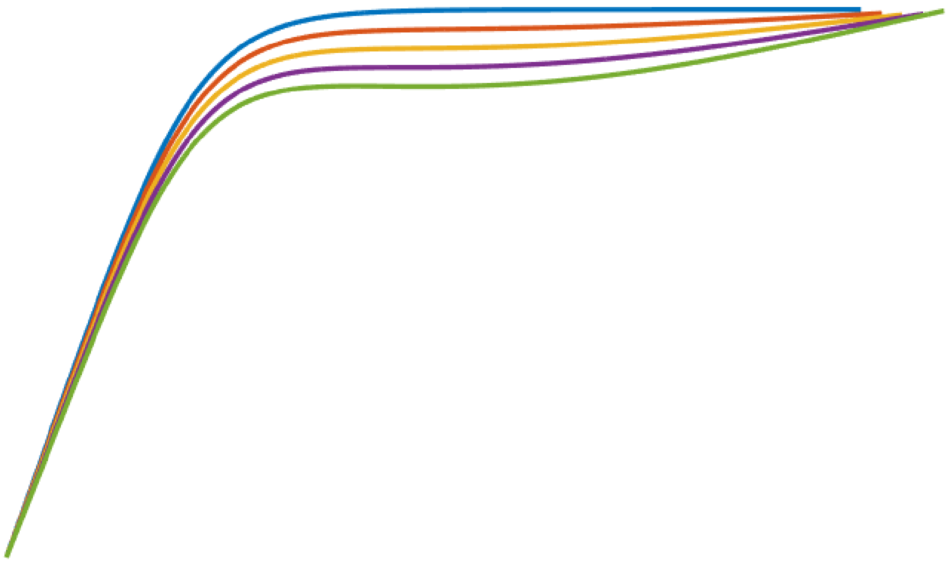

Synchronized Take-Off at Red Lights

Analyzing the impacts to traffic flow and emissions caused due to synchronized take-off of vehicles at traffic lights

Master's thesis in Sustainable Energy Systems

ADITYA RAJAN

Department of Space, Earth and Environment
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master's thesis 2018:NN

Synchronized Take-Off at Red Lights

Analyzing the impacts to traffic flow and emissions caused due to synchronized take-off of vehicles at traffic lights

ADITYA RAJAN

Department of Space, Earth and Environment
Chalmers University of Technology
Gothenburg, Sweden 2018

Synchronized Take-Off at Red Lights
ADITYA RAJAN

© ADITYA RAJAN, 2018.

Supervisor: Björn Lindenberg, Volvo Car Corporation
Supervisor: Johan Lodin, Department of Space, Earth and the environment
Examiner: Frances Sprei, Department of Space, Earth and the environment

Master's Thesis 2018:NN
Department of Space, Earth and the environment
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Drive cycles of five simulated cars that take-off synchronously

Typeset in LaTeX
Gothenburg, Sweden 2018

Synchronized Take-Off at Red Lights
Aditya Rajan
Department of Department of Space, Earth and Environment
Chalmers University of Technology

Abstract

Urban environments are growing more automobile reliant as a direct product of the affluence of societies, and externalites such as congestion, noise and CO_2 emissions are becoming more prevalent. With the advent of advanced driver assistance systems (ADAS) and vehicle to vehicle communication (V2V) and autonomy in vehicles, there are new avenues that can help relax the aforementioned impacts.

This thesis will focus on evaluating various driver models and simulating them with the objective of achieving synchronized take-off of vehicles at traffic lights. The scope will involve a specific simulation environment developed just for testing the different cases and will not fully extend to complex road networks. The simulation results will be scrutinized using a parameter study and will be used to perform a carbon emissions analysis using a complete vehicles simulation software developed by the Volvo Car Corporation.

Keywords: Drive Cycle, Driver Model, Platooning, Synchronized take-off, ACC, VSim.

Acknowledgements

I would like to extend my gratitude towards my supervisor at Chalmers, Johan Lodin and my examiner Frances Sprei. They have provided me with equal parts of advice and insight that has led to the completion of my Master Thesis.

Secondly, I would like to thank my supervisor at the Volvo Car Corporation, Björn Lindenberg and the whole Vehicle Energy Efficiency team whose kindness and guidance pushed me out of my comfort zone and helped me succeed.

Finally, I would like to thank my friends and family who have made the whole ride smooth and exciting.

Aditya Rajan, Gothenburg, August 2018

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

In recent years, the proper functioning of our society has become heavily reliant on automobiles. As such, drawbacks of an automobile-reliant society are progressively becoming more relevant.

Direct negative impacts such as congestion and CO_2 emissions cased by traffic are magnified within dense urban environments. As of 2014, the transportation sector contributed to around 20% [1] of global greenhouse gas emissions. Studies conducted on the effects of congestion with respect to emissions, concluded that due to longer trip times and inefficient engine use, there could be an increase of up to 50% in GHG emissions when compared to ideal free road conditions [3]. A significant amount of increased congestion in cities is caused due to vehicle queuing at intersections and traffic lights [2]. With an increase in driver assistance systems and overall autonomy of cars it is possible to shorten wait times at red lights and increase overall flow of vehicles.

The complexity of the road networks and consequent traffic flows make it cumbersome to measure and physically analyze different traffic scenarios. A possible solution can be found in traffic simulation. Road networks and traffic flow through them are subject to a lot of uncertainty and as such observing them proves to be a challenging task for a traffic analyst. But there has been a stark increase in data being collected by vehicles and as such traffic simulations are becoming more indicative of real world scenarios. This thesis will aim at using traffic simulation techniques to investigate the benefits of achieving synchronized take-off of vehicles at red lights.

1.2 Problem Definition

Within cities, a lot of vehicular queuing is due the presence of intersections and choke points within the road network. This queuing, if not properly managed by the road network, leads to congestion. This leads to longer trip times and slower commute, which can have a direct impact on CO_2 emissions of the vehicle fleet as well as negative externalities such as economic losses.

This queuing at traffic lights can be relaxed with more efficient take off methods for the vehicles. With the advent of Advanced Driver Assistance Systems (ADAS) and vehicle to vehicle communication, it would be possible to shorten the response time of drivers to traffic lights and achieve synchronized take-off.

The impacts of assisted take-off methods of vehicles at red lights to traffic flow (based on time saved at red light) and CO_2 emissions would be useful in evaluating these driver assistance systems and would prove insightful in infrastructure planning.

1.3 Objective

To use micro-simulation methods to evaluate the potential impacts to traffic flow and CO_2 of synchronized take off of vehicles at red lights

- Perform a comparative study based on the degree of assistance systems in the car using micro-simulation methods
- Using Volvo's complete vehicle simulation software *VSim* to calculate energy impacts.

1.4 Thesis Redefinition

The original thesis was titled "*Efficient Road Infrastructure*". It involved the use of an open source traffic simulation software *SUMO* (Simulation of Urban Mobility). The goal of the thesis was to develop a map optimization software that could fix discrepancies (such as broken roads and missing intersections) in road networks imported from OpenStreetMaps.

During the initial literature study phase and after discussions with the Autonomous Drive Fuel Economy (ADFE) team at the Volvo Car Corporation, it was discovered that due to the high degree of complexity and difference in various road networks, automating the map optimizing process would be time consuming and difficult to complete by a single thesis student. It was also discovered that updates in *SUMO's* in-built map optimizer *NetCONVERT*, would eventually fix discrepancies in imported road networks.

1.5 Limitations

Due to scope of the chosen simulation environment there are the following limitations

- Flow improvements are based on a single traffic light, and may cause externalities when introduced to a complete road network
- Accuracy of mathematical models used to represent human drivers is still a question of debate

- Effects of air drag are not considered
- Lane changing of vehicles is not considered
- System boundary of the study is restricted only to the simulation environment and does not extend to more complex road networks.
- Extended system level externalities have not been evaluated

2
Theory

In the following section, relevant literature that was used in performing the simulations will be reviewed.

2.1 Traffic simulation

Traffic simulation employs the use of mathematical models (known as driver models) and numerical methods to simulate the functioning of a traffic system or road network. It forms the basis of traffic engineering and outputs from traffic simulation can be used to analyze road networks and source potential infrastructure improvements. Traffic simulation may be done using discrete or time continuous variables whose system state are constantly tracked and updated within the system.

2.1.1 Macroscopic traffic simulation

This is a form of traffic simulation that is aimed at analyzing traffic systems to gain a system level perspective. It is used to identify traffic flow impacts like density, average fleet velocity, emissions etc. It can have a foundation in mircoscopic simulation models that have been extrapolated to form a larger system, or can be based simply on data collected coupled with projection algorithms.

2.1.2 Microscopic traffic simulation

This form of traffic simulation is based on modelling the characteristics of each individual vehicle in the simulation and how they interact with each other. A simulation environment(road network) is first created and the movement of vehicles within it is simulated by assigning specific drive conditions for each vehicle and also determined by how the vehicles directly interact with other vehicles that are in their vicinity. One relevant model used in this form of simulation is the car-following model.

2.1.2.1 Car-Following Model

Each vehicle in a simulation environment is considered as a control element, whose characteristics (vehicle speed, acceleration etc.) are determined by its interaction with the preceding control element withing the system. These characteristics are usually a function of the inter vehicle gap, i.e. the gap from the tip of the front bumper of the following car to the end of the rear bumper of the leading car. Here it is assumed that the drivers try to maintain a gap higher than the minimum allowed gap (to avoid collisions) while also trying to maximize their current speed [4].

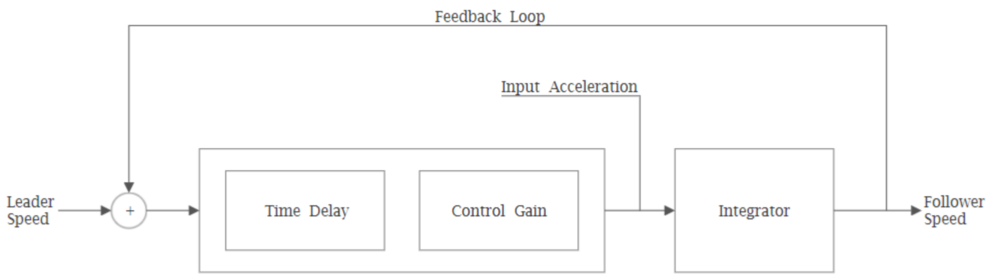

Figure 2.1: Block diagram of a standard linear car following model [4]

In figure 2.1, the time delay (also known as time headway) and control gains (depending on aggressiveness/passiveness) represent the driver, the lead vehicle speed is modified by the driver behavior and the acceleration command (acceleration/ deceleration based on current gap between the leading and following vehicles) are integrated to determine the speed of the following car.

Work on car following models has been active for over 45 years and improvements have constantly been made, and many driver models have been developed that try to mimic human drivers.

2.2 Driver Models

Accuracy of driver models (mathematical models of actual drivers), has been a hot topic of debate for many years now. The sheer scope of human driving patterns is riddled with uncertainties. The driver models were chosen based in the objective of the thesis, one driver model to represent humans (Intelligent Driver Model), one which employs Advanced Driver Assistance Systems (Adaptive Cruise Control) and finally one co-operative model with vehicle to vehicle communication (Platooning Model)

2.2.1 Human Driver Models

Finding a driver model that can adequately represent human drivers in entirety is extremely challenging due to the many random factors (driver preferences), that dominate human driving [5]. Recent developments lean towards training neural networks to drive cars, but this topic is worthy of a separate thesis in itself.

This study mainly focuses on the behaviour of drivers at traffic lights, due to this small system boundary, the question of finding a driver model that accurately represents humans, is simpler. Two driver models have been investigated and compared to arrive at a solution.

2.2.1.1 Intelligent Driver Model

The Intelligent Driver Model (IDM) is a dynamic, time continuous, crash free and single lane car following model developed by Treiber, Hennecke and Helbing in the year 2000 [6]. In this model, the velocity of a following car and consequently its updated position is calculated based on its current inter-vehicle distance and difference in velocity to the leading car. If x is the current position of the vehicles and l is the vehicle length, the current inter vehicle gap s can be written as,

$$s = x_{\alpha-1} - x_{\alpha} - l_{\alpha-1} \tag{2.1}$$

α refers to the following vehicle and α - 1 refers to the leading vehicle. The rate at which the follower is approaching the leader is an important factor in determining how much acceleration or braking is required, this can be written as,

$$\Delta v_{\alpha} = v_{\alpha} - v_{\alpha-1} \tag{2.2}$$

If the position of the following vehicle is known, then its velocity can be calculated as its first derivative with respect to time.

$$\frac{dx_{\alpha}}{dt} = \dot{x} = v_{\alpha} \tag{2.3}$$

As consequently, its movement in the next time-step will be based on its current acceleration, which is the second derivative of its position.

$$\frac{d^2x_{\alpha}}{dt^2} = \ddot{x} = a\left[1 - (\frac{v_{\alpha}}{v_0})^{\delta} - \left(\frac{s^*(v_{\alpha}, \Delta v_{\alpha})}{s_{\alpha}}^2\right)\right] \tag{2.4}$$

In the IDM model, the acceleration is calculated based on a few factors, v_0 is the desired vehicle speed (usually set to the speed limit of the road), a is the maximum desired acceleration (based on driver preference) and finally s^* which is the gap control parameter which is written as,

$$s^*(v_{\alpha}, \Delta v_{\alpha}) = (s_0 + v_{\alpha}T + \frac{v_{\alpha}\Delta v_{\alpha}}{2\sqrt{ab}}) \tag{2.5}$$

This is mainly an interaction term and is the cause for the "intelligence" of the model, by adding the maximum desired braking term b, and also a minimum allowable gap s_0 making the model crash proof [6]. The term T depicts the minimum time headway which is the minimum possible time a point on the following vehicle takes to reach a corresponding point on the leading vehicle.

2.2.2 Adaptive Cruise Control Model

Adaptive Cruise Control (also called Autonomous Cruise Control) is a form of advanced driver-assistance systems (ADAS) that controls or limits the vehicle speed and approach rate depending on the desired inter-vehicle distance that is set in the system. This is done by employing on board radar, laser sensors of by using cameras that track the distance to the front vehicle.
There are many ways of modeling the behaviour of such a system. As the name suggests, adaptive cruise control changes based on the different driving conditions. So one potential model to evaluate would be the Wiedemann river model.

2.2.2.1 Wiedemann Driver Model

The Wiedemann driver model [7] is a psycho-physical car following model that is based on naturalistic data and determines the behaviour of the following vehicle based on different driving modes and conditions based on relative speed and distance to the leading vehicle.

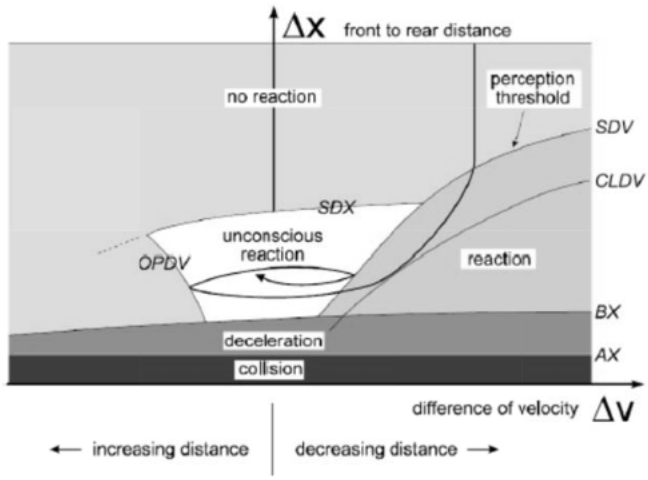

Figure 2.2: Depiction of different driving thresholds that determine driving conditions (acceleration/deceleration). ΔX is the gap between the leader and follower, and ΔV is the relative speed between them. [9]

The Wiedemann model is quite complex and can be calibrated to mimic drivers very well, but it is reliant on strong naturalistic data (actual data collected from real world trips), it is also dependant on a selected driver profile (driver preferences based on the different zones in the model) [9].

2.2.2.2 ACC Model

Another adaptive cruise control driver model, that provided optimal string stability [8] was proposed by Liang and Peng in 1999 [10]. The model was based on the two main factors between the leading and following cars, i.e. the relative velocity and current gap, but also added another term, which was the desired inter-vehicle distance that is configured into the ACC system. The acceleration of the following vehicle can be written as,

$$a_{acc} = k_v(v_{lead} - v_{follow}) - k_r(s - s_{des}) \tag{2.6}$$

Here, s refers to the current gap between the vehicles and s_{des} refers to the desired gap (the gap the system tends to) that has been configured in the ACC system. k_v is the velocity control parameter and k_r is the gap control parameter. Both the control gains must be configured empirically for specific driving modes.

2.2.3 Platooning Model

Vehicle platooning is a form of autonomous driving, where a group of vehicles are driven as one unit while they are in a platoon (spaced close together). This is usually done by vehicle to vehicle (V2V) communication. The platoon leader (first vehicles in the queue) relays its driving parameters (acceleration, speed etc.) to the following vehicles what respond by adapting their driving styles based on the inputs. This model is also known as a cooperative adaptive cruise control model.

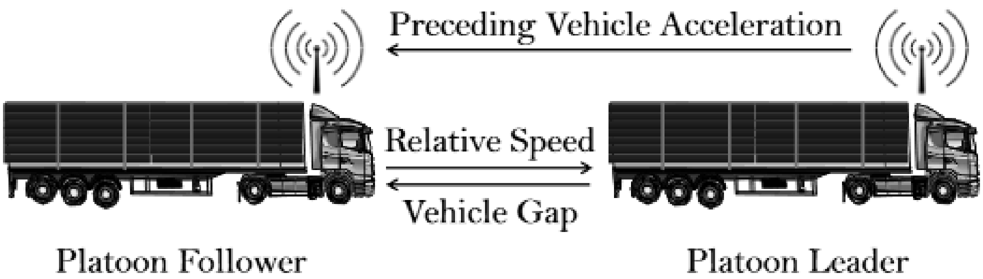

Figure 2.3: Depiction of cooperative adaptive cruise control (Platooning) [11]

This model is built similar to the the ACC model mentioned, but involves V2V communication as shown in figure 2.3. Adding this communication to equation 2.6, the acceleration of the following vehicle becomes,

$$a_{cacc} = a_{lead} + k_v(v_{lead} - v_{follow}) - k_r(s - s_{des}) \tag{2.7}$$

The relative speed and gap between the vehicles can be obtained by the use on on-board sensors, and the acceleration of the leader can be conveyed through wireless methods. The control gains can be configured empirically based on the desired output.

3
Methods

3.1 Selection of Driver Models

As mentioned in the previous section, there are various driver models that can be considered while conducting traffic studies, each with their respective pros and cons. The driver models were evaluated as follows.

3.1.1 Human Drivers

The first hurdle was to find a driver model that could adequately mimic the driving preferences of human drivers. This in itself is a cumbersome task, but there have been many studies done in the past in this domain. Two popular driving models were considered - the Gipps Model [13]and the Intelligent driver Model. They were compared qualitatively under a few specific conditions.

3.1.1.1 Braking/Deceleration

In the Gipps model, there is a constant braking term, and as such all braking maneuvers are performed at the same constant deceleration [14]. This braking strategy does not fully mimic the nuanced braking of human drivers [5].
The IDM model has an intelligent braking term, this can be seen in equation 2.5. The braking of IDM drivers is gradual and smooth during free road conditions, so there is a comfortable braking scenario, where the deceleration is lower than the maximum braking. This makes the braking of the vehicle more reactive and fluid. In addition to this, in critical braking situations, the deceleration can be higher that n the maximum possible braking configured in the model [14].

3.1.1.2 Operation and Simplicity

The Gipps model is known for its simplicity as it has few modeling parameters that must be taken into consideration. But the model relies on a safe speed v_{safe}, which determines the operation. In reality, human drivers tend to adhere more to speed limits than safe speeds, which makes this crucial parameter hard to determine.
The IDM model, is considerable more complex and involes a few more parameters, but most of there are parsimonious and as such describe specific features of driving behaviour which is beneficial during the modeling phase [14].

3.1.2 Adaptive Cruise Control

The choice of which ACC model to use in the study is based on a trade-off between complexity of the model and its applicability to the take-off scenario.

The Wiedemann model proves to be very useful as it can be applied to many driving conditions, but the shift between driving thresholds in the model is not smooth, which leads to discrete "jumps" in acceleration [14]. But, its usability is advantageous it it is used in varying driving condition opposed to just a take-off condition. Calibrating complex models is a time consuming task and as such models with fewer control parameters are more preferable [14].

The ACC model presented by Liang and Peng (1999) [10] seen in equation 2.6, is quite simple and can be calibrated and configured with relative ease. The main drawback here is that since the control parameters of the model are static, they depict only one specific driving behaviour. To perform a more robust comparison between the test cases, two different modes of this ACC model were simulated.

3.1.2.1 ACC - Cautious

In this driving mode the velocity control parameter and gap control parameter were configured as follows,

Table 3.1: Control Parameters - ACC - Cautious

$Velocity\ control\ gain\ \mathbf{k_v}$	1.70
$Gap\ control\ gain\ \mathbf{k_r}$	1.12

The optimal control gains were taken from the paper by Liang and Peng (1999) [10]. The driver in this model can be considered "cautious", and prefers to maximize the gap between vehicles (tending to the desired gap s_0) rather than maximize speed.

3.1.2.2 ACC - Take-Off

The take off condition requires the vehicles to prioritize speed, so that the get past the red light as quickly as possible. This demands that the control gains for the relative velocity be higher than the gap control gain. The values of the control parameters can be seen in table 3.2.

Table 3.2: Control Parameters - ACC - Take-off

$Velocity\ control\ gain\ \mathbf{k_v}$	2.35
$Gap\ control\ gain\ \mathbf{k_r}$	0.01

The values of k_v and k_r were configured empirically using a trial and error method based on literature, such that they offered the lowest times for the entire fleet of cars to pass the redlight.

3.1.3 Platooning - Cooperative Adaptive Cruise Control

The platooning model described in section 2.2.3 perfectly fits the trade-off between functionality and simplicity, so it was chosen as the appropriate model for modelling the behavior of vehicular platoons. The control gains for the model can be seen in table 3.3.

Table 3.3: Control Parameters - Platooning - CACC

Velocity control gain $\mathbf{k_v}$	0.3
Gap control gain $\mathbf{k_r}$	0.01

Since the size of the vehicular platoon is small (5 cars), it was decided that the time lose in wireless communication would be negligible and was hence omitted from the modeling of the system. The objective from this driver model was to achieve synchronized take-off through vehicle to vehicle communication which is why the control parameters were configured to prioritize velocity. A point to note is that the control gains required are small in magnitude as most of the acceleration term is obtained by communicating the current acceleration of the platoon leader to the followers.

3.2 Selection of Simulation tool

The initial investigation was to be done on an open source traffic simulation software tool called *SUMO* (Simulation of Urban Mobility). Sumo is a complete microscopic simulation software package that is capable of simulating complex road networks and also has an in built emissions model [12].

Since the thesis objective was specific to only the take-off case, it was determined the functionality of *SUMO* would be beyond the scope of the investigation and would increase the complexity of the study.

A decision was made to develop a case specific simulation environment from scratch using MATLAB, which could be customized for the study.

3.3 Drive Cycles

Drive cycles or speed traces are critical outputs of microscopic traffic simulation studies. They are graphical plots that show the change in velocity of vehicles over each simulation time-step. In this thesis, drive cycles for different driver models will be generated based on a few scenarios. The drive cycles will then be analyzed and compared.

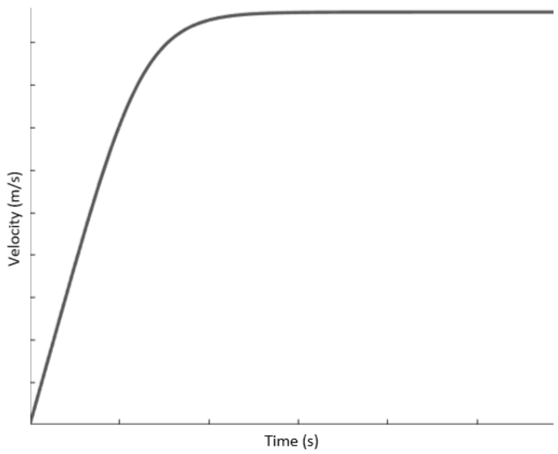

Figure 3.1: A basic drive cycle of an automobile starting at rest

Since the drive cycles depict how the car is driven (when gear changes happen, acceleration at each point etc.) under the given time-frame, most analysis methods (fuel consumption, CO_2 emissions etc.) use drive cycles as inputs to perform calculations.

3.4 Simulation Environment

A visual depiction of the simulation environment created in MATLAB can be seen in figure 3.2.

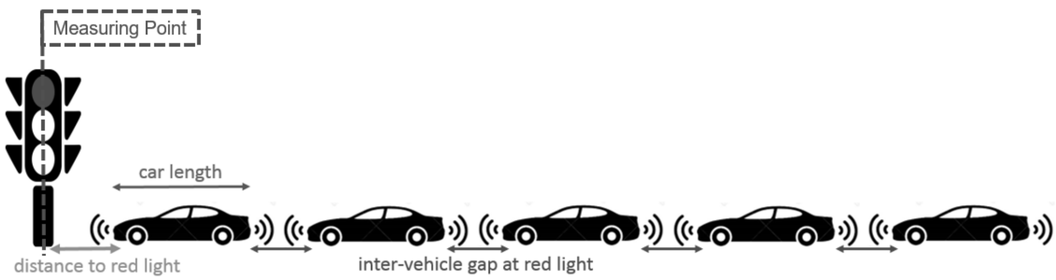

Figure 3.2: Pictorial representation of the start positions of each car in the test environment

For all test cases, the same start conditions were used. Five identical cars placed in a vehicular queue at equal distance from each other on a single lane road with a speed limit of 70 kilometers per hour. A test group of five cars was chosen to simulate an inner city traffic light. The remaining specifics of the simulation environment can be found in table 3.4.

Table 3.4: Simulation Environment Parameters

Road Type	Single Lane
Road Length	1km
Distance of first car to red light	2m
Length of cars	4m

All the different driver models were tested in this environment, and the corresponding drive cycles were obtained.

3.5 Test Cases

3.5.1 Case 1: Reference Case

The reference case was selected to represent a standard take-off case with similar vehicles at red lights, as it is today. All the drivers are represented by the IDM driver model with static parameters that have been configured to sufficiently mimic a real-world scenario [5]. The specifics and validation of each of the parameters is not of paramount importance as the investigation being conducted is a comparative study ans as such still has valuable merit.

All parameters considereded for the reference case can be seen in table 3.5.

Table 3.5: Reference Case Parameters (IDM)

Maximum Acceleration **a**	2m/s^2
Maximum braking **b**	3m/s^2
Minimum allowed gap $\mathbf{s_0}$	2m
Minimum time headway **T**	1.5sec
Inter vehicle gap at red light $\mathbf{s_{redlight}}$	4m

3.5.2 Case 2: Adaptive Cruise Control

The second case will be to evaluate the adaptive cruise control model. Here, the vehicular queue consists of 5 cars of which the first car will have an entirely unaided human driver (IDM model) followed by 4 cars that are all equipped with on-board sensors and driver assistance systems. The sensors will track the relative speed of the car and also calculate the current distance to the vehicle in the front.

The tests will be conducted under two separate cases that have been described in section 3.1.2. The two cases were selected to show the two extreme possibilities that can be achieved with this type of ADAS system. One additional parameter other than the ones mentioned for the reference case that is common for these cases is the desired gap that is configured into the system $\mathbf{s_{des}}$ = 10m, which would be considered to be a comfortable gap to have between cars.

3.5.3 Case 3: Platooning/CACC

Working off the previous case, wireless communication will be added to the cars, so that the platoon leader can communicate its communication to the rest of the cars in the platoon. The synchronized take-off case will be characterized by an IDM led platoon that all follow the CACC model described in section 2.2.3. The desired gap connfigured for the CACC in all investigations will be fixed at $\mathbf{s_{des}}$ = 10m.

3.6 Flow Improvements

Traffic flow is defined as the number of vehicles recorded passing through a point for a given observation time. If we look at intersections, improvements to flow are usually indicated by having higher traffic flow through the intersection, measured at a point after the intersection (indicating more vehicles passed through the point).

Here, we look at flow improvements through the lens of time saved at the red lights. Which, is the amount of time it takes for all the cars in the simulation to pass the red light. The time it takes for a set of cars to pass the red light (t_{test}) is compared with the time it takes for the cars to pass the red light in the reference case (t_{ref}), and hence the time saved for that test case is found.

$$(time\ saved)_{test} = t_{ref} - t_{test} \tag{3.1}$$

For positive values of time saved, there are improvements to the traffic flow and for negative values, the flow is worsened. The three test cases (ACC-Cautious, ACC Take-Off and Platooning) are simulated and their respective values of time saved are calculated.

As each of the driver models being used have many parameters that influence their performance, the effect of changes in these parameters must be verified. Using the time saved concept, a parameter study is conducted on the various parameters of each model to analyze their behaviour and importance to the objective of the thesis.

3.7 Environmental Impacts

There may be direct impacts from the improvements of flow that can result in lower trip times, there can also be more efficient driving based on how much braking is omitted, and hence there can be relevant impacts to the environment cased by the addition of ADAS systems to vehicles. The environmental impacts are calculated based on the summation of CO_2 emissions of all the cars for each test case.

3.7.1 Comparability of test cases

The first set to find the environmental impacts was to make each test case comparable so that an absolute value of CO_2 emissions can be obtained.

Initially, the tests cases were compared for a set amount of simulation time. that is, all the cases were simulated for 30 seconds and the drive cycles for each car were obtained. Then using the drive cycle, a standard car model was selected to represent the cars (will be elaborated upon more in the next section) and the fuel consumption and consequently the CO_2 emissions were calculated. The problem here was that at the end of the simulation the vehicles had all travelled different distances and ended at different speeds. This made them incomparable as all drivers would keep accelerating till they reach top speed, so the fuel consumption for those vehicles after the 30 second simulation time would still be high.

Hence a decision was made to compare all vehicles once they had reached a specific distance and were all at the same speed (with a minimal tolerance of $\pm 0.2ms^{-1}$). In this way, all models would be at a evenly comparable point and would also give absolute results.

3.7.2 VSim

VSim is a simulation tool that is used by Volvo that has been developed in-house for over two decades. VSim is a capable of complete vehicle simulations and is mainly used to conduct energy flow simulations and to compute various performance metrics of the automobile (fuel consumption, CO_2 emissions etc.). A representation of energy flows in VSim can be seen in figure 3.3 [15].

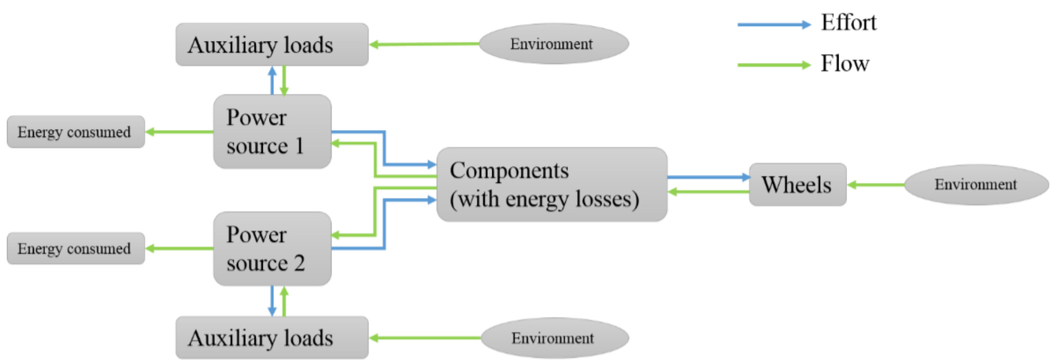

Figure 3.3: Depiction of the energy flows in VSim [15]

VSim accepts drive cycles as inputs and runs simulations based on a selected car model. Each car model has specific and tuned simulation blocks representing different components and systems (powertrain, wheels, engine etc.). Once a car model is selected and a drive cycle loaded, VSim performs the simulation and outputs energy consumed by each block and total energy used, using these vales, analyses are done to calculate the fuel consumption and CO_2 (among other performance metrics).

3.7.3 Optimal Acceleration

Finding an optimal acceleration for the tests is a difficult task as acceleration preferences differ from driver to driver. One solution was to analyze EuroFOT data which is a naturalistic data gathering project that lasted from May 2008 to June 2012, that gathered data form numerous cars equipped with ADAS systems in the European Union. The data analyzed was extremely varied and statistical methods applied to them yielded suitable results.

But, as this is a comparative study with all cases being compared to a reference case, it was more interesting to evaluate the acceleration at which CO_2 emissions for the reference case would be the lowest. This gives a strong base to make further judgment on the models, as the maximum possible reductions in CO_2 emissions can now be estimated.

Following this, emission calculations were made for all cases at the "optimal" acceleration.

4

Results

4.1 Simulation Results

The simulation environment was created in MATLAB and the different cases were simulated as per methodology found in chapter 3. All code used can be found in appendix **??**. The first outputs of the simulation were the drive cycles for all cars in each case.

4.1.1 Drive Cycles

4.1.1.1 Case 1: Reference Case

As stated in section 3.1.1, the first case simulated was the "human case" or rather the "real world" scenario. It consists of a queue of vehicles that are all equipped with the IDM driver model. It is modelled such that the first vehicle is following an "invisible" leader that is way ahead of it travelling at its speed. This was done to simplify the modelling aspect of the take-off condition. The resultant graph can be seen in figure 4.1.

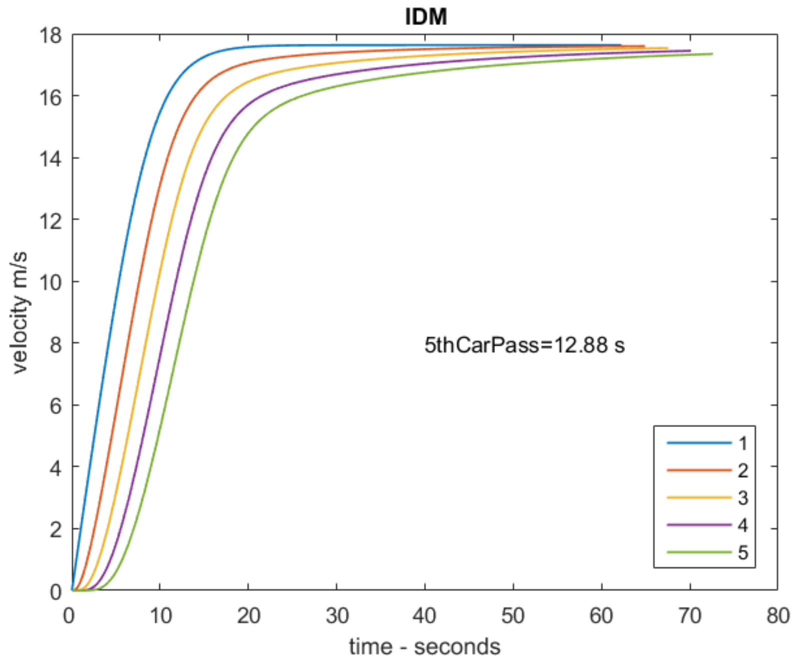

Figure 4.1: Drive cycles for 5 cars - Case 1: Reference

Here, it can be seen that the first car takes off as soon as the light turns green and the following car starts with a short time delay. This is mainly due to the minimum allowed gap (s_0) term. As the cars start at a distance of 2m away from each other and s_0=2m, the following car needs to wait a short duration before it is safe to proceed forward. As described, it can be seen that the reference case mimics human behaviour to an acceptable level.

The IDM model does not have a "desired gap" it is trying to reach, rather the gap between cars is a function of the relative velocity between followers and leaders and also the approach rate (Δv). Due to this, the cars spread out linearly as their velocity increases. This is done to ensure safety at high speeds.

4.1.1.2 Case 2: ACC

After the reference case was simulated with completely unassisted cars, the first level of ADAS systems was added to the follower cars. Sensors were added to the cars and the signal delay was considered to be negligible. As mentioned in section 3.1.2, two separate cases were modelled to depict two extreme conditions of operation of the ACC system.

Case 2.1: ACC Cautious

The first ACC test case would be the "cautious" model. It has been named as such mainly due to the way its control parameters have been configured to have high control gains for the gap control factor k_r. There is a configured desired gap that the model tries to achieve, s_{des}=10m. The resultant drive cycles can be seen in figure 4.2.

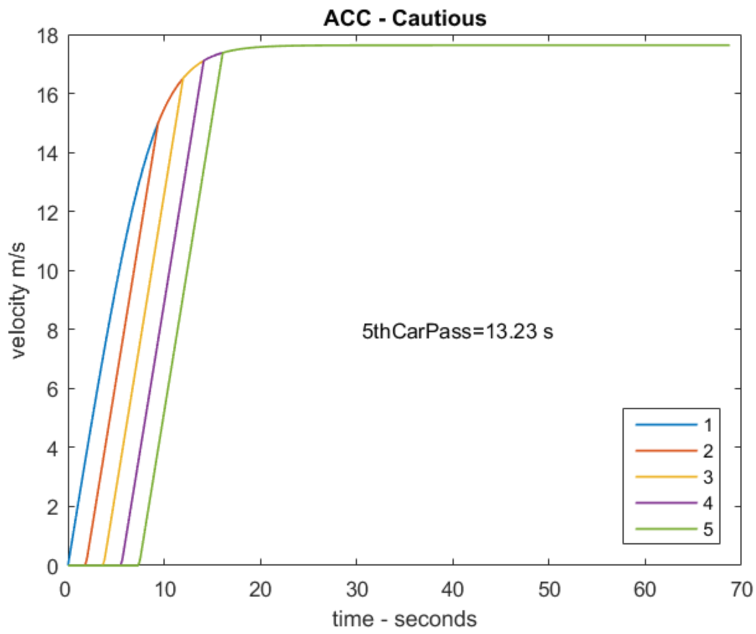

Figure 4.2: Drive cycles for 5 cars - Case 2.1: ACC Cautious

The behaviour of the followers here is characterized by safety, it can be seen that the followers do not take-off until the leader is a significant distance ahead of them. The system tries to maximize the gap to the leader and the closer it gets to the desired gap (s_{des}), the more the velocity factor kicks in that causes an increase in acceleration. So basically, the cars just wait till the leader is close to 10m away and then choose to take off from rst.

Case 2.2: ACC Take-Off

The second ACC test case has been dubbed the "take-off" case due to the system being tweaked to minimize the time it takes to pass the red light. The control parameters can be found in 3.1.2.2.

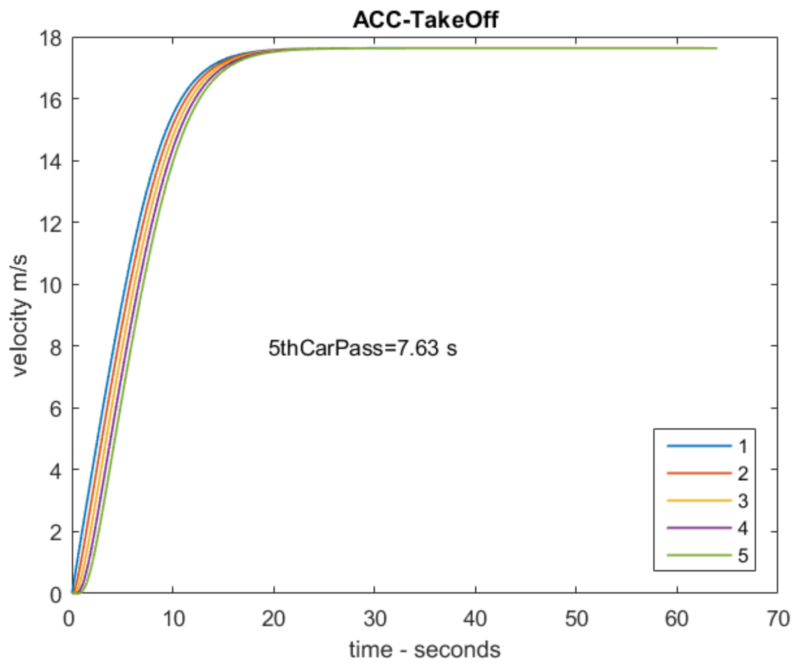

Figure 4.3: Drive cycles for 5 cars - Case 2.2: ACC Take-Off

Figure 4.3, shows the take-off case, and it can be seen that the take off case shares similarities with the reference case (figure 4.1), but the ACC test case are tweaked so they have much shorter time gaps before they take off. All vehicles eventually reach the desired gap (s_{des}), but initially take off at much lower inter-vehicle distances. This case shows that it is possible to minimize the time to pass red lights significantly by just using on-board sensors in the vehicles.

4.1.1.3 Case 3: Platooning/CACC

The final test case adds yet another aspect to the existing ACC by enabling wireless V2V (vehicle to vehicle) communication. With this added layer of communication, the acceleration of the leader is conveyed to the followers, using this, it is possible to tune the system to perform synchronized take-off. This is seen in figure 4.4.

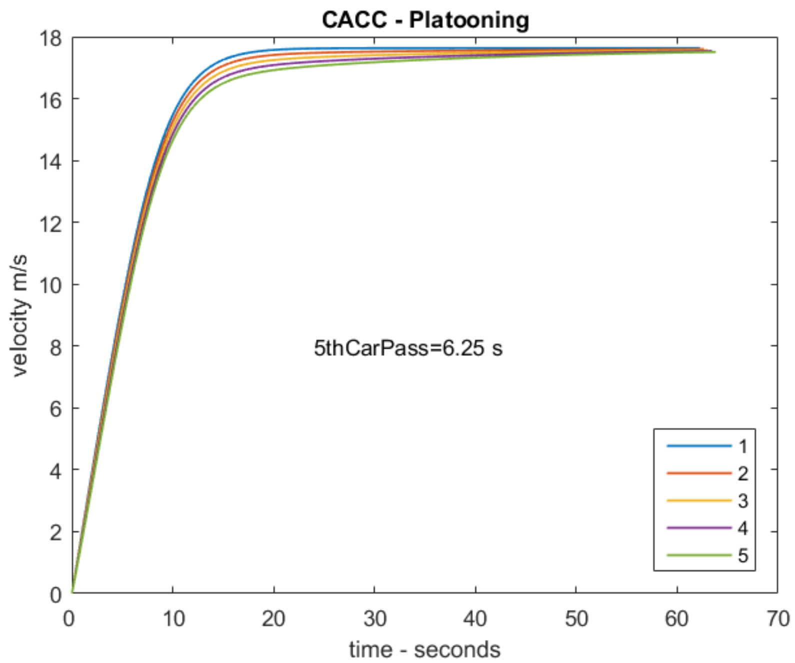

Figure 4.4: Drive cycles for 5 cars - Case 3: Synchronized Take Off

As soon as the light turns green (simulation start), all cars accelerate almost at the same rate (acceleration of the leader) and hence satisfy requirements for synchronized take-off. During the initial take off phase, velocity is prioritized and as the speed of the cars get closer to the desired speed (speed limit of the road), the weight of the gap control parameter increases and as such the cars start to decelerate to increase the gap between them to the desired gap of the system (s_{des}). This bulk of this deceleration occurs after the last car has passed the red light, so it doesn't affect the effectiveness of the model in terms of traffic flow (measured at the red light).

4.1.2 Time Saved

The first implication of the resultant drive cycles of the vehicles is the improvements to flow. The time it takes for the last car to pas the red light is recorded and compared to the reference case. Th time saved is calculated as per equation 3.1. The time saved for each case can be seen in table 4.1. The basic parameters used for the simulation can be found in table 3.5.

Table 4.1: Time saved at red light

	$Time\ for\ 5^{th}\ car\ to\ pass\ (s)$	$Time\ Saved\ (s)$
$Case\ 1:\ \ Reference\ Case$	12.88	
$Case\ 2.1:\ \ ACC\ Cautious$	13.23	-0.35
$Case\ 2.2:\ \ ACC\ Take-Off$	7.63	5.25
$Case\ 3:\ \ Platooning/CACC$	6.25	6.63

The results for time saved, reveal a few insights into the nature of the problem. Firstly, a point to be noted is that the ACC cautious case is worse than the reference case but just but a very small margin, this is mainly due to the high rates of acceleration that the cars can reach as there is a large gap between them. This is the reason the case can compete with the reference case even though the cars wait at rest for longer.

Both the ACC take-off case and the platooning case perform well in comparison to the reference, saving 5-6 seconds at red-light could potentially lead to many positive externalities in the whole traffic network. Here it can be seen that one of the dominating factors for achieving this flow improvement is the rate at which the cars accelerate and the configuration of the system.

4.2 Parameter Study

It is evident that the results of this thesis are heavily reliant on the driver models chosen and the selection of parameter values. As there are many parameters that contribute to changes in results, the only logical option was to perform a parameter study on a few important parameters to see how they affect the results. Time saved was used as a metric and was calculated for changes in parameters to get a better understanding of the models and test cases.

4.2.1 Maximum Acceleration: a

The first parameter that was observed is the maximum acceleration (driver acceleration preference). The results are seen in figure 4.5.

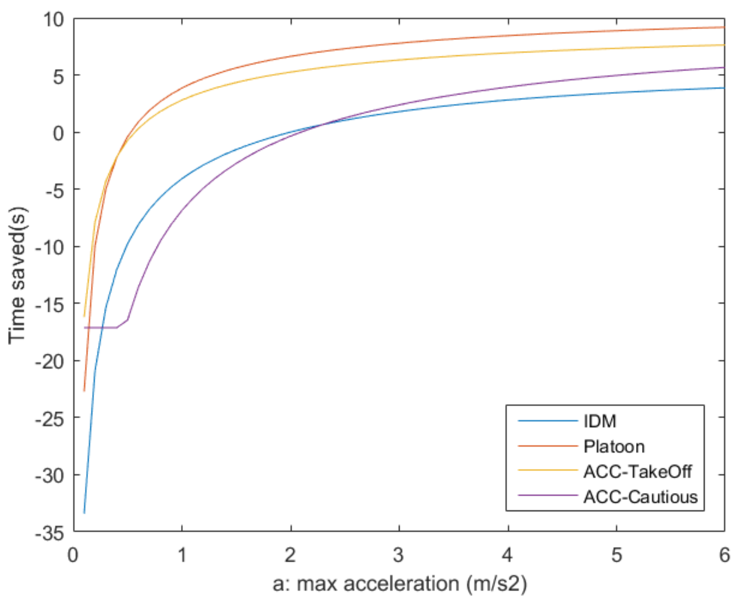

Figure 4.5: Parameter Study - Max acceleration

Observing figure 4.5, it is seen that the acceleration preferences have a huge impact on the the time saved. All models are positively affected by this parameter, but start to plateau at a certain point, indicating an upper limit. At higher acceleration all models perform better than the pure IDM model (human drivers).

it should be noted that even though this term is important, finding a single acceleration that is indicative of human driving as a whole is a very cumbersome task, and hence the comparative study has been performed.

4.2.2 Minimum Gap (s_0) and Desired Gap (s_{des})

Since The nature of a car following model is based on interaction between the cars, the gap parameter plays a very important role. As such the gap parameters that are configured into each driver model i.e. the minimum allowable gap (s_0) for the IDM and the desired gap (s_{des}) for the ACC and CACC models are crucial to the take-off scenario.

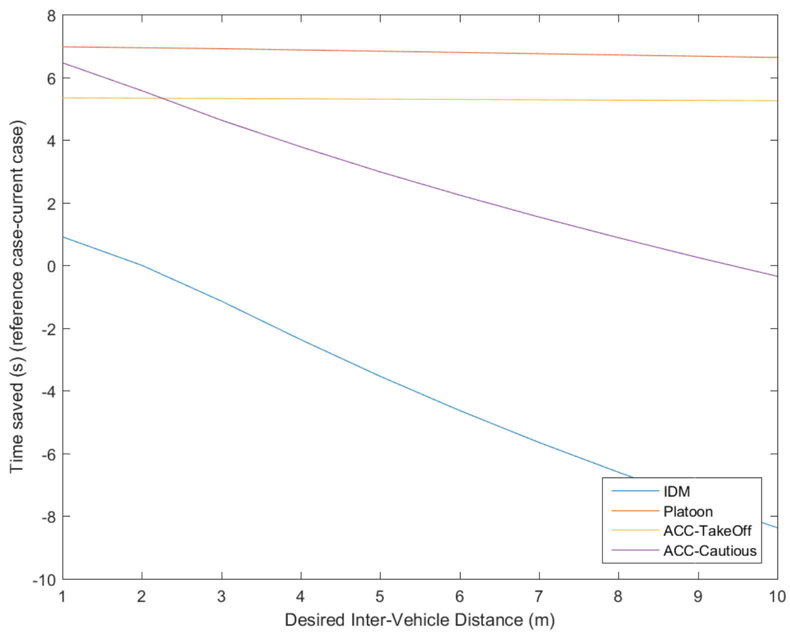

Figure 4.6: Parameter Study - Gap parameters - minimum allowable gap (s_0) for IDM and desired gap (s_{des}) for ACC/Platooning

From figure 4.6, at first glance, it is clear that the ACC Take-Off and platooning models are hardly affected by this parameter. This is due to the models giving priority to speed gains initially and then slowly spreding out to meet the desired gap the closer they get to their desired speed, and since this study is based on time saved (measuring point is the red light), this parameter does not play an important role.

Conversely, the ACC Cautious model is affected negatively by a change in this parameter, as the model will only start moving if the gap to the leader is close to s_{des}. The IDM model s_0 term is the minimum allowed gap and as such the smaller it is, the faster the follower can begin accelerating and as such is negatively affected by an increase this parameter.

4.2.3 Gap at Red Light ($s_{redlight}$)

Usually, human drivers form vehicular queues with short inter-vehicle distances at red-lights, with the hopes that they will be able to get past the traffic light faster if they are closer to it. This has been somewhat debunked in recent analytic studies performed in USA [16]. This was interesting from a flow perspective and hence this parameter was chosen.

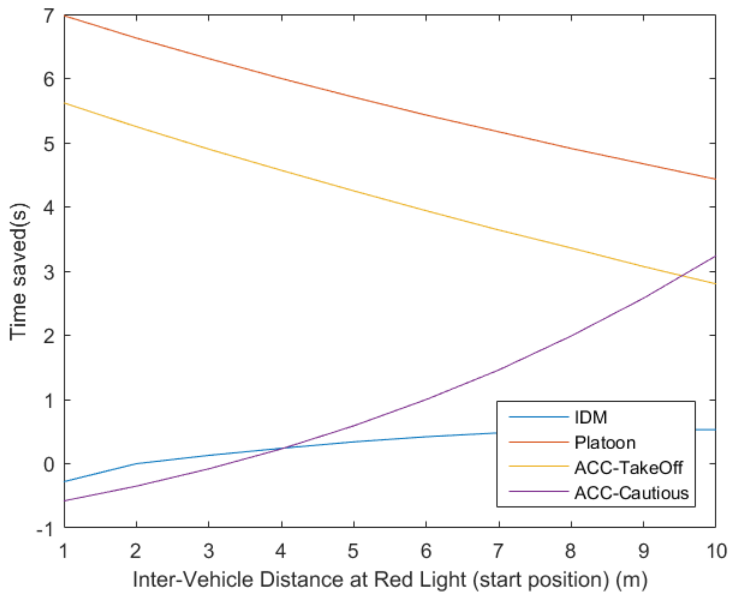

Figure 4.7: Parameter Study - Gap at Red Light ($s_{redlight}$)

Looking at figure 4.7, we see that in case of the IDM model (all human drivers), there is actually a small increase in time saved with high values of $s_{redlight}$. This is because if the cars are parked at the red-light at a distance larger than the minimum allowable gap (s_0), the cars can respond immediately when the light turns green and accelerate. Thus, if the gap is very large (ex.10m), the follower cars can accelerate and reach optimal speeds faster and as such pass the red-light faster. These results are in line with finding in the analytic study that was performed at Virginia Tech [16].

The other models behave in intuitive patterns based on their configuration. The ACC Cautious model that needs to achieve $s = s_{des}$ to perform well, starts at a high gap and as such can accelerate and performs better. The Platooning and ACC Take-Off models naturally are spaced out further and hence take a longer time to pass the red-light.

4.3 CO_2 Emissions

CO_2 emissions were estimated for each car in every case. The tool used for calculation of emissions is an in-house developed complete vehicle simulation software called VSim, owned by the Volvo Car Corporation. The analysis scripts are confidential and cannot be disclosed. Hence, the tool is being treated as a black box. But, the tool has been used by Volvo for over 2 decades to perform energy and emission calculations, so its merit and integrity have been tested.

The drive cycles obtained from the simulation are fed into VSim and a standard car model is selected. The car model selected was a standard all wheel drive car with automatic transmission. Initially, selecting a specific car model (ex. XC40/60) was prioritized, but modeling the entire car is complicated and also confidential, and as this is a comparative study, it was sufficient to select a standard model.

4.3.1 Optimal Acceleration

As mentioned multiple times in this report, finding an optimal (accurate) acceleration to represent the driving preferences of human driving as a whole is quite tough. Instead, another approach was chosen that could provide better insights into the scenarios.

Even though this is a comparative study, the results (Cumulative CO_2 emissions of cars for each test case) would not be transferable based on varying accelerations, mainly because this relationship is not linear. If the best possible case for humans could be determined i.e. the maximum acceleration value that would give the lowest cumulative CO_2 emissions for all 5 IDM cars, then that could be used as a benchmark and compared to the emissions of the car fleets of other test cases and the maximum possible savings can be determined.

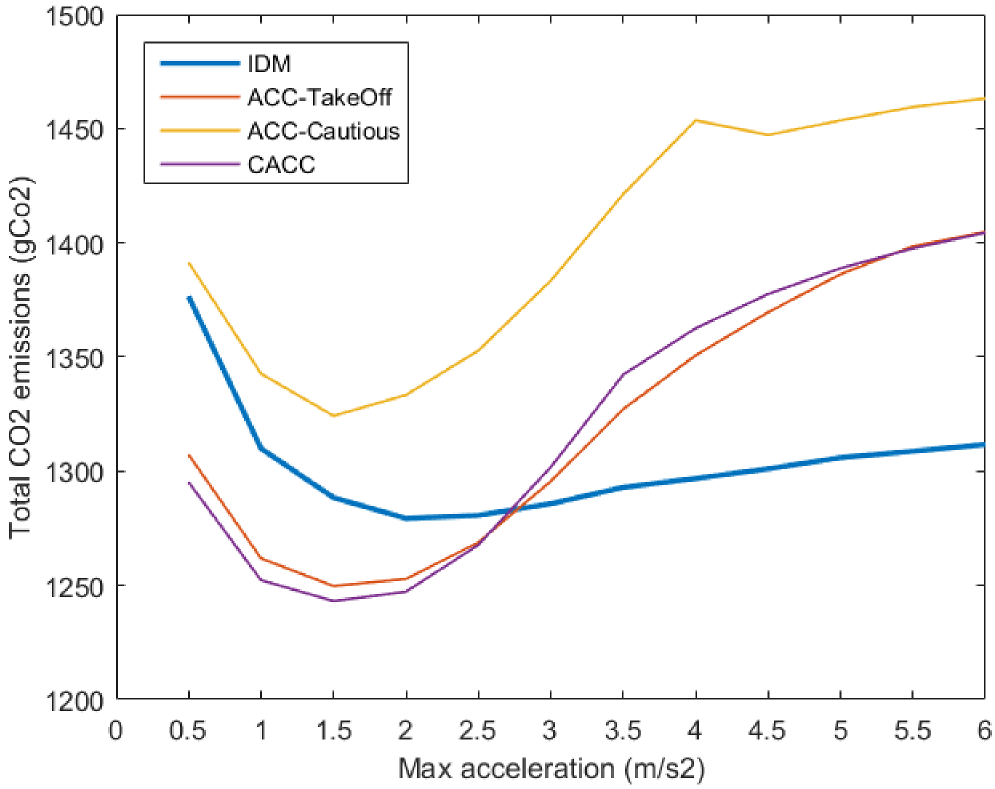

Figure 4.8: Cumulative CO_2 emissions of car fleet for each test case for varying values of maximum acceleration a

The results of this study using VSim can be seen in figure 4.8. Th lowest point on the IDM curve occurs at a=2ms^{-2} where the CO_2 emissions are 1279 gCO_2. This is the best possible case for human driving in the given simulation environment (as modelled by IDM). The reason why this happens is mostly related to the gear at which the car operational which leads to smoother operation of the engine and consequently lower emissions.

4.3.2 Maximum CO_2 Savings

As determined in the previous section, the lowest possible environmental impact occurs when a=2ms^{-2}. The simulation results at this acceleration can be seen in figure 4.9.

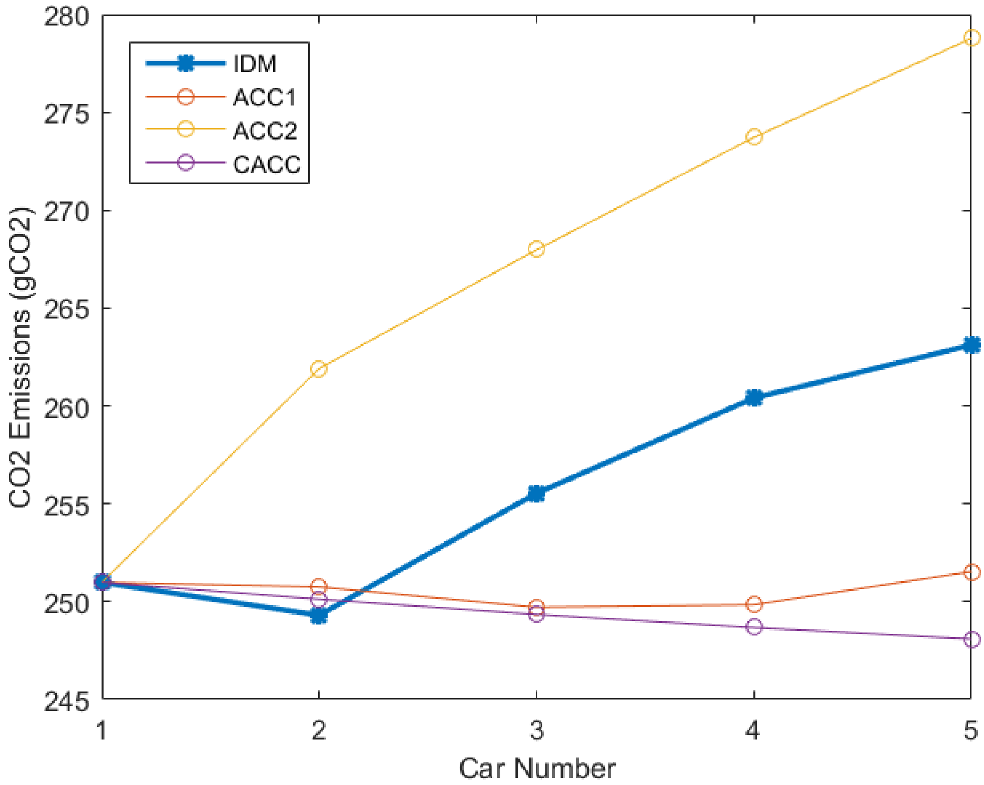

Figure 4.9: CO_2 emissions of each car for each test case for a=2ms^{-2}

The trends in emissions increase with the car number in the fleet. This is mainly due to higher interaction of cars towards the end of the queue which leads to deceleration that lowers performance. This is not seen for the platooning model as it has smooth increases and decreases in speed due to the V2V communication. The IDM considered as the reference scenario, so the cumulative emissions of the test cases can be compared to it to check the performance of the different driver models. The CO_2 savings can be found in table 4.2.

Table 4.2: Cumulative CO_2 emissions at a=2ms^{-2}

	$Total\ CO_2\ Emissions\ (gCO_2)$	$Difference(gCO_2)$
$Case\ 1:\ Reference\ Case$	1279.36	
$Case\ 2.1:\ ACC\ Cautious$	1333.38	+54.02
$Case\ 2.2:\ ACC\ Take-Off$	1252.86	-26.5
$Case\ 3:\ Platooning/CACC$	1247.25	-32.11

5
Discussion

Due to the scope and size of the simulation environment created, the results of this thesis can be further articulated and emboldened by critical discussion and scrutiny.

As mentioned earlier in section 2.2, before creation of the simulation environment, appropriate mathematical models must be chosen that are capable of appropriately representing automobiles in the real world. Most driver models vary in complexity and as such, vary in application, this leads to points of convergence where proper trade-offs are met. The relevance of a driver model may vary based on its applicability to certain scenarios, not all models perform well in varied scenarios (elaborated in section 3.1). One way of mimicking human drivers may lie in deep learning and neural networks that could be trained based on naturalistic data to perform as close to the real world case and optimized accordingly. This, again would lead to the problem of recreating and validating the model, and it would also be a victim to the quality of data collected, and would not be capable of adapting to outlying scenarios (all aspects of human driving might not be represented). But, microscopic-level modeling of vehicular traffic has been steadily growing in popularity due to the advent of better data collection methods and large data collection projects (ex. EuroFOT), the quality and magnitude of naturalistic data is improving, and as such traffic simulation results are becoming more reliable.

There is also the question of the system boundary chosen, it is restricted to a single lane, straight road that starts at a traffic light and has no intersections. Here, the question of flow improvements can be discussed, as traffic flow is a dynamic concept that is dependant on the layout of the entire road network, i.e. flow improvements at a single point could potentially cause both positive and negative externalities to traffic flow at other points in the network. Thus expanding the system boundary to include more intersections and maybe even more entry points for vehicles would prove to be beneficial in validating the results. But the current results still prove that there is potential saving to be had at traffic lights just by the use of ADAS systems, which, innately has merit in a traffic system. Additionally, the results of the platooning model are bound by the vehicles staying in the platoon, but if the vehicles had to follow different routes at intersections, the model would not be effective.

The parameters set in each model were done so based on a search for the scenario where human drivers could emit the lowest amount of CO_2. The cars are all identical and share the same parameters. This is not purely indicative, as most drivers have varied perceptions of "minimum allowed gap" and "maximum acceleration" etc. and having different cars in the queue could lead to string stability errors, and could change the results drastically. But, with the growth of ADAS systems and autonomy of driving systems, such parameters can be agreed upon on the basis of safety. These limitations were overlooked due to the time restriction of the conducted study. But, in certain highway scenarios, where the roads ate relatively straight and consists of fewer intersections, situations similar to the chosen environment could be possible.

The comparison of the the total emissions of the different test cases seen in table 4.2, show that improvements to emissions are only 0.025 %. Though this seems negligible, when applied to every vehicular queue leaving every traffic light in the world, the potential CO_2 savings would be significant. Due to confidentiality of VSim, the exact quality of results cannot be verified, but literature based on the reliability of VSim results titled "Verification of Virtual Vehicles" [15], could prove to add further insight.

6
Conclusion

The objective of this thesis was to use micro-level traffic simulation methods to ascertain the potential improvements to traffic flow and environmental impacts due to emissions caused due to synchronized take-off of vehicles at red lights. As the true nature of the working traffic systems is obscured by many different variables and uncertainties (such as human driving patterns, asymmetric intersections and synergies of traffic flows at intersections in a network), the system boundary of the simulation environment was restricted to only a single traffic light and a single lane road. Different test conditions were set up with the intention of simulating different driver preferences and also evaluating the chosen driver models, and simulation results were compared.

Flow improvements were determined by analyzing the time saved at the red light. In line with intuition, the synchronized take-off model showed the highest time savings, and as such the vehicle fleet could pass the red light the fastest among the test cases. Though this is a measure direct flow improvements, the synergy between flows at different intersections in a road network, plays an important role in determining the actual improvements to flow. As this was outside the scope of the thesis, it would be a fruitful avenue for further studies. The environmental impacts of the different test cases were calculated based on the total CO_2 emissions of the vehicle fleets. Here, the lowest impact to the environment was observed in the synchronized take-off case. The emission calculations were based on a complete vehicle simulation software developed by the Volvo Car Corporation called VSim.

It must be mentioned that achieving synchronized take-off is highly dependant on vehicle to vehicle communication, though it is still possible though just use of other proximity based ADAS systems. But in both cases, the main factor to scrutinize would be the minimum or "safe" gap that the vehicles need to maintain. As the synchronized take off method causes vehicles to accelerate at high rates with low safety distances, it may be a cause for concern to verify the models using physical tests under more critical (accident) conditions to properly validate them. The model used for synchronized take off in this case was a platooning model with all cars having the same routing path. But in reality, forming a vehicular platoon in general traffic is quite a daunting task, it requires the synchronization of onboard systems with infrastructure systems and must have fail-safes for human error. It might be that this kind of vehicular communication can be applied to specific driving conditions, such as intercity highway commute. It would also be possible to have vehicle to vehicle communication zones around points in a road network (such as intersections or red lights), which would enable vehicle fleets to interact with each other to achieve synchronized take off. The issue regarding mixed routing of individual cars in the fleet has not been investigated (due to time constraints), and could draw interesting conclusions to the test cases

Bibliography

[1] International Energy Agency (2014), *http://www.iea.org/statistics/statisticssearch/* Accessed: 05 June 2018.

[2] Mala and Varma (2016) *Minimization of Traffic Congestion by Using Queueing Theory*, IOSR Journal of Mathematics, Volume 12, Issue 1, PP 116-122

[3] Bharadwaj et.al. (2016) *Impact of congestion on greenhouse gas emissions for road transport in Mumbai metropolitan region*

[4] Rothery, Richard W. (1997) *Car Following Models*

[5] Wang, Jianqiang Li, Keqiang Lu, Xiao-Yun (2013) *Chapter 5. Effect of Human Factors on Driver Behavior*, 10.1016/B978-0-12-397199-9.00005-7

[6] Treiber, Hennecke and Helbing (2000) *Congested Traffic States in Empirical Observations and Microscopic Simulations*

[7] Wiedemann and Reiter (1992) *Microscopic Traffic Simulation: The Simulation System Mission*

[8] D. Swaroop and J. K. Hedrick (1996) *String stability of interconnected systems*, IEEE Trans. Autom. Control, Vol. 41, No. 3, pp. 349–357

[9] Higgs, Abbas and Medina (2011) *Analysis of the Wiedemann Car Following Model over Different Speeds using Naturalistic Data*

[10] Liang and Peng (199) *Optimal Adaptive Cruise Control with Guaranteed String Stability*, Vehicle System Dynamics, 32:4-5, 313-330

[11] Deng, Q (2016) *Heavy-Duty Vehicle Platooning - Modeling and Analysis*

[12] Krajzewicz, D et.al. (2016) *SUMO - Simulation of Urban Mobility*

[13] Gipps, P.G. (1981) *A behavioural car-following model for computer simulation*, Transportation Research Part B, 15, 105-11

[14] M.Treiber and A.Kesting (2013) *Traffic Flow Dynamics*, Chapter 11, Car-Following Models based on Driving Strategies

[15] Kathrikolli, A and Ratakonda, S. (2018) *Validation of Virtual Vehicles*

[16] S Farzad Ahmadi et al. (2017) *Latent heat of traffic moving from rest*, New Journal of Physics (2017). DOI: 10.1088/1367-2630/aa95f0

A

Appendix A

Matlab Code for the driver models

Intelligent Driver Model

```
clc
clear all
close all



%% Initialization

a = 2;                    %max desired acceleraion
b = 3;                    %desired braking
s0 = 2;                   %min gap
T = 1.5;                    %headway
delta = 4;                %velocity exponent
desiredSpeed = 70/3.6;
carLeng = 4;
Hz=0.01;
s_redLight = 10;           %gap from first car to redlight
target= 1000;             %target distance in m
%% Create Road
road = linspace(1,2000,2000);

%% Determine start postions of cars
carCount = 15;

x(1) = road(500);

redlight = x(1)+6;

for i1 = 1:(carCount-1)
    x(i1+1) = road(x(i1)-carLeng-s_redLight);  %positions of cars at the red light
end

%% Move the cars

for t=1:12000
    for i1 = 1:carCount
```

```
if t<2
    currentSpeed(i1,t) = 0;
else
    currentSpeed(i1,t) = v_car(i1,t-1);
end

v_delta=[];

%calculate current gap = s

s(i1,t)=50;   %Gap from first car at red light to car in front
if i1>1 && t==1
    s(i1,t) = (adv(i1-1) - carLeng) - x(i1);
elseif i1>1 && t>1
    s(i1,t) = adv(i1-1,t) - carLeng - adv(i1,t-1);
end

%Calculate approach rate = v_delta

v_delta = 0;                        %No car in front of first
if i1>1
    v_delta = currentSpeed(i1,t)-v_car(i1-1,t);
end

%Calculate velocity = v_car ---- IDM Model
if t==1
    v_car(i1,t) = 0;
else

    sStar = s0 + max(0,((currentSpeed(i1,t)*T)+
    ((currentSpeed(i1,t)*(v_delta))/(2*sqrt(a*b)))));

    acc(i1,t) = a*(1-((currentSpeed(i1,t)/desiredSpeed).^delta)-
    ((sStar./s(i1,t)).^2));

    v_car(i1,t) = max(currentSpeed(i1,t)+acc(i1,t)*Hz,0);

end
%Calculate advance of cars
if t==1
    adv(i1,t)=x(i1)+ v_car(i1,t)*Hz;
else
    adv(i1,t) = adv(i1,t-1)+v_car(i1,t)*Hz;
end

%Check when last car passes the red light
if i1==carCount
    check = ((adv(carCount,t)-carLeng)- redlight);
    if check > 0
    else
        LastCarPass=(t+1)*Hz;
    end
else
end
%Extract speed traces after crossing target distance
if (adv(i1,t)-carLeng)<(x(i1)+target)
```

```
            v(i1,t)= v_car(i1,t);
            dyc(i1).speedtrace=v(i1,:);
        end
    end
end

%%

% startv_car= zeros(5,1);
% v_car = cat(2,startv_car,v_car);

adv1= adv.';
%plot (adv1)%plot(s.')
v_car1=v_car.';
figure;
for i1= 1:carCount
    plot (dyc(i1).speedtrace,'LineWidth',1);
    hold on
    legend('1','2','3','4','5','Location','Southeast');
end
set(gca,'XTick',0:1000:8000)
set(gca,'XTickLabel',0:10:80)
xlabel('time - seconds')
ylabel('velocity m/s')
text(4000,8,sprintf('5thCarPass=%.2f s',LastCarPass));
title('IDM');
save('Dyc_IDM.mat', 'dyc')
```

Adaptive Cruise Control Model

```
clc
clear all
close all



%% Initialization

a = 2;                    %max desired acceleraion
b = 3;                    %desired braking
s0 = 2;                   %min gap
sdes=10;                  %desired vehicle gap
T = 1.5;                  %headway
delta = 4;                %velocity exponent
desiredSpeed = 70/3.6;
carLeng = 4;
s_redLight = 2;
kv = 1.70;
kr = 1.12;
c=1;
Hz=0.01;
target=1000;
%% Create Road
road = linspace(1,2000,2000);

%% Determine start postions of cars
carCount = 5;

x(1) = road(500);

redlight = x(1)+6;

for i1 = 1:carCount
    x(i1+1) = road(x(i1)-carLeng-s_redLight);    %positions of cars at the red light
end

%% Move the cars

currentSpeed = [];
for t=1:12000
    for i1 = 1:carCount
        if t<2
            currentSpeed(i1,t) = 0;
        else
            currentSpeed(i1,t) = v_car(i1,t-1);
        end

        v_delta=[];

        %calculate current gap = s
```

```
s(i1,t) = 50;   %Gap from first car at red light to car in front
if i1>1 && t<2
    s(i1,t) = (adv(i1-1) - carLeng) - x(i1);
elseif i1>1 && t>=2
    s(i1,t) =(adv(i1-1,t) - carLeng) - adv(i1,t-1);
end

%Calculate approach rate = v_delta

v_delta = 0;                        %No car in front of first
if i1>1
    v_delta = currentSpeed(i1,t)-v_car(i1-1);
end

%Calculate velocity = v_car
v_car(i1,1) = 0;

if t>1

    if i1>1     %ACC model (followers)
        acc(i1,t) = min(a,(kv*(v_car(i1-1,t) - v_car(i1,t-1))*c +
        kr*(s(i1,t)-sdes)*c));

        v_car(i1,t) = max(currentSpeed(i1,t)+(acc(i1,t)*Hz),0);


        v_car(i1,t) = min(v_car(i1,t),v_car(1,t));


    else        %IDM model  (leader)
        sStar = s0 + ((currentSpeed(i1,t)*T)+
        (currentSpeed(i1,t)*v_delta)/(2*sqrt(a*b)));

        acc(i1,t) = a*(1-((currentSpeed(i1,t)/desiredSpeed).^delta)-
        (sStar./s(i1,t)).^2);
        v_car(i1,t) = currentSpeed(i1,t)+(acc(i1,t)*Hz);
    end
end

%Calculate advance of cars
if t<2
    adv(i1,t)=x(i1)+ +(v_car(i1,t)*Hz);
else
    adv(i1,t) = adv(i1,t-1)+(v_car(i1,t)*Hz);
end
if i1==carCount
    check = ((adv(carCount,t)-carLeng)- redlight);
    if check > 0
    else
        LastCarPass=(t+1)*Hz;
        m=check;
    end
else
end
%Extract speed traces after crossing target distance
if (adv(i1,t)-carLeng)<(x(i1)+target)
```

```
            v(i1,t)= v_car(i1,t);
            dyc(i1).speedtrace=v(i1,:);
        end
    end
end

%%

% startv_car= zeros(5,1);
% v_car = cat(2,startv_car,v_car);

adv1= adv.';
acc1=acc.';

%plot (s.')
%figure(1);
% plot (adv1);
%legend('1','2','3','4','5');

v_car1= v_car.';

figure;
for i1= 1:carCount
    plot (dyc(i1).speedtrace,'LineWidth',1);
    hold on
    legend('1','2','3','4','5','Location','Southeast');
end
set(gca,'XTick',0:1000:8000)
set(gca,'XTickLabel',0:10:80)
xlabel('time - seconds')
ylabel('velocity m/s')
text(3050,8,sprintf('5thCarPass=%.2f s',LastCarPass));
title('ACC - Cautious');
save('Dyc_ACC_2.mat', 'dyc')

% figure(3);
% plot (acc1);
% legend('1','2','3','4','5');
```

Platooning Model

```
clc
clear all
close all



%% Initialization

a = 2;                  %max desired acceleraion
b = 3;                  %desired braking
s0 = 2;                 %min gap
sdes=10;                %desired vehicle gap
T = 1.5;                %headway
delta = 4;              %velocity exponent
desiredSpeed = 70/3.6;
carLeng = 4;
s_redLight = 2;
kv = 0.3;
kr = 0.01;
c=1;
Hz=0.01;
target=1000;
%% Create Road
road = linspace(1,2000,2000);

%% Determine start postions of cars
carCount = 5;

x(1) = road(500);

redlight = x(1)+6;

for i1 = 1:carCount
    x(i1+1) = road(x(i1)-carLeng-s_redLight);    %positions of cars at the red light
end

%% Move the cars

currentSpeed = [];
for t=1:12000
    for i1 = 1:carCount
        if t<2
            currentSpeed(i1,t) = 0;
        else
            currentSpeed(i1,t) = v_car(i1,t-1);
        end

        v_delta=[];

        %calculate current gap = s
```

```
    s(i1,t) = 50;       %Gap from first car at red light to car in front
    if i1>1 && t<2
        s(i1,t) = (adv(i1-1) - carLeng) - x(i1);
    elseif i1>1 && t>=2
        s(i1,t) =(adv(i1-1,t) - carLeng) - adv(i1,t-1);
    end

    %Calculate approach rate = v_delta

    v_delta = 0;                        %No car in front of first
    if i1>1
        v_delta = currentSpeed(i1,t)-v_car(i1-1);
    end

    %Calculate velocity = v_car
    v_car(i1,1) = 0;

    if t>1

        if i1>1    %CACC model (followers)
            acc(i1,t) = min(a,(acc(i1-1,t) + kv*(v_car(i1-1,t) -
            v_car(i1,t-1))*c + kr*(s(i1,t)-sdes)*c));

            v_car(i1,t) = min(currentSpeed(i1,t)+(acc(i1,t)*Hz),desiredSpeed);

        else       %IDM model  (leader)
            sStar = s0 + ((currentSpeed(i1,t)*T)+
            (currentSpeed(i1,t)*v_delta)/(2*sqrt(a*b)));

            acc(i1,t) = a*(1-((currentSpeed(i1,t)/desiredSpeed).^delta)-
            (sStar./s(i1,t)).^2);
            v_car(i1,t) = currentSpeed(i1,t)+(acc(i1,t)*Hz);
        end
    end

    %Calculate advance of cars
    if t<2
        adv(i1,t)=x(i1)+ +(v_car(i1,t)*Hz);
    else
        adv(i1,t) = adv(i1,t-1)+(v_car(i1,t)*Hz);
    end
    if i1==carCount
        check = ((adv(carCount,t)-carLeng)- redlight);
        if check > 0
        else
            LastCarPass=(t+1)*Hz;
            m=check;
        end
    else
    end
    %Extract speed traces after crossing target distance
    if (adv(i1,t)-carLeng)<(x(i1)+target)
        v(i1,t)= v_car(i1,t);
        dyc(i1).speedtrace=v(i1,:);
    end
end
```

```
end


%%

% startv_car= zeros(5,1);
% v_car = cat(2,startv_car,v_car);

adv1= adv.';
acc1=acc.';

plot(s.')
% figure(1);
% plot (adv1);
% legend('1','2','3','4','5');

v_car1= v_car.';

figure;
for i1= 1:carCount
    plot (dyc(i1).speedtrace,'LineWidth',1);
    hold on
    legend('1','2','3','4','5','Location','Southeast');
end
set(gca,'XTick',0:1000:7000)
set(gca,'XTickLabel',0:10:70)
xlabel('time - seconds')
ylabel('velocity m/s');
text(2400,8,sprintf('5thCarPass=%.2f s',LastCarPass));
title('CACC - Platooning');

save('Dyc_Platoon.mat', 'dyc')

% figure(3);
% plot (acc1);
% legend('1','2','3','4','5');
```