



Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin

Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

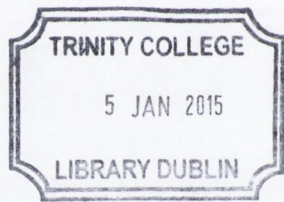
I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

**A Slot-based Approach to Optimise Freeway Traffic Using
Intelligent Vehicles**

Dan Marinescu

A Dissertation submitted to the University of Dublin, Trinity College
in fulfillment of the requirements for the degree of
Doctor of Philosophy (Computer Science)

2014



Thesis 10442

Declaration

I, the undersigned, declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.



Dan Marinescu

Dated: September 21, 2014

Acknowledgements

I owe a very grateful thank you to many people who have helped me in countless ways during my doctorate experience.

I would like to thank my supervisors Vinny Cahill and Mélanie Bouroche for their guidance and assistance over the years. Thank you to all the members of the NEMBES project and to the Irish Higher Education Authority who funded it. Many thanks also to all my colleagues in the Distributed System Group for making it such a pleasant environment to work in. In particular, I would like to thank my colleague Jan Čurn, with whom I've shared a common research interest, an apartment, and a great friendship: "Have a good time to DSG"!

Five years ago, I arrived in Ireland without knowing a single person. The friends I have made in this time have offered me invaluable support. Special thanks to: Mithileash Mohan, for being my first friend in Dublin; Tim Walsh, for adopting me in the "gang"; Raymond Cunningham, for introducing me to lunch-break football; Neil O'Connor, for letting me believe I can get better at golf when that was clearly not the case; Jenny Munnely, for all those great weekends having after-golf lunch in her parent's place; Shiu Lun Tsang, for always being available to go on holidays; Ivana Dusparic, for all those enjoyable conversations we had while walking home from Trinity; and Andronikos Nedos, for inviting me to watch the Champions League final in his place when he didn't really know me, nor enjoy watching football.

Many thanks also to my soon-to-be wife Diana Ferri, who put up with me being away for almost five years. I wouldn't have made it without your understanding and support. Finally, I would like to thank my family, for their guidance throughout my entire life.

Andrei, thanks for reviewing this thesis bro. Mom, Dad, thanks for keeping faith in me throughout puberty, I know it must have been really hard.

Dan Marinescu

University of Dublin, Trinity College

April 30, 2013

Abstract

Over the last decades, the increase in demand for road transportation has traditionally been matched by an expansion of the road network to prevent or reduce traffic congestion. This approach is unsustainable in the long term. A sustainable solution requires more efficient use of the existing road network by deploying so-called Intelligent Transportation Systems (ITS). Present day ITS implementations are generally responsible for controlling road access, incident detection and mitigation, predicting traffic conditions and providing route guidance, with the vast majority of these systems being deployed as part of the road infrastructure. However, the benefits of roadside-only ITS deployments are limited as they do not address the nature of human driving, characterized by relatively slow reaction times and competition rather than cooperation, which leads to an underutilisation of the road surface. In parallel, car manufacturers have addressed road safety issues by gradually shifting the control of the vehicle from the driver to the vehicle itself, leading to the emergence of so-called intelligent vehicles (IVs). Besides increasing safety, IVs have the potential of enabling a more efficient form of driving: vehicles equipped with sensors can react faster than human drivers, which allows a reduction of the inter-vehicle spacing, while vehicles equipped with communication capabilities can leverage inter-vehicle communication to cooperate. This thesis describes an approach that combines the sensing and communication capabilities of future IVs with roadside ITS deployments to optimise traffic on freeways. Such approaches are commonly known as Intelligent Vehicle/Highway Systems (IVHSs).

Previous research on using IVs to optimise freeway traffic employ either a centralized or decentralized approach. Centralized approaches exploit the fast reaction times of IVs by grouping vehicles into so-called “platoons” inside which vehicles maintain relatively small time-headways, thus increasing the theoretical capacity of the freeway. The platoon is employed as an interface between individual vehicles and a traffic management system (TMS) responsible for optimising traffic. Centralized approaches generally maintain a strong coupling between the higher-level TMS and individual vehicles that implement the policies of the TMS, such as following a pre-

assigned route and target speed. By contrast, decentralized approaches use local cooperation between vehicles to achieve network-wide traffic optimisation. The optimisation task is shifted from the TMS to the individual vehicles and emerges as a result of the cooperation strategy employed. Both approaches have significant drawbacks. Centralized approaches — due to their high degree of control upon vehicles — lack flexibility and scale poorly. Distributed approaches address these issues but do not leverage the optimisation benefits of a better informed central component that uses a global view of the road network and roadside traffic control to determine and implement optimisation policies.

This thesis proposes a hybrid approach that addresses the drawbacks of both centralized and decentralized approaches by introducing an abstraction layer - called a virtual slot - between the TMS and individual vehicles. A virtual slot moves in time along the freeway with a predefined behaviour (speed and trajectory profile). The combined behaviour of all virtual slots along the freeway is called the slot map. The optimisation task is the responsibility of the TMS and is implemented through the specification of the slot map. However, the TMS does not directly influence the driving task of individual vehicles. Instead, vehicles are individually responsible for coordinating with neighbouring vehicles in order to move into, drive within, and change slots, altogether referred to as slot-based driving. As such, the virtual slot abstraction achieves a loose coupling between the driving and optimisation tasks that translates into a simple set of coordination mechanisms for IVs. This leads to a flexible and scalable traffic management framework that also leverages the optimisation benefits of centralized management and integrates with roadside traffic control systems. The emergent cooperative behaviour of vehicles driving in a predefined slot map, combined with a better utilisation of the road surface due to the smaller reaction times of IVs, helps prevent traffic congestion. Another effect of the deterministic nature of a predefined slot map is that the travel times of IVs can be accurately estimated. This is very useful in the context of a global economy that is becoming increasingly reliant on just-in-time production and distribution systems that in turn depend on reliable journey-time estimation.

The main contribution of this thesis is the novel slot-based approach to optimising traffic using IVs. This approach is implemented in the form of a TMS and a set of vehicle coordination and control algorithms that facilitate slot-based driving. Furthermore, the slot-based approach is applied and evaluated in a set of scenarios in which congestion commonly occurs. The results of this evaluation indicate that the slot-based approach can be used to significantly increase the efficiency and reliability of traffic.

Publications related to this Ph.D.

O'Hara, N., Slot, M., **Marinescu, D.**, Čurn, J., Yang, D., Asplund, M., Bourroche, M., Clarke, S. and Cahill, V.: "MDDSVsim: an integrated traffic simulation platform for autonomous vehicle research", in *The First International Workshop on Vehicular Traffic Management for Smart Cities (VTM) 2012*, IEEE Computer Society.

Marinescu, D., Čurn, J., Bourroche, M. and Cahill, V.: "On-ramp traffic merging using cooperative intelligent vehicles: a slot-based approach", in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITSC 2012)*, pp. 900–906.

Čurn, J., **Marinescu, D.**, Lacey, G. and Cahill, V.: "Estimation with non-white Gaussian observation noise using a Generalised Ensemble Kalman filter", in *Proceedings of the IEEE International Symposium on Robotic and Sensors Environments (ROSE 2012)*, pp. 85–90.

Čurn, J., **Marinescu, D.** and Cahill, V.: "On the Monte Carlo representation of uncertain spatial constraints", in *Procedia Engineering: International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012)*, vol.41, pp. 37–46.

Marinescu, D., Čurn, J., Slot, M., Bourroche, M. and Cahill, V.: "An active approach to guaranteed arrival times based on traffic shaping", in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*, pp. 1711–1717.

Čurn, J., **Marinescu, D.** and Cahill, V.: "A flexible approach to management and processing of collaborative vehicular perception data", in *Proceedings of the Workshop on Emergent Cooperative Technologies in Intelligent Transportation Systems (ECTITS 2010) at the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*.

Marinescu, D., Bourroche, M. and Cahill, V.: "Towards an active approach to guaranteed arrival times based on traffic shaping", in *Advanced Microsystems for Automotive Applications 2010*, Springer Verlag, pp. 367–376.

Contents

Acknowledgements	v
Abstract	vi
List of Tables	xiii
List of Figures	xiv
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 The Case for Vehicle Automation	4
1.3 Approach	6
1.4 Scope	7
1.5 Thesis Roadmap	10
Chapter 2 State of the Art	12
2.1 State of Practice	12
2.1.1 Enabling Technologies	13
2.1.2 Demand Management	15
2.1.3 Supply Management	18
2.1.4 Discussion	22
2.2 Using Intelligent Vehicles to Optimise Traffic	23
2.2.1 Intelligent Vehicles	23

2.2.2	Intelligent Vehicle Highway Systems	33
2.3	Conclusion	46
Chapter 3	Design	48
3.1	Design Decisions	48
3.1.1	Limitations of previous work	49
3.1.2	Requirements	49
3.1.3	A Slot-based IVHS	52
3.2	System Model	55
3.2.1	Freeway Model	55
3.2.2	Vehicle Model	57
3.2.3	Slot Model	59
3.3	The TMS	61
3.3.1	Architecture	61
3.3.2	Slot Provider	64
3.4	Slot-based Driving	67
3.4.1	Moving into a Slot	69
3.4.2	Driving within a Slot	73
3.4.3	Changing Slots	74
3.4.4	Swapping Slots	76
3.5	Case studies	77
3.5.1	Addressing Reductions of Freeway Capacity	79
3.5.2	Demand Increase	85
3.6	Discussion	91
3.6.1	Feasibility of the Slot-based Approach	91
3.6.2	Matching the Requirements	94
3.6.3	Limitations	96
3.7	Conclusion	97

Chapter 4 Evaluation	99
4.1 Evaluation Framework	100
4.1.1 VISSIM integration	101
4.1.2 Slot Provider	106
4.1.3 Vehicle	111
4.1.4 Summary	118
4.2 Case Studies	118
4.2.1 Testbed	119
4.2.2 Moving Vehicles into Slots	121
4.2.3 Lane Merging	124
4.2.4 On-ramp Merging	129
4.2.5 Reliability	136
4.3 Discussion	139
 Chapter 5 Conclusion	 143
5.1 Achievements	143
5.2 Future Work	145
5.2.1 System Model	146
5.2.2 Heuristics	147
5.2.3 Slot-based Driving in Real-world Conditions	147
5.2.4 Fault Tolerance	148
5.3 Outlook	148
 Appendix A Implementation of DriverModel	 150

List of Tables

2.1 Classification of reviewed literature 44

List of Figures

1.1	Fundamental diagram of traffic flow	3
1.2	Scale model of an automated highway presented at GM's Futurama exhibition at the World Fair in 1939	5
1.3	Changing the nature of driving reflected in the fundamental diagram . . .	8
2.1	The IVHS architecture introduced by PATH (after Horowitz and Varaiya (2000))	34
2.2	The Dolphin architecture	36
2.3	The Auto21 architecture	37
2.4	IV-based framework, after Baskar et al. (2012)	38
2.5	Average travel times vs. Reliability (U.S. Federal Highway Authority, 2006)	46
3.1	A hybrid IVHS architecture	52
3.2	Virtual slots as the interface between RSUs and vehicles	54
3.3	Basic segment	56
3.4	Ramp merge segment	56
3.5	Ramp diverge segment	57
3.6	Facility Example	58
3.7	The architecture of the TMS	62
3.8	A simple slot formation	66
3.9	Slot-based driving architecture	68
3.10	Is the slot ahead of B free or about to be occupied?	72

3.11	The parallel formation	76
3.12	The chessboard formation	77
3.13	Swapping slots within the chessboard formation	78
3.14	Classification of congestion by its root cause	79
3.15	Visualisation of the lane merging algorithm	81
3.16	On-ramp merging	87
3.17	Moving to free slots to the left-front of the current slot	88
3.18	Swapping slots with a vehicle targeting another off-ramp	89
3.19	Vehicles cannot exit the freeway	90
3.20	Improved exiting probability through reduced capacity	91
4.1	Evaluation framework built around VISSIM	101
4.2	Vehicle lifecycle within VISSIM	104
4.3	Slot Provider Class Diagram	107
4.4	Interaction between the individual components of a vehicle	111
4.5	Vehicle class diagram	113
4.6	Manual vs. VTS throughput on a three-lane freeway section	123
4.7	Delay for 3-to-2 lane merging with no HGVs	125
4.8	Delay for 3-to-2 lane merging with 10% HGVs	127
4.9	Delay for 3-to-2 lane merging with 20% HGV	127
4.10	Impact of HGVs on the lane merging algorithm	128
4.11	On-ramp merging	130
4.12	Maximum on-ramp throughput	132
4.13	On-ramp delay under medium traffic conditions on the mainroad	133
4.14	On-ramp delay under heavy traffic conditions on the main road	134
4.15	On-ramp delay under medium traffic conditions on the main road with an exit ramp	135
4.16	On-ramp delay under heavy traffic conditions on the main road with an exit ramp	136
4.17	Reliability of the slot-based approach	138

Chapter 1

Introduction

This thesis presents an approach to optimising road traffic using vehicles that are capable of sensing their environment and of (semi-)automated vehicle control, commonly referred to as intelligent vehicles (IVs). The novelty of this work consists in its employment of a hybrid architecture based on an abstraction layer between roadside infrastructure and vehicles called *virtual slot*. The virtual slot concept combines the superior optimisation potential of a centralized, roadside-based infrastructure with the scalability and flexibility of distributed vehicle coordination.

1.1 Motivation

Transport represents the life blood of industrialised economies. According to the European Commission (2003), in the the European Union (EU), transport provides about 10% of wealth measured in terms of gross domestic product (GDP). Among the various modes of transportation, transport by road represents the biggest share. As of 2003, around 80% of all passenger journeys within the EU were made by car and as much as 44% of all goods transported in the EU went by road. The increasing demand for road transport, especially to the detriment of rail transportation, has led to an increase in congestion: every day 7,500 kilometres or 10% of European highways experience traffic jams.

The increase in congestion creates problems of an environmental, financial and social nature. Road traffic alone is responsible for 12% of the EU's emissions of carbon dioxide. The transport sector is responsible for 30% of the EU's energy consumption and 71% of all oil consumption, with road transport accounting for 60% (European Commission, 2007). From an economic point of view, congestion adds 6% to the EU's fuel bill (European Commission, 2003) and causes total costs of about 0.5%–1% of EU's GDP (European Commission, 2006). Furthermore, congestion causes driver frustration, which can lead to accidents and so has an indirect impact on the safety of humans. Around 40,000 people die yearly on EU roads. Besides the unquantifiable human costs, road accidents cost the EU approximately 2% of its GDP (European Commission, 2003).

The traditional approach used in the past to address the rise in demand for road transportation was to increase road capacity by building new roads or extending existing ones. This, however, is becoming increasingly untenable due to socioeconomic considerations: building new roads impacts the environment, is expensive and, in certain dense urban areas, virtually impossible. It is for these reasons that in the future, the only sustainable way of reducing - or indeed preventing - congestion in the presence of increasing demand for road transportation is through more efficient and intelligent use of the existing road capacity.

The need for a more efficient utilisation of the road network has been recognised in the past. Probably the first step in this direction was the introduction of the traffic light. First deployed in 1868 at the intersection of Great George Street and Bridge Street in London, the traffic light became popular in the US in the 1920s and was initially designed to increase the safety of horse carriages, vehicles and pedestrians at road junctions (BBC, 2009). It took, however, until the 1960s for the first computer-controlled traffic light system to be introduced in what is perhaps the first implementation of an Intelligent Transportation System (ITS) aimed at increasing road traffic efficiency (Figueiredo et al., 2001).

To this day, traffic lights have remained the dominant form of urban traffic management. However, in the case of roads that provide uninterrupted flow such as freeways,

further traffic control systems have been introduced such as ramp meters, dynamic route guidance and variable speed limitation. At the core of these techniques lies the relationship between traffic flow and density as described by the fundamental diagram of traffic flow. Depicted in Fig. 1.1, the fundamental diagram of traffic flow states that as freeway traffic density (D) increases so does the flow (v), until a certain point is reached, called critical density (D_{cr}), at which any increase in density causes a decrease in flow. The flow achieved just before the critical density is reached is called capacity flow (v_{max}) and represents the capacity of the freeway. Roadside-based freeway traffic management techniques ultimately aim at achieving capacity flow by maintaining a density just below the critical density. The capacity of a freeway section depends on a multitude of factors such as the curvature and gradient of the respective road section, as well as weather conditions. For a typical free flow speed of 33.3 m/s (120 km/h), the capacity of a freeway can be as high as 2400 passenger cars per hour per lane (pc/h/ln) (Highway Capacity Manual, 2012).

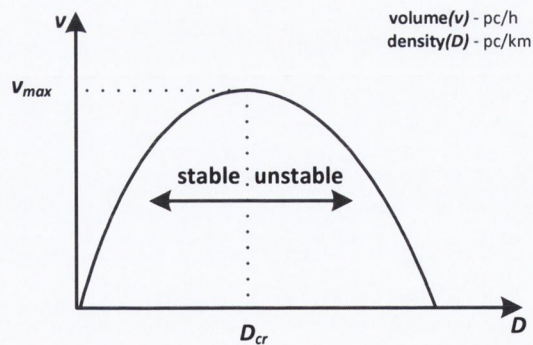


Fig. 1.1: Fundamental diagram of traffic flow

1.2 The Case for Vehicle Automation

Present day ITS deployments aimed at an efficient utilisation of the road network are almost exclusively roadside-based (i.e. there is no direct vehicle actuation). Shladover (2009) shows that for a capacity flow of 2200 pc/ln/h, with vehicles driving at an average speed of 27.8 m/s (100 km/h) and taking into account the average size of a vehicle, the average inter-vehicle headway under free flow conditions, as well as the typical width of a freeway lane, no more than 5% of the road surface is occupied by actual vehicles. This means that at its most efficient point in time, due to the large inter-vehicle space, 95% of the road surface is not utilised. In turn, the large inter-vehicle space is a consequence of the relatively slow reaction times of human drivers (according to Maciuca and Hedrick (1995), the delay for human drivers generally varies between 1.2 and 1.7 seconds) as well as the competitive rather than cooperative nature of human driving (see Vanderbilt, 2009, chap. 4). To increase the utilisation of the road infrastructure beyond this 5% these limitations need to be addressed.

In parallel to efforts from the authorities targeted towards higher traffic efficiency, vehicle manufacturers have focused their efforts on increasing safety and fuel efficiency. As a consequence, the control of the vehicle has been gradually shifting from the human driver to the vehicle itself. Automatic transmission, power steering, anti-lock braking systems (ABS) and electronic stability control are examples of widely deployed systems where control has been handed over to the vehicle. More recently introduced systems such as adaptive cruise control (ACC) and parallel parking take this a step closer to completely removing the need for a human driver. With the adoption of information technologies¹, the automotive industry has seen in recent years a rapid advance towards full vehicle automation. Probably the most prominent present-day example of full vehicle automation is Google's Self-Driving Car (Thurn, 2010), a modified Toyota Prius which finds its roots in Stanford's entry to the DARPA Urban Challenge². This shift towards full vehicle automation, driven mainly by the need to increase safety, can also

¹One can expect the software running on a modern car to have over 100 million lines of code (Charette, 2009)

²<http://archive.darpa.mil/grandchallenge/>

be used to address the problem of large inter-vehicle spaces required by human drivers. Due to the considerably faster reaction times of automated vehicles, the inter-vehicle spacing can be considerably reduced thus significantly increasing the capacity of the freeway. Furthermore, intelligent vehicles using full vehicle automation and communication technologies can coordinate with each other, changing the nature of driving from a competitive to a cooperative form and further increasing the overall traffic efficiency.



Fig. 1.2: Scale model of an automated highway presented at GM's Futurama exhibition at the World Fair in 1939

It is worth noting here that the idea of automated vehicles driving on highways is not new: indeed, as far as at the 1939 World Fair, General Motors's (GM) Pavilion exhibited "Futurama", an utopic vision of the future where, among other things, a scale model of an automated highway (i.e. automated vehicles driving on a highway) was presented (Fig. 1.2). It is important to note just how truly visionary this work was, considering that, at that point in time, the US interstate freeway system did not yet

exist. This is perhaps the first time that the potential of vehicle automation with respect to increasing traffic efficiency was recognized. A system that uses intelligent vehicles to optimise traffic is generally classified as an Intelligent Vehicle/Highway System (IVHS). This thesis describes a novel approach to IVHS that addresses the limitations of previous work.

1.3 Approach

Initial work on IVHS, performed in the mid 1990's and early 2000, focused on the benefits of vehicle automation with respect to freeway capacity and employs a centralized, hierarchical architecture that integrates roadside-based traffic control and intelligent vehicles. Vehicles are assumed to be fully automated and organised into so-called platoons - comprising of one platoon leader and several followers - within which very small inter-vehicle gaps are maintained. In the mid 2000's this approach was largely abandoned, mainly due to the fact that at that time full vehicle automation was in its infancy and a large-scale deployment was deemed unfeasible in the short term. More recent work on IVHS employs a decentralized approach that incrementally builds on the advances in the state-of-the-art with respect to vehicle automation and generally investigates the benefits of inter-vehicle cooperation. This approach is flexible and also scales well due to its reliance on self-organisation instead of centralized management to achieve traffic optimisation. Unfortunately, such an approach does not integrate with roadside-based traffic control systems and as such its benefits are limited.

This thesis introduces a hybrid approach to IVHS by combining the benefits of a centralized management architecture that integrates road traffic control systems with the flexibility and scalability of distributed inter-vehicle coordination. To achieve this, an abstraction layer is used, called *virtual slot*, that allows a loose coupling between the traffic management system (TMS) and IVs. A virtual slot is defined by its length and width as well as a predefined behaviour specified as a speed and trajectory profile. The geometry of a virtual slot is such that it can accommodate a vehicle, creating a corridor through space and time in which a vehicle can drive.

The task of the TMS is to create a set of virtual slots in which vehicles can drive safely and in a manner that increases the overall efficiency of traffic. This set of slots is referred to as the *slot map*. The combined behaviour for all slots in the slot map represents an optimisation solution to the current traffic conditions within the road network. The slot map must address situations such as a decrease in the available capacity, for example due to road works or accidents that lead to bottlenecks on the freeway, inclement weather conditions leading to lower average speed, as well situation where traffic demand increases above the capacity of the freeway, such as during rush hours and special events.

The vehicles are responsible for coordinating in finding and moving into appropriate slots in a safe manner. This means that the TMS does not directly specify the slot in which a vehicle should drive nor directly control the throttle or steering of a vehicle. Instead, the TMS merely specifies the slot map and provides this map (or just parts of it) to the vehicles along with the indication that vehicles are supposed to drive within slots.

The slot-based approach affects the critical density by addressing the limitations of human drivers. Firstly, it addresses the problem of the slow reaction times of human drivers by setting the distance between consecutive slots on the same lane at a value below that of the average inter-vehicle headway and relying on the use of IVs with a higher degree of automation to drive within closely-spaced slots. Secondly, it leverages the use of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication to enforce a collaborative instead of a competitive behaviour between vehicles. As such, the slot-based approach tackles congestion by changing the nature of driving leading to a significant increase in the value of the critical density which corresponds with a higher capacity flow, as exemplified in Fig. 1.3.

1.4 Scope

Although the slot-based driving concept can be adapted and applied to a variety of road types, the scope of this thesis is confined to freeways. Freeways make excellent study

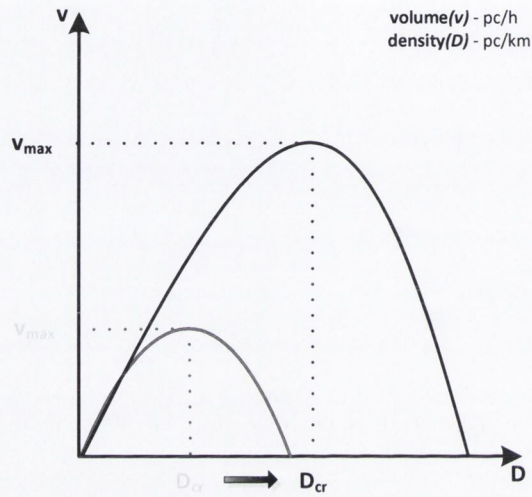


Fig. 1.3: Changing the nature of driving reflected in the fundamental diagram

candidates as they are both the least complex type of roads as well as being arguably the most important. Optimising freeway traffic can have a considerable impact on traffic within the entire road network.

Vehicles on present-day freeways share many features yet are far from being a homogeneous group. Large trucks and buses have large frames, slow accelerations and relatively lower top speeds. Personal vehicles have significantly smaller frames, faster accelerations and higher top speeds. Motorcycles further differ from both personal vehicles and trucks, while even among personal vehicles considerable differences exist. This heterogeneity considerably increases the complexity of the optimisation problem and makes the search for a generic solution very difficult. In such cases, a classic approach used in mathematically-oriented fields of science such as physics, chemistry, mathematical biology and astrophysics (Mitchell (2009) and Thorne (1994) describe several illustrious works in which this approach has been used) is to idealise the model in the hope of finding an initial solution that can then be adapted to a more complex and realistic

model. This work uses the same approach and simplifies the initial vehicle model to a homogeneous set of vehicles that resembles a typical personal vehicle. The choice of personal vehicles is based on the fact they represent the vast majority of present day freeway traffic, e.g. in the U.S. personal vehicles represent between 88% and 90% of all freeway traffic (Highway Capacity Manual, 2012).

The main limitation of the slot-based approach lies in its reliance on a set of requirements with respect to the sensing, communication and automation capabilities that all vehicles participating in freeway traffic subject to optimisation must comply with. Nevertheless, while not deployable on present day roads, the flexibility of the virtual slot approach means that the sensing, communication and automation requirements for the vehicles match the technology available in present day state-of-the-art vehicle prototypes. Furthermore, once a point is reached where a minimum set of requirements is fulfilled, any improvements in the range and precision of sensors and automation translate into an incrementally higher utilisation of the freeway network.

Slot-based driving imposes a shift towards vehicle automation that has implications with respect to the interaction between the human driver and the vehicle. The study of these implications falls under the field of human-computer interaction (HCI), a discipline at the intersection of computer science and behavioural sciences that has increased in popularity in recent years as the advances in technology have allowed humans to communicate and control machines in new ways. The automotive industry has also invested considerably in the field of HCI, especially in the context of the aforementioned switch towards a higher degree of autonomy for vehicles. The HCI aspects of slot-based driving are not considered in this thesis and only the technical feasibility of the slot-based driving concept is investigated.

The use of vehicular communication and automation raises new security concerns. Ranging from privacy issues to malicious attacks that take over the control of a vehicle, these concerns are of great importance and need to be addressed. Fortunately, existing security concepts such as cryptographic-based authentication and encryption can be used to address most of these concerns, while ongoing research is addressing the new type of

security problems that are specific to vehicular communication and automation (Raya and Hubaux, 2007).

The main contribution of this thesis to the state of the art lies in its novel slot-based approach to optimising traffic using IVs. The virtual slots are specified by the TMS in the form of a slot map which allows vehicles to coordinate in an efficient fashion, leading to a significant reduction in journey times. In addition, the deterministic nature of slot-based driving leads to a reduction in travel time variability and thus, by definition (Joint Transport Research Center of OECD and ITF, 2010), to an increase in the overall reliability of road transport. From a non-functional perspective, the slot-based approach is scalable due to its hybrid architectural approach based on a loose coupling between the TMS and individual vehicles. Another important characteristic of the slot-based approach is its flexibility. This is reflected in the lack of strict requirements imposed on the vehicular technology used (e.g. fully-automated vehicles are not explicitly required), enabling a trade-off between a faster route to deployment through less advanced vehicular technology on one side and higher efficiency on the other side. Furthermore, it addresses the likely reluctance of human drivers to hand over the control of the vehicle by permitting an on-the-fly transformation from manual to slot-based driving, thus obliging the human drivers to hand over vehicle control only when the critical density is about to be reached. Both the functional - efficiency and reliability - and the non-functional - scalability and flexibility - properties of the slot-based approach are a consequence of the design decisions made in Chapter 3, and are demonstrated by the case studies presented in Chapter 4.

1.5 Thesis Roadmap

The remainder of this thesis is organised as follows: Chapter 2 analyses the current state of the art with respect to traffic optimisation using intelligent vehicles. Chapter 3 then introduces the slot-based approach to IV-based traffic optimisation, as well as a set of slot-based algorithms that address critical traffic scenarios. In Chapter 4, the design and implementation of an evaluation framework for slot-based driving are described.

This framework is then used to evaluate the proposed slot-based approach, and the thus obtained results are also discussed in Chapter 4. Finally, Chapter 5 concludes this thesis, discusses future work that needs to be addressed, and provides an outlook on the future of intelligent vehicles and their potential to improve everyday life.

Chapter 2

State of the Art

Freeway traffic management concerns itself with the safe and efficient operation of the road surface. This chapter provides a review of state-of-the-art techniques aimed at increasing the efficiency of freeway traffic and is split into two sections. Section 2.1 provides an overview of currently deployed techniques, which are based on the idea of maintaining the density of vehicles over any section of the road below the critical density threshold. Unfortunately, this means that they are limited by the capabilities of human drivers: the critical density is a consequence of the slow reaction times of human drivers and their lack of coordination. Recent advances in vehicle automation and communication promise to address these limitations, leading to the emergence of intelligent vehicles (IVs). While not deployed on present days roads, IVs promise to considerably improve the efficiency of traffic. Section 2.2 provides a review IV-based freeway management techniques.

2.1 State of Practice

Freeway traffic management is currently almost exclusively implemented using roadside infrastructure. The techniques used are highly dependent on the sensing and actuating capabilities deployed in the road infrastructure. Section 2.1.1 provides an overview of the sensors and actuators frequently deployed on present days roads, which is necessary

for a good understanding of roadside-based traffic management techniques.

Based on the location at which traffic management measures are applied, freeway traffic management can be classified as either demand management or supply management. An overview of demand management techniques is provided in Section 2.1.2, while supply management techniques are reviewed in Section 2.1.3.

It is important to note that the scope of this section is not to provide a comprehensive review of all freeway management systems and techniques that have been deployed but rather to provide an overview of the state of the practice limited to the techniques that are the most relevant from an optimisation perspective.

2.1.1 Enabling Technologies

Freeway traffic management is made possible by a variety of technologies that can be grouped, based on their functionality, as either sensors or actuators.

Sensors

Traffic management systems use sensors to detect the presence and passage of vehicles. Most of these sensors are roadside-based and are mounted either in the roadway or over the roadway. In-roadway sensors are embedded in the pavement or attached to the surface of the roadway. The most common in-roadway sensors are inductive loops and magnetic sensors. By contrast, over-roadway sensors are mounted above the surface of the roadway. The most common over-roadway sensors are radar-based sensors and video image processing-based sensors.

Inductive loops are by far the most widely used sensors in present day traffic control systems (Klein et al., 2006). The basic inductive loop system consists of a wire loop that is embedded in the pavement and an electronic unit that energises the wire loop. When a vehicle crosses the wire loop, electrical current is induced into the vehicle, which increases the induction in the loop. The electronic unit detects this increase, thereby inferring the presence or passage of a vehicle. The simple nature of the vehicle detection provided by inductive loops means that algorithms are required to derive traffic flow

parameters such as density and speed using data aggregated from a set of inductive loops located along the freeway.

Microwave radar, laser, infrared and ultrasonic vehicle detection sensors are all based on the same principle. A detection unit is mounted on an overhead bridge or pole and transmits energy that is reflected back by passing vehicles. The low-level data obtained from the energy reflected by vehicles is processed, resulting in higher-level traffic flow data on a per-lane basis. Alternative installations on the side of the road can provide traffic flow data across multiple lanes of the highway.

Video image processing technology has dramatically improved over the last decade. Initially, video cameras were introduced for roadway surveillance and always included a human operator in the loop. Nowadays, video image processors are used to detect vehicles across multiple lanes, provide traffic flow data and even for automatic incident detection (Shehata et al., 2008).

Besides roadside-based sensors, in recent years, due to advances in mobile communication technologies, as well as the high penetration of GPS-enabled devices, crowdsourcing data from mobile sensors deployed within individual vehicles has emerged as an alternative or complementary source of traffic data. In some cases, gathering traffic data from mobile sensors is even considered a cheaper alternative to roadside-based sensors (Bacon et al., 2011). In general, to incentivise the crowdsourcing, drivers are in return provided with live congestion data and estimated travel times (e.g. within INRIX¹) or even routing and turn-by-turn navigation (e.g. Google Maps for Mobile², Waze³).

Unfortunately, none of these sensing technologies are without weaknesses. In-roadway sensors have high installation and maintenance costs (lane closures are required). Video image processors and some of the radar-based sensors perform poorly under certain weather conditions (rain, snow, fog etc.). Furthermore, most radar-based sensors have problems detecting stationary vehicles, while crowdsourcing traffic data, although showing great promise and gathering momentum, is still in its infancy.

¹<http://www.inrix.com>

²<http://googleblog.blogspot.ie/2009/08/bright-side-of-sitting-in-traffic.html>

³<http://www.waze.com>

Actuators

While the sensing technologies used to gather data are quite diverse, the actuation capabilities of traffic management systems are rather limited. Access to the freeway can be controlled using traffic lights, possibly reinforced by barriers in locations such as toll plazas or on-ramps when ramp metering is used. Variable message signs (VMSs) indirectly influence traffic by providing indications such as speed limits, lane closures, alternative routes and hazard warnings.

2.1.2 Demand Management

Demand management describes traffic control measures applied to vehicles before they enter the freeway. There are two basic complementary approaches to demand management: access control and route guidance. Access control is commonly enforced through ramp metering and is applied to vehicles that are about to enter the freeway. Route guidance is applied both to vehicles planning to enter the freeway as well as vehicles on the freeway. In this regard route guidance can be considered as a hybrid between demand and supply management.

2.1.2.1 Access Control

Empirical studies have shown that the maximum flow that can be achieved on a freeway is subject to the critical density. The fundamental diagram of traffic flow (see Fig. 1.1 in Chapter 1) shows that the flow increases directly proportional to the demand until the critical density of the respective freeway section is reached. At this point, an increase in demand causes a decrease in flow. Ramp metering is a technique that attempts to control the demand so that the traffic density is maintained below the critical density. A metering rate defines the amount of vehicles that can enter the freeway at the an on-ramp and is generally enforced using traffic lights. Ramp metering is popular in North America, with around 4000 installations (California Department of Transportation, 2011), with more recent implementations in Europe, Japan and Australia.

Ramp metering strategies can be classified as either fixed-time or reactive. Fixed-

time ramp metering strategies are computed off-line and control the on-ramp access based solely on historical demand data. The reliance on static data makes fixed-time strategies vulnerable to special events such as accidents, road works and weather hazards. Furthermore, fixed-time strategies are also vulnerable to modelling errors. Consequently, under certain circumstances, they may lead to congestion or underutilization of the freeway.

Reactive ramp metering strategies apply a control-theoretic approach where the goal is to keep the traffic conditions close to a pre-specified set of values. Unlike fixed-time ramp metering strategies, traffic demand is measured in real time. Based on the scope of operation, reactive ramp-metering strategies can be either local or coordinated. Local strategies use traffic measurements in the vicinity of the on-ramp, such as upstream flow and downstream freeway occupancy to compute the ramp metering rate. Local ramp metering is performed independently for each ramp in the network, using local measurements. Papageorgiou et al. (2003) argue that, based on empirical evidence, closed-loop local control approaches (i.e. approaches that use feedback from the system output to control the input) such as ALINEA (Papageorgiou et al., 1991) outperform open-loop approaches (i.e. approaches that do not use feedback) such as the demand-capacity strategy (Masher et al., 1975). Unlike local strategies, a coordinated ramp metering strategy uses all traffic data measurements in the network to determine the ramp metering strategy of each individual ramp metering facility. Coordinated ramp metering strategies are much harder to design and implement but they can outperform local strategies (Papamichail et al., 2010).

For a better overview of ramp metering as well as a more comprehensive taxonomy of ramp metering strategies see Papageorgiou and Kotsialos (2000). In general it can be argued that ramp metering is the most efficient form of demand management deployed on present-day roads. However, it is limited by the capacity of the respective on-ramps. A minimum metering rate is required to prevent spillback of the queue onto the adjacent road network. This means that ramp metering cannot prevent congestion under certain conditions such as a high traffic demand upstream of a bottleneck.

Another noteworthy way of controlling traffic demand is through congestion pricing (U.S. Federal Highway Authority, 2008). This technique employs a market economics approach to regulating demand by using a variable toll rate that increases in the presence of higher demand. Higher toll prices are generally employed during rush hours to motivate people to use public transport or at different times of the day. Congestion pricing is generally employed on urban roads such as in cities that are confronted with congestion problems.

2.1.2.2 Route Guidance

The total travel time experienced by individual vehicles along the course of a journey varies according to the traffic conditions along that route. In choosing the fastest route, drivers need to consider a complex set of factors that influence traffic conditions across all available routes. Driver information and guidance systems assist the driver in this process by providing route guidance. Herbert and Mili (2008) propose a taxonomy which classifies route guidance systems based on:

- the nature of data used to predict travel times;
- the form in which the route guidance is provided to the users;
- the user coverage.

Depending on the nature of data used to predict travel times, route guidance systems can be classified as either static or dynamic. Static systems use historical data where traffic demand varies according to the time of the day, day of the week and even time of the year. Dynamic systems use real-time traffic data which includes current demand as well as information about conditions such as road works or incidents along the freeway. Travel-time prediction is arguably the most complex and important component of route guidance systems. An overview of travel time prediction approaches can be found in Lint (2004).

Route guidance can be provided to drivers either as descriptive or prescriptive guidance. Descriptive guidance provides information on traffic conditions but offers no rout-

ing recommendations. Prescriptive guidance goes one step further and provides routing advice to drivers. In general, traffic operators as well as drivers prefer descriptive guidance, as prescriptive guidance systems require very accurate predictions and are thus hard to implement in practice (Papageorgiou et al., 2003). Route guidance information can be communicated to drivers using VMSs or in-vehicle technology such as turn-by-turn navigation. In Japan, the Vehicle Information and Communication System (VICS) provides, among others, information on congestion conditions and travel times. VICS is very popular with an estimated 30 million installations in 2010. Due to its very high adoption rate, it is expected that VICS can also be used in the future to disperse traffic flow and improve the general traffic fluidity (Japanese Ministry of Land, Infrastructure and Tourism, 2008).

The number of users targeted by a route guidance system can have a significant impact on the overall traffic flow. Users can be targeted individually, in which case the goal is to optimise the travel time for an individual user and hence the system is called user-optimal. When route guidance is aimed at all the users in the network with the goal of optimising the overall traffic flow, then it is called system-optimal.

In general, route guidance is used by traffic operators and individual drivers to reduce travel times and increase their reliability. However, the positive effects of these systems are limited by their advisory rather than enforcing nature. System-optimal routing is particularly hard to implement when routing decisions cannot be enforced on individual drivers, while user-optimal routing can have a negative impact on the system-wide traffic flow.

2.1.3 Supply Management

Supply management refers a variety of potentially complementary techniques used to control the traffic supply along the freeway (i.e. the vehicles that have already entered the freeway). These techniques can be applied in a proactive fashion, using historical data, or in a reactive fashion, as consequence of an incident (e.g. a traffic accident). This section provides an overview of the most relevant supply management techniques

with respect to traffic efficiency.

Note that an essential component on which many supply management techniques are based is incident detection. Incident detection systems are based on two components: a data collection component and a data processing component. The data collection component gathers data through a variety of sensors such as inductive loops, video image processing for automated vehicle identification and GPS for vehicle localisation. The thus gathered data is then processed to identify possible incidents along the highway. The processing can be done either manually by a human operator, or automatically by an incident detection algorithm. The development of new sensor technologies, as described earlier in this chapter, promises to help improve the accuracy of automated incident detection. An overview of incident detection algorithms and their respective deployments can be found in Parkany and Xie (2005).

2.1.3.1 Variable Speed Limitation

Variable speed limitation (VSL) is a technique used to address the problem of traffic jam propagation. This propagation can take place in time (i.e. the time required for the congestion to dissipate), as well as in space, due to the shockwave effect, where incrementally heavier braking eventually leads to a traffic jam that propagates upstream. Hegyi et al. (2005) classify VSL approaches based on their goals as approaches that aim to either achieve a homogenisation of the traffic flow or to prevent too high densities.

Homogenisation approaches use speed limits to reduce the speed differences between vehicles. A homogenisation of the traffic flow is believed to lead to both safer driving and more stable flows, since smaller speed differences give drivers more time to react, minimise the impact in case of accidents and do not require excessive braking that can cause a shockwave to propagate upstream. In general, the speed limit is set slightly below the current average speed but above the critical speed (i.e. the minimum speed needed to achieve maximum flow), thus slightly increasing density while maintaining the same traffic flow. The main limitation of this approach is that it cannot suppress existing shockwaves.

VSL approaches that aim to prevent high densities use speed limits below the critical speed to reduce inflow into a congested area. This means that shockwaves can be dissipated in low density areas. See Hegyi et al. (2005) for an overview of the control methodologies used to implement VSL.

VSL can also be used to optimise traffic flow at work zones. Lin et al. (2004) describe two VSL algorithms, one aimed at reducing the queue upstream of the work zone and the other aimed at increasing the throughput. The simulation-based evaluation of these algorithms shows that VSL can be used to increase the throughput and reduce delay at work zones.

Traditionally, VSL is implemented using VMS and therefore cannot be directly enforced on individual drivers. More recently, Intelligent Speed Adaptation (ISA) - a system that allows the speed of a vehicle to be controlled externally - has been considered as a more advanced way of implementing VSL to increase efficiency (Boriboonsomsin et al., 2008; Liu and Tate, 2004) as well as safety (Carsten and Tate, 2005). The generic ISA system comprises of a central traffic management component that decides the speed limit; a communication infrastructure between the central component and vehicles on the freeway; and vehicles equipped with GPS, digital maps and means of either communicating or enforcing the speed limit to the driver such as haptic feedback on the acceleration pedal. Thus, ISA systems are generally classified based on how the speed limit is imposed:

- Advisory: similarly to classic VSL, the speed limit is simply suggested to the driver
- Mandatory: the speed limit is enforced upon the driver

The rather complex combination of required technologies and collaboration between vehicle manufacturers and traffic operators means that currently ISA is mostly in a trial stage (Liu and Tate, 2004; Vlassenroot et al., 2007).

2.1.3.2 Lane Control

A complementary approach to VSL for managing the traffic supply is lane control. Lane control is used to either (temporarily) restrict access to a lane to only specific vehicles or to change the direction of the traffic flow for one or more lanes. The former is generally implemented on a freeway in the form of high occupancy vehicle (HOV) lanes and has been deployed mainly in North America, Australia and New Zealand. As the name suggests, HOV lanes are dedicated lanes for vehicles with two or more passengers which are meant to encourage carpooling leading to a more efficient utilisation of the infrastructure. A report from the year 2000 on the efficiency of HOV lanes in California, one of the US states with the highest number of HOV lanes, showed that the performance of HOV lanes is mixed and that, in general, HOV lanes are being underutilised (California, LAO, 2000).

A different approach to lane control is represented in the form of reversible lanes. They are used to deal with an increase in demand in one traffic direction such as during rush hours or during emergency evacuations. Reversible lanes are commonly found on bridges and in tunnels and operate using overhead traffic lights.

2.1.3.3 Merging Traffic Control

Freeway bottlenecks are caused by a drop in capacity encountered at on-ramps, work zones, toll plazas or at the merging point of two freeways. Under high traffic demand, these bottlenecks cause congestion. Merging traffic control techniques addresses this problem by attempting to maintain capacity flow through the bottleneck area. For example, Papageorgiou et al. (2008) addressed the merging problem as a special form of the ramp metering problem and proposed an approach based on local ramp metering, more specifically the ALINEA algorithm. It is worth noting that for toll plazas or freeway interchanges, the initial design of the road layout plays a crucial role with respect to the efficiency of traffic at those sections. Aspects of road layout design are, however, outside the scope of this work.

2.1.4 Discussion

This section presented an overview of existing approaches to traffic management that are aimed at increasing the efficiency of freeway traffic. Demand management strategies generally fail in cases where traffic demand is much higher than the available supply, e.g. ramp metering when there is the risk of spillback from the on-ramp to an adjacent road or route guidance when all the available routes are congested. Supply management measures on the other hand tend to be of a heuristic nature and few studies have been conducted to quantify their impact (Papageorgiou et al., 2003). VSL cannot directly enforce speed limits on vehicles, while the large-scale deployment of ISA requires technical challenges to be overcome.

These approaches are generally implemented independently and the lack of coordination between the control strategies leads to suboptimal results. In the short term, the focus should be on integrating of complementary demand and supply management strategies into a network-wide solution. Nevertheless, the improvements that can be achieved using roadside-only freeway traffic management with respect to traffic efficiency are limited by the nature of human drivers. Due to the relatively slow reaction times of human drivers, vehicles on freeways need to leave a safety gap. The average lateral gap is determined by subtracting the average width of a vehicle from the width of a freeway lane. In the US, the width of a highway is 3.5m (TRB, 2000). The width of a personal vehicle rarely exceeds 1.8m while the width of a bus or truck, representing 5-10% of the entire freeway traffic, can reach 2.74m (Shladover, 2009). These numbers show that a maximum of around 50% of the lateral actual width of a freeway can be used at any time due to enforcement of lanes. Furthermore, Shladover has shown that even when a maximum highway capacity is reached, no more than 11% of length of a lane is occupied by vehicles (Shladover, 2009), while the remainder represents the safety gap left by the drivers. This means that currently, due to the limitations of human drivers, even at capacity traffic flow, only approximately 5% of the freeway surface can actually be occupied by vehicles. As such, in the face of increasing traffic demand, it is the limitations of human drivers that will need to be addressed in the long term.

2.2 Using Intelligent Vehicles to Optimise Traffic

The core advances in the present generation of vehicles lie in providing basic control assistance (e.g. traction control, antilock braking system) used mainly for increasing safety, followed by a variety of functions aimed at driver and passenger comfort. The next generation of vehicles promises to take this a step further by using information gathered by vehicle sensors to assist the driver. Such vehicles are generally called *intelligent vehicles* (IV). While this term is rather generic and can be misleading, for the purpose of this thesis the definition provided by Bishop (2005) is used, which states that IVs are vehicles that “sense the driving environment and provide information or vehicle control to assist the driver in optimal vehicle operation”. Furthermore, they “operate at the tactical level of driving (throttle, brakes, steering) as contrasted with strategic decisions such as route choice”. While mainly aimed at safety applications, the capabilities of IVs can also be used to improve the flow of traffic as part of so-called Intelligent Vehicle Highway Systems (IVHS).

This section provides an overview of the state of the art with respect to IVHS and is organized in two parts. The first part provides an overview of advances in vehicular technologies that are at the core of IVs and are relevant in the context of IVHS. The second part then provides a survey of existing IVHS approaches and discusses their benefits and limitations.

2.2.1 Intelligent Vehicles

Understanding the vehicular technologies that enable IVHSs necessitates an understanding of the requirements that IVHSs impose on participating vehicles. While the exact requirements depend on the specifics of the IVHS, a vehicle must be able to fulfil three fundamental requirements:

- sense its environment;
- communicate with other vehicles and/or with the infrastructure;
- assist the driver by providing information and/or vehicle control.

Vehicular technologies that address these requirements are presented in the following paragraphs, followed by a discussion on how the current state of the art addresses the requirements.

2.2.1.1 Sensing

Currently used to increase safety and driver comfort as well as to provide navigation capabilities, vehicular sensing is a fundamental component of the IVHS of the future. There is a large variety of sensors in a modern vehicle, such as sensors used to measure the exhaust, accelerometers used by airbag systems to detect when an accident takes place, video cameras for driver face and gaze tracking and many others. Fleming (2001) provides a comprehensive overview and classification of automotive sensors, where sensors are classified based on their area of systems application as:

- Powertrain sensors, used in the engine, the transmission and to provide onboard diagnostics (e.g. engine misfire, electronic transmission control)
- Chassis sensors, used in various vehicle dynamics aspects (e.g. ABS braking, traction control, steer-by-wire)
- Body sensors, generally used to provide user comfort (e.g. navigation, adaptive cruise control (ACC))

The category of automotive sensors that is most relevant in the context of IVHS are body sensors. More specifically, shifting the driving task from manual control to fully automated control requires a very good knowledge of a vehicle's surrounding environment. To achieve this, sensors need to be able to provide both the absolute position of a vehicle in global coordinates, as well as the relative position of a vehicle with respect to objects in its neighbourhood.

Absolute positioning systems provide a vehicle with its current position, in global coordinates, and are generally available in the form of Global Navigation Satellite Systems (GNSS). The accuracy of GNSS typically ranges from a few centimetres to over ten

meters. Broadly speaking, a GNSS consists of a set of transmitters mounted on satellites in the orbit of the Earth that broadcast microwave signals towards Earth. GNSS receivers located on or inside a vehicle triangulate these signals to determine the position of the vehicle. The accuracy of the determined position is subject to a variety of factors, mainly the number of satellites available, atmospheric and relativistic effects as well as the reflection of radio waves on objects, an effect known as multipath. Only two GNSS are currently operational: the US-owned Global Positioning System (GPS) and the Global Navigation Satellite System (GLONASS), owned by the Russian Federation. Further systems such as Galileo - a joint initiative of the European Commission (EC) and the European Space Agency (ESA) - are expected to be available in the future.

The GPS is currently by far the most widely-used GNSS. Over the last decade, GPS-based navigation devices have become increasingly popular and are used in vehicles to provide navigation assistance to the human driver. However, the relatively poor accuracy of standard GPS devices - 13m horizontal accuracy (Hegarty and Chatre, 2008) - means that accuracy improvements are required to achieve the higher degree of autonomy required by intelligent vehicles. Fortunately, various approaches that improve the position accuracy have been developed. One such approach uses differential GPS (DGPS) to reduce spatially correlated errors. A reference receiver installed at a well-known fixed location receives signals from the satellites and computes a differential correction, which is then broadcast to user DGPS receivers. As the distance between the reference and the user receiver increases, the positioning accuracy decreases. However, within a radius of 100km around the reference radius, a sub-meter accuracy has been observed (Monteiro et al., 2005). Using a similar approach, Real Time Kinematic GPS (RTK-GPS) can achieve centimetre-level accuracy (Lenain et al., 2003). Satellite-Based Augmentation Systems (SBAS) such as the European Geostationary Navigation Overlay Service (EGNOS) or its US-equivalent, the Wide Area Augmentation System (WAAS), apply the same technique used by DGPS to wide areas. These systems determine errors caused by atmospheric effects (mainly ionospheric errors) and broadcast their corrections to the receivers. SBAS are capable of reaching an accuracy of under 5m (Gauthier et al., 2001).

Relative positioning provides a vehicle with its position relative to objects in its neighbourhood. Different types of sensors are being used for relative positioning, of which the most common ones are:

- Ultrasound
- Laser radar (lidar)
- Millimetre-wave radar
- Machine vision

Ultrasonic sensors are used for near-distance (a few meters) object detection (Bishop, 2005). They are low-cost, offer wide-area coverage and are generally used in parking-assist systems. Laser and millimetre-wave radars are used for far-distance object detection, usually between 2 and 150 meters (Abou-Jaoude, 2003). In general, millimetre-wave radars are more reliable in poor weather conditions than lidar. On the other hand, lidars are more accurate and provide a wider angular coverage (Fleming, 2001). Both are commonly used to provide range and closing-rate information to ACC systems. Machine vision is probably the most multi-purpose sensor and has a variety of applications (Zhang et al., 2011), the most widely used ones being tracking of road lane markings, driver face and gaze tracking, and pedestrian detection.

Sensor data fusion refers to the fusing of data describing the surrounding environment as gathered by sensors into a so-called “world model”. Such a world model is required by intelligent vehicles to be able to perform complex manoeuvres such as overtaking a vehicle or merging into a freeway. Traditionally, sensor data fusion has been an active research topic within the robotics community. Thurn (2002) provides an overview of robotic mapping techniques. Such techniques have been applied to autonomous driving within the frame of the DARPA Grand Challenge⁴ and are currently employed by the Google Self-Driving Car (Thurn, 2010). However, the mapping approaches employed

⁴<http://archive.darpa.mil/grandchallenge/>

for vehicles differ slightly from the ones employed for robots. Unlike a robot exploring an unknown environment (a problem commonly known as simultaneous localization and mapping (SLAM)), the layout of a road is well defined and can be captured in digital maps. Furthermore, vehicle cooperation can be employed to enhance the quality and range of the world model, such as in Papp et al. (2008).

2.2.1.2 Communication

Vehicular communication enables a wide-range of applications both within and beyond the scope of traffic management. It is expected that equipping vehicles with communication capabilities will provide a platform for inter-vehicle coordination that will increase the safety and efficiency of traffic as well as provide useful traffic information and entertainment to the occupants of the vehicles. In general, vehicular communication can be classified as being either vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) or hybrid. The umbrella term vehicular networking is generally used to describe all forms of vehicular communication.

Vehicular networks are inherently wireless, ad-hoc and very mobile, posing a special set of challenges. Indeed, the reflecting nature of a vehicle's body can impact on the quality and strength of the signal. Furthermore, the mobile nature of vehicles and the high speeds that they can achieve impose a very challenging set of requirements on routing and membership protocols. These are made even more difficult when safety-critical applications such as collision avoidance or safe overtaking are considered, as they require both real-time and guaranteed message delivery. Finally, security in general, and trust and privacy in particular, are crucial aspects given the safety-critical nature of driving.

More recently, research in vehicular networking was focused on creating a set of standards for Dedicated Short-Range Communication (DSRC) that would enable a variety of applications, mostly around increasing safety but also increasing traffic efficiency. Standardisation efforts in Europe, US and Japan have led to the emergence of the IEEE 802.11p protocol that describes physical and data-link layers of DSRC. While the IEEE

802.11p protocol is common to standardisation efforts in all three geographical regions, the protocols located higher on the DSRC stack vary across the three regions. In the US, the IEEE 1609 protocol set is used on top of IEEE 802.11p by projects such as IntelliDrive⁵ and California PATH⁶. This combination of IEEE 1609 and IEEE 802.11p is known as the DSRC/WAVE standard. Research efforts in Europe as part of projects such as SAFESPOT⁷, CVIS⁸, CarTalk2000⁹, COOPERS¹⁰ and PREVENT¹¹ have led to the emergence of the ISO CALM protocol suite which specifies how existing wireless technologies such as cellular communication and 802.11p can be used by the upper layers. Finally, in Japan, DSRC was already used in 2008 by approximately 20 million vehicles for electronic toll collection (Karagiannis et al., 2011).

Surveys performed over the past years by Sichitiu and Kihl (2008); Toor et al. (2008); Hartenstein and Laberteaux (2010); Karagiannis et al. (2011); Kenney (2011) indicate that vehicular communication standards are reaching a level of maturity that would permit the deployment of DSRC on production vehicles in the next years. Nevertheless, Karagiannis et al. (2011) identifies several research challenges in areas associated with vehicular communication, such as reliability, robustness, security and, in particular, geographical addressing (i.e. addressing messages based on the position of a vehicle). Finally, once these technical hurdles are surpassed, political and economic issues related to the mass deployment of vehicular communication technologies will still need to be addressed.

It is worth noting that besides new protocols such as DSRC/WAVE, which are yet to be deployed on production vehicles, initial studies indicate that cellular communication technologies such as 3G and the emerging LTE 4G, which have already been deployed on a large scale, could be used for vehicular communication (Matolak et al., 2011).

⁵<http://www.its.dot.gov/>

⁶<http://www.path.berkeley.edu/>

⁷<http://www.safespot-eu.org/>

⁸<http://www.cvisproject.org/>

⁹<http://www.cartalk2000.net/>

¹⁰<http://www.coopers-ip.eu/>

¹¹<http://prevent.ertico.webhouse.net/>

2.2.1.3 Vehicle Control

Shifting the control of the vehicle from the human driver to the vehicle itself, as argued in Chapter 1, can increase the utilisation of the road surface and therefore have a significant impact on the efficiency of traffic. The following paragraphs review the state of the art with respect to first lateral and then longitudinal control, followed by an overview of the latest advances in autonomous driving.

Lateral control refers to the automatic control of a vehicle's steering and builds on the capacity of a vehicle to detect the lanes of the road. Three approaches to lane detection have been investigated:

- magnetic markers embedded in the roadway;
- combining highly accurate GPS and digital maps;
- machine vision.

Machine vision has proven to be the most popular, mainly due to the significant efforts required to embed magnetic markers in the roadway or to create accurate digital maps on a large scale. Recent research efforts have addressed the early limitations (mainly weather-related) of machine vision (Bishop, 2005).

Lane detection has been initially deployed in production vehicles as part of Lane Departure Warning (LDW) and Road Departure Warning (RDW) systems. These systems assist the driver by means of audio or haptic feedback, such as simulating directional rumble strip sounds through sounds and vibrations of the driver seat.

With both LDW and RDW systems, the control of the vehicle is fully in the hands of the human driver. This is not the case with Lane Keeping Assistance (LKA) systems. LKA takes the LDW concept a step further by adding an automatic steering torque to the vehicle steering systems. The torque increases as the vehicle approaches the edge of the lane, creating a so-called "driving in a bathtub" effect (Bishop, 2005). Note that with LKA the attention of the human driver is still required, since LKA uses a shared control paradigm where 80% of the control is provided by the vehicle and 20%

by the human driver. Initially deployed on trucks, LKA systems have been introduced in the recent years to high-cost personal vehicles from manufacturers such as Daimler, BMW and Toyota (Oagana, 2010).

While keeping vehicles within their lane has been subject of past research and systems like LDW and LKA are available in many vehicles, current research is focusing on the next step of lateral control: automated lane-changing. This is applied to a variety of scenarios such as in collaboration with V2V communication to perform vehicle platooning lifecycle operations such as leaving and joining a platoon of vehicles (Rajamani, 2000), emergency manoeuvres such as obstacle avoidance (Swaroop and Yoon, 1999) and automating the vehicle overtaking procedure (Naranjo et al., 2008).

Longitudinal control refers to the automatic control of a vehicle's speed and acceleration. Cruise control (CC) systems have been available in production vehicles starting from the 1950s and are used to maintain a vehicle at a constant predefined speed. Present day high-cost vehicles use ACC, a further enhancement of CC, first introduced by Toyota in Japan in 1998 (Jones, 2001). The distance to the vehicle in front is measured by a lidar or radar sensor and is then combined with a measurement of the vehicle's current speed to determine the headway. The speed of the car is controlled so that a constant headway is maintained, as opposed to the constant speed provided by CC systems.

A survey of past research dedicated to ACC control schemes is available in Vahidi and Eskandarian (2003). More recent research on longitudinal control has focused on further enhancements of ACC such as cooperative adaptive cruise control (CACC) and vehicle platooning. CACC and vehicle platooning use vehicular communication to enhance the information available to ACC systems. Better information can be used to increase safety and driver comfort by anticipating the manoeuvres of the leading vehicle as well as increase traffic efficiency by enabling vehicles to drive in close formations.

Autonomous vehicles are vehicles capable of performing fully automated lateral and longitudinal control without human intervention. To achieve this, a vehicle obtains information from the sensors about its surrounding environment. Over the last decades,

the concept of a fully autonomous vehicle has evolved from a futuristic vision to the stage of working prototypes. The late 1980s and early 1990s saw the first research projects in Europe - DRIVE and PROMETHEUS (Williams, 1998), the United States - PATH (Shladover, 2007), and Japan - DEMO 2000 (Kato et al., 2002) - that investigate the use of autonomous vehicles to increase safety, decrease the impact of road traffic on the environment and increase traffic efficiency. These efforts resulted in the first demonstrators of autonomous vehicles capable of driving in platoons on highways. Over the following years, due to practical and financial limitations at that time, focus shifted from autonomous vehicles to driver assistance systems in general and safety applications in particular.

Nevertheless, over the last decade research on autonomous vehicles has been reignited, mainly due to a series of challenges sponsored by the US defence agency DARPA that combined the efforts of researchers from automotive and robotics fields. The DARPA Urban Challenge, the last of the three DARPA Grand Challenges, demonstrated autonomous vehicles capable of navigating an urban environment (e.g. MIT's Talos (Leonard et al., 2008)). Perhaps the most advanced autonomous vehicle of present day is Google's Self-Driving Car, a modified Toyota Prius which finds its roots in Stanford's entry to the DARPA Urban Challenge.

Unfortunately, having fully autonomous vehicles alone is not enough to increase the efficiency of traffic. This is because these vehicles are mainly designed around the idea of effectively replicating the way humans currently drive, with the goal of increasing driver safety, comfort and accessibility. To increase the efficiency of traffic, a system that allows these vehicles to go from a human-inspired, competitive form of driving to a collaborative one is required.

2.2.1.4 Discussion

The three fundamental requirements imposed by IVHS on IVs are addressed by next-generation vehicular technology. The sensing technology is mature and has been widely used in production vehicles. High-cost production vehicles use relative sensing to increase

safety and driver comfort. Absolute positioning is generally available in the form of GPS receivers and has been widely used in combination with digital maps to provide navigation support to drivers. However, IVHSs assume a high degree of vehicle autonomy and typically impose a high absolute positioning accuracy requirement that cannot be fulfilled by standard GPS receivers. This issue is addressed by DGPS and RTK-GPS which provide sub-meter accuracy.

Similarly to sensing, automated lateral and longitudinal control have been already deployed on high-cost or commercial vehicles but are yet to be integrated in everyday vehicles. To achieve this, the information received by various sensors needs to be fused into a model of the surrounding environment - usually called a world model - to enable the automation of complex driving manoeuvres. Systems that fuse data from various sensors into a so-called world model are generally still in the prototype phase, as are vehicles capable of driving autonomously.

Unlike sensing and vehicular control, vehicular communication is yet to be largely deployed in production vehicles. This is mainly due to the fact that while sensing and vehicle control can be used independently of other vehicles to provide increased safety and comfort, vehicular communication requires the collaboration of several vehicles and/or the infrastructure which is complicated by the high mobility and speed of vehicles, and raises security and privacy issues. Additionally, a significant part of recent research into intelligent vehicles has been ignited by the DARPA Grand Challenges, which did not involve cooperation between vehicles. Consequently, vehicular communication was not part of this initial effort and has been lagging behind ever since.

In conclusion, the state-of-the-art vehicular technology matches most of the requirements of IVHS, while the remaining issues such as accurate positioning, sensor fusion and in particular vehicular communications are currently being addressed. Perhaps the best showcase of the state of the art is represented by the Google self-driving car, which has recently completed 300,000 miles on public roads without a single accident (Rosen, 2012).

2.2.2 Intelligent Vehicle Highway Systems

Approaches that use IVs to optimise freeway traffic are generally discussed under the umbrella term Intelligent Vehicle Highway Systems (IVHS). In the following, a survey of IVHS is presented based on a classification that splits the relevant work into two categories: centralized and decentralized approaches. The first represents approaches where traffic is managed within centralized and hierarchical control frameworks while the second category represents decentralized, self-organising techniques for managing traffic with IVs. It is important to note that this review does not include IV-based traffic optimisation techniques that are specific to urban roads, such as the traffic management system proposed by Milanés et al. (2012).

2.2.2.1 Centralized Approaches

Centralized approaches are characterized by their use of comprehensive control frameworks with a hierarchical structure. At the top of this structure lies a central component that defines the traffic management policies which are subsequently implemented at the lower levels. The most relevant frameworks that have been proposed are presented in the following paragraphs.

PATH AHS

The California Partners for Advanced Transit and Highways (PATH) Program was funded in 1986 and focused a significant part of its effort on increasing highway capacity through automated driving, an approach that became known as Automated Highway Systems (AHS). Experiments with vehicle automation started in the late 1980s and by 1992 the first experiments involving automated vehicle control and communication via wireless LAN were concluded. These experiments culminated in a demonstration of a four-vehicle platoon in 1994 and an eight-vehicle platoon in 1997. In time, the PATH program adopted an incremental approach to highway automation and the focus shifted on automating heavy duty vehicles such as trucks and buses first. Nevertheless, in the process of designing an AHS, the PATH program proposed the first architecture for an

IVHS, which proved very influential and inspired most of the following designs of IVHSs.

The first sketch of the PATH IVHS architecture was introduced in 1991 by Varaiya and Shladover (1991). A more refined version of the same concept is described in Horowitz and Varaiya (2000). The network of highways is modelled as a collection of interconnected links, usually between 0.5 and 5 kilometres in length. Each link is composed of a sequence of segments, representing a highway section delimited at each end by an entry or exit point. Vehicles are assumed to be either free agents or followers - when they follow a vehicle ahead - thus creating a formation known as platoon. The PATH architecture uses a hierarchical control approach over five layers, as shown in Fig. 2.1. The lower three layers are part of a vehicle's on-board system while the upper two are part of the roadside traffic management system.

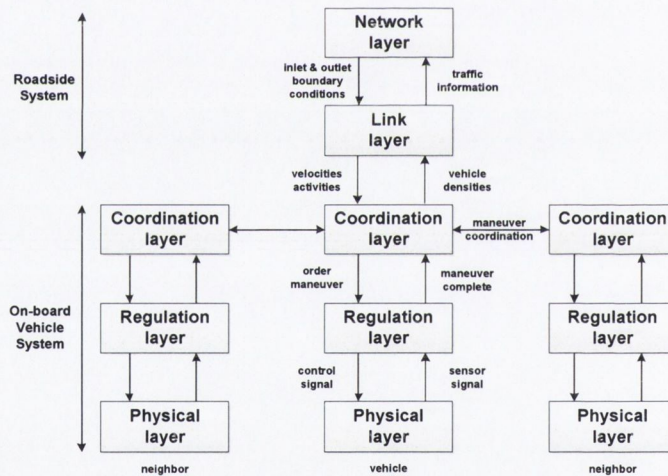


Fig. 2.1: The IVHS architecture introduced by PATH (after Horowitz and Varaiya (2000))

The physical layer deals with a vehicle's dynamics based on commands received from the regulation layer. It uses information from the vehicle's sensors to control the throttle, braking and steering. The regulation layer translates manoeuvres specified by the coordination layer, e.g. merging into the highway, joining and spitting a platoon or changing a lane, into lateral and longitudinal control. The manoeuvres that a vehicle

needs to perform are determined at the coordination layer, based on the activity plan of the vehicle as determined by the link layer and coordination with neighbouring vehicles. The activity plan of each vehicle is computed at the link layer and communicated to each individual vehicle. An activity plan contains information such as desired speed and lane, as well as desired platoon size, and is determined based on the desired traffic flow characteristics specified by the network layer. The optimisation task for the entire network is performed at the network layer, where each vehicle or platoon is assigned a route.

The hierarchical control over five layers was designed to create self-contained layer with clear interfaces between them. The platoon abstraction simplified the design and implementation of the lower on-board vehicle control layers. However, the platoon abstraction also enforces a tight coupling between the roadside and on-board control systems, where the link layer needs to communicate to each vehicle the desired speed profile, route and platoon size. This tight coupling reflects negatively in the form of a complex link layer controller, as one can observe by studying the PATH implementation described by Alvarez and Horowitz (1997). Furthermore, the need for the link layer to communicate which each individual vehicle increases the overhead of communication while the fact that the network layer needs to provide a route for each individual vehicle means that the PATH IVHS architecture scales poorly with respect to the number of vehicles in the system.

Dolphin

The first work inspired by the PATH AHS was the Japanese Dolphin framework (Tsugawa et al., 2000; Kato et al., 2002). The name comes from the platoon formation adopted, which resembles the formation adopted by swarms of migrating birds, known to be aerodynamically efficient, while dolphins also swim in this formation without collisions (Kato et al., 2002). The management architecture, described by Tsugawa et al. (2000), is composed of three layers, as shown in Figure 2.2. The vehicle control and vehicle management layers are part of the on-board vehicle control system, while the

traffic control layer is part of the roadside infrastructure.

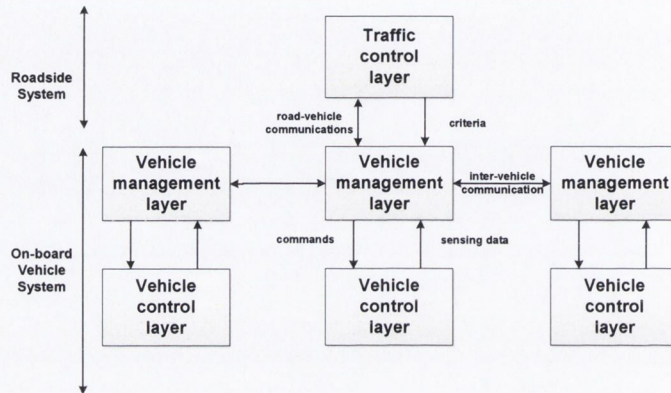


Fig. 2.2: The Dolphin architecture

The vehicle control layer is responsible for providing sensor data to the vehicle management layer and for actuating the lateral and longitudinal control based on commands received from the vehicle management layer. The vehicle management layer determines which movements the vehicle needs to perform based on the suggestions received from the traffic control layer and coordinates performing these manoeuvres with neighbouring vehicles. The traffic control layer is distributed into local traffic controllers that have a view of the cars in their neighbourhood and provide these cars with driving instructions.

Most of the work in the Dolphin project has focused on implementing cooperative platoon driving and culminated with the Demo 2000 demonstration of platoons driving side-by-side and then merging. However, the interface between the traffic control layer and the vehicle management layer is not thoroughly specified. Furthermore, to the best of the author's knowledge, no evaluation of the traffic control layer has been published.

Auto21

Inspired by the PATH AHS and Dolphin frameworks, the Canadian Auto21 project developed a framework for collaborative driving which they named the Driving Agent Architecture (Halle et al., 2004). Depicted in Fig. 2.3, the Driving Agent Architecture

comprises of three layers: the guidance layer, the management layer and the traffic control layer. While the guidance and traffic control layers are virtually identical to the corresponding layers in the Dolphin framework, the core focus of this work has been on the collaboration strategies at the management layer. The management layer is further divided into two sub-layers: the networking layer and the linking layer. The networking layer handles the coordination with respect to the place of each vehicle in the platoon, while the linking layer is responsible for merging into or splitting from a platoon, based on information received from the traffic control layer.

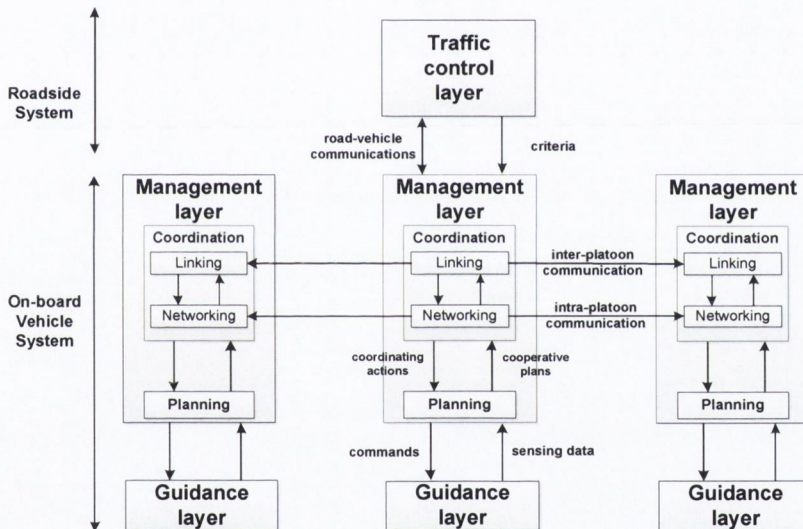


Fig. 2.3: The Auto21 architecture

Using the platoon and vehicle-agent abstractions, the management layer provides both inter-platoon as well as intra-platoon coordination. One of the characteristics of the Auto21 research has been the multiagent systems approach to vehicle coordination. In particular, Hallé and Chaib-draa (2005) proposed the teamwork model used in multiagent systems as a distributed coordination approach. Unfortunately, the evaluation scope of this work has been solely on the performance on the coordination strategy with respect to the communication and planning overhead and, to the best of the author's

knowledge, no results of the impact of this approach on traffic flow have been published.

Baskar et al.

Inspired by PATH, Baskar et al. (2007, 2008, 2011, 2012) proposed a hierarchical traffic management framework for AHS. Similarly to previously presented frameworks, this work uses the platoon abstraction and is structured as a hierarchy of controllers, depicted in Fig. 2.4, comprising a supraregional controller that supervises several regional controllers, each in turn supervising a set of area controllers. At the other end lie the vehicle controllers which, similarly to the previously presented frameworks, provide sensor data to the upper layer while performing the lower-level actuation such as steering, braking and controlling the throttle.

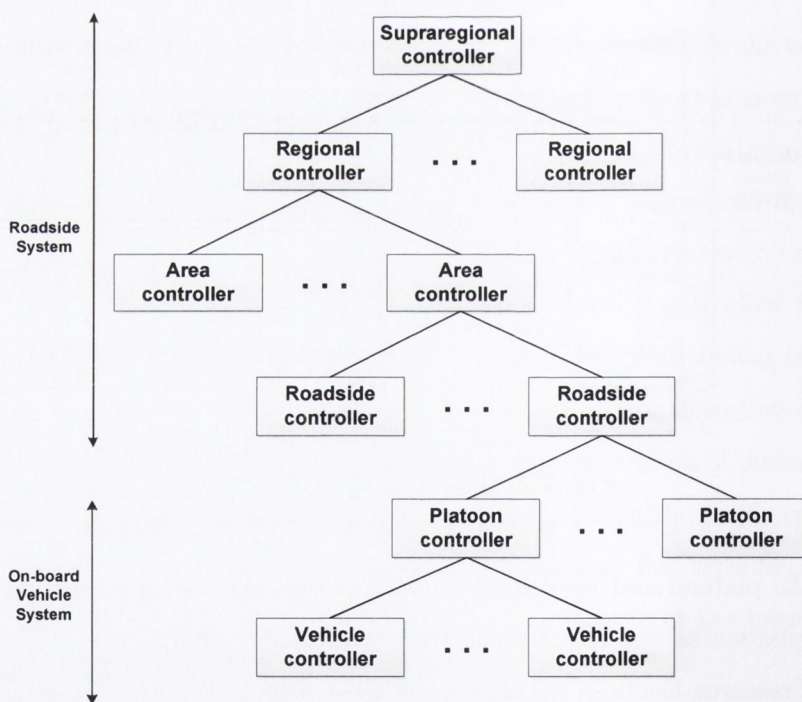


Fig. 2.4: IV-based framework, after Baskar et al. (2012)

The core focus of this work lies in the design of the roadside and platoon controllers.

The benefits of using automated vehicles are combined with traditional roadside-based traffic control techniques such as variable speed limits, route guidance and ramp metering into an integrated solution. Particular to this work, a model-based predictive control (MPC) approach is used to determine appropriate speed limits and lane allocations for platoons, as well as appropriate release times for vehicles at on-ramps. While the initial results of this work are promising, the published case studies evaluate the approach in scenarios which show an increase in capacity that is below the potential of vehicle automation. In Baskar et al. (2012), the capacity achieved is approximately 1850 vehicles/lane/hour, representing a capacity increase of only approximately 16%, which is clearly below the threefold increase in capacity that vehicle automation is expected to achieve, as argued by Varaiya (1993).

Com2React

The EU-sponsored Com2React project (Mincheva, 2009) proposed a hierarchical traffic management framework where the role of the roadside infrastructure is limited to providing traffic information such as highway congestion data or route guidance to vehicles. Unlike previous approaches, Com2React does not assume automated driving and relies heavily on V2V communication to organise vehicles into clusters, referred to as virtual sub centres (VSC). The role of these VSCs is to coordinate the sharing of information between vehicles, as well as information coming from the roadside infrastructure, that would help increase both safety and the overall traffic efficiency. This work is mainly focused on communication and data aggregation aspects and, to the best of the author's knowledge, no results have been published of the impact this approach has on the overall traffic efficiency.

2.2.2.2 Decentralized Approaches

The AHS concept began to decrease in popularity in the early 2000s as it became clear that deploying such technology in the short term was not feasible. As such, research focus shifted to the lower-levels of vehicle coordination. The resulting work generally

employs a decentralized approach and addresses a specific problem rather than trying to provide a more generic traffic management framework. Most of this work exploits emerging vehicular communications and is sometimes referred to under the umbrella term *cooperative vehicles*. An important observation to make is that focusing on the lower levels resulted in a paradigm shift, with research mainly geared towards increasing safety, while increasing traffic efficiency became a secondary target. This paradigm shift can be easily observed in the IV-related projects that the EU funded over the last years: SAFESPOT, CVIS, CarTalk2000, COOPERS and PREVENT all investigate cooperative vehicle technologies with the main goal of increasing safety. Further works (Khan and Boloni, 2005; Michaud et al., 2006; Bouroche et al., 2006; Tan and Huang, 2006; Sengupta et al., 2007; Stiller et al., 2007; Frese et al., 2007, 2008; Frese and Beyerer, 2011)) address various aspects of vehicle cooperation without directly evaluating their impact on traffic efficiency. More recent work within the framework of the Grand Cooperative Driving Challenge (GCDC) showcased cooperation between vehicles in real-world conditions with automated longitudinal control (Pérez et al., 2013).

Nevertheless, there is still a considerable amount of work that employs a decentralized approach to increasing traffic efficiency. The introduction of ACC systems in production vehicles has raised the question of the impact these systems can have on traffic flow. Answering this question quickly became an active topic of research. An early estimation by Broqua et al. (1991) of the impact of ACC on traffic flow argued that with a penetration rate of 40% ACC-enabled vehicles, set with a target headway of 1s, a 13% increase in throughput can be obtained. However, these results were partially contradicted by later studies (van Arem et al., 1995; Minderhoud and Bovy, 1999) which found that ACC-equipped vehicles had little positive impact on traffic and actually performed worse than exclusively manually driven vehicles under high-density conditions (Marsden et al., 2001).

The conclusion of this early work was that ACC systems, designed primary for driver comfort, required a redesign of the vehicle following strategy that aims at increasing both driver comfort and traffic efficiency. Kesting et al. (2007) used this insight to investigate

how the following strategy of the ACC can be modified so that it has a positive impact on traffic. The goal of this work was to increase the capacity and stability of the flow. More specifically, the authors use a scenario where an on-ramp merging into a one-lane road creates a bottleneck which under high traffic demand causes a traffic jam. The ACC strategy uses a relatively small safe-time headway to increase the capacity as well as an increase in a vehicle's desired acceleration and a reduction in the desired deceleration to increase the stability of the flow (Treiber et al., 2006). Using a microscopic simulation, the authors show that even when a small percentage of vehicles are equipped with ACC this can significantly reduce the traffic jam to the point at which a 30 % ACC-equipped penetration rate completely prevents a traffic jam in the respective scenario. In follow-up work, the system is generalised by extending the ACC with automatic real-time detection of traffic conditions and the ability to adapt the driving strategy to the respective conditions, with similar results (Kesting et al., 2008).

The emergence of CACC systems brings with it additional benefits. Ploeg et al. (2011) show that CACC can be used to reduce the time headways to under 1s while maintaining string stability¹², thus increasing the capacity of the freeway. Furthermore, the potential for vehicle cooperation provided by vehicular communication can be particularly exploited in situations that require the coordination of multiple vehicles such as merging traffic at critical zones. Past work on AHS (Kachroo et al., 1997; Uno and Sakaguchi, 1999; Kato et al., 2002; Xu and Sengupta, 2003; Lu et al., 2004), and more recently Milanés et al. (2011), studied the problem of an automated vehicle merging into a platoon of automated vehicles at an on-ramp. This work focuses on low-level vehicle control issues generally aimed at achieving a safe rather than a highly efficient merging.

Nevertheless, the question of increasing merging efficiency received considerable attention. Xu et al. (2002); Xu and Sengupta (2003) compare an ACC and a CACC system against human drivers in a scenario where on-ramp vehicles merge into an one-lane main road. By measuring the on-ramp queue length, they conclude that ACC systems can improve the efficiency of merging, while extending the ACC system with V2V commu-

¹²string stability represents a guarantee that spacing errors between vehicles are not amplified as they propagate towards the end of the string of vehicles

nication further increases this performance as vehicles on the main road can in advance create gaps for on-ramp vehicles. They show that the ACC system outperforms human drivers.

Similar results were obtained by MacNeill and Miller (2003) using the VISSIM microscopic traffic simulator and a CACC emulation to evaluate a CACC strategy in on-ramp merging and weaving scenarios. Davis (2007) proposed a cooperative merging strategy where cars on the main road slow down to create a gap into which on-ramp cars can merge. The throughput thus obtained was 20% larger than that of human driver simulations for a medium on-ramp demand and only 5% for high on-ramp demand. Wang et al. (2007) used a slightly different strategy in which vehicles on the main road proactively form a merging gap into which on-ramp vehicles can merge by adjusting their speed accordingly. As in the previous case, this approach performs significantly better under medium and light traffic conditions than under heavy traffic conditions.

The cooperative approaches to merging generally assume a one-lane main road and as such cannot take full advantage of the cooperation between vehicles on both main road and on-ramp. Collaboration between vehicles on the main road aimed at increasing traffic efficiency, in this particular case by creating more gaps on the rightmost lane for on-ramp vehicles to merge into, has received little attention. Dao et al. (2007) use V2V communication to implement a distributed lane allocation algorithm for individual vehicles and later on in Dao et al. (2008) apply the same approach to platoons of vehicles. The lane allocation algorithm is implemented in VISSIM and compared for single vehicles and platoons against human drivers with no optimised lane allocation as simulated by VISSIM, by measuring the average travel time. In general, a significant reduction in average travel time can be observed when the lane allocation mechanism is used in low and medium traffic conditions. For high traffic conditions, only the lane allocation for vehicle platoons performs significantly better than human drivers. Unfortunately, it is unclear how much of this improvement is due to platooning and how much due to the lane allocation.

2.2.2.3 Discussion

The reviewed literature was classified based on the architecture adopted by each approach into centralized and decentralized approaches. This related work can be further compared by expanding this classification into a comprehensive taxonomy based on the following criteria:

Structure describes the basic architectural characteristic of each approach and can be either centralized or decentralized.

Target refers to the entity subject to control by the optimisation approach. The approaches reviewed use either individual vehicles or platoons of vehicles as the control target.

Automation refers to the degree of automated vehicle control. This can be either manual, longitudinal control, lateral control, full automation or a mix of them.

Communication indicates the presence and type of communication employed. This can be either one or a combination of V2V and V2I.

Intelligence indicates the presence of optimisation techniques at the roadside and/or at vehicle level.

A classification of the reviewed literature based on the proposed taxonomy is shown in Table 2.1. Several observations can be made from this classification. Foremost, the classification emphasises the influence of PATH on all other centralized approaches. As such, all centralized approaches use platoons as the control target, acting as an interface between the roadside traffic management infrastructure and individual vehicles. There are two basic arguments that motivate the use of platooning. The first is that, for vehicles driving in a platoon, the inter-vehicle distance is reduced and thus the capacity

	Structure	Target	Automation	Communication	Intelligence
PATH	Centralized	Platoon	Automated	V2V, V2I	Roadside, Vehicle
Dolphin	Centralized	Platoon	Automated	V2V, V2I	Roadside, Vehicle
Auto21	Centralized	Platoon	Automated	V2V, V2I	Roadside, Vehicle
Baskar et al.	Centralized	Platoon	Automated	V2V, V2I	Roadside, Vehicle
Com2React	Centralized	Platoon	Manual	V2V, V2I	Roadside, Vehicle
Kerstig et al.	Decentralized	Vehicle	Manual/Longitudinal	V2V	Vehicle
Xu et al.	Decentralized	Vehicle	Manual/Automated	V2V	Vehicle
MacNeille and Miller	Decentralized	Vehicle	Longitudinal	V2V	Vehicle
Davis	Decentralized	Vehicle	Manual/Longitudinal	V2V	Vehicle
Wang et al.	Decentralized	Vehicle	Manual/Longitudinal	V2V	Vehicle
Dao et al.	Decentralized	Platoon, Vehicle	Automated	V2V	Vehicle

Table 2.1: Classification of reviewed literature

of the highway is increased. In practice, however, the benefits of platooning are highly influenced by the nature of the traffic, such as the trip distribution, which determines the size and lifecycle management of platoons. Further work is needed - such as wide-scale simulations with real traffic data - to tune the platoon lifecycle management approaches and determine the actual impact of platooning on highway capacity.

The second motivation for using platooning is that it reduces management complexity for the roadside infrastructure. The traffic management layers compute traffic optimisation policies that are applied to platoons of vehicles instead of having to deal with individual vehicles. However, the platooning interface partially shifts the complexity from the management layers to the coordination layers that are responsible for platoon lifecycle management. The amount and complexity of work dedicated to implementing vehicle platooning, as it can be observed in Horowitz and Varaiya (2000), indicates that this argument is correct. Furthermore, the lack of wide-scale simulations also raises questions about the scalability of the platooning concept, while the need for fully automated driving means that the concept lacks flexibility. This lack of flexibility was largely responsible for the abandonment of AHS by PATH.

More recent work focuses almost exclusively on decentralized approaches and uses individual vehicles as the control target, generally allowing for the coexistence of automated and manually driven vehicles. The focus of these approaches is on specific scenarios such as on-ramp merging or lane allocations that have not been integrated

into a common, decentralized framework. The mix of human drivers and vehicle automation means that decentralized approaches are more flexible and thus more likely to be deployed in the near future. Nevertheless, because there is no integration with roadside infrastructure, the decentralized approaches lack a global view of the road network and, consequently, the optimisation achieved through inter-vehicle collaboration generally has a local rather than system-wide impact. Unfortunately, the nature of freeway traffic is such that local improvements can have a negative impact downstream or upstream. For example, an increase in throughput over a freeway section, achieved as a result of inter-vehicle collaboration, can have a negative impact on a downstream section where an accident has caused a bottleneck. Instead, in such a situation one would aim to reduce the inflow in the congested area (e.g. using VSL as described in Section 2.1.3) by reducing the throughput on the upstream section. The lack of integration with roadside infrastructure means that decentralized approaches can not fully exploit the potential of IVHS.

2.2.2.4 Road transport reliability

In the context of this work, road transport reliability is measured by the consistency of vehicle travel times. On a freeway section, traffic is considered to be reliable if there is little variation between the travel times experienced by a vehicle travelling repeatedly along that freeway section over a longer period of time. This is different from the concept of average travel time, as shown in Fig. 2.5. Here, although the average travel time is constant across the entire year, there is a strong variation in travel times experienced at different times of the year. The three peaks observed in this figure — probably corresponding to spring, summer and winter holidays when traffic demand is high — negatively affect the user's perception of the overall system performance. This makes reliability an important performance criteria for personal road transport.

Furthermore, from a commercial perspective, in the context of a global economy that is becoming increasingly reliant on just-in-time production and distribution systems, reliable road transportation plays a very important role (Joint Transport Research Center

of OECD and ITF, 2010). More specifically, these systems depend on highly accurate total journey time estimations.

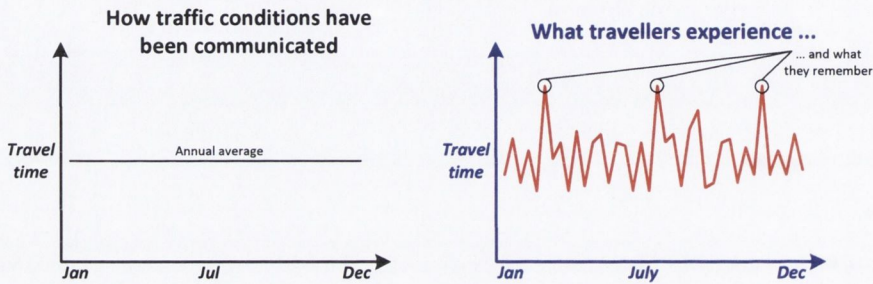


Fig. 2.5: Average travel times vs. Reliability (U.S. Federal Highway Authority, 2006)

Unfortunately, none of the reviewed IVHS approaches directly considers road transport reliability. Whether increasing the capacity through vehicle automation, or merging more efficiently through vehicle cooperation, all the reviewed approaches have the same goal: increase the efficiency of road transport. However, an increase in efficiency reflected through on average shorter travel times does not necessary imply an increase in reliability, as shown previously. Increasing the reliability of traffic should be an important feature of IVHSs.

2.3 Conclusion

The deployment of IVs on the roads is made possible by recent advances in enabling technologies such as sensing, communication and vehicle control. Their potential to optimise traffic stems from two basic properties: faster reaction times and communication capabilities. The faster reaction times allow vehicles to reduce the headway and thus increase the theoretical capacity of the freeway, while communication capabilities enable vehicles to cooperate and thus increase the efficiency of manoeuvres such as merging and overtaking.

In the short term, distributed IVHS approaches that use vehicular communication to promote cooperation between individual vehicles are better suited for an incremental deployment since allow the coexistence of human drivers and (semi-)automated vehicles. To fully exploit the potential of intelligent vehicles, however, on the long term decentralized, self-organising approaches need to be integrated with roadside-based traffic control approaches. Proposed centralized IVHS approaches integrate roadside-based traffic control, yet they lack flexibility and tend to scale poorly due to their choice of platooning as a control abstraction. Furthermore, none of these directly address the crucial aspect of road transport reliability.

In the next chapter a hybrid approach is proposed based on an abstraction layer between the TMS and individual vehicles, called virtual slot, that addresses the limitations of the previously proposed platooning concept. A framework is designed around this abstraction layer and later evaluated under a set of scenarios in which congestion commonly occurs.

Chapter 3

Design

This chapter describes the proposed slot-based IVHS and is organised as follows: the first section discusses the limitations of previous work and determines a set of requirements that an IVHS needs to fulfil. A novel IVHS is consequently proposed, which aims at fulfilling these requirements by employing a hybrid architecture built around the concept of slot-based driving. Section 3.2 then defines the system model within which the slot-based IVHS is described. A detailed description of the traffic management system employed by the proposed IVHS is provided in Section 3.3, followed in Section 3.4 by a description of how individual vehicles perform slot-based driving. In Section 3.5, the slot-based approach is applied to a set of scenarios which typically lead to traffic congestion. Finally, in Section 3.6, the feasibility and limitations of the approach are discussed, followed by an analysis of how the design matches the requirements set at the beginning of the chapter.

3.1 Design Decisions

This section provides a rationale for the design decisions at the core of the IVHS proposed in this work. The limitations of previous work, described in Section 3.1.1, result in a set of requirements for an IVHS, described in Section 3.1.2. These requirements are addressed in Section 3.1.3 through a set of design decisions that lead to a novel, slot-based IVHS.

3.1.1 Limitations of previous work

The analysis of state-of-the-art freeway management techniques performed in Chapter 2 has shown that approaches to traffic optimisation which are exclusively roadside-based are limited by the nature of human driving. Instead, they need to be complemented by IV-based solutions, forming what is commonly referred to as an IVHS. The IVHSs proposed in the past employed either a hierarchical, centralized approach with a TMS coordinating the vehicles, or a fully distributed, decentralized approach, based on collaboration between autonomous vehicles. Each approach has its advantages and disadvantages. Centralized approaches generally do a good job at increasing traffic efficiency. However, due to their high degree of direct control upon vehicles, they tend to scale poorly. Furthermore, by relying on fully automated driving, they lack flexibility. Decentralized approaches, on the other hand, scale well and are generally more flexible. However, they do not fully exploit the potential of IVHS with respect to increasing efficiency, due to their lack of integration with roadside infrastructure. Furthermore, none of IVHSs studied directly addresses the crucial aspect of road transport reliability.

3.1.2 Requirements

By analysing the advantages and disadvantages of previous IVHSs, a set of four basic requirements are identified. Firstly, the main functional requirement of an IVHS is to increase the efficiency of traffic. Secondly, an IVHS should also aim at increasing the reliability of traffic flow. Furthermore, an IVHS must address two non-functional requirements: scalability and flexibility. These four requirements are discussed in the following paragraphs.

Increase Efficiency

The primary goal of optimising freeway traffic is to increase efficiency. Within the scope of this work, the main indicator of efficiency is the average travel time. Average travel time refers to the average of the times required by all vehicles to cross a section of the road subject to evaluation. Under free flow conditions, the average travel times experienced

by drivers are a consequence of each individual's desired speed, vehicle capabilities, road design and regulations governing the respective road network. This however changes when the road network experiences congestion, which has a negative impact by reducing the average speed and thus increasing the average travel times. This work focuses on decreasing the average travel times experienced by drivers during high-demand traffic.

Increase Reliability

There are several ways of measuring the reliability of a road network. For the purpose of this work reliability is quantified on the basis of one fundamental property: the predictability of journey times across the road section subject to evaluation. Under free flow conditions travel times can be accurately predicted due to the fairly constant average speed. This is not the case when the road network is congested, e.g. during rush hours when drivers can experience a high variation in journey times from one day to another. A goal of this work is to increase the reliability of road transport by reducing the variability of travel times under congested traffic conditions.

Scalability

Managing traffic on a road network implies dealing with both a large number of vehicles and a large network of roads spanning potentially hundreds or thousands of kilometres. While an IVHS system might be deployed initially for a small section of the road network and a relatively small number of vehicles, the scalability property means that the system must be able to maintain the same level of performance when the size of the network and number of vehicles under management are increased. When designing an IVHS, the choice of architecture, its degree of control upon vehicles and the way the components of the system communicate can all have a considerable impact on its scalability. In general, an architecture which imposes a tight coupling of its components tends to scale poorly. These aspects need to be carefully addressed at the design of the IVHS.

Flexibility

The review of the state of the art in vehicular technology performed in Section 2.2 indicates that, while recent advances have been made towards deploying fully automated vehicles, this deployment will most likely be done incrementally, over a longer time period possibly spanning several decades, by gradually shifting the driving task from the driver to the vehicle itself. Currently, automated longitudinal control has been deployed in systems such as ACC, which are available in many high-cost vehicles. Automated lateral control, on the other hand, has not seen the same rate of deployment, with comparatively fewer LKA systems available. For an IVHS to be deployable in the near future, it needs to support an incremental deployment of vehicle automation.

The state-of-the-art review identified vehicular communication as one particular area of research and development still lagging behind other vehicular technologies. Similarly, while sensors have been deployed in vehicles for some time, the precision and coverage of currently-deployed sensing technologies, such as absolute and relative positioning, need to be improved to support a higher degree of vehicle automation. An IVHS system needs to consider the current state of the art in vehicular communication and sensing, and function with reduced communication capabilities as well as inaccurate sensors that potentially provide an incomplete map of the surrounding environment.

A further non-technical aspect relates to the likely reluctance of human drivers to definitively hand over the control of the vehicle, as driving is seen by many as a recreational, pleasant activity. An IVHS needs to consider this aspect and allow human drivers to control the vehicle, when doing so does not endanger the safety, efficiency and reliability of traffic.

Based on these, flexibility is defined as the property of an IVHS to be compatible with an incremental use of vehicle automation, communication and sensor capabilities, as well as allowing the use of manual driving when traffic conditions allow it. Furthermore, a necessary characteristic of a flexible IVHS is that increases in vehicle automation, communication and sensor capabilities are reflected in an improvement with respect to the functional properties (efficiency and reliability) of the IVHS.

3.1.3 A Slot-based IVHS

To address the four requirements identified in Section 3.1.2, the IVHS proposed in this work employs a hybrid architecture that combines the advantages of centralized and decentralized approaches. To leverage the superior potential of centralized approaches with respect to increasing efficiency, a global optimisation policy is determined in a hierarchical, centralized fashion by a Traffic Management System (TMS), which encompasses the entire roadside infrastructure. This policy is ultimately communicated to individual vehicles by means of road-side units (RSUs) using V2I communication. To leverage the scalability and flexibility of distributed approaches, the vehicles implement the received policy by coordinating in a distributed fashion. This hybrid architecture is shown in Fig. 3.1.

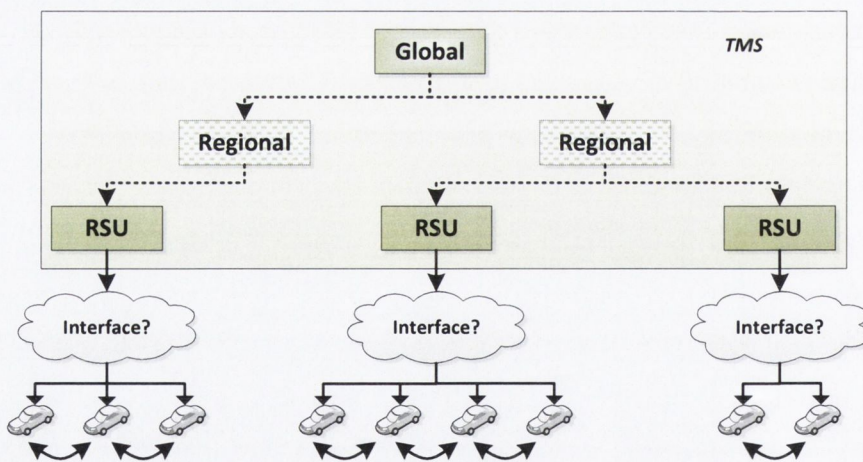


Fig. 3.1: A hybrid IVHS architecture

Interface to Vehicles

Essential to implementing such a hybrid architecture is the interface between the TMS and the vehicles. This interface must enable the TMS to influence the vehicles without directly controlling them. It acts as an indirection layer between the TMS and the vehicles that abstracts the complexity of vehicle dynamics from the TMS and decouples the traffic optimisation task performed by the TMS from the vehicle control task performed by vehicles themselves.

Past research on IVHS has used the concept of vehicle platooning to abstract away the complexity of managing individual vehicles. Unfortunately, platooning enforces a tight coupling between the TMS and individual vehicles. The high degree of direct control the TMS has over individual vehicles (finding a route to the destination, determining target platoons, setting the velocity of the vehicle) effectively means that vehicles are controlled in a centralized fashion, resulting in a purely centralized architecture with its aforementioned drawbacks.

A hybrid architecture requires an indirection layer that allows vehicles a higher degree of freedom with respect to the vehicle control task. This work proposes such an indirection layer, called a *virtual slot* (*vSlot*) or simply a *slot*. A *vSlot* represents an abstraction of a vehicle that moves throughout the freeway network being managed, with dynamics similar to that of a vehicle and a rectangular shape that can encompass a vehicle. The set of all slots is defined by the TMS so that the combined movement of slots in the slot map is both efficient and safe, i.e. collision free. The slots set is then communicated by the TMS through the RSUs to individual vehicles. Because slots mimic the dynamics of a vehicle and are large enough to surround a vehicle, vehicles replicating the movement of these slots also yield an efficient and safe form of travel. Consequently, the role of the TMS is limited to determining a set of slots that optimises the flow of traffic, passing this set to the vehicle and requesting that all vehicles drive within slots. Vehicles, on the other hand, coordinate to find empty slots, as well as to switch slots, without interference from the TMS. From a control theoretic point of view, the employment of this *vSlot* concept effectively makes this IVHS an open-loop control

system, unlike previous centralized IVHSs which used a closed-loop control approach. The lack of a feedback loop means that no communication link needs to be maintained between the TMS and the vehicles. Thus, the vSlot abstracts the complexity of directly controlling a vehicle away from the TMS and decouples the traffic optimisation task from the vehicle control task, leading to an overall hybrid IVHS architecture that scales with respect to the number of vehicles under management. This hybrid nature is best observed in the way the optimisation policy is implemented: vehicles receive information from the TMS but coordinate locally to find a slot. The use of vSlots as the interface between TMS and individual vehicles is shown in Fig. 3.2.

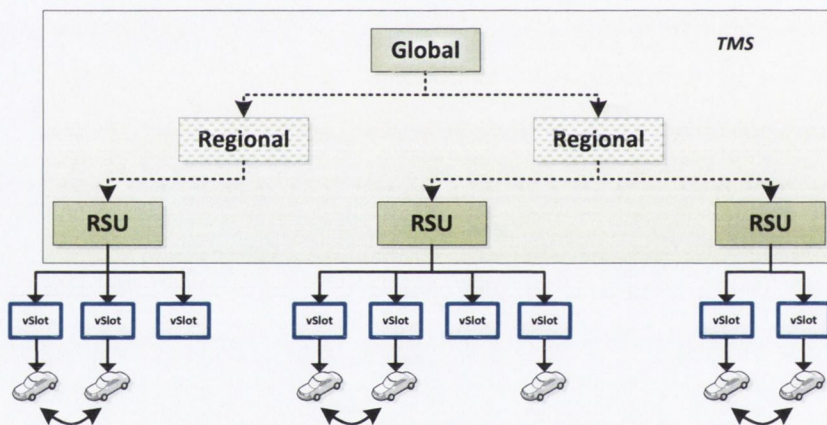


Fig. 3.2: Virtual slots as the interface between RSUs and vehicles

It is important to note at this point that the term “virtual slot” has been used before in the context of IV-based traffic optimisation, albeit to describe relatively different concepts (Morla, 2005; Ravi et al., 2007; Cahill et al., 2008). Most notably, Morla (2005) proposed that congestion-free travel can be achieved by assigning virtual slots to vehicles. In Morla’s vision, a virtual slot represents a time-space corridor negotiated among vehicles without the use of roadside-infrastructure. The slot concept is thus used

as a vehicle's "safe space" which cannot be entered by other vehicles. In situations where slots could overlap, such as at merging zones, the concerned vehicles must negotiate new slots that do not overlap and then move into these new slots. As a result, Morla's vision of a slot is an abstraction for vehicle coordination and is fundamentally different from the concept presented in this thesis of virtual slots as an interface between a TMS and vehicles. Furthermore, the work described by Morla is limited to a position paper and, to the best of the author's knowledge, has never been followed-up by a concrete description and evaluation of the concept.

3.2 System Model

This section presents the system model used to describe the slot-based IVHS throughout the remainder of this thesis. Although some of the concepts introduced in this chapter can be adapted and applied to the entire road network, the scope of this work is limited to optimising traffic on *freeways*. As such, the model of the system under study comprises three basic elements: the freeway, the vehicles and the slot model. As mentioned in the introduction, these models are simplified for the sake of reducing the complexity of the problem, with the purpose of finding an initial solution, in this thesis, that can then be generalized and deployed under real conditions. Note that this model and indeed the remainder of this thesis assumes right-hand traffic.

3.2.1 Freeway Model

A freeway is defined by the Highway Capacity Manual (2012) (HCM) as a divided highway with full control of access, with access limited to ramp locations. Freeways have two or more lanes for the exclusive use of traffic in each direction that are continuously separated. They provide uninterrupted flow and contain no signalized intersections. The freeway model is simplified by assuming a perfectly straight road (i.e. no road curvature) and a zero gradient (i.e. no slope). While the road curvature has generally little impact on freeway traffic, the gradient can cause heavy goods vehicles to slow down. Such situations are not addressed in this thesis but will need to be considered by future work.

Furthermore, because on freeways traffic directions are separated, the freeway model takes only one traffic direction into consideration. While the communication channel can be affected by vehicles travelling on the opposite direction, this is an issue that needs to be addressed by vehicular networking protocols in general and is not addressed in this work.

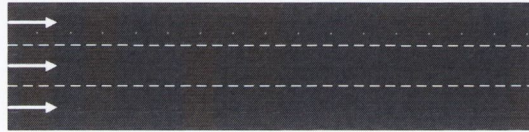


Fig. 3.3: Basic segment

By definition, a freeway can have two or more lanes. Furthermore, the number of lanes is not constant and can change along the length of the freeway. The freeway is therefore split along its length into segments, with each segment sg characterised by its length l and a constant number of freeway lanes lnr . Besides freeway lanes, a freeway segment can also contain an on- or off-ramp (also referred to as exit ramp). As such, three type of segments ty exist in the freeway model: basic, ramp-merge and ramp-diverge.

$$sg = (l, lnr, ty)$$

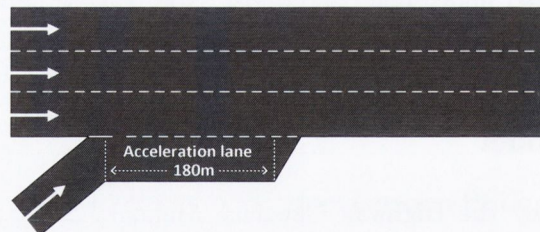


Fig. 3.4: Ramp merge segment

Basic segments are defined as freeway segments with a constant number of lanes, unaffected by merging or diverging traffic. A three-lane basic freeway segment is shown in Fig. 3.3. Ramp-merge segments represent short freeway segments where traffic on the main road merges with traffic coming from an on-ramp. A ramp-merge segment is shown

in Fig. 3.4. The acceleration lane is used by vehicles entering the ramp-merge segment to accelerate from a relatively lower speed to a speed that would allow them to merge into the first lane of the freeway without severely impacting main road traffic. Ideally, the acceleration lane is long enough to allow a vehicle to accelerate from virtual standstill to the speed of vehicles travelling on the first lane. The HCM specifies a default length of 180 meters for acceleration lanes, a value which is also adopted in this model. Similarly, a ramp-diverge segment defines a freeway segment where main road traffic diverges at an off-ramp. A ramp diverge segment example is shown in Fig. 3.5. The default value of 42 meters specified by the HCM for deceleration lanes is also adopted by this model.

Note that this simplistic freeway model does not consider weaving segments, which are segments that consist of an on-ramp followed by an off-ramp. Future work will need to address the specific challenges posed by weaving segments. Nevertheless, no reason stands out as to why such segments would be incompatible with the slot-based approach.

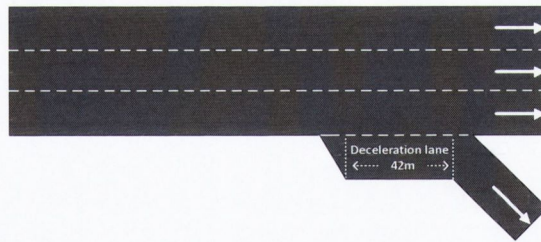


Fig. 3.5: Ramp diverge segment

The study of traffic optimisation techniques requires a model comprising connected segments. A combination of consecutive connected segments is generally called a freeway facility (Highway Capacity Manual, 2012). Fig. 3.6 shows an example of a freeway facility comprising two basic segments, one ramp-diverge and one ramp-merge segment.

3.2.2 Vehicle Model

The vehicle model used in this thesis is a greatly simplified version of an accurate vehicle model, in particular with respect to the vehicle dynamics. A vehicle c is defined as a

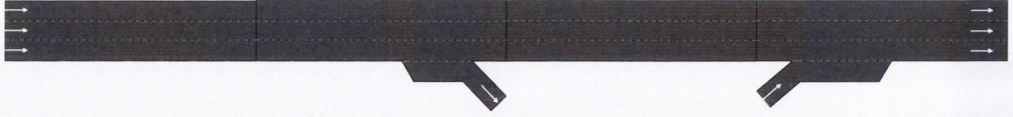


Fig. 3.6: Facility Example

tuple:

$$c = (g_c, s_{max}, \theta_{max}, a_c, b_c)$$

where g_c represents the geometry, s_{max} the top speed and θ_{max} the maximum steering angle of the vehicle. The acceleration profile is simplified, with a_c and b_c representing the uniform acceleration and deceleration of the vehicle respectively. The geometry of the vehicle g_c is further defined by the tuple:

$$g_c = (l_c, w_c)$$

where l_c represents the length of the vehicle and w_c the width.

Vehicles participating in freeway traffic form a heterogeneous set with different geometries, acceleration and deceleration capabilities, top speed, and steering. These differences are obvious when personal vehicles are compared with commercial vehicles such as trucks or buses. To further reduce the complexity of the problem, as justified in Chapter 1, the scope of the thesis is limited to a homogeneous set of personal vehicles with identical capabilities. This means that the width, length, maximum speed and steering, as well as acceleration and deceleration, are the same across all vehicles.

An introduction to the basics of freeway traffic flow is outside the purpose of this thesis. Nevertheless, for completeness, the variables which characterise freeway traffic and are particularly relevant for this thesis are briefly introduced. The following definitions are taken from Highway Capacity Manual (2012):

- *Volume* (V): “the total number of vehicles that pass over a given point or section of a lane or roadway during a given time interval and can be expressed in terms of annual, daily, hourly, or subhourly periods”.
- *Flow rate* (v): “the equivalent hourly rate at which vehicles pass over a given point or section of a lane or roadway during a given time interval of less than 1 h, usually

15 min”.

- *Average travel speed (S)*: the average speed of all vehicles over a certain section of the freeway.
- *Free flow speed (FFS)*: “average speed of vehicles on a given facility, measured under low-volume conditions, when drivers tend to drive at their desired speed”.
- *Density (D)*: “the number of vehicles occupying a given length of a lane or roadway at a particular instant”. It can be calculated using the formula $D = \frac{v}{S}$
- *Freeway capacity*: “the maximum sustained 15-min flow rate, expressed in passenger cars per hour per lane, that can be accommodated by a uniform freeway segment under prevailing traffic and roadway conditions in one direction of flow”
- *Spacing*: “the distance between successive vehicles in a traffic stream, measured from the same point on each vehicle”. In practice, the *average spacing (P)* is used, given by the formula $P = \frac{1}{D}$
- *Headway*: “the time between successive vehicles as they pass a point on a lane or roadway, also measured from the same point on each vehicle”. In practice, the *average headway (H)* is used, and can be calculated using the formula $H = \frac{P}{S}$ and $H = \frac{1}{v}$.

3.2.3 Slot Model

A vSlot is defined by its geometry and behaviour over time. These attributes are set by the TMS, which is responsible for the creation, update and deletion of all slots. Formally, a vSlot is defined by the tuple vs :

$$vs = (g_{vs}, t_{start}, t_{end}, p, s_{base}, B)$$

where g_{vs} represents the geometry of the slot, t_{start} and t_{end} determine the time horizon over which the behaviour B is defined, p represents the position of the slot at t_{start} , and s_{base} represents the base speed. The geometry of a slot is defined as a flat and rectangular surface:

$$g_{vs} = (l_{vs}, w_{vs})$$

where l_{vs} represents the length and w_{vs} the width of the slot. Given that vehicles drive within slots and that the vehicle model defines all vehicles as identical, the same geometry is used for all slots. Furthermore, as the geometry of a slot must be such that it can encompass a vehicle, we need $l_{vs} > l_c$ and $w_{vs} > w_c$.

The position of a slot relies on the lack of curvature in the freeway model and is defined by the tuple:

$$p = (x, ln)$$

where x represents the longitudinal position on the freeway and ln represents the lane. Given that, to prevent accidents, slots must never overlap, each slot can be uniquely identified using its position p at time t_{start} .

The base speed represents the speed of the slot along the freeway and can be superseded by a speed specified as part of the slot behaviour. The slot behaviour is defined over a time horizon, bounded by t_{start} and t_{end} , as a set of waypoints:

$$B = \{w_0, w_1, \dots, w_n\}$$

Each waypoint w_k is defined by the tuple:

$$w_k = (p_k, s_k, ln_k)$$

where p_k represents the position along the freeway at which the waypoint is located, s_k the new speed of the slot and ln_k the new lane of the slot relative to the current lane (1 for left and -1 for right). Thus the set of waypoints define a discrete speed and trajectory profile that a slot follows along the freeway. When driving within a slot, vehicles must replicate the slot behaviour by following these waypoints. The TMS must take into consideration the capabilities of vehicles when defining the behaviour of a slot through waypoints, more specifically the acceleration, deceleration and steering angle.

Overall, this leads to a flexible definition of a slot that can change its speed and lane, similarly to how a vehicle drives on a freeway. The TMS must ensure that slots never overlap. Furthermore, it is the task of the TMS to define slot behaviours so that, when vehicles drive within the respective slots, the efficiency and reliability of traffic are increased. The next section discusses how the TMS is built to achieve these goals.

3.3 The TMS

According to the overall design of the slot-based IVHS and the slot model, the purpose of TMS is to increase traffic efficiency and reliability by creating, updating and deleting the slots in which vehicles drive. In the hybrid architecture employed by the slot-based IVHS, the TMS represents the hierarchical, centralized part, which provides the slots in which distributed, independent vehicles drive. The TMS is designed to allow vehicles to transition from manual driving to slot-based driving when traffic conditions require such a transition. More specifically, under free flow conditions, the travel times are mainly a consequence of the desired speed of each driver. This changes dramatically under congested traffic, when the travel times increase significantly, in the process also becoming less predictable. The TMS treats the task of creating, updating and deleting slots regardless whether vehicles actually drive into these slots or not, and only imposes slot-based driving when free flow conditions are under threat.

3.3.1 Architecture

The TMS has two levels of control, as shown in Fig. 3.7. At the global level, road traffic data gathered by *RSUs* located along the freeway is fused together with data coming from external sources by the *Traffic View* into a global view of the network. This global view is used by the *Global Controller*, which coordinates with regional *Slot Providers*, each responsible for creating, updating and deleting slots on a specific section of the freeway. The freeway is split into sections within which the predicted traffic conditions are as much as possible homogeneous. The following paragraphs provide a brief overview of the functionality of these component.

Each individual *RSU* is responsible for monitoring a different road segment and, combined with the other *RSUs*, cover the entire freeway network managed by the TMS. The monitoring is achieved using currently-deployed sensing technologies described in Section 2.1.1, such as inductive loops, radar, video cameras or in-vehicle GPS systems and smartphones. The raw sensor data needs to be processed into macroscopic traffic flow variables, such as average speed and density, before it can be provided to the Traffic

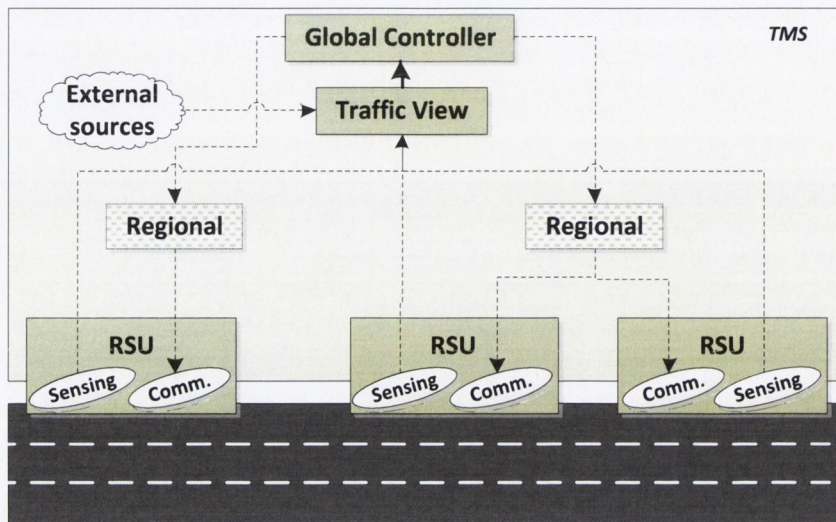


Fig. 3.7: The architecture of the TMS

View component. Besides providing sensor data to the Traffic View, the RSUs are also responsible for broadcasting the slots to the vehicles. This requires V2I communication between the RSUs and the vehicles.

The Traffic View fuses the sensor data received from the RSUs into a global view of the road network. External sources can also be used, such as a weather service or information from road maintenance authorities. The resulting higher-level traffic data contains information such as current and predicted traffic demand, weather conditions, planned road works and road accidents that are currently causing bottlenecks.

The Global Controller uses the current and predicted traffic demand information, provided by the Traffic View, in determining the capacity required from each freeway section. Unlike the traffic demand data provided by the Traffic View, which measures the number of vehicles that pass through a specific section of the road during a period of time, the required capacity is specified in slots rather than vehicles and represents the number of slots that must pass through a section of the road during a period of time.

A simple approach to doing this is to set the required capacity to match the demand on the respective road section. In practice however, the requested capacity will need to be higher than the demand, to account for the fact that traffic is not evenly distributed and as such not all slots can be filled by vehicles.

Congestion can occur if a Slot Provider cannot fulfil the requested capacity. In such cases, the Slot Provider will try to minimize the consequences by providing its maximum capacity. Besides requesting a capacity from each Slot Provider, the Global Controller also harmonises traffic by setting the base speed to be used by all the slots on the freeway. Individual Slot Providers can only change the speed of a slot for a finite length of the freeway, at most equal to the length of their road section. This ensures that when slots transit between slot providers no bottleneck will be caused at the transition point by a mismatch in slot speeds. The significance of this restriction is that Slot Providers cannot fulfil a requested capacity by increasing the speed of the slots on the entire section.

The Slot Providers fulfil the requested capacity by creating, updating or deleting slots on the respective road section, in accordance to changes in the requested capacity as well as the inflow of slots corresponding to the capacity provided by the Slot Provider directly upstream. Each Slot Provider receives a constant demand of slots from the upstream Slot Provider and needs to accommodate this demand. It does so by either adding slots, if the inflow of slots is lower than the requested capacity, or by deleting slots, if the inflow of slots is higher than the requested capacity.

The set of all slots managed by a Slot Provider is called a slot map. The Slot Provider must ensure that the slot behaviours it specifies in the slot map can be replicated by vehicles. For example, the Slot Provider cannot design slot behaviours in which the speed of the slot is higher than the maximum speed of the vehicle. In addition, if semi-autonomous driving is assumed, such as automated longitudinal control and manual lateral control, slot behaviours should account for the slow reaction and lack of precision of human drivers and not impose abrupt lane changes in narrow gaps. Finally, the Slot Provider must adapt its slot map to fulfil the requested capacity while considering the road conditions (e.g. number of lanes available, lane closure, accidents, weather) on its

respective freeway section. The next subsection describes in detail how a Slot Provider fulfils the aforementioned requirements. Given that the focus of this work is on traffic optimisation through slot-based driving, rather than the detection of traffic conditions or vehicular communication, a further description of the other components of the TMS is outside the scope of this work.

3.3.2 Slot Provider

The Slot Provider represents the core optimisation logic of the slot-based IVHS. It is responsible for generating a slot map, in which each slot has a slot behaviour defined for the entire section. A Slot Provider receives a constant flow of slots from the Slot Provider located directly upstream. Since a different slot formation could be used upstream, the Slot Provider must first allocate a subsection of the road for transitioning the slots to its slot map. During this transition phase the Slot Provider must also decide whether to enforce slot-based driving on vehicles or not. This transition is discussed in the following paragraphs, followed by a description of the slot map generation process.

3.3.2.1 Transition

As specified by the overall design of the TMS, during the transition period the behaviour of all slots is fixed to following the base speed. The transition to a slot map must first consider the case where the capacities of the neighbouring Slot Providers differ. If the downstream capacity is higher than the upstream capacity, the transition period requires the addition of slots. This can be achieved by updating the behaviour of arriving slots to create gaps in which new slots are added. For situations in which the downstream capacity is lower than the upstream one, the flow of slots must be decreased by deleting a subset of the incoming slots. If slot-based driving is activated on the upstream section, then the downstream Slot Provider needs to disable slot-based driving before it can delete slots.

After slots are added or deleted, the behaviour of all slots, including the new ones, is transitioned to match the slot map of the current Slot Provider. This is achieved by

matching each incoming slot in the transition phase to its corresponding slot in the slot map, copying its behaviour and prefixing it with a set of waypoints that smoothly adapt the speed of the slot so that at the end of the transition period, the slot is merged into its slot map match.

In parallel to the slot transitioning process, the Slot Provider decides whether to enforce slot-based driving or not. If slot-based driving is enforced when vehicles are not currently driving in slots, the transition period must allow for vehicles to move into slots. Section 3.4 describes how this transition is performed. The decision to enforce slot-based driving depends on whether the known capacity of the freeway section for manual driving can accommodate the current and predicted demand. If this is not the case, then slot-based driving is enforced and the slot map is broadcast to the vehicles at the beginning of the transition phase.

3.3.2.2 Slot Map Generation

The effectiveness of the slot-based approach is a consequence of the combined behaviour of all slots in the slot map. Creating and updating these behaviours needs to also consider the fact that slots are not allowed to overlap at any point in time. If behaviours are individually tailored for each slot in the slot map, the task of generating and updating the slot map becomes almost as complex as that of controlling the vehicles directly. A strategy is needed for the Slot Provider to allow a limited set of slot behaviours to be replicated across all slots safely and efficiently. If a slot map can be fully described by a finite set of slot behaviours, then the size of the Slot Provider's optimisation task is with respect to the size of the road section it manages and the number of vehicles travelling through the respective section, making the Slot Provider scalable.

To allow the replication of slot behaviours, the Slot Provider specifies behaviours starting after the transition period, in which the initial position of slots is arranged into a repeatable pattern, called a *slot formation*. A simple example of such a slot formation can be seen in Fig. 3.8, where slots are simply aligned across three lanes and separated at a constant headway within the lane. This pattern can then be repeated across all

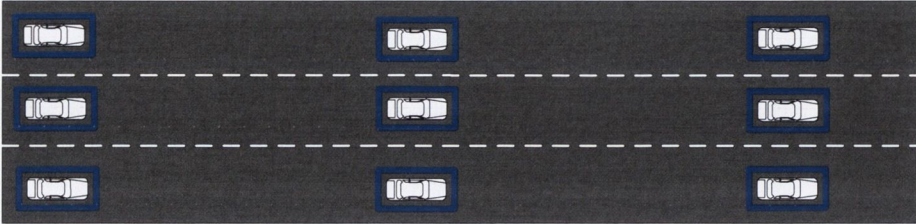


Fig. 3.8: A simple slot formation

slots. Clearly, a slot behaviour comprising a single constant speed that is replicated across all slots in a slot map using such a slot formation guarantees that slots do not collide. Thus, a slot map can be described by the slot formation it uses and a finite set of slot behaviours.

The capacity provided by a slot map is directly proportional to the slot base speed, set by the global controller, and inversely proportional to the average inter-slot headway. Thus, the Slot Provider achieves a required capacity by adjusting the inter-slot headway in its chosen formation and creating a set of slot behaviours that comply with it. For example, assuming a freeway section with a constant number of lanes and the simple formation described above, the Slot Provider could generate a simple behaviour which maintains the base speed and use it for all slots in its slot map. Nevertheless, such a slot map does not suit a wide range of scenarios, in particular scenarios in which a lane of the freeway is blocked, or in which vehicles need to change lanes. Such scenarios require the use of different slot formations and more complex slot behaviours.

When determining the slot map, besides the requested capacity, the Slot Provider must also be aware of current and predicted traffic conditions, as well as the road layout of the respective freeway section. This information can be queried by the Slot Provider from the Traffic View. For the purpose of this thesis, the *Slot Provider* is designed as a rule-based expert system that matches the current predicted traffic conditions as well as the information about the road layout against a set of predefined scenarios. Each predefined scenario has a corresponding heuristic that generates the slot map for the

respective scenario. The slot map is then adapted to fulfil the required capacity by increasing or decreasing the inter-slot headways. Section 3.5 describes a set of heuristics that aim to increase traffic efficiency and reliability by tackling the most prominent congestion scenarios. These heuristics provide a blueprint for how slot maps can be generated for different scenarios and are not intended to cover an exhaustive set of scenarios.

3.4 Slot-based Driving

The loose coupling provided by the slot abstraction relies on the ability of vehicles to perform slot-based driving. This section describes how individual vehicles perform slot-based driving, starting with an overview of the components involved and followed by a detailed description of how vehicles move into slots, drive within slots and coordinate to change slots.

The slot map specified by the Slot Provider is broadcast to the vehicles at the beginning of each transition period, which then fuse this slot information with local information obtained from sensors into a so-called *World Model*, as shown in Fig. 3.9. Based on the information stored in the *World Model*, the *Slot Driving Logic* controls the vehicle, either directly by controlling the steering and throttle of the vehicle, indirectly by assisting the human driver, or through a combination of the two (e.g. controlling the throttle and providing advice with respect to the steering)

The World Model provides the information required by vehicles for slot-based driving. To perform slot-based driving a vehicle needs to be aware of *local information*:

- the vehicle's own speed
- the vehicle's own acceleration
- the vehicle's own position
- the lane in which the vehicle is currently driving
- the presence in the direct vicinity of other objects such as other vehicles

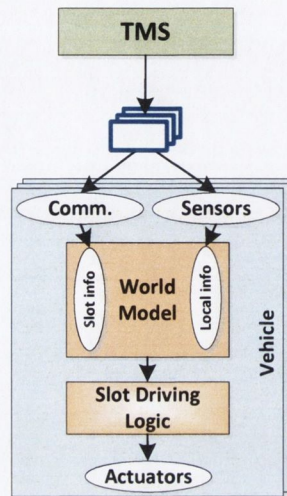


Fig. 3.9: Slot-based driving architecture

as well as *slot information* contained or derived from the slot map:

- the position of slots
- the size of slots
- the behaviour of slots.

Furthermore, vehicles need to be equipped with V2I communication capabilities to receive the slot map. As we will see in Section 3.5, depending on the scenario, vehicles might also need to communicate with each other, in which case V2V communication capabilities are required. For local information, vehicles need to be equipped with sensors. The coverage that these sensors must provide depends on the specifics of the scenario. The World Model fuses and stores slot and local information into a model of the surrounding environment which includes both physical elements, such as neighbouring vehicles, road infrastructure features and other objects, and virtual elements such as the surrounding slots as specified by the slot map. Fusing information from different

sources and storing it into a meaningful model of the surrounding environment is in itself a complex research problem, as seen in Section 2.2, and is outside the scope of this thesis. Chapter 4 describes a simplistic and idealized implementation of this component.

According to the design of the slot-based approach, vehicles are responsible for driving within the specified slots, coordinating in an efficient and safe manner. To achieve higher flexibility, slot-based driving can be employed only when traffic conditions are such that manual driving could lead to congestion. As such, vehicles capable of slot-based driving need to be able to transition from manual to slot-based driving, drive within a slot and coordinate to change slots. These tasks are accomplished by each vehicle's Slot Driving Logic and are described in the following paragraphs.

3.4.1 Moving into a Slot

Before a vehicle can drive within a slot, it needs to move into the slot, driving at the same speed as the slot. This is achieved through a vehicle control algorithm, called the vehicle-to-slot mapping (VTS) algorithm. The VTS algorithm is designed around the idea of performing an on-the-fly transition from manual to slot-based driving, thus increasing the overall flexibility of the slot-based approach. It assumes a starting point, when vehicles are provided with the slot map and notified that they need to move into slots, and an end point, where vehicles are expected to be driving within slots. The transition from manual to slot-based driving is allocated a basic freeway segment, defined by the start and end point of the VTS. The minimum length of this transition segment needs to be computed by each Slot Provider by taking into consideration the average speed of the vehicles, the capabilities of the vehicles specified by the vehicle model, as well as the base speed of the slots and the inter-slot spacing. Using these parameters, a Slot Provider can compute how long it would take a vehicle, with its given acceleration and maximum speed, to move into a slot located directly ahead.

Given that the Slot Provider does not specify the precise slot in which each vehicle should drive, the task of finding an appropriate slot is left to the vehicle itself and is part of the VTS. Since this is performed by each vehicle individually, it is necessary that the

algorithm is designed in a way that ensures both a safe and efficient transition. More specifically, vehicles must not collide with each other and the transition itself must not cause a bottleneck that can lead to congestion. The Slot Provider helps in addressing these issues by generating slot maps where the base speed of the slots is greater than or equal to the current free-flow average speed and the inter-slot headway is at most equal to the average inter-vehicle headway observed on the respective freeway section. To avoid heavy braking by individual vehicles driving at speed considerably higher than the average speed, the slot map needs to be communicated in advance, so that faster vehicles have enough time to reduce their speed without need for sudden braking.

The VTS is designed to use as little vehicular technology as possible, in line with the flexibility requirement specified at the beginning of this chapter. In particular, the use of technologies such as V2V communication, which is yet to be deployed on production vehicles, as well as prohibitively expensive sensors such as the Velodyne 360 Lidar¹, are avoided. As a consequence, the VTS algorithm assumes that vehicles are only able to communicate with the infrastructure (e.g. using third generation cellular communication) and have sensing capabilities limited to the area in front of the vehicle.

The VTS algorithm is presented in Algorithm 3.1. To ensure safety in the absence of V2V communication, the VTS algorithm first prohibits any lane changes. If the traffic load is unevenly distributed across lanes, this approach would lead to one or more lanes being less utilised while the others could experience congestion caused by a larger number of vehicles than that of slots. However, since the VTS algorithm is only applied in situations where high demand could lead to congestion, and assuming that under high-demand conditions the traffic load is evenly distributed across all lanes of the motorway, prohibiting lane changes would not lead to an suboptimal utilisation of the road surface. Since there are no lane changes between the start and end points of the VTS, safety can be ensured by making each vehicle responsible for avoiding a collision with the vehicle directly in front.

Next, vehicles must find an empty slot in their lane. The lack of V2V communication

¹<http://velodynelidar.com/lidar/lidar.aspx>

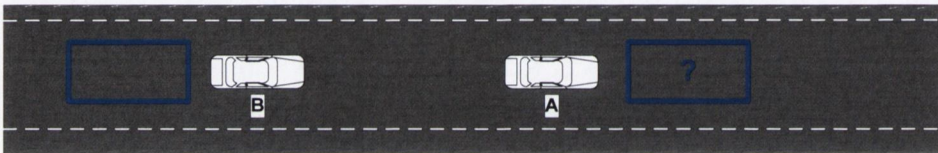
```

begin VTS
  Vehicle receives slot information and the start and end points of the VTS;
  Vehicle maintains the current lane;
  if Vehicle can sense another Vehicle ahead then
    while enough time to move into the next free slot before reaching the end point of VTS
    do
      | control the speed to be at exactly 1 inter-slot headway behind Vehicle in front ;
    end
    if the distance to the Vehicle ahead is more than 1 inter-slot headway then
      | get into the first slot ahead;
    end
  end
  else
    while Vehicle cannot sense a Vehicle ahead and enough time to move into the next free slot before reaching the end point of VTS do
      | accelerate to maximum speed and maintain this speed;
    end
    if Vehicle can sense another Vehicle ahead then
      while enough time to move into the next free slot before reaching the end point of VTS do
        | control the speed to be at exactly 1 inter-slot headway behind Vehicle in front ;
      end
      if the distance to the Vehicle ahead is more than 1 inter-slot headway then
        | get into the first slot ahead;
      end
    end
    else
      | get into the first slot ahead;
    end
  end
end

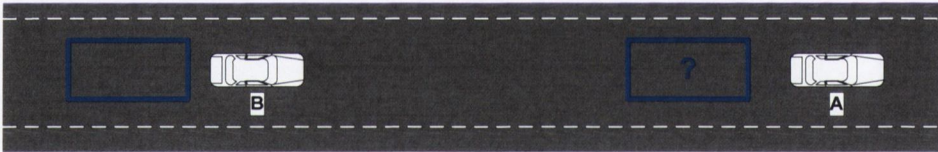
```

Algorithm 3.1: VTS algorithm

means that vehicles cannot communicate when they occupy or desire to occupy a specific slot, and therefore a vehicle cannot know for sure whether a slot is occupied or not. This problem is depicted in Fig. 3.10. Here, vehicle B targets a slot ahead which is currently empty but might be targeted by another vehicle ahead. In Fig. 3.10(a) vehicle A is between vehicle B and the target slot and could be moving forward into the target slot, while in Fig. 3.10(b) vehicle A has passed the target slot but might move backward (brake) into the slot.



(a)



(b)

Fig. 3.10: Is the slot ahead of B free or about to be occupied?

The VTS algorithm solves the empty slot problem by having vehicles follow the first vehicle that they can detect ahead at a distance equal to the inter-slot headway specified by the slot map. Assuming that a vehicle's sensing range is larger than the inter-slot headway², when a vehicle cannot detect a vehicle ahead it means that at least the first slot ahead is free, so the vehicle can target that slot. When a vehicle has moved into the target slot, a vehicle following behind at exactly one inter-slot distance will also be in a slot. In this way, all vehicles can safely move into slots without communicating with

²Radar sensors have a range of 150m (Abou-Jaoude, 2003), which is much more than the 40m representing the observed average inter-vehicle distance under capacity flow (Shladover, 2009)

each other. The only exception to this is the first vehicle in each lane that needs to move into a slot and which might have a vehicle ahead that does not perform the VTS algorithm. In such cases the vehicle will aim to get into an empty slot either in front of or behind it while making sure to maintain a safe gap to the vehicle ahead.

Finally, the VTS algorithm also deals with a potentially uneven distribution of vehicles along the lane by making sure that each vehicle targets the furthestmost empty slot ahead as long as it cannot detect another vehicle in front, and as long as getting into that new target slot can be achieved before the end point of the VTS. This way vehicles driving in close formations, known as platoons or clusters, can be broken down and evenly distributed into slots. The longer the freeway segment dedicated to the VTS transition, the larger the size of a cluster it can break without causing a bottleneck. It is interesting to note that in this regard the VTS algorithm somehow resembles the leaky bucket algorithm (Turner, 1986) used to manage the bandwidth of computer and telecommunication networks by controlling the rate at which packets enter a network. Nevertheless, there are big conceptual differences between vehicles and data packets such as with regard to motion dynamics, as well as the fact that vehicles cannot be dropped from the network as data packets are when the leaky bucket is full. These conceptual differences mean that networking techniques cannot be directly applied to road traffic without careful consideration.

The transition performed by the VTS algorithm depends on the rate at which vehicles arrive to the start point of the VTS, as well as the distribution of speed across these vehicles, both of which are stochastic. This means that the efficiency with which the VTS algorithm performs a transition from manual to slot-based driving needs to be determined empirically using real or simulated traffic data. This evaluation is performed in Chapter 4.

3.4.2 Driving within a Slot

For a vehicle to drive within a slot, it needs to be first positioned within the slot and driving at a speed equal to that of the slot. This is achieved by the VTS algorithm. From

that point onwards, to drive within the current slot the vehicle simply needs to follow the same speed and trajectory as the slot, i.e. the behaviour of the slot. It is the task of the Slot Provider to generate only slot behaviours that can be replicated by the vehicles, thus making the task of driving within a slot as simple as following a predefined speed and trajectory. Further details of how a vehicle follows its slot behaviour are available in Chapter 4.

3.4.3 Changing Slots

Changing slots enables vehicles to enter and leave the freeway and could also be used to prioritise emergency vehicles and optimise traffic in certain scenarios. The challenge posed by changing slots lies in the need to determine a slot's occupancy status before moving into it. There are two approaches to achieving this, both with their advantages and disadvantages.

The first is centralized in nature, where the TMS maintains a view of the occupancy status of all the slots on the motorway. Initially, all slots are empty. When a vehicle moves into a slot as a result of the VTS algorithm, it must inform the TMS that the respective slot is now occupied. When a vehicle wants to change its slot, it enquires about the status of the target slot from the TMS and upon successfully identifying an empty target slot it informs the TMS that its target slot is now occupied and its former slot is now free. The TMS can then accept this change, in which case the vehicle can proceed with the slot changing manoeuvre, or reject it, in which case the vehicle needs to roll back the slot changing manoeuvre. It is necessary to be able to reject changes at the TMS level to address the situation in which two vehicles concurrently intend to move into the same slot. A simple solution to this problem commonly used by databases is for the TMS to ensure that modifying the old and new slot information takes place within an atomic and synchronized transaction (Gray and Reuter, 1992). The centralized approach has the advantage of relying solely on V2I communication, but has the major drawback of having to perform a transaction each time a vehicle attempts to change slots, which scales poorly with respect to the number of vehicles on the road. Nevertheless, this is

diminished by the hierarchical nature of the slot-based TMS, which splits the task of managing traffic on the freeway between multiple Slot Providers, each responsible for a distinct freeway section. This limits the number of vehicles and slots, and thus the number of potential slot-changing requests that a Slot Provider provider must manage. The centralized approach can therefore be used without compromising the scalability of slot-based driving.

The second approach is distributed in nature. Again, slots are initially empty and vehicles move into them using the VTS algorithm. When a vehicle wants to change slots, it coordinates this manoeuvre with neighbouring vehicles by means of V2V communication to ensure that the target slot is empty and that no other vehicle is aiming for the same slot. The distributed approach scales well and fits better within the hybrid architecture of the IVHS system. Nevertheless, it requires the implementation of both group communication and a complex coordination protocol (see Slot and Cahill (2011) and Bouroche et al. (2006) for details on vehicular group communication and vehicular coordination respectively).

Since both approaches are feasible, in this work the centralized approach was used, due to its simplicity and consequently easier implementation within a simulation environment, as described in Chapter 4. Note that the distributed approach is likely to be better suited for a real-world implementation, thanks to the greater scalability and robustness of distributed communication.

The way in which slots are arranged on the freeway has an impact on the complexity of the coordination required between vehicles to change slots. A formation with slots aligned across lanes such as the one in Fig. 3.11(a) is not optimal for vehicles that need to change slots. When parallel vehicles attempt to simultaneously move into distinct empty slots ahead, their trajectories intersect, as exemplified in Fig. 3.11(b). Thus, a complex coordination across several vehicles is required to perform the manoeuvre safely.

A better formation is shown in Fig. 3.12(a). It has the advantage that the trajectories of vehicles moving into distinct empty slots ahead or behind do not intersect, as shown in Fig. 3.12(b). This means that a vehicle can change slots using a simple coordination

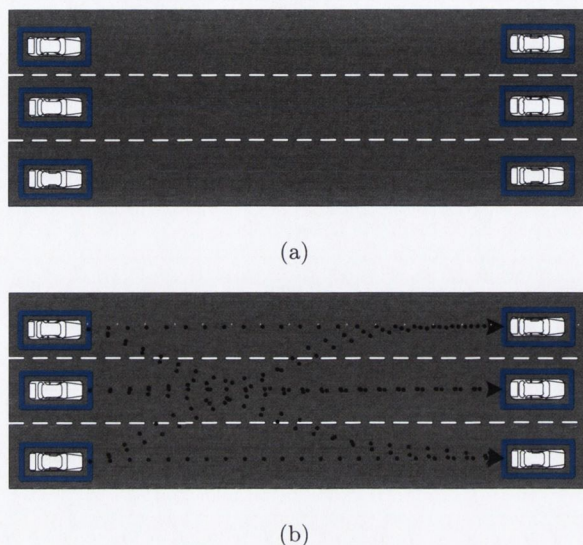


Fig. 3.11: The parallel formation

mechanism to determine that no other vehicle is currently targeting the same empty slot. This formation is called the *chessboard formation*.

3.4.4 Swapping Slots

Swapping slots is a special form of slot changing that is sometimes necessary to allow vehicles that are not located on the rightmost lane to leave the freeway. Additionally, it can also be used to prioritise certain vehicles, such as emergency vehicles. Slot swapping requires vehicles to communicate, agree on a slot swap and perform the manoeuvre in a safe fashion. As in the case of changing slots, the slot formation used affects the complexity of the coordination between vehicles. Consequently, some slot formations are better suited for slot swapping than others.

The chessboard formation is particularly suited to slot swapping. Using the chessboard formation, a slot swapping mechanism can be devised that requires little coordination between vehicles. Vehicles are only allowed to swap slots with vehicles in adjacent lanes, located either in front or behind. This guarantees that upon agreeing on a slot swap, two vehicle performing the swap will not collide with other vehicles changing or

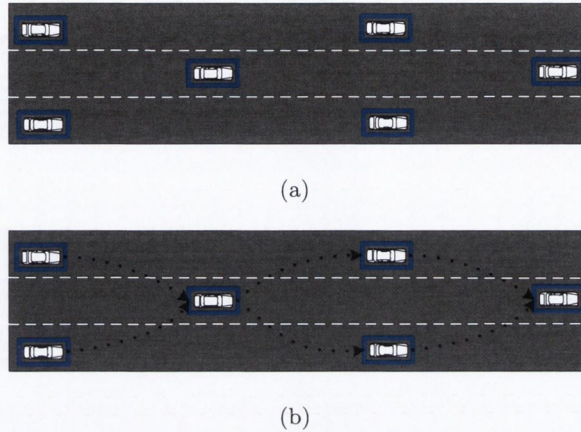


Fig. 3.12: The chessboard formation

swapping slots. The slot swapping mechanism using the chessboard formation is pictured in Figure 3.13. It shows which vehicles are allowed to swap slots. One can observe that, thanks to the chessboard formation, the trajectories of vehicles swapping slots never intersect with the trajectories of other vehicles changing or swapping slots. The chessboard formation thus enables two adjacent vehicles to swap slots without having to coordinate with other vehicles on the freeway.

3.5 Case studies

This section describes a set heuristics used by the Slot Provider to tackle the most prominent causes of congestion. This requires an understanding of what the sources of congestion are. Traffic congestion can therefore be classified based on its occurrence as either recurring or non-recurring. Recurring congestion is typically experienced by drivers as “rush hour” traffic, where high demand leads to volumes beyond the normal capacity of the road. Non-recurring congestion on the other hand is due to special circumstances that can be split into two categories: events that lead to a reduction of the normal road capacity, such as road works, accidents or inclement weather conditions (e.g. snow or fog), and special events that lead to an exceptional increase in demand, such as sport events, concerts or emergency evacuations (e.g. a natural disaster such as

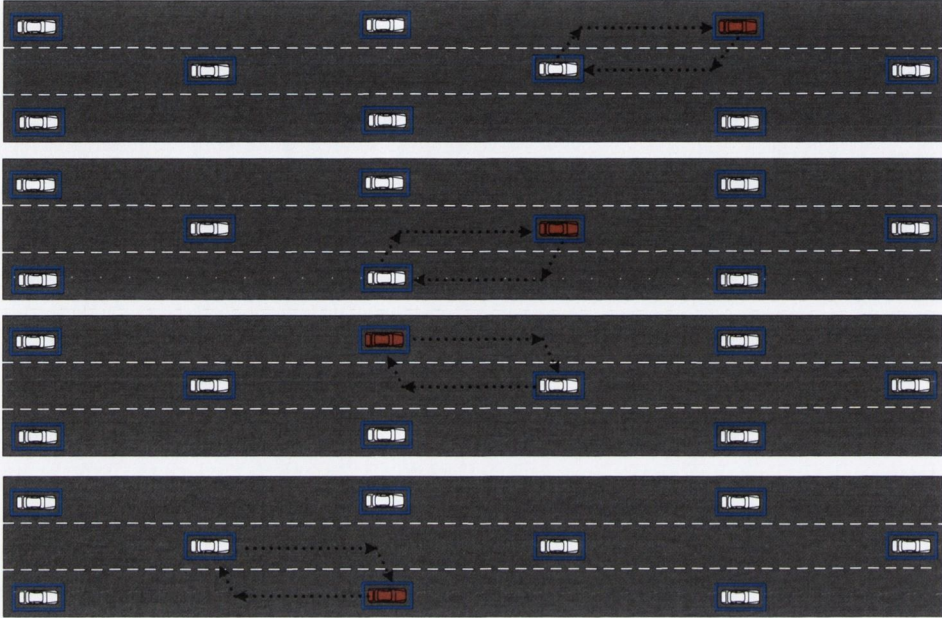


Fig. 3.13: Swapping slots within the chessboard formation

a hurricane). This classification is shown in Fig. 3.14.

According to the US Federal Highway Administration (FHWA), in the US recurring congestion is responsible for approximately half of the total congestion, road accidents for 25%, bad weather for 15%, and finally road works for 10% (Federal Highway Authority, 2012). Note that while poor road design can also be a contributing factor to occurrence of congestion, addressing poor road design is outside the scope of this work. In the following, in line with the rule-based design of the Slot Provider, a heuristic in the form of a slot map, described by a finite number of slot behaviours and a slot formation, is proposed for each of these congestion sources. These solutions are grouped based on the cause of congestion into heuristics that address the reduction of capacity and heuristics that address an increase in demand.

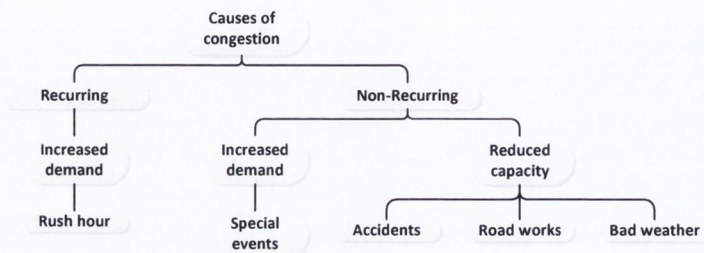


Fig. 3.14: Classification of congestion by its root cause

3.5.1 Addressing Reductions of Freeway Capacity

According to the classification of congestion shown in Fig. 3.14, reductions in traffic capacity can be caused by road works, accidents and inclement weather. Road works and accidents affect the capacity of the road by blocking one or more lanes, thus creating a bottleneck. Inclement weather conditions such as snow, ice and fog cause a reduction in average travel speed and, in extreme cases, can also lead to accidents.

Bottlenecks

First, the case is discussed where a reduction in capacity due to road works or an accident leads to a bottleneck. A scenario is devised using a simple model of a freeway facility composed of two segments of m lanes each, where $m \in \mathbb{N}^*$, and a segment of n lanes, where $n \in \mathbb{N}^*$ and $n < m$. The n -lane segment is located between the two m -lane segments, thus causing a bottleneck. Such bottlenecks can lead to congestion due to a combination of high traffic demand on the respective facility, as well as inefficient merging from m to n lanes performed by human drivers. As such, due to inefficient merging, congestion can arise even if the demand on the section is lower than the capacity of the n lanes section. A slot-based lane merging algorithm (LMA) is proposed, that addresses both of these issues.

The LMA assumes vehicles are already driving in slots. If this is not the case, then

the LMA needs to be preceded by a transition from manual to slot-based driving, which is achieved using the VTS algorithm, described in Section 3.4. For simplicity, during the course of the LMA, vehicles are prohibited from changing slots. This means that while slots do change lanes during the course of the LMA, a slot map which guarantees that slots never collide means that no collisions will take place between vehicles driving within these slots during the merging procedure. The prohibition on changing slots is only applied to the small section of the freeway on which the LMA is applied. The LMA is designed to be used with automated longitudinal control and manual steering assisted by a driver information system, as well as fully automated longitudinal and lateral control.

The core idea of the LMA is to maintain the capacity of the freeway through the bottleneck area by reducing the inter-slot headway. The slot map specified by the LMA comprises a slot formation where slots are aligned across all lanes and a slot behaviour for each slot in the formation, so that, when combined, these behaviours perform the merging of slots from m to n lanes. The vehicles replicate the behaviour of the slot they are currently driving in, thus performing a coordinated merging procedure that, as shown later in Chapter 4, is more efficient than the competitive merging procedure performed by human drivers. This hypothesis is tested in Chapter 4.

The merging procedure specified by the LMA is described on a 3-to-2 lane merging scenario, as this is arguably the most common case. A set of predefined slot behaviours are specified which together perform an efficient merging procedure. The steps involved in this procedure are shown in Fig. 3.15. The behaviours are classified in two main categories: A and B. Category A slots are followed, within the same lane, by category B slots. Each of the two categories has three subcategories, numbered 1, 2 and 3 according to the lane in which the slots are initially moving. This results in six slot behaviours that are replicated across subsequent slots, as shown in Fig. 3.15. At the starting point of the LMA, the slots are aligned and have a base speed equal or higher than the average travel speed observed at that time downstream of the bottleneck. As a result of the transition to slot-based driving, all vehicles at the starting point are driving within these slots,

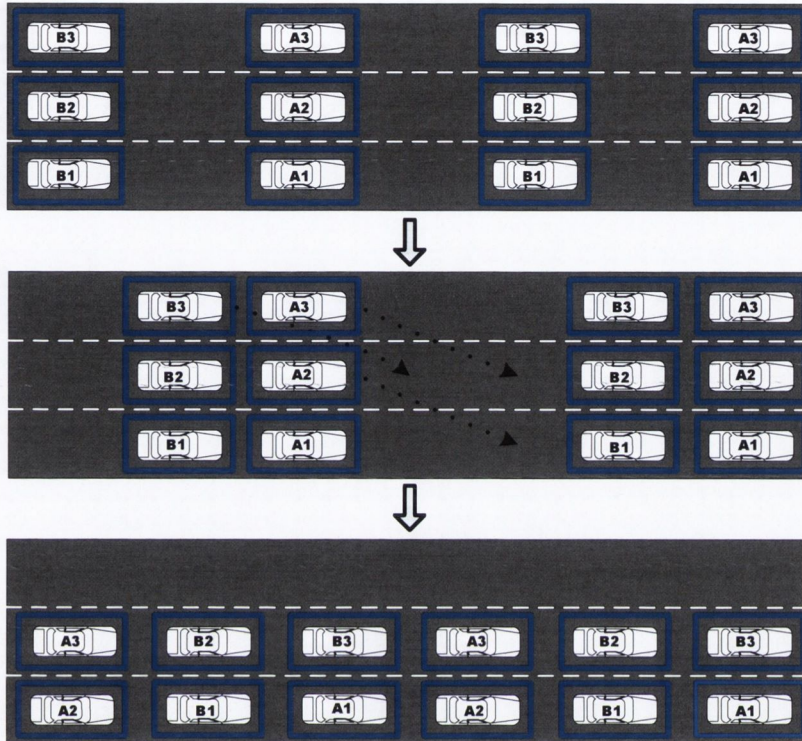


Fig. 3.15: Visualisation of the lane merging algorithm

although not all slots have to be filled by vehicles. From this position the B-slots reduce the headway to A-slots, leaving a larger empty space between these newly formed pairs (Phase 1). This space is then used by A2 slots to move to lane 1 (Phase 2). Similarly, A3 and B3 slots use the remaining space to change to lane 2 (Phase 3), leaving no slot and thus no car on lane 3. The combined behaviour of these slots is a merging procedure during which the speed of all slots never goes below their initial speed, meaning that the LMA achieves the same capacity on the two-lane bottleneck section as on the three-lane highway. Since the speed of the slots at the starting point of the LMA is set to be equal or greater than the average travel speed observed at that point in time, the LMA effectively prevents a reduction in capacity.

The LMA defines start and end points for each of the three phases, which are de-

terminated based on the specifics of the slot base speed specified by the Global Controller and the capabilities of the vehicles specified by the vehicle model, in particular the acceleration, maximum speed and maximum steering angle. The pseudocode for LMA is shown in Algorithm 3.2 from the perspective of a vehicle following the behaviour of its slot.

```

begin LMA
  while Vehicle has not reached the end point of the LMA do
    if Vehicle is within Phase 1 then
      if Vehicle is in a slot of category B then
        Follow the speed of the slot to reduce the gap to the slot ahead to a predefined headway;
      end
    else
      Maintain speed ;
    end
  end
  else if Vehicle is within Phase 2 then
    if Vehicle is in a slot of category A2 then
      Follow the speed and trajectory of the slot and change to lane 1;
    end
  end
  else if Vehicle is within Phase 3 then
    if Vehicle is in a slot of subcategory 3 then
      Follow the speed and trajectory of the slot and change to lane 2;
    end
  end
end
end
end

```

Algorithm 3.2: Lane merging algorithm

The six slot behaviours specified by the LMA are defined by the following sets of waypoints in line with the slot model defined in 3.2:

- A1={ } (maintain the base speed of the slot and don't change lanes)

- $B1 = \{\{phase1_{start}, s_{base} + q, 0\}, \{phase1_{end}, s_{base}, 0\}\}$
- $A2 = \{\{phase2_{start}, s_{base} + r, -1\}, \{phase2_{end}, s_{base}, 0\}\}$
- $B2 = \{\{phase1_{start}, s_{base} + q, 0\}, \{phase1_{end}, s_{base}, 0\}\}$
- $A3 = \{\{phase3_{start}, s_{base} + r, -1\}, \{phase3_{end}, s_{base}, 0\}\}$
- $B3 = \{\{phase1_{start}, s_{base} + q, 0\}, \{phase1_{end}, s_{base}, 0\}, \{phase3_{start}, s_{base} + r, -1\}, \{phase3_{end}, s_{base}, 0\}\}$

where q causes a small increase in speed that allows a slot to reduce the gap to the slot ahead. Similarly, r causes a small increase in speed needed when a slot changes lanes. Notice how in fact the slot behaviours B1 and B2 are identical, meaning that there are only 5 distinct behaviours in the slot map. The two behaviours were presented distinctly to better describe the way in which the LMA works.

The same idea behind the 3-to-2 LMA can be applied to any variation of m to n merging. Vehicles on the lanes that are not blocked downstream make space for vehicles on lanes that are blocked. However, the feasibility of LMA decreases proportionally with the value of $\frac{m}{n}$. More specifically, an m to n merging requires the capacity of the remaining n lanes to increase by a factor of $(\frac{m}{n} - 1)$. The LMA relies on the ability of vehicles to perform automated longitudinal control to achieve this capacity increase. Shladover (2009) uses measured freeway data to show that, for an observed maximum capacity of 2200 vehicles per lane per hour driving at 100 km/h, the average inter-vehicle headway is 1.63s, or 40.5 m, which represents 8 vehicle lengths. Assuming a theoretical best-case scenario where automated longitudinal control would allow vehicles to drive with virtually no inter-vehicle gaps, the capacity increase provided by vehicle automation compared to manual driving would be approximately 9-fold. This means that, for LMA, the theoretical limit for $\frac{m}{n}$ is 10.

In practice, the limit for $\frac{m}{n}$ is likely to be considerably smaller than 10, due to the larger than zero gaps required by automated longitudinal control, but also due to the gaps required by vehicles to change lanes during the respective LMA phases. For

manual steering, which is supported by the LMA, these gaps are significantly larger and subject to the human perception of the critical gaps, which has been subject to extensive research over the last decades due to its practical applications in the design of microscopic traffic simulators. While opinions vary on the distribution of the critical gap, field measurements have shown that, when the relative speed is close to zero, drivers are more likely to accept very short gaps, sometimes no more than 1 to 2 meters to the front vehicle (Hidas, 2002), which for vehicles driving at 100km/h corresponds to a headway of less than 0.1s. While this value can be seen as rather extreme, it shows that the impact of using manual steering in LMA does not drastically limit the capacity. In conclusion, the applicability of LMA depends on the values for m and n , as well as the vehicular technology available.

The LMA is preemptive in nature, to be used before congestion arises. While this works well for road works, which are typically known in advance, in the case of accidents sometimes congestion cannot be prevented. In such cases the LMA can be adapted to include a technique currently used on freeways. When an accident takes place, traffic operators often use variable speed limits to reduce the speed of the vehicles upstream of the accident, thus allowing the accident area to clear of vehicles. Similarly, the LMA can be applied upstream of the accident area with an initially lower slot base speed for upstream slots. Once the accident area has cleared, the base speed of the upstream slots can be increased. Increasing the speed of the slots beyond the free flow speed would allow to increase the throughput and dissipate the potential congestion caused by the initial slower speeds. Nevertheless, the initial slower speed can cause disruptions on the upstream sections, which is a limitation of this approach.

Inclement Weather

Inclement weather conditions such as fog, ice and snow cause vehicles to slow down, thus reducing the capacity of the road. While in some situations such as icy roads this might be inevitable due to the severe speed limitations, in other situations a combination of sensors and actuators can be used to assist the driver and maintain the same average

speed. For example, applying vehicular technologies used by slot-based driving such as sensors that detect neighbouring vehicles and a heads up display that can project this information to the driver can combat the poor visibility effects of fog. Dealing with inclement weather conditions is outside the scope of this work.

3.5.2 Demand Increase

During rush hours or special events, congestion typically occurs due to an increase in demand leading to densities larger than the critical density of the respective freeway section. At such high densities, the effects of the slow reaction times and inability to accurately estimate distances and speed as well as to properly coordinate, characteristic of human drivers Vanderbilt (2009), are amplified. Human drivers are more prone to brake unexpectedly under high density conditions, causing traffic shockwaves that propagate upstream and eventually can lead to stop-and-go traffic. In such situations, slot-based driving can be used to prevent sudden braking and shockwaves. Clearly, a slot map in which all slots have a constant speed is string stable, and, as long as a vehicle performing slot-based driving does not cause another vehicle to brake, so is the traffic composed of the vehicles driving within these slots.

While the LMA enforces slot-based driving for a relatively short section of the road, during high-demand traffic slot-based driving will need to be enforced over considerably larger freeway sections. This raises the problem of vehicles joining the freeway at on-ramps and in particular leaving the freeway at exit ramps. During the entire duration of the LMA vehicles maintain their slot and follow its behaviour, possibly changing lanes in the process. For larger freeway sections, a different approach is proposed, with a single slot behaviour describing a constant speed for all slots in the slot map. Slots maintain a predefined speed and lane and vehicles change slots according to their necessities, such as when aiming to change lanes in order to exit the freeway. The rules governing the changing of slots must ensure that a vehicle changing his slot does not cause another vehicle to have to brake to avoid a collision, thus ensuring that no shockwaves can occur.

Since slot-based driving is likely to be imposed for a longer part of the freeway during

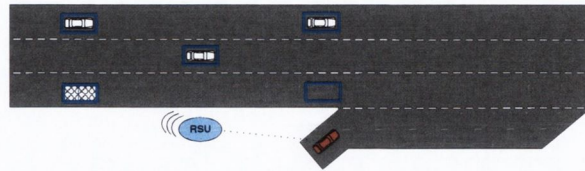
high-demand traffic, vehicles will need to be able to change and swap slots, particularly to be able to enter and leave the freeway. In such scenarios, to simplify the coordination required for these manoeuvres, the slot map uses the chessboard formation, introduced in Section 3.4. To achieve such a formation, slots on even lanes (lanes 2, 4, 6 etc.) need to be generated with half-headway offset relative to the slots on odd lanes (lanes 1, 3, 5 etc.). The behaviour of slots in this slot map is set to maintain their base speed. The following paragraphs discuss a scenario where vehicles enter the freeway at an on-ramp during high-demand traffic. This is then extended to include vehicles exiting the freeway at an off-ramp.

On-ramp Merging Algorithm

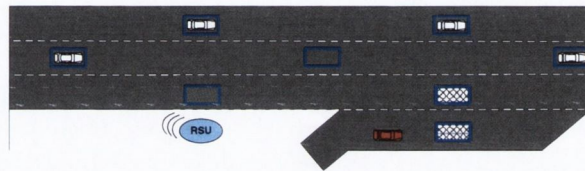
From a slot-based driving perspective, merging vehicles at on-ramps in an efficient way requires on-ramp vehicles to move into slots without disrupting the main road traffic. Such slots must be empty and located on the first lane of the motorway. Furthermore, at the time of selection, the distance between the chosen slot and the actual point where the roads merge must be large enough to allow an on-ramp vehicle to move into the slot, while driving at a speed equal to that of the slot, before the end of the acceleration section is reached. This can be calculated based on the speed of the slot, the maximum acceleration and the speed specified by the vehicle model and the length of the acceleration lane, which at a minimum must allow a vehicle to accelerate from virtual standstill to the speed of the slot.

Determining such a slot can be achieved using V2I communication with a RSU located near the on-ramp and acting as a proxy between the vehicles on the main road and the on-ramp vehicles. The RSU is able to sense the location of vehicles on the main road and can coordinate with them by using V2I communication and a copy of the slot map to determine suitable slots. When such slots are found, the RSU lets the vehicles on the main road know that the respective slots are occupied, effectively blocking any attempt of any vehicle on the main road to move into that slot. On-ramp vehicles communicate with the RSU using V2I communication and request a slot. Once the RSU

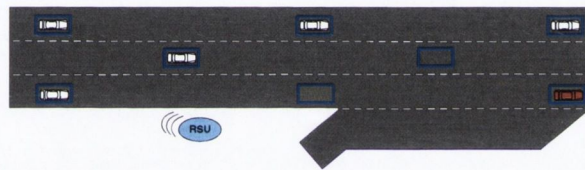
finds an available slot, it communicates the slot information to the on-ramp vehicle.



(a) Vehicle asks the RSU for a free slot



(b) Vehicle makes a copy of the slot and targets it



(c) Vehicles moves into the original slot

Fig. 3.16: On-ramp merging

After the on-ramp vehicle has received a suitable slot, the moving into the slot phase commences. For this purpose, the vehicle targets a copy of the slot which moves in parallel to the original target slot along the acceleration lane. The vehicle then moves into the on-ramp slot, using a mechanism described in Section 3.4. Just before reaching the merging point, the vehicle moves into the original slot by performing a lane change towards the first lane of the main road, finishing the on-ramp merging procedure. This process is visualised in Fig. 3.16.

This simple algorithm merges on-ramp vehicles into free slots on the first lane of the motorway. However, in doing so, it does not take advantage of free slots on the other lanes of the motorway. This can be achieved with a basic optimisation, based on the ability of vehicles to change slots, which is described in detail in Section 3.4. Vehicles driving in slots are aware that they are approaching an on-ramp (e.g., using information

available in digital road maps or communicated by the RSUs) and as such target free slots to the left front of the current slot, if such slots are available. This is shown in Fig. 3.17. The result of this optimisation is an increase in the number of free slots on the first lane which enables a higher throughput of vehicles on the on-ramp.

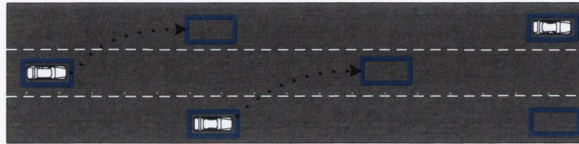


Fig. 3.17: Moving to free slots to the left-front of the current slot

The on-ramp merging algorithm resembles ramp metering in the way it forces vehicles to wait at the on-ramp until the main road traffic conditions are deemed appropriate. Although the on-ramp merging algorithm can achieve a high throughput, as shown later in Chapter 4, there is still the theoretical possibility that a queue of vehicles will form on the on-ramp which is large enough to spill back onto the adjacent road network. While unlikely, in such situations a trade-off exists between maintaining slot-based driving and thus free flow on the freeway while allowing traffic to spill back into the adjacent network or disabling slot-based driving on the freeway and allowing vehicles to freely merge into traffic in the hope that a spill back can be avoided, which will likely lead to congestion on the freeway.

Exit Ramps

The chessboard formation is also useful when vehicles need to exit the freeway. In order to do so, a vehicle which is not located on the rightmost lane must move from its current lane onto the rightmost lane. When driving within slots arranged in the chessboard formation, this means that the exiting vehicle will need to move into a slot on the rightmost lane before the targeted exit ramp is reached, assuming it has not already got one. The chessboard formation is designed to facilitate a vehicle moving “diagonally” from its current slot into an adjacent slot. In some cases it could be that none of the adjacent slots are free. In such cases, vehicles need to coordinate to swap slots, so that

vehicles that travel further and are located to the right front or right back of a vehicle looking to exit the highway earlier will swap slots with it. This is shown in Fig. 3.18 and uses the slot swapping mechanism described in Section 3.4.

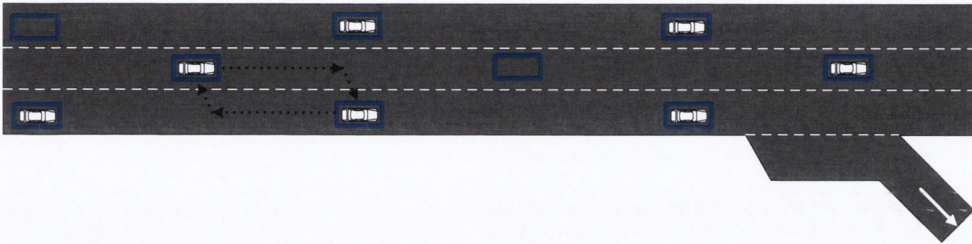
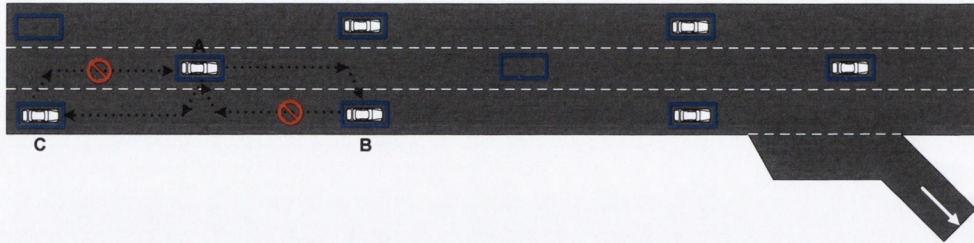


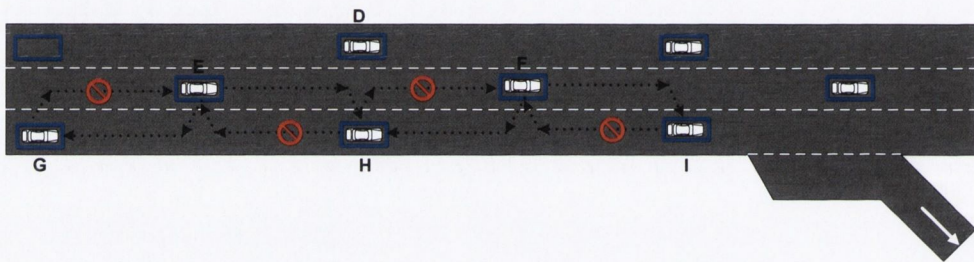
Fig. 3.18: Swapping slots with a vehicle targeting another off-ramp

Nevertheless, there might be cases where a simple slot swap is not possible. Fig. 3.19(a) illustrates such a situation, in which vehicle A, aiming to leave the freeway at the next exit ramp, cannot swap slots with vehicles B and C as they also target the next exit ramp. In such cases, vehicles B and C reject a swap and consequently are not allowed to leave their slots. Knowing this, vehicle A can merge between vehicles B and C if the inter-slot gap is large enough. When this is not the case, as well as in odd pathological cases such as the one in Fig. 3.19(b), in which vehicle D is blocked by E and F which in turn are blocked by vehicles G, H and I from exiting the freeway, then vehicles A, D and potentially E and F might not be able to leave the freeway at their desired exit point. After the exit point, the previously occupied slots on the rightmost lane are now free and the vehicles which missed their exit can move into these slots and take the next exit. Nevertheless, this is a clear limitation of the chessboard formation, as in some cases not being able to take the desired off-ramp exit is not acceptable.

Other slot formations can be used to address the aforementioned limitation. For example, the chessboard formation can be modified to ensure that vehicles D, E and F in the previous example can leave the freeway. This is illustrated Fig. 3.20, in which a formation similar to the chessboard formation is applied to a three lane freeway. However, in this modified formation every second rightmost slot is removed. Furthermore, this



(a) Vehicle A is blocked by vehicles B and C



(b) Vehicle D is blocked by vehicles E and F, which are blocked by vehicles G,H and I

Fig. 3.19: Vehicles cannot exit the freeway

formation imposes a minimal inter-slot gap which allows a vehicle to merge in-between slots, thus ensuring that vehicles D, E and F can leave the freeway at the next exit ramp by merging into the rightmost lane. Nevertheless, using such a formation with its inter-slot space restrictions and reduced number of slots relative to the chessboard formation has a negative impact on the overall capacity, which is reduced by at least 16%. This leads to a trade-off between being able to offer stronger guarantees with respect to vehicles exiting the freeway at their desired off-ramp and maximizing capacity. A decision for one or the other can be taken based on the required capacity as well as upon a careful analysis of the probability of pathological cases occurring. Further coordination between vehicles along the freeway can be used to minimise this probability, such as avoiding to swap slots with a vehicle on the left lane if the left vehicle targets an exit located further away than the one targeted by the right vehicle, or by actively promoting vehicle swaps when the right vehicle is targeting an exit ramp located before the exit ramp of the left

vehicle.

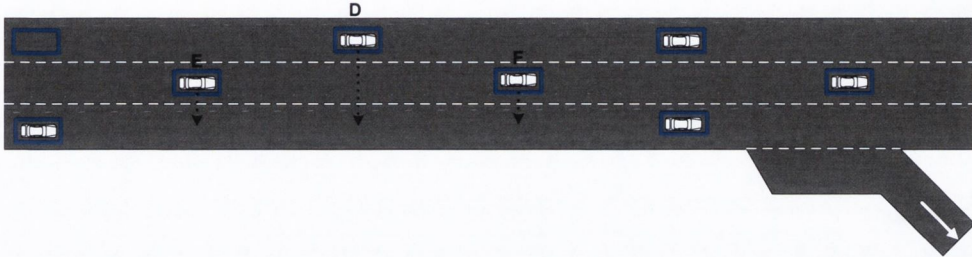


Fig. 3.20: Improved exiting probability through reduced capacity

3.6 Discussion

This chapter introduced a novel slot-based approach to optimising freeway traffic using intelligent vehicles. The IVHS designed relies on the ability of vehicles to perform slot-based driving. This section investigates the feasibility of slot-based driving with respect to the state of the art in vehicular technology as described in Chapter 2. This is then followed by an analysis of how the IVHS requirements set at the beginning of this chapter are fulfilled by the slot-based approach, as well as a discussion of the limitations of the proposed designed.

3.6.1 Feasibility of the Slot-based Approach

For a vehicle to be able to perform slot-based driving, it has to rely on information about the surrounding environment obtained from sensors augmented with the slot map communicated by the TMS. This information is processed by the vehicle's Slot Driving Logic, which then controls the vehicle either directly or indirectly by assisting the human driver. A feasibility analysis needs to consider three areas of vehicular technology: sensing, communication and vehicle control.

Sensing

A vehicle's sensors need to provide the *World Model* with information about the vehicle's own speed, acceleration, position and the presence of other objects on the road. Sensors measuring the vehicle's speed and acceleration are already available on virtually all modern vehicles and are known to be accurate (Bishop, 2005).

For positioning, a large number of vehicles are already equipped with GPS receivers, embedded either in the vehicle itself or as part of a third-party navigation device or smartphone. The poor accuracy of standard GPS receivers (Hegarty and Chatre (2008) measured it at 13m horizontal accuracy) can be partially compensated by increasing the length of the slots to ensure that, for example, even if the positioning measurements of a vehicle driving within a slot are inaccurate the vehicle is still positioned within the slot. Nevertheless, the width of a slot cannot be increased beyond the width of the lane (3.5m for US freeways) which means that the vehicle must use an additional sensor to determine its current lane. Stereo vision cameras have been used for this purpose on high-cost personal vehicles and trucks, generally as part of a LKA system. Increasing the length of the slots would, however, negatively impact the overall performance of the IVHS as it will increase the average inter-vehicle headway which directly impacts the theoretical capacity. To increase this performance, vehicles would need to be equipped with more precise positioning technologies. Fortunately, GNSS technologies such as differential GPS and RTK-GPS have already been tested in vehicles and can provide sub-meter accuracy. A combination of very accurate positioning and digital maps can also be used to detect the current lane such as stereo vision cameras.

Slot-based driving also requires a vehicle to be able to sense its local environment. In particular, the VTS algorithm requires that vehicles are able to sense the space ahead of them beyond the maximal inter-slot distance, meaning that vehicles can always sense if the space to the first slot ahead contains a vehicle or not. This can be achieved using sensors such as lidar or radar which have already been deployed in systems such as ACC. More advanced sensors such as the Velodyne 360 Lidar can extend the sensing range but are generally more useful in situations in which vehicles are fully autonomous such as

Google's Self-Driving Car.

Communication

The slot information is provided by the TMS and needs to be communicated to the vehicle. Thus, vehicles need to be equipped with V2I communication capabilities. In most cases, the slot map can be provided to vehicles in advance, before the start point of the VTS. In these cases the V2I communication does not need to be real time and can be implemented using already widely deployed 3G networks, as well as 4G technologies such as WiMAX³ and LTE⁴ that are currently being deployed on a wide scale. Nevertheless, in some cases, e.g. when a vehicle's malfunction is putting the safety of other vehicles is at risk, the communication needs to take place in real time. Furthermore, changing slots requires vehicles to be equipped with real-time V2V communication capabilities to be able to coordinate. As seen in Chapter 2, a considerable research effort is currently targeting the development of real-time vehicular communication but such technology is yet to be deployed in production vehicles. For this reason the design of the slot-based approach employs an incremental approach with respect to its use of vehicular communication and, when possible (e.g. VTS and LMA algorithms), attempts to limit the use of vehicular communication.

Vehicle Control

The *Slot Driving Logic* processes the information in the *World Model* and controls the vehicle accordingly. This control can be direct, meaning that the vehicle drives autonomously; indirect, by assisting the driver to perform the driving task; or a combination of the two where part of the driving task is performed by the vehicle itself and part by the driver.

Fully autonomous driving is the most efficient form as it addresses the issue of slow reaction times of human drivers. It has already been implemented on research vehicles but is yet to be available in production vehicles. While the technical feasibility has been

³<http://www.wimaxforum.org/>

⁴<http://www.3gpp.org/Technologies/Keywords-Acronyms/LTE>

demonstrated by projects such as Google's Self-Driving Car, where autonomous vehicles have driven in total more than 300.000 miles, legal and moral issues (e.g. the issue of liability in case of an accident or dealing with situations where a possibly fatal accident is unavoidable) are yet to be addressed.

Indirect vehicle control can be used for slot-based driving, for example by assisting the driver via a heads-up display (HUD) that indicates both the slot in which the vehicle needs to drive, as well as the current and future speed and trajectory. HUD systems have already been deployed successfully on some high-cost vehicles (Brandon, 2012). While indirect vehicle control would increase the cooperation between vehicles, it cannot take full advantage of the potential of slot-based driving, as the slow reaction times of human drivers would need to be accommodated in the inter-slot distance, which would affect the theoretical capacity of the road.

Finally, mixing direct and indirect control would result in a form of semi-autonomous driving, where part of the driving task is performed by the vehicle itself and part by the human driver. For example, automated longitudinal control that can be overridden by the human driver - much like presently deployed ACC systems - could be combined with manual lateral control. The lateral control would be assisted by a HUD system that indicates the current and future trajectory that the vehicle must follow.

In conclusion, all these forms of vehicle control are feasible and have been made possible by recent advances in vehicle automation. Incrementally increasing the degree of automation from indirect control to full automation would result in improved traffic efficiency. While semi-autonomous driving is likelier to be deployed in the short-term to medium-term and can yield a significant increase in traffic efficiency, for the long-term fully autonomous driving will be needed to achieve the full potential of slot-based driving.

3.6.2 Matching the Requirements

At the beginning of this chapter four basic requirements were set for the slot-based IVHS. The following paragraphs analyse how the design of the IVHS matches these

requirements.

Scalability

The slot abstraction provides a loose-coupling between the TMS and individual vehicles, which is critical to the scalability of the IVHS. The decoupling of the traffic optimisation task from vehicle control, combined with the ability of Slot Providers to design a slot maps with a limited number of slot behaviours replicated among all slots, allows the TMS to scale with respect to the number of vehicles and the size of the network under management. Furthermore, slot-based driving solely depends on sensing the local environment and, if necessary, negotiating with other vehicles in the neighbourhood. This means that the complexity of slot-based driving does not depends on the size of the road network or the number of vehicles. Overall, this means that the slot-based IVHS is scalable, which can also be observed from the experiments performed in Chapter 4.

Flexibility

Flexibility was defined as the property of an IVHS to be compatible with an incremental use of vehicle automation, communication and sensor capabilities as well as allow manual driving as long as it does not threaten the free flow of traffic. These requirements were considered in the overall design of the IVHS, and in particular in the slot map design in Section 3.5 as well as in the specifics of slot-based driving in Section 3.4. Scenarios such as using the LMA do not require full vehicle automation or V2V communication and work with limited and inaccurate sensing capabilities, e.g. by compensating the lack of accuracy in absolute positioning of a vehicle with an increase in the size of a slot. Furthermore, the VTS allows vehicles to switch between manual and slot-based driving on the fly, which means that slot driving can be enforced only when needed, leaving drivers the ability to drive manually if that is desired. Finally, the slot-based IVHS is designed in a way such that improvements in sensing and full vehicle automation lead to an increase in efficiency by further reducing the safe inter-vehicle gaps.

Efficiency

Slot-based driving addresses the limitations of human drivers by both allowing vehicles to drive in slots at close distances thanks to the employment of vehicle automation. Furthermore, the slot map also aims at coordinating the vehicles in a more efficient way than the competition that characterizes human driving. The hypothesis that addressing these limitations will yield an increase in traffic efficiency is validated empirically in Chapter 4.

Reliability

Similarly to efficiency, the insight is that the predefined nature of the slot behaviours, and thus the slot map, will lead to a reduction in the variability of journey times. Based on the definition of reliability (see Section 2.2.2.4), this will consequently lead to more reliable traffic. Nevertheless, this hypothesis must be validated empirically. This validation is performed in Chapter 4.

3.6.3 Limitations

Given the complexity of designing a complete IVHS solution, the design of the proposed slot-based approach uses a simplified system model. In particular, the vehicle model is limited to personal vehicles with identical capabilities and a constant acceleration, while the road model does not include gradients and curvatures. This, however, does not capture scenarios in which heavy goods vehicles driving up a slope cannot keep up with the speed of their slot. Additionally, situations in which vehicles do not comply with the rules specified by the slot map and the rules of slot-based driving in general are not explicitly addressed by this design. While safety can be ensured by equipping each vehicle with a safety controller that can override the slot-based driving controller whenever it detects a vehicle which does not comply to the rules, minimising the impact of such rogue vehicles will need to be addressed before a slot-based IVHS can be deployed. This includes addressing issues such as a security, communication reliability as well as malfunctioning vehicle control.

The fact that a base speed is shared by all slots can seem a limitation at a first glance, given that vehicles that want to drive faster are not allowed to do so. However, one of the core ideas of the slot-based approach is to only enforce slot-based driving when free flow conditions cannot be maintained by human drivers, which would drastically limit the speed of vehicles in the first place. Furthermore, the slot-based approach allows priority vehicles to drive on average at a speed higher than the slot base speed by changing and swapping slots.

Finally, the heuristics proposed a blueprint for generating slot maps while specifically addressing the most prominent causes of congestion. However, they do not cover an exhaustive list of scenarios that can occur on freeways as well as the impact of severe weather conditions on slot-based driving.

3.7 Conclusion

In this chapter, a novel IVHS approach was introduced, based on the concept of slot-based driving. The proposed design uses a simplified system model to fulfil the requirements: scalability, flexibility, increased efficiency and reliability. The analysis of the feasibility of the proposed slot-based driving system with regard to the current state of the art in vehicular technologies shows that, thanks to the flexibility of the slot-based approach, slot-based driving can be implemented using technologies that have already been deployed in production vehicles. Furthermore, incrementally adding technologies that are currently under development, such as employing more precise sensing, V2V communication or full vehicle control, would yield improvements in the efficiency of the IVHS. Issues related to driver compliance, such as when driving in a semi-autonomous mode, are not addressed in this work.

The design of the slot-based IVHS matches the scalability and flexibility requirements set at the beginning of this chapter, without sacrificing efficiency and reliability. The impact of slot-based driving on traffic depends on an efficient mapping of vehicles to slots, which in turn is subject to the distribution of traffic. To determine the efficiency and reliability of the slot-based approach, an empirical evaluation that takes into account

the stochastic nature of traffic flow needs to be carried out. The next chapter addresses this by providing a testbed implementation and evaluation of the slot-based approach in scenarios that replicate the main sources of congestion.

Chapter 4

Evaluation

This chapter describes the evaluation of the proposed slot-based approach to optimising freeway traffic. The scope of this evaluation is limited to the scenarios discussed in Section 3.5, and includes a stand-alone evaluation of the efficiency of transitioning from manual, human driving, to slot-based driving, as described by the VTS algorithm proposed in Section 3.4.

Evaluating the slot-based approach in the aforementioned scenarios requires the implementation of a Slot Provider within a testbed comprising of a dedicated freeway section and a large number of vehicles, capable of performing slot-based driving and implicitly of communicating with each other and with the Slot Provider. Unfortunately, the costs of setting up such a testbed make testing in real-world conditions infeasible. Instead, a traffic simulator was used to evaluate the slot-based approach.

The remainder of this chapter is organised as follows: Section 4.1 describes the design and implementation of an evaluation framework, built around the VISSIM traffic simulator. This framework is then used in Section 4.2 to evaluate the impact of the slot-based approach on traffic efficiency and reliability, using the heuristics defined in the previous chapter for a bottleneck scenario which requires vehicles to merge from three to two lanes, as well as a high-demand scenario in which vehicles merge from an on-ramp into freeway traffic.

4.1 Evaluation Framework

The core of a traffic simulator lies in the way in which it models the flow of traffic. Over the last fifty years, significant effort has been invested in traffic flow modelling (for a detailed review see Hoogendoorn and Bovy, 2001). Based on their level of detail, these models can be classified into three categories: microscopic, mesoscopic and macroscopic traffic flow models. Microscopic traffic flow models describe the behaviour of individual vehicles under the influence of neighbouring vehicles. A high level of detail is used to describe this behaviour, such as the speed, acceleration and lane changing manoeuvres of individual vehicles. Unlike microscopic models, macroscopic models describe traffic flow using cumulative traffic characteristics such as flow, density and average speed and do not represent individual vehicle manoeuvres. Finally, mesoscopic traffic models combine the properties of microscopic and macroscopic models to describe the behaviour of individual vehicles at a lower level of detail using aggregated data.

A high level of detail is required to simulate slot-based driving. The manoeuvres performed by individual vehicles during slot-based driving depend on the behaviour of neighbouring slots specified by the slot map, as well as the interaction with neighbouring vehicles (e.g. when changing or swapping slots). This high level of detail means that, to accurately simulate slot-based driving, a microscopic traffic model is necessary. For this work, VISSIM¹, a well-established microscopic traffic simulator, was used.

VISSIM uses a car-following model developed by Wiedemann (1974) that attempts to replicate the physical and psychological behaviour of drivers. A further adaptation of this model to the characteristics of freeway driving is provided by VISSIM in the form of the so-called “Wiedemann ’99” model. To evaluate the impact of the slot-based IVHS on traffic efficiency and reliability, VISSIM was extended with the ability to simulate slot-based driving. The following paragraphs describe the design and implementation of a framework built around VISSIM to evaluate the slot-based IVHS.

¹<http://www.ptvgroup.com>

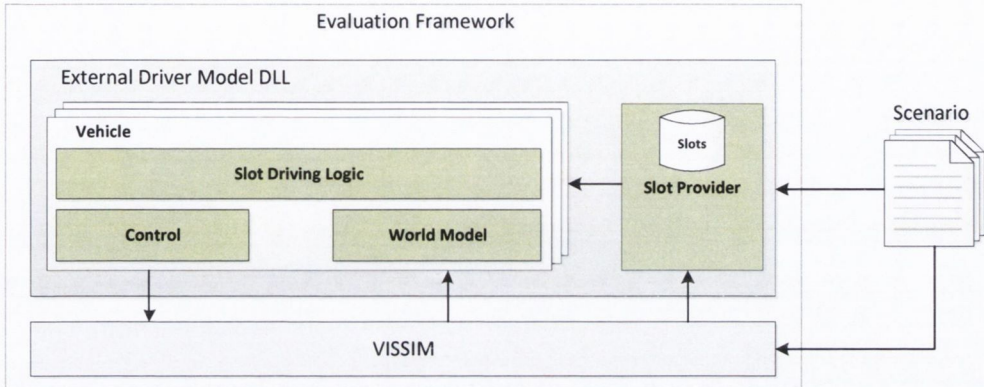


Fig. 4.1: Evaluation framework built around VISSIM

4.1.1 VISSIM integration

VISSIM is a microscopic, time-stepped and behaviour-based traffic simulator. The default driver model uses a stochastic distribution of speed and headway thresholds as parameters to the core car-following model to replicate the characteristics of individual, human drivers. These thresholds and several other parameters can be calibrated to improve the accuracy of the simulation provided by VISSIM. Most importantly, VISSIM offers the ability to completely replace this embedded driver model with an external driver model. The evaluation framework is built on this ability, which enables the implementation of slot-based driving as an external driver model for VISSIM.

The evaluation framework architecture is shown in Fig. 4.1. The VISSIM component provides the core traffic simulator functionality, while the External Driver Model replaces the built-in car-following model with slot-based driving. Inside the External Driver Model, there are two distinct component types: the Slot Provider, who is responsible for generating slots, and the Vehicles, who are responsible for driving within the slots generated by the Slot Provider. The functionality of each Vehicle is provided by a combination of three subcomponents: Control, World Model and Slot Driving Logic.

When VISSIM starts, it loads the External Driver Model and the Scenario parameters

defining the simulation. These Scenario parameters contain information such as the freeway section model, the vehicle model used, the simulation period as well as the traffic demand over that period. Additionally, these parameters specify the heuristic to be used by the Slot Provider to generate the slot map for the respective scenario, as well as whether slot-based driving should be enabled or not. Finally, the scenario parameters specify the performance metrics to be measured during the simulation.

VISSIM is implemented in C/C++ on top of the Microsoft Windows OS platform. When VISSIM is configured to use the external driver model instead of its own embedded driver model, it does so by calling a dynamic-link library (DLL). VISSIM specifies an interface containing three functions that the external driver model DLL must implement. Thus, the interaction between the individual components of the External Driver Model is determined by the specific design of this interface. VISSIM provides a stub of the external driver model interface, written in C/C++. A bridge from this C/C++ stub to the more modern C# programming language was implemented, with the goal of easing and speeding-up the development process. Thus, the entire implementation of slot-based driving within VISSIM was done using C# and as such all code extracts shown in this thesis are in C#.

The signatures of the three functions that must be implemented by the External Driver Model, are shown in Listing 4.1.

Listing 4.1: Interface for the External Driver Model DLL

```

1  int DriverModelExecuteCommand(long number); //command code number
2
3  int DriverModelSetValue(long type, //indicates the data item
4                          long index1, //index for data item
5                          long index2, //additional index for data item
6                          long long_value, //value for type
7                          double double_value, //value for type
8                          char *string_value); //value for type
9
10 int DriverModelGetValue(long type, //indicates the data item
11                         long index1, //index for data item
12                         long index2, //additional index for data item
13                         long *long_value, //plaholder for type value
14                         double *double_value, //plaholder for type value
15                         char **string_value); //plaholder for type value

```

During a simulation, VISSIM provides the External Driver Model with information

about each Vehicle, such as the vehicle's speed, current lane, acceleration and relative steering, as well as information about each of the vehicle's neighbours, such as relative distance and speed. The task of an External Driver Model is to compute and return to VISSIM values such as the desired speed, steering and target lane for each vehicle. In this way the external driver model can control both the longitudinal and the lateral driving tasks of a vehicle. VISSIM calls the `DriverModelSetValue` and `DriverModelGetValue` functions, to respectively send and receive information from the External Driver Model. The function `DriverModelExecuteCommand` is called by VISSIM to execute one of the four commands:

- `DRIVER_COMMAND_INIT`
- `DRIVER_COMMAND_CREATE_DRIVER`
- `DRIVER_COMMAND_MOVE_DRIVER`
- `DRIVER_COMMAND_KILL_DRIVER`

which are called respectively when the simulation environment is initialized, each time a vehicle is created in the simulation, each time a vehicle is about to be moved in the simulation, and each time a vehicle is removed from the simulation.

The interface between VISSIM and the External Driver Model imposes a specific interaction between the individual components of the framework. This interaction is described by the sequence diagram show in Fig. 4.2. The `DriverModel` represents the interface between VISSIM on one side, and the `SlotProvider` and `Vehicles` on the other side. When the simulation is initialised, the `DriverModel` creates and initialises the `SlotProvider`. Similarly, when VISSIM calls for a new driver to be created, the `DriverModel` creates and initialises a new `Vehicle` instance. VISSIM moves a vehicle in the simulation by calling the move command (via the `DriverModelExecuteCommand` function) on the `DriverModel`, preceded by a call to the `DriverModelSetValue` function and followed by a call to the `DriverModelGetValue` function. When the move command is called, the `DriverModel` asks the `Vehicle` to update its `WorldModel` with the latest information (e.g. current position, speed, neighbouring vehicles) and determine its future driving behaviour (e.g. desired speed and lane). Finally, when a vehicle leaves the simu-

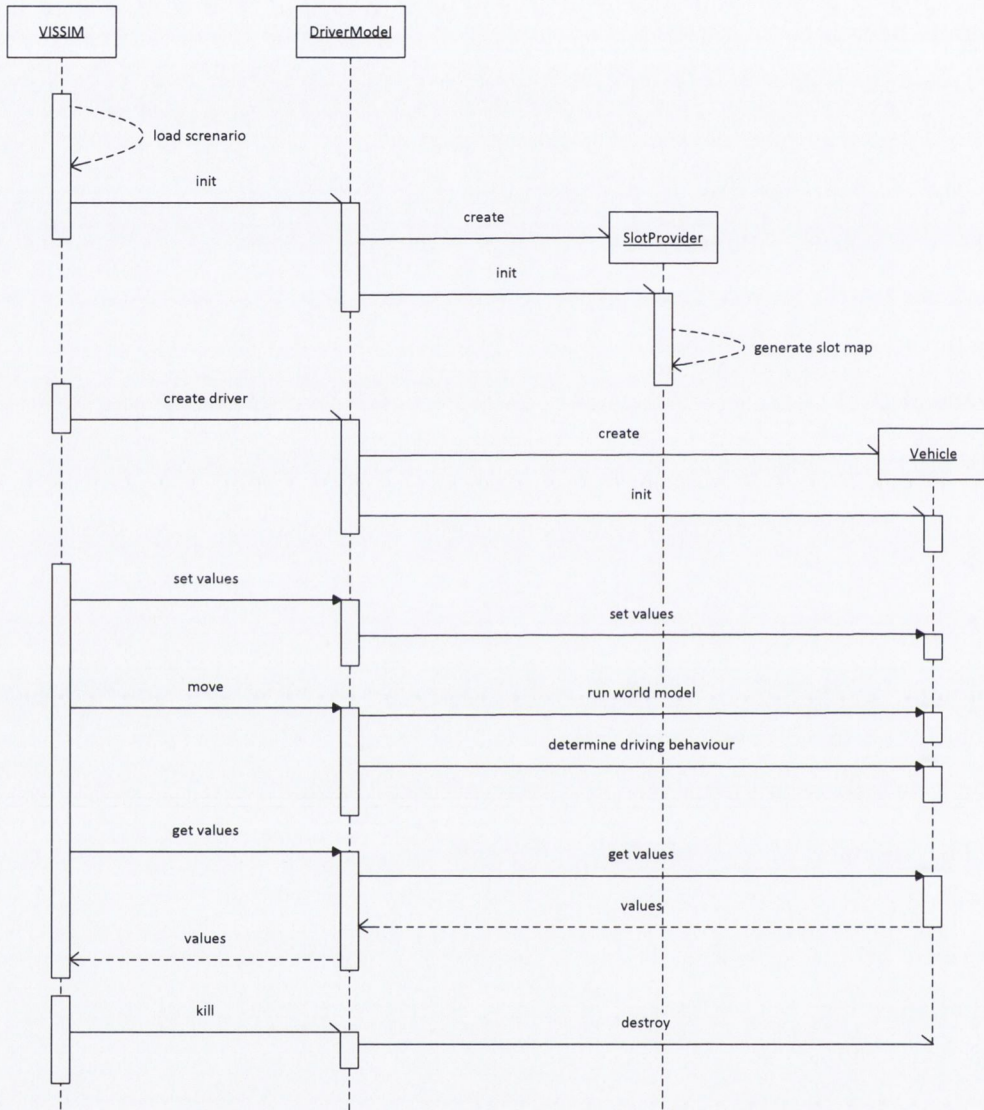


Fig. 4.2: Vehicle lifecycle within VISSIM

lated road network, VISSIM removes it from the simulation by calling the kill command (again via the `DriverModelExecuteCommand` function) on the `DriverModel`, which in turn destroys the respective `Vehicle` instance.

The `DriverModel` class implements the entire External Driver Model interface. List-

ing 4.2 shows only the implementation of the `DriverModelExecuteCommand` within the `DriverModel` class and follows the interaction described in Fig. 4.2. The full implementation of the External Driver Model interface is shown in full in Appendix A. A list of the data items that are communicated between VISSIM and the External Driver Model is available in the description of VISSIM's optional external driver model DLL.

Listing 4.2: Extract from `DriverModel.cs` showcasing the commands

```

1 public int DriverModelExecuteCommand(int number)
2 {
3     switch( number )
4     {
5         case DRIVER_COMMAND_INIT:
6             _slotProvider = SlotProvider.GetInstance();
7             _slotProvider.Initialize();
8             return 1;
9
10        case DRIVER_COMMAND_CREATE_DRIVER:
11            _activeVehicle.Initialize(_vehicles, _slotProvider.Conf, "
12                vehicleconfig.xml");
13            return 1;
14
15        case DRIVER_COMMAND_KILL_DRIVER:
16            _activeVehicle.Destroy();
17            _vehicles.Remove( _activeVehicle.Id );
18            return 1;
19
20        case DRIVER_COMMAND_MOVE_DRIVER:
21            _activeVehicle.RunWorldModel( _time );
22            _activeVehicle.DetermineDrivingBehaviour( _time );
23            return 1;
24
25        default:
26            return 0;
27    }

```

The `SlotProvider` is initialized when the `DRIVER_COMMAND_INIT` command is called (lines 5-8). This command is called only once, at the beginning of the simulation. The vehicles are created and initialized when the `DRIVER_COMMAND_CREATE_DRIVER` command is called (lines 10-12), as instances of the `Vehicle` class. They are stored as values in a dictionary (`_vehicles`), with their corresponding unique identifiers provided by VISSIM as keys, and are later destroyed and removed from the dictionary when the `DRIVER_COMMAND_KILL_DRIVER` command is called (lines 14-17), corresponding to a vehicle leaving the simulation. Finally, the desired steering, speed and acceleration of each vehicle are computed when VISSIM calls the `DRIVER_COMMAND_MOVE_DRIVER` command,

which in turn updates the world model and calls the `DetermineDrivingBehaviour()` method of the respective vehicle (lines 19-22).

The time-stepped nature of the VISSIM simulation means that, at each time-step, the `DRIVER_COMMAND_MOVE_DRIVER` command is called for each vehicle in the simulation. The variable `_activeVehicle` points to the vehicle for which the command was called. The value of this variable is set during the call of `DriverModelSetValue` function preceding the move command, by retrieving the respective `Vehicle` instance from the `_vehicles` dictionary. Additionally, while calling the `DriverModelSetValue` function, VISSIM passes relevant data to the External Driver Model. For example, before moving the vehicle, VISSIM provides the External Driver Model information such as the speed of the current vehicle and the relative position of neighbouring vehicles. When the `DRIVER_COMMAND_MOVE_DRIVER` command is sent, the task of the external driver model is to process the data provided by VISSIM into driving decisions. Finally, the computed driving decisions are queried in `DriverModelGetValue` by VISSIM. When moving a vehicle, such driving decisions would include the desired speed and acceleration of the vehicle, as well as any possible lane change.

The remainder of this section describes the implementation of the Slot Provider, used to generate slots, as well as the implementation of slot-based driving by the Vehicle and its subcomponents.

4.1.2 Slot Provider

The Slot Provider is responsible for generating the slots within which vehicles drive. Its functionality is implemented by three classes: `Slot`, `SlotProviderConfiguration` and `SlotProvider`. The relationship between these classes is shown in the class diagram depicted in Fig. 4.3.

The `SlotDrivingLogic` class represents the interface between a Vehicle and the Slot Provider. A consequence of this design is that the communication between individual vehicles and the Slot Provider is implemented using shared memory, meaning that the communication will occur with no packet loss and minimal latency. The use of shared

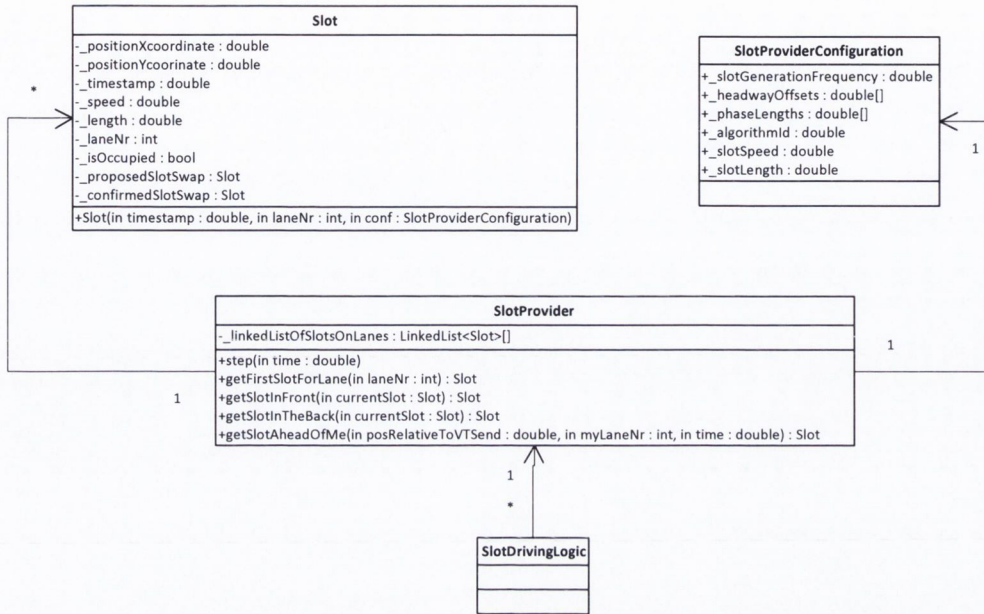


Fig. 4.3: Slot Provider Class Diagram

memory is made possible by the fact that the entire simulation runs on a single computer. Consequently, for the purpose of this evaluation, the specifics of vehicular communication are not taken into consideration. For the slot-based IVHS, the fact that vehicles are always able to communicate with the infrastructure without the occurrence of any errors means that manoeuvres that require vehicular communication to coordinate, such as changing or swapping slots, are accomplished faster in the simulated environment than in real-world conditions. The implementation can be adapted to account for communication latency in real-world conditions by simply increasing the time/space available to perform a certain manoeuvre.

The `slot` class represents an implementation of the slot model described in Chapter 3. An extract of the `slot` class is shown in Listing 4.3. The shared memory approach to communication can be observed in the attributes of a slot, such as its occupancy status (line 10) or the status of a proposed slot swap (lines 12-15). The `slot()` constructor

(lines 17-28) is used by the Slot Provider to generate slots. The slots are defined by their initial position (including the lane number), time stamp, base speed and length.

VISSIM dictates the global time used by the Slot Provider and all the Vehicles in the framework. The current position of a slot can be computed by a vehicle using the current time, the slot's initial position and timestamp, and the predefined behaviour (speed and trajectory profile over time). Note that this behaviour is not specified in the slot model's implementation. Instead, for coding convenience, the slot behaviours for each scenario are encoded in the Slot Driving Logic.

Listing 4.3: Extract from Slot.cs showcasing the slot model

```

1 public class Slot
2 {
3     private double _positionXcoordinate;
4     private double _positionYcoordinate;
5     private double _timeStamp; //created using the time from the VISSIM
        simulation
6     private double _speed;
7     private double _length; //width is irrelevant here, the slot is as
        wide as the lane
8     private int _laneNr;
9
10    private Boolean _isOccupied; //is the slot currently occupied by a
        car
11    //when a vehicle currenttly driving in a different slots wants to swap
        to this slot (currently occupied)
12    //it sets this value to point the vehicle's current slot
13    private Slot _proposedSlotSwap;
14    //when a vehicle accepts a proposed slot swap it sets the value of
        this to its current slot
15    private Slot _confirmedSlotSwap;
16
17    public Slot( double timeStamp, int laneNr, SlotProviderConfiguration
        conf)
18    {
19        //determine initial X coordinate
20        _positionXcoordinate = conf.DetermineInitialXCoord();
21        //emulate Y coordinate, not relevant for current implementation
22        _positionYcoordinate = laneNr*2-1;
23        _timeStamp = timeStamp;
24        _speed = conf.SlotSpeed;
25        _length = conf.SlotLength;
26        _laneNr = laneNr;
27        _isOccupied = false; //at the time or creation no slot is occupied
28    }
29 }

```

The Slot Provider is implemented as a singleton in the SlotProvider class. An extract from this class is shown in Listing 4.4. It showcases the attributes and methods

relevant for the creation (lines 19-25) and update (lines 27-38) of the slot map. The `SlotProviderConfiguration` class (lines 1-9) encodes the slot formation as well as the algorithm that specifies the slot behaviour that will be used in the respective scenario. This configuration is loaded from an external file, allowing it to plug-in different slot formations and behaviours without having to modify the source code. The corresponding slot map is encoded in the form of an array of linked lists, with each linked list corresponding to a lane in the freeway. The `Step()` method updates the slot map at each time-step according to the slot formation.

Listing 4.4: Extract from `SlotProvider.cs` showcasing the slot map generation

```

1  public class SlotProviderConfiguration
2  {
3      double _slotGenerationFrequency;
4      double[] _headwayOffsets;
5      double[] _phaseLengths;
6      double _algorithmID;
7      double _slotSpeed;
8      double _slotLength;
9  }
10
11 public class SlotProvider
12 {
13     private Slot newSlot;
14     private LinkedList<Slot>[] _linkedListOfSlotsOnLanes;
15     private Boolean _wasPreviousSlotLeader;
16     private SlotProviderConfiguration _conf;
17     private double _lastSlotGenerationTimestamp;
18
19     public void Initialize()
20     {
21         for (int i = 0; i < _linkedListOfSlotsOnLanes.Length; i++)
22         {
23             _linkedListOfSlotsOnLanes[i] = new LinkedList<Slot>();
24         }
25     }
26
27     public void Step(double time)
28     {
29         //generate new slots for each lane only when the time has changed
30         //and at intervals of _conf.SlotGenerationFrequency seconds
31         if (time!=_lastSlotGenerationTimestamp && time != 0 && (time %
32             _conf.SlotGenerationFrequency == 0))
33         {
34             _lastSlotGenerationTimestamp = time;
35             for (int i = 0; i < _linkedListOfSlotsOnLanes.Length; i++)
36             {
37                 newSlot = new Slot(time+_conf.HeadwayOffsets[i], i + 1,
38                     _conf);
39                 _linkedListOfSlotsOnLanes[i].AddFirst(newSlot);
40             }
41         }
42     }
43 }

```

```
39     }  
40 }  
41 }
```

The slot information is queried by vehicles directly from the Slot Provider through a vehicle's Slot Driving Logic, rather than being available locally. Since shared memory is used to pass data between components, this decision is made to simplify the implementation and has no further impact on the accuracy of the simulation. The `SlotProvider` provides a set of helper methods used to query the slot map. These methods, shown in Listing 4.5, are used by vehicles during slot-based driving to determine a target slot, change the current slot or swap slots with an adjacent vehicle, and represent implementation of the respective mechanisms described in Section 3.4.

Listing 4.5: Extract from `SlotProvider.cs` showcasing the query methods

```
1 public Slot getFirstSlotForLane(int laneNr)  
2 {  
3     return _linkedListOfSlotsOnLanes[laneNr-1].First.Value;  
4 }  
5  
6 public Slot getSlotInFront(Slot currentSlot)  
7 {  
8     if (currentSlot == null)  
9         return null;  
10    return _linkedListOfSlotsOnLanes[currentSlot.LaneNr - 1].Find(  
11        currentSlot).Next.Value;  
12 }  
13 public Slot getSlotInTheBack(Slot currentSlot)  
14 {  
15     if (currentSlot == null)  
16         return null;  
17    return _linkedListOfSlotsOnLanes[currentSlot.LaneNr - 1].Find(  
18        currentSlot).Previous.Value;  
19 }  
20 public Slot getSlotAheadOfMe(double posRelativeToVTSend, int myLaneNr,  
21    double time)  
22 {  
23    Slot currentSlot = getFirstSlotForLane(myLaneNr);  
24    while (posRelativeToVTSend > (time - currentSlot.TimeStamp) *  
25        currentSlot.Speed)  
26    {  
27        currentSlot = getSlotInFront(currentSlot);  
28    }  
29    return currentSlot;  
30 }
```

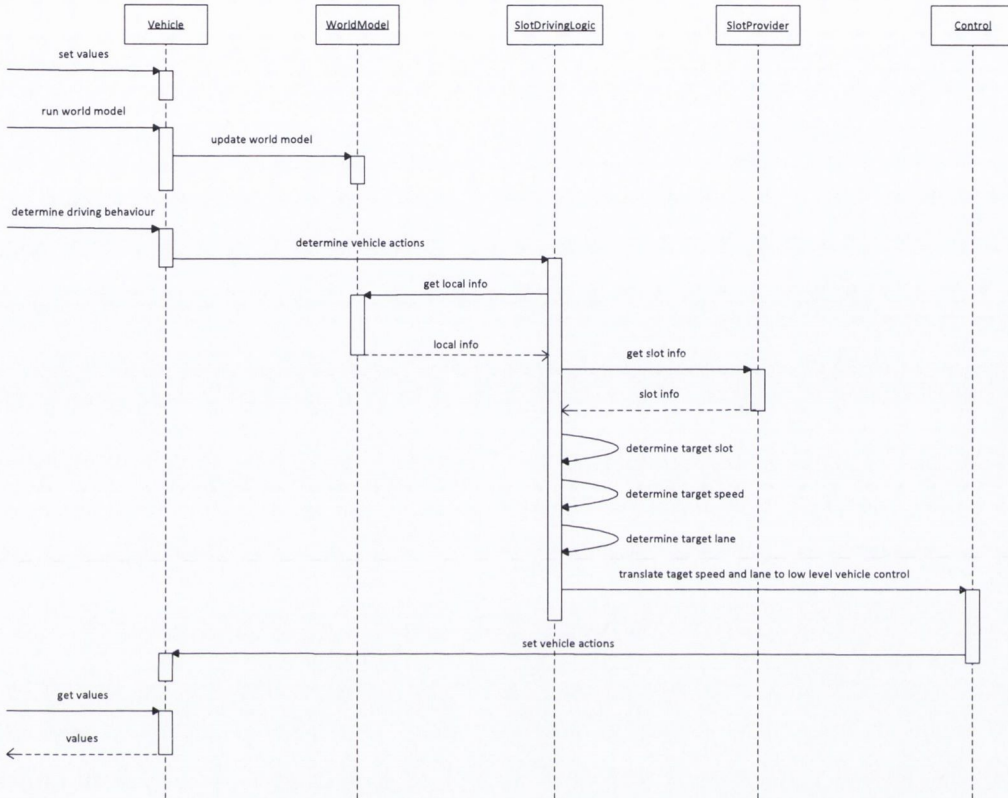


Fig. 4.4: Interaction between the individual components of a vehicle

4.1.3 Vehicle

Each Vehicle is responsible for performing slot-based driving, using local information it receives from VISSIM through the Driver Model as well as slot information it queries from the Slot Provider. The Vehicle component achieves this together with three other sub-components: World Model, Slot Driving Logic and Control. The interaction between these components, as well as the interaction with the Slot Provider, is shown in the sequence diagram depicted in Fig. 4.4.

The local information from VISSIM is forwarded by the `DriverModel` and stored by the `Vehicle`. When the `DriverModel` receives a `move` command from VISSIM, it asks

the `Vehicle` to update its `WorldModel`. The `WorldModel` processes the lower-level local information stored by the `Vehicle` and makes the obtained higher-level local information (e.g. distance to the first vehicle ahead) available to the `SlotDrivingLogic`.

After the `WorldModel` is updated, the `DriverModel` asks the `Vehicle` to determine its driving behaviour for the next time step, i.e. determined it's desired speed, acceleration, steering and target lane. This request is accomplished by the `SlotDrivingLogic`, which uses the higher-level local information in the `WorldModel` in combination with the slot information it queries from the `SlotProvider` to make high-level driving decisions, such as selecting a target slot to move into, changing or swapping slots. These decisions are then processed by the `Control` component into low-level vehicle control instructions, such as desired speed, acceleration, steering and target lane. The control instructions are then stored in the `Vehicle` and retrieved by the `DriverModel`. VISSIM then uses this information to move each vehicle in the simulation.

The class diagram depicted in Fig. 4.5 shows the implementation of these components as classes, and the relationship between them. The `VehicleConfiguration` class is used to model and store scenario-specific static attributes shared by all vehicles in the simulation, such as maximum speed and acceleration, radar range and others. The `VTS` class represents a subcomponent of the Slot Driving Logic that isolates the implementation of the VTS algorithm from the scenario-specific slot driving.

Each vehicle in VISSIM is represented by an instance of the `Vehicle` class. An extract of the `Vehicle` class is shown in Listing 4.6. It stores local information received from VISSIM (lines 9-14), as well as control instructions (lines 16-17) which are used by VISSIM to move the corresponding vehicles in the simulation. Furthermore, it is characterised by the following attributes: an instance of `SlotDrivingLogic` (line 19), which handles slot-based driving aspects; an instance of `WorldModel` (line 20), which fuses information about the surrounding environment and provides methods for querying this information; and an instance of `Control` (line 21), which handles the low-level aspects of vehicle control. The remainder of the code extract shows how a `Vehicle` instance is created (lines 23-27) and initialised (lines 29-39), as well as the implementation of the

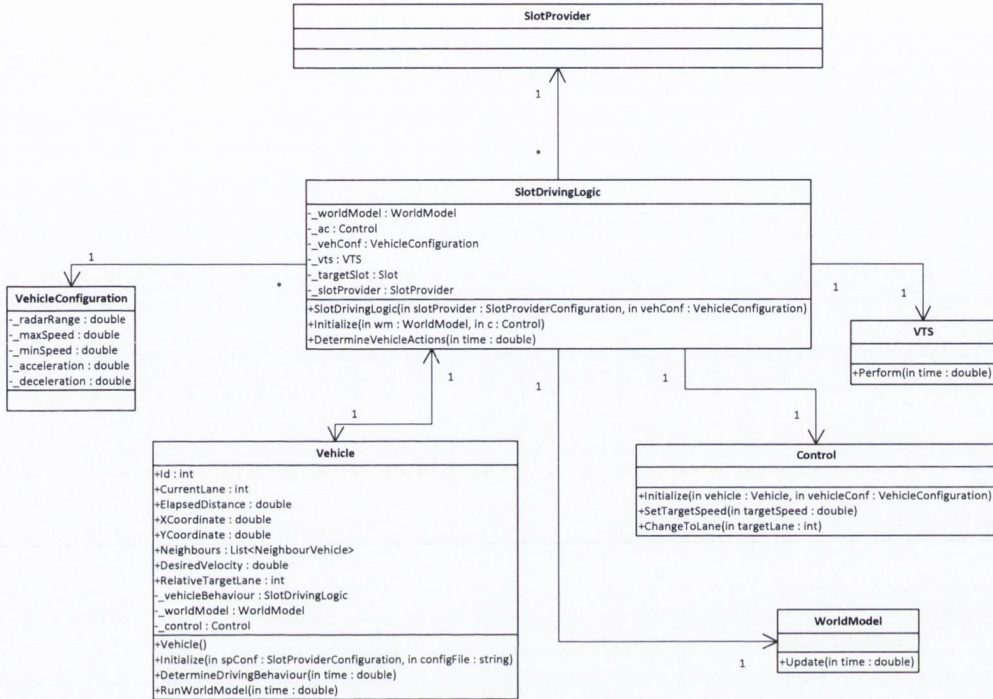


Fig. 4.5: Vehicle class diagram

DetermineDrivingBehaviour() (lines 41-44) and RunWorldModel() (lines 46-49) methods, which are called by the DriverModel when VISSIM sends it a *move* command.

Listing 4.6: Extract from Vehicle.cs

```

1 public class VehicleConfiguration
2 {
3     double _radarRange, _maxSpeed, _minSpeed, _acceleration, _deceleration;
4 }
5
6 public class Vehicle
7 {
8     //information from VISSIM
9     public int Id { get; set; }
10    public int CurrentLane { get; set; }
11    public double ElapsedDistance { get; set; }
12    public double XCoordinate { get; set; }
13    public double YCoordinate { get; set; }
14    public List<NeighbourVehicle> Neighbours { get; set; }
15    //information for VISSIM
  
```

```

16     public double DesiredVelocity { get; set; }
17     public int RelativeTargetLane { get; set; }
18
19     private SlotDrivingLogic _vehicleBehaviour;
20     private WorldModel _worldModel;
21     private Control _control;
22
23     public Vehicle()
24     {
25         _worldModel = new WorldModel();
26         _control = new Control();
27     }
28
29     public void Initialize(SlotProviderConfiguration spconf, string
        configfile)
30     {
31         _worldModelWrapper.Initialize(this);
32         VehicleConfiguration vehConf = VehicleConfiguration.Deserialize(
            configfile);
33         _driving.Initialize(this, vehConf);
34         if(spconf.AlgorithmID==LMAConstants.ALGORITHM_ID_LMA)
35         {
36             _vehicleBehaviour = new SlotDrivingLogic(spconf, vehConf);
37             _vehicleBehaviour.Initialize(_worldModel, _control);
38         }
39     }
40
41     public void DetermineDrivingBehaviour(double time)
42     {
43         _vehicleBehaviour.DetermineVehicleActions(time);
44     }
45
46     public void RunWorldModel(double time)
47     {
48         _worldModelWrapper.Update( time );
49     }
50 }

```

The abstract class `SlotDrivingLogic` implements the basic slot-based driving functionality. A code extract from the `SlotDrivingLogic` class is shown in Listing 4.7. The information from VISSIM regarding the current state of the vehicle and its neighbours is stored in the `WorldModel` instance (line 3), and updated at each time-step. When the `DetermineVehicleActions()` method (line 28) is called, the `SlotDrivingLogic` uses the local information, combined with the slot information it queries from the Slot Provider, to make high level driving decisions. These high level driving decisions result in a `_targetSlot` in which the vehicle should drive. The `performVTS()` method (lines 30-33) shows how the `VTS` subcomponent of the `SlotDrivingLogic`, represented by the `_vts` attribute (line 6), is used to determine the initial `_targetSlot`. Finally, the `_control`

attribute (line 4) of the `SlotDrivingLogic` class represents an instance of the `Control` class, and is used to transform these high-level decisions into low-level vehicle control data, such as desired speed, acceleration and target lane. This data is used by VISSIM to move the vehicle in the simulation.

Listing 4.7: Extract from `SlotDrivingLogic.cs` showcasing the basic functionality

```

1 public abstract class SlotDrivingLogic
2 {
3     protected WorldModel _wm;
4     protected Control _ac;
5     protected VehicleConfiguration _vehConf;
6     protected VTS _vts;
7     protected Slot _targetSlot;
8     private static SlotProviderConfiguration _spConf;
9
10    public SlotDrivingLogic(SlotProviderConfiguration spconf,
11                           VehicleConfiguration vehconf)
12    {
13        _vehConf = vehconf; //for a heterogenous set of vehicles
14        _vts = new VTS(_vehConf);
15        if(_spConf==null)
16        {
17            _spConf = spconf;
18        }
19
20    public void Initialize(WorldModel wm, Control c)
21    {
22        _wm = wm;
23        _ac = c;
24        _targetSlot = null;
25        _vts.Initialize(_wm, _ac);
26    }
27
28    void DetermineVehicleActions(double time);
29
30    void performVTS(double time)
31    {
32        _targetSlot = _vts.Perform(time);
33    }
34 }

```

The basic slot-driving functionality implemented by the `SlotDrivingLogic` is shown in Listing 4.8. Here, the implemented utility methods for moving into an empty slot (lines 1-7) and swapping slots with another vehicle using a three-way handshake (lines 9-35) are shown. These utility methods are made available to be used within the `DetermineVehicleActions()` method implemented by the concrete subclasses specific to each scenario. As such, creating a scenario-specific subclass is simply a matter of com-

binning these methods according to the slot behaviour and scenario-specific optimisations reflected through slot changing and swapping. To simplify the implementation, the slot behaviour specific to each scenario was encoded within the `DetermineVehicleActions()` method, rather than communicated by the Slot Provider as specified by the slot model in Chapter 3.

Listing 4.8: Extract from `SlotDrivingLogic.cs` showcasing changing slots

```

1  protected void moveToEmptySlot(Slot newTargetSlot)
2  {
3      //no risk of race conditions due to the sequential nature of VISSIM
4      _targetSlot.IsOccupied = false;
5      _targetSlot = newTargetSlot;
6      _targetSlot.IsOccupied = true;
7  }
8
9  protected void initiateSlotSwap(Slot newTargetSlot)
10 {
11     if (newTargetSlot.ProposedSlotSwap == null)
12         newTargetSlot.ProposedSlotSwap = _targetSlot;
13 }
14
15 protected void acceptSlotSwap()
16 {
17     if(_targetSlot.ProposedSlotSwap != null)
18     {
19         Slot tmpSlot = _targetSlot.ProposedSlotSwap;
20         _targetSlot.ProposedSlotSwap.ConfirmedSlotSwap = _targetSlot;
21         _targetSlot.ProposedSlotSwap = null;
22         _targetSlot = tmpSlot;
23     }
24 }
25
26 protected void acknowledgeSlotSwap()
27 {
28     if (_targetSlot.ConfirmedSlotSwap != null)
29     {
30         Slot tmpSlot = _targetSlot.ConfirmedSlotSwap;
31         _targetSlot.ConfirmedSlotSwap = null;
32         _targetSlot = tmpSlot;
33         return;
34     }
35 }

```

The VTS algorithm, introduced in Chapter 3, is performed by each vehicle at the beginning of each scenario. An extract from the `VTS` class, which provides the implementation of this algorithm within VISSIM, is shown in Listing 4.9. This extract showcases the procedure of getting into an empty slot. Based on the position of a vehicle relative to the vehicle ahead as well as the end point of the VTS, a vehicle's `Control` performs

one of the following:

- accelerate to maximum speed to break potential vehicle clusters (line 26)
- control the speed to be at exactly one inter-slot headway behind vehicle in front (line 38)
- move into the first slot ahead (lines 30 and 43)

Listing 4.9: Extract from VTS.cs showcasing getting into an empty slot

```

1  public Slot Perform(double time)
2  {
3      if (_targetSlot == null)
4      {
5          _targetSlot = determineSlot(time);
6      }
7      if (_targetSlot != null)
8      {
9          GetIntoSlot(SlotUtils.currentPositionOfSlot(_targetSlot, time) -
10                     _wm.GetCoordinates().X, time);
11         _targetSlot.IsOccupied = true;
12     }
13     return _targetSlot;
14 }
15 Slot determineSlot(double time)
16 {
17     Slot tSlot = null;
18     _wm._vehicle.ActiveLaneChange = 0; //don't change lanes during VTS
19     _wm._vehicle.DesiredLaneAngle = 0;
20     //if car can't sense a vehicle ahead
21     if (_wm.GetDistanceToVehicleAhead() > _vehConf.RadarRange)
22     {
23         if (_vtsEndPosition - _wm.GetCoordinates().X >
24             _minDistanceToGetIntoSlot)
25         {
26             //no car ahead and enough space to get into a slot, get to max
27             speed
28             _ac.SetTargetSpeed(_vehConf.MaxSpeed);
29         }
30         else
31         {
32             tSlot = _slotProvider.getSlotAheadOfMe(carSlotPosition(), _wm.
33                 GetLaneNr(), time);
34         }
35     }
36     else
37     {
38         if (_vtsEndPosition - _wm.GetCoordinates().X >
39             _minDistanceToGetIntoSlot)
40         {
41             //still enough space to get into a slot so maintain or close
42             the gap

```

```
38         createInterSlotGap(_wm.GetDistanceToVehicleAhead());
39     }
40     else
41     {
42         //need to get into a slot now, the first slot ahead is free
43         tSlot = _slotProvider.getSlotAheadOfMe(carSlotPosition(), _wm.
            GetLaneNr(), time);
44     }
45 }
46 return tSlot;
47 }
```

The remaining subcomponents of the Vehicle, represented by the `WorldModel` and `Control` classes, are simply modelled after the way in which VISSIM send and receives data from the External Driver Model. No further description of these is provided as they are specific to VISSIM.

4.1.4 Summary

This section provided a general description of the framework developed for evaluating the slot-based IVHS. As can be observed from the above paragraphs, the implementation is specific to the VISSIM traffic simulator. While some aspects such as vehicles dynamics are considered in the implementation of slot-based driving, others such as sensing and communication errors are not. The goal of this implementation is not to demonstrate how slot-based driving would be implemented in the real-world but rather to provide a framework in which the impact of slot-based driving on traffic can be evaluated.

The code extracts shown in the above paragraphs are part of the entire code base of the framework and have been selected to showcase the design employed in building an evaluation framework around VISSIM's external driver model.

4.2 Case Studies

The slot-based IVHS aims at fulfilling four requirements: scalability, flexibility, efficiency and reliability. As argued in Chapter 3, the hybrid architecture of the proposed system, based on the slot abstraction, explicitly addresses the scalability and flexibility requirements. This section discusses the efficiency and reliability of the system. For this

purpose, efficiency is measured using two performance metrics:

- *Throughput*, measured as the number of vehicles that travel during the course of an hour through a freeway section.
- *Average delay*, measured as the difference between the average travel time and the ideal travel time on a freeway section (i.e. the time required by a single vehicle to travel the respective section with no other vehicles around)

while reliability is measured using:

- *Standard deviation of delays* across repeated measurements of the same scenario using a random seed

Using the evaluation framework described in the previous section the throughput, average delay and standard deviation of average delay are measured for slot-based driving. The results are then compared against those obtained for human drivers as emulated by VISSIM's "Wiedemann'99" model, which has been used extensively to model driver behaviour on freeways (Moen et al., 2000; Bloomberg and Dale, 2000; Fellendorf and Vortisch, 2001; Park and Schneeberger, 2003; Gomes et al., 2004; Chitturi and Benekohal, 2008).

The remainder of this section is organised as follows: Section 4.2.1 describes the testbed used for running the simulations, including the vehicle and slot model parameterisation. The efficiency of the VTS algorithm is then discussed in Section 4.2.2, followed by an evaluation of the scenario-specific heuristics proposed in Chapter 3 with respect to their efficiency (Sections 4.2.3 and 4.2.4) and reliability (Section 4.2.5). The chapter is concluded by a discussion of the evaluation results.

4.2.1 Testbed

The simulations were performed using a single workstation, equipped with an Intel Core i7 CPU with 4 cores running at a clock speed of 2.93GHz and 8 GB of RAM. The evaluation framework uses VISSIM 5.20 and runs on a 64-bit version of Microsoft Windows 7.

All vehicles in the simulation have identical dynamics, in line with the system model described in Chapter 3. The desired speed of manually-driven vehicles as well as the base speed of all slots across all case studies were set at $30m/s$, which is a reasonable estimate of the average free-flow speed on freeways based on the data provided by the Highway Capacity Manual (2012). The acceleration that can be achieved by a vehicle at any time is constant at $4m/s^2$ (typical of higher-end vehicles), while the deceleration is constant at $-6m/s^2$ (which would be experienced as fairly hard braking). These rather extreme values for acceleration and deceleration were chosen to reduce the time required for manoeuvres such as getting into a slot or changing slots, thereby reducing the overall size of the road model required to simulate these manoeuvres in VISSIM. A more realistic acceleration model does not affect the feasibility of slot-based driving and simply requires the slot behaviours and transition periods to be adapted accordingly. The maximum speed a vehicle and thus a slot can achieve was set to a conservative $40m/s$, while the average inter-vehicle headway was set to $1.5s$, which is typical of high-demand traffic. The slot length is set at $5m$, which is slightly larger than the length of a personal vehicle in accordance with the car model used by VISSIM, and is shared by all slots across all case studies. Finally, an inter-slot headway of $1s$ is used across all case studies, corresponding to a capacity of 3600 slots/lane/hour.

In the design chapter, the vehicle model was limited to personal vehicles with identical driving capabilities and geometries. Since the slot model is designed to match the vehicle model, this raises the question of whether the slot-based approach can be applied to the more general case of heterogeneous vehicles with different dynamics and geometries. As argued in the previous paragraph, the slot model is flexible with respect to the driving capabilities of a vehicle. However, the fixed geometry of a slot, and in particular the length of a slot (set to $5m$, i.e. slightly larger than the typical length of a personal vehicle), is not large enough to encompass all types of vehicles, such as buses or trucks. This is clearly a problem, as slot-based driving assumes vehicles are driving within slots.

To study the effects of vehicle length on slot-based driving, the simplified vehicle model — described in Chapter 3 — was extended to include vehicles with a length

of over 10m. For this purpose, the VISSIM Heavy Goods Vehicle (HGV) model was used. The scope of this was to study the effects of vehicle length, rather than other dynamic aspects such as weight or engine power. To isolate the effect of vehicle length, the VISSIM HGV model was modified to use the same vehicle dynamics (maximum speed and acceleration profile) as personal vehicles. The case studies presented in the following paragraphs describe how slot-based driving was adapted to account for HGVs in the respective scenarios.

Finally, because the evaluation is performed using a traffic simulator, slot-based driving cannot be evaluated using semi-autonomous vehicle control, as the behaviour of human driving assisted by the vehicle cannot be simulated in a realistic fashion. As such, only fully autonomous slot-based driving is evaluated.

4.2.2 Moving Vehicles into Slots

One of the four requirements addressed by the design of the slot-based driving system is flexibility. As shown in Chapter 3, core to the flexibility of the slot-based IVHS is the ability to enforce slot-based driving only when traffic conditions on the freeway are such that the free flow is under threat. This on-the-fly switch from manual, human driving, to slot-based driving, is achieved through the VTS algorithm, described in Section 3.4. The VTS algorithm must be employed, at the latest, when traffic demand is nearing the capacity of human drivers on the respective freeway section.

The VTS algorithm uses slots that are large enough to encompass a personal vehicle but not large enough to encompass a HGV. Since increasing the size of the slots so that they can encompass HGVs would result in a large amount of unused freeway space, the VTS was adapted to cater for HGVs by allowing them to occupy two consecutive slots. Vehicles that are located directly behind a HGV detect that the vehicle ahead is a HGV, which will occupy two slots instead of just one, and adapt their target slot accordingly.

It is essential that the VTS algorithm achieves an efficient transition of vehicles to slots and does not in itself cause a bottleneck. While the Slot Provider generates slots with an inter-slot headway lower than the current average inter-vehicle headway (for

this evaluation 1s and 1.5s respectively), a bottleneck can still occur when clusters of vehicles driving close to each other arrive at the beginning of the transition. The VTS algorithm specifically addresses these situations by covering any empty gaps ahead of a vehicle during the transition period. The effect of this is a more even distribution of traffic that helps break out vehicle clusters.

For the purpose of this evaluation, a three-lane freeway section with a length of 2500m is used, at the end of which vehicles will be driving in slots. A longer road section is not needed due to the fact that vehicles driving in slots at a constant speed cannot have an impact on upstream traffic. When choosing the length of the freeway section, the same thinking applies to all other scenarios evaluated in this thesis.

Vehicles enter the freeway travelling at a desired speed averaging approximately 30m/s, and are controlled, during the first 300m, by VISSIM's human driver emulation. When slot-based driving is employed, the next 2200m are used for transitioning from manual to slot-based driving. Knowing the maximum vehicle speed of 40m/s, the constant acceleration of 4m/s², and the constant slot speed of 30m/s, and assuming a vehicle enters the transition period driving at 30m/s and the space ahead of it is free of any vehicles, this vehicle can reach the end point of the transition approximately 540m ahead of a slot entering the transition period at the same time. Knowing the length of a slot is 5m and the inter-slot headway is 1s, this means that at a maximum a vehicle can pass 15 empty slots during the transition period. For vehicles driving in close formations, this means that during the 2200m of transition, the VTS algorithm can accommodate a maximum cluster size of 15 vehicles without causing a bottleneck, assuming that there are no other vehicles 540m ahead of this cluster. While such uneven distributions of traffic are unlikely, especially in free flow conditions, this analysis shows that a transition length of 2200 is large enough to break clusters of average sizes, while in the same time small enough to make its employment feasible.

To determine the impact of the VTS algorithm on traffic, the throughput achieved on a three-lane freeway section was measured in a scenario where the VTS algorithm is applied. This was then compared against the throughput achieved by human drivers

(simulated by VISSIM) on the same road, without applying the VTS algorithm. The evaluation is performed using a penetration rate of HGVs of 0%, 10% and 20% of the total number of vehicles. According to the Highway Capacity Manual (2012), the average penetration of HGVs on freeways is currently around 10%, meaning that these penetration rates represent best-case, average-case and worst-case scenarios. Combined, this results in 6 different test cases. Each test case received 10 VISSIM simulation runs using a random seed. The results shown in Fig. 4.6 represent the average throughputs over the 10 simulation runs for each of the 6 cases.

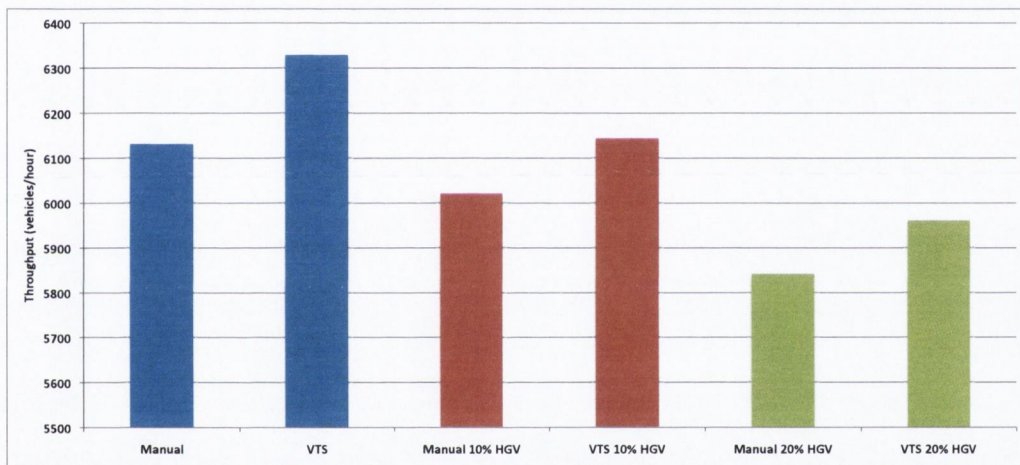


Fig. 4.6: Manual vs. VTS throughput on a three-lane freeway section

For the case with no HGVs, the average throughput achieved by human drivers over the three lanes is approximately 6150 pc/h (or 2050 pc/h/ln), which closely resembles the typical freeway capacity values specified by the Highway Capacity Manual (2012). When HGVs are introduced in the simulation, their presence has a negative impact on the throughput achieved. Nevertheless, the most important observation that can be made here is that the VTS algorithm outperforms the manual drivers, independently of the HGV penetration rate. This shows that the VTS algorithm does not cause a bottleneck when applied in heavy-traffic conditions and thus performs an efficient mapping of vehicles to slots. Note that the higher throughput achieved by the VTS compared to manual drivers is due to its strategy of breaking vehicle clusters by having individ-

ual vehicles accelerate into empty spaces ahead. Furthermore, since the VTS scenario includes an initial phase of manual driving followed by a transition from manual to slot-based driving, the overall throughput achieved by the VTS is limited by the throughput achieved by human drivers during this initial phase, rather than the transition to slots. The experiments described in the remainder of this chapter show that, by merging vehicles into empty slots at on-ramps, slot-based driving can achieve an overall higher throughput than that achieved by the VTS algorithm alone.

4.2.3 Lane Merging

As seen in Chapter 3, a sudden reduction in capacity caused by road works or accidents represents one of the main causes of congestion on freeways. The algorithm proposed in Section 3.5.1 deals with such cases by performing a merging from m to n lanes using slot-based driving. To evaluate the efficiency of slot-based driving in such a situation, a bottleneck scenario was set-up in VISSIM in the form of a three-lane freeway segment merging into a two-lane segment. This capacity drop from three to two lanes can be caused by either road works or accidents and creates a bottleneck which, under high traffic conditions, will cause congestion. In this scenario, the lane merging algorithm is preceded by the VTS transition. Follower vehicles close the gap to leader vehicles to a headway of $0.5s$, leaving a headway of $1.5s$ between the follower and the next leader. At a slot base speed of $30m/s$ this represents a gap of $45m$, which is large enough for two personal vehicles with a length of $5m$ to move into.

In this VISSIM model, vehicles join the highway at the beginning of a three-lane section, which is 2500 meters long. This section then merges into a 235 meters long two-lane section. The length of the three-lane section was chosen to sufficiently accommodate both the transition to slots performed by the VTS algorithm as well as the merging procedure performed by the lane merging algorithm. When vehicles join the freeway, their behaviour is controller by the Wiedemann '99 driver model. After 300 meters, the VTS algorithm takes control. Based on empirical evidence from previous experiments, a length of 1200 meters was this time deemed sufficient for accommodating the VTS

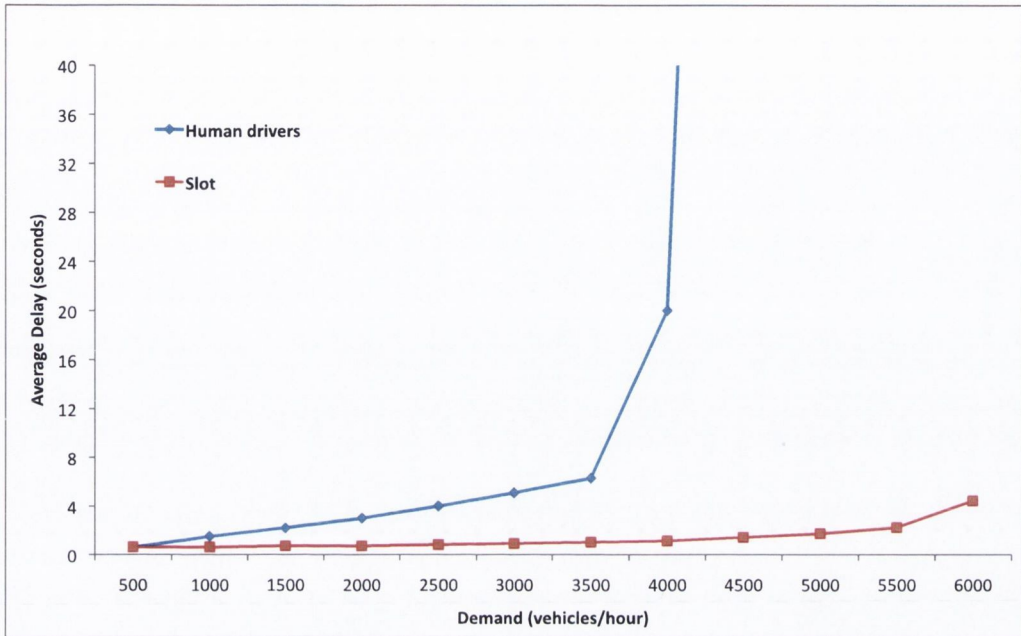


Fig. 4.7: Delay for 3-to-2 lane merging with no HGVs

algorithm. 1500 meters after joining the highway, vehicles start the LMA procedure, which is easily accommodated, given the vehicle and slot model parameters, over a 1000 meters-long stretch of the freeway.

The efficiency of the LMA was evaluated under increasing demand conditions. Slot-based driving was evaluated against VISSIM's Wiedemann '99 driver model, using the average delay as a performance metric. For the bottleneck scenario, a high delay is also an indicator that congestion has formed and therefore the throughput has dropped. Initially, all vehicles in the simulation are personal vehicles with identical capabilities. The average delay of human drivers and slot-based driving are compared as the incoming demand of traffic is gradually increased in the following simulation runs. The demand is initially set at 500 vehicles/hour, representing low traffic conditions, and increased at each new simulation run with an additional 500 vehicle/hour until a demand of 6000 vehicles/hour is reached, which represents the upper bound as it is limited by the throughput of the VTS algorithm over three lanes, as seen in Section 4.2.2. This results in 12 simulation runs, each with the duration of 1 hour of simulation time. The results of this evaluation

are shown in Fig. 4.7.

Even at relatively low-traffic conditions, slot-based driving outperforms human drivers. This is partially due to the nature of the VTS algorithm, in which vehicles attempt to close empty gaps ahead by accelerating to their maximum speed and maintaining this speed until they either sense another vehicle ahead or they approach the end of the VTS transition, leading to an overall higher average travel speed on the respective freeway section. However, as the demand is increased and traffic becomes more dense, these situations occur more rarely and the average speed converges towards that exhibited by human drivers. Nevertheless, the difference in average delay continues to increase, with slot-based driving clearly outperforming human drivers. A dramatic increase in delay for human drivers can be observed between 3500 and 4000 vehicles/hour. The explanation for this is that as traffic demand nears the capacity of the two-lane section, the effects of the inefficient merging performed by human drivers are amplified, eventually leading to congestion which is reflected by an increased average delay from a relatively low 6.3s to 20s which represents an increase of over 300%. Once congestion arises, any increase in demand leads to an exponential increase in delay experienced by human drivers. These delays of over 100s cannot be captured meaningfully and are therefore not shown in Fig. 4.7. By comparison, the delay experienced by slot-based driving is relatively stable in the face of increased demand, increasing steadily to reach 4s by the time the maximum observed throughput of the VTS over three lanes is reached, as seen previously.

Slot-based driving relies on the idea that a virtual slot surrounds each vehicle. Because the LMA merges slots in the empty space between two other slots, the notion of slot length is particularly critical. In Section 4.2.2, we have seen how the VTS algorithm was adapted to accommodate HGVs. While the size of the slot was kept to m , the VTS was modified to map HGVs to two consecutive slots. During the LMA, consecutive slots on a lane might find themselves on different lane, as dictated by the merging procedure. The only exception are the slots on the rightmost lane. As such, the LMA can be adapted by first constraining HGVs to the rightmost lane. Furthermore, because vehicles moving from the second into the first lane do so between a follower slot located

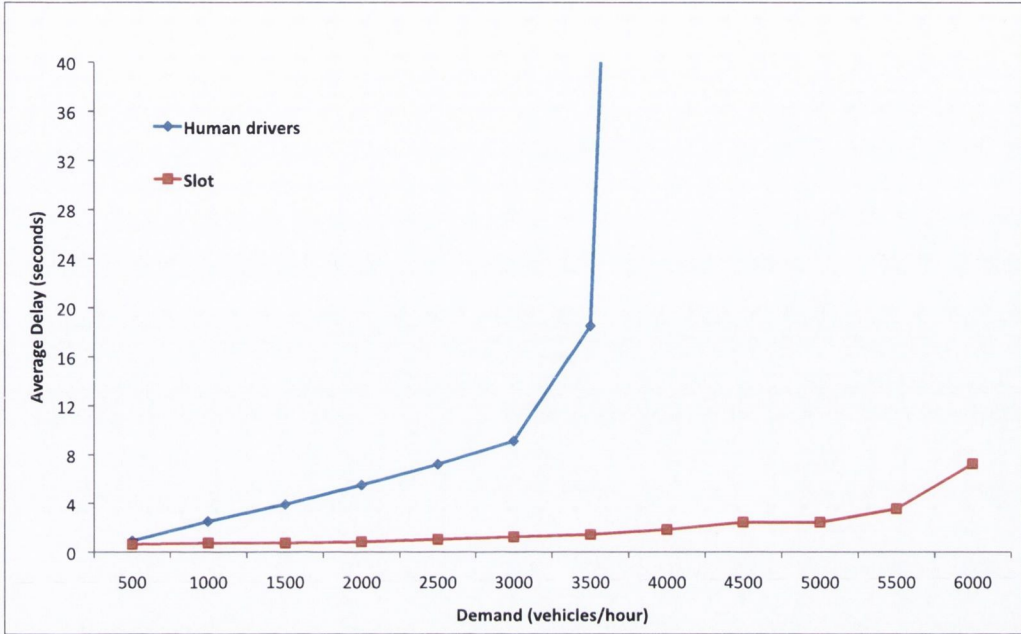


Fig. 4.8: Delay for 3-to-2 lane merging with 10% HGVs

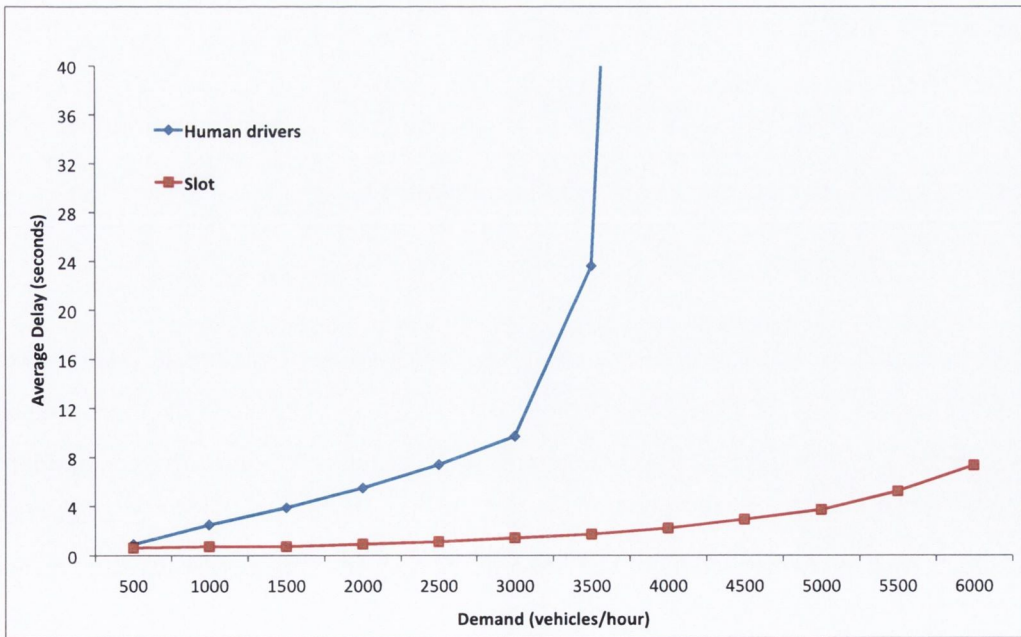


Fig. 4.9: Delay for 3-to-2 lane merging with 20% HGV

directly in front and a leader slot located directly at the back of the merging vehicle, to avoid vehicles attempting to move in between two slots occupied by a HGV, the first of the two consecutive slots occupied by a HGV must be a leader slot. This means that the space available for a HGV is equal to the sum of the two slot lengths plus the inter-slot space. Given a slot length of $5m$, a base slot speed of $30m/s$ and a reduced inter-slot headway between a follower and a leader slot of $0.5s$, this results in $25m$ available for HGVs, which is large enough to fit even the largest HGVs².

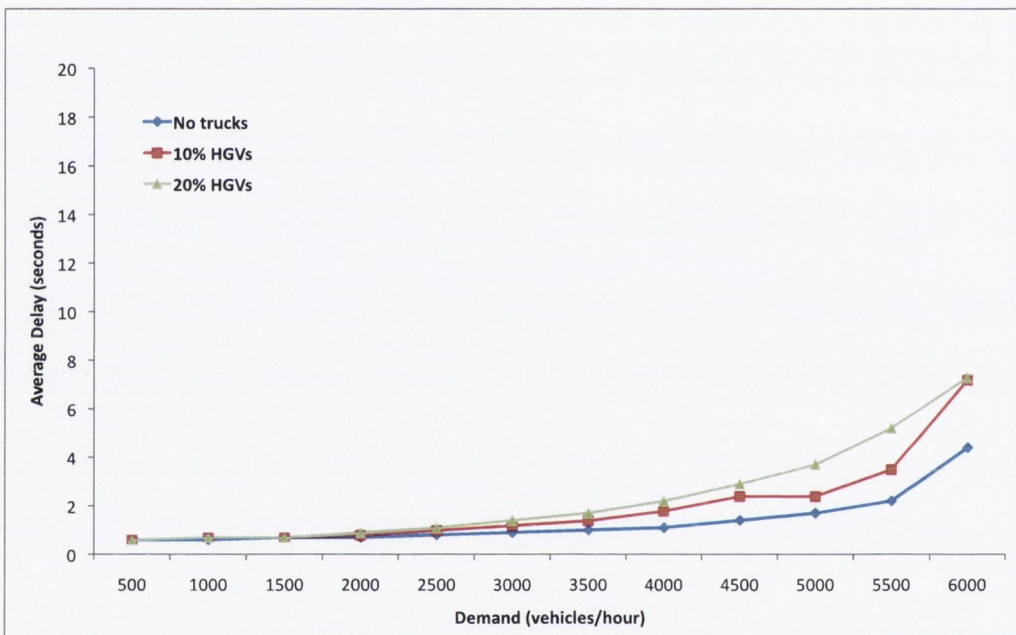


Fig. 4.10: Impact of HGVs on the lane merging algorithm

To evaluate the impact of HGVs on slot-based driving, the same simulations were repeated using a 10% and 20% penetration rate of HGVs. The results of this evaluation are shown in Fig. 4.8 and Fig. 4.9 respectively, with the delay of human drivers compared against the delay experienced by slot-based driving. Similarly to the previous evaluation, slot-based driving clearly outperforms human drivers. For human drivers, an increase of over 100% in delay is observed between 3000 and 3500 vehicles/hour for a 10% HGV

²Some exceptions exist, with some countries in Europe, as well as Australia, allowing HGVs as large as $25.25m$ and $26m$ respectively, which would require a slight increase in the minimum headway used during the lane merging algorithm

penetration rate, while for a 20% HGV penetration rate the delay increases by almost 150%. Just as in the case with no HGVs, from the point congestion arises any increase in demand results in an exponential increase in delay for human drivers. Unlike human driving, slot-based driving performs well for both 10% and 20% HGV penetration rate, reaching a maximum delay of 7.2s and 7.3s respectively. Fig. 4.10 shows the impact of HGVs on slot-based driving, where 10% and 20% HGVs penetrations are compared against the case with no HGVs. The fact that the increase in delay when HGVs are used is under 4s shows that the impact of HGVs on slot-based driving is rather minimal, demonstrating the flexibility of the slot-based approach.

4.2.4 On-ramp Merging

Out of the main causes of congestion presented in Chapter 3, recurring congestion caused by high traffic demand, commonly known as “rush hour” traffic, is responsible for approximately half of the total congestion. In Chapter 3, an on-ramp merging heuristic was proposed, followed by a discussion on how the slot formation used impacts on the ability of vehicles to exit the freeway. The following paragraphs discuss the performance of the aforementioned heuristic using the chessboard formation. For this purpose, the road model initially considers no exit ramps. This is later extended to include an exit-ramp in the vicinity of the on-ramp. Unlike the previous scenarios, in this scenario the vehicle model does not include HGVs, as they do not complicate the merging procedure and their impact on throughput is already known from Section 4.2.2.

On-ramp merging without considering exit ramps

In this scenario, the road model only consists of a three-lane freeway section followed by a four-lane on-ramp section. The on-ramp merging procedure, described in Chapter 3, is shown again in Fig. 4.11, for the sake of completeness. The main road of the freeway has a total length of 3000m. Similarly to the previous scenario, the first 1500m are used by the VTS algorithm to allow vehicles to transition to slot-based driving. When inter-vehicle cooperation is employed, the second part of the road before the on-ramp

is used by vehicles to move into free slots located to the left front of the current slot, if such slots are available. At $2600m$, the main road merges with the on-ramp. The length of the on-ramp's acceleration lane is $180m$, which is the default value suggested by the Highway Capacity Manual (2012). The RSU is located at $300m$ from the merging point and only assigns on-ramp vehicles with empty slots which are located further away from the merging point than the RSU, at the time of the assignment. The ramp length of $180m$ is larger than the $112.5m$ required by a vehicle to accelerate from virtual standstill to $30m/s$, which represents the speed of the main road slots. Furthermore, the minimum $120m$ distance between the assigned slot and the on-ramp vehicle ensures that, during the $7.5s$ required by the vehicle to accelerate to $30m/s$, the slot cannot overtake the vehicle.

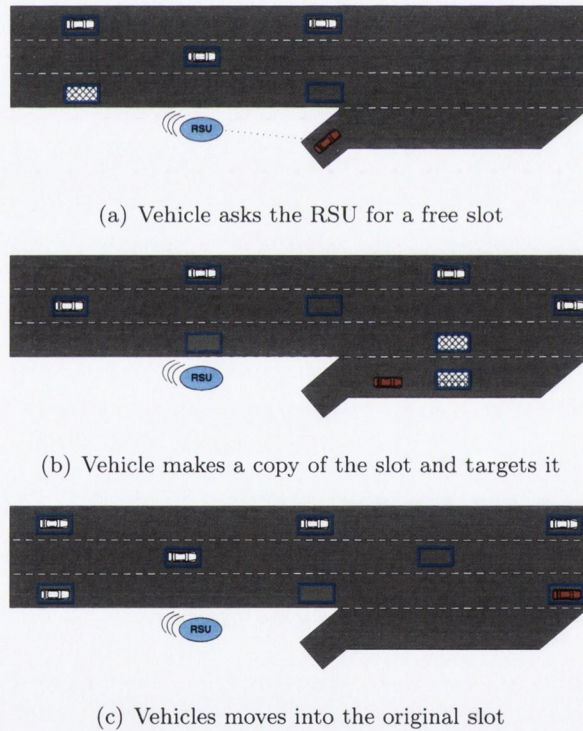


Fig. 4.11: On-ramp merging

A centralized approach based on V2I, described in Chapter 3, was used to implement the inter-vehicle cooperation, due to its relatively simple implementation within the sim-

ulation environment. The slots are generated by the Slot Provider using the chessboard formation proposed in Chapter 3 with a headway of one second. The chessboard pattern is achieved by generating slots on odd lanes with a timestamp that is 0.5 seconds earlier than that of their counterparts on even lanes. The Slot Provider communicates the slot information to the vehicles and marks slots as free or occupied when vehicles move out of or into slots. For the on-ramp merging scenario, the Slot Provider performs the task of the RSU and assigns free slots on the first lane to on-ramp vehicles entering the freeway.

The slot-based merging algorithm was evaluated with and without the optimisation proposed in Chapter 3, which involves vehicles moving into free slots located to the left-front of the their current slot in order to free-up slots on the rightmost lane for on-ramp vehicles. The results of slot-based driving with and without this optimisation are compared against those of human drivers as simulated by VISSIM.

The efficiency of merging is evaluated with respect to:

- Throughput: the number of vehicles that merge from the on-ramp into the main road within an hour;
- Delay: the average delay experienced by vehicles on the on-ramp;

under:

- Medium traffic conditions: main road demand of 3600 vehicles/hour;
- Heavy traffic conditions: main road demand of 4700 vehicles/hour.

The throughput on the on-ramp was measured for human drivers as well as slot-based driving with and without the optimisation provided by the cooperation of vehicles on the main road. Each of these three was evaluated under medium and heavy traffic conditions. The results of this evaluation are shown in Fig. 4.12. Under medium traffic conditions, the slot-based driving without cooperation achieves a 41% increase in throughput when compared to human drivers. Slot-based driving with cooperation performs better and achieves a 106% increase in throughput compared to human drivers. The two algorithms perform even better under heavy traffic conditions compared to human drivers: 230% and 452% increase for slot-based driving without and with cooperation respectively.

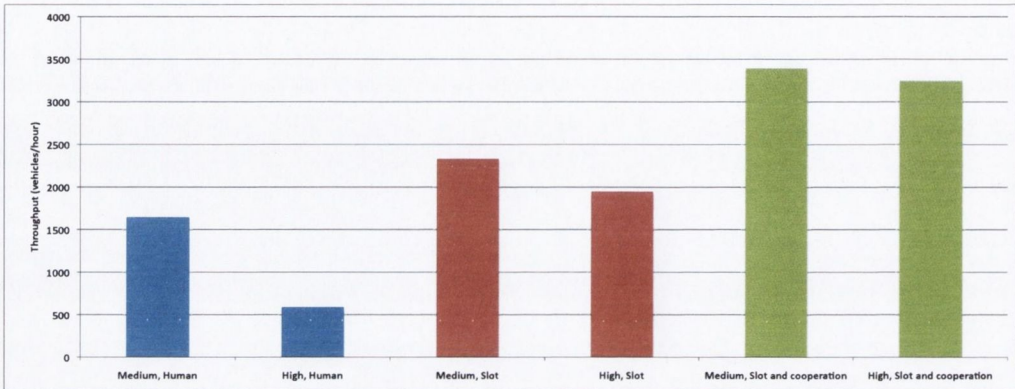


Fig. 4.12: Maximum on-ramp throughput

These results show that the throughputs achieved by both slot-based driving algorithms clearly outperform the throughput of the merging performed by VISSIM's emulation of human drivers. As expected, cooperation between vehicles on the main road has a positive effect and increases the throughput on the on-ramp. Furthermore, the slot-based driving algorithms perform even better under heavy traffic conditions when compared to human drivers. It is worth noting that heavy traffic causes a decrease of on-ramp throughput across all three merging approaches when compared to medium traffic. However, for human drivers the throughput under heavy traffic decreases by 64% when compared against medium traffic, compared to 17% and 4% for slot-based driving without and with cooperation respectively.

The on-ramp delay was measured under medium and heavy traffic conditions on the main road. Fig. 4.13 and Fig. 4.14 show the delay for human drivers, slot-based driving and slot-based driving with cooperation under medium and heavy traffic conditions respectively. The on-ramp traffic demand was gradually increased from 200 vehicles/hour to 2000 vehicles/hour which corresponds to very heavy on-ramp traffic conditions. We can observe that, for human drivers, the delay increases exponentially with respect to the on-ramp demand. For the two slot-based driving algorithms the delay remains small and slightly increases as the on-ramp demand increases. The traffic demand on the main road increases the delay for human drivers but has very little effect on the other

two approaches, thanks to the high throughput and efficient merging of the slot-based approach.

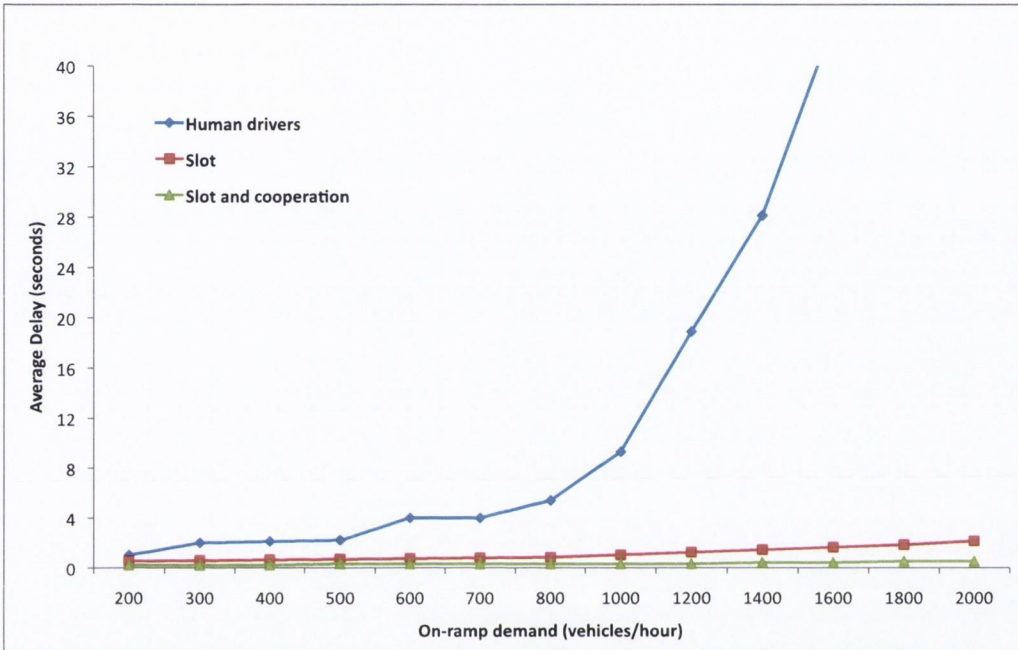


Fig. 4.13: On-ramp delay under medium traffic conditions on the mainroad

Note that in Fig. 4.14, the measured delay for 500 vehicles/hour is smaller than the delay obtained for 400 vehicles/hour. This result shows that there are other factors that can influence the manual merging procedure such as the distribution of vehicles across both main road and on-ramp. For example, a cluster of vehicles on the main road reaching the merging point at the same time as a cluster of vehicles on the on-ramp can have a significant impact on the merging. This is not a problem for slot-based driving since the VTS algorithm was designed to break-up such clusters when vehicles are mapped to slots, as shown previously.

The impact of exit ramps on on-ramp merging

Given the nature of “rush hour” traffic, slot-based driving will possibly need to be employed over a large portion of a freeway and not only bottleneck segments as was

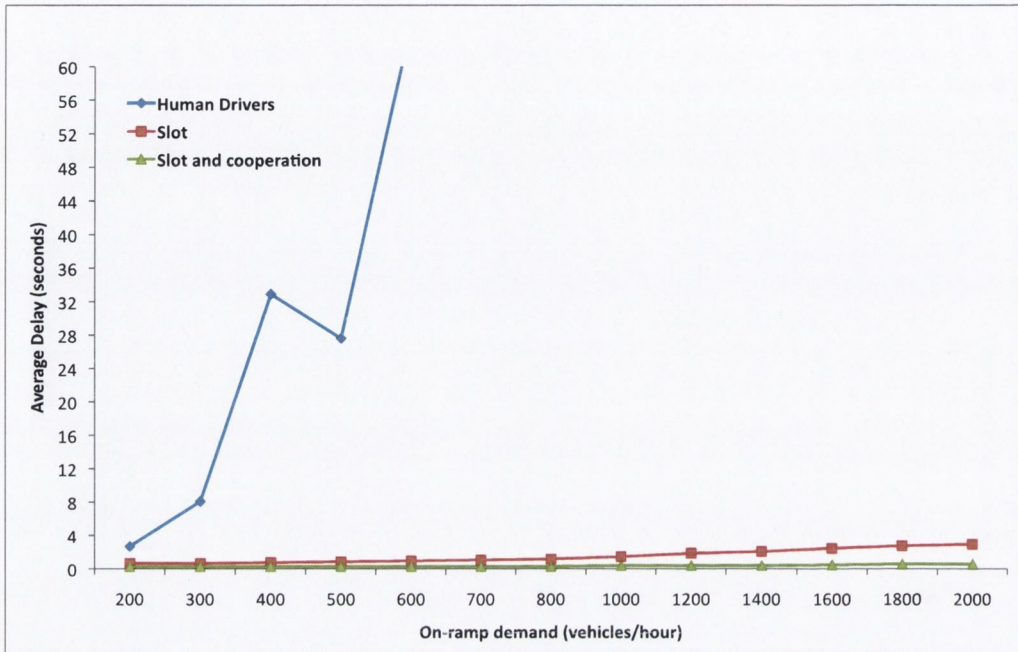


Fig. 4.14: On-ramp delay under heavy traffic conditions on the main road

the case in the lane merging scenario. This raises the issue of vehicles needing to exit the freeway. This can be achieved using the slot-changing and slot-swapping procedures presented in Chapter 3. Vehicles located on the left lanes change or swap slots in order to move to the rightmost lane. Most commonly, exit ramps precede on-ramps. This has a positive effect on the performance of the on-ramp merging procedure as more slots will become free on the rightmost lane after the exit ramp. However, in the rather uncommon case of an exit ramp located downstream, and at a relative close distance to the on-ramp, this can have a negative effect on the performance of the on-ramp merging procedure, vehicles targeting that exit ramp will compete with on-ramp vehicles for slots on the rightmost lane. To evaluate the impact of exit ramps in such an unfavourable scenario, the previous VISSIM model of a basic freeway 3-lane segment followed by an on-ramp segment was extended with a 2500m-long basic 3-lane freeway segment followed by an 180m exit segment.

The on-ramp delay was compared under medium and heavy traffic conditions for:

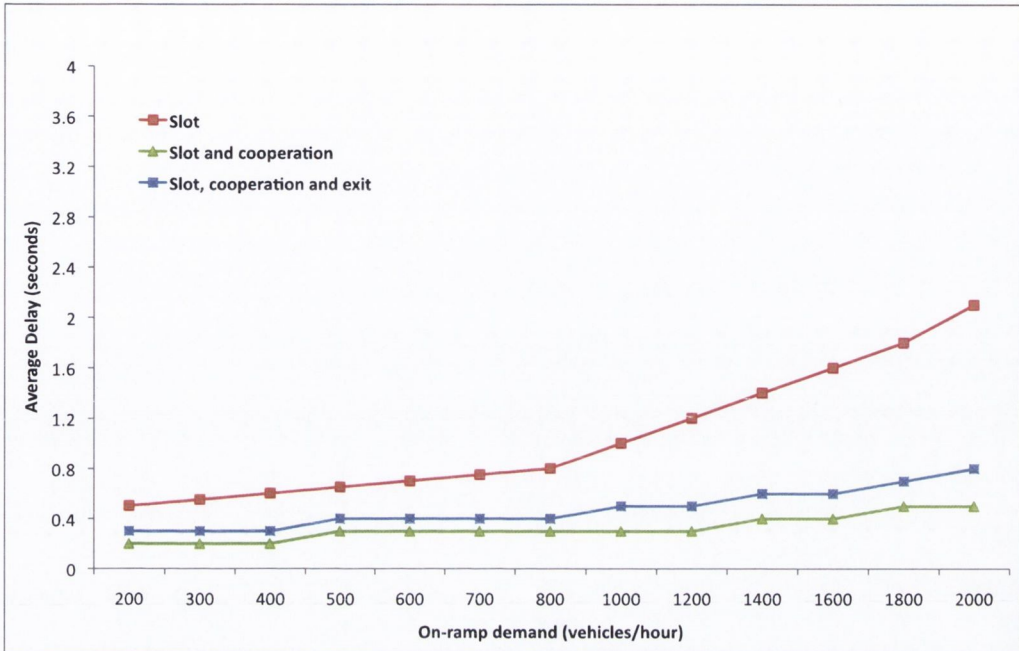


Fig. 4.15: On-ramp delay under medium traffic conditions on the main road with an exit ramp

- slot-based driving without any optimisation (i.e. without freeing up slots on the rightmost lane) and without considering exit ramp
- slot-based driving with optimisation but without considering exits
- slot-based driving with the same optimisation while considering exits

The difference between the second and the third cases is that while in the second case vehicles would attempt to free-up the rightmost lane independently of their destination, for the third case vehicles in the rightmost lane will not leave this lane if they target the following exit lane. The results for medium traffic and heavy traffic conditions, with 10% of all vehicles set to target the exit ramp while the rest continue on the freeway, are shown in Fig. 4.15 and Fig. 4.16 respectively.

These results show that even for a relatively high number of vehicles targeting the exit ramp, the increase in delay for the optimisation case is relatively small. Furthermore,

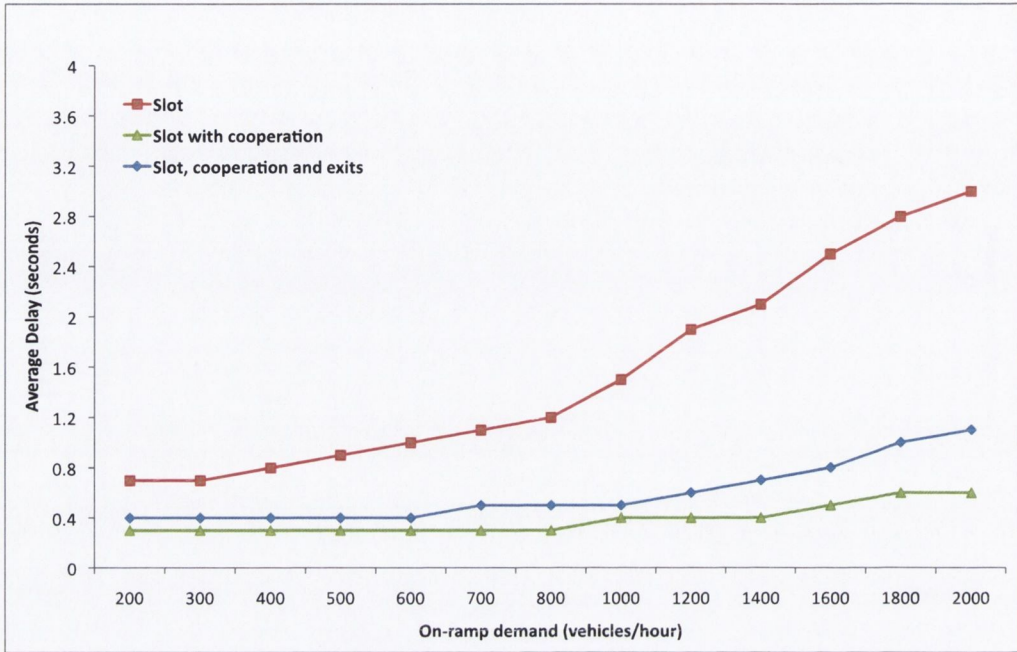


Fig. 4.16: On-ramp delay under heavy traffic conditions on the main road with an exit ramp

the optimisation with exits case outperforms the case with no optimisation and exits, showing that the optimisation strategy has a positive impact regardless of the presence of an exit lane in the vicinity of the off-ramp.

4.2.5 Reliability

The other functional goal besides increasing efficiency is increasing reliability. As defined in Chapter 3, reliability is measured as the variability of travel times and is particularly relevant under congested traffic conditions. As such, the reliability of the slot-based IVHS was evaluated against the same scenarios for which the efficiency was evaluated under traffic conditions that would normally lead to congestion.

For the lane merging scenario, a total of six test cases were evaluated, each using a traffic demand of 6000 vehicles/hour comprising:

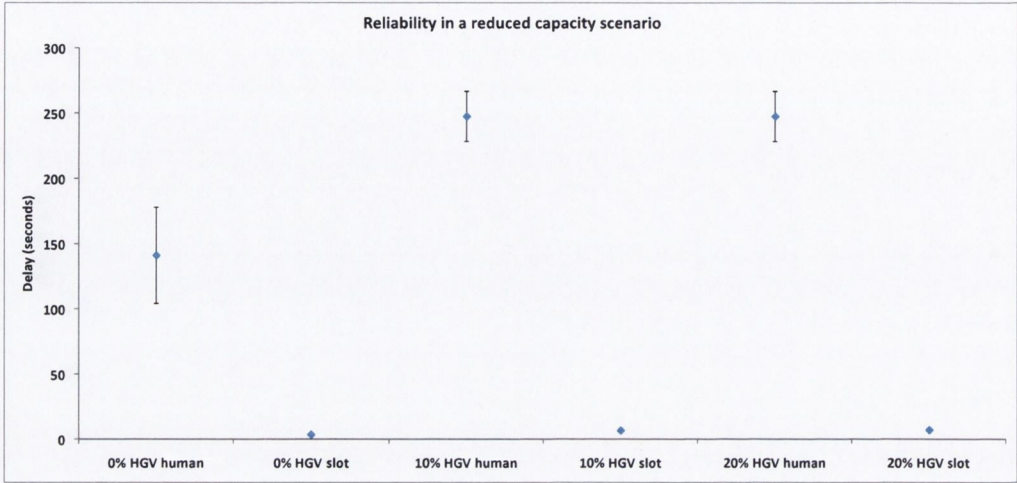
- 100% personal vehicles and 0% HGV, human driving (as emulated by VISSIM);

- 100% personal vehicles and 0% HGV, slot-based driving;
- 90% personal vehicles and 10% HGV, human driving;
- 90% personal vehicles and 10% HGV, slot-based driving;
- 80% personal vehicles and 20% HGV, human driving;
- 80% personal vehicles and 20% HGV, slot-based driving.

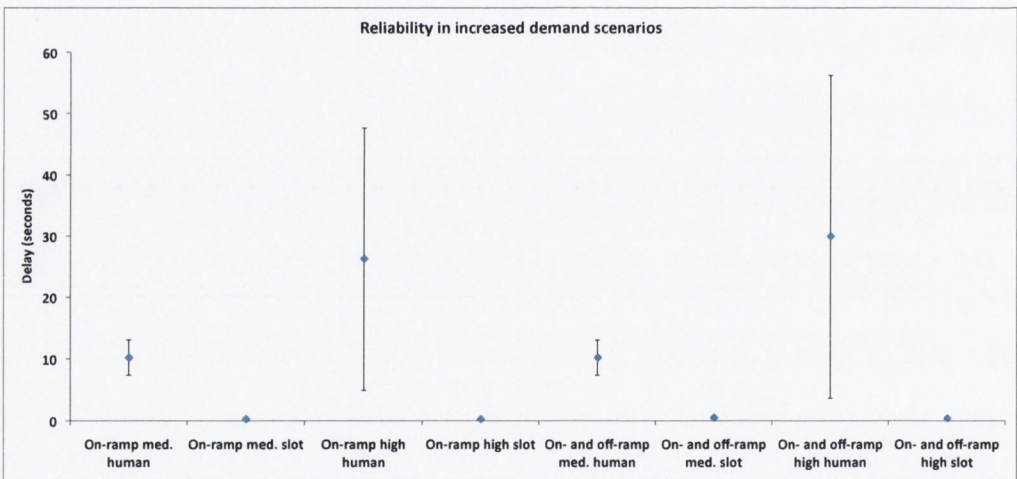
The average delay for each test case was measured during an hour of simulation time and then the process was repeated using a different random seed. This results in 10 average delay measurements for each test case, which were used to calculate the mean delay and standard deviation for the respective test cases. The results are shown in Fig. 4.17(a). The standard deviations in the delay times experienced by the human drivers are, in the order used in Fig. 4.17(a): 36.89s, 19.13s and 19.13s, while the standard deviations of delay times as a result of slot-based driving are close to zero at: 0.31s, 0.50s and 0.39s respectively.

The same testing procedure was used for the two ramp merging scenarios. No HGVs were used for these scenarios but instead the traffic conditions varied between medium traffic demand on the main road (3600 vehicles/hour) and high traffic demand on the on-ramp (1000 vehicles/hour), and high traffic demand on the main road (4700 vehicles/hour) and medium traffic demand on the on-ramp (500 vehicles/hour). This led to a total of eight test cases:

- medium main-road demand, no off-ramps, human driving;
- medium main-road demand, no off-ramps, slot-based driving;
- high main-road demand, no off-ramps, human driving;
- high main-road demand, no off-ramps, slot-based driving;
- medium main-road demand, one off-ramp, human driving;
- medium main-road demand, one off-ramp, slot-based driving;



(a)



(b)

Fig. 4.17: Reliability of the slot-based approach

- high main-road demand, one off-ramp, human driving;
- high main-road demand, one off-ramp, slot-based driving.

The same methodology as in the reduced capacity scenario was used to measure the reliability. The results are shown in Fig. 4.17(a). The standard deviations in the delay times experienced by the human drivers are, in the order used in Fig. 4.17(a): 2.85s,

21.34s, 2.85s, and 26.31s, while the standard deviations of delay times as a result of slot-based driving are close to zero at: ≈ 0 , 0.05s, 0.04s and 0.05s respectively.

Overall, these results show that while delay times experienced by human drivers are significantly varied, slot-based driving offers a considerable increase in traffic reliability as shown by its nearly zero standard deviations with respect to the delay times experienced.

4.3 Discussion

This chapter described how the efficiency and reliability of the slot-based IVHS were measured within an evaluation framework built around VISSIM. Overall, the results of the evaluation for the lane and on-ramp merging scenarios show that slot-based driving clearly outperforms human drivers as simulated by VISSIM, with respect to the aforementioned performance criteria.

It is important to note that traffic simulators such as VISSIM are known to inaccurately simulate the merging performed by human drivers (Al-Obaedi and Yousif, 2011). In the on-ramp merging case, this can be observed in VISSIM by the fact that on-ramp vehicles patiently wait for an empty gap on the main road, potentially causing a large queue to be formed on the on-ramp (Horowitz et al., 2005). It therefore does not account for aggressive merging from the on-ramp, nor for the cooperation of drivers on the main road, which might slow down or change lanes right before the merging point to accommodate on-ramp vehicles. The experiments performed in this work showed that, for manual driving under high traffic demand on both main road and on-ramp, the free-flow state of the main road is maintained despite a large queue being formed on the on-ramp. In real-world conditions, on a highway without ramp metering, aggressive merging and main road vehicles slowing down could reduce the length of the on-ramp queue. However, this often leads to congestion on the main road, which in turn can have a negative effect on the on-ramp throughput, and lead to delays on both main road and on-ramp. It is therefore difficult to precisely determine how the inaccurate VISSIM simulation affects the throughput and delay results obtained in the evaluation of merging performed by human drivers. To counter this, VISSIM can be calibrated with real-world merging

data (Gomes et al., 2004; Chitturi and Benekohal, 2008). Nevertheless, all this does not affect the accuracy of slot-based driving simulations, since the slot-based driving implementation overrides the human driver model provided by VISSIM. Consequently, a calibration of VISSIM was not performed.

The very high potential of a slot-based IVHS to increase traffic efficiency is demonstrated by the throughput achieved by cooperative slot-based merging, which approaches the theoretical limit of 3600 vehicles/hour for slots generated at a frequency of 1 second. Besides efficiency, this evaluation also showed that the slot-based approach can increase the reliability of traffic by making travel times more predictable. This aspect plays an important role in the context of a global economy that is becoming increasingly reliant on just-in-time production and distribution systems, and has not been addressed by previous work on IVHS.

The slot abstraction splits the task of optimising traffic between the Slot Provider and the individual vehicles. Thus, the Slot Provider is only concerned with generating the slots in which vehicles drive. The implementation of the Slot Provider, presented in Section 4.1.2, shows how simple the slot generation process is. The time required to generate these slots scales linearly with respect to number of lanes on the highway and does depend on the number of vehicles in the network. The generated slot map is queried by individual vehicles when determining their respective target slot. This form of communication between the Slot Provider and each vehicle can have a negative impact on the scalability of the approach. However, this problem is specific to the implementation within VISSIM. A real-world implementation can avoid this scalability problem by having the Slot Provider broadcast the slot map to the vehicles each time the slot map is updated, as specifically designed in Chapter 3. This means that a real-world implementation of the Slot Provider scales linearly with respect to the number of lanes on the highway and the complexity of generating and passing slot information to vehicles does not directly depend on the number of vehicles in the system. Furthermore, for the implementation within VISSIM, this inefficient form of communication does not affect the performance of the system, since the entire simulation is running on a single workstation

and thus the communication between the Slot Provider and individual vehicles is taking place over shared memory. This means that the implementation of the Slot Provider within VISSIM scales well with respect to the size of the road as well as the number of vehicles. This was also confirmed during the evaluation, where even for a traffic demand of over 6000 vehicles per hour across three lanes, the computing resources of this single workstation were enough to run the simulation at a resolution ranging between twenty and forty times real-time.

The VTS algorithm was shown to achieve a higher throughput than human drivers, meaning that slot-based driving can be applied on-the-fly, without negatively impacting traffic, which is key to the flexibility of the whole approach. The ease with which the VTS and LMA algorithms were adapted to work with an extended vehicle model which included HGVs is a further indicator of the flexibility of the slot-based IVHS.

Since the implementation and evaluation were performed within a framework built around a traffic simulator, rather than real-world conditions, some aspects of the slot-based IVHS could not be evaluated. Most importantly, V2I and V2V communication were replaced by shared-memory, which has very low-latency and no packet loss. This means that vehicles in real-world conditions would require more time to coordinate than in simulation, due to the higher latency and packet losses. Additionally, semi-autonomous slot-based driving could not be tested in simulation and would require a real-world evaluation. The simplified vehicle dynamics, sensing and communication models used in the simulation indicate that vehicles should be able to perform slot-based driving, provided they are equipped with accurate sensing, communication and automated vehicle control capabilities.

Finally, one aspect that is not explicitly addressed by the VISSIM implementation is the fault tolerance of the system, the most important part of which is the safety of a vehicle's passengers. This however can easily be achieved by extending the implementation of the Vehicle described in Section 4.1.3 with a Supervisor component. The Supervisor would validate the safety of the low-level driving instructions provided by the Control component and when necessary overwrite these instructions to ensure that

the vehicle does not collide with another vehicle. This would ensure that in the event of a malfunctioning vehicle, surrounding vehicles will perform necessary evasive measures, rather than blindly follow their slots. The risk of two or more vehicles malfunctioning and colliding with each other is independent of slot-based driving and will need to be addressed by manufacturers of automated vehicles. Besides safety, another aspect of a fault tolerant system is maintaining a certain level of performance in the presence of faults. For the slot-based system, this means avoiding a traffic jam and ensuring that each correctly functioning vehicle is able to reach its destination. In the worst case, a total breakdown of traffic can be prevented by gradually reverting to manual driving, leaving the individual drivers in charge of driving. A more optimal solution to this problem, as well as a comprehensive analysis of the system's fault tolerance, are subject to future work.

Chapter 5

Conclusion

This thesis describes a novel approach to optimising freeway traffic using intelligent vehicles. The proposed IVHS is built around the concept of slot-based driving, which provides an abstraction between the traffic management side of an IVHS and the vehicle control. This leads to a hybrid approach that scales well and is flexible, without having to trade-off traffic efficiency and reliability. This chapter summarises the achievements of the work and its contributions to the state of the art, and discusses potential areas for future work. The chapter concludes with an outlook on the future of intelligent vehicles and their impact on everyday life.

5.1 Achievements

An analysis of the currently-deployed freeway management techniques showed that approaches to optimising freeway traffic which are exclusively roadside-based are limited by the nature of human driving. The slow reaction times of human drivers are compensated by large inter-vehicle gaps, leading to a very low utilisation of the road surface. This underutilisation is further amplified by the competitive rather than cooperative nature of human driving. IVHS approaches to optimising freeway traffic address these issues by employing vehicle automation and communication at various degrees. The review of past work on IVHS, performed in Chapter 2, classified these systems based on their

architectural approach as either centralized or decentralized, and argued that centralized IVHSs lack scalability and flexibility, while decentralized IVHSs scale well and are flexible, albeit at the cost of trading-off efficiency. Furthermore, the state-of-the-art review has shown that previous work has not directly addressed the reliability of traffic.

This thesis presented a novel, slot-based approach to optimising traffic using intelligent vehicles. The IVHS addresses the limitations of previous work by employing a hybrid architecture which combines the advantages of centralized and decentralized approaches. The traffic optimisation task is performed in a centralized fashion, while the vehicle control task is performed by each vehicle independently, in a distributed fashion. This leads to an IVHS that is both scalable and efficient. The hybrid architecture is made possible by the slot abstraction. The centralized TMS is responsible for generating slots, which flow along the freeway with a predefined behaviour. The combined effect of all the slot behaviours is one in which slots never overlap and flow freely (i.e. without congestion). On the other side of the slot abstraction, vehicles are responsible for driving within these slots. This results in a mapping of vehicles to slots, performed by the VTS algorithm, and, consequently, in a safe and free flow of vehicles. The way in which slot-based driving is performed by individual vehicles, as well as a set of heuristics used to generate slot maps for scenarios which replicated the most common causes of congestion, were described in Chapter 3.

The evaluation of the slot-based approach to optimising traffic was described in Chapter 4. First, an evaluation framework was built by extending VSSIM, a well-established traffic simulator, with the ability to simulate slot-based driving. Then, to showcase the flexibility of the approach, the vehicle model used in the simulations was extended to include a percentage of HGVs. Within this framework, the efficiency of the transition of vehicles from manual to slot-based driving was evaluated with respect to the throughput of vehicles achieved. The results showed that this transition is efficient and does not cause a bottleneck. Furthermore, the proposed heuristics were evaluated in scenarios which replicate the most common causes of congestion. The results in the respective scenarios showed that slot-based driving clearly outperformed human driving,

both with respect to efficiency and reliability. This indicates that the slot-based approach can be used to increase traffic efficiency and reliability.

The main contributions of this thesis are summarised as:

- An overview of state of practice freeway traffic management techniques and a discussion of their fundamental limitations, followed by an overview of past IVHS approaches to managing freeway traffic and their enabling technologies. Particular attention is paid to the ability of the latter to increase traffic efficiency and reliability, as well as their scalability and flexibility.
- A novel, slot-based IVHS, which addresses the limitations of previous work by providing a scalable and flexible system without sacrificing traffic efficiency and reliability.
- An algorithm that performs an efficient transition from manual to slot-based driving, as well as a set of mechanisms that enable slot-based driving, such as procedures for changing and swapping slots.
- Heuristics for generating slot maps that address the most prominent causes of congestion.
- The design and implementation of a framework for evaluating slot-based heuristics within VISSIM, a widely-used microscopic traffic simulator.
- An evaluation of the vehicle-to-slot transition algorithm as well as the proposed heuristics, which showed that it is possible to significantly improve traffic efficiency and reliability using the slot-based approach.

5.2 Future Work

In the development of the proposed slot-based IVHS, a number of issues were identified that would be suitable for further investigation. This section outlines the key areas identified for future work. This work relates to: generalising the approach to a system model

that accurately resembles real-world conditions; extending the list of heuristics to cover an exhaustive set of scenarios; performing slot-based driving in real-world conditions; and finally, addressing fault tolerance issues.

5.2.1 System Model

The problem of optimising freeway traffic is made difficult by the complexity of the model under study. A combination of factors contribute to this complexity such as the lack of homogeneity with respect to the characteristics of the vehicles, the multitude of parameters that characterise the road model, as well as the impact of weather conditions on traffic. The approach used in this thesis was to reduce the complexity of the problem by idealising the system model, with the goal of finding an initial solution, which can then be generalised.

In Chapter 4, a first step towards this generalisation was made by extending the vehicle model to account for larger vehicles and adapting the VTS and LMA algorithms accordingly. This initial work needs to be continued by extending the vehicle model to describe a heterogeneous set of vehicles, with different geometries and accurate dynamics. In particular, one of the assumptions of the designed IVHS was that vehicles are equipped with sensing and actuating capabilities that enable them to perform slot-based driving. This means that, in its current form, a slot-based IVHS can only be deployed if all vehicles on the managed freeway are capable of performing slot-based driving. While the sensing and actuating requirements of slot-based driving can be fulfilled using already deployed vehicular technologies, future work should investigate how the proposed slot-based IVHS can be adapted to allow the coexistence of slot-based driving and fully manual driving (e.g. using dedicated lanes), and the implications this would have on the overall efficiency and reliability of the system.

The freeway model used in this work needs to be extended to account for curvature, gradients, as well as more complex layouts such as weaving segments. In particular, the gradient of a road can have an impact on ability of a HGV to comply to the speed requirements imposed by slot-based driving. Future work can also investigate if the

concept of slot-based driving can be applied to urban roads. For example, the task of optimising traffic at intersections can be approached by forcing vehicles to move into slots which cross the intersection in a predefined safe and efficient manner. Finally, future work needs to consider the impact of weather on slot-based driving. In particular, inclement weather conditions, such as snow-covered or icy roads, can force vehicles to reduce their speed, and as such will need to be considered when slot behaviours are generated.

5.2.2 Heuristics

The list of heuristics for generating slot maps presented in this thesis address the most common causes of congestion. Future work will need to extend this list so that it covers an exhaustive set of scenarios that can occur on freeways. The rule-based approach used by the Slot Providers to match a heuristic to the current and predicted traffic conditions was mainly chosen for its simplicity and relies on the existence of a heuristic for each possible scenario. Future work can investigate the use of an alternative, learning-based approach to generating slot maps, which could replace this rule-based system. Such an approach would lead to a more adaptable and flexible system and would require less human intervention.

5.2.3 Slot-based Driving in Real-world Conditions

The logistic requirements involved evaluating the slot-based IVHS in real-world conditions could not be fulfilled within the scope of this work. Instead, the evaluation was performed within a traffic simulation. This means that aspects such as communication reliability and security, sensor accuracy, as well as low-level vehicle control and semi-autonomous driving could not be properly evaluated. To demonstrate the real-world feasibility of slot-based driving, future work will need to perform experiments in real-world conditions. Because evaluating traffic efficiency requires a large number of vehicles, a hardware-in-the-loop (HIL) technique could be used to integrate one or more vehicles driving in real-world conditions with vehicles in a simulation. For this purpose, we have

already started the development of a HIL simulator which integrates VISSIM, a network simulator called OPNET and car-like robots driving in real-world conditions (O'Hara et al., 2012).

5.2.4 Fault Tolerance

Finally, future work will need to address issues such as malfunctioning vehicles or drivers which do not comply with the rules imposed by slot-based driving. This requires the development of fault tolerance mechanisms which can guarantee the safety of vehicles and ensure an efficient use of the infrastructure in the face of malfunctions. Vehicles will need to be equipped with a safety controller capable of superseding the Slot Driving Logic when their safety is at risk. Slot Providers will need to be able to adapt their slot maps so that they can maintain a high level of performance with respect to traffic efficiency and reliability. In worst-case situations, such as an accident caused by a malfunctioning vehicle which is blocking the entire freeway, slot-based driving will need to be disabled.

5.3 Outlook

The work presented in this thesis has shown the potential of intelligent vehicles to optimise freeway traffic using the slot-based approach. Besides freeway traffic optimisation, there are many other areas in which intelligent, automated vehicles can have an impact. Increasing safety has been the main driving force behind vehicle automation. Drastically reducing the number of accidents and fatalities will have a considerable socio-economic impact. Additionally, this will also have an impact on the insurance business, as insurance premiums will likely drop. However, rare as they will likely be, the occurrence of accidents caused by automated vehicles will also raise the issue of determining who is liable: the driver, or the vehicle's manufacturer.

In countries with rich economies, it is common among middle-class families to own more than one vehicle. However, many of these vehicles spend most of their time unused. A self-driving car could replace the need for additional cars by serving each family member's individual transport needs. For the car-rental industry, car-sharing could

become a better suited business model, as self-driving cars could collect passengers from their location and would not need to be dropped-off at specific locations. Additionally, the demand for parking spaces within cities would likely drop, with cars capable of autonomously driving back home or to designated parking spaces at the outskirts of cities. All this would improve the quality of life in cities by reducing pollution, noise and stress caused by traffic congestion.

While the technical feasibility of automated driving has been demonstrated by Google, as well as vehicle manufacturers such as BMW (Carfrae, 2011), Audi (Squatriglia, 2010), VW (Holling, 2011), GM (Foley et al., 2010) and Toyota (BBC, 2013), the adoption of intelligent vehicles will require the support of governments and legislators. Legislation will need to be passed which allows automated driving¹. Furthermore, the governments will need to support the costs of deploying roadside infrastructure required by an IVHS. While the long term benefits will likely outweigh the costs of deploying IVHSs, the costs of owning an intelligent car might be, at least initially, too high for the majority of the population. A cost-benefit analysis will need to determine if these costs can and should be subsidised.

In conclusion, the recent advances in automation, sensing and communication have made intelligent vehicles technologically feasible. The adoption of intelligent vehicles and automated driving will increase safety, efficiency and comfort. Furthermore, as shown in this work, intelligent vehicles can be used to significantly increase the efficiency and reliability of road traffic. As the demand for transportation increases, these benefits will likely outweigh the costs associated with a large-scale deployment of intelligent vehicles and the infrastructure required to implement an IVHS, such as the one presented in this work.

¹The US states of Nevada and California have already passed such legislation

Appendix A

Implementation of DriverModel

Listing A.1: Extract from DriverModel.cs showcasing the get and set functions

```
1 public class DriverModel : Constants, IDriverModel
2 {
3     private SlotProvider _slotProvider;
4     private Dictionary<int, Vehicle> _vehicles; // All the vehicles in
           the simulation
5     private Vehicle _activeVehicle;
6     private NeighbourVehicle _activeNeighbourVehicle;
7     private double _timeStep; // Simulator time step configuration
8     private double _time;
9
10    /* Sets the value of a data object of type <type>, selected by <index1
           > */
11    /* and possibly <index2>, to <int_value>, <double_value> or
           */
12    /* <string_value> (object and value selection depending on <type>).
           */
13    /* Return value is 1 on success, otherwise 0.
           */
14    public int DriverModelSetValue(int type,
15                                   int index1,
16                                   int index2,
17                                   int long_value,
18                                   double double_value,
19                                   string string_value)
20    {
21        switch( type )
22        {
23            case DRIVER_DATA_TIMESTEP:
24                _timeStep = double_value;
25                return 1;
26            case DRIVER_DATA_TIME:
27                _time = double_value;
28                if(_slotProvider!=null) _slotProvider.Step( _time );
29                return 1;
30            case DRIVER_DATA_VEH_ID:
```

```

31         if( !_vehicles.TryGetValue( long_value, out _activeVehicle
32             ) )
33         {
34             _activeVehicle = new Vehicle();
35             _activeVehicle.Id = long_value;
36             _vehicles[_activeVehicle.Id] = _activeVehicle;
37         }
38         _activeVehicle.Neighbours.Clear();
39         return 1;
40     case DRIVER_DATA_VEH_LANE:
41         _activeVehicle.CurrentLane = long_value;
42         return 1;
43     case DRIVER_DATA_VEH_VELOCITY:
44         _activeVehicle.Velocity = double_value;
45         return 1;
46     case DRIVER_DATA_VEH_ACCELERATION:
47         _activeVehicle.Acceleration = double_value;
48         return 1;
49     case DRIVER_DATA_VEH_X_COORDINATE:
50         _activeVehicle.XCoordinate = double_value;
51         return 1;
52     case DRIVER_DATA_VEH_Y_COORDINATE:
53         _activeVehicle.YCoordinate = double_value;
54         return 1;
55     case DRIVER_DATA_NVEH_ID:
56         _activeNeighbourVehicle = new NeighbourVehicle();
57         _activeNeighbourVehicle.Id = long_value;
58         if( _activeNeighbourVehicle.Id >= 0 )
59         {
60             _activeVehicle.Neighbours.Add( _activeNeighbourVehicle
61             );
62             _activeNeighbourVehicle.RelativeLane = index1;
63             _activeNeighbourVehicle.RelativePosition = index2;
64             return 1;
65     case DRIVER_DATA_NVEH_LATERAL_POSITION:
66         _activeNeighbourVehicle.LateralPosition = double_value;
67         return 1;
68     case DRIVER_DATA_NVEH_DISTANCE:
69         _activeNeighbourVehicle.Distance = double_value;
70         return 1;
71     case DRIVER_DATA_NVEH_REL_VELOCITY:
72         _activeNeighbourVehicle.RelativeVelocity = double_value;
73         return 1;
74     case DRIVER_DATA_NVEH_ACCELERATION:
75         _activeNeighbourVehicle.Acceleration = double_value;
76         return 1;
77     }
78 }
79 /* Gets the value of a data object of type <type>, selected by <index1
80 > */
81 /* and possibly <index2>, and writes that value to <*double_value>,
82 */
83 /* <*float_value> or <***string_value> (object and value selection
84 */
85 /* depending on <type>). */
86 */
87 /* Return value is 1 on success, otherwise 0. */

```

```
84         */
85     public int DriverModelGetValue(int type,
86                                   int index1,
87                                   int index2,
88                                   out int long_value,
89                                   out double double_value,
90                                   out string string_value)
91     {
92         switch( type )
93         {
94             case DRIVER_DATA_VEH_DESIRED_VELOCITY:
95                 double_value = _activeVehicle.DesiredVelocity;
96                 retVal = 1;
97                 break;
98             case DRIVER_DATA_DESIRED_ACCELERATION:
99                 double_value = _activeVehicle.DesiredAcceleration;
100                retVal = 1;
101                break;
102             case DRIVER_DATA_REL_TARGET_LANE:
103                 long_value = _activeVehicle.RelativeTargetLane;
104                 retVal = 1;
105                 break;
106         }
107         return retVal;
108     }
```


Bibliography

Abou-Jaoude, R. (2003), ‘ACC radar sensor technology, test requirements, and test solutions’, *Intelligent Transportation Systems, IEEE Transactions on* **4**(3), 115–122.

URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1255573

Al-Obaedi, J. and Yousif, S. (2011), ‘Microsimulation Model for Motorway Merges With Ramp-Metering Controls’, *IEEE Transactions on Intelligent Transportation Systems* **PP**(99), 1–11.

URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6060913

Alvarez, L. and Horowitz, R. (1997), Traffic Flow Control In Automated Highway Systems, Technical report, California PATH.

URL: <http://escholarship.org/uc/item/4m56q4s1>

Bacon, J., Bejan, A., Beresford, A., Evans, D., Gibbens, R. and Moody, K. (2011), Using Real-Time Road Traffic Data to Evaluate Congestion, in C. B. Jones and J. L. Lloyd, eds, ‘Dependable and Historic Computing’, Vol. 6875 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 93–117.

URL: <http://www.springerlink.com/content/7734344r61736556/>

Baskar, L. D., De Schutter, B. and Hellendoorn, H. (2012), ‘Traffic Management for Automated Highway Systems Using Model-Based Predictive Control’, *IEEE Transactions on Intelligent Transportation Systems* **13**(2), 838–847.

URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6157625>

Baskar, L. D., De Schutter, B., Hellendoorn, J. and Papp, Z. (2011), ‘Traffic control

- and intelligent vehicle highway systems: a survey', *IET Intelligent Transport Systems* **5**(1), 38.
URL: <http://link.aip.org/link/IITSAN/v5/i1/p38/s1&Agg=doi>
- Baskar, L., De Schutter, B. and Hellendoorn, H. (2007), Hierarchical Traffic Control and Management with Intelligent Vehicles, in 'Intelligent Vehicles Symposium, 2007 IEEE', pp. 834–839.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4290220
- Baskar, L., De Schutter, B. and Hellendoorn, H. (2008), Model-based predictive traffic control for intelligent vehicles: Dynamic speed limits and dynamic lane allocation, in 'Intelligent Vehicles Symposium, 2008 IEEE', pp. 174–179.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4621307
- BBC (2009), 'The man who gave us traffic lights'.
URL: <http://goo.gl/m7OeY> (accessed 28/04/2013)
- BBC (2013), 'Toyota sneak previews self-drive car ahead of tech show'.
URL: <http://www.bbc.co.uk/news/technology-20910769> (accessed 27/04/13)
- Bishop, R. (2005), *Intelligent Vehicle Technology And Trends*, Artech House.
URL: <http://amzn.com/1580539114>
- Bloomberg, L. and Dale, J. (2000), 'Comparison of vissim and corsim traffic simulation models on a congested network', *Transportation Research Record: Journal of the Transportation Research Board* **1727**(1), 52–60.
- Boriboonsomsin, K., Servin, O. and Barth, M. (2008), 'A Preliminary Design of Speed Control Strategies in Dynamic Intelligent Speed Adaptation System for Freeways', *Transportation Research* (January), 17.
- Bouroche, M., Hughes, B. and Cahill, V. (2006), Real-time coordination of autonomous vehicles, in '2006 IEEE Intelligent Transportation Systems Conference', IEEE, pp. 1232–1239.
URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1707391>

- Brandon, J. (2012), 'Top 5 HUDs in modern cars today'.
URL: <http://goo.gl/g1KrS> (accessed 28/04/13)
- Broqua, F., Lerner, G., Mauro, V. and Morello, E. (1991), Cooperative driving: basic concepts and a first assessment of " intelligent cruise control" strategies, in 'Advanced Telematics in Road Transport, DRIVE Conference', Brussels, pp. 908–929.
URL: <http://trid.trb.org/view.aspx?id=650294>
- Cahill, V., Senart, A., Schmidt, D. C., Weber, S., Harrington, A. and Hughes, B. (2008), The managed motorway: real-time vehicle scheduling, in 'Proceedings of the 9th workshop on Mobile computing systems and applications - HotMobile '08', ACM Press, New York, New York, USA, p. 43.
URL: <http://dl.acm.org/citation.cfm?id=1411759.1411771>
- California Department of Transportation (2011), Ramp metering development plan, Technical Report December.
- California, LAO (2000), 'HOV Lanes in California: Are They Achieving Their Goals?'.
URL: http://www.lao.ca.gov/2000/010700_hov/010700_hov_lanes.html
- Carfrae, J. (2011), 'An automated adventure at the wheel of a driverless BMW'.
URL: <http://goo.gl/lvXUx> (accessed 27/04/13)
- Carsten, O. M. J. and Tate, F. N. (2005), 'Intelligent speed adaptation: accident savings and costbenefit analysis', *Accident Analysis Prevention* **37**(3), 407–416.
URL: <http://dx.doi.org/10.1016/j.aap.2004.02.007>
- Charette, R. (2009), 'This Car Runs on Code'.
URL: <http://spectrum.ieee.org/green-tech/advanced-cars/this-car-runs-on-code> (accessed 11/09/12)
- Chitturi, M. V. and Benekohal, R. F. (2008), Calibration of vissim for freeways, in 'Transportation Research Board 87th Annual Meeting', number 08-2317.

- Dao, T.-S., Clark, C. and Huissoon, J. (2007), Optimized Lane Assignment Using Inter-Vehicle Communication, in ‘Intelligent Vehicles Symposium, 2007 IEEE’, pp. 1217–1222.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4290284
- Dao, T.-S., Clark, C. and Huissoon, J. (2008), Distributed platoon assignment and lane selection for traffic flow optimization, in ‘Intelligent Vehicles Symposium, 2008 IEEE’, pp. 739–744.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4621202
- Davis, L. (2007), ‘Effect of adaptive cruise control systems on mixed traffic flow near an on-ramp’, *Physica A: Statistical Mechanics and its Applications* **379**(1), 274–290.
URL: <http://www.sciencedirect.com/science/article/pii/S0378437106013690>
- European Commission (2003), Europe at a crossroads: The need for sustainable transport, Technical report.
URL: <http://goo.gl/hQjDF> (accessed 28/04/13)
- European Commission (2006), On the Intelligent Car Initiative “Raising Awareness of ICT for Smarter, Safer and Cleaner Vehicles”, Technical report.
URL: <http://goo.gl/W4DuM> (accessed 28/04/13)
- European Commission (2007), Towards Europe-wide Safer, Cleaner and Efficient Mobility: The First Intelligent Car Report, Technical report.
URL: <http://goo.gl/hE1oO> (accessed 28/04/13)
- Federal Highway Authority (2012), ‘Operations Story’.
URL: <http://www.ops.fhwa.dot.gov/aboutus/opstory.htm> (accessed 16/01/13)
- Fellendorf, M. and Vortisch, P. (2001), Validation of the microscopic traffic flow model vissim in different real-world situations, in ‘National Research Council (US). Transportation Research Board. Meeting (80th: 2001: Washington, DC). Preprint CD-ROM’.

- Figueiredo, L., Jesus, I., Machado, J., Ferreira, J. and Martins de Carvalho, J. (2001), Towards the development of intelligent transportation systems, in 'Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE', pp. 1206–1211.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=948835
- Fleming, W. (2001), 'Overview of automotive sensors', *Sensors Journal, IEEE* **1**(4), 296–308.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=983469
- Foley, H., Albano, M. and Flores, D. (2010), 'GM Unveils EN-V Concept: A Vision for Future Urban Mobility'.
URL: <http://goo.gl/kn8GM> (accessed 27/04/13)
- Frese, C., Batz, T., Wieser, M. and Beyerer, J. (2008), Life cycle management for cooperative groups of cognitive automobiles in a distributed environment, in 'Intelligent Vehicles Symposium, 2008 IEEE', pp. 1125–1130.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4621206
- Frese, C. and Beyerer, J. (2011), A comparison of motion planning algorithms for cooperative collision avoidance of multiple cognitive automobiles, in '2011 IEEE Intelligent Vehicles Symposium (IV)', IEEE, pp. 1156–1162.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5940489
- Frese, C., Beyerer, J. and Zimmer, P. (2007), Cooperation of Cars and Formation of Cooperative Groups, in 'Intelligent Vehicles Symposium, 2007 IEEE', pp. 227–232.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4290119
- Gauthier, L., Michel, P., Ventura-Traveset, J. and Benedicto, J. (2001), 'EGNOS - The first step in Europe's contribution to the Global Navigation Satellite System', *ESA Bulletin* (105), 35–42.
URL: http://www.egnos-pro.esa.int/Publications/ESA_Bulletin105.pdf
- Gomes, G., May, A. and Horowitz, R. (2004), 'Congested freeway microsimulation model

- using vissim', *Transportation Research Record: Journal of the Transportation Research Board* **1876**(1), 71–81.
- Gray, J. and Reuter, A. (1992), *Transaction Processing: Concepts and Techniques*.
URL: <http://dl.acm.org/citation.cfm?id=573304>
- Hallé, S. and Chaib-draa, B. (2005), Collaborative Driving System Using Teamwork for Platoon Formations, in 'Applications of Agent Technology in Traffic and Transportation', pp. 133–151.
URL: <http://www.springerlink.com/content/r455526v3v134742>
- Halle, S., Laumonier, J. and Chaib-Draa, B. (2004), A decentralized approach to collaborative driving coordination, in 'Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on', pp. 453–458.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1398942
- Hartenstein, H. and Laberteaux, K., eds (2010), *VANET Vehicular Applications and Inter-Networking Technologies (Intelligent Transport Systems)*, Wiley.
URL: <http://amzn.com/0470740566>
- Hegarty, C. and Chatre, E. (2008), 'Evolution of the Global Navigation Satellite System (GNSS)', *Proceedings of the IEEE* **96**(12), 1902–1917.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4745647
- Hegyí, A., DeSchutter, B. and Hellendoorn, J. (2005), 'Optimal Coordination of Variable Speed Limits to Suppress Shock Waves', *IEEE Transactions on Intelligent Transportation Systems* **6**(1), 102–112.
URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1402434>
- Herbert, W. and Mili, F. (2008), Route guidance: State of the art vs. state of the practice, in 'Intelligent Vehicles Symposium, 2008 IEEE', pp. 1167–1174.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4621289

- Hidas, P. (2002), 'Modelling lane changing and merging in microscopic traffic simulation', *Transportation Research Part C: Emerging Technologies* **10**(56), 351 – 371.
URL: <http://www.sciencedirect.com/science/article/pii/S0968090X02000268>
- Highway Capacity Manual (2012), 'Traffic characteristics', *Proteomics. Clinical applications* **6**(3-4), NA.
URL: <http://www.ncbi.nlm.nih.gov/pubmed/22539764>
- Holling, R. (2011), 'Driving without a driver - Volkswagen presents the Temporary Auto Pilot'.
URL: <http://goo.gl/wOvms> (accessed 27/04/13)
- Hoogendoorn, S. P. and Bovy, P. H. L. (2001), 'State-of-the-art of Vehicular Traffic Flow Modelling', *Journal of Systems and Control Engineering* **215**(Special Issue on Road Traffic Modelling and Control), 283 – 303.
URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.106.9913>
- Horowitz, R., May, A., Skabardonis, A., Varaiya, P., Zhang, M., Gomes, G., Muñoz, L. and Sun, X. (2005), Design, Field Implementation and Evaluation of Adaptive Ramp Metering Algorithms, Technical Report January, California PATH.
- Horowitz, R. and Varaiya, P. (2000), 'Control design of an automated highway system', *Proceedings of the IEEE* **88**(7), 913–925.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=871301
- Japanese Ministry of Land, Infrastructure, T. and Tourism (2008), 'ITS: A Collection of Effectiveness Case Studies: 2007-2008', pp. 2007–2008.
- Joint Transport Research Center of OECD and ITF (2010), *Improving reliability on surface transport networks*, OECD Publishing.
URL: <http://dx.doi.org/10.1787/9789282102428-en>
- Jones, W. (2001), 'Keeping cars from crashing', *Spectrum, IEEE* **38**(9), 40–45.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=946636

- Kachroo, P., Li, Z., Control, K. O. and Mode, S. (1997), Vehicle merging control design for an automated highway, in 'IEEE Intelligent Transportation Systems Conference', number 1, pp. 224–229.
- Karagiannis, G., Altintas, O., Ekici, E., Heijenk, G., Jarupan, B., Lin, K. and Weil, T. (2011), 'Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions', *IEEE Communications Surveys & Tutorials* **13**(4), 584–616.
URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5948952>
- Kato, S., Tsugawa, S., Tokuda, K., Matsui, T. and Fujii, H. (2002), 'Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications', *Intelligent Transportation Systems, IEEE Transactions on* **3**(3), 155–161.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1033758
- Kenney, J. B. (2011), 'Dedicated Short-Range Communications (DSRC) Standards in the United States', *Proceedings of the IEEE* **99**(7), 1162–1182.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5888501
- Kesting, A., Treiber, M. and Kranke, F. (2007), 'Jam-avoiding adaptive cruise control (ACC) and its impact on traffic dynamics', *Traffic and Granular .*
URL: <http://www.springerlink.com/index/p56714553g82r2h7.pdf>
- Kesting, A., Treiber, M., Schönhof, M. and Helbing, D. (2008), 'Adaptive cruise control design for active congestion avoidance', *Transportation Research Part C: Emerging Technologies* **16**(6), 668–683.
URL: <http://dx.doi.org/10.1016/j.trc.2007.12.004>
- Khan, M. and Boloni, L. (2005), Convoy driving through ad-hoc coalition formation, in 'Real Time and Embedded Technology and Applications Symposium, 2005. RTAS 2005. 11th IEEE', pp. 98–105.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1388377

Klein, L. A., Mills, M. K. and Gibson, D. R. P. (2006), 'Traffic Detector Handbook', *Operations Research I*(October), 462.

URL: <http://trid.trb.org/view.aspx?id=349412>

Lenain, R., Thuilot, B., Cariou, C. and Martinet, P. (2003), Adaptive control for car like vehicles guidance relying on RTK GPS: rejection of sliding effects in agricultural applications, in '2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)', Vol. 1, IEEE, pp. 115–120.

URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1241582>

Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S., Koch, O., Kuwata, Y., Moore, D., Olson, E., Peters, S., Teo, J., Truax, R., Walter, M., Barrett, D., Epstein, A., Maheloni, K., Moyer, K., Jones, T., Buckley, R., Antone, M., Galejs, R., Krishnamurthy, S. and Williams, J. (2008), 'A perception-driven autonomous urban vehicle', *J. Field Robot.* **25**(10), 727–774.

URL: <http://portal.acm.org/citation.cfm?id=1455709>

Lin, P.-W., Kang, K.-P. and Chang, G.-L. (2004), 'Exploring the effectiveness of variable speed limit controls on highway work-zone operations', *Journal of Intelligent Transportation Systems* **8**(3), 155–167.

URL: <http://trid.trb.org/view.aspx?id=742189>

Lint, J. (2004), Reliable Travel Time Prediction for Freeways, PhD thesis, Delft Univ. Technol.

Liu, R. and Tate, J. E. (2004), 'Network effects of intelligent speed adaptation systems', *Transportation* **31**(3), 297–325.

URL: <http://link.springer.com/article/10.1023%2FB%3APORT.0000025394.78857.13>

Lu, X.-Y., Tan, H.-S., Shladover, S. E. and Hedrick, J. K. (2004), 'Automated Vehicle Merging Maneuver Implementation for AHS', *Vehicle System Dynamics* **41**(2), 85–

107.

URL: <http://www.tandfonline.com/doi/abs/10.1076/vesd.41.2.85.26497>

Maciuca, D. B. and Hedrick, K. J. (1995), Brake Dynamics Effect on Ahs Lane Capacity, *in* 'Future Transportation Technology Conference and Exposition'.

MacNeill, P. and Miller, R. (2003), A new technology for a cruise control system, *in* 'Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th', Vol. 5, pp. 3341–3345 Vol.5.

URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1286300

Marsden, G., McDonald, M. and Brackstone, M. (2001), 'Towards an understanding of adaptive cruise control', *Transportation Research Part C: Emerging Technologies* **9**(1), 33–51.

URL: <http://www.sciencedirect.com/science/article/pii/S0968090X0000022X>

Masher, D. P., Ross, D., Wong, P., Tuan, P., Zeidler, H. and Petracek, S. (1975), Guidelines for design and operation of ramp control systems, Technical report.

Matolak, D., Wu, Q., Sanchez-Sanchez, J., Morales-Jimenez, D. and Aguayo-Torres, M. (2011), Performance of lte in vehicle-to-vehicle channels, *in* 'Vehicular Technology Conference (VTC Fall), 2011 IEEE', pp. 1–4.

Michaud, F., Lepage, P., Frenette, P., Letourneau, D. and Gaubert, N. (2006), 'Coordinated Maneuvering of Automated Vehicles in Platoons', *Intelligent Transportation Systems, IEEE Transactions on* **7**(4), 437–447.

URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4019445

Milanés, V., Godoy, J., Villagrà, J. and Pérez, J. (2011), 'Automated On-Ramp Merging System for Congested Traffic Situations', *IEEE Transactions on Intelligent Transportation Systems* **12**(2), 500–508.

Milanes, V., Villagra, J., Godoy, J., Simo, J., Perez, J. and Onieva, E. (2012), 'An Intelligent V2I-Based Traffic Management System', *IEEE Transactions on Intelligent*

- Transportation Systems* **13**(1), 49–58.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6121906
- Mincheva, E. (2009), ‘Intelligent vehicle clusters (Inside Com2React project)’, *Intelligent Transport Systems, IET* **3**(2), 206–215.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5071790
- Minderhoud, M. and Bovy, P. (1999), Impact of intelligent cruise control on motorway capacity, in ‘78th Annual Meeting of the Transportation Research Board’, pp. 1–9.
- Mitchell, M. (2009), *Complexity: a guided tour*, Oxford University Press, USA.
- Moen, B., Fitts, J., Carter, D. and Ouyang, Y. (2000), ‘A comparison of the vissim model to other widely used traffic simulation and analysis programs’, *Institute of Transportation Engineers*.
- Monteiro, L. S., Moore, T. and Hill, C. (2005), ‘What is the accuracy of DGPS?’, *Journal of Navigation* **58**(02), 207–225.
URL: http://journals.cambridge.org/abstract_S037346330500322X
- Morla, R. (2005), Vision of Congestion-Free Road Traffic and Cooperating Objects, Technical report.
- Naranjo, J., Gonzalez, C., Garcia, R. and de Pedro, T. (2008), ‘Lane-Change Fuzzy Control in Autonomous Vehicles for the Overtaking Maneuver’, *Intelligent Transportation Systems, IEEE Transactions on* **9**(3), 438–450.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4515884
- Oagana, A. (2010), ‘Lane Keeping Assist Systems Explained’.
URL: <http://www.autoevolution.com/news/lane-keeping-assist-systems-explained-25459.html> (accessed 01/04/13)
- O’Hara, N., Slot, M., Marinescu, D., Čurn, J., Yang, D., Asplund, M., Bourouche, M., Clarke, S. and Cahill, V. (2012), Mddsvsim: an integrated traffic simulation platform

- for autonomous vehicle research, in ‘The First International Workshop on Vehicular Traffic Management for Smart Cities (VTM)’, IEEE Computer Society.
- Papageorgiou, M., Hadj-Salem, H. and Blosseville, J.-M. (1991), ‘ALINEA: a local feedback control law for on-ramp metering’, *Transportation Research Record* (1320).
URL: <http://trid.trb.org/view.aspx?id=365587>
- Papageorgiou, M., Kiakaki, C., Dinopoulou, V. and Kotsialos, A. (2003), ‘Review of road traffic control strategies’, *Proceedings of the IEEE* **91**(12), 2043–2067.
URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1246386>
- Papageorgiou, M. and Kotsialos, A. (2000), Freeway ramp metering: an overview, in ‘Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE’, pp. 228–239.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=881058
- Papageorgiou, M., Papamichail, I., Spiliopoulou, A. D. and Lentzakis, A. F. (2008), ‘Real-time merging traffic control with applications to toll plaza and work zone management’, *Transportation Research Part C: Emerging Technologies* **16**(5), 535–553.
URL: <http://www.sciencedirect.com/science/article/pii/S0968090X07000873>
- Papamichail, I., Papageorgiou, M., Chrysoulakis, I., Gaffney, J. and Vogn, V. (2010), HERO Coordinated Ramp Metering Implemented at the Monash Freeway, in ‘89th Transp. Res. Board Annu. Meeting’.
URL: [http://2isfo.eng.hawaii.edu/Presentations/Session 18/18 - Papamichail.pdf](http://2isfo.eng.hawaii.edu/Presentations/Session%2018/18-Papamichail.pdf)
- Papp, Z., Brown, C. and Bartels, C. (2008), World modeling for cooperative intelligent vehicles, in ‘Intelligent Vehicles Symposium, 2008 IEEE’, pp. 1050–1055.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4621272
- Park, B. B. and Schneeberger, J. (2003), ‘Microscopic simulation model calibration and validation: case study of vissim simulation model for a coordinated actuated signal system’, *Transportation Research Record: Journal of the Transportation Research Board* **1856**(1), 185–192.

- Parkany, E. and Xie, C. (2005), A complete review of incident detection algorithms & their deployment: what works and what doesn't, Technical report.
URL: <http://trid.trb.org/view.aspx?id=754914>
- Pérez, J., Milanés, V., Godoy, J., Villagrà, J. and Onieva, E. (2013), 'Cooperative controllers for highways based on human experience', *Expert Systems with Applications* **40**(4), 1024–1033.
URL: <http://dx.doi.org/10.1016/j.eswa.2012.08.011>
- Ploeg, J., Scheepers, B. T. M., van Nunen, E., van de Wouw, N. and Nijmeijer, H. (2011), Design and experimental evaluation of cooperative adaptive cruise control, in '2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)', IEEE, pp. 260–265.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6082981
- Rajamani, R. (2000), 'Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons', *IEEE Transactions on Control Systems Technology* **8**(4), 695–708.
URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=852914>
- Ravi, N., Smaldone, S., Iftode, L. and Gerla, M. (2007), Lane Reservation for Highways (Position Paper), in 'Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE', pp. 795–800.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4357746
- Raya, M. and Hubaux, J. (2007), 'Securing vehicular ad hoc networks', *Journal of Computer Security* **15**(1), 39–68.
- Rosen, R. (2012), 'Google's Self-Driving Cars: 300,000 Miles Logged, Not a Single Accident Under Computer Control', *The Atlantic* .
URL: <http://goo.gl/x9ZBQ>
- Sengupta, R., Rezaei, S., Shladover, S. E., Cody, D., Dickey, S. and Krishnan, H. (2007),

- ‘Cooperative Collision Warning Systems: Concept Definition and Experimental Implementation’, *Journal of Intelligent Transportation Systems* **11**(3), 143–155.
- Shehata, M. S., Cai, J., Badawy, W. M., Burr, T. W., Pervez, M. S., Johannesson, R. J. and Radmanesh, A. (2008), ‘Video-Based Automatic Incident Detection for Smart Roads: The Outdoor Environmental Challenges Regarding False Alarms’, *IEEE Transactions on Intelligent Transportation Systems* **9**(2), 349–360.
URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4538012>
- Shladover, S. (2007), ‘PATH at 20—History and Major Milestones’, *Intelligent Transportation Systems, IEEE Transactions on* **8**(4), 584–592.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4358931
- Shladover, S. (2009), ‘Cooperative (rather than autonomous) vehicle-highway automation systems’, *Intelligent Transportation Systems Magazine, IEEE* **1**(1), 10–19.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5117654
- Sichitiu, M. and Kihl, M. (2008), ‘Inter-vehicle communication systems: a survey’, *Communications Surveys & Tutorials, IEEE* **10**(2), 88–105.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4564481
- Slot, M. and Cahill, V. (2011), A reliable membership service for vehicular safety applications, in ‘Intelligent Vehicles Symposium (IV), 2011’, pp. 1163–1169.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5940487
- Squatriglia, C. (2010), ‘Audis Robotic Car Climbs Pikes Peak’.
URL: <http://goo.gl/smPmP> (accessed 27/04/13)
- Stiller, C., Farber, G. and Kammel, S. (2007), Cooperative Cognitive Automobiles, in ‘Intelligent Vehicles Symposium, 2007 IEEE’, pp. 215–220.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4290117
- Swaroop, D. and Yoon, S. M. (1999), The Design of a Controller for a Following Vehicle in an Emergency Lane Change Maneuver, Technical report, California PATH, University

- of California, Berkeley.
URL: <http://escholarship.org/uc/item/20v299zx>
- Tan, H.-S. and Huang, J. (2006), ‘DGPS-Based Vehicle-to-Vehicle Cooperative Collision Warning: Engineering Feasibility Viewpoints’, *Intelligent Transportation Systems, IEEE Transactions on* **7**(4), 415–428.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4019449
- Thorne, K. (1994), *Black holes and time warps: Einstein’s outrageous legacy*, WW Norton & Company.
- Thurn, S. (2002), *Robotic Mapping: A Survey*, Morgan Kaufmann Publishers Inc.
- Thurn, S. (2010), ‘What are we driving at’.
URL: <http://goo.gl/1HLAZ> (accessed 28/04/2013)
- Toor, Y., Muhlethaler, P. and Laouiti, A. (2008), ‘Vehicle Ad Hoc networks: applications and related technical issues’, *Communications Surveys & Tutorials, IEEE* **10**(3), 74–88.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4625806
- TRB (2000), *Highway Capacity Manual*.
- Treiber, M., Kesting, A. and Helbing, D. (2006), ‘Understanding widely scattered traffic flows, the capacity drop, and platoons as effects of variance-driven time gaps’, *Physical Review E* **74**(1).
URL: <http://arxiv.org/abs/physics/0508222>
- Tsugawa, S., Kato, S., Matsui, T., Naganawa, H. and Fujii, H. (2000), An architecture for cooperative driving of automated vehicles, in ‘ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings’, IEEE, pp. 422–427.
- Turner, J. (1986), ‘New directions in communications (or which way to the information age?)’, **24**(10), 8–15.

- Uno, A. and Sakaguchi, T. (1999), A merging control algorithm based on inter-vehicle communication, in 'IEEE Intelligent Transportation Systems Conference'.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=821160
- U.S. Federal Highway Authority (2006), 'Travel Time Reliability: Making It There On Time, All The Time'.
URL: http://ops.fhwa.dot.gov/publications/tt_reliability/ (accessed 21/12/12)
- U.S. Federal Highway Authority (2008), 'Congestion Pricing - A Primer'.
URL: http://ops.fhwa.dot.gov/publications/fhwahop08039/cp_prim1_00.htm (accessed 28/11/12)
- Vahidi, A. and Eskandarian, A. (2003), 'Research advances in intelligent collision avoidance and adaptive cruise control', *Intelligent Transportation Systems, IEEE Transactions on* **4**(3), 143–153.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1255576
- van Arem, B., Hogema, J., Vanderschuren, M. and Verheul, C. (1995), An assessment of the impact of autonomous intelligent cruise control, Technical report, TNO Inro, Delft.
- Vanderbilt, T. (2009), *Traffic: Why We Drive the Way We Do (and What It Says About Us)*, Vintage.
URL: <http://www.amazon.com/Traffic-Drive-What-Says-About/dp/0307277194>
- Varaiya, P. (1993), 'Smart cars on smart roads: problems of control', *Automatic Control, IEEE Transactions on* **38**(2), 195–207.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=250509
- Varaiya, P. and Shladover, S. (1991), Sketch of an IVHS systems architecture, in 'Vehicle Navigation and Information Systems Conference, 1991', Vol. 2, pp. 909–922.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1623699
- Vlassenroot, S., Broekx, S., Mol, J. D., Panis, L. I., Brijs, T. and Wets, G. (2007), 'Driving with intelligent speed adaptation: Final results of the Belgian ISA-trial',

- Transportation Research Part A: Policy and Practice* **41**(3), 267–279.
URL: <http://linkinghub.elsevier.com/retrieve/pii/S0965856406000589>
- Wang, Z., Kulik, L. and Ramamohanarao, K. (2007), Proactive traffic merging strategies for sensor-enabled cars, in ‘Fourth ACM international workshop on Vehicular ad hoc networks VANET 07’, ACM, pp. 39–48.
URL: <http://portal.acm.org/citation.cfm?id=1287755>
- Wiedemann, R. (1974), ‘Simulation des straßenverkehrsflusses.chriftenreihe heft 8’, *Institute for Transportation Science, University of Karlsruhe, Germany*.
- Williams, M. (1998), PROMETHEUS-The European research programme for optimising the road transport system in Europe, in ‘IEE Colloquium on Driver Information’, pp. 1–9.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=209729
- Xu, Q., Hedrick, K., Sengupta, R. and VanderWerf, J. (2002), Effects of vehicle-vehicle/roadside-vehicle communication on adaptive cruise controlled highway systems, in ‘Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th’, Vol. 2, pp. 1249–1253 vol.2.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1040805
- Xu, Q. and Sengupta, R. (2003), Simulation, analysis, and comparison of ACC and CACC in highway merging control, in ‘IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683)’, IEEE, pp. 237–242.
URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1212915>
- Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X. and Chen, C. (2011), ‘Data-Driven Intelligent Transportation Systems: A Survey’, *IEEE Transactions on Intelligent Transportation Systems* (99), 1–16.
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5959985>