Enabling GLOSA for on-street operating traffic light controllers

Thom Glastra

Enabling GLOSA for on-street operating traffic light controllers

Improve performance of dynamic decentralized traffic light controllers on sparse real measurements by enabling GLOSA

by

Thom Glastra

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday April 15, 2020 at 1:00 PM.

Student number:	4385330	
Project duration:	September 1, 2019 – April 15, 2020	
Thesis committee:	Prof. dr. ir. D. Gavrila,	TU Delft, chair
	Dr. J. Kooij,	TU Delft, supervisor
	Dr. ir. M. Wang,	TU Delft
	Ir. A. Koek,	Siemens

TU Delft

Abstract

Nowadays traffic jams have an effect on most people in daily life. The bottleneck of the maximum road volume in urban areas, and the major reason for the occurrence of congestion in traffic, is the maximum capacity of the traffic flow on the intersection. Safe passage for all vehicles on the intersection is coordinated with Traffic Light Controllers (TLCs). A promising method to decrease the number of stops are Green Light Optimal Speed Advice (GLOSA) systems. These systems will give a speed advice to arriving vehicles based on the schedule of the TLC. TLCs needs to be predictable to give reliable GLOSA messages, i.e. the schedule of TLCs needs to be known and fixed. However, most on-street controllers are flexible to maximize the performance until the last moment.

In this thesis a predictive controller is developed that is suitable for real world application. A prediction model is used to predict the future arrivals based on available measurements to optimize and fix the schedule in advance. The proposed controller can enable GLOSA systems to improve the performance. This thesis will investigate the possible performance gain by developing such a controller.

A Long Short-Term Memory (LSTM) network was designed by Van Senden [80] to predict future arrivals. Appropriate pre-processing steps are implemented and the optimal input features are selected to improve the performance with an increasing prediction horizon. All detection data is stationary over time, i.e. remove the trend and make the input feature independent over time, by using the differenced series. The time of day data is cyclical to remove the undesirable jumps during midnight. The day of week data is divided in workdays and weekend days to create a binary input. The combination of stop line detectors, queue detectors, arrival detectors and signal states of the controlled and preceding intersections maximized the performance. All input features are included in the V-log standard which is used by operating TLCs on-street. This prediction horizon of the proposed prediction model could be extended until 50 seconds without loss of accuracy compared with the prediction model of Van Senden with a prediction horizon of 30 seconds. Over the same prediction horizon of 30 seconds, the Normalized Root Mean Square Error (NRMSE) decreased with 17%.

The proposed controller is based on the self-organizing intersection controller developed by Lämmer [37] and DIRECTOR developed by Van Senden [80]. The proposed controller extends the control horizon and uses multiple prediction models to predict the arrivals for every time bin within the control horizon. The proposed controller outperforms DIRECTOR with 14 - 38% reduction in terms of vehicle delay and 5 - 32% reduction in terms of numbers of stops based on the scheduling mode. The proposed controller is not competitive compared to the hand-crafted non-predictive on-street controller. However, the proposed controller has a fixed schedule which can be used to enable GLOSA systems.

The GLOSA system is an add-on of the controller and the controller also must be able to operate without the GLOSA system. The control horizon of the proposed controller always has a fixed length which is needed to determine the time until green. The implemented GLOSA system will determine the optimal speed based on the time until green and the expected delay due to the surrounding vehicles. The proposed controller is a cloud controlled application. Therefore, it is possible to adjust the setup (i.e. scheduling modes) during the day. Enabling GLOSA all day except during rush hours will lead to 3 - 4% reduction in terms of vehicle delay and 29 - 32% reduction in terms of numbers of stops based on the scheduling mode. This setup of the proposed controller seems competitive with the hand-crafted non-predictive on-street controller. Compared with this controller, the proposed controller will reduce the number of stops with 26% at the cost of 16% increase in vehicle delay.

The proposed controller is designed conform the safety standards used on-street. Permission is received from the provincial government to do on-street pilots with the proposed controller. Due to the safety reasons, GLOSA is not (yet) enabled in these pilots. At the time of writing, the final preparations are realized to connect the controller to the on-street TLC. The pilots will be done in small test periods at a time. The results could give new insights in the performance of the proposed controller.

Preface

This master thesis is my final work to complete my Master Mechanical Engineering. I could not have done this research and write this thesis all by myself. Therefore, I would like to thank a few people by name. First of all, I would like to thank Linda Lanphen and Patrick Verspui to give me the opportunity and guidance to conduct my master thesis at Siemens. Secondly, I would like to thank Eddy Verhoeven who started this project a few years ago and lifted this project to a higher level. I am convinced that without him it would not have been possible to get the designed controller on the street.

Eddy left Siemens before I could finish my Master Thesis. Fortunately, Alexander Koek stepped in and took place in the graduation committee and gave me guidance. Together with Hans Looijen they were always available to answer all my questions relating to the content. During my time at Siemens I worked in a motivated team; the Digital Lab. This team created a good working environment to keep the pace up and getting the most out of everything. This certainly contributed to the final result. Finally, I would like to thank Julian Kooij for the guidance throughout this thesis and keep an eye on the academic value. His critical look during all meetings helped me structuring the master thesis with this report as result.

Finalizing my master thesis also means the end of my time as mechanical engineering student. The past 5 five year I had the opportunity to broaden my knowledge and I would like to thank everyone who helped me accomplish this.

Thom Glastra
Zoetermeer, April 2020

Contents

List of Figures

List of Tables

1
Introduction

1.1. Background

Nowadays traffic jams have an effect on most people in daily life. Especially in the morning and evening rush hour these jams are a big problem and can have a huge effect on society. In 1989 the costs were already estimated at $16 billion annually for the United States only [42]. A study 10 years later estimated a cost of $410 for each registered vehicle in Chicago [64]. Present-day these costs will turn out even higher. A traffic jam can appear when the number of vehicles exceeds the road volume. Traffic jams can be caused by several reasons. For example:

- The road is inadequate for the number of vehicles, i.e. the road cannot handle the amount of vehicles. This can have many causes.
- Obstacles on the road causing a blockage and merger. This will result in a lower road volume [9].
- Traffic light signals out of sync or with inadequate green time. This will also result in a lower road volume [1].

Traffic jams that are caused by the reduction of the road volume could be prevented or at least shortened in most cases. The use of automated vehicles (AVs) can bring a beneficial change to our transportation system. This innovation will result in less crashes, travel time reduction, fuel efficiency and parking benefits which can save $2000 per year per AV [21]. Congestion in traffic that arises when the road is inadequate is harder to prevent. The best options are to increase the road volume (by e.g. making an extra lane) or reducing the number of vehicles (by e.g. making an alternative route more attractive). However, some traffic jams are caused by perturbations of human drivers in dense traffic [35]. Connected (automated) vehicles could resolve or at least reduce these delays with Vehicle-to-Vehicle (V2V) communication [71].

The last cause of traffic jams is due to traffic light controllers (TLCs). The bottleneck of the maximum road volume in urban areas, and a major reason for the occurrence of congestion in traffic, is the maximum capacity of the traffic flow on an intersection. In most cases, especially on crowded intersections, this is controlled with traffic lights. The way an intersection is controlled will have a major impact on the traffic flow. Signalized intersections will not only give delay during rush hours but also during more quiet hours. When there is no coordination between TLCs of adjacent intersections, they could operate out of sync what causes non-optimal traffic flow control. Another cause of delay is that vehicles will need to wait for a red light because vehicles will come from another (conflicting) direction. If the vehicle was informed in advance, the vehicle could have reduce speed to delay the time of arrival for a few seconds.

1.1.1. Taxonomy and topology of traffic light controllers

In this section, the basics of TLCs are explained to fully understand the subject. TLCs control the traffic lights of one intersection based on the input of the intelligent transportation system (ITS). The ITS is the control algorithm that decides which traffic lights will turn green. The ITS could manage multiple TLCs. A TLC is always placed on an intersection, therefore it is also called a signalized intersection controller.

An intersection is the point where roads of different directions come together. In the Netherlands all the terminology is documented in the Intersection Topology Format (ITF) [81], which are compliant to international standards. In ITF terminology, these directions are called links or approaches and the intersection is called the node. To summarize, an intersection is a node where multiple links will come together. A link can have multiple Signal Groups (SGs) which split at the node. An example is used to make it more visual. A schematic drawing of an intersection is seen in Figure 1.1. The node is at location X and the four links of this intersection are A, B, C and D. All links have two SGs. For example, link A has SGs named 1 and 6. SG 1 directs to link C and SG 6 will direct to links B and D.

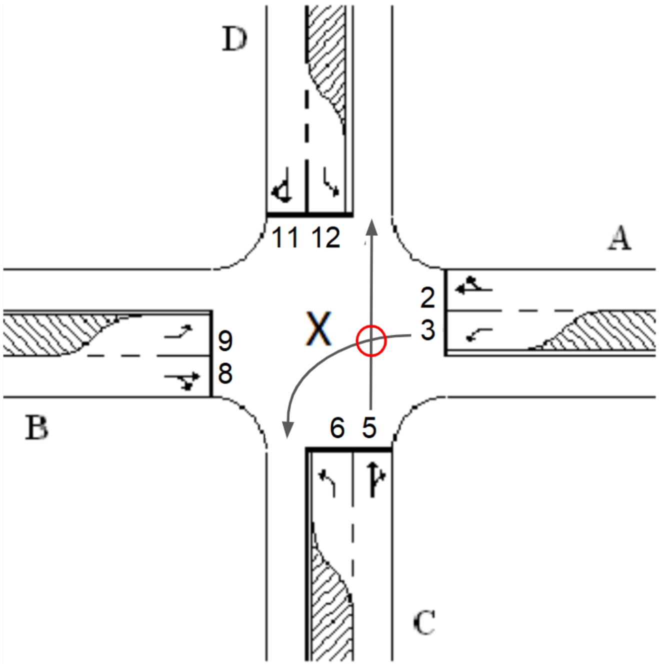


Figure 1.1: Schematic drawing of an intersection [4]

Safe passage of the intersection is important for all traffic participants: vehicles, bicycles, pedestrians and sometimes even a tram. Nowadays there are two ways of how this is organized. For the less crowded intersections this is often done with priority intersections. Some links have a higher priority and all vehicles on that link can cross the intersection first. Vehicles on the other links will need to determine if it is safe to cross the intersection. When no link has priority, the link right of the vehicle has a higher priority (and therefore the left link has a lower priority). For the opposite links, traffic straight ahead has a higher priority. The second option is to control the intersection with traffic lights. For conflicting SGs, which is when the lanes are crossing at the intersection, only one of the SGs will be granted access and vehicles from other SGs will need to wait in front of a red traffic light. The conflicting area between SG 3 and SG 5 is seen in the red circle in Figure 1.1. All SGs, which have crossing arrows with each other in the yellow area, are conflicting SGs. In most cases not all the SGs are conflicting and combined in a phase group. An intersection consist of multiple phase groups and each phase group consist of one or more SGs. An intersection will always have multiple phase groups. If there are no conflicting lanes it will not be called an intersection anymore and no control is needed if there are no conflicting SGs. Figure 1.1 is used to make this more clear. In the example all the links have two SGs. If there are no traffic lights present, link A will have priority over link C, link C over link B, etc. For opposite links, SG 8 will have priority over SG 3. If there are traffic lights, conflicting links or SGs will be granted access one after another. In this example SG 2 is conflicting with link D, link C and SG 9. This means that SG 2 could be combined with SG 3 or SG 8 in a phase group. It is important to notice that SG 2 could not be combined with both SG 3 and SG 8 because they are conflicting between themselves. The ITS application determines the optimal sequence to serve the SGs. The TLC receives this information and controls the traffic lights. The TLC will check the outputs of the ITS for safety, e.g. there is amber time between green and red, no conflicting SGs serviced at the same time or not enough clearance time between switching.

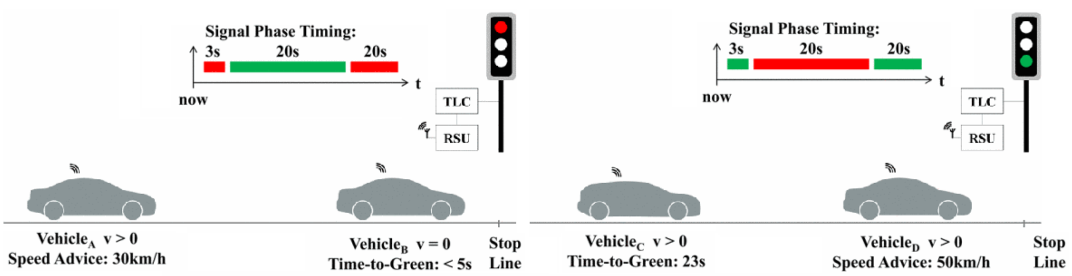
Finally, it is important to determine the performance index. The ITS is the control algorithm and has the goal to maximize the performance. The TLC will execute the control decisions based on the output of the ITS. The performance index answers the question: *how optimal is the intersection controlled?* The performance index is a combination of two Key Performance Indicators (KPIs).

- The first KPI that is widely used to determine the performance is the **total vehicle delay**. The total vehicle delay is defined as the difference between the travel time that is actually experienced by a vehicle while going across the intersection and the travel time this vehicle would have experienced in the absence of traffic signal control and other traffic [19].
- The second KPI is the **total number of stops**. This is the number of vehicles that need to make a stop (i.e. a full standstill) before they can cross the intersection.

There is not a universal agreement on the absolute importance for both KPI. The goal is to improve them both as much as possible. The KPIs are correlated: if a vehicle needs to make a stop it will also encounter travel time delay. An intersection will always have impact on the traffic flow. If one phase group will be granted a green light, the other phase groups will receive a red light at the same time. The road volume for these phase groups is temporarily set to zero. If a vehicle arrives at that moment, it will have travel delay. The objective of the ITS is to have minimal impact on the traffic flow.

1.1.2. Green light optimal speed advice

There are multiple control-methods trying to achieve the optimal control based on the available resources. Some of those control methods have a predetermined sequence of signal states and some controllers react on inputs from the vehicle detectors that measure arriving vehicles. These methods are explained in more detail in Chapter 2. Up till today, none of these methods can perfectly control the traffic. There are already solutions in development to improve the performance of the TLC and reduce the travel delay and number of stops. A promising development uses the Vehicle-to-Infrastructure (V2I) communication model. The TLC sends a message with a speed advice to the arriving (connected) vehicles. This message consists the Signal Phase and Timing (SPaT) information of the TLC [34]. Briefly, this gives information about the behavior and the future signal states (colors) of the TLC. This advice will suggest a speed for the vehicle to let the vehicle arrive during a green light period so that the vehicle does not have to stop. In the case of vehicles on conflicting SGs, one direction will get a lower speed advice so that the vehicles can cross the intersection one after another. This system is called a Green Light Optimal Speed Advisory (GLOSA) system [34] as is seen in Figure 1.2. It is possible to enable GLOSA systems when future signal states are known or there is an estimation. This feature, certainly when this information is given in an early stage, could lead to a significant performance improvement [80]. Multiple vehicles will receive GLOSA to arrive during the same green phase which leads to platoon formation. Platoons can be serviced efficiently because all these vehicles will be serviced during the same green phase for multiple intersections. An overview of the performance gain was given during the 14th International Workshop for Communication Technologies for Vehicles. The exact numbers differs a lot for each research because the result was highly dependable on the assumptions made. More realistic assumptions, in general, lead to less improvement. The waiting time reduction varied between 15% - 46% and the travel time delay decreased up to 48%. There were also studies that had no significant change in travel time delay. The saved fuel consumption due to the reduction of stops was between 14% - 25%.



(a) Scenario 1: Current phase of traffic light program is red (b) Scenario 2: Current phase of traffic light program is green

Figure 1.2: Example of a SPaT information including GLOSA. In scenario 1 the vehicle in front of the red light is already stopped but the rear vehicle receives a GLOSA of 30 km/h. In scenario 2 the front vehicle receives a GLOSA for 50 km/h. The rear vehicle is too late for the current phase and will need to wait 23 seconds. [65]

1.2. Problem statement

Unfortunately, the implementation of GLOSA systems is not as straightforward as it seems. GLOSA systems can be enabled when the time until green is known. A problem is that many TLCs change the time to green until the last moment to optimize the KPIs. If the state of the TLC can change up till the last moment it is not possible to give a reliable GLOSA message because the time to green will change accordingly. A possible way to reduce the impact of this problem is detecting arriving vehicles in an early stage. If the TLC knows the traffic flow, it can determine the optimal signal states in advance. GLOSA systems can use this information to inform the vehicles [50]. The sooner and more accurate the vehicles are detected, the sooner the correct time until green can be determined. If it turns out that the detections are not that accurate, the controller (designer) faces a dilemma: deviate from the planned states to maximize performance or stick to the planned states that were sent to the arriving vehicles with the GLOSA system. If the TLC often deviates from the planned states, the reliability of a GLOSA system will deteriorate and the system will be less effective. Therefore, it is important to maximize the quality of the early detections. These detections must be transformed to a forecast of the moment the vehicles will arrive, which is done with a prediction algorithm. However, present-day TLCs have some limitations which makes it hard to make accurate predictions. Circumstances in which present-day TLCs need to operate are:

- The locations of the arriving vehicles are not always known. Most TLCs use inductive loop detectors to detect the arriving vehicles. The vehicles can only be measured at the locations of these detectors. The exact locations of these detectors are explained in more depth in the next chapter.
- Vehicles are not controllable. There is an expected speed based on the speed limit or an advised speed when enabling GLOSA systems, but the drivers are controlling the speed. TLCs need to be able to handle vehicles that deviate from the given speed.
- Partly due to the point above, there are safety margins between switching signal states. Between switching there is an amber time first and after that an evacuation time (all lights are red).

Figure 1.3 shows an example of an intersection where a TLC must be able to function. Vehicles are only detected by the stop line, queue or arrival detectors. More information about the location of the vehicles is not available. To solve this problem the TLC needs to be predictable and flexible. The TLC needs to be flexible in order to adapt to the traffic flow and it also needs to be predictable in order to give reliable information to the arriving vehicles. The trade-off is an important design choice and effects the performance of the TLC and the GLOSA system.

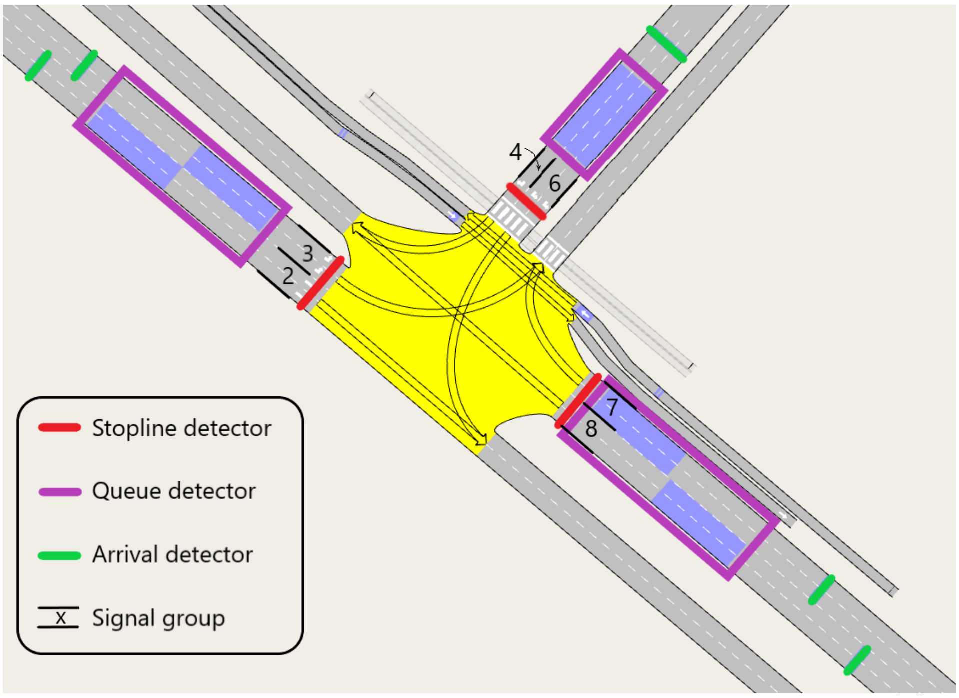


Figure 1.3: Schematic drawing of intersection located in Noord-Holland. Vehicles can only be detected by one of the detectors.

The state-of-the-art controller that is suitable for real-world application is non-predictive. The other type of state-of-the-art controller that currently exists is predictive and a GLOSA system could be enabled. However, this type of controller is not suitable for real-world application because it makes unrealistic assumptions on complete knowledge of the traffic situation. This thesis is an attempt to make the gap between these types of controllers smaller; enable reliable GLOSA systems without deterioration of the control performance but is still suitable for real-world application. The proposed controller is based on DIRECTOR, a controller that is suitable for real-world application designed by Siemens that sets the first steps in closing the gap. This controller is explained in depth in Subsection 3.2.2. DIRECTOR detects the vehicles when they leave the preceding intersection. With the use of a Neural Network (NN) the traffic flow is predicted and used to determine the signal states in advance. Simplified tests are performed to enable a GLOSA system but the performance deteriorated due to a short control horizon. Therefore, the current state of DIRECTOR is not able to operate with a GLOSA system. The problem and the associated approach to solve this is separated into multiple parts. The numbers of the list below corresponds with the numbers of the sub questions in the next section.

1. The first step to enable a GLOSA system for a TLC which is suitable for real-world application is to make it more predictable. To make early predictions of the arriving vehicles, the prediction horizon needs to be longer and more accurate than the short control horizons used before in DIRECTOR, up to 30 seconds.

2. The next step is to incorporate the prediction model into the controller. The controller will need to make decisions based on the predicted arrivals.

3. With reliable predictions it is possible to determine the time to green for all directions based on the (preliminary) time plan. This information can be used to enable a GLOSA system. GLOSA systems should not only take the desired speed to arrive during green light periods into account, but also the optimal time within these periods. When the road is crowded, the system needs to ensure that all vehicles arrive scattered during the green light period instead of all at the beginning or all at the end.

4. The performance of the proposed controllers needs to be determined for intersections as shown in 1.3. Due to safety reasons, the correct operation of a new controller needs to be demonstrated first in a simulation. A simulation setup is needed to mimic the real-world circumstances and test all proposed methods.

1.3. Research question

Currently there are state-of-the-art TLCs that improved the performance by enabling a GLOSA system. However, they make unrealistic assumptions on complete knowledge of the traffic situation and are not suitable for real-world application. A predictive controller on real-world measurements that enables a GLOSA system could potentially increase the present-day traffic flow. The objective of this thesis is to improve the data-driven NN of DIRECTOR that uses information from preceding TLCs and create a TLC that can enable a GLOSA system for real-world application. In Section 1.2 the problem is explained. This master thesis tries to solve the described problem based on a research question and design an ITS that can operate on an intersection as shown in Figure 1.3. The research question is defined as follows:

How much can the performance of dynamic decentralized traffic light controllers be improved on sparse real measurements by enabling a GLOSA system?

The research question is separated into multiple sub questions to fully and systematic solve the problem. The sub questions are based on problem statement in the previous section. The number of the sub questions corresponds to the numbers in the problem statement.

1. *Can the current prediction model in DIRECTOR achieve longer prediction horizons by optimization of the input features and the pre-processing of the input data?*
2. *How much does the predictive model improve TLC performance compared to other controllers that also work without complete knowledge of the traffic situation?*

- *How much does the performance improve compared to controllers suitable for real-world application?*
- *How much does the performance improve compared to other predictive controllers?*

3. *How much can the TLC performance be improved by enabling GLOSA systems?*
4. *What is the effect of fixing the time plan to ensure reliable GLOSA messages?*

1.4. Document Structure

Chapter 2 will give an overview on available research about TLCs and enabling GLOSA systems. This chapter will show the gaps in the literature and what this master thesis can do to close this gap and will give the basic knowledge needed to understand this master thesis. Chapter 3 will explain all the methods to design the proposed controller. This chapter is separated into five sections.

- Section 3.2 explains the baseline control approaches. DIRECTOR and CCOL are used as the baseline.
- Section 3.3 will explain the proposed method to improve the performance of the prediction model.
- Section 3.4 will incorporate this prediction model and make the structure of the controller ready to enable GLOSA.
- Section 3.5 will include the GLOSA system in the proposed controller.
- Section 3.6 will explain the simulation method to test the performance of all the baseline and proposed methods.

Chapter 4 will explain the performed experiments and show the results. This full master thesis is concluded and the results are discussed in Chapter 5. Recommendations for future work are also discussed in this chapter.

2
Related Work

This chapter will explain all the background information relating to this master thesis. This will be separated into four sections starting with general information about state-of-the-art traffic light controllers (TLCs) and ending with more sophisticated and specialized developments regarding communication between TLCs and vehicles to minimize the travel time delay and the number of stops. This chapter is based on the literature study that is written to explore the field of intelligent TLCs. This study can be seen as the introduction of this master thesis and this chapter will summarize the most important findings. The literature study can be used for a deeper understanding.

2.1. State-of-the-art traffic light controllers

This section explains the basics of traffic management for intersections. This is split up in possibilities to detect vehicles and the control methods used in a TLC. Both the currently used methods and the most recent developments are discussed.

2.1.1. Detection of the vehicles

There are different types of vehicle detectors. These detectors are separated into three categories based on the detecting capabilities [25] [70] [52].

- *Vehicle detectors that can only detect the presence of a vehicle.* The most detectors currently used on the road can only detect the presence of a vehicle at the location of the detector. The most common ones are the inductive-loop detectors [3]. When a vehicle passes the detector, the induction is temporary reduced which is denoted as a detection. If two vehicles cross the intersection the detector at the same time, only one vehicle is counted. Magnetic [7] and infrared detectors [32] have similar outputs.
- *Vehicle detectors that can detect the presence and speed of a vehicle.* Radar and ultrasonic detectors will detect vehicles based on the reflection [7]. Relatively new detectors are video cameras [49]. Cameras have a lot of possibilities but they depend on the processing techniques to extract all the information. Double inductive-loop detectors in a row can also determine the speed of vehicles. With additional software it is possible to classify the type of the vehicle.
- *Floating Car Data (FCD)* [17] is the most state-of-the-art method to detect vehicles. FCD uses the GPS data of the vehicles. This method has some huge benefits. There is no hardware required on the road. Also, the locations of all the vehicles are known at any time, not only when the vehicle crosses a detector. Sometimes the destination of the vehicle is also known which can be used to determine the associated signal group. However, there is one major drawback, this information is not available for every vehicle. Besides this, all vehicles that use this do not use the same platform. Apps like Google Maps collect this information but it is not shared publicly.

Most of the detectors named above and especially most detectors on the Dutch roads can only detect vehicles in one place. Most intersections will have multiple type of detectors with different functions. The locations are separated into three categories. The locations are shown graphically in Figure 2.1 and the locations are explained below [13].

- The stop line detector is the detector nearest to the traffic light. The detector determines if there are currently vehicles waiting for the traffic light. The TLC knows that at least one vehicle is waiting. This detector is also used to determine how many vehicles crossed the intersection.
- The downstream detector, also called the queue detector, is the detector before the stop line detector and determines if there are still vehicles present. In some cases this detector covers a long part of the road, therefore it is also sometimes called a long loop detector. If the detector is longer, the chances are higher that multiple vehicles are on the detector at the same time. Therefore, this detector is usually not used to count the number of vehicles.
- The upstream detector is the furthest away and therefore also called the far away loop or arrival detector. The other two detectors are almost always present, this detector is a bit rarer. It can be up to a few hundred meters before the traffic light. This detector will give the TLC a heads up about the arriving vehicles. This is enough time to change the traffic lights and give passage to the arriving vehicle in time (if there is no conflicting traffic).

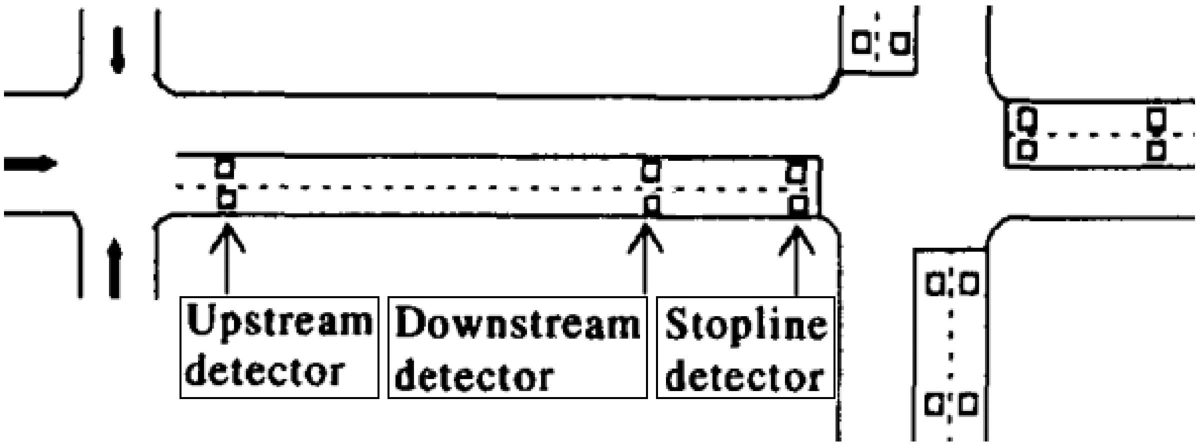


Figure 2.1: Location of vehicle detectors [13]

2.1.2. Methods for traffic control

Every TLC has the goal to minimize the delay at the intersection. There are four control methods on which a TLC can be based. These controllers are called pre-timed controllers, traffic-actuated controllers, traffic-adaptive controllers or model predictive controllers. All methods are explained below with the most important advantages and disadvantages.

Pre-timed control

The pre-timed traffic controller, also known as fixed-time controller, is the most basic traffic controller. The pre-timed controller operates in a predetermined and repeated sequence, called a time plan. Important input parameters to determine the time plan are the cycle length and the green split for every phase group [36]. The cycle length is the time until all the phase groups had at least one green signal. The green split is the fraction of the cycle length when the signal is green for that phase group. The duration that the traffic light is green is called green time. These parameters could be determined by training or testing when there is available historical data of the traffic. Pre-timed controllers operate without vehicle detectors what makes it cheaper and/or easier to purchase, install and maintain than the other types of controllers. A drawback is the lack of flexibility due to the predetermined sequence. The parameters will only be set once and this sequence is used for all traffic conditions. The traffic conditions will fluctuate over the day, especially in rush hour, and therefore the control is non-optimal most of the time. Due to changing traffic conditions, the control becomes outdated. An additional total vehicle delay of 3–5% per year is experienced as a consequence of not re-timing signals as the conditions evolve over time [45]. Besides the drawbacks concerning the control, there is also a huge benefit. Due to the repetitive sequence of the control, the future behavior of the signals is known. This information can be used to inform adjacent intersections. The controllers can use this information to adapt to each other [76]. This information can be used to inform arriving vehicles with the time until green. Vehicles can adjust their speed to arrive in time for a green signal.

Traffic-actuated control
Traffic-actuated control is seen as an improvement over pre-timed control. The actuated controller makes the schedule solely based on the active green phase. The actuated controller makes the decision to switch or extend the active green phase based on the remaining traffic on that phase group. The controller is subjected to constraints of minimum and maximum green times. This can vary for each signal phase depending on traffic flow conditions [87]. In most cases, this means extending the green time until there are no more arriving vehicles or the maximum green time has been reached. The benefit of actuated control is the possibility to adjust to the current traffic conditions extremely well. However, also this control has some drawbacks. Because the controller has no prospect of the traffic flow, it will not take future arriving vehicles into account. The consequence of this is that the performance deteriorates when there is a lot of traffic suddenly [73]. Also, the proportion of stopped vehicles is generally high.

Traffic-adaptive control
Traffic-adaptive control is once again seen as an improvement, now compared to traffic-actuated control. This controller uses the state of the entire intersection to make a control decision. Adaptive controllers can vary in semi adaptive and fully adaptive behavior [78]. While semi-adaptive controllers do not alter the order of signals but only their length, fully adaptive controllers can also change the order of signals and even leave unneeded ones out. Overall, the pros and cons are equal to the traffic-actuated control except for one extra benefit. The controller can premature switch the active green phase if there are a lot of vehicles arriving from another direction because the whole state of the intersection is taken into account. CCOL is a toolkit to design traffic-adaptive controllers that are used to design controllers for many intersections in the Netherlands [16]. Every specific version for an intersection can use different modules of the toolkit to optimize the controller for that intersection. This is an expensive design process but will result in high performance. CCOL is a widely used controller in the Netherlands and therefore this controller is used as a baseline method to test the performance of the proposed controller.

Model predictive control
Model Predictive Control (MPC) is used in many industries, including the control of traffic on intersections [72] [25]. MPC is relatively new in the traffic control industry but is seen as a promising method for the future. MPC makes an optimization over a time horizon to determine the schedule. The schedule is not fixed for the full time horizon but the effect of the schedule for the near future is taken into account. MPCs can make decisions based on information over a longer time horizon to determine the optimal time plan. However, the controller is dependent on the available information. Therefore, FCD will be highly desirable but this is currently not available on the Dutch roads. DIRECTOR is a controller designed by Siemens that uses a workaround to use MPC with the current infrastructure [27] [80] [25]. DIRECTOR includes an algorithm in the model to predict the arriving vehicles based on the available detectors. This information of knowing the number of arriving vehicles for every signal group is taken into account for the control. As already explained in the introduction, the structure of DIRECTOR is used to design the proposed controller. Therefore, the structure of DIRECTOR is explained in more depth in Section 3.2. The main benefits of MPC are the absence of controller fluctuations due to short-sighted control outputs and the vision horizon based on the time plan for the near future. In the case of DIRECTOR, the arrivals are based on the output of the prediction algorithm and the performance is therefore dependent on the quality of the predictions.

The first implementation of MPCs in traffic control are SCOOT (Split, Cycle and Offset Optimization Technique) [31] and SCATS (Sydney Coordinated Adaptive Traffic System) [63]. Both controllers are not yet model predictive, but they have some similarities. They communicate with the surrounding intersections to have a foresight of the arriving traffic. SCOOT uses a time plan for basic control. The objective of the controller is to constantly optimize this time plan according to the current traffic situation. As the name already states, SCAT is actually an adaptive controller. However, the advanced aspect of this controller is communication. This controller is installed all over the city of Sydney. There are over 1000 controllers in total that are communicating with each other. Every controller has autonomous control over its own region but there is a supervising computer that tries to optimize the total control.

Nowadays there is a lot of research related to MPCs for traffic [41]. A few promising controllers are briefly explained. The main focus of these researches are the development and integration of the predicting algorithm. They all use the same basic prediction algorithm which is explained in the next section. Below the unique aspects of the controller are highlighted. PRODYN [28] was one of the first attempts to make an MPC. Although this research was already done in 1983, it still has some advanced control ideas. Almost all MPCs still need some pre-timed control plan or parameters as input. PRODYN uses no pre-determined time plan which makes it easy and beneficial to implement the controller on different intersections without the need of determining the inputs every time. RHODES [50] divides the problem in subproblems to control in a more optimal way. The strengths of RHODES are summarized in three points. RHODES is an effective system due to its fast processing, the ability of data smoothing and the ability to respond to variations by explicitly predicting the arriving vehicles on multiple levels. One of these levels is a predicting algorithm that uses the input from vehicle detectors. Many parameters must be estimated to get reliable outputs. Another level is based on the moment such as day and time. CRONOS [12] divides the controller into three modules; forecasting, simulation and optimization module. The prediction is based on a rolling average of the arrivals in the past. The simulations investigate the optimal state for the next time step. At the next time step, it uses the new forecasting, the simulation and the optimization module to determine the final time plan. UTOPIA [47] does not optimize a single intersection but tries to optimize a large area and decompose this in a decentralized way. The goal is to minimize the total travel time and the number of stops over the whole area instead of only one intersection. This seems a desired property but it also has some drawbacks. Small changes in this area can lead to a big performance drop. Controllers like SCATS have fewer problems with this. Overall, researchers focused primarily on the prediction algorithm. In general, the prediction algorithm is the improved part of the state-of-the-art MPC. This indicates that this part has a big influence, and probably an advantage, on the performance of MPCs compared to the other traffic controllers. In the next section, the different prediction algorithms are explained.

2.2. Time horizon of traffic light controllers

MPCs include the near future in the optimization to make an optimal time plan. Subsection 2.2.1 will explain the used methods to determine the future arrivals. The added benefit of predictable TLCs is that it is possible to enable GLOSA systems. To ensure the predictability, the time plan may not change often. Due to this, the flexibility of the controller is lost which can result in a performance drop. This trade-off between flexibility and predictability is discussed in Subsection 2.2.2. The time horizon of MPC is used to make clear that the optimization includes the near future. However, MPCs have two horizons. The control horizon is the time time plan for the near future which is explained in Subsection 2.2.3. To determine this schedule, the future arrivals need to be known. The horizon of these predictions is referred to as the prediction horizon which is explained in Subsection 2.2.1.

2.2.1. Prediction algorithm

An MPC will need to know the arrivals in the near future. All future arrivals are known if FCD is available. However, this is not the case and therefore the arrivals are predicted with a prediction algorithm based on the detections of the preceding intersection. Based on these detections, the forecasting seems straightforward. Just add the seconds to determine the time of arrival for all vehicles depending on the distance and speed limit. However, this is not the case because small disruptions to the network can cause large differences to the arriving time. Things as small speed differences, vehicles that leave the road before the intersection, queued vehicles that cause delay or the different proportions that go straight, left or right at the intersection will influence the result. Especially the destination of the vehicles, which will result in an arriving vehicle for a different signal group, is hard to predict. Figure 2.2 makes the longitudinal (time until the vehicle arrives) and lateral (the direction of the vehicle) dispersion of vehicles visual.

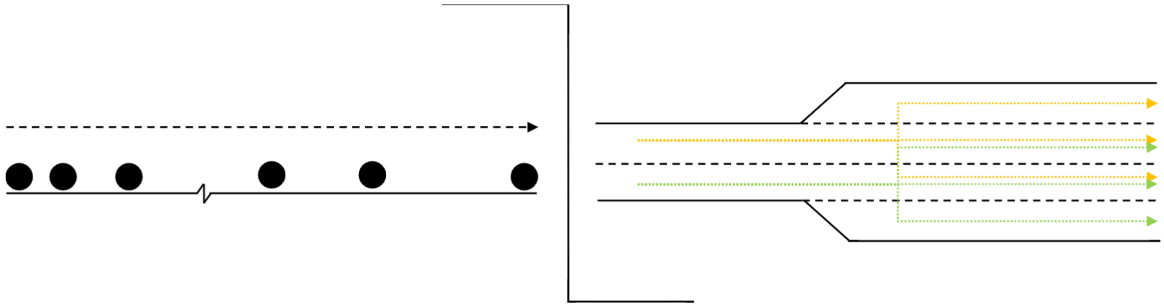


Figure 2.2: Visual representation of the longitudinal (left) and lateral (right) dispersion [27].

DIRECTOR uses a Recurrent Neural Network (RNN) to determine the arrivals as already explained in the previous section. This is a totally different approach compared to the traditional prediction algorithms. The three main analytic models for traffic flow progression are Lighthill & Witham's fluid dynamic traffic model [40], Pacey's diffusion model [55] and Robertson's platoon dispersion model [58]. Respectively, each model is a modification of the predecessor and therefore MPCs use the most recent model. The model of Lighthill & Witham's uses a fluid dynamic model to predict the arriving traffic [74] [26]. They used a continuity equation that assumed conversation of the number of vehicles. To complement this equation, the traffic flow and local speed instantaneously follow the density of the road. A higher density will lead to a slower speed. Because of this, it automatically assumes velocity is in equilibrium and thereby it will not account for stop-and-go oscillation. Pacey's diffusion model adopted a normal distribution as the distribution of vehicle speeds. Robertson's platoon model uses an empirical recursive relationship to describe the dispersion of traffic. In theory, this can give some very accurate predictions. In reality, the model needs a lot of parameter calibration to achieve this. Robertson's model has become the standard platoon dispersion model and has been incorporated in the most MPCs such as SCOOT, SCATS and TRANSYT [22].

As already mentioned above, in DIRECTOR the forecasting algorithm is based on an RNN. When such a network is set up properly, it will have some desired properties. A data-driven method has no explicit assumptions such as vehicle speed or vehicles that take an exit. Parameter calibration is also no longer necessary because the model will train itself. Most RNNs have difficulties with learning long-term time dependencies which can be a drawback [27] [25]. This is because classic RNNs have difficulties with back-propagation because gradients can vanish (tend to zero) or blow up (tend to infinity) [30]. Gradients that vanish will cause loss of time dependencies and exploded gradients will cause oscillating weights. This is mainly because error signals depend exponentially on the size of the trained weights of the RNN. An RNN that tackles this problem is Long Short-Term Memory (LSTM) network. LSTM allows constant error flow (unchanged gradients) through the states which improves the preservation of dependencies in the time series. Therefore, LSTM is a neural network that is often used for time series forecasting due to its ability to process data sequences. Therefore, DIRECTOR uses LSTM because the prediction of arriving vehicles is a time series forecasting. Due to its memory from the input, output and forget gate (sometimes called the keep gate) the ability to process sequences of data arises, as seen in Figure 2.3. The green blocks are the LSTM prediction model, the network is shown for three time steps in this figure. The gates have memory and can save dependencies in the time series [53]. The gates make decisions about what to store (from the current time bin) or read (from internal memory based on previous time bins), by opening and closing the gates. Often there is an extra gate (tanh) that creates candidate values.

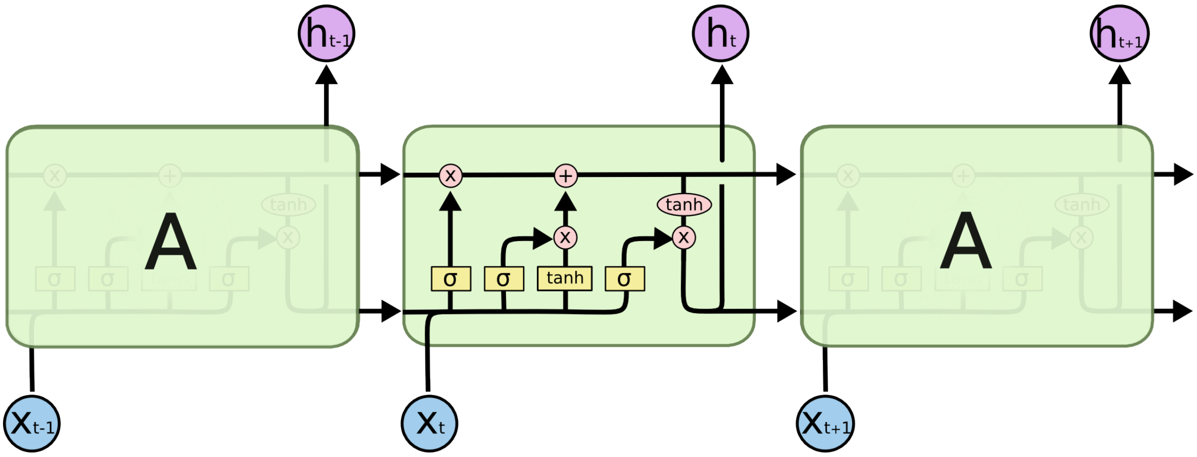


Figure 2.3: Schematic drawing of a LSTM network [54]. X_t is the input and h_t the output. The yellow blocks are respectively from left to right the forget, input, tanh and output gate.

2.2.2. Trade-off between flexibility and predictability

MPC seems to be the most promising control approach. The only drawback is that it will need information about the future arrivals, but that is not available yet. A prediction model is used to predict the arrivals for the future. As already explained in the problem statement (Section 1.2, this prediction will not always be right and the MPC will make a non-optimal time plan. The trade-off between sticking to the time plan to ensure predictability or being flexible and changing to the optimal time plan to maximize the performance is discussed in this subsection. The goal of the controller remains the same; maximizing the performance. At first sight, it seems logical to switch to the optimal time plan. However, predictability will make it possible to enable the GLOSA system which has the potential to have the highest performance in the long run.

Green Light Optimal Speed Advisory systems

Green Light Optimal Speed Advisory (GLOSA) systems are already briefly explained in the introduction. It is a system equipped by the TLC that sends speed advice to the arriving vehicles. The vehicles could, based on the given information, adjust their speed to arrive during a green phase which reduces the number of stops [11]. A deeper understanding of the system is needed to see the benefits of GLOSA systems. It is not always possible to reduce the travel delay to zero because of the scattered arrivals of the vehicles at intersections. However, if it is possible to change the scattered arrivals to more grouped arrivals, the number of stops could decrease considerably. An approach to achieve this is the use of GLOSA systems. The time until green can be shared with the arriving vehicles with a Signal Phase and Timing (SPaT) message [61]. According to this message, the vehicles can adjust their speed to arrive during a green phase. Possible benefits of the system:

- The number of stops could be decreased because the speed of the vehicle becomes a control parameter. Freedom in allowable speed is of course limited. Thereby, the advice is not always followed. This can improve when vehicles become more automated. An extra benefit is fuel reduction due to minimal use of braking [5].
- During an equal amount of green time, more vehicles cloud cross the intersection [25]. If the vehicles arrive during a green phase, they do not have to accelerate from a standstill, but they will have a flying start. Especially when one of the vehicles is a truck, this could save a lot of time.
- GLOSA systems create platoons that can cross the intersection during one green phase. If there are several TLCs behind each other, the travel time over the total trip could be reduced. In the optimal case, the first TLC collects the vehicles and informs the downstream TLCs about the arriving vehicles.

There is more needed than only the time until green to give informative GLOSA messages. It is not always possible to drive the advised speed if there are other vehicles on the road. Therefore, the GLOSA system needs to take the surrounding vehicles into account. During crowded moments, the GLOSA system should make sure that vehicles arrive scattered over the green phase. When it is extremely crowded, some vehicles may even need to be advised to arrive at the next green phase. TRANSYT is such a method that estimates the queues [59]. GLOSA is only an advice and the vehicle could still ignore this. The system will not be effective at all when most vehicles ignore the GLOSA messages. It is important to know the objective of both the TLC and arriving vehicles to determine how likely it is that vehicles will use the GLOSA system.

- The arriving vehicles will benefit from reliable GLOSA messages in time to cross the intersection without stopping. If it turns out that the advice is often correct, it is more likely that vehicles will act on following messages.
- The goal of the TLC is to maximize performance. Flexibility to change the time plan will directly lead to the highest performance [83].

At first sight, it looks like the vehicles and the TLC have different goals. However, they both have the goal to reduce the total travel time and number of stops. The only difference is that the controller tries to reduce these goals over all vehicles while vehicles on their own are more interested in their personal travel time. At first, the controller will benefit from flexibility. However, when the time plan is more predictable the vehicles could align their arrival time [10]. This ensures that the vehicles will arrive during a green light period. The TLC will also benefit from this predictability because the vehicles do not have to stop. These are the two indicators of the performance index that were explained in the introduction. The optimal controller should achieve the highest score on this index meaning there is no travel delay and no stops at all. Predictability and flexibility could help to achieve this. Predictability will help to enable GLOSA systems and flexibility will help to adapt to the current traffic flow.

In summary, enabling the GLOSA system has some clear benefits. The system will at least need to know the time until green. Fixing the time until green will ensure it will not change anymore. Flexibility is lost because of this and could deteriorate the performance of the TLC. There is a trade-off between predictability and flexibility. To what extent a TLC is predictable or flexible depends mainly on the length of the fixed control horizon.

Predictable and flexible TLCs

There is a trade-off between predictability and flexibility for every TLC. The pre-timed controller was stated as the most basic controller. However, this TLC has some desirable properties to enable GLOSA systems. For this controller, the time plan is known for the entire future because it will keep repeating the same sequence. The time plan will not change over time so it is possible to give perfect reliable GLOSA messages as far ahead as desired. Therefore, the pre-timed controller is used in most researches for testing the effect of GLOSA [5]. Traffic-actuated and traffic-adaptive controllers are the opposite. They have a lot of flexibility and can adapt to the most traffic situations. It is a lot harder to implement reliable GLOSA systems in such controllers because of their unexpected behavior. MPCs have the potential to be the golden mean. If you will only look at the implementation of GLOSA systems, the pre-timed controller will be the best option due to the known time plan. However, the primary task is not to give reliable GLOSA messages but to optimize the traffic flow. A reliable GLOSA system can help to accomplish this. MPCs have the highest potential in optimizing the traffic flow [67].

Concluded, the goal is to increase the predictability without affecting flexibility. This flexibility ensures that the controller can adjust to the current traffic demand and handle different traffic situations. In Figure 2.4 this trade-off is visualized. The fixed (pre-timed) controller has the highest predictability but no flexibility and therefore this method is not interesting for further research. Traffic-actuated and traffic-adaptive controllers are the opposite and therefore they are not optimal to work with GLOSA systems. The most promising method to still keep some flexibility but also be predictable enough to enable GLOSA systems seems to be the MPC.

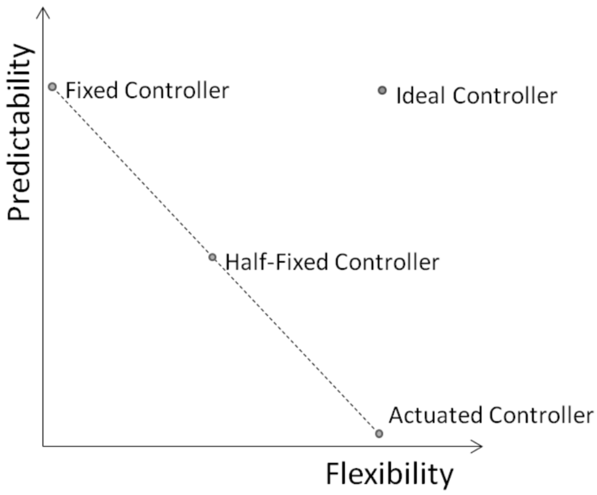


Figure 2.4: Predictability versus flexibility for the optimal TLC [27].

2.2.3. Predicted versus fixed control horizon

The length of the schedule is the control horizon of an MPC. However, it is unclear in what extend the schedule is fixed. Therefore, the time horizon is divided into the predicted and fixed control horizon. The predicted control horizon of the controller is the length of the time plan is known. It is still possible to deviate from this schedule if the predictions turns out to be incorrect. The fixed control horizon is the length of the fixed time plan. From this moment on, the schedule cannot change anymore. The trade-off between flexibility and predictability will influence these lengths. GLOSA systems will benefit from a long fixed control horizon to assure reliable speed advice. However, it is also possible that GLOSA sends the messages based on the predicted control horizon [56]. The difference is that the messages are not totally reliable. In this case, GLOSA systems can use some extra information to make it more reliable. All the options are investigated to improve the quality of the predicted control horizon. However, the only way to ensure 100% trustworthy messages is to fix the time plan which is also further investigated.

The predicted control horizon

The predicted control horizon is used by the TLC to make a time plan. The GLOSA system will determine the time until green based on this time plan. It is also possible to enable GLOSA systems when the TLC has no control horizon at all. In that case, the GLOSA system needs to make the predictions. Researches already made predictions based on FCD to enable GLOSA systems [60]. However, this is not (yet) implemented in the real world because these studies assume that this information is available for all vehicles. The other option is to use the data from inductive loop detectors. The information extracted from these sources can be divided into three categories to predict the traffic flow for GLOSA systems. Due to the diversity and uniqueness of intersections and TLCs, there are differences within a category [11]. The first category makes a prediction based on historical data [68] that were collected before the controller was implemented, sometimes even years old. This is similar to training the pre-timed controllers. The controller provides initially good predictions but over time the data becomes outdated and the quality of the predictions becomes worse.

The second category is based on previous behavior. Depending on the implementation, these data could be from weeks, days or even hours ago. For example, the predicted time plan is based on the behavior of the controller at the same time over the past days. When the traffic flow changes over a longer period, the predictions will change accordingly. However, the prediction will not include short-term fluctuations.

Both sources only use stop line detectors as input, which are present at an intersection most of the time. If there also are upstream detectors available (from a preceding intersection), the last source could be used. A prediction algorithm uses the information from the upstream detectors to make a forecast. This method is comparable with the prediction algorithm of MPCs. The advantage of this method is that it uses the most actual information; the detected vehicles will generally cross the intersection in a few moments. Both the TLC and the GLOSA system will use the same information. The drawback is that the information is not always correct due to changes that happen between the upstream detector and the intersection. Factors that most influence the arrivals are [27]:

- Vehicles that show up from side roads without vehicle detectors, also called sources.
- Vehicles that leave to side roads without vehicle detectors, also called sinks.
- Vehicles that not drive the expected speed, i.e. the speed limit.
- The uncertainty about which direction they take at the intersection.

As mentioned in the previous subsection, it is desired to make the controller predictable but also keep the flexibility. The location of the upstream detectors affects the predictability. If the detector is placed further away, it gives an earlier but less trustworthy prediction and vice versa. Especially when the detectors are placed far away, the behavior of adjacent intersections will become more important. It is also important that every link to the intersection has detectors, including the links for pedestrians or bicycles. The performance of the controller could decrease significantly when the information is incomplete.

All the approaches and factors mentioned above will influence the quality and length of the predicted control horizon. In general, the quality decreases when the length increases. As already mentioned earlier, almost all related research assumed pre-timed traffic light programs. Research has been done to increase both the length and the quality of the predicted control horizon with actuated, adaptive or predictive traffic controllers. One research combined historical data, previous behavior (last hour) and upstream traffic information [29]. This method improved the quality for peak hours, in the other cases only the use of historical data had the same result. An important side note is that they did not use intersections, only a uni-direction freeway. When the prediction is based on historical data, it is usually not used by the TLC. In that case, the TLC will make a time plan based on the vehicle detectors. Both the predicting and controlling module use their own method which can cause a decrease in quality due to incoherent behavior. When the complexity increases, it becomes harder to use the prediction algorithm in real-time. To keep the complexity acceptable, a lot of parameters have to be fixed to do a simulation of an urban traffic network with multiple intersections [41]. When combining historical data and the upstream detectors, it is even possible to make predictions for a network of fully actuated traffic lights [11]. The reliability is over 80% for a predicted control horizon of 15 seconds with accurate detectors. It is also noted that the quality of the predictions increases when the amount of available historical data increases. The most advanced control and predicting strategies, including the ones mentioned above, are often not (yet) implemented. The main reason for this is the operational complexity and the need for extra components such as reliable and multiple upstream vehicle detectors and historical data. FCD, which contains the location and sometimes even the destination of the arriving vehicles, could boost the prediction abilities [24].

The fixed control horizon

In general, both the TLC and the GLOSA system like to know the traffic flow as early and accurately as possible. This is not feasible in practice. There are limited vehicle detectors, limited computational power and uncertainties in the predictions. The TLC will benefit if changes to the time plan is allowed as long as possible, because the information is not perfect at an early stage. Contrary, the GLOSA system likes to fix the time plan in an early stage such that there is enough time to reduce the speed of vehicles to arrive during a green phase that cannot change anymore.

Before giving some insights into acceptable lengths of the fixed control horizon, it is important to know how long it takes to switch between phase groups. The safety margins from the intersection shown in Figure 1.3 are used. Before the next phase group can switch to a green phase, there is first a yellow time for 3 - 5 seconds and clearance time of 2 - 5 seconds follows [48]. In the worst case for, it will take 10 seconds to give another phase group a green light. To switch to a new phase group, there is a minimum green time which differs from 4 to 15 seconds based on the size of the intersection. This interval can be extended when needed. So, the total duration to give passage to one phase group is the sum of three parts. This is at least 9 - 25 seconds in total. When it is known how long it will take to switch between phase groups, the desired fixed control horizon could be determined. The preferred fixed control horizon is both determined as seen from the TLC and the GLOSA system.

Optimal fixed control horizon for the TLC

Fixing the time plan will give limitations to the TLC and is therefore undesired. This is true in general.

However, when fixing the time plan GLOSA messages will be more reliable. Arriving vehicles will react better and it is possible to make a more optimal situation. Beforehand, all the vehicles from different directions arrive at the same time. Due to the GLOSA messages, they will arrive one after another. If predictions are made over a 10 seconds time horizon, the signal-timing logic can only make timing decisions that extend or shorten the current phase. If the predictions are made over a longer time horizon, the signal-timing decisions can include decisions on phase sequencing and phase duration [50].

Currently, most TLCs in the Netherlands have no fixed time plan at all but only a predicted control horizon. Predictions are made but the time plan is fixed at the last moment. In this case, the maximum fixed control horizon is the duration to give passage to one phase group. All GLOSA messages that are sent earlier than that, are based on predictions. As seen earlier, it will take up to 10 seconds to switch to the next phase group. It can take up to 25 seconds to change this again because of the minimum green time. Most MPCs work with a fixed control horizon between the time to switch to the next phase (5 – 10 seconds) or the total time duration to give passage to one phase group (9 – 25 seconds). DIRECTOR is a ITS application where the control horizon can be fixed to ensure it is always at least 10 seconds long, the bare minimum to make decisions about phase sequencing and phase duration [80].

Optimal fixed control horizon for the GLOSA system

Vehicles never have to stop before a red light in the optimal scenario. Vehicles need to safely reduce the speed to arrive on time after the GLOSA message is received. This is based on a few parameters:

- Minimum allowable speed
- Minimum green time
- Yellow time
- Clearance time
- Number of phase groups

A combination of the parameters mentioned above will give a desirable fixed control horizon. All parameters are already explained above except the minimum allowable speed. Due to safety reasons it is not desired to make the minimum allowable speed to low if other vehicles still could drive the maximum speed. Besides, chances are higher that the vehicles will reduce speed when the advised speed is not too low. On the other side, if the advised speed may only differ a bit from the maximum speed, a longer horizon is needed. If there is more freedom, the time of arrival could be changed in a shorter period. One extra circumstance is also important: the surrounding vehicles. If there are many surrounding vehicles, especially with congestion, the result will be influenced. Below there is an example given of how the parameters above could be translated to the desired fixed control horizon.

In the worst case, vehicles will arrive with the same expected time of arrival from all the phase groups. All signal groups can often be serviced within 2 to 4 phase groups. There is always one phase group that will be the last to get a green phase. Again in the worst case, this group needs to delay the arrival time so that the other three phase groups can pass. As seen at the beginning of this subsection, one phase group needs 9 - 25 seconds. So, three phase groups will take up to 27 - 75 seconds. The last vehicle needs to delay the time of arrival that much. If we allow a minimum speed that is half of the maximum speed, the vehicle needs double that time to change the time of arrival. So the GLOSA message must be sent 54 - 150 seconds in advance before the last vehicle crosses the intersection. An important note, most adjacent TLCs are within this time frame which can be a cause to stop a vehicle earlier.

To give a perfect reliable GLOSA message in the example above, the time plan needs to be fixed early. Fixing the time plan early will cause problems for the TLC to keep the performance equally high. An option is to give GLOSA messages based on a time plan that is not fixed. Because of this, the GLOSA message will contain errors if the time plan changes. However, it is still preferred that the TLC will increase the fixed control horizon. The challenge is to do this without making the performance of the controller worse. A longer and more accurate prediction horizon will help to achieve this. Most TLCs use vehicle detectors to determine the time plan. To increase the prediction horizon of the TLC, predictions can be based on the detectors of the upstream TLC. Several TLCs can be used to further increase the length. However, a prediction horizon that is too long will increase the complexity and the predictions will become more unreliable. The possibilities to communicate with TLCs of adjacent intersections are discussed below.

2.3. Communication between traffic light controllers

An important requirement to enable GLOSA is to have insights in the future arrivals. FCD is only very limited available and that is why there is often a prediction model used to determine the arrivals. Sharing information between TLCs can help the prediction model. Another benefit is that TLCs can align their time plans to minimize the travel time over multiple intersections. Potential benefits and used methods are explained in Subsection 2.3.1. The communication between TLCs can go further than just the adjacent TLC but could be extended to all nearby intersections to make a network of controllers. A possible hazard is that using a network of controllers has consequences for the complexity. For networks, especially large networks, the authority of controllers becomes important. The control could be centralized by an overarching controller or remain at the intersection with decentralized control. The pros and cons for both control approaches are discussed in Subsection 2.3.2 and 2.3.3

2.3.1. Network of controllers

Every TLC tries, in its own way, to minimize the travel delay for vehicles that are crossing the intersection. However, the travel time over the total trip is far more important for the (drivers of the) vehicles. Therefore, it seems logical to optimize the total travel time over all TLCs. A short stop at the first TLC is desired if it will prevent a long stop at the next TLC. To regulate this, there must be communication between TLCs to share their (time) plans. The goal of all the TLCs together could be described as to minimize the total travel time and number of stops, preferably including everything that takes time during the trip. Communication between TLCs about the time plans is seen in Figure 2.5. The information collected by the second TLC can be used to determine the optimal time plan for all downstream TLCs [86]. Comparably, the first TLC can use the traffic flow of the upstream TLCs to prevent traffic jams by sending vehicles to busy roads.

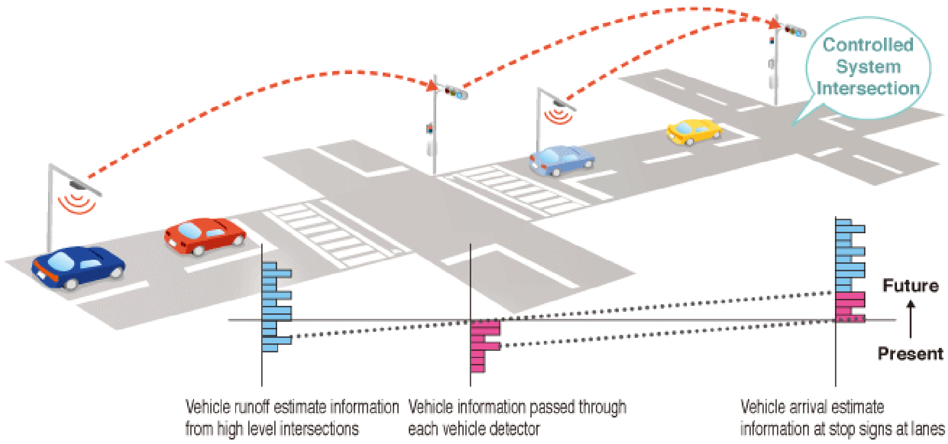


Figure 2.5: Increasing the time horizon due to communication between TLCs of adjacent intersections [69].

A good example that is already implemented on some roads, is the green-wave model [51]. In general, this initiative is implemented on relative busy main roads with side roads that have significant less traffic. All the TLCs on the main road are linked to each other and once in a while the assignment is given to initiate a green wave for the main road. Signs are placed on the side of the road with the advisory speed. When driving the given speed, all linked intersections will give a green phase consecutively. This can be seen as the first concept of GLOSA messages. The green-wave model has some drawbacks. For high traffic density, vehicles entering a green wave will be stopped by vehicles ahead of them or vehicles that enter the road. Due to the delay the vehicle misses the green wave and it will have to wait for the next green phase. On the other hand, for low densities, vehicles might arrive too quickly at the next intersection, having to stop at each crossing. The system also does not consider the traffic flow on the side roads. It is possible that the green-wave is initiated while there are many vehicles on the side road. The green-wave model is certainly better than having no synchronization at all, however, it can be greatly improved [14]. The cause is that there is no feedback implemented. It is assumed that the vehicles drive the given speed and there are no in-between measurements.

GLOSA systems can also be used to give a speed advice through a network of TLCs. Such systems will make a possible trajectory to cross multiple intersections. This may require accelerating of braking in-between but will lead to a fast and fuel efficient trip at the end. This setup is already tested with the assumption that the vehicles drive in platoons and during these tests it was noticed that correctly identifying the lead vehicle was very important for the result [66].

Again, we will look back to the pre-timed controller. Despite we already stated multiple times that MPC is the most promising TLC, the pre-timed controller has some advantages. Pre-timed controllers can be optimized in advance such that they align as perfect as possible during operation. Because the sequence is constantly repeating, this aligned control will remain. The advantage is that the TLCs will work together in an optimal scenario without increasing the complexity [6]. A side note is that when using a network of pre-timed controllers, you can doubt if there is communication. Indeed, after implementation, there is no communication anymore. However, during initialization the pre-timed controllers are optimized to work together. Model predictive controllers still have the most potential to achieve the highest performance in adapting to and optimizing in the current traffic flow. Therefore, the focus is still on these kind of dynamic TLCs.

2.3.2. Centralized traffic control

When the control is centralized, the control of a network of signalized intersections is combined in one place with one overarching controller [57]. In general, this controller has the authority to make control decisions or at least a part of the control. The size of these networks can vary, SCATS is one of the biggest networks and controls over 1000 intersections in Sydney [63]. The overarching controller focuses on the high-level control. There are still controllers at each intersection for the low-level control to assure safety. For such large networks this remains necessary to limit the complexity. When all the control is located at the signalized intersection, the control is decentralized.

Centralized control in general has a huge complexity because it needs to control multiple intersections. When there are separate low-level controllers at each signalized intersection, the complexity is reduced a bit. The advantage of centralized control is that all the available information is gathered in one place. In theory, this can give the highest performance [79]. When using overarching controllers, the biggest challenge is to limit the complexity. The problem of computational limitation raises, especially with MPCs due to the time horizon at every intersection. Along with the fast development of the transportation infrastructures, traffic networks becomes larger and the complexity increases. A possible solution is to use subnetworks [41]. The overarching controller controls multiple subnetwork controllers. These controllers have the control over multiple intersections which gives a better modular structure. This reduces the complexity and the network is also more resistant to changes in the network; only a subnetwork controller is affected. This is also called hierarchical controller coordination.

2.3.3. Decentralized traffic control

With decentralized traffic control the authority of control decisions will remain at the controller of the intersection [57]. Information of upcoming traffic and (a predicted) time plan can be shared with adjacent intersections. The main advantage of decentralized control is that the complexity of the control problem reduces significantly compared to centralized control. However, the decrease of complexity comes at a cost [79]. The controllers only have the shared information of the adjacent controllers and therefore there is a lack of information. The controller cannot make a perfect prediction of all the upcoming traffic and the optimization is not perfect. However, the control is not perfect in both centralized and decentralized controllers because the available information from vehicle detectors is not perfect and complete in most cases.

The use of FCD could help to improve the vehicle detections [17]. Unfortunately, this is not (yet) publicly available. The simplest approach in decentralized control is to only share the information of the vehicle detectors [27]. The TLC uses the information from the vehicle detectors that are located before the upstream intersection in this case. Because the upstream TLC also uses these vehicle detectors (of course), the information of these detectors can be seen as the most basic variant of information sharing. The approach is used among others by DIRECTOR to increase the time horizon. Because the time plan of the upstream controller is not shared, the vehicles can make a long stop (when waiting for a red light) what makes it harder to predict the expected time of arrival.

Zhou et al. [88] used all the information of the surrounding intersections to optimize the control. No pre-timed sequences was used, so the TLC remains adaptive to the traffic. However, only one intersection was optimized, the other intersections were only seen as supply roads and for provision of information. The travel delay was reduced with 75% and the number of stops halved. Unfortunately, the total travel time of one controller was optimized, not the total trip time.

Ferreira et al. [23] controls a network of intersections in a decentralized way by giving 'opinions'. Before every TLC determines the state for the next phase, they share some information about the traffic flow and use this to make an optimal decision. This method reduces the complexity. A drawback is that not every controller has all the information. For example, the information about the states of some controllers or feedback about their control decision is now missing. The intersections do not make an optimal time plan together, but they still do this separately. This method improved the performance up to 20%.

A similar approach is the multi-agent based method [84]. The total optimization problem is divided in small subproblems for each intersection. Each agent tries to find its local optimum with the information given by the adjacent agents.

2.4. Cooperative Vehicle Intersection Control

The goal of this master thesis is to improve the performance by enabling GLOSA for TLCs that can operate in real-world circumstances. GLOSA is a system to communicate with the arriving vehicles with the goal to improve the performance. GLOSA is a part of Cooperative Vehicle Intersection Control (CVIC). CVIC is the communication between vehicles themselves (V2V) and the communication between vehicles and the TLC (V2I) to improve the performance. GLOSA systems influence the arrival time of the upcoming traffic which can result in a more optimal scenario where all vehicles from the same signal group (in platoon formation) arrive one after another instead of all at once. The earlier a (reliable) speed advice is given, the more freedom there is to change the time of arrival [85].

GLOSA systems, as the name already says, only give an advice and vehicles have the option to ignore this advice. With the rise of Automated Vehicles (AVs) it becomes more plausible that the advice is given to the vehicle itself [2]. AVs will follow the advice while drivers tend to keep the maximum speed. A vision of the future is that vehicles are controlled by the TLC. Vehicles will not get an advised speed, but will be controlled by the intersection to arrive precisely at the desired time. The total travel times could be reduced up to 33% and the sum of the standstill times could be reduced with 99% if vehicles become a control parameter[39].

As stated in the introduction, the performance of the TLC is defined as a combination of the total travel time and the number of stops. A lot of studies that investigate GLOSA systems are focused to decrease the number of stops and hereby decrease the fuel consumption. In those studies, it was not assumed that the controller could be influenced, but it was seen as a pre-timed controller. However, when the TLC and the arriving (connected) vehicles could work together to achieve an optimal scenario, both the total travel time and the number of stops could be significantly reduced [38].

All the studies can be split up in two main approaches to control the arriving vehicles. The first approach tries to get the optimal control plan without taking the current traffic rules into account. Of course, the requirements to give a safe passage to vehicles is always preserved. A lot of studies named this as the reservation-based approach, which is discussed in Subsection 2.4.1. The second approach takes all the current traffic rules into account and could therefore be implemented right away, this is discussed in Subsection 2.4.2.

2.4.1. Reservation-based approach

The reservation-based approach is the most promising but also the most futuristic method. Although it is still far away from implementation, many studies are related to this optimal control of the signalized intersection. This approach assumes that all vehicles are connected and uses this advantages for optimal control [20]. It will give safe passage to all vehicles but will not include all the traffic regulations that apply present-day. In the future, traffic regulations could be adjusted if this method is proved to be safe in all situations. The differences are summarized below, individual studies may differ slightly.

- Traffic lights are absent. Vehicles from all directions can cross the intersection shortly after each other. The gap needs to be just large enough to guarantee safe passage. The gap is 'reserved' by one vehicle so that this vehicle can cross the intersection safely in the reserved time bin.
- Safety margins become smaller. The mandatory amber phase will be eliminated and vehicles from different phase groups can cross the intersection shortly after each other. This is needed, otherwise it is not possible to let vehicles from all directions cross the intersection shortly after

each other. An amber phase between every vehicle will again lead to a control plan where only vehicles from one phase group will cross the intersection. Of course, a (small) safety margin will remain.
- All vehicles are connected. They communicate to get a moment for safe crossing, a 'reservation'. The connection must be fast enough to avoid dangerous situations to occur.

This approach will lead to a significant increase of the complexity [18]. Besides the control of the traffic lights, all the vehicles must be controlled individually. A lot of studies are related to reduce this complexity such that it becomes possible to control in real-time. When all vehicles can cross the intersection shortly after each other, the safety margins are small and therefore the complexity increases. This can be handled with a hierarchical framework split up in four parts [8].

- Higher-level controllers for network-wide coordination.
- Roadside controllers that control platoons on highways.
- Platoon controllers use the commands of the roadside controllers to inform vehicles inside the platoon.
- Vehicle controllers receive these commands and translate this into execution.

Another option to reduce the complexity is the assumption that at each time step, each vehicle considers that the situation will not change for all vehicles but itself [46]. In this case the vehicles communicate with each other to get safe passage over the intersection. To assure collision avoidance, some extra constraints are added.

For the approach mentioned above, the system will not work when there is only one vehicle that is not connected. This is also the main reason why it will be difficult to implement this approach in the near-future. The other hard challenge is to determine the optimal speed when the time until green is known. This applies for both control approaches. Especially for the reservation based system, good communication is extremely important. To determine the optimal speed, multiple aspects can be taken into consideration, for example surrounding vehicles with their own speed, to reach the intersection in a fuel efficient way or minimize the CO_2 emission.

Many studies are related to investigate this optimal speed [43] [77]. The important requirement is that the time plan of the TLC needs to be known. Otherwise, systems such as GLOSA will not be equally efficient. The pre-timed controller is again a good TLC to test the effects of speed advice [68]. For both control approaches the effect of speed advice on the performance of the TLC is tested in many studies. In simplified tests the reservation-based approach outperformed the approach with the current traffic regulations 200 to 300 times in terms of travel delay [20]! The biggest challenge is to implement such a system with human drivers. To reach the above performance the margins for error in the system are too small for humans to control the vehicle. The margins of error could be enlarged, but the efficiency benefits come directly from the precise control of the computer.

2.4.2. Retaining the traffic lights

The approach in the subsection above has many advantages and seems to be the future, however it will not be the near future. If you want to improve TLCs in the near future, vehicles cannot cross the intersection so quickly after each other due limitations in V2I and V2V communication. Also the presence of human drivers makes this impossible. When the safety margins are expanded, it becomes again more attractive to only let one phase group cross the intersection again. Especially when not all vehicles are connected, this is still the best way to go. However, for this approach there are still a lot of improvements possible that are tested in many studies. Below, the constraints of the current control approach are summarized again.

- Between switching phase groups, there is "amber time" and evacuation time, also called yellow and red clearance duration [48]. Switching too much will lead to inefficient control and the scenario from the previous subsection is not optimal anymore.

- Vehicles are not (all) perfectly controllable, in the best case you can give a speed advice.

- With this approach only one phase group can cross the intersection at the same time. This will lead to longer waiting times. To prevent this, vehicles needs to be advised in an early stage to arrive during a green phase. This can be done with GLOSA systems. Therefore, it is important to detect the arriving vehicles in an early stage.

Currently, there are a lot of different signalized intersection configurations in the real world. Some TLCs have a lot of detectors, some none. However, a lot of information is desired to use the GLOSA system. The studies that test this approach use some simplifications in general [82]. There is a lack of studies that test the more complicated situations. DIRECTOR is a traffic management system that is designed to operate with the current traffic regulations. For the time being, no cooperative control has been implemented, only some simple tests are done. However, the vision of DIRECTOR is to detect the arriving vehicles in an early stage to make a time plan. This step is shifted forward to give the vehicles a speed advice in time. In a first approach they shifted the fixed control horizon 10 - 20 seconds in the future. The first tests proved that it is desirable to expand this horizon to utilize the GLOSA system and reduce the number of stops significant. The implementation of GLOSA in the first tests assumed that there were no vehicles queued and that acceleration and deceleration is instant [80]. Hence, it is expected that the final result with a more accurate GLOSA implementation will be better, reducing the travel time and number of stops.

2.5. Contributions

The state-of-the-art TLCs described above can be separated into two main groups. The first type of TLC can operate in real world circumstances but is non-predictive. In the Netherlands CCOL is a widely used controller. The other type of TLC that currently exists is predictive and the GLOSA system could be enabled. However, this type of TLC cannot operate in real-world circumstances because it makes unrealistic assumptions on complete knowledge of the traffic situation.

This thesis is an attempt to make the gap between these types of TLCs smaller; enable reliable GLOSA without deterioration of the control performance but still able to operate in real-world circumstances. DIRECTOR is designed by Siemens that set the first steps in closing the gap. DIRECTOR can operate in real-world circumstances, has predictive abilities and some simplified tests are performed to enable GLOSA. However, this controller is not yet tested in fully realistic circumstances. This controller is explained in depth in Subsection 3.2.2. These three (types of) controllers can be seen as the baseline for the proposed controller. It is not possible to test the fully predictable controller in simulations that mimics on-street circumstances because it makes unrealistic assumptions on complete knowledge of the traffic situation. Therefore, DIRECTOR and CCOL are only used as baseline methods.

The contributions of this thesis are explained in Chapter 3 and can be summarized in a few bullet points:

- DIRECTORs Neural Network is extended with new features to increase the time horizon of the predictive neural network based on the available data from preceding traffic light controllers. Appropriate pre-processing are implemented to maximize the performance.
- An improved design of DIRECTOR is developed which makes an optimal time plan for the full length of the prediction horizon to enable GLOSA, providing the first real-world applicable traffic light controller with a GLOSA system incorporated.
- A novel simulation setup is designed to mimic the real-world circumstances which is more realistic and is able to test the effects of GLOSA.

3

Methods

This chapter explains the methods used in the proposed controller to improve the performance of a Traffic Light Controller (TLC) by enabling Green Light Optimal Speed Advisory (GLOSA) systems. Symbols used to describe the topology of an intersection is given in Section 3.1. The baseline controllers are explained in more detail in Section 3.2. Especially a deeper understanding of DIRECTOR is needed because the structure of the proposed controller is based on the structure of DIRECTOR. Thereafter, the modifications to design the proposed controller are separated into three steps.

- Section 3.3 investigates the possible pre-processing steps and available inputs to maximize the performance of the prediction model. This improved model could be used to increase the prediction horizon.

- Section 3.4 shows the incorporation of the predicted arrivals from the prediction model in the TLC to make an optimal time plan. The structure of the proposed controller is designed to make it ready to enable GLOSA systems. This controller is referred to as P-HOR.

- Section 3.5 shows the implementation of the GLOSA system in the controller. The controller with GLOSA enabled is referred to as P-HOR+G.

The goal is that P-HOR obtains higher prediction and control horizons and improve performance. Besides, the structure is designed to make a predictable controller to enable GLOSA systems which has the ability to further improve the performance. The modifications are tested in a simulation. Section 3.6 will explain the method used to test the baseline and proposed controller, both P-HOR and P-HOR+G, in a simulation setup.

3.1. Symbol definition of the topology

The introduction already explained the basics of the topology and used taxonomy for intersection control. This section will elaborate on this and give an overview of the used corresponding mathematical symbols to explain the baseline and proposed method. All notations are denoted in Table 3.1 and explained below with an example.

Name	Variable	Set	Explanation
Controlled intersection	c	C	C is the set of all controlled intersections which is one for decentralized controllers.
Preceding intersection	i	I	I is the set of all supplying intersections.
Link or approach	a	A_c	A_c is the set of all links connecting to intersection c
Phase Group (PG)	pg	PG_c	pg is a set of non-conflicting SGs, PG_c is the set of all PGs of intersection c
Signal group (SG)	od	OD_c or OD_a	OD_c is the set of all SGs of intersection c, OD_a is the set of all SGs of link a
Lane	l_{od}	L_{od}	L_{od} is the set of all lanes of SG od
Arrival detector	d^f	D^f_{od}	D^f_{od} is the set of far away detectors of SG od
Queue detector	d^q	D^q_{od}	D^Q_{od} is the set of long detectors of SG od
Stop line detector	d^s	D^s_{od}	D^S_{od} is the set of close-by detectors of SG od
Detection	$x^z_d(t)$	$-$	$x^z_d(t)$ is a detection at time t with z the type of detector (f,q,s)
Signal state	$s_{od}(t)$	S	$S = [GREEN, AMBER, RED]$, for all SGs. $s_{od}(t)$ is the signal state of SG od at time t

Table 3.1: Naming convention of all components of an intersection.

Figure 1.3 in Chapter 1 shows the topology of an intersection. The topology of this intersection is used as an example to explain the mathematical notation. This controlled intersection is referred to as intersection c. To show some extra detail, the right side of Figure 3.1 zoomed in on this intersection. On the left side, the preceding intersection, referred to as intersection i, is included to show the entire link. The proposed method is a decentralized traffic controller and therefore $|C| = 1$, opposed to centralized traffic control methods which control multiple intersections. Intersection c has 3 preceding intersections that are connected with a link to the controlled intersection. The link shown in Figure 3.1, between intersection i and c, is referred to as link a. The size of I and A_c is often equal, except when a link egresses on a residential area without a preceding intersection. On the other hand, each preceding intersection is connected via a link, i.e. $|A_c| \geq |I|$.

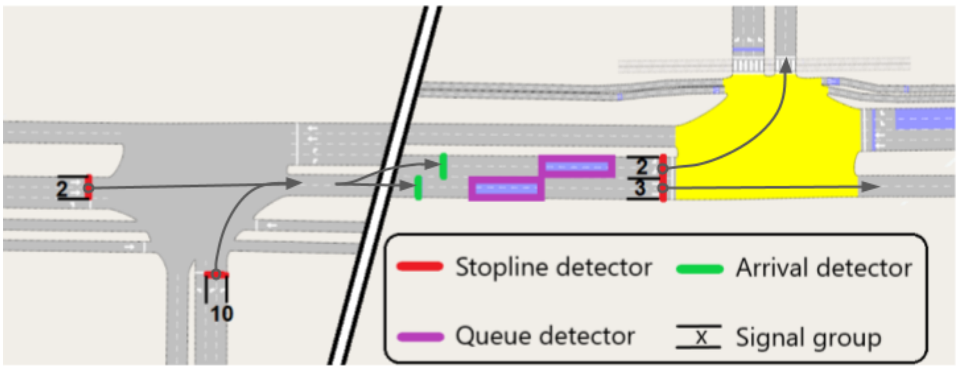


Figure 3.1: On the left side, the preceding intersection i and on the right side the controlled intersection c which is highlighted in yellow. The intersections are connected with link a. The ingress of this link is via SG 2 or SG 10 from intersection i. The egress of this link is via SG 2 for vehicles turning left or via SG 3 for vehicles that go straight ahead.

The topology of link a of between intersection i and c as seen in Figure 3.1 can now be explained in the mathematical symbols as described in Table 3.1.

- OD_a is always based on the SGs of the controlled intersection. On intersection c, vehicles can turn left following SG 2 or go straight ahead following SG 3. The set of SGs of link a is denoted as $OD_a = [2, 3]$.
- Each SG has one arrival detector covering all lanes of that SG, $|D^f_2| = |D^f_3| = 1$.

- Both SG 2 and SG 3 have two lanes, for both SGs applies $|L_{od}| = 2$.
- Each lane has a stop line and queue detector, $|D_{od}^{s}| = |D_{od}^{q}| = 2$.
- All detectors are inductive loop detectors that detects conductive objects such as vehicles, as explained in Section 2.1. When a vehicle arrives at detector d at time t, the induction is temporary reduced. At the start of this reduction the detector sends a one to the TLC, $x_{d}^{z}(t) = 1$ with z the type of detector. The TLC sends a zero at the end of the reduction, $x_{d}^{z}(t + \Delta t) = 0$. Inductive loop detector cannot distinguish individual vehicles. If the presence of vehicles overlap, it is detected as one vehicle with Δt longer than usual. The same applies for vehicles with a lower speed than usual.

The goal is to minimize the number of stops and total vehicle delay for intersection c by selecting the optimal sequence of signal states based on the detections. This is done by selecting the best phase group. The signal states of the selected phase group are switched to green and all other signal states to red. A phase group is a set of signal groups that are non-conflicting. In Figure 3.1, SG 2 and SG 3 can be combined in a phase group. But it is also possible to combine SG 3 with both signal groups on the opposite side of the intersection. All signal groups from one link are always a phase group, but there are also combinations possible that uses multiple links, i.e. $A_c \subseteq PG_c$. The selected phase group at time t should be $\chi(t)$ which maximize the overall performance as given by

$$\chi(t) = \underset{pg \in PG_c}{\arg\max}\left(f(x_{d}^{z}(t), \ldots) \right) \forall d, z, \tag{3.1}$$

where z is the type of detector and f the score function. It is not always possible to use the Key Performance Indicators (KPIs), total vehicle delay and total number of stops, as score function because not all information needed to determine the KPIs is available for TLCs used for real-world operation. This has two causes:

- Most controllers only have inductive loop detectors. These detectors give too little information to determine the KPIs.
- The consequence of a schedule, in terms of vehicle delay and number of stops, will only influences future detections which are still unknown when the schedule is determined.

Controllers determine the best schedule based on the available information from now and the past. Most controllers determines the schedule solely based on the detections as input. However, some state-of-the-art controllers can use extra inputs. Despite the fact that the score function will not directly maximize the KPIs, it will maximize a correlated function that is possible to determine based on the available information. This function is different for each controller and cannot be written down in a general form.

3.2. Baseline methods for traffic control

In this section two control approaches are explained that will act as the baseline to test the performance of the proposed controller compared to current state-of-the-art controllers. The first control method is based on CCOL, this is a toolkit to design vehicle actuated controllers. CCOL controllers are currently one of the most used on-street controllers in the Netherlands [16]. The second controller is DIRECTOR, the model predictive controller designed by Siemens. Both methods are already briefly explained in Chapter 2, in this section both baseline methods are explained in more detail for a deeper understanding.

3.2.1. Baseline method - CCOL

CCOL is a toolkit to design controllers for a specific intersection. A CCOL controller is hand-crafted for every intersection and therefore also different for every intersection. It takes a lot of effort to design and fine tune these controllers. The performance of these kind of controllers deteriorates due to the changing traffic flow. Therefore, these controllers are tuned every 2-3 year. The CCOL controller used as a baseline method is the hand-crafted version for the intersection shown in Figure 1.3 in the introduction. This version was tuned in 2017 for the last time. The used cost function to determine the optimal schedule is not exactly known because every controller is tuned differently. The intuition behind CCOL is summarized as follows [16]:

- The controller uses a basic scheduling sequence that has the fastest and safest switches in between.
- The duration of green extends until the last vehicle of the queue left or the maximum green time is reached.
- Based on the hand-crafted properties, some exceptions are made on the scheduling sequence. Two obvious and often used properties are:
  - A phase group is skipped when no vehicles are present.
  - Signal groups within a phase group can be exchanged for a signal where there are vehicles waiting.

The strength of CCOL is that it utilizes the green time optimally by choosing the optimal moment to switch, just after the last vehicle has departed. In general, vehicles never have to wait twice before a red traffic light and after the last vehicle departed another SG is serviced directly. This optimal moment is known from the moment the last vehicle has left and therefore, the time until red will change until the last second. Therefore, this scheduling model is not suitable to enable GLOSA.

3.2.2. Baseline method - DIRECTOR

DIRECTOR stands for Data-driven Intersection and Road Environment Controller for Traffic Optimization in Real-time. DIRECTOR is a controller, developed by Siemens, with the goal to enable Advanced Driver Assistance Systems (ADAS) such as GLOSA systems. To achieve this goal, the controller needs to be predictable, i.e. the time until green or red is known in advance. Based on this, the optimal speed can be determined for the arriving vehicles to minimize the number of stops. With the development of automated vehicles, this information could eventually be used to control the speed of the vehicles. The only way to guarantee a constant time until green is to fix the schedule ahead in time. On the other side, it is also desired that DIRECTOR remains flexible to handle changing traffic conditions. Live detections of the vehicle detectors are used by controllers to adapt to the current traffic conditions. However, the detectors measure the vehicles only for the near future which is within the fixed schedule that ensures a constant time until green. Therefore, DIRECTOR needs to know the arrivals in advance to adapt to the current traffic flow. Hence, DIRECTOR uses a prediction model that predicts the future arrivals mainly based on the vehicle detectors of the preceding intersection. With this model, the controller is still flexible but can also fix the schedule ahead in time.

The prediction model of DIRECTOR uses Recurrent Neural Networks (RNNs) and is developed by Helmy [27] for her master thesis that she conducted at Siemens. Van Senden [80], her successor at Siemens, improved this model by using a Long Short-Term Memory (LSTM) network. Van Senden also created the controller that makes a time plan for the near future based on the predicted arrivals. Haanstra [25] developed an add-on that uses variable weights for trucks to minimize the number of stopping trucks. Trucks cause a lot lost green time due to low accelerating abilities. The add-on can be seen as a detached part of DIRECTOR and has no effect on this master thesis. The methods that DIRECTOR use can be separated into three steps.

1. Data gathering and pre-processing steps
2. Prediction model
3. Control of intersection

All steps of DIRECTOR are explained in this subsection. All explained methods are also used in the proposed controller. For design choices and extra explanation it is recommend to read the master theses of Helmy, Van Senden and possibly Haanstra that includes a complete evaluation and possible improvements of DIRECTOR [27] [80] [25].

DIRECTOR operates in discrete time steps with the use of time bins. The index of a time bin is denoted as T. The transformation from the time interval with the index of T between t_{start} and t_{end} is given by

$$\begin{aligned} t_{start} &= t_0 + T \cdot t_{bin}, \\ t_{end} &= t_0 + (T+1) \cdot t_{bin}, \end{aligned} \tag{3.2}$$

where t_0 is the current time. For example, DIRECTOR predicts the arrivals for 30 seconds in the future with a bin size t_{bin} of 10 seconds. This means that the predictions are made for time bin $T = 2$ which is from $t_{start} = 20$ until $t_{end} = 30$.

A prediction model is trained for each link separately. Therefore, the subsections about data aggregation and the prediction model will focus on one link. For the intersection in Figure 1.3, three prediction models are needed.

Data aggregation and pre-processing
The first step of DIRECTOR is to collect all inputs needed for the prediction model from the past time bin, i.e. $T = -1$. Helmy [27] determined five inputs that are needed for the prediction model. Below the aggregation of all five inputs are explained.

- *Departures from preceding intersection:* Probably the most important inputs are the stop line detectors of the preceding intersection. These are inductive-loop detectors and ensures that DIRECTOR is able to adapt to the traffic flow of the near future. Most vehicles that cross the stop line detectors will eventually arrive at the next intersection. These detections are transformed from the presence of a vehicle at a certain time to vehicle flow per time unit, i.e. transform ones and zeros to number of vehicles per time bin. This transformation is needed because at the end, the traffic flow is the desired output of the prediction model. The aggregation of detections for the past time bin for one signal group is given by

$$\Delta\rho_{od}^{+}(T = -1) = \sum_{d^s \in D_{od}^s} \sum_{t=t_0-t_{bin}}^{t_0} x_d^s(t), \tag{3.3}$$

  where the aggregated detections are denoted as $\Delta\rho_{od}^{+}$ and $x_d^s(t)$ is the detection of a vehicle on the stop line detector d^s at time t.

- *Arrival flows (with turning percentages):* The same method as given by Equation 3.3 is used for the arrival detectors which are also inductive-loop detectors. The arrival detectors are the first detectors of the intersection and each signal group has his own arrival detector. Important is to keep in mind that these detectors are also the main output of the prediction model. The goal of the prediction model is to determine the traffic flow over the arrival detectors shifted 30 seconds in the future. Lateral dispersion of the vehicles have impact on the performance because each link can lead to multiple signal groups. The turning percentage is also given as an input based on the ratio between the number of detections for one signal group and all detections for that link. This helps to predict the correct number of arriving vehicles for each signal group based on the total number of leaving vehicles of the preceding intersection. The turning percentage $turn_{od}$ is determined for all signal groups of the approach and given by

$$turn_{od}(T = -1) = \frac{\Delta\rho_{od}^{+}(t_0)}{\sum_{od \in OD_a} \Delta\rho_{od}^{+}(t_0)}. \tag{3.4}$$

- *Signal states:* The state of the signals has an effect on the speed of the vehicles. To incorporate this effect in the model, the signal states of the upcoming traffic lights are used. For every interval the first signal state is noted. Traffic lights have three states. Green is denoted as 2, amber is denoted as 1 and red is denoted as 0.

$$s_{od}(T = -1) = \begin{cases} 2, & \text{if } s_{od}(t_0 - t_{bin}) = GREEN, \\ 1, & \text{if } s_{od}(t_0 - t_{bin}) = AMBER, \\ 0, & \text{if } s_{od}(t_0 - t_{bin}) = RED. \end{cases} \tag{3.5}$$

- *Presence of queue:* The presence of a queue also influences the speed of the arriving vehicles. These detectors work the same as the earlier mentioned vehicle detectors. Because the area of this detector is much bigger, the vehicles are detected over a longer period. It is important to distinguish vehicles that drive over the detector and vehicles that stopped in the queue. This separation is done by using a time threshold. When the detector is occupied for at least 3 seconds without a break, a queue is noted. A queue is present if

$$x_d^q(t) = 1 \ \wedge \ x_d^q(t + \Delta t) = 0 \text{ with } t_0 - t_{bin} \le t \le t_0 \text{ and } \Delta t \ge 3 \tag{3.6}$$

  is true and a one is noted in that case, otherwise a zero is noted.

- *Time:* Time is separated into time of day and day of week. For time of day the start time of the interval is taken as seconds that has passed after midnight. For example, noon has the time $12 \cdot 60 \cdot 60 = 43200$. Day of the week is noted with the numbers 0 to 6 with each day of the week its own number.

The result is a matrix for each link as shown in Figure 3.2. Every time step, the new aggregated data from $T = -1$ is included in the matrix. Each column contains all above mentioned input features. This input vector is to scale for the example shown in Figure 3.1. The rows contain the aggregated data for the interval $T = [-n, -1]$. Detections before $T = -1$ are still useful to determine the predicted arrivals. Therefore, the input for the prediction model is the aggregated data of the previous n time steps.

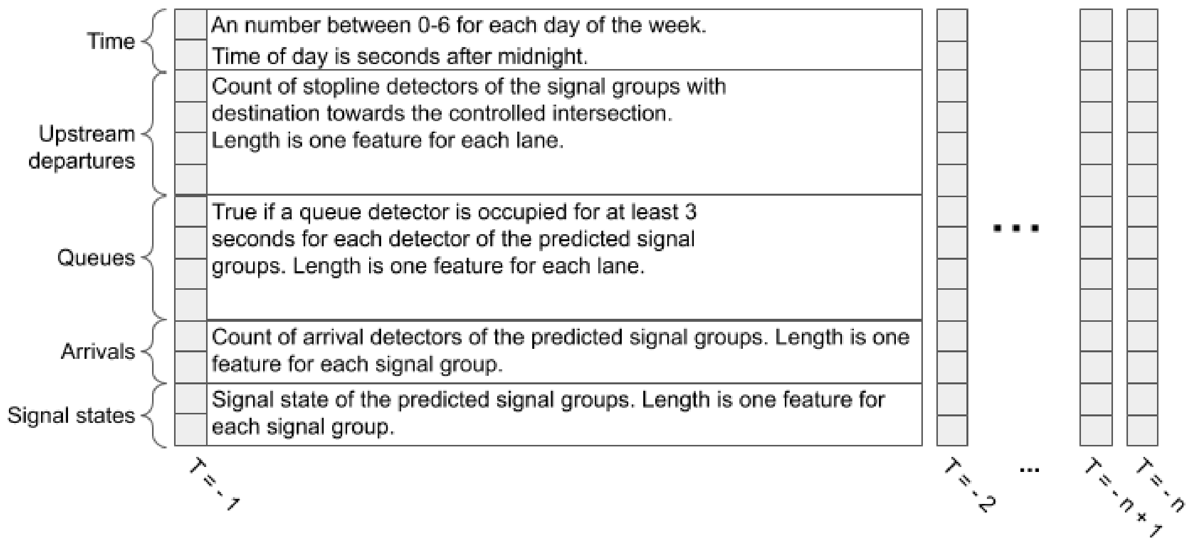


Figure 3.2: Matrix of inputs for the prediction model. The rows represent the number of input features which depends on the configuration of the link. The matrix contains n columns which represents the number of time bins from the past that are used for the prediction.

The left side of Figure 3.3 gives an example of the aggregation of arriving vehicles for the past time bin for a signal group. The detected arrivals $x_d^f(t)$ are aggregated using Equation 3.3 which results in $\Delta\rho_{od}^{+}(T = -1) = 2$.

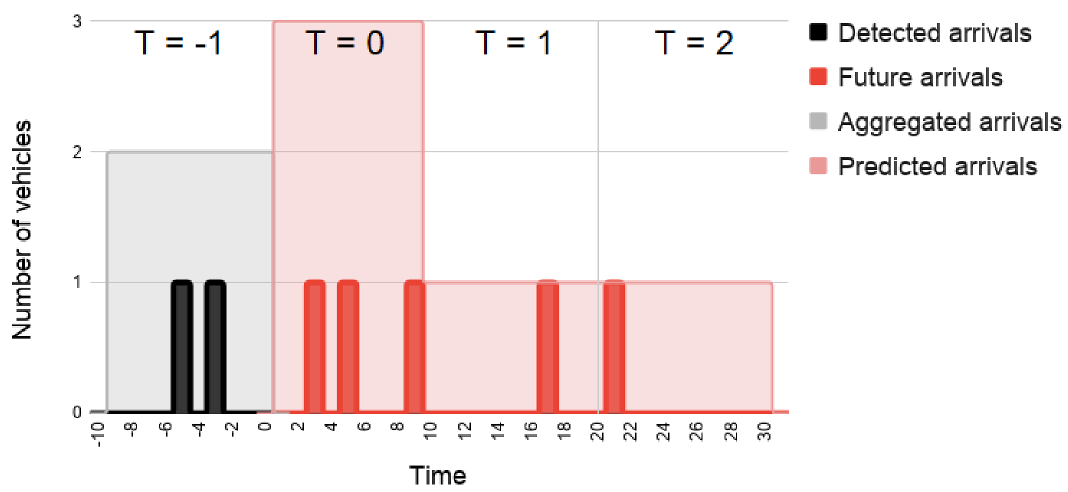


Figure 3.3: Example of aggregation of the detected (black) and predicted (red) arrivals over time. $T = -1$ is based on the detections of the arrival detectors, $T = 0$ and upwards is based on the predictions of the LSTM network.

The next step is to perform some pre-processing steps to make the data easy interpretable for the neural network. Pre-processing is not mandatory but it is highly recommendable to boost the performance. In this case, the data is standardized, i.e. the data is transformed to have a mean of zero and a standard deviation of 1.

The exact mean and deviation is unknown for the inputs. Therefore, these parameters are based on historic data that is used for the training of the model. The data set contains N observations that is collected Q time bins in the past. Each observation is the aggregated data of one time bin with all features as described in Figure 3.2. The standardization is performed for each feature separately. A feature $x(T=-1)$ is standardized which results in the input for the prediction model as given by

$$z(T=-1) = \frac{x(T=-1)-\mu}{\sigma} \quad \text{with} \quad \mu = \frac{1}{N}\sum_{T=-N-Q}^{-Q} x(T) \quad \text{and} \quad \sigma = \sqrt{\frac{1}{N}\sum_{T=-N-Q}^{-Q}(x(T)-\mu)^2}. \quad (3.7)$$

Prediction model

The prediction model uses the aggregated data to predict the future arrivals at the arrival detector. The prediction model needs to be trained once for each link based on historic data. The prediction model is a LSTM network for short-term traffic flow predictions. LSTM is a RNN that uses five different features as input corresponding with the outputs of the data aggregation. The architecture of the network is shown in Figure 3.4. The main target of the prediction model of DIRECTOR is to predict the arrivals over 3 time bins. This are the arrivals for 20 until 30 seconds in the future with a time interval of 10 seconds. The outputs of the model are the three of the five inputs described as above, but shifted over 30 seconds, i.e. 3 time bins. The master thesis of Van Senden can be used for extra information about the design choices regarding the LSTM architecture [80].

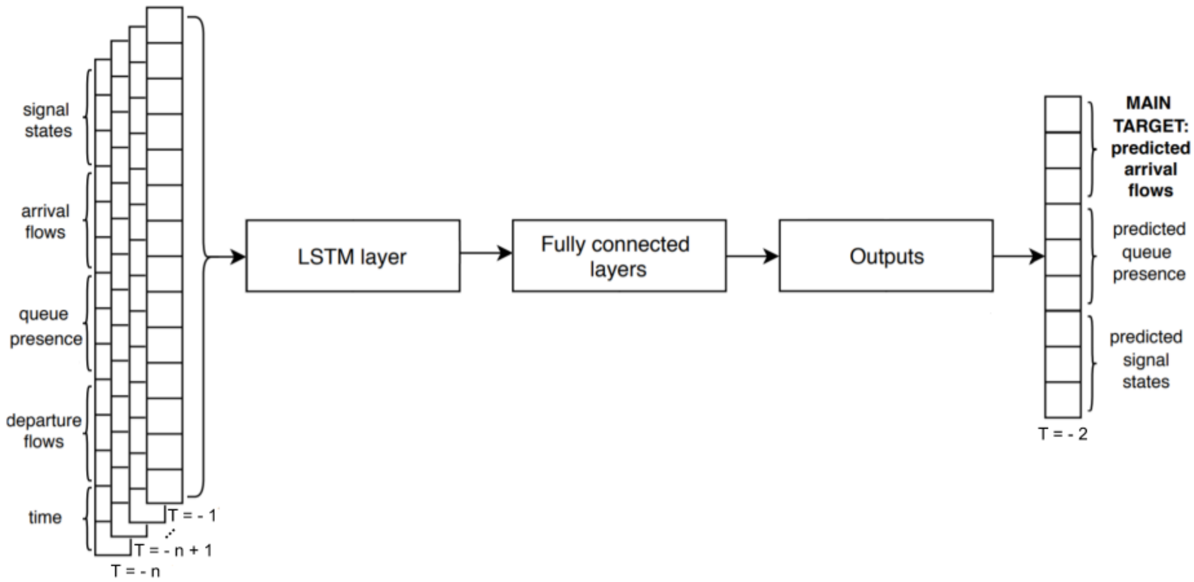


Figure 3.4: Schematic representation of the LSTM architecture used in DIRECTOR [80]. The inputs of n time bins in the past are used to make predictions for $T = 2$

The right side of Figure 3.3 gives an example of perfect predictions of the arriving vehicles for a signal group. The future arrivals are unknown at the time of predicting. The arrivals are predicted for $T = 2$. The output of the prediction model is $\Delta\rho^{+}_{od}(T = 2) = 1$. After the time interval has past, this prediction is moved to $T = 1$, etc. The result is the predicted arrivals up to 30 seconds in the future. If the future arrivals are correctly predicted, the vehicles will cross the arrival detectors as time goes on. From this moment on, they are no longer predictions but known vehicles that entered the queue. The predictions (red) are replaced for the real measurements (black).

Control of intersection
The controller is developed by Van Senden [80] at Siemens, based on the self-organizing intersection controller developed by Lämmer [37]. The controller uses the estimated arrivals to make an optimization over the near future. It is important to include future arrivals because it is not allowed, and due to safety reason not desired, to switch signal states instantly. Measures such as amber time, clearance time, minimum green time and minimum red time will ensure safety. The procedure of switching signal states (and back) effects the near future and therefore it is important to incorporate future arrivals in the optimization.

The aggregation of data and the prediction model both uses a time interval of 10 seconds. The same time interval is used by the controller. Every 10 seconds, the data is aggregated, the arrivals are predicted and the time plan is determined for another 10 seconds. A smaller interval will likely improve the performance, however, this will also increase the computational load. DIRECTOR can operate in two **scheduling modes**. The scheduling mode is an adjustable parameter of DIRECTOR which indicates the length of the fixed control horizon, i.e. within this time frame it is no longer allowed to change the schedule. For DIRECTOR, the full length of the known schedule is always fixed and therefore there is no extra predicted control horizon available. The schedule is determined for the upcoming 10 seconds in the scheduling mode with zero bins fixed. This will result in a controller that has no control horizon beyond those 10 seconds. Therefore, the second scheduling mode fixes the schedule for 10 seconds extra, i.e. the scheduling mode with one time bin fixed, to improve the predictability. Every time step, the schedule is determined for $T = 1$ and the schedule determined at the previous time step will be the schedule for $T = 0$. No modifications are allowed in the schedule of $T = 0$ to ensure a constant time until green in the last 10 seconds. Both scheduling modes are explained below.

Selection of the optimal signal group
In both scheduling modes, DIRECTOR tries to minimize the cumulative travel time delay. DIRECTOR will service the signal group that give the greatest travel time delay if not serviced. The travel time delay D_{od} over the interval (t_{start}, t_{end}) is given by

$$D_{od}(t_{start}, t_{end}) = \int_{t=t_{start}}^{t_{end}} \rho_{od}(t), \tag{3.8}$$

where $\rho_{od}(t)$ is the queue length at time t. D_{od} is equal to the area under the black line in Figure 3.5. The queue length at t_0 is based on the counts of the stop line and arrival detectors. All vehicles that crossed the arrival detector are present in the queue. All vehicles that crossed the stop line detector left the queue. It is not possible to determine the queue length for the future. To determine future queue lengths, the predicted arrivals from the prediction model are needed as given by

$$\rho_{od}(t) = \rho_{od}(t_0) + \Delta\rho_{od}^{+}(t_0, t), \tag{3.9}$$

where $\Delta\rho_{od}^{+}(t_0, t)$ are the arrivals between t_0 and t. Future departures are not taken into account because the signal group with the greatest travel time delay if not serviced is needed. The exact moments of the arrivals are unknown because the prediction model operates in time steps. The number of arrivals is only known within the used time steps of the prediction model. Because the exact moment of arrivals is unknown, an uniform distribution is assumed of vehicle arrivals within the time bin. To have a clear distinction between the average and actual queue length, a new notation is used. The average queue length during the interval $(T \cdot t_{bin}, (T+1) \cdot t_{bin})$ is denoted as $\rho_{OD}[T]$. The average queue length within a time bin is the sum of the waiting vehicles at the start of the time bin $\rho_{OD}(T \cdot t_{bin})$ summed with the arriving vehicles during that time bin $\Delta\rho_{OD}^{+}[T]$ divided by two. Equation 3.9 is rewritten to

$$\rho_{od}[T] = \rho_{od}(T \cdot t_{bin}) + \frac{1}{2} \cdot \Delta\rho_{od}^{+}[T]. \tag{3.10}$$

The travel time delay must handle the discretized version of the queue length. The travel time delay of each time step is summed to determine the total travel time delay. Equation 3.8 is rewritten to

$$D_{od}[T_{start}, T_{end}] = t_{bin} \cdot \sum_{T=T_{start}}^{T_{end}} \rho_{OD}[T], \tag{3.11}$$

where $D_{od}[T_{start}, T_{end}]$ is the average travel time delay for all time bins within $[T_{start}, T_{end}]$. This is equal to the grey area in Figure 3.5. The signal group with the highest travel time delay will be serviced in the next interval.

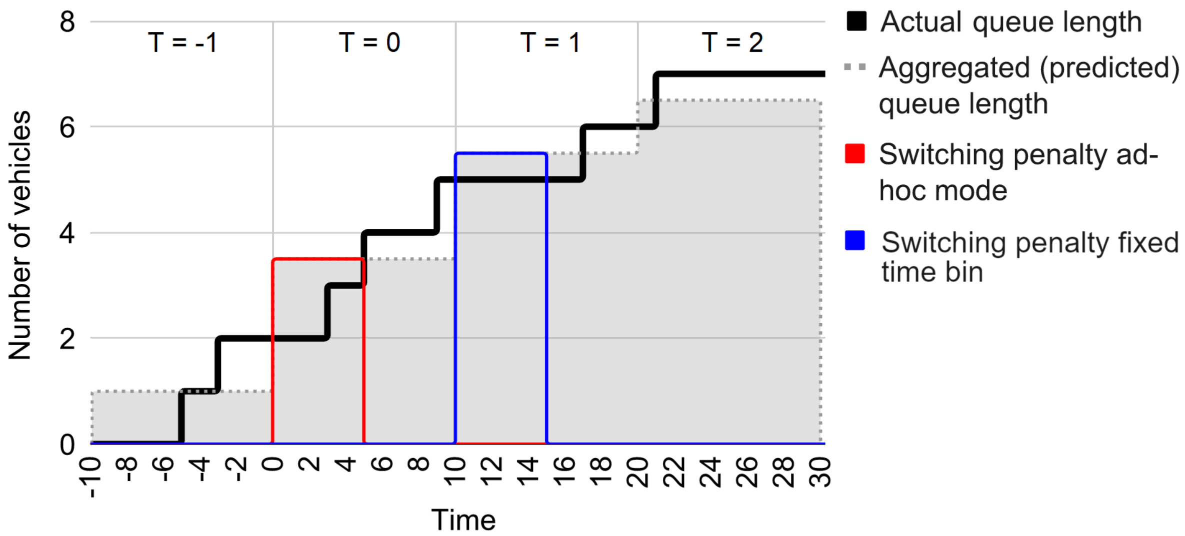


Figure 3.5: Queue length for a signal group if the signal group is not serviced based on the summed arrivals over time of the example in Figure 3.3.

Figure 3.5 gives the queue length from $t = -10$ until $t = 30$ based on the (predicted) arrivals in Figure 3.3. The actual queue length is unknown but an aggregated queue length is determined based on the prediction model. DIRECTOR in the scheduling mode with one time bin fixed, determines the best schedule for $T = 0$ based on the predicted queue length. The queue length at $t = 0$ ($\rho_{od}(0)$) is two vehicles. The number of arriving vehicles during time bin $T = 0$ ($\rho^+_{od}[0]$) is three. Filling in Equation 3.11 for $T = 0$ gives the total travel time delay $D_{OD}[0, 0]$ of 35 seconds if the signal group is not serviced during $T = 0$. This is equal to the grey area of the aggregated queue length in $T = 0$ in Figure 3.5. DIRECTOR can also determine the schedule for $T = 1$. The calculation is similar but shifted for one time interval. This scheduling mode determines the total travel time delay for $T = 1$. Using Equation 3.11 for $D_{OD}[1, 1]$ results in $10 \cdot 5.5 = 55$, which is equal to the grey area under the aggregated queue length for $T = 1$. The previous schedule of $T = 1$ is moved to $T = 0$ after the time interval has finished. The controller will determine the cumulative travel time delay for all signal groups and service the signal group with the highest travel time delay. DIRECTOR uses the same time interval t_{bin} as the prediction model. The schedule is determined for one time bin at a time based on the cumulative travel time delay of that time bin. Therefore, $T_{start} = T_{end}$. The signal group that is scheduled $\chi(t)$ during one time bin is given by

$$\chi(t) = \underset{od \in OD_n}{\arg\max}\left\{D_{od}(t, t + t_{bin})\right\} \approx \underset{od \in OD_n}{\arg\max}\left\{D_{od}[T, T]\right\} = \underset{od \in OD_n}{\arg\max}\left\{D_{od}[T]\right\}. \tag{3.12}$$

This control approach could lead to a schedule with many switches between signal groups. However, instantly switching is not allowed for operating TLCs. Switching penalties are used to incorporate the lost time of switching in the controller. The size of the switching penalty ϵ_{od} is equal to the cumulative travel time delay caused during the switch and given by

$$\epsilon_{od}(t) = \begin{cases} 0, & \text{if } od = \chi(t - t_{bin}) \\ t^{switch}_{od,\chi(t-t_{bin})} \cdot \rho_{\chi(t-t_{bin})}[T], & \text{if } od \neq \chi(t - t_{bin}). \end{cases} \tag{3.13}$$

The switching penalty is based on the queue length of the signal group serviced in the previous time bin $\chi(t - t_{bin})$ because those vehicle could continue if no switch occurred. The duration of the switch is denoted as $t^{switch}_{od,\chi(t-t_{bin})}$. The switching time also depends on the previous serviced signal group because that signal group needs to switch to red. The equation only holds when $t^{switch}_{od,\chi(t-t_{bin})} \leq$

t_{bin}, otherwise, signal groups that were serviced further back must be included in Equation 3.13. The switching penalty is incorporated in Equation 3.12 which results in

$$\chi(t) = \underset{od \in OD_n}{\arg\max} \left\{ D_{od}(t, t + t_{bin}) - \epsilon_{od}(t) \right\}. \quad (3.14)$$

Using the same example of Figure 3.5 and assuming that this signal group is serviced ($\chi(t - t_{bin})$). The switching penalty for this signal group is zero because it is already serviced and no switching is needed. Other signal groups will get a switching penalty. Assume $t^{switch}_{od,\chi(t-t_{bin})}$ is 5 seconds. Filling in Equation 3.13, the switching penalty for signal group od is $5 \cdot 3.5 = 17.5$. The area under the red line in Figure 3.5 visualizes the switching penalty. These seconds are subtracted from the travel time delay for signal group od as given by Equation 3.14. When the schedule is determined for $T = 1$, the calculation of the switching penalty is also shifted 10 seconds. This will result into the area under the blue line in Figure 3.5 when using again $t^{switch}_{od,\chi(t-t_{bin})} = 5$ but now for $T = 1$.

Assumption of no departures

DIRECTOR will service the signal group that gives the greatest travel time delay if not serviced. Based on this, the assumption was made that there are no departures included to determine the queue length. However, the queue length for $T = 1$ could significantly differ if the signal group was serviced during $T = 0$. For DIRECTOR to operate with one time bin fixed, this flaw needs to be solved. The average number of departing vehicles $\Delta\rho^-_{od}[T]$ is given by

$$\Delta\rho^-_{od}[T] = \begin{cases} 0, & \text{if } od \neq \chi(T \cdot t_{bin}) \\ \min\{\rho_{od}(T \cdot t_{bin}) + \Delta\rho^+_{od}[T],\ (t_{bin} - t^{switch}_{od,\chi(t-t_{bin})}) \cdot \Delta\rho^{-,sec}_{od}\}, & \text{if } od \neq \chi((T-1) \cdot t_{bin}) \\ \min\{\rho_{od}(T \cdot t_{bin}) + \Delta\rho^+_{od}[T],\ t_{bin} \cdot \Delta\rho^{-,sec}_{od}\}, & \text{otherwise,} \end{cases} \quad (3.15)$$

where $\Delta\rho^{-,sec}_{OD}$ is the number of departures per second. The maximum number of leaving vehicles is limited by the maximum number of vehicles in the queue. The exact number of departures per second is unknown because those vehicles have not yet crossed the stop line detector. DIRECTOR currently uses a static model to determine the expected departures based on an average in historic data. On average, every 2 seconds one vehicle per lane will leave the queue during a green phase, i.e. $\Delta\rho^{-,sec}_{od} = 0.5$ for signal groups with one lane. If the signal group was not scheduled before the previous bin, there are less than 10 seconds left for vehicles to depart because of the lost time due to switching. This deviation will only affect the signal group that was serviced during the previous time bin, otherwise, there are no departing vehicles.

The arrivals are added and departures are subtracted of the queue length to determine the queue length for the next time bin. Equation 3.10 is rewritten to

$$\rho_{od}[T] = \rho_{od}(T \cdot t_{bin}) + \frac{1}{2} \cdot \Delta\rho^+_{od}[T] - \Delta\rho^-_{od}[T-1]. \quad (3.16)$$

Requirements to make DIRECTOR suitable for real-world application

Not all signal groups are conflicting and therefore multiple signal groups can be serviced at the same time. All non-conflicting signal groups are combined in a Phase Group (PG). DIRECTOR will minimize the cumulative travel time delay by scheduling the optimal PG as given by

$$\chi(t) = \underset{pg \in PG_n}{\arg\max} \left\{ \sum_{od \in pg} D_{od}(t, t + t_{bin}) - \sum_{od \in \chi(t - t_{bin})\ \wedge\ od \notin pg} \epsilon_{od}(t) \right\}. \quad (3.17)$$

There are more requirements to make DIRECTOR suitable for real-world application, e.g. minimum and maximum green time. These extra requirements does not influence the structure of DIRECTOR and are therefore not explained in this master thesis. All these requirements are explained in the master thesis of Van Senden [80]. The next sections will explain the methods used in the proposed controller. The structure of the proposed controller is based on the structure of DIRECTOR and the methods in the next sections will be adaptions on this framework of DIRECTOR.

3.3. Optimal features for the prediction model

Now that the baselines have been discussed, this section will present the novel features for the prediction. GLOSA messages can inform arriving vehicles with the optimal arrival time. To properly adapt the arriving times of the vehicles, GLOSA messages needed to be send earlier than the case was for DIRECTOR. This can be achieved by increasing the prediction horizon without loss of accuracy. In Chapter 2, two possible improvements are discussed to increase the time horizon of controllers. The first option uses communication with the vehicles to share their location. This information is however not (yet) publicly available and therefore the second option is currently more promising. This option shares information with preceding TLCs. Expanding the data-driven model of DIRECTOR with more information from preceding TLCs could expand the prediction horizon. Chapter 2 explains that LSTM networks are the best choice for making predictions of time series such as arriving vehicles over time. Therefore, this section will not focus on the type or architecture of the prediction model. It will investigate promising features that are also available in real-world circumstances for optimal predictions. In Subsection 3.3.1 a new method for pre-processing of the features is proposed. In Subsection 3.3.2 the number of features is expanded with extra features from the preceding TLC.

3.3.1. Pre-processing of the features

Pre-processing steps could boost the performance by arranging the data is such a way that it is understandable for a neural network. This subsection covers pre-processing steps that are applicable on the input data for the proposed controller.

Stationary detection data

Standardization of the input data is a standard pre-processing method [33]. This ensures that features with big changes are not automatically more important. The mean is subtracted and the features are scaled in such a way that all values are between -1 and 1. This pre-processing step is also applied on DIRECTOR. Figure 3.6 shows an example of the aggregated arrivals per hour which gives insight in the average over time. The orange line represents the number of arrivals per hour. The stop line and queue detectors follow this same trend. Standardization will result in a data set with the same trend which is represented by the blue line. The trend remains but it scales and is centered around zero. The inputs could be converted to stationary data to remove this trend. Stationary data is constant over time, i.e. there is no seasonally dependency. To achieve this, the previous value is subtracted as given by

$$z(T) = x(T) - x(T-1), \tag{3.18}$$

where $x(T)$ is a general feature that is used in the prediction model. The differenced series $z(T)$ is the change between two consecutive observations. This model is called the random walk model and is often used in the area of finance to predict stock prices or exchange rates [75].

When predictions are made for multiple time bins in the future, all values in between are unknown at the time of predicting. For example, values $x(T-1)$ and $x(T-2)$ are unknown when predictions are made for 3 time bins in the future ($T = 2$). Equation 3.18 is rewritten to

$$z(T) = x(T) - x(T-n), \tag{3.19}$$

where n needs to be equal or greater than the number of predicted time bins. The differenced series $z(T)$ is now the change between two observations, n time bins apart.

After the prediction step, the output of the prediction model must be transformed back to the original units. This transformation is given by

$$E[x(T+n)] = x(T) + \epsilon_n(T+n), \tag{3.20}$$

where $E[x(T+n)]$ is the feature transformed back in original units and $\epsilon_n(T+n)$ is the output of prediction model. The objective of the prediction model is to predict the future arrivals. Therefore, the general feature $x(T)$ in Equation 3.20 is replaced for the aggregated arrivals as given by

$$E[\Delta\rho^+_{od}(T+n)] = \Delta\rho^+_{od}(T) + \epsilon_n(T+n), \tag{3.21}$$

where $E[\Delta\rho^+_{od}(T+n)]$ are the expected future arrivals for time bin $T+n$. $\Delta\rho^+_{od}(T)$ are the detected arrivals, i.e. detected arrivals of the past time bin. The future arrivals are the detected arrivals of the past time bin plus a shock $\epsilon_n(T+n)$. This shock is predicted with the LSTM network and is stationary over time.

The proposed method will provide a stationary data set which is represented by the green line in Figure 3.6. This pre-processing step will remove the trend and zero-center the data set. This data set is zero-centered because all values are subtracted in the n^{th} next time bin. Standardization will only zero-center the data set but not remove the trend.

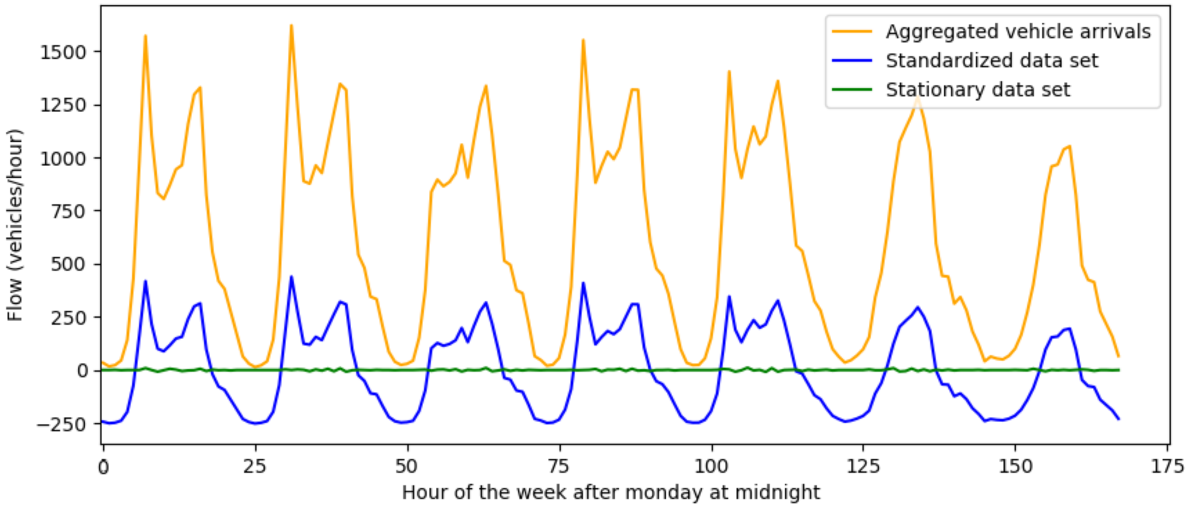


Figure 3.6: Effect of pre-processing steps on the detected arrivals over time. The data is aggregated per hour to give insight in the average over time for one week. All values of the standardized data set are between -1 and 1. There are 360 time bins within an hour, therefore, all values of the aggregated standardized data set are between -360 and 360. Stationary data sets have no trend over time which is the case for the data set represented by the green line.

Time data

As already mentioned in Subsection 3.2.2, the time data in DIRECTOR consists of two parts. The time of the day expressed in seconds after midnight and the day of the week expressed as number between 0 and 6. For the neural network to optimal perform, time data cannot be inputted as regular numbers. Time data has many unique characteristics and the neural network should benefit of knowing these characteristics. The day of the week data, which are categorical features, needs to be converted to numerical features. The time of day, which is an example of cyclic data is converted by using trigonometric functions.

Day of the week data

The blue line in Figure 3.7 represents the day of the week feature as used by DIRECTOR. Categorical features such as the day of the week must be converted to numerical features to use in a neural network [62]. DIRECTOR converts Monday to 0, Tuesday to 1, etc, as seen by the blue line in Figure 3.7. The model can misunderstand this representation by assuming that the feature has a hierarchical order were Tuesday (1) is larger than Monday (0) or the average of Tuesday (1) and Thursday (3) is Wednesday (2). One-hot encoding can prevent this by using each category as separate feature as seen in Table 3.2. The number of features will increase from 1 to 7, one feature for every day of the week. The $IsMonday$ feature is represented with the orange line in Figure 3.7. For both methods, only $1/7$ of the data is available to recognize the patterns of the day of the week. However, the outputs of Monday till Friday follow similar patterns and Saturday till Sunday follow similar patterns. This is good visible in Figure 3.6. Therefore, all weekdays (or workdays) and all weekend days are merged to give a binary output with 1 for workdays and 0 for weekend days. This representation is represented with the green line in Figure 3.7. This will reduce the complexity without deterioration of the performance.

Categorical	Numerical	One-hot encoding				Binary
DayOfWeek	DayOfWeek	IsMonday	IsTuesday	···	IsSunday	IsWorkday
Monday	0	1	0	···	0	1
Tuesday	1	0	1	···	0	1
Wednesday	2	0	0	···	0	1
Thursday	3	0	0	···	0	1
Friday	4	0	0	···	0	1
Saturday	5	0	0	···	0	0
Sunday	6	0	0	···	1	0

Table 3.2: Possible pre-processing steps on the day of the week data. DIRECTOR uses a numerical input with a value between 0 - 6 for each day of the week. One-hot encoding will transform each category in a separate feature. The binary output will combine Monday till Friday because they follow similar patterns.

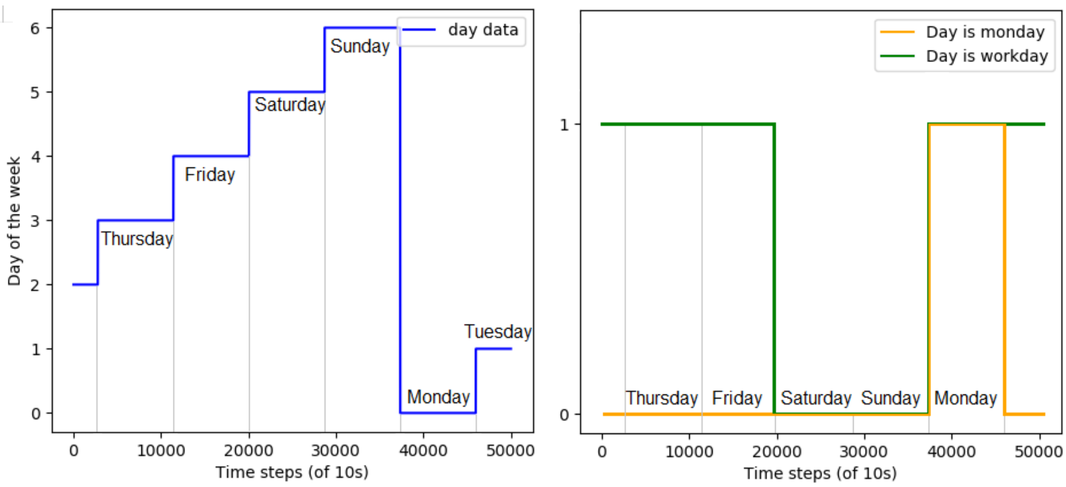


Figure 3.7: Visualization of day of week data. The left graph visualizes the input as used by DIRECTOR and the right graph shows the options for the proposed method.

Cyclic data

Time is a perfect example of cyclical data. Seconds follow a cycle within a minute, minutes follow a cycle within a hour and this continues even for years. The goal is to let our model know that time is cyclical.

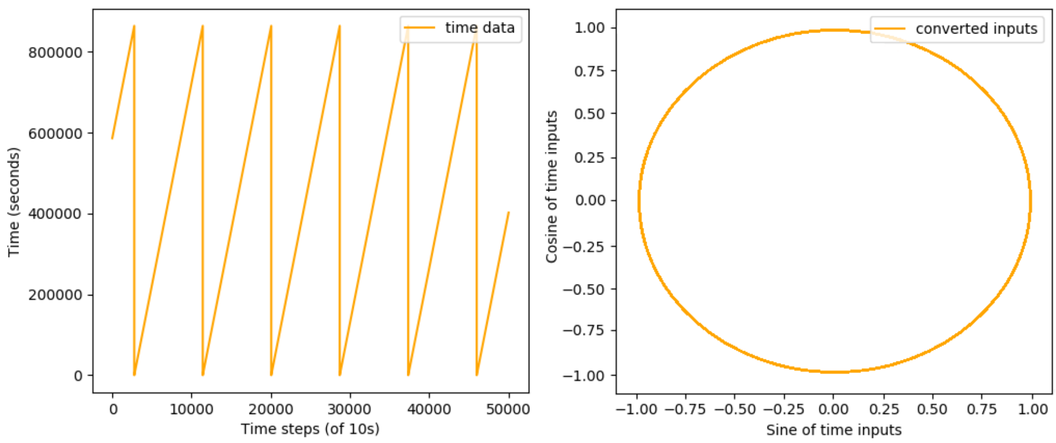


Figure 3.8: The left graph visualizes the original time input for DIRECTOR. The right graph transformed the time data by taking the sine and cosine

As seen in the left graph of Figure 3.8, the time repeats every 8640 time steps, i.e. the number of time steps within a day when using time bins of 10 seconds. However, this representation shows big jumps midnight which are not correlated with an extraordinary jump in time. In reality, a time step during midnight has the same size as all other time steps. A solution used in many other machine learning problems is to replace the time feature in two new features: the sine and the cosine of the time [44]. The result is the right graph in Figure 3.8.

3.3.2. Incorporate features of preceding intersection in prediction model

Input features

As explained in Chapter 1, GLOSA could benefit from a predictable TLC. To achieve this, the prediction horizon needs to be expanded which can be achieved using more information of the preceding intersection. All Dutch intersections collect their information using the V-log protocol. The V-log protocol is an open standard for data logging in traffic control devices [15]. This protocol collects, including the stop line detectors that are already used by DIRECTOR, all information from the signal states, queue detectors and arrival detectors. Future arriving vehicles will cross these detectors earlier because they are placed further upstream. This information could potentially boost the performance of the predictions, especially for predictions over a longer time horizon.

The proposed feature vector is shown in Figure 3.9. The size of the vector is again based on the configuration of the link in Figure 3.1. The prediction model will predict the future flow on the arrival detector (green). This link has two upstream feeding signal groups (SG 2 and SG 10) which results in 2 upstream arrival detectors and 2 upstream signal states. Both signal groups have 2 lanes each which gives 4 upstream departures and upstream queues. In total, these are 12 input features from the preceding intersection. SG 2 and SG 3 also have 2 lanes each which results in another 12 input features. There are always 3 time features, independent of the configuration of the intersection. In total, the feature vector for this link will have a length of 27.

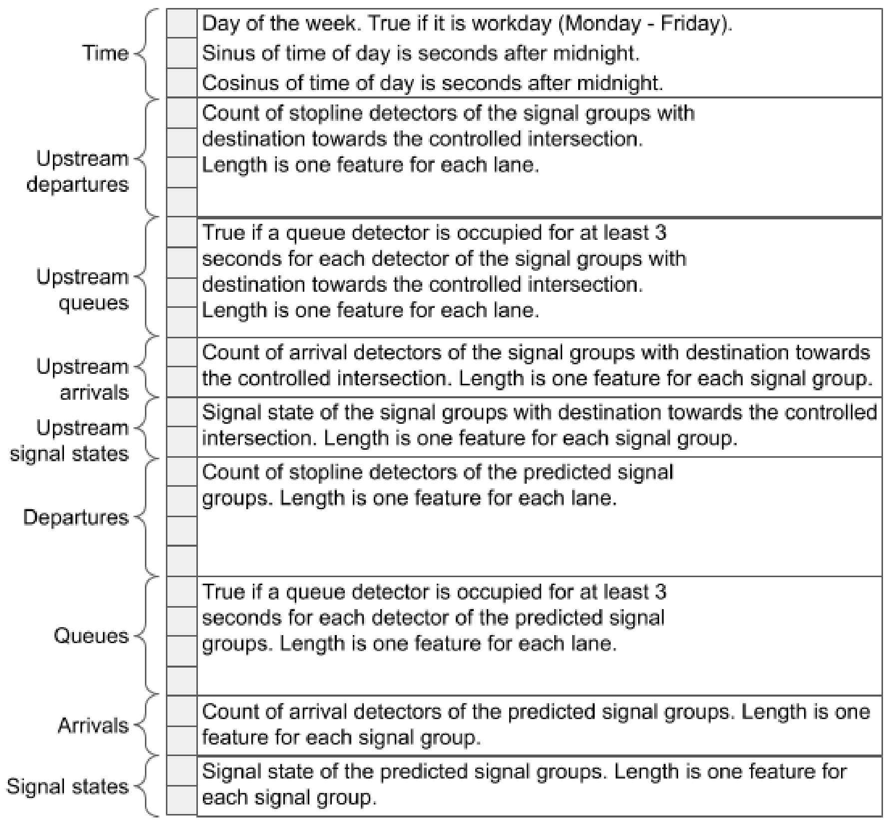


Figure 3.9: The proposed input vector for the LSTM network.

Output features

As already mentioned in Subsection 3.2.2, the outputs of DIRECTOR are the detections on the arrival detectors with turning percentages, queue detectors and signal states. However, the arrival detectors are the only used output for the control and all other outputs are discarded. Helmy [27] discovered that the extra outputs helped recognizing patterns rather than memorizing them. However, currently an LSTM network is used that has internally a remember gate to coop with this problem as explained in Chapter 2. The advantage of only using the arrival detectors as output, is that the error is not minimized over all outputs but only over the arrival detectors. Therefore, the proposed method only uses the arrival detectors as output because the performance of the prediction model is solely based on these detectors.

3.4. Proposed method to incorporate predicted arrivals in traffic light controllers

The controller makes a time plan that causes the least cumulative travel time delay which is based on the queue length and the predicted arrivals. DIRECTOR can do this in two scheduling modes: with zero and one time bin fixed. DIRECTOR makes the predictions over a time horizon of 30 seconds. Fixing zero time bins, only the first 10 seconds are used by the controller. This will be 20 seconds if one time bin is fixed. The predicted queue length after 20 seconds is not used at all to determine the schedule. This section proposes a method to extend the control horizon of the controller with optimal implementation of the prediction model. This proposed method is referred to as P+HOR and separated into two parts.

1. P-HOR extends the control horizon which could be used to enable GLOSA. The length of the control horizon is disconnected from the the scheduling mode (i.e. number of time bins that is fixed). There is always a predicted control horizon which is needed for GLOSA to operate. Therefore, it is desired to disconnect the predicted control horizon from the scheduling mode such that the predicted control horizon has the same length in every scheduling mode. The length of the fixed control horizon depends on the scheduling mode. This step will be explained in Subsection 3.4.1.
2. The second step is the optimal use of the outputs of the prediction model. The quality of the predictions deteriorate with the length of the prediction horizon of the predictions. It is beneficial to make predictions for all time bins instead of only the last time bin of the control horizon. This will maximize the quality of the queue length that is calculated with the predictions. This method is explained in Subsection 3.4.2.

The goal is to improve the performance and make the design of the controller ready to enable GLOSA. Therefore, both methods are merged at the end of this section which results in the design of the proposed controller but still with GLOSA disabled. The next section will explain the process of enabling GLOSA which should lead to further improvement of the performance. This controller is referred to as P-HOR+G.

3.4.1. Extension of the control horizon

This subsection explains the method to increase the control horizon. Figure 3.10 shows the structure of P-HOR (bottom) compared to the structure of DIRECTOR (top). P-HOR will increase the control horizon but not necessarily fix all time bins. The schedule can be redefined at the next interval if desired to increase the flexibility. Both methods will be explained in more depth which will provide clarification on the figure.

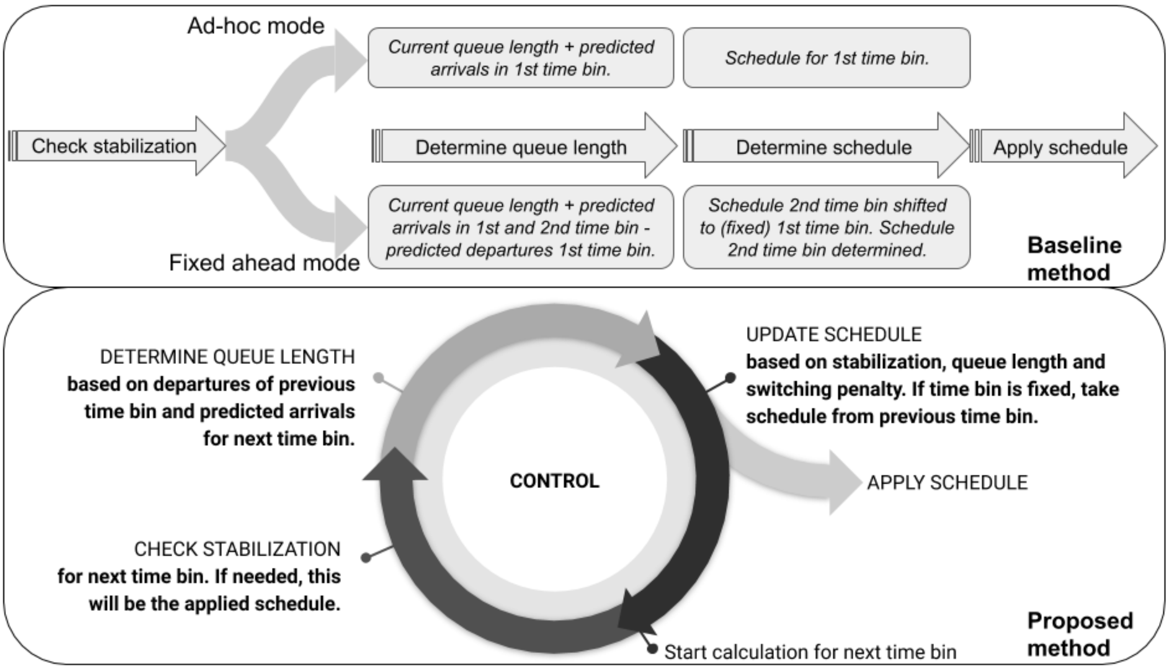


Figure 3.10: On top, the method to determine the schedule for DIRECTOR. This method is limited in two scheduling modes with an equal control horizon. The proposed method (P-HOR) uses a control loop to increase the control horizon of the controller.

Both controllers have the same four main components in their framework which are run through each time step.

- *Stabilization check:* The first step is to check if there is a situation that must be resolved immediately. This is the case when there is a (too) long queue or vehicles are waiting longer than the maximum waiting time. Priority requests for emergency services could also be handled during this check in the future. If stabilization is needed, this signal group is served directly.

- *Queue length determination:* The queue length is calculated as given by Equation 3.10 or 3.16 based on the scheduling mode.

- *Determination of schedule:* The next step is to select best phase group, i.e. the phase group with the highest priority. The priority is the sum of the travel time delay and switching penalty of each signal group within the phase group as given by Equation 3.17.

- *Applying the schedule:* If there is a stabilization task needed, that schedule is applied. Otherwise, the previously selected phase group will be applied. Based on the schedule, the time until green or red can be calculated to enable GLOSA.

DIRECTOR can run these steps in two scheduling modes depending on the number of fixed time bins. If the first time bin is fixed, the schedule from the second time bin is moved to the first time bin at the end of the interval. The schedule needs to be determined for a longer time horizon to enable GLOSA.

P-HOR calculates the optimal schedule for the full prediction horizon independent of the fixed time bins. The cumulative travel time delay is calculated for all time bins within the control horizon. The queue length in a future time bin depends on the schedule of all time bins between now and that time bin because vehicles can depart between those moments. Therefore, the time bins within the control horizon are built one by one. Each scheduled time bin has influences on the queue length for the next time bins.

The equations given in Subsection 3.2.2 are rewritten to make them suitable for longer control horizons. The switching penalty is left out for simplicity. Merging Equation 3.10 and 3.11 will result in

$$\begin{aligned} D_{od}[T_{start}, T_{end}] &= t_{bin} \cdot \sum_{T=T_{start}}^{T_{end}} \rho_{od}[T] \\ &= t_{bin} \cdot \sum_{T=T_{start}}^{T_{end}} \rho_{od}(T \cdot t_{bin}) + \frac{1}{2} \cdot \Delta\rho_{od}^{+}[T]. \end{aligned} \tag{3.22}$$

P-HOR determines the schedule for the full time horizon of the prediction model. The queue length in the future is unknown, therefore $\rho_{OD}(T \cdot t_{bin})$ is only known for $T \leq 0$. Equation 3.16 is rewritten to calculate the queue length in time bin T_x based on the current queue length as given by

$$\rho_{od}(T_x \cdot t_{bin}) = \rho_{od}(0) + \sum_{T=0}^{T_x-1} \Delta\rho_{od}^{+}[T] - \Delta\rho_{od}^{-}[T]. \tag{3.23}$$

This equation already included the departures during previous time bins which is not yet taken into account in Equation 3.22. Hence, the assumption of no departures is solved. The schedule is not determined for the entire control horizon at once but for each time bin separately. Determine one schedule for multiple time bins will probably result in worse performance because there is no opportunity to switch in between. Merging Equation 3.22 and 3.23 with $T_{start} = T_{end}$ will result in

$$\begin{aligned} D_{od}[T_x] &= t_{bin}\left(\rho_{od}(T_x \cdot t_{bin}) + \frac{1}{2} \cdot \Delta\rho_{od}^{+}[T_x]\right) \\ &= t_{bin}\left(\rho_{od}(0) + \sum_{T=0}^{T_x-1}\left(\Delta\rho_{od}^{+}[T] - \Delta\rho_{od}^{-}[T]\right) + \frac{1}{2} \cdot \Delta\rho_{od}^{+}[T_x]\right), \end{aligned} \tag{3.24}$$

where $D_{od}[T_x]$ is the travel time delay in time bin T_x. T_x is equal to the amount of previous finished loops in Figure 3.10. The structure of the P-HOR makes it possible to have an infinitely long control horizon as long as there are predictions available. Each extra loop will extend the control horizon with one time bin. The number of loops determines the length of the control horizon. To enable GLOSA, a schedule of the near future must always be available to calculate the optimal speed which can now be achieved by fixing the number of loops.

3.4.2. Optimal incorporation of the prediction model

The structure of P-HOR as explained above allows to extend the control horizon. This only influences the number of cycles the controller will run through in Figure 3.10. The time horizon of the prediction model needs to be increased accordingly. Section 3.3 already proposed a method to increase the quality of the predictions. However, it is still expected that predictions deteriorate if the prediction horizon is extended. This subsection proposes a method for optimal use of the prediction model.

P-HOR with the control loop will determine the queue length based on predictions over longer time horizons. Poor predictions will result in an incorrect queue length. To increase the quality of the queue length, predictions are determined for every time bin. The structure modification is shown in Figure 3.11. DIRECTOR determines the queue length as shown in the top design (baseline method). The prediction model determines the arrivals for the 3rd time bin which are the expected arrivals between 20 and 30 seconds in the future. This prediction is the next time step shifted to the 2nd time bin and subsequently to the 1st time bin. At the end, these predicted arrivals should have entered the queue because the 30 seconds have passed. The estimated arrivals (predictions) are now replaced for the arrivals measured on the arrival detector. With perfect predictions, these two will be the same. However, especially with predictions over a long time horizon, this will usually not be the case. P-HOR will replace the predictions during every time step, this is seen in the bottom of Figure 3.11. The advantage is that predictions in the less distant future will be in general more accurate. The number of used time bins in P-HOR can be changed as desired.

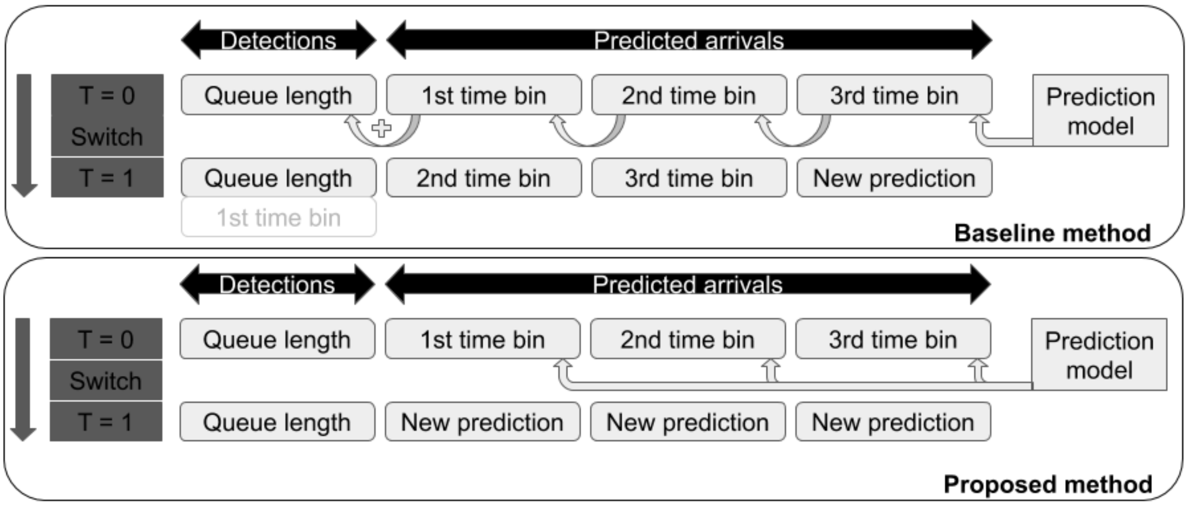


Figure 3.11: The method used by DIRECTOR to determine the queue length based on the predicted arrivals is shown in the top design. DIRECTOR predict 30 seconds in the future with a prediction horizon of 3 time bins and $t_{bin} = 10s$. These predictions are re-used for the entire control horizon of the controller. The proposed methods, shown in the bottom design, uses no depreciated predicted arrivals. Instead, it determines the predicted arrivals for every time bin.

The top design is a visual representation of the calculation of the queue length as described in Equation 3.23. All arrivals $\Delta\rho^+_{OD}$ and departures $\Delta\rho^-_{OD}$ from previous time bins are needed to determine the queue length for a future time bin. $\Delta\rho^-_{OD}$ is solely based on the green time of that signal group in previous time bins which is calculated based on the schedule of the previous time bins. $\Delta\rho^+_{OD}$ is the output of the prediction model. The prediction model of DIRECTOR outputs only the predicted arrivals for 3 time bins in the future ($T = 2$) and shifts the predictions every time step one time bin forward. If one time bin has past, the predictions for time bin T will now be for time bin $T - 1$ as given by

$$\Delta\rho^+_T[0] = \Delta\rho^+_{T-1}[1] = \Delta\rho^+_{T-2}[2] = \ldots = \Delta\rho^+_{T-n}[n]. \tag{3.25}$$

The used od subscript of $\Delta\rho^+$ is omitted for readability. This place is now used to indicate the moment the prediction is made, e.g. $\Delta\rho^+_{-1}[2]$ are the arrivals for $T = 2$ but determined with the information available at $T = -1$. This is the prediction for $T = 2$ made 10 seconds ago. This is now shifted to the prediction for $T = 1$ because 10 seconds have passed.

The prediction model of DIRECTOR only output the predicted arrivals for one time bin. For DIRECTOR, it is not possible to use Equation 3.23 because the arrivals are not predicted for all time bins in between. This equation needs to be rewritten such that it only depends on arrivals in one time bin as given by

$$\begin{aligned}\rho_T((n+1)\cdot t_{bin}) &= \rho_T(0) + \Delta\rho^+_T[n] + \Delta\rho^+_{T-1}[n] + \Delta\rho^+_{T-2}[n] + \ldots + \Delta\rho^+_{T-n}[n] - \sum_{T=0}^{n} \Delta\rho^-[T] \\ &= \rho_T(0) + \sum_{j=0}^{n} \Delta\rho^+_{T-j}[n] - \Delta\rho^-[j]\end{aligned} \tag{3.26}$$

where n is the number of time steps the prediction model can predict in the future. The departures can simply be determined based on the green time. Higher prediction horizons will generally output worse predictions. On the other side, predictions made earlier will improve the predictability of the controller. DIRECTOR determines the queue length based on Equation 3.26. P-HOR can handle multiple prediction inputs, by making predictions for each time bin, and can use Equation 3.23.

Both proposed methods, explained above, can be combined to make a controller with a long control horizon and optimal incorporation of the prediction model. The trade-off between predictability and flexibility needs to be determined. The control horizon for multiple controllers is seen in Figure 3.12. In this figure, the predicted control horizon is arbitrary set to $T = 4$ but can be increased if desired. The fixed control horizon is the length of the fixed schedule.

Schedule of controllers	0	1	2	3	4
DIRECTOR - ad-hoc	$\chi_T(T = 0)$	?	?	?	?
DIRECTOR - time bin fixed	$\chi_{T-1}(T = 1)$	$\chi_T(T = 1)$	?	?	?
Proposed - full predictability	$\chi_{T-4}(T = 4)$	$\chi_{T-3}(T = 4)$	$\chi_{T-2}(T = 4)$	$\chi_{T-1}(T = 4)$	$\chi_T(T = 4)$
Proposed - full flexibility	$\chi_T(T = 0)$	$\chi_T(T = 1)$	$\chi_T(T = 2)$	$\chi_T(T = 3)$	$\chi_T(T = 4)$
Proposed - in between	$\chi_{T-2}(T = 2)$	$\chi_{T-1}(T = 2)$	$\chi_T(T = 2)$	$\chi_T(T = 3)$	$\chi_T(T = 4)$

Fixed schedule | Preliminary schedule

Figure 3.12: Control horizon of multiple controllers. The length of the fixed control horizon of DIRECTOR depends on the scheduling mode. The length of the predicted time horizon of the proposed controller is fixed and independent from the scheduling mode. The figure shows an example with a predicted control horizon of five time bins but this can be increased if desired. A fixed control horizon will increase predictability but is based on predictions from longer back in time.

For DIRECTOR, the number of time bins that are fixed is based on the scheduling mode. If one time bin is fixed, the schedule of $T = 0$ is based on the information from the previous time bin. This will increase the predictability at the cost of flexibility. The proposed method from Subsection 3.4.1 makes it possible to increase the control horizon. This results in a fully predictable controller, but the schedule of $T = 0$ is now based on information from 4 time bins ago. The proposed method in Subsection 3.4.2 uses multiple prediction models to use the most recent available information which results in the fully flexible controller. This is the best schedule the controller can make but it is not predictable because the schedule is updated every time bin. The optimal trade-off is likely within those two methods which results in the proposed controller at the bottom of Figure 3.12. Chapter 4 will test multiple configurations to find the scheduling mode with the highest performance. Enabling GLOSA could influence this optimal configuration. The calculation of GLOSA is explained in the next section.

3.5. Proposed GLOSA implementation

The proposed controller has a schedule for the near future with fixed length to determine the time until green, amber and red. The time until a switch is explained in Subsection 3.5.1. Based on the time until a switch, an optimal speed advice is calculated in Subsection 3.5.2. The controller can communicate the time until a switch directly, so that vehicles can determine GLOSA for themselves or the controller can determine GLOSA. This thesis designs a ITS application and therefore only the second methods is used. Subsection 3.5.3 explains how the controller can communicate GLOSA with the arriving vehicles. This proposed method is an add-on module for the proposed controller and the controller should also work when GLOSA is disabled. The proposed controller with GLOSA is referred to as P-HOR+G. The goal of this section is to develop and implement the GLOSA system which should improve the performance of the TLC.

3.5.1. Calculation of time until switch

The GLOSA system needs the time until the next switch. The time until green is calculated if the current signal state is red and vice versa. The input for this module is the schedule for every time bin that is calculated in P-HOR which has a control horizon with a fixed length.

A switch from not scheduled to scheduled means the traffic light will turn green. Due to safety restrictions, the switch will not occur directly. Before the signal group can switch to green, all conflicting signal groups need to be red and afterwards, there is a clearance before the signal group can switch to green. The duration of the clearance time depends on the combination of previous and next scheduled signal group. The duration of amber depends on the signal group that was serviced (and still is before the switch occurs). The total switching time is given by

$$t^{switch}_{od,\chi(t-t_{bin})} = amber_{\chi(t-1)} + clearance^{\chi(t-1)}_{od}. \tag{3.27}$$

Every signal group has a minimum red duration (min_red_{od}) to prevent instant switching. The signal group is serviced after this minimum is met. The time until green is given by

$$ttg_{od} = \min\left(ttr_{od} + min\_red_{od}, \underset{t\,\in\,n\cdot t_{bin}}{\arg}\left\{\chi(t-1) \neq od \wedge \chi(t) = od\right\} + t^{switch}_{od,\chi(t-t_{bin})}\right), \tag{3.28}$$

with n denoted as the number of time bins of the control horizon.

A switch from scheduled to not scheduled means that the traffic light will turn red. First, the minimum green time (min_red_{od}) needs to be met to prevent instant switching. Before the signal turns red, it will first switch to amber as given by

$$tty_{od} = \min\left(ttg_{od} + min\_green_{od}, \underset{t\,\in\,N\cdot t_{bin}}{\arg}\left\{\chi(t-1) = od \wedge \chi(t) \neq od\right\}\right) \tag{3.29}$$

The time until red is slightly later after the amber time is over as given by

$$ttr_{od} = tty_{od} + amber_{od} \tag{3.30}$$

The expected time until a switch is included in the SPaT message. The SPaT messages uses the V-log standard, which is used by on-street operating TLCs. The time until switch of P-HOR+G is seen in the left graph of Figure 3.13. The SPaT message includes three different times:

- **Minimum End Time** is the guaranteed time without a switch. The switch will occur at this moment or later and it is not allowed to move this time to an earlier moment. If the signal group is scheduled within the fixed control horizon, P-HOR+G can give the Minimum End Time which is the exact time of switching. If the signal group is not scheduled within the fixed control horizon, the Minimum End Time is the length of the fixed control horizon.
- **Likely End Time** is the time until switch as calculated in the previous equations. It is expected that the switch will occur at this moment but it is not guaranteed. P-HOR+G will always give a Likely End Time which is between the Minimum End Time and Maximum End Time. This is based on the (preliminary) schedule or on the length of the predicted control horizon if the signal group is not scheduled. The Likely End Time is used to determine GLOSA.
- **Maximum End Time** is the time in which a switch is guaranteed. The switch will occur at this moment or earlier and it is not allowed to move this time to later moment. P-HOR+G can only give the Maximum End Time when the signal group is scheduled in the fixed control horizon which is the exact time of switching.

The predicting behavior of DIRECTOR is not known because this controller does not share the V-log data. The predictions of the used CCOL controller are shown in the right graph which are less accurate. CCOL is not able to determine the time until the next switch because it will change these times until the last moment.

The accuracy of P-HOR+G depends on the chosen trade-off between predictability and flexibility. More fixed time bins will result in a longer fixed control horizon with perfect knowledge of the time until green. In Figure 3.13, the Minimum End Time does not exceed 20 seconds which is equal to the length of fixed control horizon. The Likely End Time does not exceed 60 seconds which is equal to the predicted control horizon.

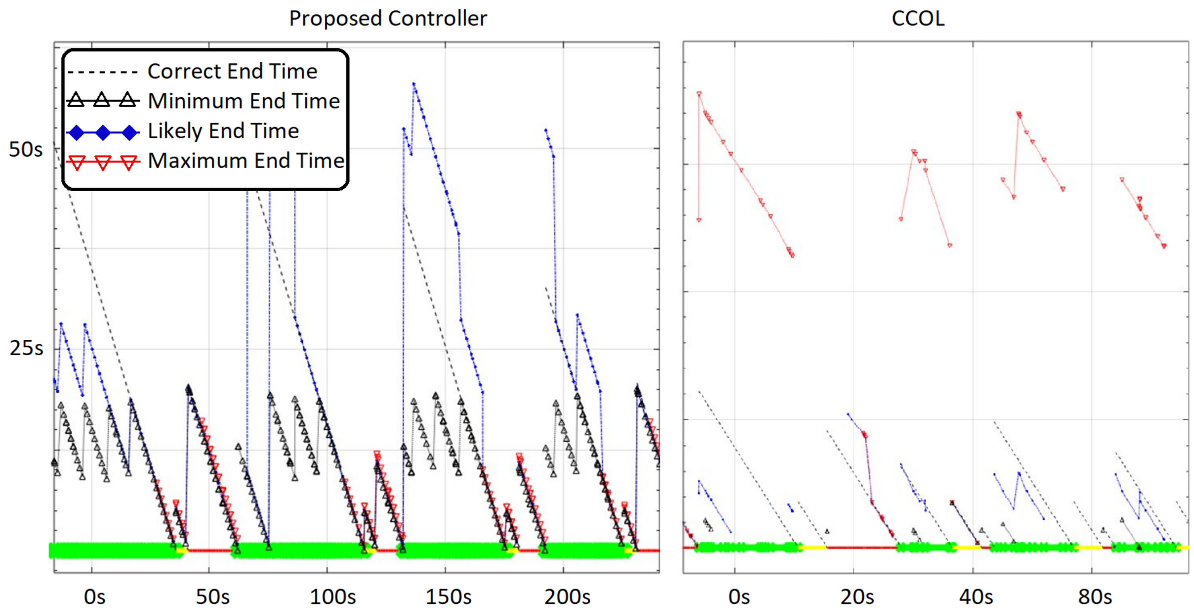


Figure 3.13: Predictions of the time until the next switch of the proposed controller and CCOL.

3.5.2. Calculation of GLOSA

The time until a switch is used to determine GLOSA. Vehicles will need to arrive after the time until green and before the time until red afterwards. This will result in a time frame in which the vehicles should arrive. The optimal moment within this time frame and the corresponding speed depends on the traffic situation. After the traffic light switch to green, the vehicles in the queue need to drive off and only then the arriving vehicles can cross the intersection. An accurate determination of GLOSA needs to take the presence of a queue into account. GLOSA is limited within certain bounds. The maximum speed is the speed limit and a minimum speed is maintained for safety reasons because large speed difference between vehicles can cause dangerous situations. All constraints are given by

$$ttg_{od} \leq t_{od}^{glosa} \leq ttr_{od}, \tag{3.31a}$$

$$ttg_{od} + t_{od}^{delay} \leq t_{od}^{arrival}, \tag{3.31b}$$

$$v^{min} \leq v_{od}^{glosa} \leq v^{max}, \tag{3.31c}$$

where t_{od}^{glosa} is the desired arrival time and v_{od}^{glosa} the corresponding speed. t_{od}^{delay} is the expected delay due to the presence of a queue before the vehicle can cross the intersection.

Besides, the proposed calculation does not make any unrealistic assumptions which makes it not suitable for real-world operation. Therefore, GLOSA cannot be updated constantly because vehicles are not (yet) continually connected with the TLC. GLOSA is only determined once for every vehicle at the moment the vehicle leaves the preceding intersection.

The last needed input to determine GLOSA is the distance of the vehicle from the intersection. GLOSA is determined once at this distance and the vehicle is advised to drive this speed until it reaches the intersection. The determination of this input is discussed in Subsection 3.5.3 and is now considered to be known. Based on the distance of the vehicle d^{veh}, the arrival time without GLOSA and the corresponding time bin is determined as given by

$$t_{od}^{arrival} = \frac{d^{veh}}{v^{max}}, \tag{3.32a}$$

$$T_{od}^{arrival} = \left\lfloor \frac{t_{od}^{arrival}}{t_{bin}} \right\rfloor \cdot t_{bin}. \tag{3.32b}$$

The queue length in time bin $T_{od}^{arrival}$ is already determined by the proposed controller in Equation 3.23. P-HOR+G also has a static model to calculate the expected departures as given by Equation

3.15. t_{od}^{delay} is equal to the time it will to take to depart for all vehicles in the queue as given by

$$\begin{aligned} t_{od}^{delay} &= \underset{t_{od}^{glosa} \in n \cdot t_{bin}}{\arg} \left\{ \rho[T_{od}^{arrival}] = \Delta\rho_{od}^{-}(ttg_{od}, t_{od}^{glosa}) \right\} - ttg_{od} \\ &= \underset{t_{od}^{glosa} \in n \cdot t_{bin}}{\arg} \left\{ \rho[T_{od}^{arrival}] = (t_{od}^{glosa} - ttg_{od}) \cdot \Delta\rho_{od}^{-,sec} \right\} - ttg_{od} \end{aligned} \tag{3.33}$$

The above equations are used to determine GLOSA as given by Algorithm 1. The min_speed and max_speed are constraints given by Equation 3.31. The schedule of P-HOR+G is used to determine $times\_until\_green$ and $times\_until\_red$ with the use of Equation 3.28 and 3.30. These are arrays with all expected switches (excluding the current signal state) within the control horizon. The $delay\_due\_queue$ is determined with Equation 3.33.

Algorithm 1 Calculation of $glosa$

Require: $min\_speed, max\_speed, distance$
Ensure: The controller provides $times\_until\_green, times\_until\_red, delay\_due\_queue$
  if $traffic\_signal == GREEN$ **then**
    if $times\_until\_red[0] > distance/max\_speed$ **then**
      # Vehicle will cross during current green light period
      $glosa = max\_speed$
    else if $distance/max\_speed < times\_until\_green[0] < distance/min\_speed$ **then**
      # Vehicle will cross during next green light period
      $glosa = distance/min(times\_until\_green[0] + delay\_due\_queue, times\_until\_red[1])$
    else
      # Stop is inevitable
      $glosa = min\_speed$
    end if
  else if $traffic\_signal == AMBER$ OR $traffic\_signal == RED$ **then**
    if $times\_until\_green[0] < distance/min\_speed$ AND $times\_until\_red[1] > distance/max\_speed$ **then**
      # Vehicle can (reduce speed to) arrive during first green light period
      $glosa = distance/min(times\_until\_green[0] + delay\_due\_queue, times\_until\_red[1])$
    else
      # Stop is inevitable
      $glosa = min\_speed$
    end if
  end if
  # Ensure that GLOSA is between min and max speed
  $glosa = min(max(glosa, min\_speed), max\_speed)$

Algorithm 1 will advise the minimum allowable speed if a stop is inevitable. In these cases, vehicles that reduces speed still have to wait before the traffic lights and could consider the system as unreliable. The result could be that these vehicles will not reduce speed the next time which will lower the effect of GLOSA systems. Therefore, it could be more effective to only give a speed advice if a stop is prevented. In the other cases, nothing or an informative message could be shared. For example, the time until green is over a minute. Vehicles can decide for themselves to reduce speed or not.

3.5.3. Communication between the controller and vehicles

A communication channel is needed to share the optimal speed with the arriving vehicles. This channel can also be used to request the last unknown input to determine GLOSA, the distance from the intersection of the vehicle.

As explained in Subsection 3.3.2, all communication with TLCs is logged in the V-log protocol. This protocol has already a placeholder for GLOSA in the SPaT message which is called *AdvisorySpeed*. It consists of three variables.

- The **type** of the vehicle. The type could influence the priority. For example, emergency vehicles could have a speed advice without a maximum speed or heavy freight trucks can have a lower maximum speed.
- The **distance** used to determine GLOSA. The optimal speed depends on the location of the vehicle. It is possible to include GLOSA for multiple distances. Vehicles no longer have to send their location but just pick the speed with the distance nearest to itself in the message.
- The **speed** is the optimal speed to arrive during a green phase; GLOSA.

Communication consists of two steps. After the TLC shared the *AdvisorySpeed* via a SPaT message, the vehicle needs to receive this message. There is not (yet) a standard method for receiving SPaT messages. The first potential method is the use of signs on the side of the road which is used in the green-wave model as explained in Chapter 2. The signs are located upstream with a known distance from the intersection. The distance of the vehicle is (almost) equal to the distance of the sign.

Another option is to use an app that can communicate with the TLC. Siemens developed such an app as shown in Figure 3.14. Based on the location of the vehicle, the downstream signalized intersection is determined. If the TLC is connected to the cloud, the V-log data (including the SPaT messages) is collected. Based on this information, the app shows the color of the traffic lights and the expected time until green or red. This app is still in development and it will probably take a while before it is publicly available.

Figure 3.14: Priotalker app developed by Siemens to inform arriving vehicle. The advisory speed is given top left in blue. The signal states of the oncoming TLC is shown in the middle with the expected time until a switch when that is available.

3.6. Setup of simulation

This section explains the simulation model to test the performance of the proposed controller. The simulation is performed in commercial simulation software that is also used in practice. Subsection 3.6.1 describes the simulation setup to simulate a real existing junction using traffic flow statistics from that junction. Previous work used a simplified simulation model, a more realistic and extensive simulation model is needed to test the effects of the GLOSA system. Subsection 3.6.2 describes an improved simulation model that is developed to test the effects of the GLOSA system. It must be possible to connect the proposed controller to a TLC that controls on-street traffic lights to minimize differences with real-world control. Therefore, the protocol of the information flow used to control on-street traffic lights is also used in the simulation as described in Subsection 3.6.3. Subsection 3.6.4 gives an overview of all modifications compared to the simulation model of DIRECTOR.

3.6.1. Case study

The controller needs to be suitable for real-world application and therefore the simulation can only use information that is available in such scenarios. The simulation mimics an intersection between the 'Drie Merenweg' and the N201, also called intersection K201234. Figure 1.3 in Chapter 1 showed the overview of this intersection, including all vehicle detectors and naming conventions for the signal groups. This schematic drawing is based on information obtained from the managing provincial government of this intersection. Besides the dimensions of the intersection, the traffic flow information was also obtained from the provincial government. The output of all detectors are shared via the V-log protocol. Siemens collected multiple data sets including the data set from January 2017 that was used in the master theses of Helmy [27] and Van Senden [80]. In this thesis, the data set of January 2019 is used that includes all extra input features needed to test the proposed prediction model. To make the proposed simulation more realistic, cyclist and pedestrians are included. All these changes could influences the proposed simulation results compared to the baseline simulation of DIRECTOR.

3.6.2. Traffic flow input for the simulation

To keep the simulation as realistic as possible, the vehicles generated in the simulation are based on real-world detections obtained from the provincial government. The initial speed of the generated vehicles is 80 km/h, equal to the speed limit. The baseline simulation generated vehicles with an initial speed of 50 km/h. A detection in the data set equals the presence of a vehicle and a vehicle is generated in the simulation. From the moment vehicles are generated in the simulation, the simulation could differ from the real-world. For example, the speed of the simulated vehicles could differ from the real-world vehicles. In the simulation, a different controller will operate compared to the real-world which results in different traffic scenarios. Therefore, some vehicles in the simulation will need to stop before a red traffic light but real-world vehicles will continue because there is a green phase.

To avoid conflicting detections, all the detectors are included in the simulation. The detections in the simulation will replace the original detections. The goal is to test the effects of different controllers (e.g. CCOL, DIRECTOR, P-HOR, P-HOR+G), but the effects of these changes are simulated. The reactions of the vehicles caused by different controllers are expected to be the same, but that can only be verified with on-street tests.

The proposed simulation also needs to tests the effects of enabling the GLOSA system. GLOSA systems will only have an effect if there is enough time to reduce the speed and delay the time of arrival until the green phase. When GLOSA messages are sent just before the vehicle arrives at the intersection, the gain will be much less compared to a situation where vehicles have more time. GLOSA messages can only be send to vehicles that are already generated. Therefore, it is desired to expand the proposed simulation compared to the baseline simulation. Figure 3.15 shows the effect of the activation distance of the GLOSA system on the number of stops and the fuel consumption.

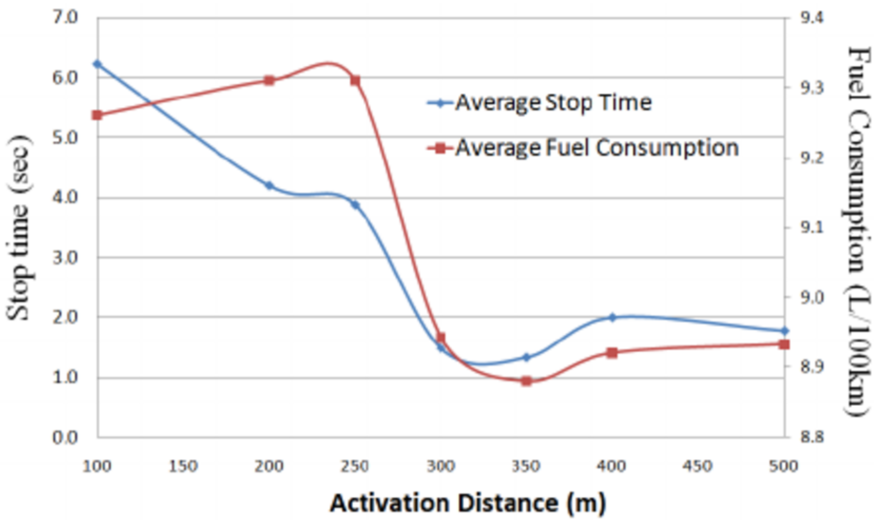


Figure 3.15: The effect of the activation distance of GLOSA at 50km/h speed limit on the performance of the controller. [34]

In the baseline simulation, the vehicles are generated on the arrival detector. The distance between the arrivals detectors and the intersection is 40 - 80 meters and covered in less than 5 seconds at full speed. The full potential of the GLOSA system can be tested if the proposed simulation will generate the vehicles at least 300 meters before the intersection. Figure 3.16 zoomed out on the intersection from Figure 1.3.

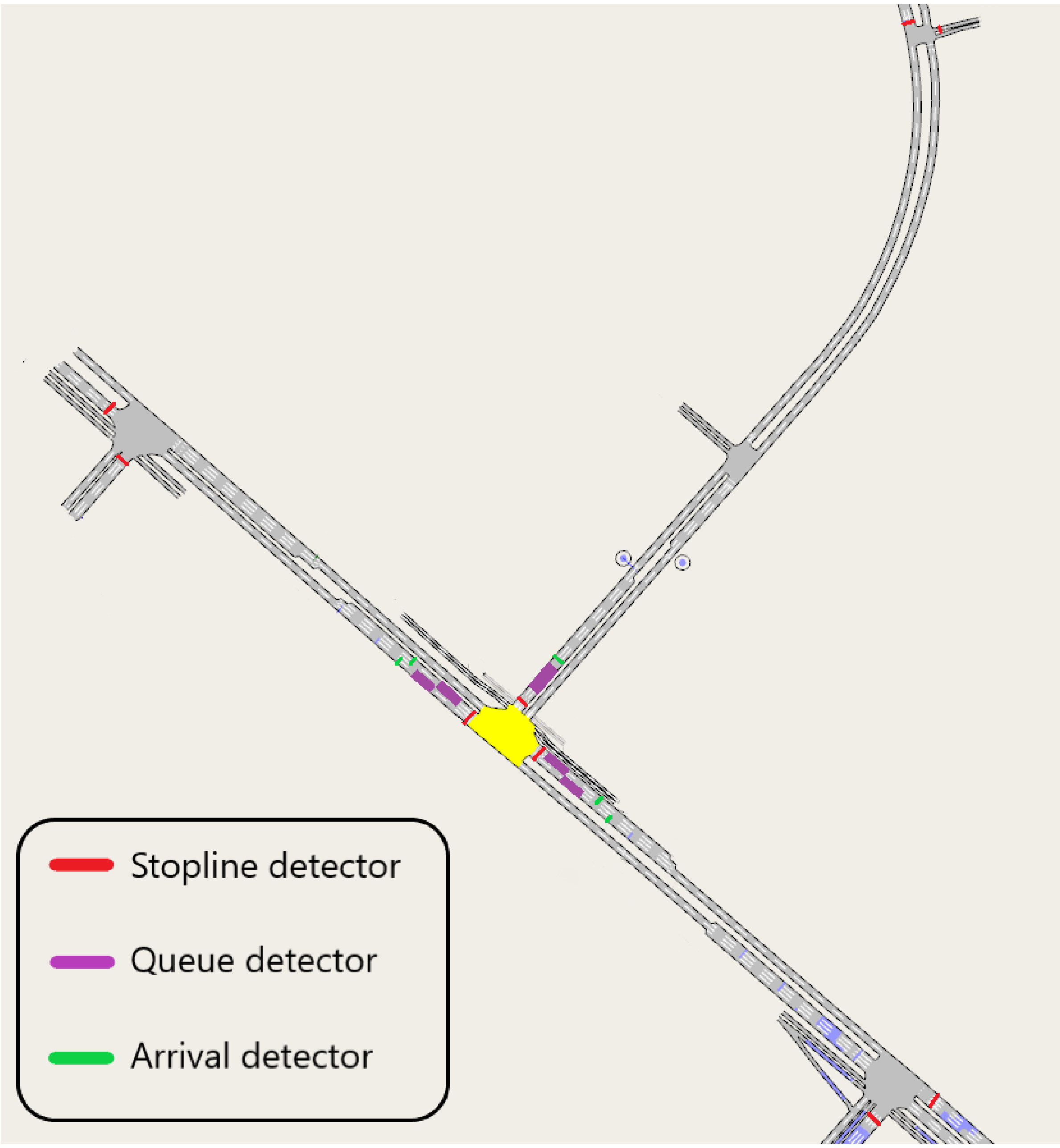


Figure 3.16: Schematic drawing of intersection K201234 including adjacent intersections

The proposed simulation is expanded up to the stop line detectors of the preceding intersections to fully utilize the potential of GLOSA. Vehicles have to travel 370 - 540 meters from the stop line detectors of the preceding intersections until they reach the controlled intersection. Expanding the simulation further is difficult because the TLC of the preceding intersection will influence the behavior of the traffic. The stop line detectors are shown in red on the edges of Figure 1.3.

An additional advantage is that the stop line detectors are single lane detectors and the arrival detectors are double lane detectors. Therefore, the stop line detectors have an higher accuracy because it has a detector for each lane.

Vehicles are not influenced by the preceding TLC after they leave the stop line detector. Vehicles may have to wait for a red traffic light when they arrive at the stop line detector. Therefore, it is not realistic to generate vehicles based on the moment they arrive at the stop line detector. Vehicles are generated when they leave the stop line detector because this is during a green phase. In this case, the color of the traffic light of the preceding intersection has no effect anymore and there are no mandatory stops until they reach the controlled intersection.

The destination of the vehicles generated at the preceding intersection is unknown. The destination of the vehicles in the baseline simulation is known because the arrival detectors are placed after pre-sorting. The destination of the generated vehicles in the proposed simulation is based on the counts of the arrival detector 30 seconds in the future. These detections are placed in a direction queue for each link. Generated vehicles will acquire a destination based on a detection in the queue for that link. If the queue is empty, i.e. there are more detections at the stop line detector than at the arrival detector, the destination is determined based on the ratio of directions of the last 10 generated vehicles for that link.

GLOSA is calculated with Algorithm 1 for vehicles that leave the preceding intersection. The effect of GLOSA also depends on the used min_speed and max_speed. Normally, with $80km/h$, the vehicles will reach the intersection between 16.7 and 24.3 seconds after leaving the upstream stop line detector. Allowing the vehicles to drive at half speed, it will take the vehicles up to double the time to arrive at the intersection depending on GLOSA. A bigger difference between min_speed and max_speed will result in a bigger range of arrival times and more possibilities to determine GLOSA.

It is not possible to implement the use of road signs or an app in the simulation as explained in the previous subsection. However, it is possible to mimic this effect in the simulation. The vehicles will receive GLOSA when they are generated at the stop line detector of the preceding intersection. GLOSA is only determined once. This could be compared with the use of road signs just after leaving the preceding intersection.

3.6.3. Connecting the traffic management installation in the simulation

The Intelligent Transportation System (ITS) is the data aggregation, prediction model and the controller combined. The ITS does not directly control the traffic lights but all information flows via the TLC that checks and redirect the information. The TLC is also incorporated in the simulation to minimize the differences with real-world circumstances. Figure 3.17 shows the information flow in the simulation. The top 3 boxes (data aggregation, prediction model and controller) will together make an optimal schedule based on the detected vehicles. The schedule is included in a SPaT message and sent to the TLC. The TLC checks if the schedule can be applied, e.g. it will reject an instant switch from green to red. It will give feedback to the ITS of the applied schedule, in general this is equal to the schedule determined by the controller. To this extent it is equal to real-world circumstances and this TLC could directly be connected to Dutch intersections. The intersection is simulated in Aimsun and the detections are replayed based on historic data.

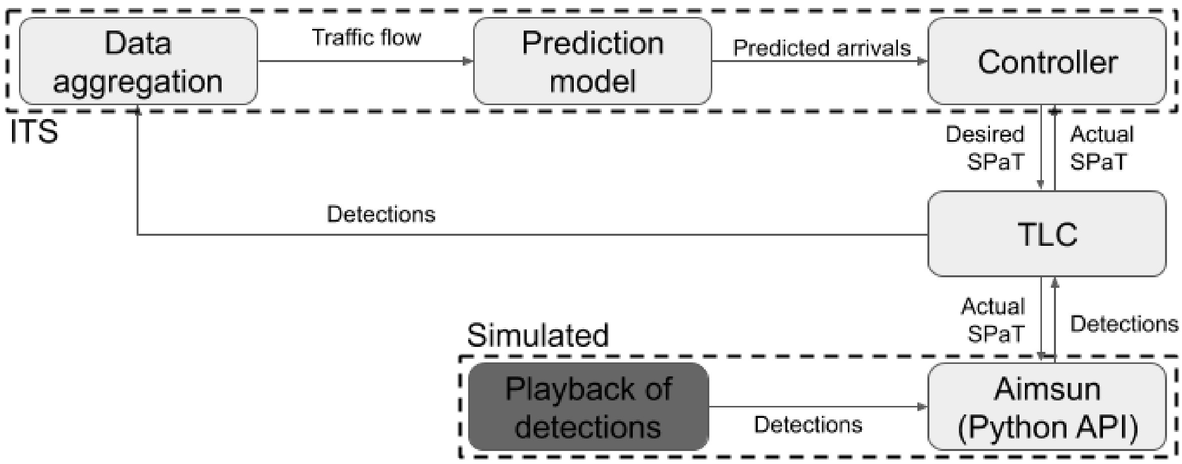


Figure 3.17: The ITS determines the optimal schedule based on the detections. This is sent to the TLC in a SPaT message which will check the schedule for safety requirements. Aimsun simulates the intersection based on historic detection data. It is possible to connect the TLC to on-street intersections.

3.6.4. Modifications compared to the simulation of DIRECTOR

Below an overview is given of all changes compared to the baseline simulation used to test DIRECTOR, the baseline controller. Some of these changes are desired and some changes insurmountable.

- The proposed simulation is performed in the commercial simulation software Aimsun instead of Vissim because of license restrictions. The Vissim license is only free to use for students and this software may not be used for free to develop products used by businesses such as Siemens. Therefore, it is not possible to do new experiments with the baseline simulation setup.
- Pedestrians and cyclist are included in the proposed simulation.
- The speed limit changed from 50 km/h in the baseline simulation to 80 km/h in the proposed simulation.
- In the thesis of Van Senden, it was not clear which (day of the) data set was used for the simulations. This case study is based on the measurements of Tuesday 15th of January 2019 from 5am till 7pm. The prediction model is trained on the 3 weeks in advance.
- The proposed simulations are shorter compared to the baseline simulations (14 hours instead of 24 hours). In this case, simulations will be finished the next day if it is started at the end of the previous workday. The nightly hours are skipped in the simulation. This will probably deteriorate the performance of the controller because these skipped hours are in general less busy. On the other side, the daily hours are the most interesting hours with the most potential for improvement.
- The vehicles are generated at the preceding intersection instead of the arrival detectors to make it possible to enable GLOSA. An additional advantage is that these detectors have a higher accuracy because they are single lane detectors.

4
Experiments

This chapter shows the performed experiments and results to test the proposed methods in Chapter 3. Section 4.1 shows the performance of the prediction model. Section 4.2 tests the quality of the simulation that is used to test the performance of the proposed controller. The goal of the simulation is to mimic real-world traffic flows. The last two sections tests the performance of the proposed controllers. Section 4.4 shows the performance of P-HOR (GLOSA disabled) and Section 4.5 shows the performance of P-HOR+G (GLOSA enabled).

4.1. Performance of proposed prediction model

The Keras package for Python with a Tensorflow back end is used to implement the prediction model. The Key Performance Indicator (KPI) of the prediction model is the Normalized Root Mean Square Error (NRMSE). The baseline method uses also the RMSE as KPI as given by

$$RMSE_{od} = \sqrt{\frac{\sum_{T=0}^{N}(\Delta\rho_{od}^{+}[T] - \Delta\rho_{od}^{+}(T \cdot t_{bin}, (T+1) \cdot t_{bin})^2}{N}}, \tag{4.1}$$

where N is the number of observations in the test set (the length in number of time bins), $\Delta\rho_{od}^{+}[T]$ are the predicted arrivals and $\Delta\rho_{od}^{+}(T \cdot t_{bin}, (T+1) \cdot t_{bin})$ the actual arrivals over time bin T. The RMSE is the absolute error defined in number of vehicles per interval which is in this case 10 seconds.

The experiments are based on the data sets from 2017 and 2019. Comparing the RMSE between different data sets will not give a realistic overview of the performance of the prediction model. The NRMSE is the normalized version by dividing the RMSE with the variance as given by

$$NRMSE_{od} = \frac{RMSE_{od}}{\sigma_{od}}, \tag{4.2a}$$

with the mean μ and σ equal to Equation 3.7 in Section 3.2 as given by

$$\begin{aligned} \mu_{od} &= \frac{1}{N}\sum_{T=0}^{N} \Delta\rho_{od}^{+}(T \cdot t_{bin}, (T+1) \cdot t_{bin}) \\ \sigma_{od}^2 &= \frac{1}{N}\sum_{T=0}^{N} (\Delta\rho_{od}^{+}(T \cdot t_{bin}, (T+1) \cdot t_{bin}) - \mu_{od})^2 \end{aligned} \tag{4.2b}$$

The results of the proposed method explained in Section 3.3 are given below. The results of the baseline version of Helmy [27] and Van Senden [80] are given in Subsection 4.1.1. The results of the proposed prediction model are given in Subsection 4.1.2 which will show the effect of the pre-processing steps and the effect of the incorporation of extra features. Subsection 4.1.3 will give an overview of all improvements and their effect.

4.1.1. Results of baseline methods - DIRECTOR and CCOL

The first version of a prediction model for traffic flow based on recurrent neural networks was designed by Helmy [27] (B-RNN) and thereafter improved by Van Senden [80] with the use of a LSTM network (B-LSTM). Table 4.1 shows the performance of these prediction models for all signal groups (SGs). These results are obtained from the master thesis of Van Senden [80]. The baseline simulation model is used and the data set of 2017 is used for the traffic input. The configuration of the signal groups is seen in Figure 1.3 in Chapter 1. This prediction performance can be seen as the baseline.

SG	RMSE B-RNN	RMSE B-LSTM	NRMSE B-CNN	NRMSE B-LSTM
2	1.38	**1.29**	0.43	**0.40**
3	0.72	**0.69**	1.00	**0.97**
4	**0.70**	**0.70**	**0.82**	**0.82**
6	0.82	**0.81**	0.74	**0.73**
7	0.92	**0.88**	0.74	**0.71**
8	1.16	**1.07**	0.42	**0.39**
Avg.	0.95	**0.91**	0.69	**0.67**

Table 4.1: RMSE and NRMSE of the predictions based on the traffic flow in 2017 [80]. B-RNN and thereafter B-LSTM are the prediction models used in DIRECTOR. Best performance highlighted in bold. B-LSTM is the best performing prediction model and will be used as the baseline model.

B-LSTM is the best performing prediction model and will be used as the baseline model. The 2017 data set used for these experiments did not contain all needed information from the preceding intersections to test the proposed prediction model as explained in Section 3.6. Therefore, a new data set from 2019 is used which included all needed information. The shift from data set can have impact on the performance of the prediction model. Therefore, Table 4.2 shows the performance of the used baseline model (B-LSTM) for both data sets. The data set from January 2017 is referred to as DS17 and the data set from January 2019 as DS19. The prediction model is trained and tested on data sets of 3 weeks in January. The data sets are split in 80% training set and 20 % test set. The first column is from Table 4.1 above.

SG	B-LSTM-DS17 [80]	B-LSTM-DS19
2	0.40	0.47
3	0.97	0.96
4	0.82	0.82
6	0.73	0.72
7	0.71	0.56
8	0.39	0.39
Avg.	0.67	0.65

Table 4.2: The effect of changing the input data set on the NRMSE. The first column is from Table 4.1. DS17 is the data set from 2017, DS19 is the data set from 2019. Exactly the same prediction models are used in both cases.

4.1.2. Results of proposed method with improved features

The proposed prediction model contains two set of adjustments. First, the proposed pre-processing steps are tested. Thereafter, the proposed extra features are tested.

Proposed pre-processing steps

The proposed pre-processing steps makes the detection data stationary, removed the jumps in the time of day data and converted the day of week to logical numerical features as explained in Subsection 3.3.1. This prediction model is seen in the 2nd column of Table 4.2. The proposed pre-processing steps reduces the average NRMSE with 5%, only SG 2, SG 7 and SG 8 have some performance deteriorating.

SG	B-LSTM	P-LSTM+PP
2	0.47	**0.46**
3	**0.96**	0.97
4	0.82	**0.74**
6	0.72	**0.64**
7	**0.56**	0.57
8	**0.39**	**0.39**
Avg.	0.65	**0.63**

Table 4.3: The effect of the proposed pre-processing steps (PP) on the NRMSE of the predictions. The same prediction model (LSTM) and the same input (DS19) is used for both experiments. The first column is from Table 4.2. Best performance highlighted in bold. The proposed pre-processing will improve the predictions.

Proposed extra features

Table 4.4 shows the results of all available extra input features as explained in Subsection 3.3.2. The baseline input feature consists of time, upstream stop line detectors, queue detectors, arrival detectors and the signal states as seen in Figure 3.2. The proposed extra features are the upstream signal states (SS), upstream queue detectors (Q) and the upstream arrival detectors (A) as shown in Figure 3.9. All extra features are tested separately and the combination is tested. The column name indicates the extra input feature besides the five already used feature in the baseline method. The extra inputs are all unused features that are logged in the V-log protocol. The last column combined all new inputs (COM) which leads to the highest performance. For all columns, the output are the predicted traffic flow over arrival detectors over 3 time bins (between 20 and 30 seconds in the future) and the data set of 2019 is used.

SG	P-LSTM+PP	P-LSTM+PP+SS	P-LSTM+PP+Q	P-LSTM+PP+A	P-LSTM+PP+COM
2	0.46	0.41	0.44	0.40	**0.33**
3	0.97	0.89	0.91	0.88	**0.83**
4	0.74	0.70	0.69	**0.67**	**0.67**
6	0.64	0.58	0.58	**0.54**	**0.54**
7	0.57	0.53	0.53	**0.51**	**0.51**
8	0.39	0.35	0.36	0.35	**0.33**
Avg.	0.63	0.58	0.59	0.56	**0.54**

Table 4.4: The effects of the proposed extra input features on the NRMSE of the predictions. All prediction model uses the LSTM network and the proposed pre-processing steps (PP). The extra input feature are the upstream signal states (SS), upstream queue detectors (Q) and the upstream arrival detectors (A). Best performance highlighted in bold. The last column has the highest prediction performance and uses all available inputs (COM) from the V-log protocol.

4.1.3. Overview of all tested methods

Figure 4.1 shows the progress of the performance of the prediction model. The prediction model was optimized by using the appropriate pre-processing steps in Subsection 3.3.1. Subsection 3.3.2 proposed extra features from the preceding intersection. The green dot with a value of 0.65 represents the average NRMSE for the prediction model developed by Van Senden (B-LSTM-DS19). The red dot with a value of 0.63 represents the average NRMSE after including the proposed pre-processing steps (P-LSTM-DS19+PP). The average NRMSE has dropped to 0.54, represented with the blue dot, after the incorporation of all proposed extra input features (P-LSTM-DS19+PP+COM).

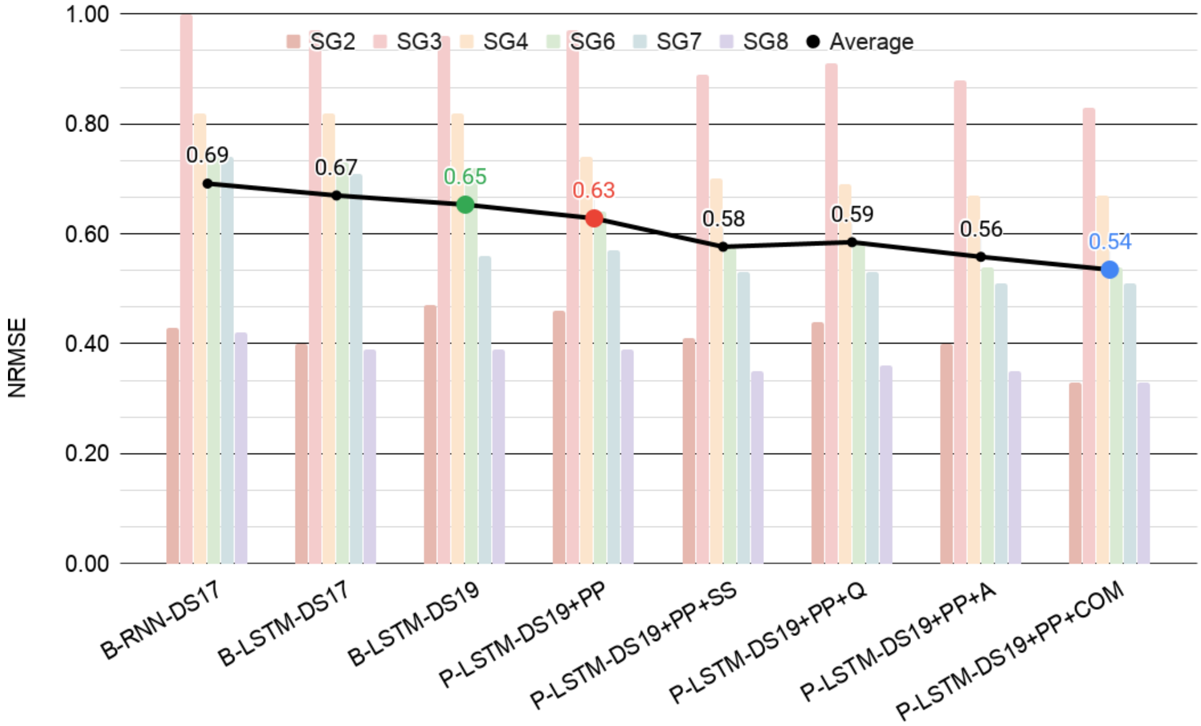


Figure 4.1: NRMSEs for all tested prediction models with a prediction horizon of 30 seconds. The green dot represents the baseline prediction model. The red dot represents the average NRMSE with the proposed pre-processing steps included. The green dot represents the proposed prediction model that combines all extra input features which has the highest prediction performance. The two experiments on the right are performed on the data set of 2017 to give insight in the switch between two data sets.

The goal of the proposed prediction model was to increase the prediction horizon without loss of performance. Figure 4.2 shows the prediction performance for all prediction horizons between 10 and 50 seconds. The prediction model predicts the arrivals per time bin. For $T = 0$, the prediction model predicts the number of arriving vehicles for 0 - 10 seconds in the future. $T = 1$ are the predictions between 10 - 20 seconds, etc.

For all tested methods applies that a longer prediction horizon will result in a higher NRMSE. The NRMSE of the best performing prediction model with a prediction horizon of 50 seconds is 0.64 (green star) which is comparable to the NRMSE of the baseline prediction model with a prediction horizon of 30 seconds (green dot). Concluded, the proposed method improves the prediction performance and achieves higher prediction horizons.

The NRMSE diverges for different prediction horizons when using the proposed pre-processing steps. This is advantageous for all prediction models up to 30 seconds, but disadvantageous for prediction horizon over 30 seconds. Predictions over 30 seconds cannot be based on the upstream stop line detectors because the vehicles have not yet crossed these detectors. As a result, the time input is more meaningful than the inputs of the detectors. The most logical cause is that the B-LSTM-DS19 prediction model with prediction horizons over 30 seconds follows the trend over time. This trend is removed for the prediction model with the proposed pre-processing steps causing the worse prediction performance for predictions over a long time horizon.

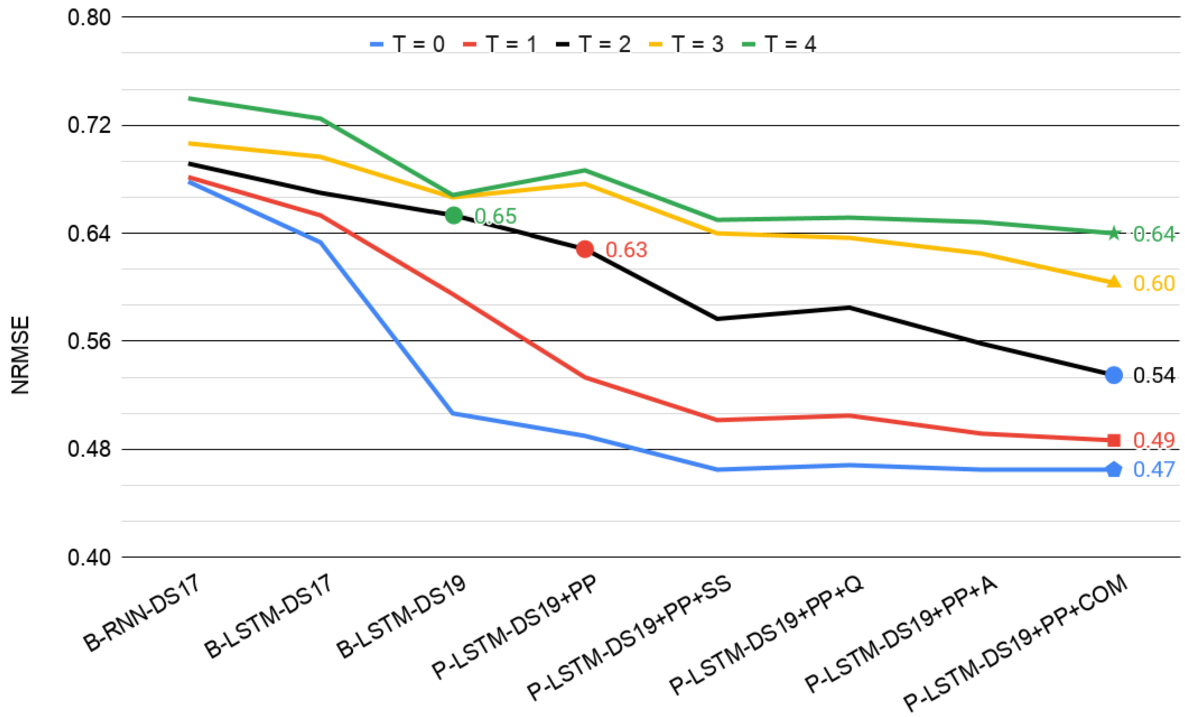


Figure 4.2: NRMSEs for tested prediction models for prediction horizons between 10 and 50 seconds. A longer prediction horizon will result in a higher NRMSE. The proposed pre-processing steps will not improve the performance for predictions above the 30 seconds. However, the proposed pre-processing steps with the proposed extra input do provide an improvement.

4.2. Truthfulness of simulation

Screenshots of both the baseline and proposed simulation are shown in Figure 4.3. The baseline simulation is performed in Vissim and shown on the left side, the proposed simulation is performed in Aimsun and shown on the right side. All modifications are summarized in Subsection 3.6.4. It is not possible to perform new simulations in the baseline simulation model because of license restrictions. All experiments in the baseline simulation are based on the results of previous work.

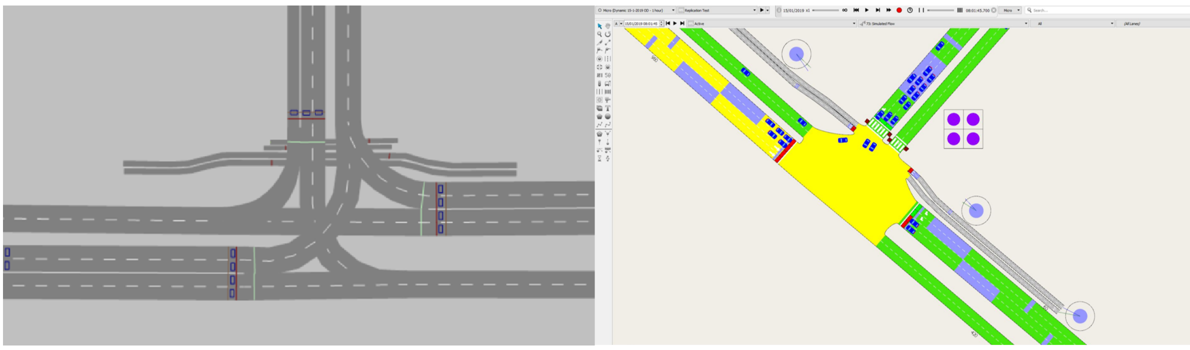


Figure 4.3: The baseline simulation is performed in Vissim. A screenshot of the baseline simulation is seen on the left. Due to license restrictions, the proposed simulation is performed in Aimsun. A screenshot of the proposed simulation is seen on the right.

The proposed simulation model needs to be able to test full potential of the GLOSA system and therefore the vehicles are generated on the stop line detector of the preceding intersection. This required some assumptions on the destination of the vehicles what could influence the truthfulness.

This section will test the truthfulness of the proposed simulation explained in Section 3.6 compared to the baseline simulation. The available data is not sufficient to check the truthfulness of the real behavior of the vehicle drivers. However, it is possible to check if the number of vehicles over time match the real-world numbers. The comparison between the simulation and the real-world situation is shown in Figure 4.4.

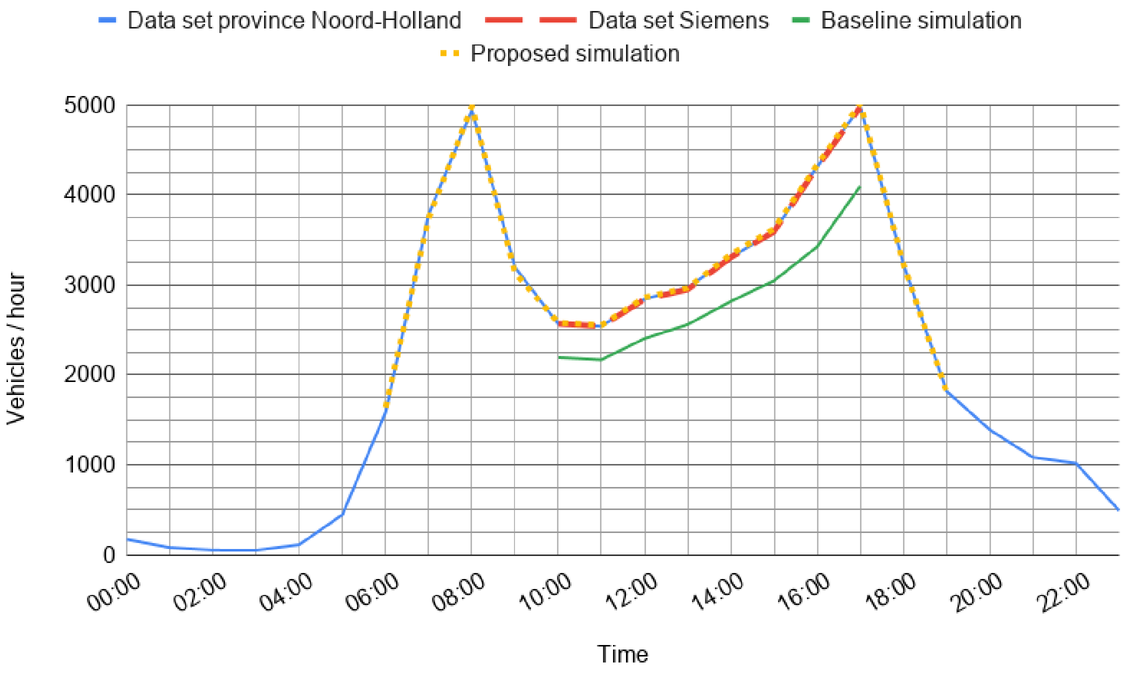


Figure 4.4: A comparison of the number of vehicles in simulations and on-street. The blue and red line represent measurements obtained from on-street stop line detections, respectively collected by the managing province and Siemens. The green line represents the number of generated based on the measurements on the arrival detectors, which is over 10% less compared to all other measurements. The yellow line represents the number of generated vehicles for the proposed simulation where vehicles are generated on the stop line detector of the preceding intersection.

The figure shows the number of (generated) vehicles per hour summed for all links of the intersection during January 5th. The vehicles counts in the real-world situation are obtained in two ways:

- The first data set is collected by the province of Noord-Holland and represented by the blue line. This data is available via a secured website that is used for the monitoring of the TLC performance.
- The red line represents the counts of the stop line detector in the data set collected by Siemens. This data set is based on the same detections only the logging is done by Siemens. This logging can have a lot of influence because it is the translation from ones and zeros to the number of vehicles that crossed the detector. As seen in the figure, the counts of both sources match.

These counts are compared to the counts of both simulations:

- The green line represents the number of generated vehicles of the baseline simulation where vehicles are generated on the arrival detector (close by).
- The yellow line represents the number of generated vehicles in the proposed simulation where vehicles are generated on the stop line detector of the preceding intersection (far away).

The figure shows that the baseline simulation generated between 10% and 20% less vehicles compared to all other counts. A logical cause is that arrival detectors covers two lanes. If vehicles on both lanes cross the detector at the same time, it is only counted as one vehicle. Due to those missed counts, the number of generated vehicles is too low. The proposed simulation matches the other counts. This simulation will generate more vehicles which could result in a deterioration of the performance because every extra vehicle can only increase the total travel time delay (and not decrease).

4.3. Definition of tested controllers and performance index

4.3.1. Baseline and proposed controllers

The contributions in Section 2.5 are combined in two proposed controllers. Table 4.5 shows all properties of the baseline and proposed methods.

Method	Performance	Data-driven	Real world	Horizon	GLOSA
B-CCOL	0	No	Yes	0	No
B-Perfect	++	No	No	∞	Yes
B-DIRECTOR	-	B-LSTM-DS19	Yes	≤ 20 s	No
P-HOR	- / 0 (?)	P-LSTM-DS19+PP+SS+Q+A	Yes	> 20 s	No
P-HOR+G	0 / + (?)	P-LSTM-DS19+PP+SS+Q+A	Yes	> 20 s	Yes

Table 4.5: Properties of all baseline (B) and proposed (P) methods. The contributions mentioned in Section 2.5 results in two proposed controller. The first goal is to extend the control horizon with optimal incorporation of the prediction model (P-HOR). The final goal is to improve the performance by enabling GLOSA (P-HOR+G).

The first baseline method is suitable for real-world application but is still non-predictive. In Table 4.5 this method is named B-CCOL (1st column). B-CCOL is a state-of-the-art controller that is currently operating on Dutch roads. The performance (2nd column) of B-CCOL is used as the baseline performance and therefore has the value 0. This controller is handcrafted to optimize the performance for that specific intersection and therefore it is not data-driven (3rd column). B-CCOL does not require perfect information of the surroundings and therefore it is suitable for real-world application (4th column). This controller has no prediction and control horizon (5th column) and therefore it is unable to enable GLOSA (6th column).

The second baseline method is predictive but not suitable for real-world application. This type is named B-Perfect in Table 4.5. The performance is high but it needs full information about and full control over the arriving vehicles and therefore it can have an prediction and control horizon of infinity (in theory). The computing power is in general the limiting factor. With this information and the ability to control the vehicles it is possible to enable GLOSA causing the high performance. In this case, GLOSA is not an advice but the speed at which the vehicle is controlled. It is not possible to test this controller because this information is not available in the simulation.

DIRECTOR is a controller that has predictive abilities and is still suitable for real-world application. In Table 4.5 this controller is named B-DIRECTOR. DIRECTOR detects the vehicles when they leave the preceding intersection. The future arriving vehicles are predicted with the use of a neural network predictions. There are already done some experiments to enable GLOSA but the performance deteriorated. Therefore, in Table 4.5 it is still noted that DIRECTOR cannot enable GLOSA to improve the performance. DIRECTOR is seen as the starting point to tighten the gap between the two controllers mentioned above.

P-HOR is separated into two methods which are explained in Chapter 3. Both proposed controllers uses the proposed prediction model as explained in Section 3.3. Section 3.4 explains the proposed method that incorporates the prediction model and makes the structure of the controller suitable to enable GLOSA. This will result in the proposed controller P-HOR in Table 4.5. With reliable predictions it is possible to determine the time until green for all directions based on the (preliminary) time plan which can be used to enable GLOSA. Section 3.5 explains the GLOSA system to design the proposed controller P-HOR+G in Table 4.5.

4.3.2. Performance index

The performance is expressed in the KPIs explained in Chapter 1. The first KPI is the total vehicle delay. Comparing this indicator between two different data sets (with a different traffic volume) will not give a realistic overview. Therefore, the average travel time delay is used by simply dividing the total vehicle delay by the total number of vehicles as given by

$$t_{avg}^{delay} = \sum_{a \in A_c} \frac{1}{N_a} \left(\sum_{n=1}^{N_a} (t_n^{travel}) - N_a \cdot t_a^{freeflow} \right), \tag{4.3}$$

where N_a is the number of generated vehicles on link a during the full simulation, t_n^{travel} is travel time that is actually experienced by vehicle n on link a while going across intersection c and $t_a^{freeflow}$ is the travel time this vehicle would have experienced in the absence of traffic signal control.

The second KPI is the total number of stops, also this KPI is divided by the total number of vehicles as given by

$$s_{avg} = \sum_{a \in A_c} \frac{1}{N_a} \sum_{n=1}^{N_a} s_n, \tag{4.4}$$

where s_n is the number of stops of vehicle n.

4.4. Performance of proposed controller

This section shows the results of the simulations performed to test the proposed controller without GLOSA (P-HOR) versus the baseline controllers (B-CCOL & B-DIRECTOR). This is done in three steps:

1. Subsection 4.4.1 shows the results obtained in the master thesis of Van Senden that will give the performance of DIRECTOR. The performance is based on the baseline simulation and the traffic volume is based on the data set of 2017.
2. Subsection 4.4.2 shows the results of the same controller but in the proposed simulation and the traffic volume is based on the data set of 2019.
3. Subsection 4.4.3 shows the results of P-HOR. These results have the same circumstances as the previous results: proposed simulation and the data set of 2019. P-HOR can operate in more scheduling modes which fixes the schedule more ahead in time.

4.4.1. Results of baseline controller in baseline simulation - DIRECTOR and CCOL

This subsection shows the results obtained in the master thesis of Van Senden of the performance of DIRECTOR and CCOL in the baseline simulation. Table 4.6 shows six different controllers. The upper four rows are all versions of DIRECTOR but with different settings. The 1st column indicates the used prediction model. The baseline LSTM prediction model is compared to a prediction model with perfect predictions to give insight in the importance of good predictions. The 2nd column indicated the length of the fixed control horizon, i.e. the scheduling mode that fixes the number of time bins. A scheduling mode with 0 time bins fixed gives a fixed control horizon of 0 - 10 seconds. At the start of the interval the schedule for the next 10 seconds is determined. The fixed control horizon will decrease to 0 seconds during this interval after which it again is determined for another 10 seconds. A scheduling mode with 1 time bins fixed gives a fixed control horizon of 10 - 20 seconds, etc. The versions in the first two rows have no fixed time bins and the version in the next two rows have one time bin fixed. The performance of DIRECTOR with zero time bins fixed is slightly better than the current on-street implementation. However, the performance decreases when one time bin is fixed. Perfect predictions countered this negative effect. The last two rows shows controllers that are currently used on Dutch roads. The current on-street controller is a hand-crafted actuated controller with green-wave coordination. CCOL is the same controller but without the green-wave coordination.

Prediction type	Fixed control horizon	Avg. delay / vehicle	Avg. # stops / vehicle
Perfect	0 - 10 seconds	**9.5**	0.40
B-LSTM-DS17	0 - 10 seconds	10.3	0.46
Perfect	10 - 20 seconds	10.4	**0.36**
B-LSTM-DS17	10 - 20 seconds	12.9	0.43
Current on-street implementation		10.4	0.40
Vehicle actuated (CCOL)		16.8	0.43

Table 4.6: Performance of DIRECTOR versus state-of-the-art controllers in the baseline simulation [80]. Best performance highlighted in bold. DIRECTOR outperforms the current on-street controller using perfect predictions. The performance is slightly worse using the LSTM prediction model. Extension of the fixed control horizon will increase the vehicle delay but decrease the number of stops.

4.4.2. Results of baseline controller in the proposed simulation - DIRECTOR and CCOL

The baseline controllers are also tested in the proposed simulation. Changes in the simulation could influence the performance of the controller. For all simulations below, all the conditions are equal except the tested controller. Therefore, the tested controller is the only difference between two experiments which will give a good comparison between the baseline and proposed controllers. The difference in KPIs is only influenced by the behavior of the controller. Table 4.7 shows the results of the baseline controllers in the proposed simulation. The three tested controllers have the same configuration as respectively the 1st, 3rd and 5th row in Table 4.6. The only adjustments to the controller are some back end changes to make the controller work in the proposed simulation setup.

Control method	Fixed control horizon	Avg. delay / vehicle	Avg. # stops / vehicle
DIRECTOR	0 - 10 seconds	23.7	0.87
DIRECTOR	10 - 20 seconds	17.5	0.62
Vehicle actuated (CCOL)		**12.1**	**0.54**

Table 4.7: Performance of DIRECTOR and CCOL in the proposed simulation model. Best performance highlighted in bold. DIRECTOR is tested in both scheduling modes that affects the length of the fixed control horizon. CCOL is the controller with a overall highest performance.

The performance of DIRECTOR deteriorated compared to the baseline simulation in Table 4.6. During the simulation, it is seen that DIRECTOR often falls back on stabilization mechanisms during high traffic flows. The most used stabilization mechanism is the check for long queues. This mechanism is activated when the arrival detector is occupied for a long time and the signal group is scheduled to clear the long queue, i.e. the stabilization mechanism will schedule the signal group until the long queue is gone.

When the vehicles are generated on the arrival detector, the vehicles in the long queue are all placed (unrealistically) on top of the arrival detector. In this scenario, the arrival detector remained occupied until the last vehicle departed and the long queue is resolved. The proposed simulation setup generates the vehicles on the stop line detector of the preceding intersection and all vehicles in the long queue will stop behind the arrival detector if there is a long queue. This difference leads to inefficient control of the stabilization mechanism.

The congested traffic starts to drive after the stabilization mechanism is activated. Due to the congestion on the double lane arrival detector, the detector was constantly occupied causing the inability to count the inflow of the queue. The stop line detector was still able to properly count the outflow because it is a single lane detector. Once the vehicles have gained speed, the arrival detector will be free between vehicles and the long queue seems fixed. The result is that DIRECTOR measures an empty queue but actually the queue reaches over the arrival detector. DIRECTOR will service another signal group because the queue seems empty. When all vehicles are back in line, the arrival detector is again occupied for a long time and again the stabilization mechanism will kick back in which results in inefficient control.

An obvious solution is to not reset the stabilization mechanism if the arrival detector is free for a short time. The long queue remains and the signal group stays scheduled if short free periods in between are accepted. The downside is that the stabilization mechanism can act when there is no long queue. During busy hours the arrival detector is occupied most of the time which makes it hard to distinguish a long queue or just normal driving vehicles. There are methods that makes the long queue stabilization mechanism less used or not used at all such as:

- The optimal controller will not need any stabilization mechanisms. High traffic density can be recognized based on the predicted arrivals. The use of the stabilization mechanism is prevented if the predictions are accurate and the control capability of P-HOR is sufficient.

- An extra priority weight is added to determine the priority of the signal group. Increasing the weight exponentially for waiting vehicles over time will ensure that the signal group is serviced before the stabilization mechanism is activated.

The performance of CCOL improved compared to the results of Table 4.6. The exact configuration of the CCOl controller of Van Senden is unknown [80]. Possibly, the CCOL controller of this subsection is more similar to the on-street controller than the CCOL controller in the previous subsection. There are a lot of adjustments between the baseline and proposed controller which influences the simulation. The adjustments given in Subsection 3.6.4 could also deteriorate the performance. As explained before, generating vehicles on the stop line detector results in higher traffic intensity. The effect on CCOL is shown with simulations in different circumstances.

- An average travel time delay per vehicle of 16.8 seconds for the baseline simulation setup used in Table 4.6.
- An average travel time delay per vehicle of 12.1 seconds for the proposed simulation setup used in Table 4.7.
- An average travel time delay per vehicle of 8.4 seconds for a simulation setup in which vehicles are generated on the arrival detector but all other changes remain as explained for the proposed simulation setup.

We can conclude that higher traffic intensity will result in worse performance of the controller. This will probably also affect the upcoming results causing worse performance of the tested controllers in the proposed simulation setup.

4.4.3. Results of proposed controller

This subsection shows the performance of P-HOR. All used methods are explained in Chapter 3. The proposed methods for the prediction model and the controller are summarized in short below:

- The prediction model is extended with extra input features and appropriate pre-processing steps are used. The outputs of the prediction model are the expected vehicles per time bin on the arrival detectors. The length of the prediction horizon is free to choose.

- The extended prediction model is incorporated in P-HOR. The control horizon is extended and uses multiple prediction models (one for every time bin) to increase the quality of the predictions. The length of the predicted control horizon is disconnected from the scheduling mode, i.e. the fixed control horizon, to enable GLOSA.

Table 4.7 is extended with the simulation results of P-HOR which results in Table 4.8. This table shows the results of the baseline versus the proposed method when the GLOSA systems is disabled. P-HOR can operate in more scheduling modes. Two and three time bins fixed will increase the fixed control horizon up to 40 seconds which increases the predictability.

Control method	Fixed control horizon	Avg. delay / vehicle	Avg. # stops / vehicle
DIRECTOR	0 - 10 seconds	23.7	0.87
DIRECTOR	10 - 20 seconds	17.5	0.62
Vehicle actuated (CCOL)		**12.1**	**0.54**
P-HOR	0 - 10 seconds	14.6	0.59
P-HOR	10 - 20 seconds	15.0	0.57
P-HOR	20 - 30 seconds	17.4	0.64
P-HOR	30 - 40 seconds	20.2	0.65

Table 4.8: Performance of the proposed controller P-HOR compared to the baseline methods. Best performance highlighted in bold. P-HOR can change the scheduling mode to increase the fixed control horizon. This increases the predictability which can be used to enable GLOSA.

Several things can be concluded based on Table 4.8:

- P-HOR improved compared to DIRECTOR. The gain is probably due to the inability of the stabilization mechanism in the proposed simulation. This mechanisms was not effective as explained in Subsection 4.4.2.
- P-HOR is still worse compared to CCOL. CCOL is a hand-crafted controller to optimal perform on this specific intersection and uses last minute changes in the schedule to maximize the performance, i.e. a fixed control horizon of 0 seconds. The drawback of CCOL is that needs to be tuned every 2 - 3 years and has no predictability at all. Predictability should provide benefits when GLOSA is enabled.
- The performance of P-HOR with 0 and 1 time bins fixed are equivalent. The overall performance of the controller depends on the chosen importance of both KPIs.
- The performance of both KPIs deteriorated when increasing the number of fixed time bins further. This is because it is not allowed to change it to a more beneficial schedule once the schedule is fixed. Enabling GLOSA could counter this effect.

4.5. Performance of proposed controller with GLOSA enabled

The last experiment tests the effect of enabling the GLOSA system on the performance. The GLOSA system will provides the optimal speed for the arriving vehicles to reduce the number of stops. GLOSA systems should not only take into account the desired speed to arrive during green phases but also the optimal time within these periods. When the road is crowded, the system needs to ensure that all vehicles arrive scattered during the green light period instead of all at the begin or end. Subsection 4.5.1 shows the results of the preliminary experiments to test the performance of DIRECTOR with GLOSA enabled in the baseline simulation. Subsection 4.5.2 shows the result of enabling GLOSA for the proposed controller in the proposed simulation.

4.5.1. Results of baseline controller with GLOSA enabled - DIRECTOR and CCOL

The GLOSA system can be enabled if the controller has a control horizon that shows the schedule for the near future. The schedule is used to determine the time until green and the corresponding speed. Therefore, it is not possible for the CCOL controller to enable GLOSA as seen in Figure 3.13. DIRECTOR can be used as baseline controller with GLOSA enabled. In this version of GLOSA the speed is constantly updated to achieve the highest performance. However, this is currently not possible for on-street operation because not all vehicles are connected. GLOSA is determined for vehicles that crossed the arrival detector because that was the outer edge of the simulation for the nearby simulation. The control horizon of DIRECTOR is limited to one time bin fixed which is long enough to determine the optimal speed for vehicles generated on the arrival detector. Table 4.9 shows the results of DIRECTOR with GLOSA enabled.

The average travel time per vehicle is used as KPI instead of the average delay because Vissim (that the baseline simulation performed) could not calculate the correct average delay per vehicle. GLOSA changes the target speed for vehicles which was used to determine the delay time. The average delay per vehicles is the delay compared to the maximum speed, not the optimal speed (GLOSA). The first two scenarios shows the performance with the GLOSA system disabled which are the same as scenario 2 and 4 in Table 4.6 but converted to the new performance index. The performance deteriorates with one time bin is fixed. However, this scheduling mode is designed to enable Advanced Driver-Assistance Systems (ADAS) such as GLOSA as shown in scenario 3.

Control method	Fixed control horizon	GLOSA	Average travel time / vehicle	Average # stops / vehicle
DIRECTOR	0 - 10 seconds	N/A	**25.4**	0.50
DIRECTOR	10 - 20 seconds	Disabled	29.0	0.50
DIRECTOR	10 - 20 seconds	Enabled	28.6	**0.36**

Table 4.9: Performance of DIRECTOR enabling a simple version of GLOSA in the baseline simulation model. Best performance highlighted in bold. The implementation assumes that there are no vehicles queued and that acceleration and deceleration is instant [80]. Enabling GLOSA reduces the number of stops with almost 30% for this implementation.

GLOSA has little effect on the travel time but can reduce the number of stops with almost 30% for DIRECTOR. There is an inexplicable difference in these results obtained by van Senden between row 2 and 4 of Table 4.6 and row 1 and 2 of Table 4.9 which are the same controllers with the same scheduling mode in the same simulation setup.

- The number of stops increases from 0.46 and 0.43 in Table 4.6 to both 0.50 Table 4.9.
- The delay time increased from 10.3 to 12.9 (increase of 2.6) with one time bin fixed but the travel time increased from 25.4 to 29.0 (increase of 3.6). As explained in Equation 4.3, the difference between travel time delay and travel time is the travel time in free flow which is a constant. Therefore, the difference in both tables should remain the same.

This makes it difficult to compare the performance of P-HOR+G with the results of Van Senden. Therefore, the effect of enabling the GLOSA system is mainly compared to the performance of the proposed controller with GLOSA disabled P-HOR+G.

4.5.2. Results of proposed controller with GLOSA enabled

Enabling GLOSA is the last proposed method explained in Chapter 3. The GLOSA system determines the optimal speed for the vehicles leaving the preceding intersection. Table 4.10 shows the performance of P-HOR+G.

Control method	Fixed control horizon	Average travel time / vehicle	Average # stops / vehicle
P-HOR+G	0 - 10 seconds	16.9	0.62
P-HOR+G	10 - 20 seconds	15.5	0.43
P-HOR+G	20 - 30 seconds	16.4	0.43
P-HOR+G	30 - 40 seconds	16.0	**0.39**
Vehicle actuated (CCOL)		**12.1**	0.54

Table 4.10: Performance of the proposed controller after enabling GLOSA in the proposed simulation model. All vehicles received GLOSA. Best performance highlighted in bold. The CCOL controller has still the lowest vehicle delay. However, the proposed controller with a scheduling mode of 3 time bins fixed and GLOSA enabled has the least number of stops.

Table A.1 in the Appendix contains an overview of the results of the proposed methods with both GLOSA enabled and disabled and the baseline methods. Based on Table 4.10 several things can be concluded:

- Enabling GLOSA in the scheduling mode with 0 time bins fixed results is an overall worse controller. The system determines GLOSA based on a schedule that still can change. Therefore, the GLOSA speed is not optimal anymore in many cases.
- Enabling GLOSA with 1 time bin fixed has more promising results. The travel time delay increased with 3 - 6% but the number of stops decreased with 25 - 27% compared to the most promising scheduling modes with GLOSA disabled (0 or 1 time bins fixed, based on which KPI is the most important).
- Fixing more time bins will improve both KPIs compared to the same scheduling mode with GLOSA disabled. This is mainly because the version with GLOSA disabled has a poor performance. Nevertheless, the travel time delay decreased up to 20% and the number of stops up to 40%. Compared to the most promising scheduling modes the travel time delay increased with 7 - 12% and the number of stops decreased with 25 - 34%.
- The performance deteriorated if more time bins are fixed in all previous experiments. In Table 4.10 the travel time delay decreased while the number of fixed time bins increased from 2 to 3 time bins fixed.

The GLOSA system will not give similar results during the entire simulation which is seen in Appendix B. Figure B.1 shows the vehicle delay and Figure B.2 the number of stops per 15 minutes over time. CCOL has the lowest vehicle delay during the entire simulation. Compared with both DIRECTOR and P-HOR, the difference in vehicle delay is bigger during the morning and evening rush hour. However, P-HOR+G has the lowest number of stops. The GLOSA system reduces the number of stops especially during the less busy hours and has no effect at all during the evening rush hour. During these hours, it is more crowded making it unable for vehicles to drive the given target speed. This is especially the case when there is a big difference in target speed for vehicles on the same link with different destinations. Another explanation is that the delay due to the queue is based on the queue length estimated by P-HOR+G. Wrong estimation of the queue length affects the optimal speed.

The GLOSA system is not yet capable to perfectly determine the speed. This affects the overall performance which is the most visible during the busiest hours. It is possible to switch between scheduling modes during the day because the proposed controller is not installed on the TLC but is running in the cloud. This makes it possible to operate without GLOSA during the busiest moments and enable GLOSA the remainder of the day. The performance over a full day is improved to 14.5 seconds of travel time delay per vehicle and a average of 0.39 stops per vehicle if the controller operates in the scheduling mode with zero time bins fixed and GLOSA disabled during the evening rush hour (between 15:30 - 17:00) and the scheduling mode with 3 time bins fixed and GLOSA enabled during the rest of the day. Disable GLOSA also during the morning rush hour will even further decrease the travel time delay. This will result in a travel time delay of 14.0 seconds per vehicle and an average number of stops per vehicle of 0.42. Both configurations have a higher performance on both KPIs compared to the best performing controller with GLOSA disabled as seen in Table 4.11.

Control method	Fixed control horizon	GLOSA	Average travel time / vehicle	Average # stops / vehicle
CCOL	0 seconds	disabled	**12.1**	0.54
P-HOR+G*	0 - 40 seconds	partly enabled	14.5	**0.39**
P-HOR+G**	0 - 40 seconds	partly enabled	14.0	0.42

Table 4.11: Performance of the proposed method that disables GLOSA during the most busy moments. The remainder of the day GLOSA is enabled and the schedule is known for 30 - 40 seconds. The experiments are performed in the proposed simulation. Best performance highlighted in bold.
*) GLOSA is disabled during the evening rush hour.
**) GLOSA is disabled during the morning and evening rush hour.

We can conclude that enabling GLOSA will not directly give an overall performance improvement. However, enabling GLOSA at the right moments will improve the overall performance. The most effective moments to disable GLOSA is during the busy moments such as the morning and evening rush hour.

5

Conclusions and future work

Nowadays traffic jams have an effect on most people in daily life. The bottleneck of the maximum road volume in urban areas, and a major reason for the occurrence of congestion in traffic, is the maximum capacity of the traffic flow on the intersection. Safe passage for all vehicles on the intersection is coordinated with Traffic Light Controllers (TLCs). There are multiple control-methods trying to achieve the optimal control based on the available resources. Despite all effort to minimize the delay for vehicles, it is still common to wait regularly before a red traffic light.

A promising method to decrease the number of stops are Green Light Optimal Speed Advice (GLOSA) systems. These systems will give a speed advice to arriving vehicles based on the schedule of the TLC. The TLC needs to be predictable to give reliable GLOSA messages, i.e. the schedule of the TLC needs to be known and fixed. However, most on-street controllers are flexible to maximize the performance up until the last moment. Currently, enabling GLOSA systems is only possible in two situations:

- Non-dynamic controllers that have a predetermined schedule and cannot adapt to a changing traffic flow
- Controllers that make unrealistic assumptions on complete knowledge of the traffic situation and are not suitable for real-world application.

A prediction model could predict the future arrivals based on available measurements to optimize and fix the schedule in advance. Controllers that predict the arrivals in advance to enable GLOSA could potentially improve the performance and are still suitable for real-world application. The proposed controller developed in this thesis, will investigate the possible performance gain by answering the following research question: *How much can the performance of dynamic decentralized traffic light controllers be improved on sparse real measurements by enabling GLOSA?*

The proposed controller is tested in a simulation which mimics real-world circumstances. The configuration and the traffic flow data are based on real life detections obtained from the managing provincial government. The proposed controller is a decentralized controller which means that it will control one intersection. The design of the controller and the setup of the simulation is in such a way that the proposed controller can be connected to operating TLCs without any adjustments.

5.1. Prediction model

The first goal was to optimize the prediction model for arriving vehicles. Appropriate pre-processing steps are implemented to boost the performance. All detection data is stationary over time, i.e. remove the trend and make the input feature independent over time, by using the differenced series. The time of day data is cyclical to remove the undesirable jumps during midnight. The day of week data is divided is workdays and weekend days to create a binary input.

Extra input features of the preceding TLC are used to increase the prediction horizon. The combination of queue detectors, arrival detectors and signal states of the preceding intersection as extra input besides the already used stop line detectors maximized the performance. All input features are included in the V-log standard which is used by operating TLCs on-street.

The result is that the prediction horizon could be extended until 50 seconds without loss of accuracy compared with the prediction model of DIRECTOR with a prediction horizon of 30 seconds. Over the same prediction horizon of 30 seconds, the NRMSE decreased (from 0.65 to 0.54) with 17%.

Interesting to investigate as future work is to use the output of the GLOSA system as input for the prediction model. The model predicts the future arrivals based on a data set where vehicles drive a target speed of 80 km/h. Due to the GLOSA system, the overall speed of vehicles changes. The prediction model can incorporate the changed behavior of the vehicles due to the GLOSA system by using the target speed of the GLOSA system as extra input. A data set with GLOSA information is therefore needed that is unfortunately not (yet) available. An option is to collect all detections and the output of the GLOSA system during a simulation and use this data set to train the prediction model. However, the prediction model is trained on the simulated data which has an unknown effect.

5.2. Predictive controller

The prediction model is used to design a predictive controller that is suitable to enable the GLOSA system. Table 4.5 describes the contributions of this thesis. Two controllers are used as baseline methods to compare the performance of the proposed controller. CCOL is a non-predictive hand-crafted controller that operates on-street and DIRECTOR is the first step towards on-street predictable controllers. The structure of DIRECTOR is used to design the proposed controller ($P - HOR$). The proposed controller extends the control horizon and uses multiple prediction models to predict the arrivals for every time bin within the control horizon.

The proposed controller outperforms DIRECTOR with 14 - 38% reduction in terms of vehicle delay and 5 - 32% reduction in terms of numbers of stops based on the scheduling mode. DIRECTOR is not yet able to optimal control under the circumstances in the simulation, therefore, the gain is probably due to the inability of the stabilization mechanism. The proposed controller is not competitive compared to the non-predictive CCOL controller.

The proposed controller has extended the control horizon to increase the predictability. Experiments are performed with scheduling modes up until a fixed control horizon of 40 seconds. As expected, the performance deteriorates because of the reduction of flexibility. The predictability of these scheduling modes is used to enable the GLOSA system. Several things are still worth investigating as future work to improve the performance of the controller. Small adjustments such as optimization of time bin size, optimal weight of the switching penalty and the implementation of a dynamic model to predict the departures will likely improve the performance. There is also some potential left in the selection of the optimal schedule. The optimal schedule is selected based on the vehicle delay during that time bin. All travel time delay after this time bin is not included in the optimization. An optimization over the full control horizon at once with possible switches in between could improve the performance because the effect on the next schedule is incorporated in the optimization. In this case, the total vehicle delay for a combination of schedules in the control horizon is determined at once and the best combination of schedules is picked. It could be beneficial to schedule a signal group with less priority first such that afterwards the signal group with the highest priority could be serviced for a longer time.

5.3. Enabling GLOSA

The GLOSA system is an add-on of the controller and the controller must also be able to operate without the GLOSA system. In Table 4.5 this controller is referred to as $P - HOR + G$. The control horizon of the proposed controller always has a fixed length which is needed to determine the time until green. The implemented GLOSA system will determine the optimal speed based on the time until green and the expected delay due to the surrounding vehicles.

Enabling GLOSA will not lead to an overall improvement of the performance directly. The number of stops decreased but the travel time delay increased. Only when comparing between the same scheduling modes with a fixed schedule more than 20 seconds, the overall performance improved. However, this is mainly due to the bad performance with GLOSA disabled.

The proposed controller is a cloud controlled application. Therefore, it is possible to adjust the scheduling modes during the day. Enabling GLOSA all day except during rush hours will lead to 3 - 4% reduction in terms of vehicle delay and 29 - 32% reduction in terms of numbers of stops based on the scheduling mode. This setup of the proposed controller seems competitive with the hand-crafted non-predictive CCOL controller. Compared with this controller, the proposed controller will reduce the number of stops with 26% at the cost of 16% increase in vehicle delay. The performance of all tested controllers is shown in Table A.1.

Despite the fact that the GLOSA system already improved the performance, there is still progress to be made. GLOSA is only determined at the moment the vehicles leave the preceding schedule. Based on new insights in the expected time until green or the expected queue length, it could be beneficial to update GLOSA. Also, the proposed method does not take speed differences into account. Vehicles on the same link with a different destination could receive a different GLOSA as big as the difference between minimum and maximum boundary of GLOSA which can cause dangerous situations. Safety would be improved when the speed differences are minimized for each link. Also, in the research it is assumed that all vehicles will change their speed according to the GLOSA system. In reality, the number of vehicles that will change their speed differs because some vehicles will still drive the speed limit. It is expected that the participation will increase with the rise of automated vehicles. However, it is important to the test the effects of GLOSA with less than 100% participation to have a realistic view. Simplified tests with 80% participation are performed, i.e. 80% of the vehicles are randomly selected and advised with GLOSA. This is tested for the scheduling mode with 1 and 2 time bins fixed. The results are shown in Table 5.1 below. These results are also included in Table A.1.

Control method	Fixed control horizon	GLOSA	Average travel time / vehicle	Average # stops / vehicle
P-HOR+G	10 - 20 seconds	80% enabled	15.5	0.56
P-HOR+G	20 - 30 seconds	80% enabled	**15.3**	**0.52**

Table 5.1: Results of simulations where randomly 80% of the vehicles received GLOSA. Best performance highlighted in bold. The results with a horizon of 10 - 20 seconds is equal to the setup with GLOSA disabled. The number of stops with a horizon of 10 - 20 seconds is between the number of stops with GLOSA enabled and disabled. The average vehicle delay is over 1 seconds per vehicle lower compared to GLOSA disabled and over 2 seconds lower compared to GLOSA enabled.

Several things can be concluded based on Table 5.1:

- The performance is equal when GLOSA is disabled or 80% enabled for the scheduling mode with one time bin fixed.
- Enabling GLOSA for 80% for the scheduling mode with two time bins fixed gives a better result. The number of stops is between the number of stops with GLOSA enabled and disabled. The average vehicle delay is over 1 second per vehicle lower compared to GLOSA disabled and over 2 seconds lower compared to GLOSA enabled.

Finally, all statements within this thesis are based on the results of simulation. Despite this simulation is designed to mimic real-world circumstances, it is not possible to say with certainty that the controller will perform equivalently on street. As already mentioned, the proposed controller is designed conform the safety standards used on-street. Together with a team at Siemens, we convinced the provincial government of the safety and quality of the proposed controller and received permission from the provincial government to do on-street pilots with the proposed controller on the intersection as shown in Figure 1.3. Due to the safety reasons described above, GLOSA is not (yet) enabled in these pilots. At the time of writing, the last adaptions are realized to connect the controller to the TLC. The pilots will be done in small test periods at a time. The results could give new insights in the performance and is a check on the truthfulness of the simulation. The objective to get the developed controller on street is accomplished. That does not mean that no more improvements can be made. An improvement that can be obtained with minimal effort and certainly gives an performance improvement is the optimization of the time bin size that is currently fixed for 10 seconds. More extensive research about the optimization of the scheduling over multiple time bins, as explained above, could lead to significant performance improvements.

A

Overview of all simulation results

Control method	Fixed control horizon	GLOSA	Average travel time / vehicle	Average # stops / vehicle
CCOL	0 seconds	disabled	**12.1**	0.54
DIRECTOR	0 - 10 seconds	disabled	23.7	0.87
P-HOR	0 - 10 seconds	disabled	14.6	0.59
P-HOR+G	0 - 10 seconds	enabled	16.9	0.62
DIRECTOR	10 - 20 seconds	disabled	17.5	0.62
P-HOR	10 - 20 seconds	disabled	15.0	0.57
P-HOR+G	10 - 20 seconds	enabled	15.5	0.43
P-HOR+G	10 - 20 seconds	80% enabled	15.5	0.56
P-HOR	20 - 30 seconds	disabled	17.4	0.64
P-HOR+G	20 - 30 seconds	enabled	16.4	0.43
P-HOR+G	20 - 30 seconds	80% enabled	15.3	0.52
P-HOR	30 - 40 seconds	disabled	20.2	0.65
P-HOR+G	30 - 40 seconds	enabled	16.0	**0.39**
P-HOR+G*	0 - 40 seconds	partly enabled	14.5	**0.39**
P-HOR+G**	0 - 40 seconds	partly enabled	14.0	0.42

Table A.1: Performance of all tested methods in the proposed simulation. Best performance highlighted in bold. The last two proposed methods disable GLOSA during the most busy moments. The remainder of the day GLOSA is enabled and the schedule is known for 30 - 40 seconds.
*) GLOSA is disabled during the evening rush hour.
**) GLOSA is disabled during the morning and evening rush hour.

B

Performance over time

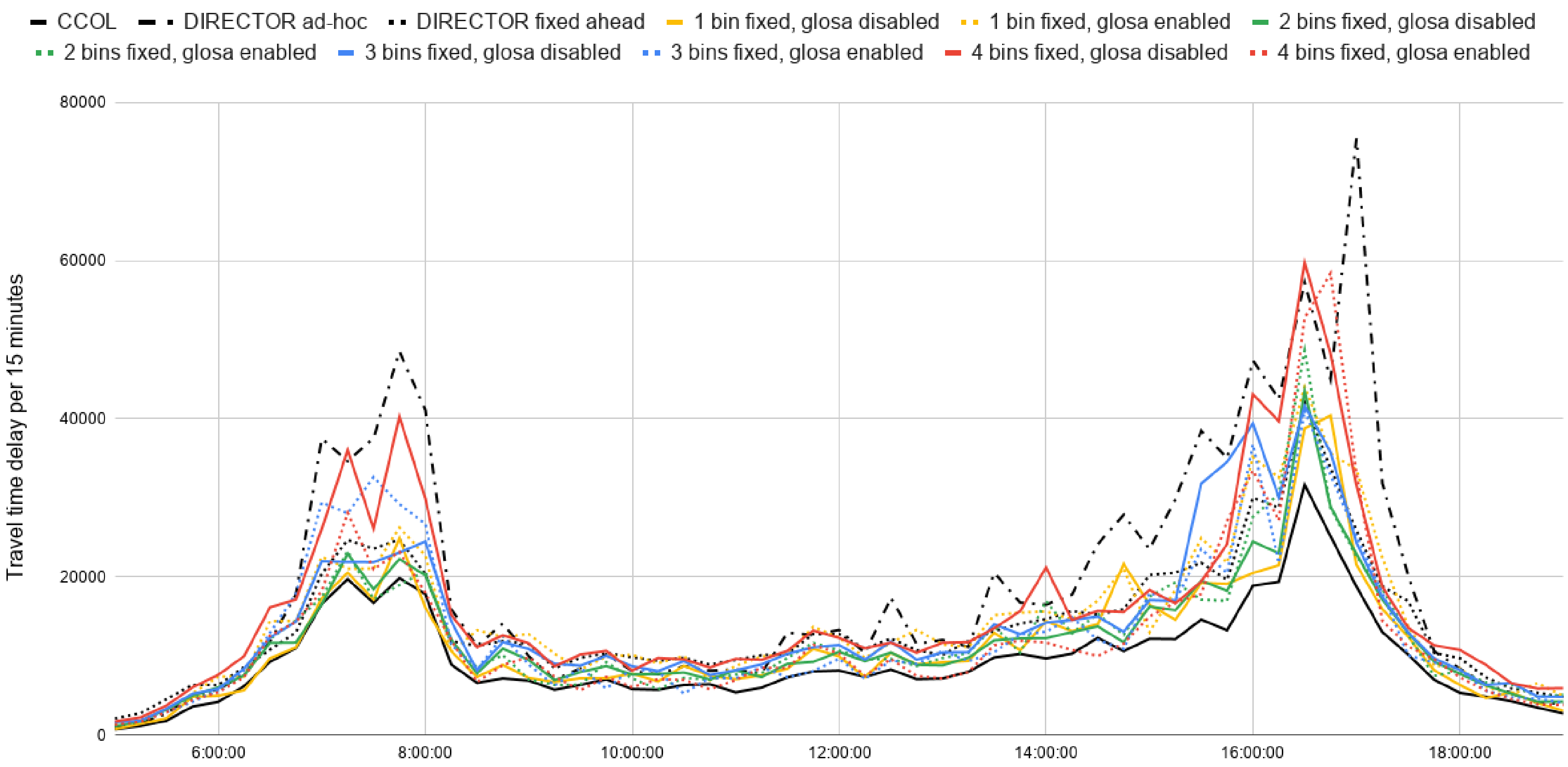


Figure B.1: Travel time delay over time for the proposed controller to see the effect of GLOSA

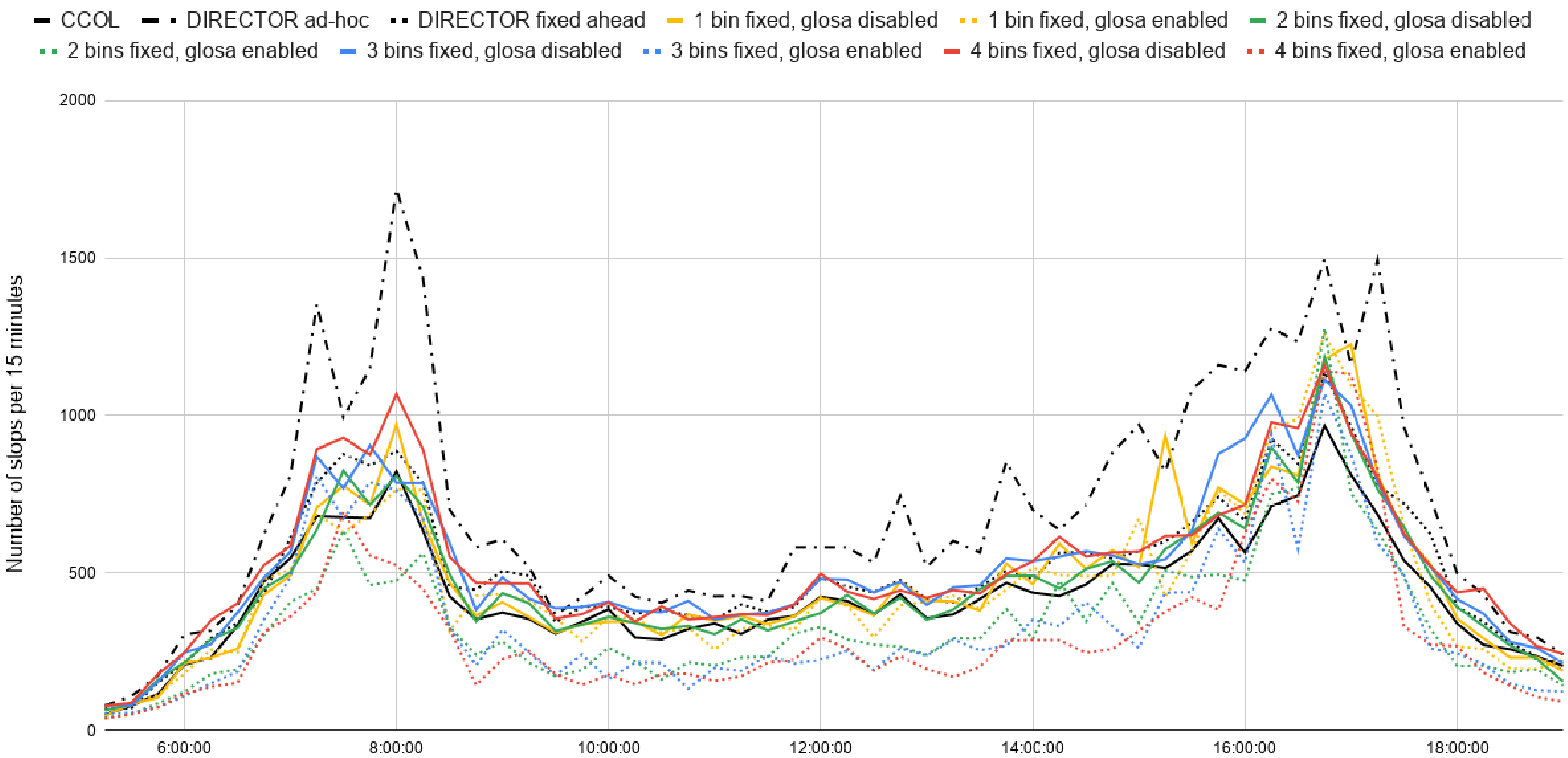


Figure B.2: Number of stops over time for the proposed controller to see the effect of GLOSA

Bibliography

[1] R. Akcelik. The highway capacity manual delay formula for signalized intersections. *ITE J.; (United States)*, 58(3), Mar 1988.

[2] H. Anany. Effectiveness of a speed advisory traffic signal system for conventional and automated vehicles in a smart city, 2019.

[3] R. L. Anderson. Electromagnetic loop vehicle detectors. *IEEE Transactions on Vehicular Technology*, 19(1):23–30, Feb 1970. ISSN 0018-9545. doi: 10.1109/T-VT.1970.23428.

[4] I. Arel and T Urbanik. U18: Traffic signal safety (phase b). *National Transportation Research Center*, 08 2009.

[5] B. Asadi and A. Vahidi. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE Transactions on Control Systems Technology*, 19(3):707–714, May 2011. ISSN 1063-6536. doi: 10.1109/TCST.2010.2047860.

[6] E. Azimirad, N. Pariz, and M. B. N. Sistani. A novel fuzzy model and control of single intersection at urban traffic network. *IEEE Systems Journal*, 4(1):107–111, March 2010. ISSN 1932-8184. doi: 10.1109/JSYST.2010.2043159.

[7] J. L. Barker. Radar, acoustic, and magnetic vehicle detectors. *IEEE Transactions on Vehicular Technology*, 19(1):30–43, Feb 1970. ISSN 0018-9545. doi: 10.1109/T-VT.1970.23429.

[8] L. D. Baskar, B. De Schutter, and H. Hellendoorn. Model-based predictive traffic control for intelligent vehicles: Dynamic speed limits and dynamic lane allocation. In *2008 IEEE Intelligent Vehicles Symposium*, pages 174–179, June 2008. doi: 10.1109/IVS.2008.4621307.

[9] A. G. Beacher, M. D. Fontaine, and N.s J. Garber. Field evaluation of late merge traffic control in work zones. *Transportation Research Record*, 1911(1):33–41, 2005. doi: 10.1177/0361198105191100104. URL https://doi.org/10.1177/0361198105191100104.

[10] R. Blokpoel and W. Niebel. Advantage of cooperative traffic light control algorithms. *IET Intelligent Transport Systems*, 11(7):379–386, 2017.

[11] R. Bodenheimer, D. Eckhoff, and R. German. Glosa for adaptive traffic lights: Methods and evaluation. In *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 320–328, Oct 2015. doi: 10.1109/RNDM.2015.7325247.

[12] F. Boillot, S. Midenet, and J. Pierrelée. The real-time urban traffic control system cronos: Algorithm and experiments. *Transportation Research Part C: Emerging Technologies*, 14(1):18 – 38, 2006. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2006.05.001. URL http://www.sciencedirect.com/science/article/pii/S0968090X06000180.

[13] G. Chang, M. Vasudevan, and C. Su. Modelling and evaluation of adaptive bus-preemption control with and without automatic vehicle location systems. *Transportation Research Part A: Policy and Practice*, 30(4):251 – 268, 1996. ISSN 0965-8564. doi: https://doi.org/10.1016/0965-8564(95)00026-7. URL http://www.sciencedirect.com/science/article/pii/0965856495000267.

[14] S. Cools, C. Gershenson, and B. D'Hooghe. *Self-Organizing Traffic Lights: A Realistic Simulation*, pages 45–55. Springer London, London, 2013. ISBN 978-1-4471-5113-5. doi: 10.1007/978-1-4471-5113-5_3. URL https://doi.org/10.1007/978-1-4471-5113-5_3.

[15] Vialis Corporation. Wat is v-log?, 2016.

[16] Vialis Corporation. Systeembeschrijving toolkit ccol, 2016.

[17] C. de Fabritiis, R. Ragona, and G. Valenti. Traffic estimation and prediction based on real time floating car data. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 197–203, Oct 2008. doi: 10.1109/ITSC.2008.4732534.

[18] A. de La Fortelle. Analysis of reservation algorithms for cooperative planning at intersections. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 445–449, Sep. 2010. doi: 10.1109/ITSC.2010.5624978.

[19] F. Dion, H. Rakha, and Y. Kang. Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections. *Transportation Research Part B: Methodological*, 38(2):99 – 122, 2004. ISSN 0191-2615. doi: https://doi.org/10.1016/S0191-2615(03)00003-1. URL http://www.sciencedirect.com/science/article/pii/S0191261503000031.

[20] K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '04, pages 530–537, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 1-58113-864-4. doi: 10.1109/AAMAS.2004.190. URL https://doi.org/10.1109/AAMAS.2004.190.

[21] D. J. Fagnant and K. Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77: 167 – 181, 2015. ISSN 0965-8564. doi: https://doi.org/10.1016/j.tra.2015.04.003. URL http://www.sciencedirect.com/science/article/pii/S0965856415000804.

[22] M. Farzaneh, H. Rakha, A. Chair, M. Trani, K. Abbas, P. Ahn, K. November, and B. Virginia. *MODELING TRAFFIC DISPERSION*. PhD thesis, Virginia Polytechnic Institute and State University, 07 2019.

[23] E. D. Ferreira, E. Subrahmanian, and D. Manstetten. Intelligent agents in decentralized traffic control. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pages 705–709, Aug 2001. doi: 10.1109/ITSC.2001.948746.

[24] N. Goodall, B. Smith, and B. Park. Traffic signal control with connected vehicles. *Transportation Research Record Journal of the Transportation Research Board*, 2381:65–72, 12 2013. doi: 10.3141/2381-08.

[25] L. Haanstra. Improvement of road traffic sustainability by implementation of priority weights for trucks in predictive signalized intersection control. Master's thesis, Delft University of Technology, The Netherlands, 2019.

[26] D. Helbing. Improved fluid-dynamic model for vehicular traffic. *Phys. Rev. E*, 51:3164–3169, Apr 1995. doi: 10.1103/PhysRevE.51.3164. URL https://link.aps.org/doi/10.1103/PhysRevE.51.3164.

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

[28] J. J. Henry, J. L. Farges, and J. Tuffal. The prodyn real time traffic algorithm. *IFAC Proceedings Volumes*, 16(4):305 – 310, 1983. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)62577-1. URL http://www.sciencedirect.com/science/article/pii/S1474667017625771. 4th IFAC/IFIP/IFORS Conference on Control in Transportation Systems , Baden-Baden, FRG, 20-22 April 1983.

[29] A. G. Hobeika and C. K. Kim. Traffic-flow-prediction systems based on upstream traffic. In *Proceedings of VNIS'94 - 1994 Vehicle Navigation and Information Systems Conference*, pages 345–350, Aug 1994. doi: 10.1109/VNIS.1994.396815.

[30] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[31] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton. Scoot - a traffic responsive method of coordinating signals. Technical report, Transport and Road Research Laboratory (TRRL), 1981.

[32] T. M. Hussain, T. N. Saadawi, and S. A. Ahmed. Overhead infrared sensor for monitoring vehicular traffic. *IEEE Transactions on Vehicular Technology*, 42(4):477–483, Nov 1993. ISSN 0018-9545. doi: 10.1109/25.260764.

[33] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[34] K. Katsaros, R. Kernchen, M. Dianati, and D. Rieck. Performance study of a green light optimized speed advisory (glosa) application using an integrated cooperative its simulation platform. In *2011 7th International Wireless Communications and Mobile Computing Conference*, pages 918–923, July 2011. doi: 10.1109/IWCMC.2011.5982524.

[35] F. Knorr, D. Baselt, M. Schreckenberg, and M. Mauve. Reducing traffic jams via vanets. *IEEE Transactions on Vehicular Technology*, 61(8):3490–3498, Oct 2012. ISSN 0018-9545. doi: 10.1109/TVT.2012.2209690.

[36] F. Kurniawan, D. Dermawan, O. Dinaryanto, and M. Irawati. Pre-timed and coordinated traffic controller systems based on avr microcontroller. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 12:787, 12 2014. doi: 10.12928/telkomnika.v12i4.497.

[37] S. Lämmer and D. Helbing. Self-stabilizing decentralized signal control of realistic, saturated network traffic. *Technical Report*, 01 2011.

[38] M. Lèbre, F. E. Le Mouël, E. Ménard, A. Garnault, B. Bradaï, and V. Picron. Real scenario and simulations on GLOSA traffic light system for reduced CO2 emissions, waiting time and travel time. *CoRR*, abs/1506.01965, 2015. URL http://arxiv.org/abs/1506.01965.

[39] J. Lee and B. Park. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):81–90, March 2012. ISSN 1524-9050. doi: 10.1109/TITS.2011.2178836.

[40] M. J. Lighthill and G. B. Whitham. On kinematic waves ii. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178):317–345, 1955. doi: 10.1098/rspa.1955.0089. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1955.0089.

[41] S. Lin, B. de Schutter, Y. Xi, and H. Hellendoorn. Efficient network-wide model-based predictive control for urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 24: 122 – 140, 2012. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2012.02.003. URL http://www.sciencedirect.com/science/article/pii/S0968090X12000150.

[42] J. A. Lindley. Urban freeway congestion problems and solutions: an update. *iTE Journal*, 59(12): 21–23, 1989.

[43] H. Liu, D. Sun, and M. Zhao. A model prediction control based framework for optimization of signaled intersection: A cyber-physical perspective. *Optik*, 127(20):10068 – 10075, 2016. ISSN 0030-4026. doi: https://doi.org/10.1016/j.ijleo.2016.07.094. URL http://www.sciencedirect.com/science/article/pii/S0030402616308609.

[44] I. London. Encoding cyclical continuous features - 24-hour time, 2016.

[45] F. Luyanda, D. Gettman, L. Head, S. Shelby, D. Bullock, and P. Mirchandani. Acs-lite algorithmic architecture: Applying adaptive control system technology to closed-loop traffic signal control systems. *Transportation Research Record*, 1856(1):175–184, 2003. doi: 10.3141/1856-19. URL https://doi.org/10.3141/1856-19.

[46] L. Makarem and D. Gillet. Model predictive coordination of autonomous vehicles crossing intersections. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1799–1804, Oct 2013. doi: 10.1109/ITSC.2013.6728489.

[47] V. Mauro and C. Di Taranto. Utopia. *IFAC Proceedings Volumes*, 23(2):245 – 252, 1990. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)52678-6. URL http://www.sciencedirect.com/science/article/pii/S1474667017526786. IFAC/IFIP/IFORS Symposium on Control, Computers, Communications in Transportation, Paris, France, 19-21 September.

[48] H. McGee Sr, K. Moriarty, and T. J. Gates. Guidelines for timing yellow and red intervals at signalized intersections. *Transportation research record*, 2298(1):1–8, 2012.

[49] P. G. Michalopoulos. Vehicle detection video through image processing: the autoscope system. *IEEE Transactions on Vehicular Technology*, 40(1):21–29, Feb 1991. ISSN 0018-9545. doi: 10.1109/25.69968.

[50] P. Mirchandani and L. Head. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6):415 – 432, 2001. ISSN 0968-090X. doi: https://doi.org/10.1016/S0968-090X(00)00047-4. URL http://www.sciencedirect.com/science/article/pii/S0968090X00000474.

[51] T. Nagatani. Vehicular traffic through a sequence of green-wave lights. *Physica A: Statistical Mechanics and its Applications*, 380:503 – 511, 2007. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2007.02.072. URL http://www.sciencedirect.com/science/article/pii/S0378437107002269.

[52] K.r Nellore and G. P. Hancke. A survey on urban traffic management system using wireless sensor networks. *Sensors*, 16(2), 2016. ISSN 1424-8220. doi: 10.3390/s16020157. URL https://www.mdpi.com/1424-8220/16/2/157.

[53] C. Nicholson. A beginner's guide to lstms and recurrent neural networks, 2018.

[54] C. Olah. Understanding lstm networks, 8 2015. URL https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[55] G. M. Pacey. The progress of a bunch of vehicles released from a traffic signal. *Road Research Laboratory Note RN/2665/GMP*, 1956(1956):36–50, 1956.

[56] A. M. Pereira, H. Anany, O. Přibyl, and J. Přikryl. Automated vehicles in smart urban environment: A review. In *2017 Smart City Symposium Prague (SCSP)*, pages 1–8, May 2017. doi: 10.1109/SCSP.2017.7973864.

[57] I. Porche, M. Sampath, R. Sengupta, Y. Chen, and S. Lafortune. A decentralized scheme for real-time optimization of traffic signals. In *Proceeding of the 1996 IEEE International Conference on Control Applications IEEE International Conference on Control Applications held together with IEEE International Symposium on Intelligent Control*, pages 582–589, Sep. 1996. doi: 10.1109/CCA.1996.558925.

[58] D. I. Robertson. Transyt: a traffic network study tool, road research laboratory report, lr 253. *Road Research Laboratory, Crowthorne*, 1969.

[59] D. I. Robertson and R. D. Bretherton. Optimizing networks of traffic signals in real time-the scoot method. *IEEE Transactions on vehicular technology*, 40(1):11–15, 1991.

[60] M. Seredynski, G. Arnould, and D. Khadraoui. The emerging applications of intelligent vehicular networks for traffic efficiency. In *Proceedings of the Third ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet '13, pages 101–108, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2358-1. doi: 10.1145/2512921.2517048. URL http://doi.acm.org/10.1145/2512921.2517048.

[61] M. Seredynski, D. Khadraoui, and F. Viti. Signal phase and timing (spat) for cooperative public transport priority measures. In *Proc. 22nd ITS World Congress*, 10 2015.

[62] R. Shaikh. Choosing the right encoding method-label vs onehot encoder, 11 2018.

[63] A. G. Sims and K. W. Dobinson. The sydney coordinated adaptive traffic (scat) system philosophy and benefits. *IEEE Transactions on Vehicular Technology*, 29(2):130–137, May 1980. ISSN 0018-9545. doi: 10.1109/T-VT.1980.23833.

[64] K. A. Small and J. A. Gómez-Ibáñez. Chapter 46 urban transportation. In *Applied Urban Economics*, volume 3 of *Handbook of Regional and Urban Economics*, pages 1937 – 1999. Elsevier, 1999. doi: https://doi.org/10.1016/S1574-0080(99)80015-7. URL http://www.sciencedirect.com/science/article/pii/S1574008099800157.

[65] R. Stahlmann, M. Möller, A. Brauer, R. German, and D. Eckhoff. Technical evaluation of glosa systems and results from the field. In *2016 IEEE Vehicular Networking Conference (VNC)*, pages 1–8, Dec 2016. doi: 10.1109/VNC.2016.7835967.

[66] S. Stebbins, M. Hickman, J. Kim, and H. L. Vu. Characterising green light optimal speed advisory trajectories for platoon-based optimisation. *Transportation Research Part C: Emerging Technologies*, 82:43 – 62, 2017. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2017.06.014. URL http://www.sciencedirect.com/science/article/pii/S0968090X17301675.

[67] Simon Stebbins, Jiwon Kim, Mark Hickman, and Hai L Vu. Combining model predictive intersection control with green light optimal speed advisory in a connected vehicle environment. In *Australasian Transport Research Forum (ATRF), 38th, 2016, Melbourne, Victoria, Australia*, 2016.

[68] A. Stevanovic, J. Stevanovic, and C. Kergaye. Green light optimized speed advisory systems: Impact of signal phasing information accuracy. *Transportation Research Record*, 2390(1):53–59, 2013. doi: 10.3141/2390-06. URL https://doi.org/10.3141/2390-06.

[69] Ltd Sumitomo Electric Industries. Traffic signal control system moderato-s, 2018.

[70] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *IEEE transactions on pattern analysis and machine intelligence*, 28:694–711, 06 2006. doi: 10.1109/TPAMI.2006.104.

[71] A. Talebpour and H. S. Mahmassani. Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*, 71:143 – 163, 2016. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2016.07.007. URL http://www.sciencedirect.com/science/article/pii/S0968090X16301140.

[72] T. Tettamanti, I. Varga, B. Kulcsar, and J. Bokor. Model predictive control in urban traffic network management. In *2008 16th Mediterranean Conference on Control and Automation*, pages 1538–1543, June 2008. doi: 10.1109/MED.2008.4602084.

[73] M. B. Trabia, M. S. Kaseko, and M. Ande. A two-stage fuzzy logic controller for traffic signals. *Transportation Research Part C: Emerging Technologies*, 7(6):353 – 367, 1999. ISSN 0968-090X. doi: https://doi.org/10.1016/S0968-090X(99)00026-1. URL http://www.sciencedirect.com/science/article/pii/S0968090X99000261.

[74] M. Treiber and A. Kesting. *The Lighthill–Whitham–Richards Model*, pages 81–126. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-32460-4. doi: 10.1007/978-3-642-32460-4_8. URL https://doi.org/10.1007/978-3-642-32460-4_8.

[75] M. I. Ullah. Random walk model, 12 2016.

[76] T. Urbanik, A. Tanaka, B. Lozner, E. Lindstrom, K. Lee, S. Quayle, S. Beaird, S. Tsoi, P. Ryus, D. Gettman, et al. *Signal Timing Manual*. Transportation Research Board Washington, DC, 2015.

[77] V. Nguyen, O. T. T. Kim, T. N. Dang, S. I. Moon, and C. S. Hong. An efficient and reliable green light optimal speed advisory system for autonomous cars. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–4, Oct 2016. doi: 10.1109/APNOMS.2016.7737260.

[78] R. T. van Katwijk. *Multi-agent look-ahead traffic-adaptive control*. PhD thesis, Delft University of Technology, 2008.

[79] R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn. A test bed for multi-agent control systems in road traffic management. In Franziska Klügl, Ana Bazzan, and Sascha Ossowski, editors, *Applications of Agent Technology in Traffic and Transportation*, pages 113–131, Basel, 2005. Birkhäuser Basel. ISBN 978-3-7643-7363-4.

[80] A. G. van Senden. Director: Enabling advanced driver assistance systems with predictive signalized intersection control using lstm networks. Master's thesis, Delft University of Technology, The Netherlands, 2018.

[81] J. Vreeswijk, M. Harmenzon, M. Barto, E. Koenders, P. Luns, E. Verhoeven, J. Zee P. Smit, K. M. Shivakumar, K. op den Kelder, W. de Smet, A. Schreuders, and B. Schiltmans. Intersection topology format (itf). Technical report, CROW, 06 2019.

[82] M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem. Connected variable speed limits control and car-following control with vehicle-infrastructure communication to resolve stop-and-go waves. *Journal of Intelligent Transportation Systems*, 20(6):559–572, 2016. doi: 10.1080/15472450.2016.1157022.

[83] M. A. Wiering, J. van Veenen, J. Vreeken, and A. Koopman. Intelligent traffic light control, 2004.

[84] J. Withanawasam and A. Karunananda. Multi-agent based road traffic control optimization. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 977–981, Oct 2017. doi: 10.1109/ITSC.2017.8317896.

[85] L. Yang, M. Zhao, D. Sun, and H. Liu. Variable speed limit control scheme for vehicle in the vicinity of signalized intersection. *Modern Physics Letters B*, 32(19):1850218, 2018. doi: 10.1142/S0217984918502184.

[86] K. M. Yousef, M. N. Al-Karaki, and A. M. Shatnawi. Intelligent traffic light flow control system using wireless sensors networks. *J. Inf. Sci. Eng.*, 26(3):753–768, 2010.

[87] G. Zhang and Y. Wang. Optimizing minimum and maximum green time settings for traffic actuated control at isolated intersections. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):164–173, March 2011. ISSN 1524-9050. doi: 10.1109/TITS.2010.2070795.

[88] B. Zhou, J. Cao, and H. Wu. Adaptive traffic light control of multiple intersections in wsn-based its. In *VTC Spring*, pages 1–5, 2011.