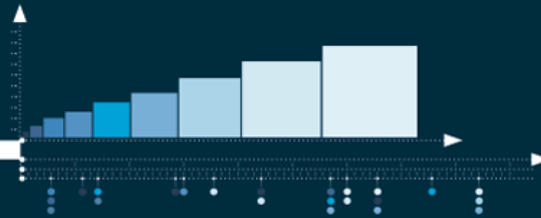


An Intelligent Traffic Flow Progression Model For Predictive Control Applications

A Proposed Approach for Making Short-Term Traffic Flow Predictions, Using A Recurrent, Multi-Task Learning Neural Network Model, to Transform Traffic Light Controllers from Adaptive to Predictive with Minimal Hardware Changes

by Noorhan Helmy - 4513843



Report completed in partial fulfillment of the requirements for the degree of
Master of Science
in Transport, Infrastructure and Logistics
at the Delft University of Technology

Chairman:

Prof. Dr. ir. Bart van Arem, OTG, Transport and Planning

Thesis Committee:

Dr. ir. Meng Wang, OTG, Transport and Planning

Dr. ir. Riender Happee, SIME, BioMedical Engineering and Cognitive Robotics

ing. Eddy Verhoeven, Siemens

John Vermeer M.Sc., Siemens

ACKNOWLEDGEMENTS

This thesis is the final deliverable for completing my Master of Science (M.Sc.) degree in Transport, infrastructure and Logistics at Delft University of Technology (TU Delft). Accordingly I would like to express my gratitude to all the people who have been vital to the completion of this project.

From my graduation committee, I would first like to thank the Chairman of the committee Prof. Bart van Arem who not only provided his valuable feedback in all the major meetings throughout the duration of this master thesis, but also had played an important role in shaping my master studies over the past two years. It was, in fact, his lectures during the Automated Vehicles course that first sparked my interest in the fields of automated vehicles and ITS, and his supervision, since then (on several projects), that has inspired me to delve into further research in these areas. Secondly, I would like to thank Dr. Riender Happee whose critical yet motivating feedback lead me to write a more a focused thesis paper, summary and presentation. I would also like to thank John Vermeer, whose unbelievable support over the last eight months has kept me highly motivated throughout, and has been vital to the successful completion of this project. Last, but not least, I would like to give a very special thank you to both Dr. Meng Wang and Eddy Verhoeven who I credit as being the two pillars of this project. During our weekly progress meetings Dr. Meng Wang not only kept me on track and focused, but he made sure to ask the difficult questions, challenge my methodology choices and be very critical of my work, which has consistently pushed me to work harder, has elevated the quality of my research and has nourished my critical thinking abilities. For the entire remainder of the week Eddy Verhoeven has been there, day in and day out, providing me with incredible support, making sure I have all the data and information I need, putting me in contact with all the right people, encouraging me to expand my scope, guiding me to develop my professional skills, and making sure to give me consistent feedback on my work. Moreover, his investment in building my ideas has lead to the completion of a well rounded concept, and his acknowledgement of the project's potential has kept me motivated to continue working on it until the very last minute.

In addition to my graduation committee, I also received a lot of aid from external members who have very generously taken the time to help me with some of the struggles I had during my thesis. Firstly, I would like to thank Dr. Maria Salomons for all the time and patience she spent helping me learn how to work with V-log data and answering all my questions regarding the VisSim COM interface. I would also like to thank Werner van Loo for taking the time to quickly answer all my questions on his V-log compiler CuteView and for compiling MV files for me when I could not find a program for it online. I would also like to give a special thanks to Wouter Moerland and Robin Huizenga who quickly provided me with a TU-Delft network license for VisSim as soon as they heard about my struggle with obtaining a VisSim license for the completion of my thesis. Moreover, I would like to thank Harm-Jan Mostert and Cyril Cappendijk for the time they spent providing me with all the data, AutoCAD drawings and configuration files I needed for my case study, and for taking the time to discuss my thesis topic and being open to considering its possible implementation on the street. Similarly, I would like to thank Peter Broekhuijsen for all the time and patience he spent helping me find a case study in Delft that served my purposes, and for taking the time to transfer large amounts of V-log data to me. Last, but certainly not least, I would like to thank Linda Lanphen and Jan Cees van Senden, who have really invested time and effort in helping me with restructuring and proof reading my report when I began to struggle with it.

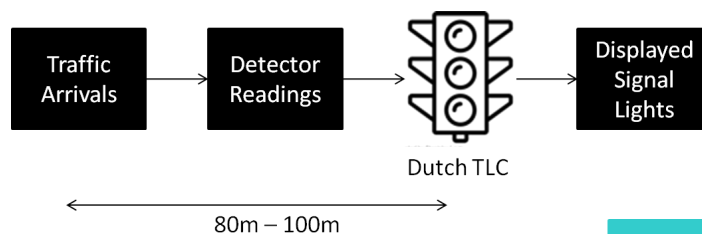
Furthermore, I am extremely thankful for my overall experience in Siemens. Being surrounded by the incredibly ambitious and talented people working on the Talking Traffic project has really encouraged me to come out of my comfort zone and explore working with different tools that were far out of my skill-set for my thesis. Moreover, everyone at the Siemens office is very welcoming and always willing to answer questions, which made the office environment extremely pleasant and inclusive.

Finally I would like to give an extremely special thanks to my incredible family and friends who have gotten me through all the stressful parts of the thesis, and whose undying support that travels through time, land and sea will always be the thing I am most thankful to have in my life.

This study explores the possibility of developing a short-term traffic flow prediction model that can be used to convert installed adaptive controllers to predictive controllers with minimal hardware changes. By using the prediction model's outputs to virtually trigger vehicle loop detectors, the outputs of an adaptive controller can be extracted in advance of actual vehicle arrivals. This will enable service providers to send out time to green/red (T2G/R) information or green light optimal speed advice (GLOSA), which are driver assistance use cases that aim to efficiently guide vehicles through intersections, in anticipation of known upcoming signal states. The main requirements for developing the prediction model are that it should be scalable to different intersection configurations, adaptable to different traffic conditions and should encompass the nonlinearity of traffic flow behavior. Since it fits these criteria, the developed model is a multi-task learning recurrent neural network with exogenous inputs (NARX), which is designed to match the traffic flow simulation abilities of a well adopted analytical traffic flow progression model. Both models were tested for a corridor in Delft that experiences almost consistent free flow conditions, and on an intersection in Haarlem with varying traffic conditions. For both case studies the neural network outperformed the analytical model. Most notably, it was better adaptable to long queues at intersections, had lower average error values, made fewer large errors, and better recognized the effect of a source/sink. When interfaced with an adaptive controller to test the predictive control methodology proposed, the superiority of the designed neural network model over the analytical model became more prominent. At no point did the neural network's prediction errors result in queue spillback, which was not true for the analytical model. However, the overall accuracy of the NARX model was still not yet satisfactory enough for practical application (especially for highly under-saturated traffic conditions) without the use of corrective measures. Nonetheless, due to its significant superiority over the widely used analytical model, with regards to both accuracy and adaptability, this model can be considered as a new starting point for traffic flow progression modeling.

The high level societal problem addressed in this study is that of travel time and fuel inefficiencies at urban intersections. Dutch traffic light controllers mitigate these inefficiencies by being highly adaptive to detected arrival and departure flows of traffic. This allows for real time optimization of traffic light phase cycle durations, minimizing global travel time and fuel losses at a signalized intersection. Nevertheless, the majority of travel time and fuel inefficiencies in an urban network remain a byproduct of red lights disturbing vehicle driving profiles. Green light optimal speed advice (GLOSA) and time to green/red (T2G/R) driver assistance use cases can be used to reduce the impact of this type of flow disturbance. These functionalities use infrastructure to vehicle (I2V) communication to inform road users of upcoming phase cycles enabling them to adapt their driving profiles accordingly. However, the mentioned adaptability of Dutch traffic light controller leads to fluctuating GLOSA and T2G/R messages, which are likely to be unsafe and cause user frustration. The underlying issue is that both approaches to traffic management are contradictory. The mentioned I2V communication solutions require foresight of upcoming schedules, while an adaptive controller’s foresight is limited to its detector inputs (at most 100m upstream the stop line). The challenge is then to increase the adaptive controller’s foresight to meet that of GLOSA and T2G/R use cases while maintaining its traffic management performance and its output behavior that drivers have grown accustomed to. The concept design proposed in Figure 1 aims to solve this problem.

Current Scenario:



Proposed Scenario:

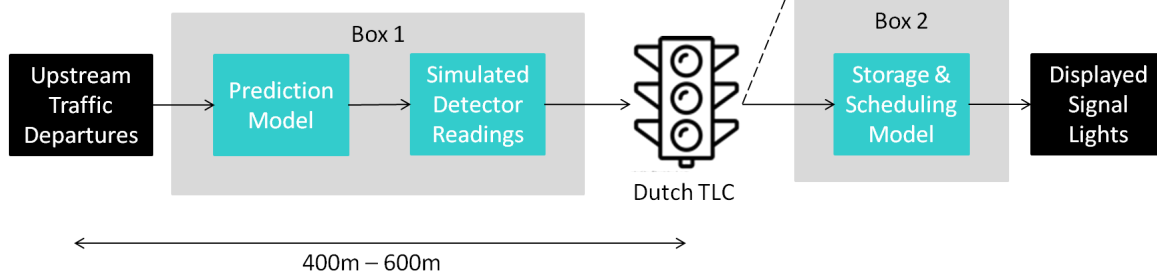


FIGURE 1: THE PROPOSED CONCEPT DESIGN, SHOWING HOW THE PROPOSED CHANGES TO THE CURRENT CONTROLLER ARE IN THE INPUT AND OUTPUT INFORMATION FLOW, WHILE THE CONTROL LOGIC REMAINS THE SAME

The objective of this design is to convert currently deployed controllers from adaptive to predictive. This can be achieved, as shown in Figure 1, by manipulating an adaptive controller’s inputs. By simulating vehicle detector readings, according to short term traffic flow arrival predictions, and feeding them into an adaptive controller, its outputs can be elicited early. The traffic flow prediction methodology proposed in this concept design is to measure platoon departures from upstream intersection stop line detectors, and use this information to estimate this same platoon’s arrival flow profile at the next signalized intersection on its path. Accordingly, it is coined as “short term traffic flow *progression* modeling”. Following this methodology, the prediction horizon is the average travel time between an upstream intersection and a downstream intersection, which can range from 20-35s, on average. By utilizing this prediction horizon, service providers will also have 20-35s foresight on upcoming traffic light controller outputs, allowing for more stable T2G/R and GLOSA messages. Similarly, the prediction horizon is also the lag time needed before outputting traffic light phase states on-street, to assure they are displayed at the correct time of actual vehicle arrivals. The flow of information can be visualized in Figure 2.

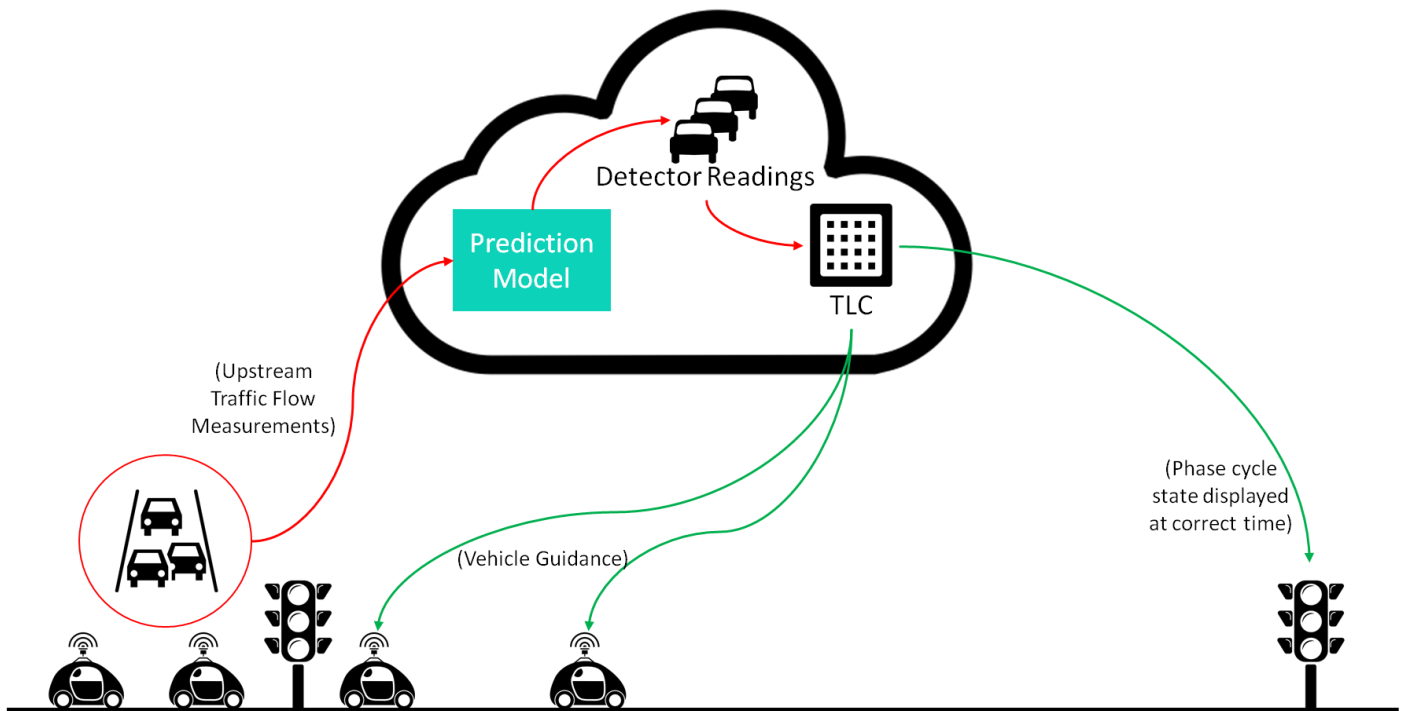


FIGURE 2: A VISUALIZATION FOR THE FLOW OF INFORMATION FOR PROPOSED TRAFFIC LIGHT MANAGEMENT METHODOLOGY

To facilitate the operation of a predictive controller proposal though, a model for short term traffic flow arrival predictions is needed, allowing the controller to continue its online optimization process as though the predicted vehicle arrivals are the actual ones. Accordingly, the focus of this report is on the design and evaluation of this model, both individually and when used to generate inputs for an adaptive controller. With that, the research question is formulated as follows:

To what extent can a short term traffic flow prediction model facilitate the conversion of a Dutch traffic actuated controller into a predictive controller, without compromising its performance?

While predictive controllers do exist in the market, inaccuracies in the short-term traffic flow arrival prediction models they rely on still remain a limitation to their performance, which they overcome through sophisticated verification and corrective measures for the model predictions. However, real time corrections to traffic light schedules compromise the predictability of these controllers. Moreover, online optimized predictive controllers rely on lane specific far away loop detectors to verify the model's prediction accuracy every time step; and, in cases where long queues cover these far away loops, online these controllers instate a system failure, and must resort to a fallback control methodology. Accordingly, a more robust prediction and validation methodology is needed.

Robertson's platoon dispersion model is identified as the main reference model, as much of the scientific papers found on short term traffic flow progression modeling use it as a starting point to improve upon. The relevance of this model is that it has been found to closely replicate the longitudinal progression of platoons, under free flow conditions, and was used in the development of many state of the art predictive controllers. However, it is unable to accommodate to sources and sinks, changing traffic conditions and long queues. Research on parameter calibrations for this model have targeted these limitations, although a critical tradeoff can be seen between parameter optimization and transferability to different locations. Other, more recently developed, analytical models do not tackle these limitations, but focus mainly on modeling the progression of traffic through a single, or multiple, signalized intersections. On the other hand, the nonlinearity of alternative machine learning models better addresses these limitations. However, to the extent of the knowledge of the author, no combined machine learning model, that collectively tackles all aspects of short term traffic progression, under complex traffic conditions, has been ventured yet. Nonetheless, this study concluded that neural networks algorithms have the highest potential to fill this gap. A neural network model does not comply to the law of conservation between inputs and outputs, allowing it to better account for the possible existence of sources and sinks; nor is it confined to a single probability distribution pattern, making it more adaptable to the different dispersion patterns resulting from varying traffic conditions, queue lengths and turning percentages.

The design methodology followed for this neural network traffic flow progression model is to begin with the simplest model architecture and iteratively add more features to the model until the optimal performance is reached. For that, three case studies, of incrementally varying complexities, are used: a synthetic case study that contains no stochasticity, a two lane link (i.e. a segment of Beatrixlaan, Delft) case study which contains traffic data under free flow conditions, and a full intersection case study that combines between two corridors with varying traffic conditions (i.e. N201/N205 intersection, Haarlem). With added complexity to the traffic datasets used, new corresponding features are added to the prediction model, and its iterative design approach continues. With each change made to the case study or model design, an analysis of the model’s prediction behavior is made. This research approach not only allows for the model design to contain the most relevant features, but additionally provides insights on the predictive abilities of a neural network with regards to different types of traffic conditions. The final model form was of a multi-task learning recurrent neural network with exogenous inputs (NARX), which operated with a time-step duration of 10s. The secondary tasks of this model (all aggregated to 10s time steps) were to recognize time-series patterns in traffic light phases, predict whether or not a long queue will exist at its prediction horizon, and predict the percentage distribution of vehicles amongst the different directions of flow. Through the inductive use of information from these tasks, and the use of upstream departure flow measurements as inputs, the model was designed to predict the arrival flow of vehicles for every 10s interval. The traffic progression theories included in Robertson’s model inspired the architectural form of this final neural network architecture as well.

Using a VISSIM simulation environment of the final, most complex, selected case study (i.e. the intersection between N201 and N205), the final NARX prediction model was used to generate inputs for the controller installed at the intersection. This allowed for an evaluation of the model’s performance in regards to its intended predictive control application. The results of this evaluation are outlined in Table 1 and Table 2, for a high demand and low demand day respectively. For comparison, the neural network model’s performance is compared to that of Robertson’s model, which has been calibrated and expanded to fit the configuration and conditions of the selected case study. In addition, the adaptive controller and the proposed predictive controller are compared to a static, fixed, 120s controller as well.

TABLE 1: VEHICLE LOSS HOURS RESULTS FOR A HIGH DEMAND DAY (TUESDAY)

| Traffic Stream | Adaptive Controller Delay (vehicle loss hours) | NN Predictive Controller <i>Added</i> Delays (vehicle loss hours) | Rob. Predictive Controller <i>Added</i> Delays (vehicle loss hours) | 120s Cycle Fixed Controller <i>Added</i> Delays (vehicle loss hours) |
|----------------|--|---|---|--|
| 2 | 25.40 | 3.13 | 105.49 | 106.5 |
| 3 | 43.39 | 8.18 | -27.81 | 11.45 |
| 4 | 56.13 | 3.68 | -16.25 | -1.26 |
| 6 | 121.77 | 27.66 | -87.94 | -28.92 |
| 7 | 8.27 | 1.53 | 62.53 | 69.71 |
| 8 | 28.27 | 11.91 | 98.01 | 107.87 |

TABLE 2: VEHICLE LOSS HOURS RESULTS FOR A LOW DEMAND DAY (WEDNESDAY)

| Traffic Stream | Adaptive Controller Delay (vehicle loss hours) | NN Predictive Controller <i>Added</i> Delays (vehicle loss hours) | Rob. Predictive Controller <i>Added</i> Delays (vehicle loss hours) | 120s Cycle Fixed Controller <i>Added</i> Delays (vehicle loss hours) |
|----------------|--|---|---|--|
| 2 | 7.80 | 0.53 | 3.44 | 48.17 |
| 3 | 11.90 | -3.66 | -3.60 | 0.33 |
| 4 | 23.98 | 7.17 | 3.50 | 7.92 |
| 6 | 40.05 | 35.89 | 37.11 | 16.46 |
| 7 | 1.21 | 0.41 | 11.21 | 48.39 |
| 8 | 6.26 | 13.79 | 20.31 | 80.41 |

From these results, and through the insights on the neural network (and Robertson's model) performance behavior derived during the design and calibration process, several conclusions can be made. Firstly, with regards to the NARX neural network's prediction accuracy, the results show that the model is still not strong enough to operate without consistent corrective procedures. Nonetheless, the model performance appears to be significantly more robust to congested conditions than the classical model, even after its parameters have been calibrated. Unlike the classical model, at no point did the neural network make a prediction error that was significant enough to cause queue spillback, and therefore a breakdown of traffic flow to a wide moving jam. Accordingly, the flexibility of an adaptive controller to respond to the needs of denser traffic streams is relatively maintained with the neural network predictions. Moreover, looking at both Table 1 and Table 2, it can be seen that the performance of the neural network between high demand and low demand days is relatively similar, except for directions 3, 6 and 8 whose performance decreases significantly. This is an expected outcome though, since, especially during less dense traffic conditions, these three traffic streams are the most difficult for the neural network to predict the arrival flows of, due to their highly fluctuating traffic states. Alternatively, for the lower demand day, Robertson's model performance improved (although it still remained inferior to the neural network), since free flow traffic occurred more consistently, which is where this model's performance strength is. However, a larger portion of errors in both prediction models occur in the off-peak evening hours, which last longer in the low demand day. This is a result of both missed vehicle and ghost car errors, which occur during less dense, highly dispersed periods of day. Missed vehicles, overlooked by a prediction model, wait for long periods of time before receiving a green light. Alternatively, predicted ghost cars cause the controller to output a green light for an empty traffic stream, taking away green time from traffic streams that need it. While these occur less for the neural network model, they still result in significant delays.

While the ideal short term traffic flow prediction model performance has not yet been reached, this designed NARX neural network does tackle some of the limitations of state of the art predictive controllers. Straight forward instructions are supplied for its input and output configurations, and model training, which make it quite user friendly when installing it and adapting it to different upstream-downstream intersection combinations. Additionally, the robustness of this model to different traffic states allows for less drastic corrective measures to be needed. Moreover, since validation checks on the model's performance are fed back into the prediction model itself, rather than just to the predictive controller, model itself receives inputs from the actual conditions downstream, and is able to, accordingly, better adapt its predictions to these conditions. Hence, when long queues propagate backwards, past the far away loop detector, this traffic state is recognized by the prediction model itself, and as part of the adaptive characteristic of the neural network, it is able to adjust its predictions to this state. Accordingly, a predictive controller operating on this model will not need to identify a long queue such as this as a "system failure" and resort to the fail-safe control method. This is concurred by the fact that under no conditions, including when the intersection sides were saturated, was there an experienced system failure in the form of a queue spill back. Additionally, a significant advantage of this study is that by designing, testing and calibrating the model for case studies that gradually increased in complexity, the strengths and weaknesses of the model, for different conditions, have been identified, which may be used for the development of systematic verification and corrective procedures.

With these conclusions, this model can be considered as a new starting point for traffic flow progression modeling. Accordingly, in the recommendations for future work, means with which the NARX neural network's prediction performance can be improved are suggested. These include the use of alternative data sources than detector counts, the use of online adaptive learning, and adding a long-short-term-memory (LSTM) layer to the model. However, since the high level objective is to industrialize the conceptual design proposed in Figure 1, recommendations for developing and evaluating the system design are also proposed. These recommendations include proposals for the validation and correction procedure that best maintain the controllers predictability, and suggested safety and efficiency tests for performance evaluation. Lastly, recommendations for how to evaluate the benefits of the T2G/R and GLOSA driver assistance functionalities are also provided; since, on the highest level, the benefits that can be derived from the operational implementation of these use cases is what achieves the intended societal contribution of this research.

TABLE OF CONTENTS

| | |
|---|------------|
| ACKNOWLEDGEMENTS | II |
| ABSTRACT | IV |
| EXECUTIVE SUMMARY | VI |
| LIST OF FIGURES AND TABLES | XII |
| LIST OF FIGURES | XII |
| LIST OF TABLES..... | XIII |
| 1. INTRODUCTION | 1 |
| 1.1. THE NEED FOR ITS SOLUTIONS..... | 1 |
| 1.2. PROBLEM STATEMENT..... | 2 |
| 1.2.1. <i>Inefficiencies at Urban Intersections (Societal)</i> | 2 |
| 1.2.2. <i>Predictability of an Adaptive Traffic Light Schedule (Scientific)</i> | 3 |
| 1.2.3. <i>Predictability of Incoming Traffic (Scientific)</i> | 5 |
| 1.2.4. <i>Concerns with Changing Controller Logic</i> | 6 |
| 1.3. SOLUTION: PROPOSED CONCEPTUAL DESIGN | 7 |
| 1.3.1. <i>The Systems Engineering Approach</i> | 7 |
| 1.3.2. <i>Main Advantage of Predicting Traffic Instead of Traffic Lights</i> | 12 |
| 1.4. RESEARCH OBJECTIVES AND SCOPE..... | 13 |
| 1.4.1. <i>Research Objectives</i> | 13 |
| 1.4.2. <i>Research question and Sub-questions</i> | 13 |
| 1.4.3. <i>Scope of Works</i> | 14 |
| 1.5. RESEARCH APPROACH AND METHODOLOGY | 15 |
| 1.6. THESIS OVERVIEW | 17 |
| 1.6.1. <i>Outline of Report Structure</i> | 17 |
| 1.6.2. <i>Scientific and Societal Contribution</i> | 17 |
| 2. REQUIREMENTS AND APPROACH FOR A SHORT-TERM TRAFFIC FLOW PROGRESSION MODEL | 18 |
| 2.1. INTRODUCTION | 18 |
| 2.2. THE MAIN CONTRIBUTIONS TO ANALYTICAL TRAFFIC FLOW PROGRESSION MODELING..... | 18 |
| 2.3. ALTERNATIVE APPROACHES USING MACHINE LEARNING ALGORITHMS | 21 |
| 2.4. TRAFFIC FLOW PREDICTIONS AND SCOOT CONTROLLERS | 23 |
| 2.5. THE GAP BETWEEN PREDICTION MODEL AVAILABILITIES AND NEEDS..... | 24 |
| 2.6. DEFINED BASELINE FOR MODEL EVALUATION | 26 |
| 2.7. CONCLUSIONS..... | 26 |
| 3. DESIGNING THE NEURAL NETWORK TRAFFIC PROGRESSION MODEL | 27 |
| 3.1. INTRODUCTION | 27 |
| 3.2. CONTEXT | 27 |
| 3.2.1. <i>Motivation for the Case Study</i> | 27 |
| 3.2.2. <i>Generating the Synthetic Data</i> | 27 |
| 3.3. SETTING UP THE NEURAL NETWORK PARAMETERS AND DATA | 28 |
| 3.3.1. <i>Model Setup</i> | 29 |
| 3.3.2. <i>Data Pre-(and Post-)Processing</i> | 31 |
| 3.4. DESIGN PROCESS OF NEURAL NETWORK ARCHITECTURE | 32 |
| 3.4.1. <i>Experimental Setup & KPIs</i> | 33 |
| 3.4.2. <i>Results and Evaluation</i> | 33 |
| 3.4.3. <i>Relationship Between Traffic Progression Theory and NARXNET</i> | 37 |
| 3.5. CONCLUSIONS..... | 39 |
| 4. VERIFICATION, CALIBRATION AND TESTING OF PROGRESSION MODEL | 40 |
| 4.1. INTRODUCTION | 40 |
| 4.2. CONTEXT | 40 |
| 4.2.1. <i>Motivation for Case Study</i> | 40 |
| 4.2.2. <i>Case Study Conditions</i> | 40 |
| 4.3. EXPERIMENTAL SETUP & KPI'S | 41 |
| 4.4. VERIFICATION AND CALIBRATION OF THE ARCHITECTURE DESIGN FOR REAL DATA | 42 |

| | | |
|-----------|---|-----------|
| 4.5. | RESULTS AND EVALUATION OF DESIGN TESTING | 44 |
| 4.5.1. | <i>Traffic Progression Conditions</i> | 44 |
| 4.5.2. | <i>Insights from Comparison with Classical Model</i> | 44 |
| 4.5.3. | <i>Insights From Different Prediction Horizons</i> | 45 |
| 4.5.4. | <i>Insights from Different Directions of Flow</i> | 46 |
| 4.5.5. | <i>Insights from Different Aggregation Levels of Time Steps</i> | 47 |
| 4.6. | CONCLUSIONS..... | 47 |
| 5. | REENGINEERING, APPLICATION AND EVALUATION OF PROGRESSION MODEL..... | 48 |
| 5.1. | INTRODUCTION | 48 |
| 5.2. | CONTEXT | 48 |
| 5.2.1. | <i>Motivation for Case Study</i> | 49 |
| 5.2.2. | <i>Case Study Conditions</i> | 49 |
| 5.3. | EXPERIMENTAL SETUP AND KPIS | 51 |
| 5.4. | MODELING EXPANSIONS FOR TRAFFIC LIGHTS AND QUEUES..... | 51 |
| 5.4.1. | <i>Reengineering and Application of the Neural Network Architecture</i> | 52 |
| 5.4.2. | <i>Expansion, Calibration and Application of the Classical Model</i> | 54 |
| 5.5. | RESULTS OF BOTH MODELS AND COMPARATIVE EVALUATION | 56 |
| 5.5.1. | <i>Aggregated KPI Results and Evaluation</i> | 56 |
| 5.5.2. | <i>Results and Evaluation Per Traffic states</i> | 57 |
| 5.5.3. | <i>Evaluation of Results per defined Error occurrences</i> | 61 |
| 5.5.4. | <i>Evaluation of Results Per Magnitudes of Errors</i> | 63 |
| 5.5.5. | <i>Quality Evaluation of Used Data</i> | 64 |
| 5.6. | CONCLUSIONS..... | 65 |
| 6. | EVALUATING THE NEURAL NETWORK TRAFFIC PROGRESSION MODEL FOR PREDICTIVE CONTROL APPLICATIONS..... | 66 |
| 6.1. | INTRODUCTION | 66 |
| 6.2. | CONTEXT | 66 |
| 6.3. | APPROACH TAKEN FOR MODELING AND EVALUATING A PREDICTIVE CONTROLLER..... | 66 |
| 6.4. | EXPERIMENTAL SETUP AND KPIS | 67 |
| 6.5. | RESULTS AND EVALUATION OF PREDICTIVE CONTROLLER..... | 68 |
| 6.5.1. | <i>Travel Time Delay Results for Both Prediction Models and Comparative Evaluation</i> | 68 |
| 6.5.2. | <i>Prediction Horizon Evaluation</i> | 70 |
| 6.6. | CONCLUSIONS..... | 71 |
| 7. | CONCLUSIONS, DISCUSSION, REFLECTION & RECOMMENDATIONS | 72 |
| 7.1. | OVERVIEW OF RESEARCH CONCLUSIONS | 72 |
| 7.2. | ANSWERS TO RESEARCH QUESTIONS | 74 |
| 7.3. | REFLECTION ON METHODOLOGY AND RESEARCH PROCESS | 75 |
| 7.4. | POSSIBLE IMPROVEMENTS TO THE WORK..... | 76 |
| 7.5. | RECOMMENDATIONS FOR FUTURE WORK | 76 |
| 8. | REFERENCES..... | 79 |
| 9. | APPENDIXES..... | 82 |
| A. | CONTROL STRATEGY ALTERNATIVES | 82 |
| B. | CURRENT DUTCH VEHICLE ACTUATED CONTROL STRATEGY..... | 85 |
| C. | PROJECT DIVISIONS, TASKS AND TOOLS USED | 87 |

LIST OF FIGURES AND TABLES

LIST OF FIGURES

| | |
|--|-----|
| FIGURE 1: THE PROPOSED CONCEPT DESIGN, SHOWING HOW THE PROPOSED CHANGES TO THE CURRENT CONTROLLER ARE IN THE INPUT AND OUTPUT INFORMATION FLOW, WHILE THE CONTROL LOGIC REMAINS THE SAME | VI |
| FIGURE 2: A VISUALIZATION FOR THE FLOW OF INFORMATION FOR PROPOSED TRAFFIC LIGHT MANAGEMENT METHODOLOGY..... | VII |
| FIGURE 3: PLATOON PROFILES AT TWO SUCCESSIVE SPACE TIME POINTS, AS PRESENTED BY BIE ET. AL. (2014)..... | 6 |
| FIGURE 4: THE PREDICTABILITY VS. FLEXIBILITY OF DIFFERENT TYPES OF TYPICAL CONTROLLERS AS PRESENTED BY KOEN DE CLERCQ (2016)..... | 6 |
| FIGURE 5: FIRST SIX STEPS OF SYSTEMS ENGINEERING APPROACH USED FOR THIS STUDY..... | 7 |
| FIGURE 6: VISION OF AN ACTUATED CONTROLLER (TOP) VS. THE VISION OF AN ADAPTIVE CONTROLLER (BOTTOM)..... | 8 |
| FIGURE 7: DETECTOR GEOMETRY FOR A DUTCH ADAPTIVE CONTROLLER | 9 |
| FIGURE 8: DIAGRAM SHOWING HOW THE PROPOSED PRODUCT WILL ONLY INTERFERE WITH THE INFORMATION FLOW OF THE CONTROLLER, BUT ALL OTHER ASPECTS WILL REMAIN THE SAME. | 10 |
| FIGURE 9: FLOW OF INFORMATION FOR INTERSECTION TRAFFIC LIGHT MANAGEMENT FROM THE CLOUD | 10 |
| FIGURE 10: VISUALIZATION OF PREDICTIVE CONTROL PROPOSAL | 11 |
| FIGURE 11: EXAMPLE INTERSECTION WHERE THE FAR AWAY LOOP IS PLACED PRIOR TO THE LANE EXPANSION..... | 12 |
| FIGURE 12: BREAKDOWN OF THE PROBLEM | 13 |
| FIGURE 13: THE PART OF THE SYSTEMS ENGINEERING APPROACH INCLUDED WITHIN THE SCOPE (COLORED PART). | 14 |
| FIGURE 14: TOP LEVEL RESEARCH APPROACH: BREAKING DOWN THE PROBLEM TO FIND ITS ROOTS AND PROPOSE A SOLUTION | 15 |
| FIGURE 15: DETAILED LEVEL RESEARCH APPROACH: PROCESS FLOW CHART FOR PREDICTION MODEL RESEARCH | 16 |
| FIGURE 16: QIAO ET. AL (2001)'S NARX NEURAL NETWORK ARCHITECTURE | 22 |
| FIGURE 17: THE FEED FORWARD NEURAL NETWORK ARCHITECTURE OF JIAO ET. AL. (2014)'S MODEL, WHICH TAKES DETECTOR DATA AS INPUT AND OUTPUTS PERCENTAGE DIRECTION OF FLOW. | 22 |
| FIGURE 18: A VISUAL REPRESENTATION OF A NAR NETWORK WITH OPEN LOOP FEEDBACK..... | 23 |
| FIGURE 19: GEOMETRY OF FICTITIOUS CASE-STUDY | 28 |
| FIGURE 20: TWO CHOICES FOR ACTIVATION FUNCTIONS AS PRESENTED BY STANFORD UNIVERSITY (2017)..... | 30 |
| FIGURE 21: DATA PREPROCESSING RECOMMENDATIONS, AS PRESENTED IN STANFORD UNIVERSITY (2017) | 32 |
| FIGURE 22: ARCHITECTURE OF A SIMPLE FEED FORWARD NEURAL NETWORK, WHICH TAKES INPUTS FROM THE TWO UPSTREAM FLOW POINTS AND PREDICTS THE FLOW AT ONE OF THE DOWNSTREAM LANES, AT THE DESTINATION POINT OF THE TIME-SPACE PREDICTION HORIZON..... | 33 |
| FIGURE 23: FLOW VS. TIMESTEPS CHART FOR BOTH TARGET (ACTUAL) FLOW AND PREDICTED FLOWS BY A FEEDFORWARD NEURAL NETWORK. THE RESULTS ARE FOR EACH ARRIVAL LANE WITHIN THE FICTITIOUS CASE STUDY. | 33 |
| FIGURE 24: FEED FORWARD MODEL EXPANSION: MULTI-TASK LEARNING THE ARRIVAL FLOWS OF ALL LANES SIMULTANIOUSLY | 34 |
| FIGURE 25: FEED FORWARD MODEL EXPANSION: MULTI-TASK LEARNING THE ARRIVAL FLOWS AND TURNING PERCENTAGES OF ALL LANES SIMULTANIOUSLY | 34 |
| FIGURE 26: NARX NETWORK ARCHITECTURE, SELECTED FOR THIS SYNTHETIC CASE STUDY | 35 |
| FIGURE 27: RESULTS FOR NARX NETWORK | 35 |
| FIGURE 28: LEFT = RESULTS OF POST-TRAINING REGRESSION ANALYSIS CONDUCTED BETWEEN THE MODEL'S PREDICTED OUTPUTS AND THE TARGET OUTPUTS. RIGHT = ERROR HISTOGRAM SHOWING WHAT ERRORS APPEARED AND HOW MANY TIMES..... | 36 |
| FIGURE 29: ARCHITECTURE OF NARNET WHICH TAKES INPUTS AS PREVIOUS TIME-STEPS OF ARRIVALS AND OUTPUTS FUTURE TIME STEPS OF ARRIVALS FROM THE SAME DATASET. | 36 |
| FIGURE 30: RESULTS FOR NARNET | 37 |
| FIGURE 31: TWO TYPES OF TRAFFIC PROGRESSIONS: ON THE LEFT IS LONGITUDINAL PLATOON DISPERSION, AND ON THE RIGHT IS THE LATERAL SPREAD OF VEHICLES AMONGST THE LANES. | 37 |
| FIGURE 33: A VISUALIZATION OF THE TIME-SPACE RELATIONSHIP BETWEEN DEPARTURE FLOWS UPSTREAM AND ARRIVAL FLOWS DOWNSTREAM AT SUBSEQUENT INTERSECTIONS | 38 |
| FIGURE 32: THE GRAPHICAL REPRESENTATION OF ROBERTSON'S MODEL, AS DISPLAYED BY THE NATIONAL PROGRAM OF TECHNOLOGY ENHANCED LEARNING (MATHIEW, 2014) | 38 |
| FIGURE 34: A VISUALIZATION OF HOW, AT THE POINT OF MEASUREMENT AND PREDICTION, THE PERCENTAGES OF ARRIVALS PER LANE AT THE TIME HORIZON HAVE NOT YET OCCURRED AND SO ARE STILL UNKNOWN. | 39 |
| FIGURE 35: DELFT URBAN CORRIDOR CASE STUDY | 41 |
| FIGURE 36: RESULTS FOR FEED FORWARD NEURAL NETWORK | 43 |
| FIGURE 37: RESULTS FOR NARNET | 43 |
| FIGURE 38: RESULTS FOR NARXNET | 43 |
| FIGURE 39: A LEFT TURN CONTROLLER'S GREEN TIME DURATIONS SCHEDULE, FOR AN OFF-PEAK HOUR (A) AND SOME PEAK HOURS (B), AS PRESENTED BY KRIJGER (2013) | 46 |
| FIGURE 40: TRAFFIC PROGRESSION FOR STRAIGHT AHEAD AND LEFT TURNING FLOW STREAMS..... | 46 |
| FIGURE 41: GOOGLE MAPS SATELLITE IMAGE OF CASE STUDY NETWORK | 49 |

| | |
|---|----|
| FIGURE 42: SCHEMATIC DRAWING AND SIGNAL HEAD CONFIGURATION FOR CASE-STUDY | 50 |
| FIGURE 43: O-D PAIR COMBINATIONS OF FLOWS MOVING FROM EXIT POINTS OF ONE INTERSECTION TO THE ARRIVAL POINTS OF THE NEXT | 50 |
| FIGURE 44: PHASE 1 OF PROGRESSION MODEL REENGINEERING (EXAMPLE FROM VERBINDINGSWEG SIDE)..... | 52 |
| FIGURE 45: PHASE 2 OF PROGRESSION MODEL REENGINEERING (EXAMPLE FROM VERBINDINGSWEG SIDE)..... | 53 |
| FIGURE 46: FINAL PROGRESSION MODEL ARCHITECTURE FOR FULL INTERSECTION APPLICATION (EXAMPLE FROM VERBINDINGSWEG SIDE) | 53 |
| FIGURE 47: BIE ET. AL.'S PROPOSAL [9] FOR HOW TO BEST USE ROBERTSON'S DISPERSION MODEL FOR FULL INTERSECTION APPLICATIONS | 54 |
| FIGURE 48: VISUALIZATION OF TIMESTAMPS AT WHICH INFORMATION IS COLLECTED FOR TRAFFIC STATE CLASSIFICATION | 58 |
| FIGURE 49: GREEN WITH CONGESTED FLOW STATE | 58 |
| FIGURE 50: GREEN WITH FREE FLOW STATE | 59 |
| FIGURE 51: RED WITH EMPTY LANE STATE | 59 |
| FIGURE 52: RED WITH SHORT QUEUED LANE STATE..... | 59 |
| FIGURE 53: RED WITH LONG QUEUED LANE STATE | 59 |
| FIGURE 54: RMSE'S FOR THE TWO MODELS, PER DEFINED TRAFFIC CONDITION..... | 60 |
| FIGURE 55: EXAMPLES OF LOGGING DATA FOUND FOR SOME DETECTORS..... | 64 |
| FIGURE 56: A VISULIZATION OF THE MISSED VEHICLES ERRORS THAT RESULT FROM DOUBLE LANE FAR AWAY LOOP DETECTORS (LEFT), WHICH DO NOT OCCUR FOR SINGLE LANE FAR AWAY LOOPS (RIGHT) | 65 |
| FIGURE 57: SCREEN SHOT OF VISSIM SIMULATION MODEL..... | 66 |
| FIGURE 58: SIGNAL GROUPS FOR FIXED 120S CYCLE CONTROLLER | 68 |
| FIGURE 59: A VISUALIZATION OF THE DIFFERENCE BETWEEN THE END-TO-END PREDICTION HORIZON AND THE END-TO-START APPLICAION HORIZON. . | 70 |
| FIGURE 60: VERIFICATION CHECK & CORRECTIVE MEASURES ADDITION TO CONCEPT DESIGN | 78 |
| FIGURE 61: AN EXAMPLE OF THREE POSSIBLE SIGNAL GROUPS FOR A TYPICAL INTERSECTION..... | 82 |
| FIGURE 62: VISION OF AN ACTUATED CONTROLLER (TOP) VS. THE VISION OF AN ADAPTIVE CONTROLLER (BOTTOM)..... | 84 |
| FIGURE 63: MULTI-LAYER, HIERARCHICAL, DUTCH TLC CONTROL STRUCTURE..... | 85 |
| FIGURE 64: A SAMPLE INTERSECTION FROM VAN KATWIJK(2008) , FEATURING 5 LABELED VEHICLE TRAFFIC STREAMS, AND ONE BICYCLE STREAM (NO. 26). <i>IMAGE WAS REFERENCED FROM SCHEEPIJENS (2015).</i> | 86 |
| FIGURE 65: BLOCK STRUCTURE FOR THE SAMPLE INTERSECTION, INCLUDING AN OUTLINE FOR THE PRIMARY REALIZATION PROCESS FROM VAN KATWIJK (2008). <i>IMAGE WAS REFERENCED FROM SCHEEPIJENS (2015).</i> | 86 |
| FIGURE 66: AN OUTLINE FOR THE ALTERNATIVE REALIZATION PROCESS FOR THE SAMPLE INTERSECTION FROM VAN KATWIJK (2008). <i>IMAGE WAS REFERENCED FROM SCHEEPIJENS (2015).</i> | 86 |
| FIGURE 67: DETECTOR GEOMETRY FOR A DUTCH ADAPTIVE CONTROLLER | 87 |

LIST OF TABLES

| | |
|---|------|
| TABLE 1: VEHICLE LOSS HOURS RESULTS FOR A HIGH DEMAND DAY (TUESDAY) | VIII |
| TABLE 2: VEHICLE LOSS HOURS RESULTS FOR A LOW DEMAND DAY (WEDNESDAY) | VIII |
| TABLE 3: COMPARISON BETWEEN THE DESING OF A NEURAL NETWORK OR A NEW METHODOLOGY FOR CALIBRATING ROBERTSON'S MODEL..... | 25 |
| TABLE 4: STATISTICAL ANALYSIS OF ROBERTSON'S PLATOON DISPERSION MODEL PERFORMANCE | 44 |
| TABLE 5: STATISTICAL ANALYSIS OF NARX NEURAL NETWORK MODEL PERFORMANCE..... | 44 |
| TABLE 6: THE PERCENTAGE FREQUENCY OF VEHICLE ARRIALS FOR THE FOUR DOWNSTREAM MEASUREMENT POINTS..... | 45 |
| TABLE 7: TESTING THE EFFECT OF CHANGES TO THE TIME-STEP HYPERPARAMETER | 47 |
| TABLE 8: OD PAIR WEIGHTS OF CONTRIBUTION BETWEEN THE VARIOUS DEPARTURE AND ARRIVAL FLOWS OF VEHICLES..... | 55 |
| TABLE 9: IDENTIFICATION OF THE VARIANCE VALUES NEEDED TO CALIBRATE THE MODEL SMOOTHING FACTORS 1 AND 2 | 55 |
| TABLE 10: KPI RESULTS FOR THE NEURAL NETWORK AND ROBERTSON'S MODEL | 56 |
| TABLE 11: HIT AND MISS ACCURACY OF BOTH PROGRESSION MODELS..... | 61 |
| TABLE 12: PERCENTAGE OF INSTANCES WHERE EACH MODEL MADE A PREDICTION ERROR OF ONE VEHICLE OR LESS | 62 |
| TABLE 13: PERCENTAGE OF PREDICTED GHOST CARS BY EACH MODEL | 62 |
| TABLE 14: PERCENTAGE OF UNPREDICTED CARS BY BOTH MODELS | 62 |
| TABLE 15: MAGNITUDES OF ERRORS ON DIRECTION 2 AND THEIR PERCENTAGE OCCURANCES FOR BOTH MODELS..... | 63 |
| TABLE 16: MAGNITUDES OF ERRORS ON DIRECTION 3 AND THEIR PERCENTAGE OCCURANCES FOR BOTH MODELS..... | 63 |
| TABLE 17: VEHICLE LOSS HOUR RESULTS FOR A HIGH DEMAND DAY (TUESDAY)..... | 68 |
| TABLE 18: VEHICLE LOSS HOURS FOR A LOW DEMAND DAY (WEDNESDAY)..... | 69 |
| TABLE 19: AVERAGE NN PREDICTION CAUSED DELAY PER VEHICLE IN MORNING AND EVENING HOURS FOR LOW DEMAND AND HIGH DEMAND DAYS | 70 |
| TABLE 20: PLANNED TASKS FOR EACH DIVISION | 87 |

1. INTRODUCTION

This chapter begins by explaining why research on intelligent traffic systems (ITS), specifically those associated with urban intersections, is relevant by outlining the need for on board vehicle guidance through intersections. Next, Chapter 1.2. presents the problems associated with the operational implementation of the specific ITS functionalities Green Light Optimal Speed Advice (GLOSA) and Time to Green/Red (T2G/R). From Chapter 1.2., a secondary need is established, which is one for a new on-street traffic control strategy that matches the requirements of GLOSA and T2G/R implementation. Moving on to what can be done to satisfy this new need, Chapter 1.3. follows the System's Engineering Approach to propose, and justify, a new solution. Towards the end of this chapter, a concept design is reached to give an outline of how this proposal can be developed. In Chapter 1.4., the scope of works is restricted to the development and evaluation of the conceptual design. A significant component of the proposed control strategy design was a novel design for an accurate short-term traffic flow progression model that can be easily adapted to different traffic conditions and intersection configurations. Accordingly, the defined research objectives in Chapter 1.4. first place focus on the development and proper evaluation of this model, followed by an assessment of its implementation into the proposed conceptual design framework. The approach taken to tackle the defined research questions is then explained in Chapter 1.5; which is, finally, followed by an overview of the report in, Chapter 1.6, where the scientific contribution of this work is also clearly stated.

1.1. THE NEED FOR ITS SOLUTIONS

In the modern day, through the countless installed, on road, detection technologies, large quantities of data is collected and stored, creating pools of resources for entities to analyze, providing them with a better understanding of the sources of road inefficiencies; and to process, in order to extract the relevant information users need to overcome them. By tapping into this data, alongside the utilization of widely published expertise on traffic management advancements, next generation consumer navigation systems can be developed, which not only provide real-time information on traffic conditions, but also provide valuable advice on how to avoid, or pass through, problematic zones (Pan et. al., 2013). With the combined availability of data and the knowledge on traffic management strategies, several opportunities arise for further advancements in on-board, location-based, systems.

One particular market gap regarding location-based services, is a widespread availability of functional driver assistance use-cases that may safely and effectively guide vehicles through signalized intersections. Within the current infrastructure, used by the vast majority of the developed world, stop lights at crossings are a necessary evil to prevent collisions between traffic moving in conflicting directions. However, even with the use of dynamic traffic management strategies, which are adaptive to the real-time traffic conditions, and aim to realize a global minimum of waiting times at each signalized intersection, traffic inefficiencies will still occur as a result of the disruption of traffic flow a red light, in itself, causes (Axer & Friedrich, 2016). In reaction to an unanticipated switch from green to amber, some drivers resort to sudden stops or speedy accelerations, which cause safety concerns. However, even when safe-driving precautions are taken, once the flow of vehicles on a road is interrupted by a red light, a backwards moving shock wave occurs, inducing a congested traffic state upstream the intersection. The stop-and-go behavior elicited by these conditions produces the largest amount of fuel consumption and CO₂ emissions along a vehicle's trip (Suthaputchakun et al, 2015). More so, once the traffic light turns green, the inability for drivers to anticipate when they should accelerate from stop, and the time it takes to accelerate to free-flow speed, results in a queue discharge rate that can be as low as 75% of the road's capacity (Brilon & Zurlinden, 2003). Accordingly, it does not suffice to localize all intersection traffic management to the controller. Clear safety, travel time and fuel efficiency gains can be achieved by informing vehicles of upcoming traffic light schedules and/or advising them on how to approach and exit the signalized intersections through infrastructure to vehicle (I2V) communication services. I2V communication messages that are designed to guide vehicles through problematic zones are coined **driver assistance use cases**.

Aside from the safety and efficiency gains at urban intersections that driver assistance use cases (for crossing signalized intersections) can achieve, the upgrade of on-board systems from 'informative' to 'supportive', in this manner, will additionally address the needs of a new target market: automated vehicles. Preemptive information on the upcoming signal light schedules facilitates for cooperative adaptive cruise control (CACC) functions to further adapt automated vehicle platoon speeds for a safe and efficient approach towards, and crossing of, intersections, which creates a new concept in the cruise control family: Predictive Cruise Control (PCC) (Asadi & Vahidi, 2010).

Hence, the upcoming signal state of a signalized intersection is one of the most relevant pieces of information that can be used for cooperative driver assistance as well (Barthauer & Friedrich, 2014). Using this information, safety and ethics concerns regarding the decision times in which automated vehicles must react to an unanticipated signal change (within a short range upstream an intersection) can be overcome (Shebeeb, 2016), and the risk of faulty sensor readings, as a result of poor weather conditions, can be mitigated. Accordingly, policy makers have conceded that, for safety reasons, a clear interaction between automated vehicles, road users, and the infrastructure is a necessary prerequisite to the launch of level 4 automated vehicles into the urban network (Feddes, 2017). This claim is supported by Rijkswaterstaat, who also believes automating a vehicle is not enough: it must also be able to connect to, and cooperate with, its environment (Alkim, 2017). Hence, facilitating an I2V functionality for signalized urban intersection crossings not only reduces the amount of opportunity costs spent on urban roads, but also serves as an important stepping stone for the realization of self-driving vehicles on these same roads.

1.2. PROBLEM STATEMENT

In 2017, the Dutch Ministry of Infrastructure and the Environment initiated the “Talking Traffic” project, which is conducted in partnership with over 30 private companies, with the intention of improving accessibility, flow, road safety and livability within the Dutch infrastructure, through the exchange of real-time information between road users and traffic systems. Use-case four of this project is to provide in-vehicle Time-to-Green/Red (T2G/R) information, and Green Light Optimal Speed Advice (GLOSA), to users via their onboard navigation services (Masters, 2017). Although Dutch urban traffic control strategies are already quite advanced, the Ministry of Transport and the Environment anticipates that this next generation traffic management tool will go even one step further, pushing past the currently inevitable inefficiencies caused by having vehicles stop at red lights. Hence, the aim of this service is to target the time and fuel opportunity costs associated with the existence of urban intersections. However, using on-street vehicle detector counts, the imbedded traffic control algorithm, installed in Dutch traffic light controllers (TLCs), is able to continuously run an optimization model, with the objective of minimizing overall delay times at an intersection, for every time step. Following this logic, as new vehicles arrive, with each time step, a controller’s outputted schedule fluctuates in response. Hence, while the strength of Dutch urban traffic controllers is in their ability to adapt their traffic light schedules, online, to real-time traffic flow conditions, this is now appearing to be their limitation as well. The adaptability of traffic light schedules to detected arrival and departure flows of vehicles is reflected in the information provided to end users in the form of fluctuating T2G/R and GLOSA messages. With unstable countdowns and variable speed advice, this functionality is likely to be unsafe and cause user frustration rather than serve its intended purpose.

Hence, the problem being tackled in this study can be broken down into the following layers: 1) at the top layer, there is a societal problem, which is that the existence of urban intersections create time and fuel opportunity costs that cannot be reduced using on the ground control strategies; 2) it is quite difficult to provide stable and reliable T2G/R and GLOSA to users in the Netherlands due to how adaptive controller schedules are to the continuously detected arriving and departing flows; and 3) the source of unpredictability for adaptive controllers is attributable to the erratic behavior of the short term traffic flow arrivals it responds to. Since the core issue is the fluctuating outputs of adaptive controllers though, an additional dimension is included in the problem analysis, which is that there are several consequences that need to be considered if an alteration to the highly efficient, nationwide used control logic is suggested, in attempts to make it more predictable. This chapter will describe the details of these aspects in order to highlight the complexity of bringing use-case four of the Talking Traffic project to an operational level.

1.2.1. INEFFICIENCIES AT URBAN INTERSECTIONS (SOCIETAL)

With economic growth, developed countries are facing an exponentially increasing demand for personal mobility (de Nunzio, 2015). As the resulting congestion becomes more prominent, not only is fuel consumption magnified, but the increased travel times create opportunity costs that, in turn, slow down this economic growth (de Nunzio, 2015). In attempts to break this cycle, policy makers are now facing the challenge of finding a balance between economic growth and the demand for private mode transport (European Commission - Directorate General for Environment, 2004). Accordingly, focus on minimizing the adverse effects of this private mode, in parallel to attempts to reduce its usage, is necessary. To do so, the sources of inefficiency for road transport are broken down

into the following three categories: the energy consumption of motors, the vehicle driving profiles along road segments, and traffic control strategies (Wu et. al., 2015). As this research is not taken from the perspective of the automotive industry, but rather from that of a mobility service provider, focus will be placed on the latter two categories; and, with that regard, urban intersections become the main area of concern.

The existence of urban intersections, at which vehicles are required to stop for certain durations, create disturbances to the cruising driving state along a vehicle's route, which is the most fuel efficient state. However, in order to create accessibility between all points within a city, conflicting traffic directions are bound to meet at certain points, where safe traffic management is required. The alternative to that would be to replace intersections with bridges, tunnels or non-signalized roundabouts, which is neither economically, nor in many cases spatially, feasible (de Nunzio, 2015). Hence, mitigation plans to urban intersection inefficiencies rely on innovations within the on-street traffic controllers. Accordingly advanced urban traffic control strategies have been developed, such as the one used in the Dutch Traffic Actuation program, which adapts the start times and durations of green/red waves to the real-time detected traffic volumes in each direction, to minimize the number of unnecessary stops. However, the remaining "necessary stops" still result in over 50% of the fuel consumption of a vehicle's trip (Wu et. al., 2015). Not only do urban intersection red lights require vehicles to brake to a stop within a short distance, but the backwards moving shockwave that occurs as a result of this behavior induces a cyclic driving state of acceleration and idling that causes two thirds of the energy usage lost at intersections (Suthaputchakun et. al., 2015)(Wu et. al., 2015). Additionally, once the traffic light turns green, the inability for drivers to anticipate when they should accelerate from stop, and the time it takes to accelerate to free-flow speed, results in a queue discharge rate that can be as low as 75% of the road's capacity (Brilon & Zurlinden, 2003). Hence, current research trends are now recognizing that even necessary disturbances to the steady cruise of traffic, that are implemented to prevent collisions between conflicting traffic directions, should be targeted as economical (Wu et. al., 2015).

Accordingly, as a supplement to on ground traffic light control, technologies for preemptively influencing driver behaviors have become the most recent trend in traffic management. Coined as "eco-driving systems", these technologies are classified as: pre-trip systems, which provide navigation services, including information on what the quickest route is, where the nearest available parking space to the specified destination is, etc; in-trip systems, which provide driver assistance services such as cruise control, incident notifications, etc; and, post trip systems, which generate statistics summarizing the time and fuel losses during trips (de Nunzio, 2015). T2G/R and GLOSA information are classified as in-trip systems, as their purpose is to assist drivers, in real-time, through urban intersections, with the most economic driving profile possible (Stevanovic et. al., 2013). These systems use timely information on traffic signal durations and traffic light locations as an input, then use this information to provide vehicles with advice for a more uniform trip, with lower stopping frequencies at traffic lights (Stevanovic et. al., 2013). However, as mentioned at the start of Chapter 1.2., with adaptive controllers, the safe and efficient implementation of these use cases becomes difficult.

1.2.2. PREDICTABILITY OF AN ADAPTIVE TRAFFIC LIGHT SCHEDULE (SCIENTIFIC)

A key prerequisite for the successful use of T2G/R and GLOSA then is for the traffic light controller's schedule to be predictable. However, with actuated controllers (i.e. controllers that adapt their signal timings online to detected traffic conditions), it becomes difficult to predict exactly when one traffic light phase will end and the next will begin. To test the effect this uncertainty has on the efficacy of GLOSA, using VisSim, and the Comprehensive Modal Emission Model (CMEM), Stevanovic et. al. (2013) modeled two case study isolated intersections: one where a fixed time controller is used, and so the signal schedule is known in advance, and one where an actuated controller is used. In both models, different penetration rates of GLOSA were inserted (using the VisSim C2X platform); and, the number of vehicle stops and the quantity of CO₂ emissions were used as key performance indicators (KPIs) . GLOSA messages were also sent at different frequencies in this study. With regards to methodology, for the fixed controller the durations of green and red waves were taken directly from the control schedule; but, for the actuated controller, averages on green and red time durations were estimated for the different times of days (to create stable messages) instead. The results showed that, for a fixed time controller, as the penetration rate and message frequencies of GLOSA went up, notable improvements could be seen in stop frequencies, although less for emission rates (Stevanovic et. al., 2013). However, when using an actuated controller, GLOSA had a negligible effect stop frequencies and emissions, due to the slight differences between predicted and real signal phase durations; and, in fact, negatively affected traffic (as measured by the increasing vehicle delays) at higher penetration rates of these erroneous

messages (Stevanovic et. al., 2013). From their study, the authors concluded that highly accurate predictions on signal phase start times and durations are necessary in order for a functionality such as GLOSA to work (Stevanovic et. al., 2013).

Despite its severely low level of predictability, some studies have been conducted in attempts to accurately predict an actuated controller's schedule, using various approaches. For instance, Weishet & Hoyer (2014) developed a Support Vector Machine (SVM) predictor that aims to predict switching times of traffic signals for vehicle actuated controllers. The approach resulted in a 97% accuracy for switch times; but, under several assumptions. Most critically this methodology was used for actuated controllers with fixed green time cycle lengths. The variability, therefore, only existed for when the start time of the green phase for each signal group would be. Additionally, other modalities such as pedestrians, public transport and cyclists were ignored (Weishet & Hoyer, 2014). Similarly, Koukoumidis et. al. (2012) used the same assumptions in their study, but following a Support Vector Regression (SVR) methodology, to achieve the same objective. On average their error was around 2.45 seconds from the actual switch time, but no indication was given in the study on the applicability of the prediction model for other case-study intersections than the one used. The fixed green time duration and modality assumptions were also found in studies that used different KPIs for evaluating controller predictability as well. This included Protchkey et. al. (2014)'s study, where they used a Kalman Filter to calculate second by second probability statements for a traffic light signal state (i.e. the probability that the phase cycle is at a certain color, each second). Despite using these simplifying assumptions, at only 71% of instances did the model output probabilities that were 95% reflective of the actual upcoming signal states (Protchkey et. al., 2014). Also focusing also on probabilities, Barthauer & Feiedrich (2014) used a Markovian Chain to predict the likelihood of a shift in green time from one signal group to the next. However, once again, the underlying assumption for their model's success was that green time durations are fixed. Within Nissan, Silicon Valley's research center though, they did venture to develop a prediction model for the start time and duration of adaptive green phases. However, after designing a prediction model using, once, a neural network and, another time, Support Vector Regression, the conclusion was that, with an adaptive controller, predictability is too low for operational driver assistance use-cases (Scheepjens, 2015). While, with the SVR, the T2G predictions became significantly more accurate, notable errors remained for the T2R predictions (Scheepjens, 2015). Accordingly, the consensus appears to be that without restricting a controller's adaptivity, to some extent, and, therefore, regressing from the waiting time optimization strides taken over the last decade, it is unlikely a prediction for the traffic light schedule can be made with sufficient accuracy.

Recognizing that controller fluctuations are a result of traffic flow behaviors, other studies chose to use incoming traffic information as an additional input into their models. For instance, Axer & Friedrich (2016) explored the use of low-frequency floating car data, at low penetration rates, to derive daily patterns in signal schedules. The assumption made in this study was that travelers, most likely, have daily travel patterns; and so, actuated controllers must converge to a relatively fixed time of day schedule as well, which can be captured using the speed and positioning measurements of probe vehicles. This was proven true for certain periods in a day, where, with the use of sparse trajectory data, the authors managed to estimate signal timings with a 95-97% accuracy rate (Axer & Friedrich, 2016). However, aside from very few time-intervals where clear patterns could be seen, their model was unsuccessful in estimating signal schedules (Axer & Friedrich, 2016). Assuming traffic will follow a daily pattern was the biggest limitation to this study; since, although, on an aggregate level, patterns could be seen, in the form of peak and off-peak hours, real-time traffic flow through intersections is very stochastic, even when comparing time-of-day flow (Yin, 2008). Accordingly, flexible control strategies do not follow clear daily patterns, for all time periods, as well. These results are reflective of Krijger (2013)'s findings, who focused his research on drawing parallels between different traffic states and controller predictability. Krijger (2013)'s main findings were that, during peak hours, when urban roads are in a saturated state, controller schedules converge to fixed times, while when traffic volumes are low controller outputs no longer follow any probability distribution pattern (Krijger, 2013). Accordingly, Krijger's study concluded that the most difficult signal groups' schedules to predict are those of turning movements, as the demand on these lanes is, on average, much lower than lanes supporting straight ahead movements (Krijger, 2013).

The final approach found in literature for predicting controller schedules was to estimate green and red time durations indirectly, by attempting to accurately predict trip durations along a signalized corridor (Baluja et. al, 2015). The objective function of the authors' prediction model, which was built using a Next-Ascent Stochastic-Hillclimbing (NASH) algorithm, was to minimize the difference between predicted and actual journey times of vehicles; however, they did so by attempting to match hypothesized light settings to the actual ones (Baluja et. al., 2015). The case study

used was a corridor of several intersections in Silicon Valley, California, and their machine learning algorithm was trained using big, cell-phone based, GPS trajectory data (Baluja et. al., 2015). Within a SUMO (Simulation of Urban MObility) simulation environment, their model results showed a correlation of 0.8 between predicted and actual controller schedules; then, during the field test, the correlation went further down to 0.5. The drop in accuracy during the field test was a result of additional trip delays that occur at intersections, most likely due to the erratic behavior of drivers when approaching or causing an intersection (Yin, 2008), which was not accounted for in the approach (Baluja et. al., 2015). Hence, in the end, while the results showed, overall, a significant minimization in route time estimation errors, from approximately seven minutes to 51 seconds (Baluja etl al., 2015), providing notably more accurate information for navigation services, this methodology was still found to be unsuccessful for facilitating driver assistance use cases through signalized intersections.

1.2.3. PREDICTABILITY OF INCOMING TRAFFIC (SCIENTIFIC)

The common motif in all studies attempting to predict an actuated controller's schedule is that the stochasticity of arrival flows is the biggest hindrance. In some discussed studies, researchers attempted to overcome the volatility of traffic by targeting controllers that do not have adaptive green time durations. These controllers are still traffic-actuated in the sense that they do not output a green-wave unless vehicles have been detected, but the duration of this signal state is fixed. Other studies, such as that of Axer & Friedrich (2016) and Baluja et. al. (2015), did recognize that a relationship existed between traffic conditions and traffic light schedules, and attempted to include them within their models in some form. However, neither anticipated the extent of volatility of the incoming traffic flow the controller responds to. Accordingly, Axer & Friedrich (2016)'s prediction model could only output sufficient results for times of day where traffic is more homogeneous, and a daily pattern in the controller's schedule could be derived; and, Baluja et. al. (2015)'s matching model failed to accurately replicate the controller behavior, as it underestimated the impact the heterogeneity in driver behaviors would have on delays in journey times along vehicle trajectories. Accordingly, the main error in their traffic light schedule prediction was that the model assumed all delays resulted from a homogenous vehicle response to a red light induced shockwave (Baluja et. al., 2015). By understanding the limitations of these studies, it can be concluded that, part of assessing the obstacles regarding the predictability of adaptive signal schedules is understanding the (un-)predictability of traffic flow arrivals.

In theory, since nearly all signalized intersections within the Netherlands have vehicle detectors, vehicle flows can be monitored along their trip, making it possible to predict traffic flow arrivals, using upstream measurements. However, in actuality, what occurs is that flows exiting an upstream intersection are released from the stop-line as tight platoons with short time-headways; but, as these platoons travel along a link, variations in their vehicles' speeds arise as a result of vehicle interactions, roadway geometries, roadside activities, different driving habits, downstream signal states, etc. (Paul et. al., 2016)(Geroliminis & Skabardonis, 2005). Resultantly, an incoming platoon's arrival rate becomes dispersed, and no longer follows any probability curve, making it quite difficult to predict its progression from one point to the next (Qiao et. al., 2001). The current benchmark for traffic progression modeling is Robertson's platoon dispersion model, which has been widely implemented in famous traffic control systems, such as SCOOT, SCATS and TRANSYT. However, many research findings have indicated that the model performance is highly dependent on the values of its parameters, and that parameter calibrations of this model are highly site specific, leaving no methodology for quantitatively calibrating them (Bie et. al., 2014); and, even when calibrated, the errors that still result from this prediction model necessitate for the controllers that rely on it to have fail safe systems (de Nunzio et. al., 2015). This entails that the traffic light schedule predicted, based on a traffic flow arrival prediction, may need to be updated when actual arrivals are measured, reducing, once again, its predictability.

Longitudinal platoon dispersion is not the only traffic progression phenomenon that needs to be taken into consideration though. To limit the backward propagation of queues, which result from the shockwave induced by a red light, the number of lanes near an urban intersection increases. This road capacity expansion both serves as a shock absorber, and facilitates the placement of a controller signal head (i.e. queue server) per traffic movement direction (i.e. turning left/right or moving straight forward). The difficulty with this intersection configuration though is that it introduces an additional, lateral, dimension to the dispersion to vehicles; and, once again, while, on an aggregated level, patterns in turning percentages are clear, for short-term predictions, stochasticity begins to arise (Jiao et. al., 2014). More so, as shown in Figure 3, when vehicles spread out laterally, platoon profiles further change (Bie et. al., 2014). Accordingly, at the point of platoon profile (b), from Figure 3, not only is it required to predict the

percentage of vehicles heading towards which signal head, but it is additionally necessary to estimate the new dispersion parameters that result from the platoon splits (Bie et. al., 2014).

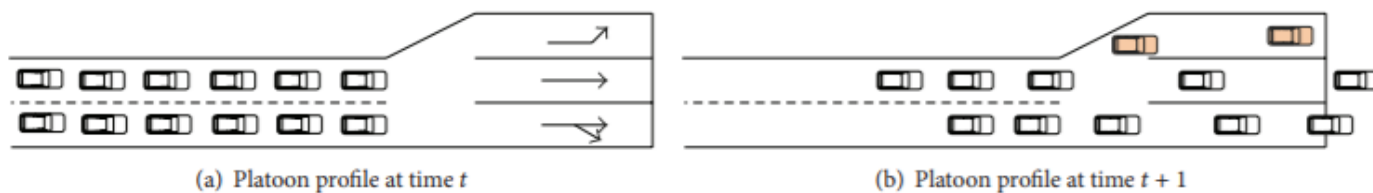


FIGURE 3: PLATOON PROFILES AT TWO SUCCESSIVE SPACE TIME POINTS, AS PRESENTED BY BIE ET. AL. (2014).

The last complication associated with the predictability of traffic flow arrivals is the existence of side streets and parking lots that vehicles may enter or exit from, undetected. With regards to traffic progression modeling, the additional difficulty becomes that total detected departure flow from one point does not always equal the total detected arrival flow to the next (Geroliminis & Skabardonis, 2005). The main source of vehicle detections is often detector readings, which provide unlabeled counts of vehicle passes. Accordingly, as successive platoons mix along a corridor, due to differences in vehicle speeds (Paul et. al., 2016), it becomes quite difficult to quantify a pattern of entering and exiting vehicles from side streets/parking lots to include within an analytical model (Geroliminis & Skabardonis, 2005). Altogether, this large number of uncertainties makes it quite complicated to develop a model for reconstructing measured upstream traffic at a downstream intersection. Resultantly, no prediction model, developed thus far, has been capable of predicting incoming queues at intersections, whether through the use of upstream detector counts, or with the additional assistance of floating car data, with enough accuracy for field application, without the use of excessive calibration and fail-safe measures (Hong et. al., 2014).

1.2.4. CONCERNS WITH CHANGING CONTROLLER LOGIC

From Chapters 1.2.2 – 1.2.3., appears that the underlying issue with regards to the application of GLOSA and T2G/R use cases with actuated controllers is that both approaches to traffic management are contradictory. Actuated controllers are a highly reactive form of traffic management, while driver assistance use cases are proactive. The prior approach does not attempt to anticipate the highly volatile vehicle arrival flows, but rather waits for vehicles to be detected and continuously reacts accordingly, while the later requires full knowledge of upcoming signal schedules for efficient operation. On one hand, the more flexible a controller's responses are to traffic, the more efficiently it can manage it; but, on the other, the less predictable its schedule becomes, due to the stochasticity of short-term traffic flow arrivals. Hence, in order to efficiently implement use case four of the Talking Traffic project, a change to the traffic controller methodology is needed, to make it more predictable.

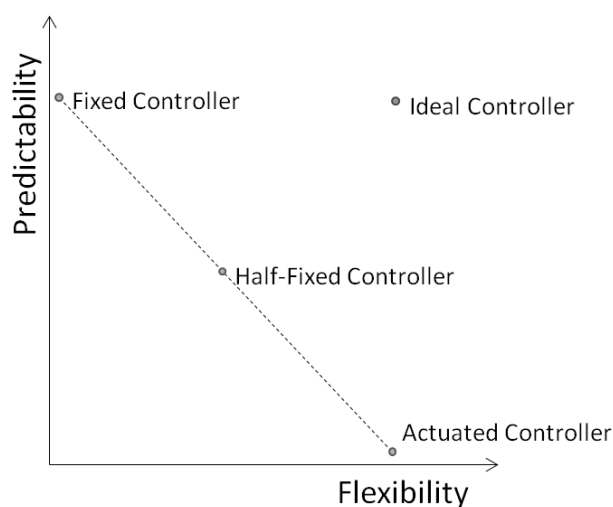


FIGURE 4: THE PREDICTABILITY VS. FLEXIBILITY OF DIFFERENT TYPES OF TYPICAL CONTROLLERS AS PRESENTED BY KOEN DE CLERCQ (2016).

Considering a change to the controller logic though adds a new dimension to the problem: a tradeoff between efficiency and predictability. For instance, one approach for increasing a controller's predictability is to follow the simplification made in the majority of studies outlined in chapter 1.2.2., which is to restrain the adaptability of a controller's green time duration to a more predictable range. However, this solution implies a regression in the waiting time optimization strides, taken in the Dutch traffic actuation program, over the past decades. The obstacle then becomes that a new controller is needed that supports both the flexibility features of an actuated controller and the predictability features of a fixed schedule controller, as shown in Figure 4. Additionally, the controller responses must not vary greatly from the currently installed actuated controllers, so as to not discomfort drivers with a new

control behavior that they are used to; and, the cost of this new control strategy must be low, so as to be a viable option to the already installed controllers. Naturally, the safety of its implementation is also a prerequisite.

1.3. SOLUTION: PROPOSED CONCEPTUAL DESIGN

At its core, the problem was concluded in Chapter 1.2.4 to be a contradiction between the ITS approach to traffic management, and the on-ground traffic light controller logic. Hence, the first step to solving the problem is establishing a proposal for this new strategy. Following this methodology, first a conceptual design objective is decided, then research goals are set for evaluating the feasibility of the design, and how well it satisfies the larger objective. As there are many complexities involved with the design of a new type of controller, the Systems Engineering Approach was used to structure the process, as shown in Figure 5. The remainder of this chapter will, therefore, focus on the concept design, by explaining the different control strategy alternatives, how the proposed one best fits the all the requirements of the problem definition, and how it is expected to stabilize the T2G/R and GLOSA messages.

For this study, the objective is not to develop a new algorithm for traffic management at urban intersections, but to propose an appropriate methodology for traffic control. Accordingly, the design specifications that need to be defined are: what source of input data will be fed into the controller (and with what means will this data be inputted), what existing control logic will be used, and how will the outputs of the controller be displayed to the drivers. Hence, the selection of the control logic itself is the core design decision that needs to be made, based on which the two other specifications will be decided.

1.3.1. THE SYSTEMS ENGINEERING APPROACH

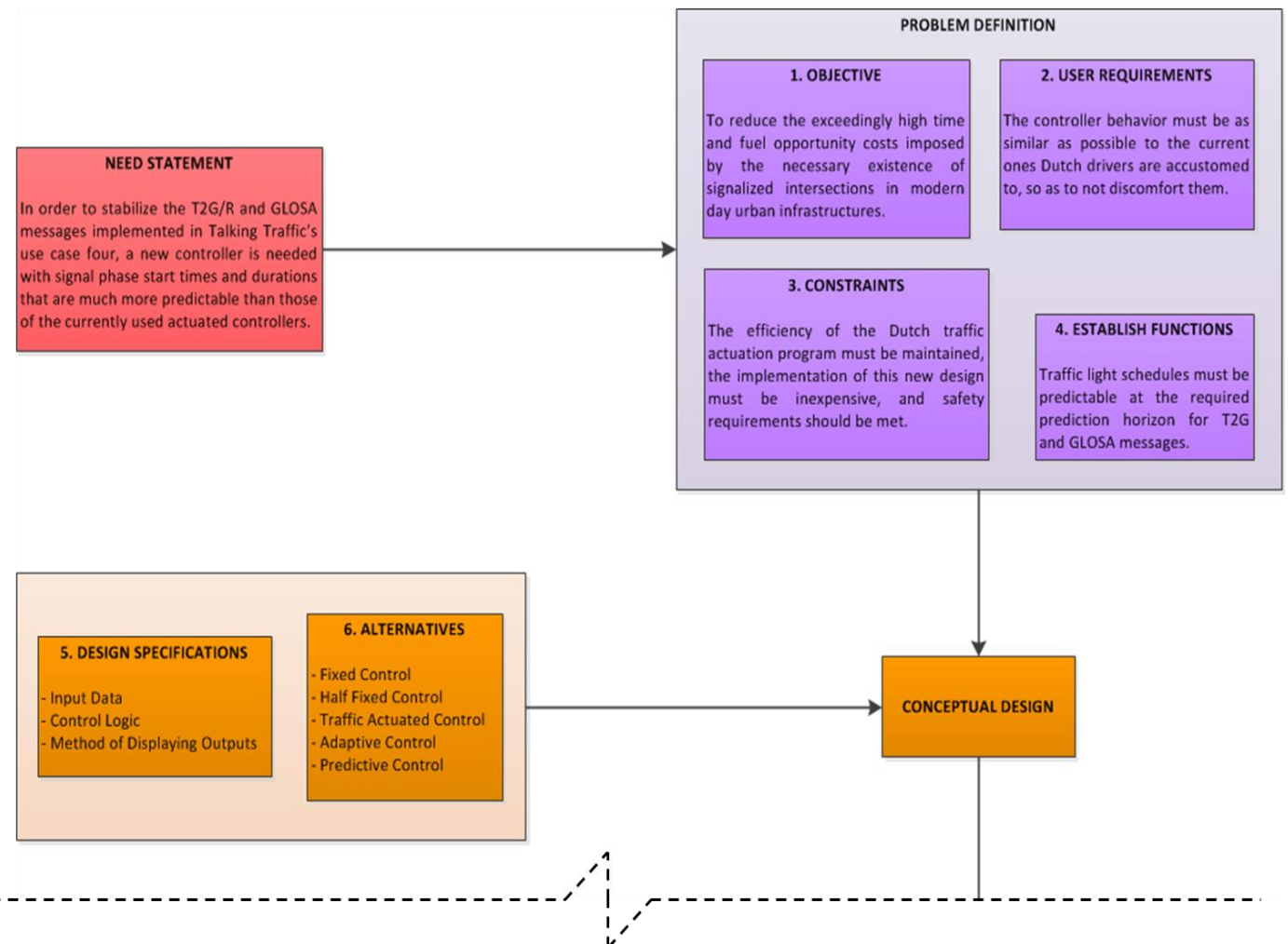


FIGURE 5: FIRST SIX STEPS OF SYSTEMS ENGINEERING APPROACH USED FOR THIS STUDY

There are several alternative types of controllers available in the market to select from; and, a literature study on these alternatives is provided in Appendix A. This literature study was not included within this chapter (i.e. within the main text), so as to not distract the reader with different traffic control logics, when the objective of the report, as mentioned, is not to design a new traffic control algorithm. However, it remains available in the appendix, for the reader to reference, and get a better understanding of the different types of control strategies, alongside examples of controllers that operate using these strategies. For the purposes of this chapter though, only the relevant definitions regarding these alternatives will be provided. Elaborations and references can be found in Appendix A.

Definitions:

Signal Group: A set of traffic light heads that output identical traffic light indications (i.e. colors) to permit the simultaneous flow of non-conflicting traffic streams, within one intersection, and accordingly prevent the flows from conflicting traffic streams, so as to avoid collisions at the intersection.

Fixed Controller: These are controllers range from the most naïve version, which outputs a single fixed green time duration that rotates amongst different signal groups, to more complicated control strategies that attempt to coordinate the signal schedules of subsequent intersections together, offline, in a matter that would result in network optimized control.

Traffic Actuated Controller: This type of control optimizes its schedules online, in response to detected vehicle counts in real time. Half-Fixed, Actuated and Adaptive traffic control strategies are all variations of this type of control, with the differences between them lying in their level of adaptability to arriving and departing flows. Half-Fixed controllers output a minimum green time that rotates in a cycle, then extend the green cycle for a certain signal group, if traffic has been detected at a related traffic stream, to clear a queue. Alternatively, both actuated and adaptive controllers can reshuffle their signal groups, to give the non-conflicting traffic streams with the most pressing demands for green a green phase as soon as possible. The difference between an actuated and an adaptive controller then, is that an adaptive controller takes in an extra input, through the existence of a far away loop, which allows it to further extend its green time, if needed, to account for incoming traffic, as shown in Figure 6.

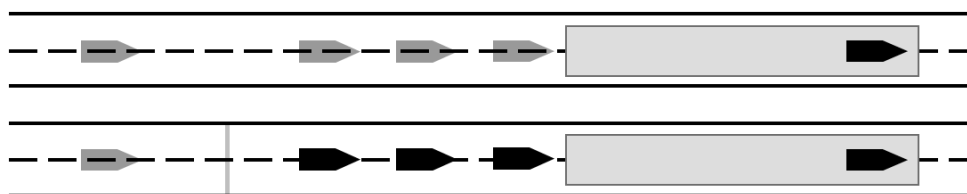


FIGURE 6: VISION OF AN ACTUATED CONTROLLER (TOP) VS. THE VISION OF AN ADAPTIVE CONTROLLER (BOTTOM)

Predictive Control: This type of controller’s line of vision extends much further back than an adaptive controller, in both space and time, allowing it to anticipate short term traffic flow arrivals, within a certain prediction horizon before their actual arrival times. By knowing what the future demands will be on all its traffic streams, this type of controller is capable further optimizing its schedules; and, accordingly, when several predictive controllers exist in a network, it is possible to create network optimized control strategies using the prediction abilities of the controllers for flow movements. However, these control strategies are additionally highly complex as they must include highly sophisticated validation and fail-safe processes to overcome the unreliability of short-term traffic flow prediction modeling, under certain traffic conditions.

After explaining the different possible control strategies, an understanding of the current state (i.e. the existing **Dutch traffic actuated control strategy**) is needed, so as to identify what changes can be made to it, to possibly upgrade it through a proposed concept design. In the Netherlands, controllers range from half-fixed to adaptive, depending on the Municipality or Province. However, in recent years, the trend has been moving towards adaptive control, as the older, less flexible, control models are being replaced. Accordingly, the focus of this study will be on Dutch adaptive controllers. A description of the quite sophisticated traffic control strategy followed by these controllers can be found in Appendix B. What’s most relevant for this study though is that these controllers are highly adaptive to incoming traffic flows, which are detected and counted through stop line detectors, long loop detectors and far away loops, as shown in Figure 7, every tenth of a second.

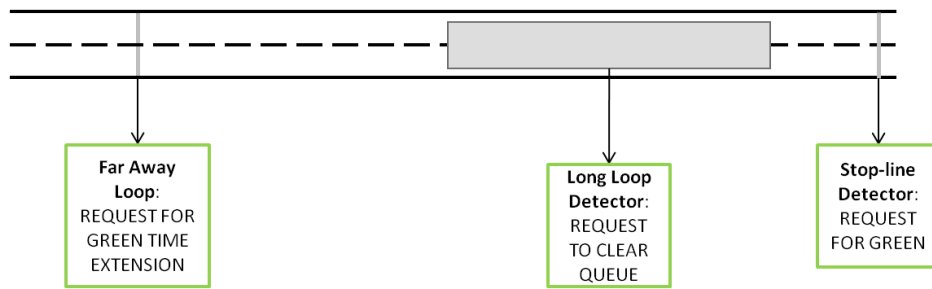


FIGURE 7: DETECTOR GEOMETRY FOR A DUTCH ADAPTIVE CONTROLLER

Selection of Alternatives:

There are essentially two possible directions to go when considering a new control strategy. The first is to go up the spectrum outlined in Figure 4 by restraining some of the controller’s adaptability. The assumption there would be that the benefits coming from T2G/R and/or GLOSA information would outweigh the lost efficiency. However, a study evaluating the trade-off between having fixed time settings with GLOSA usage, and having adaptive controllers without, showed that, with regards to travel time durations, and overall fuel consumption, adaptive controllers without GLOSA outperform fixed controllers accompanied with the use of GLOSA (Stevanovic et. al., 2013). Accordingly, it is counterproductive to make this recommendation. Hence, through a process of elimination, moving towards predictive control strategies became the more attractive option.

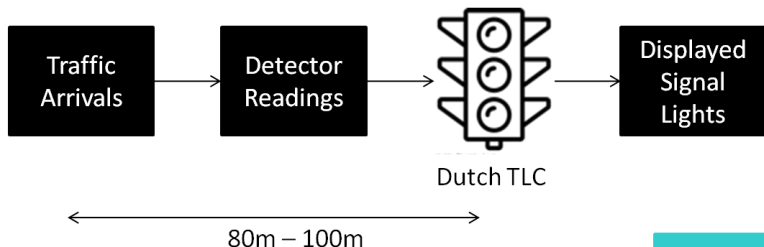
Design Specifications:

Recommending the purchase and installation of one of the mentioned predictive controllers in Appendix A is not feasible, as it would go against the user requirements and the cost constraint mentioned in Figure 5. Alternatively then, the same control concept is achieved through a manipulation of the inputs and outputs of the existing Dutch adaptive controller. The objective would then be to output the exiting controller’s schedule early enough for the operational usage of T2G/R and GLOSA, without actually having to change anything about the control logic itself. To assure that these predicted outputs match 100% the actual, they will be used in actuality as well. Accordingly, following the same methodology as SCOOT, the controller will operate in a simulated environment, responding to short term traffic flow arrival predictions; and, then its outputs will be lagged, for the duration of the prediction horizon, to be displayed at the appropriate time of vehicle arrivals in real space. This proposal is visualized in Figure 10. Instead of designing a new controller then, the suggested design is of two boxes: a predictor and a scheduler, as shown in Figure 8, with the new flow of information shown in Figure 9. The details of the design specifications go as follows:

- 1) **Input Data:** Since fluctuations in an adaptive controller’s schedule are a response to traffic flow arrivals and departures, traffic movements along an intersection can be forecasted early and distributed along detectors in a simulation environment of the isolated intersection in question. Like SCOOT, the traffic flow predictions will be in the short term, by forecasting the progression of traffic from an upstream to a downstream intersection. Using detected inputs from the stop line detectors of the directly upstream intersection, a prediction model can be used to forecast how vehicles will arrive at the far away loops of the target intersection. Hence the controller’s foresight will be extended to match that of T2G/R and GLOSA use cases. Through simulation modeling the predicted vehicles at the far away loop can be distributed out over the remaining detectors, triggering them, and therefore the controller, for an output. Hence, the form of inputs the controller will take will remain the exact same then, with the difference being that the detector triggers will be predicted and simulated in virtual space. Also like SCOOT, the actual arrival flows measured by the far away loop at the target intersection can be used to calibrate the prediction model and later to verify the prediction accuracies as well.
- 2) **Control Logic:** The controller will take the input from the simulated detectors and begin to respond as it normally does, providing fluctuating outputs until the controller decides to end a green phase and switch to a new traffic stream. With short term traffic arrival flow predictions, the controller will continue to be online optimized and reactive to relevant traffic conditions. However, the prediction horizon will allow service providers to have schedule information early, so that they can use it for T2G and GLOSA applications.

- 3) **Output Display:** Since controller outputs will be elicited early, they will need to be stored for the duration of the prediction horizon, then outputted at the correct time when vehicles will actually arrive in real time. This is the function of Box 2 in Figure 8. Additionally though, a new platform now exists to use this forecasted information for the proactive traffic management use cases.

Current Scenario:



Proposed Scenario:

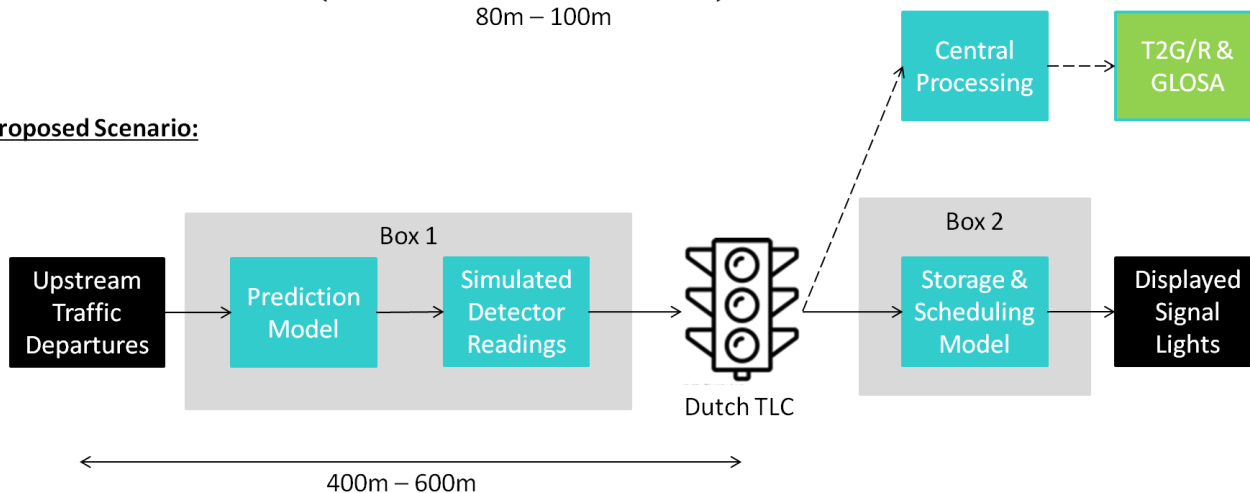


FIGURE 8: DIAGRAM SHOWING HOW THE PROPOSED PRODUCT WILL ONLY INTERFERE WITH THE INFORMATION FLOW OF THE CONTROLLER, BUT ALL OTHER ASPECTS WILL REMAIN THE SAME.

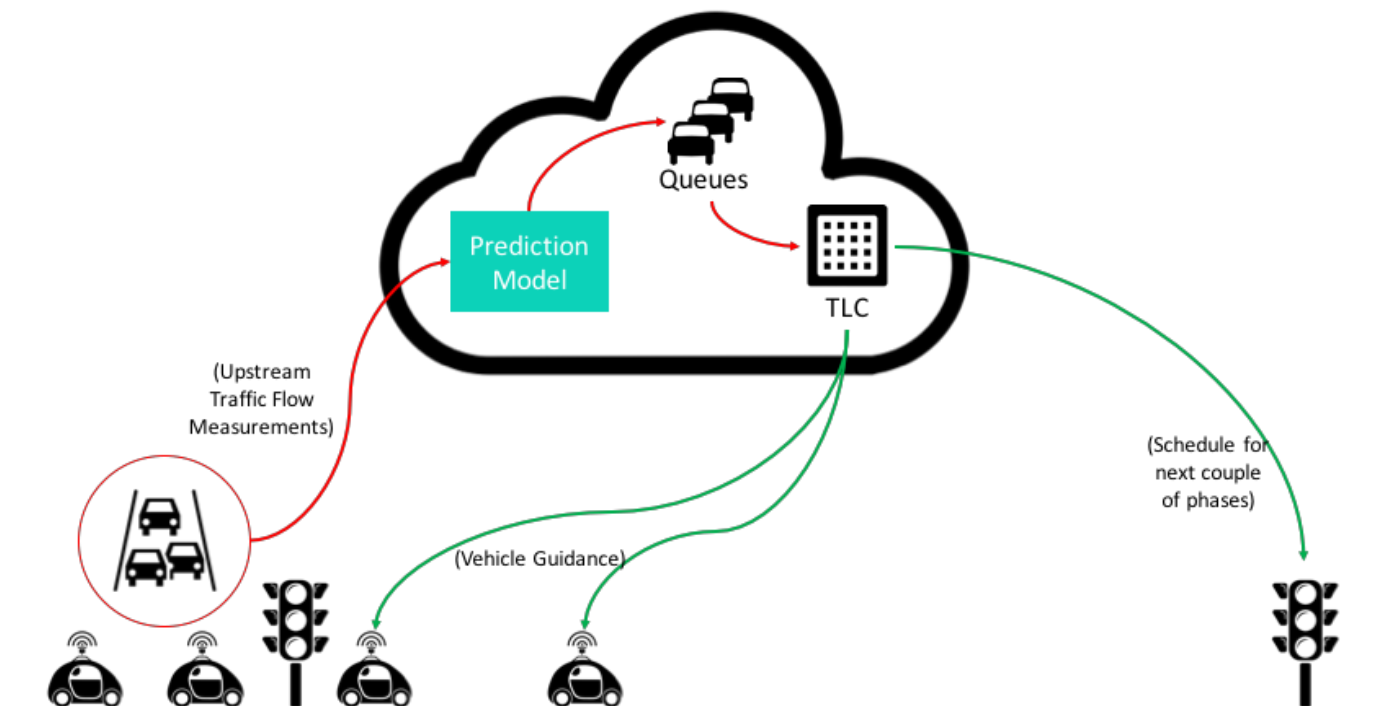


FIGURE 9: FLOW OF INFORMATION FOR INTERSECTION TRAFFIC LIGHT MANAGEMENT FROM THE CLOUD

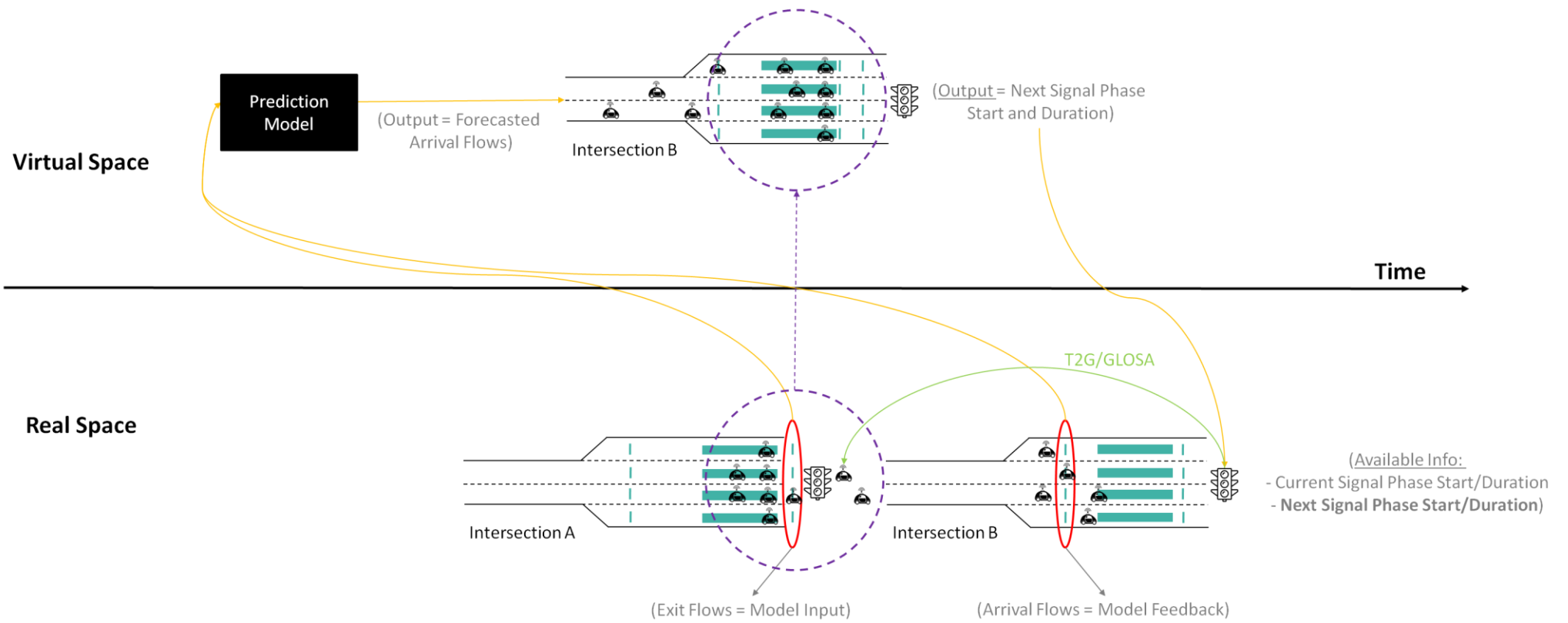


FIGURE 10: VISUALIZATION OF PREDICTIVE CONTROL PROPOSAL

With this solution, no expensive infrastructural modifications are required, users will experience the exact same control behavior they are used to, schedules will be outputted early to the service provider (i.e. better predictability), and, with short term traffic flow predictions, the controller will still be online adaptive, like SCOOT (i.e. same flexibility). Accordingly, the ideal controller point of the spectrum outlined in Figure 4 can be reached with this proposal, while satisfying both the user requirements and the defined design constraints. This also creates a good business opportunity, since these “boxes” can be sold to road owners to be interfaced with any controller, regardless of which company it was manufactured by, to upgrade it to be predictive. More so, since the traffic light controller will essentially be reacting to simulated states, the control software itself can be removed from the street and placed in the cloud, as shown in Figure 9.

1.3.2. MAIN ADVANTAGE OF PREDICTING TRAFFIC INSTEAD OF TRAFFIC LIGHTS

A lot of attention in literature is given to traffic light prediction modeling. With the trend in on board driver assistance use cases, predicting signal switch times and phase cycle durations has become a research interest as well. However, as was seen from the problem analysis, prediction models for green times can only give positive results if the controller’s adaptability is reduced. Additionally, the severity of the error risk is quite high. At best, inaccurate on board advice or predictions can cause traffic disturbances, as was seen in Stevanovic et. al. (2013)’s study, and at worst it could cause a traffic accident due to red light violations or sudden braking. To avoid this, a near 100% prediction accuracy of a controller’s schedule is needed for the safe and efficient use of T2G/R and GLOSA applications. Alternatively though, errors in the traffic prediction input of a controller, at worst would cause added delays. Accordingly, shifting the risk to this section reduces its severity. Moreover, there is a higher tolerance for controller input errors, than for errors in outputted information to the end user.

As previously mentioned, the key feature that differentiates adaptive controllers from actuated controllers is the existence of the far away loop, since it adds more flexibility to the control cycle. However, in some cases within the Netherlands, this long loop detector is placed before the road expands to several lanes, as shown in Figure 11. In these cases, the controller must additionally follow a form of prediction logic to determine which traffic stream the incoming vehicles will approach, to properly assign the extension of green (Blokpoel & Niebel, 2016). The prediction models used to estimate the turning percentages of detected vehicles, over time, are based on realized patterns from historical data, which is a methodology that results in an average prediction error of 2 vehicles (Blokpoel & Vreeswijk, 2016). Despite this error range, the added foresight benefits of this form of adaptive controllers have been found to still result in overall higher traffic throughputs at intersections than actuated controllers (Blokpoel & Niebel, 2016). Hence, even if the controller does not receive 100% accurately predicted input, additional foresight on incoming traffic still results in more optimized traffic management; meaning some of this error risk can be absorbed.

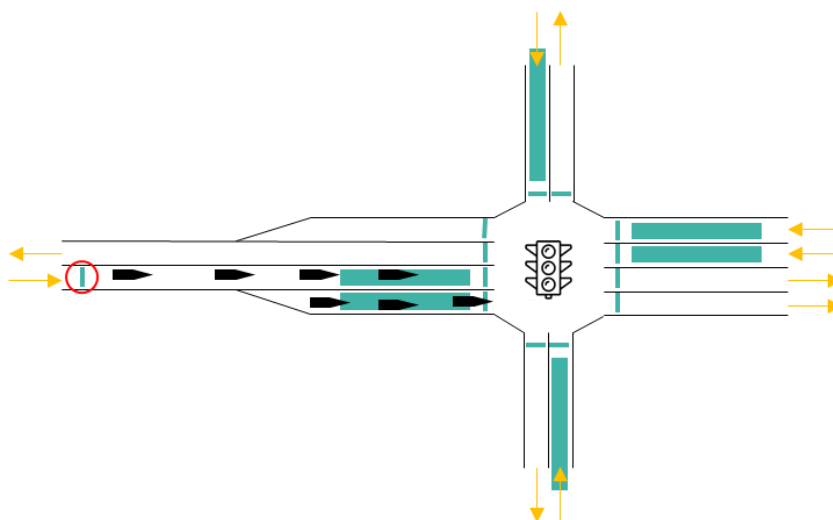


FIGURE 11: EXAMPLE INTERSECTION WHERE THE FAR AWAY LOOP IS PLACED PRIOR TO THE LANE EXPANSION

1.4. RESEARCH OBJECTIVES AND SCOPE

Now that the problem has been broken down, and a design objective has been identified to solve this problem, the next step is to define the research objective that needs to be reached to facilitate the realization and evaluation of the concept design. This includes formulating the research questions, identifying a means with which the results will be evaluated, and specifying the scope of works. The selected, lower level, research objective will serve as the initial step in realizing the predictive control design. From there, future work can proceed with completing the design, and taking the solution full circle, by using it for its intended purpose, which to solve the higher level problem introduced at the start of Chapter 1.2.

1.4.1. RESEARCH OBJECTIVES

In summary, the previous chapters take a large, multi-dimensional, problem and break it down until a single, lower level objective is reached, as shown in Figure 12. The highest level objective of this research is to find a means to output stable and usable T2G/R and GLOSA messages to end users for higher efficiency at intersections. However, the adaptive behavior of traffic light schedules makes it difficult to apply these use cases, due to their lack of predictability. The proposed predictive controller provides a possible solution to this problem; but, to achieve this type of control, a strong enough prediction model for short term traffic flow arrivals is needed that may be used to transform an adaptive controller into a predictive controller, without compromising its performance. Accordingly, finding a means with which to develop and evaluate a proposal for this model is the **research objective** of this study. As mentioned in Chapter 1.3.2., the advantage of predicting traffic, as an input for a controller, rather than predicting what a controller output will be, is that the prior method has a higher error tolerance than the latter. More so, the volatility of traffic flow arrivals is the central root of the problem.

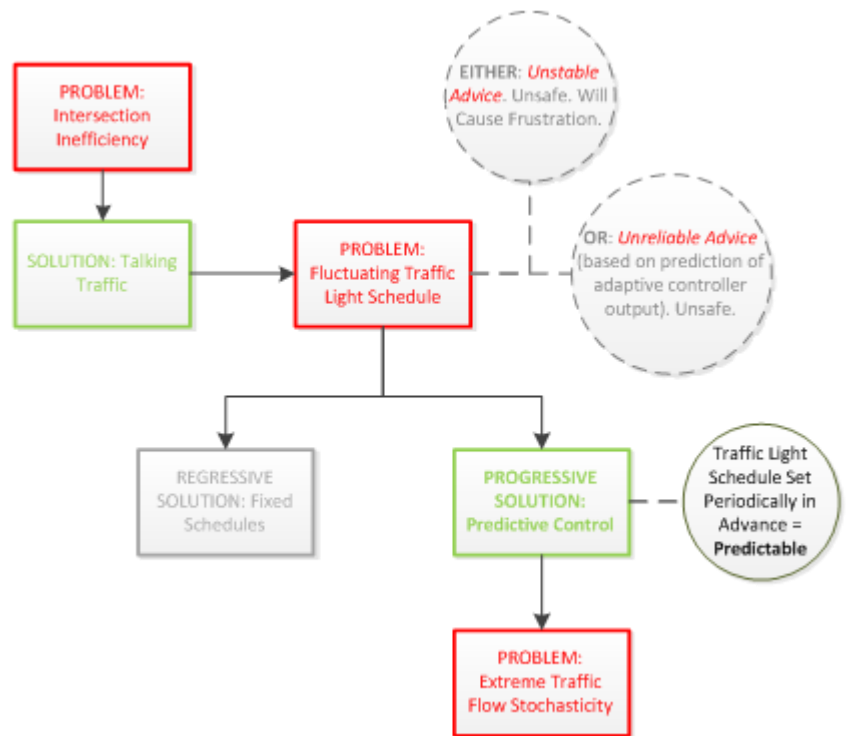


FIGURE 12: BREAKDOWN OF THE PROBLEM

1.4.2. RESEARCH QUESTION AND SUB-QUESTIONS

In order to reach the research goal of this study, and tie it to the larger research objective, the following research question, and proceeding sub-questions, were formulated:

To what extent can a short-term traffic flow prediction model facilitate the conversion of a Dutch traffic actuated controller into a predictive controller, without compromising its performance?

- *How accurately can a prediction model forecast short-term traffic flow arrivals, under what traffic conditions, and at what time/space horizon?*
- *At the maximum accuracy reached by the prediction model, how is traffic flow affected if the controller calculates and outputs its green/red phases in accordance to predicted flows rather than actual flows?*
- *When having a controller respond to predicted flows, what benefits can be achieved with regards to outputting more stable information with regards to a controller's traffic phase cycle?*

The scope of this project will be to build the prediction model, analyze its performance, and then determine how a controller that gets its outputs by reacting to predictions would perform. If a wider prediction horizon than one successive intersection is needed, this will part of the conclusions and suggestion for future works, but this will not be included within the scope. Additionally, the intersection geometry is simplified for this study to include no other modalities than vehicles. Lastly only detector data is considered for this research, due to its ready availability, in abundance, for analysis and modeling. The possible inclusion of other data sources, such as floating car data or trajectory data has not been explored in this study, as this type of data was not available, and the objective was to develop a ‘predictive controller’ that can be readily used with existing data sources. The purpose of this research report is to introduce this methodology for controlling Dutch urban intersections, and test to see if the concept design is feasible. Accordingly, the end of the scope will be to reach steps seven and eight from the Systems Engineering Report, as shown in Figure 13.

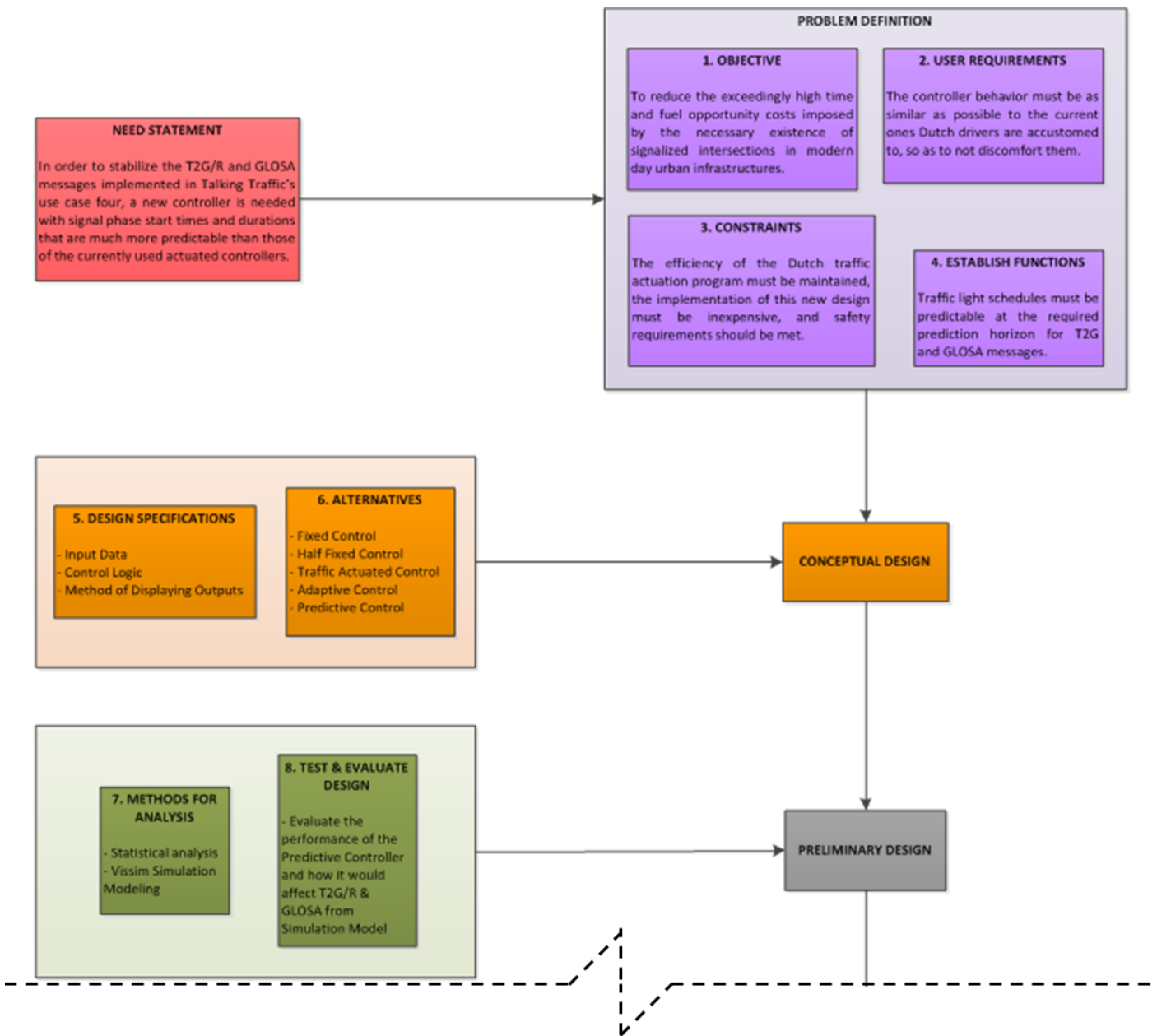


FIGURE 13: THE PART OF THE SYSTEMS ENGINEERING APPROACH INCLUDED WITHIN THE SCOPE (COLORED PART).

Within this chapter, the two levels of research approaches followed for this study are identified. On the conceptual level, the top level problem was broken down and a concept design proposal was made, while on the detailed level, a prediction model was designed and tested, to facilitate the development and feasibility testing of the concept design. As mentioned in Chapter 1.4.4., the focus of this research is to develop a prediction model, determine how it improves the conditions of the base case, and then evaluate its performance in its intended application. The intended application for the model is for its short-term traffic flow prediction outputs to be used as an input for the predictive control concept design proposed in Chapter 1.3. Accordingly, the last part of the model evaluation, before forming final conclusions, which corresponds to Chapter 6 in this report, is to determine how a controller performs when taking in this model’s outputs as an input. However, prior to testing the model for this purpose, it must initially go through an iterative design process, and then undergo several tests to determine how changes in its parameters affect its performance, and how the model’s performance is affected by different traffic conditions and different levels of traffic complexities. The purpose of these test was to provide a cohesive understanding of the model’s traffic flow prediction abilities, under different circumstances. Hence, as part of this testing process, three case studies were used: a synthetic case study, an urban corridor case study, and a full intersection case study. The reason was to slowly increase the complexity of the traffic data, and the geometry of the case study, for both gradual model calibration purposes and to better identify the types of impedances that, when existing, interfere with the model’s performance accuracy.

As shown in Figure 12, the development of this short term traffic flow prediction model presents the first, and most research intensive, step in solving the top level problem. The methodology followed for breaking down the multidimensional problem, to identify its root, and propose a solution to solve it can be found in Figure 14. The blue processes represent Chapter 1, the green Chapter 2, while the yellow are the remainder of the report, which is expanded in Figure 15. Hence, while the design, testing and application of the prediction model is the main focus of this report, it fits within a bigger framework. The predictive model can facilitate the predictive control concept design proposed, which can allow on-ground traffic management to be proactive, matching the proactive strategy of the GLOSA and T2G/Red use cases.

The colors of Figure 15 are also divided by chapters, where the orange is Chapter 3, the turquoise is Chapter 4, the purple is Chapter 5, the blue is Chapter 5, and the remaining boxes represent the conclusions, reflections and recommendations for future works, which can be found in the final chapters of this report. At the decision markers of the flow charts, the red flows represent “no”, while the green represent “yes”. There is one dashed red line in the flow chart to identify that this process would have been done if a “no” was faced, however during this particular research the answer was “yes” from the start. As will be seen, later, in Chapter 2, machine learning algorithms were selected for developing the prediction model, with Neural Networks (NN) being the algorithms of choice. The base case model is also defined in this chapter. These are the two models referenced in Figure 15

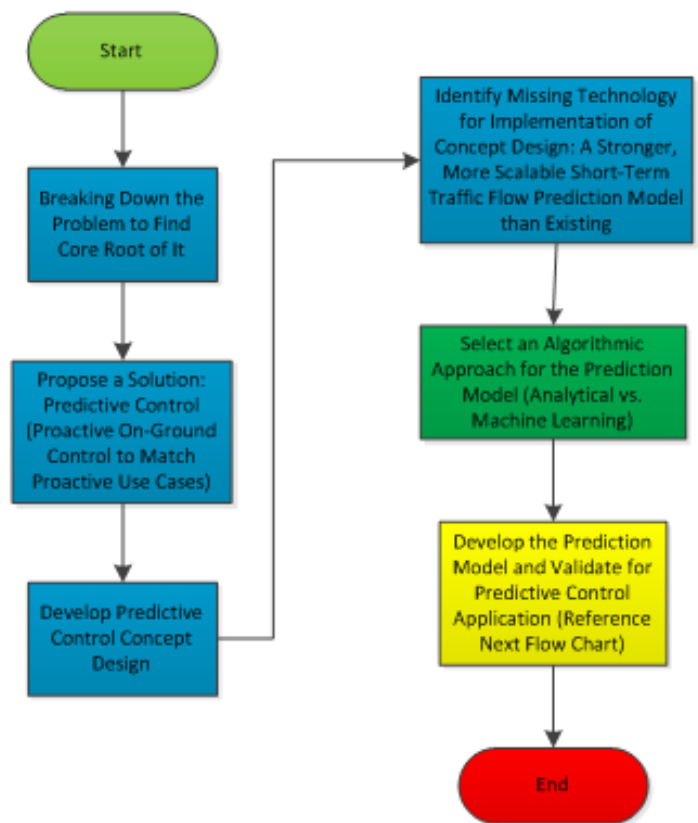


FIGURE 14: TOP LEVEL RESEARCH APPROACH: BREAKING DOWN THE PROBLEM TO FIND ITS ROOTS AND PROPOSE A SOLUTION



FIGURE 15: DETAILED LEVEL RESEARCH APPROACH: PROCESS FLOW CHART FOR PREDICTION MODEL RESEARCH

In summary, this project was divided into three divisions: 1) Defining a Concept Design to Solve the Top Level Problem; 2) Developing the Prediction Model; and 3) Testing the Concept of a Predictive Controller. Each division follows its own research and/or design methodology, and outputs results that need to be evaluated individually. Nonetheless, the completion of each division relies on the successful completion of its predecessor. Hence, the end date for each division will be considered as a project milestone. The breakdown of the tasks and the resources/tools used to complete the tasks can be found in Appendix C.

1.6. THESIS OVERVIEW

Now that the “why, what and how” questions of this research project have been answered, the outcomes of the research itself need to be presented. Accordingly, the remainder of this report will focus on the findings made throughout each phase of this iterative approach, until the final conclusions were reached. To facilitate the readability of this report though, an outline of its structure, in addition to the explicit identification of the scientific contribution, are, first, identified in this chapter.

1.6.1. OUTLINE OF REPORT STRUCTURE

There are several approaches that can be used to build a prediction model for traffic flow arrivals. Mainly, they can be clustered into two approaches: using analytical equations or using machine learning algorithms. Accordingly, Chapter 2 provides the literature review of the possible choices, and outlines the reasoning behind the selected approach. The conclusion of the chapter was that machine learning algorithms are the more appropriate approach for a predictive controller application due to their adaptability to different intersections, without requiring computationally exhaustive calibrations each time. Using a neural network was the selected machine learning algorithm, following the literature trends for short term traffic flow predictions. From there, a neural network needed to be built from scratch. Following the iterative approach outlined in Chapter 1.5., the initial neural network design was built, using a simple synthetic case study dataset and the help of the Stanford neural networks course [14], as shown in Chapter 3. The most significant outcome of this chapter is that it shows the process followed for reaching the selected type of neural network used for the prediction model. In Chapter 4, real data for a simple corridor case study, is used to verify the selected deep learning approach, and test the effect of the hyper-parameters on the model’s behavior. Due to configuration differences between the case study in Chapter 3 and the one in Chapter 4, some calibrations to the architecture are made. Similarly, some calibrations are made to fit the model input and output nodes to the configuration of the complex, full intersection case study of Chapter 5. In this chapter, the model is used to predict short term traffic flow arrivals for a full intersection, to evaluate its performance for its intended application. Despite the thorough evaluation of the model made in Chapter 5, the model needed to be applied in a predictive control setting, to complete its performance evaluation. Hence, in Chapter 6, the concept of a predictive controller is tested and evaluated, using the prediction results of the developed model. Within this chapter, the lower level research objectives are tied together with the higher level objective of this study, by providing some insight on how well the proposed predictive controller can facilitate the use of T2G/R and GLOSA applications, using the developed prediction model. Chapter 7 summarizes the overall conclusions of this study, and specifies the discovered answers to the research question and sub-questions; then, provides a reflection of the work, and, finally, gives some recommendations for future work.

1.6.2. SCIENTIFIC AND SOCIETAL CONTRIBUTION

Scientific: The main scientific contribution of this study is the development of the prediction model for short term traffic flow arrivals using a recurrent neural network, with memory delays and multi-task learning. To the extent of the knowledge of the author, this is the first time traffic flow has been predicted in this way. The approach taken for building the model was to develop it as a reconstruction of a widely used classical analytical model. In literature, only a single study was found that followed a similar approach to this (Qiao et. al., 2001); however, the parallels between their neural network and the classical model were not as explicit as they are in this study. Additionally, also to the extent of the knowledge of the author, this was the first study found where a short term traffic flow prediction model was evaluated for a full intersection application where adaptive control is applied, and where the controller responses to predicted inputs was included as a part of the evaluation. It is important to note however that the aim of this research was not to develop a new method of machine learning, but to design a neural network architecture that can be used for short term traffic flow forecasting applications.

Societal: The societal contribution of this work however, is that by using this prediction model for its intended purpose, which is as an input for a the predictive control concept design proposed in Chapter 1.3., it becomes possible to facilitate GLOSA and T2G/R use cases, which, in turn, can solve the top level problem of intersection inefficiency.

2. REQUIREMENTS AND APPROACH FOR A SHORT-TERM TRAFFIC FLOW PROGRESSION MODEL

This chapter begins by defining some relevant concepts regarding the short term traffic flow prediction modeling that will be developed in this study. It then provides some background information on the origins of traffic flow prediction modeling, to introduce the classical traffic flow algorithms that have been used as the skeleton of most short-term traffic flow prediction models in modern day. The underlying theory behind these models is that traffic progresses from one point along a corridor to the next following a particular platoon dispersion pattern. Next, within Chapter 2.2. the most superior classical progression model is further explained, and the updates that have been made to it over the years are outlined. This is followed, in Chapter 2.3., with alternative machine learning methods that have been used to model traffic flow progression as well; and with an overview, in Chapter 2.4., of how the classical model had been used for predictive control applications. From these three chapters, the limitations of the classical model, even after it has been updated, are more clearly apparent, and the need for exploring a new traffic flow progression model approach, using neural networks is established. By Chapter 2.6., a reference case is identified as the benchmark, from which the prediction model performance is evaluated. Finally, in Chapter 2.7., the conclusions of Chapter 2, are made.

2.1. INTRODUCTION

Chapter 1.4.1. identifies a need for a short term traffic flow prediction model that robust to different traffic conditions and lane configurations. Accordingly, a literature investigation is needed on the research conducted on short term traffic flow prediction modeling. For more accurate predictions, the proposed predictive controller design makes use of inputs from the exiting flows detected on the stop lines of upstream intersections. The method of predicting downstream platoon flow rates based on upstream platoon flow measurements is defined as **traffic flow progression modeling**. With this type of short term traffic flow prediction, the challenge is not to predict a pattern in vehicle arrivals over time, but to forecast how measured vehicle platoons will move in both space and time, depending on the traffic conditions. This phenomenon is referred to as **platoon dispersion**.

There are two dimensions to platoon dispersion: lateral, which refers to the split of platoons amongst the different traffic streams at an intersection, and longitudinal, which refers to the spread of vehicle headways in a platoon, due to varying driver speeds. Lateral platoon dispersion depends on the movement direction vehicles in a platoon intend to go through, which can be estimated through probability distributions of origin-destination (O-D) pairs. This lateral spread of vehicles impacts longitudinal dispersion, which is a function of vehicle interactions, roadway geometries, roadside activities, different driving habits, downstream signal states, and other impedances to the flow of traffic (Paul et. al., 2016)(Geroliminis & Skabardonis, 2005). Due to these disturbances, flows released from upstream intersection stop lines in tight platoons, with short time headways, spread out at seemingly stochastic rates. Accordingly, longitudinal dispersion is the main focus of traffic flow progression modeling, and lateral dispersion is included as a model input parameter, or expansion. Hence, this chapter investigates the related works with regards to longitudinal platoon progression, and some means with which lateral progression is also incorporated in.

2.2. THE MAIN CONTRIBUTIONS TO ANALYTICAL TRAFFIC FLOW PROGRESSION MODELING

There are three major analytical contributions made to traffic flow progression research: Lighthill and Witham's kinematic wave theory, Pacey's diffusion model and Robertson's recursive traffic flow progression model. The Lighthill and Witham model uses shockwave theory to illustrate the changes in traffic states upstream a signalized intersection and describe the platoon behavior on a signalized link. On the bases of this model, Head (1995) and Geroliminis & Skbarodinis (2005) developed their own traffic flow progression models, for simulating the movement of traffic through an urban network. The main advantage of both models lies within their ability to describe the spatio-temporal effect of traffic light signal timings on the flow of vehicles, using a Markovian process, which allows them to forecast traffic flow behavior over multiple intersections. However, shockwave theory relies on the assumptions that vehicles are conserved and that traffic behaves according to a fundamental diagram. Fundamental diagrams rely on the assumptions that traffic is homogeneous (i.e. traffic states do not change over space) and is stationary (i.e. traffic states do not change over time). Accordingly, while traffic flow progression models based on Lighthill and Witham's

theories provide sufficient interpretability of traffic flow behavior, and are quite useful for modeling applications, they are not accurate enough for real life application purposes. By following the fundamental diagram assumption, these models ignore the concept of longitudinal platoon dispersion, and, therefore, result in ineffective signal timing plans (Bhaskar et. al., 2016).

Improving upon Lighthill and Witham’s model, Pacey’s model was developed to overcome the limitations of fundamental diagrams. This model assumes platoons progress along a link with speed variations that follow a normal distribution and with unrestricted overtaking conditions (Bhaskar et. al., 2016). Accordingly, its downstream short term traffic flow predictions are closer to reality than Lighthill and Witham’s model. However, the model that showed the most agreement with real data has been Robertson’s recursive traffic flow progression model, which has been designed as a modification to Pacey’s model. It was derived from empirical data, with the specific aim of more realistically forecasting arrival profiles, to facilitate traffic signal synchronization (Qiao et. al., 2001). The fundamental difference between the approach of both models is that Robertson’s model uses a geometric distribution, rather than a normal distribution, to describe the probability distribution of vehicle travel times in a platoon. Since the tail of this distribution is longer than a transformed normal distribution, Robertson’s model predicts higher platoon dispersions than Pacey’s, for any given average platoon travel time (Geroliminis & Skabardonis, 2005). Moreover, it results in significantly lower computation times (Geroliminis & Skabardonis, 2005). The formulas for this model are as follows:

Robertson (1969):

$$q_d(t_n) = F * q_u(t_n - \beta T) + (1 - F) * q_d(t_n - 1) \dots\dots\dots(1)$$

$$F = \frac{1}{1 + \alpha \beta T} \dots\dots\dots(2)$$

In his model, Robertson describes downstream arrival flows q_d (where t_n is the modeling time step) as a function of upstream departure flows q_u (from βT time steps ago, where β is a unit-less travel time factor and T is the travel time between the upstream and downstream intersections), a smoothing factor F , and the remainder of downstream flow from one time step back. In the equation for the smoothing factor, α is a unit-less platoon dispersion factor. However, for Equation (1) to be used to predict flows for a certain time interval, it must be applied recursively. Accordingly, Seddon (Rumsey & Hartley, 1972) reinterpreted the model as shown in Equation (3), which calculates the downstream flow profile $q_d(t_n)$, based on the sum of incoming upstream flows $q_u(t_i)$ within a certain time interval. With these limits, only the previous platoons who are most likely to have vehicles still within the link are considered to contribute to the arrival flows at a certain time step.

$$q_d(t_n) = \sum_{t_i=1}^{t_n-\beta T} q_u(t_i) * F * (1 - F)^{t_n-\beta T-t_i} \dots\dots\dots(3)$$

Due to its strong ability to capture and replicate platoon dispersion along a link (specifically under free-flow conditions), Robertson’s model has been vastly used in many state of the art predictive controllers such as SCOOT, SATURN, TRAFLO, UTOPIA, and OPAC, and offline optimized controllers such as TRANSYT (Qiao et. al., 2001) (Geroliminis & Skabardonis, 2005) (Bie et. al., 2014) (Bhaskar et. al., 2016). However, the accuracy, and, hence, most successful application, of Robertson’s model is highly dependent on the proper calibration of its time and dispersion parameters (Mathew, 2014). Accordingly, advancements to Robertson’s platoon dispersion model have spanned into three separate directions: the development of some default values for the dispersion parameters that are more easily transferable to different case studies, the development of a general methodology for model calibrations, and research on case-specific calibrations that may achieve maximum model accuracy. The selection of a research track depends on the intended purpose of the model application: whether it’s to make the model’s application more adaptable to different intersections, or to customize the model to specific traffic conditions (e.g. those of third world countries (Bhaskar et. al., 2016)). The prior objective is more applicable for this study, and so the first two tracks are explored further.

In the United Kingdom (UK) and the United States (US) some studies have been carried out in order to set some default parameters for the application of Robertson’s model in predictive controllers, to facilitate its usability (Yu, 2000). For simulation modeling applications, most commonly, β is recommended as 0.8, while the value for α

ranges from 0.25 (for tight platoons) to 0.5 (for highly dispersed platoons), and is generally taken as 0.35 (Geroliminis & Skabardonis, 2005). However, assigning set values to these parameters has been proven to be unsuitable for predicting actual arrival profiles for downstream intersections, since, more often than not, actual traffic conditions deviate from the initially assumed state (Bie et. al., 2014). The methodology of fixing the travel time value to 0.8 and calibrating the platoon dispersion value is maintained for various suggested calibration processes though, since the simultaneous calibration of both α and β parameters has been found to be quite computationally intensive (Manar & Baass, 1996) (Bie et. al., 2013) (Bie et. al., 2014) (Bhaskar et. al., 2016). Following this approach Manar & Baass (1996) proposed a parabolic model to dynamically calibrate α in correlation with traffic volumes, in order to capture the ever-changing nature of traffic conditions. While this model improves upon the use of standard values, it is not applicable for non-two lane roads since the relationship between the α parameter and traffic volumes has been found to be sensitive to the number of lanes on a road segment (Bie et. al., 2013). This revelation served as a basis for Bie et. al. (2014)'s model, which considers the expansion of lanes at intersection entrances (for increased capacity) by splitting the vehicle flows into their corresponding traffic streams, and recalibrating α for each direction of movement post the expansion. This follows Yu & Recker (2006)'s proposal to expand Robertson's model to include origin-destination (O-D) pair weights for traffic flows heading in different directions. The portion of vehicles that will head to each traffic stream direction is estimated based on historical empirical data (Bie et. al., 2014).

From a different perspective, Farzaneh and Rakka (2006)'s study concurred that, without the calibration of β , signal timing plans would still be inefficient using Robertson's model, and that the value of α was not as significant in affecting outputted traffic light schedules. Hence, a tradeoff can be seen when developing a methodology for the calibration of Robertson's α and β parameters between computational efficiency and accuracy. Earlier studies, such as that of Yu (2000), have proposed a means for the model be alternatively calibrated in accordance to the standard deviation σ of a platoon's mean travel time on a link, since platoon dispersion is essentially the variation of speeds (and therefore travel times) of vehicles within a platoon. Through the use of Equation (4), the smoothing factor F can be calibrated, depending on the standard deviation of a platoon's mean travel time and the modeling time step size n . When applied into Seddon's restructuring of Robertson's model (which only uses the β parameter to define the time limits within which vehicles from previous platoons will be considered), this equation serves as an implicit simultaneous calibration of α and β . However, its disadvantage is that it is sensitive to the modeling step size, and this more generalized approach does not leave room for calibrating the model to specific impedances.

$$F_n = n * \left(\frac{\sqrt{n^2 + 4\sigma^2} - n}{2\sigma^2} \right) \dots \dots \dots (4)$$

The comparisons of these studies reveals a third dimension to the trade-off: scalability. The limitation of Manar & Baass (1996)'s study was that their model was not scalable to different road segment configurations. However, their results showed that when models overlook the effect traffic flow volume fluctuations have on platoon dispersion patterns, Robertson's model's predictions result in more frequent delays and stops when used as inputs for a predictive controller (Manar & Baass, 1996). Similarly, Paul et. al. (2016) developed an equation for calculating a value for α that takes into consideration the traffic composition and the existence of queues at downstream intersections, which are two of the biggest limitations to Robertson's un-calibrated model (Geroliminis & Skabardonis, 2005), that is only applicable for single lane road segments. The reason why these two models are not scalable is that combination of impedances on traffic flow progression (e.g. queuing and road segment width) has a non-linearly added affect on platoon dispersion patterns.

Due to this difficulty in effectively combining the effect of multiple impedances, each study chooses to calibrate one, or both, parameters in Robertson's equation based on what the researchers perceive will have the strongest affect on arrival profiles. This includes either driver behaviors (lane change, car following, etc.), traffic volume, the physical properties of the road, roadside activities (parking, pedestrians, etc.), types of vehicle classes, or downstream signal schedules. In actuality though, ALL of these factors, in addition to other non-observable ones, have an impact on platoon dispersion. Additionally, despite all the explored updates made to Robertson's model (which mainly revolve around its effective parameter calibrations), the model is still constrained by the two assumptions that: a) each vehicle in a platoon maintains constant velocity while traveling along a link, and b) between upstream departures and downstream arrivals there is traffic flow conservation (Bie et. al., 2014). In reality however, driver behavior is unstable, and sources and sinks can be found on urban roads (Head, 1995). Lastly, no methodology has yet to be proposed for adapting the model to non-homogeneous saturated flow conditions (Bhaskar et. al., 2016).

The nonlinear behavior of short term traffic flow progression makes it an interesting application for machine learning algorithms. However, research on the use of machine learning algorithms for traffic flow forecasts is less prominent than that available for calibrating Robertson’s model. Nonetheless, the outcomes of the few studies found suggest that this approach is worth exploring as well. Lv. et. al. (2015)’s study, for instance, found that by using a neural network configuration with stacked autoencoders, they were able to build a prediction model capable of forecasting traffic flows on a mesoscopic level, in a network, with over 90% accuracy, using detector count inputs. The model was trained and tested using historical detector count data, collected over several years, and is capable of satisfactorily forecasting traffic flow in a network for 15, 30 and 60minutue intervals due to its ability to capture non-linear spacio-temporal correlations in traffic data (Lv et. al., 2015). While their study was conducted on a much more aggregated level than needed for this one, it still reveals the potential of using deep learning techniques for traffic flow forecasts. However, on an urban link level, there are three main works using machine learning algorithms that are most applicable to this study, as outlined below:

Short Term Traffic Flow Progression Modeling using a NARX Neural Network (Qiao et. al.,2001):

Qiao et. al. (2001)’s study on the use of Neural Networks for simulating platoon dispersion, and predicting arrival rates at downstream intersections, was designed as an alternative for classical platoon dispersion modeling, along a double lane urban link. The biggest obstacle associated with using machine learning algorithms to formulate traffic progression models is the difficulty in quantifying the relevant parameters for achieving the required degree of approximation (Qiao et. al., 2001). Hence, the most interesting part of Qiao et. al. (2001)’s study was that it recognized the strength and weaknesses of Robertson’s model, and proposed a neural network based model that was set up as a restructuring of this classical approach rather than an entirely new replacement (Qiao et. al., 2001). This is evident in the model’s information set displayed in Equation (5).

$$Z^N = [y(t - 1), \dots, y(t - n_y); u_1(t - k), \dots, u_1(t - k - n_1); u_2(t - k), \dots, u_2(t - k - n_2)] \dots\dots(5)$$

For this dataset (Z^N), which is used as input for the Neural Network, u_1 and u_2 are upstream flow and speed respectively, y is the downstream flow, n_1 , n_2 and n_y are the number of time steps back, and k is the detection time lag, which is considered as 0 for loop detectors. Hence, similarly to Robertson’s model, Qiao et. al. (2001)’s neural network based prediction model outputs downstream flows by taking into consideration the upstream flows, downstream flows (from one time step back), and platoon speeds recursively for a two lane road segment.

Unlike analytical models though, with a black box model structure, such as a neural network, the most crucial contribution for the successful design of the model is the proper set up of the network architecture. The model architecture within which these variables were applied is a nonlinear autoregressive network with exogenous inputs (i.e. NARX network), which can be visualized in Figure 16. This is a special neural network time-series model that not only searches for a time-series pattern in data, but additionally recognizes a relationship with external data points. These architectural features are what programmed the model to simultaneously process upstream flow information (i.e. external input) and downstream flow information from $y(t - 1)$ (i.e. feedback input), similarly to Equation (1), while making its prediction for the next time step. In order to avoid an exponential growth in output error with each output though, open-loop feedback was used, allowing detector-measured arrival flows, at the time of the prediction horizon, be inserted as inputs instead, following similar a logic to model predictive control. The main advantage of autoregressive networks though, is that they have memory cells embedded into them that allow for deeper learning. Accordingly, these models can find relationships between variables within several points in time, rather than from one point in time to the next. This is similar to Seddon’s restructuring of Robertson’s platoon dispersion model into a recursive model that recognizes the relationship between platoon departures, at several points in time, and platoon arrivals, that decreases geometrically the farther back in time (starting from the prediction horizon) the platoon had departed.

The outputs of the prediction model were compared to arrival rates detected by historical data, revealing the model had an average error of 1.5 vehicles (Qiao et. al, 2001), for each five second interval. When the same test was conducted using non-calibrated versions of Pacey and Robertson’s models, their outputs were quite similar to each other, and had larger errors with wider variances (Qiao et. al, 2001). While these results are quite promising though, the case study used continues to remain too simplistic for application of a predictive controller. The road

configuration remains constant at two lanes for the entirety of their selected case-study road segment, and the distance between the upstream and downstream points of measurement is 86m, which is quite short. Using a 5s time-step, this model is able to provide a highly reliable simulation for traffic progression, but not nearly enough time in advance for predictive control to be possible. With an average travel time of 50kph, the first vehicle in the platoon would have arrived to the downstream intersection by the time the 5s time-step duration for flow measurement is complete. Hence, the validity of this model for farther away intersections (i.e. longer space/time horizons) must be evaluated.

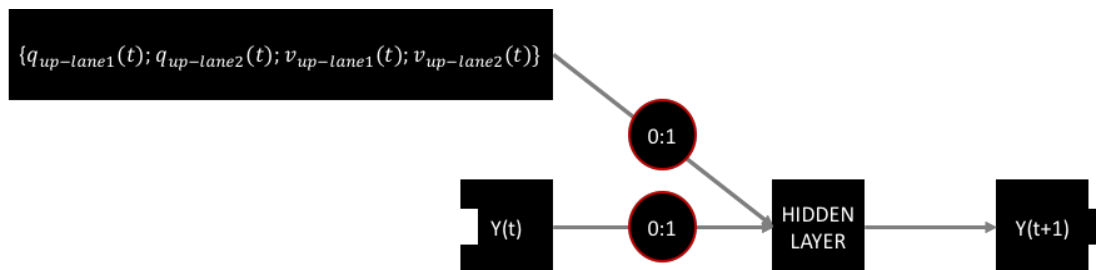


FIGURE 16: QIAO ET. AL (2001)'S NARX NEURAL NETWORK ARCHITECTURE

Turning Percentages Predictions Using a Feed Forward Neural Network (Jiao et. al., 2014):

One aspect of traffic flow prediction that was not captured in Qiao et. al. (2001)'s model is the percentage of vehicles in each direction of flow. Fortunately, this was the focus of Jiao et. al. (2014)'s research. On an urban link level, Jiao et. al. (2014), explored the use of a Kalman Filter and a neural network with Back Propagation to predict the lane direction vehicles would stand in at downstream intersections (i.e. for turning left, turning right, or traveling forward) using upstream detector count inputs. The results of their experiments showed that their machine learning models were capable of forecasting direction movement percentages of traffic flow at 85-89% accuracy, with slightly better results when using the Kalman Filter (Jiao et. al., 2014). The majority of found studies on short term traffic flow progression modeling using machine learning chose to use neural network algorithms though. Hence, to unify the algorithmic choice for this modeling approach, more focus will be placed on Jiao et. al. (2014)'s neural network model.

In their study, they developed a feed-forward neural network, designed to determine the percentage of vehicles that will move in each of the three possible directions of flow (forward, turn left, turn right), using upstream, lane-specific, detector counts, as shown in Figure 17. However, their model did not focus on making short term predictions on traffic progression, but rather whether or not a relationship exists between a time series of upstream lane occupancy and a time series of downstream turning proportions. Accordingly, their input and output data points were aggregated to 3minute time steps. Using only 2hours of data for training their model was still able to capture a relationship between traffic flow detector counts and turning percentages. However, although, for their case study, drivers were permitted to change lanes after their vehicle measuring point, the road configuration of their case study may allow for a stronger relationship between lane-specific detector counts and turning proportions than for a 2 lane road configuration which expands to four (or more) lanes downstream. Moreover, due to the aggregated precision of this model, it still is not quite suitable, as is, for the purposes of this report. Nonetheless, it does provide some information on how to approach the issue of turning percentages at downstream intersections.

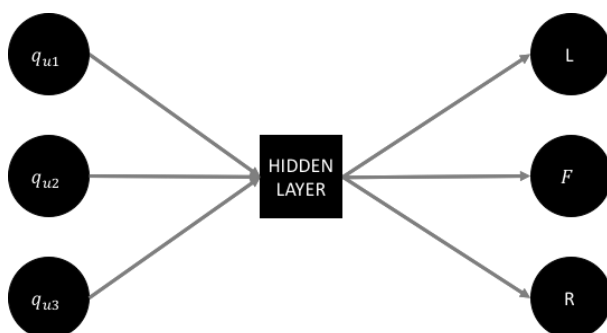


FIGURE 17: THE FEED FORWARD NEURAL NETWORK ARCHITECTURE OF JIAO ET. AL. (2014)'S MODEL, WHICH TAKES DETECTOR DATA AS INPUT AND OUTPUTS PERCENTAGE DIRECTION OF FLOW.

Traffic Intensity Predictions Using Time Recurrent Neural Networks (de Clercq, 2016):

A previous student researcher, Koen de Clercq researched the possibility of being able to predict traffic intensities, at a single intersection, by setting-up and training his neural network to detect a time-series relationship for changes in these traffic intensities (at his case-study intersection) over time. The type of network he used was a non-linear autoregressive neural network (i.e. NAR network), with an open-loop feedback. As shown in Figure 18, open loop feedback means that the input fed back into the model is not the output of the previous prediction, but the actual measurement at the time-horizon of the prediction (i.e. when it happens). This form of time-series set up allows the neural network to recognize the patterns in a dataset over time. Similarly to Qiao et. al.(2001)'s model, memory cells can be included in this type of autoregressive model as well, for deeper learning, helping the model to recognize the changing trends of a time-series dataset with time.

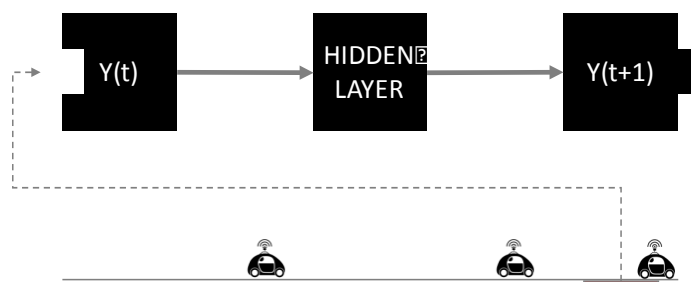


FIGURE 18: A VISUAL REPRESENTATION OF A NAR NETWORK WITH OPEN LOOP FEEDBACK.

De Clercq (2016)'s objective was to build a model capable of recognizing the trends of detected traffic intensities on a stop line detector, using a seven minute time step. A seven minute time step is too long for the purposes of short-term traffic reconstruction modeling, however the results do give some insights on the level of stochasticity within detector data. After inserting memory cells that go up to 40 time steps back, the NAR model's prediction accuracy was estimated at 80%, however no information was provided on how this percentage was estimated. Considering the aggregation level of the time steps, in comparison to the ones used for this study, these results were not too comforting. As previously mentioned though, traffic flow arrivals are not completely isolated occurrences, that follow a pattern over time. They originate from origin points and move forward. Accordingly, ignoring the flow rates of upstream traffic is likely to result in a prediction model that is highly sensitive to changes in incoming traffic.

2.4. TRAFFIC FLOW PREDICTIONS AND SCOOT CONTROLLERS

As mentioned in Chapter 2.2., Robertson's traffic flow progression model is applied in various state of the art predictive controllers. Despite its limitations, (when its parameters have been calibrated) it has been found to be, thus far, the most accurate, analytical, longitudinal platoon dispersion model developed. From the investigation on the different types of existing controllers, conducted in Appendix A, it appears the predictive controller following the closest methodology to the one proposed in Chapter 1.3. is SCOOT. It similarly operates using online predictions on incoming vehicle arrivals, based on detected vehicle departures from directly upstream intersections. It also relies on the arrival flow readings of far away loop detectors for prediction model parameter calibrations, and to verify the model's arrival flow predictions during operation (Siemens Mobility, Traffic Solutions UTC System, 2016). However, the practical application of Robertson's model in SCOOT does not mean that its limitations go unnoticed. Firstly, to maximize the accuracy of the prediction model outputs, the far away loops of SCOOT controllers are placed at locations far enough back so that traffic flow arrival profiles measured at that point are least disturbed by queues. This distance is constrained by the criteria that these loops must be close enough to the stop line so that drivers are unlikely to make a lane change or deviate their driving behavior in between. Hence, it is normally 120-150m (Siemens Mobility, Traffic Solutions UTC System, 2016). Additionally, to ensure its continuous successful operational efficiency, a SCOOT controller requires a sophisticated traffic flow prediction verification and correction process (Siemens Mobility, Traffic Solutions UTC System, 2016). In the case where queues are long enough to reach the far away detector, but not yet cover it, there are large detected prediction errors that the SCOOT controller needs to instate real-time corrective measures for. Moreover, once the far away loops have been occupied by a long queue, the SCOOT controller is no longer able to verify predictions, so instates a "system failure" and automatically resorts to a fallback control procedure until the queue has cleared (Siemens Mobility, Traffic Solutions UTC System, 2016).

While abiding by the needs and constraints outlined in Figure 5 for the proposed predictive control concept design, it is not possible to directly adapt the short term traffic flow prediction methodology applied in the SCOOT controller. Dutch adaptive controllers do not follow the same requirements for the locations of their far away loops. For these adaptive controllers, far away loops are placed at locations where detected vehicles can be considered as an extension of the platoon receiving a green wave. This is normally 80-100m upstream the stop line. At this shorter distance though, it is more common, during high density time periods, for queues to propagate back to, or past, these far away loops. Under these conditions, the controller will need to resort to harsh corrective measures and “system failure” procedures more often; both of which are counterproductive to the objective of this study. Sudden (and large) real time corrective measures to the current phase state duration of a controller will make the controller outputs less predictable. Similarly, a repetitive fall back to the fail safe method of control will either compromise the controller’s predictability or efficiency, depending on what fall back control system is implemented. Accordingly, especially considering the locations of installed loop detectors in the Netherlands, there is a need for a short term traffic flow prediction model that is more robust to different traffic conditions and queue lengths.

This gap resides within the literature study as well. From Chapter 1.2. and Chapter 1.3., it appears that Robertson’s platoon dispersion model remains the most advanced analytical traffic flow progression model available, and that its potential alternative is a neural networks based model. The common motif with regards to the advancements made using either modeling approach though, is the vast majority of studies set their focus on one specific aspect of platoon dispersion rather than aiming to develop a single model that encompasses all road impedances influencing the flow of traffic. This issue is more prominent for the updates made to Robertson’s model, due to the computational intensity of developing a calibration methodology for the model parameters that encompasses several types of influences (that are not additive) on platoon dispersion. On the other hand, the nonlinearity of a neural network makes it more equipped to simultaneously consider the effect of several impedances. In fact, the black box characteristic of a neural network makes it, alternatively, difficult to determine the individual effects of different lane compositions or traffic conditions on the progression of vehicles. Hence, with regards to the literature found using a neural network for short term traffic flow forecasting, this gap seems to be a result of a discrepancy between the research objectives of previous works and the application requirements for a predictive controller rather than a limitation to a neural network’s abilities. This discrepancy may have also been the cause for the scarcity in amount of sources found, regarding neural networks, that were directly relevant to the purposes of this study.

To fully bridge the gap between theory and application requirements, the characteristics of the needed short term traffic flow progression model need to be outlined. Revisiting the overall objective of this study, the intention is not to design a new controller to improve traffic flow management, but to add a predictive feature to the currently used control strategy, without requiring any intrusive infrastructural modifications (i.e. changes in detector locations or replacement of installed controllers). Hence, what is needed is a prediction model that can work with the standard infrastructural composition currently placed on Dutch urban roads (including the corresponding traffic and queuing conditions). With that, the following objective, requirements and functionality for the prediction model are defined:

Prediction Model Objective: The specific prediction model objective, is to predict the arrival flows at the entrance to an urban intersection, right after the lane configuration has expanded for increased capacity. Predicted flow arrivals must be specific for each individual signal head. This way, forecasted traffic streams can be distributed, in simulation, over the relevant virtual detectors, triggering controllers to optimize the schedules of these signal heads. The schedule outputs can then be used for proactive traffic management applications (e.g. T2G/R and GLOSA use cases).

Prediction Model Requirements: Scalability¹ is quite an important feature for this model. It should be adaptive to different case study locations, with different lane configurations, without requiring computationally intensive or sophisticated calibrations. Otherwise it is not practical for wide implementation. Moreover, this model must be

¹ Scalability is defined as the ability of the model to adapt its parameters to different intersection configurations (i.e. different number of lanes or different locations for the far away loops).

additionally adaptive to different traffic conditions. The model should be robust² to different levels of lane saturations, and different types of driver behaviors. The latter is especially important since the implementation of GLOSA or T2G/R would mean that driving profiles along an urban corridor will differ from what it currently is now.

Prediction Model Functionality: The intended functionality of this model is for it to measure traffic flows at one point, and then predict how this flow will progress and arrive at another point, downstream. Hence, the model that will be developed for this study will be required to take inputs from upstream stop line detector counts and predict how these measured flows will arrive on top of the existing downstream far away loops. Accordingly, it can be considered as a traffic flow progression model, or a traffic reconstruction model.

Both the objective and model functionality resemble those of state of the art predictive controllers, and can be satisfied by any of the mentioned updates to Robertson’s model. However, the requirements for joint scalability to different lane compositions (and detector distances), and adaptability to changing traffic conditions, is where the added value for developing the model using a neural network can be seen. Although the interpretability of a neural network’s behavior is quite low and it requires an excessive amount of data to train, its characteristics better fit the requirements. With enough data, a neural network can be trained to adapt to any non-stochastic data trend, and its parameters can be scaled to any intersection composition by changing the numbers of input and output nodes, provided the relevant upstream and downstream detectors are available. A neural network is not constrained by the assumption that vehicle speeds must remain constant along the urban link, this allows the model to take into account how the existence of upstream queues may cause vehicles to slow down before arriving to the intersection stop line. Moreover, it is not bounded by the conservation of flow constraint, which allows the model to capture the effect of sources and sinks; and lastly, when considering lateral dispersion, the classical model will need to include additional probabilistic modeling for estimating turning percentages, while a neural network model will be able to additionally include this parameter within the same model structure. This comparison can be summarized in Table 3. Since the advantages of the neural network approach were more relevant to the purposes of the needed traffic flow progression model, the machine learning approach was selected moving forward with the design. To the extent of the knowledge of the author, no neural network model has been made to predict short term traffic flow arrivals, for every signal stream at a complex urban intersection.

TABLE 3: COMPARISON BETWEEN THE DESIGN OF A NEURAL NETWORK OR A NEW METHODOLOGY FOR CALIBRATING ROBERTSON'S MODEL

| Model Characteristic | Designing a new neural network model for Short-term traffic flow progression modeling | Developing a new methodology for calibrating Robertson’s traffic flow progression model that encompasses all influences on platoon dispersion |
|---|---|---|
| Interpretability | | X |
| Need for Data | | X |
| Time elapsed before model usage (i.e. for training or calibrations) | | X |
| Adaptability to different traffic conditions | X | |
| Scalability to different intersection configurations | X | |
| Ability to recognize patterns from sources and sinks | X | |
| Robustness against queue propagation | X | |
| Inclusion of lateral platoon dispersion | X | |

² Robustness is defined as the extent to which the performance of a prediction model will remain unchanged, regardless of differences in traffic conditions. Hence, the ability for the model to match the general arrival flow profile of a traffic stream, a low variance in the mean error for different traffic conditions, and fewer instances where large prediction errors are made are all considered indicators of robustness.

2.6. DEFINED BASELINE FOR MODEL EVALUATION

Now that a neural networks approach has been selected, a benchmark must be identified from which the model's performance can be compared. Since Robertson's classical model is the core of traffic flow progression modeling used in current predictive controllers, it will be used as the reference case. However, some calibrations will be made to the model in order to make it more applicable for the selected case-studies. These calibrations will be made using a naïve trial and error approach though, since the purpose is not to further update Robertson's model over what has already been done in literature, but to develop a context appropriate version of the model from which to compare the performance of the neural network. Accordingly, while this is not the most efficient parameter calibration methodology, it does set a relatively good baseline from which to evaluate the neural network's performance.

Using Equation (3), the recursive interpretation of the model is replicated. To calibrate the model, the practical approach outlined by Equation (4) is used, which simultaneously calibrates the α and β parameters, implicitly, by calibrating the smoothing factor in accordance to the time step duration and the average platoon travel time variance. Following Bie et. al. (2014)'s approach, through trial and error, different variances are tested until the lowest RMSE is reached between the model prediction and the empirical data used in this study. An additional reason why this performance indicator was selected for the calibration process is, as will be discussed in Chapter 3, this is also the performance indicator that the designed neural network uses to evaluate its own performance for the optimization iterations it makes during training.

To account for more than one lane, this model is slightly expanded following Yu & Recker (2006)'s proposal. Origin-destination (OD) pair relationships are included into the model by saying that the downstream flow of each lane is a dispersed profile of the weighted sum of all upstream flows. Hence, the weighted contribution of the upstream flow on each lane must be estimated for each lane at the downstream target destination point. This weight is determined through trial and error, with the objective of achieving the lowest RMSE. The additional benefit of this reference model is that, through the interpretable calibration process, more insights can be made on the traffic conditions of the data sets used for testing the developed prediction model. This allows for a more in depth analysis of the black box neural network's performance. For the expansion of a link's lane configuration from two lanes to more (at the entrance of an intersection) though, Bie et. al. (2014)'s approach for recalibrating Robertson's model after the lane expansion (and traffic flow split into separate streams) is followed. Further details on this calibration process is explained in later chapters.

2.7. CONCLUSIONS

This chapter served as the starting point for the short-term traffic flow prediction model design. Within this chapter, the first design choice made was to forecast the traffic flow arrivals through traffic *progression* modeling. This means that predictions will be made, not just based on time related patterns, but by forecasting how measured traffic profiles at one point will move to the next. This choice was verified when looking at the results of de Clercq's work, considering the results of his neural network (which was designed to detect traffic departure patterns over time) were quite low for the number of memory cells included and the level of aggregation of the selected time step. Secondly, the modeling approach selected was to use neural network algorithms to estimate the traffic flow arrivals. This choice was made through a literature investigation, where first the most relevant works to traffic flow progression modeling were explored, and the path that best met the requirements of the needed short term traffic flow prediction model was followed. Finally the base-case scenario, from which the neural network model's performance will be evaluated, was defined as a context calibrated version of Robertson's traffic flow progression model, to create a fair platform from which the improvements the neural network achieves can be evaluated. With these choices in place, the first iteration for the prediction model design can begin.

3. DESIGNING THE NEURAL NETWORK TRAFFIC PROGRESSION MODEL

This chapter outlines the first steps in designing for a neural network. The chapter begins with outlining the relevance of initializing this design with data from a fictitious case study, and explains how the synthetic data was generated. Next Chapter 3.3. lists the technical design choices made for the set up of the neural network algorithms, and the data pre(and post)processing steps that were taken. The information provided in Chapter 3.3. is applicable for the remainder of this report. Chapter 3.4. then delves into the systematic design process that was followed for selecting the neural network architecture, and finally Chapter 3.5. outlines the chapter conclusions.

3.1. INTRODUCTION

When working with machine learning algorithms, it is important to start with the simplest possible design, then systematically add more parameters until the target performance is reached. This methodology assures that the finally selected design is neither unjustifiably too complex, nor does it over-fit the data. With a neural network there are certain algorithmic design choices that need to be made when setting up the model; however, the most important choice made is for the model architecture, as this is where the network can be customized to the application needs. Using a simple, controlled, dataset, this chapter explains the steps taken to reach the final selected architecture, providing the needed justifications for this design choice.

3.2. CONTEXT

This case study is of a fictitious intersection, the configuration of which was determined based on the typical Dutch intersection geometry. The input data for this case study was from the corridor Beatrixlaan in Delft. At several points along this two lane corridor, detectors are placed, from which vehicle counts are recorded and stored. From one of these detector positions, the data was used as the upstream flow profile. Using the classical model described in Chapter 2.2.1. (which was also expanded to allow for the traffic streams to be further divided into multiple lanes), the downstream flows were simulated. A visualization of this case study can be seen in Figure 19.

3.2.1. MOTIVATION FOR THE CASE STUDY

The motivation behind this case study is that it allowed each iteration from the neural network design to be evaluated in a controlled environment. The upstream detected flow profile was dispersed to the downstream point using a model that is known to provide a realistic representation of platoon dispersion for ideal free-flow conditions. Accordingly, the initial evaluation of the neural network design is that it can learn and reproduce this upstream-to-downstream flow relationship, as outlined by the classical model. With this case study, no external disturbances exist in the dataset. This eliminates the possibility that any errors in the neural network's performance can be due to stochasticity in the data; and so, the goal for the final neural network design is to perfectly match the target data.

3.2.2. GENERATING THE SYNTHETIC DATA

Using MATLAB, synthetic arrival flows at the fictitious intersection were generated as a non-random transformation of the detector data taken from the Beatrixlaan corridor. The detector data was placed as the upstream flow data, while the synthetic data was identified as the target, arrival flow, the neural network must train to predict. The architecture of the synthetic case-study can be seen in Figure 19. The upstream point had a two lane configuration, matching the geometry of the corridor where the data was collected. For the arrival flow, as shown, it was assumed, for this scenario, that the far away loops are placed after the number of lanes expands at the intersection entrance. This flow data used was generated using the following steps:

- 1) Real upstream flow profiles were used for this section, which were later dispersed longitudinally and laterally synthetically. The detector data came from a long Provincial road approaching intersection K001 in Delft. These flows were estimated by clustering measured detector counts into 5 second time step bins (i.e. summing the detected counts within each 5 second time step). Next, "number of vehicles per 5 seconds" was converted to "number of vehicles per hour".

- 2) The data points of both q_{u1} and q_{u2} were multiplied by percentages representing directions of flow which varied over time (e.g. for first 50 time steps 30% of q_{u1} is going left, 25% first forward lane, 25% second forward lane, and 20% right, while the opposite for left and right were applied for q_{u2} . However, for the next 50 time steps these percentages changed).
- 3) Once the data points from these distributions were divided from two to four lanes, they were inserted into Robertson's Platoon dispersion model equation in order to simulate how they would longitudinally disperse along the road segment. The speed (v) used to determine the average travel time per time step, and the α and β parameters of the model, were selected as normally distributed (slightly randomized) values, restricted by a mean and variance as follows:
 - For v : the selected mean was 50kph with a variance of 6kph
 - For α : the selected mean was 0.35 with a variance of 0.05
 - For β : the selected mean was 0.8 with a variance of 0.07

The purpose of the slight randomization of these variables was to introduce a manageable level of stochasticity into the dataset, to determine if the neural network would be able to overcome it.

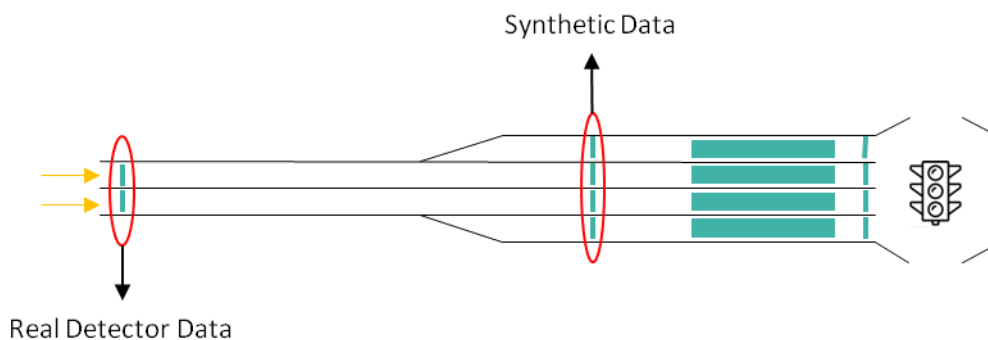


FIGURE 19: GEOMETRY OF FICTITIOUS CASE-STUDY

3.2.2.1. SIMULATING DISPERSION FOR A TARGET SET

For simulating arrival flows, Seddon's recursive re-formulation of Robertson's platoon dispersion model, displayed in Equation (6), was referenced:

$$q_d(t_n) = \sum_{t_i=1}^{t_n-\beta T} q_u(t_i) * F * (1 - F)^{t_n-\beta T-t_i} \dots \dots \dots (6)$$

This model was interpreted in MATLAB as shown in Equation (7). In order to test the model's ability for determining traffic progression multiple time steps into the future, with each time-step representing 5 seconds, a four time-step future horizon was selected (i.e. a 20s horizon). This is the minimum expected time horizon between two successive intersections.

$$q_d(t + 4) = F * q_u(t) + F * (1 - F) * q_u(t - 1) + F * (1 - F)^2 * q_u(t - 2) + F * (1 - F)^3 * q_u(t - 3) + F * (1 - F)^4 * q_u(t - 4) \dots (7)$$

Using the outputs of this model as a target set, an ideal flow situation is created that is sufficient for assessing the ability of different model architectures to detect a relationship between input and output flows. This will provide a guide for designing the appropriate neural network for this project.

3.3. SETTING UP THE NEURAL NETWORK PARAMETERS AND DATA

Aside from the neural network architecture, design choices must be made with regards to the model parameters. The choices made within Chapter 3.3.1., will continue to be applicable for the remainder of the report. These choices were reached through research, recommendations from online courses (Stanford University, 2017), and through experimentations. However, in accordance to these parameters, the data inputted into the model must

be pre-processed, to aid the neural network in pattern and relationship recognition. After the model gives its outputs, post-processing is applied to return the data to an interpretable form. This data manipulation process is outlined in Chapter 3.3.2.

3.3.1. MODEL SETUP

Regardless of the model architecture, certain technical parametric choices need to be made when designing a neural network. Accordingly, this chapter outlines all the technical design choices made for the final built neural network.

Identifying the Learning Style

The first step in building a prediction model using a machine learning algorithm is to identify the learning style needed. There are several types of neural networks that can be built, each using a different learning style to serve a certain purpose. For instance, famous deep learning networks such as a convolutional neural network or a stacked auto-encoder work best for unsupervised learning methods and/or classification problems. For this particular traffic flow forecasting problem though, supervised learning is needed. The model is required to draw relationships between specified (historical) input datasets and specified (also historical) target datasets. Once it has learned this relationship, it can begin making predictions for future data. This relationship is relatively known, from the classical mathematical algorithms used for traffic flow progression, and so the supervised learning can be facilitated with the appropriate input and output variable combinations. This can best be done using a feed forward neural network, which transfers information, from input nodes to output nodes, in a similar manner to Robertson's mathematical functions. If a simple, static, feed forward network is not strong enough for this application, more complex architectures will be needed. However, whatever network architecture chosen, it must still use supervised learning.

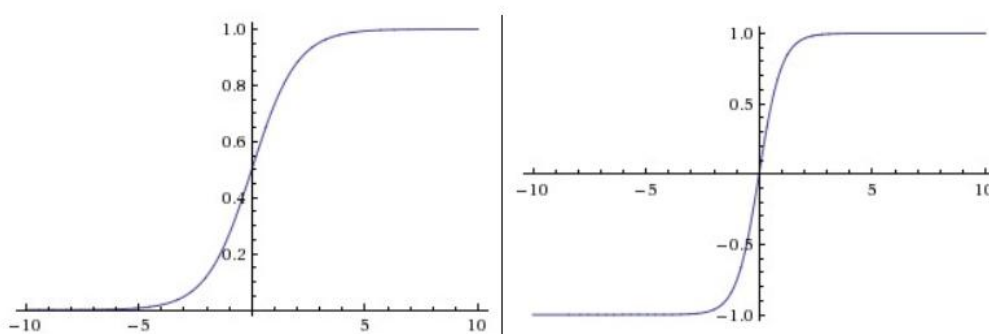
Identifying the Loss Function and Learning Process

For supervised learning, the method of back-propagation is used. With every iteration of this method, once information is transferred forward from the input nodes to the output nodes, the error between the model's prediction and the target value is calculated. The model then updates its vector weights and biases to minimize errors for the next iteration. To do, each node in the network is programmed to compute the local gradient of its input with respect to its output value (Stanford University, 2017). This backward pass is recursively applied throughout the network, using the chain rule, starting from the output nodes all the way back to the input nodes (Stanford University, 2017). These backward moving gradient signals allow each node to communicate to its predecessor the direction in which it must calibrate its weight vectors to minimize the value of the final loss function output. Since this was consistency the one selected in all the studies researched in Chapter 2.5.2, the loss function selected for this network is the mean squared error (MSE). Hence, the direction and magnitude with which to improve the network's weight space from the output layer to the hidden layer is related to the gradient of this function. From any hidden layer back, the gradient is related to each neuron's activation function.

Selecting the Activation Functions

The activation function for the output layer is always a linear function. However, for the neurons of the hidden layer, there are several choices for nonlinearity functions within which input data will be quashed. Activation functions also determine the "firing rate" of neurons for certain linear regions of its inputted datasets. Hence, the majority of reference papers that have built variations of feed forward neural networks for traffic flow predictions use the Sigmoid nonlinearity function for its interpretability. The Sigmoid activation function of a single neuron takes the weighted sums (plus bias) of the inputs from the previous layer and transforms it into a value between zero and one, with large positive values becoming one, representing a maximum frequency firing rate, and large negative values turning to zero (i.e. no output of information). However, in the lectures of the Neural Networks course (Stanford University, 2017) followed for this project, this function was strongly discouraged for its saturating form. When a Sigmoid neuron's activation saturates near zero or one, its local gradient will be zero, meaning no signal will flow from this neuron to its weights, during back-propagation, terminating the parameter optimization process of the effected neurons. Alternatively, the tan-sigmoid function is a zero centered form of the sigmoid function (Stanford University, 2017). It therefore squashes data into a range of $[-1,1]$, which is more applicable to the preprocessed data, which is

normalized to a range between -1 and 1 as well. Accordingly, while this function also has a saturating form, significantly less data is converged to zero. The difference between both activation functions can be seen in Figure 20.



Left: Sigmoid non-linearity squashes real numbers to range between [0,1] **Right:** The tanh non-linearity squashes real numbers to range between [-1,1].

FIGURE 20: TWO CHOICES FOR ACTIVATION FUNCTIONS AS PRESENTED BY STANFORD UNIVERSITY (2017)

Weight and Bias Parameter Initialization

Although, during training, the neural network is expected to recursively update its weight and bias parameters, until they are optimized, their initialization has a significant impact on the model’s training performance. For instance, while it may seem reasonable to zero-initialize all parameters, this will cause all neurons in the network to compute the same output, and therefore the same gradients during back-propagation, leading to the same parameter updates. Hence, without asymmetry in the initial weight parameters, the network is unable to find different effect each weight connection has on the total error, for their proper individual optimization. Alternatively, it is recommended to randomly initialize the neurons, but while also assuring the neurons have the same output distribution, in order to mitigate the concern that the outputs of a randomly initialized neuron would have an increasing variance the more inputs there are (Stanford University, 2017). Accordingly, the variance of each neuron’s output is normalized to one by scaling its initial weight vector by the square root of the number of neurons it takes as input (Stanford University, 2017), as shown in Equation (8).

$$InitW = \frac{randn(n)}{\sqrt{n}} \dots\dots\dots(8)$$

Where n is the number of inputs into the neuron. With this initialization, it is expected for the model to more rapidly converge during training, for the outputs of the model to be more robust (due to similar initialization values for each run), and for the model to more likely reach a local optimum closer to the global optimum. On the other hand, it is recommended to initialize all biases to zero, and allow the optimization function to incrementally increase the bias.

Selecting the Training (i.e. Weight/Bias Optimization) Function

Once the gradients have been computed with back propagation, the model’s weight and bias parameters must be updated to minimize the gap between the inputted target data and the model predictions through a numerical search procedure (Sjoberg et. al., 1995). As recommended by the reference paper (Qiao et. al., 2001), the parameters are optimized following the Gauss-Newton-based Levenberg-Marquardt (L-M) procedure. The main advantage of this second-order optimization method is that it does not rely on learning rate hyper-parameters (Stanford University, 2017). This eliminates the risk of setting a learning rate too high for an activation function, and killing (i.e. making inactive) the network neurons with improperly set learning rates (Stanford University, 2017). However, the disadvantage of this training function is that it is computationally intensive, and so takes a long time to train.

Measures Taken to Prevent Over- Fitting

In order to prevent overtraining of the neural network, which is when the model begins to memorize its training dataset instead of learning the patterns within it and relationships between the inputs and outputs, the early stopping technique was used. This technique is basically a constraint placed on the number of iterations made. With every iteration, the model validates its newly optimized weight and bias parameters by attempting to predict the

validation dataset and checking the mean squared error. After six iterations within which the model's ability to predict the validation dataset does not improve, the model stops its training. This limit for performance improvement failures on the validation set was recommended in the MATLAB neural network documentation. It provides the neural network enough trials to get out of a local optimum, but restricts it from continuously attempting to optimize with no progress. Otherwise, the model could keep trying to match its outputs with the training dataset, to the point that it would memorize it. This measure was selected since, with short-term traffic data there is a lot of noise, so it is unrealistic to assume the neural network can keep training until it reaches its zero error objective without over-fitting.

Dividing Data Into Training, Validation and Test

The dataset was divided to training, validation and testing as follows: the first 60% of the data available was used for training the model, the next 20% was used for validation and the last 20% was used for testing. During the validation phase the model makes predictions without looking at the target dataset, then calibrates its parameter weights if needed. The test phase allows the user to see how well the fully trained and calibrated neural network now performs.

Defining the Prediction Horizon

The neural network is intended to predict the arrival profile of vehicles based on their measured departure profiles at directly upstream intersections. Accordingly there is no freedom in selecting the prediction horizon. The length of the horizon is determined by the average travel time between two successive intersections.

3.3.2. DATA PRE-(AND POST-)PROCESSING

For this study, actual (historical) detector data was collected and used for analysis and modeling. Accordingly, received data was in V-log format. Hence, before beginning to use the data it had to first be compiled into numerical data. Afterwards, the translated detector count data needed to be processed in order to get the relevant information out for usage. Finally, before entering the data into the neural network, it had to be properly scaled to reduce noise, and remove differences in the units of different variables. This allows the neural network to better detect patterns in data and relationships between variables.

3.3.2.1. COMPILING AND PROCESSING V-LOG DATA

The raw data used for this study was detector count data harvested by road owners in the Netherlands. Accordingly, it first had to be compiled from V-log data into numerical data. The program CuteView, developed by Werner van Loo from the Municipality of Amsterdam, and widely used by the Dutch road owners, was used to compile the V-log data. The program takes in two inputs: the data itself, and the configuration files of the controllers at the intersection. The configuration files allow the program to assign the correct labels to the correct detector data. Hence, the user can be able to identify which data stream comes from which detector. The output of the file tells the user the exact timestamp a detector was occupied, the exact time stamp it was no longer occupied, then occupied again, and so on. The unit of measurement for time is one tenth of a second, and CuteView begins its time step counts from zero, for any data set inserted. Users must be careful then when compiling V-log data from multiple intersections to make sure the data set uploaded into the application all start from the exact same time. The translated raw data was later processed to calculate the volume of vehicles passing over detectors within certain time step bins (e.g. for five or ten second time steps), to estimate the percentage distribution of flows between different lanes over time, and to estimate traffic states from detector occupancies.

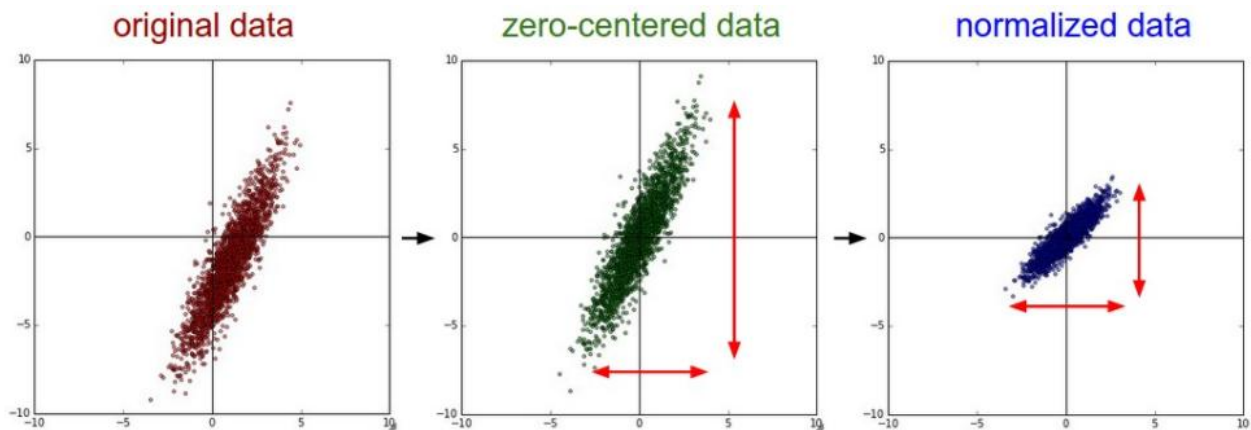
Detector counts are not the only output from CuteView's translation of V-log data. From CuteView, it is also possible to get outputs on the historical start and end times of controller phase cycles. This information includes the WUS, which is the external controller state (i.e. the green, yellow and red lights displayed to vehicles), and the GUS, which is the internal controller's state (e.g. whether it is deciding to extend green or end green). This information is also outputted with one tenth of a second time units. From the configuration files, this information is provided, with the correct label, for every controlled traffic stream.

The obstacle with V-log data and configuration files though is that they are very controller specific, even for the detector count data. Accordingly, every time a controller is updated, all the previous V-log data is no longer interpretable. This is because of two reasons: the first is that road owners tend not to keep the configuration files for

old controllers; and, the second is that often when new controllers are installed, changes to the configuration of the loop detectors are made. Consequently, there isn't actually a lot of available cohesive V-log data stored that can be used for big data analytics.

3.3.2.2. SCALING DATA FOR NEURAL NETWORK

Since the inputs and outputs of the model can have various units (i.e. flows [veh/hr] and percentages [%]), input and target datasets were first zero-centered and then normalized between -1 and 1, as recommended in Stanford University (2017) and displayed in Figure 21. Scaling data this way is very important, since neural networks do not recognize the different units of measurements for the different inputs and outputs. This preprocessing step is first computed on the training data and then 'applied' onto new data points, which are added for validation and testing. Once the model outputs its predicted datasets, this output data is then 'reverted' back from its normalized form to its original form for interpretability. This can easily be done on MATLAB with "apply" and "reverse" functions.



Common data preprocessing pipeline. **Left:** Original toy, 2-dimensional input data. **Middle:** The data is zero-centered by subtracting the mean in each dimension. The data cloud is now centered around the origin. **Right:** Each dimension is additionally scaled by its standard deviation. The red lines indicate the extent of the data - they are of unequal length in the middle, but of equal length on the right.

FIGURE 21: DATA PREPROCESSING RECOMMENDATIONS, AS PRESENTED IN STANFORD UNIVERSITY (2017)

3.4. DESIGN PROCESS OF NEURAL NETWORK ARCHITECTURE

Essentially, the methodology followed for reaching the final model architecture, was to begin with the simplest feed forward neural network form, then add more information with every iteration. It was necessary to initially perform some research on how each variable, and each variable combination, affects the ability of the model to capture the progression of traffic; since, to the best of the knowledge of this research, this is will be the first short-term traffic flow prediction model that attempts to encompass all aspects of traffic flow progression between two points on an urban link, and so no fully ready reference was available. However, there were some quite relevant previous works on short-term traffic flow prediction modeling, using neural networks, which were identified and described. While these reference models were intended for simpler purposes than to be used for predictive control, their features and the results they elicited served as guidance for each iteration made when developing the prediction model for this study. The most successful model architecture in the end though was an explicit restructuring of Seddon's interpretation of Robertson's platoon dispersion model, with an additional 'turning percentages' feature that accounted for the expansion of the number of lanes at the entrance of an intersection.

In order to have a clear benchmark for the predictive abilities of each tested neural network architecture, a synthetic case study was used. The upstream departure flows came from two short loop detectors found along a Provincial road in Delft, leading up to intersection K001, which will be discussed in the next chapter. The 'arrival flows' though were not taken from the real data. Instead, using Robertson's platoon dispersion model, and using some assumed percentages for how the vehicles would split amongst the lanes, after their number expands, arrival flows were simulated. Following this methodology, the neural network can now be tested to see how well it is capable of matching the strongest classical short-term traffic flow prediction model applied today. Once an architecture is

formed that, at least, is capable of replicating the results of Robertson’s model, it can then be tested and evaluated for actual real-life traffic situations (in later chapters).

3.4.1. EXPERIMENTAL SETUP & KPIS

Although the models presented in Chapter 2.5 are considered benchmarks for the development of this project’s model, before starting with a complicated architecture, first a simple feed forward neural network must be tested, in order to see how well it recognizes patterns within the data. From there, additional features will be added to the model, until it reaches its assumed full potential. When features are being added to the model though, while experimentation is important, the previous studies are additionally used as a guide. The first objective for developing the model, is to find an architecture that is capable of performing as well as Robertson’s platoon dispersion model. Accordingly, synthetic data is calculated, using this model, and a neural network architecture is designed to assure the behavior of Robertson’s model and the neural network are near-perfectly matched (i.e. the difference between actual and predicted arrival flows was the KPI). More information on architectural design choices are provided within the chapter itself.

3.4.2. RESULTS AND EVALUATION

For the synthetic data, several model architectures were tested until the final model was formulated. The process followed to reach this final architecture is outlined within this section. The essential methodology though was to begin with the most basic form, and then keep adding features until the optimum results have been reached.

3.4.2.1. FEED FORWARD NEURAL NETWORK

The first tested architecture was the simplest possible feed forward neural network for this case. The network took in the two inputs from the two upstream detectors along the corridor leading up to intersection K001. The defined target was one of the four downstream arrival lanes from Figure 19. This model was run four times; once for the arrival flows at each downstream lane. A visualization of the results are shown in Figure 22.

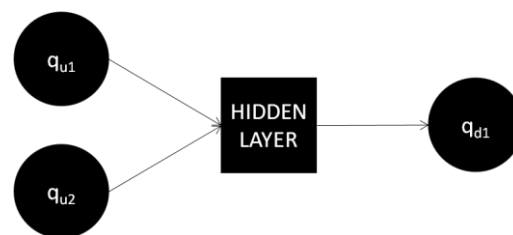


FIGURE 22: ARCHITECTURE OF A SIMPLE FEED FORWARD NEURAL NETWORK, WHICH TAKES INPUTS FROM THE TWO UPSTREAM FLOW POINTS AND PREDICTS THE FLOW AT ONE OF THE DOWNSTREAM LANES, AT THE DESTINATION POINT OF THE TIME-SPACE PREDICTION HORIZON.

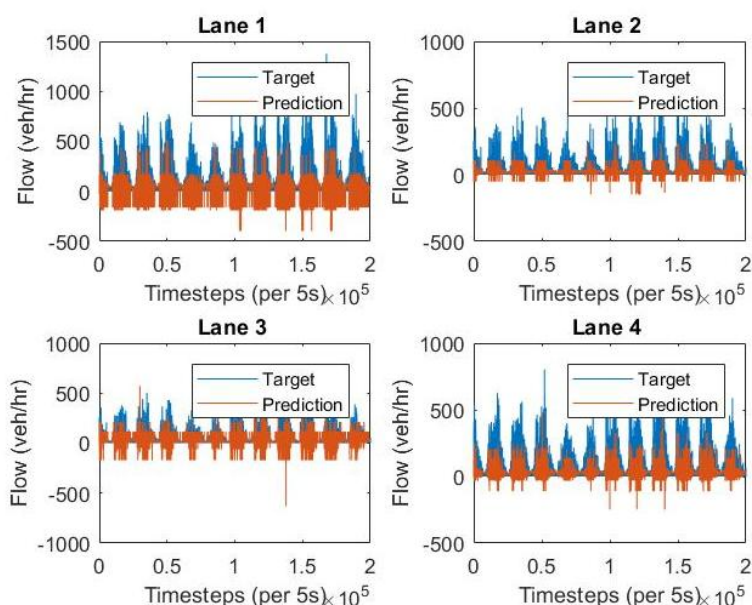


FIGURE 23: FLOW VS. TIMESTEPS CHART FOR BOTH TARGET (ACTUAL) FLOW AND PREDICTED FLOWS BY A FEEDFORWARD NEURAL NETWORK. THE RESULTS ARE FOR EACH ARRIVAL LANE WITHIN THE FICTITIOUS CASE STUDY.

As can be seen from Figure 23. A simple feed forward neural network was capable of recognizing some pattern with regards to input and output flows, but it was still unable to reliably replicate the behavior of the synthetic platoon progression. This can be fixed by adding more data, for better learning; but also, by improving on the model architecture. The most likely reason is that the split from two to four lanes made it difficult for the neural network to find relationships between upstream and downstream flows. To aid the neural network in making this relationship, the total number of nodes at the downstream intersection were inserted into the same model, as shown in Figure 24.

The aim is to allow the neural network to recognize that the total flows entering from the upstream two nodes are distributed amongst four arrival lanes.

The phenomenon where the network receives information from its outputs is called Multi-Task Learning (MTL). Simply put, how this works is, information is implicitly absorbed by the model during back-propagation training, since the objective function of the neural network is to minimize the error of all its outputs. Back propagation is the process of computing gradients on the connections of a neural network with respect to the selected loss function (e.g. mean square error) (Stanford University, 2017), to efficiently direct the updates of the connection weights and properly calibrate the properties of the elements in the hidden layer. Hence, the neurons of each layer are influenced by the partial derivative of their successor. This therefore entails, that training a neural network to meet the target values of multiple outputs simultaneously would result in an inductive transfer of information between these outputs (Caurana, 1997). Hence, correlations between the multiple outputs can be found. However, no significant signs of improvement in learning were seen, although there appeared to be slightly less noise in the prediction model's output.

MTL has been proven to be quite useful in cases where one output is a function of the other though (Caruana, 1997). In this case, the longitudinal progression is a function of the percentage distribution of the flows amongst each lane. Before disregarding the use of multi-task learning, one more expansion was tested, which is to include the corresponding turning percentage of each lane into the model outputs as well, as shown in Figure 25. While upstream flows and turning percentages are not directly related, the concept of using upstream flows as inputs to predict downstream flows at the target location was adapted from Jiao et. al. (2014)'s model, due to the positive results it elicited. Their finding was that certain input flows upstream contribute more towards certain turning percentages downstream. Hence, the purpose of including the arrival flow percentages in this way was to see if it would allow the model to recognize the pattern of turning percentages included in the synthetic data, and if that would therefore assist in determining the arrival flows of vehicles. The results of this architecture, showed that the predictive abilities of the neural network do, in fact improve with this multi-task architecture. Accordingly, it becomes the new benchmark for the model's architectural development.

However, the predictions still did not match the actual traffic flow arrivals, despite the fact that these arrivals were created through a systematic transformation of input data. Additionally, there continued to be extensively large negative predictions made by the model, which were informative of the difficulty the model continued to have with matching the peaks in the profile. These were a result of harsh corrective measures made with each optimization iteration the model made, as it attempted to adapt its predictions to capture both low and high peaks. This indicated that the neural network, while showing improvements, was still not receiving enough information that allows it to learn the relationship between upstream and downstream flow profiles. Accordingly, further expansions were made.

3.4.2.2. AUTOREGRESSIVE NEURAL NETWORK (NARXNET)

Starting from the last reached architecture, deeper learning was included into the model by adding memory cells. These cells will allow the neural network to 'remember' previous departure flows, previous arrival flows and previous turning percentages when making its next prediction. This memory allows the neural network to better

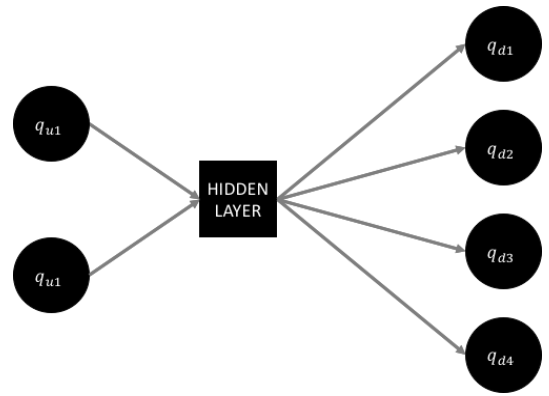


FIGURE 24: FEED FORWARD MODEL EXPANSION: MULTI-TASK LEARNING THE ARRIVAL FLOWS OF ALL LANES SIMULTANIOUSLY

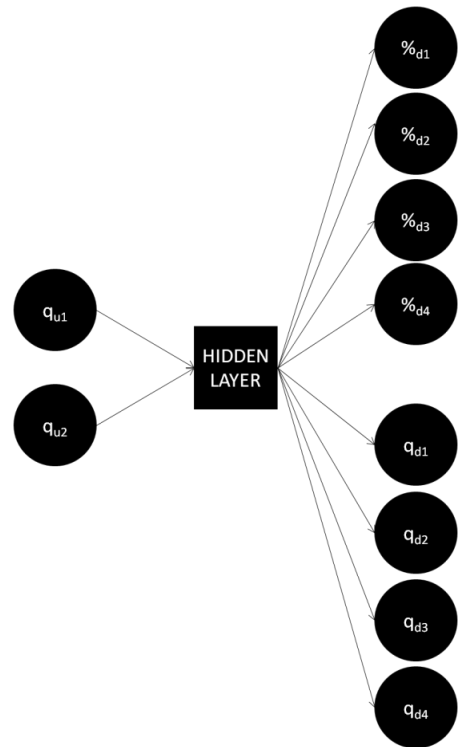


FIGURE 25: FEED FORWARD MODEL EXPANSION: MULTI-TASK LEARNING THE ARRIVAL FLOWS AND TURNING PERCENTAGES OF ALL LANES SIMULTANIOUSLY

identify time related trends, and identify error correlations in its outputs. The model is able to calculate these error correlations due to the fact that it has been designed to take in an open-loop feedback, which means that actual arrivals, from previous time steps. Essentially, an open-loop NARX model architecture is purely a feed-forward neural network trained through static back-propagation, since the “feed-back input” is actually the true measurement of the output variable (at the time of its occurrence) rather than a recursively inputted predicted value. This results in a series-parallel architecture. Hence, the same process for multi-task learning that can be used in a feed forward neural network is applicable for an autoregressive neural network as well. Most importantly though, the neural network will now be able to recognize a relationship between departure flows, at different time steps, and arrival flows. With the feed forward neural network, the recursive relationship defined in Equation (5), in Robertson’s model, is not explicitly captured due to the lack of memory included regarding previous platoon departures. With a NARX network though, which is shown in Figure 26, the memory of past inputs may aid the model to recognizing this relationship.

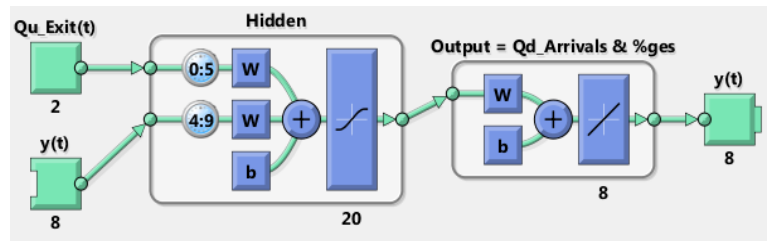


FIGURE 26: NARX NETWORK ARCHITECTURE, SELECTED FOR THIS SYNTHETIC CASE STUDY

Five times steps of memory were included into the model, as can be seen above. The reason for that that is, in accordance to Robertson’s platoon progression theory, which will be further explained in Chapter 3.4.3., arrival flows have been found to not only be attributable to a single platoon that left one prediction horizon back (i.e. the vehicles that left 20s ago, in this case-study). Slower vehicles, from platoons that left earlier, also contribute to these arrivals. Hence, the limit to the memory time steps was assumed as the time it takes for vehicles moving at 20km/hr, on average (i.e. 80% of that speed was taken as the lower variance limit of this average), are the slowest vehicles that can contribute. Trials and errors conducted with the number of memory cells concurred this was the best memory quantity to take. “Memory” is included in the model as additional replication of the inputs, but from past time steps. The exact same process, and memory capacity, is conducted for the feedback, but with a time lag of the prediction horizon, since that is when the target vehicles actually arrive and are detected.

The NARX model was capable of, almost perfectly, learning the pattern between departures and arrivals, and then making predictions for future steps with a high accuracy as can be seen in Figure 27, which displays a line-chart for target values and a line-chart for predicted values that are nearly entirely in sync. In Figure 26, the eight neurons in the output represent the arrival flows on each of the four lanes, and the percentage of the total flow on each of the lanes as well. The decision for how many memory cells are needed in this model depended on several trials and errors, as does the selection for the number of neurons within the hidden layer. These values were found to provide the best results. Any changes in them only worsened the neural network’s performance.

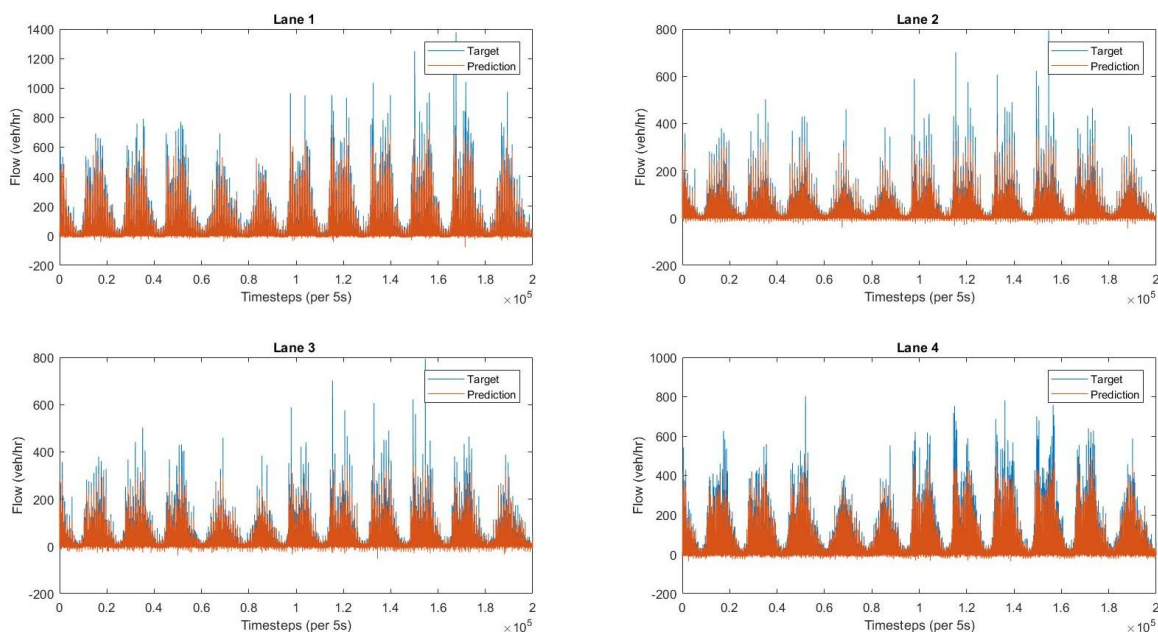


FIGURE 27: RESULTS FOR NARX NETWORK

Although the visual representations of real flows versus predicted flows do not give a precise indication of how accurately the neural network performed, they do reveal how well the model is able to match the profile shape of traffic flow arrivals. In order to check to see if the model performs as well as it appears though, additional checks were made, as shown in Figure 28. The results of the regression analysis performed between the predicted and target datasets for training, validation and testing are represented on the left of this figure. As shown, the outputted R-squared parameters are high, indicating the model has learned the simulated traffic progression conditions quite well, especially considering there was a level of randomness used for both the dispersion parameters and speeds used to estimate target arrival flows. Moreover, from the error histogram, on the right, it appears the average error is around 3 vehicles/hour for every five second time step of flow prediction. These results validate the conclusions of Qiao et. al. (2001)'s study that a NARX network is the most appropriate architecture for predicting longitudinal traffic progression, and Jiao et. al. (2014)'s recommendation for using upstream flows to predict downstream turning percentages with a feed forward neural network.

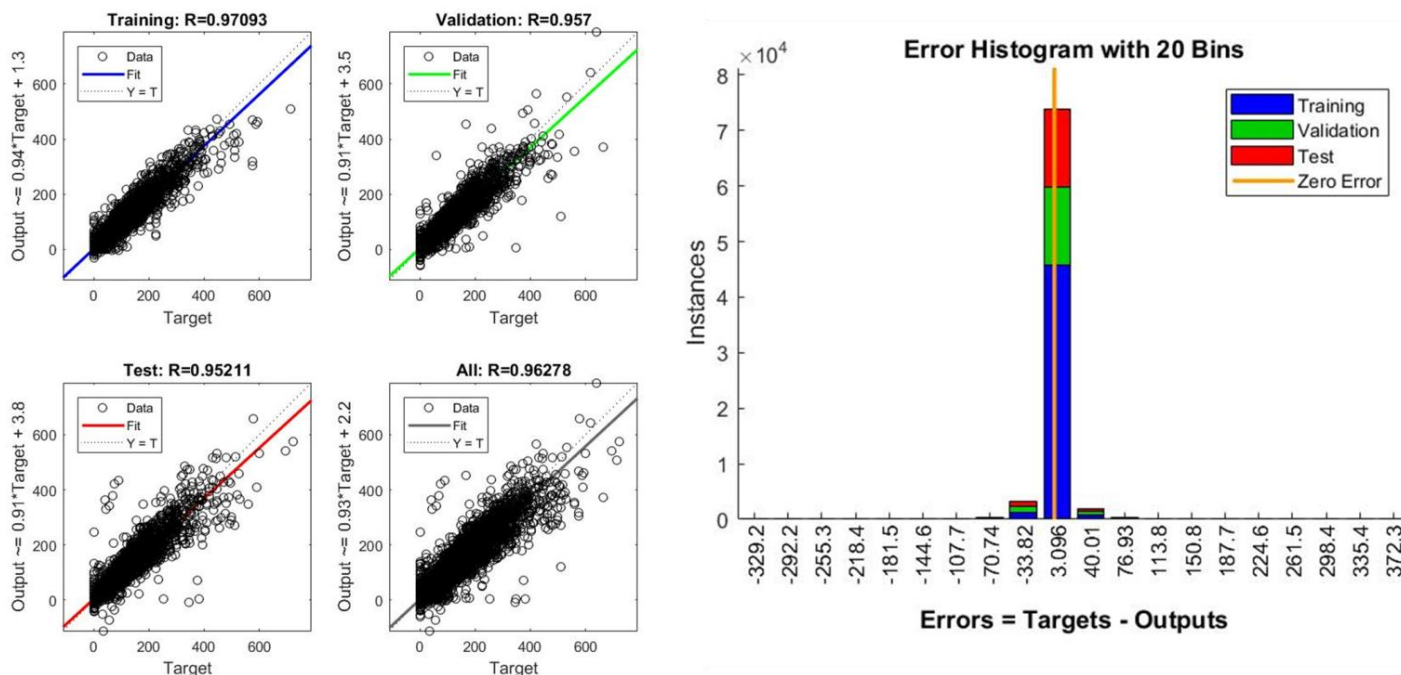


FIGURE 28: LEFT = RESULTS OF POST-TRAINING REGRESSION ANALYSIS CONDUCTED BETWEEN THE MODEL'S PREDICTED OUTPUTS AND THE TARGET OUTPUTS. RIGHT = ERROR HISTOGRAM SHOWING WHAT ERRORS APPEARED AND HOW MANY TIMES.

3.4.2.3. VERIFICATION CHECK: AUTOREGRESSIVE NEURAL NETWORK (NARNET)

A NARX neural network is essentially an extension of a nonlinear-autoregressive neural network (NAR) that includes exogenous (i.e. external) inputs than the ones included within the time series. Accordingly, in order to verify that the NARX model was in fact the ideal model to use, a NAR model was tested as well. The results of this test would verify whether the selected architecture is the correct one for its application purpose, or if the added memory of arrival flows and percentage splits was enough for the neural network to determine patterns of arrivals. Five memory cells, and twenty neurons in the hidden layer, were included for this model as well, and the results confirmed, as displayed in Figure 30, that it was the relationship between several departures to several arrivals that gave the neural network its predictability strength, not the sole memory of arrival flow patterns.

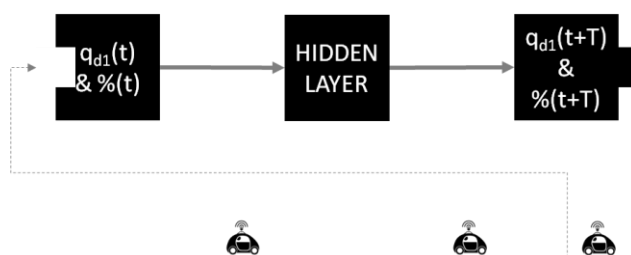


FIGURE 29: ARCHITECTURE OF NARNET WHICH TAKES INPUTS AS PREVIOUS TIME-STEPS OF ARRIVALS AND OUTPUTS FUTURE TIME STEPS OF ARRIVALS FROM THE SAME DATASET.

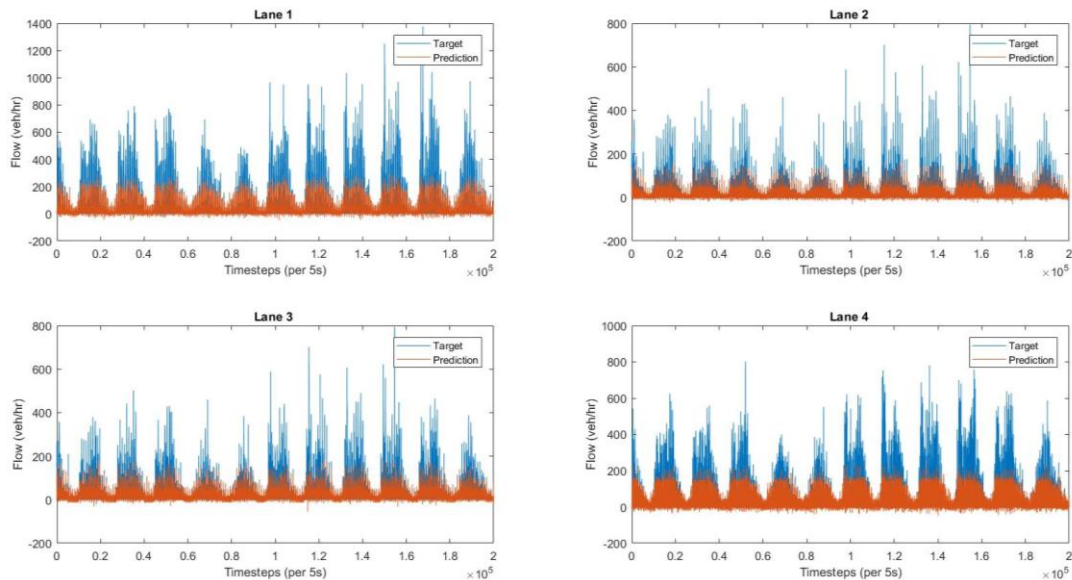


FIGURE 30: RESULTS FOR NARNET

3.4.3. RELATIONSHIP BETWEEN TRAFFIC PROGRESSION THEORY AND NARNET

With Neural Networks, determining the correct connections between the nodes (i.e. the architecture), and defining the relevant variables, is very important. It is not coincidental that the NARX network extensively outperformed the other model architectures, with the same number of neurons in the hidden layer, the same parameters; and, in comparison to the NARNET, the same number of memory cells. With this architecture, and combination of variables, the neural network can be considered as an expanded restructuring of Seddon's interpretation of Robertson's model, that not only considers longitudinal progression (i.e. platoon dispersion), but lateral progression as well (distribution of vehicles amongst an increasing number of lanes), as shown in Figure 31.

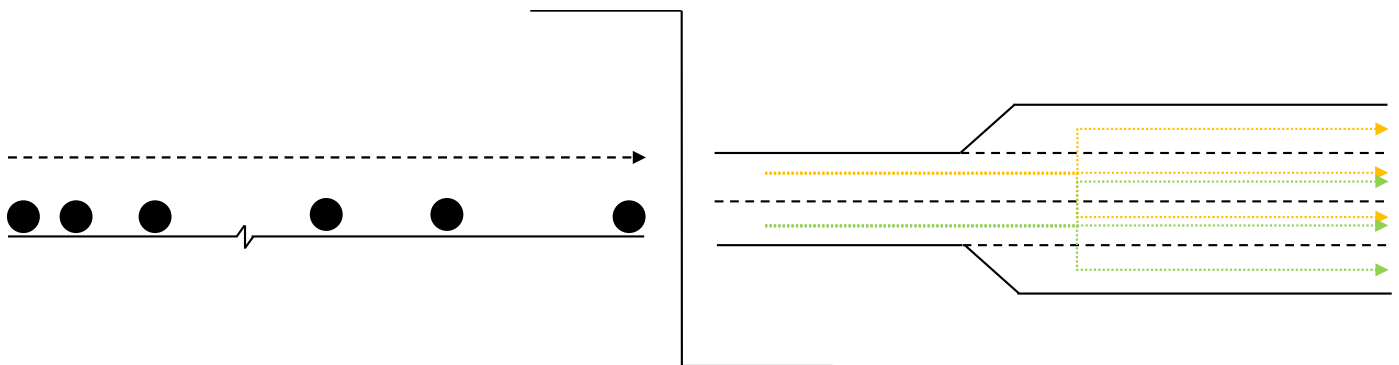


FIGURE 31: TWO TYPES OF TRAFFIC PROGRESSIONS: ON THE LEFT IS LONGITUDINAL PLATOON DISPERSION, AND ON THE RIGHT IS THE LATERAL SPREAD OF VEHICLES AMONGST THE LANES.

3.4.3.1. LONGITUDINAL PROGRESSION

According to the results of chapter 6.2, with the static feed forward neural network, the a relationship between upstream and downstream flow was quite difficult to find. This is a logical conclusion, since vehicles exiting an intersection do not remain platooned together along the corridor leading to the next intersection, nor do individual platoons simply spread out in a measurable manner without interacting with other vehicles. This makes point to point relationships for flow noisy. The theory behind Seddon and Robertson's models is that, while cars are expected to get from point A to point B at the free-flow speed limit, even under ideal free-flow conditions, this is not the case for all cars. Some drivers choose to travel slower than the free-flow speed, and so arrive at longer times. The differences in speeds and driving styles amongst drivers causes platoons to spread out to the point that they interact blend together. Accordingly, the group of vehicles arriving at an intersection consists of the fast cars that have departed " t " seconds ago (where " t " is considered to be the time it takes vehicles to travel from the previous intersection to the

target intersection at average speed), in addition to those that departed prior to that, but have been moving more slowly towards the next intersection, as shown in Figure 32.

A NARX model is a dynamic feed forward network that aims to find relationships between both input and output time-series, rather than single variables. Most basically, this model structure is designed to search for a pattern of flow arrivals (downstream), overtime, by drawing a relationship with a set of departures (upstream), at several time steps, as shown in Figure 33. The difference between this formulation and the static formulation, is that a static model attempts to draw correlations between *single instances*, at several time-steps. On the other hand, the main difference between the neural network and the classical platoon dispersion model is that Seddon and Robertson include a mathematical assumption that, as speeds decrease, the dispersion increases geometrically. This assumption is not included in the neural network. Alternatively, a trained neural network of this form is expected to begin drawing its own conclusions about the relationship between upstream departures, and downstream arrivals, that is more adaptive to changing traffic conditions. Accordingly, it is not restricted to a specific probability distribution.

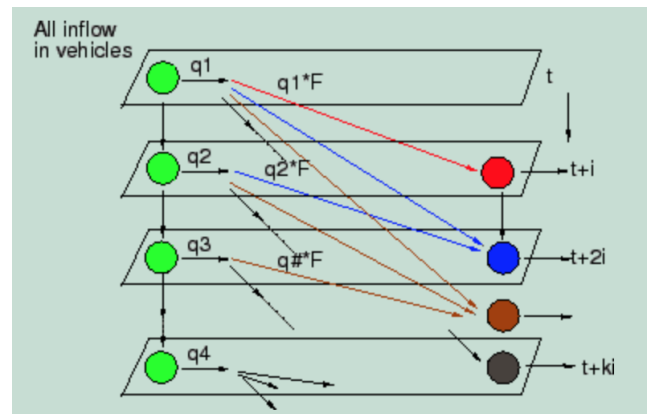


FIGURE 32: THE GRAPHICAL REPRESENTATION OF ROBERTSON'S MODEL, AS DISPLAYED BY THE NATIONAL PROGRAM OF TECHNOLOGY ENHANCED LEARNING (MATHIEW, 2014)

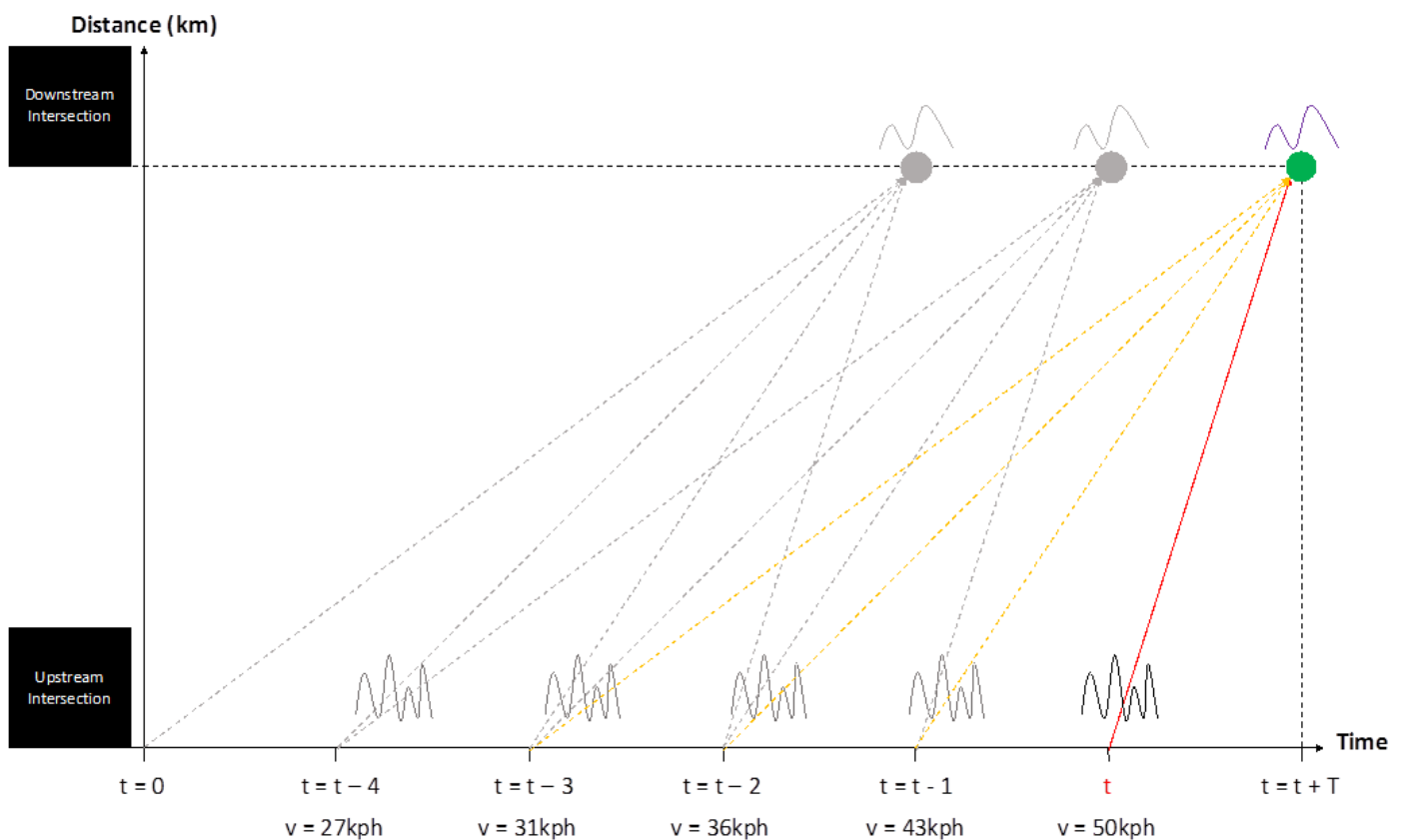


FIGURE 33: A VISUALIZATION OF THE TIME-SPACE RELATIONSHIP BETWEEN DEPARTURE FLOWS UPSTREAM AND ARRIVAL FLOWS DOWNSTREAM AT SUBSEQUENT INTERSECTIONS

3.4.3.2. LATERAL PROGRESSION

The second dimension that needs to be predicted is how vehicles will distribute amongst the different lanes, after the road configuration expands from two lanes to three, or more, lanes. Percentage distributions amongst the lanes, as identified in Figure 31 (right), are expected to follow a certain time-based pattern (i.e. based on the percentages now, and a few time-steps back, the percentage for the next time step can be predicted). Since the objective is to predict arrival flows, and arrival flows are a function of these percentages, the straight forward solution would be to include them as an input. However, as shown in Figure 34, arrival flows downstream are only a function

of the turning percentages that occur when the vehicles approach the intersection entrance, where the lane expansion is. Accordingly, these measurements are still in the future, and so cannot be measured, within the time-stamp at which the neural network is required to make its prediction..

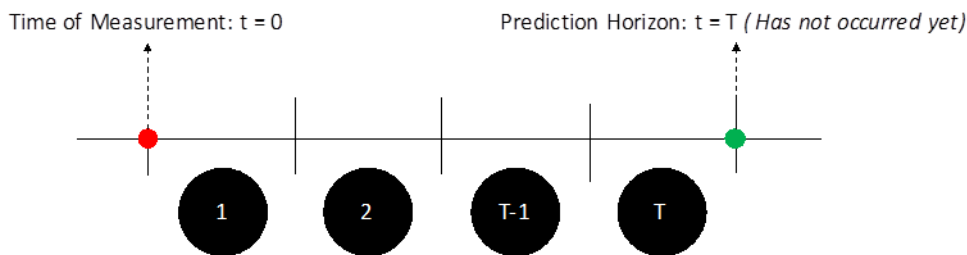


FIGURE 34: A VISUALIZATION OF HOW, AT THE POINT OF MEASUREMENT AND PREDICTION, THE PERCENTAGES OF ARRIVALS PER LANE AT THE TIME HORIZON HAVE NOT YET OCCURRED AND SO ARE STILL UNKNOWN.

Fortunately though, as previously explained, multi-task learning provides a solution for this dilemma. Through back propagation learning, the model will still be able to capture that its main output (arrival flows) as a function of its secondary output (lateral percentage distribution of vehicles) implicitly, through the optimized values of its weight and bias parameters. Moreover, with an open loop autoregressive model structure, feedback loops will allow the actual values of the predicted percentages to be reinserted into the model as time-series inputs, making this model structure highly appropriate. Moreover, from Jiao et. al. (2014)’s work, it appears knowing the number of vehicles on each of the upstream lanes improves the prediction of percentage distribution patterns downstream. For example, vehicles that enter the link on the left lane are expected to be more likely to turn left than those entering from the right lane. This means that the external (or exogenous) input of upstream turning percentages is useful for predicting this parameter as well, making a NARX model the appropriate architecture.

For the model to understand which task is the main task, and which are the secondary, the additional tasks inserted into an MTL model are given more leniency with regards to error goals (Caruana, 1997). This mitigates the issue of overworking the model to learn all tasks to the highest possible accuracy, allowing it to focus on optimizing its parameters to best predict the main variable while using some information inductively transferred from the other output variables. This is done by having the model multiply the performance parameter of the output variable (e.g. the mean squared error) by a percentage (e.g. 50%), so that it registers the error of the secondary output, for each iteration, as a value smaller than it actually is.

3.5. CONCLUSIONS

Using an iterative approach, starting from the simplest model architecture and incrementally increasing the complexity, the most appropriate model architecture for this study was found to be that of a NARX network. The number of memory cells included in this network, and the connectivity between the inputs and outputs are directly relevant to Robertson’s classical traffic progression model. Accordingly, this model framework is set up as a reconstruction of the analytical model. However, the advantage of a “machine learning” adaptation of this model lies within its flexibility to determine its own probability distribution function (and parameters) for platoon dispersion, unlike the analytical model which is restricted to mathematical assumptions. Hence, with this neural network, the assumption is that the logic behind the classical traffic progression model can now become more scalable to different traffic conditions. The validity of this traffic flow theory inspired architectural setup does not only lie within the accuracy of the final results, but from the insufficient outputs of the simpler architectures, which is why it was important to go through this process when building the architecture. Finally, only through proper model parameter setup and data preprocessing can the neural network properly process the data and attempt to find relationships. Trial and error was used to determine the ideal number of time steps back of memory to include; however, ultimately these trial and error tests concluded that the best output results come from matching the number of time step look back to the lower a lower limit of 20km/h for vehicle speeds. This way, all cars still on the corridor can be taken into consideration when making an arrival flow prediction for the next time step.

4. VERIFICATION, CALIBRATION AND TESTING OF PROGRESSION MODEL

In this chapter, the NARX neural network architecture is applied to real data, to verify that its applicability for traffic flow progression modeling. The design was calibrated in the sense that the number input and output nodes are updated to fit the new case study's geometry. As was the case with the previous case study, this chapter begins by introducing the case study and its relevance to this research approach. Next, in Chapter 4.3., a brief outline of the checks made, to assure that the selected architecture is still the most superior for real data, is provided. Following this verification check, a thorough analysis of the model's performance for this case study is made. The experimental setup and KPIs related to this study are defined in Chapter 4.4., while Chapter 4.5. displays the results themselves and their evaluation. The analysis includes some tests for its hyper-parameters and a comparison of its performance to the base-case model described in Chapter 2.6. Finally, the chapter conclusions are highlighted in Chapter 4.6. For the purposes of understanding the effect of the different parameters of this black box model, and interpreting its behavioral adaptations to different data complexities, this intermediate design step is quite necessary.

4.1. INTRODUCTION

Moving forward from completely controlled synthetic data, the first verification step for the model would be to see how it performs with real, stochastic, traffic data, using a geometrically simple case-study. Similarly to the previous case-study, the purpose is to evaluate how the model performs in an environment with the least amount of external impedances, such as the color of a traffic light, the existence of queues and/or the expansions in road geometry. This facilitates for a clearer understanding of how the model's performance changes with incremental increases to data complexities, and helps with the identification of the different conditions that affect the model's performance. Otherwise, more decisive conclusions on the conditions that are the root cause of errors cannot be made.

4.2. CONTEXT

Beatrixlaan is a long corridor that leads up to a signalized intersection with several detectors placed along several locations on this corridor. Hence by taking the section of the corridor upstream the intersection entrance, this case study becomes one of a simple two lane corridor, with no sources and/or sinks and no disturbances to the flow due to a traffic light (at the section used for this case study flow is not yet affected by queues caused by a traffic light). Accordingly, the traffic state for the data in this case study is free flow.

4.2.1. MOTIVATION FOR CASE STUDY

While the developed NARX model performed well with the synthetic case-study, this does not assure that the performance will be the same with real detector data. Accordingly, the first intention behind this case study is to verify two things: whether or not real traffic data, in free-flow, does, in fact, behave in a similar way as Robertson's traffic progression model; and, similarly, if the NARX neural network architecture is still the most appropriate for capturing this progression behavior, with a real dataset. Secondly, this study will provide some insights to how well the neural network adapts to traffic stochasticity, in comparison to Robertson's model, in an environment within which the classical model is known to perform quite well in. The limited disturbances affecting this dataset, in addition to the existence of data-points along different distance horizons, make it, also, ideal for testing the effect of the hyper-parameters, prediction horizon and time-step aggregation level, on the model's performance. Moreover, as each lane lead up to a certain direction of traffic flow, some insights on the differences between traffic streams can be made, without running into the uncertainty that queuing behavior could be the cause.

4.2.2. CASE STUDY CONDITIONS

Beatrixlaan is a simple urban corridor in Delft, that leads up to intersection K001. It is a long and undisturbed (i.e. no sources and sinks) urban road with several detectors widely spaced along of it, providing an optimal opportunity to test the neural networks ability to estimate longitudinal platoon dispersion. Within this case study, external factors such as controller light, the widening cross section of the road and the existence of queues are non-

existent to interfere with platoon dispersion modeling. Accordingly, variability in the dataset is limited to platoon disturbances that are caused by differences in driver behaviors. The corridor, and detectors, used for this case study are shown in Figure 35. Circled in red is the departure node, and in light green is arrival node one (250m downstream) and in dark green is arrival node 2 (350m downstream).

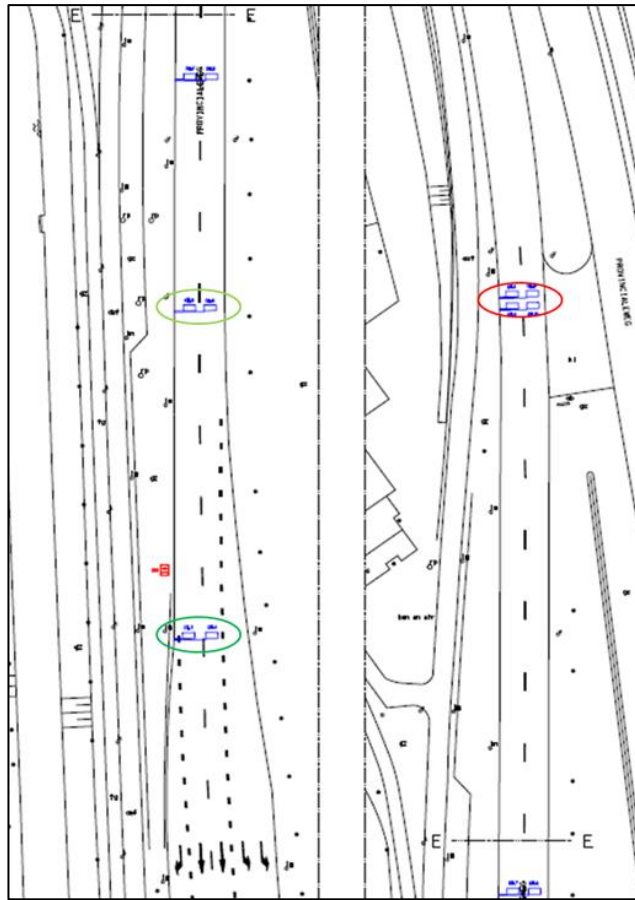


FIGURE 35: DELFT URBAN CORRIDOR CASE STUDY

4.3. EXPERIMENTAL SETUP & KPI'S

To assure that the selection of a NARX model architecture was in fact the correct choice for traffic progression modeling, three main architectures were tested: a feed forward neural network, a NARNET and a NARXNET. The selected arrival node for this validation test was arrival node one, which is 200m away from the departure node. Since the maximum speed for his corridor is 50km/h, the prediction horizon was estimated to be roughly 20s, by estimating the average speed to be around 80% of the maximum speed. A five second time step was selected for this model architecture test. The same architecture used for the synthetic case study was used for this one, with the only difference being that the number of output nodes were changed to two for arrivals and two for turning percentages, where previously they were four for both (i.e. architecture was calibrated for new case-study configuration). The lane closest to the two left turn errors was defined as Lane 2, while the other was defined as Lane 1.

Additional to the verification check for the model architecture, with this unique case-study, several tests can be conducted to evaluate model performance. The data comes from a semi-controlled environment, so the sources of dispersion that may affect the predictability of the data are limited. Additionally, detectors, which are a constraint with regards to prediction horizons, are placed at several locations. With these conditions the following tests can be conducted:

Test to see how well the model performs in comparison to Robertson's model, within an environment that the classical model performs the best: For this test, both the designed NARX neural network, and Robertson's model are used to predict arrival flows at the 200m downstream detector and the 350m downstream detector, and their performances are compared. Both models have recursive forms and rely on inputs from present and past time steps to make their predictions. Accordingly the number of input "look backs" for both models are matched during comparisons.

Test to see how the model's performance changes with different prediction horizons: For this, an analysis for differences in model performances between the 200m downstream predictions and the 350m downstream predictions is provided. When testing for the 350m prediction horizon, calibrations are made to the time prediction horizon, and accordingly the lag time it takes for feedback. As the prediction horizon becomes 30s instead of 20s, the lag in feedback is 6 time steps instead of 4. Additionally, the travel time difference between the free flow speed and the lower speed limit doubled. Accordingly, more memory steps were needed, and as memory was added the results for this horizon began to further improve. It is important to note that adding memory does not always improve the model's performance. Only the addition of relevant memory does. If additional memory is needed for the longer time-space horizon, the correlation between the traffic progression theory and the neural network is verified.

Test to see how to model performs with regards to traffic streams heading to different directions: For this, an analysis for the difference in model performances between both lanes (each of which lead to different directions of flow) is provided.

Test to see how the model's performance is affected by different data aggregation levels: For this test, two aggregation levels for the modeling time steps are tested: a 5 second aggregation and a 10s aggregation. Data was not aggregated less because it appeared that higher aggregations provide better results, and data was not aggregated more because, at higher than a 10s time step, there would not be enough prediction horizon left before vehicles reach the next successive intersection. The distance between urban intersections, in the Netherlands, is 500m – 600m, on average (Barthauer et. al., 2014) (Blokpoel & Vreeswijk, 2016), which does not leave a lot of space for further aggregation. In accordance to this new time step aggregation, the time-step-prediction horizon and the amount of memory time-steps was adjusted accordingly.

To train the neural network, input data was taken from one pair of the red-circled detectors in Figure 35, and the target data was taken from the light and dark green-circled detectors of the same figure (depending on the target horizon being tested). For both the neural network and Robertson's model (i.e. the reference case discussed in Chapter 2.6), data was clustered into time steps by adding up the detector counts within each 5 or 10 second interval. This data was not later converted to "veh/hr" to maintain the interpretability in errors that comes from calculating the average number of vehicle errors off per prediction time step. More so, now that the model architecture has been selected and validated, analytical tests were made to evaluate the model performance. While the graph charts give a visual indication of how well the predictions match the profile of the arrivals, they can be misleading. Only through proper statistical analysis, can conclusions about the model performance be made. Accordingly, the **key performance Indicators (KPIs)** were defined as follows:

- Root Mean Squared Error (RMSE): To identify the magnitude of the average model error
- Normalized Root Mean Squared Error (NRMSE): To give a proportional, or relative, indication of how severe the error is in comparison to the range of dataset being evaluated.
- R-squared parameter: To determine how well the distribution profiles of predicted flows matches the distribution profiles of actual flows.

4.4. VERIFICATION AND CALIBRATION OF THE ARCHITECTURE DESIGN FOR REAL DATA

The results of the verification check can be seen in Figure 36, Figure 37 and Figure 38 for the feed forward network, the NARNET and the NARXNET respectively. These results validate the outcome of the synthetic model test, which is that the NARX architecture is the most appropriate architecture for traffic progression modeling. The predicted arrivals are not as perfectly matched with the actual arrivals, as was seen with the synthetic data; however, this model still manages to best capture the profile of traffic. Also, it is important to note that the above results have been outputted only after the model had been trained for only two months of data, due to a lack of data availability, as a result of detector location changes and controller updates over the years. What is interesting to see though is that the simple feed forward neural network is capable of recognizing a relationship between downstream and upstream traffic, without the assistance of memory cells, better than a NAR network, which does include memory cells. This indicates that, traffic is in fact progressing with a clear pattern along a continuous corridor; and, as previously hypothesized, a purely time related pattern in arrival flows is quite difficult to detect with data of this level of stochasticity. Comparing the results of this case study with the previous one, it is clear that a lane configuration change (i.e. increase in number of lanes) is the main recognized obstacle for a feed-forward neural network, with

regards to traffic progression modeling. Similarly to the synthetic case study though, this obstacle will exist for the full intersection application intended for this neural network.

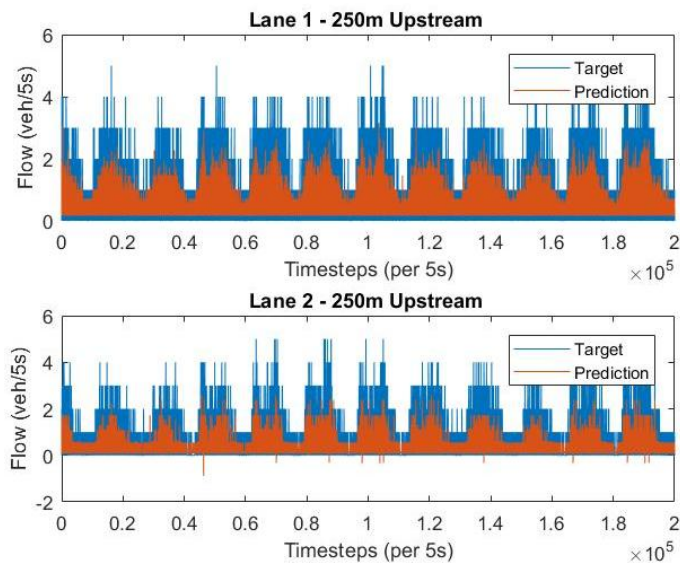


FIGURE 36: RESULTS FOR FEED FORWARD NEURAL NETWORK

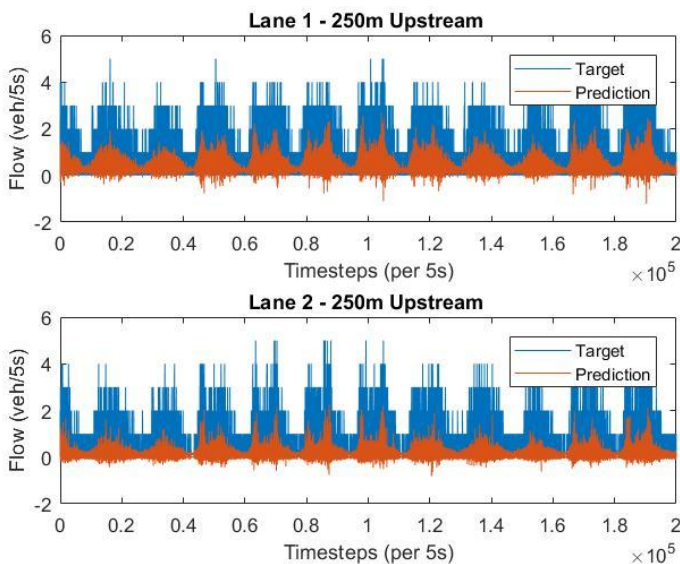


FIGURE 37: RESULTS FOR NARNET

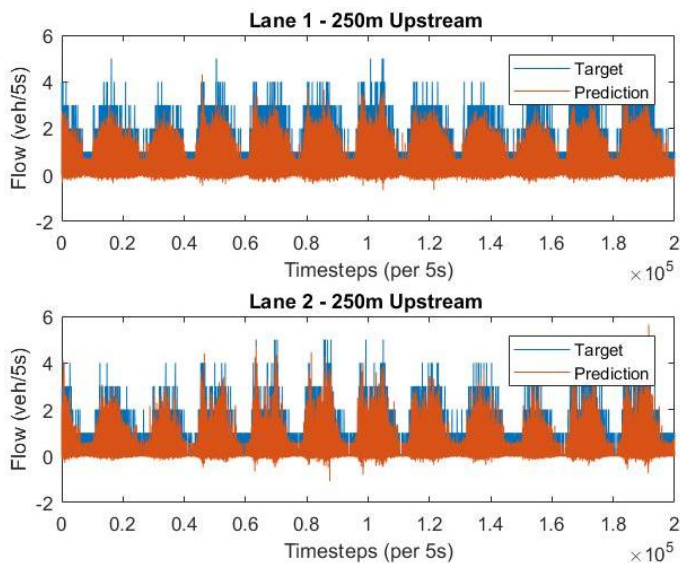


FIGURE 38: RESULTS FOR NARXNET

4.5. RESULTS AND EVALUATION OF DESIGN TESTING

Within this chapter, the results of the previously mentioned four tests are listed, and insights that were derived from the evaluation of these results are outlined. By the end of this chapter, a clearer understanding of the neural network's predictive abilities for different traffic streams and hyper-parameter settings is achieved.

4.5.1. TRAFFIC PROGRESSION CONDITIONS

TABLE 4: STATISTICAL ANALYSIS OF ROBERTSON'S PLATOON DISPERSION MODEL PERFORMANCE

| Lane – Distance | Selected σ (s) | R-Statistic | RMSE | NRMSE |
|-----------------|-----------------------|-------------|------|-------|
| Lane 1 – 200m | 5 | 0.79 | 0.37 | 1.34 |
| Lane 2 – 200m | 5 | 0.75 | 0.26 | 2.11 |
| Lane 1 – 350m | 16 | 0.62 | 0.42 | 1.69 |
| Lane 2 – 350m | 30 | 0.43 | 0.26 | 3.56 |

The interpretability of Robertson's classical, analytical model reveals information about the overall dynamics of platoon progression along both lanes. Accordingly, before comparing the results of both models together, several conclusions can be drawn from the results of Robertson's model calibrations and results alone. With regards to lateral progression, the calibration process revealed there was barely any. The percentage contribution of upstream flow measurements to each downstream flow measurement showed that the lowest RMSE was reached when 90 percent of the flow moving downstream was assigned from the same lane and only 10 percent was assigned from the other (e.g. 90% of the arrival flow on the left lane originated from the upstream flow profile on the left lane and 10% came from the right lane). This percentage contribution of upstream flows to downstream flows remained the same for both the 200m and the 350m prediction horizons. On the other hand, with longitudinal progression, clear differences can be seen between the speed variances at the 200m horizon and at the 350m horizon. At the 200m prediction horizon the amount of platoon dispersion on both lanes is quite low and very similar. However, by the time platoons reach the 350m horizon much wider travel time variances were estimated to reach the lowest RMSE's. Aside from this, it appears that the platoon dispersion on the left lane (i.e. Lane 2) are vastly higher than those for Lane 1; and, as dispersion levels (which are measured by the magnitude of the variance) increase, the R-statistic value decreases. Further insight on this is provided in Chapter 4.5.3. However, for this chapter, a comparison is next done with the results of the designed neural network, which are presented in Table 5.

4.5.2. INSIGHTS FROM COMPARISON WITH CLASSICAL MODEL

TABLE 5: STATISTICAL ANALYSIS OF NARX NEURAL NETWORK MODEL PERFORMANCE

| Lane - Distance | R-Statistic | RMSE | NRMSE |
|-----------------|-------------|------|-------|
| Lane 1 – 200m | 0.84 | 0.38 | 1.36 |
| Lane 2 – 200m | 0.75 | 0.26 | 2.09 |
| Lane 1 – 350m | 0.77 | 0.39 | 1.54 |
| Lane 2 – 350m | 0.56 | 0.27 | 3.77 |

With regards to the prediction accuracy of both models, when looking at the RMSE, it appears both the neural network and Robertson's model reach nearly identical performance levels. However, from the R-statistic, the main difference between the neural network and Robertson's Model can be seen. Robertson's model reaches this low error value by making predictions that closely match the majority condition, where the platoons are highly dispersed and there are low vehicle counts detected per time interval. This can be seen by the lower R-statistic value, which results from a lack of similarity between the predictions and actual data for larger volumes of arrival. On the other hand though, the R-statistic of the neural network reveals that this model's predictions better match the overall profile of arrivals. For the RMSE to be the same then, that means that higher neural network errors are made at the data range within which Robertson's model excels. This range where platoons are most dispersed, represents the free flow traffic conditions.

Prior to looking at the results, calibrations and tests were made to the neural network to adapt it to the new time-space prediction horizon. Firstly, as mentioned in Chapter 4.3., the time horizon for the prediction was changed to 6 time steps instead of 4. Secondly, trial and error experimentations were done to determine the quantity of relevant memory cells needed for this change. Around 9 time steps of memory were found to be ideal for the 350m prediction horizon. This is quite similar to the 10 time steps estimated, which verifies this connection to the traffic progression theory. However, for the feedback loop, it appeared that more than 4 or 5 time steps back of memory do not seem to add much additional value. This does not indicate that the neural network is not capturing the same pattern as the progression theory, but rather that it only needs to remember a certain number of arrivals to be able to capture the multiple-departures-to-a-single-arrival relationship visualized in Figure 33.

Now focusing on the differences in the neural network’s performance for the 200m and 350m, prediction horizons, the RMSE values in Table 5 are further evaluated. The results of this KPI show an average prediction error of less than half a vehicle, for the Neural Network, which indicates a high accuracy for predictions. However, they also entail that the predictive ability of the neural network is unaffected by the widening of the prediction horizon, which is an inaccurate claim. In compliment to the RMSE, the R-statistic reflects that the difference between the predicted and actual traffic arrival profiles actually increases with the prediction horizon. At farther away horizons, vehicles have vastly more opportunity to disperse and interact with traffic flow impedances, which causes the actual arrival profiles become more spread out; and, similarly to the case with Robertson’s model, as the flow of vehicles gets more dispersed, the variance in the flow profile grows. From the arrival flow profiles of the actual data, which are shown in Table 6, it is clear that much of the 350m arrival profile consists of low counts per interval, due to spread out platoons; but, there are certain instances where higher counts (i.e. less dispersed platoons) are detected, creating peaks that are much different than the remainder of the data. Therefore, the traffic progression behavior during these instances differs from the more prominent pattern appearing the vast majority of the time, making it difficult for the neural network to predict them, given that they make up such a small percentage of the training data. However, while these spikes are few, in comparison to the rest of the data, they occur with enough consistency over the days to not be considered as outliers.

TABLE 6: THE PERCENTAGE FREQUENCY OF VEHICLE ARRIALS FOR THE FOUR DOWNSTREAM MEASUREMENT POINTS

| Flow (veh/5s) | Distribution (%) Lane 1 – 200m | Distribution (%) Lane 2 – 200m | Distribution (%) Lane 1 – 350m | Distribution (%) Lane 2 – 350m |
|---------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| 0 | 73,180 | 87,173 | 74,791 | 91,941 |
| 1 | 20,476 | 10,695 | 19,927 | 7,329 |
| 2 | 5,323 | 1,777 | 4,644 | 0,676 |
| 3 | 0,933 | 0,309 | 0,610 | 0,052 |
| 4 | 0,084 | 0,043 | 0,028 | 0,002 |
| 5 | 0,004 | 0,004 | 0,000 | 0,000 |
| 6 | 0,000 | 0,000 | 0,000 | 0,000 |
| 7 | 0,000 | 0,000 | 0,000 | 0,000 |

While the lower R-statistic is useful in revealing the worsening match between the predicted and actual arrival profiles, due to the widening variance in arrival patterns, information is still needed on whether or not the RMSE does, in fact, still match the remaining majority of the data, at the 350m horizon, with as high of accuracy as it does for the 200m horizon. This information can be derived from the NRMSE, which provides a value for the magnitude of error that is relative to the mean value of the dataset. While this value isn’t a physical measurement of prediction error, it does provide an indication of how significant the error is with regards to the average data range. As platoons become more dispersed along the corridor, the arrival rates spread out; and so, the average volume of arrivals every five second time interval decreases. Accordingly, an RMSE of 0.26 at the 350m horizon of Lane 2 is much more significant than an RMSE at the 200m horizon (i.e. where the average volume of arrivals is higher every five second interval), as shown by the NRMSE. This means that it is not only the peaks that are being missed, but that the overall prediction accuracy lessens with a longer prediction horizon. These results indicate that the conclusions of Qiao et. al. (2001)’s

study on a neural network's performance with regards to traffic flow progression modeling should be analyzed critically, due to their very short, 89m, prediction horizon.

4.5.4. INSIGHTS FROM DIFFERENT DIRECTIONS OF FLOW

One of the consistencies in the results is that the predictability of Lane 2 is always lower than that of Lane 1. This is actually not coincidental, nor specific to this case study. Figure 35 shows that Lane 2 actually leads up to the left turning lanes at an intersection. As was outlined in Chapter 1.2.2., the less saturated a traffic stream is, the more difficult it is to predict its short-term arrival profile, due to the lack of consistency in low demand arrival patterns. Accordingly left/right turning flows, which are far less dense than straight ahead moving flows, tend to have more stochastic arrival profiles (Krijger, 2013). For the majority of the time, this traffic stream receives very few vehicles, at low rates, but, at certain time stamps, longer platoons arrive, with no recognizable pattern. This high variance in arrival flow profiles of a left-turn lane can be visualized in Figure 39, where Krijger (2013) displays the stochasticity of green time durations displayed by a left-turn signal head of an adaptive controller. The controller outputs higher green times as a response to longer platoons, and accordingly the green time durations do not follow a clear probability profile. This Krijger's results [36] are therefore consistent with the results of this study.

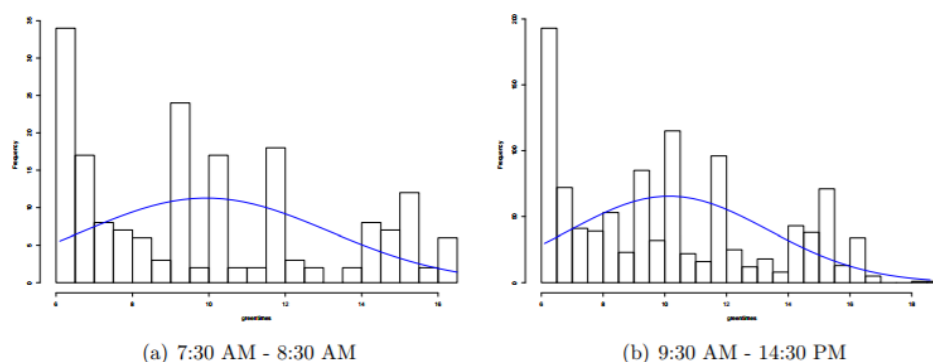


FIGURE 39: A LEFT TURN CONTROLLER'S GREEN TIME DURATIONS SCHEDULE, FOR AN OFF-PEAK HOUR (A) AND SOME PEAK HOURS (B), AS PRESENTED BY KRIJGER (2013)

The parallels between the conclusions of Krijger (2013)'s study with regards to traffic streams, flow density and predictability, can be seen in Figure 40. As shown, the least visually matched predictions are the ones for the lane heading towards the right turn at the 350m prediction horizon, despite using the same prediction model. While the image does not show details of prediction accuracy, it does visualize how the prediction profile, similarly to Figure 39, is incapable of adapting to the variances caused by the peaks. Likewise, the lower R-statistic match can be seen between the 200m and 350m prediction horizons, as discussed in Chapter 4.5.3.

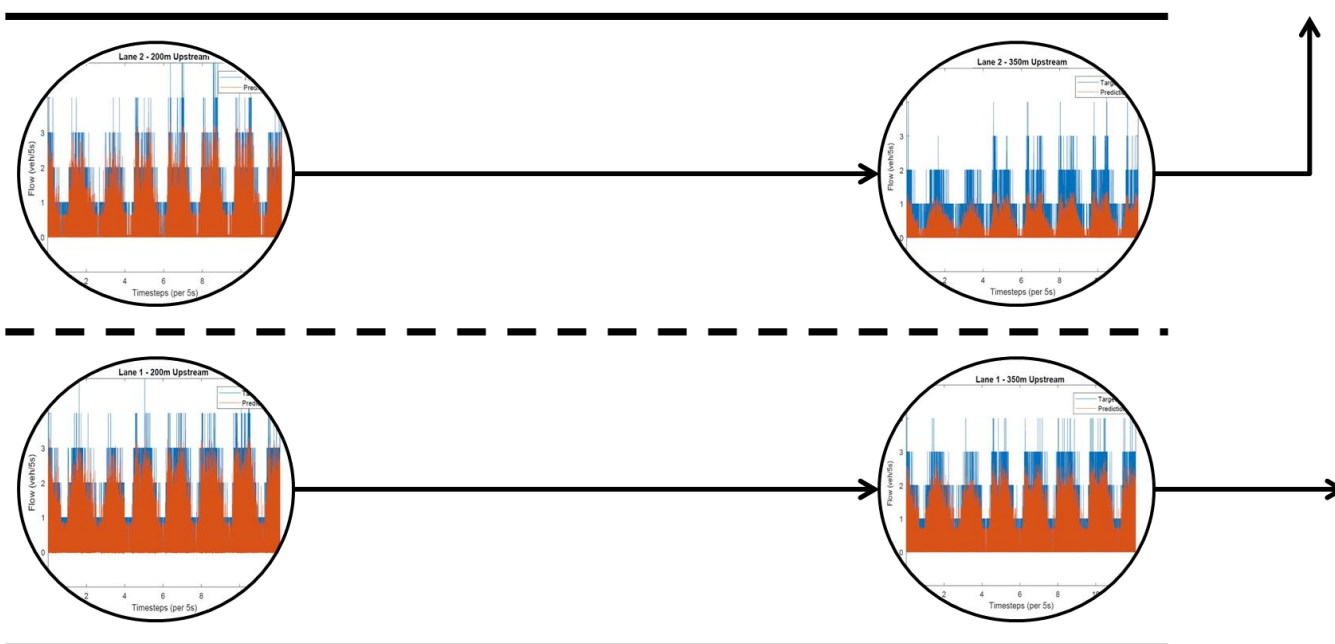


FIGURE 40: TRAFFIC PROGRESSION FOR STRAIGHT AHEAD AND LEFT TURNING FLOW STREAMS

The time step hyper parameter for this model was initially taken as five seconds. This time step was widened further to determine how it would affect the data predictability. While the RMSE values are slightly poorer, this is not an indication that the more aggregate time step provides lower predictability. By doubling the time-step aggregation, the number of vehicle counts that can be included within one bin increases, which increases the dataset range. Accordingly, the assistance of the other two KPIs is once again needed. When combining this outcome with the R-statistic values and the NRMSE, it appears that the wider time step makes the model prediction outputs more robust. More aggregate time steps are known to aid in smoothing out stochasticity in data, which appears to allow the neural network to recognize flow profiles better, as shown from the R-statistic; and, more consistently provide higher accuracy outputs, as shown from the significant improvement in the NRMSE, which is scaled to the data range of each sample set.

TABLE 7: TESTING THE EFFECT OF CHANGES TO THE TIME-STEP HYPERPARAMETER

| Lane – Distance – Time Step | R-Statistic | RMSE | NRMSE |
|-----------------------------|-------------|------|-------|
| Lane 1 - 350m – 5s | 0.77 | 0.39 | 1.25 |
| Lane 2 350m – 5s | 0.56 | 0.27 | 2.99 |
| Lane 1 - 350m – 10s | 0.82 | 0.41 | 0.97 |
| Lane 2 350m – 10s | 0.60 | 0.29 | 2.18 |

A wider time step than 10s was not tested due to space constraints. As previously mentioned, the prediction horizon for the intended model application is constrained to the distance between one intersection and the next. The distance between two consecutive urban intersections within the Netherlands is often within the range of 500-600m (Barthauer et. al., 2014) (Blokpoel & Vreeswijk, 2016). Depending on the speed limits, and average vehicle speeds, between two intersections, the prediction time horizon is expected to be around 25-35s. Accordingly, if wider time steps are taken, then by the time the neural network has finished counting the vehicles passing over one intersection, they would have already arrived at the next, defeating the purpose.

4.6. CONCLUSIONS

Several conclusions can be drawn from this study. Firstly, this study verifies that Robertson’s model can, quite accurately, progress traffic flows, from one point to the next, under free flow conditions; and, correspondingly, the proposed NARX neural network architecture, which is formed as a reconstruction of this model has been verified as the most suitable neural network model architecture for traffic flow forecasting. With regards to parameter calibration though: number of input and output nodes need to match the case study configuration, the prediction horizon must match the free flow travel time between the input and output points, and the cell quantity of input memory must be calibrated to assure that the neural network remembers all previous vehicle departures, up to the lower speed average of 20km/h. With regards to the feedback loop, 4-5 memory cells suffice.

Secondly, the results of this study show that, when the parameters of each model are optimized to the lowest RMSE, they tend to adapt to the most prominent dispersion pattern in traffic. This can be seen at farther away prediction horizons, as platoons get more dispersed, and yet certain instances occur where platoons arrive more closely together, differing from the majority condition. This behavior is even more apparent on the left turn lanes, due to the lower density on these lanes, which causes higher variability in platoon movements. However, especially for these more difficult to predict conditions, the NARX neural network has been concluded to be more robust than Robertson’s model, due to its ability to better adapt to the variances in traffic flow arrivals, while maintaining a low RMSE. Hence, in the environment where the classical model performs at its strongest, and the results show that it in fact does perform quite well, the NARX neural network is superior. This removes the doubt, for more complex case studies, that differences in the performance of both models could only be due to the fact that the classical model is not placed within the environment within which it was designed. Finally, more aggregated time steps are recommended for further robustness of the NARX networks performance. With confirmation that improvements in predictions can be achieved from the new proposed model, even in the environment the classical model was designed for, it is justifiable to move forward to a more complex case-study, to determine how the performance gap is affected.

5. REENGINEERING, APPLICATION AND EVALUATION OF PROGRESSION MODEL

Now that the model design has been verified, and it has been concluded that the NARX neural network can outperform the classical model within a free flow environment, due to its higher robustness, the model can be applied to a full intersection case study, to determine how its performance will be affected by the existence of a traffic light; and, the extent to which the neural network model will be able to better adapt to the traffic flow disturbances caused by a red light than the classical model. Hence, the chapter first begins by introducing the full intersection case study used, the motivation for picking it and the conditions. Next, in Chapter 5.3., the experimental setup is explained and the KPIs used are referenced again. Chapter 5.4. then explains the modeling expansions made to both the neural network and the classical model, to adapt them to the additional complexities of this case study. Later, Chapter 5.5. presents the results of this case study, and their evaluation. For a thorough investigation of the model's performance, the results were analyzed with regards to each of the multiple traffic states that exist at the entrance of an intersection, individually. Additionally, an analysis of how the robustness gap between the two models increases when a traffic light is included is made. Finally, conclusions for the whole of Chapter 5 are made.

5.1. INTRODUCTION

According to the research methodology, complexity is gradually added to the dataset used to test and evaluate the progression model, for better interpretability of the model's behavior towards impedances, in comparison to the classical model. Accordingly, the ideal case would have been to extend the model's prediction horizon further, while still using the Delft case study, so that it would predict traffic arrivals at the intersection, post the point where the number of lanes expands. Predictions at this point are more difficult, as traffic flow is affected by the backwards propagation of queues there. However, as can be seen in Figure 35, there were no short, far away, loop detectors after the lane expansion for this case study. Hence, there was no data available to be used as a target for neural network training, nor to use for error calculations between predicted and actual flow arrivals. Long loop detectors are not helpful for providing vehicle counts per intervals, since detectors only provide binary "occupied" or "unoccupied" data. Accordingly, if several vehicles are on top of a long loop detector, it will only read "occupied" without giving any information on the number of vehicles. Moreover, the nonexistence of far away loop detectors indicates that the controller for this intersection is an actuated controller, not yet an adaptive one, as targeted for this study (following the controller upgrade trends).

For these reasons, a new study was sought out. This new study is of a full intersection in Noord Holland, which connects the N201 provincial road with the N205 provincial road together. Predictions for traffic on all directions of this case study were made, which not only provided insight on how scalable to the model is to different traffic cases, but was additionally necessary for the evaluation that will be done in Chapter 6, where the predictions will be used as inputs for a modeled predictive controller. Prior to doing so however, a thorough statistical analysis of the prediction results will be made in this chapter. Accordingly, this chapter serves as the first step towards developing the predictive controller concept design outlined in Chapter 1.3.

5.2. CONTEXT

The intersection used as a case-study for this research was the one connecting the N205 motorway to the N201 motorway in Noord Holland (i.e. intersection 201234), which is circled red in Figure 41. It is fed into by three other intersections (also shown in Figure 41): Spaarnepoort (circled orange, to the left of 201234), Leenderbos (circled yellow, to the right of 201234) and Verbindingsweg (circled green, from the top of 201234). These intersections are also numbered: 201231, 201209, and 205195 respectively. There are several modalities approaching this intersection, however each modality is completely isolated to its own lane, and is controlled by its own separated signal head. This makes it simple to ignore the presence of these additional modalities, according to the scope of works, as will be explained further in Chapter 5.3.

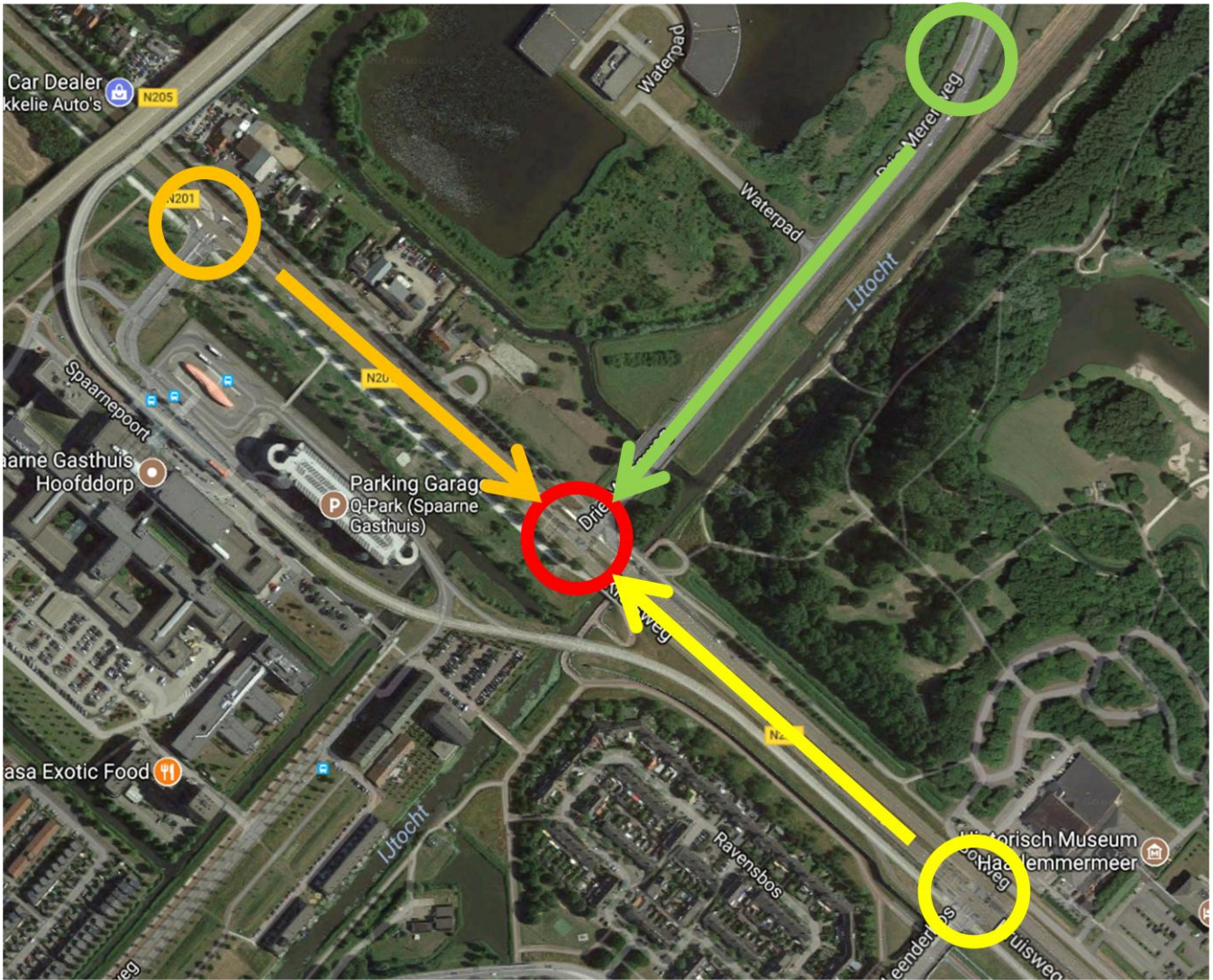


FIGURE 41: GOOGLE MAPS SATELLITE IMAGE OF CASE STUDY NETWORK

5.2.1. MOTIVATION FOR CASE STUDY

This specific intersection was selected for several reasons. Firstly, only on a single direction did it have one source/sink. While traffic coming from both Spaarndam and Leenderbos is entirely detected, there is a parking space between Verbindingsweg and the case study intersection, as can be seen in Figure 41. The performance of the neural network relative to Robertson’s model on this direction, in comparison to the others, will give some indication on whether or not the neural network is better able to better perform in the case of sources and sinks. Secondly, the traffic conditions on the six individual traffic streams at this intersection were different, providing a variety of traffic situations for evaluating the comparative performance of both the baseline model and the designed prediction model. Additionally, the detector configuration satisfied the needs of the study, there were far away loops that were specific to each traffic stream. This both indicated that the controller was adaptive, and provided the needed data source for the neural network to use for training, receive information for its open feedback loop from, and to check the prediction errors against. Lastly, representatives from the Province of Noord Holland were quite supportive of this project, and were willing to provide any resources needed..

5.2.2. CASE STUDY CONDITIONS

The three feeder intersections are at different distances from the case study, but due to the speed-limit differences between Dri Merenweg and Kruisweg, a roughly 30s prediction horizon is equally established from all sides (ranges were rounded from 26 – 32s). At the intersection itself, the lanes expand from two to four (or, in the case of Verbindingsweg’s side, three). While far away loops exist for each traffic stream after the expansion, they are not lane specific, but only direction specific, loops. Accordingly, as shown in Figure 42, these loops cover two lanes each.

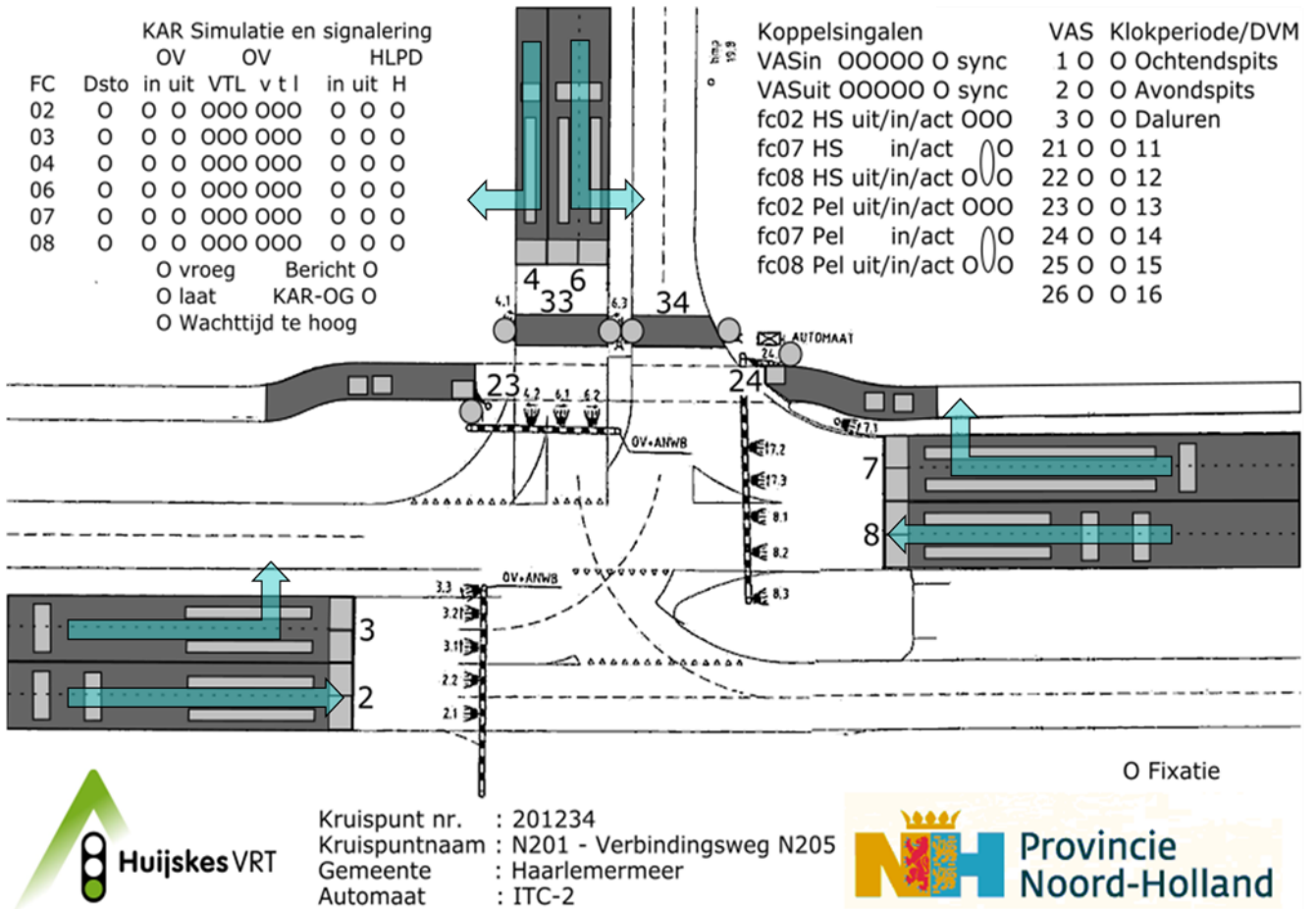


FIGURE 42: SCHEMATIC DRAWING AND SIGNAL HEAD CONFIGURATION FOR CASE-STUDY

The controller installed on this intersection is a quite new Vialis controller that was put in place at the start of 2017. While this makes the selected case study quite relevant, it also means that there is no usable V-log data for this intersection before 2017. Accordingly, a maximum of five months of data was available in total (when this data was collected from the Province). The biggest advantage of this case study though, was that this controller followed the highly adaptive vehicle actuated control strategy outlined in Appendix B; and, with a similar detector configuration.

With regards to traffic flow progression between the intersections, from the Spaarnepoort and Leenderbos feeder intersections, there were four lanes whose direction of flow, when departing, leads to the N201/N205 intersection. From the Verbindingsweg feeder, there were three. At the center destination intersection though there were a total of six points of arrival, since each of its three sides contained two possible directions of flow. The combination of eleven departure points with six arrival points was a clear indicator that there wasn't a 1:1 relationship between the Origin-Destination (OD) pairs, as shown in Figure 43. Both the neural network and the classical model had to both be calibrated to take into account this input-node-to-output-node relationship. For the neural network, this is accomplished through the definition and connectivity between the neurons. For Robertson's model, additional expansions to the model are required.

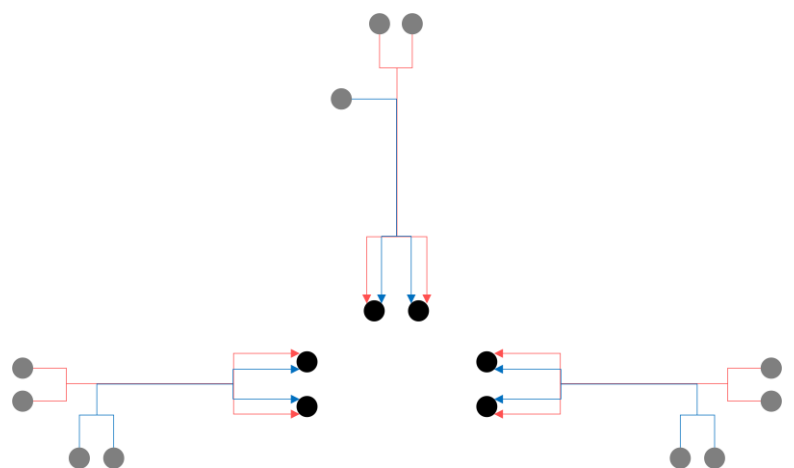


FIGURE 43: O-D PAIR COMBINATIONS OF FLOWS MOVING FROM EXIT POINTS OF ONE INTERSECTION TO THE ARRIVAL POINTS OF THE NEXT

For Robertson's model, additional expansions to the model are required.

The complexity of this dataset required reengineering of the original, validated design. With the existence of a traffic light controller, traffic flow is disturbed and queues of different lengths are formed. To account for this effect, modeling expansions were implemented to both the neural network and Robertson's model. Additionally, unlike in the previous case study, which was of an urban corridor, there are several corridors and lanes involved, including different modalities. Hence there were several aspects that needed to be dealt with, when making the experimental setup. They can be summarized as follows:

Scoping: In accordance to the scope of works, the alternative modes controlled by signal heads 23, 24, 33 and 34 were ignored for simplification. This means that predictions for arrivals of other modes than vehicles were not tested, and accordingly, in Chapter 6, these out of scope traffic streams were removed from the model. The configuration of all signal heads, and their labels can be found in Figure 42, and the ones specifically being used for this project are the ones with the following numbers: 2, 3, 4, 6, 7 and 8. In correspondence with these labels, from here on in, any arrival flow mentioned will be referenced in the following manner: $Q_d(\text{"Signal Head No."})$.

Time-step Selection: Since, from the last case study, the results showed that wider time steps result in more robust model prediction performance, the time step for this case study was defined as 10s. As previously mentioned, this time step was not further expanded so as to not lose too much time from the prediction horizon in measuring vehicle counts.

Data Division: 60% of the data was allocated for training, 20% for validation and 20% for testing. The 20% of the data left used for testing will be statistically evaluated; and then, parts of it would be used in Chapter 6 for purposes of testing and evaluating a concept design.

Calibrating: While the uniformity in the time prediction horizon facilitated the process of inserting controller predictions at the same time-horizon, in Chapter 6, the difference in the free flow speeds between the two roads entailed that more memory was needed for the higher speed corridor (i.e. N205) to reach the lower speed limit of 20km/h. Before systematically calibrating in this way, trial and error tests were made with the memory cells, to assure that the methodology derived in Chapter 4 remains consistent for a new case study. The **KPI for calibrations** was the RMSE. The number of input "look backs" selected for the neural network are also applied to the similarly recursive formulation of Robertson's model.

Reengineering (i.e. Modifying Features): Due to the inclusion of a traffic light in this case study, which disturbs traffic flow and results in queue propagation, both the neural network and the classical model (as it is the reference point) need to be reengineered in response to the new impedances. The skeleton architecture (or algorithm in the case of the classical model) remain intact, but expansions are added to them. Details of this process will be explained in Chapter 5.4. The **KPI for the reengineering** was the RMSE.

Progression Model Evaluation: The data for this case study is far more complex than before. It includes several lane directions and different traffic conditions. Accordingly, three different evaluations were made for the traffic progression model: firstly the results were generally evaluated as a whole, afterwards, they were divided per traffic state (using a model that will be proposed in Chapter 5.5.2.1.); then, finally a robustness evaluation was made by assessing how the model performs per intensity of arrivals, and how frequently it makes different magnitudes of errors. All three evaluations were made comparatively to the classical model. For the first two evaluation tests, the KPIs described in Chapter 4.3 were used, while, for the last one, percentage improvements from the classical model was the KPI considered.

5.4. MODELING EXPANSIONS FOR TRAFFIC LIGHTS AND QUEUES

Unlike the Delft case study, since this is a full intersection application, other external factors than varying driving speeds can affect the longitudinal dispersion of vehicles. Mainly, these factors include the existence of queues already at the downstream intersection and the state of the traffic light. Hence, expansions to both the neural network, and the classical model used as a bench mark, were ventured. To expand the neural network, more information was added into the model, with regards to traffic light phases and the existence of queues. Additionally, through trial and error, more neurons were added to the hidden layer until the optimal number of neurons was

reached. When working with neural networks, there is no other procedure that can be followed for defining the size of the hidden layer other than going through trial and error. Of course, calibrations to the prediction horizon and memory cells were made as well, as before. On the other hand, for Robertson's model, inspiration was taken from Bie et. al. (2014)'s study, where they suggested that, at complex signalized intersections, the platoon dispersion model should be applied twice: once for the distance from upstream to the last point where the number of lanes remains the same, then once again, with newly calibrated parameters, for the distance from the start of the road expansion to the intersection stop line. Details on these expansions are covered in this section.

5.4.1. REENGINEERING AND APPLICATION OF THE NEURAL NETWORK ARCHITECTURE

Before adding features to the architecture, the model was first tested with its original design. The appropriate calibrations were made, while making sure to test that the calibration process derived from the previous two case studies is still valid. By testing different memory cell quantities, the conclusion was that it was. For the three directions of arrivals, adding more than 5 memory steps for the feedback resulted in poorer model performance. As was the case in the Delft case study, the number of time steps back for the feedback is not correlated to a certain time duration, but is related to the number of arrival instances the neural network appears to need to remember to begin recognizing a pattern between several inputs (from different departure times) and the target arrival. Similarly, the input memory calibration was also consistent with the previous case study. Significantly larger memory steps were needed to account for the higher free flow speeds and longer space horizons for each of the three arrival directions (from the left, right and top as shown in Figure 41) to achieve the lowest RMSE. Taking the top direction as a reference (i.e. the flow coming from Verbindingsweg), around 10-12 time-steps worth of memory cells had the same performance, and 10 were estimated to include the vehicles traveling at the lowest speed limit. Similarly, for the other two directions the number of needed time steps of memory estimated per the 20km/h lower limit was 8, while the trial and error results concluded that 9 was the most optimal, but only negligibly from using 8 memory cells. Accordingly, the methodology of basing the memory cells on the lower bound speed has been validated. This calibrated architecture can be seen in Figure 44.

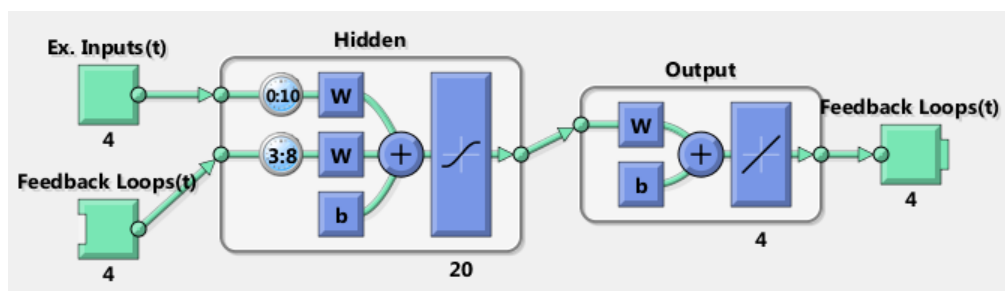


FIGURE 44: PHASE 1 OF PROGRESSION MODEL REENGINEERING (EXAMPLE FROM VERBINDINGSWEG SIDE)

Initially, three architectural modifications were made. The first was to have the model predict for a single traffic stream rather than both for each arrival direction. This contradicts the conclusion made in Chapter 3.4.2.1. that having the model output the arrival flows of all traffic streams coming from the same direction, after the flow splits up, improves its prediction abilities. The reasoning though is that, the different traffic light phases outputted, and the different queue behaviors occurring, for each traffic stream, makes it so that the correlation between their arrival flow patterns weakens. However, having the model simultaneously predict the percentage contribution of flow allocated to the target traffic stream was still found to be significant for the model's performance. Accordingly, the multitask architecture was reduced as follows: rather than having four outputs (2 traffic flow arrivals and 2 percentage contributions), the arrival flows of only one traffic stream, accompanied by the percentage flow assigned to it, are placed in the output. The second architectural modification was to add time patterns. The model can therefore recognize patterns of time of day by having a 1 to 8640 value inputted, corresponding to the 10s time step of the day it is; and, a 1 to 7 value inputted, corresponding to the day of the week. For chaotic time series, these inputs are useful to aid the model in recognizing behaviors that correspond to certain times of day or certain days of the week. The last modification made was to the number of neurons in the hidden layer. Through trial and error, the size of this layer was optimized to 55 neurons. The KPI used to monitor the model performance through this reengineering process was the RMSE. Similarly to the process of designing the network itself, each of these features was changed gradually

and the performance of the NARX neural network was monitored. The mentioned architectural modifications can be seen in Figure 45.

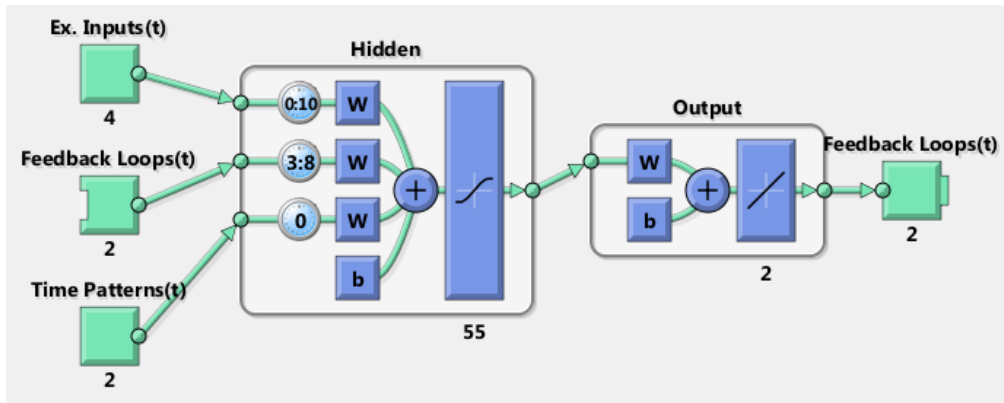


FIGURE 45: PHASE 2 OF PROGRESSION MODEL REENGINEERING (EXAMPLE FROM VERBINDINGSWEG SIDE)

So far, the modifications made to the architecture of the neural network do not yet address the main sources of arrival flow disturbances: red traffic lights and the existence of queues. As is the case with the percentage contribution of the total flow to each arrival traffic stream, the state of the traffic light phase cycle and the queue conditions that may disturb arrival flows take place at the future prediction horizon. Accordingly, they are added as multi-task learning features as well. With this expansion, the model is required to predict the following secondary targets, in total: the percentage contribution to the traffic stream, the phase cycle state of the traffic light at the start of the target 10s time step, and whether or not the long loop detector of each relevant lane will be occupied (i.e. a binary target prediction) at the start of the target 10s time step. For the last feature, two outputs are required, corresponding to each of the two lanes allocated to a single traffic stream. For the phase cycle state input, this information can be extracted from the controller’s WUS, which is the V-log data of the external state it displays. The intention behind adding the two latter secondary prediction tasks into the model was to further guide the neural network towards implicitly recognizing the external factors that affect platoon dispersion in the last few meters of platoon trips. As previously mentioned, since the main target, which is short-term vehicle arrival flows, is a function of the mentioned secondary tasks, predicting these added information improves the performance of the neural network model for its intended purpose. In total, as shown in Figure 46, this sums up to five target outputs (i.e. 4 secondary outputs and 1 main output).

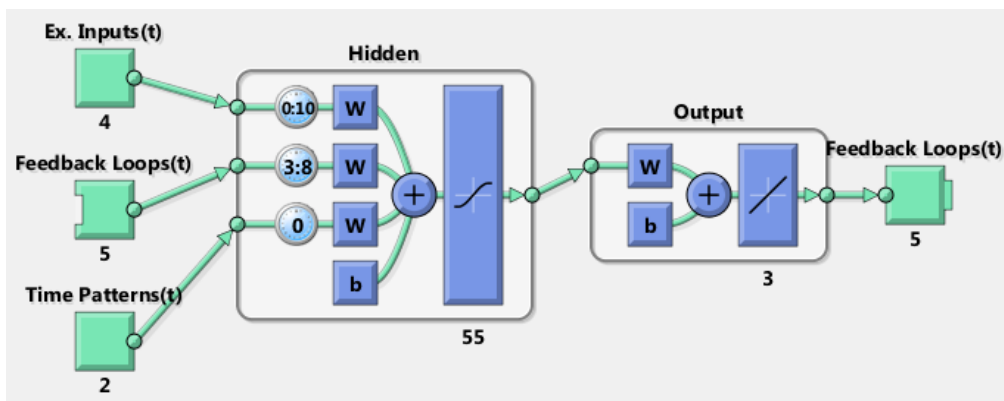


FIGURE 46: FINAL PROGRESSION MODEL ARCHITECTURE FOR FULL INTERSECTION APPLICATION (EXAMPLE FROM VERBINDINGSWEG SIDE)

The added queue and WUS information was highly significant for the neural network. Not only did it lead to improvements in the RMSE of the test set, but by comparing the R-statistics for the training, test and validation set before and after these multi-task outputs were included, it appears that, without them, the neural network has trouble finding a relationship in this more complex data, as shown by the large gap between its performance for the training and test sets. This gap indicates the model is memorizing rather than recognizing and learning patterns of progression. Alternatively, when the added secondary targets are included, the gap between the training and test sets for the different traffic streams significantly narrows. Hence, the architecture presented in Figure 46 is the final, reengineered model architecture for intersection-to-intersection traffic progression applications. This model can be

used for any case study intersection, provided the required departure and arrival data is available, and the prediction time horizon and the number of input memory cells are calibrated.

5.4.2. EXPANSION, CALIBRATION AND APPLICATION OF THE CLASSICAL MODEL

In correspondence to the calibration choices made within the neural network, the number of “look-back” measurements (i.e. flows from previous time steps) that were included in the neural network through the memory cells, will also be included within the calibration of the famously used classical model. In its recursive form, this model also relies on flow information from past time steps to forecast future arrival flows at a certain time-space horizon. However, a different, more analytical, expansion process must be followed to also aid this model in implicitly recognizing the effect of queues and signal heads on flows at intersections. In order to do so, as per the nature of the model, its longitudinal dispersion parameters must be calibrated in a way that would allow the model to recognize the disturbances both queues and red traffic lights impose onto free flow traffic. The most appropriate calibration methodology for this type of impedance found in literature was that from Bie et. al. (2014)’s study on traffic progression modeling between intersections in a network. In their study, they propose that the platoon dispersion model be applied twice on incoming traffic flows. First to progress flows along the uniform portion of the corridor, and then once again to disperse the flows after they have divided into different traffic streams, with the expansion of the number of lanes, at the entrance to urban intersections. For each time the model is applied, the dispersion parameters are calibrated differently, to account for the different conditions. This methodology can be best summed up by Figure 47, which was included in Bie et. al. (2014)’s paper, for the same clarification purpose.

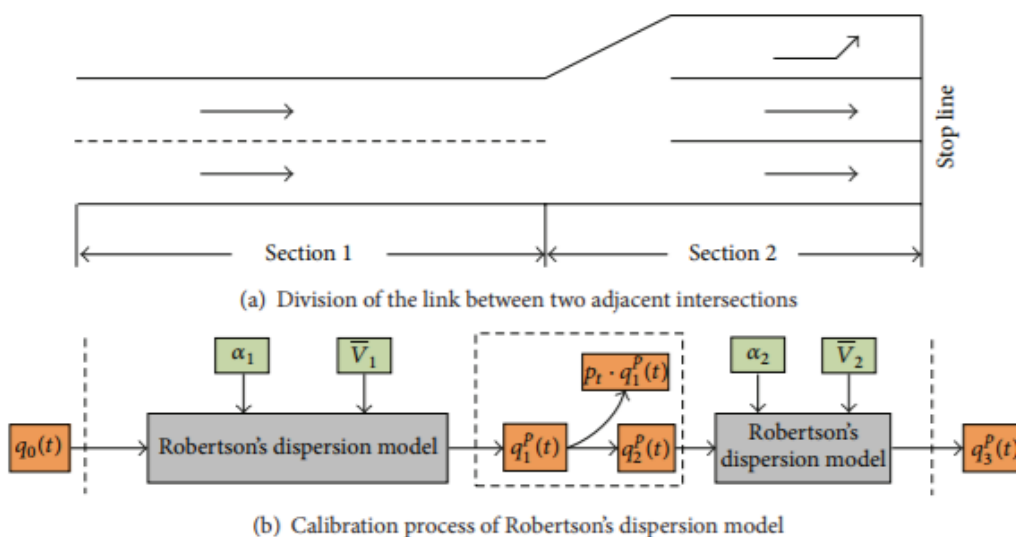


FIGURE 47: BIE ET. AL.'S PROPOSAL [9] FOR HOW TO BEST USE ROBERTSON'S DISPERSION MODEL FOR FULL INTERSECTION APPLICATIONS

Based on this methodology, a means to disperse traffic laterally, after the lane expansion is needed, Since there are four (or three in one case) departure nodes heading to six target nodes, as shown in Figure 43, the contribution of upstream flows to each lane downstream, post the expansion, was derived from a matrix of OD pair weights, which is displayed in Table 8. From this matrix, the contribution the flow from each departure node has on the flow at each arrival node could be estimated. The weighted sum of the upstream departure flows multiplied by their weight of contribution determine the “ q_u ” input variable in Seddon’s recursive restructuring of Robertson’s model. These weights were estimated through trial and error until the lowest RMSE for the predictions on each traffic stream are achieved. Since there were no detectors along the corridor (i.e. prior to the lane expansion) it was not possible to check if a different lateral flow distribution occurred along the corridor itself. However, based on the analysis made it Chapter 4.5., it is likely that flows heading to a certain traffic stream all collected at the lane along the corridor closest to their intended direction of movement, post the lane expansion. Hence this was the assumption made for the lateral flow split on corridor portion of the expanded platoon dispersion model.

TABLE 8: OD PAIR WEIGHTS OF CONTRIBUTION BETWEEN THE VARIOUS DEPARTURE AND ARRIVAL FLOWS OF VEHICLES

| Departures/Arrivals | 201234 Arrival Traffic Stream 7 | 201234 Arrival Traffic Stream 8 | 201234 Arrival Traffic Stream 2 | 201234 Arrival Traffic Stream 3 | 201234 Arrival Traffic Stream 4 | 201234 Arrival Traffic Stream 6 |
|------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
| 201231 – Departure Lane 1 | 0 | 0 | 0.15 | 0.40 | 0 | 0 |
| 201231 – Departure Lane 2 | 0 | 0 | 0.15 | 0.40 | 0 | 0 |
| 201231 – Departure Lane 3 | 0 | 0 | 0.35 | 0.10 | 0 | 0 |
| 201231 – Departure Lane 4 | 0 | 0 | 0.35 | 0.10 | 0 | 0 |
| 201239 – Departure Lane 1 | 0.3 | 0.15 | 0 | 0 | 0 | 0 |
| 201239 – Departure Lane 2 | 0.3 | 0.15 | 0 | 0 | 0 | 0 |
| 201239 – Departure Lane 3 | 0.2 | 0.35 | 0 | 0 | 0 | 0 |
| 201239 – Departure Lane 4 | 0.2 | 0.35 | 0 | 0 | 0 | 0 |
| 205195 – Departure Lane 1 | 0 | 0 | 0 | 0 | 0.60 | 0.40 |
| 205195 – Departure Lane 2 | 0 | 0 | 0 | 0 | 0.20 | 0.40 |
| 205195 – Departure Lane 3 | 0 | 0 | 0 | 0 | 0.20 | 0.20 |

In parallel to finding the most appropriate weights, two smoothing factors need to be estimated, for the two stages of dispersion platoons go through (i.e. before and after the lane expansion), in accordance to Bie et. al. (2014)'s methodology. Essentially, as conceptualized in Figure 47, the formulation is set up so that the output of the first application of Robertson's dispersion model is used as input for the second one. This allows the model to isolate the dispersion caused by traffic lights and queues and adjust the model dispersion parameters to it accordingly. For the second dispersion stage, the flow should be dispersed laterally to account for the expansion in the number of lanes. However, as is the same limitation with the neural network, it is not possible to do so since the detector covers each lane pair representing the same traffic stream, after the expansion, as shown in Figure 42. The two smoothing factors are calibrated using Equation (5), and, similarly to the neural network's optimization process, the KPI for estimating the optimal speed variances, which are displayed in Table 9, was the RMSE. These variances were determined by trial and error.

TABLE 9: IDENTIFICATION OF THE VARIANCE VALUES NEEDED TO CALIBRATE THE MODEL SMOOTHING FACTORS 1 AND 2

| Arrival Traffic Stream | 7 | 8 | 2 | 3 | 4 | 6 |
|---|----|----|----|----|----|----|
| Corridor Standard Deviation (s) | 10 | 16 | 18 | 21 | 30 | 29 |
| Intersection Entrance Standard Deviation (s) | 7 | 3 | 2 | 6 | 1 | 3 |

Similarly to the Delft case study, the interpretability behind Robertson's model and its calibration provides some insights on the behavior of traffic that cannot be extracted from the black-box structure of the neural network. For instance, traffic streams seven and three (which are right and left turns, respectively, as shown from Figure 42), have higher dispersion rates than other traffic streams, at the entrance to the intersection. This falls in line with the

observations from the Delft case study. Moreover, the results show that the most, overall, highly dispersed arrival directions are directions 3, 4 and 6. The corridor variance for directions 4 and 6 are much higher than for the rest of the directions. However this is a valid estimate as, from the VisSim simulation of this intersection (which will be elaborated on further in the next chapter), it appeared that flows entering the directions of 4 or 6 often stand in long queues that propagate to much farther back than the far away loops. Accordingly, it makes sense for the model to require an extremely high variance in travel times to be able to account for this disturbance on the corridor. With flow direction 6, there is additional dispersion that occurs at the intersection entrance, which makes this flow direction, based on the classical model speed calibrations, the least predictable one. As concluded in Chapter 4, the more travel time variance there is, the more dispersed the platoons are, which cause traffic flow progression patterns (for the same traffic stream) to be quite different between low and high volume flows, resulting in low overall predictability. The results and evaluation will clarify if the correlation between variance in travel times and predictability of traffic continues to be the same or not.

5.5. RESULTS OF BOTH MODELS AND COMPARATIVE EVALUATION

Once the two models have been built, and the neural network had been trained for each intersection combination, the results were printed and evaluated, according to the KPI's defined at the beginning of this report, but also with regards to specific scenarios. While the calibrations applied to the parameters of Robertson's model neared its performance to that of the neural network, the overall results were still in favor of the neural network, as will be discussed in this section. It is important to note however that, due to detector logging failures, much of the 5 months of data provided was not used. Only one month of data was usable in the end. More information on the quality of the data will be provided in Chapter 5.5.4.

5.5.1. AGGREGATED KPI RESULTS AND EVALUATION

Starting with the KPI's, the prediction results of the test set (i.e. the last 20% of the data set used) were evaluated, to determine the level of accuracy of the results. As previously mentioned, the results from the calibrated classical model were evaluated as well, in order for them to be used as reference points. The results can be seen in Table 10.

TABLE 10: KPI RESULTS FOR THE NEURAL NETWORK AND ROBERTSON'S MODEL

| Flow | NN_RMSE | NN_R-stat | NN_NRMSE | Rob_RMSE | Rob_R-stat | Rob_NRMSE |
|------------------------------|---------|-----------|----------|----------|------------|-----------|
| <i>q_{d7}</i> | 0.88 | 0.77 | 1.18 | 0.92 | 0.68 | 1.48 |
| <i>q_{d8}</i> | 1.13 | 0.69 | 0.98 | 1.11 | 0.45 | 1.10 |
| <i>q_{d4}</i> | 0.67 | 0.71 | 0.75 | 0.75 | 0.55 | 1.51 |
| <i>q_{d6}</i> | 0.81 | 0.68 | 1.18 | 0.90 | 0.58 | 1.40 |
| <i>q_{d2}</i> | 1.15 | 0.80 | 0.90 | 1.37 | 0.66 | 1.28 |
| <i>q_{d3}</i> | 0.70 | 0.65 | 1.34 | 0.74 | 0.54 | 1.59 |

By comparing the RMSE results of Table 10 to the results presented in Chapter 4, for the Delft case study, it seems there is a deterioration in the predictive abilities of the models for the more complex case study, despite the expansions made. However, as previously mentioned, RMSE results must be considered relatively to the dataset range. This case study has detectors that cover two lanes for five out of the six predicted traffic streams (traffic stream 4 is of a single lane). Accordingly, larger flow volumes are detected every 10s time interval, which makes the results not directly comparable with the previous case study. In fact, the NRMSE values are all less than "2", which was seen for the left turn lane in the previous case study, indicating that, relative to the datasets, the RMSE's are not too high. Moreover, with regards to the NARX neural networks KPI results in specific, the R-statistics are all higher than 0.6, for the left and right turning lanes, which entails a superiority in the performance of the redesigned prediction model, despite the added data complexity.

One of the most significant takeaways from Table 10, is that, while the performance of the NARX neural network is still not grossly superior to the classical model, a noticeably wider gap can be seen in their KPIs than for the Delft case study. This relative deterioration in the classical model's performance, in comparison to the neural network,

is consistent with the concerns expressed in literature, which are outlined in Chapter 2.2.2., on the limitations of the classical model's performance for non-free-flow conditions. The availability of different traffic conditions is the key difference between this Noord Holland, full intersection, case study and the Delft, two lane urban corridor, case study. The inability of the classical model to dynamically adapt its dispersion parameters, and turning percentages, over time, to account for changing traffic conditions, is a significant obstacle to its performance. While the neural network does not appear to fully overcome this parameter adaptability challenge, its predictions do appear to significantly better match the fluctuating traffic arrival profiles, as seen from the consistently higher R-statistics. In fact, unlike the results of Chapter 4, the NARX neural network's performance is noticeably superior with regards to all three KPIs. Accordingly, while the previous case study highlighted the strengths of the classical model, justifying the reason why it was used as an inspiration for the skeleton structure of the neural network design, the results of this case study highlight the strengths of using a machine learning approach, that does not rely on static parameters, in a highly dynamic environment.

With regards to the results of the individual traffic streams, for the neural network, some information on the traffic conditions that affect the prediction abilities of the model can be derived. For instance, the straight forward directions 2 and 8 have two of the lowest NRMSE, which is logical considering straight ahead directions usually have the most density, resulting in less dispersion and higher mean flow volumes for their profiles. However, while the R-stat of direction 2 is additionally quite high, insinuating that the model predictions match the arrival flow profile quite well (i.e. the predictions are robust to different arrival volumes), the same cannot be said about Direction 8. Despite having a very similar RMSE to Direction 2, the predicted flow profile of Direction 8 does not seem to have a comparably strong match with the actual profile, as seen by its much lower R-stat. The NRMSE of direction 8 is also higher than direction 2, despite the nearly identical RMSE, which means the mean volume of arrival is lower, indicating that the traffic stream heading to direction 8 is less dense, and therefore less predictable. However, this still does not justify the very large difference in the R-statistic. It is likely then that the V-log data for this traffic stream is more erroneous, as will be discussed in Chapter 5.5.4.

The R-statistic and NRMSE values for directions 3 and 6 indicate that the prediction model is having more difficulty matching the flow profiles of these two traffic streams as well. This outcome was already anticipated after their optimal travel time variances were estimated for Robertson's model, and showed that the longitudinal dispersion on these two traffic streams is quite high, although for different reasons. Direction 3 is a low demand left turn lane, while the traffic stream in direction 6 experiences very long queues that propagate far beyond the far away loop. When the far away loop is covered by a long queue, it provides detector readings of zero for arrival flows, despite the fact that there are vehicles arriving and queuing up at the intersection. This is problematic for traffic flow progression forecasting. However the same problem exists for direction 4, with the difference being that this traffic stream is supported by a single lane; and so, the far away loop is lane specific. What can be seen then, is that the added reliability of a lane specific detector allows for the potential of the reengineered neural network architecture to more clearly appear. With reliable, lane specific, detector data, alongside its ability to implicitly incorporate traffic signal phases and the existence of queues into its arrival flow predictions the model appears to be finding a logic behind the discrepancies of having several departure flows entering a corridor and zero measured arrival flows at the destination. Accordingly, a significant improvement can be seen for direction 4 between the classical model and the neural network prediction performance. Significant improvements can be seen for direction 6 as well, from the classical model, but it appears the double lane loop detector is affecting the overall predictability of the arrival flows on this direction, hindering the model's predictions from performing as well as for direction 4. The large difference between these two models' performances is not only an indication that the neural network can better predict vehicle arrivals when there is a long queue, but also that it can better accommodate for sources and sinks (i.e. the parking lot on that route). This is also evident since directions 2,3 and 7 have long queues that cover the far away loop at times as well (although not as often), but the performance superiority of the neural network for these three traffic streams is not as significant.

5.5.2. RESULTS AND EVALUATION PER TRAFFIC STATES

With this case study, there are several traffic conditions that exist within the data. Hence, to more thoroughly evaluate the performance of the neural network, the test data set was clustered into eight traffic cases that may occur, on each of the six arrival points. This way, the performance of the NARX neural network can be compared to that of the classical model, per classified condition. Through this clustering, a better understanding of the range of

fluctuations in the performance of both models, and the extent of superiority between each model, for the different traffic conditions, can be seen. This traffic state specific evaluation serves as a robustness indicator.

5.5.2.1. PROPOSED TRAFFIC STATE CLASSIFICATION METHODOLOGY

With only the availability of historical v-log detector pulses, it is difficult to identify the traffic states occurring within a specific dataset. Hence, a an approach for classifying traffic into different conditions, through detector occupancies, is needed. Inspired by Liu et. al. (2009)'s work on how to determine maximum and minimum queue lengths at intersections using detector occupancy information, the following traffic classification methodology was developed and used:

Aggregate Data to 1s Time Steps: At higher aggregation levels than one second, it is not possible to determine traffic states through the rate of measured detector pulses. The information derived from this methodology on the traffic conditions is later fitted to the 10s aggregated time steps to determine what the conditions were as soon as the first vehicle within the target time step arrived (i.e. the time-stamp highlighted by the red line in Figure 48).

Identify Traffic Phase Cycle State: At the one second time stamp that constitutes the start of the target 10s time step, and the end of its preceding 10s time step, the color of the traffic light is classified as either Green/Amber or Red, as shown in Figure 48. Amber and Green are placed in the same cluster as this cluster includes all vehicles that are assumed to move over the stop line detector and cross the intersection. If the color is Green, the conditions of the preceding time step are looked into. If the traffic light was red right before, then the target time step is moved from the Green/Amber cluster to the Red cluster, as the queued vehicles are only beginning to exit the intersection at this point (i.e. the red line in Figure 48), so could be considered as still queued at a red when the next platoon begins arriving.

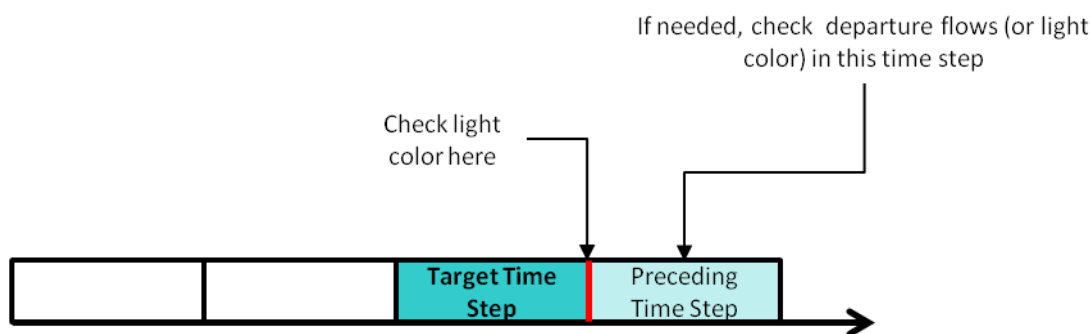


FIGURE 48: VISUALIZATION OF TIMESTAMPS AT WHICH INFORMATION IS COLLECTED FOR TRAFFIC STATE CLASSIFICATION

Identify Lane Saturation State: Now that the state of the traffic light phase cycle has been identified, the data is further clustered by the saturation state of the mouth of the intersection n(i.e. the portion of the urban corridor after the lane expansion). The following were the different clusters defined:

Green with Congested-flow: The green wave could be emptying a saturated lane. This saturated state can be identified by the rate of vehicle departures, as determined by the counts of the stop-line detector. If vehicles exit an intersection with a time headway that is less than 2.5 seconds, then the lane is saturated. However, if, right before the end of the preceding time step (i.e. before the red one second time stamp in Figure 48), the long loop detector has become unoccupied for longer than 1s, then the saturation is considered to have been cleared out before the target platoon arrives to the far away loop (around 120m from the stop line detector). The target time step is therefore placed into the green with free-flow cluster. A visualization of the green with congested-flow state can be found in Figure 49.

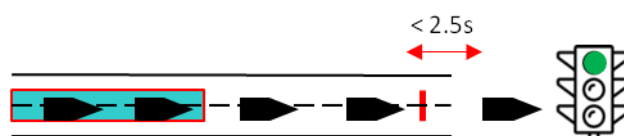


FIGURE 49: GREEN WITH CONGESTED FLOW STATE

Green with Free-flow: The green wave could be facilitating for vehicles to just pass through a green wave in free-flow. In that case, the time headway within the platoon of the preceding time step will be greater than

2.5 seconds, as shown in Figure 50. Either that or the flow is considered to have just cleared, as explained in the previous classification.

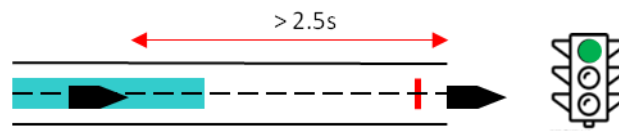


FIGURE 50: GREEN WITH FREE FLOW STATE

Red with Empty Lane: The lane could be empty during a red light, in which case, at the red 1s time stamp in Figure 48, both the stop line and long loop detectors will be unoccupied, as shown in Figure 51.

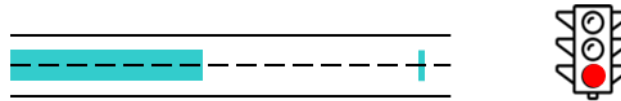


FIGURE 51: RED WITH EMPTY LANE STATE

Red with Short Queued Lane: The lane could have a “short queue”, which is defined as a queue where only the stop line detector is occupied, but the line of vehicles has not reached back enough to occupy the long loop detector, as shown in Figure 52. This detector occupancy state is determined at the red 1s time stamp in Figure 48.

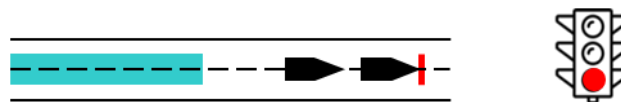


FIGURE 52: RED WITH SHORT QUEUED LANE STATE

Red with Long Queued Lane: The lane could have a “long queue”, which is defined as a queue which starts at the stop line detector and reaches back to the long loop detector (and so it is occupied as well), as shown in Figure 53. Any queue length starting from the point at which the long loop detector is occupied is considered as a long queue. This detector occupancy state is determined at the red 1s time stamp in Figure 48.



FIGURE 53: RED WITH LONG QUEUED LANE STATE

Identify Possible Lane Pair Combinations: The far away loops, at which the arrival flow predictions are made, cover all the lanes supporting the same traffic stream (i.e. same direction of flow). For five out of the six arrival traffic streams in this case study have two lanes covered by a single, joint, far away loop. Hence, possible traffic state combinations for each of the five lane pairs in this case study were identified as follows:

- Green/Amber Light:
 - Congested Flow & Congested Flow
 - Free Flow & Free Flow
 - Congested Flow & Free flow
- Red Light:
 - No Queue & No Queue
 - No Queue & Short Queue
 - Short Queue & Short Queue
 - Short Queue & Long Queue
 - Long Queue & Long Queue.

For direction 4, since it is a single lane, no combinations were needed. The possible clusters were just any of the defined saturation states above.

The KPI used to compare the performance of the two progression models was the RMSE since it is the physical estimated average of error. Accordingly, after each of the 10s time steps in the test dataset has been classified into one of the eight mentioned clusters, the RMSE for each individual cluster, for both models, is calculated, as shown in Figure 54. In the displayed bar Graphs, the green bars represent a Green/Amber phase cycle state, while the Red bars represent Red phase cycle states; and, the darker bars represent the RMSE of the NARX neural network, while the lighter bars represent the RMSE of Robertson’s classical model, per defined traffic condition.

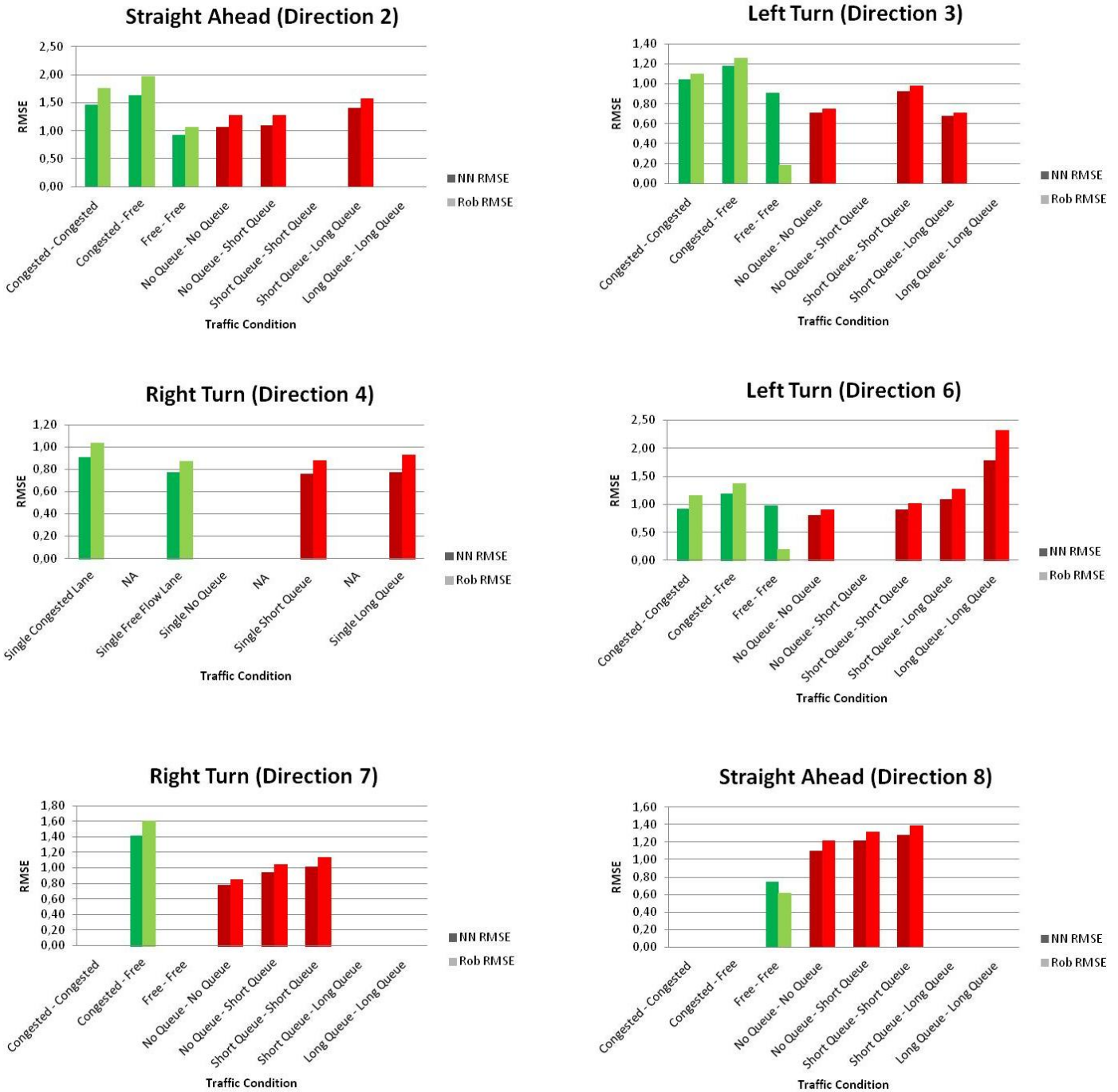


FIGURE 54: RMSE’S FOR THE TWO MODELS, PER DEFINED TRAFFIC CONDITION

The reason why the bar graphs in Figure 54 are considered to provide an indication of robustness is, as per the definition of robustness stated in Chapter 1.4.3., the more uniform the RMSE’s are per direction, the more reliability there is that the model will perform the same under any condition. For the majority of the traffic states for the different directions, it appears both the neural network and the classical model follow the same trend. However, by specifically looking at directions 3, 6 and 8, the robustness differences between the two models at the difficult to predict traffic flow directions can be seen. While the lowest overall RMSE of the classical model is achieved when its

dispersion parameters best match the free flow conditions, as shown in Figure 54 for these three directions, its errors are much higher for other traffic states. On the other hand, the neural network reaches an overall lower RMSE than the classical model, with a lower variance between its performance for all traffic states. This lower variance is an indication of its higher robustness. Accordingly, while the neural network's performance is weaker for free flow conditions in specific (in a multi-traffic state environment), it makes up for it with its more consistent performance for any and all traffic states.

Moreover, for the more saturated traffic conditions (i.e. congested lanes or lanes containing queues), much of the larger improvements of the neural network, over the classical model can be seen. This is an important advantage in the progression model's performance as it means it is able to improve upon one of the main weaknesses of the classical model, which is in its limited ability to predict traffic flow arrivals when the free flow progression of a platoon has been interrupted. The biggest advantage revealed in Figure 54 though is in the significant prediction improvements achieved by the NARX neural network on directions 4 and 6. The overall amount of improvement on these two traffic streams is visibly better than for the rest., which is a positive indicator considering the undetected parking lot that serves as a source/sink along their road link. Moreover, the largest improvements on these two directions is for the long queue and congested states. These traffic conditions include the instances where the far away loop detector is covered, due to the saturated state of the lanes, which has propagated far back beyond these loops. When the far away loop is covered, the detector reads there are no arrival flows, causing a discrepancy with the classical progression model's predictions, considering that the model is bound by the assumption that any measured flow input must progress to the destination point, with a certain pattern, within the specified time horizon. Hence, the ability of the designed neural network to better adapt to these conditions, makes it much more robust to a condition where the classical model is sure to fail.

5.5.3. EVALUATION OF RESULTS PER DEFINED ERROR OCCURANCES

The identified KPIs, even when clustered by traffic conditions, still do not provide the full picture of the differences in performance between the neural network and the classical model. Accordingly, two different lanes, representing two different directions of movements, were taken as a sample for further performance analysis. The two traffic streams selected were those of direction 2 and direction 3. The evaluations made in this chapter provide a better indication of the meaning behind the statistical parameter values of the KPIs discussed earlier. Once again, using the test set data, some observations were made on the performance of the neural network and the classical model for certain occurrences that were considered significant. These observations can be listed as follows:

Percentages of Captured Vehicles: Overall, it was found that the neural network managed to capture 16.4% more vehicles in the test set of direction 2 than the classical model, and 8.9% more vehicles in the test set of direction 3, which are notable improvements.

Hit and Miss Accuracy: The percentages of instances where the models have 100% accuracy in predicting the volume of traffic within the defined 10s time step are calculated, and displayed in Table 11. Since there are several magnitudes of arrivals that can occur every 10s, these percentages are not an indication of overall percentage accuracy, which makes it difference than the previous observation made on captured vehicles. For example, if either model predicts an arrival flow of 5veh/10s when the actual is 6veh/10s, for this KPI, it would be considered a miss (since it was not a perfect prediction), but this does not mean the model's prediction was completely wrong. However, these estimates still reveal the improvements achieved by the neural network with regards to robustness.

TABLE 11: HIT AND MISS ACCURACY OF BOTH PROGRESSION MODELS

| | Rob. Classical Model | NARX NN Model |
|-------------|----------------------|---------------|
| Direction 2 | 40.7% | 45% |
| Direction 3 | 64.1% | 66.6% |

The improvement in the perfect match seen by the neural network is 10.6% for direction 2 and 3.8% for direction 3. It is also interesting to see that, while the KPIs for the left turn lane indicate that this traffic stream is the most difficult to predict, these results show that the models are able to capture a higher percentage of the instances perfectly. Most of these perfectly matched results though are instances where there is zero flow arrivals, and the KPIs give an

evaluation for the full range of arrivals. Additionally, although the ability to predict zero arrivals is important, it is still not as challenging for a progression model to do so as it is to “perfectly” predict the exact magnitude of an arrival volume per 10s interval.

Accuracy with One Vehicle Error: Referencing the argument made in Chapter 1.3.2., the reason why traffic flow predictions was the approach taken, rather than phase cycle length and switch times predictions, was due to the higher flexibility in error acceptance, while still maintaining traffic flow control efficiency. There is no defined benchmark for what an acceptable error is for what an acceptable traffic flow prediction error magnitude is, and so it was assumed as one vehicle for this study. The percentage of instances in the test set where both models made an prediction error of one vehicle or less can be found in Table 12.

TABLE 12: PERCENTAGE OF INSTANCES WHERE EACH MODEL MADE A PREDICTION ERROR OF ONE VEHICLE OR LESS

| | Rob. Classical Model | NARX NN Model |
|-------------|----------------------|---------------|
| Direction 2 | 79% | 84% |
| Direction 3 | 94.2% | 95% |

The improvement in instances where an error of one percent or less is made by the neural network is 7% for direction 2 and 0.9% for direction 3. This correspondently entails a 26.4% reduction in errors greater than 1 for the neural network’s predictions at direction 2, and a 14.8% reduction of these errors for its predictions at direction 3. These results seem quite promising, even for the classical model. However, while one vehicle error in a platoon of multiple vehicles arriving can be considered negligible, if it is only one vehicle that arrived and was missed, or a ghost car was predicted when there was none, this error becomes far more critical. Hence further evaluation of occurrences of these two instances was needed.

Percentage of Ghost Cars: Prediction errors involve both over and under predictions. Accordingly, there are instances where both models predict vehicle arrivals when there are none. For a controller application this is critical as, using this erroneous prediction, the controller can assign a green phase to an empty traffic stream, taking away from the green time of another that needs it. The percentage of instances where ghost cars were predicted by both models can be found in Table 13.

TABLE 13: PERCENTAGE OF PREDICTED GHOST CARS BY EACH MODEL

| | Rob. Classical Model | NARX NN Model |
|-------------|----------------------|---------------|
| Direction 2 | 25.6% | 19.6% |
| Direction 3 | 36.1% | 32.4% |

The neural network appears to reduce the prediction of ghost cars by 25.5% for direction 2 and 10.3% for direction 3. However, the remaining percentage of instances where ghost cars are detected is still quite concerning.

Percentage of Unseen Cars: This is the percentage of vehicles that arrive at the intersection, but were not predicted. These entails that a controller operating on predicted arrivals would never recognize that these cars are there, and hence they are “unseen”. The percentage of occurrences of this critical error can be seen in Table 14.

TABLE 14: PERCENTAGE OF UNPREDICTED CARS BY BOTH MODELS

| | Rob. Classical Model | NARX NN Model |
|-------------|----------------------|---------------|
| Direction 2 | 7.1% | 6.9% |
| Direction 3 | 25.2% | 21.5% |

The neural network reduces the number of unseen cars by 2.8% for direction 2 and 14.7% for direction 3. The probability of this occurrence is quite high for direction 3, similarly to the ghost cars. The difference in performance between direction 2 and 3 for the last two classified occurrences gives a better indication of why direction 3 is considered to be significantly more unpredictable than direction 2.

5.5.4. EVALUATION OF RESULTS PER MAGNITUDES OF ERRORS

While the RMSE provides an indication of the magnitude of error made, on average, every 10s time step, it doesn't mean that a wide range of error magnitudes does not occur as well. Accordingly, an evaluation of the errors magnitudes of errors made for the test set and the percentage of instances with which they occur was made for both models. The intention of this evaluation was to provide some insight on how big of an error can be made by either prediction model, and how often do errors of this size occur. Once again, directions 2 and 3 were taken as the samples for this evaluation. The results can be seen for directions 2 and 3 in Table 15 and Table 16 respectively. Within this table, a positive magnitude value means that the error is an under estimation, while a negative value entails an overestimation by that amount.

TABLE 15: MAGNITUDES OF ERRORS ON DIRECTION 2 AND THEIR PERCENTAGE OCCURANCES FOR BOTH MODELS

| Magnitude of Error | Rob Occurrence (%) | NN Occurrence (%) | NN Decrease in Error (%) |
|--------------------|--------------------|-------------------|--------------------------|
| 1 | 15,426 | 16,762 | -9 |
| 2 | 5,693 | 5,652 | 1 |
| 3 | 2,462 | 2,115 | 14 |
| 4 | 1,052 | 0,764 | 27 |
| 5 | 0,411 | 0,255 | 38 |
| 6 | 0,143 | 0,082 | 42 |
| 7 | 0,053 | 0,028 | 47 |
| 8 | 0,015 | 0,005 | 66 |
| 9 | 0,002 | 0,001 | 50 |
| -1 | 22,879 | 22,757 | 1 |
| -2 | 7,640 | 5,183 | 32 |
| -3 | 2,409 | 1,079 | 55 |
| -4 | 0,785 | 0,219 | 72 |
| -5 | 0,228 | 0,036 | 84 |
| -6 | 0,057 | 0,010 | 83 |
| -7 | 0,023 | 0,002 | 91 |
| -8 | 0,006 | 0,001 | 83 |
| -9 | 0,002 | 0,000 | 100 |

TABLE 16: MAGNITUDES OF ERRORS ON DIRECTION 3 AND THEIR PERCENTAGE OCCURANCES FOR BOTH MODELS

| Magnitude of Error | Rob Occurrence (%) | NN Occurrence (%) | NN Decrease in Error (%) |
|--------------------|--------------------|-------------------|--------------------------|
| 1 | 13,671 | 13,396 | 2 |
| 2 | 3,671 | 3,052 | 17 |
| 3 | 1,005 | 0,670 | 33 |
| 4 | 0,217 | 0,148 | 32 |
| 5 | 0,044 | 0,028 | 37 |
| 6 | 0,008 | 0,003 | 67 |
| 7 | 0,002 | 0,001 | 50 |
| 8 | 0,000 | 0,000 | 0 |
| 9 | 0,000 | 0,000 | 0 |
| -1 | 16,356 | 15,036 | 8 |
| -2 | 0,880 | 1,039 | -18 |
| -3 | 0,015 | 0,037 | -145 |
| -4 | 0,001 | 0,003 | -150 |
| -5 | 0,002 | 0,000 | 100 |
| -6 | 0,001 | 0,000 | 100 |
| -7 | 0,001 | 0,000 | 100 |
| -8 | 0,001 | 0,000 | 100 |
| -9 | 0,000 | 0,000 | 0 |

The advantage the NARX neural network has over Robertson’s classical model, with regards to this error evaluation, is that the neural network appears to have optimized its parameters so that it minimizes making high magnitude errors. The percentages of instances where the classical model made large errors were already quite low, but with the neural network they became negligible. The results shown for this error evaluation are perhaps the biggest indicator of the higher robustness of the NARX neural network’s predictions. The high magnitude errors, are likely to cause queue spillback during peak hours, which would entail a failure in intersection traffic management. Hence, it is quite a significant advantage for the neural network to significantly reduce these errors as such. However, in return, it appears to be making more low magnitude errors.

With regards to Table 15, the neural network is making a one vehicle underestimation error more than Robertson’s model by 9%. There would have been some doubt that this could be a troubling value, considering this one vehicle error could lead to unseen arrivals, had it not been for the results in Table 14, which show that the neural network is actually making less “unseen vehicle” errors than the classical model. Accordingly, this additional 9% is concluded to have gone into the acceptable error cluster of Table 12. On the other hand, the much larger overestimation errors made by the neural network in Table 16 are not as acceptable. Similarly to what has been seen in Chapter 4, the neural network appears to be attempting to create predicted arrival flow profile estimates that can better match the higher volume platoon arrivals that seem to follow a different dispersion pattern than the majority of the data, which is a problematic feature of the arrival profiles of low density traffic streams. In attempts to do so, it though the neural network is, consequently, overestimating vehicle arrivals at free flow conditions. It is known that the overestimates are at free flow, since an analysis was made on the performance difference between Robertson’s model and the neural network for different volumes of arrivals, and the only volume at which Robertson’s model showed superiority was when 2 vehicles arrive within a 10s interval, which is a low density arrival state. These results correspond with the results in Figure 54, which show that Robertson’s model performs much better than the neural network for free flow states, while the neural network performs better overall.

5.5.5. QUALITY EVALUATION OF USED DATA

Out of the two years worth of v-log data provided by the Province of Noord Holland, only one month of data was used for this case study. Nearly all controllers for the four intersections in this small network had been updated around the end of 2016 or the start of 2017, making the available v-long configuration files only useful for logged data after 2017. Older configuration files were not available. Accordingly, it was not possible to compile the v-log data for 2016 and 2015 and assign it to the correct detector labels. Not only so, but for some detectors, the collected data displayed significant detector failures (or detector logging failures) , rendering the majority of the data unusable. Examples of this can be seen in Figure 55. Hence, the quantity of available data did not suffice for big data analysis. Without sufficiently large datasets, the neural network cannot train to its full potential. However, while detector errors hinder the training of a neural network, a strength of this machine learning algorithm is that once it has fully trained, it would be able to identify discrepancies in input data patterns as errors, and resultantly ignore them, making it a robust algorithm.

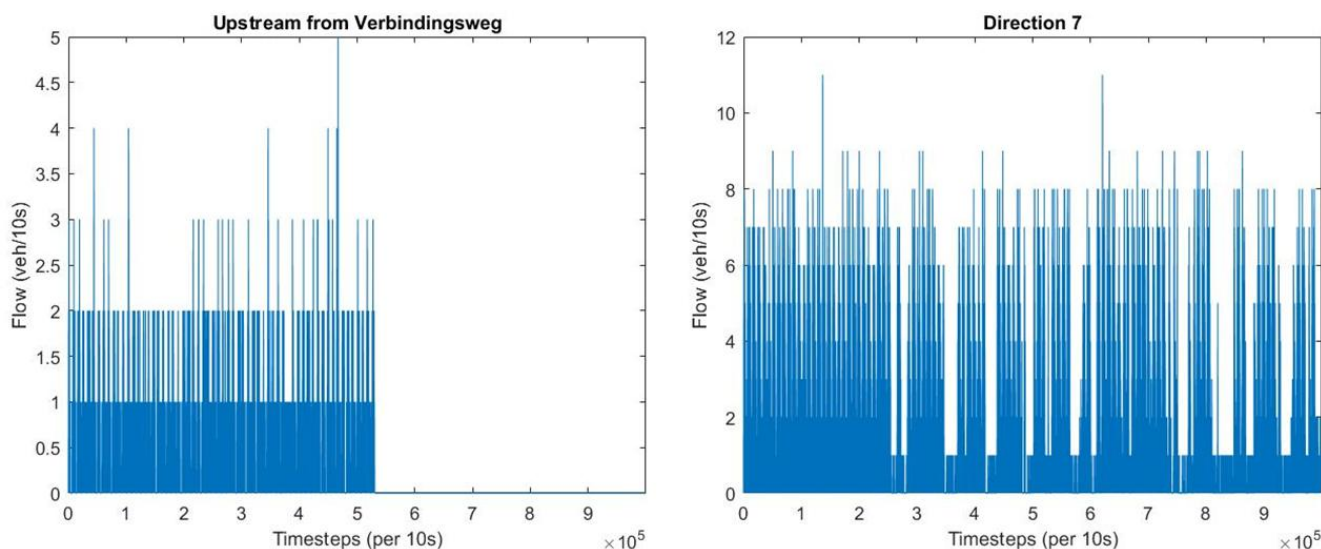


FIGURE 55: EXAMPLES OF LOGGING DATA FOUND FOR SOME DETECTORS

With regards to training though, even with the finally used data, there was noise in the detector readings of the far away loop detectors, used to measure the arrival flows. When a placed detector covers two lanes rather than one, it becomes less reliable (Liu et. al., 2009). Detectors can only give “occupied” or “unoccupied” signals, and so if two cars pass over a detector at the same time (i.e. one on each lane), the detector will only measure a single count. Hence, inconsistencies exist in the arrival flow profiles of the case study intersection, since the far away loops are missing vehicles (i.e. some vehicles are completely undetected). Figure 56 provides a good visualization of the problem, by showing how, in direction 4, where the far away loop covers a single lane, the number of arriving vehicles measured at the far away loop is the same as the number of vehicles detected later departing the intersection (during a green light) at the stop-line detector. On the other hand, for the double lane far away loop detector, the gap between the number of detected vehicles arriving and the detected vehicles later departing from the stop-lines of both lanes keeps growing as more vehicles are being overlooked at the arrival point. This noise affects the training of the neural network, and provides justification on why direction 4 has one of the highest R-statistics and lowest RMSE and NRMSE for the neural network, despite being a congested traffic stream that often has long queues far past the far away loop detector, meaning it is covered for a large percentage of the time during peak hours. Hence, with more reliable data, the neural network’s performance for the predictions on the other directions of flow could have significantly improved.

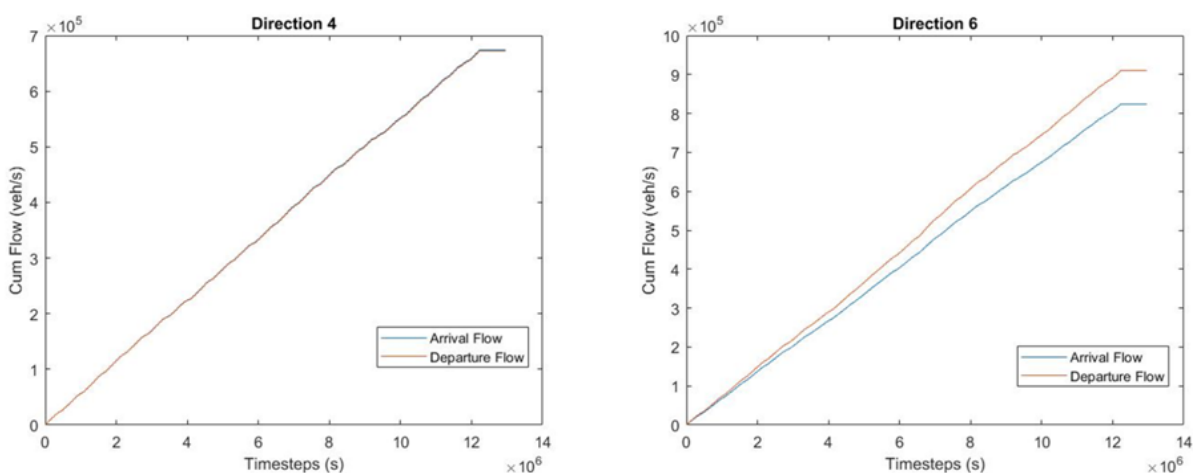


FIGURE 56: A VISUALIZATION OF THE MISSED VEHICLES ERRORS THAT RESULT FROM DOUBLE LANE FAR AWAY LOOP DETECTORS (LEFT), WHICH DO NOT OCCUR FOR SINGLE LANE FAR AWAY LOOPS (RIGHT)

5.6. CONCLUSIONS

For all the different analyses conducted in this chapter, the neural network has displayed overall performance superiority over the classical model. The model has lower average error counts, and its predictions better fit the arrival flow profiles. More specifically, the model has been found to significantly improve upon the classical model’s performance in congested conditions, especially when the far away loop is covered by a long queue, and in the presence of a source or sink. It also has a lower variance between the average errors it makes for the different traffic states, making it more reliable. Moreover, when looking at specific clusters of errors (i.e. 0 vehicle errors, 1 vehicle error, ghost cars and missed cars), the neural network outperformed the classical model on all accounts, although in some cases the performance improvement was more prominent than others. Most importantly though, the neural network’s predictions had far less instances where large errors were made, which was a clear indication of its robustness.

However, while the results of this chapter are quite numerically interpretable, alone, they do not give any indication of how a controller will be affected by the prediction errors of either model. For instance, with regards to missed cars, it could be the case that a vehicle arrives, unpredicted, at a certain 10s time step, but is followed, in the next 10s time step by a platoon of vehicles that is predicted. That missed vehicle would then get a green light with that platoon, making the severity of the risk lower. The same can be applicable for any error, since either model could be making shifted predictions, which is less negatively impactful on a controller than completely wrong predictions. Moreover, the general severity of a prediction error, for a controller, and the true magnitude of the improvements of a neural network’s performance over a classical model can only be speculated using the results of this chapter. Accordingly, to give meaning to these numbers, they need to be tested on a controller.

6. EVALUATING THE NEURAL NETWORK TRAFFIC PREGRESSION MODEL FOR PREDICTIVE CONTROL APPLICATIONS

In this chapter, the designed and evaluated NARX neural network traffic flow progression model is tested for its intended purpose. This not only allows the concept design of a predictive controller, proposed in Chapter 1.3, to be evaluated, but it additionally gives physical meaning to the KPIs calculated in Chapter 5. Hence, this chapter begins with an introduction and an explanation of the simulation environment used to test the predictive control concept. Then in chapter 6.3., the approach taken for how to model and evaluate predictive controller, in comparison to an adaptive one, is explained, followed by the experimental setup and KPIs in Chapter 6.4. Next, the results, and their evaluation, are provided in Chapter 6.5., which also includes an evaluation of how the provided prediction horizon better facilitates the use of T2G/R and GLOSA; and, finally the chapter ends with a some conclusions.

6.1. INTRODUCTION

Once the prediction model is built and its performance has been evaluated, the final step is to interface its outputs with a simulated controller, in order to determine the extent to which the errors of the prediction model negatively impact traffic management. Hence, a significant KPI for this chapter is additional measured delay time at an intersection, when predictive control is used without a fall-back plan for correcting prediction errors. This is done for both the neural network and Robertson's model to determine how severe the difference between their performance is when used for a predictive control application. The second part of this chapter will discuss the different internal phases of a controller to evaluate if the set prediction horizon will be enough for T2G/R and GLOSA information.

6.2. CONTEXT

The Province of Noord Holland has a very sophisticated VisSim simulation model of the small network used for this case study. This model includes all the necessary vehicle compositions, speeds and network configuration set up; but most importantly, the same detector configurations and traffic control system is modeled in this simulation as well, as shown in Figure 57. External controllers, compiled as CCOL7 executable files, were interfaced into the simulation environment, allowing the VisSim model's controllers to display the exact behavior as the actual adaptive controllers of this case study on the street. Hence, making experiments with this simulation model are quite informative on how an adaptive controller will behave when given predicted inputs, and how traffic could be affected.

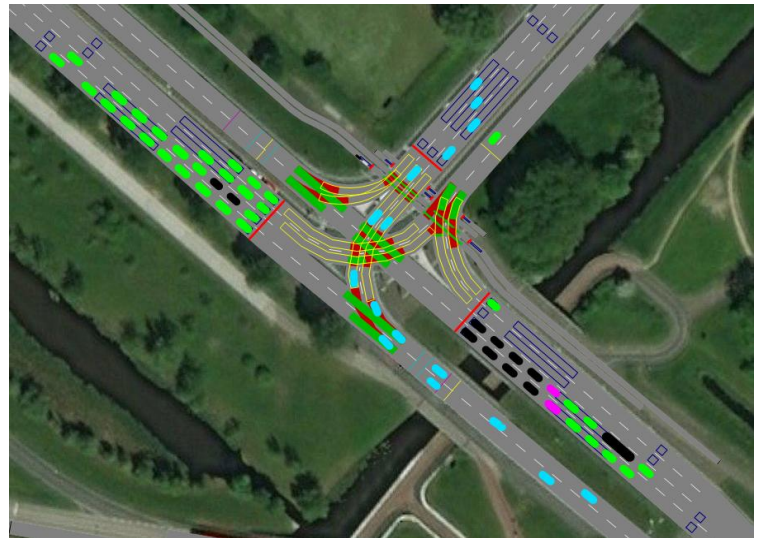


FIGURE 57: SCREEN SHOT OF VISSIM SIMULATION MODEL

6.3. APPROACH TAKEN FOR MODELING AND EVALUATING A PREDICTIVE CONTROLLER

When implementing the proposed predictive controller design, arrival flows will be forecasted at the far away detector loops of the predictive controller's intersection, based on upstream detector measurements, from the previous intersections. The movement of vehicle flows will then be simulated over the detectors, starting from the far away loop, to the stop line loop, triggering the controller, in virtual space, to provide an optimized output, as it normally does on the street. To simulate this in VisSim then, certain changes were made to the provided simulation model. Firstly, the static vehicle routing decisions were changed, since vehicles no longer need to be modeled from one intersection to the next, but from just upstream the far away loops to after the intersection crossing, to model the just described concept. Secondly, the vehicle flow averages defined for certain aggregated periods of the day were removed, and instead flows were defined in the model per each 10s time step. This allows the VisSim simulation and the prediction model to operate on the same aggregation level, which allows for them to be interfaced. The

simulation environment was therefore no longer of the network, but a highly focused, and detailed, simulation of the case study intersection.

Next, the following methodology was followed for simulating and evaluating the predictive control concept of Chapter 1.3.:

Step 1: In the 10s time intervals of the VisSim model, the corresponding actual volumes/10s (i.e. actual arrival flows) were recorded. Data collection nodes were placed at the stop line detectors of each lane for the modeled case study intersection. The model was then run and the vehicle loss times at the intersection were recorded as an output of the simulation run.

Step 2: The volumes/10s were now re-recorded, for the same corresponding time steps as before, but as per the NARX neural network's predictions. The model was then run and the controller phase cycle states (i.e. green, amber and red) start times and durations were recorded as an output of the simulation run. The exact same was done for the expanded and calibrated version of Robertson's classical prediction model.

Step 3: The actual arrival flows per 10s were placed back into the model; however, this time, the external actuated controller was removed from the intersection, and, in its place, a fixed controller was placed. This controller's schedule is fixed to the outputted control schedule of the NARX neural network, from Step 2. Predictive controllers are ones whose schedules are fixed offline, but in real time. Hence, with this formulation, an offline representation of how it would work (using historical data rather than live data) is simulated. Using the same data collection nodes as Step 1, the model was run and the vehicle loss times at the intersection were recorded as an output of the simulation run. The exact same was done for the expanded and calibrated version of Robertson's classical prediction model.

Step 4: The total vehicle loss hours per simulated day were recorded for comparison between the adaptive controller, the NARX neural network predictive controller and the Robertson's classical model predictive controller.

These four steps were conducted for a random selection of each of the two most distinct different types of days found in the data. The regular demand days, and off-peak days. These off peak days were found to be Wednesday and Thursday, which most commuters normally take off. The reason was to see how the performance accuracy differs between high and low demand days, considering the results of Chapters 4 and 5 showed that low demand leads to more dispersion and consequently higher variances in the arrival flow profiles, which lead to lower prediction accuracies. Alternatively though, according to Kerner's Three-Phase Traffic Flow Theory, at high densities, there is a much higher probability of a slight disturbance breaking down traffic to a wide moving jam. Hence, an evaluation of both types of days was needed.

6.4. EXPERIMENTAL SETUP AND KPIS

Following the approach explained in Chapter 6.3., the total vehicle loss hours for each traffic stream (i.e. direction of flow) were calculated for the two simulated days, and a comparative evaluation was made between the loss hours of the base, traffic actuated control, state, and the predictive control state (once using NARX neural network predictions and another time using the classical model's predictions). The data for these two days was extracted from the test set, which has been used for all results evaluations in Chapter 5. The test set actually only consisted of one week of data, and so Tuesday of that week was selected as the high demand day, and Wednesday was selected as the low demand day. The simulation was run three times and averages were taken for results.

Using these results, the effect of the prediction errors made by these two prediction models on a predictive controller can be seen. This better clarifies the extent to which the neural network improves upon the classical model, providing more interpretable estimates on the difference between their prediction accuracies, and give some information on the extent to which corrective measurements, which are implemented in all state-of-the-art online predictive controllers, are still needed. Accordingly, **vehicle loss hours resulting from delays at the intersection were the main KPI**. As explained in Chapter 6.3., these delays were measured using data collection nodes, which were placed on top of the stop line detectors of each lane of the simulated intersection. Delays are exported from the VisSim model in seconds per detected vehicle, but was converted to total hours lost for all vehicles during the evaluated day.

Additionally, to get a better understanding of amount of travel time savings that have been achieved by using adaptive control, and to have a bench mark from which to evaluate the severity of the prediction errors made by either the neural network or the classical benchmark model, the same simulation was also run for a naïve fixed controller, which operates with a 120s cycle. With this control strategy, each signal group gets 57s of green time and 3s of amber time. With this controller, the maximum waiting time a vehicle can experience (if there is no queue spillback) is two minutes. The signal groups for this controller are displayed in Figure 58. Vehicle loss hours were similarly measured for this scenario as well.

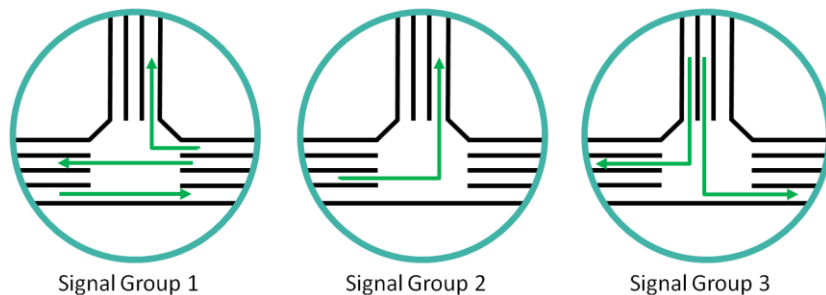


FIGURE 58: SIGNAL GROUPS FOR FIXED 120S CYCLE CONTROLLER

Since one of the observations in Chapter 5 was that a both the NARX neural network and Robertson’s model appear be predicting ghost cars or missing vehicles quite frequently, vehicle loss hours were further divided between morning 12 hours (6am to 6pm) and evening 12 hours. For only the neural network, since it is the model of interest, these values were then, each, divided by the number of vehicles in their corresponding time segment, which provided the average delay per vehicle (in seconds) for each interval. The time segments were isolated in this way to determine whether the majority of delays are occurring in the busy morning hours or during the evening and night when the density is quite low. In this way it is be possible to roughly isolate where the neural network is weakest and requires further improvements. This was followed by the final model evaluation for this chapter, which was on the potential this control methodology has for facilitating GLOSA and T2G/R applications. The **KPI for the prediction horizon evaluation was the effective length of the horizon.**

6.5. RESULTS AND EVALUATION OF PREDICTIVE CONTROLLER

Once the experimental results for vehicle delay hours were calculated, they were recorded and evaluated in order to provide some conclusions on the predictive control concept proposed for this study. Next, some research on the different internal signal states a Dutch traffic actuated controller can enter was made in order to understand what benefits he facilitated prediction horizon can have for the application of use cases such as GLOSA and T2G/R.

6.5.1. TRAVEL TIME DELAY RESULTS FOR BOTH PREDICTION MODELS AND COMPARITIVE EVALUATION

The total vehicle loss hours for the high demand and low demand days selected can be seen in Table 17 and Table 18 respectively. While the results of both days show that the NARX neural network prediction model is still not strong enough for its outputs to be used as an input for a predictive controller, without the use of corrective measures, the model did show significant improvements over the Robertson’s classical model, which had also been expanded and calibrated to match the conditions of this case study. The value of these improvements is that they show that there is potential for having predictive control applications without customizing the infrastructure for it. Especially considering the data was not ideal. Even within the one month of data used for training validation and testing, there was erroneous detector data found, the double lane far away loop detectors provided insufficient data for training, and most critically the location of the far away loops was not at a location where flow arrivals are unaffected by queue lengths. The two data errors equally affected the parameter optimization process of the two models. However, the neural network’s ability to better adapt to the obstacle of having to match predicted arrival measurements to actual ones when the far away loop is covered turned out to be one of its biggest advantages.

TABLE 17: VEHICLE LOSS HOUR RESULTS FOR A HIGH DEMAND DAY (TUESDAY)

| Traffic Stream | Adaptive Controller Delay (vehicle loss hours) | NN Predictive Controller <i>Added</i> Delays (vehicle loss hours) | Rob. Predictive Controller <i>Added</i> Delays (vehicle loss hours) | 120s Cycle Fixed Controller <i>Added</i> Delays (vehicle loss hours) |
|----------------|--|---|---|--|
| 2 | 25.40 | 3.13 | 105.49 | 106.5 |
| 3 | 43.39 | 8.18 | -27.81 | 11.45 |

| | | | | |
|---|--------|-------|--------|--------|
| 4 | 56.13 | 3.68 | -16.25 | -1.26 |
| 6 | 121.77 | 27.66 | -87.94 | -28.92 |
| 7 | 8.27 | 1.53 | 62.53 | 69.71 |
| 8 | 28.27 | 11.91 | 98.01 | 107.87 |

TABLE 18: VEHICLE LOSS HOURS FOR A LOW DEMAND DAY (WEDNESDAY)

| Traffic Stream | Adaptive Controller Delay (vehicle loss hours) | NN Predictive Controller <i>Added</i> Delays (vehicle loss hours) | Rob. Predictive Controller <i>Added</i> Delays (vehicle loss hours) | 120s Cycle Fixed Controller <i>Added</i> Delays (vehicle loss hours) |
|----------------|--|---|---|--|
| 2 | 7.80 | 0.53 | 3.44 | 48.17 |
| 3 | 11.90 | -3.66 | -3.60 | 0.33 |
| 4 | 23.98 | 7.17 | 3.50 | 7.92 |
| 6 | 40.05 | 35.89 | 37.11 | 16.46 |
| 7 | 1.21 | 0.41 | 11.21 | 48.39 |
| 8 | 6.26 | 13.79 | 20.31 | 80.41 |

When the far away loop detectors are occupied due to long queues, or even when queue lengths are propagating back, near to these far away loops, the arrival flow profiles of vehicles are disturbed, affecting the ability of the classical model to recognize, and later forecast, a progression pattern, as it was designed for free flow conditions. Even when this model's parameters have been calibrated to capture the majority condition of traffic flow progression, the model made large prediction errors, for certain instances, as seen in previous chapters. While the percentage occurrences of the large errors were few, the results of Table 17 showed that they resulted in queue spillbacks, which broke down the traffic state on the corridors where this took place. Once a queue spillback occurred, since no corrective action is taken (i.e. controllers schedules have been fixed for this study), traffic delays would progress from one phase cycle to the next, preventing queues from clearing until the peak hours were over. This resulted in the extremely high delays seen in Table 17. The major advantage of the neural network's predictions, then, was that none of its prediction errors were high enough for the controller reacting to its predicted flows to give an output that would cause queue spill backs when applied to real traffic, even under highly saturated conditions. In fact, by comparing the neural network's results in that table to the fixed 120s cycle controller, it can be seen that the neural network has maintained the optimization pattern of the actuated controller which appears to priorities the traffic streams belonging to signal group 1 (i.e streams 2, 8 and 7).

Looking at both Table 17 and Table 18, it can be seen that the performance of the neural network between high demand and low demand days was relatively similar, except for directions 3, 6 and 8 whose performance decreased significantly. From the statistical analyses of Chapter 5, it was clear these three traffic streams were the most difficult for the neural network to predict the arrival flows of, especially under less dense traffic states, which was reflected in the controller outputs. Alternatively, for the lower demand day, Robertson's model performed much better than for the higher demand day. This indicated that the less dense traffic streams are less sensitive to its prediction errors, which result in green time durations that differ from the optimal. Accordingly, its errors for that day were not as severe as for the high demand day, where the model outputs were nearly not at all advantageous over the use of a fixed, non-optimized, 120s cycle controller.

An important observation that was made from the VisSim simulation model though, was that the majority of errors in both prediction models that occurred in the off-peak hours were a result of ghost cars and missed vehicles. Missed vehicles by either prediction model would go to the detector and wait for long periods of times. When the demand is low it would take quite some time for another vehicle to approach the intersection on the same traffic stream that was, in fact, predicted, allowing the missed vehicle to finally get a green light. While this occurred less for the neural network model, as seen by its lower added vehicle loss hours, it still resulted in significant delays. On the other hand, ghost vehicles would result in a green light for an empty traffic stream, taking away from the green time of traffic streams that actually need it. Hence the high percentages of ghost cars and unseen cars calculated in Chapter 5.5.3 were in fact critical to the performance of the predictive controller. To quantify the extent of their effect, the average waiting time per vehicle (in second) was calculated for the morning and evening hours, which

provided the results shown in Table 19. As shown, the vast majority of vehicle loss hours occur at night when there is a scarce amount of vehicles passing (i.e. the conditions within which the neural network is most likely to miss cars or predict ghost cars). However, in the low demand day, the difference in the average loss per vehicle between morning and evening was not very significant, indicating that the overall performance of the model on a low demand day is weaker than for the rest of the week. This is likely to be because very low density traffic conditions occur in mornings of these days as well. On the other hand, for the high demand day, the results were quite encouraging for most of the traffic streams.

TABLE 19: AVERAGE NN PREDICTION CAUSED DELAY PER VEHICLE IN MORNING AND EVENING HOURS FOR LOW DEMAND AND HIGH DEMAND DAYS

| Traffic Stream | Low Demand Day | | High Demand Day | |
|----------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | Morning Hours (s/veh) | Evening Hours (s/veh) | Morning Hours (s/veh) | Evening Hours (s/veh) |
| 2 | 0,55 | 1,15 | -0,03 | 1 |
| 3 | -7,22 | -6,2 | 2,87 | 13,75 |
| 4 | 2,14 | 3,32 | 0,26 | 2,23 |
| 6 | 22,63 | 28,91 | 9,67 | 12,46 |
| 7 | 5,64 | 7 | 0,3 | 1,83 |
| 8 | 5,38 | 7,6 | 1,82 | 3,81 |

6.5.2. PREDICTION HORIZON EVALUATION

There is a difference between historical data evaluation and in practice application. When applying predicted traffic flows into the model, a distinction needs to be made between the prediction time horizon and the application time horizon, due to the aggregation of data to 10s time steps. The prediction time horizon is 30s, because this is how long it takes vehicles in free-flow, on average, to get from one intersection to the next in this case study. With regards to the 10s time step, this is an end-to-end prediction horizon. The model waits until the end of each 10s time step to

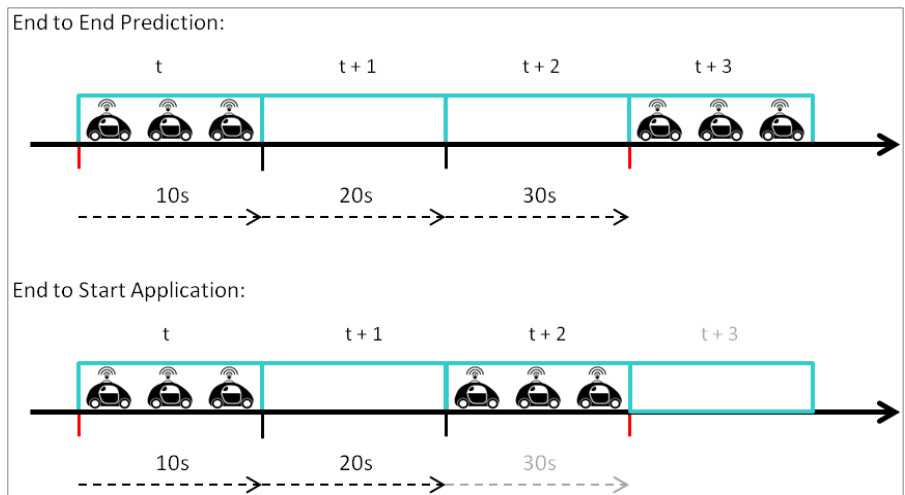


FIGURE 59: A VISUALIZATION OF THE DIFFERENCE BETWEEN THE END-TO-END PREDICTION HORIZON AND THE END-TO-START APPLICATION HORIZON.

have a complete measurement of vehicle counts per this interval. This is done both for the departure flows and the target arrival flows, which are later taken back as feedback. Hence the model estimates what the total number of vehicles arriving downstream would be, at the end of a certain 10s time step, 30s into the future, based on the number of vehicles counted at the end of the current time step from upstream. On the other hand, in application, the predicted flow must be applied end-to-start. After the 10s measurement upstream has been made, there is actually only 20s remaining before the first vehicle arrives at the downstream intersection, in free flow, as shown in Figure 59. Hence at a 20s prediction horizon, the flows for the prediction must begin to be inserted and distributed over the detectors in simulation for the predictive controller to respond to the inputs of that 10s step. For this case study then, the effective predictive control horizon is actually 20s. This explanation better clarifies the reason why more aggregated time steps were avoided for testing in Chapter 4. This application horizon will not be further reduced when spreading the flows over the virtual detectors. Once a prediction for arrival flows at a certain time step is made, the vehicles will be progressed over these detectors, in virtual space, over the short distance between the far away loop detectors and the stop line detectors, in order to trigger the detectors at a realistic rate. This process will be done via a model, that still needs to be designed. Since it is not required to simulate the vehicle movements over the detectors with a 1:1 time ratio, this process can be done instantly, immediately extracting a control output. After

another 10s, the same will be done again, and the control output will update, as it normally does with adaptive control.

A 20s prediction horizon, or even 30s traffic flow prediction horizon, as some studies recommend for GLOSA and T2G/R applications (Barthauer et. al., 2014), does not suffice for a controller's output to stabilize. A full phase cycle for an adaptive controller in the Netherlands can last up to 120s, and even 150s in some cases. Moreover, green times can undergo extensions for incoming vehicles for up to 60s. The green time extensions that were seen in the logged data of the internal controller process information (i.e. GUS data) of this specific case study, were sometimes within the range of 40s, while in others the full 60s. However, that does not mean that this horizon is insufficient. With this prediction horizon, service providers still have enough time to adapt the advice they send out to vehicles as per foreseen fluctuations. At the worst case scenario, if an extended green wave must suddenly end, to give way to a priority vehicle on a conflicting traffic stream, or to prioritize another traffic stream which is becoming to saturated, this "sudden end" to the green time will be experienced by road users 20s later. This means that the advice given to users who have less than 20s remaining to reach the intersection will remain unchanged, since the switch to red will happen after they cross the intersection anyway; and, the new platoon that has just entered the intersection will be informed that the green time is going to end upon their arrival, before they have even managed to reach half way through the corridor.

Changes in speed advice within itself is not the main concern with GLOSA and T2G/R; it is the rate at which advice changes, and the extent to which it does (Blokpoel & Niebel, 2016). The impact a change in the information given to the users has on their frustration depends on their perception of the change. For instance a change in countdown from 10s to 5s requires a sudden response from users, whereas a change from 30s to 25s does not. Hence, users are more likely to be receptive to a fluctuation in advice if the relative difference between the old advice and the new advice is smaller, even if the absolute quantity of the change was the same (Blockpoel & Niebel, 2016). With a 20s prediction horizon (which can go up to about 40s depending on the distance between intersections, the speed limit and the time step aggregation level), service providers are given a buffer time to gradually adapt the advice they give users, at a perceived acceptable rate, to avoid their frustration. Hence, the advantage of this buffer horizon is that users will no longer experience sudden changes in advice that may lead to accidents or distrust in the system. The same is applicable for an extension in green time.

6.6. CONCLUSIONS

Several conclusions can be drawn up from this chapter. With regards to the NARX neural network's prediction accuracy, the results show that the model is still not strong enough to operate without consistent corrective procedures. However, the model performance appears to be significantly more robust than the classical model, even after its parameters have been calibrated. In no case did the neural network make a prediction error that was significant enough to cause queue spillback and a breakdown of traffic flow to a wide moving jam. Moreover, while the comparison between the performances of the classical model and the neural network in previous chapters did not highlight a very large performance difference between both models, the results of this study showed the classical model's inferiority with regards to capturing the full profile of traffic flow arrivals, and its higher probabilities to make large magnitude errors are what make it unsuitable for saturated flow applications. The errors made by Robertson's model caused the controller phase cycle outputs of its predictions to create a wide moving jam, when used for the actual traffic flow data, that could not be cleared until much later in the simulated day, when the demand significantly lessened. Moreover, the performance accuracy of the NARX neural network appeared to remain consistent for the different demand days and traffic streams, except for low saturation periods, where there were ghost cars predicted or vehicles missed altogether. Although the neural network improved over the classical model with regards to these types of errors, as shown in Table 13 and Table 14, this issue still resulted in largely unacceptable delays. Finally, with regards to the prediction horizon, predictions are applied from the end of the measurement time step, to make sure all vehicles attributed to that time step have been detected, to the start of the prediction, target, time step, to make sure the prediction is for when the first vehicle in the platoon has arrived. For this case study, this results in a 20s prediction horizon. While this horizon is not enough for a full phase cycle, which lasts 120-150s, to end, it is enough for service providers to smooth out fluctuations in advice to an acceptable range with regards to user perception.

7. CONCLUSIONS, DISCUSSION, REFLECTION & RECOMMENDATIONS

In this chapter, the overall conclusions made for the several chapters of this report are first recapped in Chapter 7.1., then the specific answers to the research questions are highlighted in Chapter 7.2. Following that, in Chapter 7.3., a reflection on the work done is made, specifying the strengths and weaknesses of this research. Based on the identified weaknesses, the chapter ends with some recommendations are made on what could have been done to make this research stronger, in addition to what future work can be done to bring design one step closer to live applicability.

7.1. OVERVIEW OF RESEARCH CONCLUSIONS

The main conclusions reached in this research can be summarized as follows:

Strength of Robertson’s Classic Traffic Flow Progression Model: The strength of Robertson’s classical platoon dispersion model for predicting traffic flow progression under free flow conditions has been concurred by all research cited in this report that refers to it. Moreover, the fact that it continues to be used in modern day state-of-the-art predictive controllers reinforces these claims. However, in this study, the strengths of this model has been confirmed on several accounts. Firstly, the strongest neural network architecture, by far, with regards to traffic flow progression modeling, was the one formulated as a reconstruction of this model. Secondly, for the free flow simple corridor case study, the prediction accuracy of the classical model was highly comparable to the neural network. In fact, even for the complex case study, which contained several traffic conditions that are theoretically difficult for the classical model to adapt to, the improvements shown by the neural network, while in some cases quite significant, were not so high to the point that the classical model was rendered obsolete.

Final NARX Neural Network Design and Scalability to Intersection Configurations: The final neural network architecture was designed to be pretty scalable to different types of intersection configurations. So long as the needed inputs and outputs are available, the network parameters can be easily adapted to a new intersection. The following process can be followed for using the proposed neural network at a new intersection:

- Set the Number of Input Nodes to the number of upstream stop line detectors that are on lanes supporting traffic streams heading towards the direction of the target, case study intersection. Each input node must be connected to one of these relevant upstream stop line detectors to receive input from its detector count logging.
- There are three different outputs of this model. One main output and three supporting outputs, as per the multi-task learning feature of this model. The main output is the traffic flow arrivals at the far away loop detectors, while the supporting outputs are the detector occupancies of the long loop detectors, the percentage contribution of flow to the traffic stream and the external (WUS) output of the traffic light controller. While evaluations for the accuracy of supporting outputs was not made, their effect on the main output’s performance was monitored and was found to be positive.
 - Traffic flow arrivals (main task): number of output nodes is equal to the number of far away loops associated with the target traffic stream the neural network is attempting to predict.
 - Turning percentages: this output is dependent on a calculation that must be made online on how much percentage of flow is going to the different traffic streams of a certain side of an intersection, after the lane expansion point. For this secondary prediction task, one output node is needed.
 - Controller phase cycle state (WUS): A single output node is needed for this secondary prediction task, as only one signal head controls each traffic stream. This output node is connected to the live v-log logging of the target controller.
 - Long loop detectors: The number of nodes for this final secondary output correspond to the number of long loop detectors there are for the target traffic stream. The number of long loop detectors usually directly corresponds to the number of lanes supporting a

particular traffic stream. These nodes are connected to the long loop detectors to evaluate detector occupancy.

- With regards to the number of memory nodes included, a standard 4-5 nodes of memory are defined for the feedback loops from the outputs, while the number of memory cells needed for inputs corresponds to the extra time steps needed for vehicles traveling at the lower bound speed limit of 20km/h to arrive to the target intersection.
- The defined prediction horizon is the time it takes for vehicles to travel from one intersection to the next at free flow speed, which can be taken as 80% of the speed limit.
- The size of the modeling time step can be defined as per the preference of the user, although a warning is given that the smaller the time steps are, the less accurate the model's performance is.

Accordingly, much of the setup of the model is pretty standard, with very simple calibrations needed to different intersections, which are basically to define the correct prediction horizon, number of memory cells and the modeling time step desired. This is significantly more simple and user friendly than the computationally intensive calibration process needed to optimize the parameters of Robertson's prediction model.

Robustness of NARX Neural Network to Different Traffic Conditions: The variance in the errors of the NARX neural network for different traffic conditions was lower than that of the classical model, indicating that its prediction accuracy does not fluctuate much for different traffic conditions. Moreover, for different prediction horizons, while the performance of the NARX model worsened with farther horizons, the deterioration in the model's performance was not very significant considering the comparison was made between a 200m and a 350m prediction horizon. The latter is 1.75 times the prior, and yet the performance accuracy was not nearly halved. Moreover, while the RMSE of both the neural network and the classical model were quite similar for this simple free flow case study, the neural network best matched the traffic flow profile shape overall, as seen from its higher R-statistic, which became more prominent for the farther away horizons. Lastly, with regards to error types and magnitudes, significant neural network improvements could be seen for different defined types of errors, and when estimating the magnitudes of its errors and the frequency of their occurrence, the results showed that the neural network minimizes, almost completely, large magnitude errors, which was one of the most interpretable indicators for its robustness.

Remaining Areas of Concern for the NARX Neural Network: The main area of concern for the proposed NARX neural network, which is one that appeared on several accounts of this study, is the difficulty the model faces in capturing and forecasting dispersion patterns for highly under saturated conditions, where platoon dispersion is quite high. This can also be seen by the large number of vehicles it misses or "ghost cars" it predicts during these conditions. Often, these low density traffic streams are the ones for left turn or right turn flows.

Usability of the NARX Neural Network for Predictive Control Applications: With regards to the NARX neural network's prediction accuracy, the results show that the model is still not strong enough to operate without consistent corrective procedures. However, the model performance appears to be significantly more robust than the classical model, even after its parameters have been calibrated. In no case did the neural network make a prediction error that was significant enough to cause queue spillback and a breakdown of traffic flow to a wide moving jam. Moreover, while the comparison between the performances of the classical model and the neural network in previous chapters did not highlight a very large performance difference between both models, the results of this study showed the classical model's inferiority with regards to capturing the full profile of traffic flow arrivals, and its higher probabilities to make large magnitude errors are what make it unsuitable for saturated flow applications. The errors made by Robertson's model caused the controller phase cycle outputs of its predictions to create a wide moving jam, when used for the actual traffic flow data, that could not be cleared until much later in the simulated day, when the demand significantly lessened. Moreover, the performance accuracy of the NARX neural network appeared to remain consistent for the different demand days and traffic streams, except for low saturation periods, where there were ghost cars predicted or vehicles missed altogether. Although the neural network improved over the classical model with regards to these types of errors, this issue still resulted in largely unacceptable delays.

Advantage of this Prediction Methodology Over State-of-the-Art: The main question that comes up from these results is why would a new methodology for traffic flow progression modeling be used if a corrective process will be needed in any case. The answer to that is, the stronger the prediction model, the less sophisticated the correction process needs to be, and the less drastic the correction methods. The more harsh the corrective measures implemented the

more fluctuations in the controller there will be, which is counterproductive to the problem at hand. Moreover, a major advantage of this traffic flow progression methodology is that validation checks on the model's performance are fed back into the prediction model itself, rather than just to the predictive controller. That means that this model received inputs from the actual conditions downstream, and is able to, accordingly, better adapt its predictions to these conditions. More importantly though, when long queues propagate backwards, past the far away loop detector, this traffic state will be recognized by the prediction model itself, and as part of the adaptive characteristic of the neural network, it is able to adjust its predictions to this state. This can be seen by the very significant improvements made to the RMSE for long queue conditions of directions 4, 6 and 2, which experience this traffic state. Accordingly, a predictive controller operating on this model will not need to identify a long queue such as this as a "system failure" and resort to the fail-safe control method. This is concurred by the fact that under no conditions, including when the intersection sides were saturated, was there an experienced system failure in the form of a queue spill back, without the use of corrective measures.

Benefits of the Provided Prediction Horizon: With regards to the prediction horizon, predictions are applied from the end of the measurement time step (to make sure all vehicles attributed to that time step have been detected) to the start of the prediction (i.e. target) time step (to make sure the prediction is for when the first vehicle in the platoon has arrived). For this case study, this results in a 20s prediction horizon. While this horizon is not enough for a full phase cycle, which lasts around 120-150s to end, it is enough for service providers to smooth out fluctuations in advance to an acceptable range with regards to user perception.

7.2. ANSWERS TO RESEARCH QUESTIONS

Aside from these general conclusions, the results of this study answer the research question and sub-questions as follows:

To what extent can a short term traffic flow prediction model facilitate the conversion of a Dutch traffic actuated controller into a predictive controller, without compromising its performance?

Overall, the same efficiency of the adaptive controller was not met. With the current accuracy of the prediction model, there are still notable delay times resulting from the prediction errors. However, as previously mentioned, since this study identifies the weaknesses of the prediction model, with regards to which traffic conditions it is most likely to make the most significant errors, an error correction process can be added to the concept design to further improve the efficiency. While state-of-the-art online predictive controllers, such as SCOOT, also rely on corrective measures to prediction errors, the better prediction abilities of the proposed NARX neural network make it so that less sophistication is needed in the validation and error correction process of the proposed predictive controller. Moreover, a controller operating with this prediction model is more scalable to different intersections, without requiring infrastructural modifications or computationally intensive calibrations, so long as there are traffic stream specific long loop detectors in place, regardless of their location in relation to the maximum length of a queue. Hence, the final conclusion is that even though, by the end of this study, the prediction model had not fully achieved the research objective (i.e. facilitating a scalable predictive controller, through the development of a scalable prediction model, that operates with a similar efficiency as the adaptive controller), it has brought the current state of traffic flow progression modeling one step closer to what is needed.

- *How accurately can a prediction model forecast short-term traffic flow arrivals, under what traffic conditions, and at what time/space horizon?*

The extensive results provided in Chapters 4 and 5 provide the detailed answer to this question. However, in summary, the highest RMSE reached by the prediction model was 1.15 vehicles/10s, which is quite low. This was at a time prediction horizon of 30s, with a time step aggregation of 10s. Lower time step aggregations were found to make the prediction strength of the model poorer, although not significantly. Additionally, as the time-space prediction horizon required by the model increases, more dispersion occurs, making the prediction task for the neural network more difficult. However, interestingly enough, direction 4 was one of the directions that had a longer space horizon (although the time was the same due to a higher speed limit), and the prediction model's accuracy was actually one of the higher ones for the different traffic streams in this case study. Hence, it appears the model is more sensitive to changes in the time horizon than the space horizon. What needs to be taken into account though is that direction 4

was the only traffic stream that had a single lane far away loop detector. Accordingly, it could be that the higher predictability of this traffic stream was attributable to the better quality of the data. Nonetheless, the predictability of traffic stream 6, which had the same space prediction horizon as traffic stream 4, was also not too different than that of the other traffic streams with the shorter horizon, concurring that the neural network is not highly sensitive to changes in space horizons. Lastly, for different traffic flow conditions, while the neural network's performance fluctuated for different traffic conditions, its RMSE values were less varied than the classical model, making its average error estimates more reliable. This can also be seen by its higher R-statistic values, which indicate how well the model matches the overall flow arrival flow profile of the actual traffic. The biggest weakness of the neural network was for predicting arrivals for under saturated, highly dispersed traffic streams, which usually are at the turning lanes. Under these conditions, the model is likely to predict ghost cars, overlook short platooned actual vehicle arrivals, and under-predict the traffic volume of the rare times steps where longer platoons arrive. Nonetheless, the neural network still outperforms the classical model when at its weakest.

- *At the maximum accuracy reached by the prediction model, how is traffic flow affected if the controller calculates and outputs its green/red phases in accordance to predicted flows rather than actual flows?*

The adaptive traffic light controller used in this study was highly sensitive to arrival flows. This follows the trend in Dutch traffic light controllers towards higher reactivity. More information on how sensitive a Dutch adaptive controller is to detected traffic can be found in Appendix B. However, the answer to this question is that prediction errors resulted in longer delay times at intersections. Since the phase cycles of the traffic streams are all reliant on one another, some traffic streams received extra green time and therefore fewer delays, which was not the intension of this research, nor was it, actually, a positive result. These overestimated green times were not a product of more efficient control methods (considering the control logic itself was unchanged), but of overestimations in flow. Consequently, unnecessarily extended green times on one traffic stream lead to delays in the start of green times for traffic streams that really needed it. These errors mainly occurred during under saturated, highly dispersed, flow conditions due to ghost car predictions, overlooked cars, and underestimated platoons that, unlike the majority condition, were not as dispersed.

- *When having a controller respond to predicted flows, what benefits can be achieved with regards to outputting more stable information with regards to a controllers traffic phase cycle?*

While the intersection-to-intersection prediction horizon is too short for the controller outputs to stabilize before T2G/R or GLOSA advice can be distributed to the users, this headway provides a buffer zone for fluctuations to be smoothed out and sent to the user with gradual updates that are within an acceptable range, according to their perception. This mitigates the frustration of users with sudden and harsh fluctuations in these driver assistance use cases, allowing them to be more willing to use them. If drivers are found to trust the system more this way, then the T2G/R and GLOSA advice can be used to aid them efficiently through intersections, resolving the highest level problem of this research which is the time and fuel inefficiencies resulting from driver flow profiles when vehicles approach and cross an urban intersection.

7.3. REFLECTION ON METHODOLOGY AND RESEARCH PROCESS

As with any study, there are advantages and disadvantages to the approach followed for this study. The main **advantages** were the following: the meticulousness of the initial iterative design process; the fact that the model architecture was inspired by a well established analytical traffic flow progression model; the gradual expansion and evaluation of the model parameters; the thorough evaluation of the model's performance for different traffic states, types of errors, and intensities of arrivals; the use of a base case model for comparison that was highly relevant and calibrated to be context appropriate for each case study; and finally, that the model was tested for its intended purpose to quantify the severity of its errors. On the other hand, the **drawbacks** of this study were: the parametric analysis of the model's hyper-parameters was not thorough enough, considering more aggregation levels for predictions could have been made for both case studies used (although the prediction horizon test was constrained by the availability and location of detectors); the non-detailed evaluation of the performance of the predictive controller; the too few times the VisSim simulation models were run before averages were taken, and, the few amounts of related works that had been found with regards to neural network short term traffic flow predictions, due to it being a relatively new topic.

With regards to the research process, time management was one of the significant obstacles of this project, since certain tasks took longer than expected; and, much unanticipated time was lost during the learning curve of unfamiliar concepts and tools, and even more so during the search and procurement of the needed data, a VisSim licenses that could support external controller interfaces, and the licensed interface application that would allow the external controllers to operate within VisSim.

7.4. POSSIBLE IMPROVEMENTS TO THE WORK

With regards to the VisSim simulation runs for evaluating the predictive control concept, due to the large quantity of 10s by 10s data used for evaluation, the simulation runs took quite some time. Accordingly, there was not enough time to run several simulations before taking averages of the results. Therefore, the first improvement to the study would be for 10 simulation runs to be made, which is a ballpark average used for most simulation models, in order to mitigate the stochasticity in the model performance. Additionally, it would be even more insightful to calculate the vehicle loss hours for different times of days, in addition to the more aggregated morning and evening evaluations made for different types of days in a week. On an even more detailed level, the exact percentage of the time that the delay was caused due to a “missed vehicle” in specific could be estimated. This analysis would have provided a quantitative evaluation of how often this error occurs and for how long, on average, vehicles must wait if it does.

With regards to the neural network model, while the analysis was pretty detailed already, it would have been even more insightful if a stronger comparison between the results of Chapter 4 and Chapter 5 were made, providing a more clear analysis on the specific effects adding an adaptive traffic light controller has on the predictability of arrival flow data. Additionally, in Chapter 5 a realization is made that the model’s performance is more affected by the time horizon of a prediction than the space horizon. However, this is not explored further. More analyses can be conducted on different combinations of time-space horizons to determine how the relationship between the average speed of a platoon, the time spent on a corridor and the length of the corridor affect platoon dispersion patterns, and accordingly the predictability of these patterns.

7.5. RECOMMENDATIONS FOR FUTURE WORK

Based on the results and reflection, the following recommendations can be made for future work:

Improvements to the NARX Neural Network’s Learning Process: The experimental, iterative approach taken for developing this prediction model provided some verification that this neural network model architecture was the most ideal for short term traffic flow progression modeling. However, this does not mean that improvements to the model are not possible. The recommended improvements though do not concern the architecture, but of learning from relevant data. As previously mentioned, a month’s worth of data does not constitute big data analysis. Perhaps if the more data collected, since may, was used to further train the neural network, improvements can be seen. Regardless though, a noticed feature about the chaotic traffic flow time series is that it, on the more segregated level, the flow profiles of each day are unique, as the conditions of each day differ from the next. Accordingly, when training a neural network for 60% of a single day, it is able to make a better prediction for the remaining 20% test set than it does when trained for a larger amount of data. That is because it was trained for a highly relevant portion of data. On the other hand though, when this model is used to predict arrivals for the start of the next day, the results are quite poor, as this day does not match the day before. Hence, for a neural network to have the pattern recognition advantages of training for a large dataset, and the parameter customization advantages of training for a smaller, more relevant dataset, online adaptive learning is recommended. It is recommended for the neural network to be trained offline first for as large of a historical dataset as available, and then when implemented for online adaptive learning to be added into the model’s algorithm. This will allow the model’s already locally optimized parameters to continuously calibrate per day to the new conditions of that given day. This not only would give higher accuracy, but would be quite useful for when the traffic behavior is changed through GLOSA or T2G/R applications.

Further Scalability for the NARX Neural Network: Currently, the recurrent neural network structure proposed is still considered as shallow learning (although the use of feedback loops and memory classifies it as dynamic learning, which is deeper than the static learning of a feed forward neural network). A true deep learning upgrade to the model

would be to add in a Long-Short-Term-Memory (LSTM) layer into the model. The purpose of this layer is that it eliminates the need to select the amount of memory cells needed, and from which time steps. An LSTM recurrent model can decide this information independently. Accordingly, with an added LSTM layer, the user of a NARX neural network would only be required to plug in the correct inputs and outputs to the correct nodes, and specify what the prediction horizon is (based on the free flow travel time from intersection A to intersection B). Additionally, a very useful feature of an LSTM layer is that it may choose to keep a very old time step in its memory, which could be related to a certain impedance it wants to remember in the future, in case it occurs again. Hence, with this deep learning layer, improvements to the accuracy are likely to be seen as well, not just in the model's scalability.

Exploring the Use of Other Data Sources: Part of the reason why the neural network's performance was not as high as hoped, was due to quality concerns regarding the used data. However, it is not unusual for detector data to be faulty, since it is not thoroughly monitored and not often used on such a segregated scale. More commonly, this historical data is used for more macroscopic traffic flow analysis purposes. Accordingly, improvements to the model can be made through the assistance of other data sources, such as the emerging floating car data, trajectory data, and camera detection data. The issue with floating car data is that there is no abundance of it yet for big data analysis, and GPS information is not yet specific enough to classify which lane vehicles are on or which traffic stream they are heading towards. However, this data can be incorporated into the model as a supplement to the detector data. Similarly, while trajectory data can provide some insights on the turning percentages of vehicles throughout different time periods, it does not provide the detailed flow count information upstream that is needed for traffic progression modeling. Accordingly this type of data can be used as a supplement as well. Camera detector data on the other hand can theoretically be used as an alternative to detector data, as it provides the same type of information. However, it is missing far away loop data, which is very significant for green light extensions; and so, it will most likely not be suitable if used completely alone.

Further Analyses of Travel Time Delays: This was already discussed in Chapter 7.4., but should be referenced again here, as it is quite an important step to be done.

A thorough Analysis on the Affect of Space & Time Horizons on Model Performance: Although Chapter 4 did experiment with two different prediction horizons, more analyses are required for conclusive results. More clarity is needed on whether it's the time or physical space horizon that has the most affect on the model performance, and the role average speed plays.

A thorough Analysis of Safety and Efficiency Regarding Green Time Extensions: One of the main KPIs considered when selecting the location of the far away loop, is the driver's sight distance. Within a certain distance, the driver will be able to clearly see the traffic light; and, if green, the driver is likely to accelerate towards it. This is due to the comfort drivers have in the Netherlands that green time durations will most likely be extended for them if they have come this close to a green traffic light already. Hence, a significant feature of the far away loop, is that it is placed in a location after which if a sudden switch to red is made, the driver could be at risk of an accident due to sudden braking. For that reason, in addition to adding traffic flow efficiency, the green waves of adaptive traffic light controllers extends its green cycle phase. However, vehicles are only considered as part of a platoon, and are therefore given extra time for right of way, if they are within a certain time headway of the last detected vehicle. Otherwise, arriving vehicles are considered to be safely part of a new platoon and do not get an extension to green. Hence, with the proposed predictive control strategy in Chapter 1.3., there is a safety risk that needs to be thoroughly evaluated regarding green time extensions. If, in the prediction, a vehicle is considered as part of a new platoon, and so is not granted a green time extension, but, in actuality, the vehicle should be given one, the uninformed driver will be expecting to pass through the green, and be shocked by the sudden change to red. Accordingly, the driver will be placed in a risk of accident. A possible mitigation plan for this risk is to uniformly apply a certain safety factor of a few time steps, for the green time extended duration, for all signal groups. This could be a uniform 5s extension, for example. However, an efficiency check must be made in that case, since the extension of green time for one signal head will delay the green time start of conflicting signal groups, which result in further added delays.

Implementation of a Corrective Measures Procedure for Predictive Controller: Even after improvements to the neural network's accuracy are made, a verification and corrective measures system must be added to the predictive controller. This assures the robustness of this type of controller under any conditions that may lead to a sudden failure with regards to predictions. For now, without any improvements to the neural network's prediction accuracy,

this system is needed even more, to mitigate the small, but consistent, errors made by the model, resulting in high vehicle loss hours.

Modified Proposed Scenario:

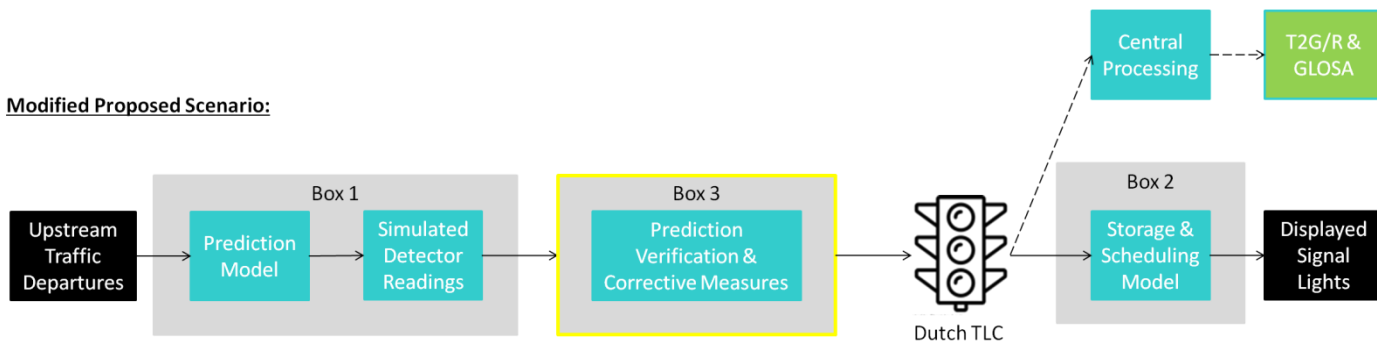


FIGURE 60: VERIFICATION CHECK & CORRECTIVE MEASURES ADDITION TO CONCEPT DESIGN

It is also recommended that the “corrective measures” be applied in the form of changes to controller inputs, so as to maintain the original goal of keeping the adaptive controller fully intact. An example of a corrective measure for inputs would be to add a ghost vehicle in the next time step, to trigger the controller to give a green phase cycle state to a missed, unseen, vehicle.

Extending the Effective Application Prediction Horizon: Even though the results of Chapter 4 showed that more largely aggregated time steps result in more robust predictions, the difference between the 5s and 10s time step aggregation was not very significant. This difference might even be reduced after the recommended improvements to the neural network, in this chapter, are implemented. On the other hand, with the lower aggregation level, the effective application prediction horizon is extended. This gives service providers more time to better smooth out fluctuations in traffic light controller outputs, allowing users to be even more receptive to the advice provided.

Evaluating User Behavioral Adaptations to Different Advice Fluctuations: Thus far, no quantitative research is provided on the acceptable rate of advice change, and the acceptable magnitude of advice change, within which users will still be receptive to the advice. Even in Blockpoel & Niebel (2016)’s study, the term “perception” is generally defined. Hence a study is needed on what range of advice change is “perceived” as acceptable by the average users, and after what frequency level of advice change will the user begin to resent and ignore the service out of frustration.

8. REFERENCES

9. Abdelaziz, M., & Baass, K. (1996). Traffic Platoon Dispersion Modeling on Arterial Streets. *Transportation Research Record: Journal of the Transportation Research Board* , 49-53.
10. Aleksandar, S., Stevanovic, J., & Kergaye, C. (2013). Green Light Optimized Speed Advisory Systems. *Transportation Research Record: Journal of the Transportation Research Board* , 53-59.
11. Allsop, R. B. (1971). SIGSET: A Computer Program for Calculating Traffic Capacity of Signal-Controlled Road Junctions. *Traffic, Engineering and Control*, 58-60.
12. Allsop, R. E. (1976). SIGCAP: A Computer Program for Assessing the Traffic Capacity of Signal Controlled Road Junctions. *Traffic Engineering and Control*, 12(2).
13. Asadi, B., & Vahidi, A. (2010, April). Predictive Cruise Control: Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time. 707 - 714.
14. Baluja, S. (2015). Approximating the Effects of Installed Traffic Lights: A Behaviorist Approach Based on Travel Tracks. *IEEE 18th Conference on Intelligent Transportation Systems* . IEEE.
15. Barthauer, M., & Friedrich, B. (2014). *Evaluation of a Signal State Prediction Algorithm for Car to Infrastructure Applications*. Retrieved 2017, from Transportation Research Procedia .
16. Bie, Y., Liu, Z., Ma, D., & Wang, D. (2013). Calibration of Platoon Dispersion Parameter Considering the Impact of the Number of Lanes. *Journal of Transportation Engineering*.
17. Blokpoel, R., & Niebel, W. (2016, June). *Advantage of Cooperative Traffic Light Control Algorithms*. Retrieved 2017, from ITS European Congress.
18. Blokpoel, R., & Vreeswijk, J. (2016). Uses of Probe Vehicle Data in Traffic Light Control. *Moving Forward: Innovative Solutions for Tomorrow's Mobility*, (p. Poland). Warsaw.
19. Blokpoel, R., Härrri, J., Caselli, F., Niebel, W., & Leich, A. (2015). *Cooperative Self-Organizing System for low Carbon Mobility at low Penetration Rates*.
20. Boillot, F. (1992). Optimal Signal Control of Urban Traffic Networks. *Road Traffic Monitoring*.
21. Brilon, W., & Zurlinden, H. (2016). Ueberlastungswahrscheinlichkeiten und V erkehrsleistung als Bemessungskriterium fuer Strassenverkehrsanlagen (Breakdown Probability and Traffic Efficiency as Design Criteria for Freeways). *Forschung Strassenbau und Strassenverkehrstechnik*(870).
22. Caruana, R. (n.d.). Multitask Learning. *Learning to Learn*, 95-133.
23. European Comission. (2017, September). *Priorities*.
24. European Commission- Directorate General for Environment . (2004). *Reclaiming City Streets for People: Chaos Or Quality of Life?*
25. Evans, L., Herman, R., & Lam, T. (1976). Multivariate Analysis of Traffic Factors Related to Fuel Consumption in Urban Driving. *Transportation Science*.
26. Farzaneh, M., & Rakha, H. (2006). Procedures for Calibrating TRANSYT Platoon Dispersion Model. *Journal of Transportation Engineering*, 548- 554.
27. Feddes, G. (2017, March). Between Rules and Riding: The (Im)possibilities in the admission of vehicles. *Presentation*. The Hague, Netherlands: Presentation at the 1st Public event of the STAD project, "Automated Driving: A Silver Bullet for Urbanized Regions?" .
28. Foss, S. (2011). The G/G/1 Queue. *Wiley Encyclopedia of Operations Research and Management Science*.
29. Gartner, N. (1983). OPAC: Strategy for Demand-Responsive Decentralized Traffic Signal Control. *Transportation Research Record: Journal of the Transportation Research Board*.
30. Geroliminis, N., & Skabardonis, A. (2005). Prediction of Arrival Profiles and Queue Lengths Along Signalized Arterials by Using a Markov Decision Process. *Transportation Research Record: Journal of the Transportation Research Board*.
31. Government of Netherlands. (2015). *Traffic Management - Mobility, Public Transport and Road Safety*.
32. Guberinić, S., & Minić, S. M. (1993). Signal Group: Definitions and Algorithms. *Yugoslav Journal of Operations Research* .
33. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on Image Net Classification. *IEEE International Conference on Computer Vision (ICCV)*.

34. Head, K. L. (1995). Event-based short-term traffic flow prediction model. *Transportation Research Record*, 45-52.
35. Henke, N., Bughin, J., Chui, M., Manyika, J., Saleh, T., Wiseman, B., & Sethupathy, G. (2016). *The Age of Analytics: Competing in a Data-driven World*. McKinsey Global Institute. McKinsey & Company.
36. Henry, J. J., Fargas, J., & Tuffal, J. (1984). The PRODYN Real Time Traffic Algorithm. *4th IFAC-IFOR Conference on Control in Transportation Systems*.
37. Hoffman, G., & Zimdahl, W. (1988). Recommended speed indication within the vehicle - a contribution to fuel saving through the "Wolfsburg Wave" information system. *Energy Efficiency in Land Transport*. Luxemburg.
38. Hong, H., Huang, W., Song, G., & Xie, K. (2014). Metric-Based Multi-Task Grouping Neural Network for Traffic Flow Forecasting. *International Symposium on Neural Networks*.
39. Hunt, P. B., Robertson, D. I., Bretherton, R., & Winton, R. (1981). SCOOT- A Traffic Responsive Method of Co-Ordinating Signals. *TRRL Laboratory Report*.
40. Jiao, P., Sun, T., & Du, L. (2014). A Bayesian Combined Model for Time-Dependent Turning Movement Proportions Estimation at Intersections. *Mathematical Problems in Engineering*.
41. Katwijk, R. V. (2008). Multi-Agent Look-Ahead Traffic-Adaptive Control. Delft: Delft School of Technology.
42. Koukoumidis, E., Martonosi, M., & Peh, L.-S. (2012). Leveraging Smartphone Cameras for Collaborative Road Advisories. *EEE Transactions on Computers*. Princeton.
43. Krijger, P. (2013, July). Traffic Light Prediction for TomTom Devices. *Thesis*. Technische Universiteit Eindhoven.
44. Little, J. D. (1966, August). The Synchronization of Traffic Signals by Mixed-Integer Linear Programming. *Informs*.
45. Liu, Z., Bie, Y., Li, Y., & Pei, Y. (2014). *Development of a New Platoon Dispersion Model Considering Turning Vehicles in Urban Road Environment*. Retrieved 2017, from Advances in Mechanical Engineering .
46. Lowrie, P. R. (n.d.). The Sydney Coordinated Adaptive Traffic System - Principles, Methodology, Algorithms. *International Conference on Road Traffic Signaling*, (p. 1982).
47. Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2015). Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*.
48. Masters, J. (2017). Dutch deployment Heralds Dawn of C-ITS Era. *Intelligent Transport Systems*.
49. Mathew, T. V. (2014). Traffic Progression Model. *Transportation Systems Engineering*.
50. Mauro, V., & Taranto, C. (1989). Utopia. *IFAC Proceedings Volumes*.
51. McKinsey Global Institute . (2011, May). *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. Retrieved 2017
52. Muller, T., & Leeuw, M. (1978). New Method to Design Traffic Control Programs . *Transportation Research Record: Journal of the Transportation Research*.
53. Murrell, D. (1980). *Passenger Car Fuel Economy, EPA and Road: A Report to the Congress in Response to the National Energy Conservation Policy Act of 1978, Public Law 95-619*. U.S Environmental Agency .
54. Nunzio, G. D. (2015). Traffic Eco-management in Urban Traffic Networks.
55. Pacey, G. (1969). The Progress of a Bunch of Vehicles Released from a Traffic Signal. *Road Research Laboratory Report*.
56. Pan, B., Demiryurek, U., Shahabi, C., & Gupta, C. (2013). Forecasting Spatiotemporal Impact of Traffic Incidents on Road Networks. *2013 IEEE 13th International Conference on Data Mining*.
57. Paul, B., Mitra, S., & Maitra, B. (2016, June). Calibration of Robertson's Platoon Dispersion Model in Non-lane Based Mixed Traffic Operation. *Transportation in Developing Economies*.
58. Protschky, V., Wiesner, K., & Feit, S. (2014). Adaptive traffic light prediction via Kalman filtering. *IEEE Intelligent Vehicles Symposium Proceedings*. Dearborn, MI, USA.
59. Qiao, F., Yang, H., & Lam, W. H. (2001). Intelligent Simulation and Prediction of Traffic Flow Dispersion. *Transportation Research Part B: Methodological*, 843-63.
60. R. A., V., & Young, C. P. (1986). Self-Optimizing Traffic Signal Control Using Microprocessors – the TRRL 'MOVA' Strategy for Isolated Intersections. *IEE Conference Publication*.

61. Rakha, H., & Ding, Y. (2002). Impact of Stops on Vehicle Fuel Consumption and Emissions. *Journal of Transportation Engineering* , 23-32.
62. Robertson, D. I. (1969). TRANSYT: A Traffic Network Study Tool. *Road Research Laboratory Report*.
63. Robertson, D. I. (1986). Research on the TRANSYT and SCOOT Methods of Signal Coordination. *Institute of Transportation Engineers Journals*.
64. Rumsey, A., & Hartley, M. (1972). Simulation of a pair of intersections. 522-525.
65. Rutsch, S. (2014). *Connected Signals*. Retrieved from Deutsche Bahn, Deutsche Bahn Digital Ventures: [www.deutschebahn.com/en/Digitalization/startups/db_startups/14616968/ Connected-Signals.html](http://www.deutschebahn.com/en/Digitalization/startups/db_startups/14616968/Connected-Signals.html).
66. Scheepjens, R. (2016). Algorithm Design for Traffic Signal Timings Predictions of Vehicle-Actuated Controlled Intersections Using Support Vector Regression. Delft University of Technology.
67. Sen, S., & Head, K. (1997, February). Controlled Optimization of Phases at an Intersection. *Transportation Science*.
68. Shebeeb, O. (2016). Helping Autonomous Vehicles at Signalised Intersections. *Conference of the Transportation Association of Canada*. ON, Toronto.
69. Siemens Mobility, Traffic Solutions UTC System. (2016). *Scoot User Guide*.
70. Sjöberg, J., Hjalmarsson, H., & Ljung, L. (1994). *Neural networks in system identification. A Post-print Volume from the IFAC Symposium* (pp. 181-184). Copenhagen, Denmark: Pergamon Press.
71. Stamatiadis, C., & Gartner, N. (2014). MULTIBAND-96: A Program for Variable-Bandwidth Progression Optimization of Multiarterial Traffic Networks. *Journal of the Transportation Research Board*.
72. Stanford University. (2017). CS231n: Convolutional Neural Networks for Visual Recognition. Stanford University.
73. Steffen Axer, B. F. (2016). *A Methodology for Signal Timing Estimation Based on Low Frequency Floating Car Data: Analysis of Needed Sample Sizes and Influencing Factors*. Retrieved from Transportation Research Procedia.
74. Stevanovic, A., Stevanovic, J., & Kergaye, C. (2013). Comparative Evaluation of Benefits from Traffic Signal Retiming and Green Light Optimized Speed Advisory Systems. *Transportation Research Board*.
75. Suthaputchakun, C., & Sun, Z. (2015). A Novel Traffic Light Scheduling Based on TLVC and Vehicles' Priority for Reducing Fuel Consumption and CO₂ Emission. *IEEE Systems Journal*, 1-9.
76. Thompson, H., Paulen, R., Reniers, M., Sonntag, C., & Engell, S. (2015). *Analysis of the State-of-the-Art and Future Challenges in Cyber-physical Systems of Systems*. Retrieved from Cyber Physical Systems of Systems.
77. Weisheit, T., & Hoyer, R. (2014). Prediction of Switching Times of Traffic Actuated Signal Controls Using Support Vector Machines. *Advanced Microsystems for Automotive Applications*, 121-129.
78. Wu, L., Yusheng, C., Chu, J., & Zhang, H. (2015). The Influence of Intersections on Fuel Consumption in Urban Arterial Road Traffic: A Single Vehicle Test in Harbin, China. *PLoS One*.
79. X. Liu, H., Wu, X., Ma, W., & Hu, H. (2009, August). Real-time queue length estimation for congested signalized intersections. *Science Direct*, 412-427.
80. Yin, Y. (2008). Robust optimal traffic signal timing. *Transportation Research Part B; Methodological* , 911-924.
81. Yu, L. (2000). Calibration of Platoon Dispersion Parameters on the Basis of Link Travel Time Statistics. *Transportation Research Board*, 89-94.
82. Yu,, X., & Recker, W. (2006). Stochastic Adaptive Control Model for Traffic Signal Systems. *Transportation Research Part C: Emerging Technologies*, 263-82.
83. Zurschmeide, J. (2015, August). *Stop Wasting Gas in the City with the App That Knows When Traffic Lights Will Change*. Retrieved from Digital Trends: www.digitaltrends.com/cars/enlighten-app-predicts-traffic-lights/.

9. APPENDIXES

A. CONTROL STRATEGY ALTERNATIVES

Essentially, to reach the needed design, an available control logic must first be selected (from the many possible alternatives), and then a concept design will be proposed to identify how this logic will be implemented in a way that satisfies the need statement. Before doing so though, the term “signal group” will be defined, as it is a key term with regards to traffic light control at intersections.

Signal Group:

At its core, traffic control on an intersection can be summed up as a mechanism for permitting and disallowing the right of way for different traffic streams with sequences of traffic light indications (Guberinic & Minic, 1993). For traffic streams that are not in conflict (i.e. their flows have no chance of collision together), a single sequence of traffic light indications can be used to control them (Guberinic & Minic, 1993). Accordingly, a set of identical traffic light indications within one intersection, used to permit or prevent the flow of non-conflicting traffic streams in the same matter, at the same time, is defined as a signal group. In traffic control documentation though, signal groups are often referred to as “signal blocks” as well. An example showing three different signal groups (or blocks) at a typical intersection can be found in Figure 61.

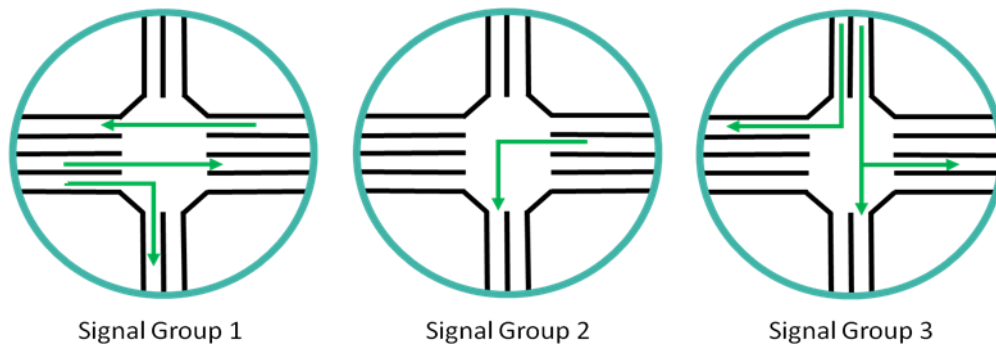


FIGURE 61: AN EXAMPLE OF THREE POSSIBLE SIGNAL GROUPS FOR A TYPICAL INTERSECTION.

Following this definition, the alternatives will now be defined. There are several types of control strategies available in the market, as listed in section 6 of **Error! Reference source not found.** However, all these controllers attempt to reach the same objective, which is to achieve an optimized signal schedules, but through different methodologies. The key differences between these controllers, are: firstly, whether they perform their optimizations offline or online (i.e. in real time); and, secondly, the time horizon of traffic information that is used as input. Within this chapter, a brief definition for each strategy will be provided, to outline the different directions that can be taken for the concept design.

Fixed Controller:

The simplest design for a TLC is to have it output fixed schedules. These types of controllers range from the most naïve version, which outputs a single fixed green time duration that rotates amongst the different signal groups, to more complicated control strategies that attempt to coordinate the signal schedules of subsequent intersections together, offline, in a manner that would result in network optimized control (De Nunzio et. al., 2015). Using historical data, empirical observations and advanced statistical modeling, the pre-set signal timings for the more sophisticated fixed controllers are determined through an optimization model that aims to maximize vehicle mobility within a network (using an estimated speed range) with the least amount of stops at signals (De Nunzio et. al., 2015). For instance, the fixed controller MAXBAND (Little, 1966) uses binary mixed integer linear programming, simplified using the branch and bound method, to achieve this optimization goal for a single corridor. An expanded version of this controller, MULTIBAND (Stamatiadis & Gartner, 1996), which additionally incorporates time clearance of existing queues, turning percentages, and multiple signal groups, is deployed on many road networks in North America.

The most famous fixed controller though is TRANSYT (Robertson, 1969). The fixed cycle schedules of this controller are optimized offline, based on a traffic model. Embedded in this model is Robertson’s platoon dispersion

algorithm (Robertson, 1969), for progressing traffic from one intersection to the next, and the fundamentals of Queuing Theory, to estimate the queue evolutions with time. The objective function of a TRANSYT controller is to create a TLC schedule that would minimize the total sum of queues, from historical data, within a modeled network, which implicitly minimizes the total number of stops as well. This is achieved using a heuristic Hill-Climbing optimization algorithm, which keeps updating model schedules until a local minimum is found (Robertson, 1969). Despite the successful use of TRANSYT in the United States, up to this modern day, its inventor had questioned, several years ago, if “as we approach the end of this century, traffic engineers and drivers will continue to tolerate signals with green and red times that were decided by flows and queues that happened to be observed on one day many years earlier, rather than in the last five minutes.” (Robertson, 1986).

Fixed control strategies have been coined as *static strategies* (De Nunzio et. al., 2015); since, for all variations of this type of controller, the traffic light schedules are optimized offline using historical data, and then fixed for long periods of time. Accordingly, the main disadvantage to that methodology is that these types of controllers are not adaptive to changes in traffic behaviors over time. Fixed controllers do not recognize that daily demands are not identical for each day, don't take into account seasonal effects on demand, don't automatically calibrate their schedules to account for population growth, and do not automatically update to accommodate for new vehicle route choices that may occur in the future, due to urban developments. For those reasons, static strategies are considered to develop “blind” controllers (De Nunzio et. al., 2015).

Traffic Actuated Controller:

The most popular alternative type of control is traffic actuated control, which optimizes its schedules online, based on detected vehicle counts in real time. “Half-fixed”, “actuated” and “adaptive” traffic control strategies are all considered variations of this type of control, with the differences between them lying in their level of adaptively to arriving and departing flows. For instance, *half-fixed control* can be described as a *stage-based strategy* (De Nunzio et. al., 2015). The defining characteristic of this type of strategy is that the green phase still goes around, in a specified order, to each signal group. However, unlike purely fixed controllers, the standard setting for the green phase of half-fixed controllers is a defined lower bound for green time duration. The green time duration remains this way for each signal group, until traffic is detected on a certain flow direction. Once a signal group gets its turn, if vehicles have been detected at its start line, the green phase for this signal group is extended to clear the detected queue (De Nunzio et. al., 2015). The benefit from the added flexibility to this type of controller, then, is that it reduces wasted green time on empty lanes, and, therefore, reaches signal groups with detected demands faster. Accordingly, the optimization objectives of these types of controllers are to minimize total intersection delay, as is the case with SIGSET controllers (Allsop, 1971). To assure the controller continues its cyclic behavior though, an upper bound for green time is also set, after which the controller will be forced to switch its green phase to the next signal group, regardless. This can be problematic for some high-demand flow directions during peak hours. However, to alleviate this concern, and avoid queue spill back, some of these types of controllers, such as SIGCAP (Allsop, 1976) also feature an additional constraint, which is that queues cannot exceed a certain amount before a signal group is serviced.

On the other hand, both actuated and adaptive controllers are considered to take *phase-based strategy* approaches towards traffic management. Similarly to state-based strategies, the objectives of this type of strategy are also to either minimize delay or maximize intersection capacity. However, phase-based strategies extend one step further to consider all different possible staging combinations (De Nunzio et. al., 2015). By processing this information, using binary mixed integer linear programming (simplified using branch and bound methods), this strategy is able to further optimize a traffic light controllers schedule, by assigning the next green phase to the signal group that will bring the most advantageous results, instead of the next one in line (De Nunzio et. al., 2015). The key difference between actuated and adaptive control, then, is that an *actuated controller* responds to real time queues at intersections, with no foresight on incoming vehicle flows (Blokpoel & Niebel, 2016); meaning, outputted green time durations are static, and do not extend to accommodate for incoming vehicles. Alternatively, for *adaptive controllers*, such as MOVA (Vincent & Young, 1986), an upstream “entrance detector loop” or “far away loop” is placed to count incoming vehicles on a link, for further optimization of green time durations (Blokpoel & Niebel, 2016). The difference between actuated and adaptive control can be seen in Figure 62, where the prior controller outputs enough green time for one vehicle in its traffic light schedule, while the latter outputs an extended green time to take into account the approaching platoon of vehicles as well.

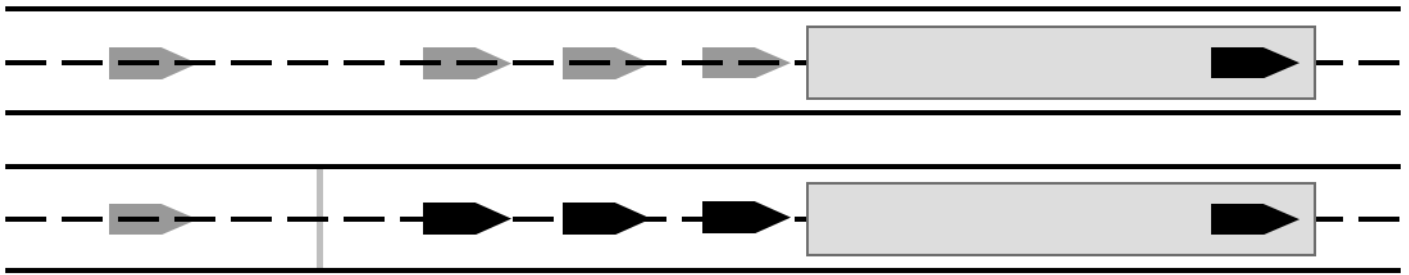


FIGURE 62: VISION OF AN ACTUATED CONTROLLER (TOP) VS. THE VISION OF AN ADAPTIVE CONTROLLER (BOTTOM)

Predictive Controller:

The upgrade between actuated and adaptive control mainly relies the differences in the horizon of a controller's vision, when it runs its optimization model for split times, cycle durations and phase sequences (Blokpoel & Niebel, 2016). This is also the key difference between adaptive and predictive control. With *predictive control*, the controller's line of vision extends much further back, in both space and time, allowing it to anticipate traffic, in the short-term, before its arrival. Unlike previous control methodologies though, there is a lot of variability with predictive control. With a prediction horizon that extends back, at least, to the exit flows of the predecessor intersection, this type of controller must be equipped to handle the large uncertainties associated with forecasting traffic flow progression that were discussed in Chapter 2.2.1. With no field-operational benchmark on how to reconstruct traffic flow patterns from point A to point B though, each predictive control designer has chosen to follow their own approach for handling these uncertainties (De Nunzio et. al., 2015). However, the underlying objective of minimizing waiting times at intersections remains the same.

There are three main approaches for predictive control used today. The first, which is implemented in the British controller SCOOT (Hunt et. al., 1981), is to follow the same process as TRANSYT, but in real time. Accordingly, this controller is also heavily reliant on traffic flow modeling for determining its signal phase schedules. However, unlike TRANSYT, which optimizes its schedules offline, SCOOT reconstructs detected departure flows from an upstream intersection (also using Robertson's platoon dispersion algorithms) online, every four seconds, into an arrival flow downstream, then uses the resulting modeled queue predictions to determine the upcoming signal phases (Hunt et. al., 1981). Every minute though, SCOOT measures the actual queues and determines whether to shorten or lengthen its predicted cycle accordingly (Hunt et. al., 1981). Hence, even SCOOT-outputted green times fluctuate. Alternatively, the Australian controller SCATS (Lowrie, 1982), adapts a semi-online-semi-offline strategy. After selecting a small sub-network of intersections for online adaptive optimization, flows exiting from this sub-network are spread out to the larger network offline (once again, using Robertson's platoon dispersion algorithms) to optimize the schedules of the remaining intersections (Lowrie, 1982). No online validation system is implemented in SCATS, and so schedules set offline, using flow predictions, remain constant. The important difference between SCATS and purely offline fixed schedule planning though is that the vehicle data spread out in the SCATS model is much more time relevant than historical data. The last approach, taken by predictive controllers such as OPAC (Gartner, 1983), PRODYN (Henry et. al., 1984), UTOPIA (Mauro & Taranto, 1989), CRONOS (Boillot et. al., 1992), and RHODES (Sen & Head, 1997), is to follow a model predictive control strategy. These controllers estimate future, optimal, online switching schedules, using current traffic flow measurements and forecasted arrival rates, with a rolling prediction horizon that averages from one to two minutes, and is updated every four seconds. The purpose of using longer horizons for traffic flow predictions and schedule optimizations is to be able to capture a full cycle, and avoid facing controller fluctuations due to shortsighted control outputs within the traffic responsive framework (De Nunzio et. al., 2015).

B. CURRENT DUTCH VEHICLE ACTUATED CONTROL STRATEGY

Before making the design choices, an understanding of how traffic control is currently implemented in the Netherlands is needed, so as to make relevant improvements. Nearly all Dutch signalized intersections operate using some variance of vehicle actuated control. In recent years, however, most older controller models are being removed, and switched with newer, more adaptive models. Accordingly, the visible trend in Dutch control strategies are in favor of the adaptive end of the spectrum. The complexity of Dutch urban intersections does not end there though. Some intersections service other modalities, such as public transport, bicycles and/or pedestrians as well; and, in the Netherlands, each modality is assigned its own infrastructure with its own signal light head. This facilitates giving priority right of way for certain modalities, if needed, which, consequently, requires more sophisticated control strategies.

Under typical conditions though, where no priority right of way is included, the underlying objective for adaptive Dutch TLCs is to create a balance in green phases between the different signal groups. Signal groups include traffic streams of all modalities. The more conflicting traffic streams there are, the more difficult this becomes; as, green times must be long enough to clear the queues of the signal group receiving right of way, but not too long to the point that the queues of other conflicting signal groups are beginning to build up. To overcome this issue, a multilayer, hierarchical, control architecture is implemented in the Netherlands. Within this architecture, each individual traffic light for each traffic stream has the single objective of optimizing green times for its own traffic stream (i.e. local optimal flow efficiency). These lower level control agents do not communicate or take each other into consideration within their optimization computations. On a higher level though, a supervisory agent is placed that follows a highly structured procedure for imposing proper constraints on the control agents. These constraints ensure the safety of vehicles, and opt for global flow efficiency within the intersection. This control structure is outlined in Figure 63, where the agent labels for this figure are chosen with reference to the sample intersection presented in Figure 64, and the constraints follow the block structure of Figure 65.

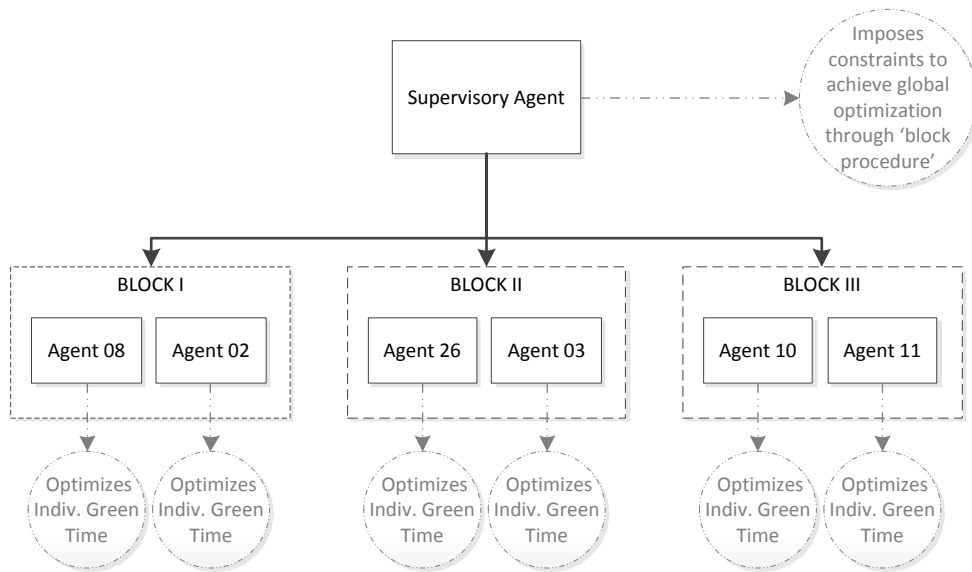


FIGURE 63: MULTI-LAYER, HIERARCHICAL, DUTCH TLC CONTROL STRUCTURE

Block procedure is a systematic procedure that aims to minimize green-time opportunity costs (Scheepjens, 2015). To outline the different stages of this procedure, the example intersection in Figure 64 will be used. For this intersection, three different blocks (i.e. signal groups) are identified, as shown in Figure 65, grouping the main non-conflicting traffic streams together. Using these defined signal groups, the first step for a block procedure is *primary realization* (Scheepjens, 2015). This entails that each block will be given a turn, in successive order, as shown in Figure 65, to be *active*. An active signal group is permitted green time for any traffic stream that has a demand for it; however, if a traffic stream is empty, a green signal phase is not imposed on it. Alternatively, the controller proceeds to *primary ahead realization* (Scheepjens, 2015). Following this strategy, the controller begins to search within the direct successor of the active signal group to see if there is a traffic stream in that block that is non-conflicting with the traffic stream in the active block that does have demand for green. If that is the case, then that traffic stream gets a

green phase as well. For example, if block one has primary realization, but there are no detected vehicles on traffic stream eight, the controller will look into block two and give a green phase to traffic stream three, as it does not conflict with traffic stream two. The controller will then continue to the last stage of the block procedure, which is *alternative realization* (Scheepjens, 2015). Once the controller has reached this stage, any traffic stream, within an inactive block, that can safely take the spot of an empty traffic stream, in an active block, becomes eligible, as shown in Figure 66.

Essentially, block procedure can be summed up as a systematic method of prioritization. Traffic streams within active blocks have first priority for receiving a green signal phase, then the traffic streams within the block that is scheduled to be active next (in the primary realization phase) will have second priority, and all remaining traffic streams have last priority. This cycle continues as the active state, which has a fixed maximum duration, shifts from one block to the next. However, this cycle can be disturbed if a priority modality (e.g. blue vehicle, such as a police car or ambulance) appears in the vicinity of an intersection. These vehicles receive *priority realization*, which means the controller will interrupt its cycle, and immediately output a new signal phases, to assure these vehicles are given right of way. While this is a highly structured procedure, it still leaves a lot of space for lower level agents to adapt their signal schedules to detected traffic along their streams, in real time.

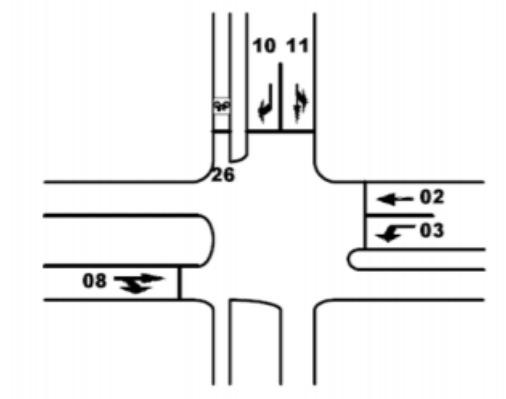


FIGURE 64: A SAMPLE INTERSECTION FROM VAN KATWIJK(2008) , FEATURING 5 LABELED VEHICLE TRAFFIC STREAMS, AND ONE BICYCLE STREAM (NO. 26). IMAGE WAS REFERENCED FROM SCHEEPJENS (2015).

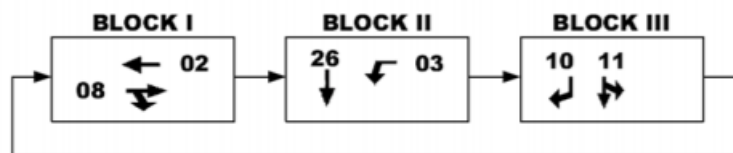


FIGURE 65: BLOCK STRUCTURE FOR THE SAMPLE INTERSECTION, INCLUDING AN OUTLINE FOR THE PRIMARY REALIZATION PROCESS FROM VAN KATWIJK (2008). IMAGE WAS REFERENCED FROM SCHEEPJENS (2015).

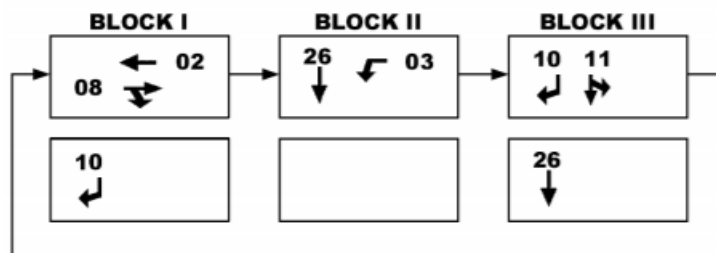


FIGURE 66: AN OUTLINE FOR THE ALTERNATIVE REALIZATION PROCESS FOR THE SAMPLE INTERSECTION FROM VAN KATWIJK (2008). IMAGE WAS REFERENCED FROM SCHEEPJENS (2015).

With a Dutch adaptive controller, traffic is detected through three types of detectors, as shown in Figure 67. Each serves a different function. When the first vehicle in a platoon, during a red cycle, reaches the stop line detector, it initiates a request for green. The controller then assigns the minimum green time, which is four seconds, to this traffic stream. If the long loop detector is also occupied, a longer green cycle is given that would be enough to clear out this detector (based on its positioning and length). Once the supervisory agent permits green time for this traffic stream, the decided upon green time can be adapted further. For instance, if the long loop detector has become unoccupied for 0.5 seconds, the controller considers that the queue has cleared and the green time can end (even if there was remaining time from the original plan). Any vehicle that after that time gap is considered to be part of a new platoon that will be serviced in the next cycle. However, the opposite is also true. So long as the long loop detector is occupied, green time is extended to clear the queue. Similarly, if a vehicle has been detected by the far away loop, which is usually placed around 60m from the stop line, a request for extension is placed to include his vehicle within the current green phase. However, similarly to the case with the long loop detectors, after a gap time of three seconds at this loop detector, new vehicles are considered to be part of a new platoon. Of course though, the start time of green and extension of green time must fall within the constraints of the higher level block procedure. This reactive behavior to traffic arrivals and departures is what causes the fluctuations in the countdowns of adaptive controllers.

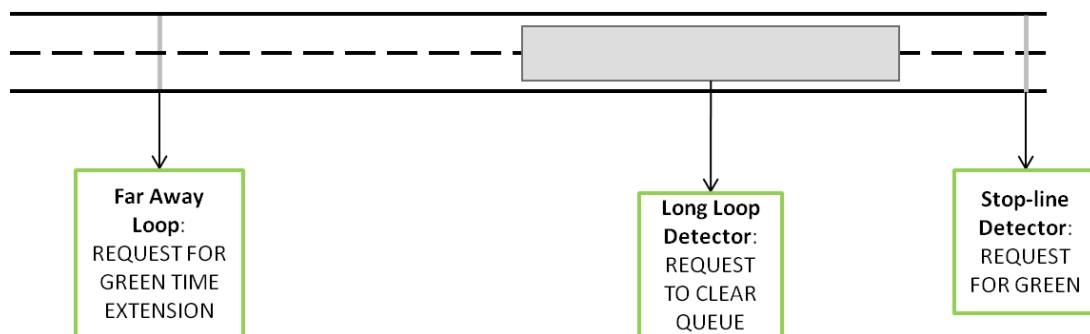


FIGURE 67: DETECTOR GEOMETRY FOR A DUTCH ADAPTIVE CONTROLLER

C. PROJECT DIVISIONS, TASKS AND TOOLS USED

TABLE 20: PLANNED TASKS FOR EACH DIVISION

| Division | Tasks | Resources and Tools |
|----------|---|--|
| 1 | Literature Review | Online Databases |
| 2 | <p><u>1) Building the Prediction Model:</u> A neural network will be constructed, aimed to output lane specific arrival flow profiles for links on urban roads. The selection of neurons and connections will be adapted from literature and online lectures.</p> <p><u>2) Selecting a Case Study:</u> Google Earth is used to find the available signalized intersections. From there the number and location of counting points</p> | <p><u>Building the Model/Data Collection and Processing:</u></p> <p><i>Literature Study:</i> other models using Neural Networks will be referenced. Moreover, literature on different optimization algorithms for training the model, different methods for initializing the weights and biases, and different activation functions are researched</p> <p><i>Online Lectures and Books:</i> through these resources, information on how to properly set up a</p> |

| | | |
|---|--|---|
| | <p>can also be checked by zooming in on the roads. Once an appropriate network is found the search for v-log data begins.</p> <p><u>3) Collecting and Compiling Data:</u> V-log data and configuration files are collected from the Municipality of Delft and the Province of Noord Holland, alongside maps of the Case study intersections detector configurations to be able to identify the detector labels needed for compiling.</p> <p><u>4) Training and Validation:</u> The model will be trained and validated using historical data, for a small network of intersections, collected from the Province of Noord Holland.</p> <p><u>5) Model Testing:</u> A new dataset will be used to test its performance to a dataset it hasn't seen.</p> <p><u>6) Model Evaluation:</u> The key performance indicator for determining the accuracy of this prediction model is how well model outputs for vehicle arrivals match measured arrivals of historical detector counts, as measured by the RMSE, NRMSE and the R-statistic.</p> | <p>neural network is be acquired in order to assure the model is correctly designed.</p> <p><i>Software Packages:</i> MATLAB Neural Networks Toolbox for building and using the model.</p> <p><i>Available Input Data:</i> Only V-log data from loop detectors will be used as input for this prediction model. The reasoning is, probe vehicle data can only output information on a vehicle's own location, speeds and proximity to other vehicles: none of which are useful for either platoon dispersion or turning percentages. GPS coordinates are not accurate enough to identify which lanes probe vehicles are on, and the overall intensity of traffic can be determined from upstream detector counts. Historical trajectory data may have been useful, but, unfortunately, it is not available for this study.</p> <p><u>Training, Testing, Validation and Evaluation:</u></p> <p><i>Historical Data:</i> Acquired from Peter Broekhuijsen, from the Municipality of Delft, and Harm Jan Mostert, from the Province of Noord Holland.</p> <p><i>Software Packages:</i></p> <ul style="list-style-type: none"> - Command Prompt Window for creating batch files to merge data from minute by minute folders to 2 month folders. - CuteView for Data Translation - MATLAB for preprocessing data <p><i>Error Estimator:</i> The extent to which the model outputs deviate from the historical data outputs will be evaluated by calculating the RMSE and R-stat between measured and estimated values.</p> |
| 3 | <p><u>1) Building the Simulation Model:</u> A case study location must be selected for evaluating how well a predictive controller performs in comparison to an actuated or adaptive controller. For this the Intersection between Kruisweg (N201) and Drie Merenweg in Noord Holland was selected, and all three surrounding intersections feeding into it.</p> <p><u>2) Evaluating a Predictive Controller:</u> For a fair comparison:</p> <p>a - the simulation model will be run using adaptive controllers at intersections, and values for travel time delays (i.e. defined KPIs) will be recorded.</p> | <p><u>Building the Model:</u></p> <p><i>Software Packages:</i> The simulation model can be build using the VisSim simulation program, which is used by all Municipalities and Provinces in the Netherlands, and is a strong platform for modeling urban networks and intersections.</p> <p><i>Historical Data:</i> Using historical data, vehicle demand profiles through the simulated case study may be designed to better mimic the actual traffic situation.</p> <p><u>Evaluating the Predictive Controller:</u></p> |

| | | |
|--|--|---|
| | <p>b - the simulation model will be run for predicted flows and the controller's output will be recorded for a certain time period.</p> <p>c - the recorded controller schedule will be fixed into the model. The historical flows will then be inserted into the model once again, but actuated control wont be used (only the control logic outputted earlier using the predicted flows). Traffic flow will be evaluated using the KPI's for this schedule and compared to those from part a.</p> <p>3) <u>Evaluating Prediction Time Horizon</u>: the duration it takes for a controller to give out a stable output is determined. If it is longer than the prediction horizon, then a longer one is needed.</p> | <p><i>Software Package</i>: To measure the KPI's, measurements from the Vissim simulation will be used to reveal the delays of vehicles at intersections. Vissim-COM interface will be used to set up the simulation as needed.</p> |
|--|--|---|