Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization:	Transport Engineering and Logistics
Report number:	2012.TEL.7708
Title:	**Traffic forecasting for improved Adaptive Traffic Control in case of a non-ideal detector configuration**
Author:	B.A.F. Krstulovic

Title (in Dutch)	Verkeer voorspellingen voor een verbeterde Adaptieve VRI controller in het geval van een niet ideale detector configuratie

Assignment:	Masters thesis
Confidential:	yes (unknown)
Initiator (university):	prof.dr.ir. J. Hellendoorn
Initiator (company):	dr.ir. R.T. van Katwijk (TNO, Delft)
Supervisor:	prof.dr.ir. J. Hellendoorn
Date:	July 20, 2012

Delft University of Technology

FACULTY OF MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department of Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Student:	B.A.F. Krstulovic	Assignment type:	Master project
Supervisor (TUD):	prof.dr.ir. J. Hellendoorn	Creditpoints (EC):	36
Supervisor (Company):	dr.ir. R.T. van Katwijk	Specialization:	TEL
		Report number:	2012.TEL.7708
		Confidential:	yes
			until: unknown

Subject: **Traffic forecasting for improved Adaptive Traffic Control in case of a non-ideal detector configuration.**

Throughout the past century a series of generations traffic controllers have been invented to better control intersections. The simplest form uses a fixed interval to switch streams. Later, actuated control was implemented which makes use of detectors in the ground to detect vehicles and more efficiently allocate its green time. The last generation of adaptive controllers also takes into account future expected traffic and can communicate with neighbouring intersections for an optimal signal plan. The Multi-Agent Look-Ahead Traffic-Adaptive Controller by R.T. van Katwijk developed at TNO and DCSC (Katwijk, 2008) belongs to the latest generation of adaptive controllers.

This adaptive controller achieves 20% improvement in average delay of a vehicle over actuated control. The controller consists of multiple components like: data gathering, queuing model, prediction model, optimizer and network control. The prediction model anticipates vehicle arrivals by placing a detector upstream of the signalized intersection. When detectors are placed further downstream this anticipation time decreases. In practice, due to cost constraints or because of a short intersection approach, detectors have to be placed close to the intersection. When the anticipation time decreases the overall performance of the controller decreases. In view of the adaptive controller placing detectors close to the intersection is therefore considered non- ideal.

This study requires insight in traffic control of discrete events and forecasting techniques. Objective is to find a solution that can increase the performance of the controller if detectors have to be placed closed to the intersection.

Direction for research can be:

- What models can forecast the vehicle arrivals?
- What is the best model to improve the controller performance?
- How can the controller be extended with a forecasting technique ?

The report should comply with the guidelines of the section. Details can be found on the website.

Prof.dr.ir. J. Hellendoorn

Master Thesis

Traffic Forecasting

for improved Adaptive Traffic Control in case of a non ideal detector configuration

By B.A.F Krstulovic
July 2012

TU Delft Delft University of Technology

TNO innovation for life

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude and heartfelt thanks to my supervisor, Dr. ir. R.T. van Katwijk for his support, invaluable guidance and critical comments throughout the project. Also, I really appreciate that you have allowed me to continue our collaboration by letting me participate in a real project for TNO after (almost) completion of my thesis.

Second I would like to thank my supervisor from the TU Delft, Prof. Dr. Ir. J. Hellendoorn for introducing me to the subject of traffic control and bringing me into contact with R.T. van Katwijk. I was unaware of traffic control as a potential subject if I had not accidently bumped into a prior meeting of the Professor and a former student J. Carriere whom was just about to hand in his master thesis on traffic control. The discussion that followed was enlightening and made it very clear to me that I wanted to graduate on the same topic. A small seven months later it is now my turn. In those seven months the subject of traffic control has grown a lot on me; I have really enjoyed the complex scenarios of discrete vehicle trajectories unfolding in my head. Above all I would like to Professor for helping me focus on the structure.

My biggest thanks go to my mom, dad and sister whom have supported me throughout my studies in more ways than just a listening ear. Student life above all was a chance to expand myself in other ways besides studying. I have expanded in many. The deviations from the straight line must have put considerable worries in their heads, thanks for the support till the finish.

My heartiest thanks to all my friends. Especially my new friends that I have gained during my masters in Transportation Engineering. Because of the study tour we organized with Willem, Ivo and Jelmer we were able to form a great team with Merijn, John and Martin whom organized the Transportkunde society. During the study tour we gained even more new friends which made both tours to Russia and China unforgettable. Thanks to these shared experiences the field of transportation engineering has become extremely lively.

Last but not least, I want to thank my girlfriend Judith for the support when attention was a little absent ;). I promise that attention^2 will be returned when we explore Asia again! X.

Alexander Krstulovic

Delft, 20 July 2012

ABSTRACT

In an ideal situation the Multi- Agent Look-Ahead Traffic-Adaptive controller by R.T. van Katwijk (2008) shows a 20% increase over actuated control in average delay of a vehicle passing a signalized intersection. In an ideal situation detectors are placed upstream (look- ahead detection) so that the controller can anticipate vehicle arrivals. When detectors are placed further downstream this anticipation time decreases. In practice, due to cost constraints or because of a short intersection approach, detectors have to be placed close to the intersection. For each link feeding the intersection that requires detectors to be positioned near the intersection a part of the 20% increased performance is lost. If 2 out of 4 links are ‘reduced’ than only 10% remains of the 20% improvement. In practice most detector configurations are not ideal. The need for a good performing controller in all situations is therefore self-evident. This thesis will focus on improving the case of how to extend the anticipation horizon if detectors are placed close to the intersection. This can be done by studying the historic events that occur at a downstream detector.

A literature study revealed that vehicles will arrive random or in groups, so-called platoons. Study of the available historic data supplied by TNO (Groene Kruisweg, Spijkenisse) and the available literature showed that individual arrivals can be aggregated in time intervals to create a traffic demand profile appearing at a detection point throughout a day. This demand profile is almost similar for weekdays (weekends are different) and corresponding days a full week in the past. This past information can be used to forecast future demand. Aggregate forecasting is the designated way to forecast arrivals when vehicles arrive random and was therefore researched for implementation in the controller.

Besides individual arrivals traffic flows also show group arrivals due to upstream signalized intersections. Platoons are under influence of dispersion which is the weakening of the group effect as a platoon moves downstream. No model was found in literature that forecasts platoon arrivals based on historic data of a single detection point. Forecasting platoons by means of aggregated demand forecasting is unjustified because information on the interspacing of platoons is lost when data is aggregated. Especially this time gap between platoons can be very important for the controller to let side streams pass without causing much delay for all streams. Therefore a new model was developed that extracts three platoon characteristics from the historic signal and forecasts these separately. These platoon characteristic are the time spacing between the leaders of two successive platoons (Inter platoon headway), the time spacing between the first and last vehicle in a platoon (Platoon size), and the amount of vehicles in a platoon (Platoon volume).

In addition to vehicle arrivals the literature study includes research on various forecasting techniques. Two techniques were chosen for further application in respect to aggregated demand forecasting: the Holt Winters and Artificial Neural networks (ANN) algorithms. The ANN algorithm can also be used to make platoon forecasts based on the newly developed model. The thesis first evaluates the forecasting potential of both techniques and determines the best model for aggregated demand forecasting by performing an offline simulating on a real world dataset. The newly developed platoon model is also evaluated through off line simulation on a real world dataset. The next stage involved implementation of the forecasting technique in the adaptive controller. To quantify the improvement of the controller in terms of delay the controller with forecasting measure was tested in the simulation environment PTV VISSIM 5.2 on an intersection with a non-ideal detector configuration.

First the results the aggregated demand forecast will be discussed. With equivalent history of one day in the past for both Holt Winters and the Artificial Neural Network the Holt Winters algorithm outperformed the Neural network, respectively RMSE 3.49 and RMSE 4.0. Both techniques were implemented in the Adaptive controller and validated in a simulation environment to see if the forecasts are able to regain the performance that is lost (from 20% to 10%). To be able to record historic data a detector is required that can count vehicle

arrivals; the stop line and extension detector that are usually present are not able to do this. Therefore a faraway detector positioned just before the extension detector is a prerequisite to be able to make forecasts. Placement of this extra detector without forecast technique extends the anticipation horizon to return a performance of 18%, however still 2% is lost. In succession the forecasting measure is added to the controller. The simulations indicated that Holt Winters outperformed the ANN in terms of delay in line with earlier findings of the offline simulations. Holt Winters gives a significant improvement of 2% during peak hours and returns the performance to 20% improvement in respect to actuated control. The best position was found at 70 metres from the stop line. Holt Winters does no result in increased performance during off peak hours and should be left off.

Development of the Platoon model indicated that a great deal of information on the traffic state can be extracted from the platoon characteristics. The signals give insight on the varying demand of the upstream conflicting links that feed the downstream link. These signals also show a recurring profile with preceding days similar to aggregated demand forecasting. For this specific intersection it was concluded that during the night (20:00h – 05:00h) the demand is sufficiently low that no forecast should be done; placement of a faraway detector should be sufficient to anticipate individual arrivals. During peak hours (05:00h – 10:00h) downstream arrivals should be forecasted with the Holt Winters method and aggregated forecasting. The reason for this is that the demand on all upstream conflicting streams is so high that there is no need to forecast the gaps between platoons. From 10:00h to 20:00h the traffic state is shaped by a dominant main stream that shows platoons with occasional appearance of side streams. After 15:00h the demand on side streams is so low that almost only the main stream is visible. A filter was developed that can remove the little demand on side streams to uncover the main stream, after 15:00h the developed platoon model is capable of forecasting platoon arrivals. The time window between 10:00h and 15:00h proved difficult because the main stream is still dominant with considerable demand on one or two side streams. The presence of these side stream make the signal more erratic. The developed filter can uncover the main stream in presence of those side streams, however the filter is still sub optimal and therefore outliers remain in the signal that make forecasting difficult.

In retrospect a mistake was identified in the implementation of the model for the time window 10:00h to 15:00h. The model is tuned to forecast the dominant stream. However if vehicles from side streams appear in the real signal then the forecasts for the main stream are consequently based on the side streams instead of the main stream; this will result in accumulating false forecasts.

The method of extracting platoon characteristics shows promise for future use. It is recommended that the model is further improved to identify platoons based on their upstream origin because forecasting a downstream signal that includes information on all upstream conflicting streams proved unsatisfactory.

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

HW	Holt Winters
ANN	Artificial Neural Networks
OD	Origin Destination
SG	Signal group
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
IP	Inter platoon headway
PS	Platoon size
PV	Platoon volume

CONTENTS

CHAPTER 1 INTRODUCTION

This chapter will shortly introduce the concept of traffic control. In addition the system will be defined, the problem and research approach will be explained.

TRAFFIC CONTROL

Seven billion people and seven billion destinations. At this time there are an estimated 1 billion (Wikipedia.nl) motorized vehicles on this planet. Getting people to their destination in a manner that is safe and accepted is the prime problem of traffic control. Getting them there fast and efficient is a bigger problem.

Transportation has always been a crucial aspect of human civilization, but it is only in the second half of the last century that the phenomenon of traffic congestion has become predominant due to the rapid increase in the number of vehicles and in the transportation demand in all modalities. Figure 1 gives an indication of the growth rate of motorised vehicles in Netherlands alone; it shows a 2% increase per year and the gradient shows no sign of stabilizing.

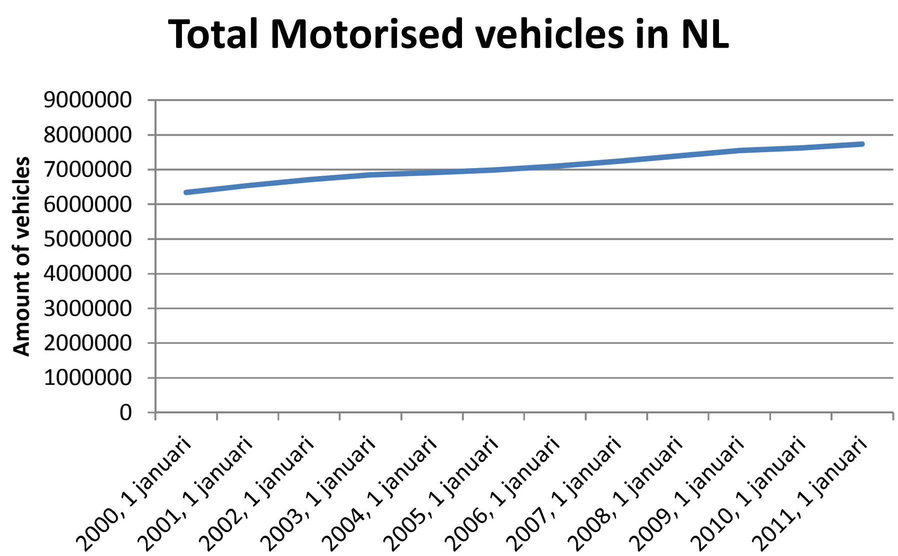


Figure 1 Total Motorised vehicles in NL, Source: Central Bureau of Statistics

Three traffic types generally appear and are managed and controlled via different solutions. These traffic types are urban, sub-urban and highway traffic. The Dutch government seems to focus primarily on improving highway congestion. It seems that highway congestion is the most felt by the commuters and therefore requires more immediate attention, yet urban congestion may account for a large part of congestion.

Urban networks are characterized by relatively short road stretches linked through intersections or round-a-bouts. Traffic on urban roads is a heterogeneous mix of transport types utilizing a common infrastructure. Sub-urban road have higher speed limits but also make use of intersections. With increased demand the need for control is self-evident in order to ensure safe passage of crossing vehicles. Traffic control ensures safety by giving alternating priority to conflicting streams. Research in traffic control aims to increase throughput and decrease environmental impact. The social relevance is discussed in more detail in the next chapter.

Throughout the past century a series of generations traffic controllers have been invented to better control these intersections. The simplest form uses a fixed interval to switch streams. Later, actuated control was

implemented which makes use of detectors in the ground to detect vehicles and can therefore more efficiently allocate its green time. The last generation of adaptive controllers also takes into account future expected traffic and can communicate with neighbouring intersections for an optimal signal plan. The centre piece of this thesis is the Multi-Agent Look-Ahead Traffic-Adaptive Controller by R.T. van Katwijk at TNO and DCSC (Katwijk, 2008). For more detailed information on the Multi-Agent Look-Ahead Traffic-Adaptive Controller the reader is referred to Appendix A.

This controller already achieves 20% improvement in average delay of a vehicle over actuated control, even so there is room for improvement. The controller consists of multiple components like: data gathering, queuing model, prediction model, optimizer and network control. A big advantage of the controller compared to other controllers of the same generation is in the way the signal plan is optimized. The optimization requires searching a tree of countless possibilities. The controller searches this tree in a clever manner so that it does not need to investigate every possible solution. This area requires therefore no immediate improvement. A great optimization algorithm, however, is pointless if the wrong data is optimized. This input data is supplied by the prediction model. The prediction model provides information (forecasts) on future arrivals. A large part of the controller's performance is determined by his knowledge of future arrivals. The prediction model can be improved on two accounts; these will be discussed in the problem definition.

Figure 2 Evolution

SYSTEM DEFINITION

Figure 3 shows a schematic diagram of the system. Each displayed intersection is controlled. The focus is on the middle intersection and its controller. The middle intersection will be referred to as the downstream intersection. The neighbouring intersections with observed wind directions will be referred to as upstream intersections.

The intersection has four arms, or links, of which the type of link is arbitrary; a link can either be connected to another controlled intersection or something else. For example a neighbourhood or a football stadium. The

system contains at least one link of a different type. Each link can have secondary roads attached which distort the arrival patterns at the downstream intersection. These disappearing or appearing cars are so called 'deaths' and 'births'.

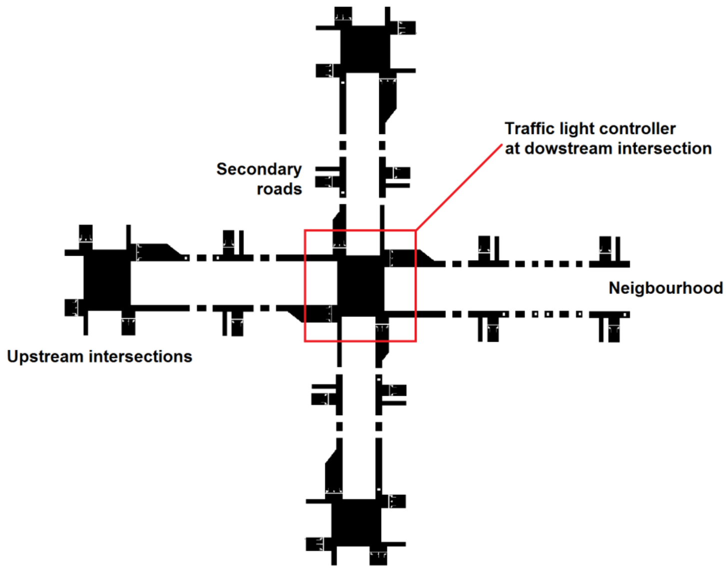


Figure 3 System overview

Throughout the report addressing the different streams or 'signal groups' will be coherent with the Dutch standard (*CT4821*, 2006). Figure 4 gives an overview of the indexes. This thesis will restrict its research to the car traffic flows (1:12). Dedicated public transportation lanes (41:52), pedestrians (31:38), and bicycles (21:28) will not be considered.

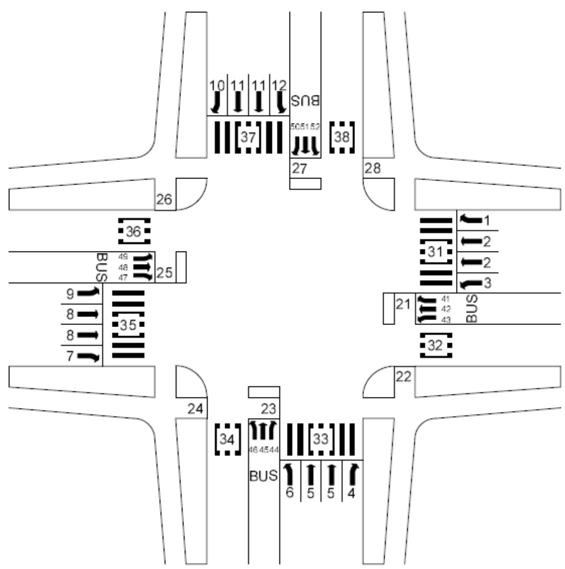


Figure 4 Dutch standard for indexing Signal groups (*CT4821*, 2006)

In a control environment the system is as follows:

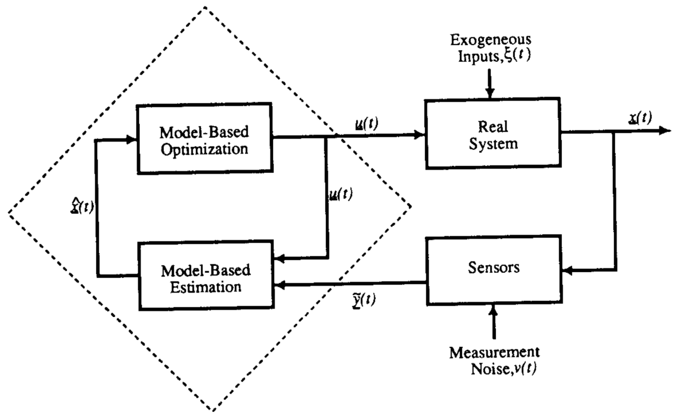


Figure 5 Control system schematic (Head & Mirchandani, 1994)

This is also referred to as model predictive control (MPC). The 'real system' describes the traffic state. The state is recorded by two sensor types. The measured traffic state is used to estimate the future state based on a model. These expected future states are implemented in the optimization algorithm to calculate a new control solution (independent variables) which is in turn applied to the real system. The new state of the system (dependent variables) is fed back in the estimator and can be evaluated with its previous estimate for that state. This feedback will results in a better estimate for the next state.

This thesis will make use of two sensor types: cameras and inductive ground loops.

Cameras recognize re-appearing license plates and can therefore record information like average speed position, or arrival time based on specific car identities. The tracking of individual vehicles allows for a detailed microscopic analysis.

Ground loops use induction to detect cars, a signal is held on detection and released upon departure. Ground loops are only able to record anonymous cars and thus allow for a more macroscopic analysis. Single ground loops detect the presence of a car whereas double ground loops can measure speed.

Ground loop detectors are named according to their position relative to the downstream stop line Figure 6 Detector positions in white (Figure 6). The detector closest to the stop line is called the 'stop line' detector (Dutch: "koplus'). The stop line detector's main function is to verify the presence of a vehicle at the signal group.

Just positioned behind the stop line detector is the 'extension' loop (Dutch: "langelus'). As the name implies the loop is several metres long. The loop functions as an observer of the queue length. If no detection is made over the full length than the controller 'knows' that the queue is resolved for the most part. A downside of the extension loop is that it is not able to count vehicles. This is because a positive signals is held if a presence is registered anywhere on the full length of the loop. As a consequence the loop cannot distinct between one or more vehicles.

Sometimes before the extension loop another normal sized loop detector is present. This loop is called the 'faraway 'detector (Dutch: "verweglus'). Before that another loop can be placed even further upstream, the far-faraway detector. Both loops have the same functionality; they detect vehicles further upstream to have knowledge of arrivals so they can extend the current green signal (Rijkswaterstaat, 2002). The faraway detector is also able to count vehicles, in addition they are also used for queue verification. If one of the detectors remains occupied for a long time vehicles are standing still on top of it. Detection of a queue at the faraway detector triggers an alarm so that the controller will prioritize emptying of the queue.

In view of the Multi-Agent Look-Ahead Traffic-Adaptive Controller one extra detector is defined: the ''look-ahead' detector. This is a detector placed very far upstream with a maximum just downstream of an upstream intersection. The look- ahead detector allows for anticipation of the vehicles. The current prediction model in the Multi-Agent Look-Ahead Traffic-Adaptive Controller is based on anticipated arrivals from look- ahead detection. The distance between a look- ahead detector and a far- faraway detector can be quite large (~2km). The controller is also able to communicate with other detectors placed between the look- ahead detector and the far- faraway detector. These detectors function as a correction mechanism on the expected arrivals. Having more look- ahead detectors on a road stretch is really uncommon. The following figure clarifies the different detectors once more:

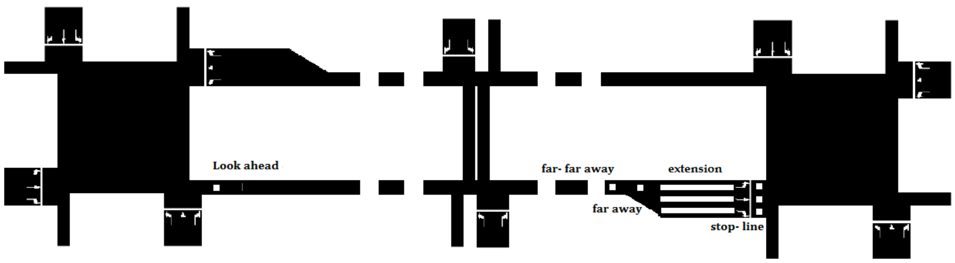


Figure 6 Detector positions in white

PROBLEM DEFINITION

The controller is an algorithm that constantly monitors the state of the system. Based on the current state and the expected future state the algorithm deducts an optimal plan for the next 120 seconds. Optimization comes down to making the right decision without squandering the possibility for future higher profits (Appendix A). In order to do this knowledge is required of the future states. This is done using prediction models. A better prediction model will directly improve overall results.

In an ideal situation the Multi- Agent Look-Ahead Traffic-Adaptive controller (Katwijk, 2008) shows a 20% increase over actuated control in average delay of a vehicle in the system. In an ideal situation detectors are placed upstream (look- ahead detection) so that the controller can anticipate vehicle arrivals. When detectors are placed further downstream this anticipation time decreases. In practice, due to cost constraints or because of a short intersection approach, detectors have to be placed close to the intersection. An intersection with varying detector configurations and varying horizons on the feeding links will be referred to as an *asymmetric* intersection.

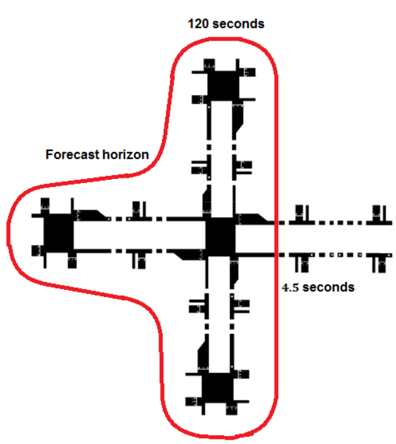


Figure 7 'Asymmetric' intersection

If a detector is placed at 100m from the stop line the anticipation horizon is equivalent to 4.5 seconds (80 km/h). With a small anticipation time vehicles are detected late, as a result the signal plan cannot respond to the latest variations in demand. The controller can optimize a 120 seconds plan for the future; placing the detector close to the intersection results in a gap of 120 – 4.5 = 115.5 seconds of unknown arrivals. To verify the decreased performance a simulation in PTV VISSIM 5.2 was done where 2 side streams had upstream detection removed, so the horizon for these links equals 0 seconds.

PTV VISSIM 5.2 is a program that can simulate traffic in a pre-designed network on microscopic scale. For this test an isolated intersection was used and 10 simulations were run for a symmetric (1) and asymmetric case (2). Each simulation was executed with a different random seed.

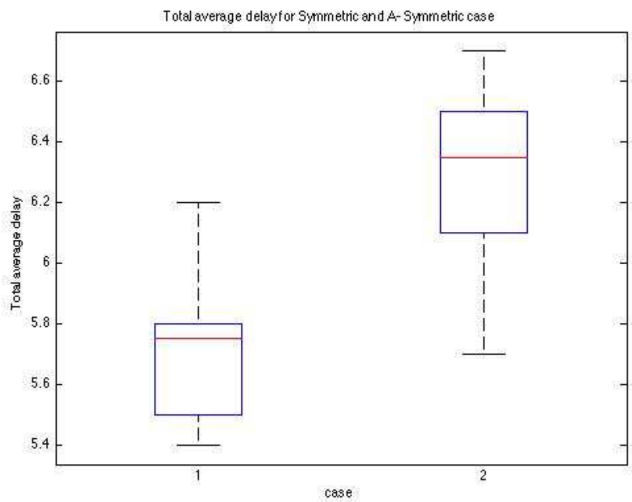


Figure 8 Deterioration of asymmetric intersection

The results clearly show a significant difference. The asymmetric case shows an approximate 10% increase in total average delay of a vehicle in the system. Delay is defined as the time that a vehicle needs to pass the intersection including waiting time in the queue minus the time that is needed to pass the intersection without interruption (in free flow).

For every intersection link reduced from look- ahead detection to stop line detection a part of the 20% improvement will be lost. Techniques and models have to be researched and developed to increase the knowledge of future arrivals when only nearby detection is available. This field of research belongs to forecasting techniques.

Besides the main problem of the asymmetric intersection another problem still remains even when vehicles are detected upstream and forecasted downstream. Assume that vehicles are detected at the beginning of the controller horizon of 120 seconds, this translates to a distance of 1.7 kilometres. The current prediction model detects a vehicle and then propagates the vehicle downstream under the assumption that is in free flow. A lot can happen in 1.7 kilometres that changes the trajectory of a vehicle. The free flow forecast is therefore often wrong. A better model is required to forecast downstream vehicle arrivals with upstream detection that travel under influence of external factors and therefore have different trajectories than for free flow conditions.

In practice most intersections won't be symmetric. The need for a good performing controller in all situations is therefore self-evident. This thesis will therefore focus on improving the case of how to extend the forecasting horizon if an intersection is asymmetric. The non-satisfactory free-flow model is left for future study. The following research question arises:

How should an asymmetric intersection be configured so that it performs at its best?

This in turn leads to numerous sub questions:

1) Which kind of arrival patterns appear?
2) What modes can forecast the vehicle arrivals?
3) Which model delivers the best forecast?
4) What is the best model to improve the controller performance using one faraway detector?
5) What are the variables that determine the correct placement of an extra detector?

RESEARCH APPROACH

This section discusses the research approach in order to able to answer the research questions.

The first step involves a literature study. The aim of the literature study is to acquire knowledge on the traffic behaviour in front of a traffic light. What are the arrival patterns, what are the reasons for arrivals patterns? What happens when they have arrived, how do they queue and how do queues resolve? Are there known methods that can predict arrivals based on a given parameters like, road geometry, upstream signal timings etc. If there are no descriptive forecast methods than what are techniques that can be used to forecast downstream arrivals?

This knowledge should be able to answer the first research question on what arrival patterns appear at downstream intersections and what influences them. The second research question can also be answered because it should have clarified which models are best suited and whether development of new model is required.

To fully answer the third research question the models have to be tested offline. Offline testing is needed to evaluate and validate the forecasting potential of the model. The most promising models found in the literature will be tested and compared based on their forecast error. If no model can be found in the literature a new model has to be developed. This model also needs to be simulated off line to evaluate its forecast error.

To answer the fourth research question; the forecast model that performed best during the off line simulations needs to implemented online to quantify the improved performance of the controller for an asymmetric intersection. Online testing in a simulation environment can shed light on the real gains in terms of delay. So:

Table 1 Research approach

Research question	Goal	Method
1	Study of arrival pattern	Literature study
2	Potential forecasting models	Literature study
3	Find best forecasting model	Simulate offline
4	Implement forecast in controller to quantify performance increase or decrease	Simulate online

Model development and offline testing will be done using a real world dataset. The dataset is real data and therefore naturally captures discrepancies. Models that can make sense of the data are more difficult to develop but should eventually also be more generic. To evaluate the impact of a good or bad forecast it needs to be implemented in the controller.

First this requires a correct implementation of the forecast in the controller. Second the new adaptive controller with forecasting measure needs to be tested against the old version without forecasting measure. This is done in a simulation environment. The simulation environment tries to fully replicate real life situations and can shed light on the possible gains. For this research a simulation can quantify the possible improvement or deterioration in delay over the system, as well as the amount of vehicle stops. The simulation environment can also provide an answer to research question five. Varying the detector location will give insight on the main drivers for detector placement and perhaps also the optimal location.

Research question	Goal	Method
5	Determining detector placement	Simulate online

The simulation environment on the other hand is a representation of the real world and will not be able to fully replicate the real signal, it is a more idealized version. Results should be evaluated carefully because a model that works in simulation might not work in reality because the signals are subject to noise.

The research process will be as follows:

- Literature study
    - Arrival patterns
    - Queues
    - Forecasting techniques
        - Descriptive
        - Generic
- Which models are appropriate and is a new model required?
    - Choose and or build appropriate model
    - Run offline forecasting simulations for model validation
    - Evaluate forecasting
    - Implement the model in the controller
    - Run 10 online simulations with and without forecast
    - Evaluate which model results in the greatest improvement
    - Simulate and evaluate impact of detector replacement

All algorithms are designed in MATLAB to analyse a real world dataset. In order to use these developed forecasting methods in the simulation environment there are two possibilities: embedding the code in the JAVA code of the adaptive controller or calling the MATLAB algorithms via an interface. Option one means recoding every algorithm within the controller, which is a lot of work. Even more so because many of the vector/matrix handling functions that are present in MATLAB are lost and have to be reinvented. For proof of concept this methodology is sufficient. In case of real world implementation the algorithm will have to be hard coded afterwards. The following diagram shows how all software programs are interconnected:

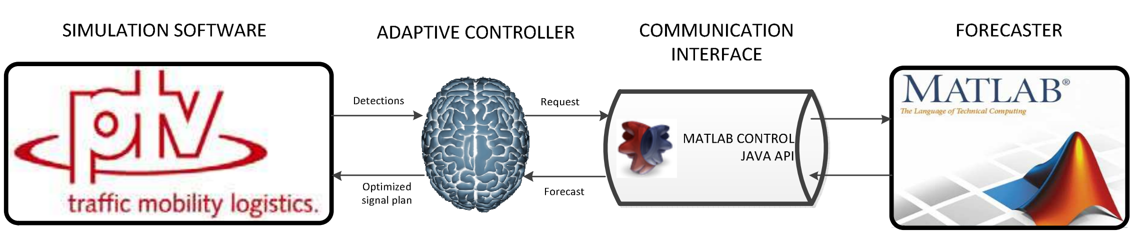


Figure 9 Software setup

REAL WORLD DATASET

The dataset was chosen because of its availability, size and applicability. The intersection in the dataset is asymmetric and holds information for both a link that is fed by an upstream intersection (Groene Kruisweg eastbound, signal group 7,8,9) as well as a link that is fed by a business/commercial area (Malledijk, signal group 4,5,6). The available dataset consists of 10 successive days: 1 November 2011 to 10 November 2011.

The dataset is quite extensive and has stored information based on *camera* observations and information based on *inductive loop detectors*. The dataset is used by TNO to evaluate ODYSA, a system that gives speed advice at multiple locations further upstream on the Groene Kruisweg to improve throughput of the intersection. It is important to keep in mind that the observed dataset on the Groene Kruisweg east bound is most likely affected by the speed advice and consequently will show 'distorted' patterns. At this point it is assumed that the distorted patterns are not likely to affect evaluation of models, a forecasting model should be able to forecast any situation regardless of the distortion. Implementation in the simulation environment will shed more light on the general applicability of the model.

The cameras are positioned along the Groene Kruisweg (the positions of the cameras in Figure 10 are approximate; the exact locations are -2090m,-1340m,-725m,0m). The inter-spacing of the cameras starting at the upstream intersection and all the way down to the downstream intersection plus the fact that cameras can track individual cars based on license plate recognition make this system ideal for a transient analysis. Downside is that cameras do no register the speed of an individual car at point of passage; they can merely indicate the average speed over the road segment.

Figure 10 Real world dataset

The intersection is covered with a series of loop detectors indicated by the white/red squares. The detectors are relatively close to the intersection and will therefore show little information on the evolution of the platoon dispersion. They do, however, supply a great deal information on arrivals patterns, inter arrival times, occupancy, and speed.

Figure 11 Groene Kruisweg and Malledijk impression

SIMULATION ENVIRONMENT

The simulation environment is PTV VISSIM 5.2. The software program simulates driving behaviour on a microscopic scale. Networks can be constructed using various building blocks; most important are the road segments or links, reduced speed areas, detectors and traffic light infrastructure. After constructions many characteristics can be implemented like origin destination matrices, car driving behaviour, and the type of traffic light controller. The multi-agent look-ahead controller is a module that was implemented on top of PTV VISSIM.

To test the algorithms two different networks were required; one network that consists of multiple controlled intersections and one isolated intersection. The isolated intersection is located in *Helmond*. This network was also used for previous research by TNO and has detailed information on demand (origin- destination) for 24 hours.

The second network is located in *Assen* and consists of three linked intersections. Due to the upstream signal interventions different arrival patterns emerge.

Both networks have been extensively calibrated in the past and therefore closely resemble real life situations.

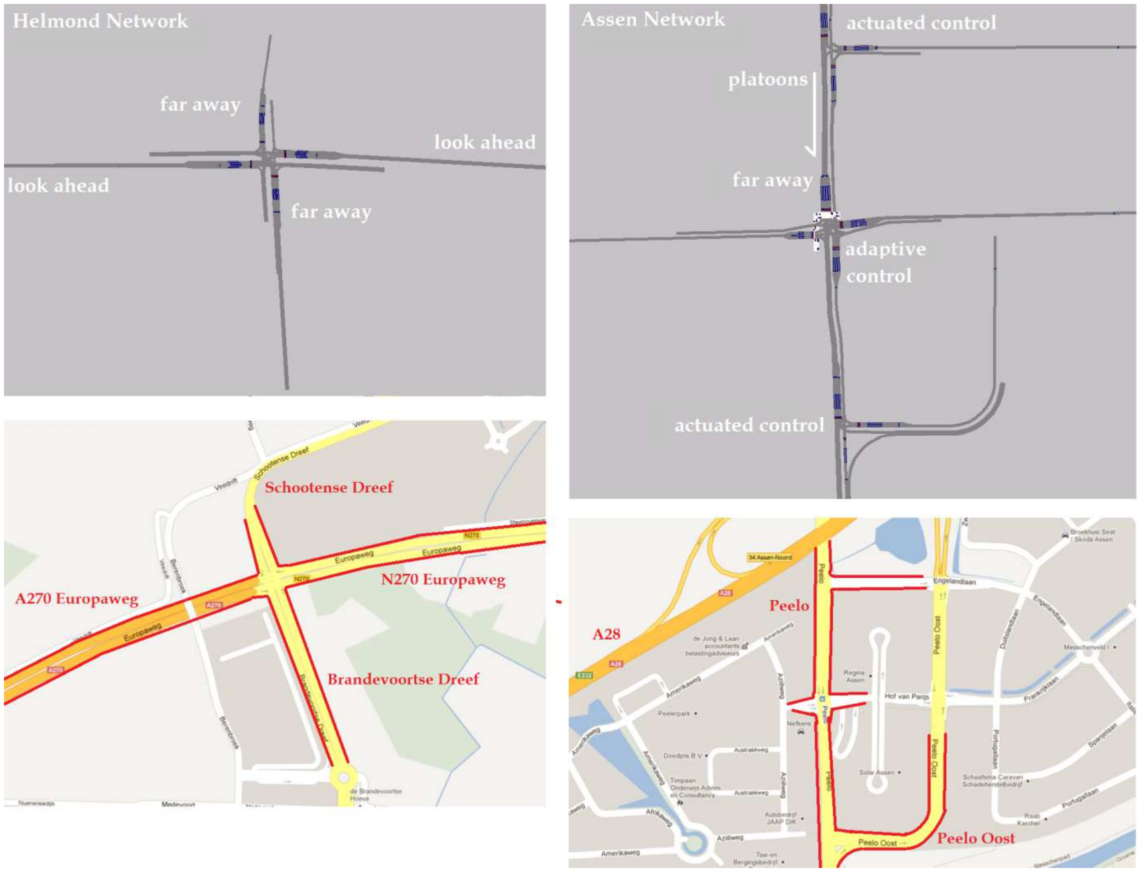


Figure 12 Helmond and Assen simulation network in PTV VISSIM 5.2

CHAPTER 2 SOCIAL RELEVANCE

Improved traffic control affects how people perceive travel times, enjoy travelling, and enjoy going to work. Apart from the directly visible results long term problems can be battled that eventually also affect the quality of life. A few of these are elaborated on in this chapter.

TRAFFIC GROWTH

A 14 % increase in vehicle –kilometres is expected between 2010 – 2015 . With a normal economic growth delay due to traffic jams is expected to grow 16% between 2010 – 2015 (Francke, Derriks, Gordijn, Groot, & Savelberg, 2010). This is a directly related to the capacity growth lagging behind the demand growth. Other important factors are fuel prices and economic growth (Mourik, W.Groot, & J.Francke, 2008). It is not mentioned how much urban traffic adds to the congestion.

Milosivic (2010) focuses on urban congestion but also fails to mention the total contribution of urban congestion to overall congestion. Milosivic (2010) argues that a large part of urban congestion is due to 'passing traffic' (dutch: 'doorgaand verkeer'). A highly dispersed area of functionalities (residential, commercial, industrial) around the city casts a large web of destination-traffic (dutch:' bestemmingsverkeer'). Passing traffic is defined as traffic that does not originate or is not destined for a certain area but passes through and thus participates in the network. The congestion on urban road is largely attributed to multiple functions that urban arterials fulfill; distributing and collecting to dispersed functional areas around the city centre, aswell as traffic intended for the city centre. Milosivic (2010) argues that due to limited space capacity is hard to increase and solutions should be sought in optimizating flows and mobility management. Optimized urban traffic can ultimately contribute to economic growth as well as better perception of quality of life.

EMISSION REDUCTION

Like a twin brother, emissions follow wherever traffic goes. Besides the logistical challenge an increasing awareness has grown the past few years in respect to air pollution and global warming. Getting from A to B as fast as possible is not so important anymore if everything lies below sea level. Not to mention the more apocalyptic scenarios.

In the Netherlands in 2008 transportation sector was responsible for 21% of the total emissions. The road sector therein vouches for 70% of all greenhouse gas emissions (Compendiumvoordeleefomgeving.nl). Urban contribution is unfortunately not mentioned. Figure 13 gives an impression of the C02 emissions per country:

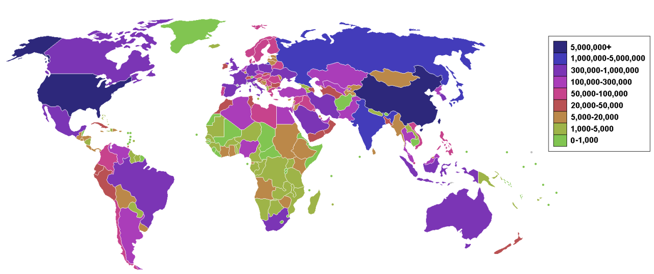


Figure 13 Countries by carbon dioxide emissions world map (source: Wikipedia)

Intensive research is done to make cars produce less emissions or have none at all. Cars are built lighter, smaller, have improved efficiency or use alternative technologies like hybrid motors, fuel cells, biofuels or harvest the sun's power. The NUNA Solar car is a great example of what can be achieved by clean solar energy.

However, it is questionable whether the reduced pollution from newly developed cars can offset the yearly increase in vehicles. In addition the highest emissions rates are during traffic congestion where cars repeatedly accelerate and decelerate at low speeds.

Barth and Boriboonsomsin (2008) have researched more closely the contribution of congestion to overall emissions. They argue that one of the following three strategies could each lower emissions (in California) by 7 to 12 %. If combined C02 reductions of even 30% could be achieved.

- Congestion mitigation strategies that reduce severe congestion and reduce traffic speeds (e.g. ramp metering, incident management, congestion pricing).
- Speed management strategies that bring down *excessive* speeds to more moderate speeds.
- Traffic smoothing strategies that reduce the number of acceleration and deceleration events.

Improved traffic light control can reduce emissions on two counts; first if delay decreases in the system cars are less idle and thus reduce pointless emissions. Second, acceleration and deceleration events which cause relatively the most emissions, can also be reduced.

FOSSIL FUEL REDUCTION

Whereas emissions are concerning everybody and the mind-set is predominantly uniform throughout the world, one can argue that it is not main priority for the major growing economies. China even surpassed the USA and now tops the list of C02 producers (pbl.nl) . Seemingly in contrast, China is also leading in renewable energy sources (guardian.co.uk).

Figure 14 Economic drivers

"However, the reality is that China's government is beginning to unleash a low-carbon dragon which will power its future growth, development and energy security objectives."

Ultimately greenhouse reduction is everybody’s concern.

Fossil fuel reduction, however, is not in everybody’s best interest. Fossil fuels originate from organic decomposition for periods over 500 million years. Fossil fuels are turned into gasoline, the main driver of the world’s economy. Complete transition to none fossil fuels will become the biggest challenge of all since it means redefining the economy.

Optimizing traffic flows is trivial to decreased energy spillage. Electric vehicles indirectly also add to fossil fuel reduction when less energy is spilled in traffic. This is due to the fact that electric vehicles are generally charged through the electricity net which at this point is still predominantly fed by non – renewable sources.

AIR POLLUTION

Air pollution is also caused by emissions but the effect to human health is direct. Air pollution is the release of chemicals particulate matter or biological materials that cause harm or discomfort to all living organisms. According to the World Health Organization (WHO.int) air pollution can cause respiratory infections, heart disease, and lung cancer. The most common sources of air pollution include particulate matter, ozone, nitrogen dioxide, and sulphur dioxide.

Geographical setting, climate, and structure of the city are important factors that can amplify the impact of air pollution. Well known are the pictures of smog trapped over high density cities that are located between mountains; for example Athens or Mexico City. The most heavily affected are cities in developing countries with heavy industry, the poor naturally receive the bulk since they live out in the open or have to work in these conditions. Traffic optimization will also reduce air pollution even if the effect is insignificant compared to other measures (e.g. soot filter, electric driving).

Figure 15 Mexico City smog

CHAPTER 3 LITERATURE STUDY

This chapter gives an overview of the researched literature on the subject of traffic behaviour around intersections and forecasting techniques. The aim is to acquire more knowledge on arrival patterns on (sub) urban roads, queuing models, and candidate models for forecasting.

STUDY OF VEHICLE ARRIVALS

Driving behaviour can be classified among the following strategies (Mashros, 2007):

- Free flow
- Car following
- Lane change
- Overtaking
- Presence of on-coming vehicles

These strategies generally describe the interactions among vehicles on a road stretch. The consequence of these interactions is that a departure profile will continuously change over time resulting in a different arrival profile.

On a more detailed level one can describe the dynamics between two interacting vehicles using a number of different car- following models. For a detailed comparison of car- following models the reader is referred to Olstam and Tapani (2004). The bottom line of driving behaviour in presence of other vehicles comes down to readdressing the driver's trailing distance and speed and the other driver's behaviour and consequently react.

The combination of multiple degrees of freedom (lane, acceleration and desired speed) and many influencing factors like other vehicles, weather, road geometry and incidents make individual arrivals very unpredictable. Individual arrivals can therefore be described by a probability distribution. The inter arrival times of vehicles can vary from long to very short. A distribution shapes how often a inter arrivals time is likely to appear. Duinkerken (2009) name a series of distributions that represent real stochastic events. If the probability distribution of a process is known it can be used for further development of a model. Probability distributions can also be implemented in simulation environments to mimic real life events. Some of the named distributions are a Normal distribution, Exponential distribution, Uniform distribution, Erlang-k distribution.

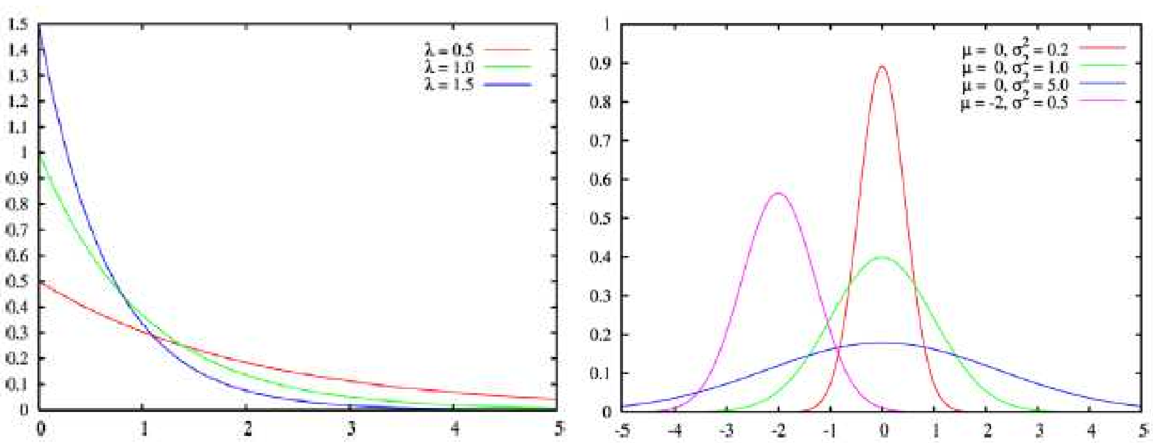


Figure 16 Exponential distribution (left) and Normal distribution (right) (Duinkerken, 2009)

A special situation emerges when vehicles are forced to leave as a group (Mashros, 2007). The group departure is also referred to as a platoon. Platoons tie individual vehicles together between the rear and the platoon leaders. Platoon leader's behaviour can be very influential to the group behaviour, it can determine its overall travel speed or cause shockwaves. The platoon rear on the other hand can determine a large part of the observed platoon compactness; with a pushing rear platoons can remain tighter or vice versa. Aside from studying individual behaviour it is therefore also possible to study group behaviour and develop appropriate models. Some of the characteristics that describe platoon behaviour:

- Inter platoon headway
- Intra platoon headway
- Platoon size
- Platoon speed

Headway is the time difference between the head of the leading vehicle and the head of the following vehicle.

Headway
$$H = t_{i+1} - t_i \tag{1}$$

Where:

H is headway, or time difference between two successive vehicle detections.
i is vehicle number.

Inter platoon headway is the time difference between two leaders of successive platoons. Intra platoon headway describes the average headway of vehicles within a platoon. Platoon size is the time between the first and last vehicle in a platoon. Platoon speed can be the average speed of all vehicles in a platoon, or of the average of the first and last vehicle.

Platoons can be defined by a group of cars that is able to depart during a green phase. This requires knowledge of the signal timing history in order to identify platoons. The other approach is to define platoons based on a critical headway. In other words, if the time-spacing between two successive vehicles is larger than a threshold value (critical headway) a cut is made and each car is assigned to its own platoon. Jiang, Li, & Shamo (2003) have investigated 30.000 different headway measurements and concluded that 2.5 seconds is a proper value.

The grouping effect weakens as a platoon moves downstream. This phenomenon is also known as platoon dispersion. Platoon dispersion is a thorn to traffic control because it makes coordination of signal timings over multiple intersections very difficult.

Traffic that initially leaves an upstream intersection in a tight platoon tends to fall apart while moving down a link. Drivers have different desired speeds and maintain different safety distances between cars. Cars at the head of the platoon will rush forward or can keep an entire platoon back. In the back of the platoon cars have a slower start up and tend to fall behind. Figure 17 gives a schematic representation of platoon dispersion. Take notion of the 100% saturation which indicates that a road segment has a maximum capacity. Also important to note is that the head of a platoon is actually drawn by the left side of the distribution and the tail is depicted by the right; at first sight this might be somewhat counter- intuitive.

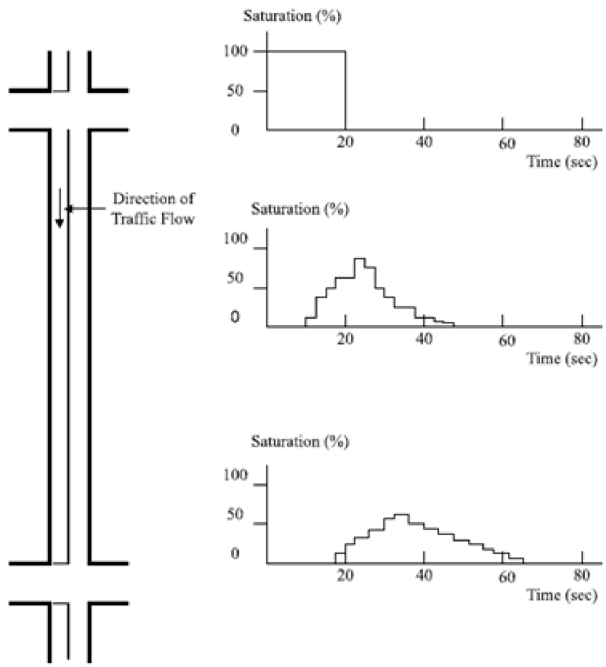


Figure 17 Platoon dispersion (Qiao, Yang, & Lam, 2001)

Platoon dispersion will increase with longer travel times (longer links). This makes the timing of sequential signalized intersections more difficult if a green wave is desired. In respect to the maximum speed limit's effect on platoon dispersion no literature was found that describes the relation. Platoon dispersion also shows a relationship with traffic densities. If a city has dense traffic, platoons will disperse less and remain compact (Mashros, 2007).

Two classical models that forecast the arrival of platoon sections under influence of dispersion are generally mentioned in platoon dispersion, one by Pacey (1956) and one by Robertson (1969). The models are probabilistic in nature. This means that they model the reality by acknowledging the probabilistic occurrence of events. Because the input of the model is sampled from the probability distribution the outcome will be different each time.

Pacey (1956) developed a probability based model where an individual car's speed is sampled from a normal distribution. The speed is then assumed to be constant for the rest of the link. The different speeds will result in a dispersed arrival profile. Mathematically the downstream arrival flow is expressed as:

$$q_d(j) = \sum_{i=1}^{j} q_o(i) g(j-i)$$

$$q = N/T \tag{2}$$

$$g(L) = \frac{a}{L^2 S\sqrt{2\pi}} \exp\left[-\frac{\left(\frac{a}{L-\bar{v}}\right)^2}{2S^2}\right]$$

Where:

q_o and q_d are the upstream and downstream flow rate at time j.
N is number of vehicles counted by detector during time period.
T is the interval.
L is the travel time between two points.
$g(L)$ the normal distribution.
a is the length of the surveyed road segment
$\bar{v}$ is the mean speed.
S is its standard deviation.

Robertson(1969)wrote a recurrent equation to describe the next state depending on the previous state. The new state is found by multiplying the old state with a smoothing function:

$$q(t+T) = F \cdot q_t + [(1-F) \cdot q_{t+T-1}]$$
$$F = 1/(1+\alpha\beta T) \tag{3}$$

Where:

$q(t+T)$ is the flow rate at time interval T.
$q_0$is the upstream flow rate at t = 0.
F is the smoothing factor.
T is 0.8 times the mean travel time.
α is an empirical platoon dispersion coefficient and varies per road link.

Both models work well in free flow environment with no disturbances. However in reality disturbances and vehicles behave non- linear, also cars are not likely to maintain a constant speed. The use of these models further in this paper at this point is questionable because both models rely on upstream information to do a forecast. However it is important to note that dispersion behaviour can generally be described by a normal or geometric distribution. Also the empirical factor makes for loss of generality. The empirical factors were often deducted from aggregated data analysis which is time consuming.

Bonneson et al. (2010) tried to tackle the generality issue by fitting a nonlinear equation to the empirical values for various situations. The above calibration coefficients are usually determined using an aggregate flow profile. The functions can be used in a wide range of street segments without the need for local calibration. The paper also displays an enlightening diagram of different departure flows depending on the signal group and how they could be combined, see Figure 18:

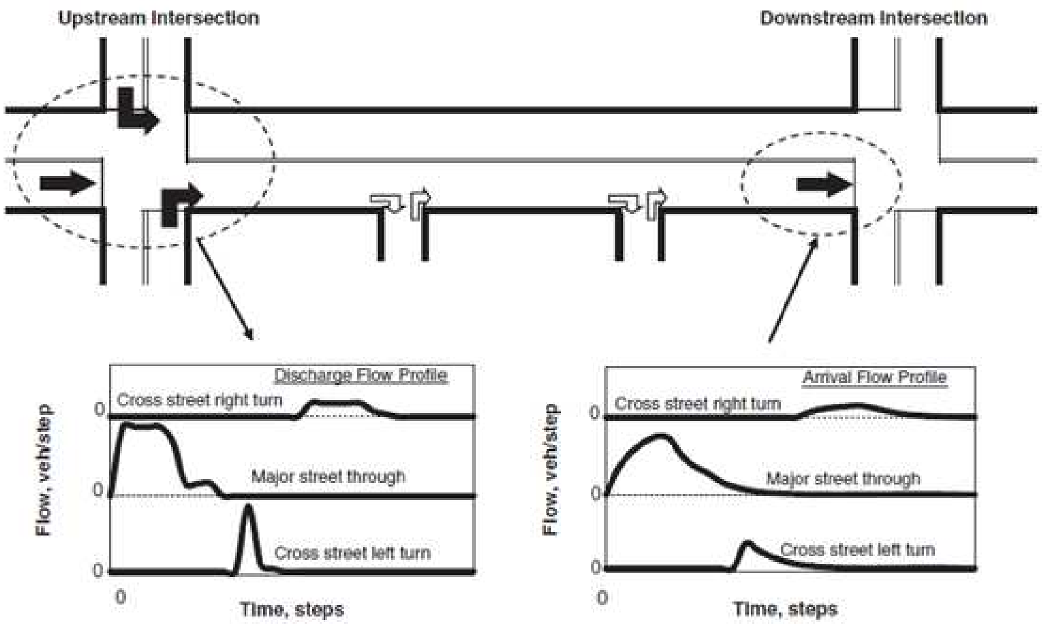


Figure 18 Different arrival profiles depending on signal group (Bonneson et al. 2010)

In view of this paper, analysing arrival profiles on a second by second basis it is important to keep in mind that in general three types of alternating arrival profiles will appear.

QUEUING MODELS

During red time vehicles accumulate in queues waiting for the next green phase. Queues are important for several reasons. From a driver's point of view queues contribute largely to a negative travel experience because every moment that the vehicle is idle the vehicle is not getting closer to the destination. Queues determine for a large part the delay of a vehicle. Delay is defined as the time that a vehicle needs to pass the intersection including waiting time in the queue minus the time that is needed to pass the intersection without interruption (in free flow). Queues also cause relatively high local congestion because of the frequent stopping and acceleration of vehicles. The law of conservation of mass sheds more light on an additional problem that comes with queues:

$$N(t) = N(t-1) - T(q_{out} + q_{in}) \qquad (4)$$

Where:

N is the amount of vehicles in the queue at time t
T is the time interval (s)
q_{out} is the flow rate of vehicles leaving the front of the queue (veh/s)
q_{in} is the flow rate of vehicles arriving at the back of the queue (veh/s)

If $q_{out} > q_{in}$ then there is no problem and the queue is called under-saturated because the signal group can unwind the queue faster than it is growing. A zero- vehicle queue remains after every green phase. In case that $q_{out} < q_{in}$ a few vehicles are left waiting for the next green phase. If both terms remain constant or q_{in} increases even more then vehicles will accumulate and the queue will increase over time. In extreme cases this can cause

blockage of another link or a spillback. Spillback means that the back of a queue has reached the upstream intersection causing a blockage and impossibility of the upstream intersection to dissolve its own queues. This can spiral out of control. When q_{out}<q_{in} the queue is saturated.

For the controller two aspects of queues are important: knowing maximum queue length in metres and knowing the amount of vehicles in the queue. Knowledge of the maximum queue length helps the controller anticipate spillback or blockage. In addition, the controller minimizes the total delay in the system; knowledge of the amount of vehicles in the queue plus the expected arrivals directly translates to delay on a link and serves as controller input.

The difference between amount of vehicles and length of the queue requires more insight than simply multiplying the amount of vehicles times the average vehicle length plus a safety distance. First of all the concentration of trucks and vehicles varies over time, trucks count as 1 vehicle but take up space of 3 vehicles. If the concentration is not known than conversion from amount of vehicles to occupied space will be biased. Furthermore the back of the queue is not decreasing synchronous with the vehicle departures at the front of the queue, there is a lag before the last vehicle in the queue can take off. This results in a long queue length but with a relatively small amount of vehicles in the queue.

In literature two main approaches are studied in respect to queues. The classical model uses the above law of conservation to keep track of the amount of vehicles in the queue. The classical model is extended with a Kalman filter in (Vigos.G & Papageorgiou, 2010)to give a better estimate of the amount of vehicles in a queue subject to measurement errors. The Kalman filter is discussed in more detail in the next section. The other research approach makes use of shockwave theory to model queue dynamics (H. X. Liu, Wu, Ma, & Hu, 2009).

The biggest difference between both models is the so-called vertical versus horizontal queue perception. The classical model is a vertical queue and could be visualized as all vehicles stacked on top of each other occupying one unit of space; when a vehicle departs it is taken from the bottom and the tower decreases by one. However, still only once space is occupied. The shockwave model puts the vehicles behind each other so that they occupy 'real' space. It can therefore also model the aforementioned lag of the last vehicle departure. In that sense the shockwave model is closer to reality. Both the classical model with Kalman filter and the shockwave models are at the forefront of queue modelling yet the estimates are still insufficiently reliable.

The biggest problem with both models is the influence of measurement errors. Detectors are not ideal and can miss a considerable amount of vehicles. By studying equation 2 it can be seen that a measurement error results in a bias for the next queue length update. The error accumulates with each new update of the residual queue. The addition of the Kalman filter to the classical model aims to correct for the measurement error.

Both models rely on an entry and an exit detection point to make the right estimates. In case the entry detection point is not available an arrival distribution can be assumed. Depending on the arrival distribution numerous equations are formulated that estimate the amount of vehicles in the queue (Duinkerken, 2009).

FORECASTING TECHNIQUES

Forecasting is attempting to "predict" the future values of a sequence of data.

Aforementioned probabilistic models by Robertson and Pacey take an explanatory approach to the phenomena and try to predict based on some underlying mechanism. More recent models take the underlying mechanism as a black box and predict based on empirical data. These models have been heavily researched in the past half century and have shown considerable advancements even so that they are able to predict better than the abovementioned classical methods. With these models the underlying mechanism is implicitly taken into account by analysing the past. Due to its generality they can be applied to many different types of data,

including traffic flows. This also means that they can be applied to the links of an asymmetric intersection. These models can be classified among two types.

"The first assumes a specific functional form and then estimate the parameters of this functional form. This approach is called parametric estimation. Alternatively, distributions or functions can be estimated directly based on observed historical data, without assuming any *a priori* functional form. This approach is called nonparametric estimation." (Talluri & Ryzin, 2004). Choosing between a parametric or non-parametric approach to forecasting is a basic design decision. Also important to note is the difference between estimations and forecasts. Estimations are derived values for the internal part, or parameters, of the equation where a forecast is directed at the output of the equation. The (parameter) estimations thus affect the quality of the forecast.

The most important parametric models include 'Structural' forecasting, 'Time Series' analysis, Bayesian methods, and Kalman filtering. Non parametric methods include machine learning or more specifically 'Artificial Neural Networks' (ANN), 'Support Vector Machines' (SVM), Fuzzy Logic, Chaos theory and Kernel Density Estimation.

Some important constraints to keep in mind while continuing throughout this section are:

1) Accuracy: the model should be able to predict well for a 120s horizon. This is dictated by the fact that the controller optimizes for a 120s horizon.
2) A model that can forecast well based on a single detection point.
3) Computational efficiency e.g. can it deliver good forecast results in less than a second?
4) Robustness (performs well for all saturations)
5) Generic (specific case or generally applicable)
6) Online learning capability

STRUCTURAL FORECASTING METHODS

Also referred to ad hoc solutions they apply to one specific situation. This is because they proceed by assuming a compositional structure of the data, breaking up and composing the series into hypothesized patterns.

A series is composed of a level, trend and seasonality component. Level describes the average value that the series fluctuates about. Trend is a predictable increase or decrease over time. A seasonal component describes a periodically returning pattern.

Moving average or Historical average (MA) and exponential smoothing are the two most important structural forecasting methods. MA looks back a predefined period of time and then takes the average of the nearby area; this time-window 'moves' along as time progresses in the future. This allows for including trend whereas a normal average would average out the effect of trend because it considers only the whole dataset. Exponential smoothing forecasts a value based on past values but the values that lie further in the past are given less weight and thus 'smoothed out'.

Holt- Winters (Holt, 2004) is a more advanced exponential smoothing method that incorporates trend and seasonality. It consists of three update functions for each component: level, trend, and seasonality and one forecasting function.

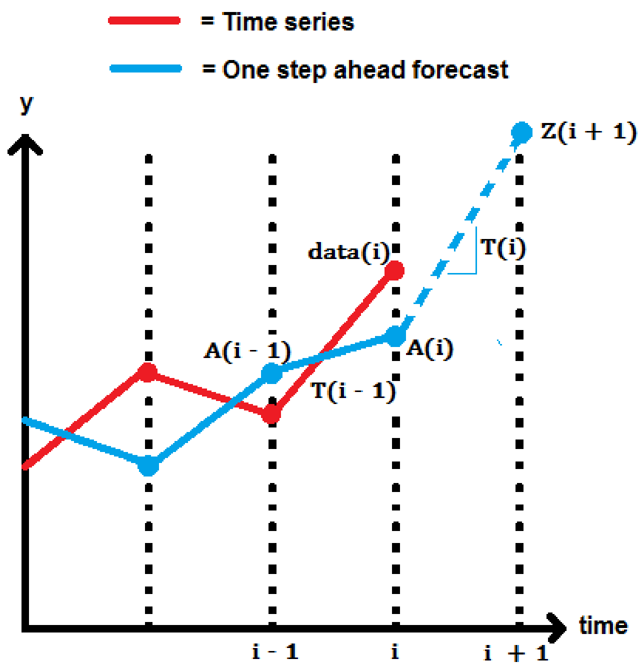


Figure 19 Visualisation of Holt Winters

The level is controlled by the parameter *alpha* which shifts the balance between the last recorded data point (i) and the previous forecast for this data point (i) as a starting point for the next forecast:

$$A(i) = \alpha \cdot \frac{data(i)}{S(i-L)} + (1-\alpha) \cdot (A(i-1) + T(i-1)) \tag{5}$$

Seasonality is not shown in Figure 19. Seasonality acts as a normalization of the effects, when the forecast is made it is 'de-normalized' taking into account the past seasonal component for one step ahead (equation 14).

Second, the trend is controlled by the parameter *beta* which shifts balance between the gradient which led to the previous forecasting of data point (i) and the gradient which led to the forecast of the point before that.

$$T(i) = \beta \cdot \big(A(i) - A(i-1)\big) + (1-\beta) \cdot T(i-1) \tag{6}$$

The last function updates seasonality. Seasonality is controlled by the parameter *gamma* which shifts the balance between the newly calculated seasonality at point (i) and the corresponding seasonality factor one period back S(i-L). This newly calculated seasonality factor won't actually be used until after one full period which also becomes clear from the forecasting function.

$$S(i) = \gamma \cdot \left(\frac{data(i)}{A(i)}\right) + (1-\gamma) \cdot S(i-L) \tag{7}$$

And the forecasting function:

$$Z(i+1) = (A(i) + T(i)) * S(i+1-L) \quad (8)$$

The research by Ghosh, Basu, and O'Mahony (2004) showed a better performance of Holt Winters over a seasonal ARIMA method for traffic forecasting. ARIMA will be discussed in the next section. The responsive nature of the algorithm is a plus because it takes into the most recent past data, Holt Winters determines an optimal historical trail. This ensures robustness and generality. On the flip side the responsive nature means that a lag exists between new information and applied forecast. For instance a sudden detection of high congestion is detected and applied in the new forecast, at this point the congestion might have already been resolved.

TIME SERIES MODELS

With time series models it is important to differentiate between stationary and non-stationary series. Even non- stationary series can be modelled but first need to be differenced to a stationary component. If a time series $z_i, \ldots\ldots, z_n$ is described as follows:

$$Z_t = \mu + \xi_t - \psi_1 \cdot \xi_{t-1} - \psi_q \cdot \xi_{t-2} \quad (9)$$

Where $\mu$and ψ are constant parametersandξ is the stochastic disturbance. The time series is said to be stationary if each disturbanceξ is sampled from the same normal distribution with mean zero and a standard deviation all independent of time *t;* $\xi_t \sim N(0, \sigma_\xi^2)$. Equation 9 is used to describe a Moving Average (MA(*q*)) process if a finite number of ψ suffices (not to be confused with earlier mentioned historical average). From (9) the following equation can be derived by continuously substituting itself (Talluri & Ryzin, 2004):

$$Z_t = \delta + \xi_t - \theta_1 \cdot Z_{t-1} - \theta_p \cdot Z_{t-2} \quad (10)$$

This equation describes the Auto Regressive (AR(p)) process. Both formulas can be combined to form an ARMA process. This can be done to reduce the number of parameters (*p,q*). Search techniques can be applied to find the right parameters by minimizing the error between the fit and the series.

If a time series shows a trend or seasonality the series is non- stationary. The series is then often differenced *d* times to find a stationary series. Again the ARMA model is applied to differenced series. This process is called the Auto Regressive Integrated Moving Average (ARIMA(p,d,q)) process. The result is then transformed back to incorporate the trend/seasonality.

Williams and Hoel (2003) perform a study into a seasonal ARIMA process to forecast one step ahead 15 minute aggregated intervals for a highway. They first perform an autocorrelation study to show that high correlation for a data point is found with exactly one day and one week before; this strengthens the use for a seasonal ARIMA method. They show that the seasonal ARIMA outperforms a historical average method.

These methods rely on the representativeness and recurrence of the training set. The parameters are fitted once and consequently used to forecast the new time series. The weakness is the inability to react to incidental

differences. On the plus side the forecast is less likely to suffer from lag if the training was done properly and recurrence is high.

KALMAN FILTER

A Kalman filter requires a model of the system as a set of dynamic equations(Taragna, 2011). This model is represented in a so called state- space model. The purpose of the Kalman filter is to have a 'second opinion' on the observed new state. Since the observed measurements and the Kalman model's prediction both describe the same system you can use them parallel to better identify the true measurement and its error. The so-called 'Kalman Gain' parameter tweaks whether more trust is put in the measurements or the model's predictions.

What if the system is a black box and does not lend itself for a set of dynamic equations like for instance the law of conservation for queuing? It is still possible to write most time –series models to a Kalman state space representation.

To further illustrate one could write an AR time series model (10) in the following state space representation:

In the following state equation:

$$\begin{bmatrix} Z_t \\ Z_{t-1} \\ \delta \end{bmatrix} = \begin{bmatrix} \theta_1 & \theta_2 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Z_{t-1} \\ Z_{t-2} \\ \delta \end{bmatrix} + \begin{bmatrix} \xi_t \\ 0 \\ 0 \end{bmatrix} \sim \boldsymbol{y}_t = \boldsymbol{A} \cdot \boldsymbol{y}_{t-1} + \boldsymbol{v}_t$$

(11)

and measurement equation:

$$Z_t = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} Z_t \\ Z_{t-1} \\ \delta \end{bmatrix} + 0 \sim \boldsymbol{z}_t = \boldsymbol{H} \cdot \boldsymbol{y}_t + \boldsymbol{\xi}_t$$

The state and measurement equation are used to forecast the next state at which point they are updated according to the error and the Kalman gain parameter. For more in depth analysis of the updating procedure the interested reader is referred to Taragna (2011).

The Kalman filter becomes an adaptive times series model and can be used to make a forecast of the future value. This is interesting because it tackles the main disadvantage of the time series model. The difficulty of a Kalman filter lies in the proper initialization of the model and determining the Kalman gain. The Kalman gain needs to be determined for every individual case. This is key to getting good results.

The Kalman filter was researched by Xie et al. (2007) in combination with a discrete wavelet decomposition (DWT) as a pre-processor. He used a dataset of 5 minute intervals to forecast the next time step. The DWT was used as a low pass filter to de-noise the dataset from high frequencies and expose the underlying true form. At first he shows the improvement of the direct Kalman filter over a historical (moving?!) average method. Xie et al. also show that de-noising is recommended when using the Kalman filter.

Hu, Madanat, Krogmeier, and Peeta (2001)use the Kalman filter to make better estimations of the amount of cars residing in a highway segment. Here the Kalman filter is used as it is predestined because the system dynamics can be described by the law of conservation. The Kalman model is used to evaluate the detector measurements. Hu et al. (2001) shows that including the Kalman filter can give better estimations of the vehicle counts.

ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks are inspired by research performed on human's central nervous system. The ANN model takes the shape of the bio physiology of the brain (TUDelft, 2008). ANN is classified among non-parametric methods because it does not assume a set function beforehand of which parameters are estimated and tuned. ANN tune so-called weights of the neurons in the network by learning just like a human would learn from his mistakes. Through this, in theory, any nonlinear function can be fitted after learning.

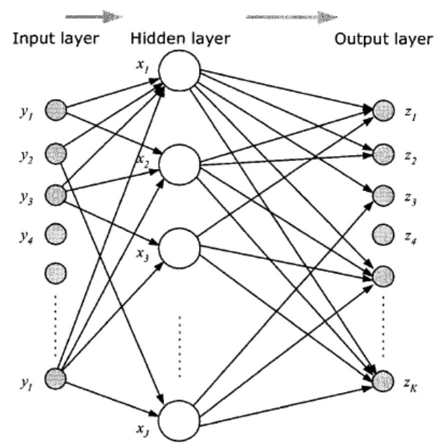


Figure 20 General Artificial Neural Network (Talluri & Ryzin, 2004)

The input layer holds static input nodes (dendrites). One or multiple hidden layers hold the neurons. The choice in amount of neurons N is arbitrary but important. Too many will result in over fitting which makes the trained function only applicable to the specific trained set. Too few will result in under fitting and won't produce accurate forecasts for new data. The static nodes are connected to each hidden neuron through weights x_N, the weights are trained parameters. At the hidden neuron the weighted inputs are summed and passed through an activation function. The choice of function is arbitrary as long as it can produce a threshold. A sigmoidal function is often used.

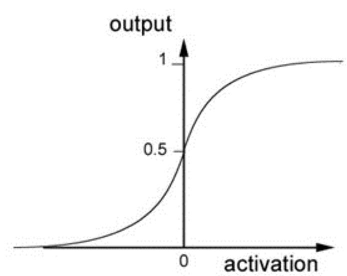


Figure 21 Sigmoidal function

ANN generally take the shape of Figure 20 but can have many variations. Most important ANN networks that apply to this thesis are the Nonlinear Autoregressive Neural Network (NAR) and the Nonlinear Autoregressive Neural Network with external input (NARx) (Beale, Hagan, Demuth, 2012).These are ANN models with specific purpose of fitting time-series and use the trained network for forecasting purposes. The NAR network is useful to perform extrapolation forecasts based on a recent data set. NARx does the same but can take into account other possible exogenous factors, like for instance historic data, speed v or the change in speed dv.

Qiao, Yang, & Lam(2001) perform a comparative study of a NARx network in comparison to the models of Pacey and Robertson. As input parameters they use the upstream flow rate as well as the upstream speed. They require 12 neurons to get a good fit for the flow rate downstream. They do aggregated flow rate

estimations for 5 second time intervals. It can be seen that the neural network clearly outperforms both classical models.

Kirby, Watson, and Dougherty (1997) perform a comparative study between a Neural Network and ARIMA time series. They use three upstream locations plus downstream location in most likely a NARx model (not mentioned).They required around 30 hidden neurons to fit the data. They used the ANN network to forecast one data point ahead which is equivalent to half an hour aggregation. This makes it difficult to evaluate the forecasting potential in respect to this thesis because a 30 minute aggregation level far exceeds 120seconds. For this aggregation level the time series model slightly outperformed the neural network.

Dia (2001) uses an ANN to forecast downstream arrivals on a highway. Data is aggregated in 20 second intervals. Both upstream speed and flow rate are used. In line with the abovementioned studies the network performs really well (93% accurate) if upstream speed is included.

The neural networks are debatable due to its black box nature. One can argue that it is not wise to heavily depend on a method where one does not fully comprehend the inner workings. Yet many papers show that 'very' good results can be achieved nonetheless. The neural network approach is similar to the time series models in that they rely on a training set to acquire a predictive model. The neural network, however, can also be made adaptive by incrementally adjusting the weights based on new information. The potential shown by researched papers in respect to the more classical methods make the neural networks an interesting candidate for further application.

SUPPORT VECTOR MACHINES/REGRESSION (SVM/SVR)

Support vector machines/regression (Basak et al., 2007) try to find subtle patterns in data and classifies the input among two sets (binary). Before being able to do this the algorithm needs to learn how to distinct the classes; this is done by supplying a training set where classes are predetermined. Figure 22 illustrates the problem of how to correctly classify the data points:

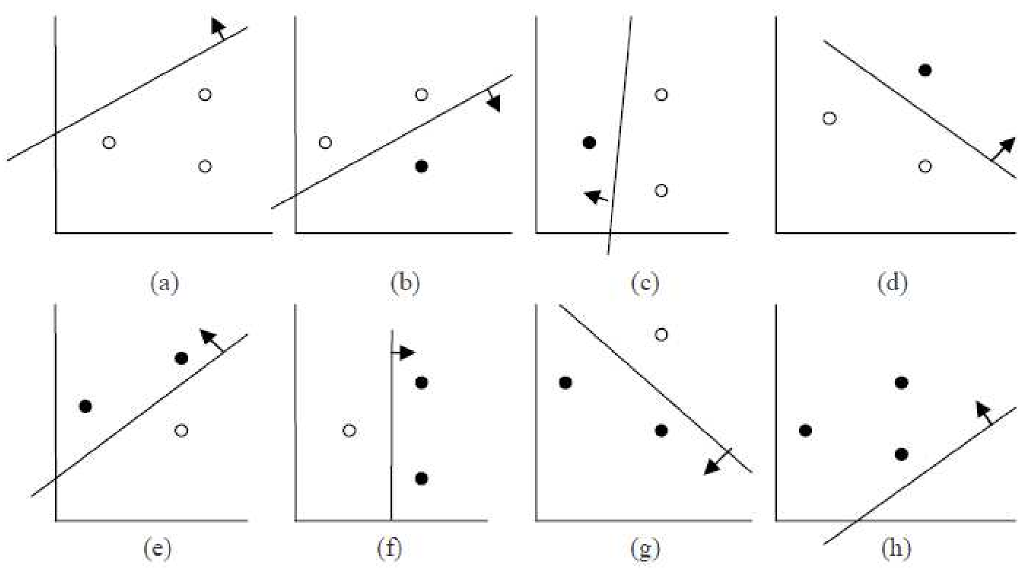


Figure 22 SVM classification (Basak, Pal, & Patranabis, 2007)

There are many types of hyper planes that might classify the data but the original one maximizes the separation between the data points and the hyper plane (linear classifier). The techniques have improved over

the years that they now also allow or non-linear hyper planes (in Figure 22 the line or hyper plane would not be straight but with a curve adding/deleting more data points from its class. SVM can be used to generalize complicated classification decision for the so-called 'grey' area.

The mechanism of Support Vector Machines can also be used for regression (SVR) so that a function is fitted on a dataset. "Traditional/statistical regression procedures are often stated as the processes deriving a function f(x) that has the least deviation between predicted and experimentally observed responses for all training examples. One of the main characteristics of Support Vector Regression (SVR) is that instead of minimizing the observed training error, SVR attempts to minimize the generalized error bound so as to achieve generalized performance. This generalization error bound is the combination of the training error and a regularization term that controls the complexity of the hypothesis space."(Basak et al., 2007) This can be interpreted more easily when considering the outliers of a dataset, traditional regression methods will treat each data point equally and therefore the outliers will also be included equally. SVR will judge a data point before including it in a regression. Outliers are more likely to be excluded resulting in a 'generalized error bound'.

X. M. Liu and Lu (2007)collect detector data for 2 minute intervals and use SVR to predict the aggregated traffic flow at the downstream intersection in section similar to Robertson's model. A big question mark arises in how they capture the dispersion if data is aggregated over 2 minutes; this is not clearly explained in the paper. For training the upstream dataset is used as input and downstream dataset as target. They show that SVR outperforms Robertson although their methodology is questionable.

With Support Vector Regression the forecast will improve if more comparable situations can be found in the historic data. The quality of the forecast is therefore heavily reliant of the quality and size of the dataset. An increased dataset also means increased computation. Based on these two requirements SVR at this point is not a viable candidate for further research.

FUZZY LOGIC

Fuzzy logic is a methodology that can map human knowledge to a set of mathematical rules. 'Human knowledge' is translated to non-crisp inputs. For instance instead of set of ages 1 to 80, one can define 'young', and 'old'. In addition one can now do an analysis by requesting what happens if someone is only 'slightly old'. There are many more applications but in the field of traffic none were found that use fuzzy logic solely for forecasting.

Yin, Wong, Xu, and Wong (2002) developed a hybrid model using fuzzy clustering to segment the data and apply the appropriate neural network. The big differences in flow rate that occur during day are better fitted with this fuzzy-neural model. Positive side effect is that computation times are also lowered. The model also uses upstream information as input, data is aggregated in 2 minute intervals. Even so the addition of fuzzy systems seems to be beneficial for overall control regardless of the aggregation level.

BAYESIAN METHODS

Bayesian methods merge a prior belief about forecast values with information obtained from observed data. The methods are especially useful when there is *no historical data*, so one assigns a believed probability for an occurrence. This probability is then updated along the way as more relevant information is gathered. It is a useful method for when for example new products are introduced and no information is known about demand. Bayesian forecasting could be useful in the sense that as time passes and at each time instance that not arrival is detected the chance increases that cars should arrive any time soon. Nevertheless this method is not further investigated since plenty of historic data is available. It seems logical to take this knowledge into account when doing a forecast.

LITERATURE EVALUATION

The literature study revealed that individual arrivals are very unpredictable due to complex vehicle interactions. This is the case with upstream detection even more so with downstream detection. Downstream three different arrival profiles can be expected in coherence with the upstream signal cycle.

Many studies use aggregate forecasting in the case of individual arrivals and predict the demand progression for the next time interval. Aggregate forecasting is the designated way to forecast arrivals when vehicles arrive individually. The question remains how this aggregate forecast should be implemented in the controller and which forecasting technique is best suited.

Besides individual arrivals traffic flows also show group arrivals. The grouping effect, or platoons, allow for new possibilities to look at arrival profiles. Additionally it also provides us with some extra tools to predict arrivals. Platoons are under influence of dispersion which again is direct result of individual behaviour and make forecasting platoons difficult on another level.

An arrival profile that shows platoons could also be forecasted by means of aggregating demand. However, aggregation of platoons at this point seems unjustified because important information on the interspacing between two platoons will be lost. Especially this interspacing between platoons can be very important for a controller because it could supply a small time window to let side streams pass. Based on these findings individual arrivals and platoon arrivals will be treated separately. Moreover each requires its own forecasting model.

Robertson, Pacey, and Bonneson give insight in the evolution of platoons, yet they depend on upstream detection. Since this paper focuses on downstream forecasting these approaches are temporarily parked. In order to do forecasts based on a single detection point one needs to study the history of that detection point. This history is a series of detection events which can be translated to a time series signal. In literature a considerable amount of generic forecasting techniques are available that can forecast time series. The researched techniques included: historical moving average, Holt Winters, ARMA and ARIMA model, Bayesian methods, Kalman filter, Artificial Neural Networks and Support Vector Machines.

The historical moving average is a robust algorithm but makes no effort to find a pattern in higher frequencies (smoothes the higher frequencies). At this point it can be assumed that the aggregated demand also consists of highly frequent demand variations, it seems unjustified to remove the possible forecast of higher frequencies beforehand. The historical moving average algorithm is therefore not researched.

Williams and Hoel (2003) researched the autocorrelation of a data point with past data points and showed that for aggregated forecasting it is wise to incorporate a previous day and week day into the forecast. Holt Winters takes into account seasonality and can therefore make use of these autocorrelations. Seasonal time series (SARIMA) are also a viable candidate because of the possibility to include a seasonal component. However,Ghosh et al. (2004)confirmed a better performance of Holt Winters over a seasonal ARIMA method for traffic forecasting.

Bayesian methods are intended for forecasts where no historic knowledge is available. Since historic knowledge is available it seems logical to include this in the forecast. Fuzzy logic is not used for mere forecasting and is used as complement to a forecasting technique. Since the aim is to find a forecasting technique fuzzy logic is left for future improvements.

The Kalman filter’s true strength lies in observing and correction of the measurement error of known dynamic systems that are controlled via system measurements. Since the dynamics of the researched traffic system are unknown as well as prior knowledge of the measurement error for downstream detection the Kalman filter seems less suitable. In addition the initialization of the Kalman filter and the Kalman gain is difficult and

determines greatly the performance of the algorithm. The Kalman gain would have to be studied for every specific case. The Kalman filter is therefore no longer researched.

Literature on Artificial Neural networks shows promising results and can therefore not be ignored. The main counter argument is often the black box approach. Critics don't like that the algorithm delivers an answer which cannot be explained. But on a very abstract level; if the internal dynamics are too hard to comprehend and model is it then not justified to use a method which can replicate the symptoms?

Holt Winters is on the other side of the spectrum and is comprehensible and robust; yet the performance seems to be bounded. Therefore these two techniques, Holt Winters and Artificial Neural networks, were chosen for further research. Incidentally both a parametric and non-parametric technique are covered.

Lastly the queuing model needs to be discussed. Queuing of vehicles can affect arrival patterns if the system is saturated and queues start to increase. This can result in constant occupation of a detector and arrivals can no longer be detected. This will distort arrival patterns in the sense that vehicles are detected later than when they are travelling in free flow. This should be kept in mind with later research.

For the controller a good queuing model is needed to deliver good estimates of delay in the system and to have knowledge of the maximum queue length. In respect to delay calculations multiple discussions about the current vertical queuing model implemented in the controller led to the conclusion that the current vertical queuing model suffices. The controller does lack knowledge of the back of the queue. This field of research was abandoned because it deviates to far from the research questions. Implementation is left for future study. Appendix B discusses extra research that was done on a horizontal queuing model by Liu et al (2009). The model makes use of the shockwave theory to estimate the back of queue. Some additional improvements are suggested.

It shows that two typical type of traffic flows appear at intersections; random arrivals or platoon arrivals. Both cases will be addressed separately and can be analysed with help of the real world dataset. The Groene Kruisweg eastbound (SG: 7,8,9) direction shows platoons due to the upstream intersection. The northbound (SG: 4,5,6) direction coming from the Malledijk shows random arrivals.

NON- PLATOON FORECASTING

Before continuing it is important to verify the unpredictability of individual arrivals. This is proven by the following diagram which shows a Poisson distribution of the inter arrival times of consecutive cars at the Malledijk:

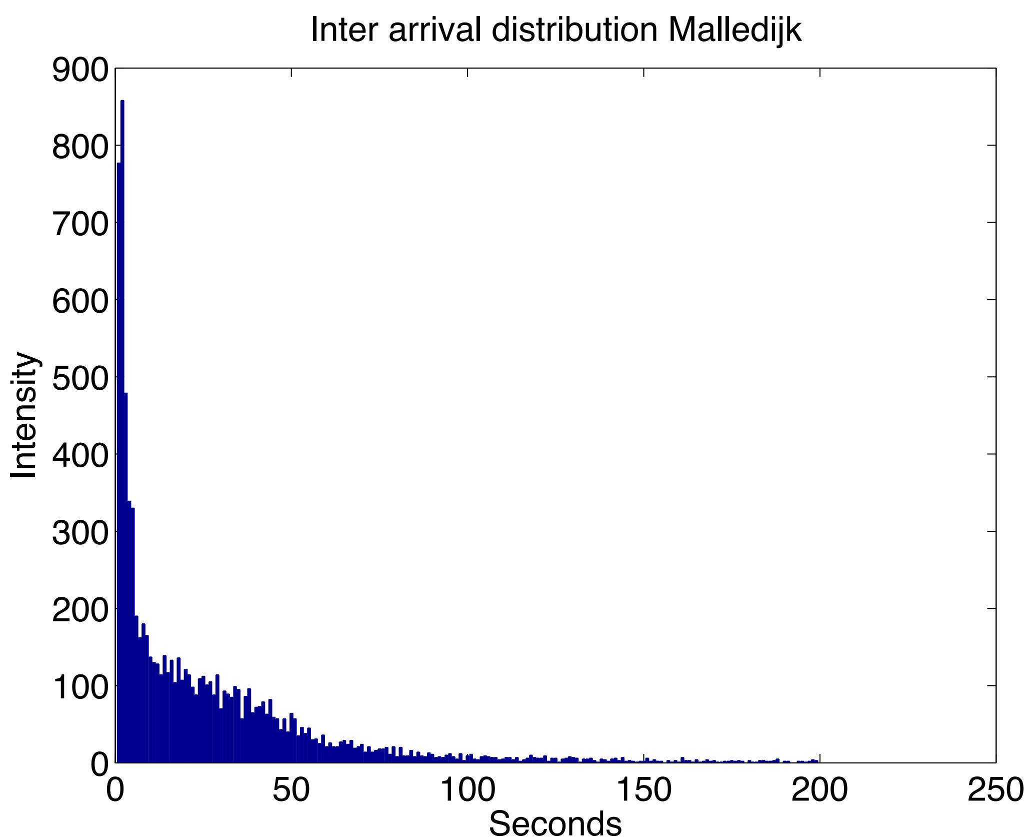


Figure 23 Poisson distribution indicates unpredictability of individual arrivals

The Poisson distribution reveals that it is not possible to predict a vehicle based on the previous one. What can be predicted, is the amount of arrivals at a certain time. If one aggregates historic data in intervals it will show a clear progression of demand throughout a day (Figure 24). Comparing these with previous days a periodic pattern can be observed. Monday until Friday show very similar time series. Saturdays and Sundays are very different from weekdays in that demand is much lower. The profile is also different, weekdays have two demand peaks, in the morning and afternoon, when people commute to work. Weekends show a slow increase of demand throughout the day reaching a peak midday and then a slow reduction to minimal demand.

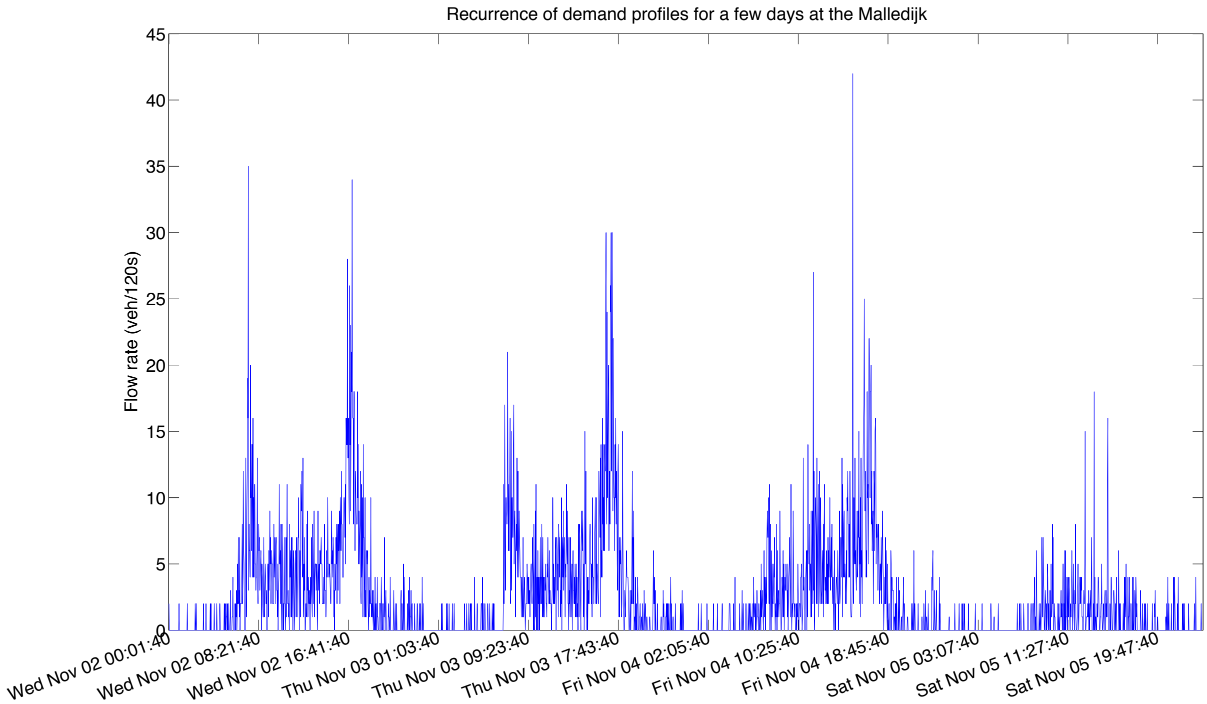


Figure 24 Aggregated demand for Malledijk

A 'continuous' signal is a prerequisite for the forecasting techniques to work properly. Aggregating arrivals into greater intervals bins will create a smoother signal but detail will be lost. Most important is to choose the interval size such that there are no more disruptions in the signal, i.e. that there are suddenly no cars present and the signal drops to zero. For the Malledijk signal an interval of 120 seconds was chosen (Figure 24). The main reason is that the interval is consistent with the optimization horizon, it can be seen that with an aggregation level of 120 seconds the signal still drops to zero at some points during the day. To solve this problem another solution was implemented; this will be discussed in the Training section.

The literature study revealed that so-called aggregated forecasting is thoroughly researched. Two promising methods are:

- Holt Winters (Exponential smoothing)
- Artificial Neural Network

Both methods will be evaluated for a one-step ahead forecast using the above shows dataset. One-step ahead means the total amount of cars that arrive in the next 120 seconds will be forecasted. This will be done for an entire day. The forecasting methods are judged by taking the root mean squared error (RMSE). Since both techniques are applied on the same dataset the RMSE is sufficient for comparison. For a more general comparison (with other papers) MAPE is advised which gives a percentage error (normalized). However MAPE is problematic in this case because the dataset also has 'zero' values which will result in an impossible division by zero.

To keep the comparison fair for both techniques a history of 1 day was used to train the weights/parameters. Most likely the performance of both techniques could be improved by increasing the historic knowledge.

HOLT WINTERS FORECASTING PROCESS

The following process was undertaken for Holt Winters. The process consists of 4 stages: initialization, training, forecast and evaluation. The first stage involves loading and preparing the necessary input. For ease of reference the datasets are named in accordance with the above dataset shown in figure 24.

The Holt Winters algorithm has three parameters which allow the algorithm to adapt to the nature of the recurring data. Similar to a race car that is tuned through various parameters to best adapt to the circuit and circumstances; type of tyres, injection mixture etc. the Holt Winters algorithm can also be tuned for optimal performance. The levers are 'alpha', 'beta' and 'gamma'.

Training is done on Thursday, the day prior to the day that will be used for validation. This means that the seasonality component is exactly one full day. It was found earlier in literature that a full week is also a good seasonal component. Unfortunately that would require a dataset of 3 full weeks (explained in next section) to be able to fully initialize and train the algorithm. The available dataset only consists of 10 successive days: 1 November 2011 to 10 November 2011.Training with seasonal component of a week is therefore not possible. The training stage involves finding the right set of parameters; this will be discussed in more detail in the next section. Important to keep in mind is that Saturday and Sunday cannot be forecast with a Seasonality of one day because the demand profile is too different from weekdays; they require Seasonality of a full week.

Thirdly the data is simulated and validated; in this case Friday. The forecast runs parallel and is one-step ahead of the data. In the last stage the forecast and validation for Friday are evaluated.

Table 2 Aggregated forecast process

	Holt Winters
Initialization	• Load aggregated datasets o If not aggregated -> aggregate • Initialize Level, Trend, Seasonality for Tuesday

Training	• Repeat until Mean Absolute Error (MAE) is minimal (Optimization of alpha, beta and gamma)
	    o New set of parameters
	    o Stage 1: Calculate Seasonality for Wednesday using the seasonality of Tuesday and the new parameters (explained in the next section)
	    o Stage 2: Train on Thursday using the seasonality for Wednesday and the new parameters
	    o Calculate MAE
	    o If MAE = minimal then save the parameters and seasonality for Thursday

Forecast	• For begin to end of Friday do every second
	    o Add new detections to database
	    o Aggregate database for 120 second intervals
	    o every 120 seconds
	        ▪ Update Level, Trend and Seasonality Friday given the new information
	        ▪ Do one step ahead forecast using the seasonality for Thursday and the optimized parameters

Evaluation	• Calculate Root Mean Square Error over Friday

HOLT WINTERS INITILIAZATION AND TRAINING

To solve the aforementioned problem of zero values in the dataset, each data point is offset by an arbitrary number of '1000'. This is due to the Level update function in the Holt Winters algorithm which cannot deal with zero values (zero detections); this results in an 'infinity' Level. Further forecasts will fail in this case. The offset is allowed because there is only one input which is being offset. All further calculation are based on (Level) or relative (Trend, Seasonality) to this dataset. The final output requires subtraction of '1000' to cancel out the offset.

All components of the algorithm, Level, Trend and Seasonality have to be initialized before the actual forecasting process can take place. This is due to the recursive nature of the algorithm i.e. Holt Winters looks back in time to make a forecast but if there is nothing to look back to then a forecast is impossible. Initialization of Level and Trend are not difficult; the forecast can wait until two time steps have passed before making a forecast. The information of these two time steps then becomes the starting position of the algorithm. For the aggregated demand profile this is even more simple because the start of a period is set at midnight, 00:00h. At this time traffic is negligible and can be assumed zero.

Seasonality is different because in the algorithm it is always called for one full period in the past. Seasonality is initialized by normalizing each data point by the mean of the period:

$$S(i)_{tuesday} = \frac{data(i)}{\overline{data}} \tag{12}$$

The next step involves the optimization of the Holt Winters parameters, alpha, beta, and gamma.

The optimization is done through a function in MATLAB called '*fmincon'*. It is a search algorithm that finds the minimum of a constrained (non-) linear function. The Holt Winters algorithm has 3 parameters resulting in a 3 dimensional search space; it can be assumed that the parameters have a non- linear relation. All three parameters are constrained within the set $0 \leq x \leq 1$. The minimum is not necessarily a global minimum. The objective of the algorithm is to minimize the Mean Absolute Error (MAE) between the forecast and the validation data. The optimization takes only a second.

The training of the parameters consists of two stages; the first is a 'second initialization' stage of Seasonality for one full day prior to the actual training day, this is Wednesday. The second stage runs a simulation on Thursday which is evaluated for its error, this error is the objective function for the optimization algorithm. Because Seasonality requires one day of initialization and two days of training are needed; a total of 3 days are needed to set up Holt Winters for future use.

The need for the second 'initialization stage of Seasonality' becomes clear from the forecast function which is used to train Holt Winters on Thursday:

$$Z(i+1)_{thursday} = (A(i) + T(i)) * S(i)_{wednesday} \tag{13}$$

The *gamma* parameter is not trained when only training on Thursday because the Forecast function of Thursday uses the Seasonality function of Wednesday. Therefore in order to also train *gamma* Wednesday needs to be incorporated in the training in addition to Thursday. However in order to include Wednesday in the training it needs to be able to look back an additional day: Tuesday. Hence, Tuesday is initialized using equation 11. Stage 2 involves including the Seasonality of one day prior to the training day in the optimization. This

training process is repeated until all parameters *alpha, beta, gamma* have resulted in a minimal error over Thursday.

For more detail regarding the execution and code the reader is referred to appendix C.

Figure 25 shows the results of the Holt Winters training process and Malledijk dataset. It is a matrix of scatterplots where each parameter is plotted against each other. The histograms on the diagonal indicate the search area for the individual parameters. The histograms can also be viewed as a blueprint of the scatterplots in vertical direction. The peaks in the histogram indicate the area where the best results were obtained for the Mean Absolute Error. The minimum MAE will therefore also be enclosed by the peaks.

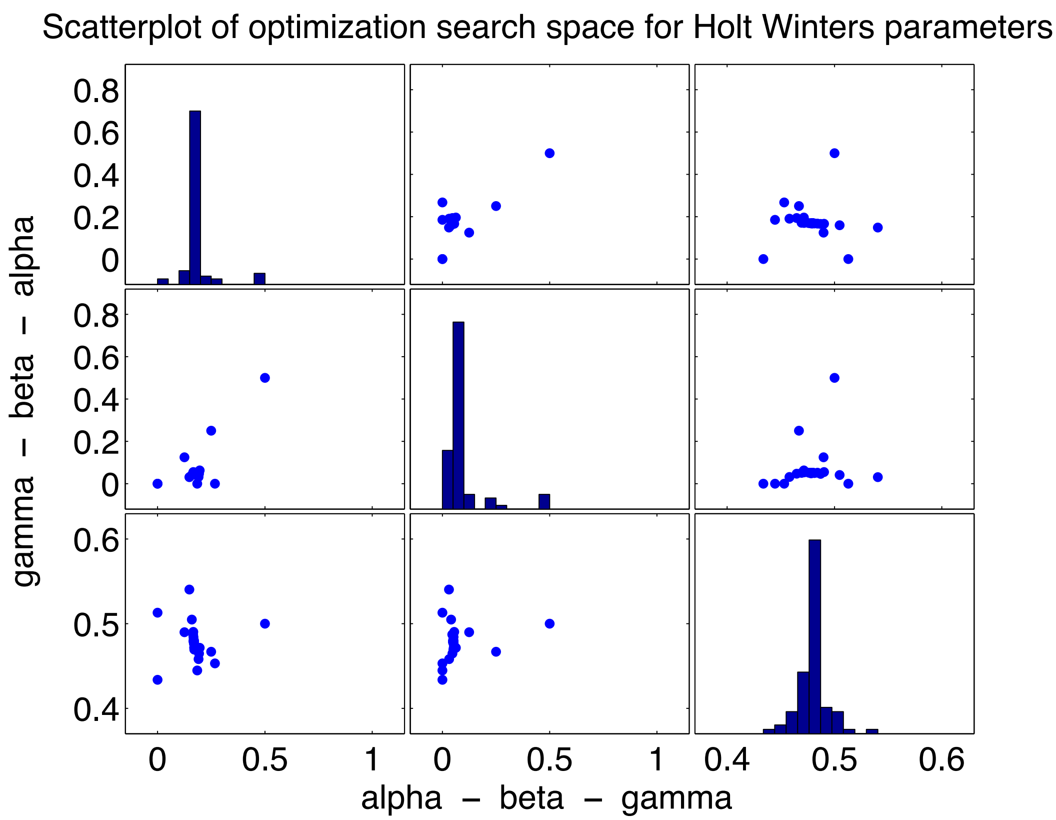


Figure 25 Scatterplot for training Holt Winters parameters

For Thursday this resulted in 90 runs to find the optimal set:

- Alpha 0.1687
- Beta 0.0509
- Gamma 0.4795

These values will later be used in the actual forecasting process. For the training set this resulted in the following error distribution:

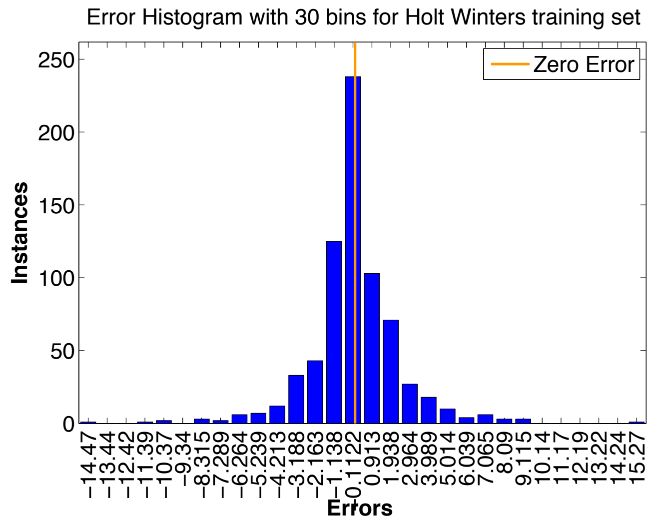


Figure 26 Error distribution of training set

The error distribution shows a good cluster around the zero error with a relatively steep descend towards larger errors. The slim profile, or 'low standard deviation', indicates good forecasting capabilities. The Mean Absolute Error weighs negative and positive errors equally and thus results in a relatively symmetric distribution. In a real world environment the algorithm will therefore overestimate and underestimate the amount of vehicles arrivals equally.

If this system were to be implemented in the real world a few important questions would have to be dealt with. When and how many times do you optimize? Optimizing after finishing a full period seems logical but optimization could also be done when a sufficient amount of new data has presented itself; so in between periods. This would require further study.

ARTIFICIAL NEURAL NETWORKFORECASTING PROCESS

The following process was undertaken for the Artificial Neural Network. The different stages are similar to the Holt Winters process. In contrast to Holt Winters, ANN has no parameters to train. However, the algorithm must also somehow learn from past observations how it should respond for the future. The equivalence, in that sense, to the Holt Winters parameters are the so-called ANN weights. As explained in the literature section the weights resemble a threshold value for which a neuron is activated.

The weights are trained by supplying an input and target dataset; the neural network will consequently train the weights such that it finds the relationship between the input and the target set. The input set can also be past observations of the target set. For predicting time series this is the case. In addition to past observations other exogenous variables can be supplied for an input. The network will try and find the relation between past observation in addition to past observations of the exogenous input. For aggregated demand forecasting the exogenous input is the dataset of the preceding day. This type of network is called a NARx network. The MATLAB neural network toolbox has a functionality to construct a NARx network. Training is done on Thursday similar to the Holt Winters training.

The Friday dataset is simulated, the trained neural network runs in parallel and delivers a one-step ahead forecast based on the most recent observations. The moment in time calls the corresponding data of the exogenous input.

Table 3 Artificial Neural Network forecast process

<table>
<tr><th colspan="2">Artificial Neural Network</th></tr>
<tr><td>Initialization</td><td>• Load aggregated datasets
    o If not aggregated -> aggregate
• Construct feed forward network with 2 inputs
    o one for Wednesday's input,
    o one as delay line for Thursday
    o and one output
• 10 steps for input delay, 15 neurons</td></tr>
<tr><td>Training</td><td>• Train the weights for Thursday using seasonality of Wednesday with Levenberg- Marquardt back propagation algorithm</td></tr>
<tr><td>Forecast</td><td>• For begin to end of Friday do every second
    o Add new detections to database
    o Aggregate database for 120 second intervals
    o every 120 seconds
        ▪ if database > input delay
            • Adjust weights given the new information (Adaptive = optional)
            • Do one-step ahead forecast</td></tr>
<tr><td>Evaluation</td><td>• Calculate Root Mean Square Error</td></tr>
</table>

ARTIFICIAL NEURAL NETWORK INITILIAZATION AND TRAINING

The same algorithms as with the Holt Winters initialization are used to prepare and aggregate the data. The MATLAB Neural network toolbox automatically prepares the dataset for three different subsets: training, validation and testing. It assigns the data points randomly via a predefined distribution. In this case training 70%, validation 15% and testing 15%. The training set is the subset used to adjust the weights on the neural network. The validation subset is used to minimize over fitting. So the weights of the network are not adjusted with this data set, it verifies that any increase in accuracy over the training data set actually yields an increase in accuracy over a data set that has not been shown to the network before. If the accuracy over the training data set increases, but the accuracy over then validation data set stays the same or decreases, then there is over fitting network and training should be stopped. The testing set is the subset used only for testing the final solution in order to confirm the actual predictive power of the network.

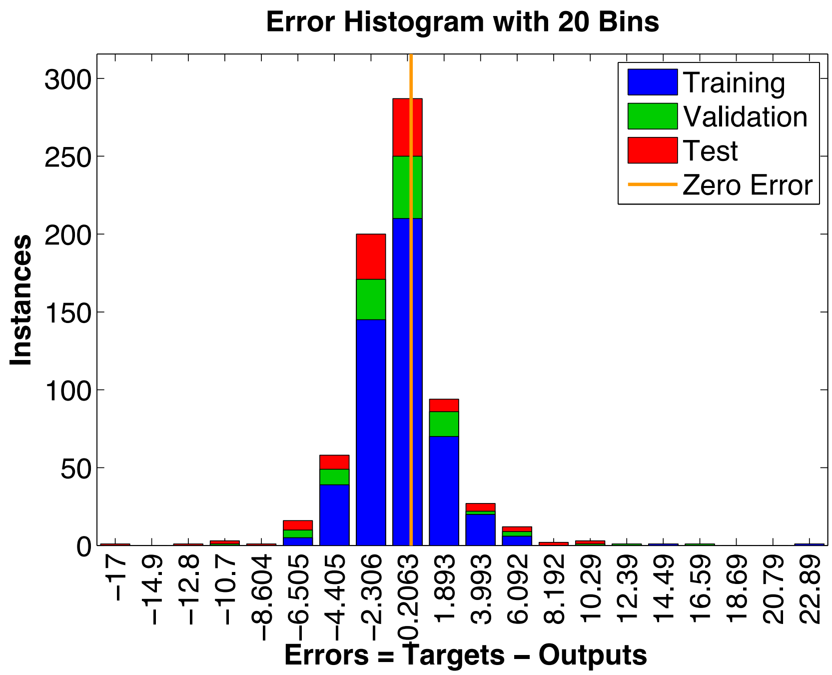


Figure 27 Error histogram for ANN training on Malledijk

The actual training is done with the Levenberg- Marquardt algorithm. The Levenberg- Marquardt(LM) algorithm is the ANN equivalent to the *'fmincon'* algorithm in Holt Winters in that it trains the weights (parameters for HW). The LM algorithm also tries to minimize the Mean Absolute Error (MAE) between the fit and the validation set, similar to the training of HW. This algorithm finds a local minimum, not necessarily a global minimum. It is important to note that a new training exercise on the same dataset will yield different results, this is due to initialization of the weights just prior to training. The initialization creates random values for the weights between -1 and 1. This can be seen in the error histogram (Figure 27) in respect to the asymmetry of the profile; the output (forecasts) tend to overestimate the target set because a bias value (general offset for all values from zero) is initialized unequal to zero.

Looking at the validation and test set in more detail it appears that the distribution over the error range is similar; for instance there are no excessive peaks for the test set in high error bins. This leads to the conclusion that the weights will perform similar with a newly presented set.

The error distribution of the neural network shows worse results than Holt Winters. Looking at the error of -4.405 for ANN it can be seen that it appears approximately 45 times, for HW this error is more in the range of 15 instances. The ANN training has a 'fatter' profile than HW which will result in lesser forecasts.

OFFLINE SIMULATION EXAMPLE

Consequently Holt Winters and ANN were used on the real world dataset of the Malledijk to forecast Friday. Both methods are evaluated with the Root Mean Squared Error:

$$RMSE = \sqrt{\overline{(target - forecast)^2}} \qquad (14)$$

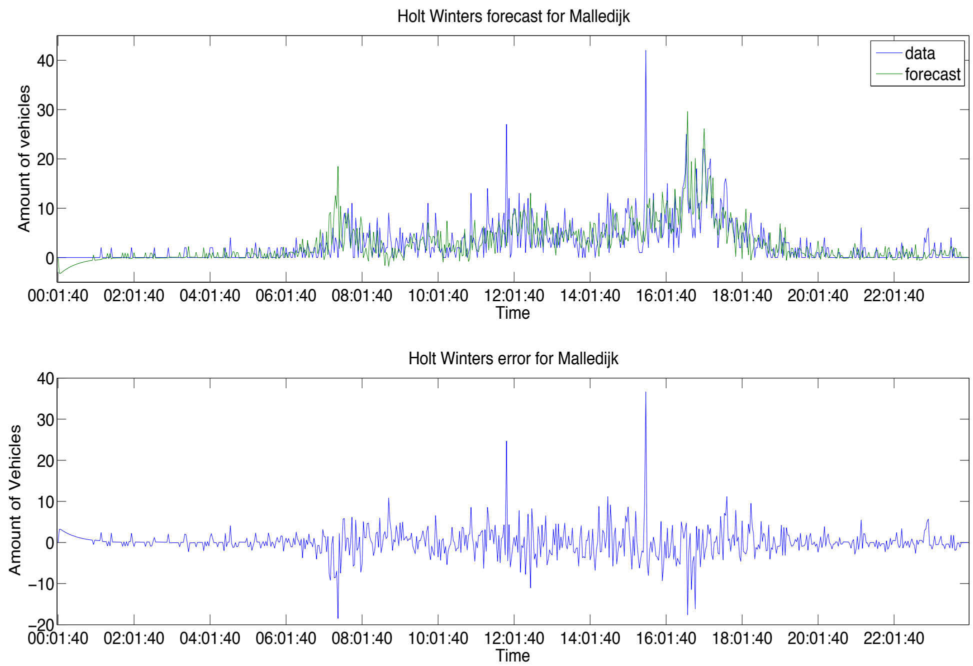


Figure 28 Holt Winters forecast for the Malledijk

Holt Winters forecast resulted in a RMSE of 3.49. From 19:00h until 07:00h there is hardly any traffic, the forecasts during this period are almost all wrong and appear for two reasons. First of all, due to the reactive nature of the algorithm adjusts its components after detection; as a result of a detection the algorithm will predict a later detection. Because the traffic demand is so low there is no assurance that a new vehicle will arrive in the next interval and the prediction is wrong.

Second, the seasonality of the previous day emerges and leads to a false prediction. A detection the previous day will lead to a partial forecast at the same exact point in time on this day. This is pointless because a vehicle arrival yesterday is an independent event of a vehicle arrival today.

The problem for 19:00 h until 07:00 h actually comes down to prediction of individual arrivals again. As shown earlier with the Poisson distribution, this is impossible. It is safer to predict no arrivals during this time window for this signal group, otherwise green will be given to ghost cars. Holt Winters is therefore better turned off during the night. In practice a single arrival could very well be solved without delay by placement of a faraway detector at 100m without the need of a forecast. The anticipation time of the controller should be enough to let the vehicle pass before a full stop is required.

Furthermore it can be seen that Holt Winters follows the general profile quite nicely. It is not able to forecast the sudden high peaks in the signal. These can happen for two reasons, detector failure or an accidental congestion. If a detector is failing than Holt Winters might actually be forecasting the real situation correctly because it follows the general trend. In the other case Holt Winters will fail because it cannot deal with highly volatile changes that are not long-lasting.

The Artificial Neural Network forecast resulted in a RMSE of 4.0. The first observation is the slight offset of the whole forecast set in the positive which was also deducted from the training error. This offset is sometimes larger or zero for different training sessions. This is due to initialization of the bias value. Apparently different initializations of the bias result in different local minima. Obviously this is an unwanted feature for real world implementation. It could be solved by fixing the bias to zero.

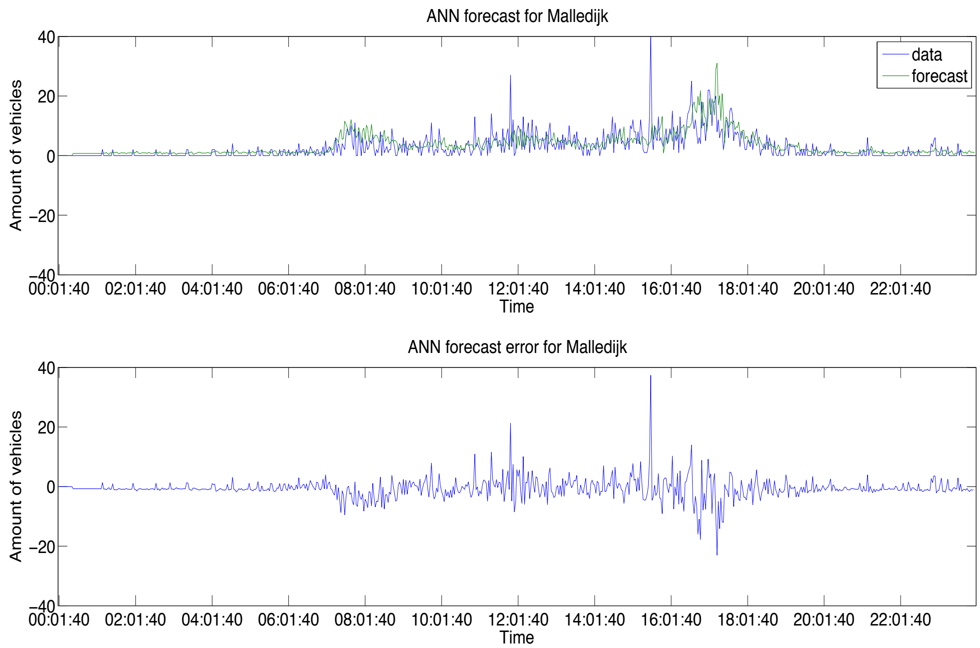


Figure 29 Artificial Neural network forecast for the Malledijk

Similar to Holt Winters the ANN follows the main profile quite nicely although it seems to overestimate the validation set a bit more. The literature study mentioned that an ANN is capable of predicting or at least reacting to volatile disturbances. In this model it is clearly not the case. for the ANN to recognize similar situations it requires a lot more training data or additional exogenous input of other congestion determinants.

EVALUATION

With equivalent history for both Holt Winters and the Artificial Neural Network the Holt Winters algorithm outperformed the neural network, respectively RMSE 3.49 and RMSE 4.0. Note should be taken that for both methods more history can be added.

Holt Winters can be altered to include more cycles. Taylor (2010) has extended the conventional HW method to deal with double and triple seasonal cycles. His extensions simply involves an additional smoothing equation and smoothing constant for each extra cycle. Taylor tested these methods on half-hourly electricity demand data and found that the triple-cycle method was more accurate than versions of Holt-Winters with fewer cycles.

Artificial Neural Networks can also be improved if more history is taken into account. The addition of more history is as straight forward as adding more exogenous inputs. More problematic is the initialization and training of the network. The fact that the network gets stuck in different local minima for each training sessions whereas Holt Winters results in the same forecast if the same input is supplied leads to the conclusion that Holt Winters is more reliable. In an online environment this is desired.

The elegant, robust and comprehensible workings of the Holt Winters algorithm also favour online use.

In respect to disturbance forecasting ANN has the benefit over Holt Winters because the exogenous input does not necessarily have to be historic data. The faraway detector currently used can supply additional data like

occupancy, density, (speed), or its derivatives. These variables are also indicators of congestion and can be added to the equation. The forecast of disturbances for downstream detection is an interesting idea for a new study.

Both Holt Winters and Artificial Neural networks will be tested in the simulation environment even though Holt Winters performed better. This is because both algorithms were coded such that they can be used online and offline; the online testing therefore requires little extra work.

PLATOON FORECASTING

The literature study revealed the arrival of platoons at downstream signals. Platoons are formed due to upstream compacting of vehicles at the traffic light and letting them through in one group. Forming platoons at the edges of a network can be very smart because if the network is configured properly platoons could ideally travell passed intersection unhindered while maintaining the group formation; the so-called geen waves. Gaps in between between platoons can be used to switch signals and allow passage of conflicting streams.

This is a very ideal situation because in reality the group formation will diminish over time and the utopic signal plans can not be realized. The literature study showed that individual behaviour will affect the observed group's behaviour. The group behaviour can be described by platoon characteristics like Inter platoon headway or platoon speed.

To my knowledge, no model was found in the literature that tries to forecast platoons for a downstream detection point. Moreover, no use was found of the described platoon characterstics to forecast platoon arrivals either based on upstream or downstream information. This resulted in the idea to use platoon chararacteristics to build a model for platoon forecasting.

An early decision was made to develop the model based on the upstream signal of the Groene Kruisweg eastbound (SG: 7,8,9). To be precise the camera positioned 2090m from the stopline was chosen. This camera is positioned just downstream of the upstream intersection. The decision was made for multiple reasons:

1. The upstream signal will show a cleaner signal less influenced by dispersion. Identifying individual platoons should therefore be easier.
2. The arrival patterns at the downstream signal could be distorted by accumulating vehicles or queues. If the back of a queue has passed the detector or is near the detector the vehicles will either be standing still or decelerating earlier. In either case the arrival pattern will be distorted making analysis and forecasting more difficult.
3. Building the signal on the downstream signal could result in inability to explain certain observations because the signal is influenced by multiple factors. Relating them to upstream observations where platoons have formed in their earliest stage should allow for a better analysis of the observed phenomena.

The model will be built on the upstream signal to extract the right signals. Secondly a forecast will be made based on the upstream information to validate its forecasting potential. The same model will then be used to extract the downstream signal. The downstream signal will be compared to the upstream signal. If the model works well on the upstream signal it could also work on the downstream signal. If the model can not be configured to work on the upstream signal then it will surely not work on the downstream signal. Figure 30 shows an example of platoons leaving the upstream detection point.

Figure 30 Upstream signal with passing platoons

MODEL DEVELOPMENT

Instead of trying to forecast all the individual vehicles on a second by second basis an approach was taken to forecast certain platoon characteristics. More specifically:

- Inter platoon headway (in seconds)
- Platoon size (in seconds)
- Platoon volume (in amount of vehicles)

To achieve this a few algorithms were devised to extract the platoon characteristics from the original signal.

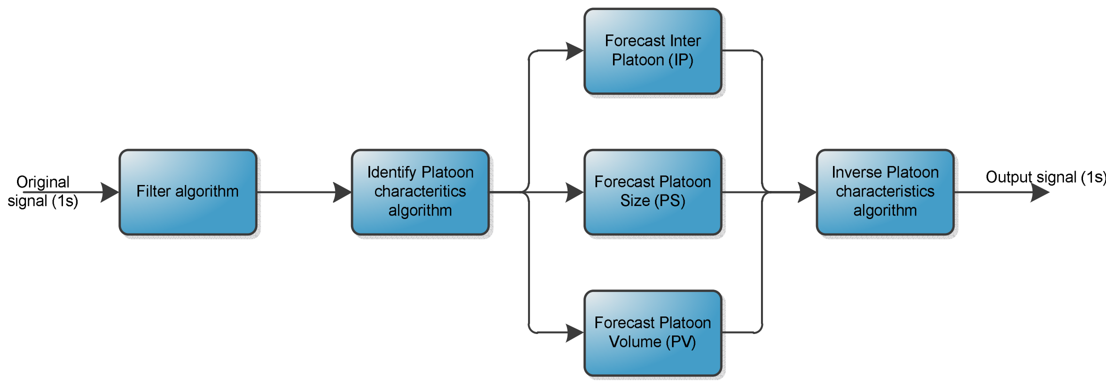


Figure 31 Platoon characteristics forecasting process

The 'identify platoon characteristics' algorithm will be discussed first to have a better understanding of the three signals and how they are determined. With a better understanding of the three separate signals the need for a pre-processing filter will also become apparent. The filter can 'de-noise' the original signal so the three separate signal can be better extracted. The next section will discuss the inverse algorithm which converts the separate signals back to an output signal on a second basis. This output signal is a forecast of 120 seconds in the future. The code for the algorithms can be found in Appendix C.

The following sections will elaborate on the forecast for the separate signals. The forecasts are made using Artificial Neural Networks. The forecasts consists of four stages: initialization, training, forecast and evaluation. Each stage will be discussed in respective order.

IDENTIFY PLATOON CHARACTERISTICS ALGORITHM

These algorithms are all based on one critical component: the critical headway. The critical headway acts as threshold to classify the vehicles among platoons. The critical headway is the key to pinpointing the start of a platoon. This is done by finding the successive vehicles which have a greater headway than the critical headway.

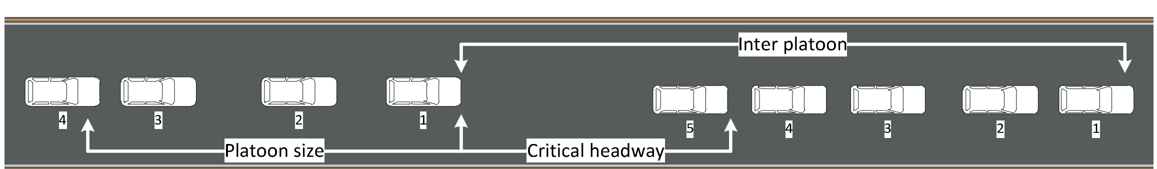


Figure 32 Platoon characteristics and critical headway

$$t_{start} \leftarrow t_{headway \geq critcal\ headway} \tag{15}$$

Consequently the end positions are determined by:

Figure 33 Key positions

$$t_{end,plat_1} = t_{start,plat_2} + headway_{veh_1,plat_2-veh_5,plat_1} \tag{16}$$

By knowing these positions the *Inter platoon (IP)* headway is determined by taking the difference of all starting points:

$$\boldsymbol{IP} = diff(\boldsymbol{t_{start}}) \tag{17}$$

Additionally *Platoon size (PS)* is determined by measuring the time between the start and end locations:

$$\boldsymbol{PS} = \boldsymbol{t_{end}} - \boldsymbol{t_{start}} \tag{18}$$

Lastly an algorithm was devised that counts all cars between a start and end point resulting in the *Platoon volume (PV).*

$$\boldsymbol{PV} = \sum_{\boldsymbol{t_{start}}}^{\boldsymbol{t_{end}}} veh_i \tag{19}$$

FILTER ALGORITHM

To clearly identify these key-points the original signal first needs to be de-noised. De-noising means removing single arrivals from the original signal that are not taking part in platoons. This is desired because single vehicles that appear in between platoons will otherwise also be considered as 'single' vehicle platoons and distort the extracted signals. For example, if the Platoon volume signal generally records platoons of 15 vehicles and suddenly there is a platoon of 1 vehicle it will cause a huge disruption in the signal. These disruptions are unwanted at a later stage because they make forecasting more difficult.

'Platoons' of two vehicles are considered equal to single vehicle platoons and are also removed from the signal. A good example of this is shown in Figure 34 around 13:20:39; the two vehicles are close enough together to form a 'platoon'. But closer inspection shows that they are separate arrivals and both are not taking part in any platoon.

A filter algorithm was devised that pinpoints single vehicles based on the critical headway. A single vehicle is identified if the critical headway is exceeded before and after the arrival. In case of two vehicle groups, given

the above criteria, both vehicles would not be removed because the headway among these two vehicles is below the critical headway. In other words, considering one of the two vehicles the critical headway is only exceeded on one side and should therefore not be removed. This is resolved by checking if the critical headway is exceeded on both sides of the two-vehicle platoon, regardless of the intra platoon headway. If this is the case then they are still both removed from the signal.

The choice of two- vehicles as a minimum Platoon volume is arbitrary, the filter features the possibility for changing this criteria to three, four, or any other amount of vehicles. The choice of minimum Platoon volume is difficult because choosing a value too high will eventually result in removing actual platoons. Removing actual platoons is also unwanted because it will create large gaps causing again more 'noise' in the signal. In other words, the filter is needed to reduce noise but can also induce noise. For this reason a relatively safe value of two vehicles was chosen.

As a consequence of the filter, single arrivals shall never be predicted in the output signal. This 'loss' is permissible by keeping in mind that bigger gains in delay reduction can be achieved by prioritizing the clear passage of platoons over single vehicles.

Below shows a sample of the original 'Groene Kruisweg' signal (bottom) and the filtered signal (top):

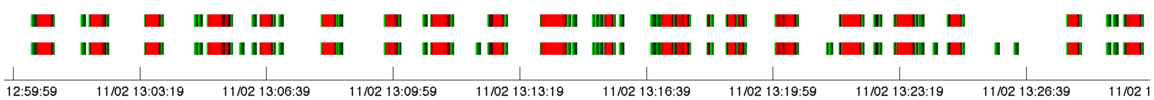


Figure 34 Original and filtered signal

INVERSE PLATOON CHARACTERISTICS ALGORTIHM

The final stage involves transforming the three separate forecasts IP, PS and PV back to an array of expected arrival times on a second by second basis. This is done by taking the t_{start} of the lastcompleted platoon and successively add the Inter platoon forecasts as new starting points. t_{start} is only known if a full platoon has passed the detector; this happens if there has been no new detection for a timespan equal to the critical headway. If the critical headway safety margin has passed a platoon is considered completed (Figure 35). If the detector is in the middle of a newly arriving platoon then t_{start} is pointed at the last completed platoon.

The start of a new platoon is registered when there are more than two successive vehicles within the critical headway, otherwise the filter removes them. Again even when the beginning of this new platoon is registered, the start point t_{start} is only assigned after full completion.

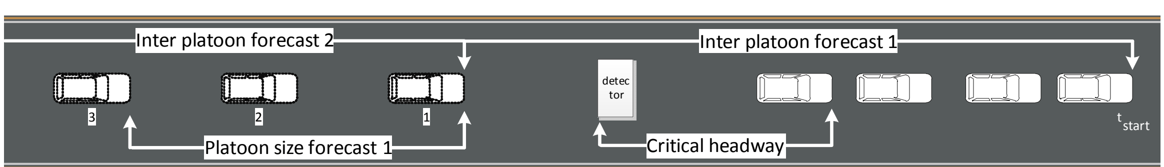


Figure 35 Inverse Platoon characteristics algorithm

The reason for assigning t_{start} to a fully completed platoon is because of the forecasting stage. The forecast algorithm makes a forecast every second. This forecast does not change for a certain period until new information becomes available for IP, PS and PV. This happens only when a platoon has fully completed. If t_{start} were reassigned without completion of the new platoon then no new information is available. Relocating t_{start} to this newly arriving platoon would propagate the old forecast belonging to the old completed platoon. The same unchanged forecast would thus be used twice at different points in time.

Therefore t_{start} is only repositioned after full completion of a platoon. Figure 36 might help visualize the problem.

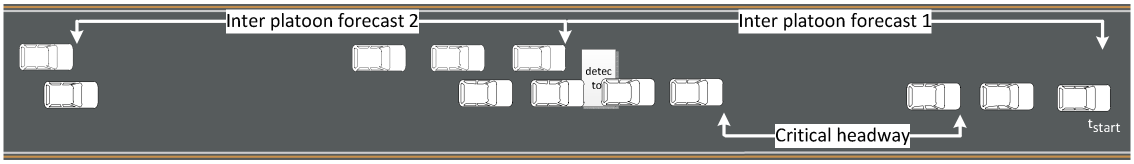


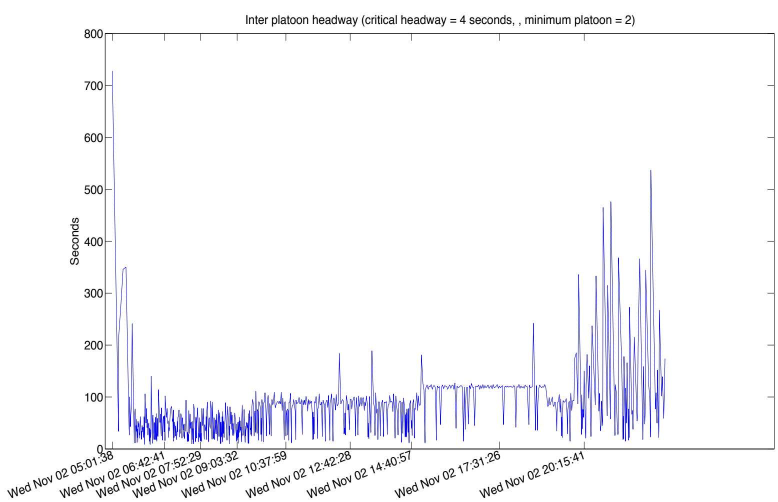
Figure 36 Position of t_{start}

The amount of vehicles forecasted in PV are *uniformly* distributed between its forecasted start and end point. A uniform distribution is untrue to the dispersed arrival patterns that are found in literature and found by investigating the data, even so a uniform distribution was chosen to simplify the problem. The reason for this is that applying any other distribution will most likely blur out the start and end positions of the forecasted platoons. This would make it harder to judge the method currently being researched because these start and end points are actually key.

PLATOON CHARACTERISTICS ANALYSIS

A critical headway was chosen of 4 seconds which is equivalent to the minimum yellow time for a 80 km/h maximum speed limit (Rijkswaterstaat, 2002). This value is small enough to identify the gaps between platoons departing from conflicting streams and hopefully large enough to make unnecessary cuts in dispersing platoons.

The following three graphs show the progression of IP, PS and PV for the Groene Kruisweg throughout a day.



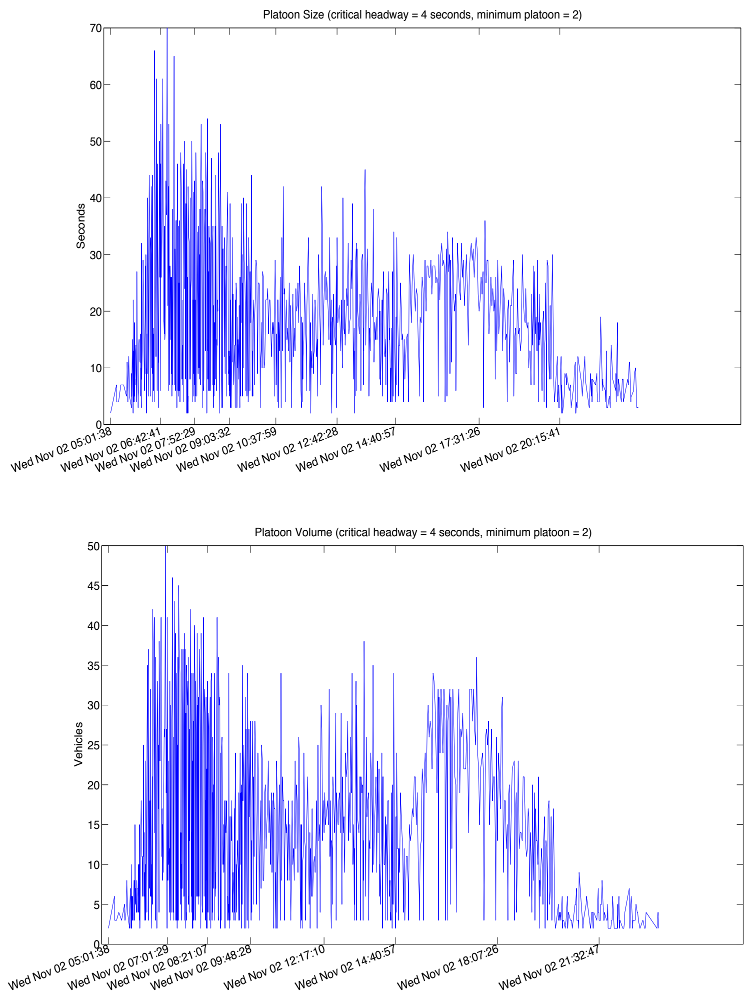


Figure 37 Platoon characteristics for critical headway = 4 and minimum Platoon volume = 2

An interesting observation can be made in respect to the IP signal. During peak hours (05:00h – 09:00h) the signal fluctuates between 30 and 60 seconds; then there is a gradual increase to 90 seconds around 10:30h and a sudden increase to 120 seconds around 15:00h. This phenomena can be explained by varying demand on the side streams (SG: 3,4,12).

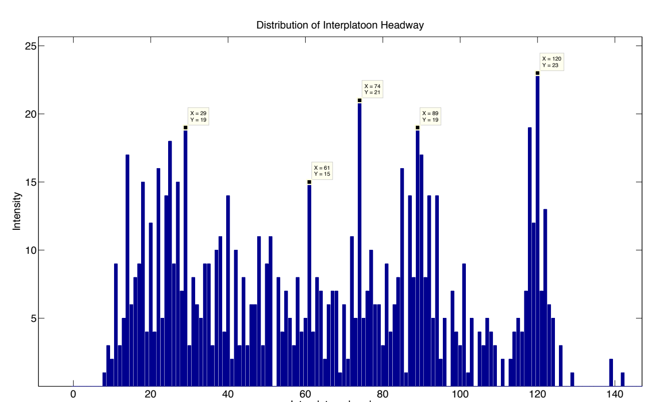


Figure 38 Distribution of Inter platoon headway

In the IP signal five dominant values can be seen re-appearing that require further explanation (Figure 38). The IP values at 30, 60, 90 and 120 seconds can be explained in respect to the signal cycle. For the moment assume that a full cycle is 120 seconds. A full cycle consists of 4 conflicting stream and 3 feeding links. On the receiving link therefore 3 types of platoons (SG: 4,8,12) should be expected. These platoons are interspaced by 1/4th of 120 seconds = 30 seconds. The dominant value of 80 seconds could be explained by premature ending of one or phases which results in a deviation from the regular 90 second phase.

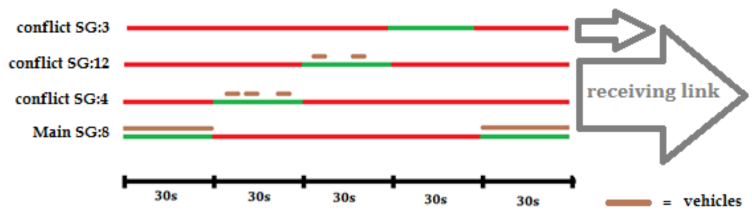


Figure 39 Explanation of observed platoon characteristics signal

In case that no vehicles depart from a conflict stream, or in the other case that the amount of vehicles is smaller than the minimum Platoon volume of 2 vehicles (and therefore removed by the filter), no platoon will be recognized leaving a gap. In case that only platoons on the main stream (SG:8) are present this gap is exactly 120 seconds. In every other case either a gap of 30, 60 or 90 seconds can be expected. The IP signal shows that the 60 second IP value appears more often than the 30 second value, even during peak hours when demand is present on all stream. This can be accredited to the fact that the SG:3 conflict stream feeds another link and therefore shows no platoons on the receiving link; so this stream always results in a gap or an IP value of at least 60 seconds.

For low demand periods during the night (20:00-05:00h); Inter platoon headways are large and unpredictable because platoons consist of a single vehicles (see Figure 37: Platoon volume).

During the peak hours (05:00h - 08:00h) the IP signal is centred around 30 seconds or 60 seconds. This means that demand is present on every side stream (SG: 3,4,12). Taking a closer look at Platoon size and Platoon volume for peak hours it can be seen that the differences in demand on the conflicting stream are quite big (from 3 to 40 vehicles). Because of the erratic signal the peak hours are not suited for platoon forecasting. This traffic state should be forecasted with aggregated demand forecasting.

The interesting period begins after the peak hours (08:00h – 20:00h) where seemingly a stationary time series emerges. At 15:00h something noticeable happens in all three signals. Especially IP shows a clearly visible change: a shift upward and the variance of the signal decreases in respect to the window before 15:00h. Demand on all the conflicting streams (SG:3,4,12) in respect to the main stream (SG:8) is so low that vehicles departing from the side roads upstream are never departing in groups large enough to be recognized by the filter. The filter therefore removes the individual vehicles when they fall in between two platoons of the on-going stream (SG:8). The signal after 15:00h then only resembles the on-going traffic (SG:8). The occasional outlier is a slip through of a group of vehicles on the side stream that exceeds three vehicles. The filter should be further tuned to remove all conflict streams in this time window, the main stream should then be easy to predict.

Before 15:00h the more erratic signal can then be explained by a relatively high demand on one or two side streams causing more groups of vehicles to pass through the filter which explains the bigger variations in the Inter platoon signal. The average 90 second Inter platoon headway still resembles the main on- going stream (SG:8), a 30 second IP value means that a conflict platoon is let go the succeeding phase. This time window requires further study.

Not plotted in Figure 37 but was discovered by looking at platoon characteristics of another day it was found that the signals are very similar to the signal shown in Figure 37. The same traffic states can be identified for the same time windows. This recurrence is interesting because it can be used for forecasting with a seasonality component similar to the use of aggregated demand forecasting.

Based on these findings four traffic states can be defined and each region requires its own solution:

Table 4 Traffic state and required solution

Time	Traffic state	Solution
20:00h - 05:00h	The signal shows random arrivals of individual vehicles	No forecast, placement of faraway detector should provide sufficient anticipation time
05:00h – 10:00h	The signal shows significant demand on all streams but with highly varying demand for the conflicting stream	Forecast with Holt Winters aggregated demand
10:00h – 15:00h	The signal shows one dominant stream the SG:8 and one or two side streams with low demand	Will be studied further
15:00h - 20:00h	The signal shows one dominant stream with an occasional side stream platoon	Filter out all remaining conflicting platoons and forecast main stream

The focus shifts to the time window of 10:00 – 15:00h because it shows that multiple conflicting streams are present but the differences in volume are not as big as during peak hours. Figure 40 shows what happens if the filter is tuned to a threshold minimum Platoon volume of 2 vehicles to 3 and 4 vehicles (increasing to 5 vehicles shows no additional change). The uncovering of the main stream can be explained by the higher demand and relatively larger platoons in respect to the side streams with lower demand. If demand on the main stream is dominant it is expected that side stream 'platoons' will be removed by the filter leaving only the main stream signal (SG:8):

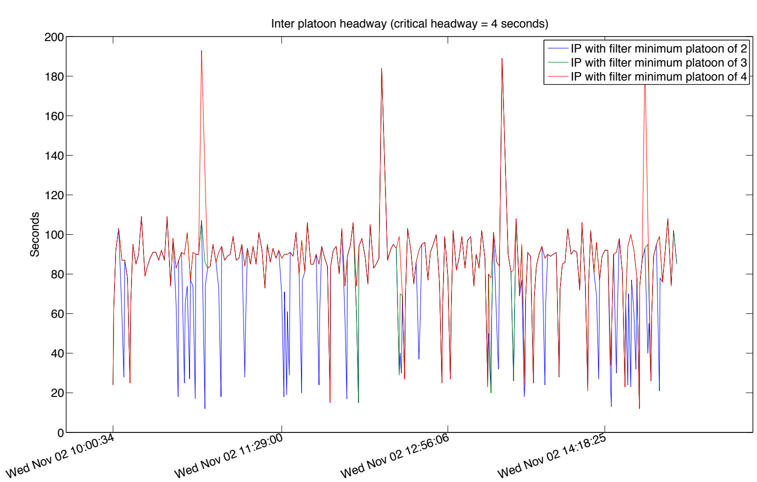


Figure 40 Uncovering the main stream by up-tuning the filter

A similar uncovering happens for the PS and PV signal if the filter is tuned to a higher minimum Platoon volume. What can be seen is that indeed more groups of vehicles are removed from the side stream leaving a signal that represents the main stream in presence of side stream platoons. The blue outliers indicate side stream platoons at 30 and 60 second intervals, in the red signal where the minimum Platoon volume is set to 4 vehicles, the filter has removed most of the blue outliers. On the other hand, the red signal has also introduced some new peaks in the region of 180 seconds: this means that the filter has removed small platoons from the main stream, i.e. the filter introduced noise.

An autocorrelation graph (ACF) for the period, 10:00 to 15:00h gives even more insight in the predictability of a time series. An ACF indicates the correlation with points in the past (lag) e.g. every point on a sine wave is highly correlated with $n \cdot 2\pi$ full periods back and inversely correlated with $n \cdot \pi$ periods back. A high correlation coefficient means that extrapolation based on these past points (forecast) is possible. The 95% confidence borders are indicated by the horizontal lines. The ACF was constructed using the Wiener–Khinchin theorem. Wiener–Khinchin theorem makes use of a Fourier Transformation to find high intensity frequencies. An ACF is plotted for a minimum platoon volume of 2 vehicles (left) and 4 vehicles (right).

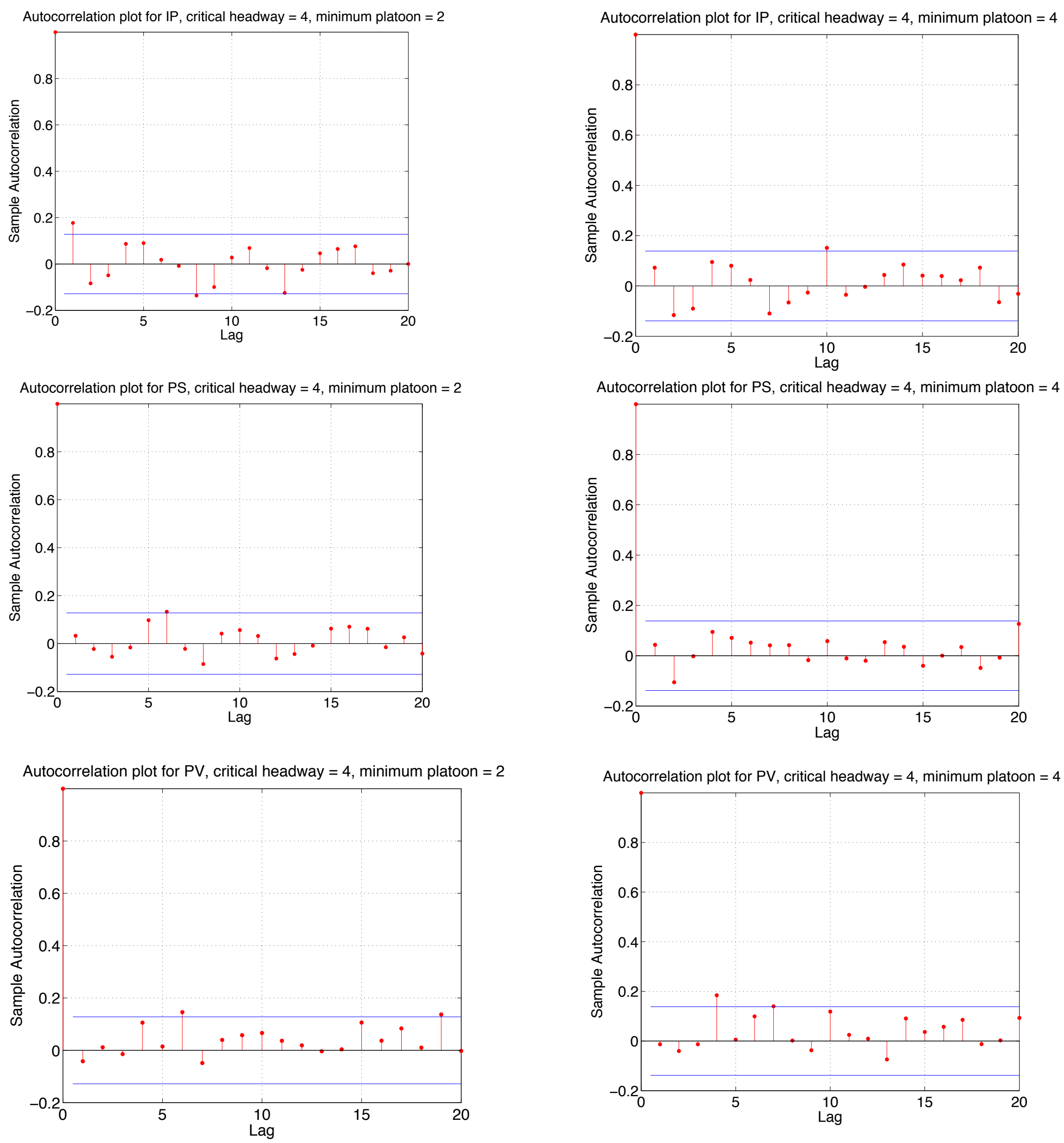


Figure 41 ACF plots for platoon characteristics with min. Platoon size = 2 (left column), and min. Platoon size =4 (right column)

Most interesting would be to see if the Wiener–Khinchin theorem can find a periodicity in the signal that has most of the side stream vehicles remaining, so with a minimum Platoon volume of 2 vehicles. In the signal with minimum Platoon volume = 2 one would expect to find that every fourth platoon shows resemblance with the first (so lag 3, 6 and 9) in the second signal one would expect to find no correlation because the signal represents only one stream and all platoons are therefore 'alike' in terms of platoon characteristics.

Interesting to see that the IP and PS signal show an almost significant sine wave for the min. platoon = 2. A similar sine wave profile can actually be seen if an ACF of a Sine wave is tested for auto correlation. So the ACF hints that there is little periodicity. However, closer inspection shows that the almost significant values are at lag 4, 5, and 8. This is not in line with expectations (3, 6, 9). This I cannot explain at this time.

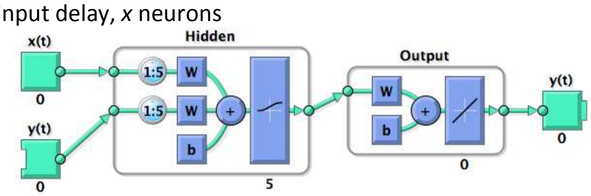
ANN FORECASTING PROCESS

Two neural network types were evaluated for this problem. Because of the recurrence found in the signal in two consecutive days one could argue that a NARx network can be used similar to the use in aggregated demand forecasting.

However, there is a difference. Demand on two separate days at the same point in time are both related because they are driven by the same, yet 'incomprehensible', mechanism of people having similar origins and destinations. They are both dependent on the same OD matrix for that system which is more or less static for that point in time. A platoon characteristic, like 'Inter platoon' at a point in time today is independent of what happened with a platoon yesterday at the same time. In that sense, it seems wrong to use yesterday's information to do forecasts for today. The only thing that justifies use of the past data is the average level.

Table 5 Platoon characteristics process

	Platoon Characteristics Forecast NARx process

Initialization	• Prepare dataset 'Wednesday' and 'Thursday'
	    o Filter signal (de-noise)
	    o Extract platoon characteristics (IP, PS, PV)
	• Construct multiple feed forward network's for IP, PS and PV each with 2 inputs (one for Wednesday's input, one as delay line for Thursday's past recordings) and one output
	• *n* steps for input delay, *x* neurons
	
	Figure 42 NARx network with n = 5, x = 5

Training	• Train the weights on Thursdayusing seasonality of Wednesday for IP, PS and PV with Levenberg- Marquardt back propagation

Forecast	• For begin to end Friday do every second
	    o Add new detections to database
	    o Extract IP, PS, and PV and $t_{start}$of last completed platoon
	    o If IP, PS, and PV > input delay then
	        ▪ For IP, PS, and PV
	            • Close the loop
	            • Do 3 step ahead forecast using yesterday and today's input
	    o Transform IP, PS and PV to an arrival forecast on a second basis
	    o Remove all forecasts smaller than $t_{now}$and greater than the 120s horizon

Evaluation	• Calculate Root Mean Square Error of IP, PS, PV for Friday • Calculate correctly predicted vehicles • Calculate falsely predicted vehicles • Calculate non-predicted vehicles

Multiple off line simulations were run for different settings for number of input delays and number of neurons. The best setting was found for input delay of 5 time steps and 5 neurons. The improvements, however, are not of an order that require further attention. This will become clear during the next section.

TRAINING RESULTS

The simulation and training example will focus on the time window of 10:00 to 15:00, first the signal that shows side stream arrivals (minimum platoon = 2) will be trained and second the trimmed down signal (minimum platoon = 4) will be trained that represents only the main stream (SG:8) in presence of conflicting streams (SG:3, 4, 12). Figure 43 shows the results for the NARx network.

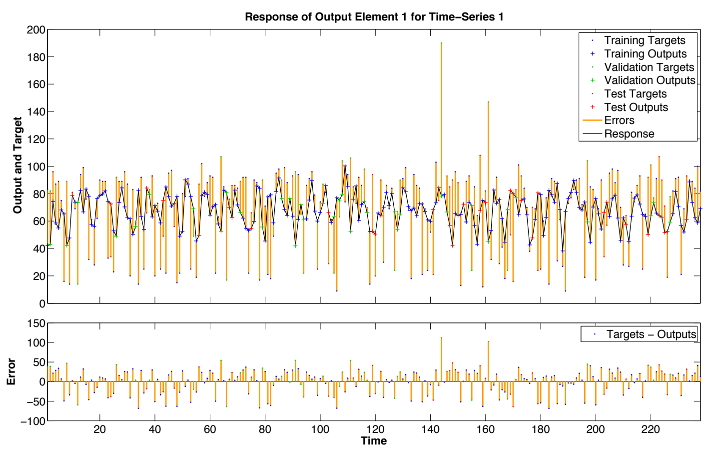


Figure 43 Regression of Inter platoon headway training with critical headway = 4, minimum Platoon size = 2

The dotted points at the end of the yellow lines indicate the real data set, the fitted function of the neural network is indicated by the black line and the crosses '+'. The difference is indicated by the yellow line and gives the error.

The most important observation is the impact of outliers. The outliers in the real dataset are there for two reasons: side stream 'platoons' and filter failure. To elaborate; the peaks around 30 and 60 seconds represent the side stream platoons. It can be seen in the above fit that the ANN is not capable of fitting the side stream platoons. The outlier in the region of 180 is the result of filter failure i.e. a mainstream platoon is removed. At this point it can be concluded that trying to forecast side stream and the main stream in one signal will not be successful. Therefore the scenario was tested where the side stream platoons are removed and only the main stream in presence of the side streams is visible. This resulted in the following training for the platoon characteristics. The critical headway is still set to 4 seconds, the minimum Platoon size is increased to 4 vehicles:

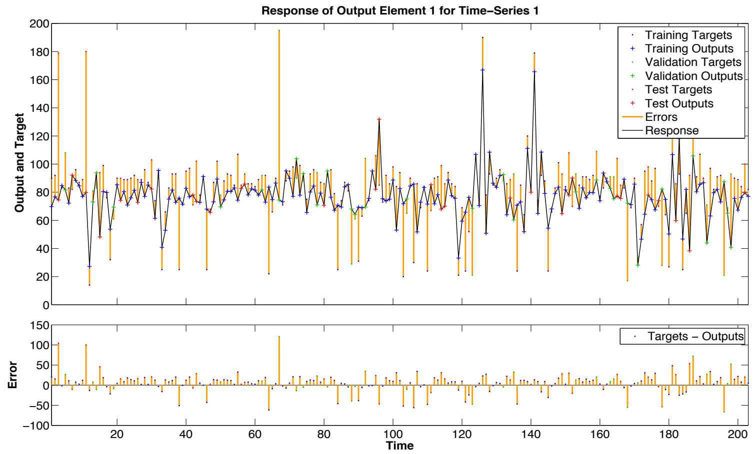


Figure 44 Regression of Inter platoon headway training with critical headway = 4, minimum Platoon size = 4

In this case it is desired to remove all platoons except the main stream. If side streams platoons slip through the filter this can be also be viewed as filter failure. It can be seen that still some values around 30 and 60 seconds appear. Also the filter introduced some extra peaks around 180 seconds. These outlier make a fit more difficult. Everything that happens with the signal that stays within a bandwidth of 70 to 110 seconds Inter platoon headway can be accredited to dispersive behaviour of platoon departures, this is random behaviour.

Figure 45 shows the training regression for Platoon size (PS).

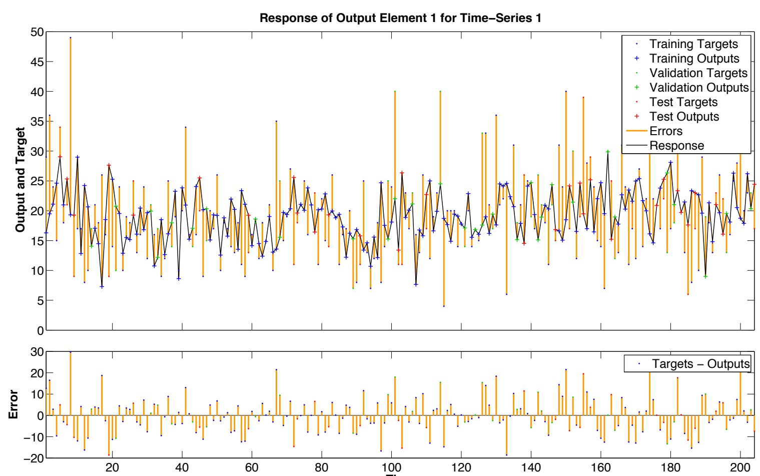


Figure 45 Regression of Platoon size training with critical headway = 4, minimum Platoon size = 4

Two things affect the time series of PS: dispersion and demand together determine the size and spread of vehicles and therefore the PS signal.

The differing demand at the (conflicting) signal group(s) will cause differences in volumes of successive platoons and consequently the outliers. Different volumes are correlated with greater Platoon sizes because more vehicles require more space.

The second reason for a difficult fit is due to the phenomena of dispersion. At the upstream intersection dispersion will still be minimal but the little observed dispersion can be accredited to differences in vehicle accelerations. In addition, the literature study of Bonneson et al. (2010) also showed the different upstream

profiles for vehicles going through turns (Figure 18). Turns are reduced speeds areas that disperse vehicles even further. There are no specific drivers that constrain the amount of dispersion. The Platoon size therefore also has a large stochastic part in the time series, this makes a good fit more difficult.

In future research a dispersion model can be included to further improve the fit. Another interesting idea for the future is to subtract Platoon volume from Platoon size, this should remove the 'constant' part of Platoon size and leave a signal which only depicts dispersion.

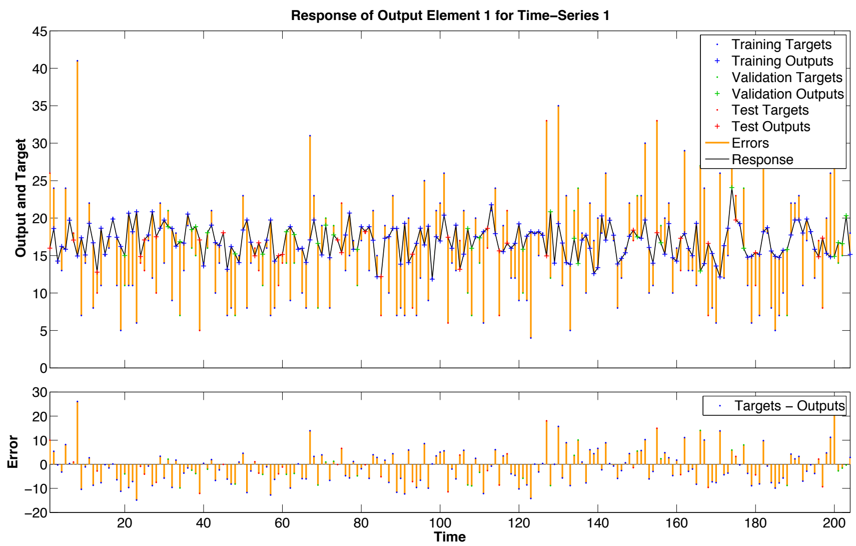


Figure 46 Regression of Platoon volume training with critical headway = 4, minimum Platoon size = 4

For Platoon volume one would expect to see a similar profile as the aggregated demand (Figure 50). In other words, if the average amount of vehicles grows over time than this should result in higher volume platoons. The large outliers ≈ 5 and > 30 can be accredited to the sub-optimal filter. Due to the filter platoons of less than 5 vehicles cannot exist; therefore the lowest visible value is 5 vehicles around for example t = 124. The outliers in the high region > 35, a twofold of the average 17 vehicles, suggest that at times two platoons are merged together to form one platoon. These outliers, high and low, should also match with outliers in PS because high and low volumes are correlated with the length of the platoon. Furthermore the outliers that give small platoons should also match with the outliers for a moment in time with the conflicting streams that managed to get through the filter. The values that range between 20 and 10 vehicles are varying demand on the main stream. These relatively great differences in demand for just the isolated main stream are difficult to fit by the ANN.

For future research it is advised to make use of this matching principle to create a second stage filter for removing outliers in IP, PS, and PV. Removing the outliers will create a more clean signal and lets the ANN focus on the main stream signal (SG:8). At this point the Artificial Neural Network training gives reasonable expectations for IP. The other two characteristics are trained but can only follow the general mean of the series. One could argue that the neural network acts as a moving average for PS and PV.

OFFLINE SIMULATION EXAMPLE

An offline simulation was done for the dataset from 13:00 to 14:00. The first 13 minutes no forecast is made because the algorithm has to gather enough input data to start forecasting. Reruns of the simulation will result in different forecasts each time due to random initialization of the network. Below one such example will be discussed; reruns can obviously perform better or worse than the discussed example. Some important remarks can be made based on this single simulation example nonetheless.

For evaluating the forecasting techniques three criteria were defined. The criteria differentiate the types of errors that can be made with a forecast. This follows from the logical progression that it is more frowned upon to give a green to non-existing cars than to give red to non-predicted cars. The following schematic clarifies the need for different criteria:

Forecasted ? / Detected?	Yes	No
Yes	Good	Annoying
No	Bad	Good

Table 6 Evaluation criteria

The evaluation algorithm also allows for slight 'wrong' predictions. Meaning that a prediction is also considered good if it matches a detection within a certain bandwidth, in the example below a forecast is also considered correct if it falls within two seconds of validation.

Figure 47 gives a forecasting example of the aforementioned Groene Kruisweg sample:

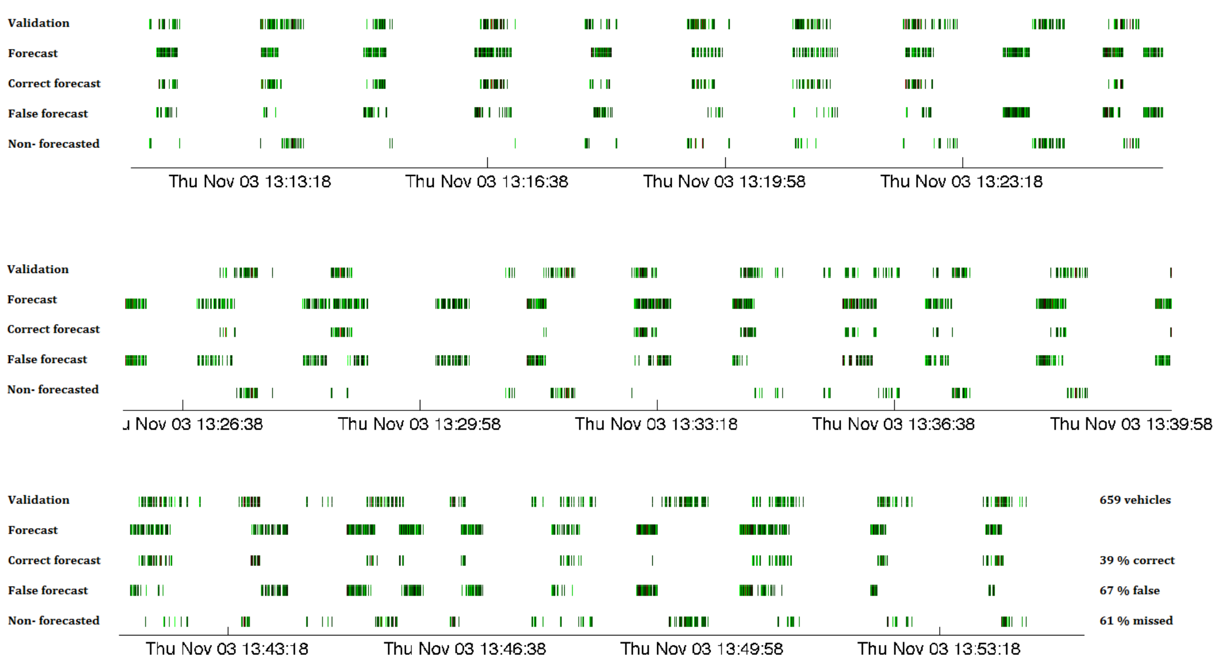


Figure 47 Platoon characteristic forecast example

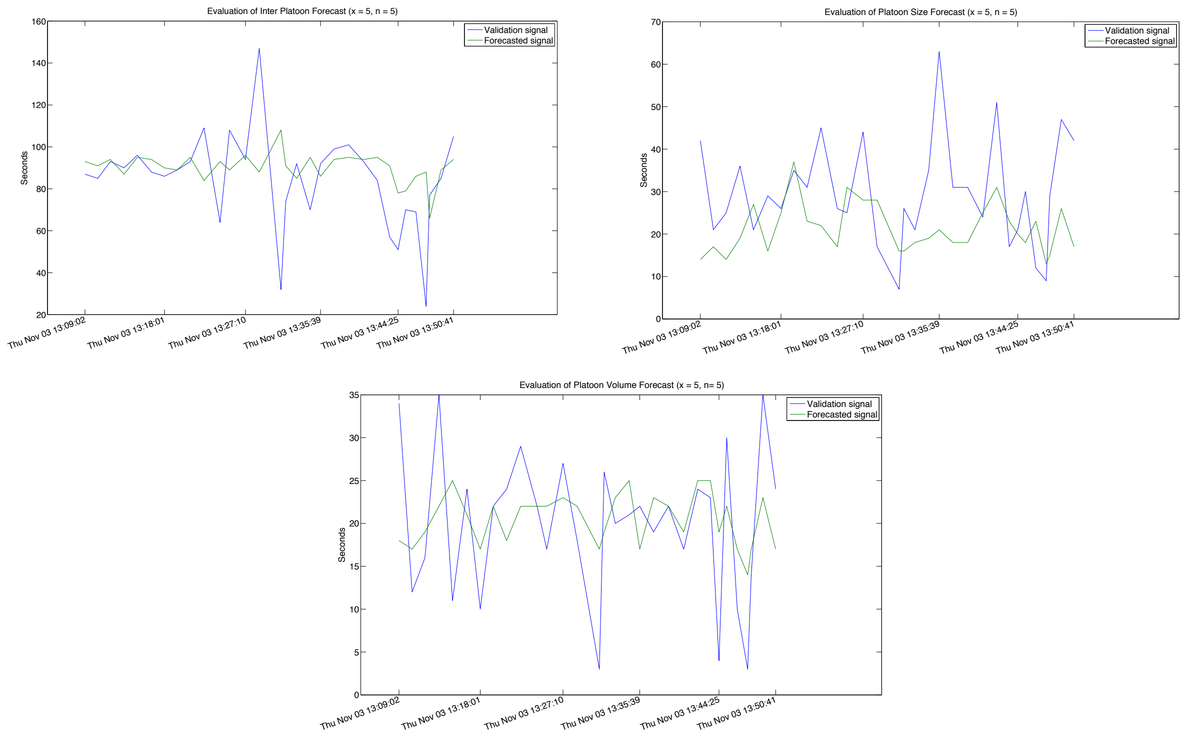


Figure 48 Evaluation of Platoon characteristics for Groene Kruisweg eastbound forecast

The first 15 minutes the algorithm performs well when it forecasts platoons from the main stream (SG:8, all around 90 seconds). IP forecasts are almost spot on. Around 24 minutes the validation signal suddenly deviates from the mean i.e. the main stream (SG:8) is skipped (nothing appears at 90 seconds) and instead one of side streams is let go with an offset (unfinished) phase of 20 seconds (the validation IP signal is therefore 90 seconds + 20 seconds ≈ 110 seconds). In the overall plot it can be seen that a main stream platoon was forecasted at this point and not one of the side stream platoons, therefore the forecast and the validation signal do not match and both contribute to the total error. A second observation that can be made here is the premature ending of a phase which is suggested by the phase that ended after 20 seconds. This adds another variable the expected arrival times making the forecast even more difficult.

Then in the validation signal the main stream is released again at a full double cycle of 180 seconds in respect to the last platoon from the main stream: 110 seconds + 60 seconds ≈ 180 seconds. The forecast signal which remained close to the mean and therefore completely first missed the platoon from the side stream then again captures the platoon from the main stream. After 25 minutes the real platoon arrivals start to show erratic behaviour and quickly results in many false forecasts. The distinction between main stream and side stream platoons can no longer be made. Around 45 minutes forecast and validation are better matched.

In retrospect a mistake was made in the forecast algorithm. The decision was made to take the starting point of every detected and completed platoon to propagate the forecasts into the future. In the case that only the main stream is dominantly present (after 15:00h) it is allowed to remove all other outliers. Before 15:00h apparently two stream are dominantly present resulting in dominant Inter platoon times of 90 seconds and 30 seconds; together they make a full cycle of 120 seconds.

The earlier removal of the side stream to focus on the main stream signal in presence of the side stream is not allowed. Because when a platoon from a side stream actually appears in the signal with an offset of 30 or 60

seconds from the 90 second Inter platoon time. Then the forecast for a mainstream platoon is propagated based on the starting point of a side stream platoon. This will result in both the side stream not predicted and consequently the main stream not predicted. In other words, forecasts for main stream platoons should only be based on starting points of main stream platoons and only with a full 120 second cycle.

Platoon size and Platoon volume go hand in hand. If PS is small and PV high than the density is high. Even when the forecasted platoon matches a real platoon for a large part many forecasted vehicles will still be counted as false because they can only match once. This can be seen in the very first platoon of the sequence; the platoons are matching in position and size so one would expect no false predictions, but directly underneath many false predictions still appear because of the high density in the forecast (this is also indicated by the darker colour). The question remains how problematic this would be for the controller. More vehicles would add more weight due to greater expected delay. An increased weight does not have to be a problem. It just is more likely to switch signal, which can even be positive because if the timing was right all the cars can pass. Needless to say is the opposite effect, if no real vehicles are present and the forecast has increased weight due to highly forecasted volume then the controller will give green to a group of ghost vehicles.

Looking at all three forecasting signals (Figure 48) one could argue that each basically forecasts the mean of the main stream signal (SG:8) with a little variance. In general, an overall forecast is good enough for 50% correctly forecasted vehicles but around 70% to 80% falsely predicted vehicles. The false forecasts are considered *bad* according to table 3. Therefore the 70% to 80% failure rate weighs heavily and disappoints.

DOWNSTREAM ANALYSIS

For platoon forecasting it is important to have a detector at a sufficient distance from the stop line. Too close to the stop line will distort the arrival patterns because of queues. If the back of the queue is very close to the detector or worse, passed the detector then the detector won't be able to fully record the platoons. On the other hand, placing the detector too far before the split of the turning directions will only record a general profile which does not include the branching off cars. In that case signal group 8 (SG:11 in Figure 49) would always be informed of too high demand. The distances from the stop line for the detector are as follows, figure 48:

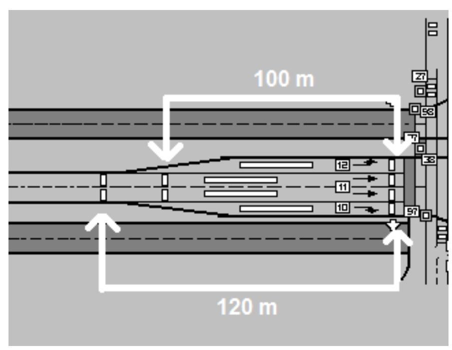


Figure 49 Detector location at the Groene Kruisweg

The data set was checked for expected queue length with aggregated demand for 120 second intervals. It was deducted earlier that the window past 15:00h only depicts the on- going traffic on the SG:8. Figure 44 shows that the platoons belonging to the main stream (SG:8) depart with a 120 second time difference after 15:00h, a full signal cycle is therefore 120 seconds. The red time is therefore 3/4th of the 120 second cycle because it has to wait for three other conflicting signal groups.

At 120metres an average of 23.26 vehicles arrive every 120 seconds (0.19 veh/s) during off peak hours (Figure 50 green section), but during peak hours (Figure 50 red section) the average is 50.46 vehicles per 120 seconds (0.42 veh/s). During red time the outflow $q_{out} = 0$, assume that the initial queue $N(t-1)$ = 0, then:

$$N(t) = N(t-1) - T(q_{out} + q_{in}) = 0 - 90 \cdot 0.19 = 17.1 \ vehicles \qquad (20)$$

This is 18 vehicles during off peak hours and 38 vehicles during peak hours building up in a queue whilst waiting for red. Assuming an average vehicle length of 4m and a safety distance of 1m the 100m link can hold 20 vehicles per lane, so a maximum queue capacity of 40 vehicles for on-going traffic. A part of the 18 vehicles will queue for the turning directions. The remainder of vehicles will therefore surely be beneath the critical queue capacity of 40 vehicles. It can be safely concluded that the queue will not reach the faraway detectors at 100m. Peak hours are not considered for two reasons. Traffic becomes so dense that the distinction between platoons becomes blurry. The arrival pattern resembles more a continuous flow which is better forecasted with aggregated demand and the queues will make detection impossible.

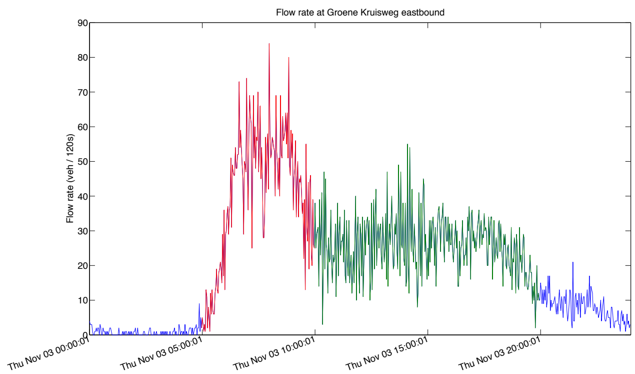


Figure 50 Aggregated demand at Groene Kruisweg eastbound

The chosen detectors are located at 100 metres from the stop line. The data for both lanes are merged together to form one dataset. The above schematic, Figure 50, is true to the real situation in that the detectors are placed just passed the widening of the road. Therefore an unknown part of the platoons will still relocate to one of the turning directions. At this point nothing can be done about that. In the control environment this could be corrected for by a distribution coefficient.

The following figures were extracted for the platoon characteristics. The platoon characteristics at the downstream location have been compensated for the travel time, or, the phase shift. The phase shift was calculated as 2090m/ (80(km/h) / 3.6) = 94 seconds.

INTER PLATOON HEADWAY

The downstream IP signal remains relatively stable although the variance increases in respect to the upstream signal. The increased variance is more clearly visible for the period after 15:00h where the upstream signal shows almost no variance. It was concluded earlier that the signal after 15:00h resembles the on-going traffic on the main stream (SG:8). The variance that is shown here is therefore purely the result of platoon dispersion on the link.

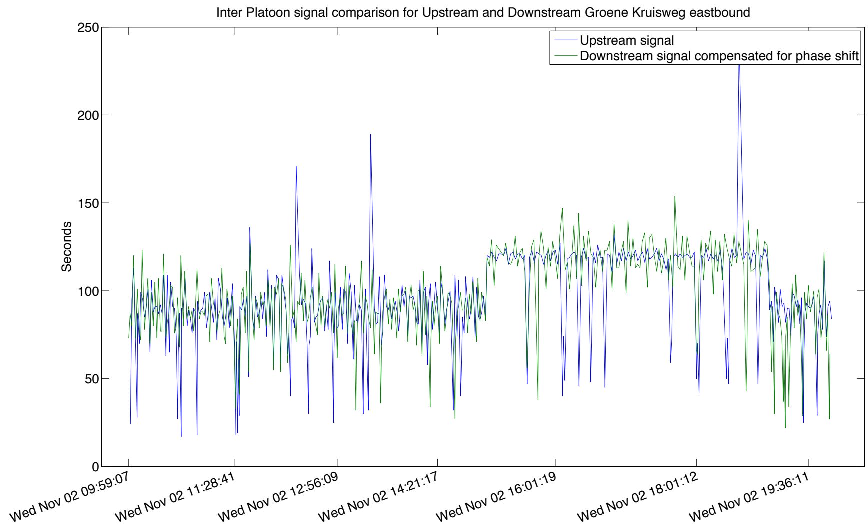


Figure 51 Inter platoon headway for upstream and downstream location

Some outliers are missing in the downstream signal and are present in the upstream signal, or vice versa. These differences can be proof of filter failure or merging of platoons. For instance the large peak around 18:45h for the upstream signal is missing in the downstream signal; this can be interpreted as a platoon that was removed from the main stream but should not have been removed. Just before the high peak a low peak is shown that resembles a side stream platoon. This peak is missing in the downstream signal which suggests that the platoon has caught up with the main platoon and merged at the back.

At 18:00h two *matching* outliers are visible in the upstream and downstream signal; this outlier can also be identified as a side stream platoon. During the offline simulation run such an outlier of a side stream platoon was also identified. This side stream platoon could not be forecasted because the algorithm was trained to forecast main stream platoons. In the case that an upstream and downstream signal are available this side stream platoon could be forecasted because it can clearly be seen that the peak in the upstream signal gives prior notion of the side stream platoon. In case that upstream detection is available the side stream platoon can be anticipated.

Downstream forecasting based on upstream information using platoon characteristics could be a promising method for further study. As explained before in the problem definition another flaw remains in the prediction model of the controller in respect to the free flow expectation of individual vehicles. Considering the signal as platoons characteristics could simplify the dynamics.

When the upstream signal is subtracted from the downstream signal the following signal remains. This signal shows the effect of dispersion on the Inter platoon headway. The high peaks can be explained by the aforementioned mismatches. The bulk of the dispersion stay below 10 seconds. It is unsure if this amount of dispersion is too much for a good forecast for a downstream detection point.

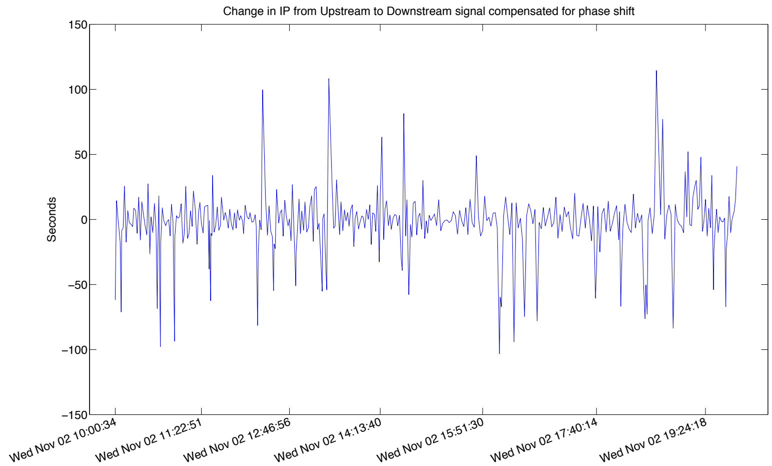


Figure 52 Change in IP from upstream to downstream signal compensated for phase shift

PLATOON SIZE

Interesting to see here is that the general profile seems to remain intact and the downstream signal is shifted upwards. The shift upward is the result of platoon dispersion. The fact that the profile remains intact is a very interesting observation because it could mean that the amount of dispersion is constant and therefore predictable. Figure 53 will elaborate on this observation; it subtracts the upstream from the downstream signal:

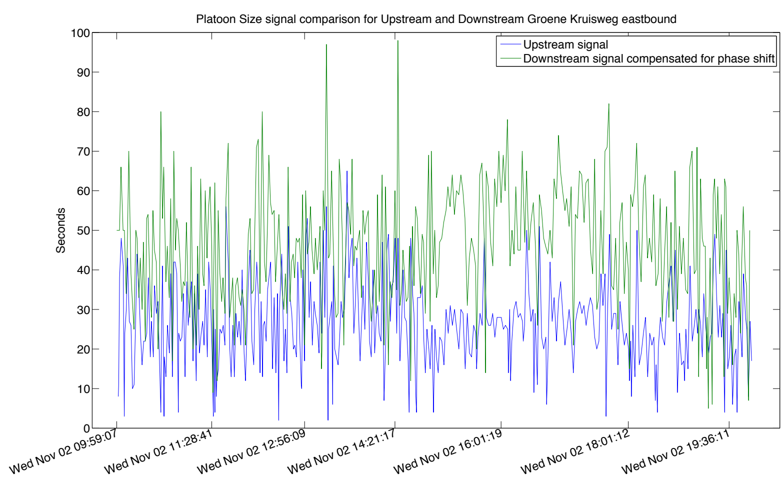


Figure 53 Platoon size for upstream and downstream location

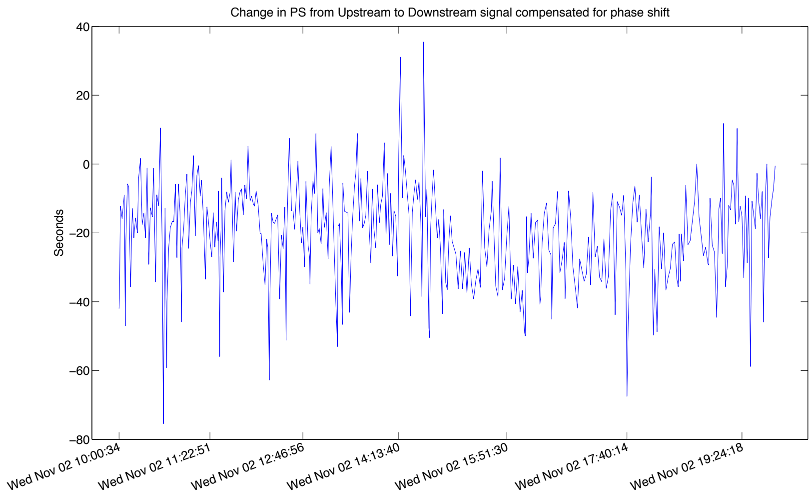


Figure 54 Change in PS from upstream to downstream signal compensated for phase shift

It can be seen that the amount of dispersion is relatively constant with an increase of approximately 20 seconds. Remember that the erratic signal in Figure 53 is still subject to changing Platoon volume and could explain the rest of the variance. If the Platoon volume signal were to be subtracted from the Platoon size signal an even clearer image of the amount of dispersion could be achieved. This is left for further study.

In respect to forecasting the time series based on a single detection point the ANN already found it difficult to train on the history of the upstream detector, the downstream signal shows a similar profile and will result in a similar bad fit.

PLATOON VOLUME

One would expect to see a downstream signal that completely matches the upstream signal because the Groene Kruisweg eastbound has no secondary roads attached so births or deaths of vehicles are not possible.

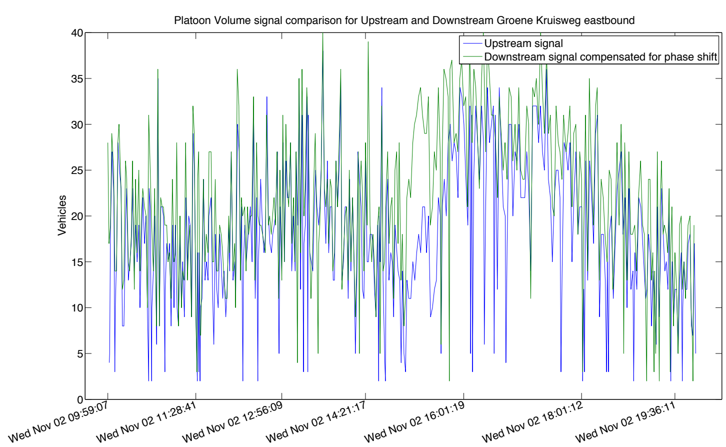


Figure 55 Platoon volume for upstream and downstream signal

Figure 56 gives a better view of the error:

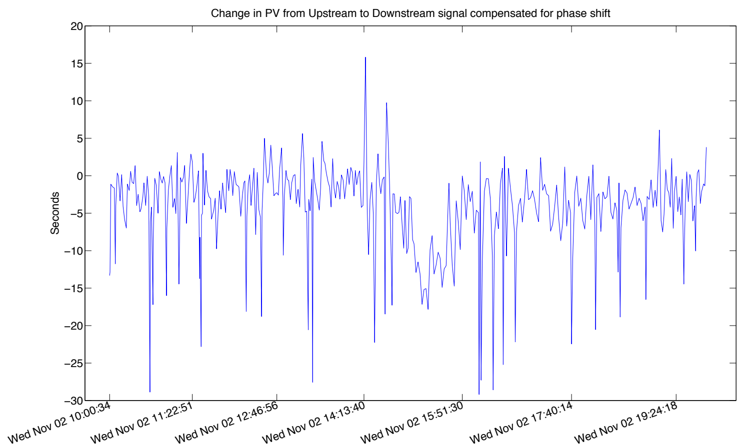


Figure 56 Change in Platoon volume for upstream and downstream signal

The differences can again be accredited to the filter and the dynamics of the traffic flow. Vehicles that were removed earlier, re-appear or have merged with other platoons resulting in higher volumes. The differences in Platoon volume are around 5 vehicles which is coherent with the general volume of a side stream platoon. If this platoon merges with a main stream platoon an increase of 5 vehicles is expected, or vice versa.

A first glance at the Platoon volume error compared to change in Platoon size supports the idea that subtracting both signal could result in an almost constant dispersion factor.

In respect to forecasting the time series based on history the ANN already found it difficult to train on the history of the upstream detector, the downstream signal shows a similar profile and will result in a similar bad fit.

EVALUATION

Overall the forecast algorithm for a single detection point performed worse than expected. The upstream signal showed an impossible fit for the signal where both the main stream and the conflicting streams are included (minimum Platoon volume of 2 vehicles). To exclude the side streams in the signal the filter was tuned to allow only platoons of 4 vehicles or greater. A signal emerges that is trimmed to show a dominant main stream.

During training of the network it became clear that a good fit is especially difficult for PS and PV, even when the signal is trimmed down to show main stream platoons. The fit for IP showed better results. The difficult fit can accredited to outliers. The outliers appear because of a sub optimal filter and varying demand on the side streams. After training a simulation was done on a new day of the real world dataset.

The simulation resulted in an average 50% correct predictions but 80% false predictions for the upstream detection point. A mistake was made with the implementation of the forecast. The decision during training was made to focus on the main stream signal and ignore the side stream by tuning the filter. The side stream vehicles are filtered out of the signal. Then when the forecasts are made in real time they are propagated, sometimes based on main stream platoon, sometimes on side stream platoons.

If main stream platoon are succeeding each other the forecast is correct because it can only forecast the main stream platoons. If a side streams platoons appears with an offset behind a main stream platoon it is not forecasted because the ANN cannot forecast the offset. As a further consequence the next main stream

platoon is also forecasted incorrectly because the forecast is propagated on the offset. This results in a cumulative error.

It can be concluded that the model failed on the upstream signal because of an implementation mistake. For the model to work the Inter platoon signal needs to be further disintegrated to show separate signals for each conflicting stream. The forecasts should then be implemented based on the origin of the platoon to form an overall forecast. At this point implementation of the model on the downstream detection point will give similar or worse results because of the increased dispersion in the signal. Online implementation in PTV VISSIM 5.2 will not result in improved performance and is hereby abandoned.

When comparing the upstream signal to the downstream signal some interesting observations were made. First of all the increased dispersion ± 10 seconds in the IP signal might be small enough for a reasonable overall forecast based on historic information. Unfortunately this cannot be tested because the model is flawed.

In addition the dispersion of the Platoon size signal suggests that dispersion is constant. For proof the Platoon size signal needs to be corrected for the changes in Platoon volume. If this is the case than platoon characteristics are a promising candidate for forecasting downstream arrivals based on upstream detection. This is left for further study. In respect to forecasting Platoon size and Platoon volume based on its history no improvement should be expected compared to the upstream signal. This is concluded based on the similar profile of Platoon size for the upstream and downstream signal.

DATA ANALYSIS EVALUATION

Two models were researched for two different arrival profiles: single arrivals and platoons. For forecasting individual arrivals an aggregated demand forecast was tested. Holt Winters showed better results, RMSE 3.49, than the Artificial Neural Networks, RMSE 4.0. Literature indicated that an extra seasonality component may be useful a full week back. It is advised to implement the extra seasonal factor in the future. The off line simulation also indicated that Holt Winters should not be used during the night with practically zero demand.

To forecast platoons a new model had to be developed. The model consists of multiple algorithms. An extraction algorithm, forecast algorithm and inverse algorithm. The extraction algorithm is preceded by a newly developed filter. The combination of the filter and the extraction algorithm made extraction of the platoon characteristics possible. It showed that these platoon characteristics are able to supply a great deal of information about the traffic state: for instance the upstream signal cycle and knowledge of the varying demand of the side stream throughout the day. The filter can also be tuned to focus on the main stream platoons.

The forecast algorithm could not be trained on a platoon characteristic signal that includes information for all conflicting streams. Therefore the decision was made to solely forecast the main stream. However a mistake was made in the inverse algorithm which constructs the platoon characteristics back to an overall forecast on a second by second basis. Therefore the model's performance was disappointing. In the future the model could be corrected for the mistake, however it is advised to first further disintegrate the Inter platoon signal to uncover the 4 signals for each of the conflicting stream. Forecasts can then be made based on the origin of the platoon. Downstream implementation was abandoned because the model did not work on the upstream signal.

The downstream signal was analysed compared to the upstream signal and indicated that the signals are showing much similarity. The dispersion effect is clearly visible and can possibly be captured in a model if further research is done on the platoon characteristics. Due to the similarities of both signal a possible implementation of a forecasting model based on a single detection, assuming the model improved and working correctly, can still be researched in the future.

CHAPTER 5 ONLINE SIMULATION RESULTS

The algorithms were coded such that they can be used online and offline. Hence the performance of both forecasting techniques found by researching the Groene Kruisweg and Malledijk should be representative for online performance. The simulation goes a step further and allows us to test the effect of the forecast algorithms when used in the adaptive control on an asymmetric intersection. The simulation environment also allows for easy modifications to a 'real time' situation to see what could be the possible effect of, for instance, re-locating the faraway detector.

The platoon forecasting model was not implemented in an online environment because the results of the platoon forecasting model proved insufficient for the real world dataset. This chapter will therefore only discuss the results of non- platoon forecasting.

NON-PLATOON FORECASTING

The output of the forecast is an aggregated number of vehicles that are expected to arrive in 120 seconds. The adaptive controller cannot handle the output as an aggregated number; therefore the 120 seconds horizon is divided by the forecasted amount of vehicles to give the average headway. The average headway is used to add expected arrivals to the adaptive controller based on the last actual detected vehicle.

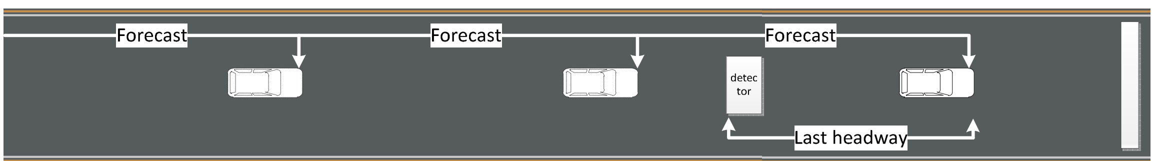


Figure 57 The aggregated forecast gives the average headway on a second by second basis

The following schematic gives an overview of the different scenarios.

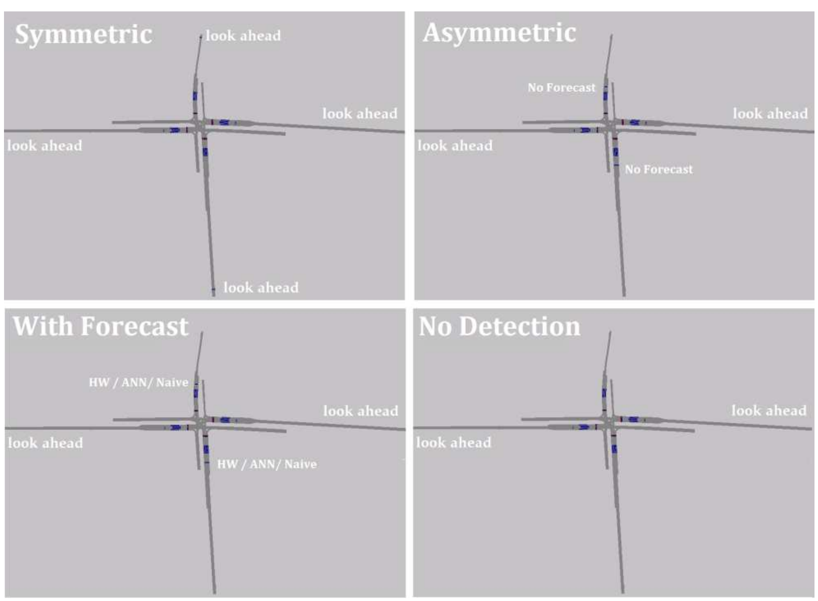


Figure 58 Different simulation scenarios

Each scenario was executed 10 times to account for the variance and simulated for peak and off – peak hours. Peak hours are generally in the morning from 06:00- 09:00 h. Demand is characterized by a steep growth and an average level that is 1.5 to 2 times the average demand during off peak hours. For off peak hours a window from 12:00 – 13:00 was chosen because the average level is representative for the average level during off peak hours. Demand around this time is stable and shows no steep gradient.

The measured performance indicator is the average delay of a car in the system.

Symmetric scenario indicates the situation where detectors are placed furthest upstream for all directions; this situation should present an upper bound where delay is minimal to which the other scenarios can be compared. The asymmetric intersection is simulated with and without forecast measures with a faraway detector at 70 meters. The three different forecast measures are Holt Winters, Artificial Neural Networks and Naïve predictions.

The Naïve predictor records the average headway at the stop line and propagates this average headway into the future similar to the Holt Winters and ANN algorithm. The Naïve predictor only takes into account the most recent vehicles to calculate the current level of demand, it therefore misses a trend and seasonality component that are present in available demand and which are present in the aforementioned forecasting techniques. The Naïve predictor acts as a benchmark for the Holt Winters and ANN algorithm.

There are two Holt Winters scenarios, one where vehicles are added to the expected arrivals array until the end of the optimization horizon (=120s) and one until a 40s horizon. In the last scenario 'No detector' the faraway detector is completely removed so that there is no more look-ahead functionality on the side streams.

PEAK HOURS

All results, except for the 'no detector scenario' are depicted in Figure 59. All scenarios give the improvement or deterioration relative to the asymmetric network with no forecast. The 'no detector' scenario is not included because the deterioration was off the chart and including it in Figure 59 would make the results unreadable.

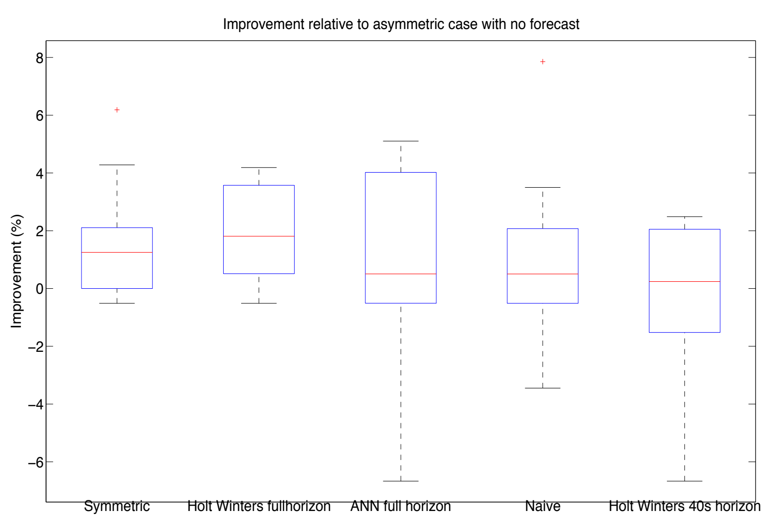


Figure 59 Simulation results for Helmond during peak hours

Table 7 Significance test for online scenarios during peak hours

Case	Symmetric	HW full	ANN full	Naive	HW 40s	No detector
Paired t-test for significance (5%)	Passed	Passed	Rejected	Rejected	Rejected	Passed

The Holt Winters forecast achieved an average RMSE of 3.37 with a variance of 0.0408 for detector 8040605 during 10 runs. The Neural network forecast achieved an average RMSE of 3.81 with a variance of 0.2296 for detector 8040605 during 10 runs. For the simulation study under similar conditions Holt Winters outperforms the neural network, in addition the lower variance indicates that the Holt Winters algorithm is much more robust. Other detectors showed similar behaviour.

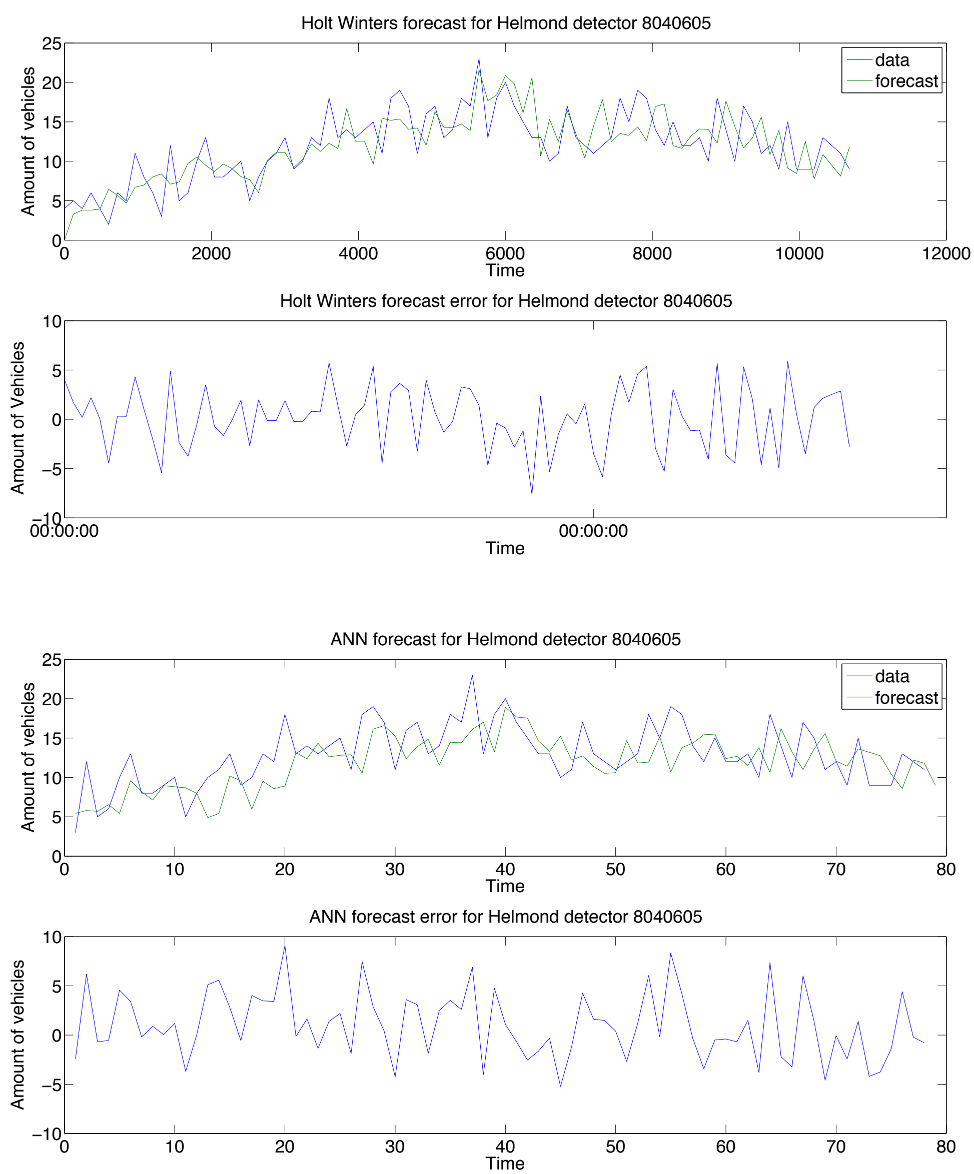


Figure 60 Examples of online VISSIM forecast for detector 8040605

What immediately draws attention is that the Holt Winters forecast seems to outperform the symmetric upstream scenario; which theoretically should be impossible. This is not entirely true because the comparison, on second thought, is not entirely fair. Namely the upstream detectors for the symmetric case are placed 400 meters away, in terms of the optimization horizon this is equivalent to 30 seconds. Changing the horizon for Holt Winters to 40 seconds confirms this by showing that the performance indeed becomes worse than the symmetric upstream scenario. Moreover, a 40 second horizon does not show an improvement over the last scenario which does no forecast, and is therefore not desirable.

The Holt Winters forecast with full 120s horizon gives a 2% improvement over the last scenario. A paired t-test with 95% confidence bounds for Holt Winters gives a significant difference and leads to the conclusion that Holt Winters is beneficial in peak hours. The neural network forecasts perform slightly worse than the Holt Winters forecast which is coherent with the earlier findings for the Malledijk, in addition the improvement is not significant according to the t-test.

The last scenario, 'No detector', was not included in Figure 59. To show the influence of the little look-ahead functionality the detector was removed for another 10 simulations. Delay during peak hours was off the chart with average delay per car over 100 seconds because of unstable queue growth on the side streams. This was because the controller was not able to cope with the peak demand due to the lack of knowledge of future arrivals. It can be concluded that 'little' look- ahead functionality of a detector at 70m actually makes quite a lot of difference for the controller.

OFF PEAK HOURS

All scenarios except for ANN and the Naïve predictor were run again for off –peak hours from 12:00 to 13:00h. The ANN and Naïve predictor were not simulated because both already performed worse during peak hours. Again all scenarios are plotted relative to the asymmetric intersection with no forecast:

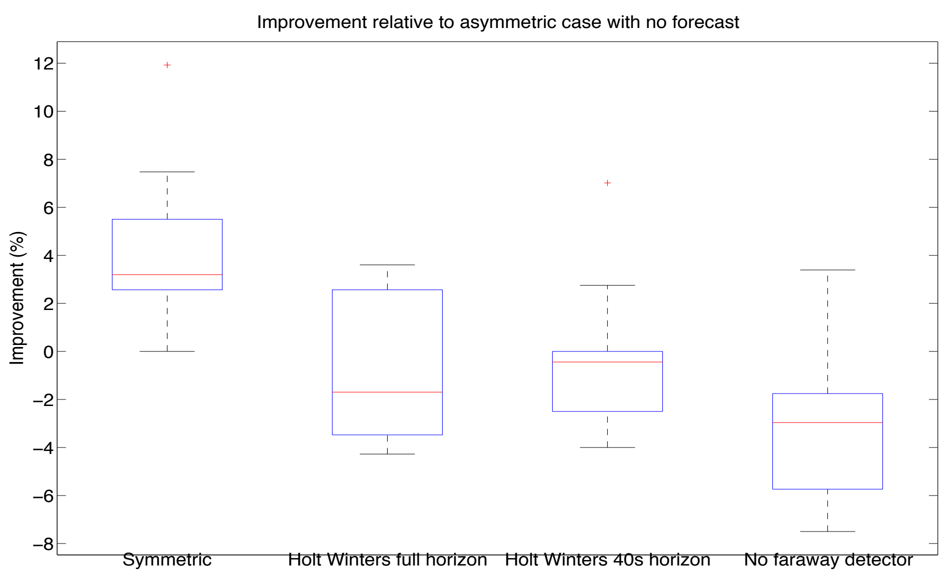
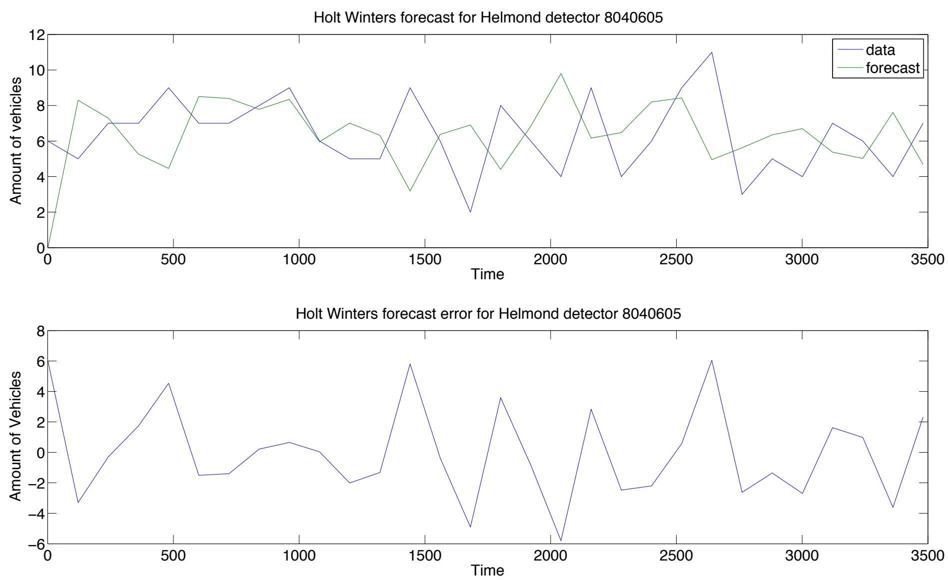


Figure 61 Simulation results for Helmond during off-peak hours

Table 8 Significane test for scenarios during off peak hours

Case	Symmetric	HW full	HW 40s	No detector
Paired t-test for significance (5%)	Passed	Rejected	Rejected	Passed

The Holt Winters forecast achieved an average RMSE of 3.28 with a variance of 0.49 for detector 8040605 during 10 runs.



Interesting to see here is that the Holt Winters forecast causes the controller to perform *worse* than with no forecast. In absolute terms the forecast error of RMSE 3.28 during off peak hours is almost equivalent to a forecast of 3.37 during peak hours, yet the controller performed worse overall. This can only be explained as a result of different traffic volumes because it is the only altered variable. During peak hours the volume ranges from 5 to 20 vehicles and during off peak hours 5 to 10 vehicles.

Logically the worse performance must be the result of the implementation of the aforementioned average headway. With low volume traffic the error between the forecasted average headway and the actual headway is larger because the traffic flow is less dense. During peak hours this error is smaller and the signal settings are less likely to be miss-timed.

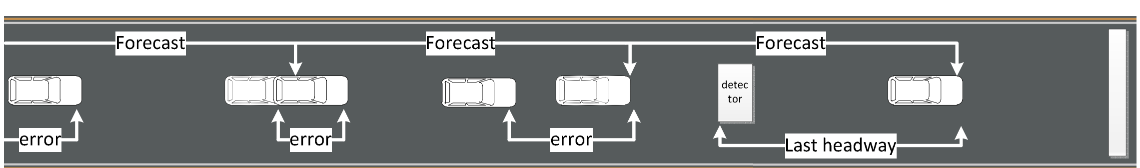


Figure 62 High density traffic

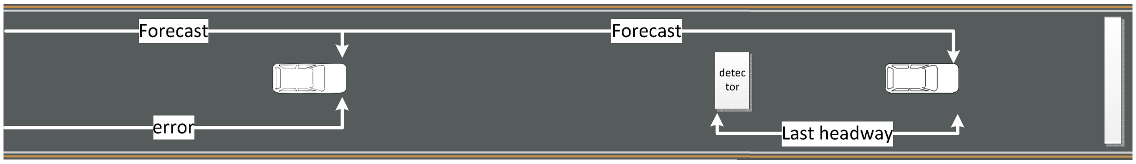


Figure 63 Low density traffic

In addition 10 simulation are run with the detector completely removed. During off peak hours the controller is now able to cope with the demand even when no look-ahead functionality is present. The difference between this scenario and the symmetric scenario is 8% deterioration. This is coherent with the earlier simulations done for the problem definition.

ALTERNATIVE DETECTOR LOCATIONS

All above simulations were run again with placement of the faraway detector at 50m and 90m from the stop line. First simulations were done with a faraway detector without forecast. This to visualize the pure impact of detector replacement in respect to controller performance. Simulations show the same profile of above results however delay increases for both cases (1 - 2%), although not significant.

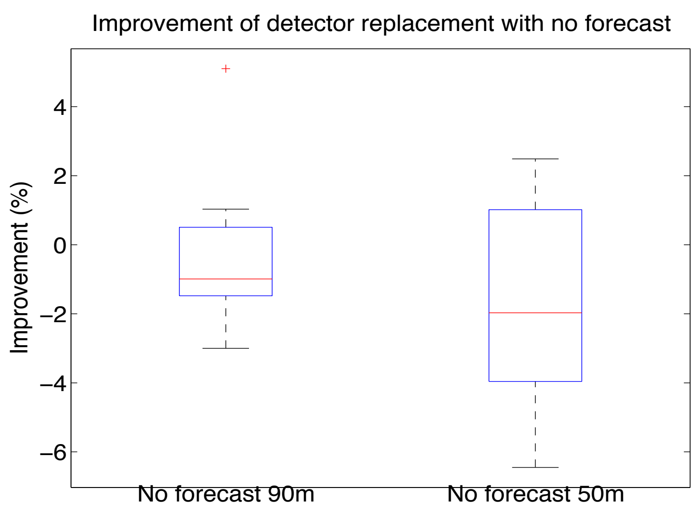


Figure 64 Impact of detector replacement for asymmetric intersection with no forecast

Bringing the detector closer than 70m will decrease the anticipation time and will negatively affect the performance of the controller. It is expected that moving the detector further away should increase the anticipation time and therefore the performance. This is not supported by the graph which suggest a small decrease in performance, however because the difference is not significant this remains inconclusive.

Table 9 Significance test for detector replacement without forecast

Case	90m	50m
Paired t-test for significance (5%) vs. 70m no forecast	Rejected	Rejected

Detector replacement can also affect the Holt Winters performance because of decreased 'correction' time. To elaborate, the Holt Winters algorithm predicts that a vehicle will arrive at a certain point in time, until that point in time is reached the controller will assume that the vehicle actually exists and will really appear at that moment in time. Consequently it will determine his signal plan on this information. When $t_{expected\ arrival} = t_{now}$ the vehicle should have appeared at the detector. If not, than the expected arrival list is corrected and the vehicle is removed. If the detector is closer to the stop line the controller has less time to adjust his signal plan for the correction and vice versa. Moving the detector closer will put more trust in the predictions; false predictions will therefore have increased impact.

Figure 65 confirms that the 1 – 2% losses of detector replacement are also translated to the Holt Winters forecast. The translation is not completely parallel for both cases. The effect on 50m is bigger than for 90m, this confirms our thoughts that moving the detector closer will cause a greater deterioration because of the Holt Winters algorithm. Otherwise HW 50m would have to be equally decreased as HW 90m parallel to the 'No forecast' replacements. The Holt Winters forecast at 70m gives the best result.

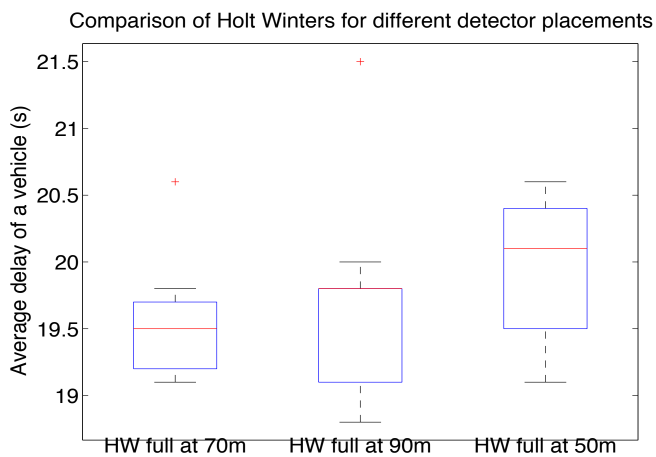


Figure 65 Holt Winters performance for detector replacement

CHAPTER 6 CONCLUSION AND RECOMMENDATIONS

This final chapter reviews the overall research, its findings, improvements and failures.

CONCLUSION

In an ideal situation the Multi- Agent Look-Ahead Traffic-Adaptive controller(Katwijk, 2008) shows a 20% increase over actuated control. In an ideal situation detectors are placed upstream (look- ahead detection) so that the controller can anticipate vehicle arrivals. When detector are placed further downstream this anticipation time decreases. In practice, due to cost constraints or because of a short intersection approach, detectors have to be placed close to the intersection. If a detector is placed at 100m from the stop line the anticipation horizon is equivalent to 4.5 seconds (80 km/h). With a small anticipation time vehicles are detected late, as a result the signal plan cannot respond to the latest variations in demand. The controller can optimize a 120 seconds plan for the future; placing the detector close to the intersection results in a gap of 120 – 4.5 = 115.5 seconds of unknown arrivals. To verify the decreased performance a simulation was done where 2 side streams had upstream detection removed, so the horizon for these links equals 0 seconds; the 20% increased performance is reduced to 10%. For every intersection link reduced from look- ahead detection to stop line detection a part of the 20% improvement will be lost. An intersection with varying detector configurations on the feeding links is called an asymmetric intersection. Techniques were researched to increase the knowledge of future arrivals when only nearby detection is available. This field of research belongs to forecasting techniques.

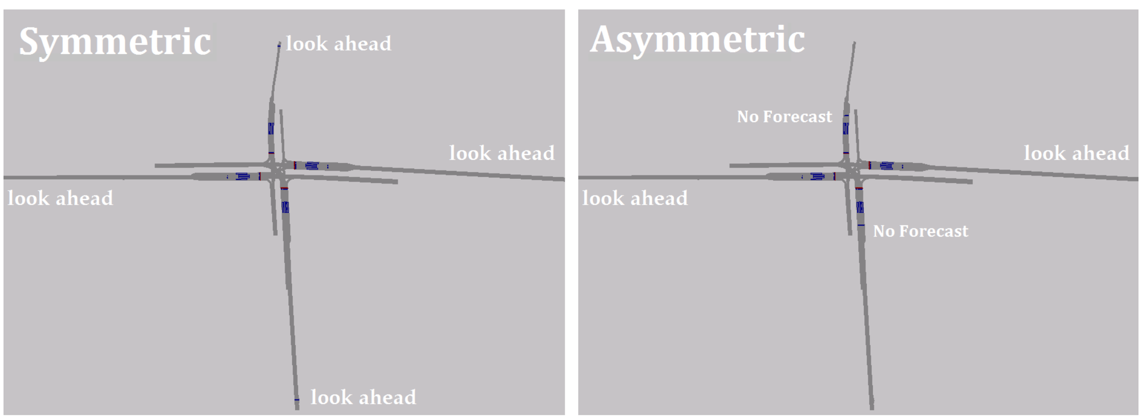


Figure 66 Visualization of both configurations

The literature study indicated that two separate forecasting techniques are necessary: one for individual arrivals and one for platoons. The literature study also revealed that Holt Winters and Artificial Neural networks are two promising forecasting techniques.

A specific arrival time for individual vehicles cannot be forecasted because the inter arrival times are Poisson distributed. In other words, there is no relation between arrivals of two successive vehicles. Individual arrivals are therefore forecasted by predicting the aggregated demand for the near future. The aggregated demand forecast is then transformed to the average distance between successive vehicles (headway) to have a forecast on second by second basis. Holt Winters and Artificial Neural Networks (ANN) were researched for forecasting aggregated demand. Both were implemented in the controller to answer the question if the forecasting techniques can improve the performance for an asymmetric intersection. In addition a Naïve predictor was implemented and tested in the controller that acts as a benchmark for the forecasting techniques.

The Naïve predictor records the average headway at the stop line and propagates this average headway into the future similar to the Holt Winters and ANN algorithm. The Naïve predictor only takes into account the most recent vehicles to calculate the current level of demand, it therefore misses a trend and seasonality component that are present in available demand and which are present in the aforementioned forecasting techniques.

Platoon forecasting required the development of a whole new model because no prior studies were found that tried to forecast platoons based on a single detection point. The model consists of multiple algorithms. The first algorithm extracts platoon characteristics, the second forecasts these separately using Artificial Neural Networks, the third ultimately construct a forecast on second by second basis. These platoon characteristics are Inter platoon headway, Platoon size, and Platoon volume. Inter platoon headway is the time difference between the start points of two successive platoons. Platoon size is the time difference between the first and last vehicle of a platoon. Platoon volume is the amount of vehicles in a platoon. An additional filter algorithm was developed to ensure a clean extraction of the platoon characteristics. The forecasting potential of the model proved to be insufficient that controller implementation was abandoned.

INDIVIDUAL FORECASTING

A 20% increase in performance was shows between the adaptive and actuated controller. With every link reduced from look- ahead detection to stop line detection the 20% increase in performance deteriorates step by step and starts to resemble an actuated controller again if the 20% improvement is completely lost. In the problem definition a deterioration, or increase in average delay of a vehicle, of 10% was measured if look-ahead detection is reduced to stop line detection for two intersection links in a simulation network. In terms of optimization horizon the removal of the look- ahead detector reduces prior knowledge of vehicle arrivals from 30 seconds to 0 seconds. This is again confirmed during the off peak hours simulations for the Helmond network where an 8% increase in delay is shown when look-ahead detection is removed on two links of the intersection. For peak hours, the same setup, resulted in bigger failure. The controller cannot cope with the increased demand and fails to unwind the queues on the side links. Because of the large queues the average delay of a vehicle increases with 423% compared to look- ahead detection.

First the effect of replacing a detector at 70m from the stop line will be discussed. Then the second step is adding a forecasting algorithm to the detector to see if further improvements can be realized. The addition of the forecasting techniques will be discussed separately for peak hours and off peak hours.

Placement of a detector at 70m allows for some 'look-ahead' detection with an equivalent horizon of 5 seconds. It showed that these 5 seconds of prior knowledge are already quite able to reduce the losses. During peak hours it is able to stabilize the system and unwind the queues. The 423% increase in delay now reduces to 2% increase in delay. During off-peak hours the 8% increase in delay reduces to 4%. The question is whether addition of a forecast at 70m can bring the 2% deterioration for peak hours and 4% for off peak hours back to 0%.

To answer which technique, Holt Winters, Artificial Neural Networks, or the Naïve predictor works best all were tested for peak hours and off peak hours. Peak hours are generally in the morning from 06:00- 09:00 h. Demand is characterized by a steep growth and an average level that is 1.5 to 2 times the average demand during off peak hours. For off peak hours a window from 12:00 – 13:00 was chosen because the average level is representative for the average level during off peak hours. Demand around this time is stable and shows no steep gradient.

PEAK HOURS

Addition of the forecasting techniques to the detector at 70m, Holt Winters, Artificial Neural Networks and Naïve predictor, only showed a significant improvement for Holt Winters during peak hours. During peak hours Holt Winters achieved a gain of 2% which was tested and passed for significance with a paired t-test with 95% confidence. The 2% improvement results in a 0% deterioration compared to look- ahead detection.

The Naïve predictor acts as a comparison for Holt Winters and the ANN performance. The fact that the Naïve predictor performed worse than the other predictors proves the importance of a trend and seasonality component in the forecast.

Closer inspection shows that Holt Winters actually slightly outperforms the look- ahead detection. At first this seems strange but the explanation is logical: it can be accredited to the fact that look- ahead detection had detectors at 400 meters which is equivalent to a 30 second horizon and Holt Winters forecasting technique is able to fully forecast the 120 second horizon. The additional prior knowledge results in an increased performance. To verify the impact of the extra horizon the Holt Winters forecast was also reduced to a 40 second horizon. This scenario indeed performed worse than equivalent look- ahead detection. Moreover it resulted in no overall improvement compared to the same configuration with no forecast. This emphasises the importance of Holt Winters to fully forecast the 120 second horizon.

OFF PEAK HOURS

Artificial Neural Networks and the Naïve predictor were not tested for off peak hours because they already resulted in a lesser performance than Holt Winters during peak hours.

During off-peak hours the Holt Winters forecast did not result in an improvement. Figure 61 even suggest a decrease in performance of 1% although this decrease is not significant. At least it can be safely concluded that there is no significant improvement. In other words, the forecasting technique is better left off. The aforementioned deterioration of 4% compared to look- ahead detection remains unchanged.

It is deducted that this loss can only be due to the reduced traffic demand. This is the only altered variable of the simulations and the forecast errors were equal to peak hours. When demand is low, traffic is less dense and the distance between successive vehicles can vary more greatly than when they are 'packed' together. Since Holt Winters delivers an average headway the forecast is more likely to be off target when the actual headway variance is greater. The Holt Winters algorithm should therefore be used during peak hours only.

OPTIMAL DETECTOR PLACEMENT

Three detector positions were researched: 50m 70m and 90m. The optimal point, although not significant, suggest detector placement between 50m and 90m. With 70m the only other location researched, the best location was determined here, with and without aggregated forecast, for peak and off peak hours. The exact optimal point would require further research. The study of detector placement resulted in two conclusions in respect to driving variables.

First, placing the detector too close will result in a smaller anticipation time for the controller to adjust to fast demand variations. Second, if a forecast technique is operational on the detector then placing the detector too close will decrease performance even further. One would expect that the deterioration of 90m and 50m compared to 70m with *no* forecast translates 1 to 1 for the simulations *with* Holt Winters forecast. However, the detector positioned at 50m showed a larger deterioration than the 90m case. When a forecast is

operational placing the detector closer to the stop line than 70m is therefore not advised because it results in an additional performance decrease. This is due to the inability of the controller to correct for false forecasts. The controller has less time to correct for false forecasts that were considered as true up until to the expected moment of arrival. The impact of the false forecasts is therefore greater.

PLATOON FORECASTING

Platoon forecasting required the development of a separate model. Instead of trying to forecast individual arrivals the model focusses on platoons and tries to forecast the arrivals of platoons accordingly. Platoon arrivals can still be translated to individual arrivals on a second by second basis for a 120 second horizon. The algorithm identifies three platoon characteristics, 'Inter platoon headway', 'Platoon size', 'Platoon volume' and forecasts these separately using Artificial Neural networks. An offline dataset was used to evaluate the forecasting potential of the model.

The model was built on upstream 'departures' and later tested on downstream arrivals. The upstream data is recorded just downstream of the upstream intersection. The reason for this was that the signal will appear more clean: less influenced by dispersion and queue interference. It will be easier to make observations for important signal features that can be sought for in the downstream signal. The upstream signal should be viewed as the ideal signal. Consequently, if the forecasts for the ideal signal are in line with expectations then there is a chance for success for the downstream signal, on the other hand, if it fails to work upstream than it will surely not work downstream.

PLATOON CHARACTERISTIC EXTRACTION

Extraction of the three platoon characteristics from the upstream signal resulted in interesting findings for the real world dataset. The platoon characteristics are able to supply a great deal of information about the traffic state: for instance the upstream signal cycle and knowledge of the varying demand of the side stream throughout the day. The varying demand on the side stream resulted in four different traffic states: from 20:00h - 05:00h the signal shows random arrivals of individual vehicles, from 05:00h – 10:00h the signal shows significant demand on all stream but with highly varying demand for the conflicting stream, from 10:00h – 15:00h the signal shows one dominant stream the SG:8 and one or two side streams with low demand, from 15:00h - 20:00h the signal shows one dominant stream with an occasional side stream platoon. Unfortunately the varying demand on the side streams results in an inconsistent appearance of these streams, the only dominant stream is the main stream following the SG:8. The inconsistent appearances of the side streams cause an erratic signal.

The filter was tuned to focus on the main stream by removing 'platoons' smaller than 4 vehicles. This further exposes the Inter platoon signal that represents the main stream in presence of side streams. After 15:00h the Inter platoon signal depicts purely the main stream because demand on the side streams is negligible. An occasional outlier appears that resembles a side stream but can be removed by up tuning the filter so that it completely uncovers the main stream platoon after 15:00h interspaced by a full signal cycle of 120 seconds.

The downside of up tuning the filter is a greater chance for accidentally removing platoons in the main stream. This will leave a large gap twice the size of the normal Inter platoon headway. The filter can therefore also create outliers.

The original signal which holds information on all streams and the trimmed signal that shows only the main stream platoons were both tested for autocorrelation. In the original signal one would expect to find that every fourth platoon shows resemblance with the first, in the second signal one would expect to find no correlation

because the signal represents only one stream and all platoons are therefore 'alike' in terms of platoon characteristics.

Indeed more correlation is found in the original signal than in the trimmed signal. The trimmed signal shows no significant correlation with past data points for all characteristics. However, the correlation that appears in the original signal shows relatively high correlation around lag 1,5 and 8 for the Inter platoon signal. This is not coherent with the expected lags of 3, 6 and 9. The shift remains unexplained.

The platoon characteristics were also extracted for the downstream signal. Both signals are very similar but at the downstream signal platoons are less coherent because the platoons disperse. However, the increased dispersion in the Inter platoon signal might be small enough for a reasonable overall forecast based on historic information. Platoon size also shows a similar profile to the upstream signal but there is a clearly visible increase.

The difference of the upstream and downstream Platoon size signal shows the increased dispersion of the platoons over the link. The dispersion of the Platoon size signal suggests that dispersion is constant. For proof the Platoon size signal needs to be corrected for the changes in Platoon volume. If this is the case then platoon characteristics are a promising method for forecasting downstream arrivals based on upstream detection.

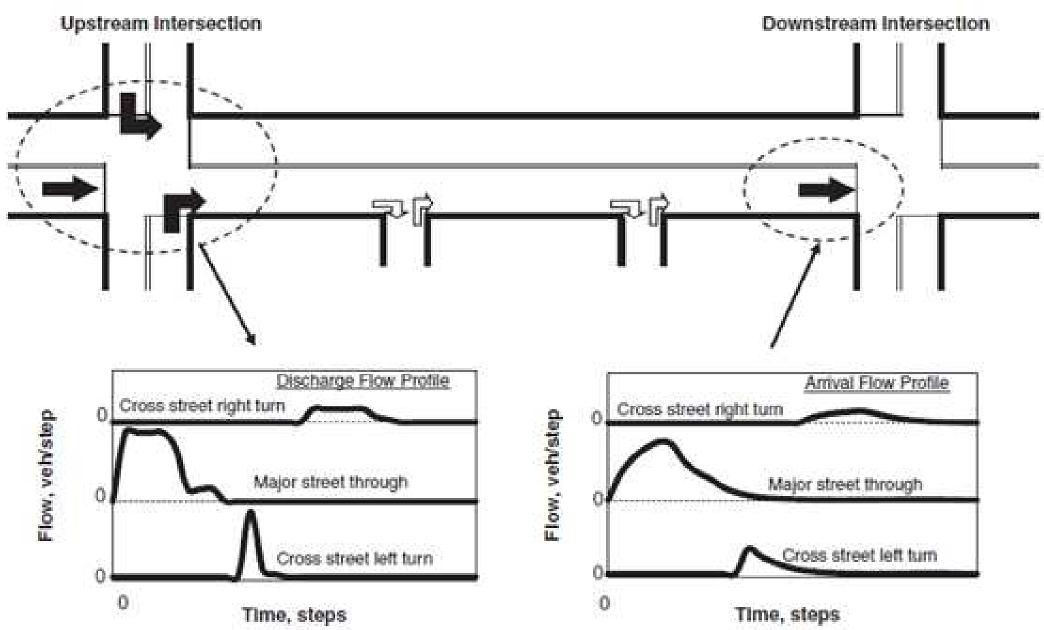


Figure 67 Upstream versus downstream signal

Platoon volume is almost similar to the upstream signal, the changes can be accredited to platoon dynamics throughout the link: platoons can merge or separate showing differences in platoon volume. The differences are around 5 vehicles which is coherent with the general volume of a side stream platoon. If this platoon merges with a main stream platoon an increase of 5 vehicles is the expected, or vice versa.

The method of extracting platoon characteristics shows promise which could also be used in future research. With further study the autocorrelation could be further uncovered. Furthermore an extensive transient analysis could be done on the progression from the upstream signal to the downstream signal. These early results already give a good visualization of the evolution of the platoon characteristics.

TRAINING THE NETWORK

Before the neural network can make forecasts it needs to be trained on a representative dataset. During training the network tries to fit a non-linear function on the data; this function can then later be used for forecasts. Training of the network was done for the ideal upstream dataset.

A first attempt was made to fit the ANN on the signal that includes information on all streams. The side streams appear as outliers because the main stream is the dominant stream. The outliers make for a very difficult fit and almost all the outliers are not reached by the fit. In order to forecast side stream platoons the network must also include the outliers. Since the outliers are not included it can be expected that the overall forecast will not include side stream platoons.

The decision was made to focus on the main stream signal and remove the side stream vehicles. This can be done by up tuning the filter to exclude platoons consisting of 4 vehicles or less. The training was reasonable for Inter platoon headway, platoon size and Platoon volume resulted in worse fit. The reasons for the bad fit can be accredited to two reasons. In all signals outliers were removed by filter but some new outliers were also created. This is due to unwanted removal of small platoons in the main stream.

The Platoon size signal shows a relatively high amount of variance which can be accredited to the little dispersion that already appears at the upstream detection point but mostly to the varying platoon volumes. When looking at the Platoon volume signal it becomes clear that the amount of vehicles in the main stream platoons varies continuously between 10 and 20 vehicles. There are some outliers in platoon volume but they are negligible compared to the variations in demand. Both signals are related; the bad fit for Platoon volume therefore also results in a bad fit for Platoon size which includes an additional stochastic effect; dispersion.

FORECAST EVALUATION

To evaluate the forecasting potential simulations were run for an hour on the real world upstream dataset. The separate forecasts of the platoon characteristics are transformed back to an expected arrival list on a second by second by second basis. For ease of reference this is called the overall forecast.

The results for the overall forecast showed an average 50 % of vehicles predicted correctly for the upstream detection point. But on the down side the model forecasted 70 to 80 % vehicles falsely. The falsely predicted vehicles are weighed more heavily because allocating green time to vehicles that are not there is inefficient and is perceived as very annoying by the drivers. The forecasting performance of the platoon algorithm is disappointing.

A mistake was identified in the inverse algorithm that constructs the platoon characteristics back to an overall forecast on a second by second basis. Therefore the model's performance was below expectations. In the future the model could be corrected for the mistake, however it is advised to first further disintegrate the Inter platoon signal to uncover the 3 signals for each of the conflicting streams. Forecasts can then be made based on the origin of the platoon. Downstream implementation was abandoned because the model did not work on the upstream signal.

FINAL CONCLUSION

To conclude, Holt Winters is a robust and comprehensible algorithm and outperformed the Artificial Neural Network for individual forecasting. Moreover, relative to the Artificial Neural Networks the Holt Winters algorithm should be fairly easy to embed in the source code of the controller. It is advised to use the Holt Winters algorithm in peak hours only.

Although the difference is not significant, the best position seems to be between 50m and 90m. The only other position that was measured for this interval was 70m and shows an improvement with respect to the 50m detector. It should definitely not be positioned closer than 70m, placing it further away than 70m remains inconclusive.

The platoon forecasting research started promising but ended disappointing. Nonetheless the use of platoon characteristics as way to study platoons and platoon behaviour is recommended for future use. This study showed that a lot of information can be extracted from the platoon characteristics.

Further development of the platoon forecasting model for use on a single detection point is advised if the model can be improved to work on the upstream signal. A clustering technique could be used to extract separate signals for the each of the conflicting streams. If signals can be extracted that are based on origin extra patterns might be uncovered that can be used for forecasting.

RECOMMENDATIONS

Some final remarks for further research. First of all the Holt Winters algorithm was implemented with only one seasonal component. The literature study revealed that there can be high correlation with data a week back. For further improvement of the Holt Winters forecasting capability it is advised to include more historical knowledge.

The research focussed on an asymmetric intersection. Holt Winters was proven to increase overall performance by extending the forecasting horizon for a small horizon. This exact same algorithm could also be applied to look-ahead detection to extend the horizon from 120 seconds to 240 seconds. If the optimization algorithm allows for such a change Holt Winters could also prove to be beneficial in this situation.

In respect to the platoon forecasting algorithm there are the following remarks. First of all more research is required to verify the general applicability of platoon characteristic extraction or that this is very case specific. If this is the case the main determinants should be identified that result in clean platoon characteristic signals.

The platoon characteristic analysis showed that the filter is sub optimal. A few recommendations are made in respect to filter improvement. The different platoon characteristics can be cross-validated to identify matching outliers. If upstream information is included the outliers can be even be more clearly identified.

A clustering technique is advised to identify the origin of the platoons. If the origin of a platoon is known then a signal can be constructed based on an origin. This could uncover more patterns which improve the forecasting potential.

Lastly, the evolution of platoon characteristics from the upstream signal to the downstream signal requires further study. Early results showed that platoon characteristics could be a good method to determine the amount of dispersion. A quick inspection even hints that dispersion is constant if Platoon size is corrected for Platoon volume. An Artificial Neural network was tried outside the scope of this paper to see if the downstream signal can be forecasted with the upstream signal. This resulted in the finding that the downstream signal cannot be compensated for the phase shift if a forecast is desired. The upstream and downstream signal have to be supplied to the network without the compensation. The network should then determine the phase shift by itself. With such a difficult signal it can be said beforehand that the ANN is not capable of doing this. A method other than Artificial Neural networks should be developed in that case.

REFERENCES

Barth, M., & Boriboonsomsin, K. (2008). Real-World Carbon Dioxide Impacts of Traffic Congestion. *Transportation Research Record*(2058), 163-171.

Basak, D., Pal, S., & Patranabis, D. C. (2007). Support Vector Regression. *Neural Information Processing, 11*(10).

Bonneson, J. A., Pratt, M. P., & Vandehey, M. A. (2010). Predicting Arrival Flow Profiles and Platoon Dispersion for Urban Street Segments. *Transportation Research Record*(2173), 28-35.

CT4821, C. n. (2006). ET4-024 Traffic Guidance Systems. *TU Delft*.

Dia, H. (2001). An object- oriented neural network approach to short- term traffic forecasting. *European Journal of Operational Research, 131*, 253-261.

Duinkerken, M. (2009). WB3420 College Slides Logistic Engineering.

Francke, J., Derriks, H., Gordijn, H., Groot, W., & Savelberg, F. (2010). Verkenning mobiliteit en bereikbaarheid 2011-2015. *KiM*.

Ghosh, B., Basu, B., & O'Mahony, M. (2004). Time Series Modeling For Forecasting Vehicular Traffic Flow in Dublin. *Unpublished*.

Head, K. L., & Mirchandani, P. B. (1994). Hierarchical Control Architecture for Real-Time Traffic Management. *Automatic Control - World Congress 1993, Vol 3*, 145-148.

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting, 20*(1), 5-10.

Hu, S. R., Madanat, S. M., Krogmeier, J. V., & Peeta, S. (2001). Estimation of dynamic assignment matrices and OD demands using adaptive Kalman Filtering. *Its Journal, 6*(3), 281-300.

Katwijk, R. T. v. (2008). *Multi-agent look-ahead traffic-adaptive control*. Delft: TRAIL Research School.

Kirby, H. R., Watson, S. M., & Dougherty, M. S. (1997). Should we use neural networks or statistical models for short-term motorway traffic forecasting? *International Journal of Forecasting, 13*(1), 43-50.

Liu, H. X., Wu, X. K., Ma, W. T., & Hu, H. (2009). Real-time queue length estimation for congested signalized intersections. *Transportation Research Part C-Emerging Technologies, 17*(4), 412-427.

Liu, X. M., & Lu, S. F. (2007). Vehicular traffic flow dispersion prediction of coordinated signal control based on support vector regression. *Proceedings of the 26th Chinese Control Conference, Vol 6*, 49-52.

Mashros, N. B. (2007). Platoon dipsersion downstream of traffic signal.

Milosivic, M. (2010). Wegwijzer Doorgaand Verkkeer.

Mourik, H. v., W.Groot, & J.Francke. (2008). Verkenning wegverkeer 2012.

Olstam, J. J., & Tapani, A. (2004). Comparison of Car-following models. *Swedish National Road and Transport Research Institute*.

Pacey, G. M. (1956). The Progress of a Bunch of Vehicles released from a Traffic Signal. *Unpublished*.

Qiao, F. X., Yang, H., & Lam, W. H. K. (2001). Intelligent simulation and prediction of traffic flow dispersion. *Transportation Research Part B-Methodological, 35*(9), 843-863.

Rijkswaterstaat. (2002). Signaalgroepafwikkeling detectieconfiguratie.

Robertson, D. I. (1969). TRANSYT - A Traffic Network Study Tool. *RRL Report, LR 253*.

Talluri, K. T., & Ryzin, G. v. (2004). *The theory and practice of revenue management Online resource*. Boston: Kluwer academic.

Taragna, M. (2011). Kalman Filter Theory.

Taylor, J. W. (2010). Triple seasonal methods for short-term electricity demand forecasting. *European Journal of Operational Research, 204*(1), 139-152.

TUDelft. (2008). Reader Knowledge Based control systems.

Vigos.G, & Papageorgiou, M. A. (2010). A simple estimation scheme for the number of vehicles in signalized links.

Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering-Asce, 129*(6), 664-672.

Xie, Y. C., Zhang, Y. L., & Ye, Z. R. (2007). Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. *Computer-Aided Civil and Infrastructure Engineering, 22*(5), 326-334.

Y.Jiang, Li, S., & Shamo, D. E. (2003). Development of Vehicle Platoon Distribution

Models and Simulation of Platoon Movements on

Indian Rural Corridors.

Yin, H. B., Wong, S. C., Xu, J. M., & Wong, C. K. (2002). Urban traffic flow prediction using a fuzzy-neural approach. *Transportation Research Part C-Emerging Technologies, 10*(2), 85-98.

WEBSITES

http://en.wikipedia.org/wiki/Motor_vehicle

http://www.compendiumvoordeleefomgeving.nl/indicatoren/nl0082-Emissies-naar-lucht-per-doelgroep%2C-volgens-IPCC-en-NEC.html?i=5-70

http://www.pbl.nl/en/dossiers/Climatechange/moreinfo/Chinanowno1inCO2emissionsUSAinsecondposition

http://www.guardian.co.uk/environment/2008/aug/01/renewableenergy.climatechange

http://www.who.int/mediacentre/factsheets/fs313/en/index.html

APPENDICES

APPENDIX A MULTI-AGENT LOOK-AHEAD TRAFFIC-ADAPTIVE CONTROL SYSTEM

The Multi- Agent Look- Ahead Traffic- Adaptive controller has been engineered by R.T. van Katwijk at TNO and DCSC(Katwijk, 2008). The controller is the centre piece of this thesis. The working principle will be discussed and the features that set it apart from the establishment. The place of the controller in the system becomes clear from the following diagram:

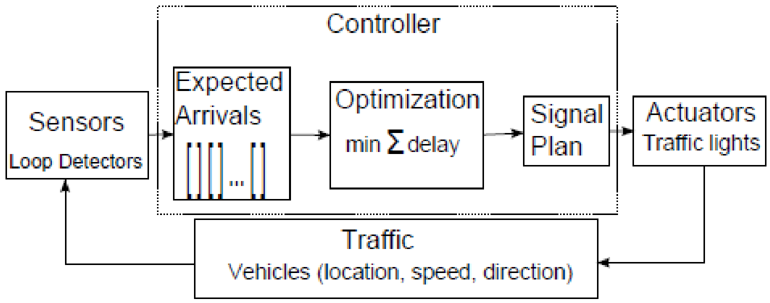


Figure 68 Position of Adaptive controller in system (Katwijk, 2008)

When inspected closely it can be noticed that this diagram is coherent with the control scheme in Figure 5.

The algorithm optimizes a certain cost function. Optimizing means finding the right sequence of decisions to minimize the total cost.

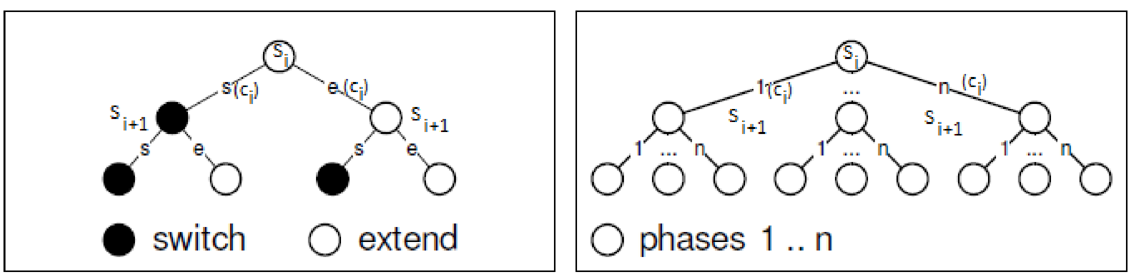


Figure 69 Different search trees(Katwijk, 2008)

Above two different decision trees are shown. In both trees each node represents a system state S_i. The line that leads to the next state incurs a costc_i; the number of possible next states or number of emerging lines represents the possible outcomes from a decision variable. Already at this point it becomes clear to formulate a decision variable$u \in U$ that gives the minimal amount of choices. The third generation restricts itself to a simple decision variable; switch or extend which keeps the search space relatively small.

The objective function takes the following shape:

$$f^*(i) = \min_{u \in U(i)} \left\{ c_i(u) + \gamma \sum_{j \epsilon S} p_{i,j}(u) f^*(j) \right\} \qquad (21)$$

It should be interpreted as follows; determine the sequence of decisions (u_k … … u_k) so that the current cost plus the future cost is minimized. Future cost is the probability $p_{i,j}$ for getting to the next state times the cost for going to next state j multiplied by a reduction factor γ.

Although all adaptive controllers have the same goal to optimize for short term demand variations they differ on numerous aspects:

- Architecture (central, distributed)
- Search algorithm (stage, or movement based)
- Control decision (phase time, sequence)
- Prediction model
- Planning Horizon

Architecturally there are arguments for using central coordinated control versus distributed control. Central coordination communicates with individual intersections an optimal plan for minimizing delay over the whole network. In theory this sounds very nice because the biggest gains can be achieved in networks. However it goes without saying that the problem becomes very large and takes a long time to solve. If not done properly the same problem arises as with 1st generation controllers where due to delay an aged signal plan is applied to a changed current state. Also a central system becomes a single point of failure.

A distributed approach has the benefit that it is easily scalable as well that there is no single point of failure. Achieving optimal results for the network is more problematic. There are systems that communicate with neighbouring intersections [PRODYN, UTOPIA-SPOT]. However an optimal plan achieved for A and B might not be coherent with the plan for B and C.

There are also systems that try to combine a distributed approach with a higher level algorithm that tries to optimize for the network. Examples are UTOPIA-SPOT, RHODES, and OPAC-VFC. These are hierarchical systems. This can be achieved by optimizing for the network and add the results as constraints in the local optimizations.

The Multi- Agent Look-Ahead Traffic-Adaptive controller implements a distributed approach but can be linked to neighbouring controllers.

The search tree is explored by a search algorithm. Since the size of the search tree is correlated with the computation time methodologies have been developed to narrow the search space or cleverly explore it. Doing a complete search will ensure an optimal solution; a good solution, however, is in many cases good enough. A distinction can be made between constructive and move-based algorithms.

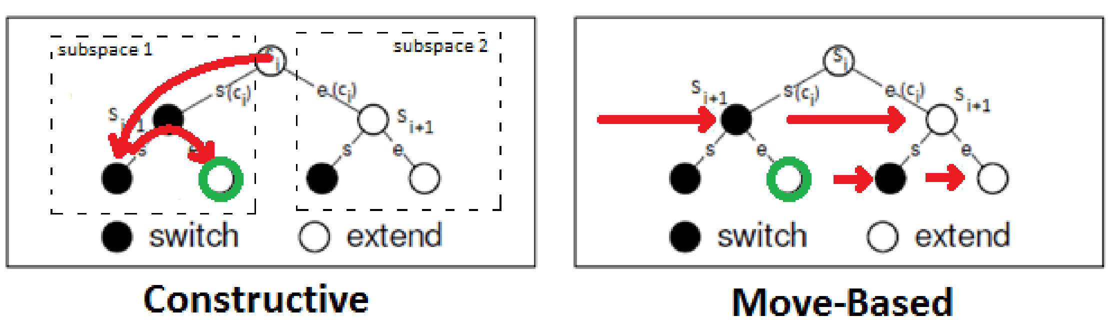


Figure 70 Search algorithms

A move-based approach will first inspect local solutions before delving deeper in the search tree. This can result in missing the optimal solution (indicated by the green circle) but will mostly result in a good solution more quickly. Whereas most 3rd generation controllers apply a constructive search method, the Multi- Agent Look-Ahead Traffic-Adaptive controller utilizes a move-based algorithm.

Optimization comes down to making the right decision without squandering the possibility of a future higher profits. In order to do this some knowledge is required of the future states. This is done using prediction models. A better prediction model will directly improve overall results. In the airline industry the same method is used to calculate airline tickets. A 20 % improvement is said to have revenue increase of 1%(Talluri & Ryzin, 2004).

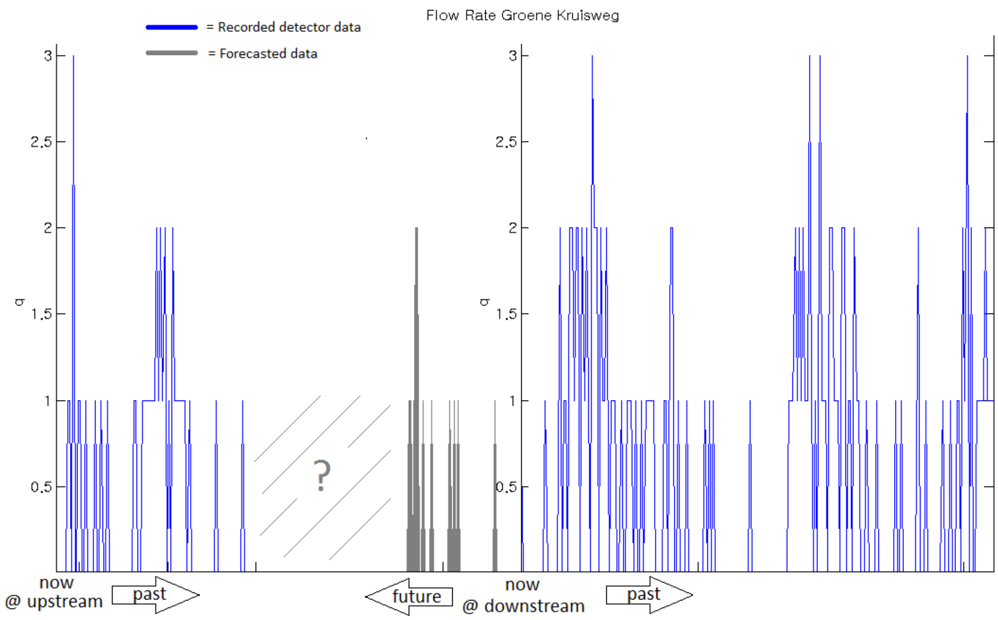


Figure 71 Schematic diagram of upstream and downstream detector recordings

In traffic control the established controllers either use prediction of near future arrivals by placing detectors relatively close so a the forecast horizon is long enough and the uncertainty minimal. Or, alternatively, prediction is done based on information received by detectors just downstream of the upstream intersection (look- ahead).

The competitive advantage of the Multi- Agent Look-Ahead Traffic-Adaptive controller primarily comes from the move –based approach. It is able to reduce the search space by eliminating impossible phases beforehand and by cleverly allowing extra non-conflicting signal groups to be added without the need of current phase cancellation. The interested reader is referred to the paper for elaborate analysis. Consequently the algorithm finds good results and achieves better computation speeds due to its move based approach.

APPENDIX B QUEUING MODEL RESEARCH

Liu et al. developed a model that estimates the maximum queue length for a saturated flow using shock wave theory. They use a far-away loop detector positioned at 100 m from the stop line together with signal information to derive some important flow information that is used in their shockwave model to estimate the maximum queue length during the last cycle. In our case the maximum length is of less importance; more important is the residual queue D at the start of the new red phase.

Their model is used to extract a set of equations to estimate the residual queue. In addition a shortcoming is identified and improved. Firstly the original model by Liu et al. shall be shortly discussed:

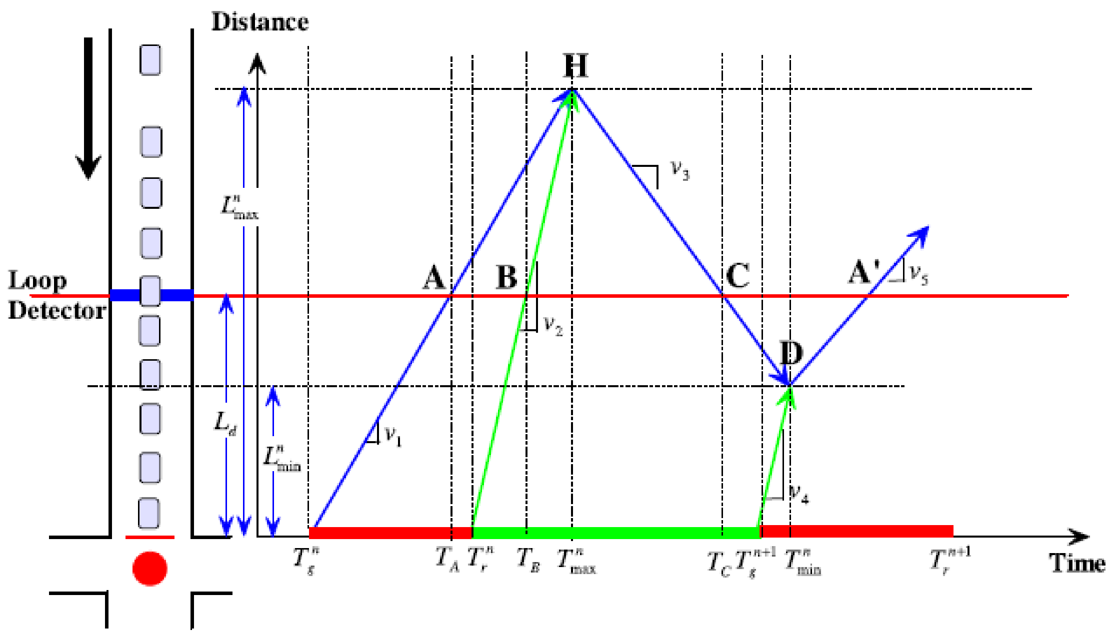


Figure 72 Queuing model for saturated conditions by Liu et al.

Five vectors are indicated by their slope $v_{1,2,3,4,5}$. In order of appearance: the first vector v_1. indicates the shockwave of the growing queue during the red phase. At the start of the green phase an unwinding shockwave starts v_2. and travels through the queue to the last car. The last car is indicated by the point H and also indicates the maximum queue length. The moment the unwinding shockwave reaches the last car the queue can actually start to resolve itself; indicated by the third vector. The moment the green phase ends a new shockwave is propelled backwards and leaves a residual queue indicated by the point D. After this point the queue starts to grow again because of new arrivals v_5.

The horizontal red line indicates the position of the loop detector. The vectors must be constructed based on observations made at this detector. These observations are made at point A,B,C.

Point A is reached when the detector is continuously occupied.
Point B is reached when the detector is no longer permanently occupied.
Point C is reached when a 'large' inter-arrival is recorded between the last car of the queue and the new arrivals.

SHORTCOMING

The shortcoming of the model lies in the assumption that new arrivals will only appear after start of the new red phase. The vector v_5 can actually appear anywhere along the line H-D. Researching our own dataset is became clear that new arrivals might actually arrive during the next green phase (point D). Extending the model to include this extra feature is easier than it might seem:

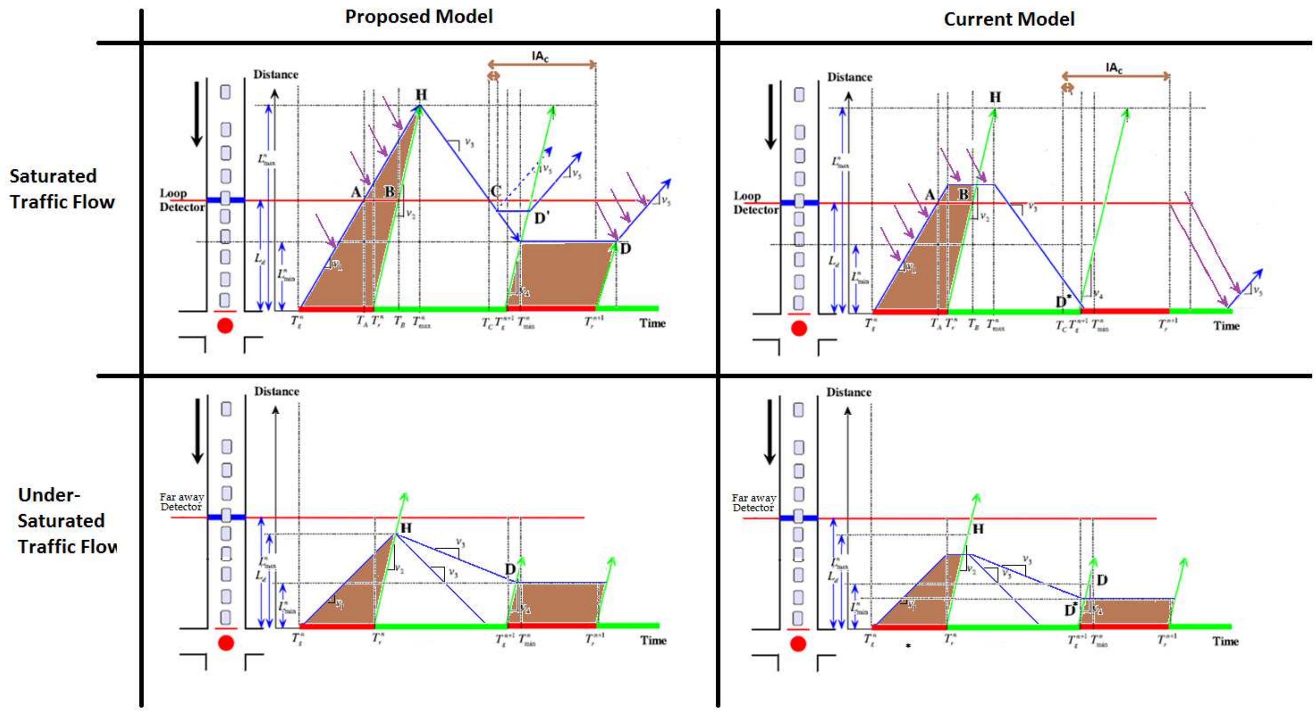


Figure 73 Shortcoming of vertical queuing model and shockwave model by Liu et al.

The detected inter arrival time at point C tells us when the queue starts rebuilding. If new cars start arriving before the old queue was able to resolve itself; point D lies on H-C, it is not possible to make a proper estimate of the residual queue.

The new model is easily verified by a sample from the Groene Kruisweg east bound. The left lane faraway detector records point A,B and C and the newly arriving platoon is given by IA_c:

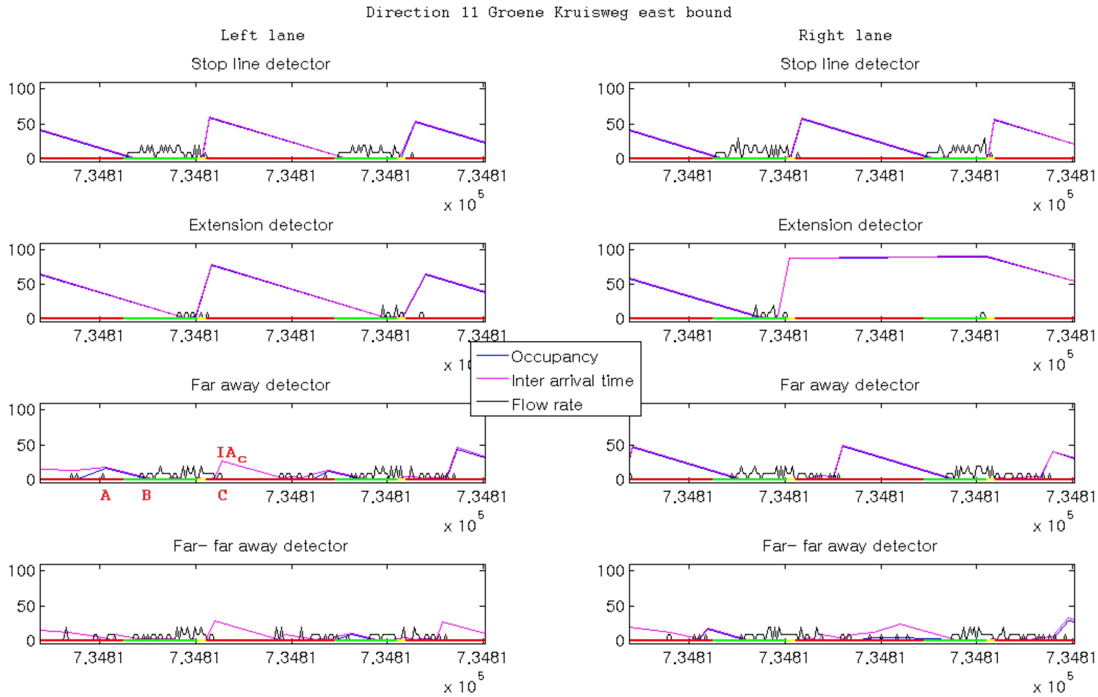


Figure 74 Identifying characteristic points for shockwave model

POINT C

An extra traffic state makes identification of C difficult. This is when point C cannot be located due to very fast arrival of new vehicles. This situation becomes more clear with the following time-space diagram; the saturation headway lies below the threshold value for identifying point C:

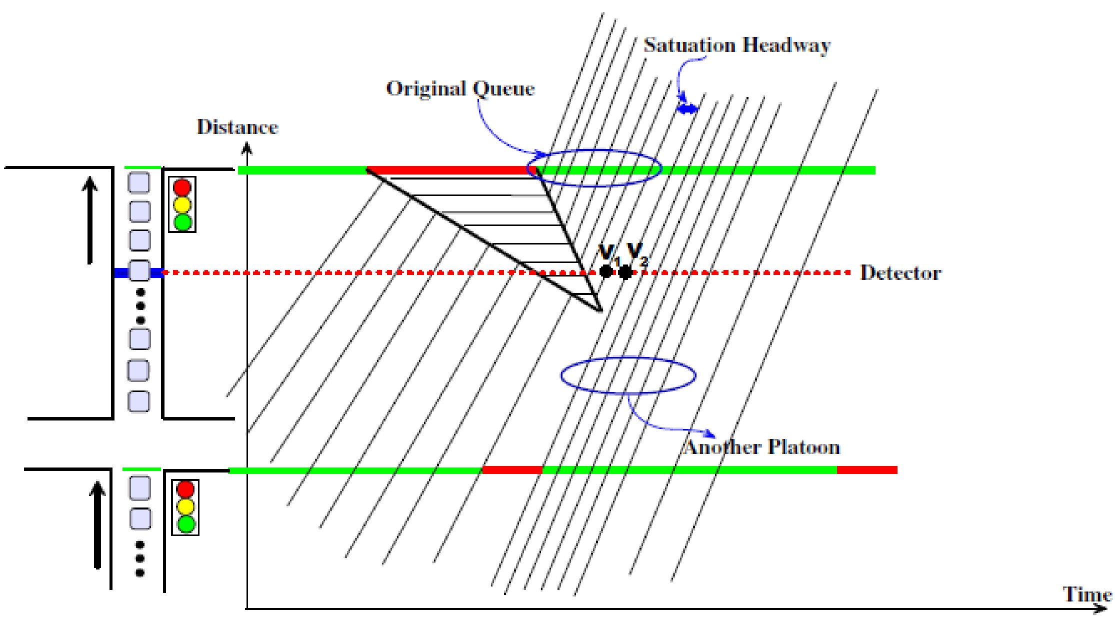


Figure 75 Introducing speed measurements in shockwave model

For future research I advise to investigate the speed difference between the two vehicles. Possibly this can be used to identify point C when $v_2 \gg v_1$, or $occ_2 \gg occ_1$

DELAY

The brown area indicates the total delay caused during a red phase. Visually it shows that the current vertical queuing model will generally underestimate the delay in both the saturated and under-saturated case. In case of under saturated flow the impact of underestimation might be questionable. In case of saturated flow the problem of underestimation becomes more apparent.

In the above example; in reality the maximum queue length ‘H’ lies much further upstream than is presumed in the vertical queuing model. The vertical queuing model therefore falsely assumes that it has no residual queue left at the end of the green phase. Or even worse, the controller might have determined to terminate the end of the green phase at that point due to it’s false assumption that queue is resolved. The actual residual queue is indicated in the top left picture by the brown parallelogram.

In addition the real maximum queue length is of extra importance due to its possible blocking of other lanes.

In terms of decision making, the decision comes down to choosing the right moment to change from green to red. The delay caused in the past is only important for registration purposes but is not a property that can be affected for more optimal control. The following diagram shows the cycle for the current green phase and a conflicting stream. To clarify, the brown area indicates the realized delay in the past whereas the grey area is the controlled delay.

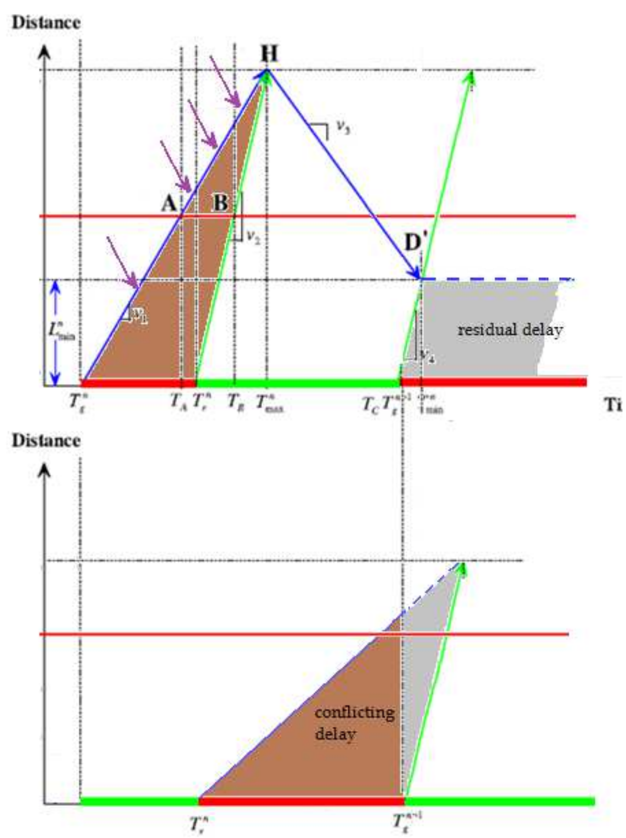


Figure 76 Idea or a new control scheme

After the point T_g^{n+1} is determined the above diagram reinitiates, the control decision remains the same only a new stream becomes the main stream residing in green phase.

The control problem could also be described as choosing a the right sequence of T_g^{n+1} so to minimize the sum of the surface areas given by residual delay and all the conflicting streams.

$$residual\ delay(T_g) = D \cdot (T_r - T_g^{n+1})$$

$$conflicting\ delay(T_g)^i = 0.5 \cdot (T_g - T_r^n) \cdot \tan(v_5) \cdot (T_g - T_r^n) + grey\ area \tag{22}$$

Find the right sequence T_g so to:

$$Min\,\{residual\ delay + \sum_{i=1}^{k} conflicting\ delay^i\} \tag{23}$$

For: k conflicting streams.

To calculate point D it is not necessary to first calculate point A and B as will become clear:

APPENDIX C CODE

```
%This is the main process for running an offline Holt Winters simulation,
%the most important function in this script are initHW: initializes Holt
%Winters and updateHW: updates and makes a forecast for every second, the
%last section generates the necessary graphs for evaluation

clear all;

%online or offline simulation? (online = 1, offline = 0)-------------------
global online;
online = 0;

%Initialize HOLT WINTERS algorithm-----------------------------------------
initHW_test('/Users/alexanderkrstulovic/Documents/MATLAB/SimulationFunctions/Helm
'act_noforec_assym_yest', 'act_noforec_assym_db_yest', 'act_noforec_assym_dbb_yes


%Load dataset--------------------------------------------------------------

load('det18 120 split.mat');
data.q  = q{1,2}.q(2:end);      %tuesday
data.t  = q{1,2}.t(2:end);


%Simulate forecast for dataset---------------------------------------------
for i = 1:1:length(data.q)-1;
        [Forecast] = updateHW_test(data.q(i),i, 18, 120, 1);
end

%Evaluation----------------------------------------------------------------
subplot(2,1,1)
plot(data.t,data.q);
hold all;
plot(data.t,Forecast_History{1,1});
legend('data','forecast');
plotDateStr(data.t, 60) ;
title('Holt Winters forecast for Malledijk')
ylabel('Amount of vehicles')
xlabel('Time')


subplot(2,1,2);
error = data.q - Forecast_History{1,1};
plot(data.t, error);
plotDateStr(data.t,60);
title('Holt Winters forecast error')
ylabel('Amount of Vehicles')
xlabel('Time')

linkaxes;

%Calculate RMSE------------------------------------------------------------
rmse = sqrt(mean(error.^2));
```

```
function [ ] = initHW_test(folder, fileName_yest, fileName_db_yest,...
                fileName_dbb_yest, location, aggregation)
%This function initializes Holt Winters and can be called online or offline
%This function only needs to be called once
%Holt winters is initialized for each individual detector "j"

global online;
global S_dbb_yest;
locationMapping = [];

%Initialze Holt Winters for all the required detectors---------------------
for j = 1:length(location)

%load data depending on online or offline simulation-----------------------
if online == 1
    locationMapping = [locationMapping location{j}(1)];

    [data_dbb_yest.t data_dbb_yest.q] = readMESdata_test(folder,...
    fileName_dbb_yest, locationMapping(j));

    [data_db_yest.t data_db_yest.q] = readMESdata_test(folder,...
    fileName_db_yest, locationMapping(j));

    [data_yest.t data_yest.q] = readMESdata_test(folder,...
    fileName_yest, locationMapping(j));

    %Aggregate-------------------------------------------------------------
    if diff(data_yest.t(1:2)) ~= aggregation;
        [data_dbb_yest.t data_dbb_yest.q] = doQ_test(data_dbb_yest,aggregation);
        [data_db_yest.t data_db_yest.q] = doQ_test(data_db_yest,aggregation);
        [data_yest.t  data_yest.q] = doQ_test(data_yest, aggregation);

    end
else
    locationMapping = [locationMapping location(j)];
    load('det18 120 split.mat')

    data_dbb_yest.q  = q{1,2}.q(2:end)  % tuesday
    data_dbb_yest.t  = q{1,2}.t(2:end)
    data_db_yest.q  = q{1,3}.q(2:end)   % wednesday
    data_db_yest.t  = q{1,3}.t(2:end)
    data_yest.q     = q{1,4}.q(2:end)   % thursday
    data_yest.t     = q{1,4}.t(2:end)
end


%Offset data by arbitrary amount of 1000----------------------------------
data_yest.q = data_yest.q + 1000;
data_db_yest.q = data_db_yest.q + 1000;
data_dbb_yest.q = data_dbb_yest.q + 1000;

%Initialize Seasonality for two days back (S_dbb_yest)---------------------

for i = 1:length(data_db_yest.q);
    S_dbb_yest{j}(i) = double(data_db_yest.q(i)) / mean(data_db_yest.q);
end


%Optimize Parameters-------------------------------------------------------
```

```
% starting values for alpha, beta, gamma are set to 0.5
% the parameters are optimized within the set 0 to 1

global x_hist;
global S_yest
x_hist = [];

[x fval] = fmincon(@(x) holtwintererror_test2(x, j, data_yest.q,...
           data_db_yest.q), [0.5,0.5,0.5],[],[],[],[],[0 0 0],[1 1 1]);

x_scat{j} = [x_hist]; %for scatterplot of search space


%Assign optimized parameters for future use--------------------------------
alpha{j} = x(1);
beta{j} = x(2);
gamma{j} = x(3);
Data_History{j} = [];
Vissum_History{j} = [];
Forecast_History{j} = [0];
Level{j} = [];
Trend{j} = [];
end

%Save all important parameters to a 'base' workspace-----------------------
assignin('base','locationMapping',locationMapping);
assignin('base','alpha',alpha);
assignin('base','beta',beta);
assignin('base','gamma',gamma);
assignin('base','S_yest',S_yest);
assignin('base','Data_History',Data_History);
assignin('base','Level',Level);
assignin('base','Trend',Trend);
assignin('base','headwayArrival',120);
assignin('base','Vissum_History', Vissum_History);
assignin('base','Forecast_History', Forecast_History);
assignin('base','x_scat', x_scat);

end
```

```matlab
function [MAE] = holtwintererror_test2(x,j, data_yest, data_db_yest)
%This function optimizes the parameters alpha, beta and gamma for Holt
%Winters. This is done by minimizing the Mean Absolute Error between the
%forecast and the validation set.
%This function is called in "initHW"

global S_dbb_yest
global S_yest
global x_hist;

% data in goede formaat zetten

[m n] = size(data_yest);
if m > n; data_yest = data_yest'; end;
[m n] = size(data_db_yest);
if m > n; data_db_yest = data_db_yest'; end;


% initialize Level (A), Trend (T), for first two timesteps for day before
% yesterday

A(1) = data_db_yest(1);
T(1) = 0.5;
Z(1) = data_db_yest(1);
Z(2) = data_db_yest(2);

k =1;
L = size(data_db_yest,2);

% initialize Seasonality for day before yesterday (S_db_yest)
for i= 2:L-1

   %Update Level (A), Trend (T), and Seasonality S_yest--------------------
   A(i)         = x(1) * (data_db_yest(i) / S_dbb_yest{j}(i)) +...
                  (1 - x(1)) * (A(i-1) + T(i-1));

   T(i)         = x(2) * ( A(i) - A(i-1)) + (1 - x(2)) * T(i-1);

   S_db_yest(i) = x(3) * ( data_db_yest(i) / A(i)) +...
                  (1 - x(3)) * S_dbb_yest{j}(i);


   %Forecast k steps ahead
   Z(i + 1)         =  (A(i) + k*T(i)) * S_dbb_yest{j}(i + 1);

end
S_db_yest(i + 1) = 1;

% initialize Level (A), Trend (T), for first two timesteps for yesterday

A(1) = data_yest(1);
T(1) = 0.5;
Z2(1) = data_yest(1);
Z2(2) = data_yest(2);


k =1;
L = size(data_yest,2);
% do forecast for yesterday using initialized Seasonality (S_db_yest)------
```

```
for i= 2:L-1

   %Update Level (A), Trend (T), and Seasonality S_yest--------------------
   A(i)          = x(1) *(data_yest(i) / S_db_yest(i)) +...
                   (1 - x(1)) * (A(i-1) + T(i-1));

   T(i)          = x(2)  *(A(i) - A(i-1)) + (1 - x(2))  * T(i-1);

   S_yest{j}(i) = x(3) *(data_yest(i) / A(i)) + (1 - x(3)) * S_db_yest(i);


   %Forecast k steps ahead
   Z2(i + 1) =  (A(i) + k*T(i)) * S_db_yest(i + 1);

end
S_yest{j}(i + 1) = 1;

%Calculate the objective function: the Mean Absolute Error (MAE)
MAE = mean(abs(Z2 - data_yest));

%Save the search space of the parameter optimization
x_hist = [x_hist; x];
```

Published with MATLAB® 7.13

```matlab
function [Forecast] = updateHW_test(lastDataPoint, time, location, aggregation, h
%updateHW stores all t

global online;

%Identify detector--------------------------------------------------------
locationMapping = evalin('base','locationMapping');
j =  find(locationMapping == location);

%Add detection to database------------------------------------------------
Data_History = evalin('base','Data_History');
Data_History{j} = [Data_History{j} lastDataPoint];
headwayArrival = evalin('base','headwayArrival');
Forecast = 0;

%Do a forecast when a aggregation point is reached-------------------------
if mod(length(Data_History{j}),120) == 0 && (isempty(Data_History{j})== 0);

    %Load necessary data
    S_yest = evalin('base','S_yest');
    alpha = evalin('base','alpha');
    beta = evalin('base','beta');
    gamma = evalin('base','gamma');
    A = evalin('base','Level');
    T = evalin('base','Trend');
    Vissum_History = evalin('base','Vissum_History');
    Forecast_History = evalin('base','Forecast_History');

    %Offset data by a '1000'
    lastDataPoint = lastDataPoint + 1000;

    %Starting time during the day
    i = time;

    if i == 0; i = 1; end;

    %Update Level (A), Trend (T), Seasonality (S)--------------------------

    if i == 1 || i == 2
        A_new{j} = lastDataPoint;
        T_new{j} = 0;
    else

        A_new{j} = alpha{j} *( lastDataPoint / S_yest{j}(i) ) + (1 - alpha{j}) *

        T_new{j} = beta{j}  *( A_new{j} - A{j}       ) + (1 - beta{j})  * T{j};

        %S_new = gamma *( data.q(i) / A)       + (1 - gamma) * S_yest;
        %alleen nodig bij online systeem waar meerdere dagen worden gesimuleerd

    end
    %Forecast------------------------------------------------------------
        Forecast =  (A_new{j} + horizon*T_new{j}) * S_yest{j}(i+1);
        Forecast = Forecast - 1000;
        Forecast_History{j} = [Forecast_History{j} Forecast]; % voor analyse

    %Save all important parameters to a 'base' workspace-----------------------
    assignin('base','Level',A_new);
    assignin('base','Trend',T_new);
```

```
    headwayArrival = aggregation / Forecast;
    assignin('base','Forecast', Forecast);
    assignin('base','headwayArrival',headwayArrival);

    assignin('base','Vissum_History', Vissum_History);
    assignin('base','Forecast_History', Forecast_History);

end
assignin('base','Data_History',Data_History);
end
```

```
Error using updateHW_test (line 8)
Not enough input arguments.
```

Published with MATLAB® 7.13

```
%This is the main process for running an offline Artificial Neural Network simula
%the most important function in this script are initNAR: initializes the ANN
%and updateNAR: updates and makes a forecast for every second, the
%last section generates the necessary graphs for evaluation

clear all;

%online or offline simulation? (online = 1, offline = 0)-------------------
global online;
online = 0;

%Initialize Artificial Neural Networks algorithm-----------------------------------
delay = 10;
neurons = 15;
aggregation = 120;

initNARx_test('/Users/alexanderkrstulovic/Documents/MATLAB/Simulation Functions/F

%Load dataset-----------------------------------------------------------

load('det18 120 split.mat')
data.q  = q{1,2}.q(2:end);
data.t  = q{1,2}.t(2:end);

%Simulate forecast for dataset---------------------------------------------

for i = 1:1:length(data.q)-1;
    [Forecast] = updateNARx_test(data.q(i),i, 18, aggregation, delay);
end

%Evaluation---------------------------------------------------------------
subplot(2,1,1);
plot(data.t,Forecast_History{1,1});
plotDateStr(data.t, 60) ;
legend('data','forecast');
title('ANN forecast for Malledijk')
ylabel('Amount of vehicles')
xlabel('Time')

subplot(2,1,2);
error = Data_History{1,1} - Forecast_History{1,1}(1:end-1);
plot(data.t(1:end-1),error);
plotDateStr(data.t, 60) ;
title('ANN error for Malledijk')
ylabel('Amount of vehicles')
xlabel('Time')

linkaxes;

%Calculate RMSE-----------------------------------------------------------
rmse = sqrt(mean(error.^2));
```

```
data_db_yest =

    q: [719x1 double]
```

```
function [] = initNARx_test(folder, fileName_yest, fileName_db_yest, location, ag
%NEURALNETWORK_1_DET Summary of this function goes here
%This function initializes The Artificial Neural Network and can be called online
%This function only needs to be called once
%Holt winters is initialized for each individual detector "

global online;
locationMapping = [];

%Initialze ANN for all the required detectors---------------------

for j = 1:length(location)

%load data depending on online or offline simulation-----------------------
if online == 1
    %locationMapping = [locationMapping location{j}(1)];
    locationMapping = [locationMapping location(j)];

    [data_yest.t data_yest.q] = readMESdata_test(folder, fileName_yest, locationM
    [data_db_yest.t data_db_yest.q] = readMESdata_test(folder, fileName_db_yest, 

    %Aggregate
    if diff(data_yest.t(1:2)) ~= aggregation;
    [data_yest.t  data_yest.q] = doQ_test(data_yest, aggregation);
    [data_db_yest.t data_db_yest.q] = doQ_test(data_db_yest,aggregation);
    end
else
    locationMapping = [locationMapping location(j)];
    load('det18 120 split.mat')

     data_db_yest.q  = q{1,3}.q(2:end) % wednesday
     data_db_yest.t  = q{1,3}.t(2:end) % wednesday
     data_yest.q     = q{1,4}.q(2:end) % thursday
     data_yest.t     = q{1,4}.t(2:end) % thursday
end

%Put data in correct format
[m n] = size(data_db_yest.q);
if m > n; data_db_yest.q = data_db_yest.q'; end;
[m n] = size(data_db_yest.t);
if m > n; data_db_yest.t = data_db_yest.t'; end;
[m n] = size(data_yest.q);
if m > n; data_yest.q = data_yest.q';end;
[m n] = size(data_yest.t);
if m > n; data_yest.t = data_yest.t';end;

Yesterday{j} = [data_yest];
%Offset data by arbitrary amount of 1000---------------------------------
data_yest.q = data_yest.q + 1000;
data_db_yest.q = data_db_yest.q + 1000;


%Normalize data-------------------------------------------------------------
normalize{j} = [norm(data_db_yest.q)];
normalize{j} =1;
input_data = data_db_yest.q/normalize{j}; %independent exogenous variable
target_data = data_yest.q/normalize{j};

%The ANN requires the data in cell format if the data is a timeseries
```

```
input_data = (num2cell(input_data)); % static omdat tijd wel een rol speelt en
target_data = (num2cell(target_data));

% Network Creation---------------------------------------------------------

%net = CreateNarx(1:delay,1:delay,1,neurons)
net = narxnet(1:delay,1:delay,neurons);
net.layers{1}.transferFcn = 'logsig';
net.performFcn = 'mae';
%net.trainParam.showWindow = false;
%net.trainParam.showCommandLine = false;
%net.initFcn = 'initlay';
%net.layers{1}.initFcn = 'initwb';
%net.inputWeights{:,:}.initFcn = 'initzero';
%net.layerWeights{:,:}.initFcn =  'initzero';
%net.biases{:}.initFcn = 'initzero';
net = init(net);   %werkt dit wel goed?!?!
%mse

%Training the network-----------------------------------------------------
[Xs,Xi,Ai,Ts] = preparets(net,input_data,{},target_data);
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
net = trainlm(net,Xs,Ts,Xi,Ai);

net_all{j} = [net];

Data_History{j} = [];
Vissum_History{j} = [];
Forecast_History{j} = [0];
headwayArrival{j} = [120];
end

%Save all important parameters to a 'base' workspace-----------------------
assignin('base','Pf', []);
assignin('base','Af', []);
assignin('base','wb', []);
assignin('base','net', net);
assignin('base','normalize',normalize);
assignin('base','Yesterday',Yesterday);

assignin('base','locationMapping',locationMapping);
assignin('base','Forecast_History',Forecast_History);
assignin('base','Data_History',Data_History);
assignin('base','Vissum_History',Vissum_History);
assignin('base','headwayArrival',headwayArrival);
```

```
Error using initNARx_test (line 12)
Not enough input arguments.
```

```matlab
function [Forecast] = updateNARx_test(lastDataPoint,time,location, aggregation, d
%NEURALNETWORK_1_DET Summary of this function goes here
%   Detailed explanation goes here

global online

%Identify detector-----------------------------------------------------------
locationMapping = evalin('base','locationMapping');
j =  find(locationMapping == location);

%Add detection to database---------------------------------------------------
Data_History = evalin('base','Data_History');
Data_History{j} = [Data_History{j} lastDataPoint];
headwayArrival = evalin('base','headwayArrival');
wb = evalin('base','wb');
Forecast = 0;

%Do a forecast when a aggregation point is reached-------------------------

if mod(length(Data_History{j}),120) == 0 && (isempty(Data_History{j})== 0);

        normalize = evalin('base','normalize');
        Forecast_History = evalin('base','Forecast_History');

        %Start forecasting when there are enough input datapoints
        if length(Data_History{j}) > delay

            net = evalin('base','net');

            %Prepare data---------------------------------------------------------

            Yesterday = evalin('base','Yesterday');
            input_data = Yesterday{j};


            input_data = input_data.q + 1000;
            target_data = Data_History{j} + 1000;

            input_data = input_data/normalize{j};
            target_data = target_data/normalize{j};

            input_data = (num2cell(input_data));
            target_data = (num2cell(target_data));

            input_data = input_data(length(target_data)-delay+1:length(target_dat
            target_data = [target_data(end-delay +1 :end) nan];
            %[Xs,Xi,Ai,Ts] = preparets(net,input_data,{},target_data);

            Xs = [input_data(end);target_data(end)];
            Xi = [input_data(1:end-1);target_data(1:end-1)];

            %Do ANN forecast----------------------------------------------
            Forecast = net(Xs,Xi);

            Forecast = cell2mat(Forecast) - 1000;
            if Forecast < 0; Forecast = 0; end;


            %---------------------------------------------------------------------
```

```
        end

        %Save all important parameters to a 'base' workspace---------------------
        assignin('base','Forecast', Forecast);
        headwayArrival{j} = [aggregation / Forecast];
        Forecast_History{j} = [Forecast_History{j} Forecast];
        assignin('base','Forecast_History', Forecast_History);



end
assignin('base','headwayArrival',headwayArrival);
headwayArrival = headwayArrival{j};
assignin('base','Data_History',Data_History);
assignin('base','wb',wb);
```

```
Error using updateNARx_test (line 9)
Not enough input arguments.
```

Published with MATLAB® 7.13

```
%function [RMSE] = doTestrun_platChar(neurons, input_delay,...
    feedback_delay, horizon, treshold, location)
%This is the main process for an offline platoon characteristic forecasting
%simulation. The main functions are initNARx which initializes and trains
%a Neural network for each platoon characteristic. The function doPlat_char
%is called every second to make a forecast. The last section evaluates the
%forecast and construct a topview plot of vehicle arrivals (plotFlat)

global online
global min_platoon_size;
online = 0;

%global parameters
 neurons = 10;
 input_delay = 5;
 feedback_delay = 5;
 horizon = 5;
 treshold = 4;
 location = 2;
 min_platoon_size = 2;

%Initialize Platoon characteristics algorithm------------------------------
initNARx_multistep_test(...
    '/Users/alexanderkrstulovic/Documents/MATLAB/341_equal',...
    'AssenNoord_adaptive_greenwave', 'AssenNoord_adaptive_greenwave',...
    location, neurons, input_delay, feedback_delay, treshold)

%Load dataset------------------------------------------------------------
if online == 1
    [data.t data.q] = readMESdata_test(...
        '/Users/alexanderkrstulovic/Documents/MATLAB/245',...
        'AssenNoord_adaptive_greenwave11', location);
else
    uur = 13;
    duur = 1;
    load ('q_cam_1s_split.mat')
    %load ('q_det_1s_split.mat')
    data.q = q{1,5}.q(uur*3600+1:(uur+duur)*3600);
    data.t = q{1,5}.t(uur*3600+1:(uur+duur)*3600);
end

[m n] = size(data.q);
if m > n; data.q = data.q';end;
[m n] = size(data.t);
if m > n; data.t = data.t';end;

%Extract platoon characteristics for validation set
[q_trans_val headway_val] = Untangle_test(validation,nan, 0);
[platChar_val.IP platChar_val.PS platChar_val.PV platChar_val.t_A...
    t_B_val] = getPlatChar_test(headway_val, validation, nan, treshold);

%Simulate forecast for dataset-------------------------------------------
for i = 1:1:length(data.q)-500;
    disp(length(data.q)-i);
    [Forecast newForecast] = doPlatChar_test(data.q(i),data.t(i),...
        location, input_delay, feedback_delay, horizon, treshold);
end
```

```
Forecast_History = evalin('base','Forecast_History');
platChar_History = evalin('base','platChar_History');

 Forecast_final.q = [];
 Forecast_final.t = [];
 for i = 2:length(Forecast_History{1})-1
     if i ~= 2;Forecast_final = checkDoubles(Forecast_final,...
             Forecast_History{1}(i,1)); end;
     Forecast_final.q = [Forecast_final.q Forecast_History{1}(i,1).q];
     Forecast_final.t = [Forecast_final.t Forecast_History{1}(i,1).t];
 end


 %Evaluation-------------------------------------------------------------
 [Forecast_final.t IX] = sort(Forecast_final.t);
 Forecast_final.q = Forecast_final.q(IX);
 Forecast_final.t = Forecast_final.t + (94/(24*3600));

 start = find(datenum_round_off(validation.t, 'second') ==...
     datenum_round_off(Forecast_final.t(1,1), 'second'));
 close = find(datenum_round_off(validation.t, 'second') ==...
     datenum_round_off(Forecast_final.t(1,end), 'second'));
 validation.t = validation.t(start-1:close+1);
 validation.q = validation.q(start-1:close+1);

 figure(1);
 % plot vehicle arrivals with a top view
 plotFlat_test(validation,10,0,10,1,1)
 plotFlat_test(Forecast_final,10,-1,10,1,0)

 % identify the correct, false and mis- forecasted vehicles
 [kpi1 kpi2 kpi3 sumKPI] = doKpi(validation, Forecast_final,2)

 plotFlat_test(kpi1,10,-2,10,1,0) % wel aanwezig wel geforecast
 plotFlat_test(kpi3,10,-3,10,1,0) % wel geforecast, niet aanwezig
 plotFlat_test(kpi2,10,-4,10,1,0) % wel aanwezig niet geforecast
 plotFlat_test(kpi2,10,-5,10,1,0) % wel aanwezig niet geforecast


[id id2] = ismember(platChar_val.t_A, platChar_History{1,1}.t_A);

 % plot the progression of the individual platoon characteristics
 figure(2);
 plot(platChar_val.t_A(id),platChar_val.IP(id));
 hold all;
 plot(platChar_History{1,1}.t_A(id2(id))',platChar_History{1,1}.IP(id2(id)));
 title('Evaluation of Inter Platoon Forecast (x = 5, n = 5)');
 legend('Validation signal','Forecasted signal');
 ylabel('Seconds');
 %xlabel('Time');
 plotDateStr(platChar_val.t_A(id), 6);
  h = gca;
 rotateticklabel(h,20)

 figure(3);
 plot(platChar_val.t_A(id), platChar_val.PS(id));
 hold all;
 plot(platChar_History{1,1}.t_A(id2(id))',platChar_History{1,1}.PS(id2(id)));
 title('Evaluation of Platoon Size Forecast (x = 5, n = 5)');
 legend('Validation signal','Forecasted signal');
```

```
 ylabel('Seconds');
 %xlabel('Time');
 plotDateStr(platChar_val.t_A(id), 6);
  h = gca;
 rotateticklabel(h,20)

 figure(4);
 plot(platChar_val.t_A(id),platChar_val.PV(id));
 hold all;
 plot(platChar_History{1,1}.t_A(id2(id)),platChar_History{1,1}.PV(id2(id)))
 title('Evaluation of Platoon Volume Forecast (x = 5, n= 5)');
 legend('Validation signal','Forecasted signal');
 ylabel('Seconds');
 %xlabel('Time');
 plotDateStr(platChar_val.t_A(id), 6);
  h = gca;
 rotateticklabel(h,20)


RMSE.IP = doRMSE(platChar_val.IP(id), platChar_History{1,1}.IP(id2(id)));
RMSE.PS = doRMSE(platChar_val.PS(id), platChar_History{1,1}.PS(id2(id)));
RMSE.PV = doRMSE(platChar_val.PV(id), platChar_History{1,1}.PV(id2(id)));
```

```
Error: File: /Users/alexanderkrstulovic/Documents/MATLAB/Simulation Functions/Tes
Unbalanced or unexpected parenthesis or bracket.
```

```matlab
function [] = initNARx__multistep_test(folder, fileName_yest,...
fileName_db_yest, location, neurons, input_delay, feedback_delay, treshold)
%This function initializes the platoon characteristic algorithm by
%extracting the extracting the characteristics from the data and
%consequently train a network.

% TODO
%
%
global online

locationMapping = [];
time = nan;

for j = 1:length(location)

    %All parameters will be saved according to its mapped location "j"
    if online == 0
    locationMapping = [locationMapping location(j)];        %for offline use
    else
    %locationMapping = [locationMapping location{j}(1)];    %for online use
    locationMapping = [locationMapping location(j)]
    end

    %Read in past data to use for network training


    load ('q_cam_1s_split.mat')
    %load ('q_det_1s_split.mat')
    uur = 1
    duur = 23;
    data_db_yest.q = q{1,3}.q(uur*3600+1:(uur+duur)*3600);
    data_db_yest.t = q{1,3}.t(uur*3600+1:(uur+duur)*3600);
    data_yest.q = q{1,4}.q(uur*3600+1:(uur+duur)*3600);
    data_yest.t = q{1,4}.t(uur*3600+1:(uur+duur)*3600);

%     data_yest.q = q_det{1,2}.q(uur*3600+1:(uur+duur)*3600);
%     data_yest.t = q_det{1,2}.t(uur*3600+1:(uur+duur)*3600);
%     data_yest.t = data_yest.t - (94/(24*3600));


    %put data in correct format
    [m n] = size(data_db_yest.q);
    if m > n; data_db_yest.q = data_db_yest.q'; end;
    [m n] = size(data_db_yest.t);
    if m > n; data_db_yest.t = data_db_yest.t'; end;
    [m n] = size(data_yest.q);
    if m > n; data_yest.q = data_yest.q';end;
    [m n] = size(data_yest.t);
    if m > n; data_yest.t = data_yest.t';end;

    %filter signal
    [input] = cutSingles_test(data_db_yest,time, treshold);
    [target] = cutSingles_test(data_yest,time, treshold);

    %extract the headway from the original signal mapped to time
    [q_trans_input headway_input] = Untangle_test(input,time, 0);
    [q_trans_target headway_target] = Untangle_test(target,time, 0);
```

```
    %from headway signal extract the platoon characteristcs

    [interplatoon_input platoonsize_input platoonvolume_input...
        t_A_input t_B_input] = getPlatChar_test(headway_input,...
        input,time, treshold);
    [interplatoon_target platoonsize_target platoonvolume_target...
        t_A_target t_B_target] = getPlatChar_test(headway_target,...
        target,time, treshold);

    %train each characteristic separately
   for i=1:3
        display([length(location)-j 3-i]);

        if i == 1;
        input_data =  interplatoon_input;
        target_data = interplatoon_target;
        end;
        if i == 2;
        input_data = platoonsize_input;
        target_data = platoonsize_target;
        end
        if i == 3;
        input_data = platoonvolume_input;
        target_data = platoonvolume_target;
        end


        Yesterday{j}{i} = target_data;


        %make input & target equal in size
        k = min([length(input_data), length(target_data)]);
        input_data = input_data(end-k+1:end);
        target_data = target_data(end-k+1:end);
        %plot([input_data' target_data'])

        %normalize data
        normalize{j}{i} = [norm(input_data)];
        normalize{j}{i} = 1;
        input_data = input_data/normalize{j}{i}; %independent exogenous variable
        target_data = target_data/normalize{j}{i};

        %convert data to cell format for so-called 'dynamic' training (see Matlab
        input_data= (num2cell(input_data)); % static omdat tijd wel een rol speel
        target_data = (num2cell(target_data));

        %network Creation

        net = narxnet(1:input_delay,1:feedback_delay, neurons);
        net.layers{1}.transferFcn = 'logsig';
        %net.trainParam.showWindow = false;
        %net.trainParam.showCommandLine = false;
        %net.initFcn = 'initlay';
        %net.layers{1}.initFcn = 'initwb';
        %net.inputWeights{:,:}.initFcn = 'initzero';
        %net.layerWeights{:,:}.initFcn =  'initzero';
        %net.biases{:}.initFcn = 'initzero';
        net = init(net);   %werkt dit wel goed?!?!
        %mse
```

```
        %Training the network

        [Xs,Xi,Ai,Ts] = preparets(net,input_data,{},target_data);
        net.divideParam.trainRatio = 70/100;
        net.divideParam.valRatio = 15/100;
        net.divideParam.testRatio = 15/100;
        net = trainlm(net,Xs,Ts,Xi,Ai);

        %save weights and biases
        weights{j}{i} = getwb(net);
        assignin('base',['net_' mat2str(j) '_' mat2str(i)], net);
    end

%Save all important data
Data_History{j}.q = [];
Data_History{j}.t = [];
Vissim_History{j} = [];
Forecast_History{j} = [];
platChar_History{j}.t_A = [];
platChar_History{j}.IP = [];
platChar_History{j}.PS = [];
platChar_History{j}.PV = [];
headwayArrival{j} = [120];
old{j}=[99999; 999999; 999999];
end

assignin('base','old',old);
assignin('base','normalize',normalize);
assignin('base','Yesterday',Yesterday);

assignin('base','locationMapping',locationMapping);
assignin('base','Vissum_History',Vissim_History);
assignin('base','Forecast_History',Forecast_History);
assignin('base','platChar_History',platChar_History);
assignin('base','Data_History',Data_History);
assignin('base','headwayArrival',headwayArrival);
```

```
Error using initNARx_multistep_test (line 15)
Not enough input arguments.
```

```
function [transformed_data1 transformed_data2] =...
    Untangle_test(original_data,time, level)
%this function untangles the original signal in a signal that displays the
%headway (transformed_data2) and the intensity(transformed_data1) both are
%mapped to the time the event occured.


%   TODO
%
%
global online;

try
    idx = find(original_data.q > level);
    firstentry = idx(1)-1;
    %lastentry = size(original_data.q,2) - idx(end);
    if online == 1
    lastentry = time - idx(end);
    else
    %lastentry = time - (idx(end)/(24*3600));
    lastentry = (time - original_data.t(idx(end)))*(24*3600);
    end
    delta = diff(idx) -1;
    transformed_data2.q =[firstentry delta lastentry];
    idx = [1 idx];
    transformed_data2.t = original_data.t(idx);

    %plot(transformed_data2.t,transformed_data2.y)

    idx2 = find(original_data.q <= level);
    transformed_data1.q = original_data.q;
    transformed_data1.t = original_data.t;
    transformed_data1.q(idx2)= [];
    transformed_data1.t(idx2) = [];

    %plot(transformed_data1.t,tranformed_data1.y)

catch
    transformed_data1 = nan;
    transformed_data2 = nan;
end

end
```

```
ans =

   NaN
```

```matlab
function   [target_data_sample] = cutSingles_test(target_data_sample,time,...
                 treshold);
%This is the filter that removes vehicles based on their critical headway
%(treshold) and the minimum platoon size


global online
global min_platoon_size;

min_platoon_size = 2;

try
[q_trans_target headway_target] = Untangle_test(target_data_sample,time, 0);
id1 = find(headway_target.q > treshold);
catch
    target_data_sample = target_data_sample;
    return;
end

try
while any(diff(id1) <= min_platoon_size)

    id2 = find(diff(id1) <= min_platoon_size);

     if length(headway_target.t) - id1(end) < 5 &&...
            length(headway_target.t) - id1(end) ~= 0;

         endOfWindow = find(ismember(target_data_sample.t,...
            headway_target.t(id1(end)+1:end)));

         target_data_sample.q(endOfWindow) = 0;
     end

    %Use if time vector records date and time in 'datenum' (offline purposes)
    if online == 0;
    singles = headway_target.t(id1(id2)) +...
        ((headway_target.q(id1(id2))+1)/(24*3600));
    else
    %Use if time vector records time in seconds (online simulation purposes)
    singles = headway_target.t(id1(id2))' +...
        ((headway_target.q(id1(id2))))';
    end

    %OPTIONAL for verification purpose
%     subplot(2,1,1);
%     plot(target_data_sample.t,target_data_sample.q);
%     hold all;
%     stem(singles,ones(length(singles)));
%     plotDateStr(target_data_sample.t,600);


    id4 = [];
    for i = 1:length(singles)
        id3 = find(target_data_sample.t <= (singles(i)));
        id4 = [id4 id3(end)];
    end

    %remove singles
    target_data_sample.q([id4 (id4 -1) (id4 + 1)]) = 0;
```

```
    %OPTIONAL for verification purposes
%     subplot(2,1,2);
%     plot(target_data_sample.t,target_data_sample.q);
%     plotDateStr(target_data_sample.t,600);
%     linkaxes;

    [q_trans_target headway_target] =...
        Untangle_test(target_data_sample,time, 0);
    id1 = find(headway_target.q > treshold);
end
catch
    target_data_sample = nan;
    return;
end

if isempty(diff(id1)) == 1; target_data_sample = target_data_sample; end;
```

```
Error using cutSingles_test (line 16)
Not enough input arguments.
```

```
function [interplatoon platoonlength platoonvolume t_A t_B] = getPlatChar_test(he
%This function extract the platoon characteristics from the signal that
%depicted the headway. t_A corresponds to t_Start in the paper and t_B to
%t_end.
%TODO
%
%
%
%
global online
%OPTIONAL for verification purposes only
% edges = 1:1:max(headway_input.q);
% [n] = histc(headway_input.q, edges);
% bar(edges, n);

try

    idx = find(headway_input.q > treshold);

    t_B = headway_input.t(idx);
    newPlatoonHasArrived = 1;

    if online == 0
    t_A = t_B + headway_input.q(idx)./(24*3600) + 1/(24*3600);   %For offline use
    %if time  <= t_A(end)+ 1/(24*3600); t_A(end) = [];newPlatoonHasArrived = 0; e
    else
    t_A = t_B + headway_input.q(idx) + 1;                  %For online use
    %if time == t_A(end) - 1; t_A(end) = []; newPlatoonHasArrived = 0; end;
    end;
    t_A = t_A(1:end-1);

    %OPTIONAL for verification purposes only
%        plot(input.t,input.q)
%        hold all;
%        stem(t_B(2:end),ones(length(t_B(2:end))));
%        hold all;
%        stem(t_A,ones(length(t_A)));

    if online == 0
    %For offline use
     interplatoon = diff(t_A)*(24*3600);
     platoonlength = (t_B(2:end) - t_A)*(24*3600);
     platoonvolume = doVolume_test(input, t_A, t_B(2:end));

    else
     interplatoon = diff(t_A);
     platoonlength = (t_B(2:end) - t_A);
     platoonvolume = doVolume_test(input, t_A, t_B(2:end));

    end

    %OPTIONAL for verification purposes only
%        figure(2);
%        plot(interplatoon);
%        figure(3);
%        plot(platoonlength);
%        figure(4);
%        plot(platoonvolume);
```

```
catch
    interplatoon = nan;
    platoonlength = nan;
    platoonvolume = nan;
    t_A = nan;
    t_B = nan;
end

end
```

```
ans =

   NaN
```

Published with MATLAB® 7.13

```matlab
function [Forecast newForecast] = doPlatChar_test(lastDataPoint, time, location,


%TODO
% mismatch in de tijd van input en target signaal
%
%
%
%

newForecast = 0;
locationMapping = evalin('base','locationMapping');
j =  find(locationMapping == location);


Data_History = evalin('base','Data_History');
Forecast_History = evalin('base','Forecast_History');
platChar_History = evalin('base','platChar_History');


Data_History{j}.q = [Data_History{j}.q lastDataPoint];
Data_History{j}.t = [Data_History{j}.t time];

headwayArrival = evalin('base','headwayArrival');
Forecast.q = 0;
Forecast.t = time;

target_sample = Data_History{j};


%put data in correct format
[m n] = size(target_sample);
if m > n; target_sample = target_sample'; end;

%cut out single arrivals that can distort platoon forecasting
[target_sample] = cutSingles_test(target_sample,time, treshold);

%extract from the original signal headway signal mapped to time
[q_trans_target headway_target] = Untangle_test(target_sample,time, 0);

%from headway signal extract the platoon characteristcs
[interplatoon platoonsize platoonvolume t_A t_B] =...
    getPlatChar_test(headway_target, target_sample,time, treshold);


if length(interplatoon) > input_delay && (isempty(Data_History{j})== 0);

    normalize = evalin('base','normalize');
    %weights = evalin('base', 'weights');
    Yesterday = evalin('base','Yesterday');

    for i = 1:3

        %select the right input and target sample to train weights
        if i == 1;
        target_sample = interplatoon;
        input_sample = Yesterday{j}{1}(length(target_sample)-...
            input_delay+1:length(target_sample) + horizon);
        end;
```

```matlab
    if i == 2;
    target_sample = platoonsize;
    input_sample = Yesterday{j}{2}(length(target_sample)-...
        input_delay+1:length(target_sample) + horizon);
    end
    if i == 3;
    target_sample = platoonvolume;
    input_sample = Yesterday{j}{3}(length(target_sample)-...
        input_delay+1:length(target_sample) + horizon);
    end

    net = evalin('base',['net_' mat2str(j) '_' mat2str(i)]);

    %Update & Forecast-----------------------------------------------------

    input_sample = input_sample/normalize{j}{i};
    target_sample = target_sample/normalize{j}{i};

    target_sample = [target_sample(end-input_delay +1 :end) nan(1,horizon)];

    input_sample = (num2cell(input_sample));
    target_sample = (num2cell(target_sample));

        netc = closeloop(net);
        [Xs,Xi,Ai,Ts] = preparets(netc,input_sample,{},target_sample);
        Forecast = netc(Xs,Xi,Ai);

        Forecast = cell2mat(Forecast);
        Forecast(Forecast < 0) = 0;

        if i == 1; interplatoon_pred = round(Forecast); end;
        if i == 2; platoonsize_pred = round(Forecast); end;
        if i == 3; platoonvolume_pred = round(Forecast); end;


    %end
end

old = evalin('base','old');
if any(interplatoon_pred ~= old{j}(1,:))==1 && any(platoonsize_pred ~=....
        old{j}(2,:))==1 && any(platoonvolume_pred ~= old{j}(3,:))==1;
        newForecast = 1;
end;
old{j} = [interplatoon_pred; platoonsize_pred; platoonvolume_pred];
assignin('base','old', old);


% construct forecast back to an overall forecast on second by second
% basis
[Forecast] = getPlatChar_i_test(interplatoon_pred, platoonsize_pred,...
    platoonvolume_pred, t_A(end));

%for evaluation purpose
if newForecast == 1;
platChar_History{j}.t_A = [platChar_History{j}.t_A; t_A(end)];
platChar_History{j}.IP = [platChar_History{j}.IP; interplatoon_pred];
platChar_History{j}.PS = [platChar_History{j}.PS; platoonsize_pred];
platChar_History{j}.PV = [platChar_History{j}.PV; platoonvolume_pred];
end;
```

```
    % remove all forecast greater than horizon and smaller than t_now
    removePast = find(Forecast.t <= time);
    if isempty(removePast); removePast = 1;end;
    removeFuture = find(Forecast.t >= time + 120);
    if isempty(removeFuture);
        Forecast.q = Forecast.q(removePast(end):end);
        Forecast.t = Forecast.t(removePast(end):end);
    else
    Forecast.q = Forecast.q(removePast(end):removeFuture(1));
    Forecast.t = Forecast.t(removePast(end):removeFuture(1));
    end;

    if length(Forecast.q) > 1;
    if newForecast == 1; Forecast_History{j} = [Forecast_History{j}; Forecast];en
    end;
end

assignin('base','Forecast', Forecast);
%headwayArrival{j} = diff(Forecast);
%assignin('base','headwayArrival',headwayArrival);
%headwayArrival = headwayArrival{j};

%Forecast_History{j} = [Forecast_History{j} Forecast];
assignin('base','Forecast_History', Forecast_History);
assignin('base','platChar_History', platChar_History);



assignin('base','Data_History',Data_History);
```

```
Error using evalin
Undefined function or variable 'locationMapping'.

Error in doPlatChar_test (line 12)
locationMapping = evalin('base','locationMapping');
```

```matlab
function [q_pred] = getPlatChar_i_test(interplatoon_pred,...
    platoonsize_pred, platoonvolume_pred, t_A);
%This function constructs an overall forecast on a second by second basis
%using the forecasts for IP, PS and PV and propogates the forecast based
%on the startpoint t_A of the last completed platoon.

% TODO
%
%   - unirnd vervangen voor exponentiele verdeling
%
%   - altijd zelfde verdeling gebruiken?
global online;

q_pred.q = [];
q_pred.t = [];
offset = 0;

for i=2:length(interplatoon_pred);
    interplatoon_pred(i) = interplatoon_pred(i) + interplatoon_pred(i-1);
end
test = [];

for i = 1:length(interplatoon_pred);
hw = platoonsize_pred(i)/platoonvolume_pred(i);
hw_vector = 0;
for j = 1:platoonvolume_pred(i)
    hw_vector = [hw_vector hw_vector(end)+hw];
end
test{i} = [(interplatoon_pred(i) + hw_vector)];
end


for i=1:length(test);
    q_pred.t = [ q_pred.t test{i}];
end
q_pred.t = sort(q_pred.t);

edges = q_pred.t(1):1:q_pred.t(end);
q_pred.q = histc(q_pred.t,edges);

if online == 1
q_pred.t = t_A + ((edges + offset));
else
q_pred.t = t_A + ((edges./(24*3600) + offset));
end
```

```
Error using getPlatChar_i_test (line 18)
Not enough input arguments.
```

```
function [kpi_1 kpi_2 kpi_3 sumKPI] = doKpi(validation,prediction, treshold);
%This function identifies the vehicles that were predicted correct, false
%or were not forecasted at all. The treshold value (in seconds) indicates
%the bandwith that accepts forecast that not were not perfect on target
%TODO
% -
% -
% -
% -

global online
idx3 = [];
idx4 = [];

idx1 = find(validation.q ~= 0);
idx2 = find(prediction.q ~= 0);
sumKPI.v0 = length(idx1);

step =1;
if online ==0;
    prediction.t = datenum_round_off(prediction.t, 'second')
    validation.t = datenum_round_off(validation.t, 'second')
    treshold = treshold/(24*3600);
    step = step/(10 *24*3600);
end

for i = 0:step:treshold
   try
    [idx1_1a, idx1_1b] = ismember(validation.t(idx1),...
        [(prediction.t(idx2)-i) ]);

    idx2(idx1_1b(logical(idx1_1a))) = [];
    idx3    = [idx3 idx1(logical(idx1_1a))];
    [idx2_1a, idx2_1b] = ismember(validation.t(idx1),...
        [(prediction.t(idx2)+i) ]);

    try
    if i~=0; idx2(idx2_1b(logical(idx2_1a))) = [];end;

    idx3    = [idx3 idx1(logical(idx2_1a))];
    catch
        display('fout in KPI')
    end

    idx1(logical(idx1_1a)) = [];
    idx1(logical(idx2_1a)) = [];

   catch
       display('fout in KPI2')
   end
end
idx3 = sort(idx3);

kpi_1 = doStretch(idx3, validation); % correct forecast
kpi_2 = doStretch(idx1, validation); % not forecasted
kpi_3 = doStretch(idx2, prediction); % false forecast
sumKPI.v1 = length(idx3);
sumKPI.v2 = length(idx1);
sumKPI.v3 = length(idx2);
```

```matlab
function out = doStretch(pointer, in)
% stukje code om alle tussenliggende seconden toe te voegen
%global online;

try
    if online == 0
        step =0.1/(24*3600);
    else
        step = 1
    end
    out.t = in.t(pointer(1))-step:step:in.t(pointer(end))+step;
    out.q = zeros(1,length(out.t));
    id = histc(in.t(pointer), out.t);
    while max(id) > 1
        id2 = find(id>1);
        id(id2 +1) = id(id2 +1) + 1;
        id(id2)= id(id2)-1;
    end
    id3 = find(id~= 0);
    out.q(id3) = in.q(pointer);
catch
    out = pointer;
end
end
end
```

```
Error using doKpi (line 15)
Not enough input arguments.
```

```matlab
function [] = plotFlat_test( data, tail, positionvector, width,Stretch,...
    doDatestr )
% This function plots vehicle arrivals with a top view
%
% TODO
% -
% -
% -
% -
global online;

if Stretch == 1; data = doStretch2(data); end

[m n] = size(data.q);
if m < n; data.q = data.q'; end;
[m n] = size(data.t);
if m < n; data.t = data.t'; end;


data.q = [data.q; zeros(tail,1)];
%xtraTime = 1:(data.t(2)-data.t(1)):tail;
if online == 1
xtraTime = (1:1:tail)/(10);
else
xtraTime = (1:1:tail)/(10*24*3600);
end
data.t = [data.t; xtraTime' + data.t(end)];

load ('colormap.mat')
signal = [];

% give the vehicles a dissipating color tail to be able to spot them
idx = find(data.q ~= 0);
tailMap = [data.q];
for i = 1:tail
    id_next = 0.8.^i * data.q(idx);
    tailMap(idx+ i) = tailMap(idx + i) + id_next;

end

        obj.signal     = [tailMap, tailMap, nan(size(tailMap))]';
        %obj.signal     = [tailMap]';
        obj.time       = [data.t, [data.t(2:end); data.t(end)]...
            nan(size(data.t))]';
        obj.signal     = obj.signal(:);
        obj.time       = obj.time(:);

        obj.signal(obj.signal > 3) = 3;
        obj.signal = round(obj.signal .*10);
        obj.signal(isnan(obj.signal)) = 1;
        obj.signal(obj.signal == 0) = 1;


        time = obj.time;
        signal = obj.signal;

        h     = patch(time,ones(size(time))*positionvector,'b',...
            'linewidth',width);
```

```
        set(h,'FaceVertexCdata',colormap(signal,:),'EdgeColor','interp')
        if doDatestr == 1;plotDateStr(time,6000);end;
    %end

    h = [h(:)];
     hold on
%     plot(get(gca,'xlim'),repmat(max(positionvector),1,2)+5,'k')
%     plot(get(gca,'xlim'),repmat(min(positionvector),1,2)-5,'k')


end

function out = doStretch2(in)
% stukje code om alle tussenliggende seconden toe te voegen
global online
if online == 1
    step = 0.1
else
step =0.1/(24*3600);
end
out.t = in.t(1)-step:step:in.t(end)+step;
out.q = zeros(1,length(out.t));
id = histc(in.t, out.t);
while max(id) > 1
    id2 = find(id>1);
    id(id2 +1) = id(id2 +1) + 1;
    id(id2)= id(id2)-1;
end
id3 = find(id~= 0);
out.q(id3) = in.q;
end
```

```
Error using plotFlat_test (line 12)
Not enough input arguments.
```

Published with MATLAB® 7.13