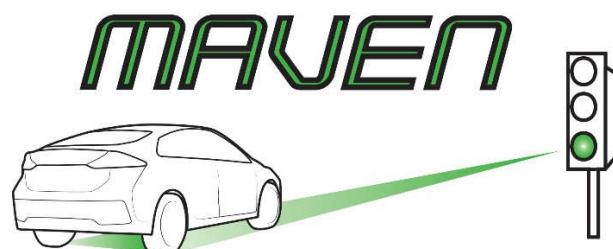


MAVEN

Managing Automated Vehicles Enhances Network



Deliverable nº: 4.4:
Cooperative adaptive traffic light with automated vehicles
(Final version)
Dissemination Level: PU

Version	Date	Release	Approval
1.4	21-12-2018	Robbin Blokpoel (DYN)	Steven Boerma (MAP)



Document Log

Version	Date	Comments	Name and Organisation
0.1	17-12-2017	Initial template and content	Robbin Blokpoel (DYN)
0.2	17-01-2018	Content provision	Jan Přikryl (CTU) Daniel Wesemeyer (DLR) Xiaoyun Zhang (DYN) Aleksander Czechkowski (DYN) Robbin Blokpoel (DYN)
0.3	25-01-2018	Internal review	Martijn Harmenzon (MAP)
1.0	30-01-2018	Final version for submission	Robbin Blokpoel (DYN)
1.1	19-11-2018	Template for updates	Robbin Blokpoel (DYN)
1.2	07-12-2018	Full initial content	Andre Maia Pereira (CTU), Daniel Wesemeyer (DLR), Xiaoyun Zhang (DYN), Robbin Blokpoel (DYN)
1.3	18-12-2018	Internal review	Steven Boerma (MAPtm)
1.4	21-12-2018	Final version	Robbin Blokpoel (DYN), Andre Maia Pereira (CTU)

Authors

Beneficiary nº (1 to 5)	Partner	Country
1	Robbin Blokpoel, Dynniq Xiaoyun Zhang, Dynniq	The Netherlands
4	Jan Přikryl, CTU Andre Maia Pereira, CTU	Czech Republic
2	Daniel Wesemeyer, DLR	Germany
3	Leonhard Lücken, DLR	Germany
9	Martijn Harmenzon, MAP Steven Boerma, MAP	The Netherlands



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Executive Summary

The scope of MAVEN focuses on the urban environment and specifically at signalized intersections and signalized corridors. Within this scope, WP4 deals with elements of C-ITS such as Green Light Optimal Speed Advisory (GLOSA), adaptive traffic light control algorithms, routing, queue algorithms and green waves. These are developed for organising the flow of infrastructure-assisted automated vehicles. Furthermore, the development of a Local Dynamic Map (LDM) is emphasized here due to its core property of infrastructure implementation.

This deliverable presents the joint work on road automation that has been done by consortium partners. To give a brief description, four types of traffic control methodologies are described in order to give an overview of them regarding the predictability of the traffic control algorithm. The tools and methodology that are relevant to WP 4 are also described. The most relevant tool is simpla and it captures the characteristic properties of automated vehicles in comparison with conventional vehicles by an appropriate parametrization of vehicle models provided by the simulation software. Queue modelling research is the first key output described in the document. Then two GLOSA implementations are discussed, one based on actuated traffic control and another on a plan stabilization algorithm for adaptive control. Both are extended for multiple intersection, for actuated combined with GLOSA, for adaptive with GLOSA-based green waves. This was based on a new green wave design methodology for static control.

The document also contains preliminary results of the queue modelling, traffic control stabilization and green wave. The queue length estimation shows large improvements when information from automated vehicles is fused into the model. Up to 40% reduction for the average error was shown. The largest benefits were visible for high traffic volumes, because this increases the chance of receiving vehicle information at 20% penetration rate. For adaptive control stabilization, results showed 25% reduction in average prediction error, while maintaining similar traffic efficiency. More advanced parameters were added to combat specific side effects, like the prediction stagnating at a certain value due to an extension. This resulted in a small further improvement, but most notably in a solution for the stagnation problem. The green wave method showed up to 56% reduction in stops for vehicles on the route.

For local level routing an extensive description of the route advice system was given. This is based on a distributed architecture that can operate both locally in different agents like road side units, or in a central point. The routing system showed promising initial results. For special road user categories the deliverable presents a study showing the benefits of plan stabilization for VRUs. Mitigation measures are presented for the potential disruption of traffic for unmanned logistics vehicles after an analysis of the current state-of-the-art of this category of road users. Guidelines for prioritizing automated vehicle platoons have also been presented, which were the result of MAVEN workshops held during the course of the project. Lastly, an appendix contains paper reviews for a thorough analysis of the state-of-the-art.



Table of Contents

Executive Summary	3
1 Introduction	11
1.1 Structure of this document.....	12
2 Traffic control methodologies	13
2.1 Static control	13
2.2 Actuated control	13
2.3 Semi-fixed time control.....	14
2.4 Adaptive control	15
3 Tools and methodology.....	16
3.1 Simpla.....	16
3.2 Local Dynamic Map.....	17
3.3 Position simulation	19
4 Queue modelling.....	20
4.1 Meaning of queue length.....	21
4.2 Model structure	21
4.3 Lane change model.....	21
4.4 Estimation without using information from probe vehicles.....	22
4.5 Occupancy correction.....	23
4.6 Remarks.....	23
4.7 Troubles	23
4.8 Queue length estimation using information from probe vehicles	24
4.9 Remarks.....	24
4.10 Preliminary results.....	24
5 Local level routing	30
5.1 Traffic Network	34
5.1.1 Static Data	35
5.1.2 Dynamic Data.....	36
5.1.3 List of Required Data.....	38
5.2 Vehicle Generation.....	39
5.3 Trip Definition	41
5.3.1 Definition of Edge, Junction Vehicle Lane, Connection and Movement Group	42
5.3.2 Estimate the Departure of Discharging Vehicles.....	43
5.3.3 Update Connections' Capacity and Delay	47
5.4 Estimation of Travel Times	48
5.4.1 Travel Times Using Probe Vehicles.....	48
5.4.2 Travel Times Using Queueing Delay Model	49
5.5 Traffic Controllers Knowledge Sharing	50
5.6 Conclusion and further research.....	52
6 Actuated traffic control with green light optimal speed advisory.....	53
6.1 AGLOSA on a single intersection	53



6.1.1	C2X interfaces.....	54
6.1.2	Queue length estimation	56
6.1.3	Results.....	57
6.2	Multi-Intersection optimization AGLOSA.....	59
7	Plan stabilization for adaptive control	62
7.1	Algorithm design	62
7.2	Simulation methodology	65
7.3	Results.....	70
8	Green waves for automated vehicles	75
8.1	Introduction	75
8.2	Q-learning approach.....	77
8.2.1	Background.....	77
8.2.2	Results.....	77
8.3	Adaptive green wave with speed advice.....	80
8.3.1	Theory.....	80
8.3.2	Simulation network.....	81
8.3.3	Results.....	84
9	Special road user categories.....	87
9.1	Prioritized public transport.....	87
9.2	Vulnerable road users	87
Results	89
9.3	Prioritization management of automated vehicle platoons	92
9.4	Unmanned logistics	92
9.4.1	Historical contest developments.....	93
9.4.2	Current developments	94
9.4.3	Impact on urban network	95
9.4.4	Unmanned Logistics and MAVEN	96
9.4.5	Simulation approach.....	97
10	Conclusion and further research	99
10.1	Conclusion	99
10.2	Further research.....	101

List of Figures

Figure 1-1:	WP4 work in the MAVEN context	11
Figure 2-1:	Actuated control dynamics	13
Figure 2-2:	Semi-fixed time control dynamics	14
Figure 2-3:	Queue and arrival flow modelling in adaptive control.....	15
Figure 3-1:	LDM signal data message design.....	18
Figure 3-2:	LDM detection data message design.....	19
Figure 4-1:	Block diagram of the MAVEN queue length model structure	21
Figure 4-2:	Intersection approach overview	25
Figure 4-3:	Prediction (red) and real (blue) values of vehicle counts at stop-bar detectors	25
Figure 4-4:	Prediction (red) and real (blue) values of vehicle counts at stop-bar detectors	26



Figure 4-5: Queue length estimation without correction from CAVs at two-lane approach. Note the significant error of the model on the left lane, which is due to integrating nature of vehicle conservation law.....	27
Figure 4-6: The same approach as in Figure 4-5, with queue model that incorporated the information from CAVs. Simulated CAV penetration rate 20%, flow approximately 1200 veh/hr.....	28
Figure 4-7: The same two lanes again, 20% penetration rate, incoming flow approximately 600 veh/hr.....	29
Figure 5-1: Communication among CAVs and Traffic Controllers (TCs) in a Traffic Network (TN).....	30
Figure 5-2 Component UML diagram containing the proposed Local Level Routing sub-systems.....	32
Figure 5-3 Example of some information required for the Network Representation sub-system.....	35
Figure 5-4 Division of a sliding prediction horizon time window into shorter intervals.....	37
Figure 5-5 Division of movement group's, J1MG1 , dynamic data by its green signal plan.	37
Figure 5-6 Composition of the Vehicles on Lane edge instance.	38
Figure 5-7 Different types of arrivals on lane. Fixed and dynamic inflow are responsibility of Vehicle Generation sub-system, while discharged from upstream is done by the Trip Generation sub-system.	39
Figure 5-8 Example of selecting vehicles to discharge for green phase ph0 . Vehicles already on lane will be discharged, as well as the earliest one arriving. The last arriving will not manage to arrive before the end of the green phase at $t = 20$	43
Figure 5-9 Key events to estimate vehicle state in 4 cases.....	44
Figure 5-10: Format of exchanged messages.....	51
Figure 6-1: GLOSA speed advice zone concept	54
Figure 6-2: GLOSA working principle	55
Figure 6-3: Queue length estimation principle, Source: [26].....	56
Figure 6-4: Basic scenario intersection	57
Figure 6-5: Results for the basic intersection scenario (red: fixed signal plan, green: AGLOSA). Left: average time loss, right: average CO ₂ emissions	58
Figure 6-6: Scenario 2 intersection	58
Figure 6-7: Results for scenario 2 (red: fixed signal plan, green: AGLOSA). Left: average time loss, right: average CO ₂ emissions	59
Figure 6-8: Multi-intersection scenario network.....	60
Figure 6-9: Results for the multi-intersection scenario (red: fixed signal plan, green: AGLOSA, blue: AGLOSA with coordination factor). Left: average time loss, right: average CO ₂ emissions.....	61
Figure 6-10: Time losses for the different traffic directions (green: main flow direction, blue: minor roads)	61
Figure 7-1: Stabilized adaptive control dynamics	64
Figure 7-2: Case study of Helmond (left) and the simulation network in SUMO (right).....	65
Figure 7-3: Helmond Intersection 701, Hortsedijk/ Europaweg	66
Figure 7-4: Results of scenarios with diff c, given no α , β and EL	71
Figure 7-5: Results of scenarios with diff α , $c=60$, no β and EL	72
Figure 7-6: An example of small consecutive jumps, “flat line” situation	72
Figure 7-7: Results of scenarios with $c=60$, diff α , β and EL	73
Figure 8-1: Ideal bi-directional green wave	75
Figure 8-2: Artificial network used for evaluation	78
Figure 8-3: Offset creation with lagging/leading left turns	80
Figure 8-4: Combination of lagging/eading left with speed advice	80
Figure 8-5: Splitting platoon with speed advice	81
Figure 9-1: XCycle dynamic sign with capability to inform about active bus priority	87
Figure 9-2: Six consecutive intersections of the case study in Helmond (left) and the corresponded simulation network in SUMO (right).....	88



Figure 9-3: From west to east, schematics of six consecutive intersections road layout, showing only the bicycle lanes (red) and corresponding signal controls	89
Figure 9-4: Relation of FOM_unified to different weight in scenario 0-5	90
Figure 9-5: Relation of FOM_unified with different weight in scenario0-5 for EL=0 and 1	91
Figure 9-6: Left the MailRail, right the AVG used at ECT(TN).....	93
Figure 9-7: Left the VAL Lillie, right the first autonomous Tram driving in Potsdam	93
Figure 9-8: Left to Right, Volvo in car delivery, PicNic small busses and the delivery robot of Domino's	94
Figure 9-9: Left to Right,Sartup Embark, DHL start testing	94
Figure 9-10: Left to Right,2getthere, Easymile, Renault, conceptcar VW	95
Figure 9-11: Left above to Right below, Starship, post, Marble, Nuro, Postmates, Telerital andDispatch	95

List of Tables

Table 4-1: Results for queue length estimation at high traffic volume	28
Table 4-2: Queue estimation results for low traffic volume.....	29
Table 5-1: Selected attributes for a vehicle, i , on its j^{th} junction/edge/lane of its route.....	41
Table 6-1: Hourly traffic demands for the basic intersection scenario	57
Table 6-2: Hourly traffic demands for scenario 2	58
Table 6-3: Hourly traffic demand for the multi-intersection scenario.....	60
Table 7-1: Cost incurred on the stabilized signal group 8, using cost function 7-2	63
Table 7-2: Simulation scenarios list	68
Table 7-3 Data of scenario 0 to 11, $c=0\sim 600$	70
Table 8-1: Results of Q-learning versus dynamic programming.....	78
Table 9-1: Simulation scenario designs overview	89



Abbreviations and definitions

Abbreviation / Term	Definition
Acceptance criteria	The criteria that a product must meet to successfully complete a test phase or meet delivery requirements.
Acceptance test	Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the acquirer to determine whether to accept the system or not.
Adaptive	The network optimizer looks forward to minimize cost (queue, stops, wait time, queue spillback, etc.) based on the configured policies. The signal group is activated at the start of the current stage (containing the signal group) when a demand is set for the signal group. The signal group is activated in a planned stage (containing the signal group) when an implied demand is set for the signal group.
Adaptive Unconditional	Same as traffic adaptive, with the differences that the signal group is activated unconditionally at the start of a stage (containing the signal group). This detection type should be configured for coordinated signal groups to allow the optimizer to begin changing the stage in anticipation of the demand.
ADAS	Advanced Driver Assistance Systems
Architecture	The organisational structure of a system, identifying its components, their interfaces, and a concept of execution amongst them.
AV	Automated Vehicle
C-ITS	Cooperative Intelligent Transport Systems
CAM	Cooperative Awareness Message
CAV	Cooperative Automated Vehicles
CEN/ISO	European Committee for Standardization / International Organization for Standardization
Conflict	Two traffic streams are conflicting and traffic will collide if one does not yield for the other, generally this situation is prevented by traffic lights.
CPU	Central processing unit
CV	Connected Vehicle
Design	Those characteristics of a system or components that are selected by the developer in response to the requirements.
Dynamic green wave	Green wave that is not based on fixed cycle times and offsets, but based on current traffic demand.
EC	European Commission
EPA	Environmental Protection Agency
ETSI	European Telecommunications Standards Institute
FCD	Floating Car Data
Figure of merit (FOM)	A figure of merit is a quantity used to characterize the performance of a device, system or method, relative to its alternatives. Two types of FOM: un-unified and unified are proposed in this research, as shown in section 7.2.
GLOSA	Green Light Optimised Speed Advisory
GPS	Global Positioning System
HAD map	Highly Automated Driving map
HGV	Heavy Goods Vehicles
HMI	Human Machine Interface
HW	(Hardware) Articles made of material, such as cabinets, tools, computers, vehicles,



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

	fittings, and their components [mechanical, electrical and electronic]. Computer software and technical documentation are excluded.
I2C	Infrastructure-to-car
I2V	Infrastructure-to-vehicle
ICT	Information and Communication Technologies
ID	Identifier
ImFlow Configurator	The ImFlow Configurator is a standalone application used to configure and simulate an ImFlow system. The ImFlow Configurator is designed to be used by Traffic Engineers.
KAR	Korte Afstands Radio (Dutch for Short range radio), an older technology to request priority at an intersection. It generally has more interference and much less bandwidth than 802.11p.
KPI	Key Performance Index
LDM	Local Dynamic Map
MAVEN	Managing Automated Vehicles Enhances Network
Measure of effectiveness (MOE)	Measures of Effectiveness (MOE) are measure designed to correspond to accomplishment of mission objectives and achievement of desired results.
Movement group	Generic version of a signal group that also covers uncontrolled intersections. In that case a movement group are all connections that must yield to the same set of connections.
MPC	Model Predictive Control
MPR	Market Penetration Rate
OBU	On-board Unit
Permissive green	In this case a signal group has multiple movements (e.g. straight and right turn) of which one conflicts with another signal group. Common permissive greens that have green light at the same time are VRUs together with straight/right turn of vehicles and straight/left turn of opposing vehicle signal groups.
PMT	Project Management Team
R-ITS-S	Roadside ITS Stations or roadside unite (RSU) as part of the Cooperative Intersection
ReMP	Requirements Management Plan
RSU	Road Side Unit
Saturation flow	The maximum flow in vehicles per hour that can be achieved. Often used in queue modelling to predict the speed at which a queue discharges during green after 6 seconds of acceleration/ reaction time.
Signal group	Set of signal heads that are always green at the same time. There can be multiple lanes in one signal group, for example 2 left turn lanes will be in the same signal group, but also two directions can be in one signal group when they are on the same lane, like one lane that allows both right, straight and left turns.
Simpla	SUMO extension to simulate automated vehicles.
Specification	A document that describes the essential technical requirements for items, materials or services including the procedures for determining whether the requirements have been met or not.
Spillback	Phenomenon that a queue is long enough to block an upstream intersection. Effectively this means switching the light to green upstream will not result in any flow due to the intersection being blocked by waiting vehicles.
SRS	System Requirements Specification
Stage	Set of signal groups that are usually green at the same time. Most traffic light controllers optimize stage-based, but small differences in timing between signal groups can occur due to constraints of the conflict matrix. When one signal group



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

	has no more traffic, some control strategies allow alternatives for it during a stage.
Stakeholders	The people for whom the system is being built, as well as anyone who will manage, develop, operate, maintain, use, benefit from, or otherwise be affected by the system.
SUMO	Simulation software (Simulation of Urban Mobility)
SW	(Software). Computer programmes and computer databases.
TC	Traffic Controller, this is a more generic version of the traffic light controller that can also operate on a non-signalized intersection. It gathers data about the junction and the edges connected to it.
TLC	Traffic light controller
TMC	Traffic Management Centre
Traceability	Ability to trace the history, application or location of that which is under consideration.
V2I	Vehicle-to-infrastructure
V2V	Vehicle-to-vehicle
V2X	Vehicle-to-anything
VA	Vehicle Actuated. Also refer to 'Vehicle Actuated Control'.
VECOM	COMpact VEhicle identification and priority, based on transponders communication through inductive loops.
Vehicle Actuated Control	Virtual detectors are detectors used when physical detectors are not available (in the field). Data from virtual detectors is estimated by the system based on the measured flow from upstream or downstream physical detectors. This data is used to estimate the intersection exit of entry flows based on the applicable traffic movements.
VRU	Vulnerable road user
WP	Work Package



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

1 Introduction

This deliverable (D4.4) contains the results of work package 4 (WP4), road automation. It describes the scheduling and signal timing strategy for Traffic Light Controller (TLC) optimization (task 4.1) as well as Floating Car Data (FCD)-based lane dependent queue estimation (task 4.2). This report is the final version deliverable of the work package on road automation and extends the initial version (D4.1). This version also includes results from tasks 4.3, Multi-intersection optimization and local level routing, and task 4.4, which is about the inclusion of special road user categories and vulnerable road users.

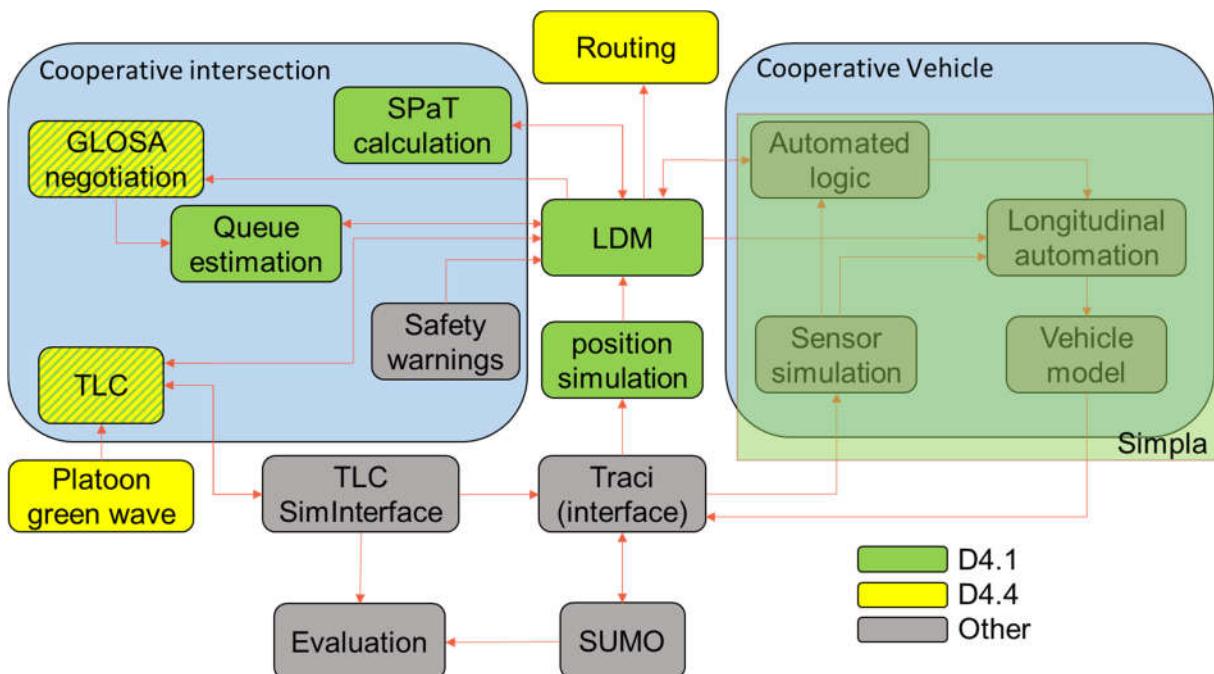


Figure 1-1: WP4 work in the MAVEN context

Figure 1-1 presents the work in this deliverable in the context of the MAVEN simulation architecture from deliverable D2.2 [1]. The colours indicate which part of the work is in the current deliverable and which is in other work packages. It should be noted that D4.2 and D4.3 are demonstrators of the work in this report and demonstrate all green and yellow blocks, except routing, green wave and position simulation.

The components in green were completed and reported in deliverable D4.1. It should be noted that the Local Dynamic Map (LDM) considered here is only for the infrastructure. This enables simulation of vehicles as well, but in WP3, a different vehicle LDM will be implemented.

The components in yellow are new in D4.4. The routing and green wave development work is new in this deliverable. The GLOSA negotiation use case and the TLC algorithms are further extended for multi-intersection AGLOSA. Therefore, they are yellow/green dashed. It should be noted that the communication perspective of negotiation is described in D5.1.

The new research on special road user categories affect all components of the infrastructure and provide policy guidelines or investigate the impact of a certain use case on another user category.

Several components (grey) are not covered by this deliverable. First of all, the evaluation element is based on an implementation from the COLOMBO project, of which details can be



found in the COLOMBO evaluation paper [2]. The evaluation methodology will only be described in this report where it is relevant. Similarly, the software required for interfacing the traffic light controller with the Simulation for Urban MObility (SUMO) software, was also developed in the COLOMBO project and is published in the COLOMBO simulation environment paper [3]. Second of all, the safety warnings element was modelled in WP3 and will be tested in the field using results from WP5, enabling technologies. Since there will be no SUMO simulations or direct algorithm development related to the traffic light controller with regards to safety, this element is not covered in this deliverable. Lastly, the vehicle aspect is covered by WP3, vehicle automation. However, a model is required for simulations using the SUMO software. The simpla extension implements this and is completed for MAVEN.

1.1 Structure of this document

The document continues with Section 2, the background information of contemporary traffic control methodologies. Section 3 describes the methods and tools required for the research. These are simpla, the LDM and the position simulation. Queue modelling is discussed in a separate section, Section 4, as this is a key research topic in MAVEN. It forms the basis for accurate GLOSA and efficient adaptive traffic control. The local level routing is described in Section 5. Two of the traffic control methodologies, actuated and adaptive control, are used to implement the GLOSA use case and will be described in Section 6 and Section 7 respectively. The new work on the green wave is in Section 8. The inclusion of special road user categories into the framework is analysed in Section 9. Section 10 gives the conclusion and the further research is presented. Lastly, an appendix includes extensive literature review to give an overview of the state-of-the-art.



2 Traffic control methodologies

This section gives an overview of the most common control methodologies used. Since one of the main use cases in MAVEN is GLOSA, special attention is given to the predictability of the traffic control algorithm.

2.1 Static control

The simplest form of traffic control is static or fixed-time control. Even though little intelligence is required inside the controller cabinet nor any investment for sensor technology, the maintenance costs can still be high. This is due to the manual calibration effort required to keep the plans effective. Formulae and software tools [4] are available to calculate these plans, but for every significant change in traffic demand, the procedure has to be repeated.

The plans are calculated based on average flow and include a margin to cope with cycle-by-cycle demand fluctuations and prevent queues from forming. Most of the time these margins are unnecessary and increase the delay time for all other traffic. When average demand fluctuates by time of the day, multiple static programs are often loaded, which are switched on at predefined times of the day.

Day-to-day differences can still cause unnecessary queues and System Activated Plan Selection (SAPS) is often used to cope with this. For this system, a few sensors are placed at strategic locations in the network to detect congestion and/or measure traffic volume. The system dynamically decides when to switch between several pre-configured plans.

Irrespective of the amount of static plans and the plan selection method, the dynamics for GLOSA are the same. The control strategy is perfectly predictable, but has a risk of forming congestion, which impedes the efficiency of a speed advice.

2.2 Actuated control

Actuated control is based on sensors detecting whether traffic is present or not. Typically, two functions for detection are used: stop line detection and gap detection. Stop line detection checks if there is any demand at a signal group. If there is no traffic in all signal groups of a stage, this stage will be skipped. Gap detection is used for extension of green light beyond the minimum duration. This means as long as there is demand, the green duration will be extended until the maximum green time is reached. This is illustrated in Figure 2-1. The solid green rectangles represent the minimum green time and the hatched rectangles the optional time available for extension.

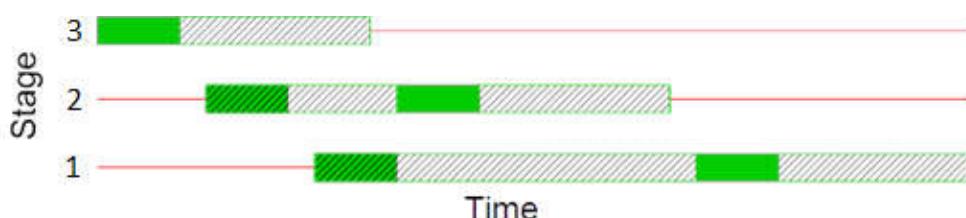


Figure 2-1: Actuated control dynamics

Investment costs of actuated control are higher due to the required sensors. Apart from added sensor maintenance, the calculation of the signal plans requires much less updating. The traffic engineer sets the minimum green time based on safety requirements, since in general drivers do not expect very short green durations. The maximum green time is based



on the maximum desired cycle time. This may require rebalancing when traffic demand changes considerably.

The plan stability is very low as can be seen in Figure 2-1. The plan in the example has a minimum green duration of 6 seconds and a maximum of 20. This means that there is 14 seconds uncertainty when the next stage starts and 28 seconds for the start of the third stage as is indicated by the increasing hatched areas. Providing speed advice based on this data will be difficult.

2.3 Semi-fixed time control

Most commonly used for contemporary GLOSA solutions are semi-fixed time control strategies, like for ODYSA [5]. These are based on a fixed time control plan, but the switching moment can occur between a configured minimum and maximum time. This is illustrated in Figure 2-2, which shows the guaranteed green with solid green rectangles, the default configured green duration is indicated as solid green with hatched light green on top of it. The maximum allowed flexibility is the total hatched box. A default green time of 20 seconds is used, while both at the beginning of a stage and at the end there is a flexibility of (-3, 3) seconds. Meaning the absolute minimum and maximum green times are 14 seconds and 26 seconds respectively.

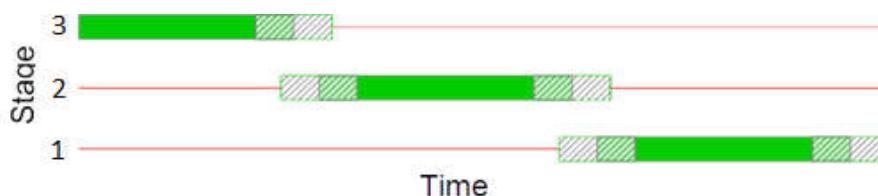


Figure 2-2: Semi-fixed time control dynamics

Important for the stability is that there is a fixed cycle time. This means the flexibility is not cumulative, i.e. if the first stage is extended up to $t = 23$ seconds, the second cannot reach the maximum green time of 26 seconds anymore. It would have to extend to the maximum switching moment to reach the default green time of 20 seconds. This also shows a weakness of this method in congested situations. If the first stage is slightly congested it will use up all flexibility, even if the second stage is heavily congested.

Despite the constrained flexibility, the plan stability is still problematic, due to the moment the decision is taken. Until the previous stage enters the (light green) hatched area, there is still 6 seconds uncertainty for the switching moment. Only once the switch is initiated, there is certainty. The amber time is left out of the figures to keep them easy to understand, which is typically 3 seconds. This is followed by typically 2 seconds of all-red clearance time before the next phase can start. Therefore, until 5 seconds before the start of green there is a 6-second uncertainty for the moment of switching.

The original GLOSA application for this system was developed to display speed advice on a static panel at approximately 500 m upstream of the intersection. The speed advice was intended to be used until approximately 100 m before the stop line at a point where the driver starts to slow down, anticipating on a slightly delayed start of green. For connected and automated vehicles, the speed advice potential is much bigger as they can receive continuous updates of the speed advice. When continuous updates are applied to this semi-fixed time control strategy, it would imply large changes in speed advice are possible. A sudden decrease of time to green prediction from 8 to 5 seconds (green starts early) or a freeze for 3 seconds when reaching 5 seconds to green are both likely with the previously discussed flexibility of (-3, 3) seconds. This can lead to a jump in the speed advice of up to 60%.



2.4 Adaptive control

Adaptive control is based on a model of the approaches towards the intersection. As an adaptive control example, in Figure 2-3, a schematic view of a queue and arrival model is shown. Vehicles enter the model when they are detected by the entry detector. The x-axis represents the distance to the stop line in travel time. In this example the historical travel time from the entry detector for queue 1 (Q1) is 15 seconds and therefore the vector reaches up to $t = 15$. Every second, vehicles in the arrival pattern are moved one field closer to the stop line, which is indicated by the "0" column. The queues accumulate at the stop line and discharge with counts from the stop line detector. Going back to the example of Q1, after one second, the 0 in cell "1" gets added to the 2 of cell "0", resulting in 2. The 1 of cell "2" shifts to cell "1", and so on.

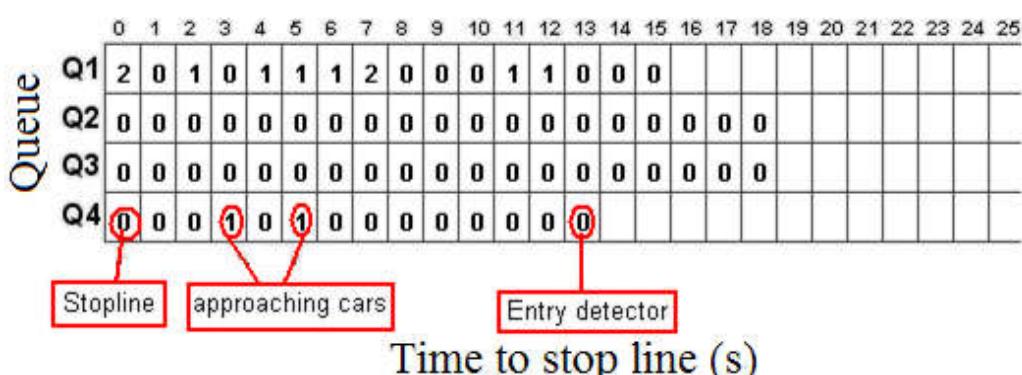


Figure 2-3: Queue and arrival flow modelling in adaptive control

In work package 4, the adaptive control algorithm ImFlow [6] is used. ImFlow uses the model of approaching and waiting vehicles to evaluate different possible control solutions. They are evaluated using a cost function that minimizes delay and stops for all traffic approaching the intersection. Calibration effort for this control method is minimal, since the algorithm optimizes the green duration by itself. Precise configuration of safety timings and detector location is required. Maintenance costs are minimal except for sensor maintenance. Throughput and delay for this control method are optimal, since every cycle can be precisely adjusted to cycle-by-cycle demand fluctuations. In case of congestion, the model knows which stages are most congested or could even cause spillback to other intersections and allocate most green time there while respecting a maximum waiting time. This in contrast to semi-fixed time control, which allocates the spare time according to a first-come-first-serve principle and actuated control, which has a pre-configured amount of extra time for each stage.

In theory, the predictability could be as low as for actuated control. However, with the modelling of the approaching vehicles, the control algorithm already knows beforehand how much a certain phase will be extended beyond the safety minimum. Disrupting factors can be detection errors, signal groups without entry detection (e.g. a pedestrian or bicycle approach with only a push button), signal priority and pre-emption calls.



3 Tools and methodology

This section gives an overview of the tools and methodology required for the work on the infrastructure in MAVEN. There are subsections for *simpla*, which is an extension to SUMO for automated driving, the LDM, which is a local dynamic map holding all relevant information for the use cases and the position accuracy simulation required for realistic position information from simulated vehicles.

3.1 Simpla

A promising possibility to enhance the future traffic efficiency is the formation of platoons by automated vehicles. In this work, we regard a platoon as a group of automated vehicles following each other with a reduced time headway and possibly employing additional control schemes to maintain a coherent state within the group. Although and because the jurisdictional details for such an operation are still to be clarified, it is highly relevant to study the expectable effects on city traffic. MAVEN focusses these studies on (signalized) junctions, where the greatest effect may be expected from a compactification of traversing traffic flows.

To evaluate the impact of automated vehicles, and especially platooning, on city traffic, MAVEN employs traffic simulations with the program *SUMO* (Simulation of Urban MObility), which is a microscopic traffic flow simulator. The term ‘microscopic’ refers to the fact that the simulation represents each vehicle as an individual entity.

As no platooning functionality was included in *SUMO* at project start, MAVEN developed a lightweight, highly configurable plugin called *simpla* (SIMple PLAtooning), which controls the dynamical adaptation of operational regimes for defined groups of vehicles. Such operational regimes correspond to the vehicles’ control modes which can be either the solitary mode or one of several platoon modes. The full list of configurable modes is:

- Solitary (mode when not driving in a platoon);
- Platoon leader (vehicle driving at the front of a platoon);
- Platoon Follower (vehicle driving within a platoon, but not at the front);
- Catch-up leader (vehicle which committed to join a platoon further downstream and is either driving solitary or at the front of a platoon); and
- Catch-up follower (vehicle within a platoon whose leader travels in catch-up mode).

Technically, this is realized by specifying different ‘vehicle types’ in *SUMO*, i.e., parametrizations of the vehicle properties including its dynamics, corresponding to the different operational modes, and switching between them based on the current traffic situation and additional platooning parameters provided by the user.

We call a vehicle with an original vehicle type for which the user specified a mapping to the platooning types, a Platooning-vehicle (P-vehicle). Only these vehicles will be able to go in platooning modes. For P-vehicles *simpla* assures that they adhere to the following platooning rules:

1. The P-vehicle switches to the appropriate platoon mode if it follows (or is followed by) another P-vehicle on the same lane within a configurable ‘platooning range’.
2. A P-vehicle, which travels solitarily or as platoon leader switches to the catch-up leader mode if another P-vehicle is within a configurable downstream range (larger than the platooning range) on the same lane. For this it is not required that a direct following situation is encountered.



3. A P-vehicle in a platoon whose leader switches to the catch-up mode switches to catch-up follower mode.
4. A P-vehicle leaves the platooning mode if condition 1.) for forming a platoon is violated for more than a configurable amount of time.
5. Before switching from one operational regime to another a safety check is performed, which ensures that the vehicle may safely operate without exceeding the desired maximal deceleration specified for the target vehicle type. If the switch is not successful, the vehicle slowly reduces its speed (with a configurable rate) to establish a situation where it may safely execute the switching.

For more details, see the extended documentation, which was integrated into the *SUMO*-documentation: <http://sumo.dlr.de/wiki/Simpla>

simpla does not pose strong requirements on the vehicle types mapped by user (although it does require that they have the same length for instance). However, there are some guidelines for the parametrization of the platooning models, which we follow in MAVEN:

1. Parameters for human imperfection (parameter `sigma=0`) are disabled for all modes.
2. The time headway (parameter `tau`) of the platoon follower and the catch-up follower modes are reduced.
3. The values for the desired speed (parameter `speedfactor`) of the catch-up modes are slightly increased.
4. The lane-change behaviour of the follower modes is set to ‘strategic only’. This means that the platoon may only be left for strategic reasons (i.e., if the route of the follower deviates from the route of the leader) and not due to cooperative or tactical considerations.

3.2 Local Dynamic Map

The Local Dynamic Map (LDM) is at the centre of most C-ITS systems. It is a database containing information about vehicles, traffic light status and other traffic relevant information. The key advantage of the LDM architecture is that the data is stored based on map references. A possible query would be to ask all vehicles approaching a certain signal group. This is opposed to geo-databases, where a user would first have to calculate the geographical area of the approach to the signal group and then filter which vehicles are heading towards the intersection and which are actually on the egress lanes. As explained in deliverable D2.2 in the architecture, the LDM connects many components in MAVEN. Apart from the WP5 specific content, work undertaken on the LDM has been finalized.

Two messages are defined to which a MAVEN component can subscribe to. The first is the LDM signal data message, shown in Figure 3-1.



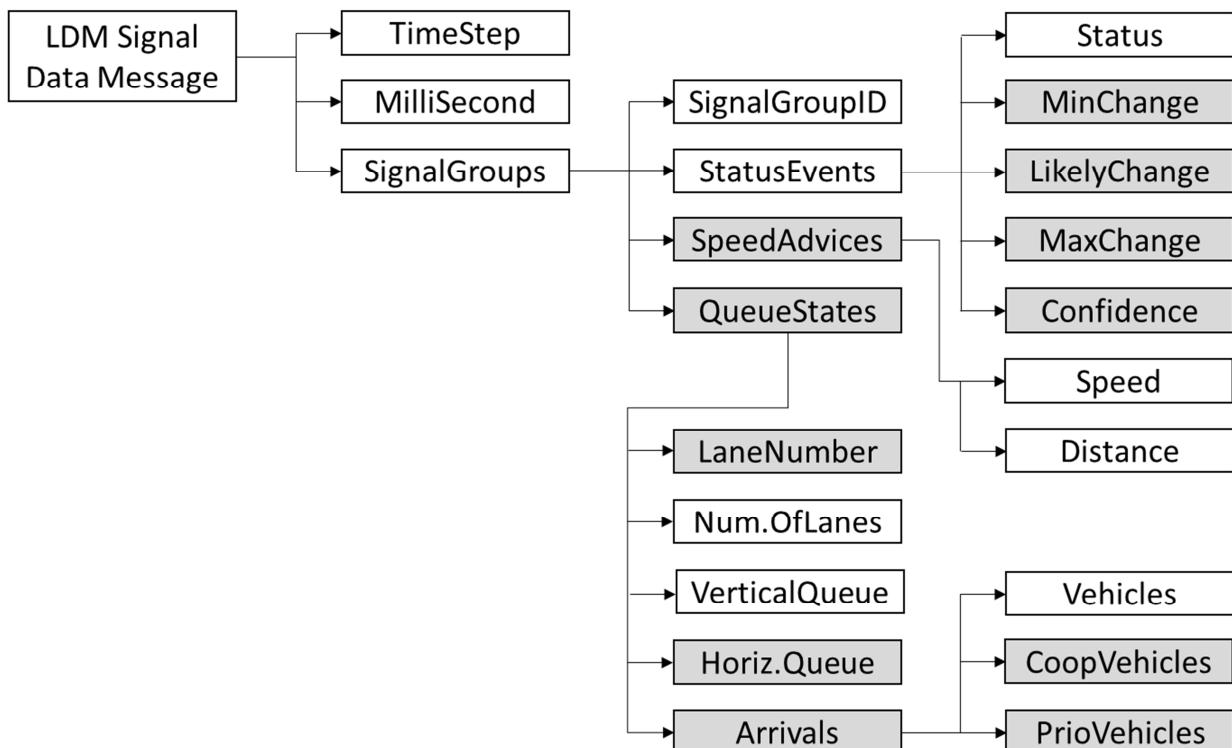


Figure 3-1: LDM signal data message design

The LDM signal data message is built up following the ASN.1 encoding. Unlike V2X messages, which use ASN.1 UPER (Unaligned Packet Encoding Rules), these messages are PER (Packet Encoding Rules). The main difference is that UPER is slightly more compressed, but much slower due to the many bit shifting operations needed for encoding and decoding.

The Figure 3-1 indicates mandatory elements as white and optional as grey. It starts with mandatory timing information followed by a compound object of the signal groups. A compound element consists of several sub-elements and there can also be more than one object. For each signal group, the ID and the status is mandatory. Speed Advice is optional, because it is not always available (for instance when there is no demand on a signal group there is no prediction and thus no speed advice). The queue states are optional as well, but always provided in case of ImFlow controllers for MAVEN. Within the status there are many options to indicate predictions for the next change, these are optional because predictions are not always available. For the queue length there is a distinction between horizontal – the actual length in meters – and vertical queue length, which is the number of vehicles waiting. Arrivals are also always provided in the MAVEN case, but can distinguish between equipped cooperative (automated) vehicles and vehicles with priority. The value in “vehicles” is cumulative. For example, if there are three vehicles approaching, of which one is unequipped, one is unequipped, but has priority and one cooperative automated vehicle (CAV), then the values would be: vehicles = 3, CoopVehicles = 1 and PrioVehicles = 1.



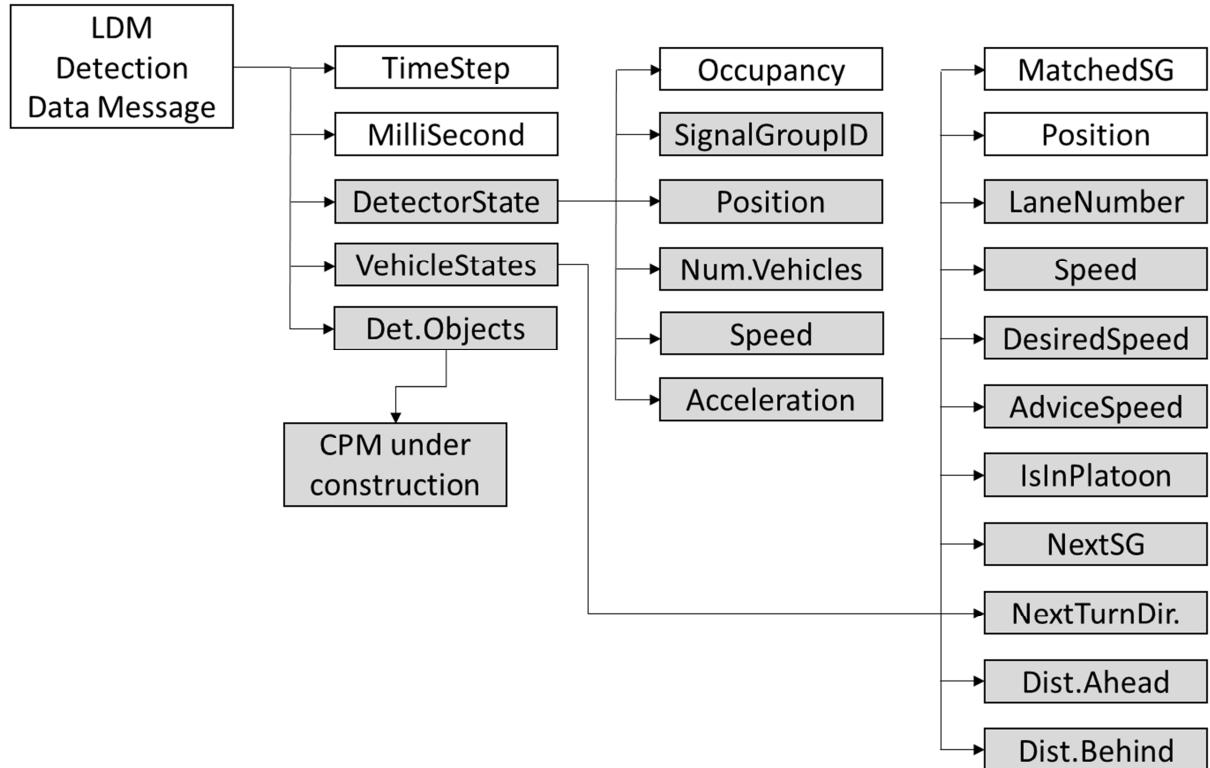


Figure 3-2: LDM detection data message design

The detection data message is shown in Figure 3-2, the figure is built the same way as for the signal data message. Traditional detectors used for presence are included in “DetectorState” objects. The “DetectedObjects” element, on the other hand, is based on the Collaborative Perception Message (CPM), which is part of WP5 and will be merged into the LDM later. However, this is based on object tracking sensor data for safety and therefore not applicable for traffic control simulations.

3.3 Position simulation

The position of a vehicle measured by a Global Positioning System (GPS) sensor in real world scenarios is not always quite precise. On one side, it is influenced by cost factors, forcing the commercially available units to use only one GPS frequency, on the other side the urban scenario that we consider poses a difficult challenge for any GPS system as the urban corridors have only a limited view of the sky. In such a situation the GPS signal is blocked by buildings that surround the corridor and the precision of the position drops significantly [10].

In our case we simulate automated vehicles within a microsimulation framework. Every such simulated vehicle can provide the user with its exact (up to machine precision) position coordinates. In order to simulate the real world conditions, we use additive Gaussian noise to distort the original data,

$$(x_r, y_r) = (x_{sim}, y_{sim}) + N(0, \Sigma) \quad 3-1$$

where (x_{sim}, y_{sim}) is the “true” simulated position of the vehicle, (x_r, y_r) is the approximate real world position of the vehicle, N denotes the normal Gaussian multivariate distribution, and Σ is the covariance of the distribution parameters, which is a heuristic 2×2 covariance matrix in this case. The resulting positions will be used as simulated positions in the LDM as if they came from a CAM message. Speed and other information is later derived from the position.



4 Queue modelling

Queue length has been regarded as one of the key parameters in the process of signal plan design, as estimates of queue length may be used as a part of a criterion that is minimised by systems that provide coordinated control of signalised intersections. Numerous studies discuss the problem of queue development and its influence on travel delay, for example [12][17][22][25].

Typical queueing models from the first group are defined in [7][8][16]. These models are the first principle models, meaning that they are derived from underlying physical principles of the queue formation and dissipation process with some ad-hoc corrections accounting for the stochastic nature of the queuing process. In cases where the installed equipment allows for dense or event-driven sampling of detectors, more elaborate deterministic approaches may be used to estimate the position of queue tail [15][20].

Literature on queue estimation from first principle models can be categorized into two modelling classes [19]:

- models based on the conservation law, using the cumulative traffic input–output information; and
- models based on shockwave theory.

Stochastic properties of queue development are directly taken into account by Markov chain models, as for example [23][25]. These models describe the queueing as a stochastic process with probabilities of queue change being given by probability distributions.

The third class of models found in literature are black-box models trying to predict the queue length based on known “training” data. These include autoregressive models [13], neural networks [14], combination of neural networks and fuzzy logic [18], or neural network constructed with the help of genetic algorithms [24].

With the development of vehicles equipped with Radio Frequency Identification (RFID) and GPS technologies, more theoretically formulated models appeared, refining the original estimate of the queue length from the input-output formulation using the conservation law to vehicle delay estimates [9][21]. The obstacle with which the proposed methods have to deal with in this case is the significant inaccuracy in determining the position of the vehicle in urban areas by GPS.

A promising, yet largely theoretical approach is outlined by Comert [10], in which the author studies the probability distribution of queue tail at an isolated signalised intersection in case that the queue contains a certain amount of cooperative automated vehicles (CAVs). Typically, the CAVs are able to determine their position much more precisely and, with the help of other sensors built into the vehicle, they announce to the Road Site Unit (RSU) also other important data from their surroundings. The approach has been tested only in a simulated environment and currently assumes a certain arrival profile (i.e. Poisson, it does not and probably cannot completely include the effects of gating by an upstream intersection).

The literature review of A.3 revealed a promising paper of Yang and Rakha in D4.1. It describes an optimisation algorithm that adapts speed of an automated vehicle in such a way that the vehicle passes the downstream signalised intersection with minimum fuel consumption. Their method computes optimal value for deceleration and consecutive acceleration of the vehicle, taking into account information about green and red signal (i.e. SPaT message, although this has not been explicitly stated) and possibly also the estimate of queue tail position. The position of the queue tail is derived from the original Lighthill-



Whitham-Richards shock-wave model and its approximation of fundamental diagram, unfortunately the accuracy of queue length model is not directly assessed in the paper. The authors report up to 40% fuel savings in case of 100% penetration rate of automated vehicles, and – for multi-lane roads – a limit of ca. 30% CAV penetration below which their method has adverse effects and actually causes increase in fuel consumption.

The results of the enhanced queue modelling developed in MAVEN and described in this Chapter are available to other components through the LDM of Section 3.2. Under WP5 a data fusion algorithm to combine various external queue sources with the ImFlow queue model has been developed. More on this can be found in D5.2 and the data fusion can be seen in action in the demonstrators of D4.2 and D4.3. This same data fusion can be used to take results of this Chapter into the control algorithm.

4.1 Meaning of queue length

The queue length reported by different models is not always related to the same quantity. Different models work with different queue length definitions. An overview of possible queue length characteristics is given for example in [22] and [17]. In the further text the queue length will always correspond to the *uniform maximum queue reach* as defined by [22] and provided by the percentile correction to the Highway Capacity Manual (HCM) 2000 average back of queue formula in Chapter 16, Appendix G of [7].

4.2 Model structure

The queue length model used by WP4 in MAVEN has the following schematic structure:

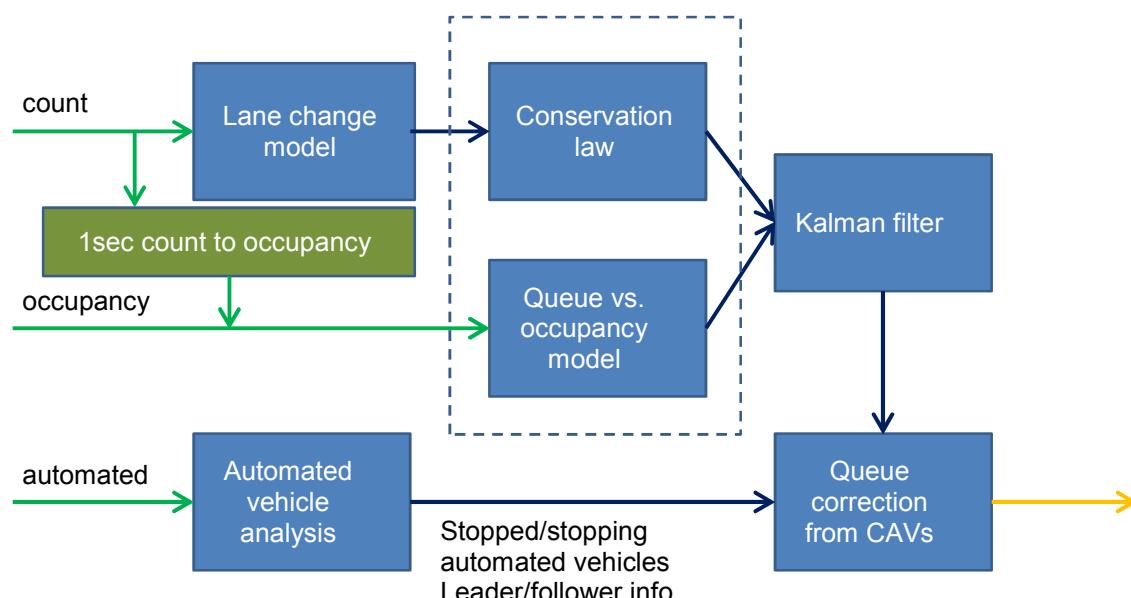


Figure 4-1: Block diagram of the MAVEN queue length model structure

4.3 Lane change model

The first component of the model is a subsystem that models how vehicles change lanes on multi-lane roads after being recorded by upstream detectors at the beginning of the modelled intersection arm. This model is data-driven, based on autoregressive identification of most



likely discrete distribution of lane changes so that the input-output balance of the whole section remains intact, i.e.

$$\sum_i c_{\text{upstream},i} = \sum_j c_{\text{stopbar},j} \quad 4-1$$

where $c_{\text{upstream},i}$ denotes vehicle count on the upstream detector for lane index i , $c_{\text{stopbar},j}$ is the vehicle count on the stop-bar for lane j . Written in matrix form it means continuously estimating matrix R such that

$$\mathbf{c}_{\text{stopbar}} = \mathbf{R}\mathbf{c}_{\text{upstream}} \quad 4-2$$

where \mathbf{c}_* are vectors containing upstream and stop-bar detector measurements.

4.4 Estimation without using information from probe vehicles

For the estimation we start with the basic state-space model in the form of conservation law

$$q(t) = q(t-1) + I(t) \times S \times z(t) \quad 4-3$$

i.e. the initial queue is updated by the number of discharged vehicles and by the number of new additions into the existing queue. Here, $q(t)$ is the queue length, $I(t)$ is the demand – the count of incoming vehicles on the lane, S is the saturation flow of the lane, and $z(t)$ is the green length (or green ratio in case that the model is cycle-based).

In case of no information about queue discharge, an additional model is used to estimate the number of vehicles that left the queue

$$y(t) = q(t) - \hat{q}(t-1) + I(t) \quad 4-4$$

where $y(t)$ is the amount of cars passing the stop-bar, and $\hat{q}(t-1)$ is the estimate of the last queue length.

The above model holds for an overflow queue, i.e. for situation when even after the end of the green signal, the queue length is nonzero. If during the green signal the whole queue discharges, we set explicitly

$$q(t) = I(t) \times r(t) \quad 4-5$$

with $r(t)$ the red duration. This means the queue is formed only by cars probably caught by the red signal, and

$$y(t) = I(t) \times z(t) + \hat{q}(t-1) \quad 4-6$$

meaning that from the total number of arrivals, only those that arrive during green will continue, and that the initial queue completely discharges.

The above equations form a state-space model with $q(t)$ as the state and $y(t)$ as a measurable output variable. In this formulation, a Kalman filter (or its non-linear variant) can be used to correct the model prediction based on measured model output. If the passage over the stop-bar $y(t)$ cannot be measured, we must rely on the measurements provided by



“strategic detectors” at connected road segments and use vehicle turning rates to reconstruct back an estimation of $y(t)$.

These relations are typically determined by traffic engineers at the moment of intersection design. Unfortunately, the common traffic engineering praxis works with daily averages, and the turning rates often vary during the day.

4.5 Occupancy correction

In the case that the upstream detector overflows and the queue tail reaches behind it, the standard approach described above does not work anymore. Similarly, as the queue length grows, the estimation becomes less precise. In order to address this, our model is augmented with linear queue length – occupancy model in the form

$$q(t) = \kappa O(t) + \sigma \quad 4-7$$

where $O(t)$ is the time occupancy of the detector and κ and σ proportionality constants that depend on the distance of the detector from the stop-bar and on detector size.

4.6 Remarks

1. The equations mentioned above are formulated for average values and detector sampling with relatively low frequency. That is, in the time period (typically say 5 minutes) we have average demand denoted by I_t , average length of the green signal z_t and average output y_t . However, real queue lengths are instantaneous at time instants t or $t - 1$. By incorporating only small corrections, the model can be adopted to dense detector sampling as it is used e.g. at the Helmond test site.
2. The formulations in Sections/paragraphs 4.4 and 4.5 are simplified and hold for a constant cycle length of traffic lights. In this case, $z(t)$ is expressed as a ratio of green length to the whole cycle length. The model can be easily extended to variable cycle lengths.
3. During the estimation the model is valid practically only when the queue exists. When there is no queue, the model degenerates. However, this point can be used for calibration – the queue is zero and we can start the computation if its length anew.
4. The model taking care of the beginning of the queue does not have the property of observability. This drawback is compensated by the second part of the model holding for the end of the queue.

4.7 Troubles

1. For the standard state-space estimation the model matrices are assumed known. Here, the matrices are formed by saturated flow S and possibly also by turning rates. Both these variables should be known from the intersection design. In reality, their knowledge is only approximate and they should be continuously estimated.
2. The vehicle count entering most of the equations is measured on a “strategic detector” at the entrance to an intersection arm (which could be the start of an egress approach) or at least sufficiently distant from the stop-bar. For this detector we suppose that it is never saturated, i.e. it could not happen that the queue tail reaches behind it.
 - a. The existence of such a detector at every intersection arm may be a too strict assumption
 - b. The condition of an unsaturated detector during the whole day is also quite ambitious.



- c. To certain extent the occupancy-based correction mentioned in Section 4.5 helps in this case at least when the queue tail approaches the detector.
- d. We do not explicitly address the case when the detector is completely saturated over the whole signal plan cycle and assume that from the viewpoint of the whole system this constitutes an exceptional case (due to traffic accident, weather conditions, roadworks etc.).

4.8 Queue length estimation using information from probe vehicles

A significant improvement to the problem of queue estimation using the input–output formulation can be achieved by incorporating information from probe vehicles into the queueing model. The information obtained from CAVs can be used for more precise estimation of the actual queue length on a lane and use it to for calibration. Our previous standard approach could calibrate only if the queue completely discharged which on high-volume intersections typically does not happen during the rush hours.

In the MAVEN setup, every CAV that approaches an intersection can be used as a probe vehicle as it provides the RSU with regular updates of vehicle position, speed and its surrounding vehicles on the lane.

We assume that the information coming from the CAVs can be used for our purposes constitutes at least from:

- *lane index*, a number of lane that the vehicles are travelling on at the moment (at least MAVEN CAVs are able to measure this);
- *vehicle position*, precise enough that we can estimate the distance to the stop-bar (and possibly also estimate the lane in case that lane index is not present); and
- *vehicle speed*, measured in a sufficiently short period that from its drop we are able to recognize the moment where the car joined the back of the queue.

Our approach works on the following simple principle: We continuously analyse data provided by CAVs to the RSU and from the data-set we extract time and vehicle position from the stop-bar at the time instance when the vehicle enters the queue. When a car enters the queue, its speed drops below the so-called crawling speed (in our case defined as 1 m/s) for some time. Alternatively, we can analyse the differences in vehicle position to estimate the vehicle speed. The queue length determined by CAVs is then defined as the maximum distance from the stop-bar of all CAVs with a low enough vehicle speed and driving on the same lane.

The queue length estimated by the model is then corrected using the estimated position of the last queueing CAV by merging the Kalman-filtered values from the conservation law model (which are normally distributed) and the (possibly not completely accurate, see 3.3) CAV position for which we also assume normal distribution.

4.9 Remarks

The correction code as outlined above assumes that when a vehicle stops, it stops in the queue. Situations as unguarded pedestrian crossings are not taken into account yet.

4.10 Preliminary results

Lane change model

The lane change model has been tested on Prague-Zlicin and Helmond networks. The tests show that for moderate-to-intense traffic conditions our data-driven approach works well, as



the measured data provides enough evidence for estimating the intra-lane movements. The situation becomes worse as the vehicle count decreases as the model does not have enough evidence to prefer a particular variant against all others.



Figure 4-2: Intersection approach overview

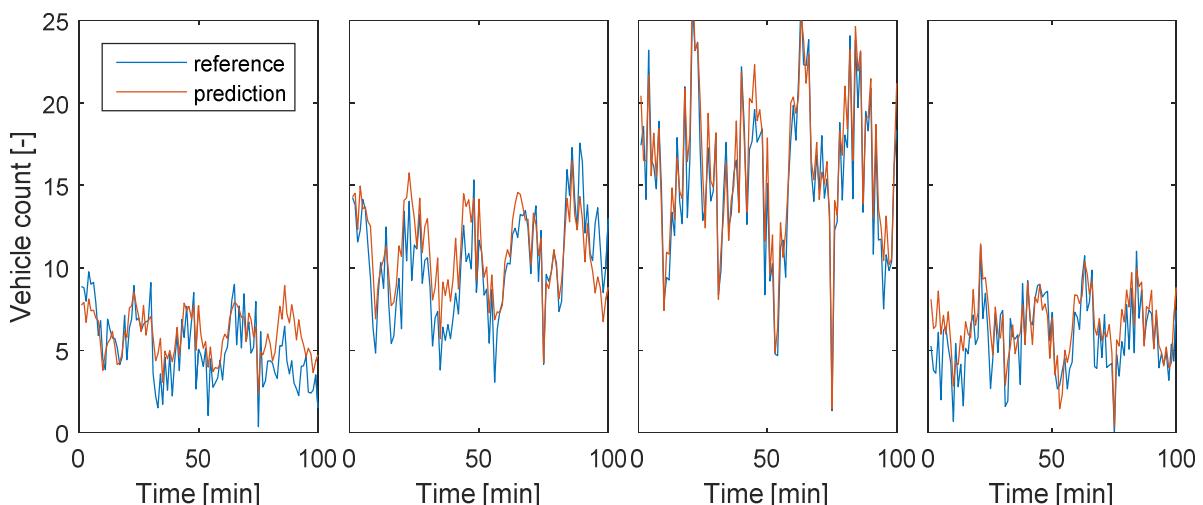


Figure 4-3: Prediction (red) and real (blue) values of vehicle counts at stop-bar detectors

Figure 4-2 shows a satellite image of the situation in Helmond, intersection 701 (Europaweg/Hortsedijk), while Figure 4-3 shows the results of the prediction and real value of vehicle counts at stop-bar detectors. The situation is at a two-lane urban arterial road that branches into four lanes. Traffic volume is approximately 2400 veh/hr, absolute error is 4.13–5.98, standard error is 0.154–0.199, and mean standard error over all lanes is 0.182.



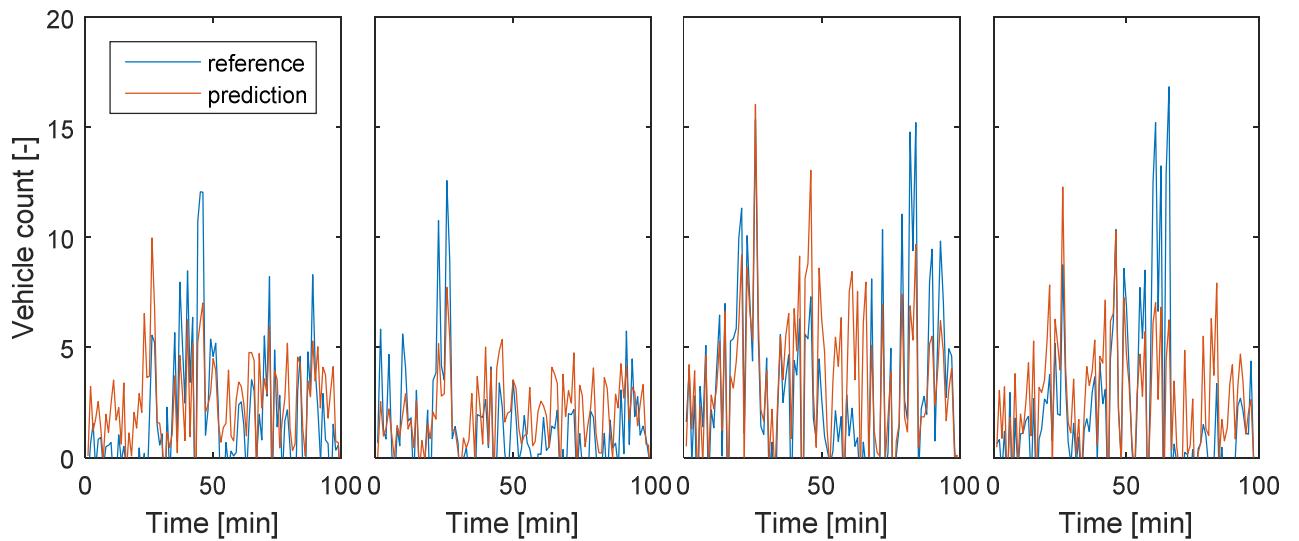


Figure 4-4: Prediction (red) and real (blue) values of vehicle counts at stop-bar detectors

Figure 4-4 shows the prediction and real value of vehicle counts at stop-bar detectors at a two-lane urban arterial that branches into four lanes. Traffic volume is approximately 600 veh/hr, absolute error is 4.24–8.23, standard error is 0.158–0.273, and mean standard error over all lanes is 0.202.

Queue length estimation

The queue length estimation model has been tested on Prague-Zlicin and Helmond networks as well. It uses information about lane changes to estimate the number of vehicles that arrive into the queue tail at every modelled lane and provides the user with an estimate of the horizontal queue starting at the stop bar. The current results indicate that the CAV correction significantly improves the initial accuracy of the model, see the figures below. Of course, the accuracy of the proposed model is limited by the accuracy of the lane change model: In case that the lane change model does not predict the arrivals into the modelled lanes correctly, the resulting queue length estimate will be incorrect as well. However, this occurs mostly in periods with low traffic volume and very short queues (usually up to 3 vehicles) that discharge completely on the next green signal.



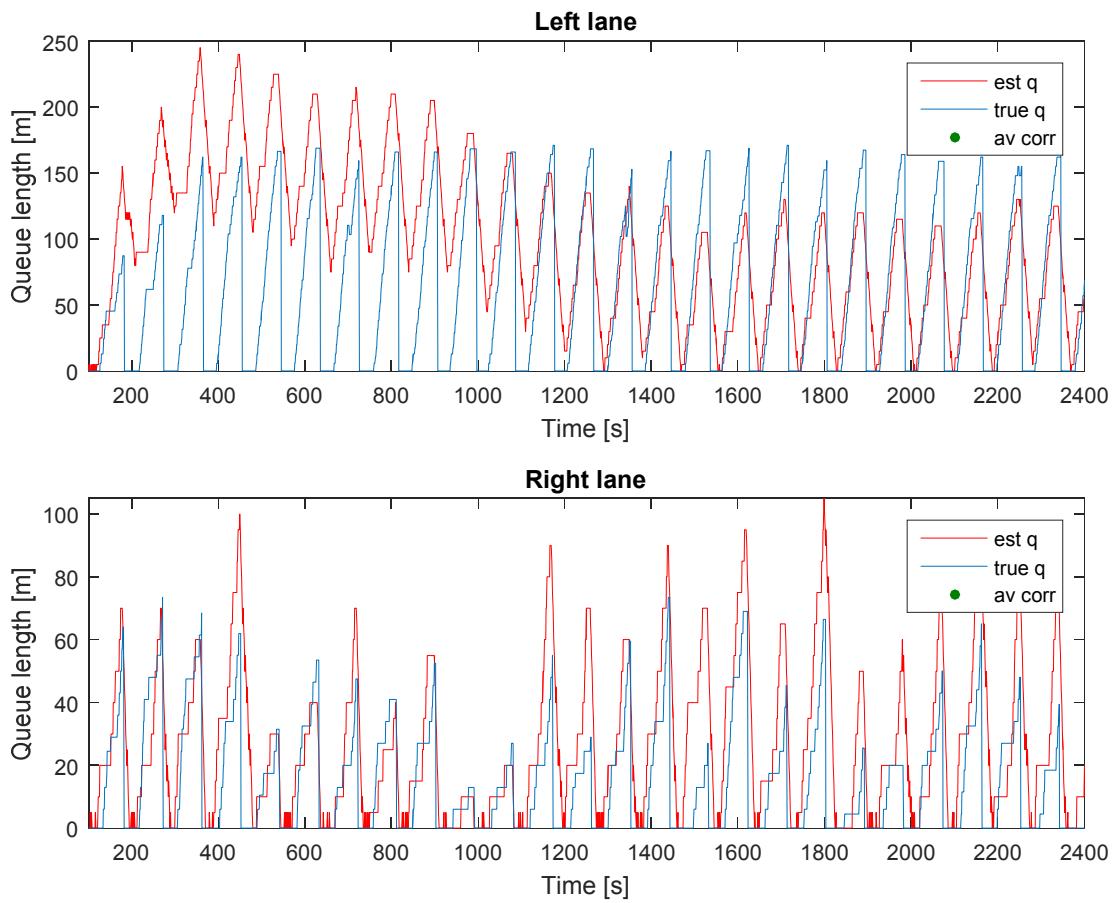


Figure 4-5: Queue length estimation without correction from CAVs at two-lane approach. Note the significant error of the model on the left lane, which is due to integrating nature of vehicle conservation law.



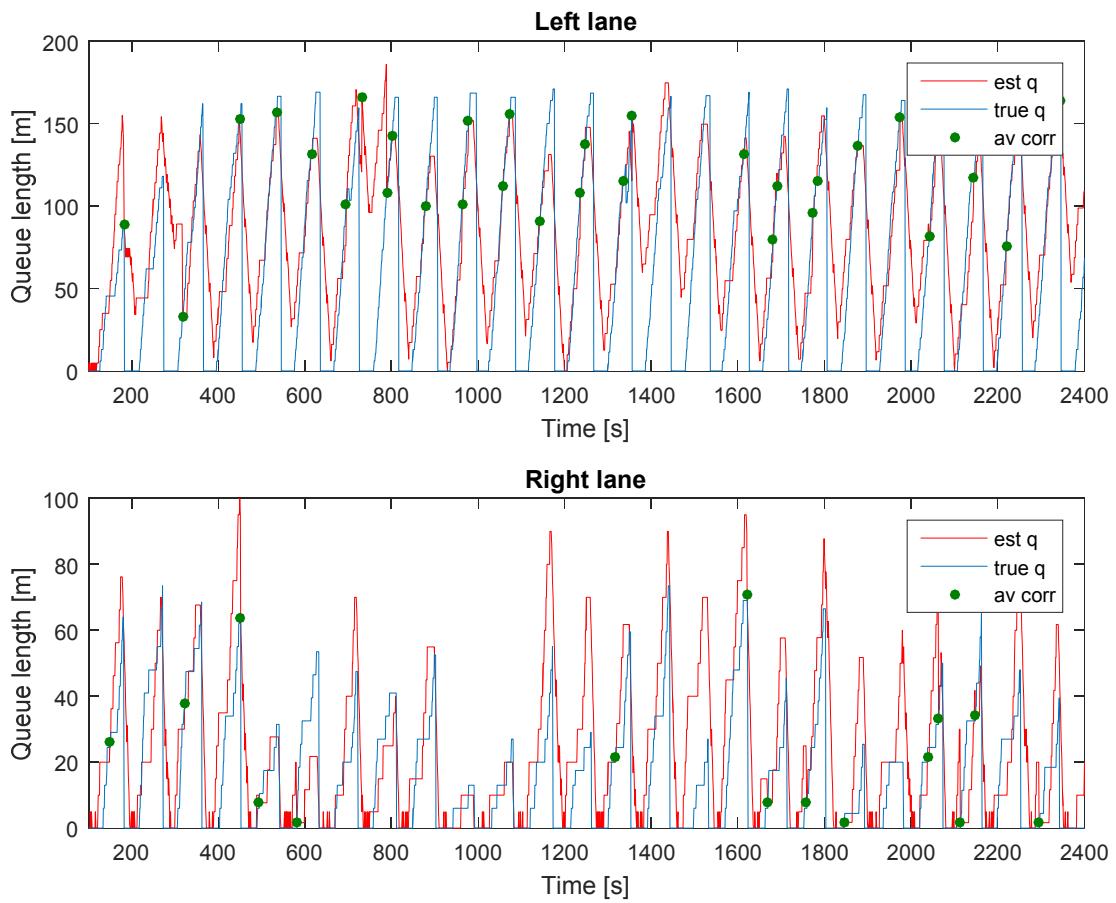


Figure 4-6: The same approach as in Figure 4-5, with queue model that incorporated the information from CAVs. Simulated CAV penetration rate 20%, flow approximately 1200 veh/hr.

The summary of results for the experiment from Figure 4-5 and Figure 4-6 is shown in Table 4-1. We can see that in denser traffic conditions even for 20% CAV penetration rate the information from automated vehicles significantly improves the mean error for longer queues and has certain positive effect in correcting outliers as well.

Table 4-1: Results for queue length estimation at high traffic volume

	Maximum difference [m]			RMSE [m]		
	No CAVs	CAV correction	Improvement	No CAVs	CAV correction	Improvement
Left lane	210	166	21.0%	1.35	0.799	40.8%
Right lane	95.0	73.7	22.4%	0.423	0.354	16.3%



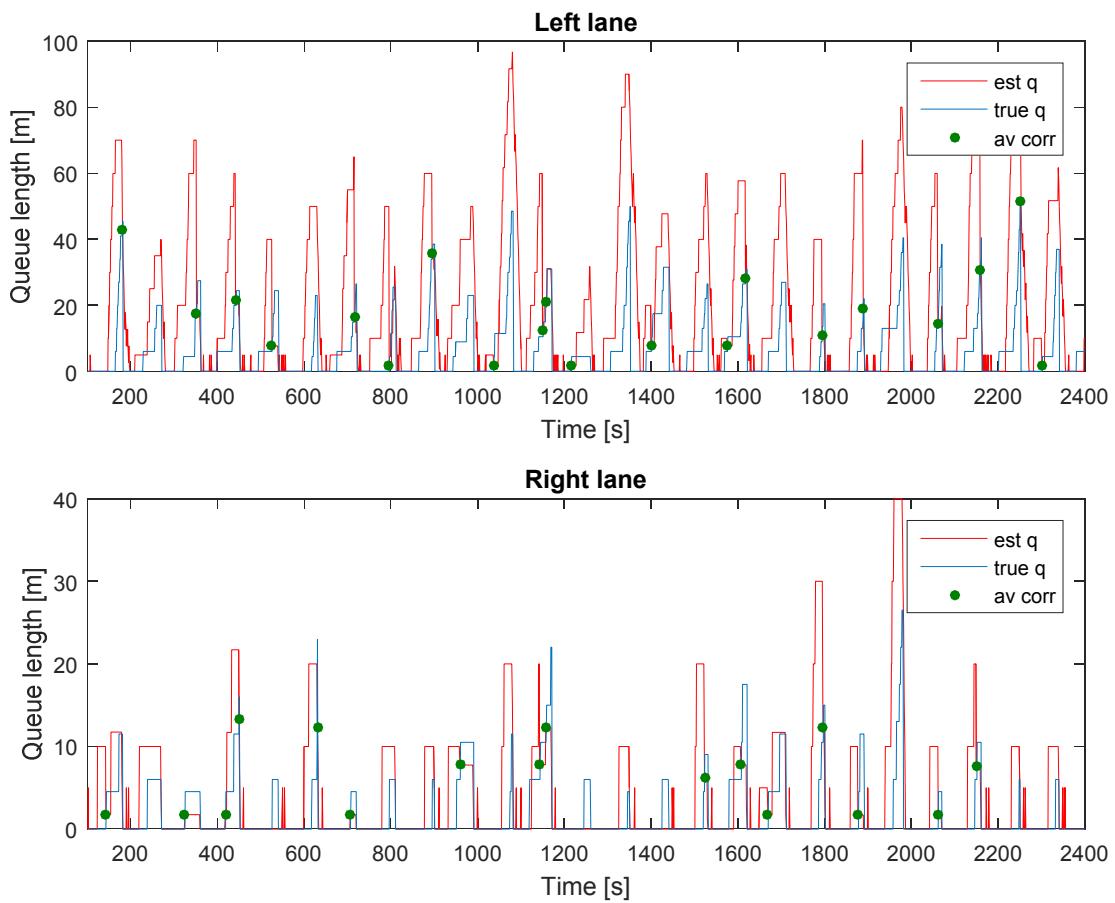


Figure 4-7: The same two lanes again, 20% penetration rate, incoming flow approximately 600 veh/hr.

Table 4-2: Queue estimation results for low traffic volume

	Maximum difference [m]			RMSE [m]		
	No CAVs	CAV correction	Improvement	No CAVs	CAV correction	Improvement
Left lane	70	70	0%	0.289	0.271	6.2%
Right lane	13	13	0%	0.0673	0.0673	0%

As shown in Figure 4-7 and Table 4-2, the lane change estimation does not work so well in this case. Due to very low traffic volume, when the queue estimation drifts away from the true value, there are in many cases no automated vehicles that would provide on-line information to correct the estimate. The mean estimate is still slightly better for the lane with more traffic, but the outliers are not corrected in this case.



5 Local level routing

A *local level routing* system is a type of route guidance system. Originally, route guidance systems were invented to facilitate driving in unfamiliar places and compute the shortest routes in a static map without real-time traffic information. Nowadays, drivers also use route guidance systems to get information of real-time traffic conditions and suggestions on alternative routes to avoid congestion, for parking availability, etc.[34].

Under the C-ITS environment, local level routing systems aid to find optimal CAV routes using short-term predictions based on local information exchanged between CAV and infrastructure. Some authors approach this integration by developing systems that are able to route vehicles given predictions of traffic signal plans (or using historical plans) with the input of vehicle routes into the traffic controller ([35],[36]) or without such input ([37],[38],[39],[40]); while others aim to propose an integrated solution for both routing and signal control problems ([41],[42],[43],[44]). We propose a local level routing system by assuming that CAVs have the knowledge of average travel times of each road in the whole network, but providing more accurate short-term travel times of the local area around them will help to improve mobility and reduce negative environmental impacts. The key aspect is the estimation of the travel time as reliable as possible that a vehicle would experience if it would enter certain road at a specific time.

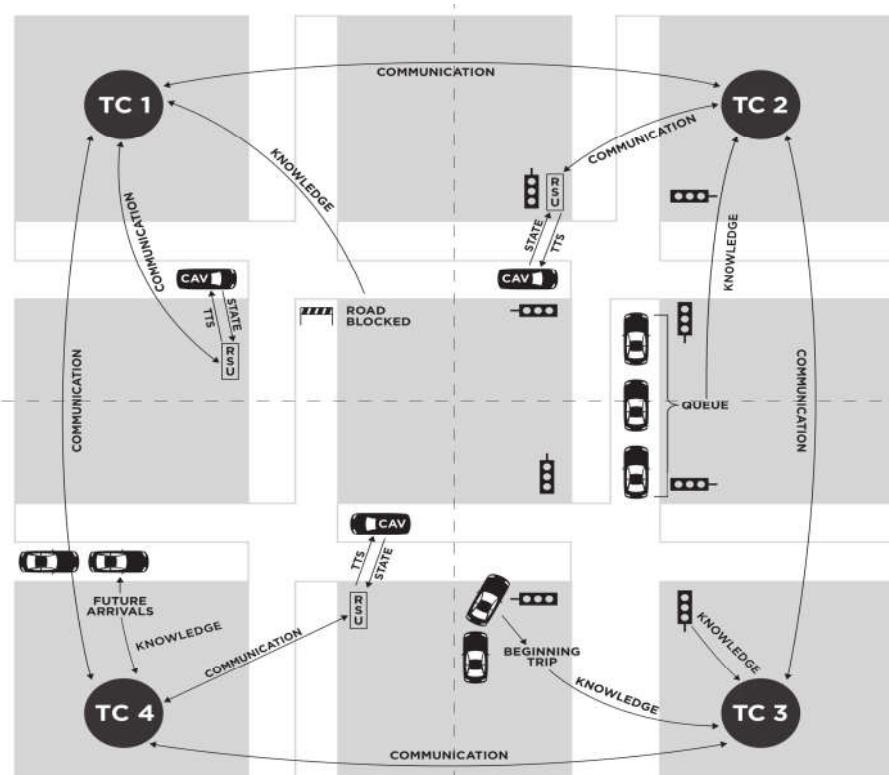


Figure 5-1: Communication among CAVs and Traffic Controllers (TCs) in a Traffic Network (TN)

Figure 5-1 illustrates the data flow for the proposed local level routing system. The Traffic Network (TN) represents a network where cooperative agents (e.g. Cooperative Intersections and CAVs) are connected and information can be shared among themselves. Considering a traffic network with junctions (intersections, highway entry/exit points, etc.) linked by edges (road, streets, etc.), the travel time of traversing each edge may use information of 1) **vehicles on lane and their attributes**, corresponding to when and which vehicle will be passing on the edge throughout a planning horizon; 2) **unexpected events**, e.g. representing edges not available to be used or with reduced speed; and 3) **queue length** as well as 4) **signal plans** (begin and duration of green phases) for calculating delays, given the



junction's capacity of discharge. From the infrastructure perspective, while TLCs control traffic lights, the Local Level Routing uses Traffic Controllers (TCs) that model the traffic (i.e. movement of vehicles) throughout certain sub-network (part of the traffic network), which can be the incoming edges into a junction controlled by a TLC or even several junctions (e.g. arterial), including unsignalized ones. Additionally, these TCs can communicate between each other in the same Traffic Network (TN). On the road user side, Cooperative Automated Vehicles (CAVs) are expected to have embedded On-board Units (OBUs) that contain off-line information of historical discrete travel times or they are able to compute them for all edges in the network (which is defined by the vehicle manufacturer in some fashion). These OBU's must be able to get on-line information with better estimation of travel times (TTs) of the edges within the TN from TCs through Roadside Units (RSUs), once they are inside their communication range. Moreover, CAVs may send information related to their state (position, speed, etc.), capabilities (e.g. acceleration and deceleration) and planned intended route. The infrastructure only provide discrete-time estimated travel times to CAVs, while the routing part is responsibility of the OBU/CAV. The format of this TTs information is described in Section 5.5.

From this point, we need to understand the behaviour of two groups: CAVs; and *modelled* vehicles. Both of them always have a *local destination*, which is the last junction within the area modelled by the Traffic Controller (TC) each of them is at certain moment, and they may have a *global destination*, outside the area modelled by the same TC. Once CAVs obtain information of the TN's travel times they may change their local destination, because from their entire route perspective it is better to change completely its route, instead of finding an alternative between current junction and local destination. This is important as a faster alternative route taking more green waves to a local destination may not improve the route to the global destination, because the lack of information from the edges outside the TN. Another point is that it would be too demanding for a traffic controller to calculate best routes for each vehicle within the whole network in a very short period of time, though that could be possible for a fixed single local destination given Origin-Destination pair flows. In one hand, CAVs may share their intended route (valid for a short period of time). In the other hand, modelled vehicles are vehicles that do not communicate with TCs, CAVs that don't share their intended route, or a predicted vehicle to arrive on certain edge within the planning horizon.

The proposed Local Level Routing (LLR) system can be more (or totally) centralized, or distributed. Its scalability and level of "centralization" depends on the available information and, of course, the acceptable running time of the algorithm for each TC. For instance, if a Traffic Controller (TC) knows the real-time information of CAVs, signal plans, and queue length (when necessary, e.g. traffic lights) of an entire area (e.g. a neighbourhood), then the LLR system can be centralized for the area. Considering Figure 5-1, you would have only one TC with all information needed for the system and not 4 of them having to communicate with each other the 1) inflow profiles, 2) TTs and 3) queue prediction. This would also mean that Section 5.5 (Traffic Controllers Knowledge Sharing) would not be necessary (though the TTs messages are still sent to CAVs), as a central TC would get information direct from the sources (each TLC and RSU) to estimate these 3 values for each edge. On the other hand, this central configuration is not commonly seen in real life, as there are limitations in the communication bandwidth and availability of communication links between intersections in the infrastructure.

Transforming the system's concept into a system-wide view, the LLR system is composed by five sub-systems, seen in the component UML diagram in Figure 5-2. The Component UML diagram illustrates the subdivision of the local level routing system (from a TC that communicates to another TC perspective), in which each sub-system has inputs, processes and outputs. The arrows connecting them illustrate the required input from the sub-system the arrows are pointing to. When these arrows touch a circle it means the interface is already



provided in the algorithm. The sub-systems represented in Figure 5-2 are embedded in the same algorithm installed on each TC, so the communication between them (output from one sub-system into input to another) is done in the same TC instantly in the same algorithm. The only communication outside the TC are the data messages exchanged between Traffic Controllers, what needs an interface for it (which is the meaning of the half open circles in Figure 5-2). Although how each TC will communicate is out of the scope, the content and format of these messages exchanged between each TC is explained in section 5.5. The paragraphs after Figure 5-2 until the end of section (before Section 5.1) contain a summary of the characteristics of each sub-system shown in Figure 5-2. From Section 5.1 until the end of section 5 there are more detailed explanation of the sub-systems.

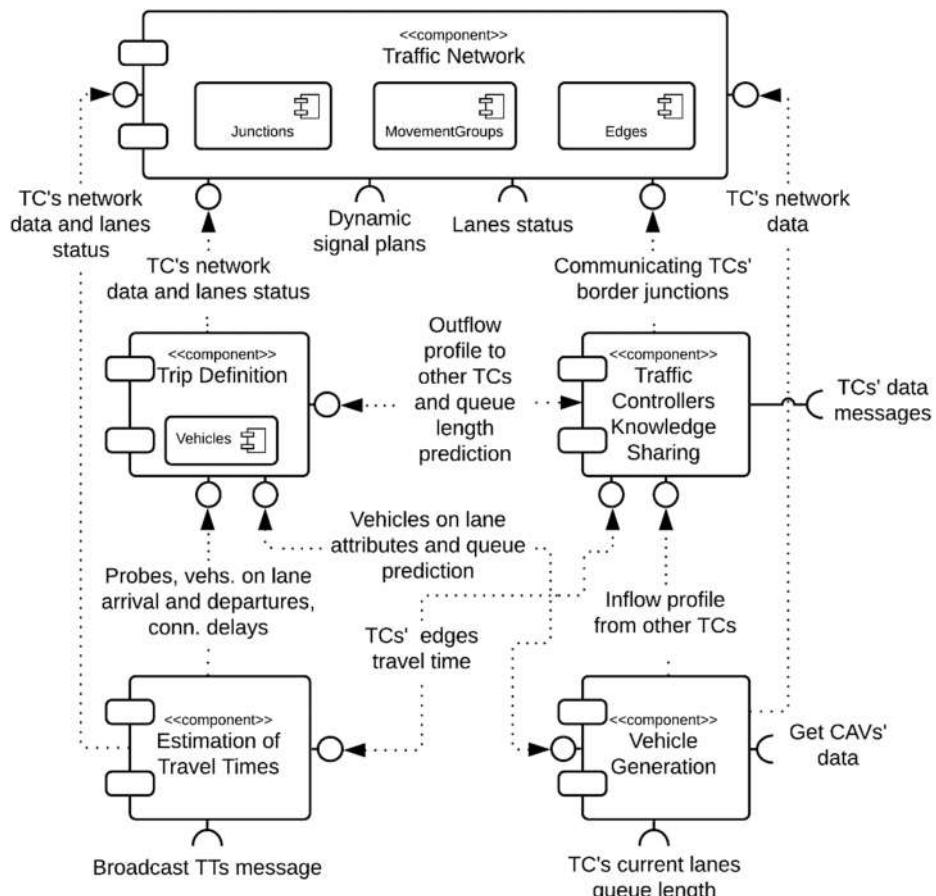


Figure 5-2 Component UML diagram containing the proposed Local Level Routing sub-systems.

The **Traffic Network** (Section 5.1) sub-system is responsible for defining the Traffic Controller's sub-network attributes, from the incoming and outgoing edges as well as the possible connections between them for each junction it controls, up to the aggregation of these connections into movement groups (this term is described below). It not only provides such static information to all sub-systems, but it is also a single point for other sub-systems retrieve or change dynamic information like maximum lane speed and lane status (e.g lane blockage) what requires an interface, queue length prediction, and inflow profile. It is composed by three classes:

- *Junctions*, containing information related to the junctions, like their type, signal plans, planning horizon (prediction time), the data received from CAVs and link to the other classes:



- *Edges*, considering the information of each edge such as travel times, lane queue prediction, inflow and outflow profiles, connection critical gap times, saturation flow and capacity; and
- *Movement Groups*, representing the groups of connections that either always get green light at the same time (when signalized) or yield to the same connections (when unsignalized) for the TC's modelled junctions, including information related to the duration of each movement group green phases as well as its vehicles considered as probe for estimating the travel times.

The **Vehicle Generation** (Section 5.2) sub-system comprises the set of functions that generate vehicles on the lanes to have their trip defined by the *Trip Definition* sub-system. Notice that this generation is only in the modelling algorithm of the LLR system and not for the simulation (the simulation only simulates the traffic where TCs gather data and exchange messages between themselves and CAVs which also reroute using received information). This sub-system defines 1) the future arrival of vehicles based on the lane inflow profile by estimating the arrival headways; 2) the vehicles starting on lanes by estimating vehicles beginning their trip on each lane (data from the *Traffic Network* sub-system) and current queue length (data from a required interface), but also constrained by the current vehicles on lane and queue prediction from *Trip Definition* sub-system; as well as 3) the vehicles already on lane, i.e. CAVs (required interface) and those expected to be on lane from last time step. The sub-system can also generate artificial probes that interact with a vehicle in front and traffic light but a vehicle behind does not interact with it.

The **Trip Definition** (Section 5.3) sub-system defines which edge, junction and lane (as well as specific attributes for each) CAVs and modelled vehicles (i.e. any vehicle) will pass through the TC's sub-network until it reaches its local destination. This happens only in the LLR model, the system doesn't define trips in the simulation, the simulator itself reroutes vehicles based on the TTs each vehicle receives. It predicts future traffic conditions within the planning horizon by modelling the movement of vehicles along the lanes, but the movement modelling is split by each green phase of the lane a vehicle will run on. The vehicles are either generated by the *Vehicle Generation* sub-system, or estimated by the *Trip Definition* sub-system itself at previous time steps, to be on each lane. For instance, when a CAV is detected (and also its intended route), the *Trip Definition* sub-system, in general, estimates the 5 events of section 5.2.2 on each lane for each vehicle's intended route. This is done to know what will be the predicted future traffic on each edge and thus its travel times. It basically considers: 1) the duration of green phases that each movement group receives; 2) the possible connections to other lanes, which vehicles have may use at certain time (stemmed from *Traffic Network* sub-system); and 3) queue length prediction of outgoing edge which may delay the departure and change vehicle's next edge's lane, and movement group. This queue length prediction is different from the one proposed in Section 4, because it is based on prediction of the future traffic (by the LLR system) and the vehicles will have to stop in the future. This sub-system also defines probe vehicles, which are those that their arrival and departure time on a certain lane will be used for estimation of travel times. It uses these probe vehicles (instead of all CAVs) due to the fact that If a CAV is starting its trip in the middle of the edge, it is not known when it arrived on the edge. This would underestimate the travel times. Moreover, the sub-system calculates connection delays, given the constrained capacity of discharge if vehicles must yield to preferential traffic (traffic stream that has the priority at intersections when there is a conflict between traffic streams).

The **Estimation of Travel Times** (Section 5.4) sub-system accounts each vehicle's lane arrival and departure times as well as their movement group (from the *Trip Definition* sub-system) to estimate an average travel time of the incoming edges. When no trip of a probe vehicle is defined within a time window of the planning horizon for a certain lane and movement group, it is possible to use a queuing delay model considering all vehicle arrivals and departures as well as connection delays and signal timings, or use travel times of the



artificial probes. These artificial probes are vehicles that can be generated (only in the LLR model and not in the simulation or real life) and interact with the vehicle in front in the LLR model, wait for green light and yield to preferential traffic, but vehicles behind them don't interact with them (to avoid disturbing the real travel demand). However, in a quite large network with long planning horizon and short time window intervals, generating an artificial vehicle for each lane and each movement group can increase considerably the running time of the algorithm. Additionally, at very high flows the travel time don't need to consider these interactions between vehicles microscopically and the queuing delay model can be a reasonable solution with a short running time. In case the lane status is blocked, it also influences the travel time. Moreover, the estimated travel times (TTs) of all TCs in the same traffic network are broadcast to CAVs (requires an interface).

Traffic Controllers Knowledge Sharing (Section 5.5) assumes that Traffic Controllers (TCs) share among themselves (through a required interface) the travel times of their incoming edges to the junctions they model traffic, the arrival profile on their outgoing edges to the neighbouring junctions are modelled by other TCs, and the queue length prediction (which is different from the one proposed in Section 4 as it uses the LLR model future traffic predictions with a longer time horizon) of their incoming edges that are outgoing edges of junctions modelled by other TCs. The inflow profile is used to generate vehicles by the *Vehicle Generation* sub-system, which is the outflow of an upstream junction estimated by the *Trip Definition* sub-system that needs the queue length prediction of outgoing edges to delay vehicle departures when there is a maximum queue (not possible to send any vehicle until it starts to dissolve). A TC needs information of which junctions modelled by other connected TCs (and their respective edges) it will send or receive data (i.e flow profile and queue prediction), provided by the *Traffic Network* sub-system.

The algorithm for the LLR system can be summarized as follows: RSUs detect CAVs and get data from them, in which this data may contain position, speed and route. The Vehicle Generation sub-system will insert these detected CAVs and modelled vehicles (coming from other TCs or being generated using historical information) into the LLR system traffic model. Then the Trip Definition sub-system (mainly) predicts the time, speed and distance to the stop line each vehicle will have at the begin and end of each green phase (on the lane each vehicle will run throughout its TC's route) of the signal plans (when unsignalized there is only one infinite green phase and modelling is done only by junction priority rules). The TTs are estimated based on predicted traffic and multicasted to each TC within the Traffic Network (TN), which then distributes that to RSUs. Afterwards, all RSUs broadcast the discrete time dependent TTs of all edges within the traffic network. CAVs receive the TTs, (some of them) change their route and on the next algorithm update time step, share again the position, speed, new route which will be considered for estimating TTs that other vehicles will use to reroute themselves.

The traffic simulator will only simulate the real condition of TCs exchanging messages and get information about signal plans and CAVs while also sending TTs to CAVs which will reroute based on the TTs data that they have. There is no command to insert/generate or remove vehicles or change routes and trips in the simulator by the LLR system algorithm.

5.1 Traffic Network

This sub-system stores the characteristics of the TC's network, it is the database needed by other sub-systems with static and dynamic data. There are five main network objects (junctions, edges, lanes, connections, and movement groups) that need information not only related to them, but also the link/mapping between each of them (e.g. the edges of a junction or the lanes of an edge). We will start explaining the required static information and then describe the dynamic information that can be changed by other sub-systems.



5.1.1 Static Data

Figure 5-3 exemplifies some information present in this sub-system. We can see few junctions/edges, some modelled by Traffic Controller 1 (TC1) and others by TC2. Here, modelled means which TC is responsible to model the discharge process of each modelled junction, i.e traffic flowing from the junction's incoming edges to its outgoing edges. Although we usually have two main types of junctions (signalized or unsignalized, i.e. whether a junction contains a traffic light or not), we categorize them further into the following list, where a junction may have characteristics of more than one category.

- *Cooperative*, for cooperative junctions, e.g. an intersection that has Roadside Unit (RSU) able to communicate with Cooperative Automated Vehicles (CAVs).
- *Modelled Non-Connected*, for junctions not connected to the TC, i.e. unsignalized or those that don't send their queue length and signal plans to their TC.
- *Modelled Connected*, for connected junctions, i.e. signalized and sharing queue length as well as signal plans.
- *Connections*, junctions that don't have traffic control, i.e. lane merge or lane split.

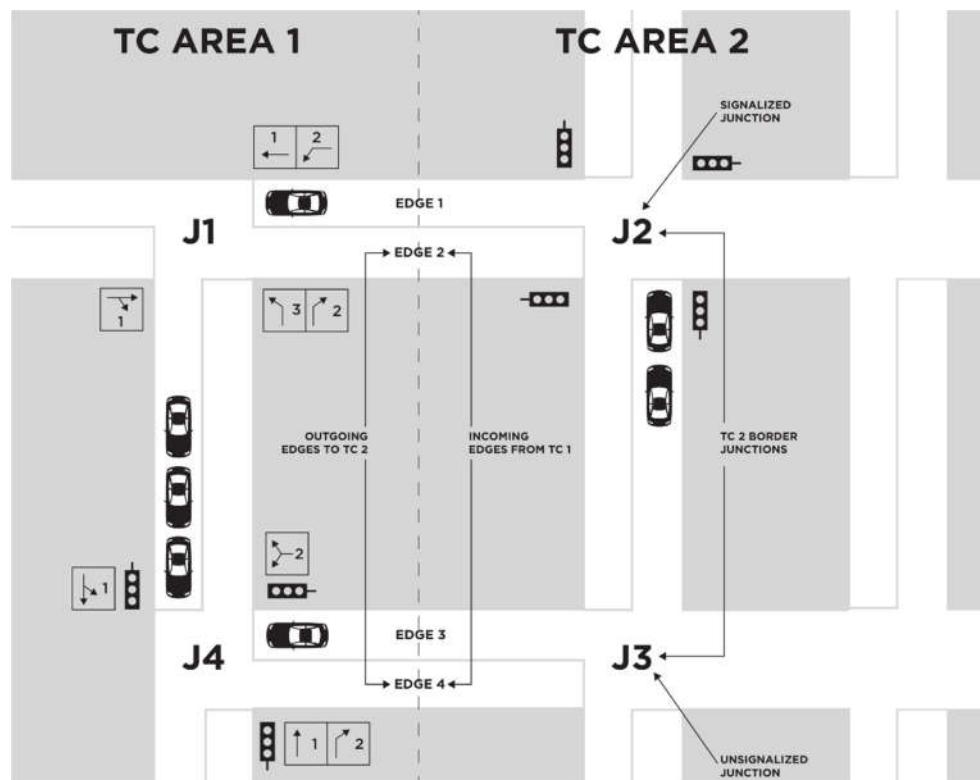


Figure 5-3 Example of some information required for the Network Representation sub-system.

Vehicles from one junction to another in a logical order. For a TC, there is a sequence of junctions to model the traffic. From one junction it estimates the departures of vehicles on all incoming edges, then goes to the junction that receives these vehicles from this upstream junction and so on. Each junction has incoming and outgoing edges, and each edge has its own lanes, as well as connections (the arrows in the TC Area 1) that represent which lanes are connected at the junction, i.e. the possible movements at junctions. Logically, edges may be incoming to certain junction while outgoing from another, e.g. edge 1 is outgoing from J1 but incoming to J2. Incoming edges need their turning rates (probability to a possible outgoing edge) as well as the probability that vehicles will finish and begin its trip on the



edge, in which for the latter it should be given the probability per distance to stop line and the average number of vehicles per length of the time window of the planning horizon.

Figure 5-3 also shows that these connections may be aggregated and have an identification number, which represents the group of movements that contains a common status (always green at same time for signalized and yield to same connections in unsignalized). For instance, the unsignalized Junction 1 (J1), is a 3-leg junction where vehicles running West-East directions have the priority. In this way, the through and right-turning movements don't yield to any traffic, so they are in the Movement Group (MG) 1. Vehicles taking the connection from West to South must yield to vehicles from East taking any connection, while vehicles coming South to West also need to yield to East coming vehicles, defining MG 2. The last, Movement Group (MG) 3, must yield to all connections which have their incoming lanes from West-East directions. The signalized junction J4 could have the same configuration of MGs, but as its MGs depend on the traffic signal plan (which we set only two phases as example), the set of MGs is different. Figure 5-3 also illustrates the TC border junctions, which are junctions that have either incoming or outgoing edges leading to a junction modelled by a different TC able to communicate with the TC of the junction been taken as reference. This is important for defining which edges need information to be sent to or receive from other neighbouring TCs.

Although the data described in this section is defined as static data, the information may be changed in case of unexpected events, e.g. reduction of maximum speed of a lane, but it is not considered dynamic data because its value is not assumed to variate throughout the planning horizon. The static information related to the TC's network must be inserted using the following order of functions provided in the system (i.e. add all junctions, then all edges and so on):

- *addJunction*, adding new junctions and their attributes;
- *addEdgeLanes*, adding edges, their respective lanes and their attributes;
- *addConnection*, adding connections between edges and their attributes;
- *addMovGroup*, adding connections' movement groups and their attributes;
- *addConnYield2Conns*, adding which connection yield to and their attributes; and
- *setUnsignalizedJctMovGroupPlans*, setting the traffic priority rules (permissive or protected greens) for unsignalized junctions and their attributes.

5.1.2 Dynamic Data

Analyzing again Figure 5-1, but looking at the flow of information in the illustrated traffic network, we see that the TC has the knowledge of some traffic data, but the information is referenced by the time interval it is valid.

For edge and lane data, e.g. like dynamic inflow profile, predicted lane queue length (by the LLR model and not the proposed in Section 4), and edge travel times (TTs), the planning horizon period, denoted t_{max} , is a sliding prediction time subdivided into a constant length $lenrge$ intervals called time windows, denoted as rge_i (as shown in Figure 5-4). In addition, at each update of the algorithm, t_τ that occurs between $lenrge$ time intervals, it is estimated the system dynamic data. The planning horizon defines the limit where it is desired to make estimations of the system's dynamic data. This t_{max} is an input for each junction and should be, at least, the time of one cycle when a signalized junction is considered in order for all movement groups to be able to discharge. The value of $lenrge$ is fixed for the whole traffic network and arbitrary chosen. For instance, the format of the travel times (TTs) is time dependent (specific for each time interval/resolution, e.g. 15 s) estimated throughout a planning horizon (certain prediction time, e.g. 180 s), which is equally divided into time windows (e.g. $180/15 = 12$ "windows") with fixed length (time resolution) for the whole TN.



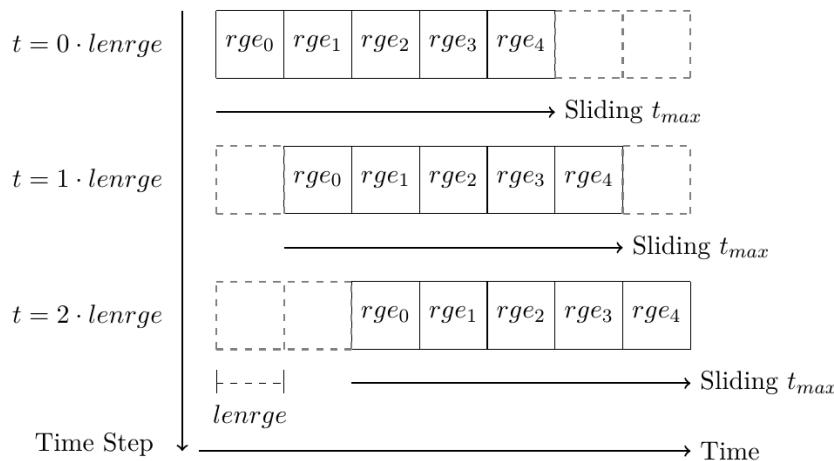


Figure 5-4 Division of a sliding prediction horizon time window into shorter intervals.

The dynamic data of connections and movement groups use the duration of the green phase for the movement group (and consequently also the connection), which is logical due the fact that such objects don't variate within the green phase. The dynamic data related to the signal plan has to be given for each movement group (MG) separately in respect to each phase it gets green, as it follows:

1. MG phase number begin time;
2. MG phase number end time;
3. MG phase number state, protected is no need to yield or permissive if vehicle must yield to preferential traffic;
4. MG phase number red + amber time; and
5. MG phase number amber time.

Assuming a traffic signal with 3 phases planned within t_{max} and a movement group, which we will call $J1MG1$, has green time in 2 of these phases, Figure 5-5 illustrates an example for the format of dynamic data. For instance, at the first algorithm update time step $t = 0$, a dynamic capacity of discharge of a connection that is in the movement group $J1MG1$, has values only for phase number 0, ph_0 , and number 1, ph_1 , because at phase number 2, ph_2 , it has red light. At the next update $t = 1$, there are values of capacity again only for the same phase numbers (0 and 1) but at different times.

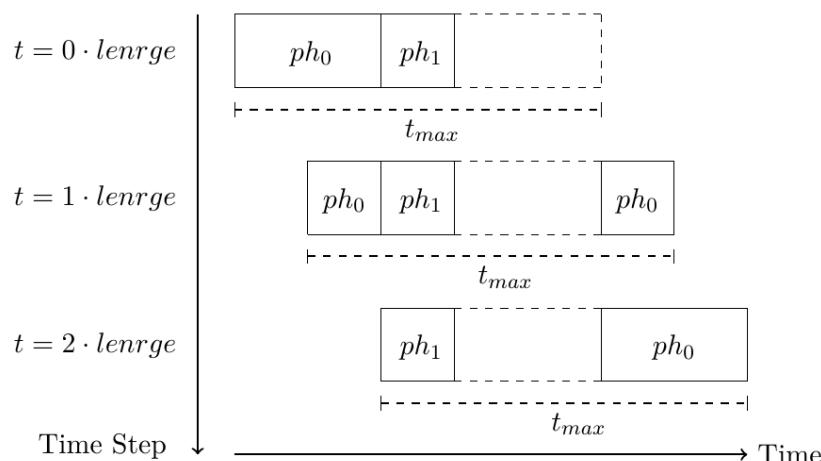


Figure 5-5 Division of movement group's, $J1MG1$, dynamic data by its green signal plan.



There are some dynamic data which are auxiliary for other sub-systems and not referenced by time. Sub-systems change their values according to the way they are able to retrieve the information later. For instance, the *Vehicles on Lane* edge instance (illustrated in Figure 5-6), denoted as *vehicleslane*, aggregates the list of all vehicles expected to be on each lane throughout the time (but without the information of when they will be on lane). It has three components, described in the following list.

- *Already on Lane*, *alreadylane*, represents the vehicles expected to be on lane due estimations of previous time steps or CAVs detected on the lane at the time the algorithm is going to update its calculations.
- *Starting on Lane*, *startinglane*, corresponds to vehicles that should be generated on the lane because they were not expected to be there given estimations of previous algorithm time steps.
- *Arrivals on Lane*, *arrivalslane*, accounting vehicles that will be arriving within the planning horizon, i.e. $[actualt, actualt + t_{max}]$.

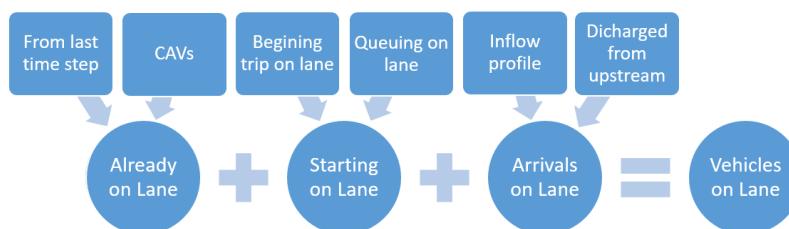


Figure 5-6 Composition of the Vehicles on Lane edge instance.

5.1.3 List of Required Data

The following list summarizes the required information related to each TC.

- Modelled junctions and their type in the order to be done the estimations by the system.
- Definition of which junctions are TC border junctions as well as their lanes in common with neighbouring different TCs.
- Each junction's edges, lanes, connections, and movement (signal) groups.
- Each edge's lanes, incoming edges turn rates (probability to a possible outgoing edge), and probability that a vehicle will finish its trip on the edge.
- Each lane's length, expected gap time between vehicles at the speed limit, maximum speed, fixed inflow profile, number and distance probability of vehicles beginning their trip, acc./decel. adjustments.
- Each connection's length, fixed capacity, average critical gap time and follow-up time, as well as which connection to yield to.
- Junction's roadside unit (RSU) communication range (when it equipped with one).
- Junction's planning horizon.
- Junction's next traffic signal plan within planning horizon (when signalized junction) for each movement group.

Additionally, the following list summarizes the information required regarding the TC border junction controlled by a different connected TC.

- Edge's lanes.
- Incoming lane's connections and mov. groups. connection's mov. group and outgoing Lane. movement groups's egressing lanes.

At a traffic network level, the required information is:



- The planning horizon time window individual length (which is also algorithm update time step interval);
- Low, mid, and high maximum flow threshold to classify the intensity of the traffic flow.

5.2 Vehicle Generation

The system models traffic microscopically, which means that every vehicle must be generated into the local level routing (LLR) system model. Vehicles are generated on the lanes of the vehicle's route (i.e. sequence of edge with respective lane), and it represents the predicted traffic. Based on the interaction between these generated vehicles and signal plans we can estimate the travel time each generated vehicle will experience along its route. There are three types of vehicles: *Already on Lane*, *Starting on Lane* and *Arriving on Lane*, and together they correspond to all predicted *Vehicles on Lane* throughout a planning horizon.

Vehicles already on a lane are CAV and the vehicles expected to be on certain lanes at the time the algorithm updates, which occur at fixed intervals of time steps. For this category, only newly detected CAV must be generated on a lane and the information needed is provided by the CAV. Vehicles starting on a lane represent vehicles that beginning their trip on the lane (given a probability to appear a new vehicle which don't come from upstream) and the number of queuing vehicles (presented in Section 4). These vehicles are generated at a vacant place along the edge with speed zero (for queueing) or a predefined speed (for beginning trip vehicles). For the vehicles arriving on lane, they are always placed at the beginning of the edge, and the task is to define when and at which speed they arrive (at the beginning of the lane) within the planning horizon. Figure 5-7 shows the types of arriving vehicles. When vehicles come from a upstream junction modelled by the same TC (so called *discharge upstream*) it is not necessary to generate the vehicle, as the Trip Definition sub-system has the information of these vehicles. When they come from a junction not modelled by the same ego TC modelling the arrivals it is needed information about the inflow profile to generate vehicles per time window:

- number of vehicles, arr_n ;
- mean speed, v_{mean} ;
- mean of headways, h_{mean} ; and
- standard deviation of headways, h_{std} .

This inflow profile data can use 1) historical values (called *fixed inflow profile*) when vehicles come from non-connected upstream junctions; 2) real-time values in an inflow message from a different TC sending vehicles (name as *dynamic inflow profile*).

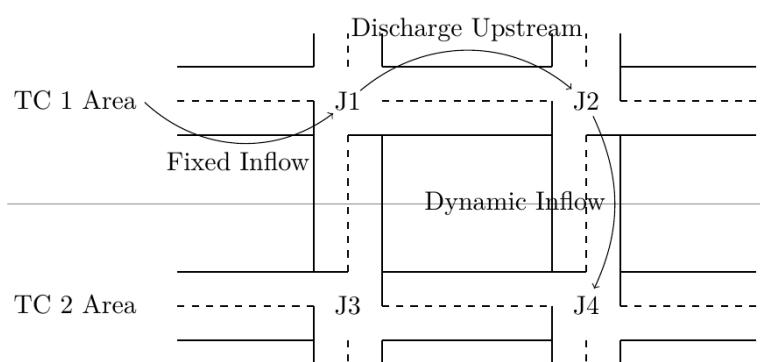


Figure 5-7 Different types of arrivals on lane. Fixed and dynamic inflow are responsibility of Vehicle Generation sub-system, while discharged from upstream is done by the Trip Generation sub-system.

We can find the flow of vehicles on lane l , λ_l^{arr} , by:



$$\lambda_l^{arr} = \frac{arr_n}{lenrge}. \quad 5-1$$

Based on maximum predefined thresholds to categorize the intensity of the flow (i.e. low, mid and high), we can choose the suitable statistical distribution for the estimation of headways. For each arriving vehicle, we set its arriving speed as v_{mean} and it is selected a random number between 0 and 1, which corresponds to a probability value of the chosen statistical distribution's Cumulative Distributive Function (CDF), denoted $F(h)$, and consequently vehicle's respective headway, h .

If λ_l^{arr} is low flow we use Negative Exponential Distribution due to almost or no interaction between the arriving vehicles from [45]:

$$F(h) = 1 - e^{-\lambda_l^{arr} h}. \quad 5-2$$

When λ_l^{arr} is mid flow it is assumed Pearson Type III Distribution, as certain vehicles will have interaction with the other (platoons) or not [45]:

$$F(h) = p(t \leq h) = 1 - \int_h^{\infty} f(t) dt. \quad 5-3$$

Although there is no closed solution for Equation 5-3, we can approximate the probability the probability for h as it follows:

$$p(h \leq t \leq h + \Delta t) \approx \left[\frac{f(h) + f(h + \Delta t)}{2} \right] \Delta t \cdot h. \quad 5-4$$

The probability density function, $f(t)$, of the Pearson Type III Distribution is:

$$f(t) = \frac{\lambda_p}{\Gamma(K)} [\lambda_p(t - h_{min})]^{K-1} e^{-\lambda_p(t-h_{min})}, \quad K, h_{min} \in R, \quad 5-5$$

where h_{min} is the minimum headway and K is the shape factor greater than 0 calculated as

$$K = \frac{h_{mean} - h_{min}}{h_{std}}, \quad 5-6$$

while λ_p is the flow rate parameter (and not the flow rate λ itself) estimated by

$$\lambda_p = \frac{K}{h_{mean} - h_{min}}, \quad 5-7$$

and $\Gamma(K)$ is the gamma function, as it follows:

$$\Gamma(K) = (K - 1)!. \quad 5-8$$

At high flows the headways assumed to follow Normal Distribution, where vehicles are expected to arrive close to each other in platoons [45]:

$$F(h) = p(t \leq h) \approx p_{table} \left(t \leq \frac{h - h_{mean}}{h_{std}^*} \right), \quad 5-9$$

in which h_{std}^* is a correction of the standard deviation of headways when the random variable t cannot be negative. This correction is given by



$$h_{std}^* = \frac{h_{mean} - h_{min}}{3}, \quad 5-10$$

where the 3 comes from the fact that if $h_{min} = h_{mean} - 3h_{std}$, which means that 99% of the headways will be greater than h_{min} .

When vehicles arrive from an upstream junction modelled by the same ego TC, it is not needed to make any statistical assumptions as the arrival time and speed is known and estimated by the *Trip Definition* sub-system.

5.3 Trip Definition

The modelling of a vehicle movement along the lane is done per green phase of the lane it is moving on. In other words, from the time the vehicle gets right-of-way until it either crosses the stop line or the green phase finishes. Here, right-of-way means when there is a green phase for a movement (signal) group of the lane and vehicles may go (the delay is due waiting for green and priority rules), or in the case of unsignalized junctions, there is a single “green” phase for all movement groups (they delay only due priority rules). The modelling for signalized and unsignalized is in the same way. The only difference is that at signalized junctions vehicles must wait the green signal before they move, while in the unsignalized this is not necessary. However, in both cases vehicles may have to yield to preferential traffic (e.g. when signalized and vehicles has minor green because it is left-turning or when unsignalized but vehicle must yield to a main road). Each generated vehicle has several attributes, a selection of the most important attributes is seen in Table 5-1.

Table 5-1: Selected attributes for a vehicle, i , on its j^{th} junction/edge/lane of its route.

Vehicle Attribute	Notation
Length	len_i
Minimum Gap	$mingap_i$
Arrival Time	$arrt_{i,j}$
Phase Start Time	$startt_{i,j}$
Phase Start Speed	$startv_{i,j}$
Phase Start Dist.	$startd_{i,j}$
Phase End Time	$endt_{i,j}$
Phase End Speed	$endv_{i,j}$
Phase End Dist.	$endd_{i,j}$
Lane Acceleration	$lacc_{i,j}$
Lane Deceleration	$ldecel_{i,j}$
Lane Desired Speed	$vdes_{i,j}$
Vehicle Length Crossed Time	$lent_{i,j}$
Vehicle Travel Time	$vehtt_{i,j}$
Vehicle Crossing	$vehc_{i,j}$

The minimum gap of vehicle i will stop from the vehicle in front of it is denoted $mingap_i$. The lane acceleration and deceleration of the vehicle adjusted to a specific edge, $lacc_{i,j}$ and $ldecel_{i,j}$, respectively. The lane desired speed is the expected maximum speed vehicle i will



try to achieve, i.e. $vdes_{i,j} = vfac_i \cdot lvmax_j$, in which $lvmax_j$ is the maximum allowed speed on edge/lane j . The vehicle length crossed time, $lent_{i,j}$, is the time in which the whole vehicle (its length, len_i) crosses the stop line, and it is calculated only when the vehicle will cross the stop line (given by the vehicle Boolean crossing flag variable $vehc_{i,j}$), while the vehicle travel time on the edge/lane is denoted $vehht_{i,j}$. The main attributes are the set of attributes phase start and phase end, representing the distance ($startd_{i,j}$ and $endd_{i,j}$), time ($startt_{i,j}$ and $endt_{i,j}$), and speed ($startv_{i,j}$ and $endv_{i,j}$) of a vehicle at the start and end of its movement during each analysing phase of its movement group on the lane, while $arrt_{i,j}$ is the arrival time. The analysed green phases are within the planning horizon and correspond to intervals vehicles will move as seen in Figure 5-5.

5.3.1 *Definition of Edge, Junction Vehicle Lane, Connection and Movement Group*

For non-CAV vehicles or CAV not sharing its intended route, the edge route (and consequently the junctions) is given by turning rates, but constrained to avoid loops and including certain probability to finish the trip in certain edge. For the lane choice (and consequently the connection and movement group), it is estimated which lane the vehicle will departure from the stop line. It is based on the next 2 edges the vehicle will take, and it assumes that vehicles will not stay in a long queue if they can take another which is shorter as well as go to a closed lane. It is important to say that the system doesn't model lane changes, it just put the vehicle on the lane that would be "better" for the vehicle.



5.3.2 Estimate the Departure of Discharging Vehicles

For each signal phase, it selects the vehicles that will move on the lanes that will get green and it will estimate the departure time from the stop line for each vehicle, as illustrated in Figure 5-8. The vehicles to be discharged are also ordered based on their distance to stop line (when already on lane) or arrival time on lane (if arriving vehicle). Certain vehicles may arrive on the lane before the end of the planning horizon but may not cross the stop line. When this happens, it is assumed the same timings of the last cycle during the modelled period to estimate such vehicle's departure. However, vehicles arriving after the planning horizon are not modelled.

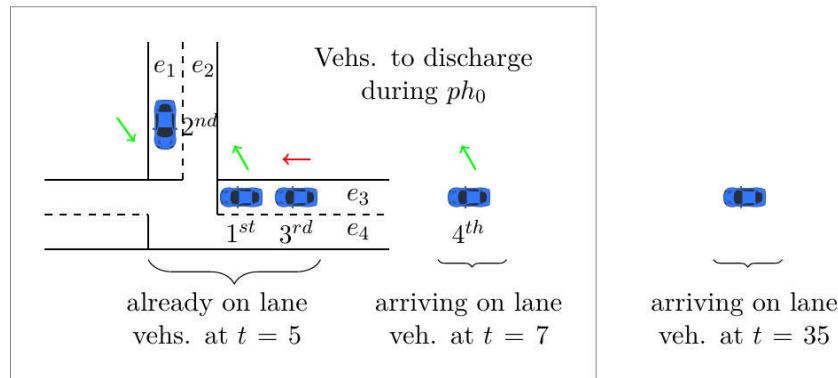


Figure 5-8 Example of selecting vehicles to discharge for green phase ph_0 . Vehicles already on lane will be discharged, as well as the earliest one arriving. The last arriving will not manage to arrive before the end of the green phase at $t = 20$.

Figure 5-9 illustrates how the system divides a real-world scenario, containing the events in which it is needed to estimate the state of each vehicle (time instant, distance to stop line and speed). The vehicle's key event attributes (1 up to 5) will be denoted as $evtd_{i,j}^e$ (distance to stop line in meters), $evtte_{i,j}^e$ (time instant in seconds) and $evtv_{i,j}^e$ (speed in m/s), where e stands for the event index, as event (0) correspond to the phase start attributes ($startt_{i,j}$, $startv_{i,j}$ and $startd_{i,j}$). One may notice that phase end attributes ($endt_{i,j}$, $endv_{i,j}$ and $endd_{i,j}$) represent values for event (4). If a vehicle will not cross the stop line, then its phase end values will be used as start values on the next green phase. Figure 5-9 also shows 4 main cases of such modelling described in the following list.

1. The 1st case is the vehicle departing earliest at stop line, and it represents a vehicle already stopped at its event (0) phase start movement, i.e $startv_{i,j} = 0$, that waits until it gets right-of-way.
2. The 2nd case is the second vehicle to depart at stop line, it is the movement when a vehicle must brake and completely stops until it gets green.
3. The 3rd case is similar to the second case (last vehicle to arrive at stop line), but on this one the vehicle slows down but don't stop completely. Notice that after short braking time, there is a displacement at constant speed due the reaction time of vehicles.
4. The 4th case is the only vehicle that didn't cross the stop line, it represents vehicles that their phase end values correspond to their final state and distance to stop line at the end of green time, opposite to other cases that have phase end values are assigned at the departure of the stop line.

The edge (and consequently its lanes) starts/end between the end of the junction's area, i.e. at the stop line for incoming edges (when exists, when not it depends where the intersection area starts) and after the pedestrian zebra (when exists, when not also where the intersection area ends). Therefore, between the edges there is the space of the junction, and it is accounted the time the vehicle will accelerate (or have constant movement if already at its lane desired speed) during the connection length from one lane to another. This is not



shown in Figure 5-9 because if a vehicle must yield to preferential traffic, its departure from stop line (event 4) is delayed, instead of modelling the movement inside the junction.

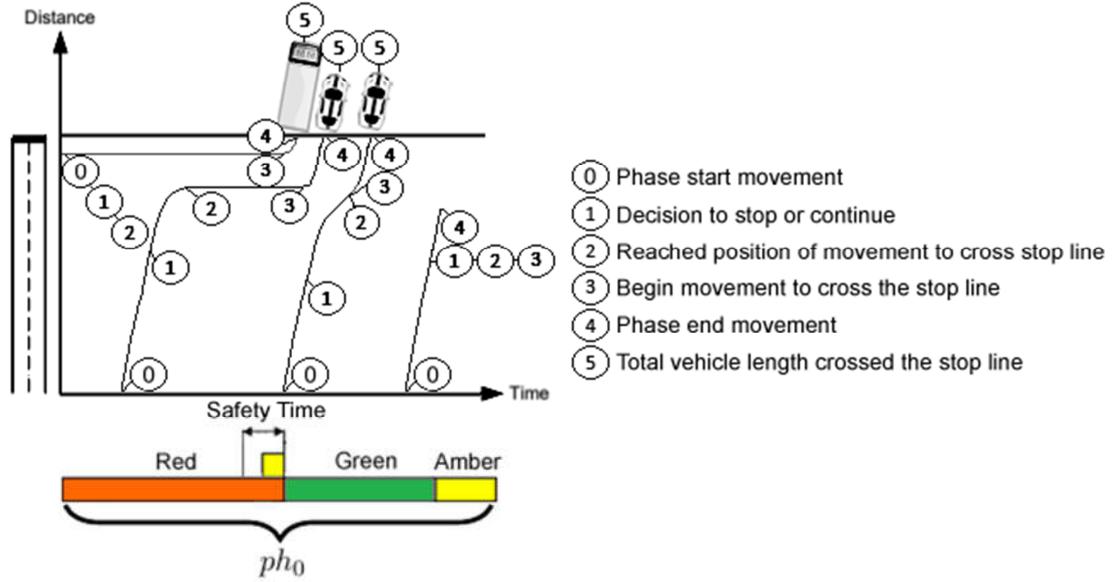


Figure 5-9 Key events to estimate vehicle state in 4 cases.

We use 3 sets of equations (acceleration, deceleration and threshold distance) to model the acceleration and deceleration between each event, in which the output of one event is the input of the next event. Based on the simplified Gipps' model presented in [46] we have the threshold distance from the final speed, $finalv$, of the accelerated and/or constant movement before start braking given the distance of acceleration Δd_{acc} , as it follows:

$$decelfinald_{i,j} = \begin{cases} evtd^e_{fronti,j} + len_{fronti} + mingap_i & \text{if } \exists vehc_{fronti,j} = 0 \\ mingap_i & \text{otherwise,} \end{cases} \quad 5-11$$

$$\Delta d_{acc} = \frac{vdes_{i,j} - startv_{i,j}^2}{2 \cdot lacc_{i,j}}, \quad 5-12$$

$$finalv = \sqrt{startv_{i,j}^2 + 2 \cdot lacc_{i,j} \cdot \Delta d_{acc}}, \quad 5-13$$

$$thresdfinalv = finalv \cdot react - \frac{finalv^2}{2 \cdot ldecel_{i,j}} + decelfinald_{i,j}, \quad 5-14$$

where $decelfinald_{i,j}$ is the deceleration minimum final distance (given by the existence of a vehicle in front, $fronti$, and that didn't cross the stop line), and $react$ is the reaction time. However, if this calculation leads to $startd_{i,j} < \Delta d_{acc} + thresdfinalv$, then vehicle is either braking already, or it will accelerate only to certain speed below its desired speed before start braking. For the former case, we have:

$$thresdinitv = -\frac{startv_{i,j}^2}{2 \cdot ldecel_{i,j}} + decelfinald_{i,j}. \quad 5-15$$

In the latter case, we need to calculate the distance of the acceleration constant movement (due reaction time) between acceleration and deceleration events given by:



$$\begin{aligned}
a &= 4 \cdot ldecel_{i,j}^2 \\
b &= -(2 \cdot lacc_{i,j} \cdot ldecel_{i,j}^2 \cdot react^2 \\
&\quad + 4 \cdot ldecel_{i,j}^2 \cdot startd_{i,j} \cdot 2 \cdot ldecel_{i,j} \\
&\quad + 4 \cdot ldecel_{i,j}) \\
c &= 4 \cdot startv_{i,j} - 4 \cdot ldecel_{i,j}^2 \cdot startd_{i,j}^2 \\
&\quad + 4 \cdot ldecel_{i,j} \cdot startd_{i,j} - 1 \\
\Delta d_{acconst} &= \max\left(\frac{-b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}, 0\right), \\
thresdperv &= \max(startd_{i,j} - \Delta d_{acconst}, decelfinald_{i,j}). \tag{5-16}
\end{aligned}$$

The threshold distance, which is the event (1), is calculated as:

$$vtd_{i,j}^1 = \begin{cases} thresdfinalv & \text{if } \Delta d_{acc} + thresdfinalv \leq startd_{i,j} \\ thresdinitv & \text{if } thresdinitv \geq startd_{i,j} \\ thresdperv & \text{otherwise.} \end{cases} \tag{5-17}$$

For the acceleration and deceleration movements we derive two different sets of equations from the fundamental equations of constant translational acceleration in a straight line from Physics. One set when vehicle i will be accelerating between key events e and $(e - 1)$ given by Equations 5-21, 5-26 and 5-27), constrained by the maximum displacement, $maxd$, and the available time for the accelerate $maxt_{acc}$:

$$maxd = evtv_{i,j}^{e-1} - accfinald_{i,j}, \tag{5-18}$$

$$maxt_{acc} = \min\left\{\frac{vdes_{i,j} - evtv_{i,j}^{e-1}}{lacc_{i,j}}, acc_{exp}\right\}, \tag{5-19}$$

in which $accfinald_{i,j}$ is the acceleration minimum final distance and acc_{exp} is the expected time for the acceleration. Both variates according to the event number and the context. The distance of accelerated movement, Δd_{acc} , also depends on the difference of the speed at the beginning of event e , $evt v_{i,j}^{e-1}$, and the desired speed $vdes_{i,j}$:

$$\begin{aligned}
\Delta d_{acc} = \min\{ & \frac{vdes_{i,j}^2 - (evt v_{i,j}^{e-1})^2}{2 \cdot lacc_{i,j}}, maxd, \\
& evt v_{i,j}^{e-1} \cdot maxt_{acc} + \frac{1}{2} \cdot lacc_{i,j} \cdot (maxt_{acc})^2 \}, \tag{5-20}
\end{aligned}$$

what give us the speed at the end of event e , $evt v_{i,j}^e$ and the time spent accelerating Δt_{acc} ,

$$evt v_{i,j}^e = \sqrt{(evt v_{i,j}^{e-1})^2 + 2 \cdot lacc_{i,j} \cdot \Delta d_{acc}}, \tag{5-21}$$

$$\Delta t_{acc} = \frac{evt v_{i,j}^e}{lacc_{i,j}}. \tag{5-22}$$

Consequently, the maximum time for the constant movement, $maxt_{const}$, as well as the distance and time of the constant movement, Δd_{const} and Δt_{const} respectively:

$$maxt_{const} = acc_{exp} - \Delta t_{acc}, \tag{5-23}$$

$$\Delta d_{const} = \min\{maxd - \Delta d_{acc}, maxt_{const} \cdot vdes_{i,j}\}, \tag{5-24}$$



$$\Delta t_{const} = \begin{cases} \frac{\Delta d_{const}}{evtv_{i,j}^e} & \text{if } evtv_{i,j}^e > 0 \\ 0 & \text{otherwise.} \end{cases} \quad 5-25$$

Finally, the distance, $evtd_{i,j}^e$, and time, $evtt_{i,j}^e$, of event e is given by:

$$evtt_{i,j}^e = evtt_{i,j}^{e-1} + \Delta t_{acc} + \Delta t_{const}, \quad 5-26$$

$$evtd_{i,j}^e = maxd - \Delta d_{acc} - \Delta d_{const}. \quad 5-27$$

For the set with deceleration, we have Equations 5-29, 5-30 and 5-31. The deceleration minimum final distance, $decelfinald_{i,j}$, is given in Equation 5-11, while the decelerating time Δt_{decel} is constrained by the expected time for the deceleration, $decelt_{exp}$, reaction time $react$ and the time needed to stop completely:

$$decel = \begin{cases} \min(decelt_{exp} - react, -\frac{evtt_{i,j}^{e-1}}{ldecel_{i,j}}) & \text{if } decelt_{exp} \geq react \\ 0 & \text{otherwise.} \end{cases} \quad 5-28$$

The time of the event after decelerating, $evtt_{i,j}^e$, is the minimum between the beginning until the expected time for the deceleration and the end of decelerating time, while the speed $evtv_{i,j}^e$ based on this elapsed time:

$$evtt_{i,j}^e = \min\{evtt_{i,j}^{e-1} + react + \Delta t_{decel}, evtt_{i,j}^{e-1} + decelt_{exp}\}, \quad 5-29$$

$$evtv_{i,j}^e = evtv_{i,j}^{e-1} + ldecel_{i,j} \cdot \Delta t_{decel}. \quad 5-30$$

For the distance of the event $evtd_{i,j}^e$, it depends on the expected time for the deceleration ($decelt_{exp}$) and it is calculated in three different ways as the calculations always add the movement during reaction time:

$$evtd_{i,j}^e = \begin{cases} decelfinalt & \text{if } decelt_{exp} \geq react \\ deceldpartt & \text{if } 0 < decelt_{exp} < react \\ evtd_{i,j}^{e-1} & \text{otherwise,} \end{cases} \quad 5-31$$

where

$$decelfinalt = \max\{ evtd_{i,j}^{e-1} - [evtv_{i,j}^{e-1} \cdot react + \frac{(evtv_{i,j}^e)^2 - (evtv_{i,j}^{e-1})^2}{2 \cdot ldecel_{i,j}}], decelfinald_{i,j} \}, \quad 5-32$$

$$deceldpartt = \max\{ evtd_{i,j}^{e-1} - [evtv_{i,j}^{e-1} \cdot (evtt_{i,j}^e - evtt_{i,j}^{e-1})], decelfinald_{i,j} \}. \quad 5-33$$

Once a vehicle is stopped, its rear distance to stop line is accounted on the predicted queue length on the lane. After the estimation of each event, the system also flags vehicles that are expected to be on lane at the next algorithm update. Additionally, it corrects the event values for vehicles that would move too early compared to the vehicle in front or the beginning of green time. Moreover, vehicles may have their departure from stop line delayed due to a predicted full queue length on their next lane. Another reason for delay is yielding to



preferential traffic, in which if vehicle's event (4) is within t_{max} then it uses event (5) time of preferential traffic that they must yield (considering their connection critical gap time $avec_{conn}$ and follow-up time fup_{conn}), otherwise it uses the connection delay, $pdt_{conn,ph}$, given the departures within t_{max} .

5.3.3 Update Connections' Capacity and Delay

This process estimates the delay, $pdt_{conn,ph}$, that a vehicle will experience when taking a connection based on the connection's capacity of discharge during certain green phase, ph , of its mov. group, mg . The capacity, $cap_{conn,ph}$, is calculated in three different ways according to the system-level flow thresholds, and the flow of vehicles departing on a lane to yield yl for its mov. group, $ylmg$ at same phase, denoted $\lambda_{yl,ylmg,ph}^{deps}$, that is given by the number of vehicles departing, $numdeps_{yl,ylmg,ph}$, in the list $deps_{yl,ylmg,ph}$, as it follows:

$$\lambda_{yl,ylmg,ph}^{deps} = \frac{numdeps_{yl,ylmg,ph}}{phendt_{ph} - phbegt_{ph}}, \quad 5-34$$

in which $phendt_{ph}$ is the phase end time and $phbegt_{ph}$ the begin time.

If $\lambda_{yl,ylmg,ph}^{deps}$ is low flow, the dynamic capacity on connection $conn$ assumes *Negative Exponential Distribution* of the headways on the preferential traffic [47]:

$$cap_{conn,ph} = \frac{\lambda_{yl,ylmg,ph}^{deps} \cdot e^{-\lambda_{yl,ylmg,ph}^{deps} \cdot avec_{conn}}}{1 - e^{-\lambda_{yl,ylmg,ph}^{deps} \cdot fup_{conn}}}, \quad 5-35$$

where $avec_{conn}$ is the critical gap time and fup_{conn} is the follow-up time.

If $\lambda_{yl,ylmg,ph}^{deps}$ is mid flow, then it is assumed a *Dichotomized Distribution* (some vehicles in platoon and some not) of the arrivals on the preferential traffic, which is also used to estimate the capacity for non-signalized junctions [47], what give us:

$$cap_{conn,ph} = (1 - \lambda_{yl,ylmg,ph}^{deps} \cdot h_{min}) \cdot \frac{\lambda_{yl,ylmg,ph}^{deps} \cdot e^{-\lambda_{yl,ylmg,ph}^{deps} \cdot (avec_{conn} - h_{min})}}{1 - e^{-\lambda_{yl,ylmg,ph}^{deps} \cdot fup_{conn}}}. \quad 5-36$$

In both cases, that this capacity cannot be higher than the fixed capacity of the connection, $cap_{conn,sf}$, which represents its saturation flow. This connection delay is constrained by its capacity $cap_{conn,ph}$ and given by:

$$pdt_{i,conn,ph}^{lowmid} = \max\{conndt - \max\{slarr_i - refhdwy, 0\}, 0\}, \quad 5-37$$

$$pdt_{i,conn,ph}^{lowmid} = \min\{pdt_{i,conn,ph}^{lowmid}, phendt_{ph} - slarr_i\}, \quad phendt_{ph} > slarr_i, \quad 5-38$$

where $conndt$ is delay without considering the time interval between arrivals (or begin green time and actual arrival of vehicle), the stop line arrival time $slarr$, and the reference time for headway $refhdwy$. They are calculated as it follows:

$$conndt = \begin{cases} 1/cap_{i,conn,ph} & \text{if } cap_{conn,ph} > 0 \\ \max\{phendt_{ph} - slarr_i, 0\} & \text{if } cap_{conn,ph} = 0, \end{cases} \quad 5-39$$



$$refhdwy = \begin{cases} slarr_{fronti} & \text{if } \exists fronti \\ phbegt_{ph} & \text{if signalized junction} \\ slarr_i & \text{otherwise,} \end{cases} \quad 5-40$$

in which Equation 5-38 is valid only if this process is called by the *Estimate the Departure of Discharging Vehicles* process, from where *fronti* is the vehicle in front.

In case $\lambda_{yl,ylmg,ph}^{deps}$ is high flow, then there is too much traffic on the preferential lanes. It is expected the vehicle may cross only after the last one on the preferential traffic crosses.

5.4 Estimation of Travel Times

The travel times are estimated for the modelled edges by the Traffic Controller (TC) per range (time window), rge , of the division of the planning horizon t_{max} . The system calculates travel times at the precision (time resolution) of each range for each movement group per lane but only the average for the edges is shared. The format of the broadcasted messages to CAVs is seen in Section 5.5, and it is the same format which is exchanged between TCs.

5.4.1 Travel Times Using Probe Vehicles

In general, a probe vehicle is a vehicle that has an arrival time on the lane. For example, a newly generated queueing vehicle on certain lane doesn't have arrival time (on the beginning of the lane) as it is generated somewhere along the lane. Those probe vehicles are used to estimate the travel time based on their arrival time on lane, as it follows:

$$vehtt_{i,j} = endt_{i,j} - arrt_{i,j}. \quad 5-41$$

The estimation using probe vehicles takes advantage of the fact that vehicles are modelled from their arrival on the edge until their departure time. Therefore, travel time on edge, k , for range, rge , denoted as $edgett_{k,rge}$ is given by:

$$edgett_{k,rge} = \frac{\sum vehtt_{i,j}}{numprobes_{l,mg,rge}} \quad \begin{array}{l} \forall i \in probes_{l,mg,rge}, \\ \forall l \in lanes_{e_k} \text{ and} \\ \forall mg \in movgs_{e_k}, \end{array} \quad 5-42$$

where l is the index of the lanes of edge k and mg the mov. group, $lanes_{e_k}$ is the list of edge e_k lanes and $movgs_{e_k}$ its mov. groups, while $numprobes_{l,mg,rge}$ is the number of probes. The probe vehicle, i , is in $probes_{l,mg,rge}$, which correspond to its j^{th} junction/edge/lane of its route and edge k .



5.4.2 Travel Times Using Queueing Delay Model

There is an option in which the system can generate artificial probe vehicles for each mov. group that didn't have itself assigned to a modelled vehicle. However, this considerably increases the number of calculations and slow down the system, what is still acceptable when the traffic controller doesn't control too many junctions and/or model too many vehicles. When probe vehicles are not assigned to a certain range rge of an edge, then a deterministic queueing model is used [47]. The choice of this model is due the fact that as we model vehicle arrivals and departures, as well as the initial queue is also modelled by the vehicles already on lane, we use our traffic theoretic model as input for this queueing model, in which:

$$q(t) = q(actualt) + A(t) - D(t), \quad 5-43$$

$$adpv(rge) = \frac{1}{A(rgendt_{rge})} \int_{actualt}^{rgendt_{rge}} q(t) dt, \quad 5-44$$

Where $q(t)$ is the predicted queue length at time instant t ; $rgendt_{rge}$ is the end time of the range rge being estimated the travel time; $A(t)$ is the cumulative number of arrivals from the current time of updating the algorithm, $actualt$ until t ; $D(t)$ is the departures under continuous presence of vehicle queue from the current time of updating the algorithm $actualt$ until t ; and $adpv(rge)$ is the average delay of vehicles queuing during the time period $[actualt, rgendt_{rge}]$.

The model needs the "arrivals" and "departures" from the stop line what we don't really have in some cases if the vehicle is not a probe vehicle and it couldn't depart from stop line before the end of the planning horizon. To solve this, we look into how it models the queue. Vehicles are considered to stack at the stop line, the so called vertical queue. Therefore, the arrival time means the time vehicle would arrive at stop line without any influence of vehicle in front or signal plan, $qmodelarr_{i,j}$, given by:

$$\begin{cases} qmodelarr_{i,j} = arrt_{i,j} + \frac{l\!len_l}{(l\!maxv_l \cdot vfac_i)} & \text{if } \exists arrt_{i,j} \\ qmodelarr_{i,j} = nupt_{i,j} + \frac{nupd_{i,j}}{(l\!maxv_l \cdot vfac_i)} & \text{otherwise,} \end{cases} \quad 5-45$$

where $arrt_{i,j}$ is vehicle's arrival time on the edge (entering it), $nupd_{i,j}$ is vehicle's distance when being generated (as $startd_{i,j}$ is updated with new values), $llen_l$ is the lane length (in metres), $l\!maxv_l$ the lane maximum speed, and $vfac_i$ vehicle i speed factor. The vehicle i is classified as part of the initial queue, $q(actualt)$, if $qmodelarr_{i,j} \leq actualt$, otherwise it is assigned to the respective arrival on range, rge , when $qmodelarr_{i,j} \geq rgstartt_{rge}$ and $qmodelarr_{i,j} < rgstartt_{rge+1}$, in which $rgstartt_{rge}$ is the begin time of the range rge . For the departures, the system accounts vehicle departure only if the vehicle crosses the stop line, the value used is the phase end time $endt_{i,j} = evt_{i,j}^4$.

Equation 5-44 only estimates the delay due to the queue, we still need to calculate the delay due red light for a vehicle arriving on the edge at the begin time of the range rge , denoted $rgstartt_{rge}$, and also the connection delay when yielding to preferential traffic. As the idea is to estimate the travel time for each range but there is no probe vehicle, what is possible is to estimate these delays if a vehicle would enter the edge at begin time of the range, $rgstartt_{rge}$. In this way, we need to calculate the travel time without any delays, $llentt_l$, in order to find the arrival time at the vertical queue, $vtarr_{l,rge}$:



$$llentt_l = \frac{l\text{len}_l}{l\text{vmax}_l}, \quad 5-46$$

$$vtarr_{l,rge} = rgstartt_{rge} + llentt_l. \quad 5-47$$

For the traffic light delay, $tsdt_{l,mg,ph}$, we need to find a suitable phase, ph , which is the first phase the vehicle may cross and it is based on the arrival time at stop line:

$$tsdt_{l,mg,ph} = \max\{begint_{mg,ph} - vtarr_{l,rge}, 0\}. \quad 5-48$$

For the calculation of the queue delay, it is necessary to know the queue at the time a vehicle arriving at $rgstartt_{rge}$ would experience. This is given by a suitable range, $rgevtar$, that is the first possible range of the arrival time at stop line. Finally, the travel time for the movement group mg of lane l at range rge is estimated by:

$$lanemovtt_{l,mg,rge} = llentt_l + tsdt_{l,mg,ph} + adq(rge) + pdt_{conn,ph}, \quad 5-49$$

where $adq(rge) = adpv(rge) \cdot q(rgevtar)$ represents the average queue delay at rge , while $pdt_{conn,ph}$ is the connection delay using $slarr = vtarr_{l,rge} + adq(rge)$. The travel time on the edge, $edgett_{k,rge}$, is the average travel time for all mov. groups of the lanes belonging to the edge k .

5.5 Traffic Controllers Knowledge Sharing

This sub-system is responsible for the definition of the content and format of the messages exchanged between traffic controllers as it is required an interface for the real exchange of messages. Figure 5-10 illustrates the format of the exchanged messages. The information is valid from certain initial time (*actualt* corresponding to the time the algorithm updated) and specific for each range (time window), rge , of the planning horizon t_{max} . As the length of the ranges, $lenrge$, is fixed for the whole system we only need to know from which time the received information is valid (t_{max} is not necessary). Notice that the messages are junction-based in order to reference which TC border junction would need to send information to which TC border junction:

- source junction ID (*jctID*) which is the identification code of the source junction within the traffic network that send a message;
- destination junction ID (*jctID*) which is the identification code of the destination junction within the traffic network.



Source Junction: $jctID \Rightarrow$ Dest. Junction: $jctID$

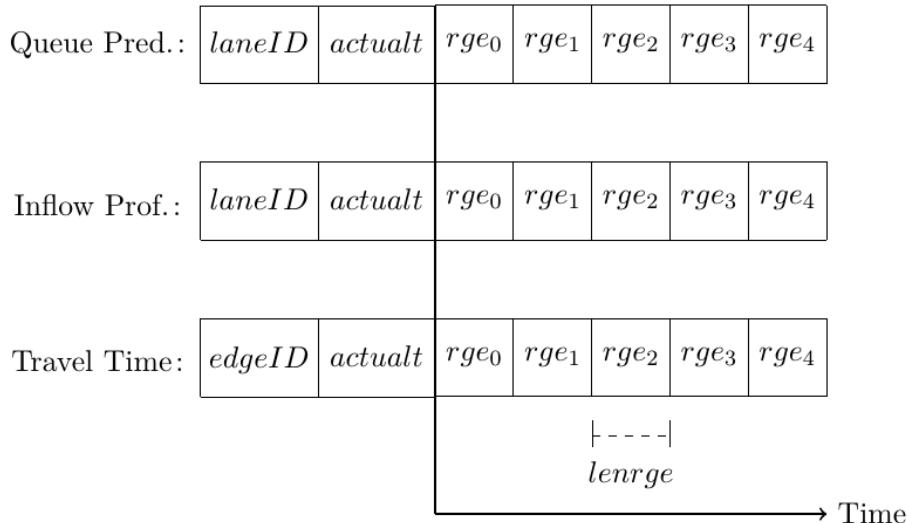


Figure 5-10: Format of exchanged messages.

The **lane inflow profile** messages are exchanged at each algorithm iteration (twice per update). The information to be shared is the parameters of headways distribution functions:

- lane network identification (*laneID*);
- number of vehicles;
- flow (veh/s);
- mean speed (m/s);
- mean of headways (s); and
- standard deviation of headways.

The **lane queue length prediction** is sent only once at the second iteration (i.e $tcit = 2$) as it will be useful only on the next algorithm update. Notice that this lane queue length prediction is based on estimation of the LLR system and not the proposed queuing model in Section 4. The needed data is the rear distance of the last stopped vehicle on each range throughout the planning horizon stored by the *Trip Definition* sub-system:

- lane network identification (*laneID*); and
- queue length (m).

The **edge travel times** (TTs) message must be multicast to all traffic controllers connected on the same Traffic Network (TN) and it is sent only at the second iteration (i.e $tcit = 2$):

- edge network identification (*edgeID*); and
- travel time (s).



5.6 Conclusion and further research

The Local Level Routing (LLR) algorithm is currently functional and tested in a real-world network of the Luxembourg City with typical European mobility patterns. The LLR system enables vehicles (with a destination outside the area modelled by the system) to find the optimal route considering nearby more accurate predictions and faraway stored information, avoiding route advice to an unnecessary local destination. In addition, the system can work with simultaneous entrance points where vehicles would start receiving real-time information accounting vehicles that have passed any entrance points. This leads to vehicles being responsible to do the routing, while sharing their intended route benefit the accuracy of the traffic predictions but it is not necessarily a requirement. These predictions are based on modelling vehicles microscopically within the road and statistically (but maintaining the number of vehicles) between junctions throughout the planning horizon. The travel time estimation is specific for each time window/range (division of the planning horizon) and considers the available information, even when there is or there will be no vehicle on the lane, as well as the waiting due to yielding to preferential traffic. The following list presents a few preliminary findings.

- One important issue is that benefits are realised when there is a time interval between reroutes, i.e. recommended from 180 up to 300 s for the Luxembourg City network [48], and vehicles should reroute at different timings. The reason is due to unreliable predictions caused by too short intervals that worsen the performance [49], [50]) and many vehicles rerouting at the same time.
- The implementation of the system in an infrastructure composed by connected TCs that exchange messages between each other allows the system to be flexible as totally distributed (e.g. one per Cooperative Intersection) or more centralized (e.g. one Cooperative Intersection modelling for all junctions connected for the LLR system) according to the required running time and communication restrictions.

The initial qualitative results of the LLR system are promising. The system will be further evaluated in WP7, where it will be assessed in a network of Prague and the results will include the quantitative performance of the system related to mobility and emissions.



6 Actuated traffic control with green light optimal speed advisory

6.1 AGLOSA on a single intersection

The AGLOSA algorithm is implemented in the programming language Python and is coupled with the traffic simulator SUMO. The main concept of this algorithm is the on-line calculation of an optimal signal plan given a traffic demand coupled with a GLOSA approach. In brief, the algorithm predicts the arrival times of all detected vehicles at their stop lines and calculates an optimal sequence of phases according to this arrival scheme and a given optimization goal (e.g. number of stops, time loss). This optimization is done in a traffic flow simulation.

The AGLOSA algorithm is presented in detail in [27]. Here, we only give a short overview.

AGLOSA is a cooperative algorithm that depends on information exchange between the infrastructure (traffic light) and the vehicles at an intersection. Vehicle information typically contains things like an identifier, a position and a speed, whereas infrastructure information contains switching times. The problem is to find an optimal sequence of phases that minimizes an optimization goal (here: time loss) for the given traffic demand.

The configuration data needed for its calculations is:

- For each approach, static information:
 - Maximum allowed speed boundary, e.g. 50 km/h
 - Minimum recommended speed boundary, e.g. 30 km/h
 - Incremental step size for speed recommendations, e.g. 5 km/h
 - Maximum covered distance from stop line for GLOSA speed recommendations, e.g. 500 m, also depending on distance to neighbouring intersections

The algorithm uses dynamic programming for the solution of the optimization problem. More specifically, the solution to the optimization problem must satisfy Bellman's principle of optimality. As a reminder:

"An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." [26]

This means, an optimal solution found for a transition between two phases must also be optimal for the transition to a following third phase. If this second transition is not optimal, then there must also be a better solution for the first one.

The first step of the AGLOSA algorithm is that positions from all detected vehicles are received. The possible arrival times for each vehicle are calculated. In the second step, it is determined what the switching times are and what switching times should be assigned to each vehicle to minimize the overall time loss. The time loss for a vehicle is determined as the difference $t_{actual} - t_{min}$, where t_{min} is the earliest possible time for passing the intersection and t_{actual} is the actual passing time. At this point, it is possible to determine specific weights for different vehicle types, e.g. busses or HGVs. This enables the pre-emption of public transport.

This problem is then decomposed into repeating sub problems for all possible future states. To find the next optimal state, we need to keep track of the time loss, the minimum remaining time until the next switch. The storage of this additional information is needed because there might be two competing solutions where one has less time loss but the other one has less switch delay and leads to a better overall solution.



When the optimal switching times are calculated, they are sent to the vehicles which in consequence adapt to the given speed advice.

Possible actions of AGLOSA on the signal plan can be:

- Extension of phase durations up to a maximum duration,
- Premature phase abortion,
- Phase skipping due to lack of demand.

The big advantages of AGLOSA are that it is able to process C2X information and that it can adapt to all traffic situations. A shortcoming of several conventional control approaches is that they work well as long as the vehicles arrive in “typical” patterns and perform poorly in unusual situations, such as induced traffic due to an event. AGLOSA on the other hand, is able to handle these situations.

6.1.1 C2X interfaces

The AGLOSA algorithm is designed to handle generic detector input. This means, it can process conventional induction loop detectors but should optimally be operated with C2X messages. The image of the actual traffic at an intersection can only be approximated through induction loops since we can only measure the time of detection, but we do not know how the vehicle is approaching the intersection from there. This problem could be avoided with C2X messages since the vehicles could be continuously detected.

In foregone projects, the detection of conventional vehicles through induction loops has already been tested. In the context of the project MAVEN, an interface for handling incoming Cooperative Awareness Messages (CAM) was developed. This interface listens to incoming messages and processes each CAM into a simulated vehicle. The insertion is realized through SUMO’s TraCI interface.¹ Since CAM messages are transmitted more frequently than once a second (usual simulation step size in SUMO), the position of the simulated vehicle can continuously be corrected using the received positioning data. If no CAM should be available in any time step, the next vehicle position can be approximated by the simulation. This fall back makes the approach very robust.

The next interface that was created in the MAVEN context was a Signal Phase and Timing (SPaT) message generator. Since all necessary information for SPaT (e.g. next switches, current signal state) are generated by the AGLOSA optimization, it can be directly written into a message. This message informs approaching C2X enabled vehicles of imminent switching events at their next intersection. Apart from the switches, a SPaT contains GLOSA information. This information consists of optimal approaching speeds depending on the distance from the stop line. The examples in Figure 6-1 and Figure 6-2 illustrate the functionality of GLOSA more in depth.

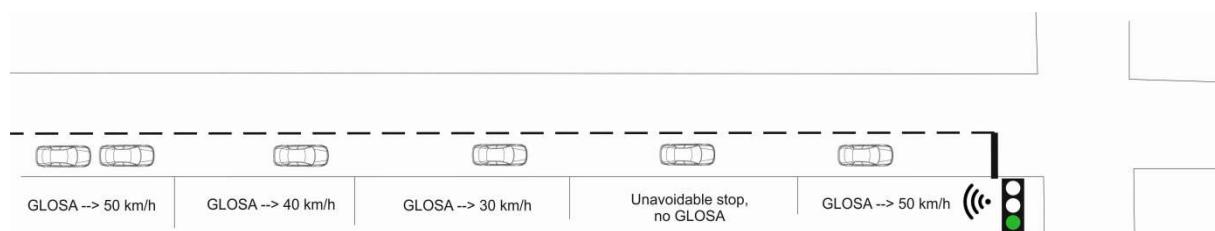


Figure 6-1: GLOSA speed advice zone concept

¹ TraCI: Traffic Control Interface (see <http://sumo.dlr.de/wiki/TraCI>)



The road is separated into several sections with distinct speed recommendations. The vehicles that are located in these sections are all approaching the traffic light. They all receive speed advisories based on the switching times transmitted by the signal controller. The speed advisories correspond to the caption inside the respective section. The recommendations for the approach speeds vary between the minimum speed and the maximum allowed speed boundary, incrementing with the defined step size. Speed recommendations below the minimum speed boundary indicate that a stop is unavoidable.

The last interface implemented in MAVEN was an interface that sends Lane Advice Messages (LAM) from the infrastructure to vehicles. These messages are recommendations of lane changes to C2X enabled vehicles. There are various possible reasons for a lane change, in this context we only consider the queueing of vehicles on the current lane. This means, a lane change is favourable if there is a queue of vehicles on its current lane and at least one lane exists that has none or a shorter queue and also leads to the vehicle's destination.

As aforementioned, conventional vehicles can be detected through induction loops and fed into SUMO and the AGLOSA optimization. The fact that conventional vehicles are put into the simulation was exploited since it enables us to directly measure the queue length on all lanes. In the MAVEN project, other queue length measuring and estimation approaches will be investigated.

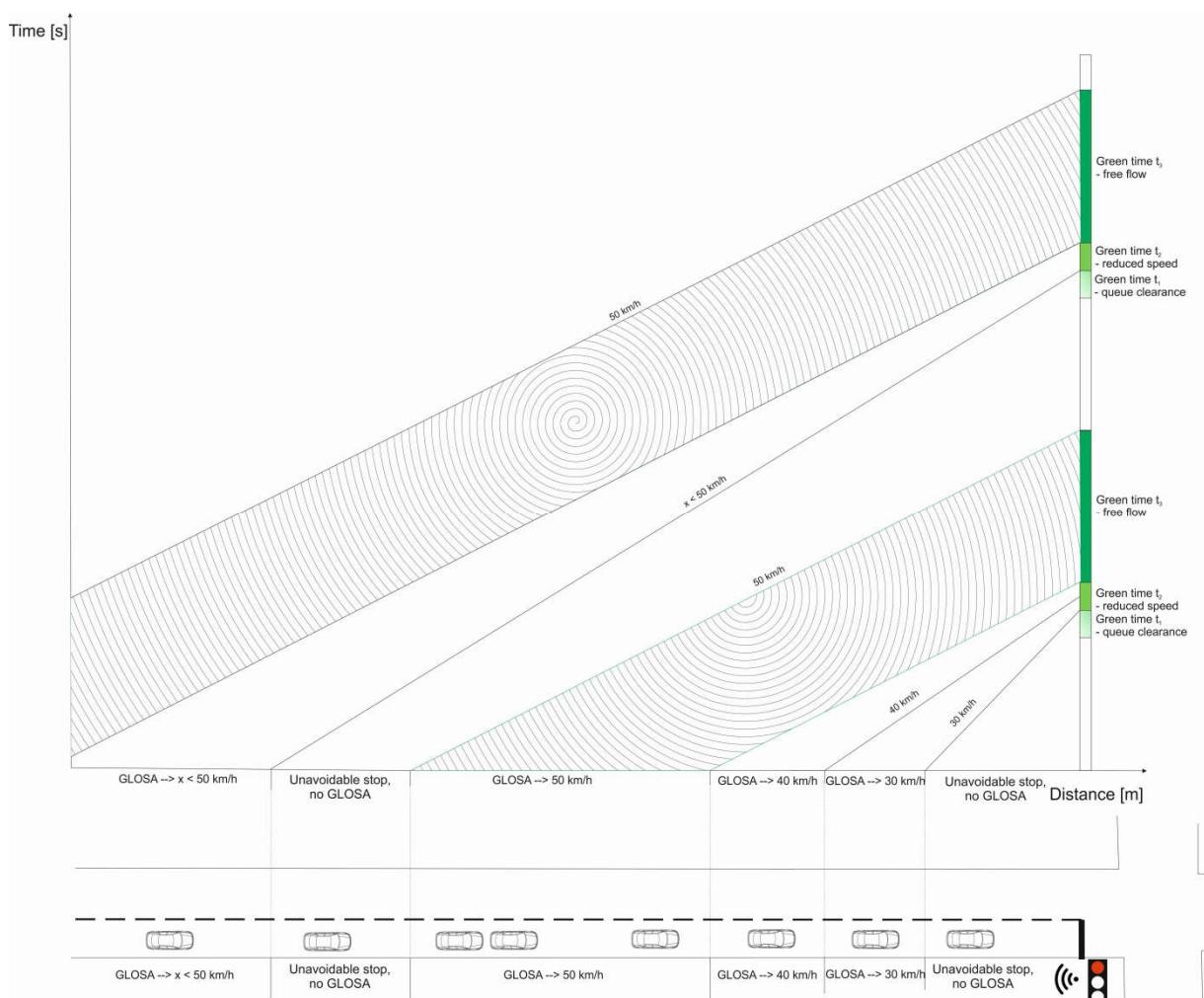


Figure 6-2: GLOSA working principle



6.1.2 Queue length estimation

In order to further smooth the traffic flow, queue length estimation will be used to clear the line of waiting vehicles at the beginning of a green phase before the platoon of cooperative vehicles arrives. The GLOSA recommendations always refer to this moment of clearance. It also takes into account the vehicles in the arrival pattern (see Figure 6-2).

To estimate the queue length at a traffic light we propose the method described in [26], which will be briefly described in the following. The principle of this estimation method is depicted in Figure 6-3.

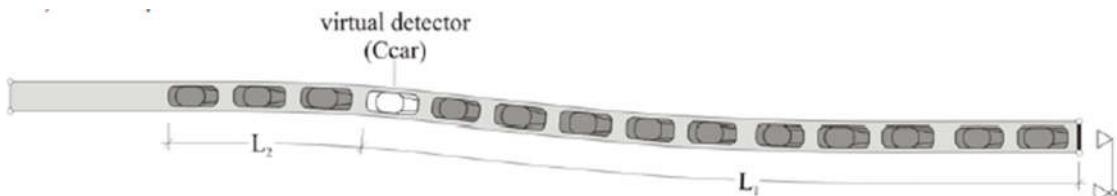


Figure 6-3: Queue length estimation principle, Source: [26]

We have a queue of vehicles waiting in front of a red traffic light. The total length of the queue is L , composed of a measurable length L_1 from the stop line to a virtual detector and a length L_2 from a virtual detector to the end of the queue that has to be approximated. The virtual detector is any vehicle that has vehicle-to-infrastructure (V2I) communication technology on-board. Its communication devices transmit its position and its speed to the traffic light controller.

As soon as the detection vehicle stops at a red light, we can measure the length L_1 given the detector's position and determine the number of vehicles waiting in front of it.

The approximation of L_2 will be described in the following. Equation 6-1 states that the expectation value of the maximum queue length of L_2 is equal to:

$$E(\max L_2) = \lambda_2 * E(\max w_2) \quad 6-1$$

Where λ_2 is the arrival rate of the vehicles within L_2 and $E(\max w_2)$ is the expectation value of the maximum waiting time. The arrival rate of vehicles within one red phase is assumed to be constant which means, its estimate results in:

$$E(\lambda_2) = \lambda_1 = \frac{d_v}{\Delta t} \quad 6-2$$

This means, the arrival rate is simply the distance between detector and stop line divided by the time interval between the arrival of the detector and the begin of the red time period.

The last value we need is the expectation value for the maximum waiting time. This value is composed of the remaining red time t_{rs} and the blocking time $t_b(d)$, the service time of the vehicles in front of the last vehicle in the queue, or:

$$E(\max w_2) = t_{rs} + t_b(d) \quad 6-3$$



The blocking time is, of course, dependant on the distance between the last vehicle and the stop line.

This method will be adapted in the MAVEN context to match the specific use cases and test areas. For the positioning of the detector vehicles, a high precision positing tool (e.g. DGPS) has to be used to enable the assignment of detector vehicles to single lanes. This might be important for the estimation of queue lengths for different OD relations at an intersection or if more than one lane exists for one or more directions.

6.1.3 Results

The application of AGLOSA on single intersections has been evaluated in the traffic flow simulation SUMO. The following results were computed for the intersection shown in Figure 6-4.

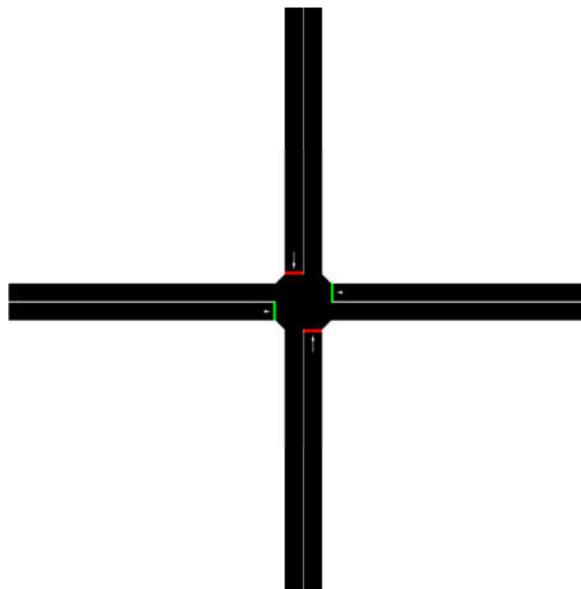


Figure 6-4: Basic scenario intersection

Vehicles can only pass straight over this basic intersection. The traffic light program consists of only two phases. The simulation was run with two different traffic demands (see Table 6-1) and two different control methods, a fixed signal plan and AGLOSA.

Table 6-1: Hourly traffic demands for the basic intersection scenario

	Horizontal demand (west- / eastbound)	Vertical demand (north- / southbound)
Demand 1	500	200
Demand 2	500	500

The results for the simulation runs shown in Figure 6-5 point out two things in particular. First, the AGLOSA algorithm is able to reduce time loss and emissions at intersections compared to a fixed signal plan. The second point is that AGLOSA works better with asymmetrical



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

demands. This is immediately clear if we imagine our basic intersection with equally high vehicle arrival rates on all approaches. As soon as one direction is given green, the vehicles in the other direction accumulate time loss. This means, the controller wants to switch to the other phase where the same effect happens. Thus, the controller would jump back and forth between the phases and – given a high enough traffic demand – eventually converge to a quasi-fixed signal plan.

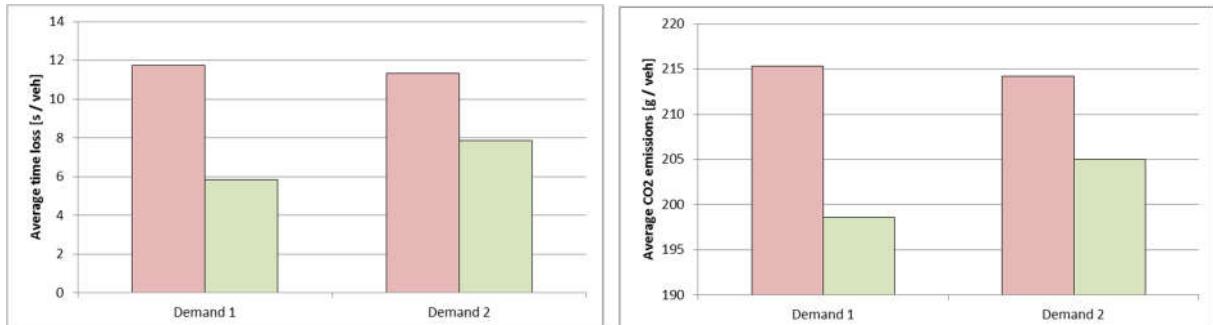


Figure 6-5: Results for the basic intersection scenario (red: fixed signal plan, green: AGLOSA). Left: average time loss, right: average CO₂ emissions

Of course, the basic intersection scenario is not a realistic one, but it sure is illustrative. To show the effects of the AGLOSA algorithm on the traffic flow of a more realistic intersection, we introduce a second scenario (see Figure 6-6).

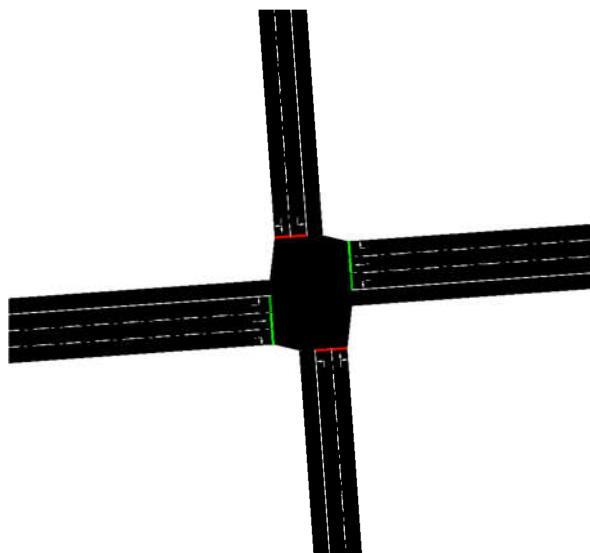


Figure 6-6: Scenario 2 intersection

The topology of this intersection is more realistic since there are directional lanes and it has a clear main traffic flow direction. The signal plan has distinct phases for straight and right-turning drivers and the left-turning traffic. The simulation is run for three different traffic demands (see Table 6-2). The turning ratios were assumed to be 15% left-turning and 30% right-turning. Like the basic scenario, both fixed signal plan and AGLOSA are used as control methods.

Table 6-2: Hourly traffic demands for scenario 2

	Horizontal demand (west- / eastbound)	Vertical demand (north- / southbound)
Demand 1	500	500



Demand 2	1000	500
Demand 3	2000	500

Like in the basic scenario, the AGLOSA algorithm is able to reduce both time loss and emissions at the intersection compared to the fixed signal plan. Even under dense traffic conditions of "Demand 3", its performance is slightly better.

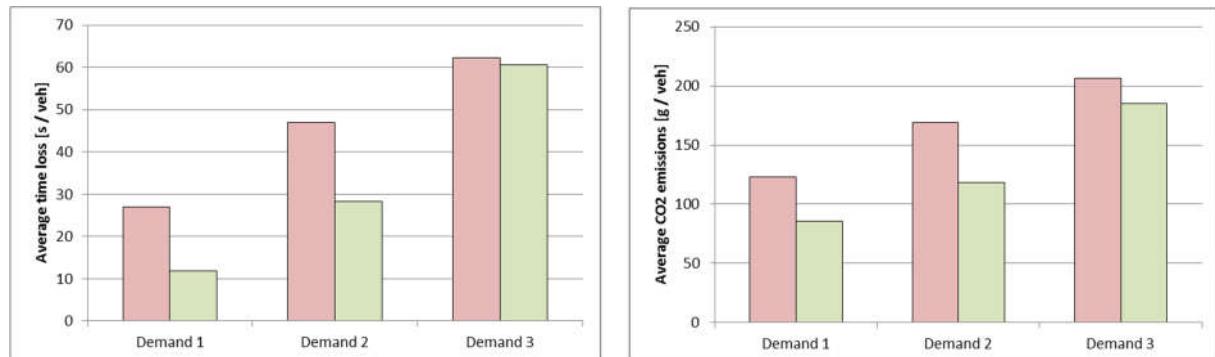


Figure 6-7: Results for scenario 2 (red: fixed signal plan, green: AGLOSA). Left: average time loss, right: average CO₂ emissions

6.2 Multi-Intersection optimization AGLOSA

The next step in the development of AGLOSA is to test it in a network where more than one intersection is controlled by this method. Markowski investigated this topic and found that the plain, single-intersection AGLOSA algorithm is capable of controlling more than one intersection efficiently. The vehicles passing these intersections experience a quasi-coordination of the green times which leads to reduced number of stops and lower time loss. There are moments when the coordination fails due to higher traffic demand in a conflicting direction, but if there is a clear main traffic flow direction, these moments should be rare. Also, this coordination effect is achieved without any communication between the traffic signals. [28]

Since the possibilities of coordination are limited due to the decentral approach of AGLOSA, [28] introduced a "coordination factor" that favors certain roads in the optimization process to achieve coordination. This factor is applied during the optimization process as an additional weight to the optimization parameter (e.g. time loss). This means, AGLOSA no longer optimizes over the sum of the observed time losses $\sum_i t_{V,b,i}$ of the traffic flows i but over the sum of virtual time losses $\sum_i t_{V,v,i}$ which are calculated by the following equation:

$$t_{V,v,i} = f_i * t_{V,b,i}$$

The factor f_i can either have the value of 1, if the direction of the traffic is not coordinated or the value of $c > 1$. The factor should be chosen larger than one to ensure the respective direction is actually valued higher.

The first results seem promising, since the number of stops and the delay time in the coordinated directions can be reduced further than with the standard AGLOSA algorithm. Yet, this improvement is achieved at the expense of all other directions and decreases the overall system performance. [28]

In the following, the multi-intersection AGLOSA algorithm will be used in a scenario. The network is depicted in Figure 6-8. It has a basic topology, consisting of five four arm



intersections and a corridor connecting them. The distance between each of these intersections is 200 m. To simplify the scenario, all intersections only allow straight driving.

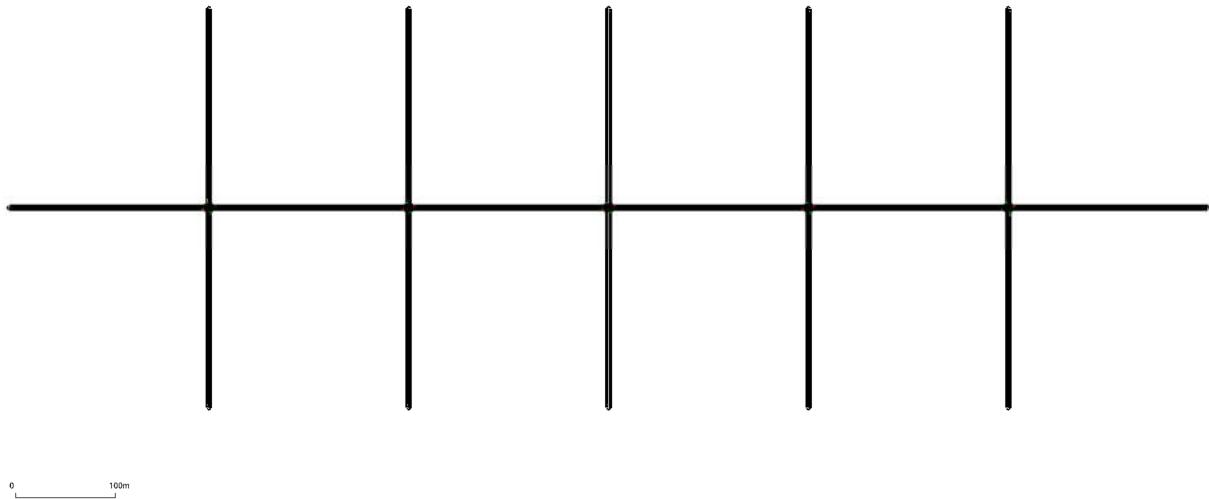


Figure 6-8: Multi-intersection scenario network

This scenario is designed as follows: The corridor is the main traffic flow direction; the north- and southbound approaches of the intersections are minor roads with little traffic volume. Table 6-3 shows the actual demands used for this scenario. There were three different control algorithms tested: Fixed signal control, AGLOSA and AGLOSA with coordination factor.

Table 6-3: Hourly traffic demand for the multi-intersection scenario

	Horizontal demand (west- / eastbound)	Vertical demand (north- / southbound)
Demand 1	500	100
Demand 2	1000	100

The results shown in Figure 6-9 illustrate drastically what impact AGLOSA can have on the traffic flow. The standard AGLOSA algorithm enables a time loss reduction of 10 – 30% and a reduction in CO₂ emissions of 10-40% compared to the fixed time control. The AGLOSA variant with coordination factor yields results comparable to the fixed time concerning the time loss. As for the CO₂ emissions, AGLOSA considerably decreases the emissions, at least for low traffic demand.



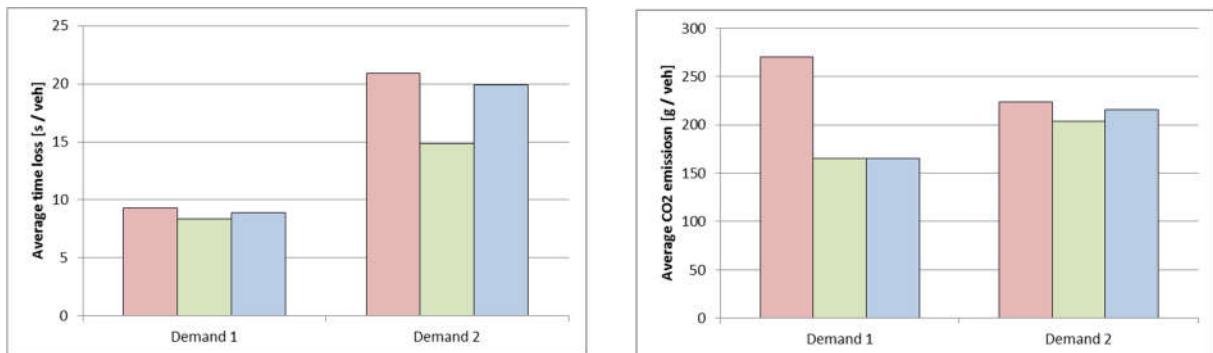


Figure 6-9: Results for the multi-intersection scenario (red: fixed signal plan, green: AGLOSA, blue: AGLOSA with coordination factor). Left: average time loss, right: average CO₂ emissions.

As said before, the variant of AGLOSA that uses a coordination factor yields slightly worse results than the plain algorithm. This effect is depicted in detail in Figure 6-10. The time loss in the main direction has slightly decreased in the coordinated AGLOSA variant while the time losses on the minor roads have significantly increased. So if the objective is the minimization of the overall time loss, the standard AGLOSA algorithm should be used. For coordination scenarios in which the time loss of one major traffic flow direction should be minimized, the AGLOSA variant with coordination factor is the better one.

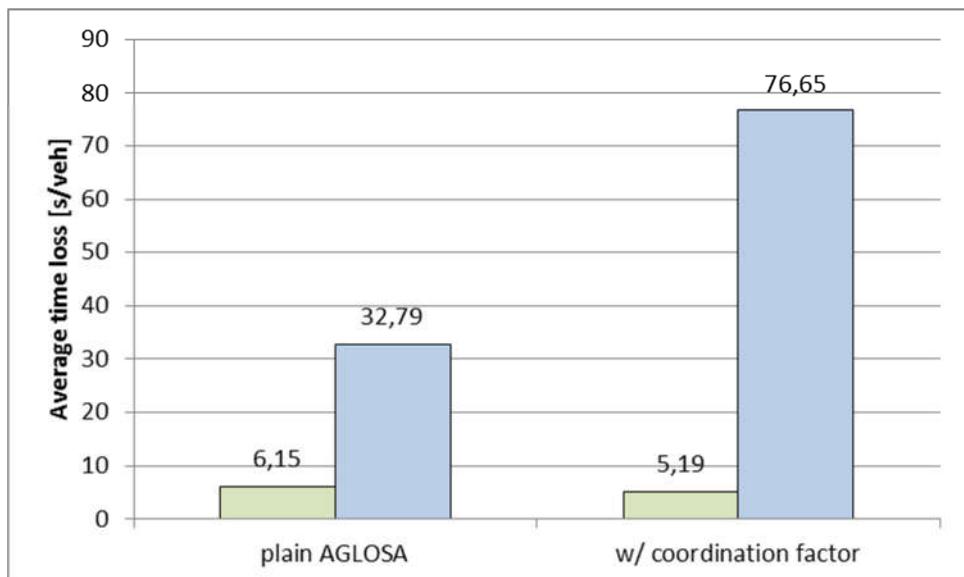


Figure 6-10: Time losses for the different traffic directions (green: main flow direction, blue: minor roads)



7 Plan stabilization for adaptive control

7.1 Algorithm design

Traffic control algorithm ImFlow is designed for adaptive real-time model predictive traffic light control. The principle concept of the ImFlow system is the optimiser, which uses the cost formula to optimize traffic signal timing, see the following schematic formula:

$$StateCost = Cost_1 + Cost_2 + Cost_3 \dots + Cost_n \quad 7-1$$

This *StateCost* is applied to each signal group of each intersection to calculate the intersection cost for the planned signal timing, during the whole configured planning horizon. The optimiser will compare many alternative signal-timing plans and execute the plan with the lowest intersection cost. Specific policies can be configured by the user, respectively to *Cost₁* to *Cost_n*. ImFlow uses this state cost function to evaluate different possible control solutions. From this it chooses the optimal one that minimizes delay and stops for all traffic approaching the intersection, more details can be found in Section 2.4.

The plan stabilization of traffic light controllers in MAVEN benefits from the “configurable cost” character of adaptive control-ImFlow, to prevent the optimizer to change the planning frequently or by a large deviation. Instead of giving more priority to drivers on the main corridor directions, the plan stabilization stabilizes the planned signal timing - particularly close to the green phase - so that drivers on these GLOSA available directions, receive reliable speed advice in order to pass the green light.

The extensibility of this adaptive control algorithm allows for adding new elements to the cost function, which makes the ad-hoc intended plan not preferable anymore comparing to the original plan. By doing so, it helps to overcome overly frequent change of signal plan, and to increase the reliability and accuracy of predictions for the time to green. Furthermore, it helps drivers modifying their speed to meet the green phase of the traffic light.

As shown in formula 7-1, the adaptive algorithm allows for adding new elements to the state cost formula, for instance, adding a configurable cost to *Cost₁*. These new elements aim at adding cost to an ad-hoc intended plan (for example, a deviated plan from original plan after the time to green advice already announced to drivers) by ImFlow, if this plan disrupts stability of GLOSA function on the main directions. A patent for a new algorithm adding such plan stabilization was applied. This should make the control algorithm more suitable for GLOSA, without deteriorating the average traffic delay significantly. The implementation of this cost function (C) is further explained in the following formulae 7-2 and 7-3:

$$C = \frac{c \cdot d^2}{TTG_{t-1}} \quad 7-2$$

$$d = TTG_{t-1} - TTG_t - T \quad 7-3$$

The configurable weight for stability (c) is a parameter that allows the traffic engineer to configure the importance of stability with respect to the other control targets. The deviation (d) is calculated using the difference between the time to green (TTG) of two consecutive time steps. The time period of a time step (T) is used for the expected decrease of the TTG as time elapses. The cost is quadratic with respect to the deviation because higher deviations are increasingly worse for the driver acceptance of a speed advice. Furthermore, the cost is inversely proportional to the *TTG_{t-1}*. This is because the closer to green, the more impact a deviation in the plan has. This close-to-green “punishment cost” feature is a major improvement compared to semi-fixed time strategies, which allow for flexibility around the stage transition and could result in change of time to green prediction very close to the actual



moment of the transition. The application of this cost function during a real-time simulation of a scenario with $c=10$ from Section 7.3 is shown in the table below. This table only gives 11 seconds of calculation for reference purpose.

SG8	c	TTG_t	TTG_{t-1}	T	d	Jumped	Cost
1	10	14	15	1	0	no	$10 \times (0)^2 / 15 = 0$
2	10	20	14	1	-7	yes	$10 \times (-7)^2 / 14 = 35$
3	10	19	20	1	0	no	$10 \times (0)^2 / 20 = 0$
4	10	18	19	1	0	no	$10 \times (0)^2 / 19 = 0$
5	10	17	18	1	0	no	$10 \times (0)^2 / 18 = 0$
6	10	16	17	1	0	no	$10 \times (0)^2 / 17 = 0$
7	10	15	16	1	0	no	$10 \times (0)^2 / 16 = 0$
8	10	14	15	1	0	no	$10 \times (0)^2 / 15 = 0$
9	10	5	14	1	8	yes	$10 \times (8)^2 / 14 = 45.71$
10	10	4	5	1	0	no	$10 \times (0)^2 / 5 = 0$
11	10	3	4	1	0	no	$10 \times (0)^2 / 4 = 0$
...

Table 7-1: Cost incurred on the stabilized signal group 8, using cost function 7-2

As can be seen from Table 7-1, the intentions to jump were not penalized enough in this scenario, and they result in successful unwanted jumps (a change in TTG , $d \neq 0$). To solve this, a higher c can be configured here. From a mathematical perspective, the results of cost function increase linearly to c but quadratic to deviation d . However, a jump of -7 followed by a jump of +8 is only +1 for the actual realization of the green phase. Therefore, increasing the weight c , would stabilize this scenario without any large cost to the traffic efficiency. The cause of these jumps are new approaching platoons detected by the queue model, which would justify changing the duration of some phases.

Regarding configuration of cost function 7-2, there are two advantages that can be adapted to get the best results of different networks. First one is that it can be activated on a per signal group basis. For example, this new element is only applied to the signal groups that require stabilization. The through direction of the main road generally has the largest amount of traffic and, therefore, the highest benefit of GLOSA and plan stabilization. Other directions can be more flexible because they do not require a cost for plan stability. In this way the controller is able to combine the best outcomes of two aspects: stability for the main direction and flexibility for the others. This is shown schematically in Figure 7-1; only the third stage has a fixed start of the green. The other stages are completely flexible and even their order could be changed if this is more optimal for the traffic flow (i.e. first stage 2, then 1 and finally 3).



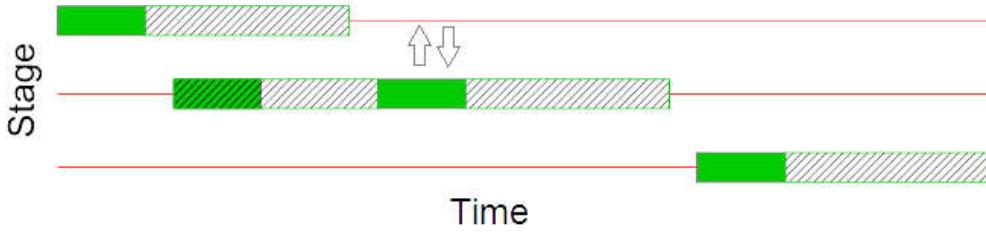


Figure 7-1: Stabilized adaptive control dynamics

During early testing, two unwanted cases of extending TTG (negative d) are seen during simulation:

1. “flat line” of several consecutive seconds have been observed where ImFlow keeps making small consecutive deviation d of 1 second to avoid high quadratic cost. This mostly occurs when a queue departs slower than expected. When this “flat line” situation where cost function didn’t prevent TTG jumps and TTG is successfully frozen each time step, TTG_{t-1} stays the same and the cost will not change for the duration.
2. Under the same situation of case 1, cost function prevented TTG jumps for the first few seconds, and TTG_{t-1} is going down 1 second each time step as planned. However, at the end the choice is between either ending the current green phase at that instant or delaying the planning of the stabilized direction 1 second further. This means for certain, that all remaining vehicles have to wait for another cycle. This gives a relatively high cost due to a vehicle actuated element in the original cost function.

MAVEN has worked further to advance the cost function in order to deal with the above mentioned situation on the Helmond network in the MAVEN project. The idea of memory, which “remembers” the previous TTG deviation d and adds more cost to small consecutive jumps (also can be seen as “creeping up/down” on TTG suggestion or speed advice) by using the cost function, is proposed. Accordingly, three new terms are introduced: memory accumulation parameter α , memory dissipation parameter β and extension level EL. In short, α and β are two configurable parameters that are used to calculate memory at the end of each time step, shown in formulae 7-4, 7-5 and 7-6. Literally, d_t is the TTG deviation at time step t between an intention plan and the old plan of ImFlow controller. M_t is the memory at time step t . Extension level refers to the extension level of vehicle actuated (VA) control. EL can be set to 0, 1 or 2. When EL is set to 0, VA extension is enabled as the baseline scenario setting of the network; When EL is set to 1, VA extension is disabled for all signal groups if the next planned stage is stabilized; When EL is set to 2, VA extension is disabled for all signal groups if any of the future planned stage is stabilized. The last one of EL=2 is very extreme and should be used with caution.

$$\text{if } d_t \times M_t > 0, \quad M_{t+1} = \alpha M_t + d_t \quad 7-4$$

$$\text{if } d_t = 0, \quad M_{t+1} = \beta M_t \quad 7-5$$

$$\text{if } d_t \times M_t < 0, \quad M_{t+1} = \beta M_t + d_t \quad 7-6$$

Where $\alpha > 0$, and $0 < \beta < 1$; $d_t \times M_t > 0$ means d_t and M_t have the same sign; $d_t \times M_t < 0$ means d_t and M_t have the opposite signs.

M_t is calculated at the end of each time step, d' is introduced here to take the higher value between d_t and $d_t + M_t$, see formulae 7-7 and 7-8. Then d' is used to update d in cost function 7-2.

$$\text{if } d_t \times M_t > 0, \quad d' = \text{Max}(|d_t|, |d_t + M_t|) \quad 7-7$$

$$\text{if } d_t = 0 \text{ or } d_t \times M_t < 0, \quad d' = d_t \quad 7-8$$



Formulae 7-4 to 7-8 showed the logic process of this advanced cost function. The basics of this algorithm is, that the memory of previous TTG deviation is accumulated via parameter α to punish continuous TTG deviation in the same direction, thus “remembering” the persistent and unwanted behaviours of TTG deviation. On the other hand, the memory of previous TTG deviation is dissipated via parameter β , for situations where none ($d_t = 0$) or opposite direction TTG deviation are shown, thus “forgetting” and punishing less to encourage stabilization behaviours. This is also to enable future stabilization in a different direction. Once the target signal group of stabilization switches to green, the memory is reset to 0.

The advantages of this advanced cost function algorithm are threefold. First, tracking the flexibility of adaptive control ImFlow to further smooth and stabilize planning effectively; second, the introduction of β insures the functionality of cost function, by preventing memory overflow or even preventing unpredicted situations where cost are “too high” for ImFlow to optimize; Lastly, these parameters are subjected to configuration for an individual network according to local polices. So it takes advantages of the flexibility of adaptive control, without deteriorating the network performance, see impact results from Section 7.3 for reference.

7.2 Simulation methodology

To implement the algorithm design introduced in the Section 7.1, a corridor with multiple intersections in Helmond was built up using SUMO, shown in Figure 7-2. Six intersections: intersection 701, 702, 704, 101, 102, 103 are distributed on this stretch of corridor, with the same main direction, east-west through directions 2 and direction 8 for each aforementioned intersection. Respectively, signal group 2 of each intersection is east-west bound and signal group 8 is west-east bound.

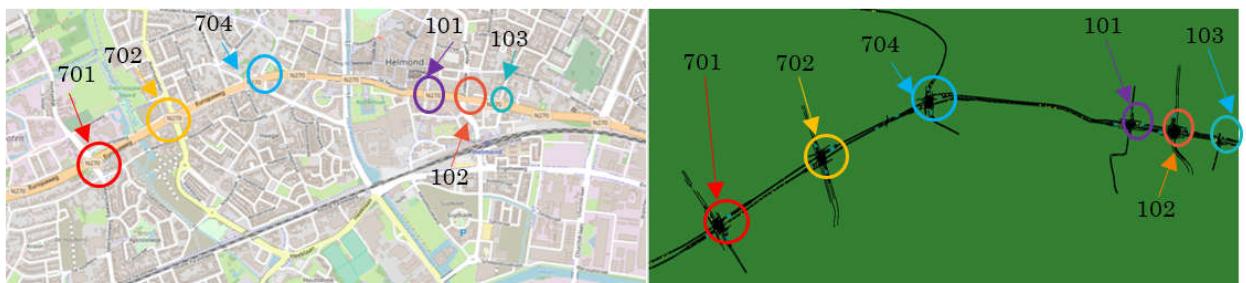


Figure 7-2: Case study of Helmond (left) and the simulation network in SUMO (right)

- Intersection 701, Hortsedijk/ Europaweg
- Intersection 702, Boerhaavelaan/ Europaweg
- Intersection 704, Prins Hendriklaan/ Kasteel-Traverse
- Intersection 101, Zuid Koninginnewal/ Kasteel-Traverse
- Intersection 102, Zuidende/ Kasteel-Traverse
- Intersection 103, Penningstraat/ Smalstraat/ Kasteel-Traverse



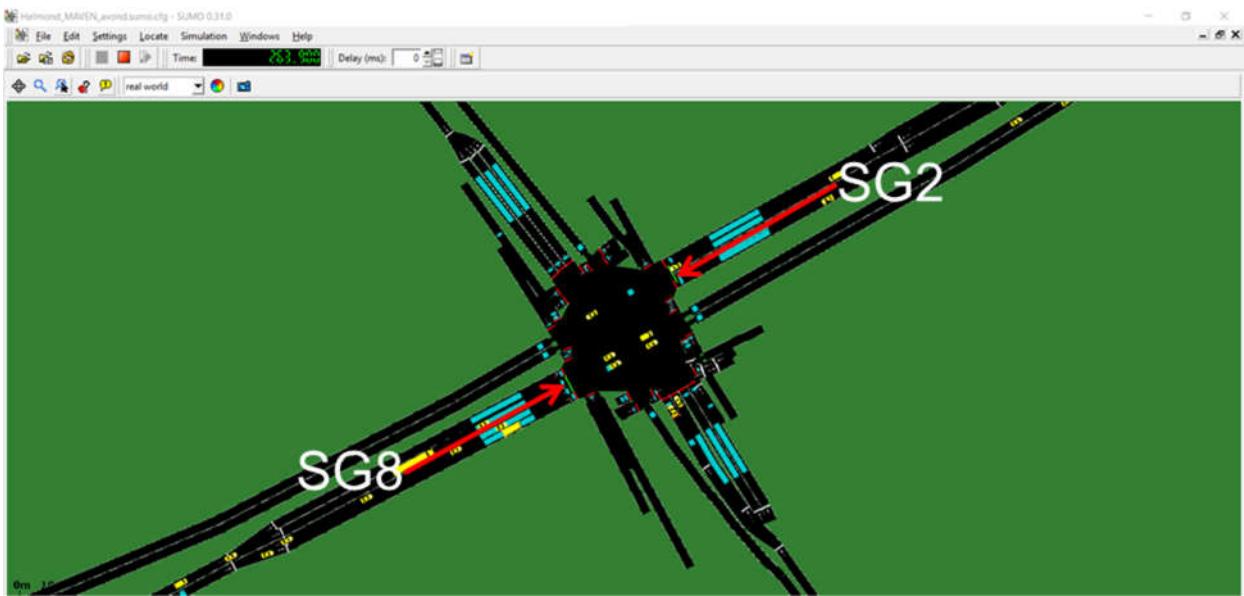


Figure 7-3: Helmond Intersection 701, Hortsedijk/ Europaweg

Figure 7-3. The configuration of the two signal group, SG 2 and SG 8 of intersection 701 are almost identical. They contain the same number of lanes (two lanes each), have the same saturation flow (1800 vehicle/hour), same number of signal heads and they both appear in the same stages/ stage assignment. The simulated traffic is detected in SUMO, then the detected vehicle information is sent back to ImFlow to calculate and optimize the signal timing plan. After making the decision of which plan to choose (the algorithm design of Section 7.1 applies during this process), ImFlow sends back the chosen plan to SUMO to continue the simulation. The detection type of SG 2 and SG 8 are both set to adaptive unconditional in ImFlow configurator. Therefore, stabilized GLOSA can be provided to these two signal groups together with the same configuration of parameters mentioned in Section 7.1.

The automated vehicles using these signal groups will follow GLOSA and plan stabilization is enabled when applicable. For the overall performance of the network, it is important that the largest stream of traffic receives the speed advice, which are main directions, SG 2 and SG 8. In addition, the signal groups with plan stabilisation are set to only SG 2 and SG 8 because they act almost as one. Adding more signal groups could impede flexibility, and in turns negatively impact the performance of the network, while applicability of GLOSA on local roads is much less. The developed simulation method is also intended for implementation for cyclists on this corridor in a later phase of MAVEN.

The speed advice is directly applied to traffic participants using the TraCI interface of SUMO. Speed advice is applied from 350 m before the stop line and is subject to a range of 30 - 50 km/h for vehicles. Slower speeds are not considered realistic. Faster speeds are simply above the speed limit. The speed advice algorithm is straightforward: the distance to the stop line is divided by the time the vehicle can first pass the stop line. This time includes both the time to green (TTG) and the time required for any queue ahead to discharge. The prediction of TTG is logged during the entire simulation. Afterwards, the actual TTG can be calculated by stepping back from the moment the light turned green. The comparison is not considered relevant when $TTG > 60$ seconds, because this is too far in the future, traffic participants would still be at an upstream intersection. Therefore, these cases are filtered out of the statistics.



The simulation network has sufficient space for vehicles to enter the network even in case of long queues due to congestion. If there is no severe congestion, the throughput is the same for all simulations. Another possibility is that one or more signal groups are blocked upstream and had too little traffic entering. This makes it easier for the controller to serve the other traffic with low delay and results in an unfair comparison to normal traffic demand situation of all other simulations because the solution would be unacceptable in the field.

To build statistically significant data, the simulations are performed with 5 runs of 2 hours evening peak hour per traffic control scenario, which comprises of different configured parameters, such as c , α , β and EL. The simulation study is set up according to the following hierarchy rules:

1. Higher configurable value of a chosen parameter when a low value is not sufficient to deliver satisfactory performance. In this study, we increase the c value of function 7-2.
2. Trigger coordination of more parameters when a single parameter (after analysing the simulation results following rule 1) is not enough or has a specific unwanted element, like the flat line described in the previous section. In this study, α is triggered to coordinate with c when increasing c alone is not effective; β and EL are triggered when the best results from c and α combinations are found. The latter case is to see whether β and EL can improve the performance further when c and α combinations are already giving good results.
3. The parameters α , β and EL are advanced tuning parameters at a level of detail not likely to be used in commercial exploitation. Therefore, it is important to monitor the statistical significance, because these parameters may also introduce unexpected statistical noise. They may cause situations that would not trigger every simulation run. Therefore, their effect cannot be determined by taking two data points and assume other values can be extrapolated.

Following the aforementioned rules of simulation scenarios built-up, all simulation scenarios are based on the same network configuration, same simulation method and operations. Thus, different scenarios are listed in Section 6.1 according to different parameters, and results of these scenarios will be discussed in Section 7.3. The numbering follows the logic of scenario's and sub-scenario's. A major scenario 1, 2, 3, etc. has an increasing c value. The first level of sub-scenario has a different α , e.g. 2.1, 2.2, etc. The last level varies β and EL, resulting in for example 2.2.1 and 2.2.2.

During the simulation, the delay time and the amount of stops are tracked for every traffic participant. Overall averages are reported in the results Section 7.3 for impact, delay and stops. The impact is a measure of traffic efficiency that can give a quick overview of the performance of a simulation scenario.



Table 7-2: Simulation scenarios list

Scenario No.	c	α	β	EL
0 Baseline	0	0	0	0
1	30	0	0	0
2	60	0	0	0
2.1	60	2	0	0
2.2	60	4	0	0
2.2.1	60	4	0.5	0
2.2.2	60	4	0.5	1
2.2.3	60	4	0.95	0
2.2.4	60	4	0.95	1
2.3	60	8	0	0
2.3.1	60	8	0.5	0
2.3.2	60	8	0.5	1
2.3.3	60	8	0.95	0
2.3.4	60	8	0.95	1
2.4	60	16	0	0
2.5	60	32	0	0
3	90	0	0	0
4	120	0	0	0
5	150	0	0	0
6	180	0	0	0
7	210	0	0	0
8	240	0	0	0
9	300	0	0	0
10	480	0	0	0
11	600	0	0	0

As explained in Section 2.4, one can expect that adaptive control such as ImFlow has higher flexibility, therefore better performance and less impact on a traffic network. Respectively, we can identify impact as a MOE that indicates the performance of an adaptive control algorithm. It is defined by the following formula:

$$impact = \frac{\sum_{i=0}^{I=I} delay_i + 8 stops_i}{I} \quad 7-9$$

The formula sums over all traffic participants (I) and calculates the average impact. It can be applied to the total network or to a single signal group. In this study, the impact is calculated based on traffic participants using intersection 701 because for one, the network is quite big to observe changes from different scenarios if all participants are taken into account. Actually, they are “diluting” the changes in performance; for two, only parameters of signal groups 2 and 8 of intersection 701 are changed for each scenario. Naturally, intersection 701 is the point of interest.

The value 8 in formula 7-9 is often used as a rule-of-thumb factor by traffic engineers. It is based on CO₂ emissions and road user comfort of not stopping. The most interesting signal groups are the ones where the stabilized GLOSA service is applied. Therefore, these signal groups will be reported separately as well on measures of effectiveness such as MRE (mean relative error) and PC (perceived change).

A mean square error (MSE) is calculated as a good indicator for overall reliability of the data and is commonly used in many fields of science. However, the TTG_{t-1} in formula 7-2 indicates the sensitivity of TTG deviations close to the actual moment of turning on green and we need



to penalize deviations harder by adding a higher cost. Therefore, another measure of effectiveness, the mean relative error (MRE) was added, which divides the error by the remaining *TTG* and expresses this as a percentage. In this study, MRE is designed as a stability measurement that specifically targets the potential performance of GLOSA.

The other stability measure is the Perceived Change (PC), which represents the percentage change between two consecutive predictions relative to the remaining *TTG*. The calculation of this measure is described in the formula 7-10:

$$pc = \frac{\sum_{t=1}^T \frac{\alpha TTG_{t-1} - TTG_t}{\min(TTG_{t-1}, TTG_t)} 100\%}{\sum_{t=1}^T \alpha} \quad 7-10$$

$$\alpha = \begin{cases} 0, TTG > 60 \\ 1, TTG \leq 60 \end{cases}$$

The PC measure serves to estimate the users' perception of the system. For example, an original plan of a *TTG* prediction sequence of three time steps is 50, 49, 48, an alternative intention plan A is 55, 44, 53 and another alternative intention plan B is 55, 54, 53. Plan A and plan B have the same MSE if we calculate their values: MSE of plan A = [(55-50)² + (44-49)² + (53-48)²]/3=25 and MSE of plan B = [(55-50)² + (54-49)² + (53-48)²]/3=25. However, the user will see the GLOSA prediction jumping around in plan A more quickly than plan B, and the user will consider the information as an unreliable source. This would hurt compliance rate greatly. Therefore, a low value for this PC is important for users' perception.

To sum up, lower MRE and lower PC are preferred from a GLOSA stability perspective. At the same time, network performance (*impact*) should not be deteriorated too much to hurt the flexibility of adaptive traffic control ImFlow. A figure of merit, FOM_un-unified is proposed to evaluate the performance of each simulation scenario, as shown in formula 7-11. In a nutshell, the lower the FOM_un-unified is, the better the result of the scenario is.

$$FOM = Impact^2 \times MRE \times PC \quad 7-11$$

FOM_un-unified takes into account the balance between traffic efficiency (indicated with *Impact*) and stability (indicated with *MRE* and *PC*). Square value of impact is to balance the appearance of MOEs in formula 7-12. To conduct data analysis more conveniently and to have overview of comparing to baseline scenario, this arbitrary formula of FOM_un-unified is transformed to FOM_unified in formula 7-12.

FOM_unified:

$$FOM_{uni} = Impact_{uni}^2 \times MRE_{uni} \times PC_{uni} \quad 7-12$$



Where,

$$\begin{aligned} Impact_{uni} &= Impact \div Impact_{baseline} \\ MRE_{uni} &= MRE \div MRE_{baseline} \\ PC_{uni} &= PC \div PC_{baseline} \end{aligned}$$

Section 7.1 outlined the algorithm design of plan stabilization for adaptive control ImFlow. Section 7.2 introduced the network of simulation study area and showed the simulation methodology directing this research. Based on the algorithm and the simulation methodology, simulations were conducted and the results are shown in Section 7.3.

7.3 Results

The network configuration and simulation scenarios are explained in Section 7.2. First, scenario 0, 1, 2, 3 ...11 (different c in the cost function 7-2, given no α , β and EL in functions 7-4 to 7-8) are simulated on the study network with different c , ranging from $c=0\sim 600$. When $0 < c \leq 240$, the interval of c is 30; when $c > 240$, bigger intervals of c are used. This is due to hardware constraints of around 200 minutes for a simulation with 5 runs. Also, as we carry out simulations, similar patterns of results in accordance to expectations. With increasing c , the MRE and PC reduce at a limited damage on the impact. After a certain c value, increasing c alone is not making much of a difference on performance results. Most likely all cases where the cost function is used to evaluate a change of TTG were already filtered out and the only remaining instability is due to constraints in the control plan. Therefore, a smaller interval of c is redundant in this situation for simulation results purpose. The simulation raw data and calculation of MOEs can be seen in the table below:

Table 7-3 Data of scenario 0 to 11, $c=0\sim 600$

N o.	c	α	β	EL	impact	MRE	PC	Impact_uni.	MRE_uni.	PC_uni	FOM_uni.
0	0	0	0	0	24136	17.737	4.28	1.0000	1.0000	1.0000	1.0000
1	30	0	0	0	24288	16.402	3.37	1.0063	0.9247	0.7888	0.7387
2	60	0	0	0	24262	15.753	3.11	1.0052	0.8881	0.7276	0.6529
3	90	0	0	0	24595	15.602	3.29	1.0190	0.8796	0.7703	0.7036
4	120	0	0	0	24236	14.085	3.23	1.0041	0.7941	0.7562	0.6055
5	150	0	0	0	24041	13.526	3.55	0.9961	0.7626	0.8301	0.6280
6	180	0	0	0	24032	13.860	3.35	0.9957	0.7814	0.7838	0.6072
7	210	0	0	0	23600	14.221	3.12	0.9778	0.8017	0.7289	0.5587
8	240	0	0	0	24033	12.878	3.14	0.9957	0.7260	0.7350	0.5291
9	300	0	0	0	24057	12.583	3.039	0.9967	0.7094	0.7106	0.5008
10	480	0	0	0	24279	12.594	3.236	1.0059	0.7100	0.7567	0.5436
11	600	0	0	0	24150	13.866	3.034	1.0006	0.7818	0.7094	0.5552



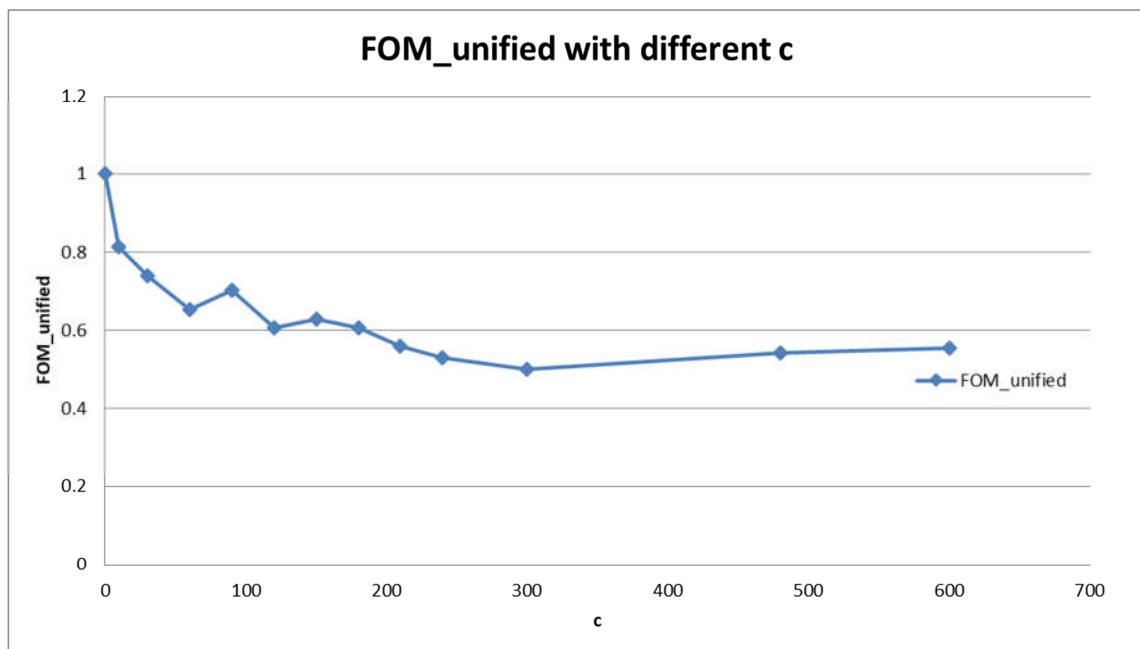


Figure 7-4: Results of scenarios with diff c , given no α , β and EL

With these data above, the FOM_unified curve respect to different c is shown in Figure 7-4. As can be seen from the graph below, configuring $c \neq 0$ has an obvious decrease of FOM_unified. Considering the value of FOM_unified, scenario 0 is 1 and scenario 1 ($c=30$) is 0.738, with 26% decrease comparing to scenario 0. It means that scenario 1 has an improvement of 26% comparing to baseline scenario, scenario 0. Seeing this, an ad-hoc scenario of $c = 10$ is simulated (data is not included in), which gives an improvement of 19%. It proves the expectation of “significant improvement when $c \neq 0$ ”. FOM_unified of scenario 2 to scenario 9 shows much less improvement with FOM_unified fluctuating to lower values, then towards stable values.

These results show that the advanced cost function algorithm design has significant improvement regarding plan stabilization. MOEs of stabilization, such as MRE and PC (shown in Table 7-3), decrease when a c value is configured, which means the signal groups with GLOSA are more and more stabilized. At the same time, impact of all traffic on intersection 701 does not deteriorate much if we look at the Impact_unified column in Table 7-3. Scenario with extremely high c value is also simulated. The results of FOM_unified are even increasing and the performance deteriorates beyond $c = 300$. Observing the simulations, there are some cases where c alone cannot improve the stabilization anymore. For example, the consecutive small jumps caused by constraints as explained in Section 7.1. Beyond $c = 60$, the cost function still improves stability, but at the same time causes the controller to hit constraints more often. Therefore, the FOM_unified does not decrease steadily anymore as a lot of random noise is introduced in the simulations.

Now that the results and indicated pattern of FOM_unified with increasing c value alone are shown, scenario 2 of $c = 60$ is chosen for further simulation study. There are three reasons to choose

$c = 60$. Firstly, higher c value did not cause FOM_unified to decrease and is not improving the performance. Secondly, $c = 60$ shows generally good and with less random noise, even after 20 runs of testing for one scenario. Lastly, increasing c value alone is deteriorating impact on the network level, especially when c is too high, for example, $c > 600$.



According to the rule 1 and rule 2 of simulation methodology in Section 7.2, the next parameter in line, α is triggered, to study whether α can solve some situations where increasing c did not.

As listed in Section 6.1 different α , $\alpha = 0, 2, 4, 8, 16, 32$ are configured for scenario 2.1 to scenario 2.5, under the same $c = 60$. In this situation, the scenario 2.1: $\alpha = 0, c = 60$ with no β and EL, is used as the benchmark for the calculation of Impact_unified, MRE_unified, PC_unified, and eventually FOM_unified. The results of FOM_unified are graphed in Figure 7-5 below.

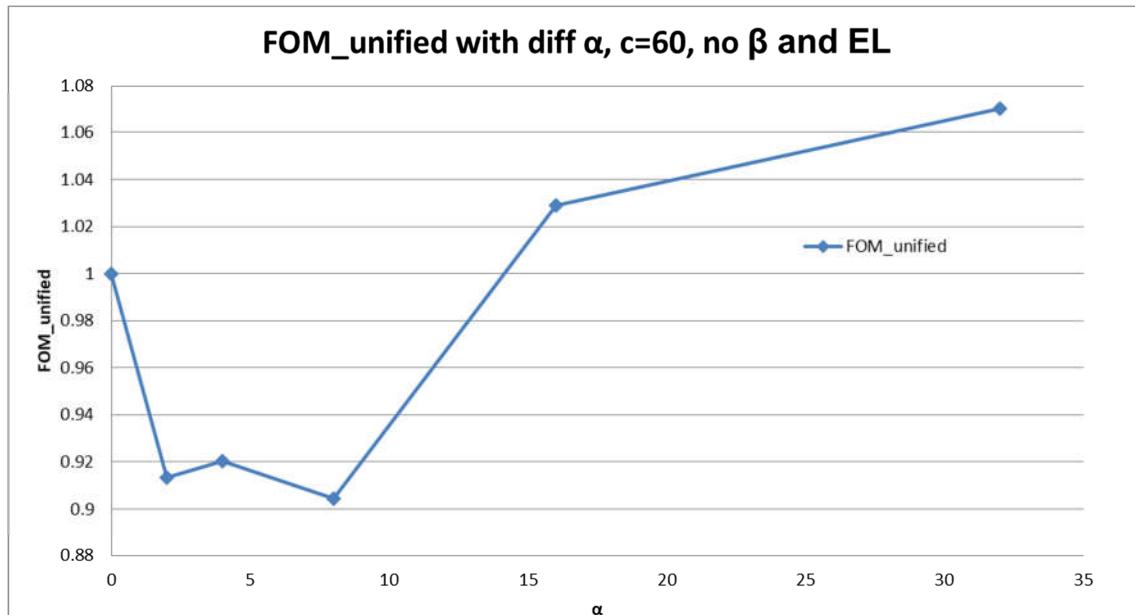


Figure 7-5: Results of scenarios with diff α , $c=60$, no β and EL

Tentatively, $\alpha = 4$ and 8 have lower FOM_unified comparing to $\alpha = 0$ and they are giving better performance. It means that the algorithm of accumulating memory in cost functions 7-4 to 7-8 are solving some consecutive small deviation situation.

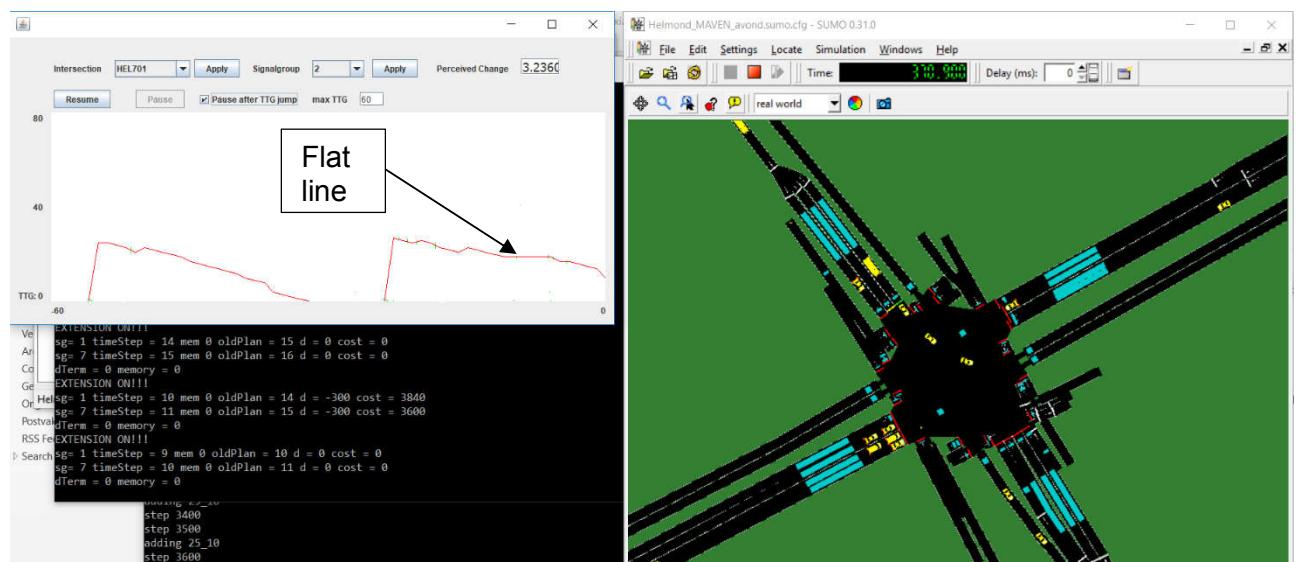


Figure 7-6: An example of small consecutive jumps, “flat line” situation



By closely observing the simulations and cost calculation output of each time step, we can see that the situation of several small deviation of 1 second (shown in Figure 7-6) in a prediction cycle are significantly reduced. Since this type of TTG deviation resemble the shape of keeping flat for several seconds, we define them using the analogy “flat line”. These kinds of changes in the field would result in a speed advice that keeps slowly decreasing, which is especially uncomfortable and frustrating for drivers who are trying to follow the GLOSA and keep having to slow down. When a count-down bar would be displayed to users, they may even think it is malfunctioning when it freezes.

On the other hand, configuring α with extreme higher values such as 16 or 32, gives an opposite impact as expected, shown in Figure 7-5. The results are worsen comparing to no α . This unwanted effect is preliminarily regarded as the result of too extreme punishment cost from accumulating memory that introduced other complicated and more deviated situations for ImFlow to optimize. This would not give a fair comparison anymore with the baseline scenario due to more cases of randomness and complication.

From Figure 7-5, the results of scenario 2.2 and 2.3 are the most promising ones. Further simulation of scenarios 2.2.1-2.2.4 and scenarios 2.3.1-2.3.4 are built according to rule 2 of the simulation methodology in Section 7.2. Parameters β and extension level (EL) are triggered in these eight scenarios. After performing these simulations, data was collected, analyzed and compared to baseline scenario 2.2 ($c = 60, \alpha = 4, \beta = 0, EL = 0$). The results of introducing β and EL are graphed in Figure 7-7.

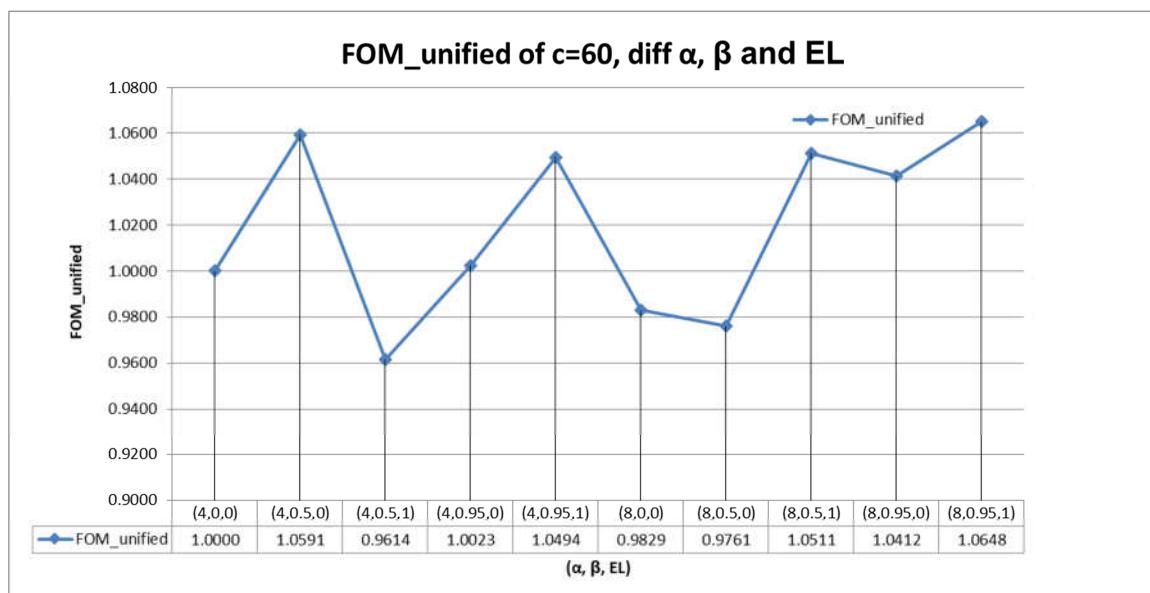


Figure 7-7: Results of scenarios with $c=60$, diff α , β and EL

As seen from above graph, for $c = 60, \alpha = 4$, setting β to 0.5 or 0.95 and EL to 0 makes the results a bit worse, but setting EL to 1 may ($c = 60, \alpha = 4, \beta = 0.5, EL = 1$) or may not ($c = 60, \alpha = 4, \beta = 0.95, EL = 1$) give an improved result. Additionally, for $c = 60, \alpha = 8$, setting β to 0.5 or 0.95 and EL to 0 makes the results relatively better, but setting EL to 1 give worse results ($c = 60, \alpha = 8, \beta = 0.5, EL = 1$) than ($c = 60, \alpha = 8, \beta = 0.95, EL = 1$). Setting EL to 2 was also tried, the FOM_unified of scenario ($c = 60, \alpha = 4, \beta = 0.5, EL = 2$) is 1.0232. It is 2.32% worse than scenario ($c = 60, \alpha = 4, \beta = 0, EL = 0$) and 6.43% worse than scenario ($c = 60, \alpha = 4, \beta = 0.5, EL = 1$). Similarly deteriorating results are produced from other scenarios that have a configuration of EL = 2. Therefore, all EL = 2 scenarios are left out of the comparison in Figure 7-7.



With more simulations from other interesting scenarios, for instance, $c = 120$ or $c = 210$ (see acceptable results for these scenarios in Figure 7-4), Similar uncertain conclusions are shown. Therefore, their data are excluded here.

More simulations can be conducted for the further research of Helmond network or other traffic networks in SUMO or Vissim, using ImFlow for the adaptive traffic control. This would result in better calibration guidelines. At this stage, the algorithm was proven to be effective and several conclusions can be drawn:

- The advanced cost function algorithm of Section 7.1 is effective. The algorithm is well-designed, especially targeting on balancing the traffic efficiency of adaptive traffic control and the stability for GLOSA functionality.
- The simulation methodology derived from this algorithm is functional and can be developed further. The methodology was already disseminated so other researcher in the field can use similar traffic simulation methodology. In particular, it can deliver better plan stabilization analysis and debug the network until the best results are shown.
- The advanced memory features β and EL show very little effect. More research is needed to arrive at calibration guidelines. Preliminary, it can be concluded that β is probably too ineffective to be configured in the future. This would increase operational costs and endanger exploitation of the system. The EL can be set at a default value without the need of calibration as it filters out specific unwanted cases that have little impact on the overall results.



8 Green waves for automated vehicles

8.1 Introduction

Many approaches to green waves are possible. Similar to giving effective speed advice to vehicles, it is again a trade-off between control flexibility and success rate of the use case. The most effective strategy for guaranteed green waves is static control, which is also the traditional approach to the use case. Software packages exist since the 1970's that automatically calculate cycle time, split and offset for such green waves. Transyt [4] is a commercial product that combines well with the SCOOT control method to add some flexibility to the control plan. COCON is a Dutch variant that calculates fixed time plans and supports upgrading them to actuated control plans. Lisa+ is a German version compatible with the local control standards.

The advantages and disadvantages of different control methods are already described in Chapter 2. Therefore, in the current state-of-the-art, a static control green wave is never the most efficient solution when it comes to overall traffic efficiency. It's generally used as a low-cost solution or when a very specific policy requires such a green wave. Semi-static control and adaptive control are considered the best solutions in the market at the moment. The choice on which particular system is used depends on the network and local standards. In general more flexibility is used for smaller networks.

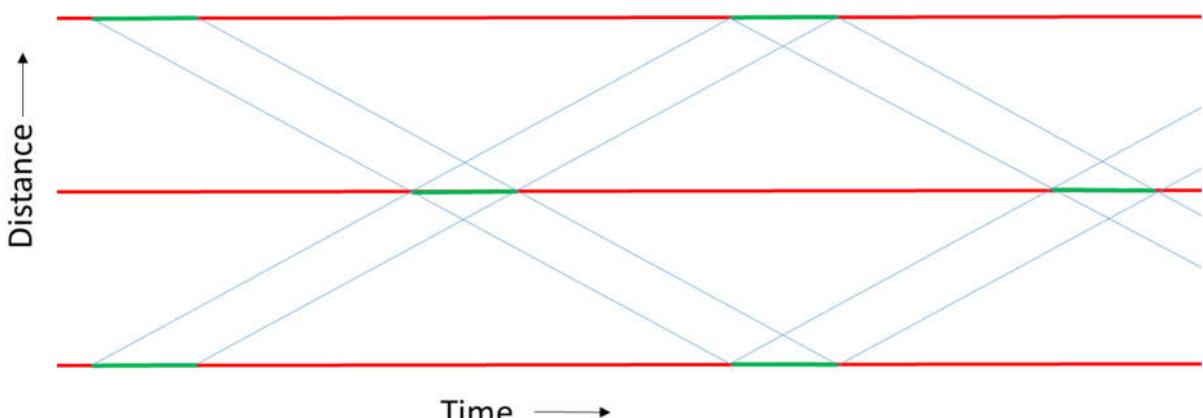


Figure 8-1: Ideal bi-directional green wave

A unidirectional green wave is trivial as any offset between intersections can be chosen and the only requirement is that the cycle time is synchronized. For bi-directional green waves, the offset in one direction needs to be equal to the offset in the opposite direction. Figure 8-1 illustrates this as the bands of both directions perfectly overlap and vehicles arrive at the second intersection in the middle of the cycle. This is only possible on a stretch of equally spaced intersections and the cycle time has to be equal to twice the travel time between the intersections. With large distances between the intersections it is also possible for the travel time to be half a cycle plus an integer amount of cycles. Therefore the following condition must be satisfied for a perfect bidirectional green wave:

$$t_{i,i+1} = \left(n + \frac{1}{2} \right) C \quad \text{with } n \in (0, 1, N) \quad 8-1$$

In practice, it is very difficult to find networks that have equally spaced intersections at a distance that results in a convenient cycle time. A small extra offset can be created by leading or lagging left turns as described in [33]. This is often insufficient resulting in green windows for different directions to not overlap, which reduces network capacity.

In specific for green waves, the platoon dispersion is also an important factor and often a problem for networks that are close to saturation. To give a smooth green wave, it should be



taken into account that the front of a platoon moves at the speed limit, while the back is a bit slower. To avoid vehicles falling out of the green window or being tempted to violate the red light, the green phase should be a bit longer for each consecutive intersection. This effect can be countered with platoon shaping via a speed advice. For non-cooperative vehicles the best solution to do this is using dynamic signs with speed advice, like in the previously discussed ODYSA system [5]. This way the head of the platoon can be slowed down while the back can catch up. Using C-ITS, the advice can be given more precise and continuous, rather than only at the specific locations of the dynamic signs. A GLOSA application in combination with green wave would take over the role of platoon shaping.

To go beyond the state-of-the-art, several directions were explored by MAVEN:

- The first direction is reported in Section 6.2, which is the extension of AGLOSA for multiple intersections. The newly introduced coordination factor is successful at creating a better green wave, but at the cost of some overall performance. It does give extra policy control to authorities operating actuated controllers that want to add GLOSA and green wave to their systems.
- Self-learning algorithms are another trend that seems to have potential for green waves. The second section of this chapter will describe how a self-learning algorithm was used to create an adaptive green wave.
- Adaptive control can get further constraints to force a bi-directional green wave, by setting a custom speed limit in certain road sections that make a bi-directional green wave easier.
- Speed advice in combination with static control has more possibilities when interacting with automated vehicles. The range of acceptable speed limits can be further increased and platoons can be split and recombined when this is optimal.



8.2 Q-learning approach

8.2.1 Background

The approach of this section is based on an internship sponsored by Dynniq. Although it was not funded by MAVEN, the results are relevant for the project and brought in by Dynniq. The full report can be found in [30].

With the emergence of computational intelligence (CI), as a hot research field, methodologies that facilitate problem solving that was previously difficult or impossible, were developed. [31] states why CI methodologies are a good approach to cope with the optimization problem in traffic signal control systems:

- Traffic signal control optimization is a complex non-linear stochastic system. Therefore, it is hard to find optimal traffic signal settings, CI provides a feasible way to obtain optimal or suboptimal solutions.
- Although it is complex to precisely model dynamic traffic, most traffic signal optimal methods rely on precise mathematical traffic models. Most CI methodologies do not require precise models, sometimes no model is required at all.
- CI can be used to enable or facilitate intelligent behaviours in complicated, uncertain, and changing environments, therefore traffic signal control plans can be determined based on real-time traffic conditions and historical reasoning.

The chosen CI method for this research is Q-learning, which is a reinforcement learning method. The mathematics of Q-learning are further described in [32]. The goal of this method is to learn a policy, which tells an agent which action to take under which circumstances. Q-learning is a model free approach, thus no model of the environment is required, instead rewards can directly be measured from the environment. The optimal policy is learned by mapping an environment's state s to the best action a based on accumulating rewards $r(s, a)$. Every state-action pair (s, a) has a value called the Q-value, that represents the expected reward for the action a performed in state s . All Q-values are stored in the Q-table, where initially these values are zero, but as the agent is observing the current state s and selecting an action a from the set of available actions A , the Q-values are updated according to the reward $r(s, a)$ and the maximum expected Q-value in the state of the next decision point s' as follows:

$$Q'(s, a) = (1 - \alpha)Q(s, a) + \alpha[r(s, a) + \gamma \max Q(s', a')] \quad 8-2$$

where $\alpha, \gamma \in (0, 1)$ are the learning rate and discount rate, respectively. The learning rate decreases as a state-state action pair gets visited by the agent during a simulation run. The discount rate describes the influence a future reward has compared to the immediate reward, as the value of γ gets closer to 1 the Q-value $Q(s, a)$ is more heavily influenced by the reward in the future. This is something generally desired for green wave systems due to the long-horizon network-wide scope of it.

The state of the intersection s includes the current set of green traffic lights (stage) and for how long they have been green and the queue and arrival patterns of all approaches towards the intersection. Actions a consist of extending the current stage or switching to another one.

8.2.2 Results

The research was conducted in an exploratory manner to test the potential of the algorithm. Therefore, the control solutions cannot be considered as end-products. At higher TRL levels the performance is different, extra constraints due to regulations deteriorate the performance, while a mature product will have more safeguards against getting into an erroneous state that will improve performance. The results should therefore be treated as indicative to



expose strengths and weaknesses of a certain method, but it cannot directly determine the tipping point when a certain method is better than another. The simulation networks used are artificial and shown in Figure 8-2:

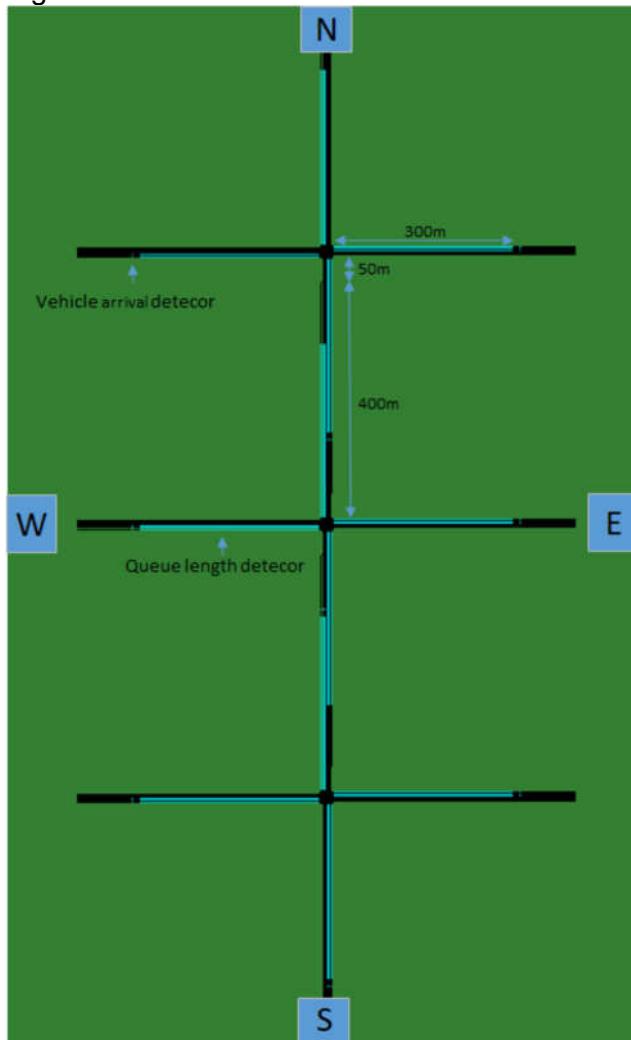


Figure 8-2: Artificial network used for evaluation

From this layout two networks are derived, network 1 has 450m distance between intersections (as shown in the figure), while network 2 has only 250m. Generally, very short distances between intersections are more difficult to form bi-directional green waves due to the decreasing cycle time that would facilitate that. The intersections have 4 stages with a volume of 100, 125, 430 and 750 veh/h on the busiest movement.

The results for the two networks are shown in Table 8-1:

Table 8-1: Results of Q-learning versus dynamic programming

Network 1 \ volume factor	0.5	0.75	0.9	1.0	1.1	1.25	1.50
Dynamic programming	6,7	21,6	35,8	45,2	60,6	85	120
Q-Learning	13.7	28.0	35.2	37.3	42.8	69.5	100.3
Network 2							
Dynamic programming	5.8	16.8	29.7	44.8	58.1	67.0	82.6
Q-Learning	15.1	34.5	55.7	69.7	82.7	87.1	96.4



The dynamic programming is a simplified adaptive control strategy, that is similar to Utopia and ImFlow, but without advanced constraints. Q-learning is the new strategy. With a volume factor of 1.5, the network is really close to saturation, while 0.5 can be considered an off-peak situation.

Generally off-peak requires more flexibility to control efficiently and this is where dynamic programming clearly defeats Q-learning. For higher volumes we see that the distance between the intersections is very important. With a speed limit of 50 km/h, the travel time is 35 seconds, which means the cycle time for a bi-directional green wave is 70 seconds when following formula 8-1. However, for the second network the distance is much smaller and a cycle time of 35 seconds would be optimal. Given that every stage change requires 5 seconds of intergreen time (amber and clearance), it means that there is only 15 seconds per cycle of green available. That's too few for an efficient control program. Therefore, Q-learning is able to learn the optimal program when it's not too difficult to find it, then it will hold on to it even if it is not optimal on a short horizon. Dynamic programming keeps optimizing and can lose track of the long term benefits. However, since those cannot be found for the second network it clearly outperforms by adjusting continuously to the current situation.

A strength of Q-learning is robustness against model errors. Since it does not track individual vehicles through the network, it is easier to operate at conditions close to saturation when shockwaves disturb the micromodels.

Given that networks like network 2 with 400-800m distance between intersections and all intersections equally spaced are very rare, it will be difficult to deploy Q-learning controllers in practice. A lot of guidance has to be given to such an algorithm, which effectively makes it behave like a control strategy with dynamic programming again.



8.3 Adaptive green wave with speed advice

8.3.1 Theory

As already explained in Figure 8-1, the optimal bi-directional green wave requires a network with equally spaced intersections. Since this is very rare a few methods can be employed to compensate for this like the leading and lagging left turns described in [31]. However, this is often still not sufficient as is demonstrated in Figure 8-3:

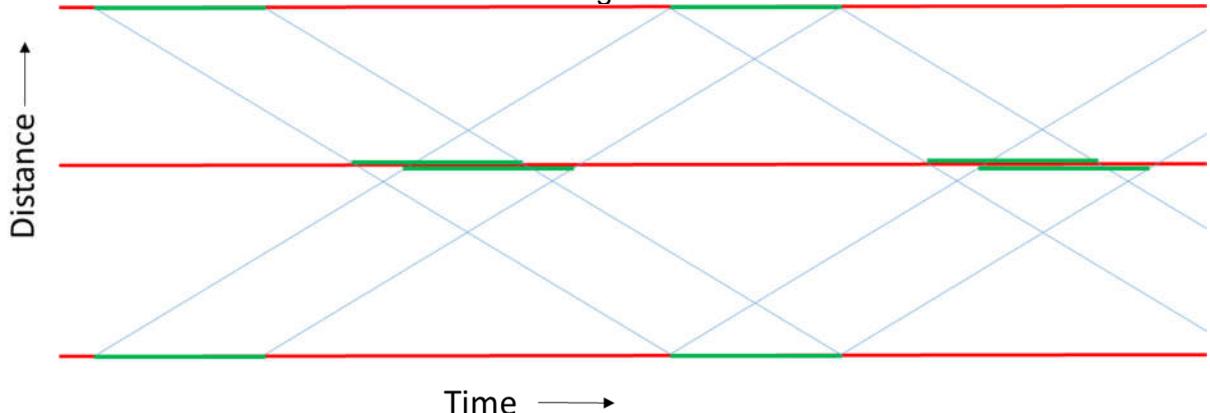


Figure 8-3: Offset creation with lagging/leading left turns

The second intersection is at 45% of the distance between the first and the third. This is about the maximum that can be compensated by assigning the left turns together with the individual main directions. When combined with speed advice, a single intersection that is closer than the others can be compensated by lowering the speed limit as demonstrated in Figure 8-4:

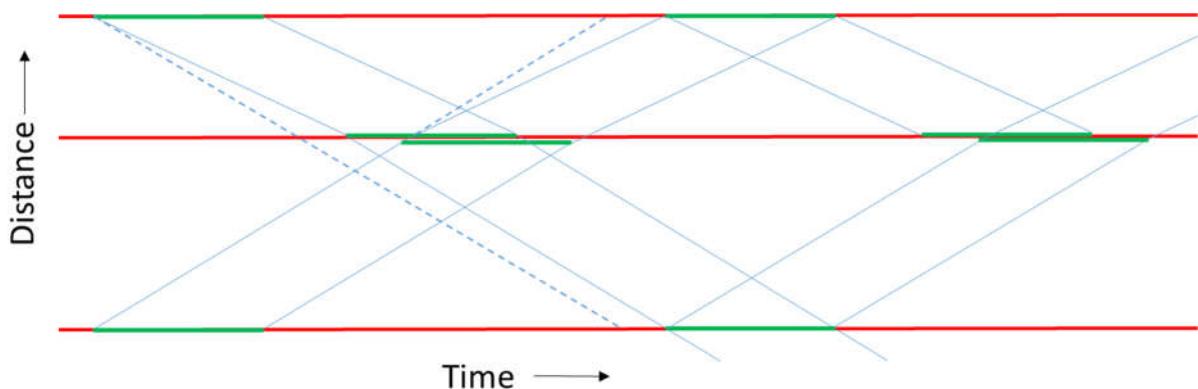


Figure 8-4: Combination of lagging/leading left with speed advice

In this case the second intersection is at 39% of the distance of the total distance between the first and the third. With a speed advice that is half of the maximum speed limit, the second could even be at 33%. The dashed line indicates where the vehicles would end up when following the maximum speed limit. In Figure 8-2 and Figure 8-3, the total distance between the first and third intersection remained the same. With the speed advice, only the distance between the first and second was decreased. This means that if the third intersection is further away, the cycle time has to be increased. This may not always be desirable, especially if the main direction takes up most of the cycle time. In such a situation the platoon would have to go over the speed limit to make it within the cycle that is too short. Splitting the platoon is then the best solution as is shown in Figure 8-5.



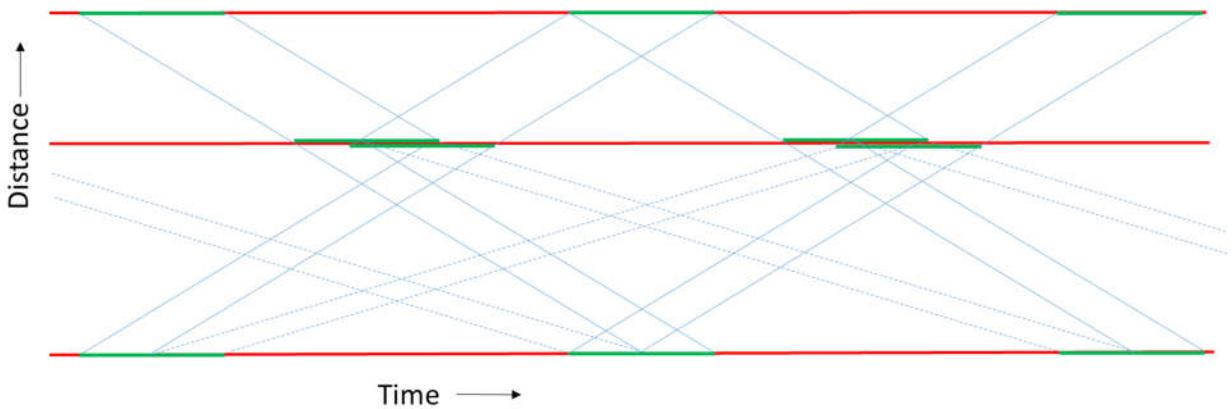


Figure 8-5: Splitting platoon with speed advice

It should be noted that this example has relatively short green phases in the green wave, this is to keep the visualization comprehensible. With a larger share of green the difference in speed advice for the part of the platoon that is split off will be smaller. With a 10 second green phase on a 60 second cycle the second half of the platoon needs to be delayed by 55 seconds, when there is 30 seconds of green time, this becomes 45 seconds.

In Section 778.2 it could be concluded that adaptive control using dynamic programming is the best solution in most situations. This especially true for the Helmond network in MAVEN that has three closely spaced intersections and another that is further away. This same network was also used in [31], but in the meantime the traffic demand and safety constraints have been updated. Therefore, new results of this deliverable cannot be compared to those results anymore.

This section will focus on the speed advice solution for green wave, which is approaching the problem from another direction than GLOSA with plan stabilization of Chapter 7. In the latter case the traffic controller is given total freedom on the control plan as long as it is kept after the first announcement of a time to green countdown, while in this case the speed advice is fixed inside the control algorithm. The vehicle will follow this speed while the traffic light controller is expected to adjust to the pre-shaped bidirectional platoons. The solution did not require the platoon splitting solution, as is further explained in the next section which presents the network, the derived control plan and the simulation results.

8.3.2 Simulation network

In light of the theory in section 8.3.1, there will be two types of experiments in this case study: adaptive control (ImFlow) with speed advice experiment and static control (SUMOonly) with speed advice experiment.

Both types of experiments take place on the network of Helmond, intersection 101, 102, 103 and 104. Figure 8-5 shows the network topology in maps (left) and in SUMO simulation (right).

The intersections for green wave study in this case study are listed under Figure 8-5. This network currently has adaptive traffic control system-ImFlow operating on this road stretch.

The network layout is suitable for green wave study due to the following characteristics: a) the corridor shaped network with consecutive intersections; b) for all four intersections, the highest traffic demands are on the corridor directions- signal group direction 8 and signal group direction 2 (see Figure 7-3); c) relatively minor traffic demands on the other directions, such as north-south directions and pedestrian/cyclist directions; d) speed limit on the network is consistently 50km/hr.



On the other hand, the network layout, like all real world traffic network, poses the following challenges for green wave study: a) the various distances between adjacent intersections, for example, the distance between intersection 101 and 102 is 179m while the distance between intersection 103 and 104 is 485m; b) asymmetric distances between two adjacent intersections from stop line to the next stop line, for example, the distance between intersection 102 and 103 eastbound is 200m while the distance between intersection 102 and 103 west bound is 164m; c) different traffic demands and different traffic directions for cyclist and pedestrian, which result in long and various intergreen for conflicts between VRU and vehicle directions.

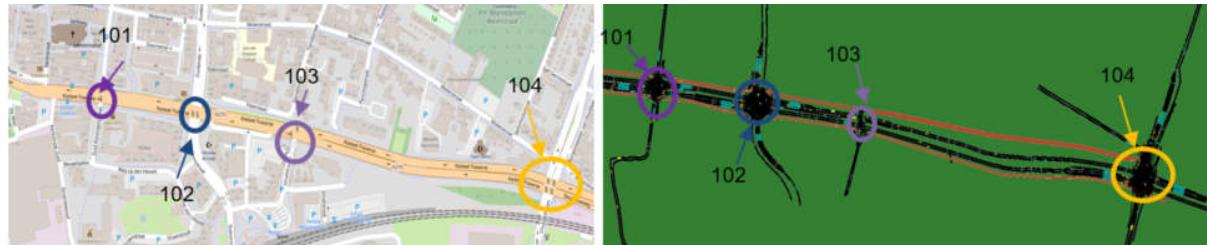


Figure 8-5: Case study of Helmond (left) and the simulation network in SUMO (right)

- Intersection 101, Zuid Koninginnewal/ Kasteel-Traverse
- Intersection 102, Zuidende/ Kasteel-Traverse
- Intersection 103, Penningstraat/Smalstraat/ Kasteel-Traverse
- Intersection 104, Churchilllaan/Burgemeester van Houtlaan/ Kasteel-Traverse

According to the aforementioned real world network layout of the case study, the signal control on the main directions 8 and 2, as well as the distance between intersections are re-created in Figure 8-6.

X-axis is the distance, which resembles the network length and the intersection size. Y axis is the time, which has a fixed cycle time of 70 seconds for all four intersections. 8 and 2 indicate the bi-directional traffic stream (eastbound and westbound) according to the signal group directions. These two directions are of our interest and will be referred to as SG8 and SG2 in the case study.

In principle, this fixed plan uses the theory in section 8.3.1 for unevenly distributed network, specifically, see Figure 8-4 and 8-5. This fixed plan takes into account the calibrated demand of each direction at each intersection. Therefore, the minimum green times are satisfied to achieve no-congestion network. In addition, constraints such as intergreen, minimum green, minimum amber and red also comply with the traffic configuration currently in operation. Lastly, the green times on direction SG8 and SG2 are maximized to achieve the optimal green wave.

After setting up the fixed time plan individually for each intersection, we can see from Figure 8-6 that the green time for SG8 and SG2, and for each intersection are different, for example, it is relatively simple to achieve green wave on intersection 102 and 103 due to long and symmetrical green time on SG8 and SG2, while there are limited green time possibilities on intersection 101 and 104.

Afterwards, the light and dark green lines can be drawn to depict vehicle trajectories, aiming to achieve green wave. The dashed lines between intersection 101 and 102 indicate that the speed is adjusted lower than speed limit of 50km/hour to catch green wave. In this case, the speed advice is 23km/hr on the SG8 direction (eastbound) and 18km/hr on the SG2 direction (westbound). This is quite slow, but with automated vehicles lower speed advice can be given as compliance is not a problem.



For adaptive control of this case study, the state plan of ImFlow considers the same calibrated demand and traffic constraints. Therefore, the signal stages and split of each stage are comparable as well. Since the goal of adaptive control is to achieve optimal traffic efficiency (low impact) instead of green wave, we adjusted the speed of road and vehicle speed between intersection 101 and 102 according to the fixed time plan. Assuming the adaptive control takes the new traffic arrival pattern and traffic speed into consideration, the adaptive plan would automatically adapt to achieve green wave when this is optimal.

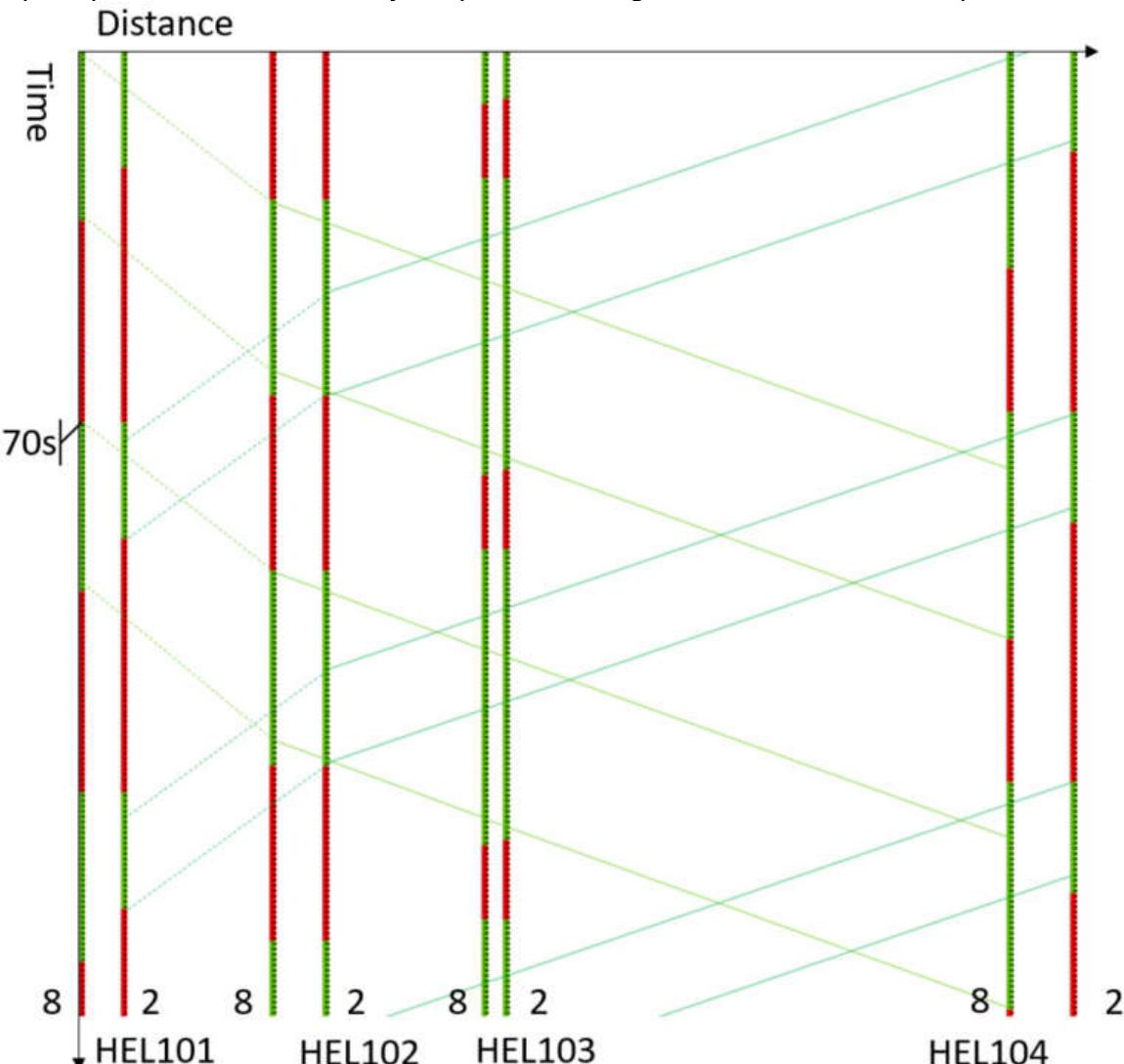


Figure 8-6: Fixed signal plan for the network, with maximum green wave bands.

In summary, four simulation scenarios are set up to perform green wave simulation experiment on the case study network, see Table 8-2.

Table 8-2: Case study scenario set-up

Scenario	Description
ImFlow Baseline	Current adaptive traffic plan in operation
ImFlow SA	Current adaptive traffic plan with speed advice, 23km/hr and 18km/hr between intersection 101 and 102
SUMOonly Baseline	Static traffic plan with no speed advice
SUMOonly SA	Static traffic plan with speed advice, 23km/hr and 18km/hr between



8.3.3 Results

The results of all four scenarios are collected after simulations experiments (10 runs per scenario and average results are used), see Figure 8-7.

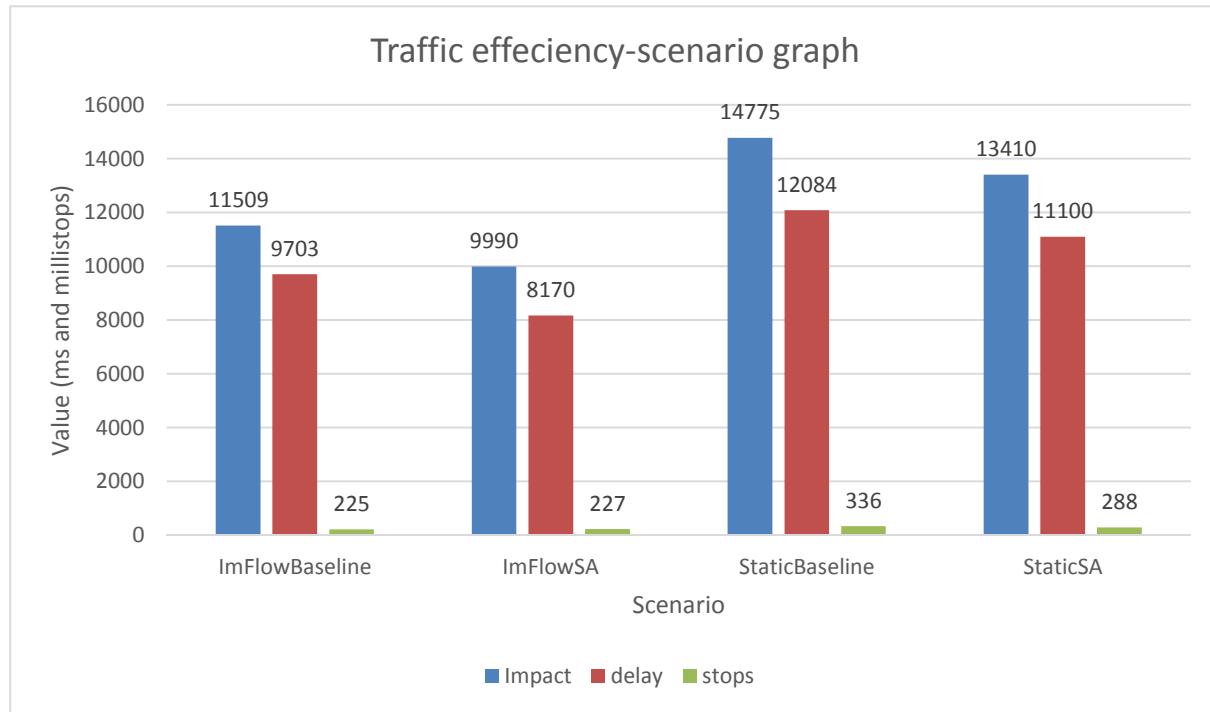


Figure 8-7: Impact, delay and stops comparison of four scenarios

As expected, the adaptive control (ImFlowBaseline) shows lower impact, delay and stops comparing to static traffic plan (SUMOonlyBaseline). Since the adaptive control aims for traffic efficiency, the arrivals of vehicles, cyclist and pedestrian are evaluated at run-time and the lowest cost solution is chosen to optimize the network performance, see section 7-1 for the benefit of adaptive control.

The adaptive control with speed advice scenario shows better network performance comparing to adaptive control, which is the contribution of less delay from speed advice.

The static plan with speed advice scenario shows an impact decrease of 9.2% comparing to static plan without speed advice. This can also be explained in the lower delay and stop, thanks to speed advice aiming for green wave.

To study the stops for red signals in the corridor itself, sets of E3 detectors (see [wiki](#)) are configured on the simulation network. A set of E3 detector (entry and exit) can detect the mean vehicle halt below a configured threshold, which can indicate the stops for red signal on one link or one stretch. It adds the possibility of filtering for vehicles that pass SG2 or SG8 on 2 or more consecutive intersections and not consider vehicles that entered from a side road.

In the case study network, eight sets of E3 detectors are configured, see the x-axis labels in Figure 8-8. The name of the E3 detector indicate the position, for example, 102SG8 means the detection of vehicle encountering red signal at intersection 102, direction SG8.



The results of vehicles stopping for red are shown in Figure 8-8. It can be observed that, for static plan, vehicles stop for red generally less with speed advice on (see the different between purple line and green line). While for adaptive control, the speed advice with ImFlow has some positive effects on some of the intersection and not on the others (see the different between blue line and red line). As explained in Chapter 7, the adaptive control is flexible and optimal designed for lowest cost control for the whole network. The control algorithm still has the same relative importance for stops compared to delay time for other traffic directions and pedestrians/cyclists. Therefore, the results of green wave plan are not significantly observable when it comes to stops in the corridor. It should also be noted that the stops are already lower than for static control with speed advice, resulting in less room for improvement. Delay time is significantly reduced because traffic generally arrives more synchronized, leaving more green time for other traffic.

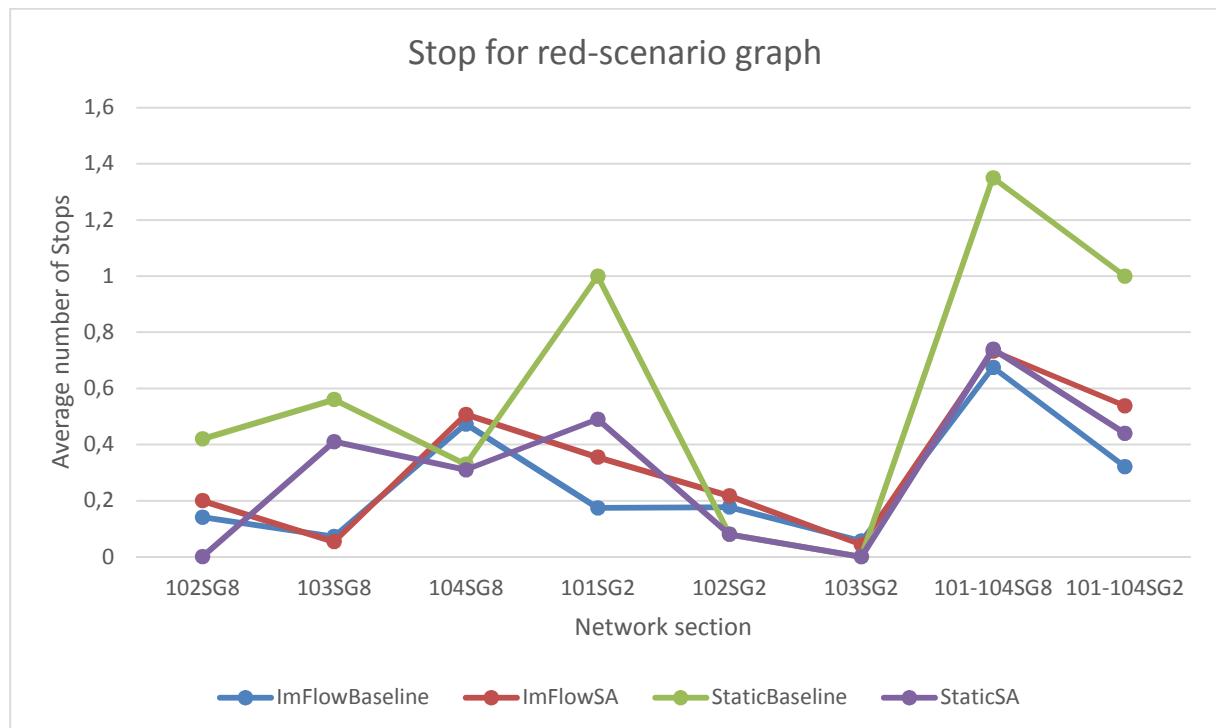


Figure 8-8: Stops for network subsections of four scenarios

To study the green wave improvement between with and without speed advice in static plan scenario, results are further demonstrated in Table 8-2. The positive increase from no advice to advice is significant while only two intersection/direction with already excellent green wave performance (due to less traffic direction in these two intersections) show no change.

Table 8-2: Case study scenario set-up

	102SG8	103SG8	104SG8	101SG2	102SG2	103SG2	101-104SG8	101-104SG2
SUMOonly Baseline	0.42	0.56	0.33	1	0.08	0	1.35	1
SUMOonly SA	0	0.41	0.31	0.49	0.08	0	0.74	0.44
Greenwave	100%	26.8%	6.1%	51%	0	0	45.2%	56%



Improv.							
---------	--	--	--	--	--	--	--

While the results for adaptive control are overall much better than for static control, the solution will not combine well with GLOSA-type speed advice. This will undermine the green band shaping, quickly restoring the old situation with unsynchronized platoon arrivals for large platoons arriving from the start of the corridor. During simulations it was observed that for the static control solution the platoons sometimes had to stop due to variations in queues of vehicles origination from upstream side roads. Adding a dynamic GLOSA-type speed advice on top of that would also eliminate those stops. Therefore, the static solution will be used further in WP7 of MAVEN in combination with GLOSA to achieve maximum green wave performance improvement. Additionally, platoon splitting will be applied for vehicles arriving at the first intersection of the corridor.



9 Special road user categories

9.1 Prioritized public transport

Most of the use cases of MAVEN rely on predictable control and key advancements have been made to increase the predictability of the control algorithms used. The use cases speed change advisory (UC7), the green wave (UC13), signal optimization (UC14) and negotiation (UC15) all rely on a high predictability of the time to green. Therefore, it is important to look at the disrupting factors for this predictability.

A key disruption is a priority request of a public transport vehicle. While this is in line with policies of road operators to offer a reliable public transport service, it disrupts the planning of traffic light controllers. This is because the priority call generally only arrives 10-20 seconds before arrival, further upstream the arrival time is still considered too uncertain due to bus stops and other intersections.

The first action that can be taken is letting the public transport vehicle sign in further upstream. This can be done by connecting to the system inside the bus that indicates whether a stop is requested in combination with monitoring the bus stop area and connecting to upstream intersections.

Another possibility is to inform road users about the reason for the disruption. This way, the system will not be perceived unreliable and user acceptance will increase. The XCycle project did this by displaying a bus symbol on the countdown panel for cyclists whenever a bus priority call was active. More on this system can be found in Section 9.2, the sign is shown in Figure 9-1:

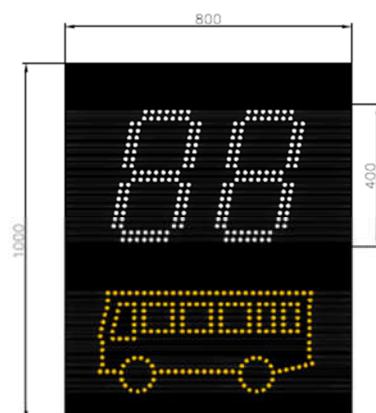


Figure 9-1: XCycle dynamic sign with capability to inform about active bus priority

For MAVEN this can be done with a special field in the SPaT message. This is the “signalPriorityIsActive” of the “intersectionStatusObject”. Setting this field active would open the possibility to show a bus symbol on an HMI.

9.2 Vulnerable road users

Traditionally, pedestrians and cyclists are disturbing factors for adaptive control when it comes to predictability. This is due to the detection method, which is usually with a push button or a loop close to the stop line. Once they are detected they are immediately waiting and therefore the control plan has to be adjusted on a short horizon to keep the waiting time low. However, when the predictability of the control is actively improved by new policies like explained in Chapter 0, this information can be fed to pedestrians and cyclists as well. Using an app or a sign like shown in Figure 9-1, the information about the time to the next green phase can be easily disseminated to the VRUs. Especially for cyclists this is interesting as



they can adjust their speed to arrive at the intersection during green. If the cyclist has to slow down from its desired speed, it will save energy, which can be used after the intersection to go faster again, compensating for the lost time. The alternative of stopping and accelerating for standstill will not save nearly as much energy. This effect should more than offset the small increase of delay time due to lost flexibility of the control algorithm.

To quantify the effects on cyclists, a simulation study was done in Cooperation with the XCycle project. The research chose the Helmond network because of its characteristics of multiple intersections on a corridor in the city centre of Helmond, shown in Figure 9-2.

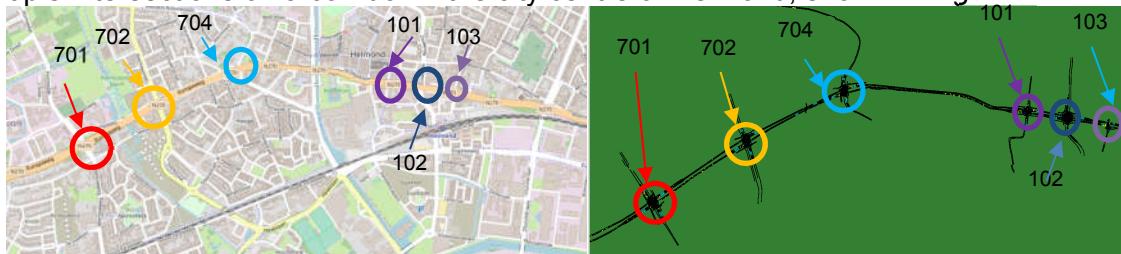


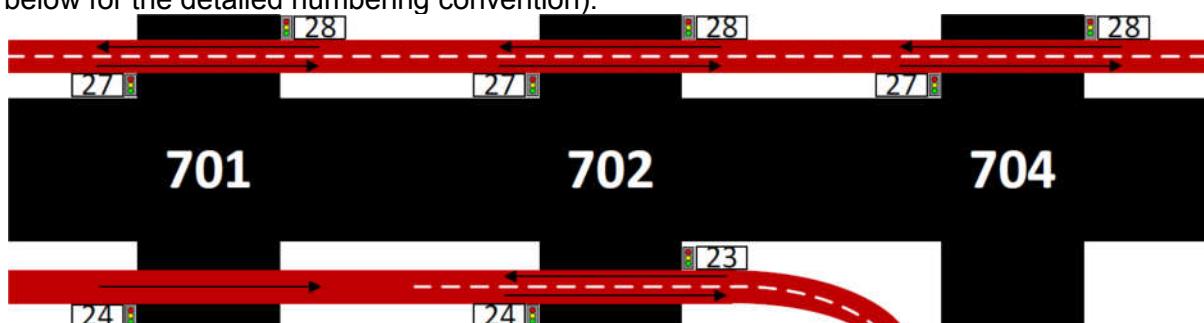
Figure 9-2: Six consecutive intersections of the case study in Helmond (left) and the corresponded simulation network in SUMO (right)

The intersections in the figure are the following:

- Intersection 701, Hortsedijk/ Europaweg
- Intersection 702, Boerhaavelaan/ Europaweg
- Intersection 704, Prins Hendriklaan/ Kasteel-Traverse
- Intersection 101, Zuid Koninginnewal/ Kasteel-Traverse
- Intersection 102, Zuidende/ Kasteel-Traverse
- Intersection 103, Penningstraat/Smalstraat/ Kasteel-Traverse

The points of interest on this real world network layout are six consecutive intersections that contain bicycle lanes (only the east-west/west-east directions are considered here). The Helmond-based simulation network is modelled and calibrated in SUMO and it focusses on the traffic control related scenarios, primarily bicycle traffic controls. Supported by predictable adaptive control, a new approach of bicycle detections is applied to the case study. Current approach of detection type at these intersections are either no detection or actuated, which is a push-button at the stop line. Providing speed advice using a push button is nearly impossible, because the arrival of the cyclist cannot be predicted and the traffic light controller will try to give green as soon as possible after the button is pressed. In this solution upstream detection will be used to predict arrivals and plan the green phase in advance. In the real-world this detection could be implemented either with an app or with infrastructure sensors.

As shown in Figure 3, these bicycle lanes are composed of dual carriageway (car lanes in the middle) with one-way/two-way bicycle lanes, for example, link 27, 28 and link 24 on intersection 101, or link 27, 28 and link 23, 24 on intersection 702. Additionally, they can be composed of single carriageway with one-way/two-way bicycle lanes, for example, link 27, 28 on intersection 704, or link 24 on intersection 103 (Referred to as Link 103₂₄, see Note 1 below for the detailed numbering convention).



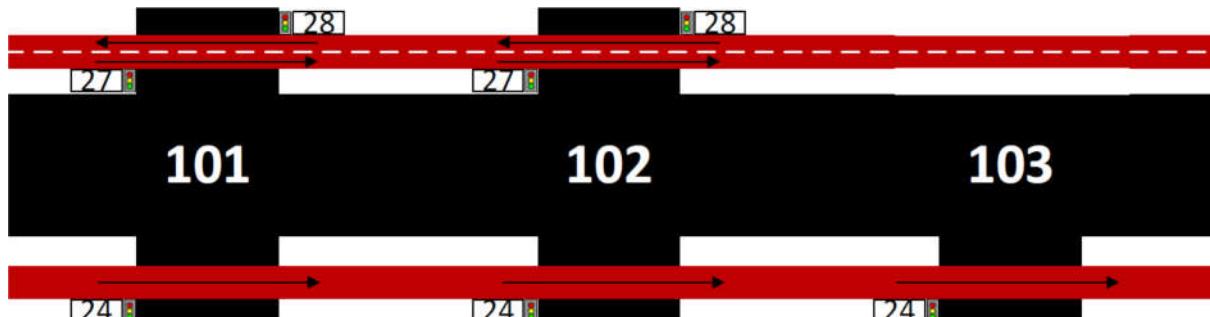


Figure 9-3: From west to east, schematics of six consecutive intersections road layout, showing only the bicycle lanes (red) and corresponding signal controls

The numbering of signal controls replicates real world signal group situation. Signal group number 24, 23, 27 and 28 are used here. The bicycle link segment at the intersection also follow this numbering convention, i.e. $701_{27,28}$ refers to the two signal head 27 and 28 at intersection 701; Link $701_{27,28}$ refers to the two-way bicycle lane segment at intersection 701. The goal of simulations in this paper is to compare the performance of the current signal control plan with a control plan including GLOSA functionality, targeting the bicycle signal groups. Six scenarios, scenario 0 to 5 are designed in order to compare the effect of baseline (scenario 0), single and multiple intersections with GLOSA, as shown in Table 2.

Table 9-1: Simulation scenario designs overview

	Description	Traffic control configuration	GLOSA
Scenario0	Baseline, do nothing	Current control plan	NO
Scenario1	$701_{27,28}$	Adapted control plan	YES
Scenario2	$701_{27,28} + 702_{27,28,24,23} + 704_{27,28}$	Adapted control plan	YES
Scenario3	$101_{27,28,24}$	Adapted control plan	YES
Scenario4	$101_{27,28,24} + 102_{27,28,24} + 103_{24}$	Adapted control plan	YES
Scenario5	$701_{27,28} + 702_{27,28,24,23} + 704_{27,28} + 101_{27,28,24} + 102_{27,28,24} + 103_{24}$	Adapted control plan	YES

The adapted control plan consists of three elements, the first is adding detection upstream in order to predict the arrival of cyclists. Additionally, constraints in the control plan that conflict with predictability have been removed and finally, the possibility to configure a weight for predictability as indicated in Chapter 0 was added. It should be noted that the main directions of the cars are in the same stage. For each scenario, sub-scenarios of different parameter sets (different c and EL) are configured and respective simulations are performed with 10 runs/sub-scenario and 2-hour simulation/run during the evening peak. As indicated before, cyclists are assumed to estimate their required speed themselves based on the TTG countdown. Since they learn quickly over time, when passing the system every day, a speed advice is calculated and applied to the cyclists in this simulation and is assumed to be fully complied by the cyclists. Speed advice is applied from 200 meters before each stop line and they are subject to a speed range of 6- 20 km/h. Slower or faster speeds are not considered realistic.

During the simulation, delay time and the amount of stops are tracked for every traffic participant. The evaluation parameters are the same as used in Chapter 6 in the formulae 7-9, 7-10, 7-11 and 7-12.

Results

The results showed a clear success for the green wave by applying GLOSA. In the baseline cyclists could pass the green light without stopping in only 44% of the cases. The effect on green wave success was already optimal when $SBW=60$ was configured for all intersections. This resulted in 64% green wave success rate. At the same time the effect on traffic efficiency was limited with an increase of the impact by 4.9% from an average impact of 26.6s to 27.9s. The MRE dropped from 35% to 12% and PC from 7.6% to 4.1%. With higher values of SBW and setting $EL=1$, this could decrease further to an MRE of 9.1% and PC of



2.7% ($SBW=480$, $EL=1$). However, this was at the cost of traffic efficiency, with an increased impact of 32.6s. Looking at the subnetworks, the scenario of 701, 702 and 704 was most successful with GLOSA success of 72%. When only 101, 102 and 103 were enabled the success rate was 60%. Enabling isolated intersections was less successful than several intersections in a corridor with 64% success for only 701 enabled.

Looking in more detail, the figures of merit: Impact, MRE and PC are extrapolated, and results are analysed for traffic efficiency and GLOSA functionality. Comparing to baseline scenario 0 (flat line with FOM value of 1), Figure 9-4 shows that for all other scenarios (with adaptive GLOSA), the synthesized performance figure $FOM_unified$ decreases with increasing weight in the cost function to configure for predictability. When increasing the weight from 0 to 720, the figure of merit for scenario1-5 tends to converge at a low value around 0.16, which showed a 84% decrease comparing to the baseline and around 60% decrease comparing to $SBW=0$. The case of $SBW=0$ already has the adjusted configuration where cyclists are detected upstream and certain control constraints are removed. Unexpectedly, there is one exception: scenario1 (only intersection 701 with GLOSA) already shows good results with $SBW=0$; increasing the weight to 60 induced a worsened result from 0.10 to 0.17. Intersection 701 is the entry intersection of the network with a high traffic demand. While other intersections receive the vehicles in platoons from upstream, this intersection has vehicles arriving from the west through a Poisson arrival process. Increasing SBW was therefore less effective.

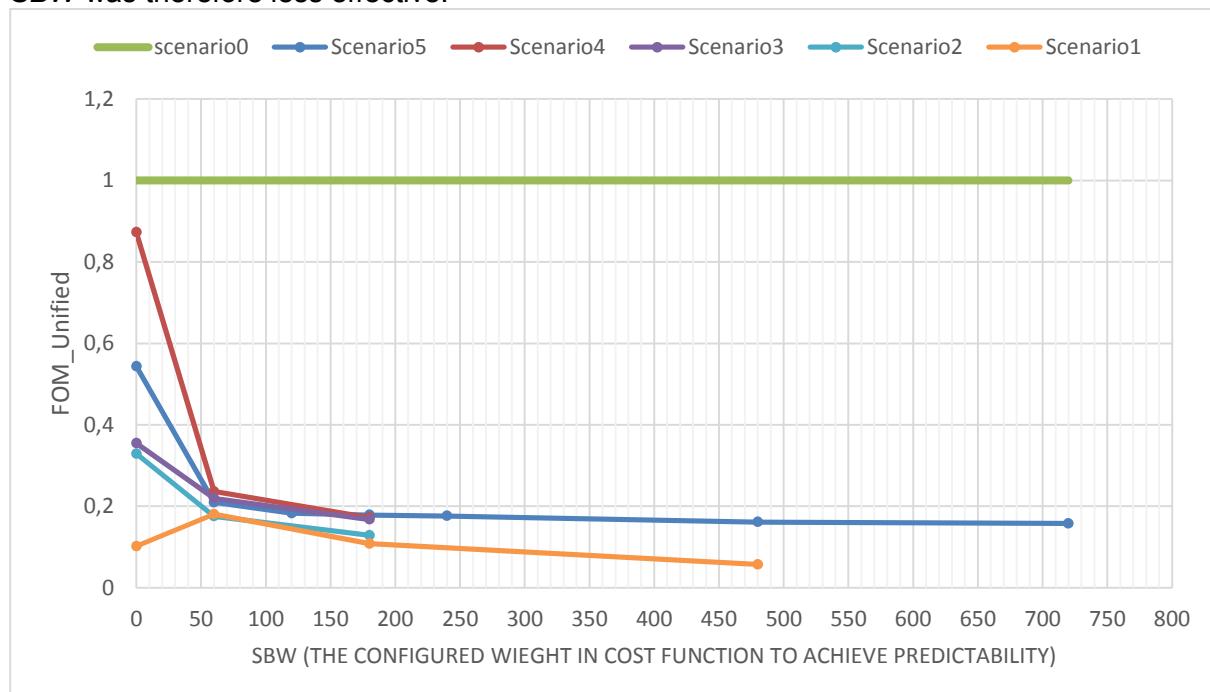


Figure 9-4: Relation of $FOM_unified$ to different weight in scenario 0-5



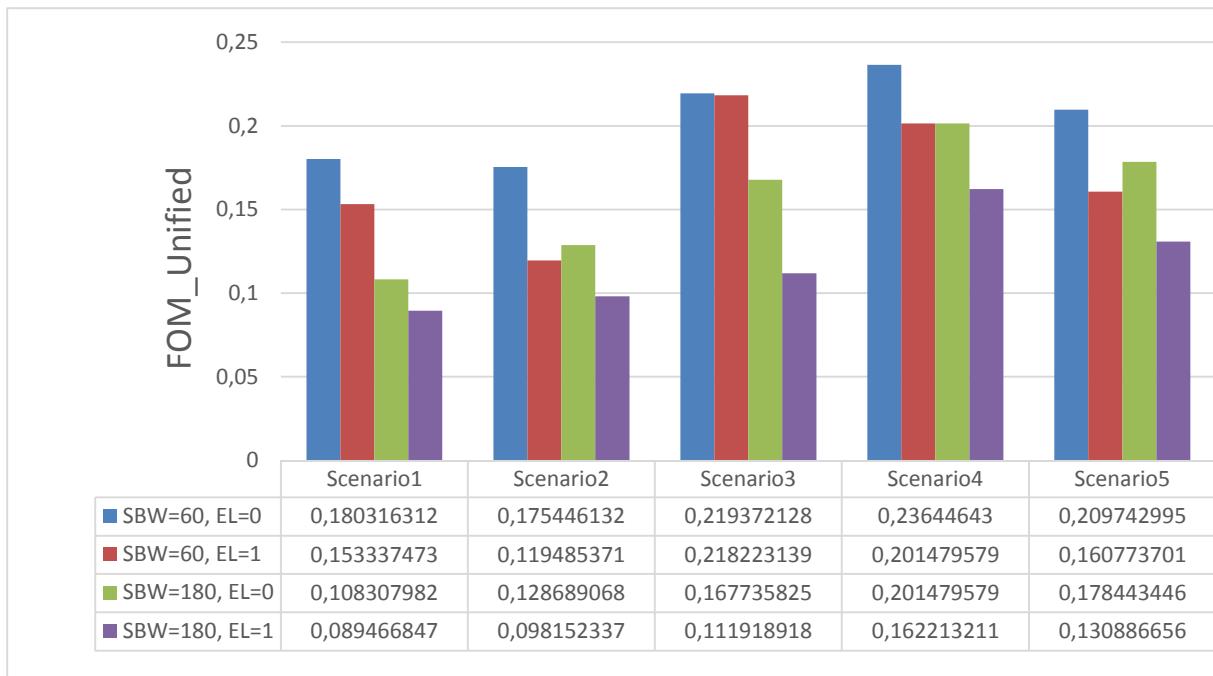


Figure 9-5: Relation of FOM_unified with different weight in scenario0-5 for EL=0 and 1

Furthermore, results of simulations with $EL=0$ and 1 are shown in Figure 9-5. With $EL=1$, which means extension of vehicle actuation for the next stage is disabled to have a more predictable plan, a visible decrease ranging from 10% to 20% can be observed for almost all sub-scenarios of scenario1-5. This feature is an effective way to configure more predictability for GLOSA for those signal groups because it can be customizable down to the level of a signal group at an intersection. Simulations for scenario 5 (six intersections with GLOSA) are conducted the most. Optimistic result of Scenario 5, with extension level ($SBW=720$, $EL=1$) shows a 9.65% decrease comparing to Scenario 5, without extension level ($SBW=720$, $EL=0$). Combining with Figure 4, it demonstrates that when SBW is at extreme high value, doesn't bring much benefit to further lowering $FOM_{unified}$. Setting EL to 1 can still have a better result.

These results show a high potential with green wave success increasing from 44% in the baseline up to 72% when the GLOSA function is used. At the same time impact on other traffic is kept minimal with an increase of only 4.9%.

It is important to consider the cyclist's behaviour was modelled in these simulations. While real cyclists are probably better at interpreting the countdown and aiming for the green, the tolerance to fluctuations of the behaviour model is higher than in reality. Therefore, the success rate is expected to be better in reality with the lower MRE and PC values that could be achieved by configuring more importance to predictability. The results also showed that consecutive intersections are more effective and very closely spaced intersections are less effective.

The trade-off between predictability and traffic efficiency was captured in a figure of merit measure. Increasing weight in the adaptive control algorithm and imposing rigid plan of no extension in the next stage, shows that green wave for cycling is feasible with ImFlow without deteriorating the performance of other conflicting traffic too much. Nonetheless, the configuration of constraints and the upstream detection of cyclists are essential for this application, especially for large-scale deployment, which needs careful calibration on the scenario 0 first before configuring the importance of predictability in the algorithm.

Future research on this topic can be carried out on this aforementioned attention point, to study if more flexibility of adaptive control can be kept regarding the trade-off between flexibility and predictability.



9.3 Prioritization management of automated vehicle platoons

Priority was already discussed in Section 9.1, but this section will look at options for giving priority to automated vehicles. Deliverable D2.1 mentions the priority management use case (UC10), but focusses on the trade-off between different vehicle classes. In the workshops held by the MAVEN project, however, it turned out there was no support for regular priority for automated vehicles. Such a regular priority would reduce the delay time by applying a higher weight to an automated vehicle in the cost function of the control algorithm. Automated vehicles could increase the efficiency of the traffic network, but this was considered too few to warrant a priority as it would stimulate usage of private cars over other modalities.

This means that the initial use case of priority management that would balance priorities between public transport, trucks and automated vehicles would basically fall back to the regular C-ITS service of green priority. This was already deployed in previous projects like eCoMove, and is being further scaled up in C-MobILE at this moment. Therefore, MAVEN came up with an alternative approach, which focusses on keeping platoons together, but does not give priority to the first vehicle in a platoon. It stimulates vehicles to join a platoon, which is beneficial for traffic control efficiency. Vehicles that actively stay together form denser platoons requiring less green time to let the same number of vehicles pass. In some cases this could increase the waiting time for other traffic if a green phase has to be extended for a large platoon, but it does make the control easier at downstream intersections. Therefore, overall traffic efficiency should increase. This measure could count on the support of road authorities that participated in the MAVEN workshops.

The implementation of the priority that keeps platoons together is relatively straightforward. It is implemented as a conditional priority that would result in a high cost for stopping the follower vehicles in a platoon if the first vehicle can pass through green. The cost also applies to so-called double stops, which means the platoon has stopped, but the next green phase would be too short to let the full platoon pass, once the first vehicle passes green, the priority applies to the rest of the platoon irrespective of having stopped already.

This concept was already applied in combination with regular green priority for a large platoon of automated trucks during the “experience week connected transport”, which was held from 1-5 October 2018 in the Netherlands. The trucks passed through many cities, including Helmond, where an ImFlow based solution was in place. Since this was done with trucks, regular priority is effective and justified because stopping a truck causes much more emissions than a regular vehicle due to its weight. MAVEN modified this to only keep the priority for keeping the platoon together.

While the field trial of the “experience week connected transport” proved the technical feasibility of the concept, the impact on other traffic can still be investigated further. Therefore, the solution will be used in the evaluations of WP7 for further impact assessment and reported in D7.2.

9.4 Unmanned logistics

Within the MAVEN project there is also a focus on new concepts for unmanned logistics and service vehicles and how to manage them in an urban environment. Paragraph 9.4.1 gives a glance of applications of unmanned logistics now and in the past. Paragraph 9.4.2 gives an overview of current developments where in paragraph 9.4.3 the impact of these developments on the urban network are discussed. Paragraph 9.4.4 sums up research questions for MAVEN and paragraph 9.4.5 sets an number of scenario's to be simulated.



9.4.1 Historical contest developments

Unmanned logistics is mainly developed out of specific tasks to be fulfilled by a system. The “first” autonomous vehicles were forklifts which processed orders and distributed them within a warehouse. However, in the early 20th century a MailRail tunnel system was built underneath the City of London due to an overcrowded road network above. The idea was inspired by the Chicago Tunnel Company. The system ran from 1927 to 2003 and at its peak carried four million letters daily between the main sorting offices.



Figure 9-6: Left the MailRail, right the AVG used at ECT(TN)

The main developments in the 20th century were on automating specific tasks carried out mostly within its own infrastructure like the Automated Guided Vehicles of the ECT terminal in Rotterdam which was opened in 1993. Similar development took place in public transport, a good example is the VAL-system (Véhicule Automatique Léger) which operates since 1983 after extensive testing. Siemens presented in 2018, a joint research and development project with ViP Verkehrsbetriebe Potsdam GmbH, the first autonomous tram which is tested in Potsdam in real life traffic situation



Figure 9-7: Left the VAL Lillie, right the first autonomous Tram driving in Potsdam

Nowadays there are numerous developments on autonomous driving and experiments introducing new concepts with more autonomous behaviour in, as it is called, the last mile delivery. A few examples are the Volvo In-Car delivery system, the Online supermarket PicNic which delivers goods with small electrical busses and delivery droids like Domino's, which is going to test delivery robots in Germany and The Netherlands for pizza delivery.





Figure 9-8: Left to Right, Volvo in car delivery, PicNic small busses and the delivery robot of Domino's

The main motivation for the development of unmanned logistics are the expected benefits like

- Higher safety
- Lower impact on the environment
- Higher efficiency

For the MAVEN project it is helpful to have a short overview on current developments and to find specific trends that are applicable for the urban environment. Then, how can we deal with this regarding the use cases on platooning, speed change, lane change, emergency situations and signal optimisation

9.4.2 Current developments

Delivery by drones is not taken in to account because this has little relationship with the objectives of MAVEN to bring CAV's and signalized intersection and corridors into reality. Therefore, the focus lies on autonomous transport, which (at least) partly uses the same infrastructure as cars, bikes, pedestrians and Public Transport in urban areas. Underneath just a grasp out of current developments where there are three types of unmanned logistics to distinguish at the moment, trucks, shuttles/pods and robots.

Trucks

In the development of autonomous trucks big steps are made. Mainly level 2 is put to the test also in combination with platooning. Level 4 should be realised in the next few years. At the moment Embark drives with Level 2 trucks between Texas and California. DHL Deutsche post will start testing with self-driving trucks with equipment from NVIDIA and ZF.



Figure 9-9: Left to Right, Startup Embark, DHL start testing

In regards of trucks, the dimensions and configuration of trucks do not alter as yet.

Shuttles/Pods



Starting in 2019 the Parkshuttle (initiative of the Metropool region Rotterdam started in 1999), which operates now on its own closed network but crossing public roads, will extend its line to the public road under the experimental law for self-driving vehicles. 2getthere is the developer of the system.



Figure 9-10: Left to Right, 2getthere, Easymile, Renault, conceptcar VW

EasyMile developed pods, which can convey up to 10 persons. The pod has a maximum speed of 15 km/h. The pods are tested, among other places, in Appelscha driving at a bicycle lane, in Frankfurt on the Campus and in Paris on the Charles de Gaulle bridge.

Vehicle manufacturer Renault has introduced the Renault EZ-PRO, an electric, autonomous robo-pod concept for the future of last-mile delivery, which has been developed in partnership with delivery firm DPD. The front POD is occupied with an employee who oversees the delivery service. The other pods will follow the front pod but also can deliver independent parcels at the delivery point. The car has the specifications of a minivan.

Robots

As mentioned in the previous paragraph Domino's starts with delivery by robots (Starship). In Australia delivery of mail was tested in Brisbane. Issues that there has to be a postman nearby and that only one address can be delivered per run are yet to be solved. There are more initiatives and developments. See pictures below:

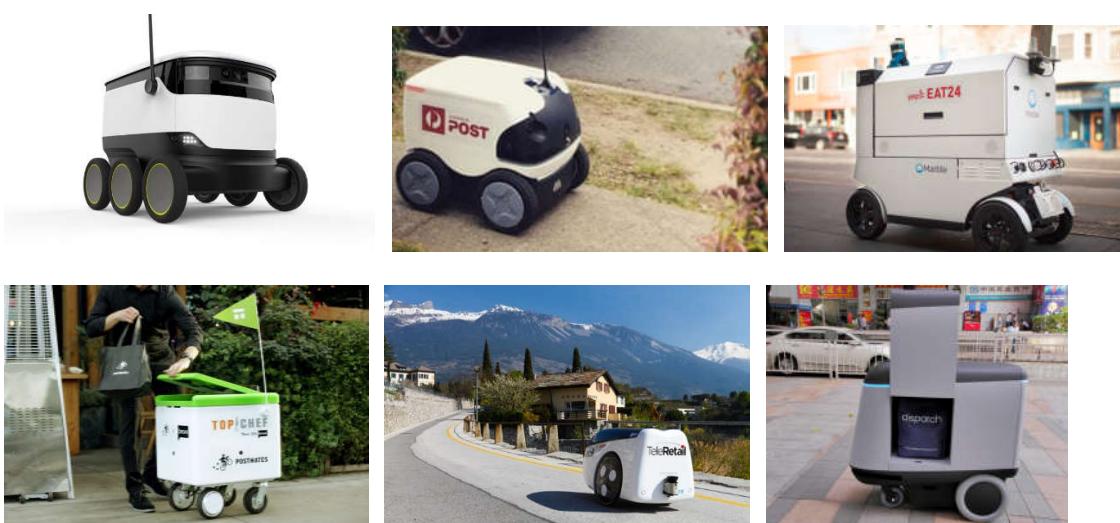


Figure 9-11: Left above to Right below, Starship, post, Marble, Nuro, Postmates, Telerital and Dispatch

These robots seem to use only the bicycle area or the sidewalks.

9.4.3 Impact on urban network

A first question could be “are the unmanned vehicles in practice a new modality” or is it and will it be another form of the existing modality. Looking at the evolution of transport with cars we see that they at first occupy the road space designated to pedestrians, horse carriage



and trams. Nowadays these groups sometimes use the same space but more often a big part is dedicated to cars.

As mentioned in the previous paragraph there are three trends at this moment. We can look at:

- Types of transports
- Forms of distribution
- Impact on the urban infrastructure
- Could steering from the government be feasible

Trucks

On main arterial roads, which have city to city traffic trucks will be part of the traffic flow. Only if there are environmental reasons to prohibit this. These trucks mostly occupy the same space as the cars and will, in urban environment, not have their own infrastructure. In relation with energy saving (emissions if not electrical or running on hydrogen) the policy could be to support platooning and prioritising. For trucks that have no real business to be on this specific arterial road prioritising could be disabled

Shuttles/Pods

Looking at the pilots and deployment of shuttles and pods there is a more diffuse situation in urban areas. Because of safety precautions most deployed systems, at this moment, operate on own infrastructure. During the test in Appelscha, where the shuttle used the bicycle lane, the shuttle hindered the bikers because there was hardly room to overtake/ get past the shuttle.

Depending on types (dimensions and speed) the shuttles can use bicycle and car infrastructure. If there is a large and specific number of shuttles/pods using the same part of a route there will be impact on the usage of the infrastructure. Here specific infrastructure or their own lane at a signalised intersection is possible. Maybe space can be shared with public transport lanes like cabs often have permission to do so.

Distribution of packages, road cleaning service, waste collection, urban green maintenance etc. are mostly based on an erratic kind of time table. This could be a point that the local council can cooperate with the distributors to develop a more specific schedule of time frames where (for instance outside the peak hours) distribution is more facilitated with infrastructure, curb side management (using public transport stops, less need for parking space, higher need for loading/off-loading bays) and priority/buffering/platooning at intersections.

These measures can keep the network more robust. Of course, this should be in combination with urban planning where the use of hubs, last mile distribution etc is taken into account.

Robots

Robots seem to be destined to use the side walk and bicycle lanes. Robots will probably mostly be used for delivery over short distances like a borough. The robots will originate from a distribution hub of any kind like Domino's or a post distribution centre or pick up goods at a bus/ shuttle stop. Near these distribution hubs, specific infrastructure could be realised. Also at intersections, lanes to create or facilitate platoons is feasible.

The legislation for deploying robots on the side walk is generally simpler than using the main road. So the more "simple" robots will probably use the sidewalks mainly.

9.4.4 Unmanned Logistics and MAVEN

The focus of MAVEN lies at signalized intersections and corridors.

Points of interests of handling autonomous logistic vehicles are:



- How to handle different characteristics of unmanned logistic vehicles stand alone and in groups apart and mixed between “normal” traffic
- Tactical and functional aspects of buffering and prioritizing at intersections and corridors also using techniques like geo fencing and different weight factor depending on type of transport and destination
- Curb-side handling nearby intersections. Should buffer locations be also the stop-and-distribution point?
- Handling uncoordinated approach of unmanned units versus scheduled approach (time tables and time windows) and effects of the network saturation
- Unmanned units that are ETSI compatible versus non communicating or dialect talking units

9.4.5 *Simulation approach*

To see the impact of different types of automated units these could be simulated in SUMO. It should be a picture of traffic in about 20 years. Scenarios that can be considered are:

- Shuttle simulation with and without scheduling to create small platoons
- Simulate intersection with and without specific infrastructure for unmanned units, own lane, buffer area or nothing to see the impact on the performance and technical possibilities.

This also implies that the SUMO network needs to be adjusted to add a source for the shuttles. Most suitable is the southern arm of intersection 103 in the Helmond green wave network, which is currently only for public transport. This is a good place to buffer vehicles between the arrival of busses and send them either onto the main road, a dedicated lane or the cycle path. The scenario results will be included in D7.2.

The results of the 4 scenarios will result in guidelines for:

- The design and layout signalized crossings
- The design and layout main road, bicycle lanes and Curbs
- The design of the LTC software and hardware
- The design of an effective scheduling and buffering strategy





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

10 Conclusion and further research

10.1 Conclusion

The main objective of this deliverable is to provide a joint understanding of work package 4, road automation, from the project partners.. While MAVEN has a separate WP for evaluation, preliminary results of some components were important to prove the algorithm effectiveness.

Since this is the final deliverable of WP4, it is good to look back at the initial objectives of the work package. The scheduling and signal timing strategy for automated connected vehicles was successfully developed to upgrade both adaptive and actuated existing control strategies. Both also include multiple-intersection scheduling elements targeting green wave and GLOSA. The queue estimation models give lane dependent information beyond the state-of-the-art like envisioned in the objectives. A new bottom-up routing strategy with micromodel-level data fulfils the objective of a local-level routing algorithm to provide routing advice to vehicles or platoons. Lastly, scheduling and infrastructure measures were proposed for special road user categories like unmanned logistic vehicles, while a deeper analysis was carried out to investigate the benefits VRUs get from the new signal timing strategy. Therefore, all objectives of the WP were successfully achieved.

The most significant results of this deliverable are about the queue model and plan stabilization for adaptive control. Data fusion of information from traditional detectors and automated vehicles resulted in up to 40% reduction for the average error of the queue length estimation. The largest benefits were visible for high traffic volumes, because this increases the chance of receiving vehicle information at 20% penetration rate. For adaptive control stabilization, results showed 25% reduction in average prediction error, while maintaining similar traffic efficiency. More advanced parameters were added to combat specific side effects, like the prediction stagnating at a certain value due to a green extension. This resulted in a small further improvement, but most notably in a solution for the stagnation problem. Green wave extensions to the actuated and adaptive strategies showed potential for improving the green wave performance, while static control remained the best option for guaranteed green waves. Therefore, a new methodology to design static control plans for automated vehicle platoons was developed. Depending on the local traffic management policy, a certain strategy can be chosen.

More detailed conclusions are listed below:

- Analysis of the state-of-the-art revealed that among the four types of traffic control (see Section 2), semi-fixed time control is most commonly used for contemporary GLOSA solutions. Adaptive control has advantages due to its flexibility in the control algorithm yet it needs to be stable enough to be accepted by users.
- The LDM is a central point for all cooperative functionality. It can be seen as a data array with location references, which stores dynamic data from various relevant actors. Extensions for automated driving were made and successfully tested by using it for the queue modelling research.
- Position simulation with realistic measurement error emulation plays an essential role for testing queue modelling. The proposed advanced model was used successfully for the queue model research.
- The simple extension of SUMO for automated driving is a promising possibility to enhance the future traffic efficiency by the formation of platoons of automated vehicles. It will be implemented in work package 7.



- The lane-based queue length model still has room for improvement, mainly in situations where the traffic flow over multiple lanes is low and the detector measurements do not provide enough insight into lane changes between the detectors. This results in incorrect identification of vehicle counts contributing to particular lane queues. The resulting error may be corrected by CAV in case that their penetration rate is reasonably high, but for low traffic volumes and low penetration rates of CAVs there is insufficient opportunity for corrections. This is acceptable, because at low traffic volumes the impact of incorrect queue information is less important.
- The configurable cost function for plan stabilization of adaptive control with ImFlow shows significant effects even when c is set as small as 30. Increasing this value shows further improvement, leading to the conclusion that this parameter can be used to implement different policies with respect to level of importance of stability.
- The advanced cost function parameters α , β and EL were effective against stagnating predictions (so-called “flat line” where the prediction freezes at the same value for 3 or more seconds).
- The extension level (EL) parameter is the most straight-forward for combatting the aforementioned flat-lines. As it is basically a constraint, it guarantees certain flat-line situations will not occur anymore.
- Both values of c beyond 150 and the advanced cost function parameters α , β and EL show very small changes in traffic efficiency and plan stability. While a general downward trend for higher values of c could be observed, individual data points are inconclusive due to statistical noise. The scaling up study with cyclists concluded that only $EL = 1$ should be used while using α and β would severely complicate deployment/calibration costs with insignificant benefits.
- Applying the plan stabilization to generate speed advice for cyclists, shows that they can also benefit from the same system. Their green wave success rate increased from 44% in the baseline up to 72% when the GLOSA function with stabilization is used. At the same time impact on other traffic is kept minimal with an increase of only 4.9%.
- AGLOSA is an effective system for speed advice when an adaptive control algorithm is not present. It can reduce the average loss time of vehicles by as much as 50% when compared to a static control strategy and CO₂ emission per vehicle from 215g to 198g. In general AGLOSA performs relatively better with lower demand on isolated intersections, while for coordinated intersections the performance gains increase with increasing volume. The strategy highly favours ongoing traffic, which can be further increased by adding the coordination factor.
- The local level routing presented a good framework for route advice, which can handle both signalized and unsignalized intersections. The rerouting frequency should be configured between 180 and 300 seconds while spreading the route advice in time over the vehicles to prevent oscillations in the underlying model.
- The new method of designing green waves with automated vehicles taking into account the improved GLOSA capabilities could reduce the number of stops with up to 56%.
- Scheduling for platoon forming and infrastructure adaptations seem to be effective measures to efficiently handle slower unmanned logistic vehicles.



10.2 Further research

The goal of this deliverable was to explain the concepts and algorithms developed under WP4. The results serve to prove the viability of the algorithms, but are not intended as a full impact assessment. Therefore, the evaluation work continues in WP7, in which different penetration rates and different traffic demand levels will also be considered.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

References

- [1] MAVEN consortium, D2.2
- [2] COLOMBO evaluation paper
- [3] COLOMBO simulation environment paper
- [4] Wong, S.C., Group-based optimisation of signal timings using the TRANSYT traffic model, *Transportation Research Part B: Methodological*, Volume 30, Issue 3, 1996, pp. 217-244.
- [5] Katwijk, R., Bakri, T., Vukovic, D., Hogema, J.H., Providing Speed Advice to Vehicles Approaching an Intersection: Evaluation and Lessons Learned, *Transportation Research Part B* 92nd annual meeting, Washington DC, USA, 2013.
- [6] Van Vliet, K., Turksma, S., ImFlow: Policy-based adaptive urban traffic control First field experience, European Congress on Intelligent Transport Systems, Dublin, Ireland, 2013.
- [7] HCM2000, Highway Capacity Manual. Transportation Research Board. 2000.
- [8] Akçelik, R., Time-dependent expressions for delay, stop rate and queue length at traffic signals. 1980.
- [9] Comert, G., Simple analytical models for estimating the queue lengths from probe vehicles at traffic signals. *Transportation Research Part B: Methodological*, 2013, 55: 59-74.
- [10] Comert, G., Queue length estimation from probe vehicles at isolated intersections: Estimators for primary parameters. *European Journal of Operational Research*, 2016, 252.2: 502-521.
- [11] Cui, Y., Ge, S.S., Autonomous vehicle positioning with GPS in urban canyon environments. *IEEE transactions on robotics and automation*, 2003, 19.1: 15-25.
- [12] Friedrich, B., et al. Data fusion techniques for adaptive traffic signal control. *IFAC Proceedings Volumes*, 2003, 36.14: 61-66.
- [13] Ho, C.H., Hwang, T.L., Modeling real-time dynamic queue length for urban traffic control systems. In: *Intelligent Vehicles' 94 Symposium, Proceedings of the*. IEEE, 1994. p. 438-442.
- [14] Ledoux, C., An urban traffic flow model integrating neural networks. *Transportation Research Part C: Emerging Technologies*, 1997, 5.5: 287-300.
- [15] Liu, H.X., et al., Real-time queue length estimation for congested signalized intersections. *Transportation research part C: emerging technologies*, 2009, 17.4: 412-427.
- [16] Mück, J. Using detectors near the stop-line to estimate traffic flows. *Traffic Engineering & Control*, 2002, 43.11.
- [17] Mystkowski, C.; Khan, S. Estimating Queue Length Using SIGNAL94, SYCHRO3, TRANSYT-7F, PASSER II-90 and CORSIM. In: *Transportation Research Board. Paper Submitted for Publication and Presentation at the 78th Transportation Research Board Annual Meeting*. Washington DC, USA. 1998.
- [18] Quek, C., Pasquier, M., Lim, B.B.S., POP-TRAFFIC: A novel fuzzy neural approach to road traffic analysis and prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2006, 7.2: 133-146.
- [19] Ramezani, M., Geroliminis, N., Queue profile estimation in congested urban networks with probe data. *Computer-Aided Civil and Infrastructure Engineering*, 2015, 30.6: 414-432.
- [20] Sharma, A., Bullock, D., Bonneson, J., Input-output and hybrid techniques for real-time prediction of delay and maximum queue length at signalized intersections. *Transportation Research Record: Journal of the Transportation Research Board*, 2007, 2035: 69-80.
- [21] Tiaprasert, K., et al., Queue length estimation using connected vehicle technology for adaptive signal control. *IEEE Transactions on Intelligent Transportation Systems*, 2015, 16.4: 2129-2140.



- [22] Viloria, F., Courage, K., Avery, D., Comparison of queue-length models at signalized intersections. *Transportation Research Record: Journal of the Transportation Research Board*, 2000, 1710: 222-230.
- [23] Viti, F., van Zuylen, H., Modeling queues at signalized intersections. *Transportation Research Record: Journal of the Transportation Research Board*, 2004, 1883: 68-77.
- [24] Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. *Transportation Research Part C: Emerging Technologies*, 2005, 13.3: 211-234.
- [25] Van Zuylen, H.J., Viti, F., Delay at controlled intersections: the old theory revised. In: *Intelligent Transportation Systems Conference*, 2006. ITSC'06. IEEE. IEEE, 2006. p. 68-73.
- [26] Bellman, R. E. (2003). *Dynamic Programming*. Princeton, NJ: Princeton University Press.
- [27] Erdmann, J. (2013). Combining Adaptive Junction Control with Simultaneous Green-Light-Optimal-Speed-Advisory. Dresden.
- [28] Markowski, R. (2017). Evaluierung und Erweiterung der LSA-Steuerverfahren VITAL im Hinblick auf die Steuerung von koordinierten Straßenzügen. Dresden.
- [29] Priemer, Christian, Friedrich, Bernhard, A Method for Tailback Approximation Via C2I-Data Based on Partial Penetration, 15th World Congress on Intelligent Transport Systems, New York, United States, 2008.
- [30] Hillebrink, A. J., Comparison of optimal traffic signal control methods, Master Thesis, 2018.
- [31] Zhao, D., Dai, Y., and Zhang, Z. (2012). Computational intelligence in urban traffic signal control: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):485{494.
- [32] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):pp279-292, 1992.
- [33] Islam, F., Vreeswijk, J., Blokpoel, R., Impact analysis of ecoBalanced Priority and ecoGreen wave applications, 10th ITS European congress, Helsinki, Finland, 2014.
- [34] Z. Liang and Y. Wakahara, "Real-time urban traffic amount prediction models for dynamic route guidance systems," pp. 1-13, 2014.
- [35] R. Blokpoel, J. Vreeswijk and D. Leistner, "Micro-routing using accurate traffic predictions," *IET Intelligent Transport Systems*, vol. 6, pp. 380-387, 2012.
- [36] B. Tatomir and L. Rothkrantz, "Hierarchical Routing in Traffic Using Swarm-Intelligence," in *2006 IEEE Intelligent Transportation Systems Conference*, 2006.
- [37] B. Yang and E. Miller-Hooks, "Adaptive routing considering delays due to signal operations," *Transportation Research Part B: Methodological*, vol. 38, pp. 385-413, 2004.
- [38] K. Faez and M. Khanjary, "UTOSPF with waiting times for green light consideration," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pp. 4170-4174, 2009.
- [39] M. Khanjary, K. Faez, M. R. Meybodi and M. Sabaei, "PersianGulf: An autonomous combined traffic signal controller and route guidance system," *IEEE Vehicular Technology Conference*, 2011.
- [40] H. Chai, H. M. Zhang, D. Ghosal and C. N. Chuah, "Dynamic traffic routing in a network with adaptive signal control," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 64-85, 2017.
- [41] P. Li, P. Mirchandani and X. Zhou, "Solving simultaneous route guidance and traffic signal optimization problem using space-phase-time hypernetwork," *Transportation Research Part B: Methodological*, vol. 81, pp. 103-130, 2015.
- [42] X. Chen, Z. S. Yang and H. Y. Wang, "A multi-agent urban traffic control system cooperated with dynamic route guidance," *Proceedings of the 2006 International Conference on Machine Learning and Cybernetics*, vol. 2006, pp. 330-335, 2006.
- [43] L. Ximin, "Based on Hybrid Genetic Algorithm and Cellular Automata Combined Traffic Signal Control and Route Guidance," *2007 Chinese Control Conference*, pp. 53-57, 2006.



- [44] J. V. Kampen, "Route guidance and signal control based on the back-pressure algorithm," 2015.
- [45] T. V. Mathew, "Lecture Notes in Transportation Systems Engineering - Microscopic Traffic Simulation," *NPTEL Online courses*, pp. 1-20, 2017.
- [46] M. Treiber and A. Kesting, Traffic Flow Dynamics, vol. 2013, 2013, pp. 1-505.
- [47] H. Lieu, "Revised monograph on traffic flow theory," *US Department of Transportation Federal Highway Administration*, 2018.
- [48] Codeca, L., Frank, R., Faye, S., and Engel, T. (2017). Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation. *IEEE Intelligent Transportation Systems Magazine*, 9(2):52–63.
- [49] Watling, D. and van Vuren, T. (1993). The modelling of dynamic route guidance systems. *Bilingualism*, 1:159–182.
- [50] Herbert, W. and Mili, F. (2008). Route guidance: State of the art vs. state of the practice. *Intelligent Vehicles Symposium*, 2008 IEEE, pages 1167–1174.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Appendix A: Literature review

In addition to the literature review carried out for MAVEN D2.2, a more extensive review was carried out for the work in WP4. This is to ensure we built on the latest work and not duplicate it. A MAVEN representative visited the TRB conference in 2017 to study and discuss the state-of-the-art with active scholars. From the presentations and discussions the most relevant papers for the MAVEN project were selected and extensively reviewed. The results of that are presented in this appendix. Note that due to personal discussions, some papers are not from TRB, because an author referred the project to another paper that would be more relevant. For some papers there are also references that have been analysed in detail to understand the work completely, these are also listed at the end in the “citation” section of the reviews.

1. Intersection Control System for Cooperative Automated Vehicles (CAV)

Title
Developing an Optimal Intersection Control System for Cooperative Automated Vehicles
Summary of the work
<p>In this paper, an enhanced algorithm: Optimal Control Effort - real-time (OCRT), is developed to optimize the movement of CAVs through intersections (only single lane intersections are considered in this paper). The aim is to find a solution that reduces the delay while at the same time minimizing the control effort (i.e. avoid aggressive acceleration levels to ensure that the ride is comfortable for the passengers).</p> <p>This paper begins with proposing a comprehensive and extended vehicle model, using nonlinear equations of motion, Kinematic and dynamic constraints. The nonlinear features produce computational challenges to obtain a solution in real-time. Necessary simplifications such as convexifying the problem is used to guarantee a fast solution process. Through these simplifications, a good compromise between optimality and fast computation is achieved.</p> <p>Specifically, this paper solves the problem numerically instead of providing analytical solutions. The system presented in [1] is discretized using Model Predictive Control (MPC). Pontryagin's minimum principle is used to formulate the problem and then it is used in convex optimization.</p> <p>Since the developed intersection control is novel with no traffic light control, a scheduler is placed at the intersection. Time slots for each vehicle to cross conflict zones, within the intersection is allocated. With the introduction of the scheduler and time slots for each vehicle, a combination of speed advisory and traffic control is presented. Simulations with different traffic inflows for a major and a minor roads are performed. The major road inflows ranges from 500 veh/h to a maximum of 1200 veh/h for a roadway saturation flow rate of $q_c = 1700$ veh/h/lane. The minor road inflows range from 250 veh/h to 600 veh/h. Detailed vehicle information is captured during the simulation: velocity, acceleration, delay, CO₂ emissions, fuel consumption, etc. Note that the fuel consumption is computed along the vehicle trajectory using the VT-Micro model. The results are compared to the results of [1] (i.e. Optimal Control Nonlinear (OCN)) and when the intersection is controlled by a roundabout (R), a stop sign (SS) and a traffic signal (TL). The results of the (R), (SS) and (TL) approaches are obtained using the INTEGRATION software, using the same input as the proposed algorithm. Noted that the INTEGRATION software also uses the Rakha-Pasumathy-Adjerid (RPA) car-following [2] and acceleration model that are embedded in the proposed algorithm.</p> <p>The results also show that the proposed algorithm outperforms the other strategies</p>



producing lower delays up to 55 % and an average CO₂ emissions reductions of 10%, relative to the best intersection control strategy (in this case the roundabout).

One disadvantage of this proposed algorithm is the estimate time of the vehicle reaching the intersection, in order to compute the solution. Once that is achieved a real-time solution is obtained through convex optimization. Initial estimates of the arrival times can be based on historical data, however further research is needed to develop good initial solutions.

Results relevant to MAVEN

The focus of MAVEN is on automated vehicles which can communicate and cooperate with traffic lights control. The novel algorithm in this paper can enlighten traffic light control design in MAVEN in the following aspects:

The non-linear modelling of CAV behaviour is relatively accurate and it could be compared with simple. Although non-linear can cause computational problem, simplifying the problem such as convexifying can help to solve this numerical computation problem; This control algorithm applied cost function from [1]. The cost function can be comparable to the cost function in section 0 of this deliverable, D4.1. The RPA car-following model [2] can model the car-following behaviour of CAVs by calibrating the CAV parameters. This can be compared with the car-following model in SUMO and check its applicability to implement and load a novel car-following model into simulations of MAVEN; This algorithm had better results comparing to Optimal Control Nonlinear (OCN), (R), (SS) and (TL) approaches. Results are built and tested on single intersection network with arbitrary simplification. The usability of this control algorithm on complicated adaptive control intersection is yet to be tested.

Citation (multiple if relevant, e.g. multiple deliverables)

Youssef Bichiou and Hesham A. Rakha, Developing an Optimal Intersection Control System for Cooperative Automated Vehicles, Transportation Research Board Annual Meeting, 2017.

Referenced papers:

[1] Bichiou, Y., H. A. Rakha, and A. Roman, Developing an Optimal Intersection Control System

for Automated Connected Vehicles. Submitted to IEEE Transaction, 2016.

[2] Rakha, H., P. Pasumarthy, and S. Adjerid, A simplified behavioural vehicle longitudinal motion model. Transportation letters, Vol. 1, No. 2, 2009, pp. 95–110.

2. Model Predictive Control (MPC) and GLOSA in Connected Vehicle (CV) environment

Title

Combining Model Predictive Intersection Control With Green Light Optimal Speed Advisory In a Connected Vehicle Environment

Summary of the work

Combination between GLOSA and signal control is not a new idea. But most inductive loops are unable to provide frequent updates about individual approaching since they are fixed in limited locations. Upcoming V2I technology will allow vehicles to communicate more accurate movement data more frequently. It is also capable of providing information back to vehicles approaching the intersection. This means that V2I communication is able to influence driving behaviour and trajectories. Several authors have proposed mechanisms for determining Green Light Optimal Speed Advisory (GLOSA) for vehicles approaching an intersection. However, they do not use MPC to find optimal solutions. Connected Vehicle (CV) technology



allows Model Predictive Control (MPC) to be integrated with GLOSA, making the best use of this future technology to improve traffic conditions for all motorists.

This paper proposes the optimal control schedule that minimises delay. It suggests that this optimal control schedule can best utilise traffic data that varies second-by-second and it can be found via MPC with suitable state space reduction techniques. Since the control algorithm utilises an underlying microscopic model, entering vehicles' trajectories can be modified with Green Light Optimal Speed Advisory (GLOSA). This allows drivers to adjust their speed profiles in order to have an efficient approach trajectory.

Unlike traditional signal control that determines the parameters such as cycle time, phase splits and offsets, this paper uses an alternative formulation that considers traffic control to be a scheduling problem. The scheduling algorithm takes as input a set of vehicles, and the intended routes for each vehicle. In addition, each vehicle has a location, speed and acceleration at the current time. All of this information could be sent to the intersection controller using V2I communication. The control algorithm works by enumerating possible future schedules and calculating their associated total cumulative delays within the planning horizon, attempting to find the optimal schedule with minimum delay. Since there are a finite number of arriving vehicles there are a finite number of ways to sequence vehicles through the intersection. However, it is impractical to enumerate them all in real-time because the number of possible sequences is (at least) exponential compared to the number of vehicles on all routes [1]. To overcome this computational complexity problem, Xie et al. [2] developed a polynomial time algorithm by framing single intersection traffic control as a job shop scheduling problem, and using elimination criteria to reduce the size of the search state space. These elimination criteria are also examined. In this paper, an algorithm is developed that works with a more realistic, car-following traffic model, and the effect of combining GLOSA with MPC signal control is examined. The car-following model used in this paper are defined with several rules. Rule (1) and (2) ensure that, minimum headway between vehicles are kept and no collisions will occur. Braking at the maximum speed will occur if these minimum distances are not kept. Rule (3) is employed only if the antecedents of rules (1) and (2) are false. Rules (4) and (5) adjust the vehicle's updated acceleration, a' , for the following simulated time step, to make it within the range of the vehicle's maximum acceleration rate and maximum deceleration rate. Once a' is known, the updated values v' and x' can be calculated.

Two techniques, platoon clustering and elimination criteria are applied to achieve state space reduction when the proposed algorithm grows exponentially. Generally, it is not beneficial to split platoons of vehicles apart due to intersection signalisation. Therefore, platoons are kept together in order to reduce the computational complexity of the algorithm. In order to reduce the computational complexity of the scheduling algorithm, similar elimination criteria to those used by Xie et al. [2] were introduced to prune the state space.

Four scenarios were tested: MPC with GLOSA, MPC without GLOSA, fixed time signal control with GLOSA, and fixed time signal control without GLOSA. Overall, the performance indicators reveal that model predictive signal control significantly outperforms fixed time signal control. Combining MPC with GLOSA improves performance further. These results are limited to several assumption/simplification of network, such as one lane single intersection, traffic only from two perpendicular direction, north and east bound.

Results relevant to MAVEN

A significant contribution of MAVEN is that it can improve urban intersections by combining CAVs with GLOSA to modify vehicle approaching pattern. In MAVEN, GLOSA will also be coupled with the existing platooning approach simpla (see Section 4). This results in platoons and not individual vehicles requesting signal switches. As explained in this paper, platoon clustering can be used for elimination criteria to achieve state space reduction. This can be an important factor to consider in MAVEN. In addition, the Literature review of signal control



approaches in this paper is quite extensive and related to MAVEN, especially considering GLOSA, which can be used for work package 4 of MAVEN.

Another interesting point to consider is the car-following model in this paper. It can be compared to the model being used in simpla, to examine how CAVs is forming platoon and how to keep the platoon clustering.

Lastly, the generic scheduling algorithm that utilizes the structure of the state space is shown with pseudocode in this paper. It shows that the main problem is the large number of potential sequence with this algorithm. Elimination Criteria is introduced to prune the state space. Further possible research can be carried out to study/improve ImFlow selection process of control strategies.

Citation (multiple if relevant, e.g. multiple deliverables)

Simon Stebbins, Jiwon Kim, Mark Hickman and Hai L. Vu, Combining Model Predictive Intersection Control With Green Light Optimal Speed Advisory In a Connected Vehicle Environment, Transportation Research Board Annual Meeting, 2017.

Referenced papers:

[1] Papageorgiou, M., C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. Proceedings of the IEEE, Vol. 91, No. 12, 2003, pp. 2043–2067.

[2] Xie, X.-F., S. F. Smith, L. Lu, and G. J. Barlow. Schedule-driven intersection control. Transportation Research Part C: Emerging Technologies, Vol. 24, 2012, pp. 168–189.

3. Cooperative Adaptive Cruise Control (CACC) at intersections

Title

<1> Eco-Cooperative Adaptive Cruise Control at Signalized Intersections Considering Queue Effects

<2> Eco-cooperative Adaptive Cruise Control at Multiple Signalized Intersections: Network-Wide Evaluation And Sensitivity Analysis

Summary of the work

In general, eco-driving research can be categorized as freeway-based and city-based strategies. Unlike freeways, traffic stream motion along arterial roads is typically interrupted by traffic control devices. The study focus of MAVEN is at urban intersections with traffic controls. To include the works of eco-driving with CACC at intersections, two related papers are grouped and introduced.

<1> Unlike existing studies related to eco-driving on arterial corridors, which attempt to minimize idling time and smooth acceleration/deceleration manoeuvres without considering the impact of surrounding traffic, this paper considers not only the SPaT information, but also the vehicle queues at signalized intersection approaches.

Queue estimation and discharges are studied in the following literatures: [1], [2] [3], [4]. However, these studies didn't integrate the queue estimation into the algorithm to apply them in real-time applications. Moreover, most researches focus on optimizing speed profiles of equipped vehicles (EVs) upstream of the intersection and ignore the acceleration behaviour downstream of the intersection after the traffic signal turns green, which results in more fuel usage of vehicles proceeding through intersections. This paper conducts a comprehensive analysis of Eco-CACC systems on arterial roads, predicting vehicle queues upstream of a signalized intersection to develop fuel-optimum vehicle trajectories.

Firstly, this paper investigates the impact of the SPaT and the vehicle queue information on fuel consumption levels at signalized intersections. Secondly, the impact of vehicle queues on the Eco-CACC algorithm proposed in [5] is analytically investigated. [5] has demonstrated



that the most critical strategy to reduce fuel consumption levels is to prevent vehicles from coming to a complete stop at the stop bar. This paper develops an Eco-CACC algorithm considering queue effects (Eco-CACC-Q) to minimize vehicle fuel consumption levels while proceeding through an intersection with the consideration of realistic deceleration and acceleration levels. Lastly, this paper also evaluates the benefits of the algorithm within a microscopic simulation environment (both single-lane and two-lane intersection approach are studied separately), and evaluate the impact of market penetration rates of probe vehicles and intersection configurations on the system performance.

The results illustrates the impact of queue length on the total fuel consumed. Both the Eco-CACC algorithms (Eco-CACC-O doesn't consider queue and Eco-CACC-Q considers queue) significantly reduce the fuel consumption of the Eco-CACC vehicles. The Eco-CACC-O algorithm reduces the fuel consumption by as high as 25% while the Eco-CACC-Q algorithm reduces the fuel consumption by 32%. Comparing the two algorithms, the Eco-CACC-Q algorithm produces fuel consumption levels that are 10% lower. Moreover, with longer queues, the cruise speed is smaller and the fuel consumption is larger. These results show that the Eco-CACC-Q algorithm can further improve the fuel efficiency of Eco-CACC vehicles.

With simulation results on single-lane and two-lane intersection approaches, the following findings are summarized: For single-lane intersection approach, the Eco-CACC-Q algorithm is the most efficient control strategy with reductions in fuel consumption levels as high as 11.4%; and compared with Eco-CACC-O, it reduces fuel consumption levels by approximately 4.5%; For two-lane intersection approach, the Eco-CACC-Q algorithm provides the most efficient control with reductions in fuel consumption levels for Eco-CACC vehicles as high as 19.2%; and compared with Eco-CACC-O, it reduces fuel consumption levels by approximately 5.6%.

As expected, both algorithms have a negative impact on the overall fuel consumption rate under lower Market Penetration Rate (MPR) of Eco-CACC vehicles. This is caused by the intense lane changes around the controlled vehicles. Once the MPR is greater than 30%, the cutting in and lane change behaviours of non-Eco-CACC vehicles are reduced. Hence, the Eco-CACC algorithms generate fuel consumption savings at higher MPRs. These savings increase as the MPR increases. If all vehicles are Eco-CACC vehicles, the fuel consumption rate is reduced by approximately 17.0% for Eco-CACC-O, and 18.3% for Eco-CACC-Q demonstrating the benefits of the Eco-CACC-Q system.

This paper only investigated the impact of MPRs on the algorithm performance. Other factors, such as the length of the controlled segments, the traffic demand levels, the SPaT plan and the robustness of the algorithm to errors in wireless communication, should also be considered in future research (see paper <2>). Moreover, further improvements to the proposed Eco-CACC-Q algorithm should considering multiple signalized intersections in the optimization logic. Furthermore, the proposed algorithm only applied vehicle-to-signal communications to gather road traffic information, which was not sufficient to estimate the queue length accurately due to lane-changing and passing behaviour. In the future, we will introduce vehicle-to-vehicles communications to queue length estimation algorithm [6]. Finally, one drawback of the algorithm is that it fails once the road is over-saturated. Hence, we propose that we combine a speed harmonization algorithm [6] on arterial roads and signal optimization with the Eco-CACC-Q algorithm to solve this problem and to further improve the system.

<2> This paper is an extension of previous research <1>, Eco-Cooperative Adaptive Cruise Control for Multiple Signals (Eco-CACC-MS) is introduced and tested on multiple intersection networks using the INTEGRATION microscopic traffic assignment and simulation software, aiming at reducing vehicle fuel/energy consumption. It is also being comparing to Eco-CACC-O algorithm proposed in <1>, which is simply referred as Eco-CACC in this paper. The Eco-CACC-MS uses SPaT data received from each signal controller via V2S communication, to predict the queue length and to compute the fuel-optimum vehicle



trajectory for an equipped vehicle. The algorithm estimates three optimal acceleration/deceleration rates for the equipped vehicle to minimize the total fuel consumption to pass the two intersections. Then the algorithm provides an advisory speed limit that allows the vehicle to pass multiple consecutive intersections without stopping.

For roads with more than two intersections, the system will always estimate the optimal trajectories for the equipped vehicles with the consideration of their two immediate intersections. Once an equipped vehicle passes one intersection, its optimal trajectory will be recalculated with the SPaT information of the two downstream intersections.

The sensitivity analysis of this paper considered factors such as market penetration rate (MPR) of equipped vehicles, number of lanes of the controlled segment, traffic demand rates, offset between the traffic signals, and distance between intersections.

For travel demand level test, different traffic flow demands from 300 to 700 vphpl (vehicle per hour per lane) were tested. The first test is a single-lane network and the second test is a two-lane network, both tests with 45s offset for the second signal with respect to the first signal. The results show some savings for all demand levels, but the overall demand of 300 vphpl has the best savings of 7.7%. These trends are the result of the increase in the number of the probe vehicles in the network. As the road gets close to the oversaturation level, lower-than-optimal results are seen and the network-wide savings decreases. Savings for the two-lane road are similar to the single-lane result, Eco-CACC-MS for a demand of 300 vphpl saved 7.9% more fuel than Eco-CACC. Similar to the single-lane road, the network-wide savings in fuel consumption decreased as the demand neared the oversaturation point.

For distance between intersections, which is an important variable that impacts fuel consumption savings, the results show that the shorter the distance between the intersections, more savings are achieved by the algorithm, which is 9.9% for 400 m between intersections. The range between the intersections cannot be very small based on the effective distance of the Dedicated Short-Range Communications (DSRC) technology that is implemented in the V2I communications between probe vehicles and signals.

For different offset, the best savings were captured at a 600-m distance between intersections by using a 0-s offset. The fuel savings for Eco-CACC-MS is 0.5% more than Eco-CACC.

For the number of lanes, the fuel savings of the single-lane road are greater at higher MPRs. Under 100% MPR, the reductions in fuel consumption are 7.3% higher for Eco-CACC-MS compared to the Eco-CACC. On the multi-lane road, due to lane-changing behaviour, there were small savings in fuel consumption for MPRs lower than 20%. For the two-lane network, savings in fuel consumption are 7.4% higher for the Eco-CACC-MS compared to the Eco-CACC system under 100% MPR. The results show that considering the queue enhances the algorithm performance.

However, these findings are not general and need further investigation for different demand levels, signal offsets, and phase splits.

The proposed Eco-CACC-MS system has limited application for oversaturated conditions due to rolling queues. Possible solutions include using vehicle-to-vehicle (V2V) communication to estimate the queue or to introduce speed harmonization [6] to restrict the traffic entering the intersection to maintain an under-saturated condition at all the times.

Results relevant to MAVEN

The queue estimation in these two papers are of interest to MAVEN because accuracy of queue length estimation is a major factor to GLOSA functionality. Moreover, MAVEN introduces V2V communications to queue modelling, which can help on estimating the queue length accurately due to lane-changing and passing behaviour. Relevant literature [6] can be referred regarding V2V communication.

In MAVEN, a constant/ fuel consumption optimum deceleration rate can be applied to CAVs which have received GLOSA. This information should be communicated back from vehicle to



infrastructure, to assist cooperative intersection SPaT calculation and platoon green wave. Paper <2> is relevant considering the algorithm Eco-CACC-MS is based and tested on multiple intersection networks.

The tests in this paper studied the performance of the Eco-CACC (O and Q) algorithms on both single-lane intersection approaches and multi-lane roads. But it only investigated the impact of MPRs on the algorithm performance. MAVEN can include the SPaT plan, explore on the traffic demand levels variation (such as peak and off-peak hour), and also examine the robustness of the system architecture. Most relevant studies, like <1> and <2>, would fail once the road is over-saturated. MAVEN can test the oversaturated scenario and can refer to a speed harmonization algorithm in [6] for further research.

Finally, the offsets and distances between multiple intersections on arterial road are to be considered in D4.3 of MAVEN.

Citation (multiple if relevant, e.g. multiple deliverables)

Source Papers:

<1> Hao Yang, Hesham Rakha and Mani Venkat Ala, Eco-Cooperative Adaptive Cruise Control at Signalized Intersections Considering Queue Effects, IEEE, 2016, pp(99):1-11.

<2> Hao Yang and Hesham Rakha, Eco-cooperative Adaptive Cruise Control at Multiple Signalized Intersections: Network-Wide Evaluation And Sensitivity Analysis, 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017.

Referenced papers:

[1] G. Qian and E. Chung, "Evaluating effects of eco-driving at traffic intersections based on traffic micro-simulation," in 34th Australasian Transport Research Forum (ATRF), Adelaide, South Australia, Australia, vol. 34, no. 0092, 2011.

[2] G. Qian, "Effectiveness of eco-driving during queue discharge at urban signalised intersections," Ph.D. dissertation, 2013.

[3] Z. Chen, "An optimization model for eco-driving at signalized intersection," Ph.D. dissertation, Texas A&M University, 2013.

[4] Q. Jin, G. Wu, K. Boriboonsomsin, and M. J. Barth, "Power-based optimal longitudinal control for a connected eco-driving system," IEEE Transactions on Intelligent Transportation Systems, pp. 1–11, 2016.

[5] R. K. Kamalanathsharma, H. A. Rakha, and H. Yang, "Network-wide impacts of vehicle eco-speed control in the vicinity of traffic signalized intersections," in Transportation Research Board 94th Annual Meeting, no. 15-4290, 2015.

[6] H. Yang and H. Rakha, "Developemnt of a speed harmonization algorithm: Methodology and preliminary testing," 2015, submitted to Transportation Research Part B.

4. Decision making tool for applying Adaptive Traffic Control

Title

Big Data Analysis Based Decision Making Tool for Applying Adaptive Traffic Control Systems

Summary of the work

This paper develops methods to guide traffic engineers and decision makers to decide whether adaptive control is best suited for a given traffic corridor and/or intersections than existing control systems especially actuated traffic control systems. The methods are based on big data analysis using large amount of data from various sources such as loop detectors, travel time systems, 511 systems, weather, and special events. The ultimate goal is to develop an easy-to-use and easy-to-understand decision making procedure to help make decisions about where adaptive control systems should be deployed.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Existing methods for such decision making are mainly based on before and after comparisons, using statistical or simple data analysis methods, e.g. using data collected for a few days by probe vehicles. With technologies advance, more and more traffic data can be collected from a variety of sources, including loop detectors, video cameras, mobile sensors, and connected and automated vehicles. It is thus interesting to investigate whether one can conduct decision making regarding adaptive control by leverage the large amount of traffic and related datasets. Big data analysis methods can be suitable for this purpose.

In this study, big data based quantitative analyses, such as volume, occupancy, volume capacity ratio, event, and weather on the study corridor, are conducted to evaluate whether adaptive signal is best suitable for a corridor. The data analysis methods, Support Vector Machine (SVM) to develop the decision-making tool. Linear SVM for LOS classification method is applied to classify data point into two classes. Using SVM, “good” signal performance (i.e., LOS A-C) can be distinguished from “bad” signal performance (LOS D-F). Results from the SVM models show that the misclassification errors decrease as the number of features that the model considers increases. Based on the SVM methods, a decision-making procedure is then developed to help deploying adaptive traffic signal control systems. To summarize, the steps of the SVM-based decision making procedure for adaptive control deployment are:

1. Collect data (volume, $v_{critical}$, occupancy, event, weather, etc.), ideally for a few months or longer.
2. Check the data points LOS-based SVM to map the data point into graphs, actuated (section 1) or adaptive (section 2) or the other sections. Each data point represents a “vote” that section.
3. Count the votes. If section 2 gets more votes, adaptive control should be deployed. Otherwise, adaptive control is not recommended.

One limitation of the current study is that the data size is rather small and is limited to only one corridor. Although the data is inherently heterogeneous, dynamic, and contains errors, technically it is not true “big” data yet. More critically, due to this limited data size, the results from the above procedure may not be very accurate or reliable. The immediate next step is to accumulate more data from more corridors in terms of before and after the deployment of adaptive control systems. The results should be improved dramatically if sufficiently large datasets are collected and used.

This big data analysis methods are suitable for decision making. The key philosophy in big data analysis methods is for data “to speak.” That is, one only cares about the correlations among data elements (i.e., the “what”), but not the underlying reasons behind the corrections (i.e., the “why”). This philosophy may not fit rigorous scientific investigation but, in any case, such methods can help better utilize the increasingly large amount of traffic data collected and being archived by transportation management agencies and the private sector, to make more informed decisions about traffic operations and other related decisions.

Results relevant to MAVEN

MAVEN is to manage automated vehicles that are connected with an intelligent environment in an urban environment. This goal of MAVEN contributes to the EU objective of reconciling growing mobility needs with more efficient transport operations, lower environmental impacts and increased road safety. Many of the test sites in MAVEN are real world traffic networks that contain pre-existed traffic control measures. Research of MAVEN so far already shows significant improvement on the existing networks.

This paper provides methods to determine whether the existing traffic control is efficient. It can also determine the improvement of changing the existing control to adaptive control. The data analysis method is applicable to case studies in MAVEN because the data collected in MAVEN are inherently heterogeneous, dynamic, and contains errors. In section 0 of this deliverable, some randomness of simulations are observed with fix seed of adaptive control



algorithm ImFlow. These randomness may be better analysed using the data analysis methods in this paper.

Future work of MAVEN includes data collection through camera detection. The methodology of this paper can also help the data analysis in that case.

Citation

Wan Li, Xuegang (Jeff) Ban, Big Data Analysis Based Decision Making Tool for Applying Adaptive Traffic Control Systems, Transportation Research Board Annual Meeting, 2017

5. Intersection communication for CAVs

Title

STIP: Spatio-Temporal Intersection Protocols for Autonomous Vehicles

Summary of the work

This paper focused on developing a new intersection protocols, Spatio-Temporal Intersection Protocols (STIP), with a realistic GPS model. STIP protocols are V2V protocols that aim to increase the throughput at intersections while avoiding collisions. STIP protocols enable cooperative driving among approaching vehicles to ensure their safe passage through the intersection. In this paper, STIP is implemented in a hybrid emulator-simulator for vehicular networks, called AutoSim.

A family of vehicular network protocols to manage the safe passage of traffic across intersections relies on vehicle-to-vehicle (V2V) communications and localization to control and navigate vehicles within the intersection area. Generally speaking, V2V communications using Dedicated Short Range Communications (DSRC) and Wireless Access in a Vehicular Environment (WAVE) to broadcast intersection safety messages to other vehicles in their communication range. Hence, CAVs approaching an intersection use DSRC and WAVE to periodically broadcast information such as position, heading and intersection crossing intentions to other vehicles. These vehicles then decide among themselves regarding such questions as who crosses first, who goes next and who waits.

STIP can be described as follows: all vehicles are assumed to follow the First-Come, First-Served (FCFS) policy, in which the vehicle with the lower arrival time to the intersection has the higher priority. In the scenarios that two or more vehicles arrive almost at the same time, they break the ties in favour of vehicles approaching on main roads. If the tie still holds, it is broken by Vehicle Identification Number (VIN), which has uniquely assigned to each vehicle. Three classes of STIP are introduced in this paper. 1) Minimal Concurrency Protocols (MCP) includes Throughput Enhancement Protocol (TEP) and Concurrent Crossing-Intersection Protocol (CC-IP) [1, 2]. In this category, the conflicting vehicle with higher priority can ignore the intersection safety messages from other lower priority vehicles and cross the intersection without slowing down or stopping. However, any lower-priority vehicle is super-cautious and, when it loses a competition, it comes to a complete stop before entering the intersection boundaries, and waits till it receives an EXIT message, from the higher priority vehicle. This message informs the lower-priority vehicle that the higher-priority vehicle has crossed the intersection and now the intersection area is safe for its passage. This protocol is applied across all priority levels. 2) High Concurrency Protocols (HCP) includes the Maximum Progression Intersection Protocol (MP-IP) and the Advanced Maximum Progression Intersection Protocol (AMP-IP) [3]. The main goal is to increase the parallelism inside the intersection area by allowing more vehicles to cross the intersection at the same time. This goal is achieved by allowing even conflicting vehicles to make maximal progress inside the intersection area, without sacrificing the primary goal of safety. This category allows even potentially conflicting vehicles to progress inside the intersection area, and the lower-priority



vehicle gets to a complete stop before entering the conflicting cell, and waits till the higher-priority vehicle has crossed and cleared that cell. 3) High Concurrency Protocols with lowdown (HCPS). HCPS includes Advanced Cross Intersection Protocol (ACIP) and advanced Progression Intersection Protocol (APIP). In MCP and HCP, STIP protocols, potentially conflicting vehicles with lower priority must come to a complete stop outside or inside the intersection area to allow the safe passage of higher-priority conflicting vehicle. HCPS protocols allow lower-priority conflicting vehicles to slow down while approaching an intersection and prior to the conflicting cell, to provide the higher priority-vehicle with necessary time gap to cross. This will minimize a vehicle's need to get to a complete stop, and also the total number of stops and startups will be decreased significantly.

The results indicate that STIP protocols are significantly outperforming the traffic light models. AP-IP has very negligible delay when dealing with low and medium-volume traffics. AP-IP outperforms the traffic light models with 30 seconds and 10 seconds of green light duration, respectively by 91.04% and 80.21%.

Clearly, localization and positioning accuracy is crucial for safety applications such as intersection collision avoidance. Different methods can be deployed to improve the position accuracy such as using high-accuracy Differential GPS (DGPS), Wide Area Augmentation System (WAAS), gyroscopes and local sensing. However, all GPS receivers have finite accuracy, with commonly-used inexpensive GPS receivers having errors of up to a few meters.

When dealing with high levels of positioning inaccuracy, each vehicle will use an updated safe distance parameter based on its GPS positioning error parameter. This increased buffer distance among following vehicles prevents vehicles from getting very close to each other and gives them the capability to slow down without causing an accident when the leader vehicle brakes suddenly.

This paper designed a new generation of V2V-based intersection protocols: STIP, which significantly increase the throughput of the intersections and avoid collisions. The effects of GPS position inaccuracies are also studied by implementing realistic GPS models. Although STIP protocols are designed for autonomous vehicles that use V2V communication for cooperative driving in future intelligent transportation systems, they can be adapted to a driver-alert system for manual vehicles at traffic intersections. Local sensing technologies such as cameras and lasers can be combined with V2V and V2I communications to avoid any potential collisions and enhance localization accuracy.

Results relevant to MAVEN

A major focus of MAVEN is on opportunities that arises from traffic control and management that are based on automated vehicles, V2V and V2I communication. WP4 of MAVEN investigates these opportunities and challenges of connected environment and its effect on signal timing. To do this, the communication protocols of MAVEN play a leader role throughout the project. This paper provides such a novel protocols, STIP. These protocols are also implemented on the arbitrary network, which showed that these protocols are significantly outperforming the traffic light models. To the least, these protocols can trigger some ideas for MAVEN studies.

Another interesting aspect of this paper is the GPS localization and positioning accuracy. This is of great importance to section 3 of this deliverable and the future work. This paper also proposed methods of GPS correction which is practical if applicable in MAVEN.

Citation (multiple if relevant, e.g. multiple deliverables)

Reza Azimi, Gaurav Bhatia, Ragunathan (Raj) Rajkumar and Priyantha Mudalige, STIP: Spatio-Temporal Intersection Protocols for Autonomous Vehicles, ICCPS'14, Berlin, Germany, April 14-17, 2014



Referenced papers:

- [1] R.Azimi, G. Bhatia, R. Rajkumar, P. Mudalige. Vehicular networks for collision avoidance at intersections, society for automotive engineers (SAE) world congress. April 2011.
- [2] R.Azimi, G. Bhatia, R. Rajkumar, P. Mudalige. Intersection management using vehicular networks, society for automotive engineers (SAE) world congress. April 2012.
- [3] R.Azimi, G. Bhatia, R. Rajkumar, P. Mudalige. Reliable intersection protocols using vehicular networks, ACM/IEEE 4thinternational conference on cyber-physical systems. April 2013.

6. Environmental impacts of CAVs

Title
Anticipating The Emissions Impacts of Smoother Driving by Connected and Autonomous Vehicles, Using The MOVES Model
Summary of the work
This paper examines the emission impacts of CAVs, presuming that CAVs are programmed to drive more smoothly than humans. It uses the US Environmental Protection Agency's (EPA's) Motor Vehicle Emission Simulator (MOVES) to estimate CAVs' emissions based on driving schedules or profiles. In this paper, only urban road emissions were simulated. Several emission pollutants, such as Volatile Organic Compounds (VOC), particulate matters 2.5 (PM2.5), carbon monoxide (CO), nitrogen oxides (NOx), sulphur dioxide (SO2) and carbon dioxide (CO2) are considered to anticipate some of the emission impacts of CAVs. CAV driving profiles are envisioned to be smoother than those of human-controlled vehicles (HVs), because CAVs are expected to be faster and more precise than human drivers, in terms of reaction times and manoeuvring. Human drivers tend to create significant, frequent speed fluctuations (i.e., hard brakes and rapid accelerations) and have relatively long reaction times (e.g., 1.5 seconds). CAV technologies may rarely suffer from such fluctuations, allowing for smoother driving profiles, referred to here as Eco-Autonomous Driving (EAD) cycles. Hard braking and rapid acceleration events are associated with increased emissions, so, by smoothing HVs' existing driving cycles, this work anticipates the emission benefits of CAVs. Differences in HV vs. CAV emissions estimation suggest valuable air quality improvement from CAVs, assuming CAVs are driven no more than HVs would be. The results from EPA cycles suggest that, in general, if HVs are replaced by AVs, greater emission benefits (up to 14% emission reductions) are anticipated in driving conditions where there are many hard acceleration and braking events, and for drivers with aggressive driving styles. The results from Austin cycles indicate the mean emission reductions are 10.89% for VOC, 19.09% for PM2.5, 13.23% for CO, 15.51% for NOx, and 6.55% for SO2 and CO2. Regression models revealed that passenger cars were found to be associated with lower emission reductions for VOC, PM2.5, CO, and NOx than passenger trucks. Diesel vehicles are linked with smaller emission reductions for these six types of emissions. The road links with higher average speeds have greater emission reductions for all emission species. At this point, the discussion of emission impacts of AVs is limited to the differences between the anticipated EAD profiles of CAVs and existing HV driving cycles. CAV profiles are envisioned to be smoother than HV cycles as compared to HV cycles. Other CAV-based technologies (like platooning of vehicles and CACC) may also save fuel and reduce emissions further.
Results relevant to MAVEN
Besides accommodating growing traffic and increasing road safety, a main objective of EU



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

transport project is to obtain environmental sustainability and to achieve lower environmental impacts. MAVEN fits this goal by preparing and adjusting infrastructure systems. Combining with V2I and V2V communications, MAVEN is able to provide GLOSA to vehicles in order to reduce stop-and-go movements as well as sudden and fluctuated acceleration/deceleration. This paper shows the emission impacts of HVs and CAVs. In this paper, the emissions evaluation model MOVES are enhanced with the driving profiles of CAVs. The comparison of HVs and CAVs behaviours and emission impact are relevant to MAVEN infrastructure systems and they are important to be taken into account. Platooning of vehicles in D2.2 Annex 2 and CACC in urban environment mentioned in 3. Cooperative Adaptive Cruise Control (CACC) at intersections, also demonstrate achievability on these environmental aspects.

Citation

Jun Liu, Kara M.Kockelman and Aqshems Nichols, Anticipating The Emissions Impacts of Smoother Driving by Connected and Autonomous Vehicles, Using The MOVES Model, Transportation Research Board Annual Meeting, 2017.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.