



**Aalto-yliopisto**  
Insinöörیتieteiden  
korkeakoulu

Pablo del Río Barral

## **In-car advisory system for lane-changing in a connected vehicle environment**

Master's thesis, which has been submitted as a thesis for examination for a Master's degree.

A Coruña 25.07.2018  
Supervisor: Claudio Roncoli



---

**Author** Pablo del Río Barral

---

**Title of thesis** In-car advisory system for lane-changing in a connected vehicle environment

---

**Master programme** Master's Programme in Spatial Planning and Transportation Engineering

**Code** ENG26

---

**Thesis supervisor** Claudio Roncoli

---

**Thesis advisor(s)** Claudio Roncoli

---

**Date** 25.07.2018

**Number of pages** 60 including Appendix

**Language** English

---

### Abstract

This thesis investigates the potential of in-car advisory systems to suggest location and timing where and when lane-changes should be executed, by evaluating traffic flow conditions with data that is available using vehicle-to-vehicle communication. After investigating existing literature regarding car-following and lane-changing models, as well as driving support assistance systems and vehicle communication applications and practice, a new lane-changing model is introduced, with the objective to serve as a basis for the development of the in-car advisory system. In particular, the model accounts for information about position and speed of vehicles that are downstream from the considered vehicle current position, namely, out of the sight of a driver. Based on the proposed model, a decision system to deliver lane-changing advices to the driver is implemented, with the goal of avoiding or reducing traffic congestion. A set of simulations using the microscopic traffic simulator AIMSUN are performed to test the effectiveness of the proposed system.

---

**Keywords** lane-changing, in-car advisory system, V2V communication, microscopic traffic simulation

---





# Contents

1. Introduction.....	1
2. Literature collection.....	4
2.1. Car-following models.....	5
2.2. Congestion assistants.....	7
2.3. Lane-changing models .....	11
2.4. Lateral Support Systems.....	15
2.5. Vehicle Connectivity.....	19
2.6. Psychological models .....	24
3. Lane-changing model .....	27
3.1. Model structure.....	27
3.2. Implementation of the model .....	30
4. Simulations .....	38
4.1. Automated Vehicles scenario.....	38
4.2. Connected Automated Vehicles Scenario .....	41
5. Conclusions.....	49
6. Acknowledgments .....	50
References.....	51
Appendix 1 .....	1
Lane-changing model code in C++. Developed with Aimsun microSDK. ....	1

# 1. Introduction

How to analyse, model and manage traffic flows has been the basis for many research studies since several decades ago. In Pipes (1953), the author related that in the decade of 1950 the method of “operations research” was being applied for the study of traffic flow, in investigations such as the statistical study of the behaviour of traffic at intersections or the dynamics of a line of traffic, that is the field in which the author focused his research on, as it is commented later in this work.

Since then, multiple models have been proposed to represent how vehicles interact between them, either regarding a line of traffic where each vehicle follows the preceding one (car-following) or movements of vehicles across the lanes of a road (lane-changing). Nowadays, besides developing more accurate models, the introduction of ADAS (Advanced Driver Assistance Systems) and vehicles with a degree of automation is another factor to consider, as well as the use of available traffic data gathered from spot detectors and floating-car data, like in Schakel and van Arem (2014) or in Roncoli et al. (2016). These systems can improve safety and comfort of drivers, but also traffic flow efficiency, because they can surpass the capabilities of human drivers in some tasks.

The improvement of traffic efficiency is crucial nowadays. The number of daily trips by road is already very high, especially during hours when people drive to or from their working places. Therefore, peaks in the number of displacements by vehicle are produced, mainly, early in the morning and during afternoon. Also, most of road users drive their vehicles alone, which means that the number of people traveling on a road could be translated to near the same number of vehicles. That reduces drastically the efficiency in mobility in cases where private vehicles are majority, like in highways (urban areas tend to have a larger number of public transports, or motorcycles and bicycles).

This poor performance often leads to traffic congestion, which is a major problem as it increases travel time and fuel consumption for vehicles involved. As stated in Roncoli et al. (2015), “the European Commission estimates that the yearly cost of road congestion in Europe exceeds 120 billion €”. To solve this problem, many different options can be considered. One typical approach is to increase road capacity, improving current infrastructures or creating new ones. In cities where there is a limited number of roads to access, offering new routes for drivers can allow to distribute traffic per their different origins. But this option is not possible in several cases, and increases recourses needed, and thus, costs.

Another option consists on reducing the number of vehicles on the road, which is not easy as most of the cities still tend to grow (especially in developing countries), so road trips as well. It could be possible encouraging people to use public transports instead of private cars, or to share their cars. This is becoming more popular nowadays thanks to software applications that can show users all the different options available for them to moving without needing a private vehicle, easily accessible via smartphone or other internet connected devices. But, for such a change in mentality of people, it could take years until reaching significant results.

Road capacity can also be increased changing drivers' behaviour. For example, reducing the headway they apply between their vehicle and the preceding one; this way, size of vehicle platoons is reduced and therefore, road congestion. Nevertheless, is not simple to apply solutions like this, as headway is also directly linked to safety. Drivers must maintain a headway large enough to stop the vehicle in case the leading vehicle suddenly breaks. Some systems, like the one introduced by Schakel and van Arem (2014), propose to deliver advices to drivers to make them more attentive and adjust the gap themselves when driving in congested traffic. Van Gent et al. (2017) explained how these advice systems must be correctly adjust to deliver just the most necessary messages, to avoid distracting the driver. Reaction time of drivers must also be taken into account, which changes depending on several factors (driver's age, his level of awareness, visibility conditions...).

The way vehicles are distributed along and across the road also affects traffic flow, because to reach cross-lane capacity, capacity flow must be achieved for each lane, as declared in Roncoli et al. (2017). Asymmetric traffic rules difficult these balanced distribution, as left lanes (in case of right-hand side traffic situation, as it is considered in this document if not specified the opposite) are used most of the time to overtake, and traffic must use the rightmost lane as the default one. In model like MOBIL from Kesting et al. (2007), how these differences between symmetric and asymmetric traffic rules affect to traffic modelling are taken into account. Also, lane changes can be a source of disturbances in the flow as they affect speed of vehicles surrounding the one carrying out the manoeuvre.

Connectivity is another feature present in vehicles nowadays and, according to Lytrivis et al. (2011), will become widely available in the near future. V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure) allow to receive and transmit information to vehicles, so traffic can be analysed based in all that available data and act in consequence. This actions can be executed by drivers, informed about the traffic or even suggested what to do in a specific situation through some advice system, like the mentioned one from Schakel and van Arem (2014); or even executed directly by autonomous vehicles.

Regarding automated vehicles, it is not within the scope of this project to go deep in that field, but some driver support systems involving a certain degree of automation are mentioned, like Adaptive Cruise Control systems. Some of these systems, like the Stop&Go from van Driel and van Arem (2008), can take control over the driving task in congested situations, maintaining automatically the headway with the preceding vehicle, improving not just comfort but also efficiency, as they can use smaller headways while maintaining safety.

In order to contribute to this research field, a new lane-changing model is investigated. The main objective of this model is anticipating the appearance of breakdowns and acting consequently, distributing upstream vehicles across all the lanes when traffic perturbances are detected. To achieve this, measures of speed and headway are obtained to evaluate the traffic flow in each lane and detect a potential congestion, making vehicles a certain distance upstream to move to those lanes with better conditions (those with higher average speeds or larger gaps).

The model is intended to be the base of a lane-change advisory system, that would alert drivers about an increasing risk of congestion downstream of their current position, so they would be suggested to change lane in advance to reduce the delay in travel time. However, in this first approach, it is assumed that information on position and speed from every vehicle on the road is available, like if the penetration rate of connected vehicles was 100%. All the lane changes commanded by the system are followed as well, which could be interpreted as a scenario with automated vehicles capable of executing the manoeuvre without the interaction of the driver, or with a full compliance rate for drivers following delivered advices. Another assumption could be that the system delivers more lane-changing advices than actual number of lane changes executed by drivers, so a scenario with a lower compliance rate would be feasible too.

This document is organized as follows. First, in chapter 2 it is presented a review of different articles regarding car-following and lane-changing models proposed by various researchers. Driving assistance systems that use these or other similar models are also considered, as well as some research works about vehicle connectivity and interaction between drivers and assistance systems. Chapter 3 is dedicated to the proposed lane-change model. It is detailed how the model has been coded to apply it in a traffic simulator software. The software that has been employed in this project is AIMSUN. Then, in chapter 4, results from some initial simulations are presented and commented. The test network employed to run the simulations is a ring road around the city of Antwerp, in Belgium. Conclusions are



## 2. Literature collection

Developing driving models that could represent how vehicles behave is the purpose for several studies carried out in the last decades. In this chapter, some of the most relevant research articles regarding development of these models are presented and summarized, in order to establish a background for developing a new behavioural model to manage lane changes.

Two main categories of behavioural models are considered:

- Car-following models. These represent how a vehicle behaves within its current lane. In most models, two scenarios are taken into account. First, when the vehicle is traveling in free flow conditions, without traffic in front that could conditionate its behaviour, so the desired speed of the driver is the parameter that governs the model. The other scenario, when a vehicle is traveling among others, is determined by the desired headway (distance between a vehicle and the one ahead) that the driver wants to maintain with the preceding vehicle.
- Lane-changing models. These models aim to represent movements of vehicles from one lane to another, evaluating every aspect of them, from the reasons that lead to start a lane change manoeuvre, to searching an appropriate gap in the target lane and the interactions with surrounding vehicles. These different levels in the lane-changing process (strategic, tactical and operational) are detailed later in this chapter as a review of Tideman et al. (2007) investigation.

Other studies, regarding features like advisory systems, connectivity solutions or human-machine interfaces are also considered in this chapter.

## 2.1. Car-following models

The main objective of Car-following models is to describe the longitudinal movement of a vehicle based on the movement of the preceding one. According to Pipes (1953), back in the decade of 1950, there were already studies about traffic flow focused on the dynamics of a line of traffic, so he proposed a mathematical analysis of this feature assuming that vehicles must follow a certain traffic regulation when following each other. He considered the California Vehicle Code, that suggested a safe distance between vehicles as that of “the length of a car (about fifteen feet) for every ten miles per hour” speed.

In his study, Pipes obtains the dynamical equations that govern a line of traffic for two situations:

- A line of vehicles initially at rest starts moving, specifying the motion of the leading vehicle among different options (suddenly acquisition of motion, gradual acceleration and constant acceleration).
- A line of vehicles moving with the same cruising speed begins to decelerate when approaching a stop point. In this case, the sudden stop and exponential deceleration of the leading vehicle are considered.

For the most complex mathematical problems regarding this analysis, the author proposes to use an electrical analogue computer, proving how connected are since their beginnings the analysis of traffic and the computer engineering.

Robert E. Chandler was also a pioneer in the field of car-following models. Chandler et al (1958) developed mathematical models based on the theory of servomechanisms and network analysis, establishing analogies between traffic and communication theory. These models take into consideration how drivers program their driving differently when they drive alone or following other vehicles.

In this last case, one essential variable in the model is the inter-car spacing, which depends on the velocity of the vehicle. The gap (or headway) that a driver is willing to maintain with its leader is what defines how vehicles move in a platoon, in the same way that the desired speed configures behaviour in free flow situations. The gap between vehicles is the core of every car-following model consulted for this text.

One of the car-following models that are widely cited by researchers is the one from Gipps (1981). He constructed a car-following model that focuses on following three main objectives:

- Mimic the behaviour of real traffic. This point is the most obvious, as accuracy is needed in order to run realistic simulations and obtain valuable results, to study actual traffic situations and develop solutions.
- Make it easy to be calibrated (using reasonable vehicle and driver parameters).

- Adapt time between recalculations to the reaction time of the driver. This aspect, regarding the moment the article was released, probably would not be conflictive nowadays due to advances in computing technology. However, as many of these models are intended to be integrated in vehicle on-board units, like embedded computers, time between calculations should always be a main point to focus on as it is critical for traffic safety.

The previous model calculates a safe speed respect to the preceding vehicle applying limits to the performance of the vehicle and the driver and considering that the driver selects his speed to make it possible to stop the vehicle safely. This limitation on the speed would be different in free-flow traffic conditions or in a congested situation, but the model ensures a smooth transition between these cases, except for cases like the leader braking harder than what the following vehicle could anticipate, the leader leaving the lane or another car moving in between the two.

Gipps model also establishes that if the velocity of the leader exceeds the desired speed of the follower, it stops following. The follower vehicle would not react instantly to a change in speed of the leader, it would wait to adapt the spacing to the new velocity.

The validation of the model show that it is able to mimic real traffic behaviour when parameters corresponding to drivers and vehicles take reasonable values, being the three main factors that control this behaviour: the distribution of desired speeds ( $V_n$ ), the reaction time for drivers ( $\tau$ ) and the ratio between rate to driver's estimates of the mean braking rate, ( $\bar{b}/\hat{b}$ ). Other advantage of the model is its speed of calculation.

Trying to simplify how car-following models are developed, Newell (2002) proposes a simple car-following rule: "the time-space trajectory of the  $n$ th vehicle is essentially the same as the  $(n - 1)$ th vehicle except for a translation in space and time". With this statement, he affirms that a vehicle following another is not just influenced by it, but it almost mimics its trajectory, maintaining a gap. This spacing between vehicles, noted as  $S_n$ , depends on their velocity. If the leading vehicle increases it, the following vehicle will increase it in the same level, with a delay in time ( $\tau_n$ ) and space ( $d_n$ ), and applying a new larger gap, as shown in figure 1.

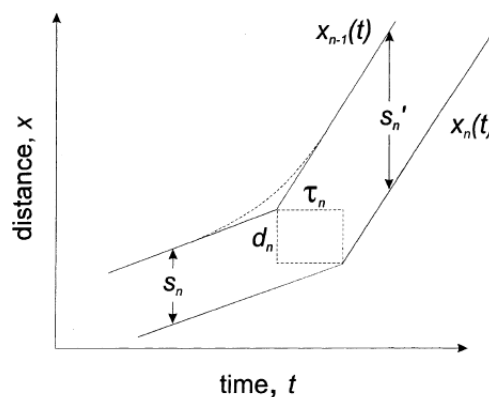


Figure 1. Relation between leading and following vehicle displacement. Newell (2002)

Newell compares his work with other traditional car-following models, like the mentioned from Chandler et al. (1958). Studies on how to fit curves to observed values of flow ( $q$ ) and density ( $k$ ) (for a macroscopic approach of the relation between velocity and spacing) are also reviewed in the paper, to prove that the presented linear relation between velocity and spacing is realistic. The conclusion that he reaches by this is that “there seems to be little, if any evidence of these non-linear effects except possibly for vehicles close to the desired velocity, flows close to the maximum, or for very low velocities”.

Wang and Coifman (2008) published a paper, trying to solve the problems that Newell’s car-following theory presents when there are frequent lane-change manoeuvres, due to its impossibility to predict microscopic behaviour of vehicles. The purpose is to evaluate the impact of lane-changes in these model. They offer a review of Newell’s work, and mention other papers which provide empirical support for Newell’s theory, at least under certain conditions.

In their article, Wang and Coifman describe how spacing between vehicles change when one of them performs a lane-change. All the affected vehicles in the manoeuvre must adapt the distance to the vehicle in front, so during this “accommodation period”, the spacing and speed deviate from the preferences of drivers. However, as it can be seen in figure 2, results from simulations show that, even with considering these discontinuities in the relation between spacing and speed, it is still strong and not far from Newell’s model.

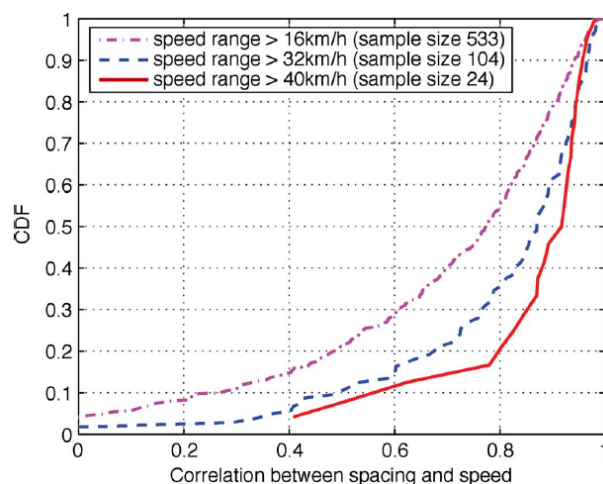


Figure 2. Cumulative Distribution Function of correlation between spacing and speed for different groups of cars regarding their speed. Wang and Coifman (2008).

## 2.2. Congestion assistants

Car-following models can be employed as a basis to develop systems that assist drivers in situations in which the driving task is basically reduced to following a line of traffic. For example, in a traffic jam or when approaching to a platoon. In this section of the chapter, some of the solutions provided by various authors are presented.

Kesting et al. (2007) presented an adaptive cruise control (ACC) traffic-assistance system to “improve traffic flow and road capacity”. Both traffic efficiency and driver comfort are within the aim this system, so it uses an intelligent driving strategy to avoid possible conflicts. The driver can adjust the desired velocity and safety time gap, so the system takes that into consideration to calculate appropriate accelerations and decelerations.

This ACC is modelled by a car-following approach, that needs to meet some requirements. The model must be accident free, permit smooth driving and adaptability, as well as count with a reduced number of parameters (like those adjusted by the driver) and different driving styles to apply. Although human drivers should be modelled in a different way than ACC systems due to human particularities like reaction time or the better capacity to be aware of their environment, an equally simple car-following model could be used too to simulate them.

The ACC is able to detect different traffic situations and adapt its behaviour model to those scenarios. In free traffic, the driving comfort is the main objective. As drivers approach to a jam, the system focus in early braking to approach in a more safe and efficient way to slow vehicles. Within congested traffic, the ACC settings are back to default values, but when leaving a jam, the maximum accelerations increases so the minimum time gap decreases. Smaller gaps are also considered in bottleneck sections, to increase capacity.

Floating-car data is employed by the presented system to detect the traffic situation in which the vehicle is at a certain point. Information is also gathered, for example, from vehicle sensors (speed of leading car) or digital map databases (road geometry, potential bottleneck regions). To improve the detection system, non-local information, provided with V2V or V2I communications, could be incorporated to the system.

The ACC system was tested through simulation, using the MOBIL lane-changing model by Kesting et al. (2007) to represent more complete and realistic situations. The objective is to evaluate the impact of proportion of vehicles equipped with ACC systems (penetration rate of 0%, 5%, 15% and 25% were applied), driving strategies and boundary conditions on traffic. The road chosen for calibration was the A8 between Munich and Salzburg. As a result it was discovered that increasing the number of ACC vehicles congestion was reduced, and thus travel time, as it can be observed in figure 3, with total traffic breakdown avoided with the 25% rate.

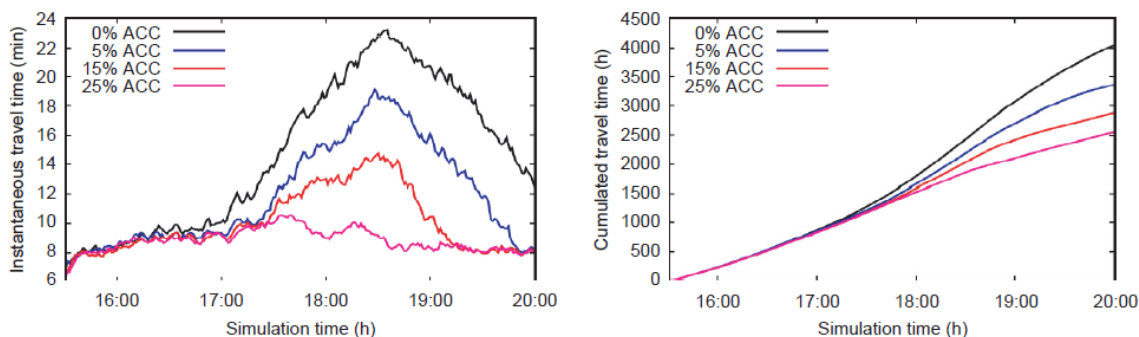


Figure 3. Instantaneous and cumulative travel time during simulation. Kesting et al. (2007).

Van Driel and van Arem (2008) developed their own ACC system, adding an Active Pedal function to this congestion assistant aimed to improve traffic flow. They check as well the system through microscopic simulation. The ACC function is based on a Stop&Go system, which takes control of the longitudinal movement of the car during a traffic jam, activating it only within a range of velocities. The Active Pedal produces a force on the gas pedal to warn the driver if approaching too fast to a traffic jam.

Simulations of the system show that both functions have positive results in reducing congestion, but the benefits of the Stop&Go are much larger, as it can be observed below in figure 4. In the displayed cases vehicles are equipped (with a penetration rate of 10% or 50%) with the active pedal system, activated 500 m before of a traffic jam; with the Stop&Go function, with a time gap of 0,8 s and with both systems acting together in the last case.

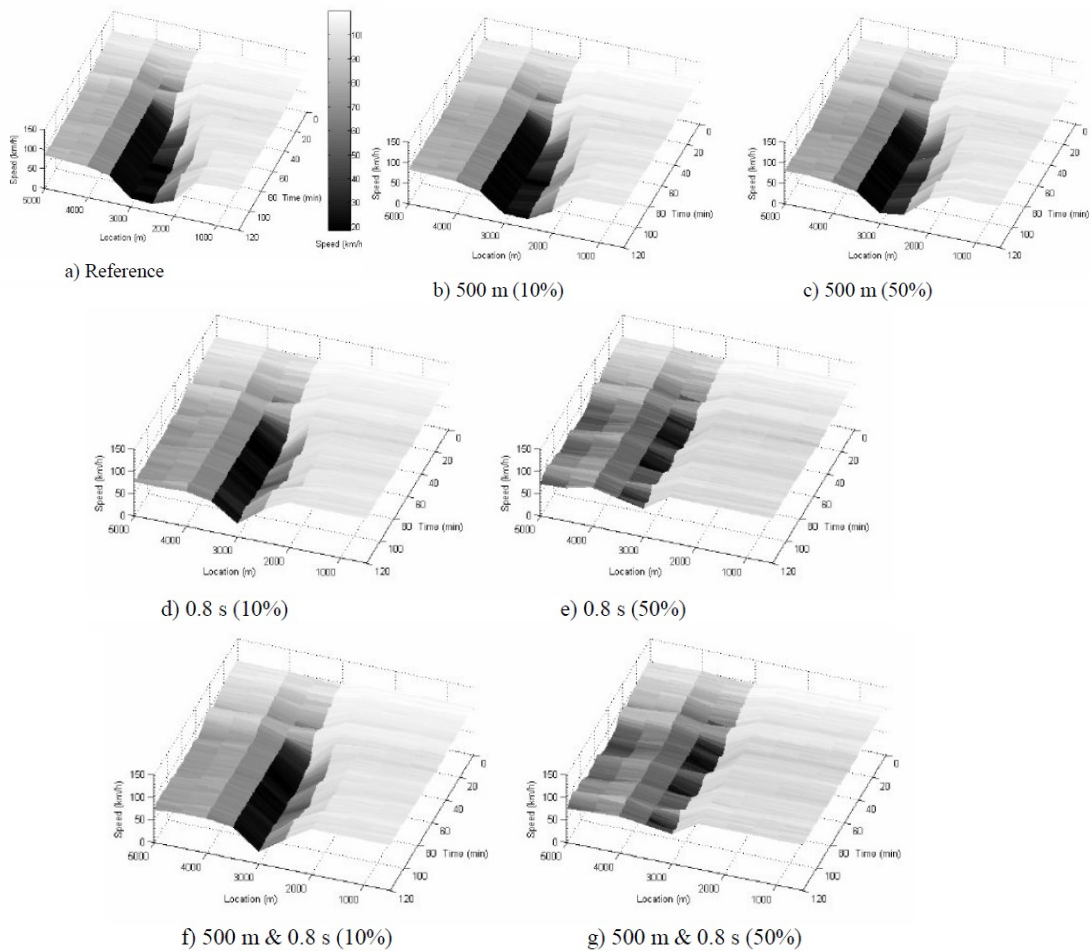


Figure 4. Effects of congestion assistant on traffic speed.  
van Driel and van Arem (2008).

This Congestion Assistant uses V2V communication to detect traffic jams downstream of the position of the vehicle.

A particular type of lane changes are merging situation, as they imply the execution of mandatory lane changes. Pueboobpaphan et al. (2010) presented a decentralized merging assistant for mixed traffic situations, involving manual and C-ACC (Cooperative Adaptive Cruise Control) vehicles. To increase stability around the merging area, an algorithm is

proposed to control the C-ACC so mainline vehicles decelerate when approaching a ramp to create gaps for incoming vehicles to merge. After simulation, authors found that the merging assistant is able to generate the gaps for ramp vehicles, but this are sometimes unnecessarily large.

Wang et al. (2015) worked on controllers and implementable algorithms for CFC (Car-Following Control) and C-CFC (Cooperative-CFC). Decentralized and distributed algorithms are implemented to deal with control of several vehicles, studying the impact of the controllers in traffic flow. Communication between vehicles was used to increase their awareness level, through a centralized communication scheme. The more vehicles are involved, the more complex and challenging this scheme gets. Thus, efficient algorithms are proposed in the article, making each CFC vehicle optimize its own situation and using a distributed algorithm for cooperation between vehicles, sharing their latest state information together with predicted control information. The impact of this controllers is tested through simulation in a two-lane highway with randomly distributed CFC and C-CFC vehicles.

The presented CFC system is divided in two modes: following mode and cruising mode. Both consider the need to maximize comfort and travel efficiency, while following mode also focusing on safety. A gap threshold is established to distinguish both modes, determined by a desired time gap that is decided by the driver; a running cost function allows to have a smooth transition between modes. Variable time gaps can be implemented. With the decentralized algorithm, each vehicle, working as a subsystem, solves its local autonomous optimal control problem and predicts the behaviour of its leader using constant speed heuristics. If the leader is a C-CFC vehicle, its state is predicted with the assumed acceleration trajectory.

Centralized optimization scheme is implemented for platoon control, using a distributed algorithm. C-CFC vehicles are supposed to have both forward and backward sensors to gather gap and speed information from their leaders and followers, respectively. Also, they equip V2V communication capabilities, which makes possible to receive state information from the leader and transmit its own to the follower. Sensors are used when the surrounding vehicles are human-driven, while the V2V information are considered for other C-CFC vehicles. A joint cost function is specified to minimize costs of safety, efficiency and comfort for the vehicle and its follower. Optimal acceleration is determined by the marginal costs of the vehicle own relative speed and its follower speed.

Centralized optimization would not be feasible due to computation and communication requirements, so each vehicle works as a subsystem, as it was mentioned before. Once each vehicle solves its local cooperative optimal control problem, determining its optimal trajectory.

The simulation process consists in creating bottlenecks applying VSL (Variable Speed Limits) at some point of a freeway, which produces traffic waves. It is observed that, when the penetration rate of CFC vehicle increases, capacity drop and microscopic hysteresis are mitigated. Also, as CFC vehicles accelerate faster to the high-speed state at the head of the traffic jam, producing smaller gaps between vehicles and thus an increase of traffic flow. This reduction of stop-and-go wave means a reduction in fuel consumption.

### 2.3. Lane-changing models

An investigation on the lane changing process with a microscopic approach was developed by Brackstone et al. (1998). To evaluate the gap acceptance process, a driver in an equipped vehicle (with a radar, a laser speedometer and a video recording system) was asked about his/her intention to lane change at random intervals, recording instrumental data for setting “gap acceptance” and “gap rejection” cases, as well as driver opinion for not measurable variables like “motivation” or “opportunity” to lane change. The paper demonstrated the possibility “to formulate reasonably accurate behavioural models for lane changing”.

Zheng (2013) reviewed several lane-changing models, analysing how each one deals with the decision-making process and how lane changes impact on surrounding traffic.

Gipps (1986) proposed a structure to deal with decision process of a lane-changing model, studying potentially conflicting goals. This goals are defined on the basis of an assumed general travelling objective for the driver, and consist in specific objectives like driving with a particular desired speed, using the correct lane or limiting accelerations and decelerations employed. The framework presented by Gipps covers urban traffic situations.

According to Gipps, the factors to take into account to decide the aptitude of a lane change are various. Safety has to be assessed, avoiding situations that could lead to a risk of collision, so an appropriate gap has to be identified before executing the manoeuvre. The model must reflect as well the awareness of drivers respect to obstacles in the road, like permanent obstructions or heavy vehicles, and also respect to lanes which are not allowed to be used by them, like transit lanes reserved for public transport. Other considered factors, like in most of lane-change models discussed in this text, are the distance to an intended turn or the potential speed advantages derived from changing to another lane (or remaining in the current).

To describe the behaviour of vehicles in the road, the presented model uses different patterns based on the distance at which a vehicle is from its intended turn. The closer it gets to it, the more important becomes for the vehicle to be in an appropriate lane for turning, ignoring gradually speed advantages derived from changing to other lanes. The model is as well designed to be integrated with the car-following model (Gipps 1981) commented above, and calculates using it a safe speed to be maintained with the preceding vehicle for manoeuvring.

The decision process of this model to change lane has several steps. A preferred lane has to be defined for the vehicle, so that lane would be the first option when selecting a target. Then, the model checks feasibility of the manoeuvre, so there are no obstructions or excessive decelerations needed. The next step evaluates if the vehicle is close to its intended turn, increasing willingness of the driver to accept smaller gaps when reaching that critical point. The model must consider transit lanes too, allowing non-transit vehicles to use them only if their current lane is obstructed, and making them leave the transit lane as soon as the obstruction has been overtaken. If the vehicle is not close to its intended turn, but at a middle distance from it, limits will be applied to selection of acceptable lane changes.

If factors above do not apply for the vehicle at its current situation, relative advantages of current and target lanes would be considered to decide the lane to be used. The vehicle will change lane if the target one offers a speed gain high enough, or if it allows to drive faster



by avoiding heavy vehicles. The last step is to evaluate safety of the manoeuvre, a criterion that could change depending on the urgency of the situation. If after the whole process, change to the target lane is rejected, the opposite lane is considered as a new target.

MULTISIM is an investigation tool to analyse the effects of lane changes in traffic flow through micro-simulation. In a paper for the 11Th IMACS World Congress in Oslo, Gipps (1986) explains how the model let users to adjust parameters like geometry of the simulated road (length, lanes, turnings, obstructions) and volume, composition and characteristics of the traffic.

Vehicles in MULTISIM are generated by a subroutine that set up their class and attributes. A car following model determines behaviour of vehicles within their lane. Lane changing depends on different objectives that may conflict, so a decision process must be followed, as described in the previous cited article from Gipps (1986). Speed and position of every vehicle is updated at fixed intervals, equivalent to reaction time of drivers.

The data is collected from monitors assigned to each vehicle, and written onto a file as vehicles depart from the network. MULTISIM provides graphic displays of the simulation operation, basically time-distance diagrams where trajectories of the vehicles in each lane can be observed (figure 5), represented by lines that go through gaps in the graph (vehicles crossing at intersections with green traffic lights) or stop when they encounter a solid rectangle (red lights). Comparison between trajectories among each lane graph allows to determine when trajectories leave a lane or enter another.

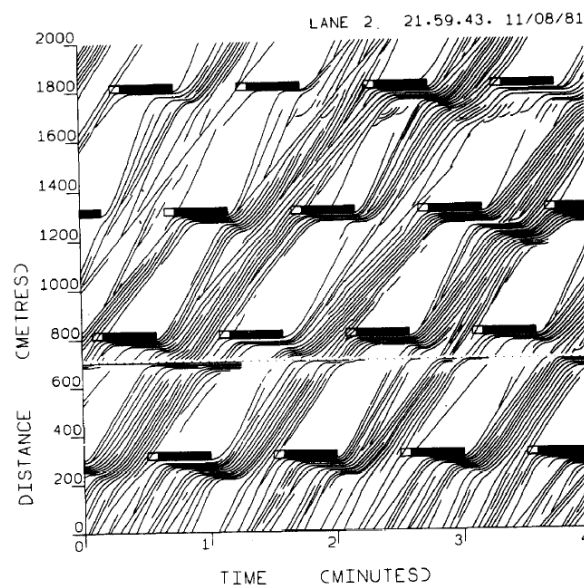


Figure 5. Time-distance representation of trajectories of vehicles in the second rightmost lane. Gipps (1986).

Interactive graphics are also produced while simulations are running. Possible applications of MULTISIM cited by the author are the study of congestion, travel time, number of stops of vehicles... but other applications derived from those are cited as well, like modelling fuel consumption. It is interesting to notice, considering the time by which the paper was published, that scenarios including electric vehicles or dynamic advisory speed signs are also taken into account.

Kesting et al. (2007) developed one of the most habitually cited lane-changing models, called MOBIL (Minimizing Overall Braking Induced by Lane-change). With this model, the lane-changing process is faced as a multi-step process:

- Strategic level. Evaluation of the route that the driver must follow.
- Tactical stage. Lane change is prepared and initiated (acceleration, deceleration)
- Operational stage. Decision about desirability and safety of the change, as a gap-acceptance process, comparing available gaps to a critical gap, that depends on the relative speed of the driver respect to the leader and the follower in the target lane.

The presented lane-changing model considers the expected advantages and disadvantages derived from the manoeuvre, anticipating and comparing vehicle accelerations before and after a potential lane change. If the accelerations in the target lane are closer to free flow conditions, the attractiveness increases. This acceleration based criteria allows for easy integration with car-following models.

Another contribution of this model is the introduction of a “politeness parameter”, so a driver would not change lane for obtaining just a marginal advantage if this obstructs other drivers. Also, aggressive drivers would make slower vehicles to yield, leaving the faster lane. This parameter can be adjusted to apply a more egoistic or more generous behaviour.

The model has diverse criteria to execute lane changes. Safety criteria is based in longitudinal accelerations, guarantying that deceleration of the new follower vehicle in target lane does not exceeds a safe limit. Gap acceptance is also related to acceleration, as a new follower approaching fast in the target lane requires for larger gaps to apply the lane change. This way, crash situations are excluded.

Not only advantages that a potential lane change could mean for the considered vehicle are taken into account by this model. To satisfy the criterion, these advantages had to be compared with the weighted sum of the disadvantages for the new and old follower of the vehicle that this manoeuvre could lead to, and exceed a threshold ( $\Delta a_{th}$ ). In the next equation, “c” subscript for acceleration refers to the changing vehicle, “n” for the new follower and “o” for the old follower. Terms with an accent refers to the situation after change. The politeness factor, “p”, is the degree of altruism of the changing vehicle.

$$\tilde{a}_c - a_c + p \cdot (\tilde{a}_n - a_n + \tilde{a}_o - a_o) > \Delta a_{th} \quad (1)$$

With symmetric traffic rules, if incentive is satisfied in both adjacent lanes, the target lane would be that offering best overall traffic advantages. With asymmetric rules this will be limited, as overtaking in the right lane is prohibited (unless traffic is congested) and the rightmost lane must be the default one. Slow vehicles would change to let faster ones to pass in the left lane. In figure 6, obtained from the MOBIL paper, it can be seen how speed changes depending on traffic density. In both cases, it decreases as density grows, and there are barely differences between lanes for symmetric traffic rules. It is remarkable how global speed is higher when the politeness factor is  $p=1$ , this is, drivers show a more altruistic behaviour (with  $p=0$ , they act egoistically).

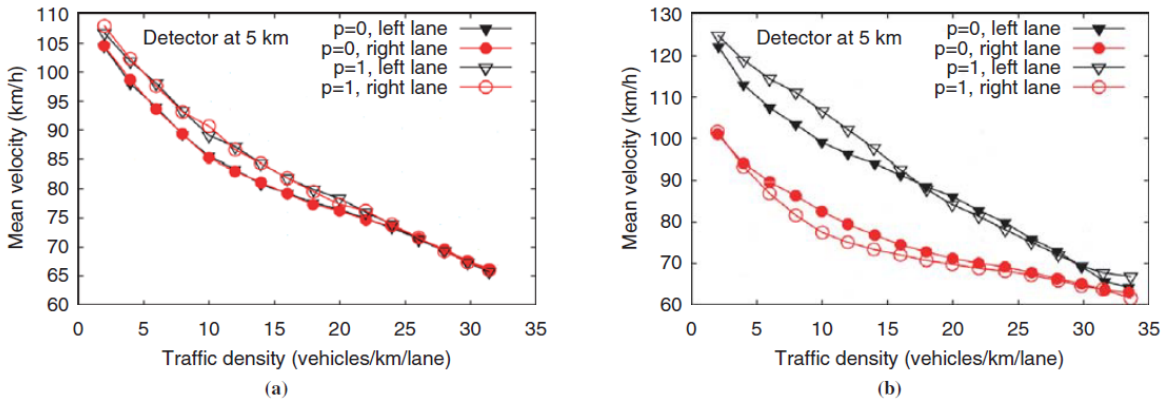


Figure 6. Mean velocity vs traffic density for symmetric traffic rules (a) and asymmetric (b). Kesting et al. (2007).

Schakel et al. (2012) presented a lane-change model, aiming to reproduce accurately real world traffic situations regarding lane distribution and lane speed. Adaptability is another goal for the model, so multiple lane change incentives are included for the decision process while a limited number of parameters to simplify calibration. The model also includes circumstances like relaxation, that reflects the willingness of drivers to accept headways smaller than usual and small decelerations during a lane-changing manoeuvre; and synchronization, that is the preparation for lane change when drivers adapt their speed to adjacent traffic and align with a gap.

In this model, the desire to perform a lane change, which is the core for the decision making, and therefore driver behaviour, depends on the followed route, the desire to gain speed or keep-right rules. With a higher level of desire, the driver willing to change lane will accept smaller headways and greater decelerations. If the desire level is too small, the driver will not execute the change, but when level increases different manoeuvres would be considered (figure 7), from free lane change (FLC) to synchronized change (SLC) and finally cooperative change (CLC), in which a gap has to be created by the driver in order to get to the adjacent lane, and the potential follower vehicle should collaborate to achieve that goal.

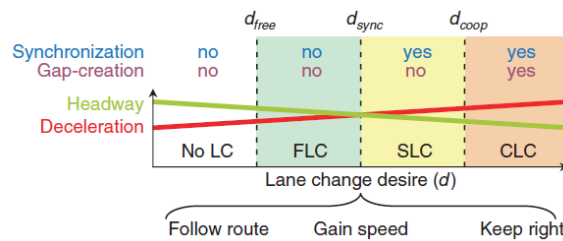


Figure 7. Types of lane-changes executed depending on the level of desire. Schakel et al. (2012).

The mentioned incentives that the model considers, to be able to adapt to different scenarios, are anticipation speed (awareness of drivers for vehicles downstream), incentive speed (desire to increase speed, only considered for the left lane), route (based on remaining time or remaining distance, depending on how high speed is, to change to required lane) and keep right rule (only respected if speed and route are not negatively affected). Voluntary lane change incentives, speed ( $d_s^{ij}$ ) and keep right ( $d_b^{ij}$ ), are ignored if a conflict with mandatory

lane change desire, route incentive ( $d_r^{ij}$ ), occurs. The level at which voluntary incentives are included is indicated by  $\theta_v^{ij}$ .

$$d^{ij} = d_r^{ij} + \theta_v^{ij} \cdot (d_s^{ij} + d_b^{ij}) \quad (2)$$

Gap acceptance and relaxation processes of lane-changing manoeuvres must consider deceleration values obtained from car-following integrated model, rejecting decelerations that are too large. Synchronization is initiated when the lane change desire is above the synchronization threshold.

## 2.4. Lateral Support Systems

Like car following models, lane-changing models can also be employed as the basis to implement driving support systems, including in this case the movement of vehicles among the lanes of the road. In this section, some of the support systems investigated and developed in recent years are presented. Some of them need the compliance of the driver to operate the vehicle, because the system is limited to deliver advice messages. Other take advantage of vehicle autonomous capabilities to act directly.

Tideman et al. (2007) reviewed different lateral driver support systems. The research includes methods and technologies of detection, focusing on sensor technology, detection algorithms and safety assessment algorithms. Among the functions of these systems are the prevention of accidents and the increase of traffic flow. The authors affirm that a lateral support system has different functional components, which must:

- **Sense.** Gathering information, directly from sensors or receiving information through communication networks V2X. Sensors can be active or passive, if they emit or not electromagnetic energy (passive can just detect it) or a combination of both. Most common are optical (passive) sensors, like cameras (figure 8).

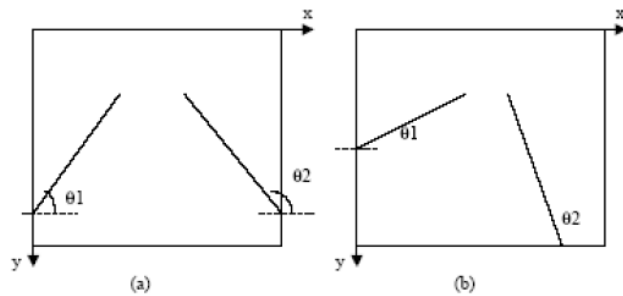


Figure 8. Camera images of lane border for a car driving in a lane (a) and leaving it (b). Tideman et al. (2007).

- **Think,** what means, to interpret information. The detection algorithms are used to determine the context of use of the lateral assistant system, detecting lanes or other vehicles through the generation and verification of hypothesis based on information gathered by the system. The safety assessment algorithms evaluate the hazard of situations like lane departures and lane-change.
- **Act.** Execute actions, through human machine interfaces or taking control over.

These components must work together to achieve the main objective of the system, that is “to support the driver during movements along the lateral axis of the vehicle”.

Schakel and van Arem (2014) developed an in-car advice system to avoid breakdowns in traffic flow that could lead to capacity drop (queue discharge rate lower than road capacity) and spillback (traffic not passing the bottleneck). Achieving this would improve traffic flow efficiency using already existing technology. This system operates on the tactical scale of driving, focusing on driving aspects like lane-changes, which makes its study relevant for this project.

This system uses a traffic management centre to gather information from detector and floating-car data from in-car devices and predicts traffic state, sending advices to vehicles based on this evaluation. This way, vehicles receive information from a downstream traffic situation that drivers cannot perceive on their own. The system loop is executed every minute, limiting the number of advices each driver receives to avoid workload saturation for them. This idea has inspired the lane-changing model developed in this project, that is detailed in later chapters.

The advice algorithm that the system employs has four steps. In the first one, infrastructural properties, mainly regarding geometry of the road, are assigned to sections (or “cells”) of the network. Then, advices regions are created, to define distribution in time and space of the advices and their content. Advices are triggered based on traffic state and following one of the three next principles:

- Acceleration advice principle, by which drivers are suggested to stay attentive and maintain “a short but safe headway at the end of congestion”, in order to maximize capacity.
- Distribution advice principle, that triggers advices to maximize utilization of all the lanes when one of them is too “busy”; for example when it is next to a merge lane. In a case like that, drivers in the busy lane are suggested to change to an adjacent not-congested lane or to yield for traffic from the merge lane.
- Spillback advice principle, with the intention to avoid congestion due to spillback in an off-ramp by telling drivers in the right lane to move to an adjacent one.

In the third step, advice regions are filtered following a priority order. In the last step, drivers to receive advices are selected based on their position and other floating-car data.

Responses to advices given are incorporated to the LMRS lane-changing model from Schakel et al. (2012), applying a compliance rate. How desired speed and headway of vehicles are modelled in the LMRS is affected by quantitative advices following the aforementioned principles. For example, modifying maximum acceleration applied by drivers to maintain short headways. In case of the lane advices, following an advice is incorporated to the LMRS lane change desire expression, increasing desire according to the compliance rate and depending on the time since the advice was received.

The system is evaluated implementing it in the LMRS simulation framework, on the A20 freeway near Rotterdam. Detector data from the road was employed. The simulated scenarios used different values for the penetration rate ( $\lambda$ ) and the compliance rate ( $\omega$ ) of the system. In the graphic below (figure 9), simulations from two different days (a and b) show how delay time (difference between actual travel time and ideal conditions travel time) decreases gradually when increasing both compliance and penetration rates, except for a peak with 10% penetration rate conditions.

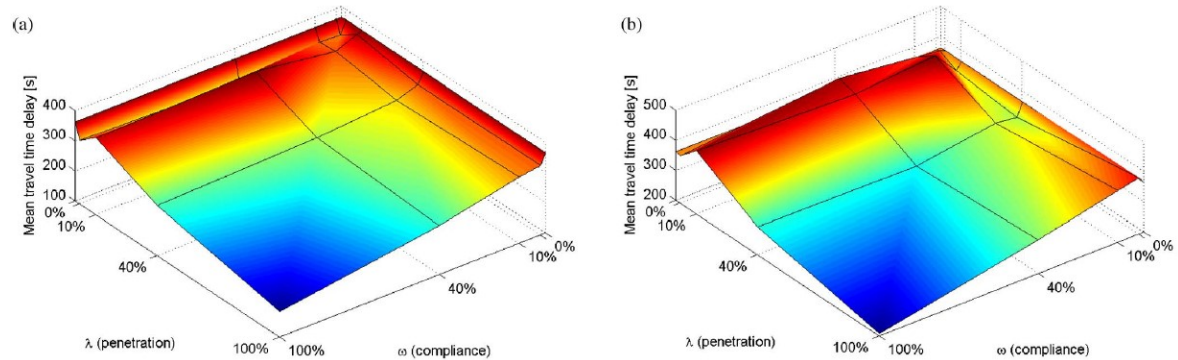


Figure 9. Travel time delay for different levels of compliance and penetration rates. Schakel and van Arem (2014).

To “assess the effects of detector data delay on traffic state prediction” of an in-car advice system as the described, Schakel and van Arem (2015) evaluated the system. They conclude that data delay, up to 180s, has no influence on the improvements in travel time delay provided by in-car tactical advice.

A macroscopic traffic flow model to be used within a model-based control strategy is described in Roncoli et al. (2015). This model includes variable speed limits (VSL) and lane-changing control strategy, and simplicity is one of its main objectives so it could ensure the efficiency of the optimal control calculation process and be applied on large-scale networks. To compute lateral flows, densities for every segment and lane of the road are considered, measuring flows of vehicles entering and exiting a segment, lateral flows between lanes and flows entering and leaving from on-ramps and off-ramps, respectively.

To apply the presented control strategy, and due to the number of interdependent factors that lane-changes are subjected to, like human driver behaviour, geometry of the road or environmental conditions, a basic lane-change model is employed, in which attractiveness for a driver to change lane is based on differences in density between them. This model is affected by situations like exiting the road, changing to facilitate merging of on-ramp traffic or because of the proximity of a lane-drop that require empirical calibration. Aggressiveness in lane-changing is another parameter of the behavioural model, and available space in a lane to receive vehicles is described by flow acceptance. The model is able to provide the right lateral flow to satisfy the off-ramp flow.

Regarding longitudinal flow, one of the main objectives of this model is to represent the empirical phenomenon of capacity drop accurately and with a linear formulation, assuming that in case of congestion (above a critical density value) the flow linearly decreases with a fixed slope. As stated in the paper, introduction of automated driving systems could alter capacity drop.

In Roncoli et al. (2017), a feedback control strategy is proposed for lane assignment of vehicles upstream of a bottleneck to “maximize throughput, targeting critical densities at bottleneck locations as set points”. Another controller, like mainstream traffic flow control, is used together with the control strategy to ensure that the flow approaching the bottleneck does not exceed its capacity. The control strategy is evaluated through several simulations in a hypothetical highway. The stretch employed is divided in different segments, so each lane within a segment is considered a cell. A lane drop is placed as well in one of the segments.

In the first case, without control actions, traffic congestion occurs as lane-changes are only executed a few segments upstream the lane-drop and spillback is also created. When optimal control strategy with constant set points is applied, it manages to avoid the creation of congestion, keeping bottleneck density at its critical value; lateral flows are much more homogeneously distributed and total travel time is reduced. In the last case a different control strategy, with the objective of distributing the total density at the bottleneck, is tested and, although congestion is avoided too, it shows no improvements comparing with the previous case. In figure 10, contour plots of densities are displayed for each lane in each case.

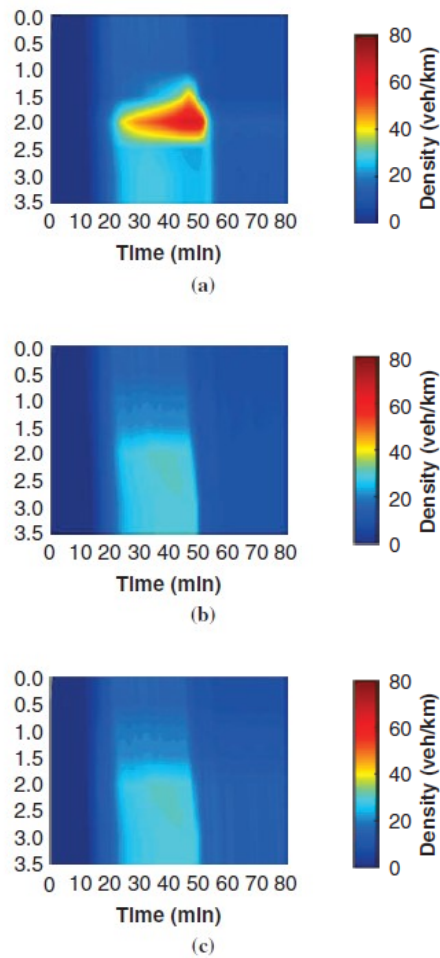


Figure 10. Contour plot of densities in central lane of the stretch for no-control case (a), control strategy with constant set points (b) and density distribution strategy (c). Roncoli et al. (2017)

## 2.5. Vehicle Connectivity

Recent developments on vehicle automation and communication make necessary to create new methods for modelling, estimation and control of traffic. Connectivity capabilities like V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure) may allow for an improvement in traffic estimation accuracy with a reduced number of spot sensors used to measure conventional traffic situations.

Lytrivis et al (2011), for example, developed an advanced cooperative path prediction algorithm, to improve road safety by gathering and sharing information among connected vehicles on the road: position, velocity, acceleration, heading and yaw rate, as well as road geometry. This algorithm allows drivers to have a better perception of the road environment.

A VANET (Vehicular Ad-hoc Network) is employed, as it offers the possibility to predict other vehicles paths from information delivered by these, like speed, acceleration, yaw rate... Whereas a radar can just estimate some of these parameters, so it is not accurate enough for safety purposes, like collision warning, collision avoidance or emergency electronics brake lights (EEBLs). The system equipment consists of the mentioned VANET router, GPS, inertial sensors and an unscented Kalman filter (UKF).

The algorithm is aimed for highway and rural situations, not for urban environments, that are highly dynamic. Track data is provided by the LRR sensor (that offers position and velocity measurements) and the vehicular network. LRR has low latency, but also higher refresh rate than the network. Position of the targets is obtained accurately thanks to the LRR, while VANET offers yaw rate and acceleration of other vehicles directly measured by them. Information from both systems is merged to obtain vectors that describe the state of tracked objects. Beacons messages are transmitted to inform about the situation of each vehicle. In addition, event messages alert about imminent situations.

The transmitted VANET messages are stored and deleted after a time window, using a VANET ID to consider just one message for every car. Estimations of future paths of all vehicles are compared between each other. Alert message can be sent to drivers with paths that intersect, using different models to predict the paths, being the CTRA (constant turn rate with tangential acceleration) the more realistic and complex, as it considers both the yaw rate and the acceleration values. Other models are the CV (constant velocity), CA (constant acceleration) and CTR (constant turn rate). Through a Dempster-Shafer reasoning system the most suitable model is selected for each situation and each road geometry.

The EEBL alerts drivers if the car in front executes an emergency braking. This function demands very low-latency V2V communication to be effective, but is useful as it makes possible for drivers to anticipate to this situation even with bad weather conditions or obstructed line of sight, and increases the range of action allowing to react much in advance. The system uses the information provided by the vehicle braking to predict its path (as depicted in figure 11) and decide if it is relevant to give an alert message to the driver.



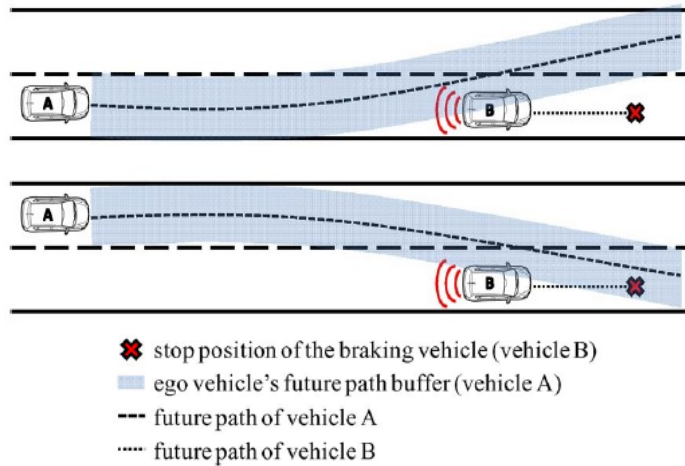


Figure 11. Intersection of vehicle paths. Lytrivis et al. (2011)

The EEBL is tested to validate the path prediction algorithm. After the first tests, results show that using the DS reasoning system to decide which model is used at each moment, performance is improved. Comparing the LRR results with the VANET and a fusion of measurements from both, the advantages of using V2V communication are clear, as estimations are largely improved. The researchers demonstrated also that time horizon for the predictions must be short (around 4s) to offer reliable values.

A second set of tests focus on the performance of the cooperative association, the path prediction algorithm and the notification system, to evaluate if they are correctly delivered or not. The algorithm checks if the paths of the braking vehicle and its follower intersect, so if it occurs, a notification must be displayed to the driver. Different scenarios are considered, showing a correct operation of the algorithm in 14 out of 15 cases, showing the great performance of the use of VANET networks and its potential for improve road safety.

Other vehicle communication application is presented by Awal et al. (2015), that propose a Cooperative Lane-changing Algorithm (CLA) aimed to offer better performance for discretionary lane changes, in terms of travel time, fuel consumption, pollutant emissions and impact on traffic. It also considers mandatory lane changes, reducing merging time. The authors declare that it outperforms MOBIL model from Kesting et al. (2007).

Related lane-change models are discussed, focusing on MOBIL and the fact that this model only takes into account the adjacent gap, not including synchronization or considering all the lanes of the road.

The proposed Cooperative Lane-changing Algorithm considers the advantages for the subject vehicle, the follower in the current lane and several lag vehicles in the target lane, minimizing the impact that lane-change manoeuvres have on traffic flow. Safety and utility of lane-changes are defined using car-following models.

The minimum velocity of the vehicles ahead is what determines which among all the available lanes offers the best option. For discretionary lane-changing, the velocity differences of the surrounding vehicles are compared with a threshold to establish the margin of advantage. In case of mandatory lane changes, the vehicle looks for a gap in front of other vehicle in the target lane for synchronization, establishing a decision-making point and a lane-changing point (instead of just using the same point as MOBIL does) based on the information about position and speed obtained from other vehicles through V2V communication. The vehicle that is carrying out the manoeuvre also sends signal and spatial information to the vehicles in the target lane, so they can adjust their velocity.

Comparing with MOBIL, this CLA algorithm achieves lower average merging time for vehicles in on-ramps, as well as a better merging rate. It also improves the waiting time for ramp vehicles, which in the end, as it looks for global benefit, makes travel time of the main road vehicles to be slightly higher. The results for flow and fuel consumption are improved in both cases with CLA (figure 12).

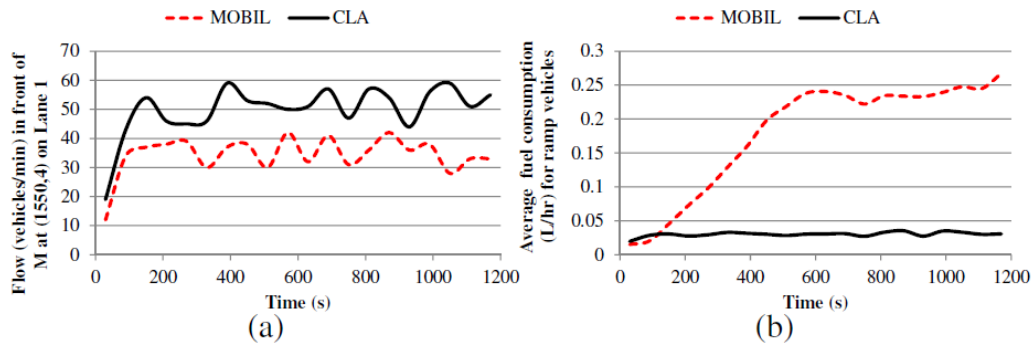


Figure 12. Comparison between MOBIL and CLA, regarding flow (a) and fuel consumption in on-ramp (b). Awal et al. (2015).

In a paper from Roncoli et al. (2016), a macroscopic model-based approach for the estimation of traffic conditions is presented. The average speed of connected vehicles in a segment of the highway (assumed to be similar to the average speed of all vehicles in the segment), total entry and exit flow of vehicles on the studied stretch and flow at ramps (or mainstream flow between them) are measured. A Kalman filter operates using that data to estimate density of vehicles in every segment.

The paper includes description of a case study, in which NGSIM (Next Generation Simulator) microscopic traffic data from a congested stretch of the I-80 highway in California is employed to evaluate performance of the estimation algorithm for different penetration rates of connected vehicles. Although it is shown that performance on traffic density estimation decreases for lower rates, for a reduced number of connected vehicles it is still accurate, with some delay comparing with real density trajectories, as it can be seen in figure 13.

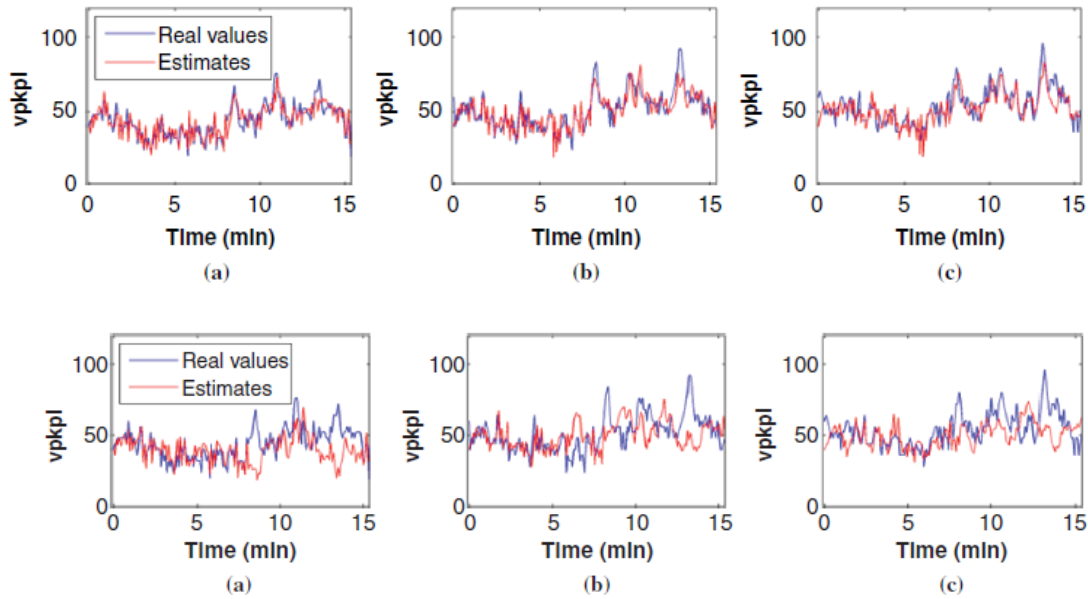


Figure 13. Trajectories of real and estimated densities with 100% (up) and 5% (down) penetration rate of connected vehicles. Roncoli et al. (2016).

A second case study is carried out with data from Schakel & van Arem (2014) from detectors on a stretch of the A20 in Netherlands, using a limited number of flow measurements and speed data, similar to what could be acquired from connected vehicles. To check realistic performance of the estimation, noise was added to both flow and speed measurements. In the next figures (14 & 15), the real and estimated congestion pattern of the traffic are compared.

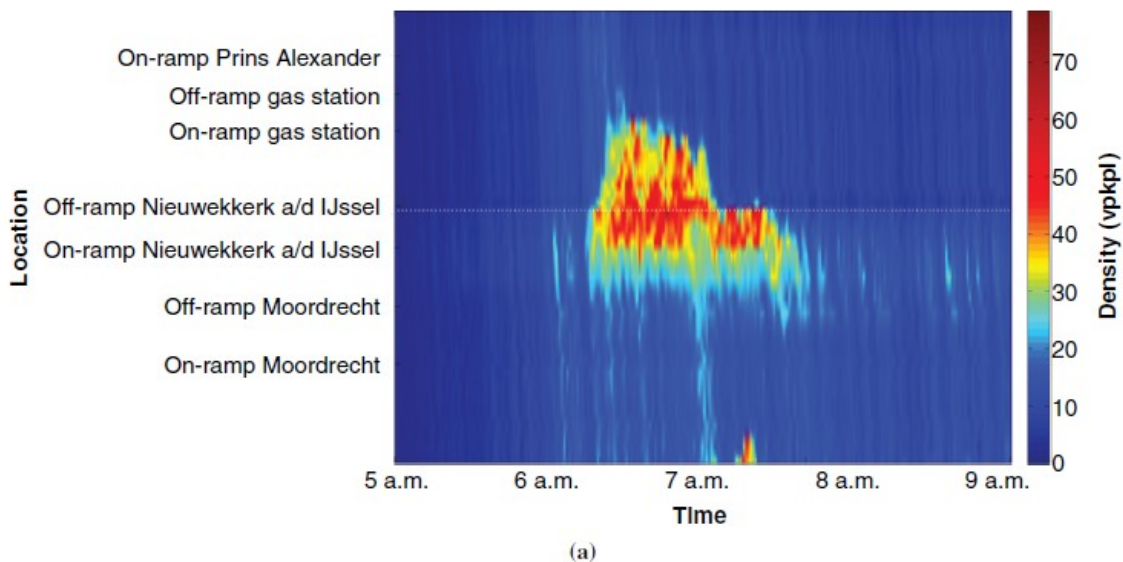


Figure 14. Real congestion patterns of A20 stretch. Roncoli et al. (2016).

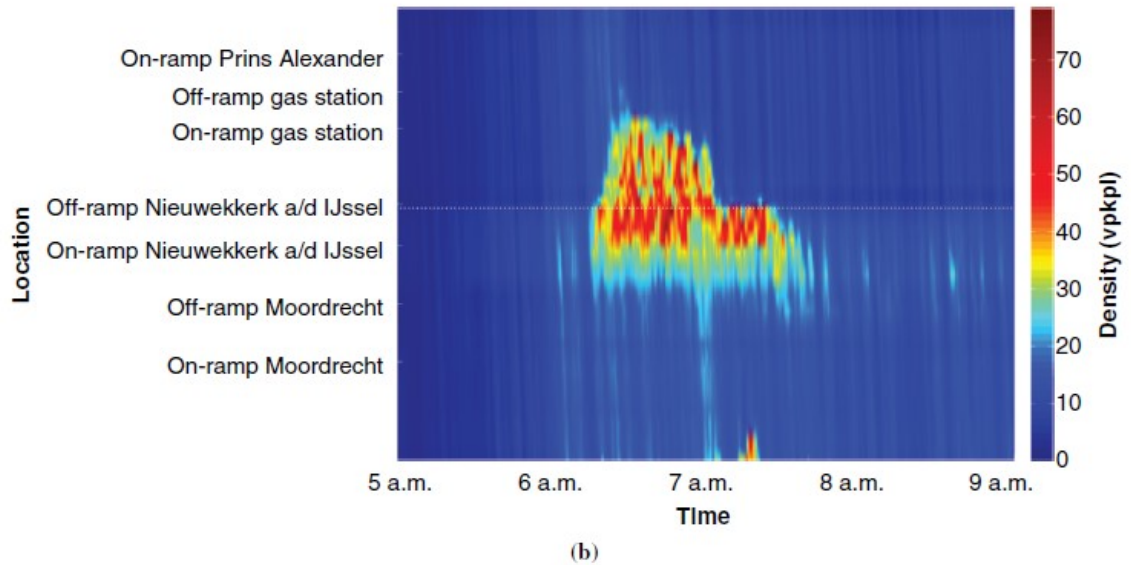


Figure 15. Estimated congestion patterns of A20 stretch. Roncoli et al. (2016).

In another paper from Roncoli et al. (2018), microscopic simulation is employed for evaluation of a per-lane density estimation that uses information gathered from connected vehicles, reducing the number of necessary detectors for spot flow measurements to a minimum. These simulations are executed considering different penetration rates of connected vehicles.

The highway stretch considered is divided in segments as in Roncoli et al. (2017), defining a cell for every segment of a lane. Mean speed, density and lateral flows of connected vehicles are measured for every cell. It is assumed that these measurements are representative of all vehicles in the stretch, because the behaviour of human-driven and connected vehicles is expected to be the same. Mainstream total flows of vehicles are measured with fixed detectors. With the gathered data, a Kalman filter is applied to estimate per lane total density.

AIMSUN is the software employed for microscopic simulation of the estimation scheme. This is also the software used for traffic simulation in the present work, so further information is provided in the next chapters. A stretch of the A-20 highway in Netherlands, the same road from Schakel and van Arem (2014) and Roncoli et al. (2016), is the case study network. It combines a lane drop and several on-ramps and off-ramps. Real traffic conditions are simulated using real demand data, including a congestion situation with spillback. To ensure the availability of data at every time step, even with low penetration rates of connected vehicles, the speed reported at the last time step or an average of the last reports within a larger time window could be considered if the report is not available. For lateral flows, when these are unstable a smoothed version of them is employed.

The performance of the estimation scheme is evaluated with a Coefficient of Variation (CV) for various penetration rates. The performance is found to be sensitive to this rate, as it decreases for lower values (figure 16).

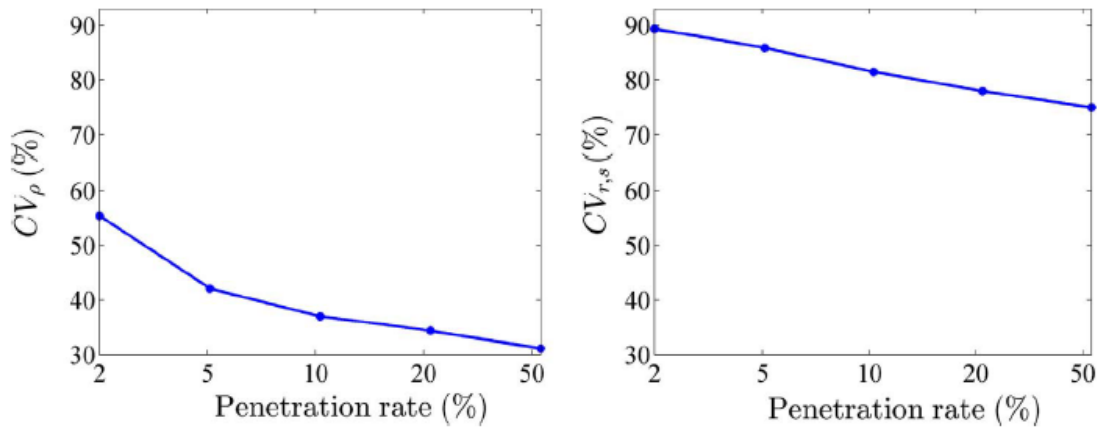


Figure 16. Performance of density (left) and ramp flow (right) estimations.  
Roncoli et al. (2018).

Sensitivity of the estimation regarding variation of the model parameters “p” (percentage of diagonal lateral movements) and “a” (smoothing factor) is also evaluated, performing an experimental analysis to obtain their optimal values. Through this analysis, authors discovered, for example, that for low penetration rates the smoothing effect improves performance and the optimal diagonal flow appears to be higher. In general, the estimation scheme “reproduces reliably the challenging traffic conditions in space and time”.

## 2.6. Psychological models

Most of the developments reviewed in this chapter are focused on behavioural models, both car-following or lane-changing approaches. The aim of these studies usually is focused on representing how vehicles move along and across the road. But other studies go deeper into actual driver behaviour, trying to understand the decision process underlying all vehicle movements in a road. Authors working on this field also want to find ways to influence this decision process, as advice systems that maintain people at the centre of the driving task could be implemented much faster in the near future than completely autonomous cars.

Van Driel & van Arem (2004) discuss in their research the way to integrate in-vehicle assistance systems from drivers’ perspective, considering their needs, preferences and capacities to process information, to develop safe systems. The aim of the project is also to evaluate the impacts of integrated driver assistance on both the driver and the traffic flow.

A pilot test carried out in order to develop a suitable survey showed that drivers tend to prefer the use of assistance systems on motorways, where these increases primarily comfort, rather than urban or rural roads where the driving task is more difficult and the main requirement from drivers to the systems is to increase safety. In the second case, for instance, assistance is appreciated during lane change manoeuvres (oncoming traffic warnings), but not in other kind of situations (intersections). The assistance in case of an imminent crash is perceived as very important.

Risto and Martens (2013) carried out an experiment to evaluate how drivers estimate the compliance rate of the users of a Connected Cruise Control (CCC) system. As other advisory systems, they rely on drivers to take control and just give instructions to improve their driving behaviour. Because of this, the effectiveness of the system is dependent on the response of the drivers.

This kind of systems give advice on optimal speed, driving lane and headway, aiming to optimize the distribution of cars, so an incentive for drivers to use them would be the possibility to appreciate the benefits on traffic flow and throughput. One possible effect of using these systems is when a general benefit in terms of traffic conditions is obtained, but not an individual one for each user. So, one major demand from users to be willing to follow advices is that the rest of users must follow them as well. The capability of drivers to appreciate the compliance of others is therefore an important factor.

To evaluate this capability, participants of the experiment were divided in two groups, one of them being aware just about the general purpose and goals of the system and the driver-in-the-loop approach used. In the other hand, the second group received more specific information about what kind of advices were given to users of the CCC and the goal of that advice. Also, three zones with different situations to trigger the advices (a lane drop, an on-ramp and a straight section) were considered, as well as different compliance rates for each situation (10%, 50% and 90%).

After completing different trails in the experiment, participants were asked to give an estimate of the compliance rate of other vehicles. Those who had received less information estimated compliance to be higher than participants from the other group, but no main effect of actual compliance rate on participants estimate of compliance was found. The authors suggest that uninformed drivers could have thought optimistically that the lack of congestion was due to a high level of compliance and looked to general traffic behaviour, as they did not have specific information, while informed users were more focused on particular indicators of compliance.

As a conclusion, the authors estate that the task of estimating compliance rates is difficult, and that when the success of a system depends on perceived compliance rate, information about how this system works should not be provided to users.

Another study involving drivers was carried out by van Driel et al. (2006), to evaluate the impact of a Congestion Assistant on participants driving behaviour and acceptance of the system. The assistant provides information and congestion warnings, and uses an active gas and a Stop&Go system. This ACC functions were evaluated later by van Driel and van Arem (2008) as it was detailed in a previous section of the present work.

The assistant delivers warnings in form of sound signal and an icon at first, displaying next a message informing the driver about the distance to a jam and applying a counterforce on the gas pedal if the approaching speed is too high. Once in the traffic jam, the Stop&Go system controls the situation after informing the driver about its activation with a spoken message. In general,

Van Gent et al. (2017) review in their study several research works on how to influence driver behaviour and develop a conceptual model to guide the research on this topic, to improve in-vehicle information systems efficacy. The requirements of the system are to be safe (not distracting the driver with the advices or overload his/her capabilities) and to persuade the driver to follow instruction and keep using the system.

Different persuasive methods are mentioned in the previous work, like gamification (introduction of game design elements), or behavioural economics (incorporation of insights of behavioural sciences into economics). These methods can be used together as various of the systems reviewed in the text do.

The proposed system by van Gent et al. (2017) to develop behavioural models is divided in three levels.

- In the system level safety and persuasiveness are evaluated.
- In the interface level the way to communicate with the driver is determined; an important point as delivering valuable advices would be worthless if they distract or saturate the driver. In a situation like this, when timing displaying messages is inappropriate or the information communicated is not relevant for the driver, probably the system would be ignored or deactivated, as the driver could considers than the system distracts him instead of helping.
- The last one, the driver level, is a guide to describe expected behavioural effects. This is directly connected with the previous level, as expected behaviour depends on how the message is transmitted.

### **3. Lane-changing model**

The main objective of this project is to investigate a lane-changing model that could be simulated and studied using a traffic modelling computer software. This model aims to improve traffic distribution across the lanes of the highway, to prevent congestion to happen due to underuse of road capacity. Vehicles are demanded to change lanes in advance, according to traffic flow situation in sections of the highway towards where they are traveling, but which are not in sight. So, lane-changes are triggered based on the traffic flow downstream of the current position of the considered vehicle.

In a normal situation, the area nearby the vehicle is evaluated by the driver, to be aware of other vehicles surrounding and estimate if any of the adjacent lanes offer advantages in terms of speed gain or in order to follow the predetermined route. For example, a driver could have the intention to move to the left lane if the vehicles ahead are traveling at a reduced speed, or at least at a speed lower than the desired one for the considered driver. The driver would be willing to change lane too if the vehicle in front is from a different class with respect to the traffic rules that must follow. For example, trucks and other heavy vehicles are usually restricted to lower speed limits, so they must use the rightmost lane every time it is possible to. Vehicles following would try to overtake them in order to increase their speed.

However, a driver could be willing to move to the rightmost lane, even if traffic is slower than in its current lane, if there is a need to leave the current road to follow a predetermined route. This way, the driver could prevent being stuck in its current lane, not valid to follow the intended route, if there is not an available gap in the adjacent one when reaching the off ramp or other kind of diversion that he/she has to take. Gipps (1986) mentioned how this aspect affects the way a lane-change model is configured, as the closer the distance is to an intended turn, more restrictive is the selection on valid lanes to change to.

To achieve the cited objectives, it is assumed that vehicles in the road are equipped with communication capabilities, so every one of them is aware of the position, speed, movements across the road and other parameters of surrounding vehicles that are necessary to check the traffic situation in a deeper way than just considering information gathered from the driver or by the sensors installed in the vehicle. This assumption is made to avoid limitations when developing a logic for the model, so every accessible information from vehicles in the simulation program can be used to evaluate the situation in the network and allow other vehicles to take actions in consequence with that data.

#### **3.1. Model structure**

To evaluate the situation in the highway, regarding traffic flow, different approaches have been considered during the development of the lane-changing model. Basically, one quantitative aspect of the traffic, like average speed of a platoon of vehicles or the average headway between them, is compared among lanes to distinguish which one offers best conditions for driving. The variables that are considered in this evaluation are the following:



**Headway (space).** It is described as the distance between the front end of a vehicle and the rear end of the vehicle ahead, although this distance is called “gap” in other texts, while headway usually refers to the distance between both front ends of the considered vehicles. In this document, the term gap is used mainly to describe that distance for vehicles in adjacent lanes. Headway is directly related with density, because once both length of vehicles and headway between them are known in a certain section of the road, density can be easily calculated. At a microscopic level, this parameter is more helpful than density, as it can be locally controlled by the driver, that can choose to maintain a larger headway when driving at high speed for safety reasons or reduce it in congested situations.

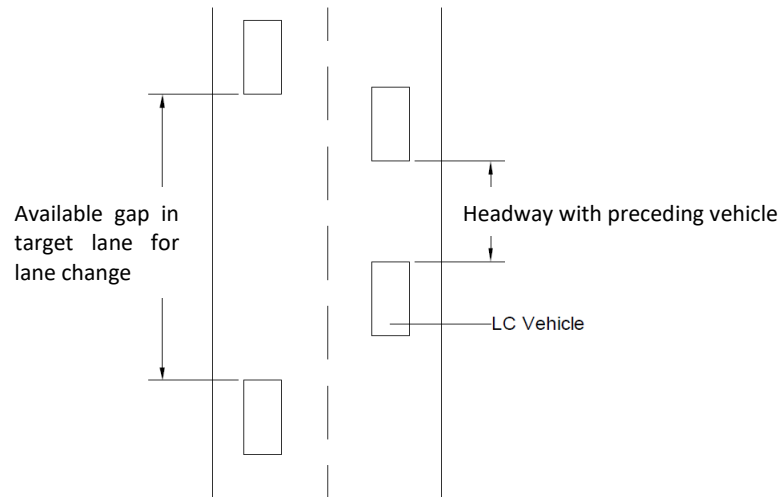


Figure 17. Headway and gap

**Time headway.** Headway can be described as a temporal parameter, being the time that takes for a vehicle to reach a certain point of the road after the vehicle that is following has passed it. It is as well the result of dividing the space headway by the speed at which the following vehicle is traveling, considered that it is constant.

Measures of traffic situation are used during the decision process of lane-changing. The final objective for the vehicle is to reach its destination in the shortest time. Thus, attempting to use always the fastest (or less dense) lane every time it allows the vehicle to follow its route. But in the case of the developed behavioural model, instead of taking these measures from the area surrounding the vehicle, traffic conditions at a certain distance downstream of the vehicle current position are taken into account (figure 18), trying to anticipate a potential congestion. This way, vehicles would distribute across all the lanes, being those where traffic is more fluent more appealing for vehicles upstream of that location.

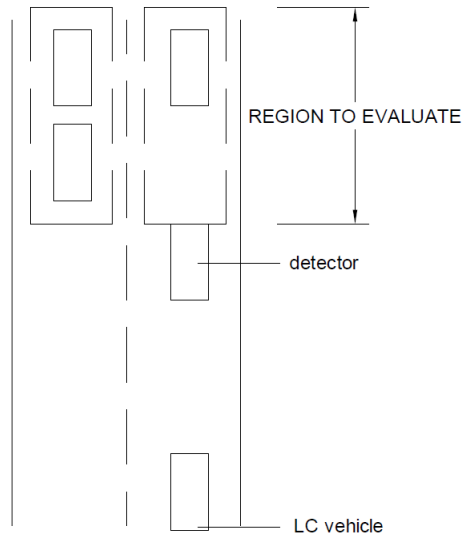


Figure 18. Evaluation of lane situation downstream for LC vehicle

To make this possible, the aforementioned variables have to be measured by each vehicle on the road in order to classify lanes, establishing which ones are more appealing. A detector vehicle is selected, acting as a reference to take the measures. This detector is the first vehicle placed at a certain distance from the considered vehicle to evaluate the lane-changing action, like in figure 19.

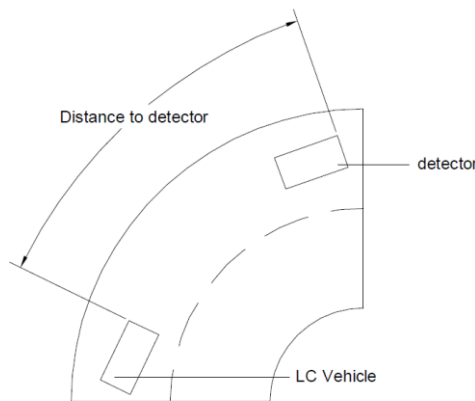


Figure 19. LC vehicle and selected detector

Once the detector is established, it gathers all the necessary information from vehicles ahead in its current lane. A maximum number of vehicles is considered, and these must be within a distance from the detector, except for the last one in the group, that is the placed just after this detection distance. In case the vehicle directly in front the detector (this means, its leader) is already beyond this distance, only that one would be considered to take measures. In figure 20, it is depicted how the selection must be carried out. This way, even when there is no any vehicle within the considered detection distance, one beyond that limit will be considered.

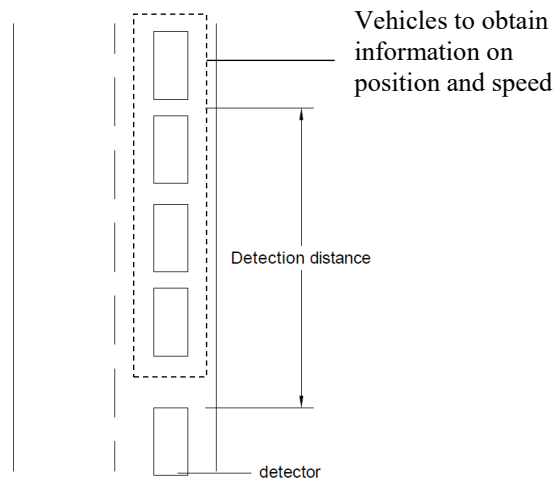


Figure 20. Vehicles to evaluate from the detector

Speed and headway between these selected vehicles is measured, as well as the time headway, that is directly calculated as it is mentioned above. The instantaneous speed of the vehicles at the moment of measuring is considered constant to calculate this last parameter. The detector repeats the same process with vehicles in the left and right adjacent lanes. Then, average speed and headway of this groups of vehicles is calculated for every lane, so it is possible to distinguish where vehicles are moving faster, or in which lane it is more likely to find a suitable gap to change.

Based on this information, the driver would be suggested to change lane by a message displayed on the vehicle dashboard. The lane suggested would be that with a higher average speed, or the one with larger average gaps, depending on how the behavioural model is set up. A full compliance rate for drivers following those advices is considered.

### 3.2. Implementation of the model

The lane-changing model has been coded in C++ language, as a modified version of a software provided by Konstantinos Mattas from the JRC in Ispra (Italy), one of the authors of the paper (Makridis et al. 2018) on impact assessment of Cooperative Active Cruise Control (CACC) systems on a case-study of the ring road of Antwerp. This computer program has been developed with the microSDK toolkit included with the AIMSUN software. Only the lane-changing model section of the program has been modified for this project. The car-following model applied is the original one present on the provided file, a modified version of the default AIMSUN version, that is based on Gipps (1981) model.

**Note: The full C++ code of the lane-changing model to be implemented in AIMSUN is presented in the Appendix 1.**

Some parameters are defined at the beginning of the lane-changing model file, so they can be easily modified to change the way vehicles behave:

- **Distance to detector.** Minimum distance between the front end of the vehicle that is considered to follow the lane-changing model (LC vehicle from this point forward) and the front end of the vehicle acting as a detector (detector vehicle).
- **Maximum number of vehicles ahead.** It is the number of vehicles that the detector considers to take all the traffic situation measures. The detector pick this number of vehicles from its own lane to evaluate flow in the current lane, or from adjacent lanes to evaluate left and right traffic flow.
- **Maximum distance ahead (detection distance).** It is, as explained above, the distance in front of the detector vehicle within the group of vehicles acting as sensors are included, except for the last vehicle in the group, that is placed beyond this distance.
- **Lane change threshold.** It is the threshold applied in every comparison between lane conditions to avoid an excess of lane changes, even if traffic conditions in other lanes are better. If the ratio between the considered variable values in an adjacent lane and the current lane is still under the threshold, the LC vehicle would remain in its current lane.
- **Fluent Speed.** It is the speed considered to offer a fluent traffic situation. Its value can be easily set up, but it is fixed for all the vehicles in the network. It also limits the number of lane changes, so when a vehicle is already traveling with a speed higher than this, no lane-changing manoeuvres are triggered by the model. This parameter was defined to avoid certain vehicle behaviours observed during the simulation runs, like groups of vehicles moving together back and forth from one lane to the previous one. It is introduced in a similar way than in the In-Car advice system from Schakel & van Arem (2014), which employs a speed limit to avoid giving advice on reducing headway to drivers when they are traveling at free-flow speed (considered in this particular case to be higher than 80 km/h).
- **Free flow rate.** As an alternative to using the fluent speed threshold, it can be considered that when a vehicle is traveling with a speed such as the rate between this and the free flow speed is high enough, the vehicle is not in a congested situation. For the simulation cases in which this factor is employed, a rate of 0,6 (60% of free flow speed) is applied to the free flow speed of the detector vehicle to set the non-congestion threshold.

**Note for the AIMSUN case:** One initial condition applied for the vehicles is to follow the internal default behavioural model of AIMSUN for lane-changing, based on the model by Gipps (1986), if they are in an on-ramp lane or just next to one. This step is considered necessary for the same reason than in the case of the fluent speed limit, as erratic behaviour for vehicles on lane-changing manoeuvres to enter the highway from on-ramps was observed, in a similar way than in the case mentioned before, moving back and forth from the ramps.

Next, it is presented the procedure that the lane-changing model follows to evaluate if an advice to change lane must be sent to a vehicle.

**Note:** The vehicle considered to change lane is named as “LC vehicle”. The “detector” is the reference vehicle to collect information on speed and position from other vehicles on the road.

>> Creation of a “detector” vehicle class.

>> If the initial parameter “distance to detector” = 0, then → “LC vehicle” = “detector”

>> In opposite case → Pick the vehicle preceding the “LC vehicle” as the detector

In the case the “LC vehicle” is also the “detector”, all data about position and speed from vehicles would be gathered using the vehicle itself as the reference to take those measures, similar to what a human driver would do. However, this analysis capacity is above human abilities, because drivers would try to estimate this information considering just the vehicles within their vision range. If the preceding vehicle is selected as the “detector”, then:

>> Get positions of “LC vehicle” and “detector”.

>> Position of the LC vehicle =  $pos_{LC}$

>> Position of the detector vehicle =  $pos_{det_j}$

**Note:** The subscript “j” is the position of the current considered detector in the line of traffic ahead of the LC vehicle (with  $j = 1$  for the first vehicle). The LC vehicle is at position 0.

>> Measure the distance between them.

**Note for the AIMSUN case:** It is relevant to mention that networks represented in AIMSUN micro simulator are divided in sections, known as “A2KSections”. In the case of the GUI, networks are divided in “GKSections”, that can include one or more A2KSections. Position of a vehicle is expressed using the beginning of the section where it is at the specified time as a reference. The “shift” variable indicates the distance between the beginnings of the respective sections of two vehicles, as it is shown in figure 21. In this case, between the LC vehicle and the detector. If both vehicles are in the same section, this value is null.

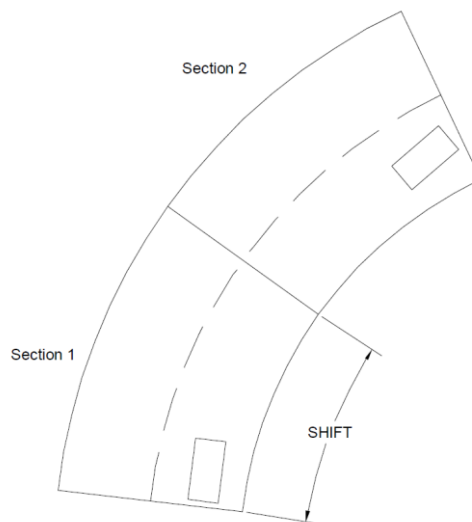


Figure 21. Shift in AIMSUN

In a general case, using another traffic simulation software, the division of a network in different sections will be considered as well. To measure the distance between two vehicles in the same lane, the process would be similar. First, the position of each vehicle in its section, defined as the distance between the beginning of the section and the front end of the vehicle in a certain lane, is obtained. Then, to calculate the “shift”, it is necessary to add up the length of the group of sections between both vehicles, from the section where the first vehicle is to the one before the second vehicle current section. For example, in figure 21, section 1 is at the same time the current section of the first vehicle and the one before the second, so the “shift” is equal to the length of just that section.

>> Calculate distance between the LC vehicle and the detector:

>>  $n_j$  = nth section within the group of N sections of the road between the vehicles in the jth and the (j-1) position, respectively. In the following case, the N1 would be the one before the section of the first vehicle ahead of the LC vehicle.

>> Length of the  $n_j$  section =  $l_{n_j}$ . Measured as the length of the middle line of the considered lane in that section.

$$shift = \sum_{N_j}^{n_j} l_{n_j} \quad (3)$$

$$distdet_j = posdet_j + \sum_{N_j}^{n_j} l_{n_j} - posLC_0 \quad (4)$$

If the selected detector vehicle is within the specified “distance to detector”, then a loop process is initialized to select a new detector.

>>WHILE  $distdet_j <$  “distance to detector” defined parameter  $\rightarrow$  select a “new detector”, picking the vehicle preceding the old detector. To define parameters regarding this new detector, the subscript is increased every time the loop is entered. Inside the loop:

$$j = j + 1 \quad (5)$$

>>Get the position of the “new detector”  $\rightarrow posdet_j$

>>Update the variable  $distdet_j$ , adding the distance between the “new detector” (j) and the old one (j-1), to obtain the separation between “LC vehicle” and the “new detector”  $\rightarrow$

$$distdet_j = distdet_{j-1} + posdet_j + \sum_{N_j}^{n_j} l_{k_j} - posdet_{j-1} \quad (6)$$

The process to obtain all the data from vehicles in the lane to calculate their average values, taking the detector as a reference to start, is very similar. The system picks the first vehicle ahead of the detector at first, and gets its position, speed and length, calculating the headway between the detector and this new vehicle and the time headway as well.

>> Select the “vehicle ahead”, the one preceding the final chosen “detector” (the vehicle in the j position).

>> Get position and length of the “vehicle ahead”.

>> Position of the vehicle ahead =  $posahead_k$

>> Length of the vehicle ahead =  $lahead_k$

**Note:** The subscript “k” is the position of the current considered vehicle in the line of traffic ahead of the detector (with k = 1 for the first vehicle).

>> Get speed of the “vehicle ahead”.

>> Speed of the vehicle ahead =  $spahead_k$

>> Calculate headway between the “vehicle ahead” and its follower (in this first case, it is the “detector”).

$$hwahead_k = posahead_k + \sum_{Mk}^{mk} l_{mk} - posdet_j - lahead_k \quad (7)$$

>> Calculate time headway → “time headway” = “headway” / “speed”

$$thwahead_k = \frac{hwahead_k}{spahead_k} \quad (8)$$

The distance between the detector and the vehicle ahead is calculated as well.

>> Calculate distance to the vehicle to take measures from →

$$distahead_k = posahead_k + \sum_{Mk}^{mk} l_{mk} - posdet_j \quad (9)$$

A set of variables are defined to store the addition of speed and headway values from all the vehicles considered in the measurement process. Example with speed →

>> Calculate speed sum. At this point, the addition is just equal to the first considered vehicle measure, as  $spsum_{k-1} = spsum_0 = 0$ .

$$spsum_k = spsum_{k-1} + spahead_k \quad (10)$$

>> Calculate headway (and time-headway) sums in the same way.

>> k acts as well as a counter of the vehicles ahead of the detector from which measures will be taken.

In case this vehicle ahead is within the limit “detection distance”, a new loop is executed. The loop will continue while the selected vehicles are within this distance limit and the number (k) of vehicles does not exceed the “maximum number of vehicles ahead” to consider.

>> WHILE  $distahead_k < \text{“maximum distance ahead”}$  AND  $k < \text{“maximum number of vehicles ahead”}$  → select the vehicle preceding the “vehicle ahead” (“new ahead”). Inside the loop:

$$k = k + 1 \quad (11)$$

>> Get position and length of “new ahead” →  $posahead_k$  and  $lahead_k$   
 >> Update distance between the “detector” and the last “new ahead” vehicle.

$$distahead_k = distahead_{k-1} + posahead_k + \sum_{Mk}^{mk} l_{mk} - posahead_{k-1} \quad (12)$$

>> Get speed of the “new ahead” vehicle →  $spahead_k$   
 >> Calculate headway and time-headway for “new ahead”

$$hwahead_k = posahead_k + \sum_{Mk}^{mk} l_{mk} - posahead_{k-1} - lahead_k \quad (13)$$

>> Update speed, headway and time-headway. Example with speed →

$$spsum_k = spsum_{k-1} + spahead_k \quad (14)$$

Once the process exits the loop, the average speed, headway and time headway are calculated.

>> Calculate average values of the variables for the current lane. Example for the speed →

$$avspeedcurrent = \frac{spsum_k}{k} \quad (15)$$

This procedure is repeated to obtain values from the left and right adjacent lanes.

>> Get position of the vehicles ahead in the left and right adjacent lanes.

>> Position of the vehicle ahead in left lane =  $posleft_l$

>> Position of the vehicle ahead in right lane =  $posleft_r$

**Note:** Subscript “l” and “r” are the positions of the two current considered vehicles in the lines of traffic in the left and right lane, respectively, ahead of the detector (with  $l = 1$  and  $r=1$  for the first vehicle in each lane).

**Note for the AIMSUN case:** The function used to get a vehicle in an adjacent lane picks both the vehicle ahead of the reference one (the detector) and the follower vehicle in the adjacent lane. The relevant vehicle is the one ahead, the downstream vehicle. The target lane must be specified as well as an argument of the function to access this vehicle, so this argument of the function is set to -1 for the left lane or 1 for the right lane.



The process to get measures from the vehicles ahead of the “detector” in adjacent lanes is the same as for the current lane, but considering the “downstream vehicle” as the first to take measures from instead of the “vehicle ahead”. Average values of the considered variable are obtained for each lane. Example for speed:

$$avspeedleft = \frac{spsum_l}{l} \quad (16)$$

$$avspeedright = \frac{spsum_r}{r} \quad (17)$$

**Note for the AIMSUN case:** AIMSUN considers obstructions, stop lines or traffic lights as vehicles for its car-following model. This feature is coherent with statements from Gipps (1986), as car-following models are “relatively unaffected by what constitutes the previous vehicle”. To differentiate real vehicles from these special objects, specific microSDK functions can be employed, that take into account only actual vehicles. For example, “getRealUpDown” function present in the code in Appendix 1.

The next step is to find out which lane offers the best traffic conditions for the LC vehicle. One of the aforementioned variables (average speed, headway or time headway) is chosen, and its value in the current lane is compared to that for both adjacent lanes. If the value the variable takes in the adjacent lane is higher than in the current lane, multiplied by the adjusted lane change threshold, that lane becomes the “target lane”

**Note for the AIMSUN case:** The microSDK of AIMSUN includes a function to check that not solid lines on the road will be crossed when moving to the next lane, and that other lane-changing manoeuvres are not being executed at the same time by the LC vehicle. These conditions must be satisfied to accept the “target lane”.

>> Check if it is possible to change to left/right adjacent lane.  
 >> Evaluate adjacent lanes conditions. Example for speed, comparing current and left lanes:  
 >> IF  $avspeedright > LCthreshold \cdot avspeedcurrent \rightarrow$   
 >> Variable “change left” is positive and “lane change direction” = left.

If both lanes are considered valid to change to, and both offer advantageous conditions respect to the current one, then the comparison is made between them.

>> Compare left and right lane conditions:  
 >> IF  $avspeedleft > avspeedright \rightarrow$   
 >> Variable “lane change direction” = left

However, if the calculated average speed in the LC vehicle current lane downstream of its position is higher than the free flow speed of the detector vehicle, multiplied by the free flow rate, no lane changes are triggered. This measured is applied to avoid unnecessary lane changes.

>> IF *avspeedcurrent* > *freeflowspeed* · *freeflowrate* →

>> Variable “lane change direction” = 0

The alternative version is to use a fixed value limit for the speed, defining it with the rest of initial parameters:

>> IF *avspeedcurrent* > *fluentspeed* →

>> Variable “lane change direction” = 0

Once the lane-changing direction is known, the manoeuvre is started. The current model used is a modified version of an AIMSUN sample lane-change model. First, the LC vehicle gets its equivalent position in the target lane, like if it was projecting its shadow. Then, it locates the downstream and upstream vehicles in the target lane, as it was done to take traffic measures before. Afterwards, it checks for an acceptable gap in the target lane and if it is valid, the LC vehicle is added to a list of vehicles to change lane. The AIMSUN algorithms decide the priority of all the vehicles willing to change lane.

A process to look for a gap for cooperation in the target lane is executed too. Like in the previous case, the LC vehicle gets the position of vehicles in the target lane and it is introduced in a list to cooperate with them. The behaviour of the upstream vehicle is conditioned by its cooperation parameters.

Once the vehicle was allowed to execute the lane change, an advice message would be sent to the considered lane-changing vehicle and displayed to the driver.

## 4. Simulations

The lane-changing model is investigated by running traffic simulations using the AIMSUN software. The software allows to build a network, configure traffic conditions (input flows, composition of the traffic...), apply a behavioural model for vehicles and run simulations. The software offers statistic results on travel time, delay time (difference between expected travel time with ideal conditions and actual value), traffic flow, density...

The test network employed to run the simulations, the ring road of Antwerp, was provided (like the initial AIMSUN model files) by Konstantinos Mattas from the JRC. Results of impact assessment of CACC systems in this network are presented in Makridis et al. (2018). The ring road contains sections of different highways.

Two different scenarios have been employed to simulate the lane-changing model, depending on the behavioural model algorithms employed for vehicles. For the first simulations, the AIMSUN automated vehicles (AVs) scenario was used, but the results obtained were not satisfactory, with too high delay times and large travel times, and vehicles showing a strange behaviour during the simulation run, as it is detailed in the next section. As it was explained by Konstantinos Mattas from the JRC later, this was probably due to a problem with DLC (Discretionary Lane Changes) algorithms used in the AIMSUN behavioural model for the AVs scenario, because “AVs are expected to be more conservative than human drivers and have larger gaps [with their respective leaders]”

However, simulations using the second considered scenario, designed for CAVs (Connected Automated Vehicles), returned results that are much more feasible. With this configuration, the gap acceptance process works properly and vehicles behave in a more realistic way, accepting reasonable gaps. In both scenarios, AIMSUN default car-following model is employed, confirming that differences appreciated in the behaviour of the vehicles between scenarios are probably due to internal issues of the software.

### 4.1. Automated Vehicles scenario

In this scenario, it was employed the fluent speed threshold to limit the number of lane changes, instead of the free flow threshold. The criterion to choose which lane offers the best conditions, was to compare average speed among them. The model initial parameters were set to:

- Maximum number of vehicles: 8
- Lane change threshold: 1,3
- Fluent speed: 50 km/h

	Distance to detector [m]	Detection distance [m]
Simulation run 1	200	200
Simulation run 2	1000	400
Simulation run 3	0	400
Simulation run 4	1500	500

Table 1. Initial parameter values for AVs simulation scenario

Selecting a small distance between the vehicle and the detector, fluctuations in traffic can be observed during the simulation, with vehicles changing constantly from one lane to the adjacent, back and forth.

During the simulation runs, it was observed that the leftmost lane is barely used by vehicles, even during congested situations, so highway total capacity is not used. This is noticeable for every simulation run with the AV scenario, as it is shown in figures 22 and 23 that show sections of the highway where congestion has occurred.

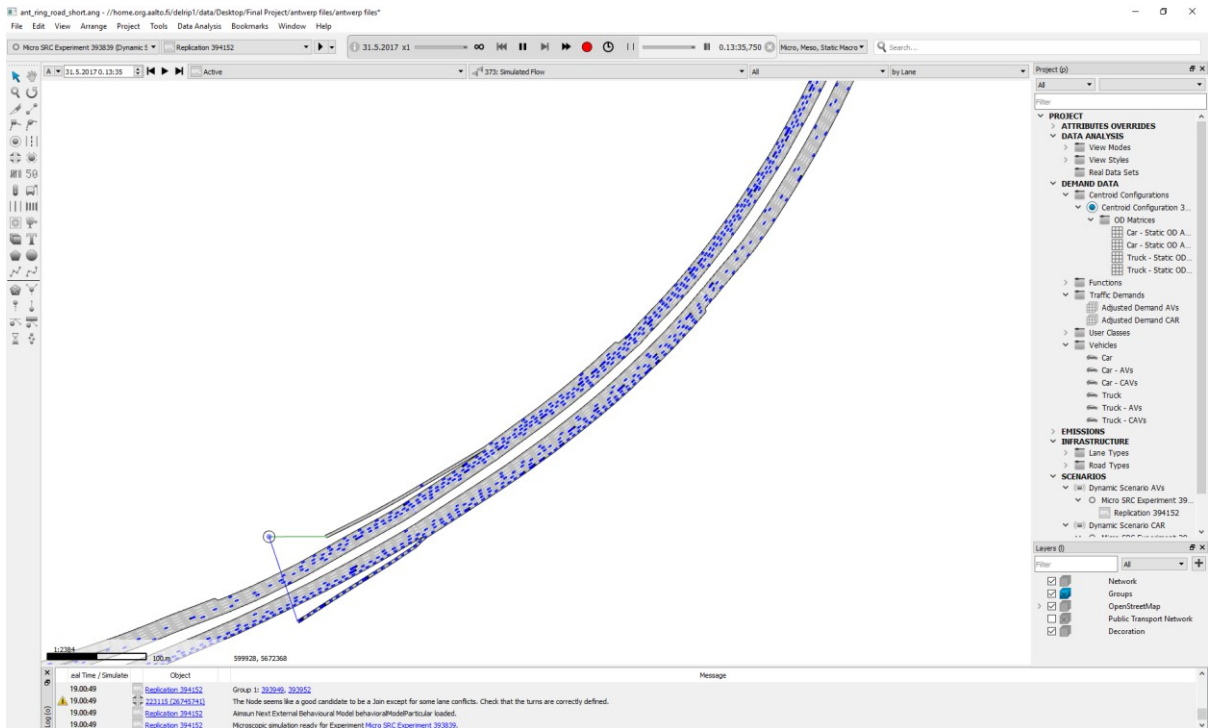


Figure 22. AIMSUN software interface showing a south section of the network, simulation run 2.

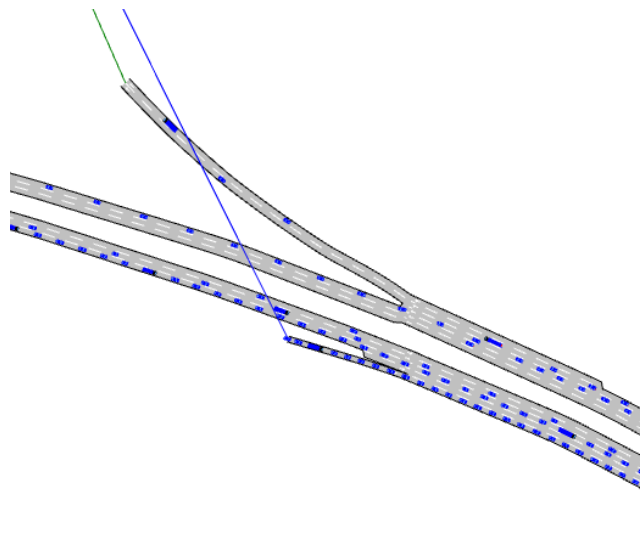


Figure 23. Detail view of an on-ramp in a southwest section of the network, simulation run 4.

In the section of the network displayed in figure 23, it can be observed how vehicles in the network do not yield for those trying to enter and merge from the on-ramps, in both directions of the traffic.

In the following graphics (figures 24-27), it is shown how results on average travel time, delay, traffic flow and density vary between simulations.

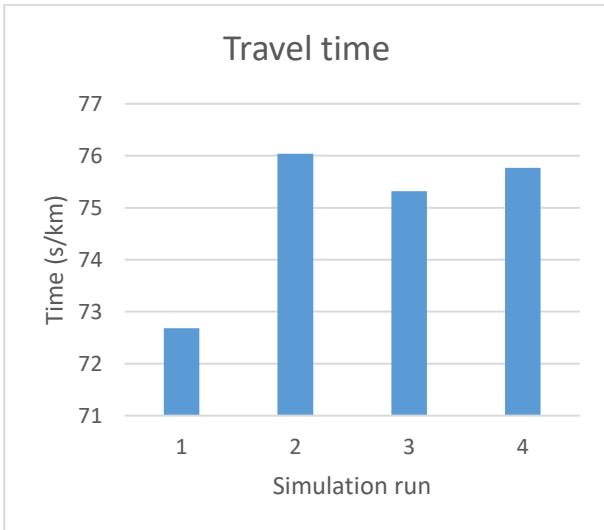


Figure 24. Vehicle travel time for each simulation.

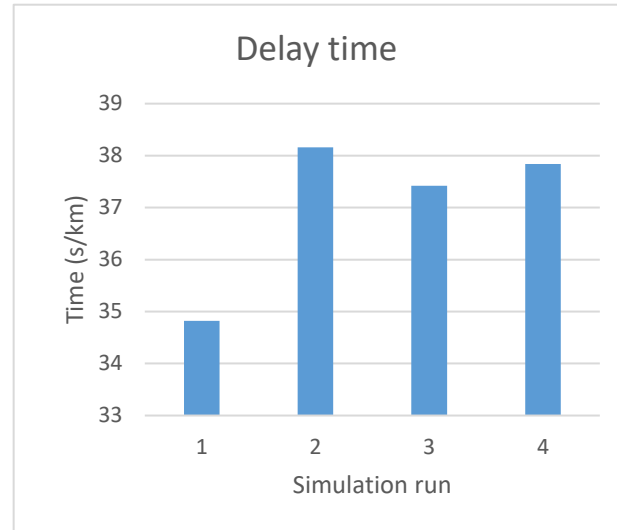


Figure 25. Vehicle delay time for each simulation.

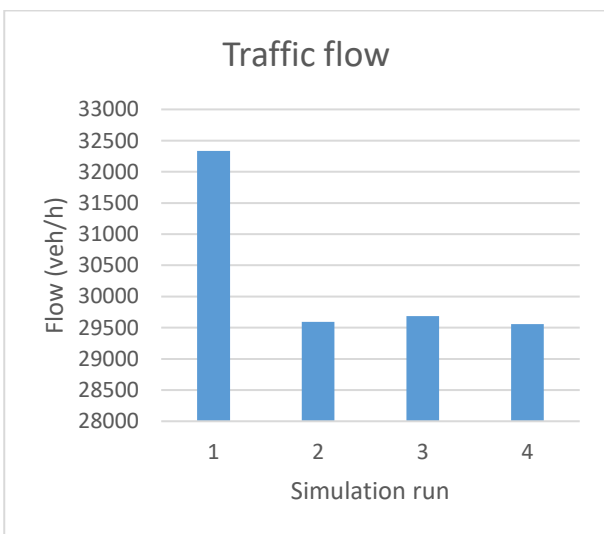


Figure 26. Traffic flow for each simulation.

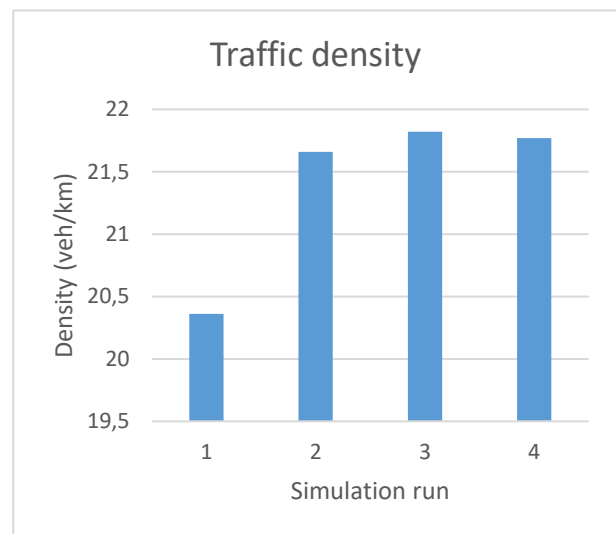


Figure 27. Traffic density for each simulation.

As it was mentioned before, these results are not very reliable. However, some aspects of the simulations can be commented. Traffic flow is higher in the first run case, as figure 26 shows, and thus both travel time and delay time are lower than in the rest of cases. This could be due to the lower detection distance applied (200m), comparing with the other runs (400-500m), so fewer vehicles would be considered when calculating the average speeds and headways to classify lanes, and thus, these average values employed in lane comparison will be more unstable, triggering a larger number of lane changes. On the other hand, the number

of vehicles to calculate these average values is limited to 8, so in congested situations the 200m detection distance should be enough to select them.

Another notable feature is that in the third simulation run, delay time decreases while traffic density rises. According to these results, in this case, capacity of the highway is used in a higher rate, although still far from its maximum.

This analysis has been carried out based just in assumptions. Therefore, as it was previously stated, results are of limited validity, and limited interest, because of the incorrect behaviour showed by vehicles in the AV scenario.

## **4.2. Connected Automated Vehicles Scenario**

The CAVs scenario allowed to obtain more realistic results. Several cases were simulated, comparing again average speed between the current lane where the vehicle is in and the potential left and right target lanes. Two different “lane change thresholds” were applied to decide which lane to use:

- Average speed in target lane  $> 1,3$  times the average speed in current lane.
- Average speed in target lane  $> 1,1$  times the average speed in current lane.

In the second case, more lane changes manoeuvres should be executed, as these are considered even for a marginal improvement of traffic conditions in an adjacent lane. Three different values of “distance to detector” are considered. In the cases that value is zero, the vehicle acting as the “detector” is the one to change lane. The other options are to pick a detector that is further than 1000m or 1500m downstream, respectively.

The other initial parameters take the following values in every simulation:

- Detection distance: 500m
- Maximum number of vehicles for detection: 8
- Free flow rate: 0,6

The next graphics (figures 28-33) show how some traffic characteristics change with respect to the distance to the detector (for a lane change threshold = 1,3). The values shown are average values considering all the vehicles in the network (cars and trucks):

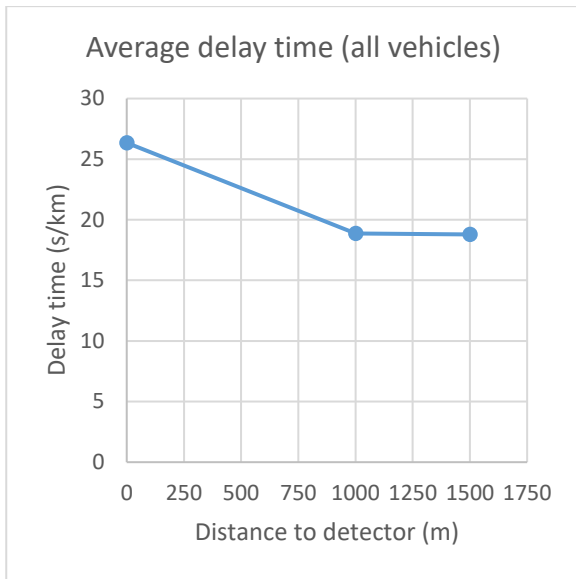


Figure 28. CAVs average delay time for LC threshold = 1,3

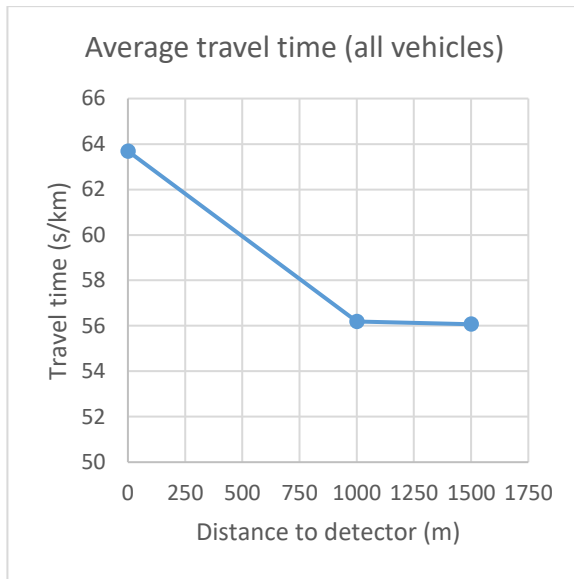


Figure 29. CAVs average time for LC threshold = 1,3

The average delay time (and thus, total travel time) is slightly reduced when a vehicle considers the situation downstream instead of checking its surroundings. According to these results, the lane change model is working correctly, as changing lane in advance allow vehicles to get closer to the ideal situation (when there is no delay in travel time, as driving at the desired speed of the driver is always possible). There are no big improvements on travel time between the case checking traffic conditions 1000m downstream and the one checking 1500m downstream.

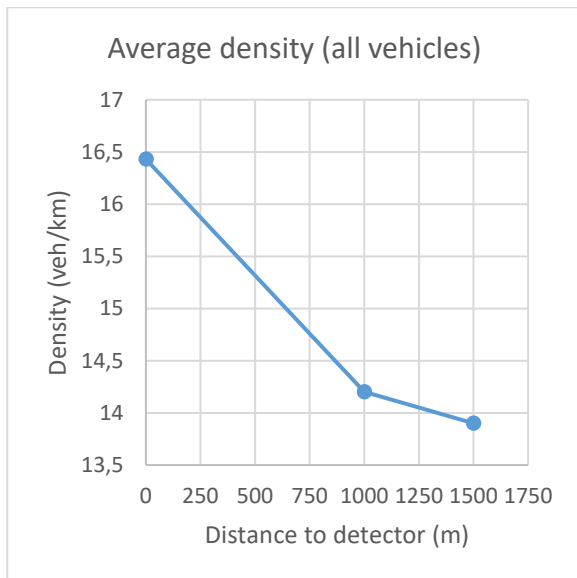


Figure 30. CAVs average density for LC threshold = 1,3

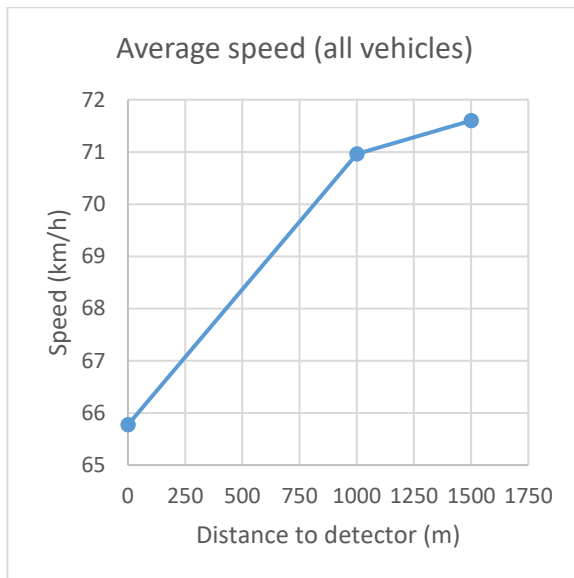


Figure 31. CAVs average speed for LC threshold = 1,3

Density decreases for larger separations between the lane-changing vehicle and the detector. As potential congestion is detected in advance, vehicles act consequently and move to lanes that are less busy in the downstream section of the road towards they are going. Vehicles decide to change to lanes that allow them to drive faster, as the speed results show

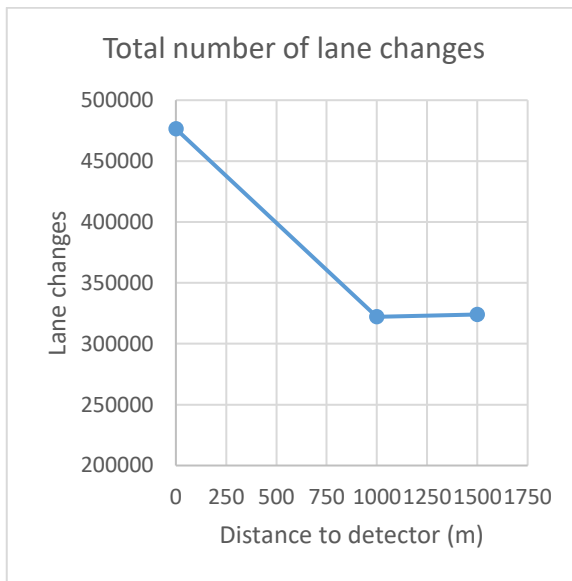


Figure 32. CAVs total lane changes for LC threshold = 1,3

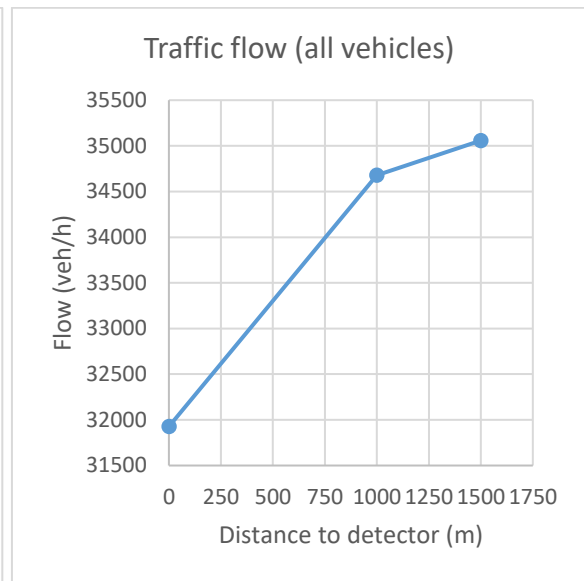


Figure 33. CAVs traffic flow for LC threshold = 1,3

The total number of lane changes decreases when vehicles execute the manoeuvre in advance. As density gets lower and speed gets higher, the flow of vehicles improves.

The next graphics (figures 34-39) show variation of traffic characteristics with respect to the distance to the detector (for a lane change threshold = 1,1):

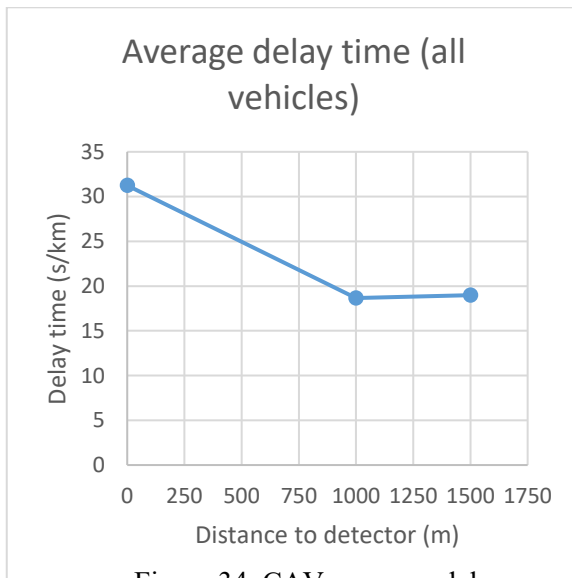


Figure 34. CAVs average delay time for LC threshold = 1,3

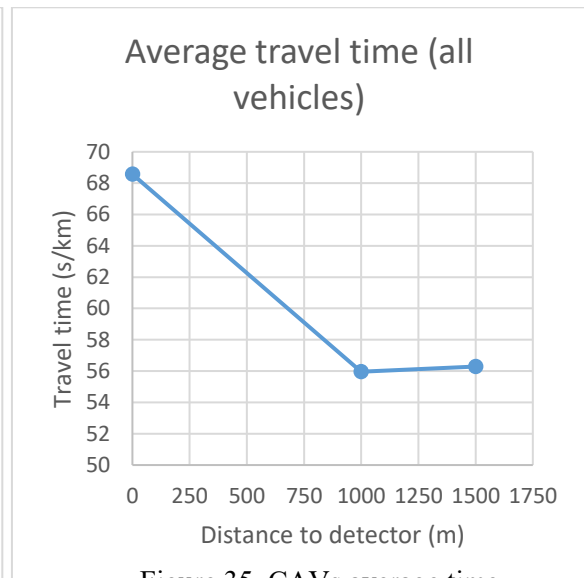


Figure 35. CAVs average time for LC threshold = 1,3



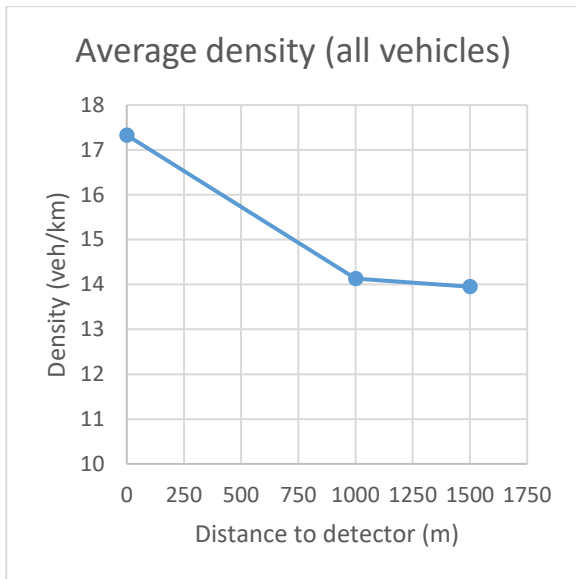


Figure 36. CAVs average density for LC threshold = 1,1

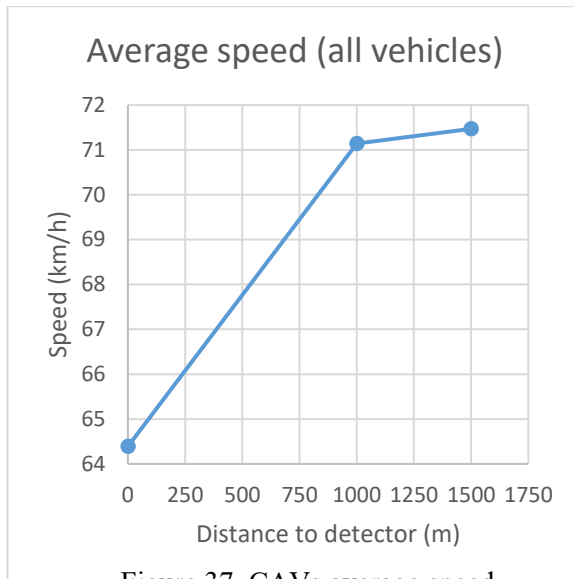


Figure 37. CAVs average speed for LC threshold = 1,1

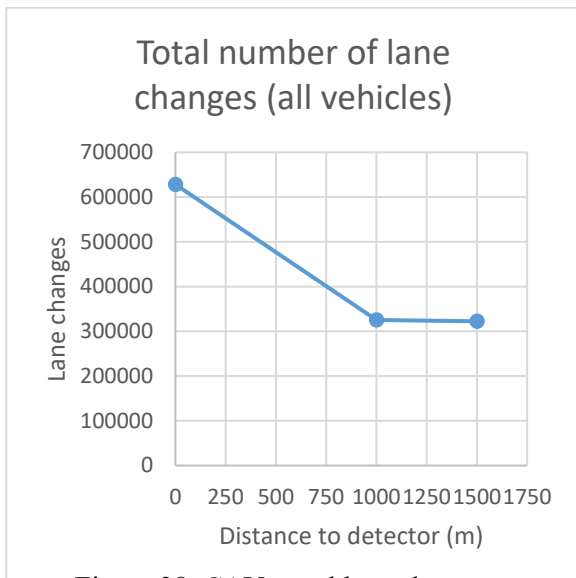


Figure 38. CAVs total lane changes for LC threshold = 1,1

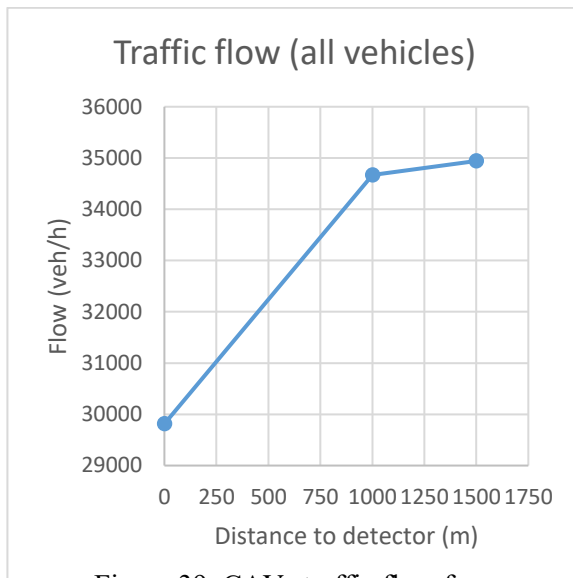


Figure 39. CAVs traffic flow for LC threshold = 1,1

Selecting the lower “lane change threshold”, results barely change for the cases checking downstream traffic, obtaining similar delay time, speed or flow values, but they do for the case with “distance to detector” = 0. As every vehicle considers the situation just ahead to take the decision to change (and it changes for every minimum speed gain), it is possible that, when some vehicle move to an adjacent lane, the situation perceived by the vehicles following it changes, which probably causes more lane-change manoeuvres. In this case, the situation is more unstable.

In table 2 and table 3 below, a summary of results from simulations is presented, including a case in which the AIMSUN default lane-change model is employed.

		Parameters							Units
Lane change threshold	Default lane-change model	1,3							
Maximum number of vehicles ahead	Default lane-change model	8							Veh
Detection distance	Default lane-change model	500							m
Free-flow rate	Default lane-change model	0,6							
Distance to detector	Default lane-change model	0		1000		1500		m	
Time series	Mean value	$\sigma$	Mean value	$\sigma$	Mean value	$\sigma$	Mean value	$\sigma$	
Delay Time - All	18,94	25,42	26,35	30,08	18,88	22,33	18,79	24,92	s/km
Delay Time - Car	19,08	25,06	26,76	29,9	19,06	21,76	18,95	24,71	s/km
Delay Time - Truck	17,48	28,74	22,16	31,62	17,08	27,42	17,12	26,85	s/km
Density - All	13,95	N/A	16,43	N/A	14,2	N/A	13,9	N/A	veh/km
Density - Car	11,9	N/A	14,2	N/A	12,14	N/A	11,86	N/A	veh/km
Density - Truck	2,05	N/A	2,23	N/A	2,06	N/A	2,04	N/A	veh/km
Flow - All	34822	N/A	31927	N/A	34679	N/A	35058	N/A	veh/h
Flow - Car	31701	N/A	29084	N/A	31585	N/A	31926	N/A	veh/h
Flow - Truck	3121	N/A	2843	N/A	3094	N/A	3132	N/A	veh/h
Input Flow - All	40686	N/A	39663	N/A	40801	N/A	40786	N/A	veh/h
Input Flow - Car	36774	N/A	35835	N/A	36885	N/A	36856	N/A	veh/h
Input Flow - Truck	3912	N/A	3828	N/A	3916	N/A	3930	N/A	veh/h
Number of Lane Changes - All	986,89	N/A	1471,51	N/A	994,88	N/A	1000,47	N/A	LC/km
Number of Lane Changes - Car	891,46	N/A	1347,24	N/A	898,21	N/A	904,35	N/A	LC/km
Number of Lane Changes - Truck	95,43	N/A	124,27	N/A	96,67	N/A	96,12	N/A	LC/km

Table 2. Summary of results from simulations in CAVs scenario with “lane-changing threshold” = 1,3 and different values of the “distance to detector” parameter.

		Parameters							Units	
Lane change threshold	Default lane-change model	1,3								
Maximum number of vehicles ahead	Default lane-change model	8							Veh	
Detection distance	Default lane-change model	500							m	
Free-flow rate	Default lane-change model	0,6								
Distance to detector	Default lane-change model	0		1000		1500		m		
Time series	Mean value	$\sigma$	Mean value	$\sigma$	Mean value	$\sigma$	Mean value	$\sigma$		
Speed - All	71,64	19,52	65,77	22,05	70,96	19,29	71,6	19,28	km/h	
Speed - Car	72,38	19,73	66,21	22,4	71,63	19,53	72,32	19,49	km/h	
Speed - Truck	64,21	15,41	61,26	17,39	64,18	15,05	64,25	15,16	km/h	
Total Number of Lane Changes - All	319535	N/A	476447	N/A	322123	N/A	323935	N/A		
Total Number of Lane Changes - Car	288638	N/A	436212	N/A	290824	N/A	292813	N/A		
Total Number of Lane Changes - Truck	30897	N/A	40235	N/A	31299	N/A	31122	N/A		
Total Travel Time - All	3831,69	N/A	3917,34	N/A	3833,54	N/A	3856,21	N/A	h	
Total Travel Time - Car	3297,65	N/A	3414,42	N/A	3307,37	N/A	3323,08	N/A	h	
Total Travel Time - Truck	534,04	N/A	502,92	N/A	526,17	N/A	533,13	N/A	h	
Travel Time - All	56,22	25,59	63,69	30,13	56,19	22,49	56,07	25,11	sec/km	
Travel Time - Car	55,68	25,17	63,41	29,95	55,68	21,86	55,55	24,86	sec/km	
Travel Time - Truck	61,76	28,95	66,5	31,75	61,35	27,57	61,37	27,03	sec/km	

Table 2 (cont.). Summary from results for simulations in CAVs scenario with “lane-changing threshold” = 1,3 and different values of the “distance to detector” parameter.

		Parameters								Units
Lane change threshold	Default lane-change model	1,1								
Maximum number of vehicles ahead	Default lane-change model	8								Veh
Detection distance	Default lane-change model	500								m
Free-flow rate	Default lane-change model	0,6								
Distance to detector	Default lane-change model	0		1000		1500		m		
Time series	Mean value	$\sigma$	Mean value	$\sigma$	Mean value	$\sigma$	Mean value	$\sigma$		
Delay Time - All	18,94	25,42	31,24	38,75	18,66	21,8	19	25,17	sec/km	
Delay Time - Car	19,08	25,06	31,75	38,88	18,87	21,25	19,14	24,75	sec/km	
Delay Time - Truck	17,48	28,74	25,82	36,98	16,49	26,66	17,54	29,09	sec/km	
Density - All	13,95	N/A	17,33	N/A	14,13	N/A	13,95	N/A	veh/km	
Density - Car	11,9	N/A	14,9	N/A	12,09	N/A	11,9	N/A	veh/km	
Density - Truck	2,05	N/A	2,43	N/A	2,04	N/A	2,05	N/A	veh/km	
Flow - All	34822	N/A	29814	N/A	34670	N/A	34942	N/A	veh/h	
Flow - Car	31701	N/A	27216	N/A	31564	N/A	31820	N/A	veh/h	
Flow - Truck	3121	N/A	2598	N/A	3106	N/A	3122	N/A	veh/h	
Input Flow - All	40686	N/A	38121	N/A	40809	N/A	40782	N/A	veh/h	
Input Flow - Car	36774	N/A	34365	N/A	36888	N/A	36850	N/A	veh/h	
Input Flow - Truck	3912	N/A	3756	N/A	3921	N/A	3932	N/A	veh/h	
Number of Lane Changes - All	986,89	N/A	1940,29	N/A	1005,62	N/A	995,74	N/A	LC/km	
Number of Lane Changes - Car	891,46	N/A	1801,23	N/A	906,5	N/A	900,34	N/A	LC/km	
Number of Lane Changes - Truck	95,43	N/A	139,06	N/A	99,12	N/A	95,4	N/A	LC/km	

Table 3. Summary of results from simulations in CAVs scenario with “lane-changing threshold” = 1,1 and different values of the “distance to detector” parameter.

		Parameters							Units
Lane change threshold	Default lane-change model	1,1							
Maximum number of vehicles ahead	Default lane-change model	8							Veh
Detection distance	Default lane-change model	500							m
Free-flow rate	Default lane-change model	0,6							
Distance to detector	Default lane-change model	0		1000		1500		m	
Time series	Mean value	$\sigma$	Mean value	$\sigma$	Mean value	$\sigma$	Mean value	$\sigma$	Units
Speed - All	71,64	19,52	64,39	24,14	71,14	19,21	71,47	19,43	km/h
Speed - Car	72,38	19,73	64,8	24,53	71,79	19,47	72,19	19,65	km/h
Speed - Truck	64,21	15,41	60,13	19,1	64,63	14,86	64,11	15,28	km/h
Total Number of Lane Changes - All	319535	N/A	628230	N/A	325601	N/A	322403	N/A	
Total Number of Lane Changes - Car	288638	N/A	583206	N/A	293509	N/A	291513	N/A	
Total Number of Lane Changes - Truck	30897	N/A	45024	N/A	32092	N/A	30890	N/A	
Total Travel Time - All	3831,69	N/A	3803,5	N/A	3819,19	N/A	3843,6	N/A	h
Total Travel Time - Car	3297,65	N/A	3331,5	N/A	3294,3	N/A	3313,73	N/A	h
Total Travel Time - Truck	534,04	N/A	472,01	N/A	524,89	N/A	529,87	N/A	h
Travel Time - All	56,22	25,59	68,57	38,9	55,96	21,99	56,28	25,36	sec/km
Travel Time - Car	55,68	25,17	68,41	39,08	55,48	21,39	55,74	24,87	sec/km
Travel Time - Truck	61,76	28,95	70,15	37	60,77	26,87	61,81	29,31	sec/km

Table 3 (cont.). Summary of results from simulations in CAVs scenario with “lane-changing threshold” = 1,1 and different values of the “distance to detector” parameter.

## 5. Conclusions

In general, checking the situation downstream of a vehicle current position instead of directly ahead for lane-change decision increases the overall performance of the system. The model seems to reduce the risk of traffic breakdown, as vehicles carrying out the lane-changes in advance make possible to achieve a better distribution of the traffic across all the lanes of the road, improving traffic conditions such as flow, density, average speed and reduced delays.

However, these improvements are much lower if the results are compared to those obtained from a simulation applying the default AIMSUN lane-changing model. Regarding delay time, for example, the best performing case using the presented lane-changing model is the one with a “distance to detector” = 1000m and a 1,1 “lane change threshold”, obtaining an average delay time of 18,66 s/km. Applying the internal model, the average delay time is 18,94 s/km. The reduction is just around 1,47%.

Therefore, it has been proved that, for the presented lane-changing model, the best results are obtained when vehicles evaluate the traffic situation considering information that is not available for human-driven not connected vehicles, like position and speed of vehicles traveling downstream on the road, out of the field of vision. But it has not been proved that this model, even assuming a 100% penetration rate of connected vehicles and a full compliance rate from drivers to follow given advices, could offer a better performance compared to the AIMSUN default model, after simulations in a Connected Automated Vehicles scenario.

The model would need some adjustments and an optimal calibration of its parameters in order to increase its performance. It should be simulated in different scenarios, like mixed traffic with human-driven vehicles and connected vehicles at different penetration rates. Compliance of driver should be considered as well, to reach realistic results.

Other ideas to expand this work in the future could be to increase the complexity of the lane-changing model, including mandatory lane-changing situations, adjustments on the gap search and acceptance process, etc. Development of a car-following model to be integrated with it could be interesting as well.

## **6. Acknowledgments**

I would like to thank Claudio Roncoli and the people in the Department of Built Environment of the Aalto University, for giving me the opportunity to work with them and write my Master's Final Project. They helped me with everything I needed and provided all the tools to do my work in the best conditions.

I would like also to thank Aalto University and the Universitat Politècnica de Catalunya for this chance to complete my studies abroad. It has been an incredible experience to study and live for the last 5 months in Finland, and for the last 2 years in Barcelona.

And last but not least, thanks to my family and friends for supporting and encouraging me all this time.

## References

- Awal, T., Murshed, M. & Ali, M. (2015) An efficient cooperative lane-changing algorithm for sensor and communication-enabled automated vehicles. *2015 IEEE Intelligent Vehicles Symposium (IV)*. COEX, Seoul, Korea. June 28 – July 1, 2015. p. 1328-1333. ISBN 978-1-4673-7266-4.
- Brackstone, M., McDonald, M. & Wu, J. (1998) Lane changing on the motorway: factors affecting its occurrence, and their implications. *Road Transport Information and Control*. 21-23 April 1998. p. 160-164.
- Chandler, R. E., Herman, R. & Montroll E. W. (1958) Traffic dynamics: Studies in car following. p. 165-184.
- Gipps, P. G. (1981) A behavioural car following model for computer simulation. *Transportation research Part B*. Vol. 15B. p. 105-111.
- Gipps, P.G. (1986) A model for the structure of lane-changing decisions. *Transportation research Part B*. Vol. 20B:5. p. 403-414. ISSN 0191-2615.
- Gipps, P.G. (1986) MULTISIM: A model for simulating vehicular traffic on multi-lane arterial roads. *Mathematics and Computers in Simulation*. Vol. 28. p. 291-295. ISSN 0191-2615.
- Kesting, A., Treiber, M. & Helbing, D. (2007) General lane-changing model MOBIL for car-following models. *Transportation Research Record: Journal of the Transportation Research Board*. No. 1999. p. 86-94. DOI 10.3141/1999.10.
- Kesting, A., Treiber, M. & Schönhof, M. (2007) Extending adaptive cruise control to adaptive driving strategies. *Transportation Research Record: Journal of the Transportation Research Board*. No. 2000. p. 16-24. DOI 10.3141/2000-03.
- Lytrivis, P., Thomaidis, G., Tsogas, M. & Amditis, A. (2011) An advanced cooperative path prediction algorithm for safety applications in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*. Vol 12:3. p. 669-679. DOI 10.1109/TITS.2011.2123096.
- Makridis M., Mattas K., Ciuffo B., Raposo M.A., Thiel C. (2018) Assessing the Impact of Connected and Automated Vehicles. A Freeway Scenario. In: Zachäus C., Müller B., Meyer G. (eds) *Advanced Microsystems for Automotive Applications 2017. Lecture Notes in Mobility*. Springer, Cham. p. 213-225. ISBN 978-3-319-66971-7.
- Newell, G. F. (2002) A simplified car-following theory: a lower order model. *Transportation research Part B*. Vol. 36. p. 165-205. ISSN 0191-2615
- Papadopoulou, S., Roncoli, C., Bekiaris-Liberis, N.; Papamichail, I. & Papageorgiou, M. (2018) Microscopic simulation-based validation of a per-lane traffic state estimation scheme for highways with connected vehicles. *Transportation Research Part C*. Vol. 96. p. 441-452. ISSN 0968-090X.



Pipes, L.A. (1953) An operational analysis of traffic dynamics. *Journal of Applied Physics*. Vol. 24:3. p. 274-281.

Pueboobpaphan, R. Liu, F. & van Arem, B. (2010) The impacts of a communication based merging assistant on traffic flows of manual and equipped vehicles at on-ramp using traffic flow simulation. *2010 13th International IEEE Annual Conference on Intelligent Transportation Systems*. Madeira, Portugal. September 19-22, 2010. p. 1468-1473. ISBN 978-1-4244-7659-6.

Risto, M. & Martens, M.H. (2014) Assessing driver's ability to estimate compliance rates to in-car advisory driver support. *European Research Rev.* No. 6 p. 287-294. DOI 10.1007/s12544-013-0129-6.

Roncoli, C., Papageorgiou, M. & Papamichail, I. (2015) Traffic flow optimisation in presence of vehicle automation and communication systems – Part I: A first-order multi-lane model for motorway traffic. *Transportation Research Part C*. Vol. 57. p. 241-259. ISSN 0968-090X.

Roncoli, C., Bekiaris-Liberis, N. & Papageorgiou, M. (2016) Use of speed measurements for highway traffic state estimation. *Transportation Research Record: Journal of the Transportation Research Board*. No. 2559. p. 90-100. DOI 10.3141/2559-11.

Roncoli, C. Bekiaris-Liberis, N. & Papageorgiou, M. (2017) Lane-changing Feedback control for efficient lane assignment at motorway bottlenecks. *Transportation Research Record: Journal of the Transportation Research Board*. No. 2625. p. 20-31. DOI 10.3141/2625-03.

Schakel, W.J., Knopp, V.L. & van Arem, B. (2012) Integrated lane change model with relaxation and synchronization. *Transportation Research Record: Journal of the Transportation Research Board*. No. 2316. p. 47-57. DOI 10.3141/2316-06.

Schakel, W.J. & van Arem, B. (2014) Improving traffic flow efficiency by in-car advice on lane, speed and headway. *IEEE Transactions on Intelligent Transportation Systems*. DOI 10.1109/TITS.2014.2303577.

Schakel, W.J. & van Arem, B. (2015) In-car tactical advice using delayed detector data. *2015 IEEE Intelligent Vehicles Symposium (IV)*. COEX, Seoul, Korea. June 28 – July 1, 2015. ISBN 978-1-4673-7266-4.

Tideman, M., van der Voort, M.C., van Arem, B. & Tillema, F. (2007) A review of lateral driver support systems. *Intelligent Transportation Systems Conference*. Seattle, WA, USA, Sept. 30 – Oct. 3, 2007. p. 992-999. ISBN 1-4244-1396-6.

van Driel, C.J.G. & van Arem, B. (2004) Integration of driver support functions: the driver's point of view. *International Conference on Systems, Man and Cybernetics*. 2004. IEEE. p. 3998-3994. ISBN 0-7803-8566-7.

- van Driel, C.J.G., Hoedemaeker, M., van Arem, B. (2007) Impacts of a congestion assistant on driving behaviour and acceptance using a driving simulator. *Transportation Research Part F*. Vol. 10. p. 139-152. ISSN 1369-8478.
- van Driel, C.J.G. & van Arem, B. (2008) Traffic flow impacts of a congestion assistant. *TRB 2008 Annual Meeting*. 2008.
- van Gent, P., Farah, H., van Nes, N. & van Arem, B. (2017) The use of persuasive in-car technology to persuade drivers at the tactical level. *Proceedings of the Road safety and Simulation*. The Hague, Netherlands. 17-19 October 2017.
- Wang, C. & Coifman, B. (2008) The effects of Lane-Change Maneuvers on a Simplified Car-Following Theory. *IEEE Transactions on Intelligent Transportation Systems*. Vol. 9:3. p. 523-535. ISSN 1524-9050.
- Wang, M., Daamen, W., Hoogendoorn, S.P. & van Arem, B. (2016) Cooperative car-following control: Distributed algorithm and impact on moving jam features. *IEEE Transactions on Intelligent Transportation Systems*. Vol. 17:5. p. 1459-1471. DOI 10.1109/TITS.2015.2505674.
- Zheng, Z. (2014) Recent developments and research needs in modelling lane changing. *Transportation Research Part B*. Vol. 60. p. 16-32. ISSN 0191-2615.



## Appendix 1

Lane-changing model code in C++. Developed with Aimsun microSDK.

```
bool behavioralModelParticular::evaluateLaneChanging(A2SimVehicle *vehicle, int  
threadId)
```

```
{
```

```
    //Parameters
```

```
    double distance2Detector = 1500;
```

```
    int maxNbVehiclesAhead = 8;
```

```
    double maxDistanceAhead = 500;
```

```
    double changeThreshold = 1.3;
```

```
    double timeLimit = 15;
```

```
    double fluentSpeed = 60;
```

```
    double freeFlowRate = 0.6;
```

```
    //Vehicle ID
```

```
    int vehicleID = vehicle->getId();
```

```
    //On ramps exceptions
```

```
    bool isOnRamp = vehicle->IsLaneOnRamp(0);
```

```
    bool leftOnRamp = vehicle->IsLaneOnRamp(-1);
```

```
    bool rightOnRamp = vehicle->IsLaneOnRamp(1);
```

```
    if (isOnRamp || leftOnRamp || rightOnRamp) return false;//to deactivate it
```

```
    else {
```

```
        //Selection of the detector
```

```
        A2SimVehicle* detector = NULL;
```

```
        if (distance2Detector = 0) detector = vehicle;
```

```
        else {
```

```
            double shiftDetector = 0;
```

```
            detector = vehicle->getRealLeader(shiftDetector);
```

```
            if (detector == NULL) return false;
```

```
            double posVehicle = vehicle->getPosition(0);
```

```
            double posDetector= detector->getPosition(0);
```

```
            double distBetweenVeh = posDetector + shiftDetector - posVehicle;
```

```
            A2SimVehicle*newDetector = NULL;
```

```
            while (distBetweenVeh < distance2Detector) {
```

```
                newDetector = detector->getRealLeader(shiftDetector);
```

```
                if (newDetector == NULL) return false;
```

```
                double posNewDetector = newDetector->getPosition(0);
```

```

        distBetweenVeh = distBetweenVeh + posNewDetector +
shiftDetector - posDetector;

        detector = newDetector;
        posDetector = posNewDetector;
    }
}

int detectorID = detector->getId();
double posDetector = detector->getPosition(0);

//Current lane measures
double shiftAhead = 0;
A2SimVehicle*ahead = NULL;
ahead = detector->getRealLeader(shiftAhead);
if (ahead == NULL) return false;

double posAhead = ahead->getPosition(0);
double speedAhead = ahead->getSpeed(0);
double lengthAhead = ahead->getLength();
double headway = posAhead + shiftAhead - posDetector - lengthAhead;
double timeHeadway = headway / speedAhead;

double distanceAhead = posAhead + shiftAhead - posDetector;
double sumSpeed = speedAhead;
double sumHeadway = headway;
double sumTimeHeadway = timeHeadway;
int nbVeh = 1;

A2SimVehicle*newAhead = NULL;
while (distanceAhead < maxDistanceAhead && nbVeh <
maxNbVehiclesAhead) {
    newAhead = ahead -> getRealLeader(shiftAhead);
    if (newAhead == NULL) return false;

    double posNewAhead = newAhead->getPosition(0);
    distanceAhead = distanceAhead + posNewAhead + shiftAhead -
posAhead;

    double speedAhead = newAhead->getSpeed(0);
    sumSpeed = sumSpeed + speedAhead;

    double lengthNewAhead = newAhead->getLength();
    double headway = posNewAhead + shiftAhead - posAhead -
lengthNewAhead;
    sumHeadway = sumHeadway + headway;

    double timeHeadway = headway / speedAhead;
    sumTimeHeadway = sumTimeHeadway + timeHeadway;
}

```

```

        nbVeh++;
        ahead = newAhead;
        posAhead = posNewAhead;
    }

    //Average speed and headway in current lane
    double averageSpeedCurrentDownstream = sumSpeed / nbVeh;
    double averageHeadwayCurrentDownstream = sumHeadway / nbVeh;
    double averageTimeHeadwayCurrentDownstream = sumTimeHeadway /
nbVeh;

    //Left lane measures
    double XPosLeftLane = detector->getPositionInTargetlane(posDetector, -1);
    double shiftUpLeft = 0, shiftDownLeft = 0;
    A2SimVehicle*vehUpLeft = NULL;
    A2SimVehicle*vehDownLeft = NULL;
    detector->getRealUpDown(-1, XPosLeftLane, vehUpLeft, shiftUpLeft,
vehDownLeft, shiftDownLeft);
    if (vehDownLeft == NULL) return false;

    double posDownLeft = vehDownLeft->getPosition(0);
    double speedDownLeft = vehDownLeft->getSpeed(0);
    double lengthDownLeft = vehDownLeft->getLength();
    double headwayLeft = posDownLeft + shiftDownLeft - XPosLeftLane -
lengthDownLeft;
    double timeHeadwayLeft = headwayLeft / speedDownLeft;

    double distanceDownLeft = posDownLeft + shiftDownLeft - XPosLeftLane;
    double sumSpeedLeft = speedDownLeft;
    double sumHeadwayLeft = headwayLeft;
    double sumTimeHeadwayLeft = timeHeadwayLeft;
    int nbVehLeft = 1;

    A2SimVehicle*newDownLeft = NULL;
    while (distanceDownLeft < maxDistanceAhead && nbVehLeft <
maxNbVehiclesAhead) {
        newDownLeft = vehDownLeft->getRealLeader(shiftDownLeft);
        if (newDownLeft == NULL) return false;

        double posNewDownLeft = newDownLeft->getPosition(0);
        distanceDownLeft = distanceDownLeft + posNewDownLeft +
shiftDownLeft - posDownLeft;

        speedDownLeft = newDownLeft->getSpeed(0);
        sumSpeedLeft = sumSpeedLeft + speedDownLeft;

        double lengthNewAhead = newDownLeft->getLength();

```

```

        headwayLeft = posNewDownLeft + shiftDownLeft - posDownLeft -
lengthNewAhead;
        sumHeadwayLeft = sumHeadwayLeft + headwayLeft;

        timeHeadwayLeft = headwayLeft / speedDownLeft;
        sumTimeHeadwayLeft = sumTimeHeadwayLeft + timeHeadwayLeft;

        nbVehLeft++;
        vehDownLeft = newDownLeft;
        posDownLeft = posNewDownLeft;
    }

    //Average speed and headway in left lane
    double averageSpeedLeftDownstream = sumSpeedLeft / nbVehLeft;
    double averageHeadwayLeftDownstream = sumHeadwayLeft / nbVehLeft;
    double averageTimeHeadwayLeftDownstream = sumTimeHeadwayLeft /
nbVehLeft;

    //Right lane measures
    double XPosRightLane = detector->getPositionInTargetlane(posDetector, 1);
    A2SimVehicle*vehUpRight = NULL;
    A2SimVehicle*vehDownRight = NULL;
    double shiftUpRight = 0, shiftDownRight = 0;
    detector->getRealUpDown(1, XPosRightLane, vehUpRight, shiftUpRight,
vehDownRight, shiftDownRight);
    if (vehDownRight == NULL) return false;

    double posDownRight = vehDownRight->getPosition(0);
    double speedDownRight = vehDownRight->getSpeed(0);
    double lengthDownRight = vehDownRight->getLength();
    double headwayRight = posDownRight + shiftDownRight - XPosRightLane -
lengthDownRight;
    double timeHeadwayRight = speedDownRight * headwayRight;

    double distanceDownRight = posDownRight + shiftDownRight -
XPosRightLane;
    double sumSpeedRight = speedDownRight;
    double sumHeadwayRight = headwayRight;
    double sumTimeHeadwayRight = timeHeadwayRight;
    int nbVehRight = 1;

    A2SimVehicle*newDownRight = NULL;
    while (distanceDownRight < maxDistanceAhead && nbVehRight <
maxNbVehiclesAhead) {
        newDownRight = vehDownRight->getRealLeader(shiftDownRight);
        if (newDownRight == NULL) return false;
    }

```

```

double posNewDownRight = newDownRight->getPosition(0);
    distanceDownRight = distanceDownRight + posNewDownRight +
shiftDownRight - posDownRight;

    speedDownRight = newDownRight->getSpeed(0);
    sumSpeedRight = sumSpeedRight + speedDownRight;

    double lengthNewAhead = newDownRight->getLength();
    headwayRight = posNewDownRight + shiftDownRight - posDownRight
- lengthNewAhead;
    sumHeadwayRight = sumHeadwayRight + headwayRight;

    timeHeadwayRight = headwayRight / speedDownRight;
    sumTimeHeadwayRight = sumTimeHeadwayRight +
timeHeadwayRight;

    nbVehRight++;
    vehDownRight = newDownRight;
    posDownRight = posNewDownRight;
}

//Average speed and headway in right lane
double averageSpeedRightDownstream = sumSpeedRight / nbVehRight;
double averageHeadwayRightDownstream = sumHeadwayRight / nbVehRight;
double averageTimeHeadwayRightDownstream = sumTimeHeadwayRight /
nbVehRight;

//Lane change decision
int laneChangingDirection = 0;

bool leftChangePossible = vehicle->isLaneChangingPossible(-1);
bool rightChangePossible = vehicle->isLaneChangingPossible(1);

bool leftChange = false;
bool rightChange = false;

if (averageHeadwayLeftDownstream >
changeThreshold*averageHeadwayCurrentDownstream && leftChangePossible)
{
    laneChangingDirection = -1;
    leftChange = true;
}
if (averageHeadwayRightDownstream >
changeThreshold*averageHeadwayCurrentDownstream && rightChangePossible)
{
    laneChangingDirection = 1;
    rightChange = true;
}

```



```

if (leftChange && rightChange)
    if (averageHeadwayLeftDownstream >
averageHeadwayRightDownstream)
    {
        laneChangingDirection = -1;
        rightChange = false;
    }
    else
    {
        laneChangingDirection = 1;
        leftChange = false;
    }

//Check when last lane change happened
//double lastLeft = ((simVehicleParticular*)vehicle)->getLastLeft();
//double lastRight = ((simVehicleParticular*)vehicle)->getLastRight();

//Limit lane-changes
//if (laneChangingDirection = -1 && lastLeft < timeLimit)
laneChangingDirection = 0;
//if (laneChangingDirection = 1 && lastRight < timeLimit)
laneChangingDirection = 0;

//Check if vehicles in the evaluated section downstream are traveling close to
free flow speed
double freeFlowSpeed = detector->getFreeFlowSpeed();
if (averageSpeedCurrentDownstream > freeFlowRate * freeFlowSpeed)
laneChangingDirection = 0;
//if (averageSpeedCurrentDownstream >= fluentSpeed / 3.6)
laneChangingDirection = 0;

//Lane change maneuver
if (laneChangingDirection != 0) {

    double XPosTargetlane = vehicle->getPositionInTargetlane(vehicle-
>getPosition(0), laneChangingDirection);

    //Lane Changing attempt
    double ShiftUp = 0, ShiftDw = 0;
    A2SimVehicle* pVehDw = NULL;
    A2SimVehicle *pVehUp = NULL;
    vehicle->getUpDown(laneChangingDirection, XPosTargetlane, pVehUp,
ShiftUp, pVehDw, ShiftDw);
    bool GapAcceptable = vehicle-
>isGapAcceptable(laneChangingDirection, XPosTargetlane, pVehUp, ShiftUp, pVehDw,
ShiftDw);

```

```

if (GapAcceptable) {
    vehicle->assignAcceptedGap(laneChangingDirection,
XPosTargetlane, (const simVehicleParticular*)pVehUp, ShiftUp, (const
simVehicleParticular*)pVehDw, ShiftDw, threadId);
    return true;
}

//Target New Gap
double ShiftUpReal = 0, ShiftDwReal = 0;
A2SimVehicle * pVehDwReal = NULL;
A2SimVehicle * pVehUpReal = NULL;
vehicle->getRealUpDown(laneChangingDirection, XPosTargetlane,
pVehUpReal, ShiftUpReal, pVehDwReal, ShiftDwReal);
vehicle->targetNewGap(laneChangingDirection, XPosTargetlane,
pVehUpReal, ShiftUpReal, pVehDwReal, ShiftDwReal, threadId);
if (pVehUpReal || pVehDwReal) {
    vehicle->assignNewTargetGap(XPosTargetlane, (const
simVehicleParticular*)pVehUpReal, ShiftUpReal, (const
simVehicleParticular*)pVehDwReal, ShiftDwReal, threadId);
}
}

double timeStep = AKIGetCurrentSimulationTime();
if (laneChangingDirection < 0)
    ((simVehicleParticular*)vehicle)->setLastLeft(timeStep);
else if (laneChangingDirection > 0)
    ((simVehicleParticular*)vehicle)->setLastRight(timeStep);

return true;
}
}

```



