Smart pairing of autonomous vehicles at lightless intersections

Tobias Glauser
Simon Eck

Master Project Thesis
Civil Engineering

July 2020

Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Table of Contents

Master Project Thesis, Civil Engineering

Smart pairing of autonomous vehicles at lightless intersections

Tobias Glauser
IVT ETH Zurich
Stefano-Franscini-Platz 5,
8093 Zurich, Switzerland

E-Mail: glauseto@ethz.ch

Simon Eck
IVT ETH Zurich
Stefano-Franscini-Platz 5,
8093 Zurich, Switzerland

E-Mail: seck@ethz.ch

July 2020

Abstract

This Master Project Thesis investigates the optimal crossing of vehicles at a four-leg intersection without traffic lights. Each leg has two approaching and two departing lanes. One approaching lane is dedicated to left-turning vehicles and the other one to vehicles driving straight and turning right. It is assumed, that all vehicles are connected and fully autonomous. If two vehicles have a conflict-free movement, they can build a pair and pass the intersection simultaneously. The simulation is implemented in MATLAB. Various parameters of the model are studied using a sensitivity analysis. As a result, the optimal choice of the parameters is proposed, and the simulation results are discussed. In the end, possible future work and improvements to the simulation setup are suggested.

Keywords

Connected and autonomous vehicles; Lightless intersection; Virtual platoon; Pairing

Preferred citation style

Glauser, T. and Eck, S. (2020) Smart pairing of autonomous vehicles at lightless intersections, Master Project, IVT, ETH Zurich, Zurich.

1 Introduction

Today there are many ways to control the traffic at intersections. There exist intersections with a traffic light or without, or completely other solutions like roundabouts. The different solutions have their advantages and disadvantages and are used in different situations. At Intersections with high demands from all directions, traffic lights are one of the most used solutions in Switzerland. They allow optimising the capacity by controlling the traffic flows. They also simplify the situation at complex intersections, where a lot of traffic signs can lead to confusion.

Traffic lights at intersections have the advantage, that the cars from all directions are fairly considered when the stages are well adjusted. There are traffic lights with a fixed cycle length, where the green time for every stage is predefined. Additionally, using traffic detectors, the green time for all users can be optimized. This control system is called actuated traffic light control. There are two categories: semi-actuated and fully actuated. A semi-actuated controller uses sensors to detect waiting vehicles at an intersection on some or on all lanes. The traffic light controller then calculates the optimal green and red time according to the presence of vehicles from other directions. The cycle length is not adjustable. At fully actuated traffic lights, the demand of all phases influences the duration of the stages. This way the cars can be served more efficiently.

Nevertheless, traffic lights also have disadvantages. There is unproductive time, where no car can pass the intersection. The reason for this is, that there must be some extra time between the green phases to ensure a safe crossing of all the vehicles. This time is needed to ensure, that cars have enough time, to clear the intersection if they pass at a yellow light. It reduces the risk of a collision at the intersection. On one side, one would like to increase the cycle length as much as possible in order to minimise the all-red time. On the other side, people don't like to wait for a long time until they can pass the intersection. Especially, if no cars are approaching from other directions. Therefore, the cycle length shouldn't be too long. This dilemma can partly be solved with actuated control. But even then, a fast switch between the stages is not possible since it would increase the all-red time and therefore would be inefficient.

In the future connected autonomous vehicles (CAVs) can solve these problems. The technology develops fast and today automated vehicles are already on the streets. Thus, they still require a driver who monitors the system. However, if CAVs can drive completely autonomous and can communicate with each other and with the infrastructure, full automation can be achieved.

If all vehicles in a traffic network are CAVs, traffic lights would not be needed anymore. The controller can coordinate the movements of all vehicles, allowing for rapid succession of vehicles from different directions. Nowadays a lot of intersections without traffic lights exist. They have the advantage, that the cars can drive as soon as there is enough space for them. This works fine for small demand but with increasing traffic from all directions, this gets difficult and can be impossible for cars from side roads to enter or cross the main traffic stream. This is one of the reasons why traffic lights are necessary.

Xu, et al., 2018 proposes that all approaching cars at an intersection are projected onto a virtual lane. On this lane, they can order themselves and later cross the intersection without the need for a traffic light. During this process, the cars are communicating with a controller which optimises the speed of all cars for an efficient and conflict-free crossing. Cars with non-conflicting movements can be paired together to reach a higher capacity. The intersection can be designed lightless.

In this thesis, a simulation of such a lightless intersection with CAVs is investigated. For the simulation, various parameters to control the behaviour of the cars are needed. This thesis contains a sensitivity analysis of these parameters to understand their influence. The goal is to find optimal values for an accident-free and efficient crossing. In chapter 3, the methodology of the simulation is explained. The simulation setup and problem statement are described as well as the solution approach. In chapter 4, all results of the simulation are shown and in the following chapter 5, these results are discussed. At the end of this thesis, some possible future researches and improvements are proposed.

2 Background

The topic of connected and autonomous vehicles is a popular research topic. Xu, et al., 2018 proposes, as mentioned in the introduction, a virtual lane to order and optimise the crossing of the intersection for all CAVs. This virtual lane is also named virtual platoon. He, Zheng, Lu, & Guan, 2018 further proposes, that the lanes are no longer dedicated to one direction. This makes lane changes unnecessary. In the end, they achieved better performance with this new model. Another proposal is the use of model predictive control (MPC) as in Makarem & Gillet, 2013. The use of MPC results in smoother driving behaviour.

At the ETH Zurich, several projects have been done on this topic. In the bachelor thesis of Josia Meier (Meier, 2019) a simulation of a lightless intersection for CAVs was done, based on the principle of a virtual platoon which was introduced by Xu et al. The simulation was coded in MATLAB. The master project thesis of Andrea Galli (Galli, 2019) continued this work and replenished it with an animation of the simulation. Further, a more detailed analysis of the simulation was done.

This thesis continues the work of Meier and Galli with some changes and improvements to the simulation. It started with the existing MATLAB code from these two theses. The theses of Galli and Meier are based on an intersection layout with four legs. Every leg has six lanes, three of them approaching the intersection and three lanes leading away from it. The vehicles that turn right are neglected for the simulation since they have no conflict with any other vehicles crossing the intersection. All directions have a dedicated approaching and departing lane. In this report, the intersection layout has been changed to a more realistic one, with only four lanes per leg and where all directions are considered.

The existing simulation used a lot of parameters to describe the dynamics of the vehicles. For example, the minimum distance between cars in the same lane and on the virtual platoon were predefined. Furthermore, weightings to control the optimisation problem were introduced. However, the influence of these variables was not further investigated. A major part of this thesis is a sensitivity analysis for the parameters to understand which values are the most optimal ones.

For a better use of the intersection capacity, the cars build pairs to cross the intersection together. These pairs haven’t been visible in the animation so far. Now, the new improved animation shows exactly which cars are paired. This helps for a better understanding of the results of the simulation.

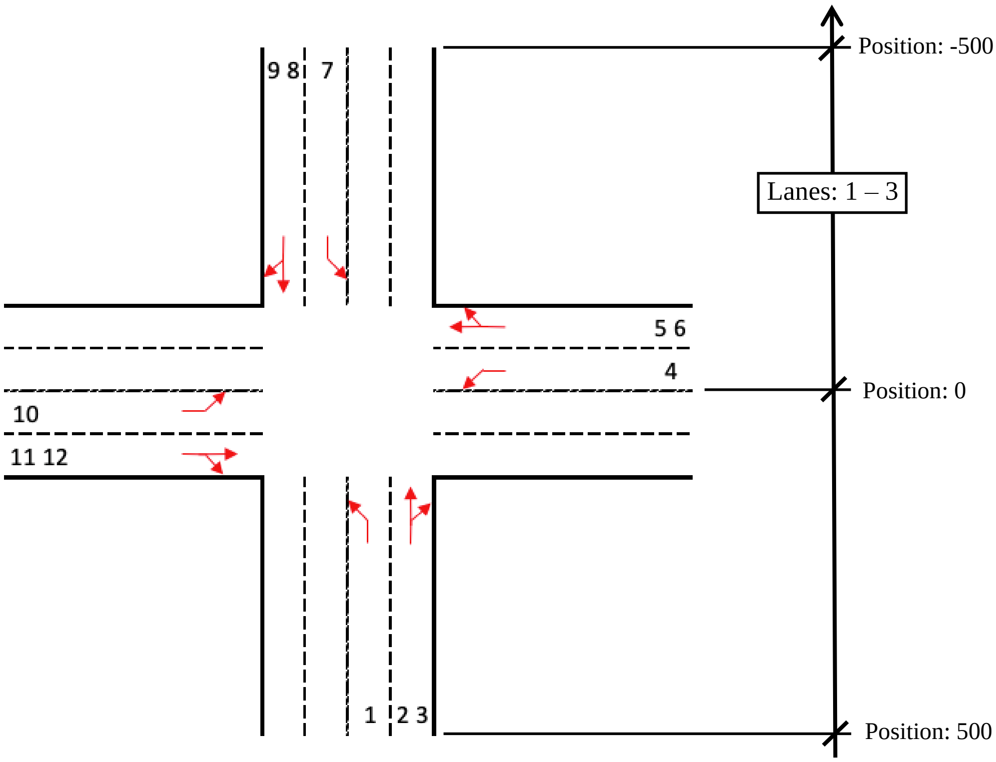
3 Methodology

In this chapter, the functionality of the simulation is explained. First, the used intersection layout is presented in chapter 3.1. Then, the necessary assumptions are briefly listed in chapter 3.2. In chapter 3.3, the concept of the virtual platoon, which is used in the simulation is explained. For the simulation, various equations are necessary for the basic dynamics of the vehicles and further for the optimisation. All of this is explained in chapter 3.4. Chapter 3.5 is about the setup of the simulation and in chapter 3.6 a short overview of the MATLAB code is given. In chapter 3.8, it is explained how the sensitivity analysis was done and in the last part of this chapter, the used performance metrics are shown.

3.1 Intersection Layout

In Figure 1, the new intersection layout is shown. The intersection consists of four legs. Every leg has four lanes, two approaching and two departing lanes, respectively. One of the approaching lanes is for the cars turning to the left and the other for those that are driving straight or turning right. The directions are numbered counterclockwise. On every entering side, there are two directions are using the same lane, for example, the direction number two and three. The legs have each a length of 500 meters, where the entry position is at the point 500 and the intersection at the point 0. The total length the cars travel inside the system is therefore 1000 meters. For this report, it is assumed that the cars do not change lanes, which means they already approach in the lane of their intended direction. For this simulation, no vehicles are neglected as in the thesis of Galli (Galli, 2019) and Meier (Meier, 2019). In the simulation, two scenarios with a different demand split will be considered, which are specified in chapter 3.4.4.

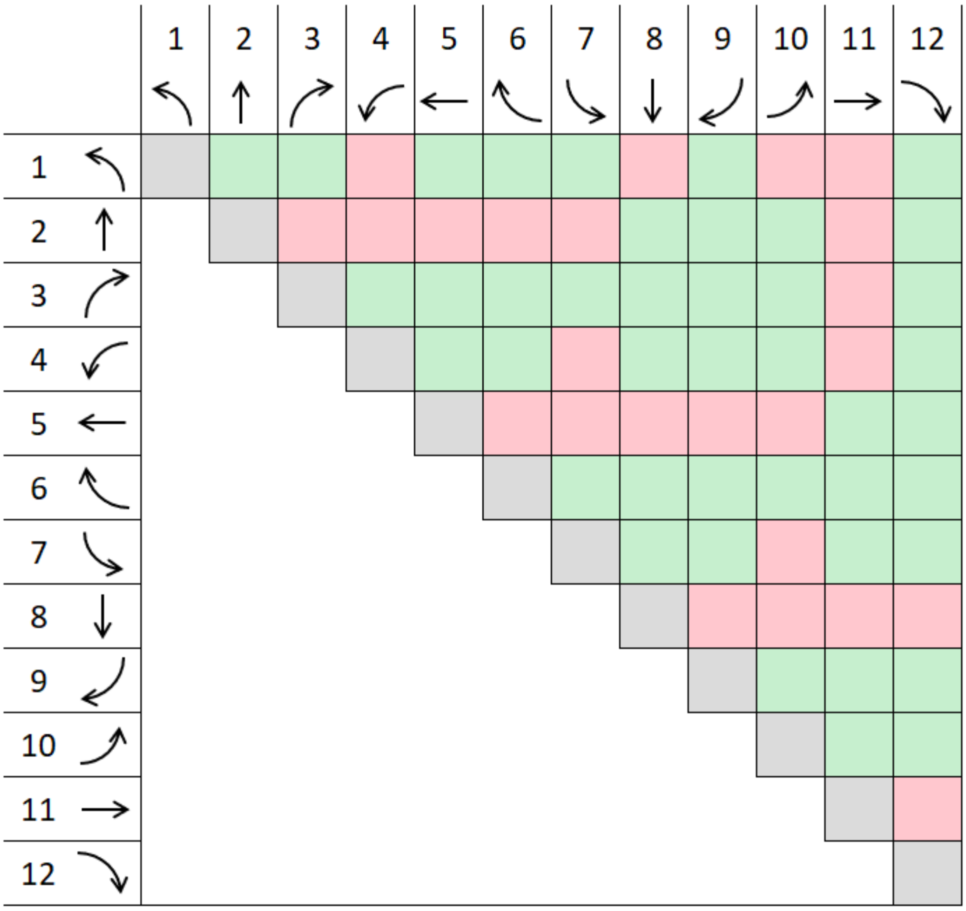
Figure 1 Intersection Layout



On this intersection, certain cars can cross at the same time. This depends on their approaching and departing direction. In Figure 2 all compatible movements are shown. The colour green indicates that the two directions have a compatible movement, where they can cross the intersection simultaneously. On the other side, the colour red indicates that there is a conflict, so these directions cannot cross at the same time.

The number of possible compatible movements depends on the direction of the vehicle. For vehicles turning right, there are nine compatible movements. For vehicles going left seven movements are compatible and for the direction going straight only five. It would be possible that more than two cars can cross the intersection simultaneously with this layout. But for this project, only pairs with up to two cars are considered.

Figure 2 Compatible Movements



3.2 Assumptions

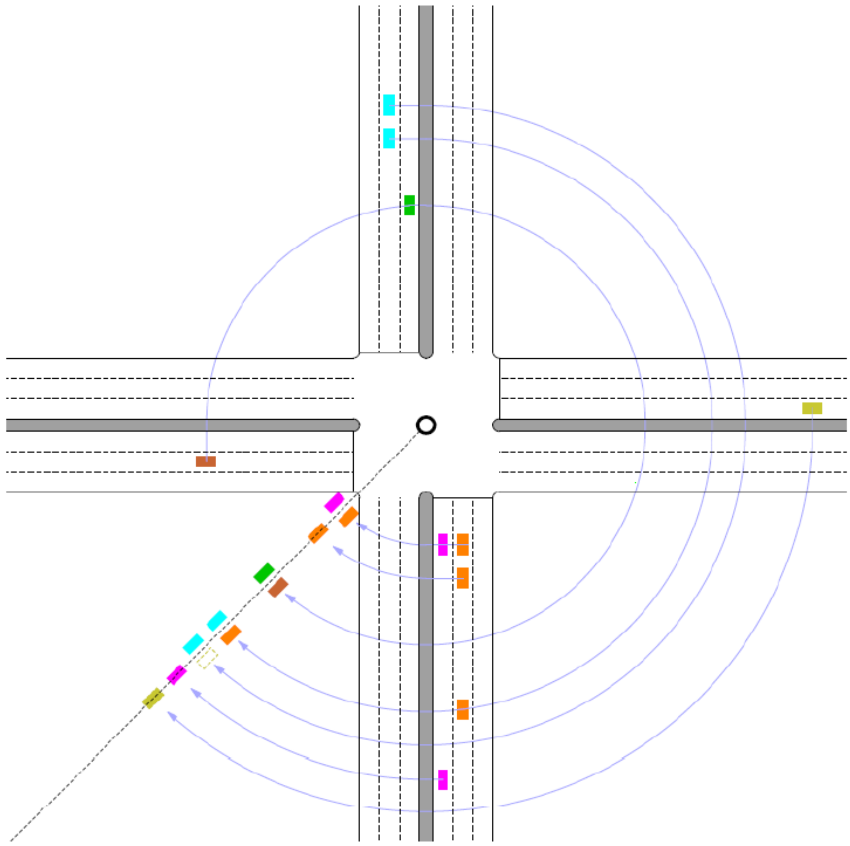
For the simulation model, some assumptions are made. Every model is a simplification of reality. It helps to understand and examine a complex problem. The assumptions are the following:

- Cars are modelled as a point
- No lane-changing of cars in the system
- Only pairs with two cars
- No external influences
- Connection and automation of vehicles is assumed, but not specified
- 100% of vehicles are CAVs

3.3 Concept of Virtual Platoon

For the simulation, the concept of a virtual platoon (Xu, et al., 2018) was used. The cars from all directions are projected onto one virtual lane. With this method, it is possible to see in which order the cars cross the intersection. On the approaching lanes, the cars adjust their velocity to order themselves on the virtual lane. They keep a distance to their virtual leader that a conflict-free crossing is possible. Further, cars can build pairs. The partner car can order himself to the same position as his leader on the virtual platoon. In Figure 3 an example of a virtual platoon is visible. All cars are ordered and kept the distance to their leader. In addition, pairs are visible in this figure.

Figure 3 Virtual Platoon



Source: (Meier, 2019) based on (Xu, et al., 2018)

3.4 Optimization Problem

The goal of the simulation is that all cars can cross the intersection conflict-free and as fast as possible. For this, an optimisation problem is solved. It is implemented in MATLAB. For the objective function, various boundary conditions must be considered. The optimization occurs at every time step for every car in the system.

3.4.1 Car-Model

The Car-Model, which was introduced in the thesis of Meier (Meier, 2019), needs to be applied to all cars for every timestep. The equation can be derived using simple physics for moving objects. It calculates the position x_i (Equation 1) and the velocity v_i (Equation 2) of every car i. a_i denotes the acceleration of the car i, t is the time in the simulation and Δt the timestep of the simulation. The signs are negative because the cars enter the system at 500 meters and the position decreases to -500 meters. The intersection is located at position 0.

$$x_i(t+1) = x_i(t) - v_i(t) * \Delta t - \frac{1}{2} a_i(t) * \Delta t^2 \tag{1}$$

$$v_i(t+1) = v_i(t) - a_i(t) * \Delta t \tag{2}$$

3.4.2 Intelligent Driver Model

For determining the distance between two cars the intelligent driver model (IDM) was used (Treiber, Hennecke, & Helbing, 2000). The IDM-distance is a distance between two following cars and is based on the actual speed of both cars. It is applied between the vehicles in the virtual platoon and therefore it’s also the distance with which the cars will cross the intersection. The IDM-distance d_{IDM} consists of three parts. In the first part, s_0 is a minimum safety distance that always must be kept between the cars. This term would remain if the speed of both vehicles is zero. The second part defines a headway for the speed of the car where $v_i(t)$ is the speed of car i, t is the time step in the simulation and T denotes the desired headway between the cars. The third term is related to the relative difference of the velocity of both vehicles. Here, $v_{\mathrm{vl}}(t)$ represents the velocity of the virtual leader on the virtual platoon. Further, $a_{\max}$ describes the maximum acceleration and b_{desired} the desired breaking acceleration for car i.

$$d_{\mathrm{IDM}} = s_0 + v_i(t) * T + \max\left(0, \frac{v_i(t) * \left(v_i(t) - v_{\mathrm{vl}}(t)\right)}{2 * \sqrt{a_{\max} - b_{\mathrm{desired}}}}\right) \tag{3}$$

3.4.3 Objective function

The objective function is formulated according to the options illustrated in Figure 4. These options are introduced in order to assign to each vehicle a leader on the virtual platoon, except for the first one. Each vehicle i can be assigned to one of the four different options. The options are described in the following.

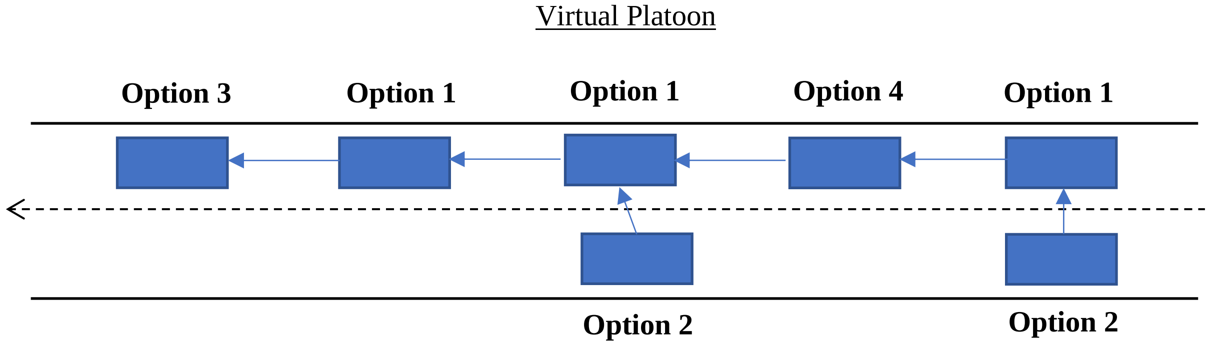
Option 1
For a vehicle assigned to option 1, its leader in the virtual platoon is directly in front (i following $i-1$).

Option 2
The relationship of vehicles, which are partnered with another vehicle, is described as option 2. The partner (leader) in the virtual platoon is directly in front of the vehicle $i+1$ ($i+1$ partnered with i). The optimisation problem for the partner vehicle is solved in the same iteration as for the leader of the pair.

Option 3
Only one vehicle is assigned to option 3. It is the vehicle, which is the first one on the virtual platoon.

Option 4
Option 4 corresponds to vehicles, which are the leader of a pair. In contrast to option 1, their leader on the virtual platoon has a partner vehicle (i following $i-2$).

Figure 4 Options of Vehicle Relationships on the Virtual Platoon



For all options, the objective function must be minimised. All terms are squared in order to get only positive numbers. In general, the objective function consists of three parts. Each part is indicated in the equations below:

1. The distance to the leader on the virtual platoon d_{IDM} (Equation 3) should be minimised.
2. The difference to the desired velocity v_{des} should be minimised.
3. The change in velocity $a_i(t)$ should be minimised.

To combine the different terms in one single objective function, weights were introduced. They are the following:

- w_{IDM}: Weight of the IDM-distance term, ensuring a safe the spacing of the vehicles on the virtual platoon
- $w_{\mathrm{No-IDM}}$: Weight, if the IDM-distance is not used, the car builds a pair with its leader
- w_{des}: Weight of the desired speed term
- w_{acc}: Weight of the acceleration term

In the following, the objective function for the four different options is stated. The goal is to minimise the output of the function subjected to the boundary conditions described in chapter 3.4.4.

Option 1

$$\min_{a_i} f\big(a_i(t)\big) = w_{\mathrm{IDM}} * \underbrace{(x_i(t) - x_{\mathrm{vl}}(t) - d_{\mathrm{IDM}})^2}_{1} + w_{\mathrm{des}} * \underbrace{(v_{\mathrm{des}} - v_i(t))^2}_{2} + w_{\mathrm{acc}} * \underbrace{a_i(t)^2}_{3} \tag{4}$$

s. t. Equation (8), (9) and (10)

Option 2

$$\min_{a_i} f\big(a_i(t)\big) = w_{\mathrm{No-IDM}} * \underbrace{(x_{i+1}(t) - x_{\mathrm{vl}}(t))^2}_{1} + w_{\mathrm{des}} * \underbrace{(v_{\mathrm{des}} - v_{i+1}(t))^2}_{2} + w_{\mathrm{acc}} * \underbrace{a_{i+1}(t)^2}_{3} \tag{5}$$

s. t. Equation (8), (9) and (10)

Option 3

$$\min_{a_i} f\big(a_i(t)\big) = w_{\mathrm{des}} * \underbrace{(v_{\mathrm{des}} - v_i(t))^2}_{2} + w_{\mathrm{acc}} * \underbrace{a_i(t)^2}_{3} \tag{6}$$

s. t. Equation (8), (9) and (10)

Option 4

$$\min_{a_i} f\big(a_i(t)\big) = w_{\mathrm{IDM}} * \underbrace{(x_i(t) - x_{\mathrm{vl}}(t) - d_{\mathrm{IDM}})^2}_{1} + w_{\mathrm{des}} * \underbrace{(v_{\mathrm{des}} - v_i(t))^2}_{2} + w_{\mathrm{acc}} * \underbrace{a_i(t)^2}_{3} \tag{7}$$

s. t. Equation (8), (9) and (10)

3.4.4 Boundary Conditions

The vehicles must fulfil some boundary conditions for a realistic and collision-free movement. The acceleration $a_i(t)$ of the car i must be within reasonable physical boundaries. They are the predefined boundaries of the maximum breaking acceleration b_{max} and the maximum acceleration a_{max} (Equation 8). The speed $v_i(t)$ of the car i must be positive and below the maximum speed v_{max} (Equation 9). If there is a vehicle in front of car i in the same lane (actual leader), the safety distance s_0 must be kept to this car, where $x_i(t)$ represents the positions of car i and $x_{\mathrm{al}}(t)$ the position of the actual leader (Equation 10). These boundary conditions have also been formulated in the thesis of Meier (Meier, 2019).

$$b_{\mathrm{max}} \leq a_i(t) \leq a_{\mathrm{max}} \tag{8}$$

$$0 \leq v_i(t) \leq v_{\mathrm{max}} \tag{9}$$

$$x_i(t) \geq x_{\mathrm{al}}(t) + s_0 \tag{10}$$

3.5 Simulation Setup

The time step of the simulation was defined as one second. For the simulation, a fixed number of cars is used. The simulation duration was chosen long enough, that all vehicles were able to pass the whole system. This was done, to have a good comparison between the results of the sensitivity analysis and the different demand splits. Every simulation scenario was repeated five times, because the results have some variation, due to the Poisson distributed generation of the vehicles (Chapter 3.6.3). With multiple runs, it was possible to calculate an average value and the standard deviation. The simulation setup can be seen in Table 1.

Table 1 Simulation setup for MATLAB

Variable	Value
Timestep of simulation [s]	1
Simulation duration of one run [s]	4000
Generated cars of one run	500
Runs per scenario	5

The simulation was done for two different demand splits. They are called scenario 1 and 2. Scenario 1 uses a regular demand split over all four approaching legs and assumes that more cars pass the intersection straight then turning left or right. For the second scenario, the demand split of the example of chapter 7.1 in the book "Principles of Highway Engineering and Traffic Analysis" (Mannering & Washburn) was used. The demand split for both scenarios is shown in Table 2.

Table 2 Demand split of Scenario 1 and 2

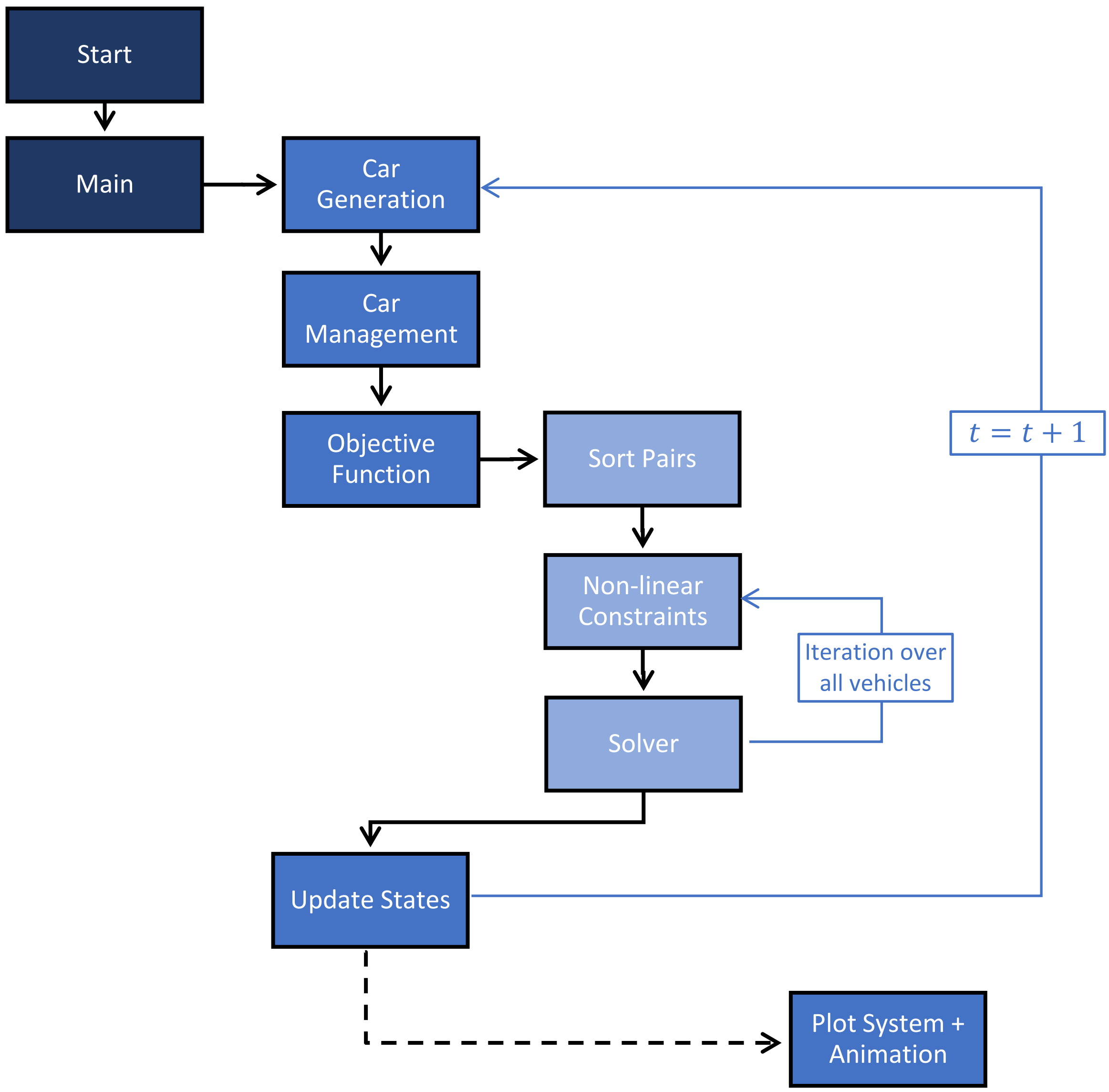
Direction	Scenario 1 [%]	Scenario 2 [%]
1	6	2.4
2	13	9.1
3	6	1.3
4	6	6.7
5	13	26.9
6	6	4
7	6	1.9
8	13	8.3
9	6	1.6
10	6	8.1
11	13	24.2
12	6	5.4
Total	100	100

Source (Scenario 2): (Mannering & Washburn)

3.6 MATLAB Code

The simulation is implemented using MATLAB. The implementation of the plain system was utilised from (Galli, 2019). Nevertheless, due to the new intersection layout (Chapter 3.1) some extensions are necessary. The code consists of different functions (Figure 5). In this chapter, they are briefly explained.

Figure 5 Overview of MATLAB Functions



3.6.1 Start

In this function, all parameters can be defined. One can also determine the number of desired runs with the chosen parameters. The function "Main" will be executed according to the defined runs. The average value and standard deviation of the performance metrics are calculated according to the number of runs. Finally, the results of the simulation are stored and plotted.

3.6.2 Main

This is the main function of the code. The function gets the input-parameters of the start function. Next, it iterates over the following functions calling them in the right order.

3.6.3 Car Generation

In this function, the cars are randomly generated with a Poisson distribution for the given arrival rate per hour. Then the generated cars are randomly distributed unto the twelve directions with the given weighting of the demand split.

3.6.4 Car Management

The generated cars of the former function are now placed on the lanes. Before they get placed, the function checks if the lane is in uncongested condition. A lane is considered uncongested if the actual leader on the same lane is beyond the IDM-distance (3.4.2) for a new car. In this case, the car is put in the system with the desired speed. Otherwise, if the gap to the actual leader is shorter, but the IDM-distance can be kept by vehicles entering the system with a velocity of 0 m/s, the vehicle can still enter the system. In this case, the condition on this lane is considered to be congested. If it's not possible to put the car in the system, the cars have to wait in a queue.

In the next calculation step, this procedure will be repeated until all cars can either be placed in the system or must wait in the queue. All information about the vehicles entering the system is saved in three matrixes. The matrix X stores the position, the matrix V the velocity and the matrix A the acceleration. For every matrix the columns $1 - n$, with n denoting the number of cars, store the data of one vehicle and represent the order of the cars on the virtual platoon. The rows $1 - k$, with k being the simulation duration, represent the time step of the simulation. The matrix X has three extra rows at the end to store additional information about the vehicles. In row $k + 1$, the information, if a car has a partner can be found. In row $k + 2$, the information about the car's direction is stored and in the last row ($k + 3$) the number of the pair is stored, if the car has a partner.

3.6.5 Objective Function

In this function, the objective function described in chapter 3.4.3 is implemented. First, the vehicles are getting paired with the function “Sort Pairs”. For all new entering cars, it is checked, if there is a possible partner in the system. Possible partners must still be single, must have a conflict-free movement and must be within reachable distance. This is done with the use of the pair-building distance ($d_{constraint}$). In the next step, the function iterates over all vehicles in the system. The “Nlcon” function determines the nonlinear constraints. They are explained in chapter 3.4.4. Then the optimisation is done with the MATLAB function “Fmincon”. The result is the acceleration for the vehicle in the specific time step.

3.6.6 Update States

In this part of the code, the calculated accelerations are applied for the next time step by using the formulas in chapter 3.4.1.

3.6.7 Plot System and Animation

The last part of the code plots the space – time, velocity – time, and acceleration – time diagrams. An example is given in chapter 4.7. Additionally, an animation of the simulation can be made. The animation is explained in more detail also in chapter 4.7. Both outputs can be suppressed in the “Start” function.

3.7 Sensitivity Analysis

To evaluate which range of the parameters are practical to use in the simulation, a sensitivity analysis of parameters was done. For this, all parameters were fixed and only one was changed. In Table 3 the defined standard values are shown for the fixed parameters.

Table 3 Used standard parameters for simulation

Parameter	Value	Description
b_{max} [m/s^2]	3	Maximum breaking acceleration
a_{max} [m/s^2]	1.5	Maximum acceleration
b_{des} [m/s^2]	1	Desired breaking acceleration
v_{max} [m/s]	14	Maximum speed
T [s]	1.2	Desired headway
Arrival rate [veh/h]	1000, 3000, 5000, 7000	Total arrival rate for all directions
S_0 [m]	5	Safety distance between cars
$d_{constraint}$ [m]	50	Pairing distance for the cars
v_{des} [m/s]	11	Desired speed
w_{acc} [-]	10	Weight of the acceleration term in the objective function
w_{des} [-]	10	Weight of the desired speed term in the objective function
w_{IDM} [-]	10	Weight of the IDM distance term in the objective function
$w_{No\text{-}IDM}$ [-]	10	Weight of No-IDM distances term in the objective function

In Table 4 the parameters and the analysed values for the sensitivity analysis are shown. The simulation was performed for four different arrival rates to see the influence of the arrival rate.

Table 4 Value of parameters for sensitivity Analysis

Parameter	Analysed values
Arrival rate [veh/h]	[1’000 2’000 … 10’000]
S_0 [m]	[0 1 5 10 20 100]
$d_{constraint}$ [m]	[0 25 50 75 100 200]
v_{des} [m/s]	[5 7 9 11 13]
w_{acc} [-]	[0 1 10 100]
w_{des} [-]	[0 1 10 100]
w_{IDM} [-]	[0 1 10 100]
$w_{No\text{-}IDM}$ [-]	[0 1 10 100]

3.8 Performance Metrics

As stated above every scenario was executed multiple times. By taking an average and the standard deviation of the five runs, a good estimation of the performance metrics was possible. For this, the 95% confidence interval was plotted. This corresponds to the average value +/- two times the standard deviation assuming normally distributed data. There were four performance metrics observed and calculated. These are explained below.

3.8.1 Vehicle Hours Travelled

The vehicle hours travelled (VHT) measures the sum of the total time each car needs to pass through the system after it has been generated. This means, that if a car is generated but not able to enter the system, the VHT is still increasing. A car is considered to have left the system when his position is smaller than -500. The VHT is calculated by taking the number of all cars in the system N_S and in the queue N_Q at each timestep k of the simulation. The result is then multiplied by the duration of the timestep Δt. The VHT can indicate how efficient cars are served.

$$VHT = \Delta t * \sum N_S(k) + N_Q(k) \tag{11}$$

3.8.2 Queue Time

The queue time is part of the VHT. It is the sum of the time each car needs to wait in the queue before entering the system. Therefore, the queue time increases every timestep if cars are waiting at the entrance of the system. A high queue time means, that the system (lanes) is at its full capacity and thus prevents vehicles from entering the system.

$$Queue\ Time = \Delta t * \sum N_Q(k) \tag{12}$$

3.8.3 Average Speed

For this simulation, the average speed is only calculated for vehicles that are in the system. This means vehicles in the queue are neglected. If the observed average speed is high, it can be assumed that there are no congestions in the system. For the calculation, all values in the matrix V at the end of the simulation are summed up and divided by the number of entries with a number N. Entries without a number are called “Not a Number” (NaN) values. The MATLAB Function “nanmean” performs this calculation by ignoring all NaN values.

$$v_{\text{average}} = \frac{\sum V}{N} = nanmean(V) \tag{13}$$

3.8.4 Total Time Spent

The total time spent (TTS) defines the time passed from the start of the simulation until the moment where all vehicles have passed through the system. In this simulation, it was the moment when the last car has a position which is smaller than -500. The TTS shows how fast all generated cars can be served. The TTS is defined as the last row of the matrix X where an entry is still a number and not a NaN entry.

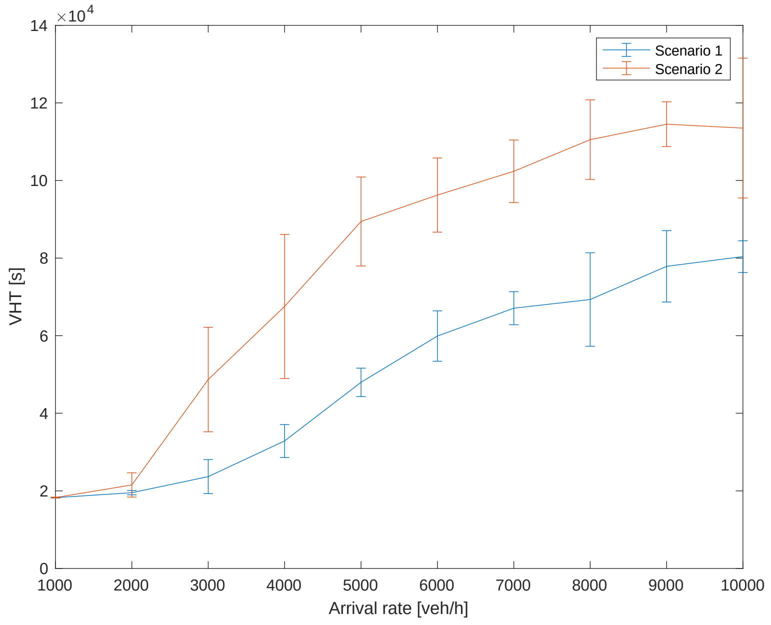
4 Results

The results are dived into different sections. In the first part, the comparison between the scenarios with different demand split is presented (4.1). After that, the results of the sensitivity analysis are stated. Each part is about one studied parameter of the analysis. The plots of all performance metrics can be found in the appendix A 1.

4.1 Comparison between different Scenarios

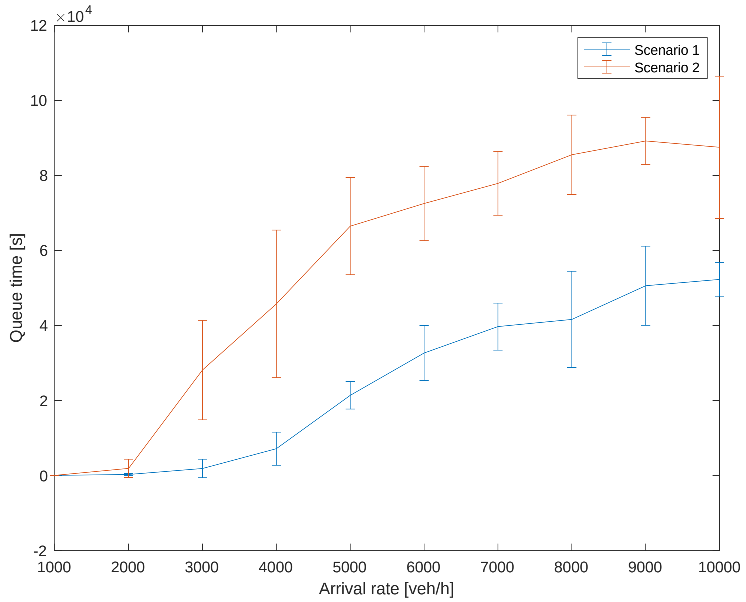
When comparing scenario 1 and 2 (Table 2) it can be stated, that the relative difference between the two results can be neglected. The trends when changing parameters are similar.

Figure 6 VHT versus Arrival rate



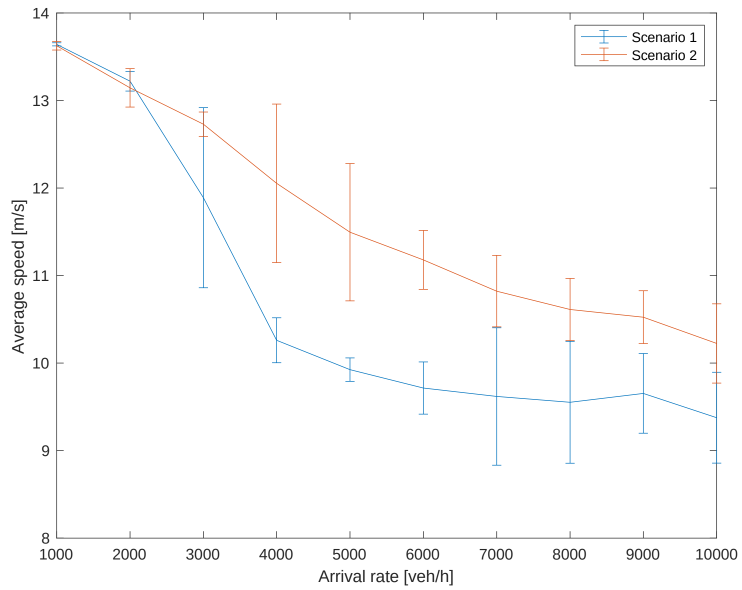
If one looks at the absolute values in Figure 6, it can be seen, that scenario 2 has a higher VHT than scenario 1. Additionally, the queue time (Figure 7) shows, that the main difference of the VHT results from a difference in queue time. For low arrival rates, the queue time is identical. But as the arrival rate increase, the queue time for scenario 2 increases much faster than for scenario 1.

Figure 7 Queue time versus Arrival rate



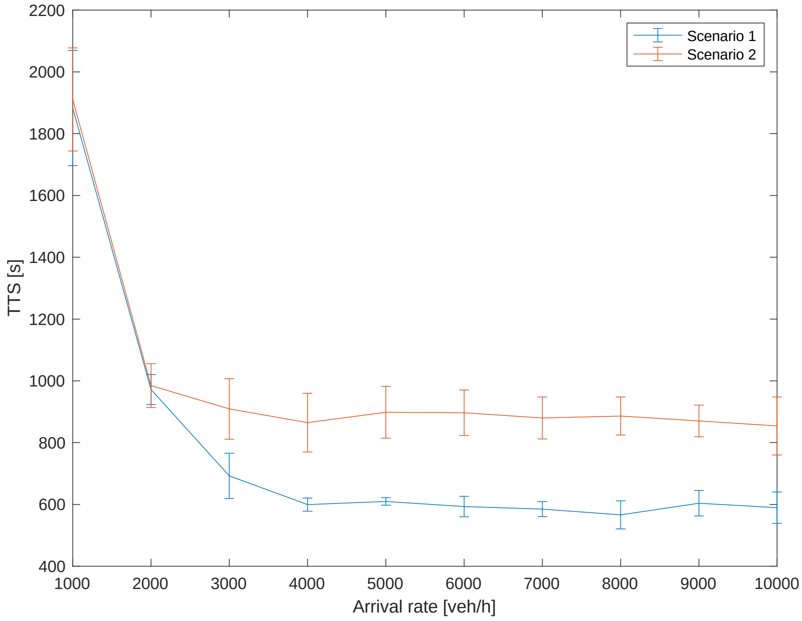
In Figure 8, the average speed versus the arrival rates is plotted for scenario 1 and 2. For low arrival rates, the velocity is the same. After reaching an arrival rate of 3000 vehicles per hour, the average speed for scenario 1 is lower than for scenario 2. The contradiction in the results with different demand scenarios is explained in chapter 5.1.

Figure 8 Average speed versus Arrival rate



The TTS shows similar behaviour for the two scenarios as the average speed. They are identical until an arrival rate of 3000 vehicles per hour is reached. From there, scenario 1 shows a lower TTS. The TTS is insensitive for middle and high arrival rates.

Figure 9 TTS versus Arrival rate



For the following observations, only plots of scenario 1 are considered, as both scenarios show similar behaviour and the standard deviation of scenario 1 is lower.

4.2 Arrival Rate

The VHT increases with higher arrival rates (Figure 6). The queue time shows a similar behaviour as Figure 7 indicates. The standard deviation increases for higher arrival rates too, leading to a wider confidence interval. With high arrival rates there appears to be a saturation.

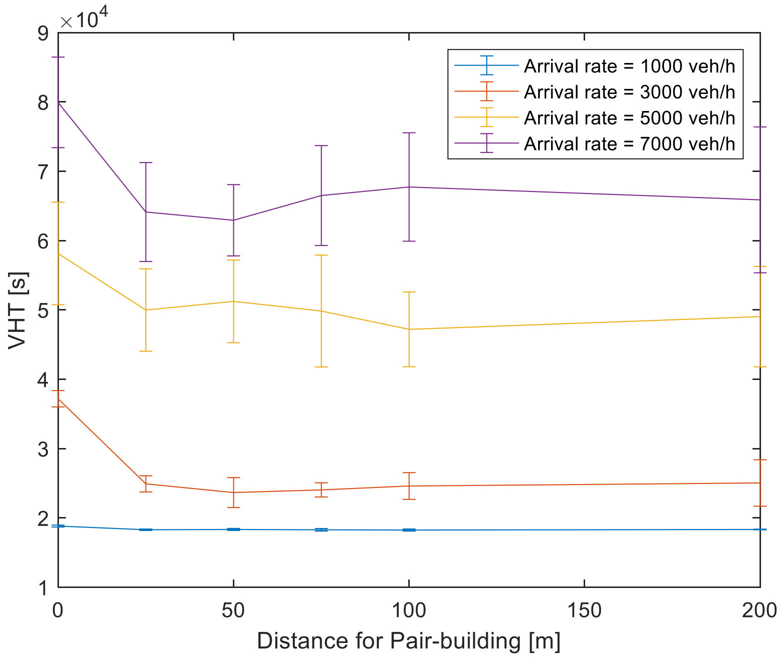
As shown in Figure 8, the average speed shows a fast drop with increasing arrival rates. However, when reaching a rate of 4000 vehicles per hour, the decrease is moderate. Again, a high standard deviation can be observed for high arrival rates. Surprisingly, for an arrival rate of 3000 vehicles per hour the deviation is the highest.

The TTS undergoes an even faster drop with increased arrival rates (Figure 9). At a rate of 4000 vehicles per hour, the TTS remains steady. The standard deviation is generally lower for the same arrival rates as in the average speed plot.

4.3 Distance for Pair-Building

If the distance where cars can search a partner for crossing the intersection together is zero, the VHT is the highest for all arrival rates (Figure 10). For longer distances the VHT is insensitive. There is some variation of the average value for higher arrival rates.

Figure 10 VHT versus Distance for Pair-building



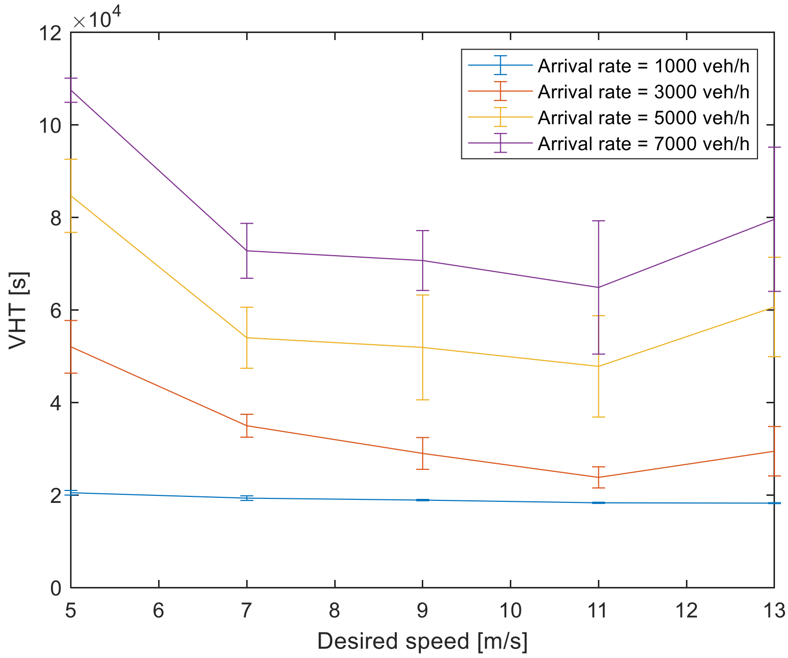
The queue time also shows a drop for a pairing distance larger than zero, except for the arrival rate of 5000 vehicles per hour, which shows a slight increase in queue time. However, the drop is, in general, more moderate.

For middle and high arrival rates the increase in the average speed with a distance for pair-building going from 0 to 25, is high. For low arrival rates, this increase is smaller. After that, the average speed is insensitive to an increased pairing distance.

The TTS is overall insensitive to a change of the distance for pair-building. It can be observed, that it decreases slightly with an increase in the distance.

4.4 Desired Speed

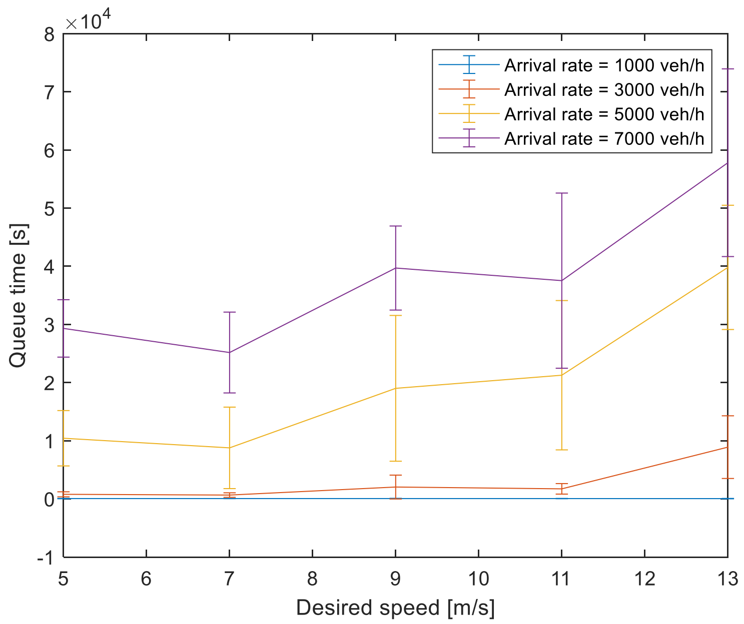
Figure 11 VHT versus Desired speed



The VHT for different desired speeds shows a minimum at 11 m/s for all arrival rates (Figure 11). If the desired speed is increased, the VHT increases too.

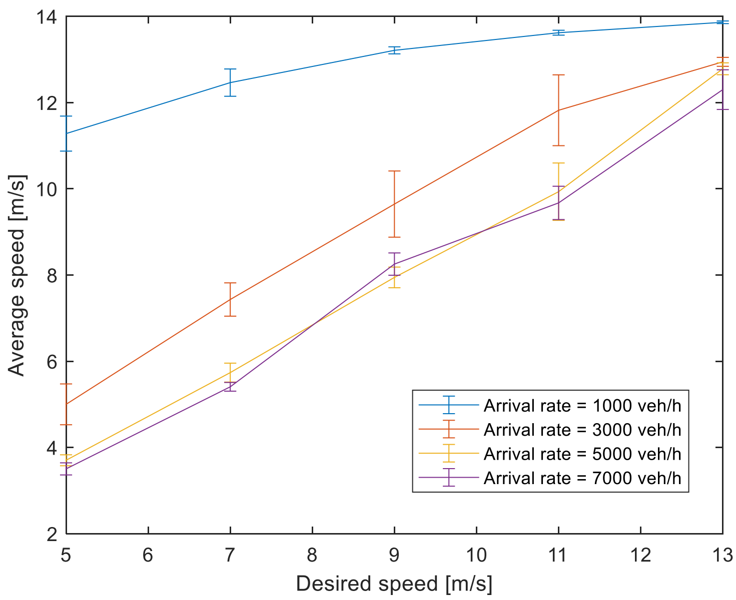
Figure 12 shows that the queue time generally increases with a faster desired velocity. For high arrival rates, one can observe a drop between 5 m/s and 7 m/s. For an arrival rate of 7000 vehicles per hour, there's also a decrease between 9 m/s and 11 m/s. Nevertheless, it must be added, that the standard deviation is high.

Figure 12 Queue time versus Desired speed



The average speed increases with a faster desired speed (Figure 13). For an arrival rate of 3000 cars per second, the average speed and the desired speed are congruent, except for the highest value of 13 m/s. The average speed for lower arrival rates is higher, whereas the velocity for higher arrival rates are lower and remains insensitive to the arrival rate.

Figure 13 Average speed versus Desired speed



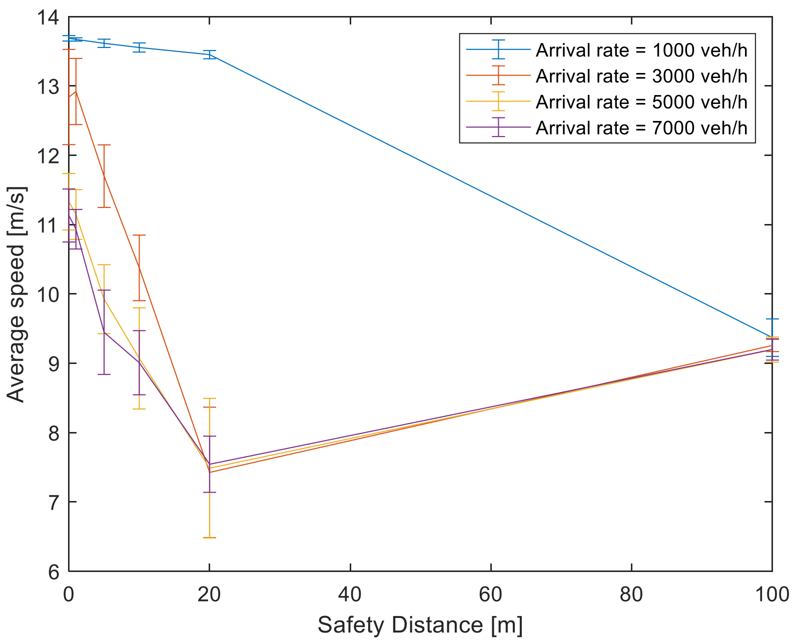
For low and middle arrival rates, the TTS decreases slightly with faster desired velocities. For arrival rates over 3000 vehicles per hour, the TTS drops until a desires speed of 7 m/s and then remains constant until a speed of 11 m/s is reached. Finally, the TTS increases again for the desired velocity of 13 m/s. The TTS for this speed is almost equal for middle and high arrival rates.

4.5 Safety Distance

The behaviour of the VHT for middle and high arrival rates is relatively similar, whereas the absolute value is the highest for the highest arrival rate. For a low arrival rate, the VHT is constant until a rise for long safety distances is reached. The standard deviation is low. The queue time shows the same behaviour.

For a low arrival rate, the average speed drops moderately with an increase in the safety distance. This decrease is much steeper for middle and high arrival rates. With a safety distance of 100 meters, the average speed converges to 9 m/s for all arrival rates as shown in Figure 14. For this safety distance, the standard deviation is also low.

Figure 14 Average speed versus Safety distance



The TTS is insensitive for a low arrival rate until safety distance increases to 100 meters. For the other arrival rates, the TTS increases with a longer safety distance but is overall smaller than for a low arrival rate. With a safety distance of 100 meters, the TTS is converging to a value of 2800 seconds for all arrival rates.

4.6 Weights of the Objective Function

As mentioned in chapter 3.7 the sensitivity analysis was performed for the different parts of the objective function. The weight of zero was introduced, to get an understanding, whether the code works correctly. A choice of zero does not make sense since all parts of the objective functions needed to be optimised. In the following, the outcome for the discrete weights is described. Unfortunately, only a few values could be examined, and a more in-depth study should be performed in future research (Chapter 6).

4.6.1 Weight of Acceleration

The VHT has a minimum at a weighting of 10 for the acceleration term. It increases very much with a weight of 100. The queue time shows the same behaviour for all arrival rates.

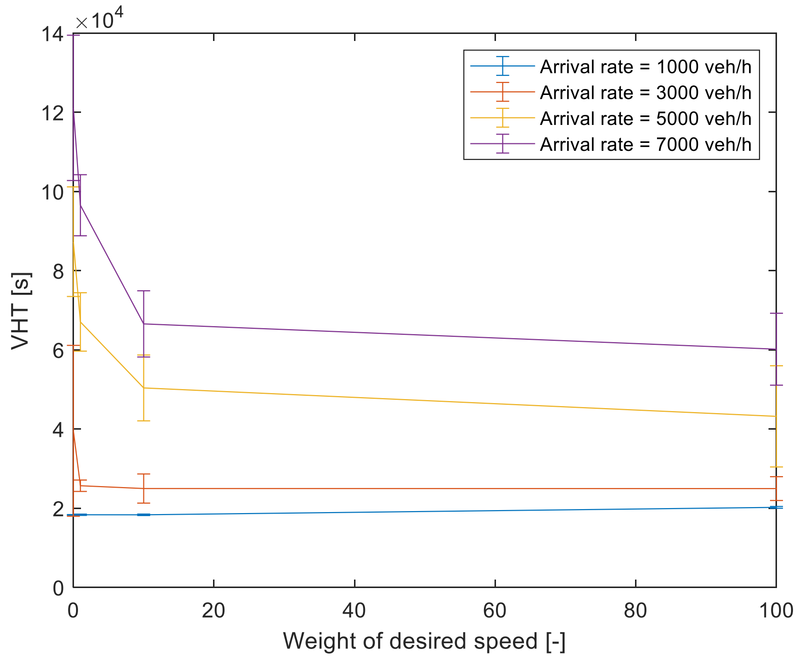
The maximum of the average speed is reached with the highest weight, except for the arrival rate of 3000 vehicles per hour. For this rate, the average speed with a weight of 10 is slightly higher, but at the same time has a high scatter. For low arrival rates, the speed is insensitive to different weights of the acceleration.

The TTS generally shows an insensitive behaviour. It has a minimum at the highest weight for low arrival rates, while the maximum is found for middle and high arrival rates. Overall the change of the TTS is small.

4.6.2 Weight of Desired Speed

For the sensitivity analysis of the weight of the desired speed, a clear decrease can be observed for middle to high arrival rates with an increased weighting (Figure 15). The decrease is faster when increasing the weight from 1 to 10, while only a slight decrease can be noted for the change from 10 to 100. For a low arrival rate, one observes a little increase with a larger weight. Again, the queue time shows the same picture.

Figure 15 VHT versus Weight of Desired speed

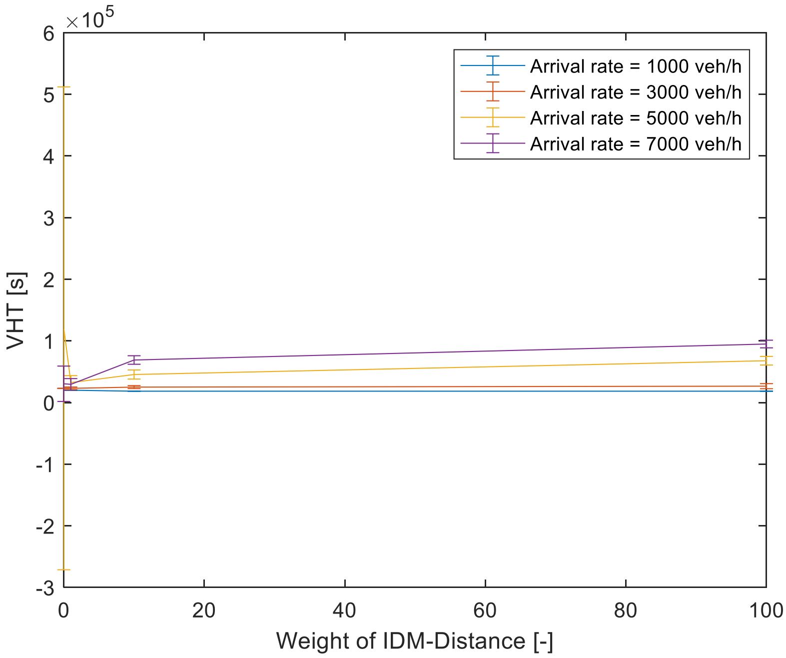


The average speed has a minimum with at a weight of 10 for middle and high arrival rates. It converges to a value of 11 m/s for the highest weight. For a low arrival rate, a constant decrease can be observed.

The same can be said for VHT, queue time, and average speed: between middle to high arrival rates and low arrival rates a distinction has to be made for the TTS. For low rates, the TTS is found to be constant whereas for the other arrival rates a decrease can be noted until the weight of 10. Beyond this weight of the desired speed, the TTS remains constant. A high deviation is observed for the weight of zero.

4.6.3 Weight of IDM-Distance

Figure 16 VHT versus Weight of IDM-Distance



In Figure 16 high scatter can be observed for a weight of the IDM-distance part of the objective function of zero. Moreover, for all arrival rates, an increase of the VHT is found with a heavier weight, as this is also the case for the queue time.

The average speed shows the same behaviour with high arrival rates with a minimum value related to the weight of 10. For low and middle arrival rates this minimum occurs at a weighting of 1.

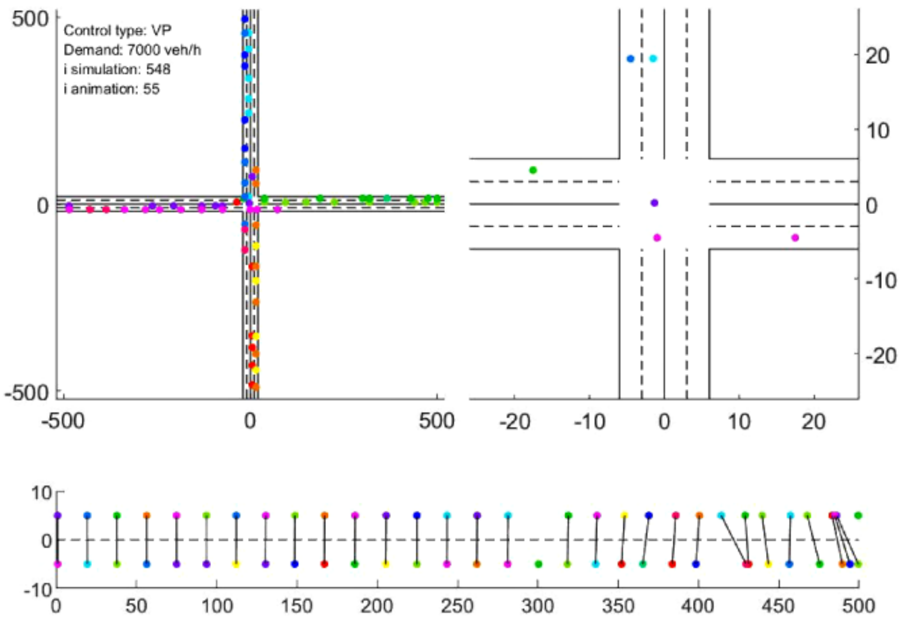
The TTS is insensitive to a change of the weights of the IDM-distance, although a minimum can be observed for a weight of 1.

4.6.4 Weight of No-IDM-Distance

Since the weight of the No-IDM-distance term is only used for the partner of a car it results in an insensitive behaviour of all performance metrics to different weight values.

4.7 Animation and Illustrations

Figure 17 Snapshot of Animation

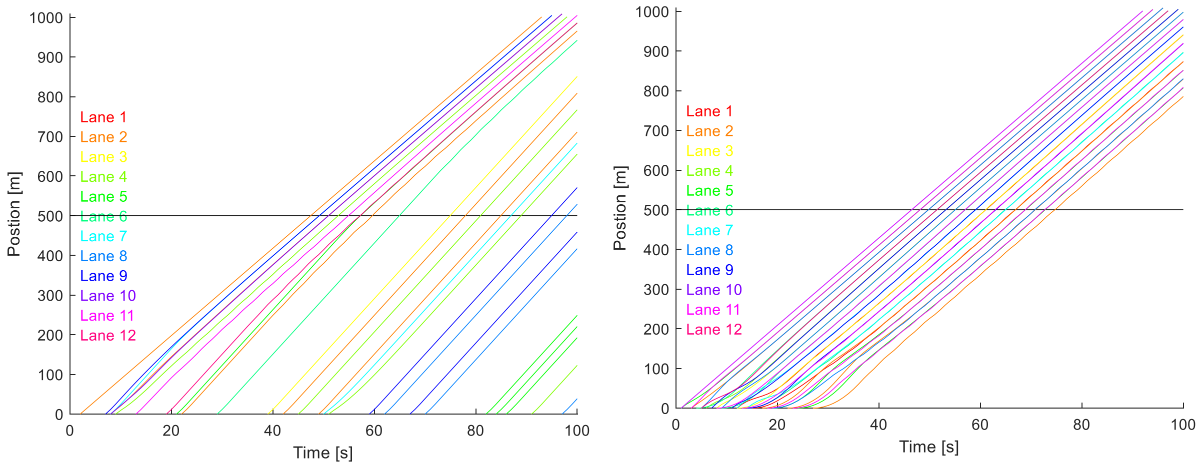


To increase the understanding of the results of the simulation, an animation was implemented. A snapshot of the animation can be found in Figure 17. In the top left picture, the whole system is shown. The simulation timestep is ten times refined in the animation. The current time steps as well as the arrival rate can be seen. The coloured points represent the vehicles on their lane. Their colour is according to their desired direction. Next to it, an enlargement of the intersection is visible. In this way, a better observation of the pairing and crossing of the cars over the intersection is possible. At the bottom, the virtual platoon is plotted. If two vehicles are paired, they are connected via a black line.

Besides the animation, the MATLAB code generates a space-time, a velocity-time, and an acceleration-time diagram. The colour of the trajectory corresponds to the lane on which the vehicle was generated. For the discussion, the illustrations of two different arrival rates are shown in Figure 18 – Figure 20. The arrival rate for the right figure was 1000 vehicles per hour and for the figure on the left, it was 7000 vehicles per hour. The number of vehicles generated was limited to 30.

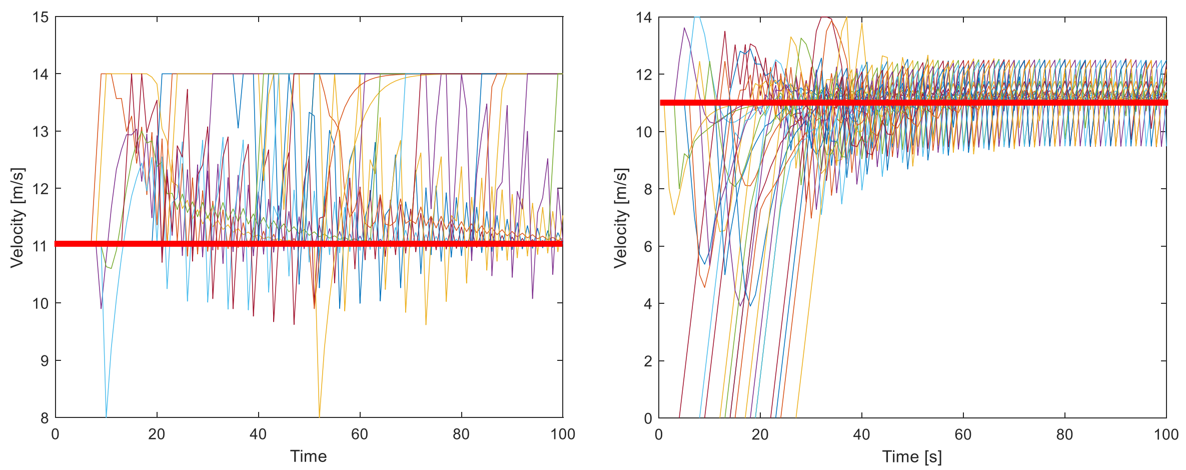
Figure 18 shows that with a low arrival rate mainly straight lines are visible. On the right side, one can observe that a lot of vehicles switch their places. The trajectories are curved in the first few time steps and closer to each other. After some time, the form straight parallel lines.

Figure 18 Space – Time diagram: 1000 veh/h (left) vs. 7000 veh/h (right)



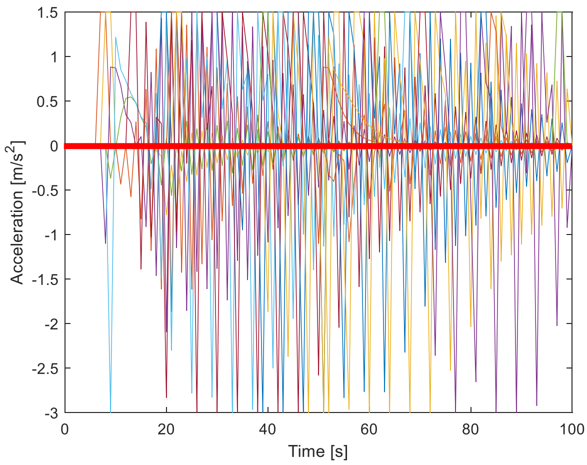
In Figure 19 one can see that for low arrival rates the observed speeds are always above 8 m/s. Although there is some oscillation the velocity of the cars converges fast to the desires speed of 11 m/s. With a high arrival rate, the conversion takes longer. Some vehicles enter the system with a speed of 0 m/s.

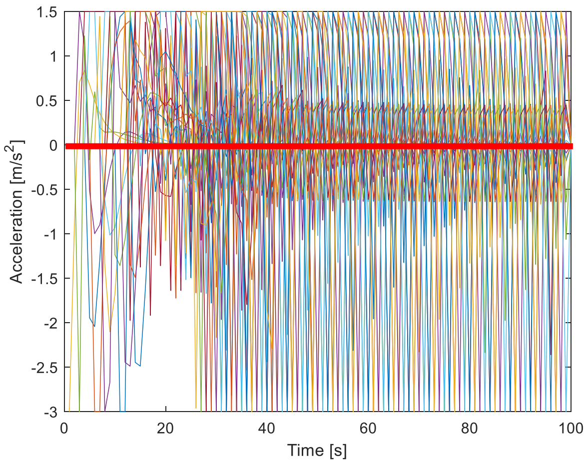
Figure 19 Velocity – Time diagram: 1000 veh/h (left) vs. 7000 veh/h (right)



The acceleration – time diagram in Figure 20 shows for both arrival rates high oscillation of the acceleration. It jumps from the lower boundary ($b_{\max}$) to the upper boundary ($a_{\max}$). The cars change from accelerating to breaking and only slowly converge to a steady velocity. The reasons for these results and an improvement of the model are explained in chapter 5.7 and 6.2.

Figure 20 Acceleration – Time diagram: 1000 veh/h (left) vs. 7000 veh/h (right)





5 Discussion

5.1 Comparison between different Scenarios

When comparing scenario 1 and 2 it is essential to keep in mind, that the demand split in scenario 2 is much more uneven (Table 2). There are a lot of vehicles generated on lane 5 and 11. As mentioned before, the VHT is strongly related to the queue time. The reason for the higher VHT and queue time is the high number of vehicles generated on lane 5 and 11. On these lanes, a lot of vehicles are put into the queue, due to congestion. Since the lane is shared with direction 6 and 12 (Figure 1), it additionally affects those cars. The capacity of the system is the same, but due to the uneven distributed arriving traffic, the efficiency is worse than in scenario 1.

The average speed is higher for scenario 2. This seems contradictory to the previous statement. But since queuing vehicles are neglected for calculating the average speed, the result can clearly be explained. The TTS shows that in scenario 2 it takes longer to serve all vehicles. Due to the large queue on lane 5 and 11 and the limited capacity on these lanes, it takes a lot of time until all vehicles can be put into the system. Once they are in the system their speed can be higher towards the end of the simulation since the vehicles from other directions have already left the system. Therefore, the average speed is higher compared to scenario 1.

5.2 Arrival Rate

The increase of VHT with higher arrival rates can be explained by taking the queue time into account. If the vehicles are generated faster, the lanes will be occupied more often. Therefore, when the lanes are occupied additional cars cannot enter the system and are put in the queue. With higher arrival rates the number of vehicles in the queue grows faster and the VHT increases. The larger scatter with higher arrival rates can be explained by the importance of the distribution of the vehicles onto the lane. If the distribution is one-sided (a lot of cars generated in one lane), no pairs can be built towards the end of the simulation. If the vehicles are distributed more evenly, the probability of finding a partner is high until the end of the simulation, resulting in a lower VHT and queue time

With low arrival rates, the vehicles can pass the system with the maximum speed as there are no conflicts and the distance between the vehicles is large enough. With an increase in the arrival rate, more cars will enter the system when the lane is congested. Therefore, their speed is zero and the cars need to accelerate. Because the cars are queued at the entrance, the arrival rate does not have a big impact on the average speed anymore.

In the case of the TTS with low arrival rates, it takes a lot of time to put all vehicles through the system. The time until all cars are generated is longer for low arrival rates. Similar to the average speed, saturation can be observed, where the TTS isn't decreasing anymore. If vehicles are generated at a higher rate only the queue time increases because the capacity of the system is already reached.

5.3 Distance for Pair-Building

If the distance where a car looks for a partner is zero, no pairing will happen, and the capacity of the intersection cannot be increased. Therefore, such a choice is not desired. Besides the value of zero, the response of the simulation is insensitive to different pairing distances. A choice between 25 and 100 meters is however optimal. If the pairing distance is too high, the vehicles will not be able to catch up to their leader. This leads to an unsafe crossing of the intersection as the partnered cars do not cross the intersection simultaneously.

5.4 Desired Speed

The sensitivity analysis shows that with a value of 11 m/s the VHT can be minimised. Even though the queue time increases a higher desired speed ensures a fast passing through the system. The reason for the increased queue time lies in the required IDM-distance. If the speed of a vehicle is higher, the IDM-distance (Equation 3) is increased to have a higher safety level. Therefore, more vehicles have to enter the queue before entering the system. Although if the desired velocity is too close to the maximum velocity, the vehicles aren't able to catch up to their leader on the virtual platoon. For a good choice, the desired speed should be as high as possible, but the difference to the maximum speed should be high enough to allow the vehicles to catch up to the leader in the virtual platoon.

5.5 Safety Distance

The safety distance mainly defines the spacing of cars on the virtual platoon and in the lane. It is obvious that with an increase in distance, the number of vehicles served per time will drop and the VHT will increase. Because a high capacity is beneficial, the safety distance should be as low as possible. But there is a lower limit. On one hand, the accuracy of the system is maybe not perfect. To deal with a certain deviation the safety distance ensures a safe crossing over the intersection and safe distance between cars in the same lane. On the other hand, user comfort is important. Especially, when introducing a new system people tend to mistrust the system. To make the passengers feel safe in congested conditions the spacing between two vehicles should be at least one meter. If the speed is increasing the spacing is increased too according to the IDM-distance (Chapter 3.4.2).

Some quite interesting observations can be made on the results of a high safety distance. The reason for the choice of this value was to validate the simulation model. If the safety distance is higher than the pairing distance no pairs will be formed. All cars will pass the intersection alone. Therefore, the TTS and average speed are the same for all arrival rates. With this validation, it could be shown that the simulation works correctly as the expected results were obtained.

5.6 Weights of the Objective Function

The choice of accurate weights of the objective function is crucial but at the same time complex. With some values, the system does not behave correctly. For example, if the acceleration or the difference to the desired speed is too important, the IDM-distance cannot be kept. To find a good combination of all terms in the objective function one has to do a try and error approach. While varying the relative weight of the different parts and observing the outcome, a good functioning combination can be found. The simulation shows that with an equal weighting good results can be achieved. The weight of the desired speed may even be higher in order to minimise the VHT. To make a well-founded statement though, a wider and deeper analysis would be necessary.

5.7 Animation and Illustrations

The idea of the animation was to get an understanding of the results from the simulation. Due to its arrangement, a collision can be identified quickly, and one can identify whether the pairing is working.

The illustrations confirm the observations made in chapter 5.2. A High arrival rate leads to an adjustment of the velocity to reach the space on the virtual platoon. Once the position is reached the pairs can pass the system and the intersection with the desired velocity. In this way the capacity of the system can be reached. With low arrival rates, almost no pairing can take place, since the vehicles are too far apart from each other.

The velocity–time diagram (Figure 19) shows a fast conversion to the desired speed for low arrival rates as cars don't need to adjust the speed in order to arrange themselves on the virtual platoon. With high arrival rates, the lanes are sometimes in a congested condition leading to an entering speed of the following vehicles of 0 m/s.

Due to the objective function high oscillations are observed in Figure 20. This leads to poor user comfort as the cars frequently change from acceleration to braking. This result is not realistic. With the use of model predictive control this issue may be solved (Chapter 6.2).

5.8 Overview of optimal Parameters

The most optimal choices of the values for the different parameters as explained in chapters 5.3 to 5.6 are summarized in Table 5 below. For most of the cases, these values are a good choice for acceptable results. For some cases with extremely low or high demand or other specific conditions, some adjustments to the parameters may be necessary.

Table 5 Overview of optimal Parameter choices

Parameter	Optimal choice
S_0 [m]	1 – 5
$d_{constraint}$ [m]	25 – 100
v_{des} [m/s]	11
w_{acc} [-]	10
w_{des} [-]	10
w_{IDM} [-]	10
$w_{No\text{-}IDM}$ [-]	10 – 100

6 Future Research

In this chapter, some suggestions are made for future research. The ultimate goal is to make the simulation more realistic and improve the solution approach. With a more realistic problem formulation and a better solution method, the simulation could one day be used for traffic control of a lightless intersection.

6.1 Improvement of the Simulation Setup

In the first part, some possible improvements to the simulation setup are explained. For this simulation, all vehicles are modelled as a point. In reality, this is not true. Vehicles have different dimensions. By implementing the real geometry or a close approximation (e.g. a square) of the vehicles, the size of a vehicle can have an impact on the results. Larger vehicles like trucks or buses take a larger space which needs to be accounted for when calculating the IDM-distance. This distinction leads to a second possible improvement. Usually larger cars have a different driving dynamic. They are not able to accelerate and brake as fast as smaller vehicles. Additionally, the restrictions concerning user comfort are stricter in the case of a bus. Those refinements will lead to a change in the results. This change though, is expected to be not as large as it mainly affects the spacing of the vehicles. The impact is mainly dependent on the share of large vehicles.

A larger impact on the results can be expected when adding the dynamic of the vehicles when they approach and cross the intersection. Normally a car will reduce its velocity when approaching an intersection especially when it needs to change direction. Therefore, the speed with which vehicles perform a right or left turn will be much slower than their normal speed. This was neglected in the simulation and a constant speed was assumed. An implementation of the speed reduction will lead to a lower capacity of the system. There may be some shockwaves caused by turning vehicles. This can lead to congestion on the lanes and therefore a higher VHT. The pairing of turning vehicles with vehicles going straight will be governed by the dynamic of the turning vehicle.

In this simulation, the pairing was only done with two cars. This could be extended to a maximum of six cars. If four vehicles from all four directions intending to turn right and additionally two vehicles with a compatible movement intending to turn left approach the intersection, a crossing at the same time is possible. This step can easily be implemented and will result in a minor improvement in capacity. The increase of capacity will primarily be determined by the demand split.

A big simplification made, was the assumption that all cars are already on the lane according to their direction. Two things can be detailed to get a more realistic simulation. In the first step, lane changes can be allowed. This will lead to a much more complex system where the optimal time for the change of the lane must be found as well as the dynamic of the lane changes must be implemented. However, this scenario matches reality better as cars are not automatically on the lane to their desired direction. The second step is the definition of the length of the approaching leg. Due to the limited space in urban areas, the directional lane split on a leg happens only shortly before the intersection. A two-lane approaching leg of 500 meters length is unusual. Therefore, the question should be asked if a pairing is possible when the approaching length is short. Vehicles will not be able to overtake each other to reach the assigned position on the virtual platoon. The distance for pair-building must be shortened to ensure that a catch up is possible.

Another limitation is the observation of an isolated intersection. In a traffic network, the intersections form a complex system. They influence each other. It is a huge task to optimise the crossing of vehicles at a network level and implement the interaction between the different intersections.

To sum up there are several steps on how to improve the simulation and make it more realistic. Some implementations can be done easily, whereas others are much more complex. In the end, one can always find a higher level of detail. A compromise between the simplification and still staying as close to reality as possible with the modelling must be found.

6.2 Model Predictive Control

A second approach to achieving better results is to use a different solution approach. One approach can be found in the paper of Tachet, et al., 2016. The concept of this solution is based on time slots assigned to each vehicle to cross the intersection. In this approach, pairing is not implemented.

Another interesting approach using pairing and a virtual platoon is the use of model predictive control (MPC). This has been done by Makarem & Gillet, 2013. MPC can be used to take future states of the system into account when deciding on the value of the decision variable. A famous example is an autonomous car approaching a curve on a street. In this example, the decision variables could be the change of the angle of the steering wheel or the change in acceleration of the car. If the car model is only applied for the next time step, when the car reaches the curve, it will not be able to reduce the velocity and change the steering angle in order to stay in its lane. This can lead to collisions with cars on the opposite lane. With the use of MPC the simulation can calculate the position and velocity of the car for the next few timesteps. A penalty term is introduced stating the deviation from the middle of the lane. The controller now chooses an adequate acceleration to reduce the value of the penalty term. This calculation is repeated for every time step. For every calculation, the optimal value of the decision variables is chosen to achieve the desired goal. In this way, the car will reduce the velocity when approaching the curve and it can stay in its lane.

This principle can be applied to the problem described in this project. The simulation will calculate the state of the system for the next time steps and then choose an optimal value for the decision variable. In this case, the decision variable is the acceleration of the car. By the use of MPC, one expects a smoother acceleration and breaking due to the preview capability of the MPC. If the leader of a vehicle on the virtual platoon is slowing down, the follower could also reduce his velocity. In this way, the oscillation illustrated in Figure 20 can be stopped. This process already takes place with non-autonomous vehicles. Drivers always try to anticipate the behaviour of their leader by looking ahead.

The number of calculated future states of the system is called the prediction horizon. It has a huge impact on the computation time of the simulation. For every time step, the model must be applied and the calculations for all cars need to be executed. Therefore, it is essential how far the prediction horizon is chosen. If it is chosen too far the calculation will take too long and the output of the result can be too late. On the other hand, if the prediction horizon is not long enough the benefit of the preview capability has no impact on the simulation.

Another issue is the optimal choice of the value for the acceleration. It’s complex to define which value should be chosen out of all the results of the MPC. But with the rights choice of the prediction horizon and decision variable MPC can lead to a big improvement of the simulation.

7 Conclusion

This chapter contains a brief conclusion. It's a summary of the most important findings resulting from the discussion as well as an insight into future research possibilities.

The simulation setup used for this project is closer to reality than previous projects. For sure, an approaching and departing leg with three lanes would be desirable, but due to the limited available space, especially in bigger cities, it is often not possible. Even the intersection layout described in chapter 3.1 requires a lot of space. It's not realistic to have an approaching lane of 500-meter length. Besides that, the simulation model still neglects other issues as well (Chapter 6.1). It is difficult to find a trade-off between the necessary detailing and still simplifying the problem. For better results and a more founded conclusion, the suggestions for an improved simulation setup in chapter 6.1 must be implemented. Nevertheless, this project shows important results and can be used as a basis for future work.

The sensitivity analysis was an important part of this project. It demonstrates the influence of different parameters. Through the analysis, optimal values of the parameters can be found (Table 5). With the help of the modified animation and the illustrations, one can get a good understanding of the results. This improves the validation process of the simulation. Again, a more in-depth study of the parameters with wider sensitivity analysis is possible and could lead to better results and conclusions.

The use of MPC can be a big improvement to the simulation (Chapter 6.2). A major issue was the oscillation of the acceleration of the vehicles (Figure 20). The results obtained by solving the objective function (Chapter 3.4.3) are not practical and far from reality. User comfort is essential for the acceptance of new technologies such as CAVs by the population. MPC may solve this problem. Another attempt could be the introduction of a boundary condition to prevent big changes in the acceleration from one time step to the next.

8 Acknowledgement

First, we would like to thank Dr Anastasios Kouvelas. Through his inputs in the meeting hours during our project, we could benefit from his big knowledge and experience in the field of traffic engineering. His chair at IVT made our project possible in the first place.

Furthermore, we would like to give thanks to Alexander Genser. He supervised our work throughout the whole project. He helped us greatly with his comments and suggestion. He was always at our disposal for technical issues. His comments concerning the presentation and report were extremely helpful and we really enjoyed working with him.

Finally, we also want to thank Kimia Chavoshi Boroujeni. As she supervised the previous projects on this topic we could benefit greatly from her knowledge.

9 Bibliography

Cover picture: Adobe Stock. *Traffic speeds through an intersection at night in Gangnam, Seoul in South Korea,* [Date of Download: 02.06.2020]

Galli, A. (2019). *Virtual platooning for conflict-free crossing of AVs through a lightless intersection.* Master Project, ETH Zurich, IVT, Zurich.

He, Z., Zheng, L., Lu, L., & Guan, W. (2018, June 2). Erasing Lane Changes From Roads: A Design of Future Road Intersections. *IEEE Transactions on Inteligent Vehicles, Vol. 3*(No. 2), 173.

Makarem, L., & Gillet, D. (2013, October 6-9). Model predictive coordination of autonomous vehicles crossing intersections. *Proceedings of the 16th International IEEE Annual Conference on Inteligent Transportation Systems (ITSC 2013).*

Mannering, F. L., & Washburn, S. (n.d.). *Principles of Highway Engineering and Traffic Analysis.*

Meier, J. (2019). *Automated intersection: Optimal conflict-free lightless crossing for autonomous vehicles.* Bachelor Thesis, ETH Zurich, IVT, Zurich.

Tachet, R., Santi, P., Sobolevsky, S., Reyes-Castro, L. I., Frazzoli, E., Helbing, D., & Ratti, C. (2016). Revisiting Street Intersection Using Slot-Based Systems. *PloS ONE.*

Treiber, M., Hennecke, A., & Helbing, D. (2000). Congested Traffic States in Empirical Observations and Microscopic Simulations. *Phys. Rev. E, Vol. 62*(No. 2).

Xu, B., Li, S., Bian, Y., Li, S., Ban, X., Wang, J., & Li, K. (2018). *Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections.* Transportation Research Part C.

Appendix

A 1 Results for Scenario 1

Arrival rates

Figure 21 VHT versus Arrival rate

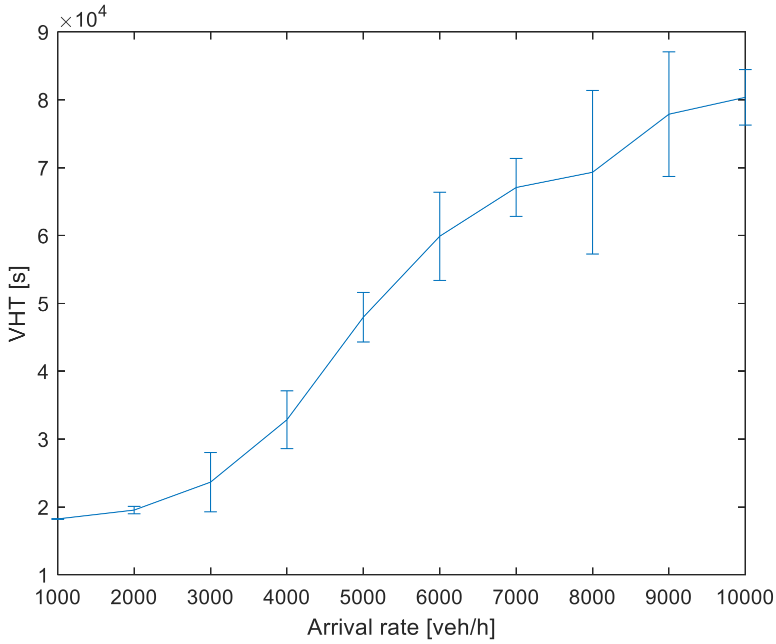


Figure 22 Queue time versus Arrival rate

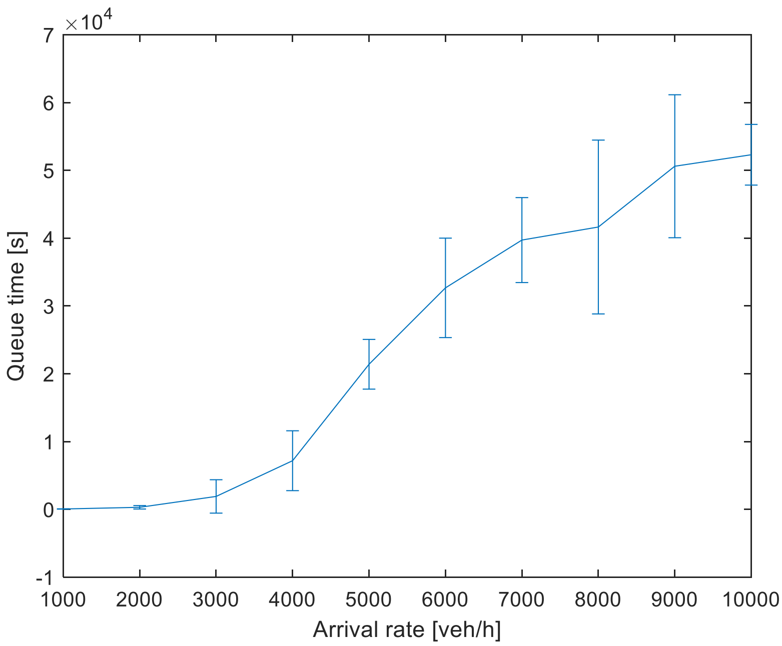


Figure 23 Average speed versus Arrival rate

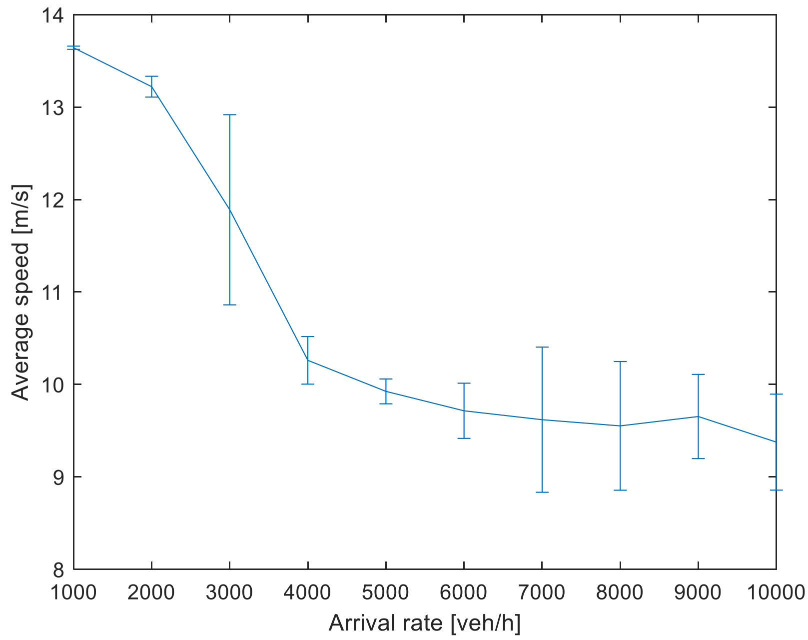
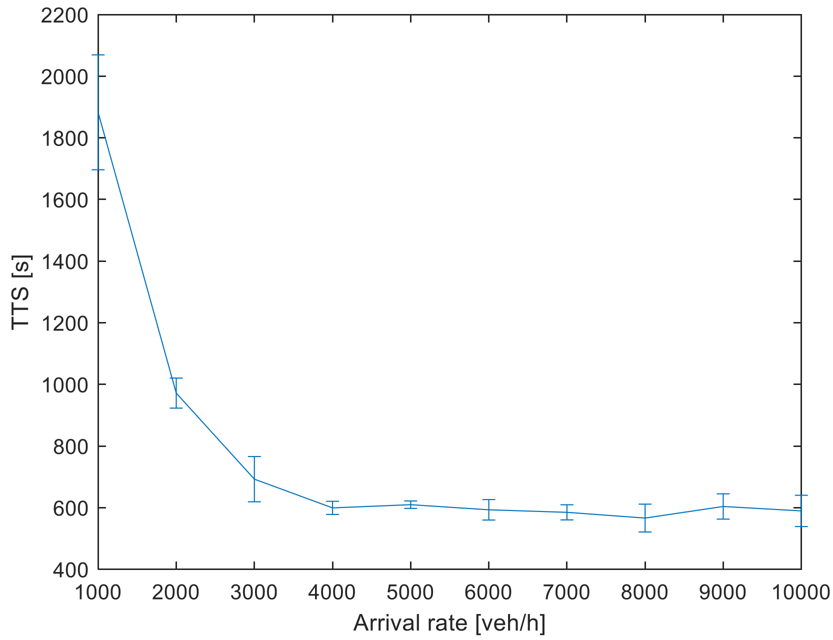


Figure 24 TTS versus Arrival rate



Distance for Pair-building

Figure 25 Queue time versus Distance for Pair-building

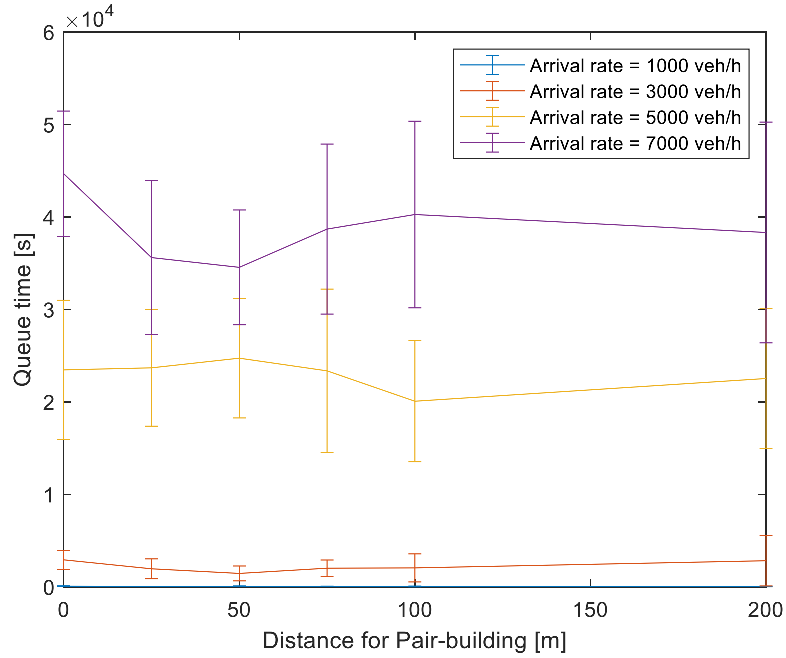


Figure 26 Average speed versus Distance for Pair-building

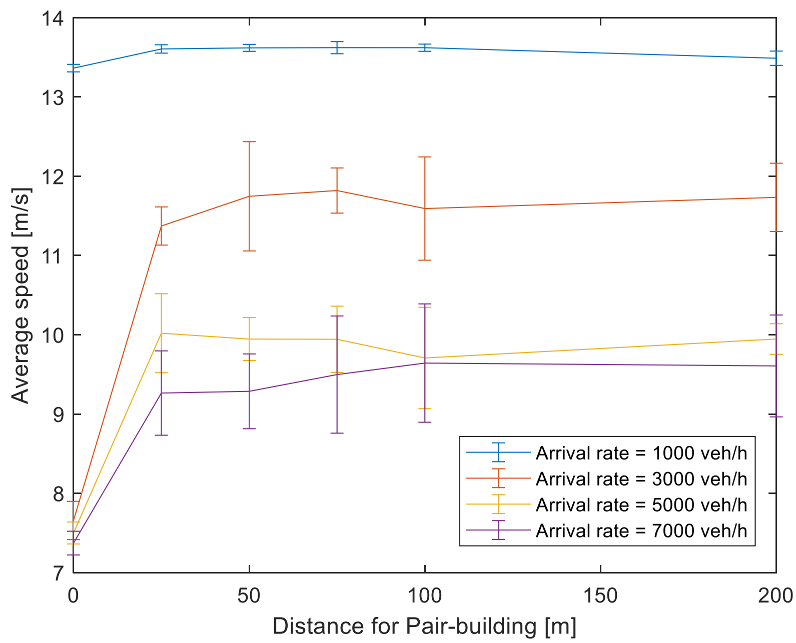
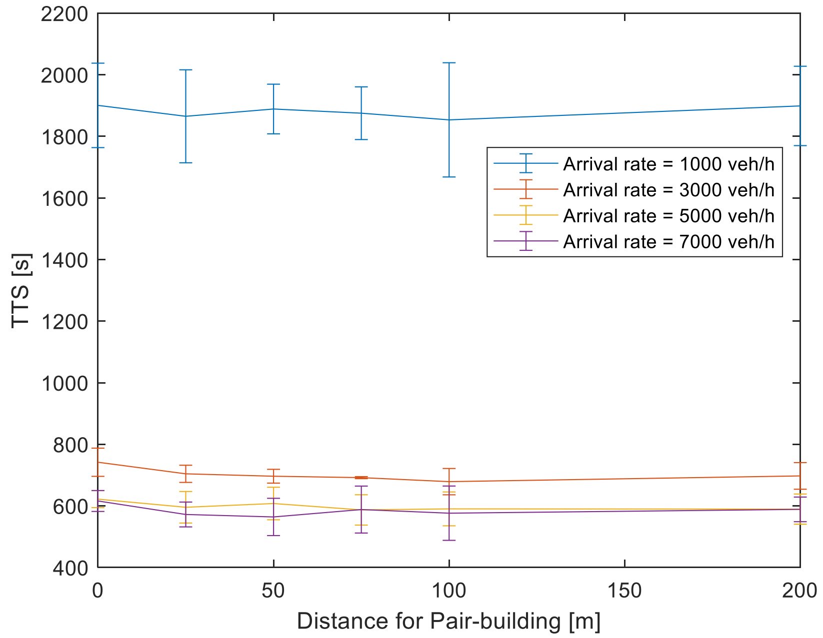
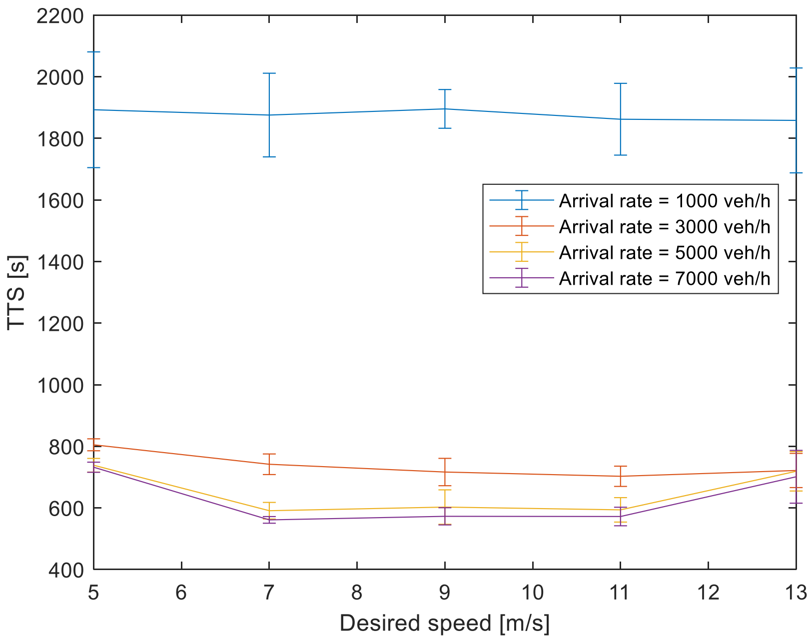


Figure 27 TTS versus Distance for Pair-building



Desired speed

Figure 28 TTS versus Desired speed



Safety distance

Figure 29 VHT versus Safety distance

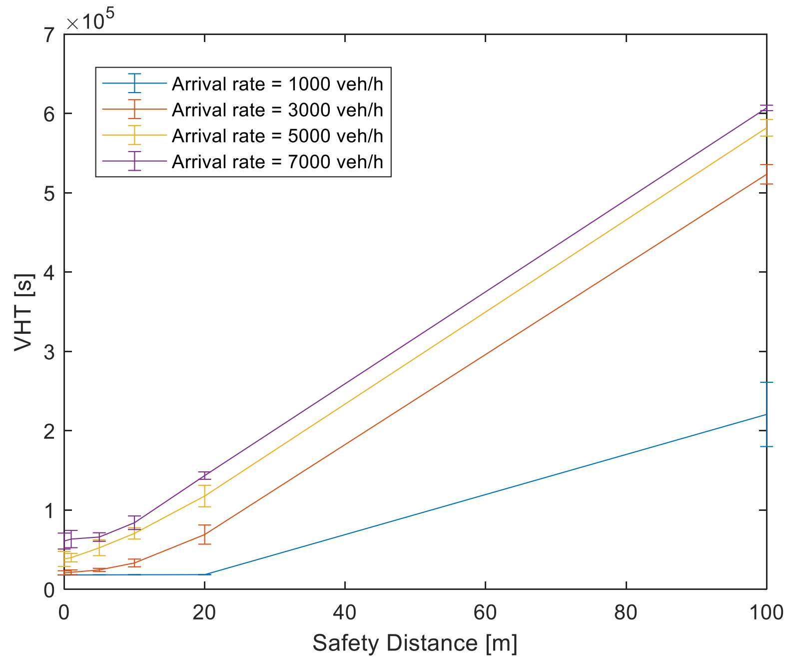


Figure 30 Queue time versus Safety distance

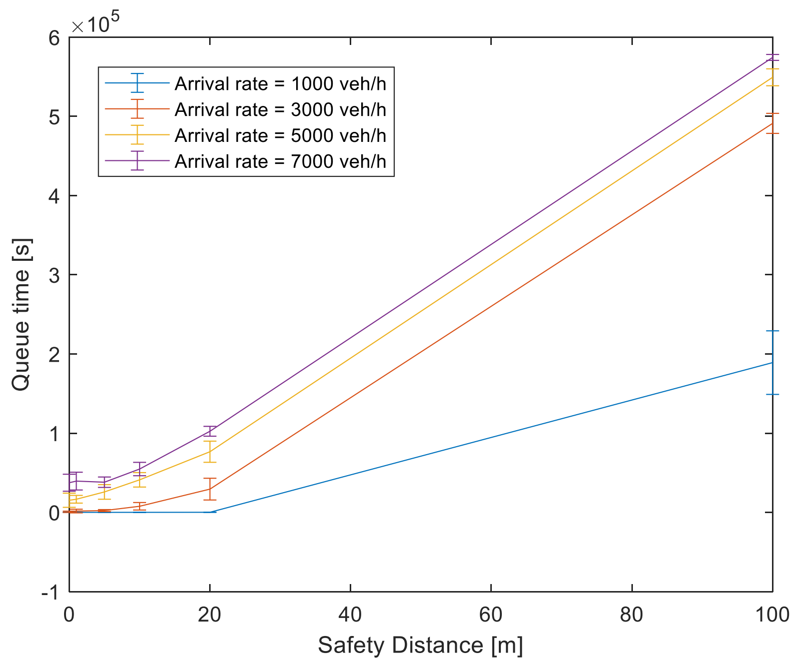
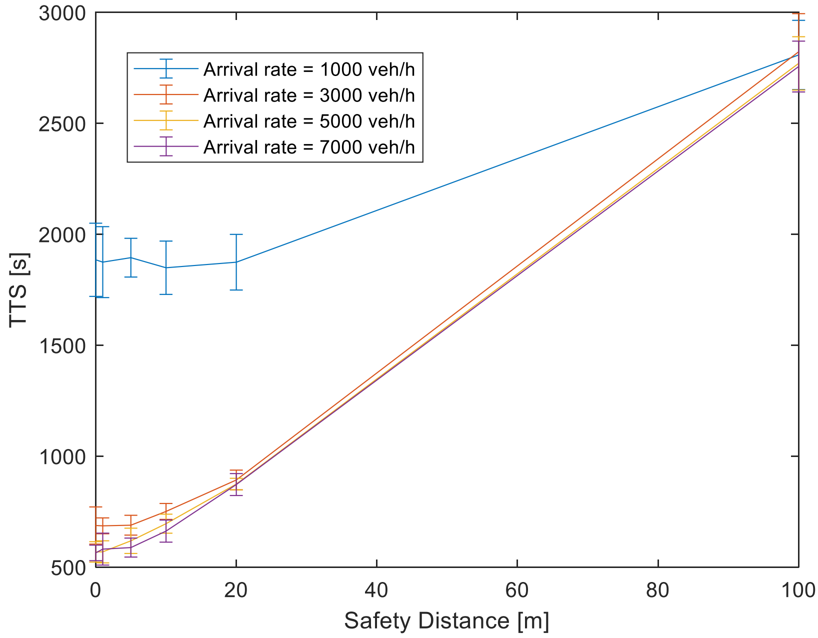


Figure 31 TTS versus Safety distance



Weight of Acceleration

Figure 32 VHT versus Weight of Acceleration

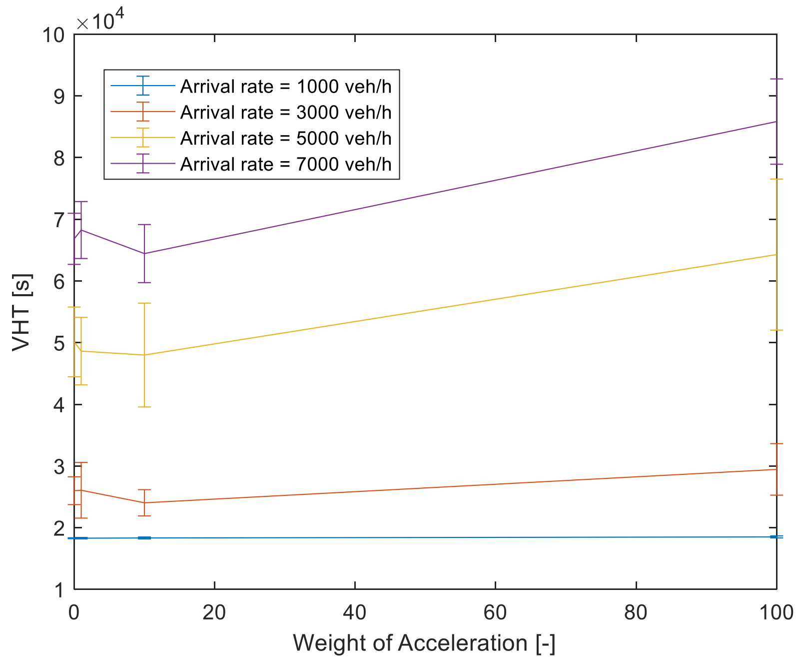


Figure 33 Queue time versus Weight of Acceleration

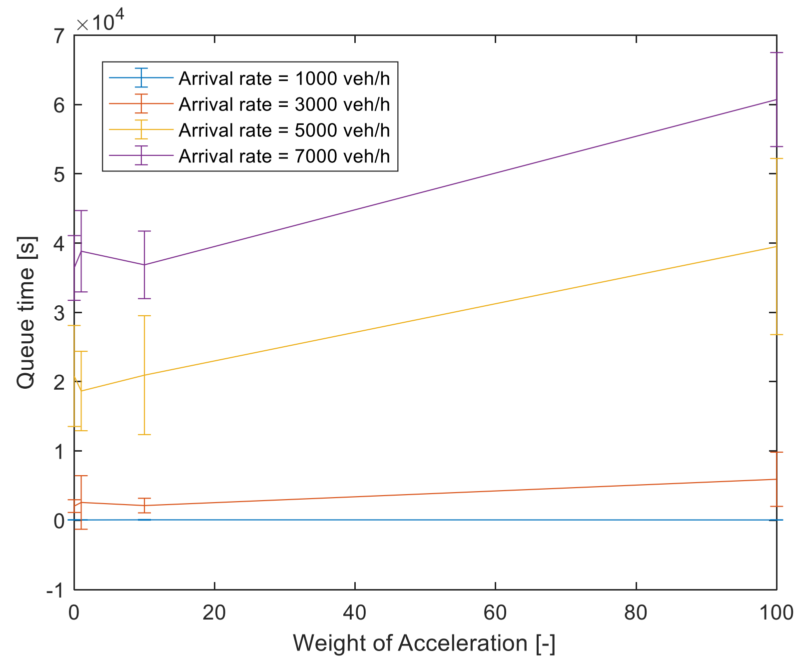


Figure 34 Average speed versus Weight of Acceleration

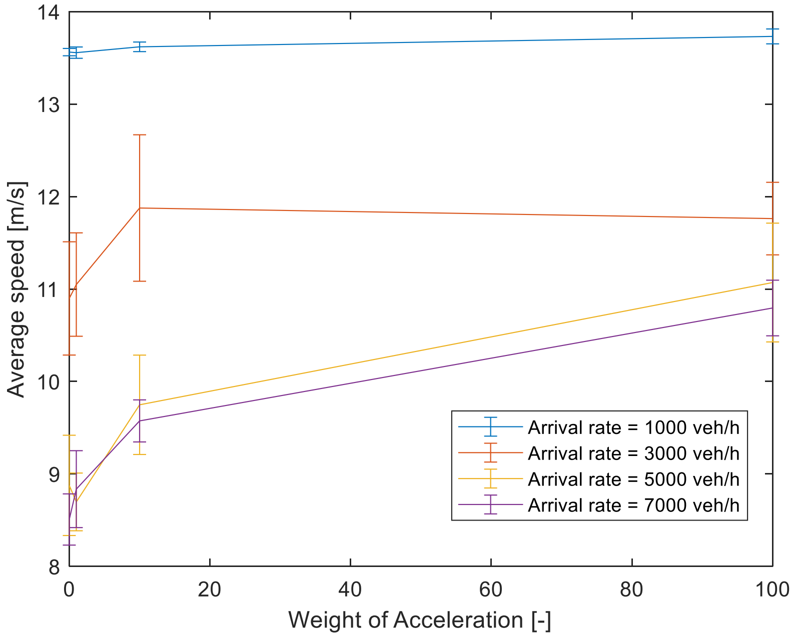
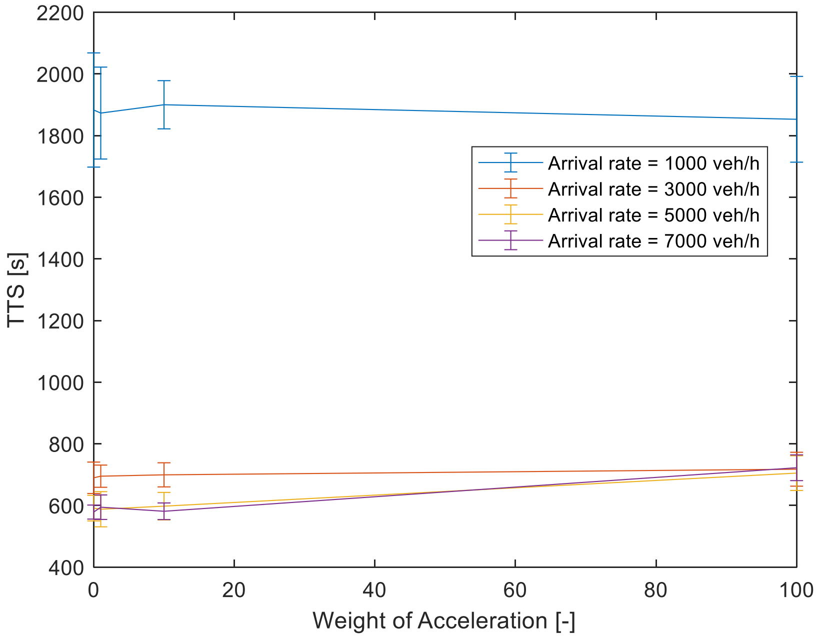


Figure 35 TTS versus Weight of Acceleration



Weight of Desired speed

Figure 36 Queue time versus Weight of Desired speed

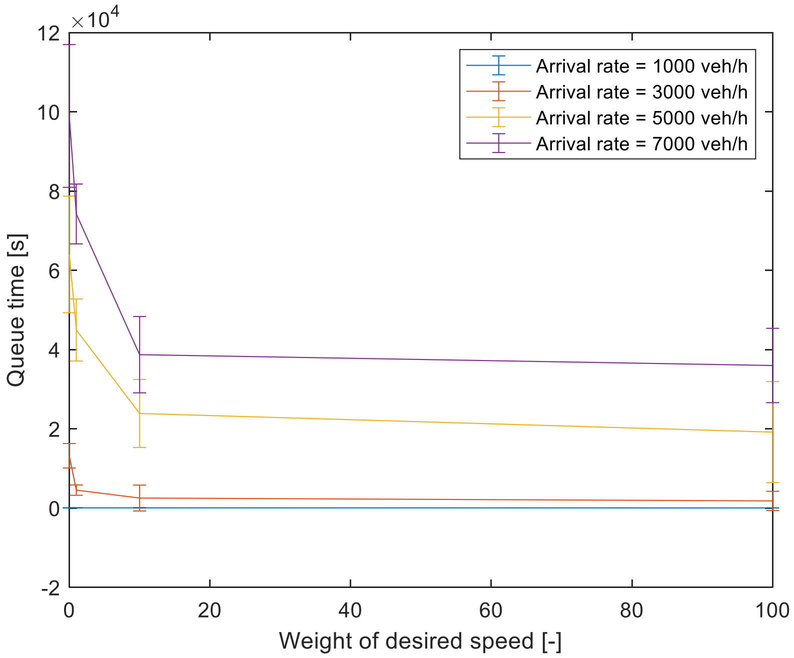


Figure 37 Average speed versus Weight of Desired speed

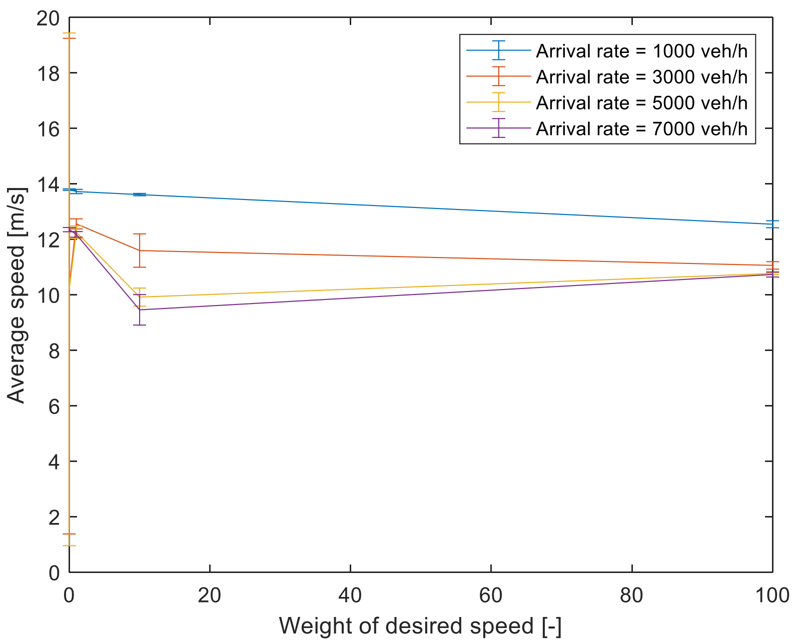
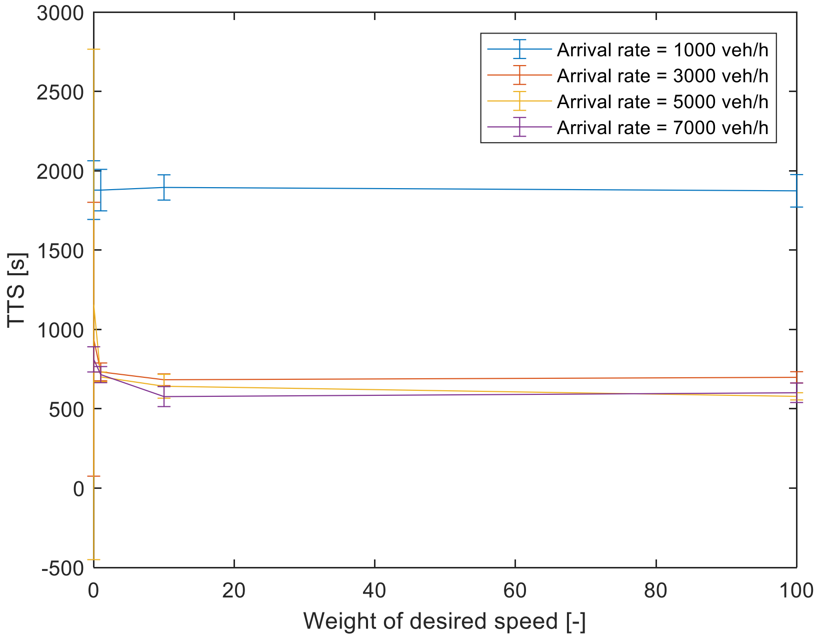


Figure 38 TTS versus Weight of Desired speed



Weight of IDM-Distance

Figure 39 Queue time versus Weight of IDM-Distance

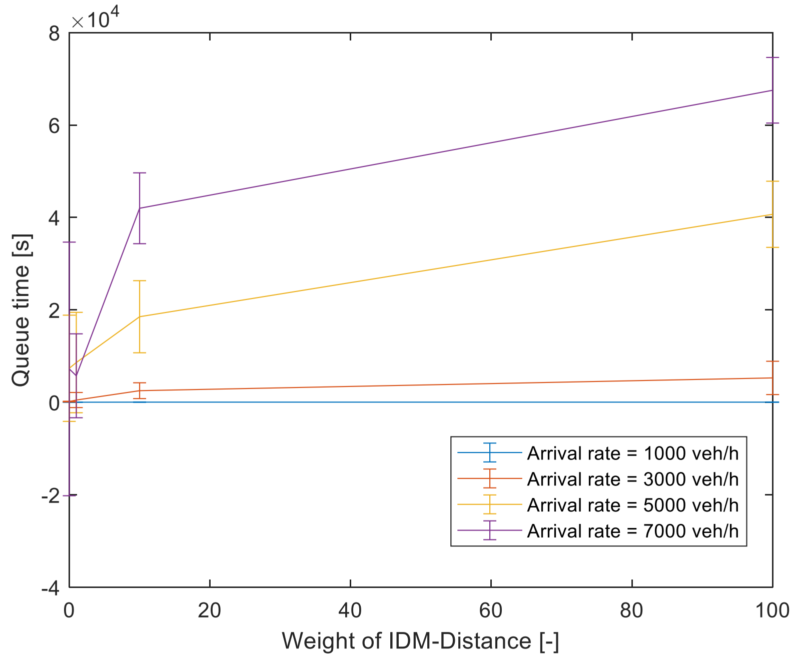


Figure 40 Average speed versus Weight of IDM-Distance

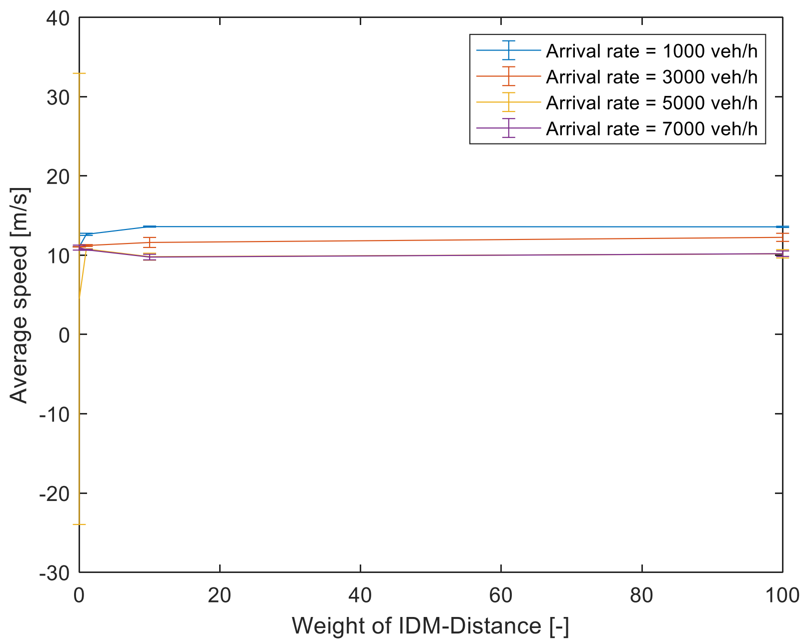
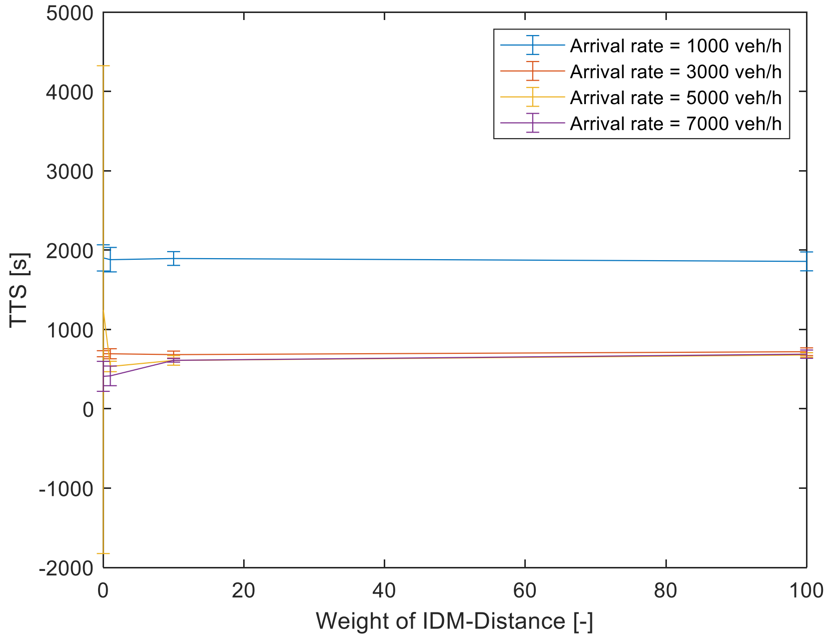


Figure 41 TTS versus Weight of IDM-Distance



Weight of No-IDM-Distance

Figure 42 VHT versus Weight of No-IDM-Distance

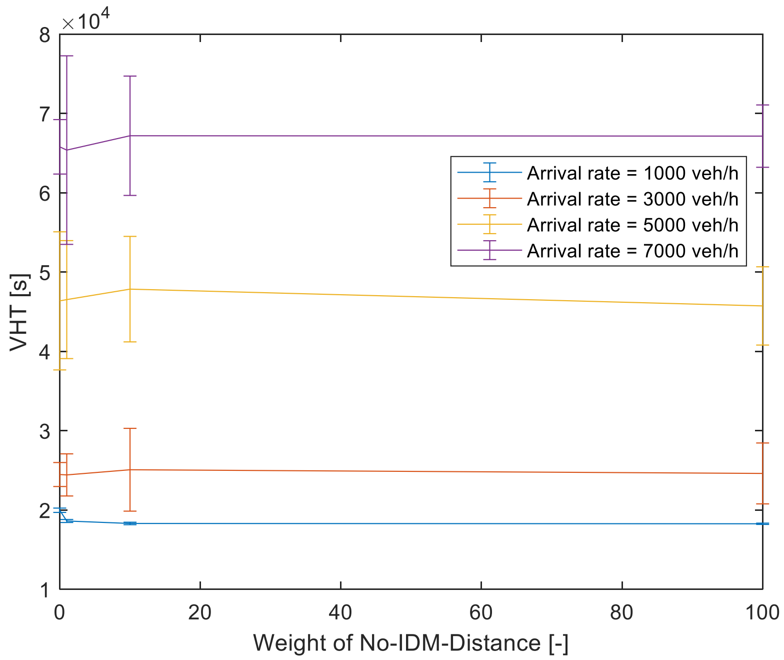


Figure 43 Queue time versus Weight of No-IDM-Distance

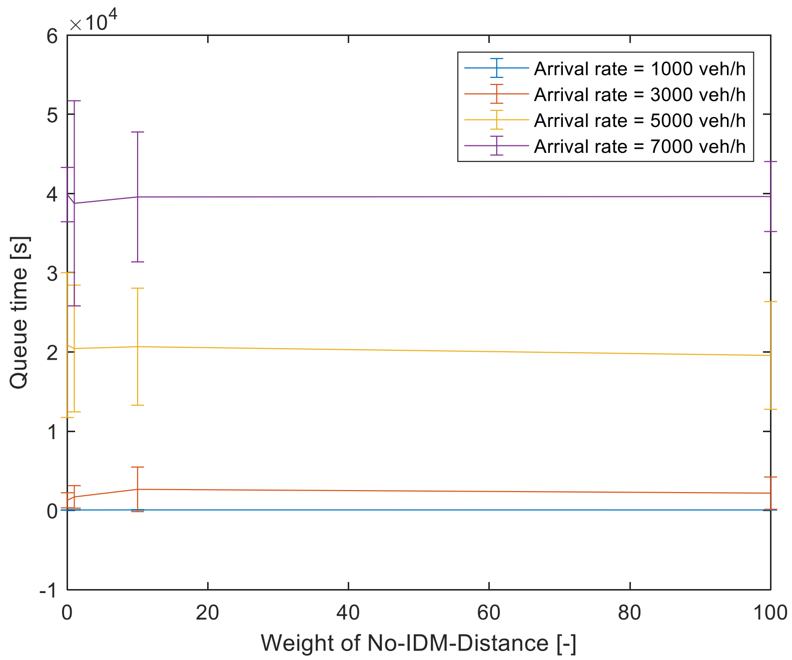


Figure 44 Average speed versus Weight of No-IDM-Distance

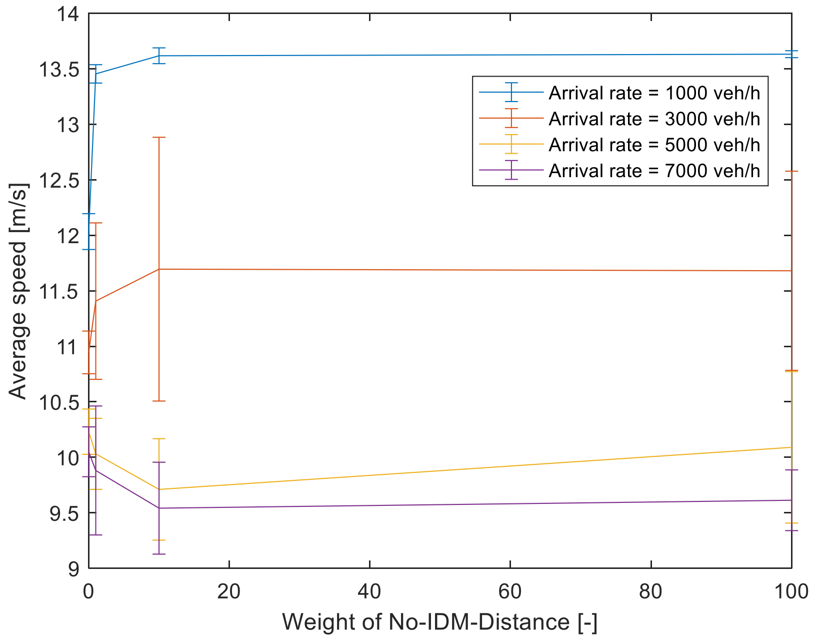
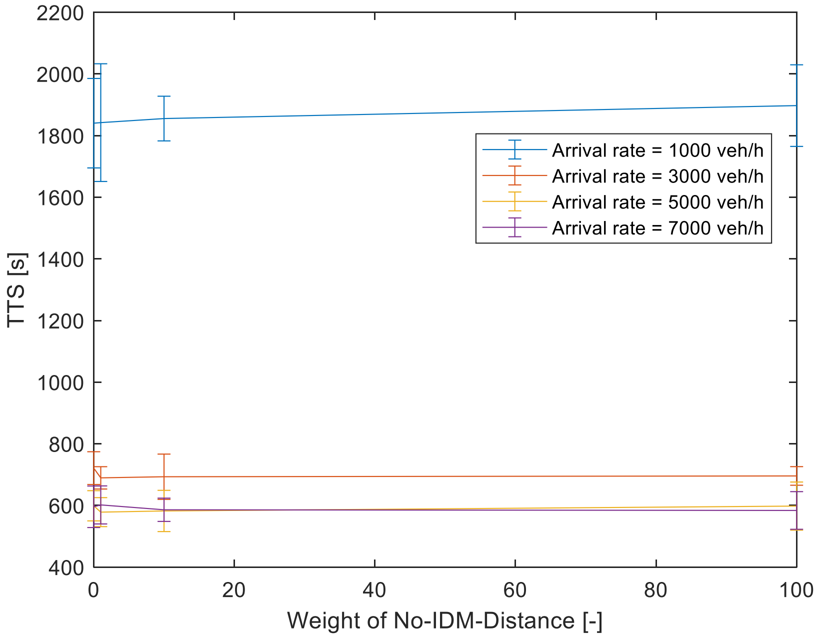


Figure 45 TTS versus Weight of No-IDM-Distance



A 2 Results for Scenario 2

Arrival rates

Figure 46 VHT versus Arrival rate

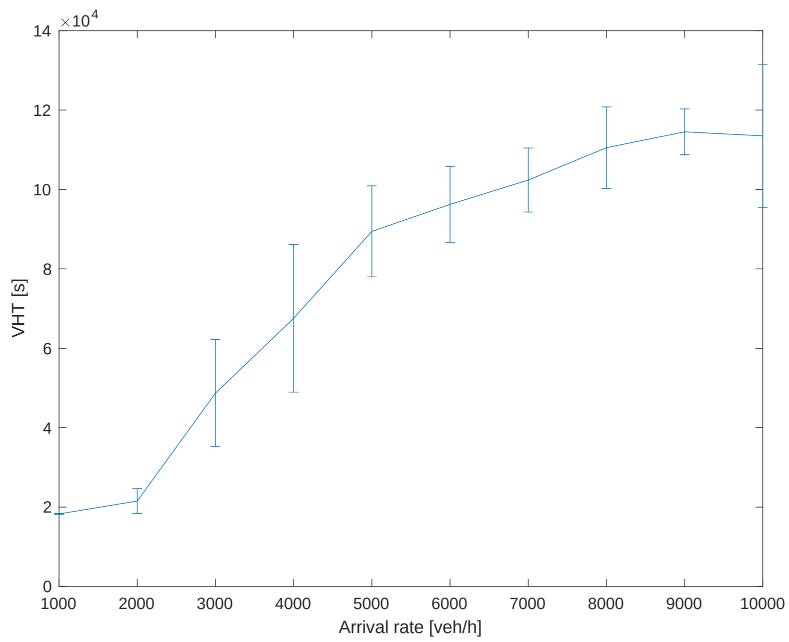


Figure 47 Queue time versus Arrival rate

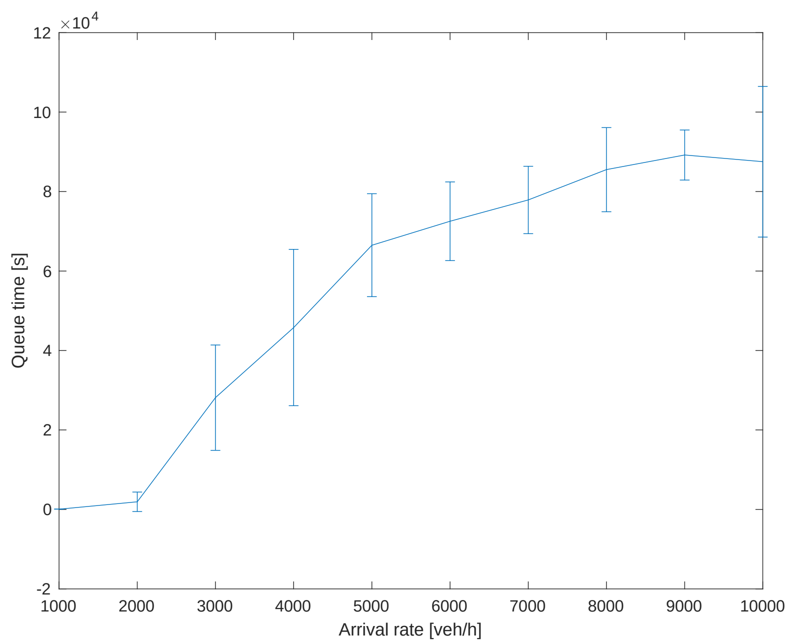


Figure 48 Average speed versus Arrival rate

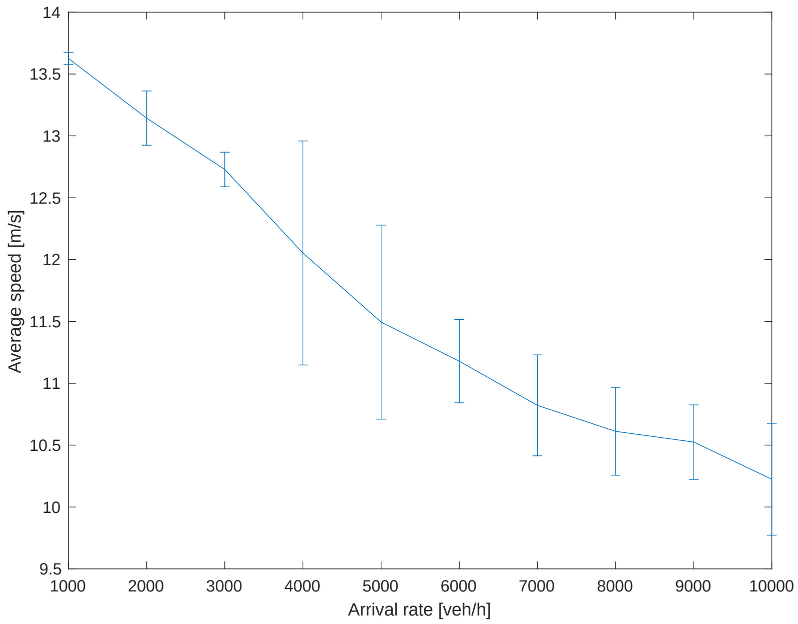
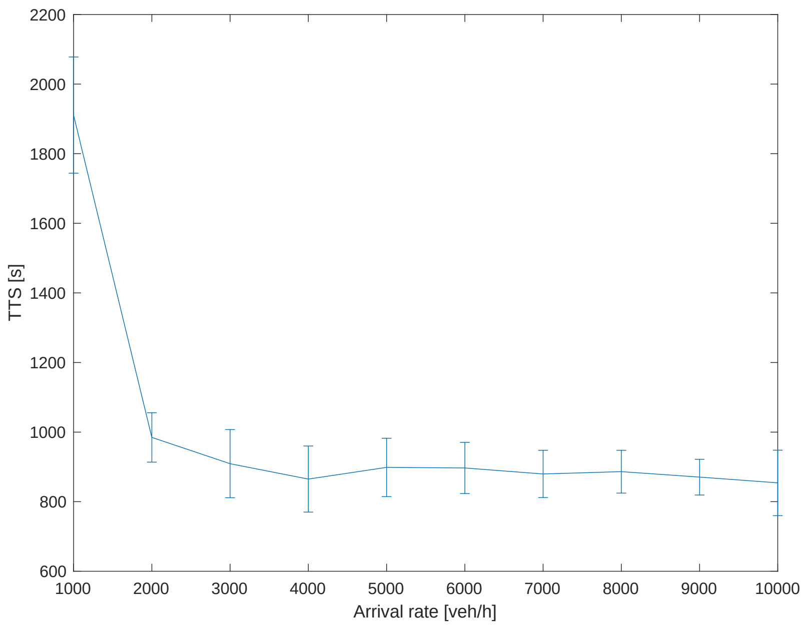


Figure 49 TTS versus Arrival rate



Distance for Pair-building

Figure 50 VHT versus Distance for Pair-building

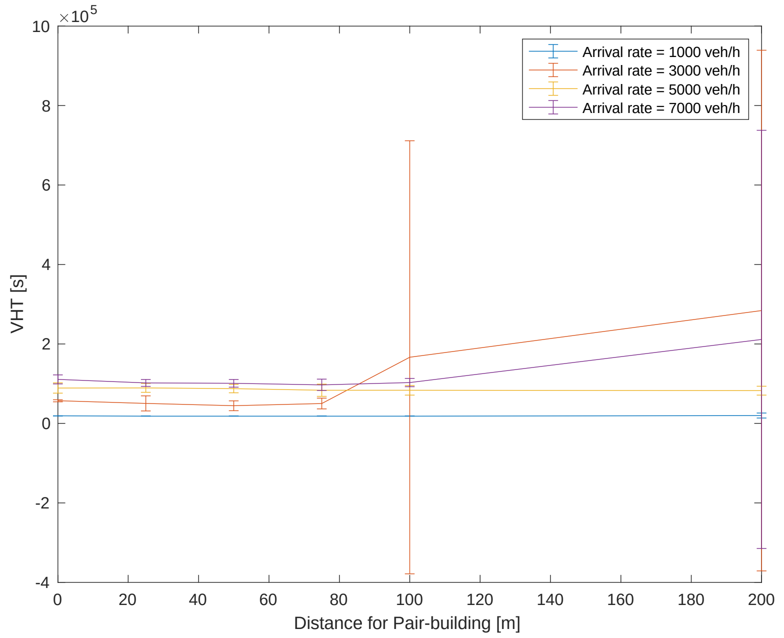


Figure 51 Queue time versus Distance for Pair-building

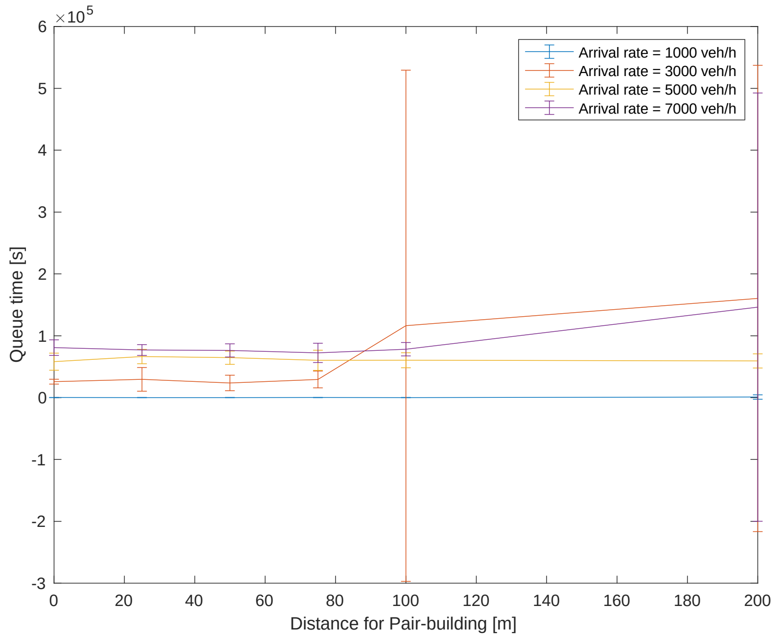


Figure 52 Average speed versus Distance for Pair-building

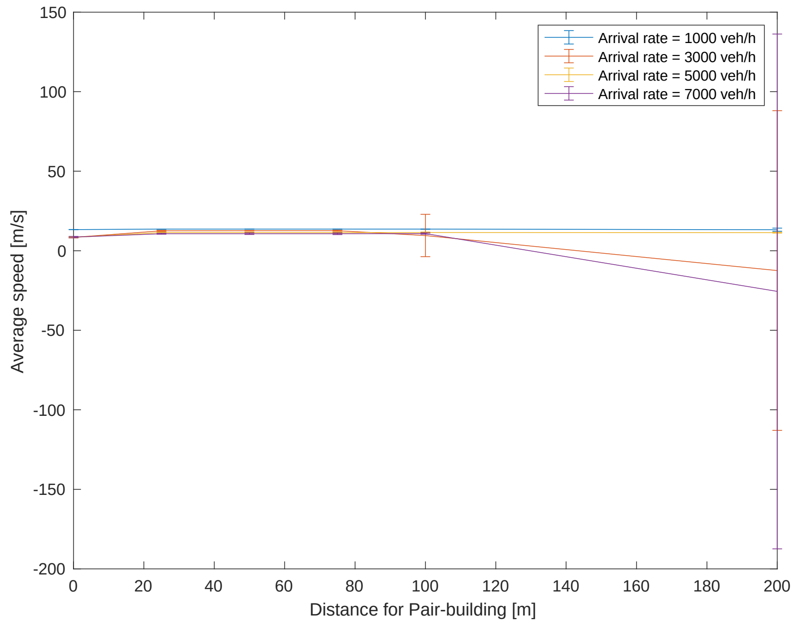
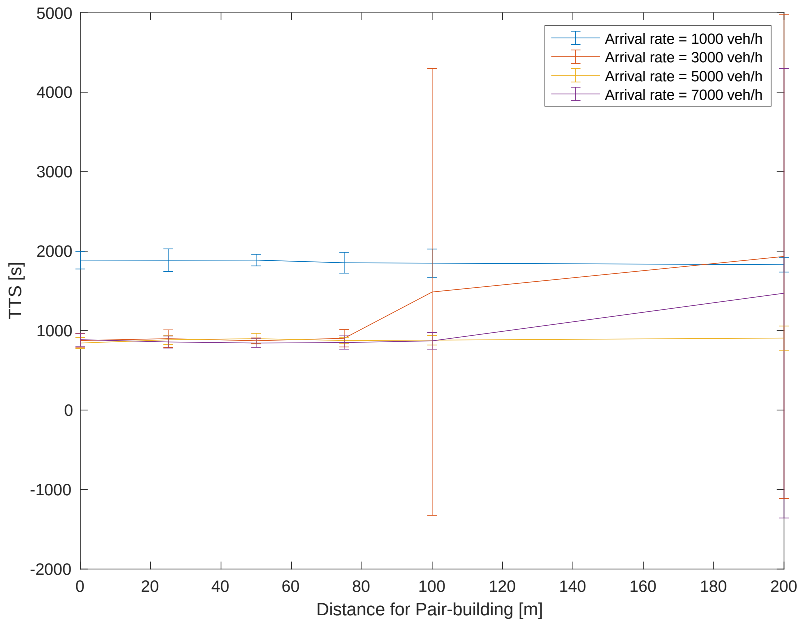


Figure 53 TTS versus Distance for Pair-building



Desired speed

Figure 54 VHT versus Desired speed

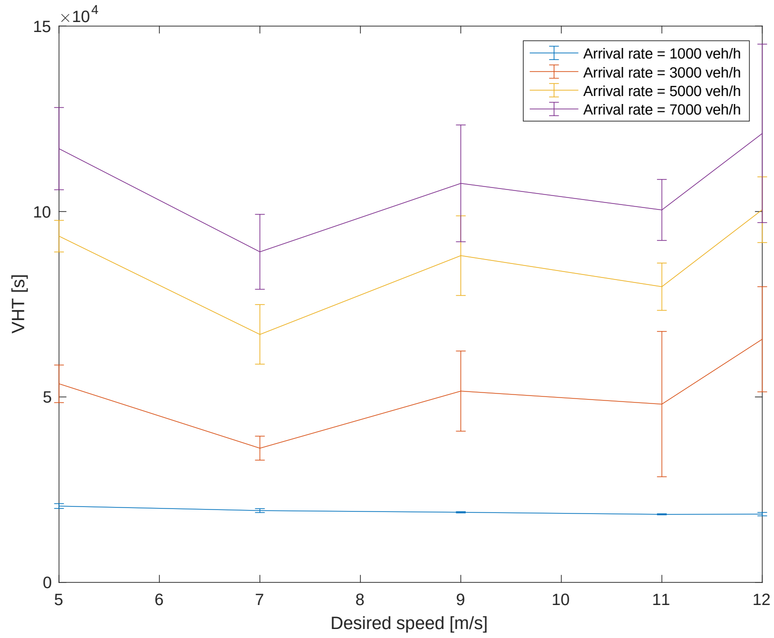


Figure 55 Queue time versus Desired speed

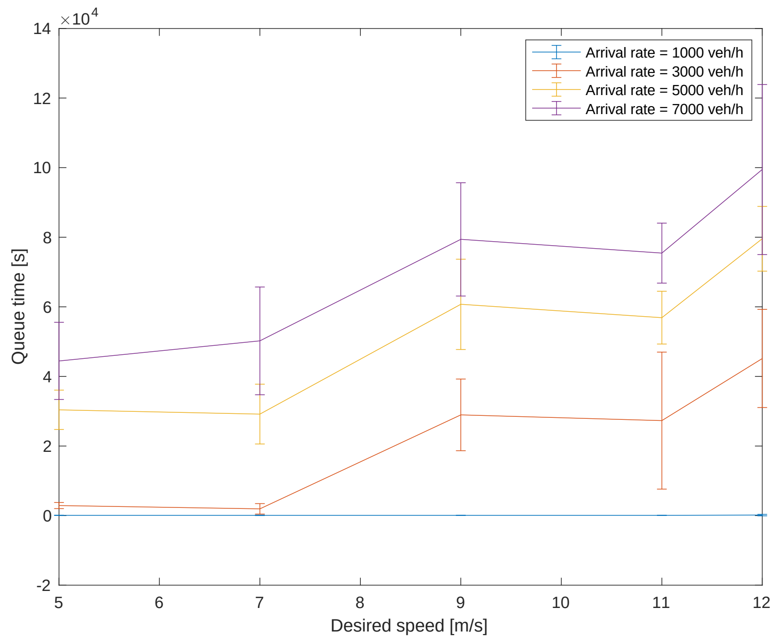


Figure 56 Average speed versus Desired speed

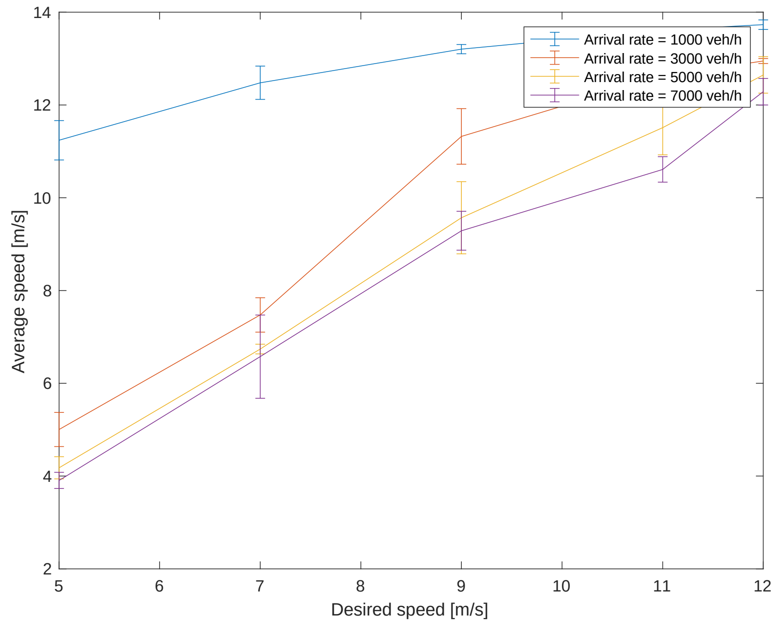
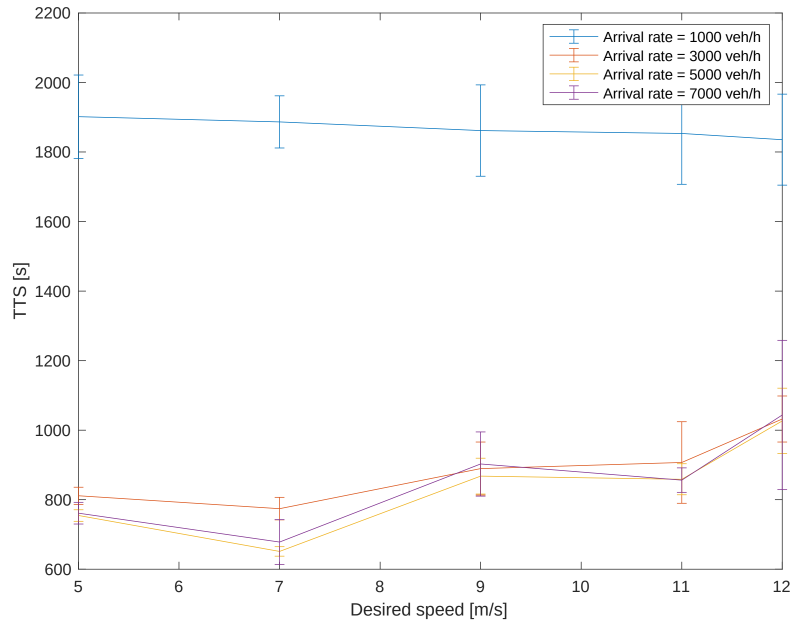


Figure 57 TTS versus Desired speed



Safety distance

Figure 58 VHT versus Safety distance

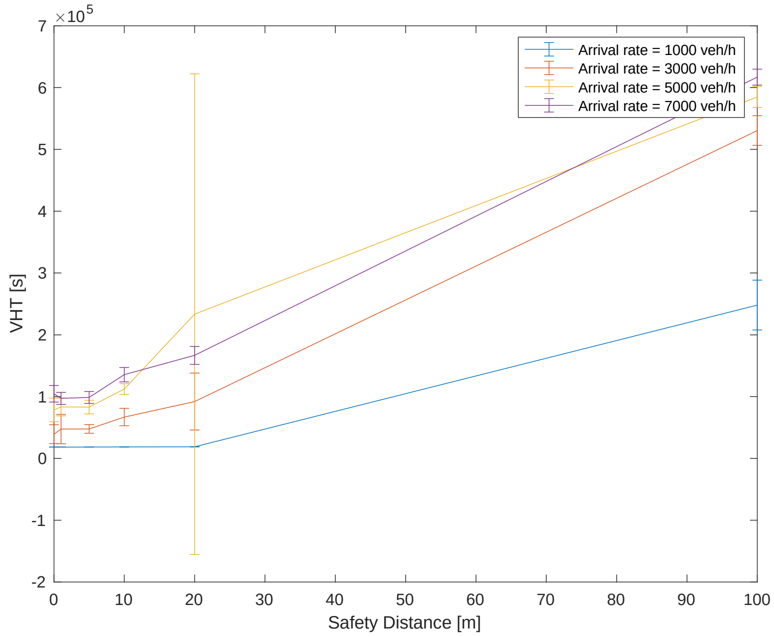


Figure 59 Queue time versus Safety distance

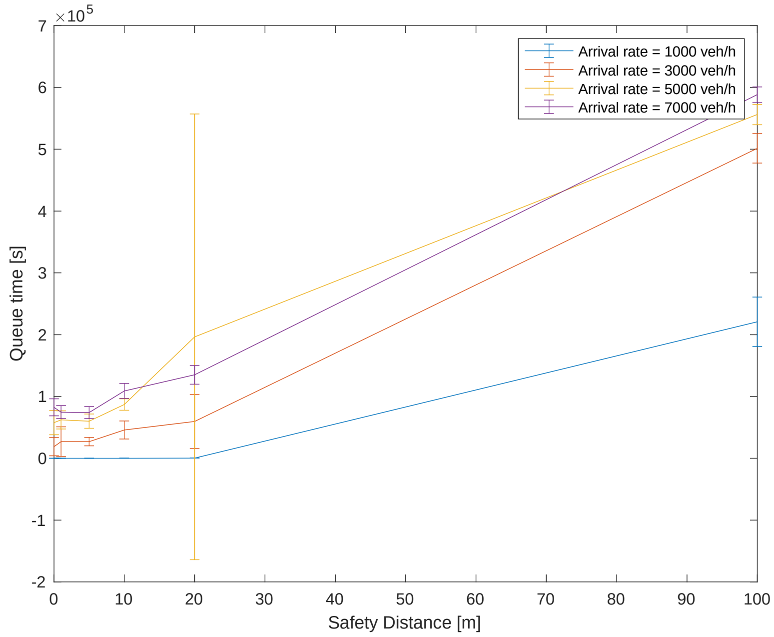


Figure 60   Average speed versus Safety distance

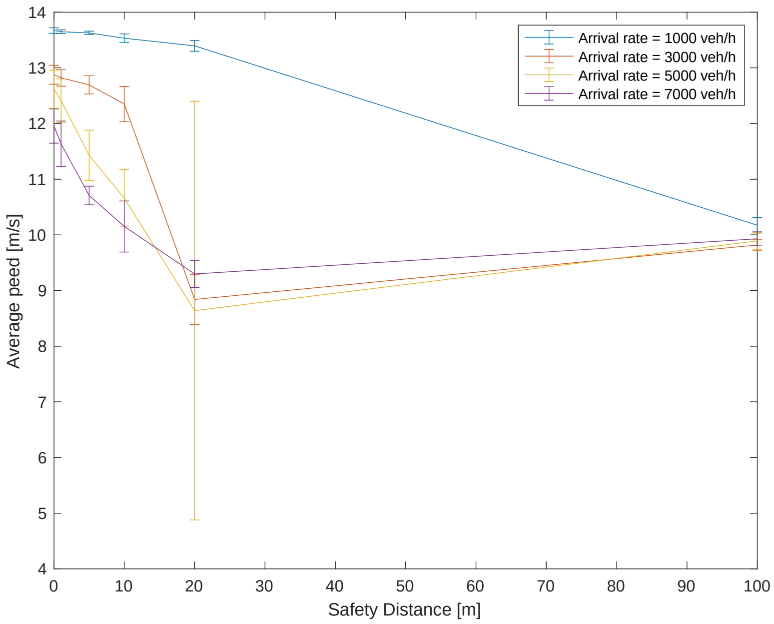
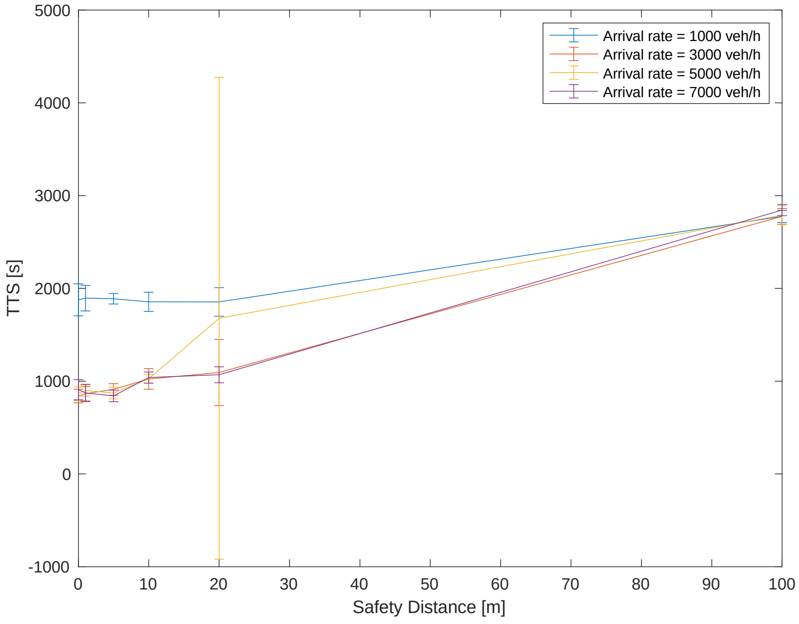


Figure 61   TTS versus Safety distance



Weight of Acceleration

Figure 62 VHT versus Weight of Acceleration

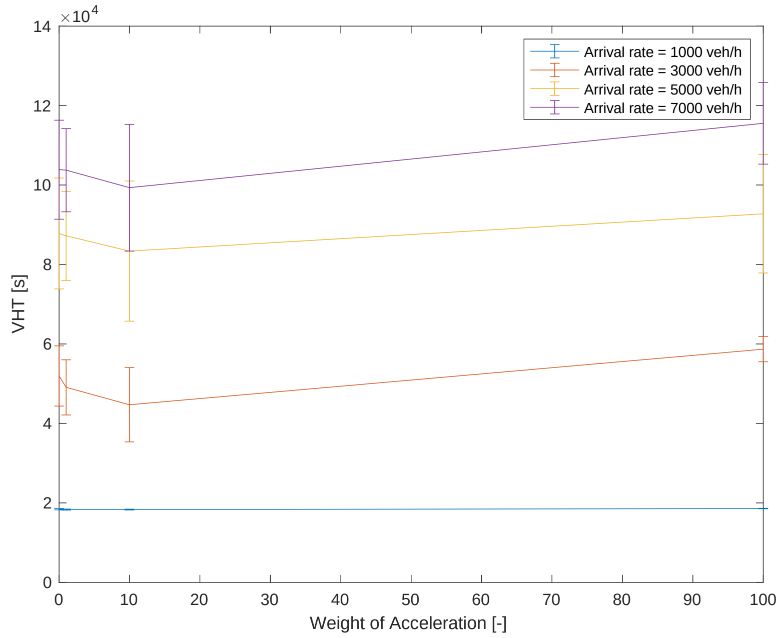


Figure 63 Queue time versus Weight of Acceleration

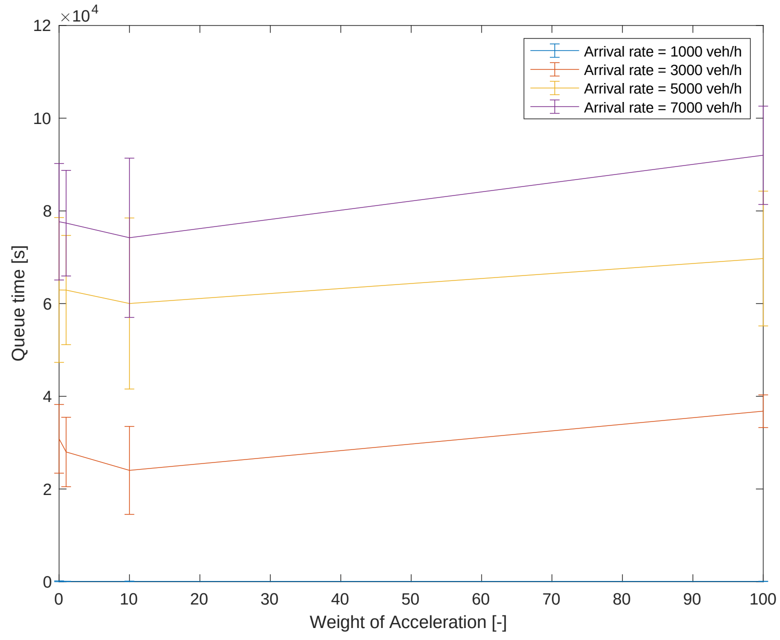


Figure 64 Average speed versus Weight of Acceleration

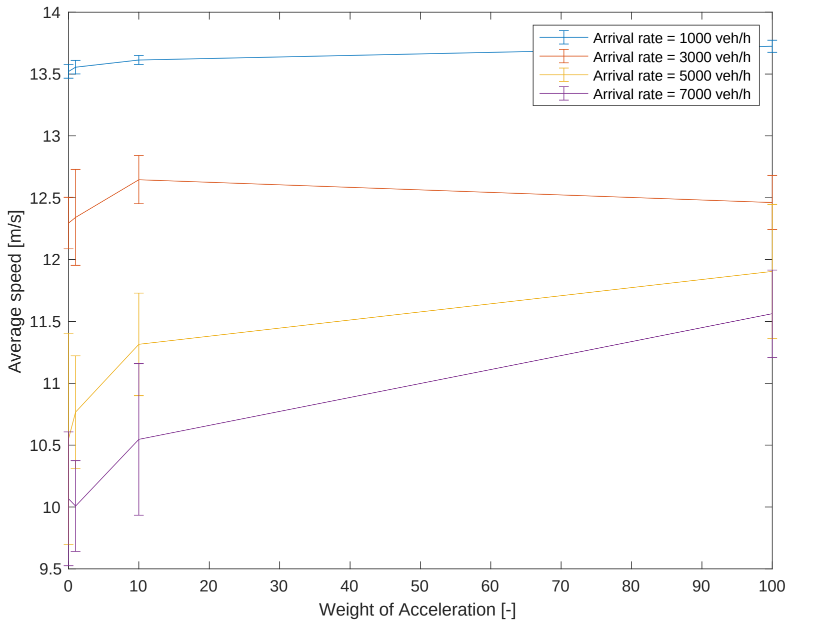
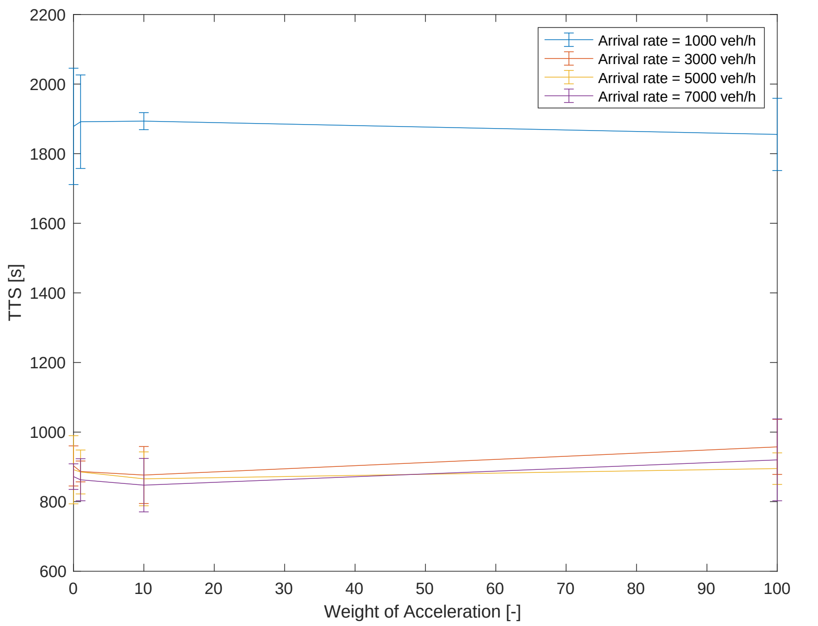


Figure 65 TTS versus Weight of Acceleration



Weight of Desired speed

Figure 66 VHT versus Weight of Desired speed

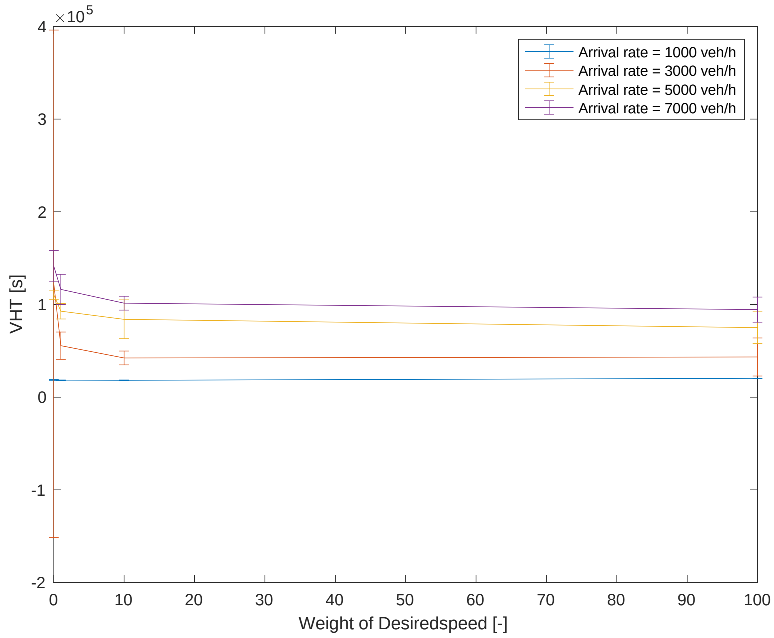


Figure 67 Queue time versus Weight of Desired speed

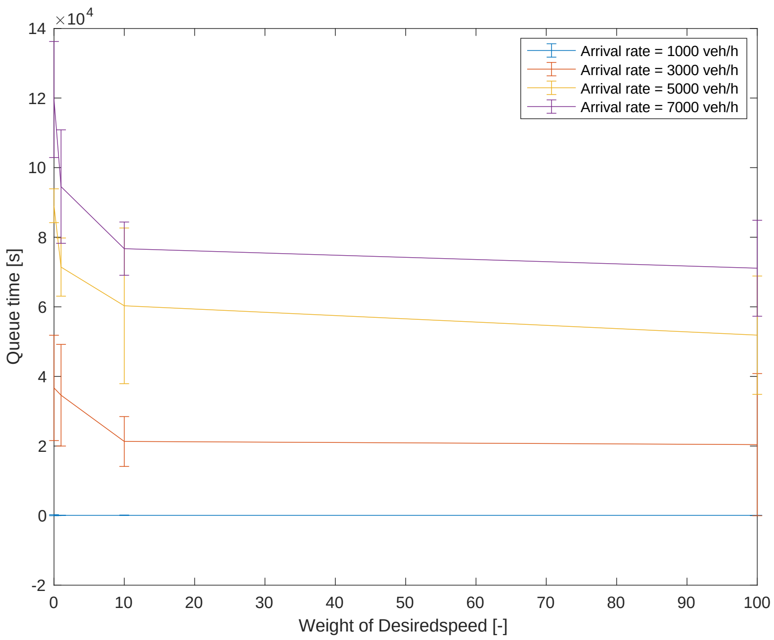


Figure 68 Average speed versus Weight of Desired speed

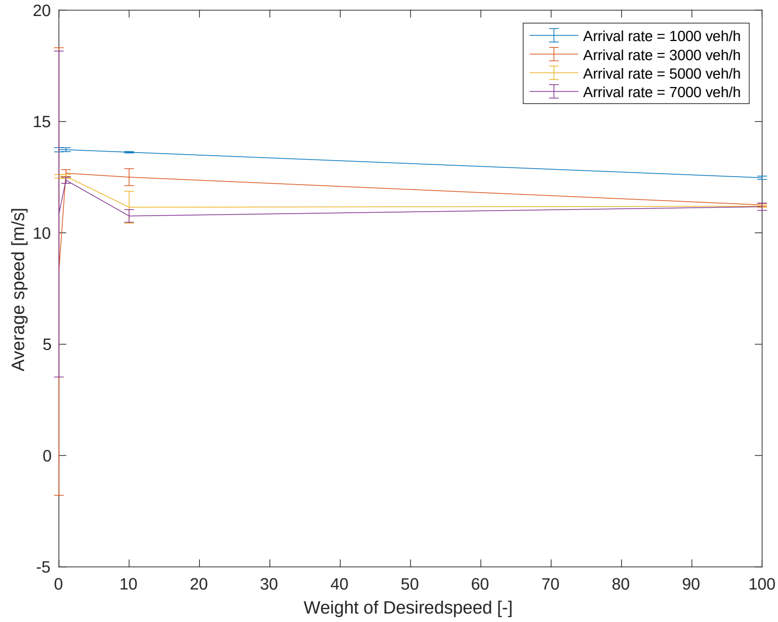
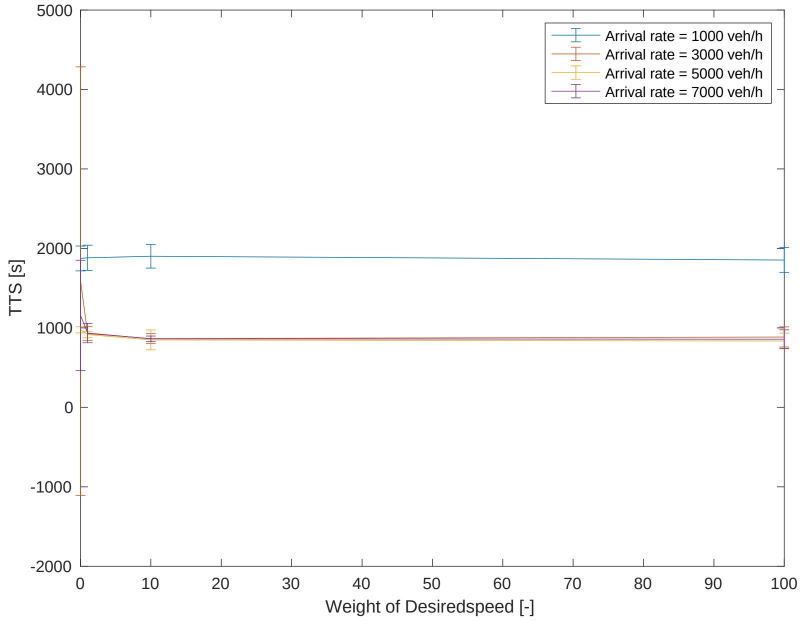


Figure 69 TTS versus Weight of Desired speed



Weight of IDM-Distance

Figure 70 VHT versus Weight of IDM-Distance

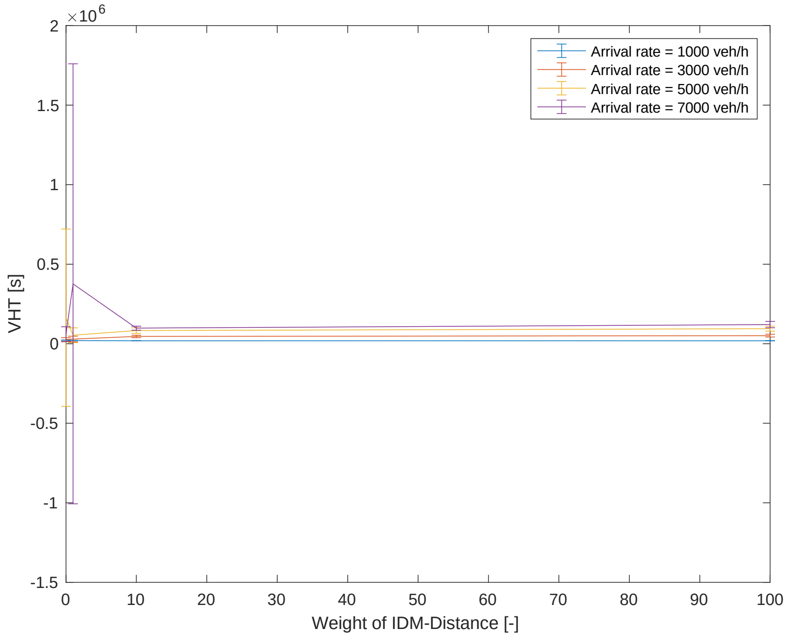


Figure 71 Queue time versus Weight of IDM-Distance

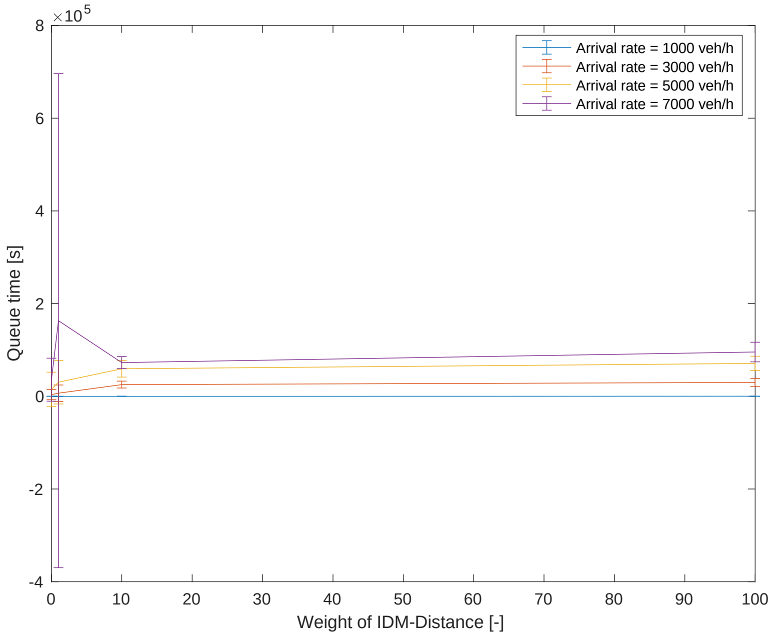


Figure 72 Average speed versus Weight of IDM-Distance

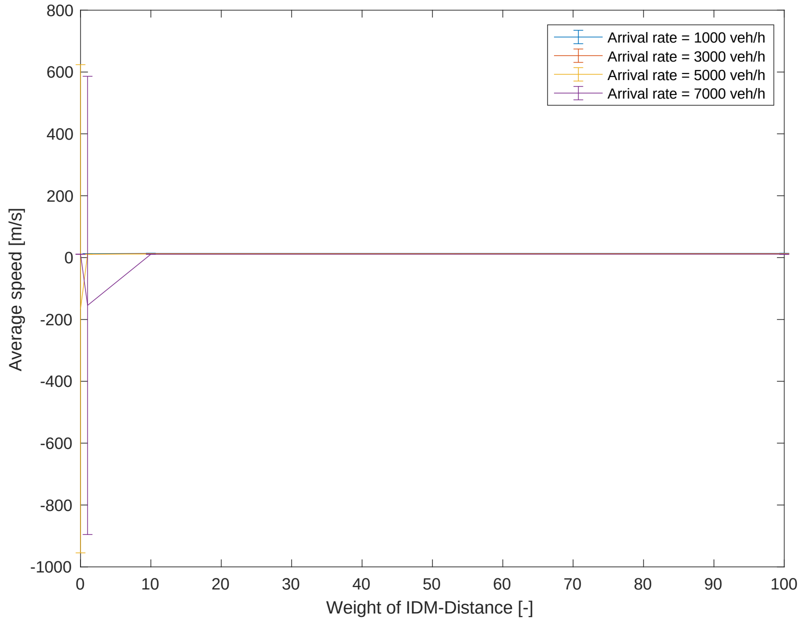
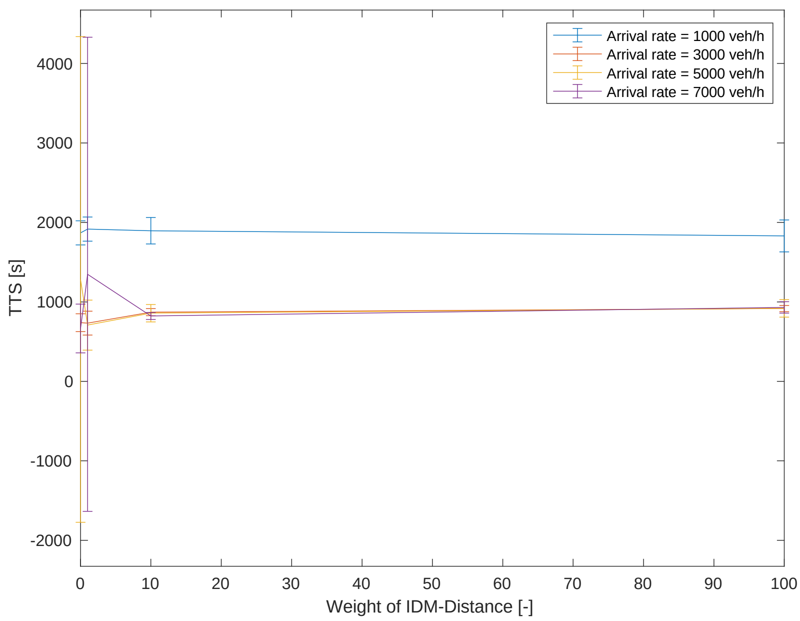


Figure 73 TTS versus Weight of IDM-Distance



Weight of No-IDM-Distance

Figure 74 VHT versus Weight of No-IDM-Distance

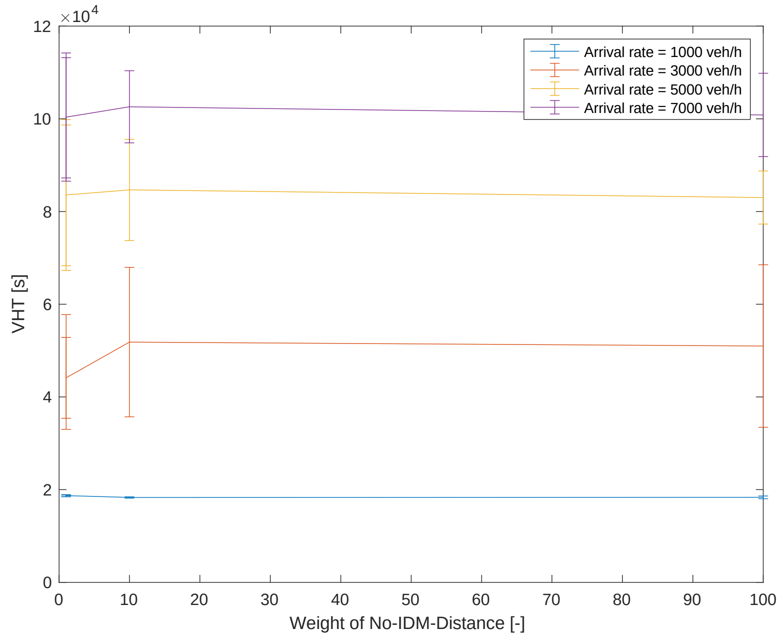


Figure 75 Queue time versus Weight of No-IDM-Distance

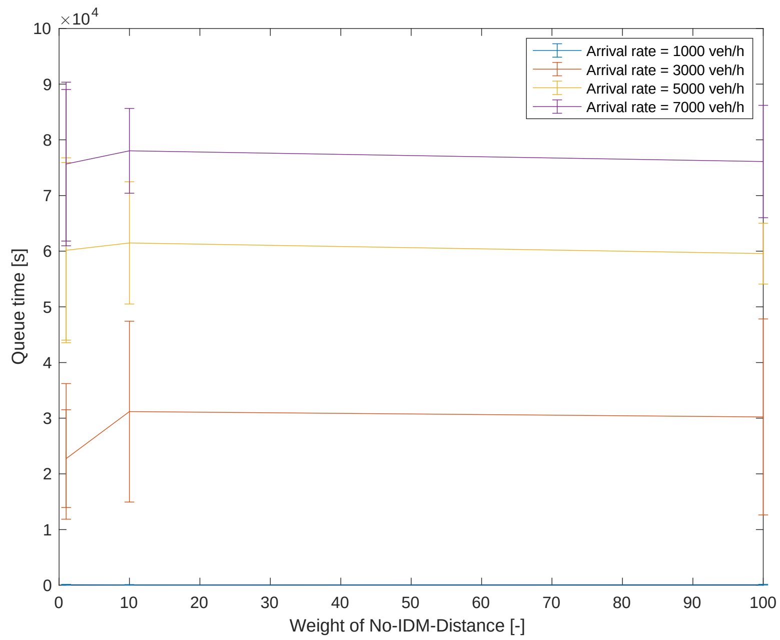


Figure 76 Average speed versus Weight of No-IDM-Distance

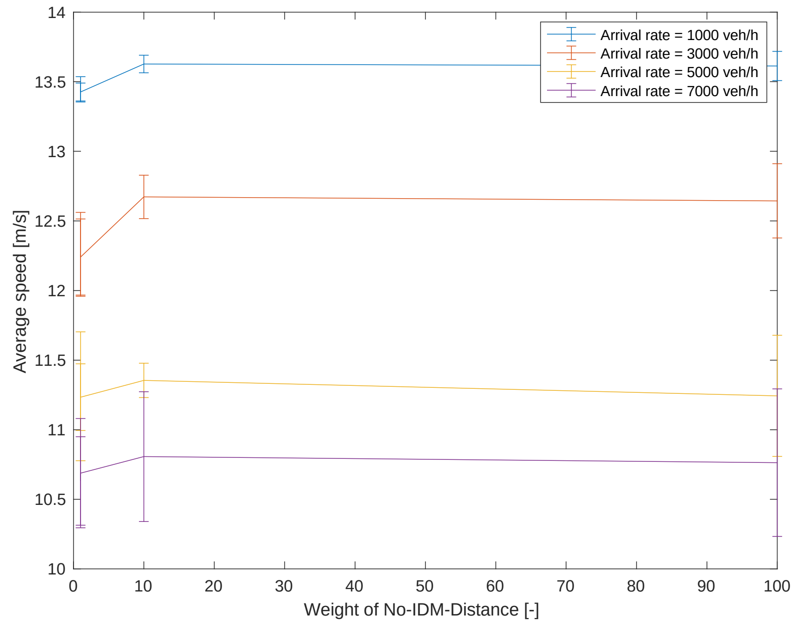


Figure 77 TTS versus Weight of No-IDM-Distance