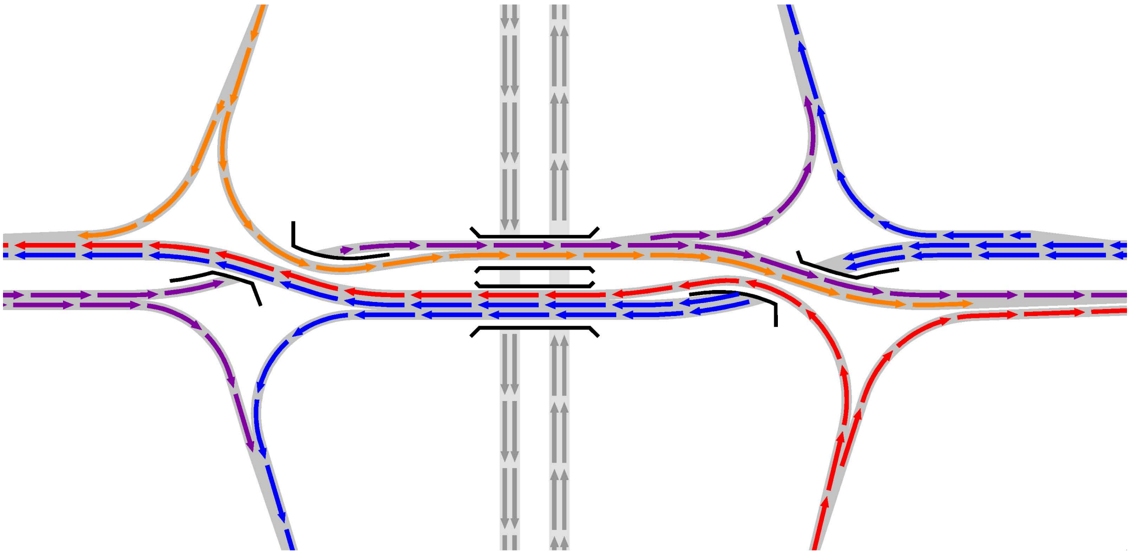
CHALMERS
UNIVERSITY OF TECHNOLOGY



Simulation-based Analysis of Partially Automated Vehicular Networks

A Parametric Analysis Utilizing Traffic Simulation

Master’s thesis in Product Development

FREDRIK NILSSON

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Simulation-based Analysis of Partially Automated Vehicular Networks

A Parametric Analysis Utilizing Traffic Simulation

FREDRIK NILSSON

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2019

Simulation-based Analysis of a Partially Automated Vehicular Networks
A Parametric Analysis Utilizing Traffic Simulation
FREDRIK NILSSON

© FREDRIK NILSSON, 2019.

Supervisor: Eric Bjärkvik, Alten Sweden
Examiner: Balázs Adam Kulcsár, Chalmers University of Technology

Master's Thesis 2019
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Illustration of traffic flow at a diverging diamond intersection.

Typeset in LaTeX
Gothenburg, Sweden 2019

Simulation-based Analysis of Partially Automated Vehicular Networks
A Parametric Analysis Utilizing Traffic Simulation
FREDRIK NILSSON
Department of Electrical Engineering
Chalmers University of Technology

Abstract

For microscopic traffic simulation to be useful it is critical that the results can be trusted. This thesis proposes a method for tuning traffic simulation against real world detector data by using a black-box optimization algorithm to minimize the difference (with respect to some objective function) between simulation and reality with driver parameters from the car following and lane change models as input variables.

This method of fitting the simulation to data was then used to model, fit and test a specific area in Gothenburg within the simulation environment. The purpose was to see both what behaviour in human drivers that lead to traffic congestion and to introduce different fractions of autonomous vehicles into the tuned simulation to see their impact. Finally, the autonomous vehicles also had their behaviour optimized with the same optimization algorithm (but of course a different objective function) to see how much they could potentially improve traffic while constituting 10%, 25%, 50% and 100% of the total traffic.

The optimized simulations show significant improvement in accuracy as compared to a default setting simulation and this is presented both numerically and graphically in the report. The results also show that while there is room for improvement in human drivers, a lot of the congestion in the area is unavoidable even with a fully automated system if the in- and outflow of vehicles is kept the same. This can be due to road design or other factors but the way in which the objective function was defined penalizes the autonomous vehicles slightly for sticking to the speed limit. Despite this fact, having only AVs in the area would increase the average speed by over 14% during rush hours while also never going above the speed limit.

Keywords: Autonomous vehicles (AVs), traffic simulation, human driver behaviour, optimization

Acknowledgements

This thesis project has been conducted at Alten Sweden in Gothenburg during the spring of 2019. I would like to thank Alten for hosting my thesis and a special thanks to my industrial supervisor Eric Bjärkvik for all the help and discussion he has provided. I would also like to thank the Department of Electrical Engineering at Chalmers, and more specifically my examiner Balázs Adam Kulcsár for all his active help in the project.

I want to thank Tommy Niittula and the rest of Trafikverket for providing me with detector data for the project. Finally, I would like to thank the Jakob Erdmann and others at the Sumo mailing list for assisting with issues related to Sumo.

Fredrik Nilsson, Gothenburg, May 2019

Contents

List of Figures

List of Tables

1
Introduction

Traffic and the complex interplay between physics and the psychological behaviour that drive it have been studied for over 100 years. Many models to describe the flow of traffic and the behaviour of drivers have been developed including the Kinematic wave model [1], Newell-Daganzo merge model [2][3] and various car-following models, see Section 2.1.1.

These types of models have lead to the development of many different simulation software, both commercial (e.g. *Vissim* [4]) and open-source (e.g. *Sumo* [5]). In this report, simulation is used to study the effects of human driver behaviour in partially automated vehicular networks.

This thesis is done at Alten Sweden and is done as the final project for a master's degree in Product Development, however the thesis is hosted by the Department of Electrical Engineering at Chalmers University of Technology.

1.1 Background

As the automotive industry becomes more focused on autonomous vehicles there is an increased need for understanding how traffic works, why it works this way and how it can be manipulated to optimize the flow of vehicles on the roads. Of course, the transition from manually driven vehicles to autonomous vehicles will happen continuously as more autonomous vehicles are introduced, rather than immediately reaching the supposed end state where all cars are autonomous. The point of this project is to take a deeper look at how the introduction of autonomous vehicles will affect areas prone to congestion and what type of behaviour (from both autonomously driven and human driven vehicles) lead to traffic congestion.

1.2 Aim

The aim of the project is to find which driver parameters trigger traffic congestion in vehicular networks both with and without the presence of autonomous vehicles. This will be done by modelling a real location in a simulation environment, tune the simulation to match real world detector data and to then analyse the simulation results when introducing different driver behaviour and autonomous vehicles.

1.3 Limitations

There are certain limitations on the project that are mostly due to the nature of simulations. One of main challenges of the project will be to adequately model and differentiate between the behaviour of the autonomously driven vehicles and human driven vehicles, since the human drivers are still following a certain algorithm in the simulation. There might be good reason to introduce some stochastic behaviour in human driven vehicles to better replicate the drivers on the road today.

The project will focus only on one specific area, but the goal is for the findings to be universal to some extent. Specific behaviour might be related only to the chosen location but other more general trends may be applicable anywhere.

Finally, one important consideration is that not all behaviour that might impact traffic is tracked in neither simulation nor camera data from the location, such as drivers giving one another gestures through the car window. There is a risk that one or more of these types of behaviours have a major impact that will be missed.

1.4 Research questions

The thesis aims to answers the following research questions:

- What driver behaviour leads to more or less congestion in the area?
- How does introducing different fractions of autonomous vehicles impact traffic in the area?
- How can autonomous vehicles behave to improve traffic?
- What do the findings imply for the future development of autonomous vehicles?

1.5 Method

The project follows a specific structure that is presented in detail in Chapter 3. A flowchart describing the process is presented in Figure 1.1.

1.5.1 Simulation software

The thesis used *Sumo* as its simulation software. The main reasons behind this decision was:

- It's free and open source, meaning that others can relatively easily replicate the study.
- It has a command line interface, making it easy to use together with other tools.
- It has previously been used in Gothenburg with good results [6].

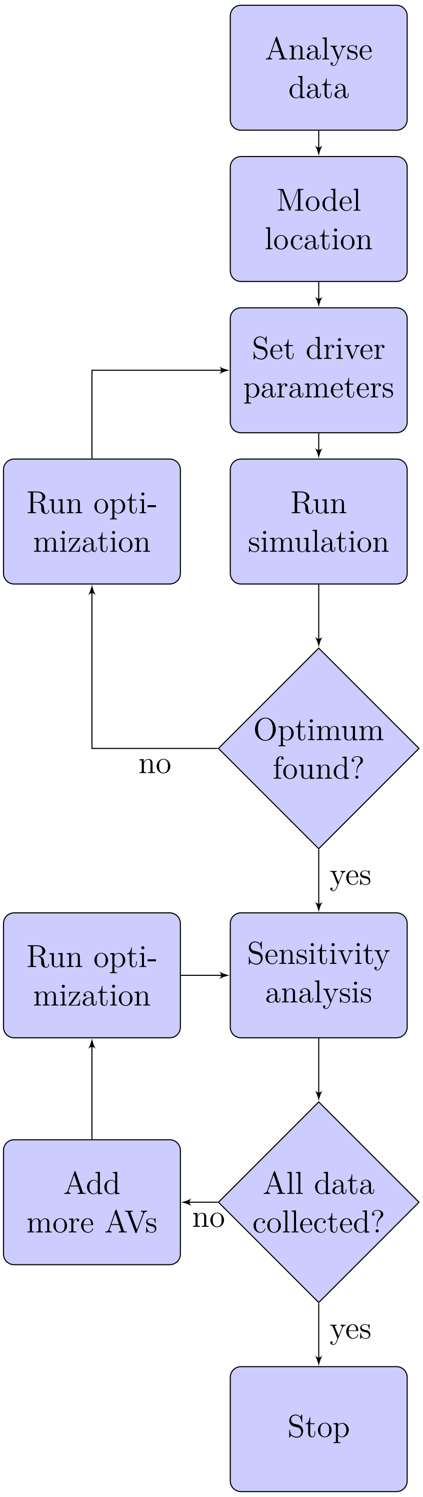


Figure 1.1: Flowchart describing the process used in the project.

2

Theory

This chapter aims to describes the theory used throughout the report, including traffic specific theory and other mathematical models. In some cases, the theory is only described briefly in broad terms and other papers on the matter are referenced instead.

2.1 Simulation parameters

The simulation can take different input parameters which will produce different outputs. This section describes those input parameters.

2.1.1 Car-following model

The most fundamental part of traffic simulation is perhaps the car-following model as it dictates how vehicles interact, follow and avoid each other longitudinally while driving. This project uses the SUMOKrauß model for human drivers, which is based on the original Krauß model [7]. There are two changes that have been made in SUMOKrauß as compared to the original [8]:

- Different deceleration capabilities among the vehicles are handled without violating safety (the original model allowed for collisions in this case).
- The formula for safe velocity was adapted to maintain safety when using the Euler-position update rule. This was done by discretizing some of the continuous terms. The original model was defined for the Ballistic-position updated rule and would produce collisions when using Euler.

The implementation of the model in C++ can be seen on SUMO's GitHub page [9]. The parameters available in SUMOKrauß are listed and described in Table 2.1.

Parameter	Description
accel	The acceleration ability of vehicles of this type [m/s^2]
decel	The deceleration ability of vehicles of this type [m/s^2]
emergencyDecel	The maximum deceleration ability of vehicles of this type in case of emergency [m/s^2]
sigma	The driver imperfection (between 0 and 1)*
tau	The driver's desired (minimum) time headway [s]
minGap	The minimum distance between two vehicles at stand-still [m]

Table 2.1: Parameters available in SUMOKrauß.

*The driver imperfection applies a uniform random deceleration from [0, sigma * min(accel, currentSpeed)], effectively meaning that the larger the driver imperfection the less stable driving behaviour becomes.

Rather than using the same model for autonomous vehicles, a *Cooperative Adaptive Cruise Control (CACC)* model is used. CACC has previously been assessed using SUMO [10] and the same paper also describes the transfer functions that the CACC model implements in SUMO. The parameter names and descriptions in SUMO's nomenclature can be seen in Table 2.2.

Parameter	Description
speedControlGainCACC	The control gain determining the rate of speed deviation (Speed control mode)
gapClosingControlGainGap	The control gain determining the rate of positioning deviation (Gap closing control mode)
gapClosingControlGainGapDot	The control gain determining the rate of the positioning deviation derivative (Gap closing control mode)
gapControlGainGap	The control gain determining the rate of positioning deviation (Gap control mode)
gapControlGainGapDot	The control gain determining the rate of the positioning deviation derivative (Gap control mode)
collisionAvoidanceGainGap	The control gain determining the rate of positioning deviation (Collision avoidance mode)
collisionAvoidanceGainGapDot	The control gain determining the rate of the positioning deviation derivative (Collision avoidance mode)

Table 2.2: Parameters available in CACC.

2.1.2 Lane-changing model

While the car-following model deals with longitudinal behaviour it does not cover the cars movement laterally, most importantly lane-changes. For this purpose a separate set of models exist, namely *lane-changing models*. In this project the SUMO default model *LC2013* [11] is used. The parameters available in *LC2013* are listed and described in Table 2.3 [12]

Parameter	**Description**
lcStrategic	The eagerness for performing strategic lane changing. Higher values result in earlier lane-changing. default: 1.0, range [0-inf[
lcCooperative	The willingness for performing cooperative lane changing. Lower values result in reduced cooperation. default: 1.0, range [0-1]
lcSpeedGain	The eagerness for performing lane changing to gain speed. Higher values result in more lane-changing. default: 1.0, range [0-inf[
lcKeepRight	The eagerness for following the obligation to keep right. Higher values result in earlier lane-changing. default: 1.0, range [0-inf[
lcOpposite	The eagerness for overtaking through the opposite-direction lane. Higher values result in more lane-changing. default: 1.0, range [0-inf[
lcLookaheadLeft	Factor for configuring the strategic lookahead distance when a change to the left is necessary (relative to right lookahead). default: 2.0, range]0-inf[
lcSpeedGainRight	Factor for configuring the treshold asymmetry when changing to the left or to the right for speed gain. By default the decision for changing to the right takes more deliberation. Symmetry is achieved when set to 1.0. default: 0.1, range]0-inf[
lcAssertive	Willingness to accept lower front and rear gaps on the target lane. The required gap is divided by this value. default: 1, range: positive reals

Table 2.3: Parameters available in LC2013.

2.1.3 Other parameters

There are many other simulation parameters in SUMO that are not part of the car-following or lane-changing models and the relevant ones are discussed here.

By default, lane-changes are instantaneous in SUMO, however the parameter *lane-change duration* can be set to change this behaviour. In real world traffic, drivers do not follow the exact speed limit and may choose to drive faster or slower depending on external factors such as road condition and visibility. This behaviour is captured by the *speedFactor* parameter which is the factor by which the drivers interpret and follow the speed limit. The *speedFactor* parameter is accompanied by a *speedDev* parameter which is the standard deviation of drivers from the mean *speedFactor*. Finally, SUMO lets vehicles teleport if they get stuck for a period longer than a given threshold, called *time-to-teleport*. If a vehicle is teleported it is inserted on the next edge (essentially the next available road segment) on its trip.

2.2 Space-mean speed vs time-mean speed

Average speed is one of the most critical parameters in traffic flow theory and the choice of how it is calculated can be detrimental to the results of analysis. The average speed can be represented as a space-mean or a time-mean, where the average is weighted spatially or temporally, respectively.

Mathematically this means that the time-mean speed v_t can be expressed as the arithmetic mean

$$v_t = \frac{1}{n}\sum_{i=1}^{n} v_i \tag{2.1}$$

and after some algebraic manipulations [13] it can be shown that the space-mean speed v_s can expressed as the harmonic mean

$$v_s = \frac{n}{\sum_{i=1}^{n}\frac{1}{v_i}}. \tag{2.2}$$

2.3 Fundamental diagram of traffic flow

In traffic theory, the fundamental diagram (FD) of traffic flow is a diagram showing the relation between traffic density and traffic flow. An illustration of this diagram can be seen in Figure 2.1.

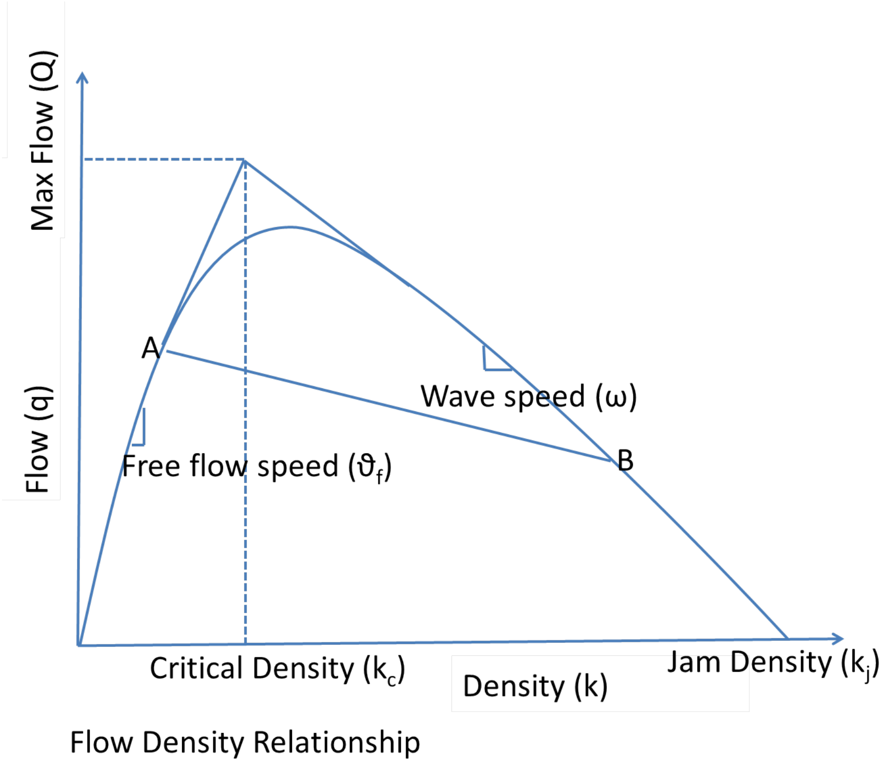


Figure 2.1: Illustration of the fundamental diagram of traffic flow.

Traffic flow q will increase as the density k increases up until it reaches it the maximum flow Q at the critical density k_c. Increasing the density further (i.e. moving to the right of the dotted line in the diagram) will reduce the flow until it reaches zero at the jam density k_j. Flow, density and speed v are related through Equation 2.3.

$$v = \frac{q}{k} \tag{2.3}$$

Many different authors have written about the fundamental diagram dating back to 1969 [14] and more recently, in 2008, verification of its existence on a macroscopic level was published [15].

2.4 Optimization

Optimization is the mathematical field focusing on finding the best (e.g. largest, smallest) solution to a given problem. These problems are usually expressed by an objective function that is to be either maximized or minimized subject to certain constraints. The standard way of formulating such a problem is using *negative null form* as shown in Equation 2.4.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0,\ i = 1, ..., m \\ & h_j(x) = 0,\ j = 1, ..., p \end{aligned} \tag{2.4}$$

In this formulation $m \geq 0$ & $p \geq 0$, maximization problems are handled by setting either $1/f(x)$ or $-f(x)$ as the objective function and x can be a vector of several variables.

If $f(x)$ can be expressed in analytical form and meet certain other criteria such as continuity and differentiability the problem can be solved either analytically or numerically by utilizing the function gradient $\nabla f(x)$ to find the minimum. Unfortunately, when optimization is used in this thesis the objective function is described, in part, by terms that can only be generated by running a simulation.

To deal with this, a different type of optimization that is not contingent on knowing the gradient must be used. This type of optimization is called *Black-box optimization* or *derivative-free optimization* and the specific optimization package that is implemented in this project is called *RBFOpt* [16].

2.4.1 Objective function for fitting simulation to data

A major part of this thesis is tuning the simulation environment to match real data using optimization. This raises the question as to what it means for the simulation and reality to match. Since traffic is often described by the fundamental diagram (as described in Section 2.3), it was decided that the goal would be to have the fundamental diagram scatter plots of real traffic and simulated traffic be as similar as possible.

2.4.1.1 Distance

Two datasets each containing points (k, q, t), i.e. density, flow and time, for both the real traffic data and the simulated traffic data can be compared on a point-by-point basis by calculating the distance in the kq-plane for each point in time. Important to note here is that because flow and density should be equally important in the optimization it is necessary to scale both k and q to be in the range [0, 1] by dividing by the max value for each variable. After scaling the parameters, the distance d can be found by simply using the distance formula in Equation 2.5.

$$d = \sqrt{\left(\frac{k_i - \hat{k}_i}{\hat{k}_{max}}\right)^2 + \left(\frac{q_i - \hat{q}_i}{\hat{q}_{max}}\right)^2} \tag{2.5}$$

2.4.1.2 Error

The distance d is now the error at each point in time t and calculating this value for all t gives a vector of errors. There are several valid ways of summarizing this error vector into a single error value E, but in this case the *root mean square error* (*RMSE*) (Equation 2.6) was chosen.

$$E = \sqrt{\sum_{i=1}^{n} \frac{d_i^2}{n}} \tag{2.6}$$

Substituting in d and cancelling the square and square root gives the complete expression for E in Equation 2.7.

$$E = \sqrt{\sum_{i=1}^{n} \frac{\left(\frac{k_i - \hat{k}_i}{\hat{k}_{max}}\right)^2 + \left(\frac{q_i - \hat{q}_i}{\hat{q}_{max}}\right)^2}{n}} \tag{2.7}$$

2.4.1.3 Weighting

The error expression presented in Equation 2.7 is reasonable from a mathematical standpoint but in reality, it turns out that the distribution of points along the k-axis is very non-uniform in the chosen location. The result of this is that by using a the standard *RMSE* the optimization will almost exclusively focus on fitting the cluster of points where k is small (i.e. free flow) and not consider the sparse points where the k is large. For this reason, a modified version of the *RMSE* was used, a *weighted root mean square error.*

The points are divided into equal-width bins by binning along the k-axis. The number of bins B was chosen based on the largest number of bins that would ensure that no bin was empty. Each bin is assigned a weight based on the number of points in that bin and the sum of all weights add up to 1 to keep the magnitude of the final value the about the same as without weights. The error E is then calculated as shown in Equation 2.8.

$$E = \sqrt{\sum_{i=1}^{n} w_i \left(\left(\frac{k_i - \hat{k}_i}{\hat{k}_{max}}\right)^2 + \left(\frac{q_i - \hat{q}_i}{\hat{q}_{max}}\right)^2 \right)} \tag{2.8}$$

It should be noted that the sum is being calculated rather than the mean and this is because the sum of all weights w is already scaled to add up to 1. Calculating w comes down to solving a system of equations, satisfying Equations 2.9, 2.10 and 2.11.

$$\sum_{i=1}^{l} w_i = B \tag{2.9}$$

$$\frac{w_i}{w_{i+1}} = \frac{b_{i+1}}{b_i}, i = 1, ..., B \tag{2.10}$$

$$\frac{w_B}{w_1} = \frac{b_1}{b_B} \tag{2.11}$$

Where b is the number of points in a given bin. Solving this system explicitly gives the solution.

$$w_i = \frac{1}{Bb_i}, i = 1, ..., B \tag{2.12}$$

As can be seen by Equation 2.12, $b \neq 0$. Substituting w into Equation 2.8 gives Equation 2.13.

$$E = \sqrt{\sum_{i=1}^{n} \frac{1}{Bb_i} \left(\left(\frac{k_i - \hat{k}_i}{\hat{k}_{max}} \right)^2 + \left(\frac{q_i - \hat{q}_i}{\hat{q}_{max}} \right)^2 \right)_j} \tag{2.13}$$

Finally, since there are a number of different detectors being considered the final step is to simply average these detectors, as shown in Equation 2.14.

$$E_{tot} = \frac{1}{m} \sum_{j=1}^{m} E_j \tag{2.14}$$

Substituting in E into Equation 2.14 gives the final objective function, see Equation 2.15.

$$E_{tot} = \frac{1}{m} \sum_{j=1}^{m} \sqrt{\sum_{i=1}^{n} \frac{1}{Bb_i} \left(\left(\frac{k_i - \hat{k}_i}{\hat{k}_{max}} \right)^2 + \left(\frac{q_i - \hat{q}_i}{\hat{q}_{max}} \right)^2 \right)_j} \tag{2.15}$$

2.4.2 Objective function for improving traffic

Fitting the simulation to real data is not the only objective of this thesis, in fact it is only a means to investigate how traffic can be improved with the introduction of autonomous vehicles. This requires a metric for describing how well the traffic flows. There are many different proposed ways of doing this ranging from simple ones such as *critical density* and *critical flow* to more sophisticated ones like the *Road Congestion Index* [17] and the *Road Segment Congestion Index* [18]. In the end this project utilizes the *Speed Performance Index* (R_v), see Equation 2.16, which is the basis for the *Road Segment Congestion Index.*

$$R_v = \frac{v}{V_{max}} \times 100 \tag{2.16}$$

Where v is current speed and V_{max} is the speed limit. This means that R_v essentially describes at what per centage of the speed limit drivers are driving at any given moment. If traffic is uncongested R_v will be around 100 and if it is congested it may be significantly lower.

Implementing the mean of the *Speed Performance Index* on one lane gives Equation 2.17.

$$R_{lane} = \frac{1}{n}\sum_{i=1}^{n} R_{v_i} \tag{2.17}$$

Inserting the expression for R_v into Equation 2.17 and rearranging slightly gives Equation 2.18.

$$R_{lane} = \frac{100}{n}\sum_{i=1}^{n} \frac{v_i}{V_{max}} \tag{2.18}$$

Finally, taking the mean over all the lanes gives Equation 2.19.

$$R_{tot} = \frac{1}{m}\sum_{j=1}^{m} \left(\frac{100}{n}\sum_{i=1}^{n} \frac{v_i}{V_{max}}\right)_j \tag{2.19}$$

2.4.3 Optimization algorithm and global search method

As previously discussed, this project implements the *RBFOpt* library which contains two optimization algorithms and three global search methods. The optimization algorithms are:

1. Gutmann's Radial Basis Function (*RBF*) [16]
2. Metric Stochastic Response Surface Method (*MSRSM*) [19]

Further explanations on how these algorithms work is beyond the scope of this report and full derivations can be read at the sources cited for *RBF* and *MSRSM* respectively in the list above. The global search methods are:

1. Genetic
2. Sampling
3. Solver

Again, full explanations of these methods are beyond the scope of the report, but very brief explanations will be given. Genetic utilizes the principal of survival of the fittest by spawning a population and letting the best performing ones survive. Sampling simply samples several points and picks the best ones. Solver finds the next points by finding a local minimum of the response model, however this is not very effective when the real model is extremely non-linear and non-continuous.

All combinations of optimization algorithms and global search methods are possible but not all have great performance, either from a results or iteration time perspective. For example, *Solver* has poor performance with *MSRSM* and Gutmann's RBF performs best with solver unless iteration speed is an issue.

3
Method

In this chapter, the methods used in the project are presented. A flowchart describing the process can be seen in Figure 1.1.

3.1 Data preparation

Detector data used for this project was provided by *Trafikverket* and had to be extensively processed before it could be used. Because of the sheer amount of data (for reference, the total data provided, had 21 columns and over 600 000 rows in csv-format), a visual tool like *Excel* would not be able to open the files. Instead, the data was processed using the *pandas* library in *Python*.

The data had to be filtered a lot in order to extract the information that was relevant to the project. Filtering the data made it possible to search for a location (as described in Section 3.2) easily.

3.2 Location

The basis for the analysis conducted in this project is the modelling of a real-world location and the subsequent calibration of the model to match detector data. Because of this, choosing a good location is a crucial part of project.

There are certain criteria for what constitutes a good location in regard to this project, of which the first is to have all in- and outflows covered by detectors. If this is the case, the location can be modelled without having to guess critical parameters. Many different types of data from detectors can be used depending on availability and calibration strategy. Examples of such data are speed, flow, occupancy, gap and headway.

If detector coverage is sufficient, both spatially (as discussed above) and temporally (i.e. the detectors worked throughout the sample period), the location should experience both free-flow and some level of congestion. Finally, the location should general enough that conclusions drawn from the analysis can at least in some respect be applicable elsewhere.

3.3 Dynamic traffic assignment

Even with detector coverage at all in- and outflows of the studied location there is one crucial piece of information that is not known, the origin-destination (OD) matrix. This matrix contains information about the paths (or at least in- and outflow locations) are being taken during the studied time period. This information is hardly ever known from data since it requires the tracking of each individual vehicle, instead it must be estimated by Dynamic Traffic Assignment (DTA).

Different methods of estimating OD matrices have been proposed [20][21] but in this project it was found by using the SUMO tool *flowrouter* [22]. *Flowrouter* is a Python script that finds the OD matrix by solving a maximum flow problem and setting the measured flows as the maximum flow capacity for the given time period. This can be aggregated over the whole time period studied or over shorter intervals (e.g. 1 min, 5 min or 1 hour).

3.4 Optimization

In order to make the simulation match reality, an optimization (in this case a minimization) of the objective function presented in Equation 2.15 had to be conducted. With this formulation in mind, the constants of this particular optimization can be inserted.

First, there are four detectors on four adjacent lanes that are of interest in the optimazation. This means that in the expression for the objective function $m = 4$. The simulation is four hours, or 240 minutes, long. The data is aggregated into 10 minute intervals meaning that there are $240/10 = 24$ data points per detector, which means that $n = 24$. Each detector is binned horizontally along the density axis into five bins, i.e. $B = 5$. Finally, the number of points per bin b_i, the maximum density $\hat{k}_{max}$ and the maximum flow $\hat{q}_{max}$ are all different per detector or bin. Putting the constant values into Equation 2.15 gives Equation 3.1.

$$E_{tot} = \frac{1}{4}\sum_{j=1}^{4}\sqrt{\sum_{i=1}^{24}\frac{1}{5b_i}\left(\left(\frac{k_i - \hat{k}_i}{\hat{k}_{max}}\right)^2 + \left(\frac{q_i - \hat{q}_i}{\hat{q}_{max}}\right)^2\right)_j} \tag{3.1}$$

Finding the remaining values k_i and q_i for each time interval and detector comes down to running the simulation and extracting the data. For this it is necessary to decide which input parameters should be input.

3.4.1 Optimization parameters

For the optimization to reach a result it needs to be able to alter the input to the simulation on each iteration and evaluate the results. The input parameters available are described in Section 2.1.

Not all available parameters were used as input to the optimization and the ones that were not used were simply using the default values instead. The reasons for not including every possible input parameter was the optimization time and complexity it would incur, with no particularly large upside. The most important parameters were chosen and are presented in Table 3.1 along with their limits. Each limit will not be justified individually but they were all picked based on the simulation or physical properties applicable. Description of the parameters can be found in Section 2.1.

Parameter	**Range**	**Distribution type**	**Unit**
accel (mean)	0.5-1.5	Normal	m/s^2
accel (std)	0.05-0.5	Normal	m/s^2
decel (mean)	4.0-6.0	Normal	m/s^2
decel (std)	0.1-0.8	Normal	m/s^2
sigma (mean)	0.01-1.0	Normal	N/A
sigma (std)	0.05-0.2	Normal	N/A
tau (alpha)	8.0-36.0	Gamma	$s*$
tau (beta)	8.0-60.0	Gamma	$s*$
minGap (mean)	1.5-4.0	Normal	m
minGap (std)	0.1-1.0	Normal	m
lcStrategic (mean)	0.01-10.0	Normal	N/A
lcStrategic (std)	0.05-2.0	Normal	N/A
lcCooperative (mean)	0.01-1.0	Normal	N/A
lcCooperative (std)	0.05-0.01	Normal	N/A
speedFactor (mean)	0.7-1.3	Normal	N/A
speedFactor (std)	0.1-0.3	Normal	N/A
time-to-teleport	5.0-300.0	Fixed value	s
lanchange-duration	1.0-5.0	Fixed value	s

Table 3.1: Parameters used in optimzation.

3.5 Sensitivity Analysis

While it is obvious that not all the parameters listed in Table 3.1 are equally important to the result, it is not as obvious which parameters have the biggest impact on the outcome.

To figure out which parameters were the biggest and smallest contributors respectively, a sensitivity analysis was used. The setup of the analysis was that each parameter would be independently varied, the simulation would be run and the result of the objective function would be noted. Each parameter would be evaluated at four different levels: ±10% and ±25%.

3.6 Introduction of Autonomous Vehicles

One of the main points of the thesis is the evaluation of traffic when different fractions of autonomous vehicles are introduced into regular traffic. The outcome of the simulations are compared based on the objective function presented in Equation 2.19. There are four lanes ($m = 4$), the speed limit is 70 $km/h \approx 19.44$ m/s on all lanes ($V_{max} = 19.44$) and just as in Equation 3.1 there are 24 data points per detector ($n = 24$). This results in Equation 3.2.

$$R_{tot} = \frac{1}{4} \sum_{j=1}^{4} \left(\frac{100}{24} \sum_{i=1}^{24} \frac{v_i}{19.44} \right)_j \tag{3.2}$$

This can be simplified to Equation 3.3.

$$R_{tot} \approx 0.0536 \sum_{j=1}^{4} \left(\sum_{i=1}^{24} v_i \right)_j \tag{3.3}$$

This objective function was used to optimize the behaviour of autonomous vehicles at different fractions of AVs (0%, 10%, 25%, 50%, 100%). The controller parameters in Table 2.2 were *not* used as input parameters to the optimization because changing them can cause odd dynamics behaviour and they are already tuned beforehand. Instead their default values were used according to Table 3.2.

Parameter	**Value**
speedControlGainCACC	-0.4
gapClosingControlGainGap	0.005
gapClosingControlGainGapDot	0.05
gapControlGainGap	0.45
gapControlGainGapDot	0.0125
collisionAvoidanceGainGap	0.45
collisionAvoidanceGainGapDot	0.05

Table 3.2: CACC default parameters used.

Instead, the parameters that are shared between the *SUMOKrauß* and *CACC* are used as input parameters to the optimization. For autonomous vehicles all parameters are fixed values since they are not expected to have any variation in their behaviour and for the same reason the driver imperfection is set to 0. The parameters and their ranges can be seen in Table 3.3.

Parameter	**Range**	**Unit**
accel	0.5-1.5	m/s^2
decel	4.0-6.0	m/s^2
tau	0.5-3.0	s
minGap	1.5-4.0	m/s^2
lcStrategic	0.01-10.0	N/A
lcCooperative	0.01-1.0	N/A

Table 3.3: Parameters used in AV optimzation.

4
Results

4.1 Location

The process of finding and deciding upon a location was a process of elimination based on the criteria presented in Section 3.2 and the set of possible locations was defined by the detector data provided by Trafikverket. Initially, the geographical location of the detectors was provided as a *shapefile (.shp)* and a quick conversion to *Keyhole Markup Language (.kml)* allowed the detectors to be overlaid on *Google Maps* as shown in Figure 4.1.

After carefully looking through all possible locations, three candidates were found. The candidate locations all had in- and outflows covered with detectors and were known to experience a decent amount of traffic. The first location is located along *Lundbyleden* and is shown as a map and a satellite image in Figures 4.2 and 4.3 respectively.

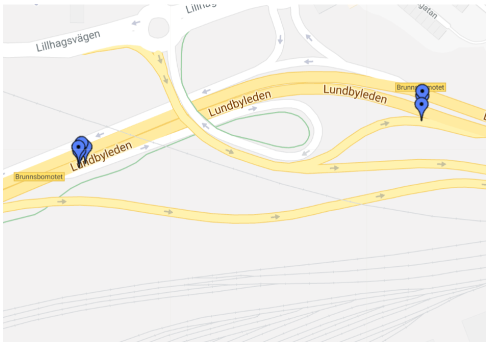


Figure 4.2: Map view of location candidate 1.

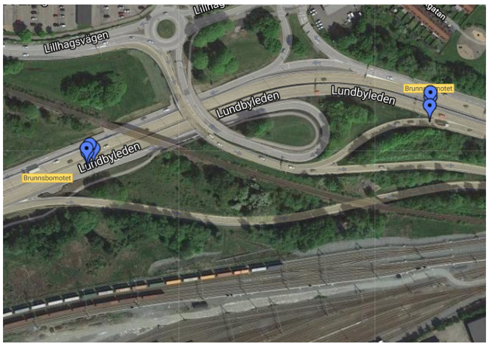


Figure 4.3: Satellite view of location candidate 1.

The second location candidate is relatively close to the first one at the entrance to *Tingstadstunneln* and is shown the same way as location 1 in Figures 4.4 and 4.5.

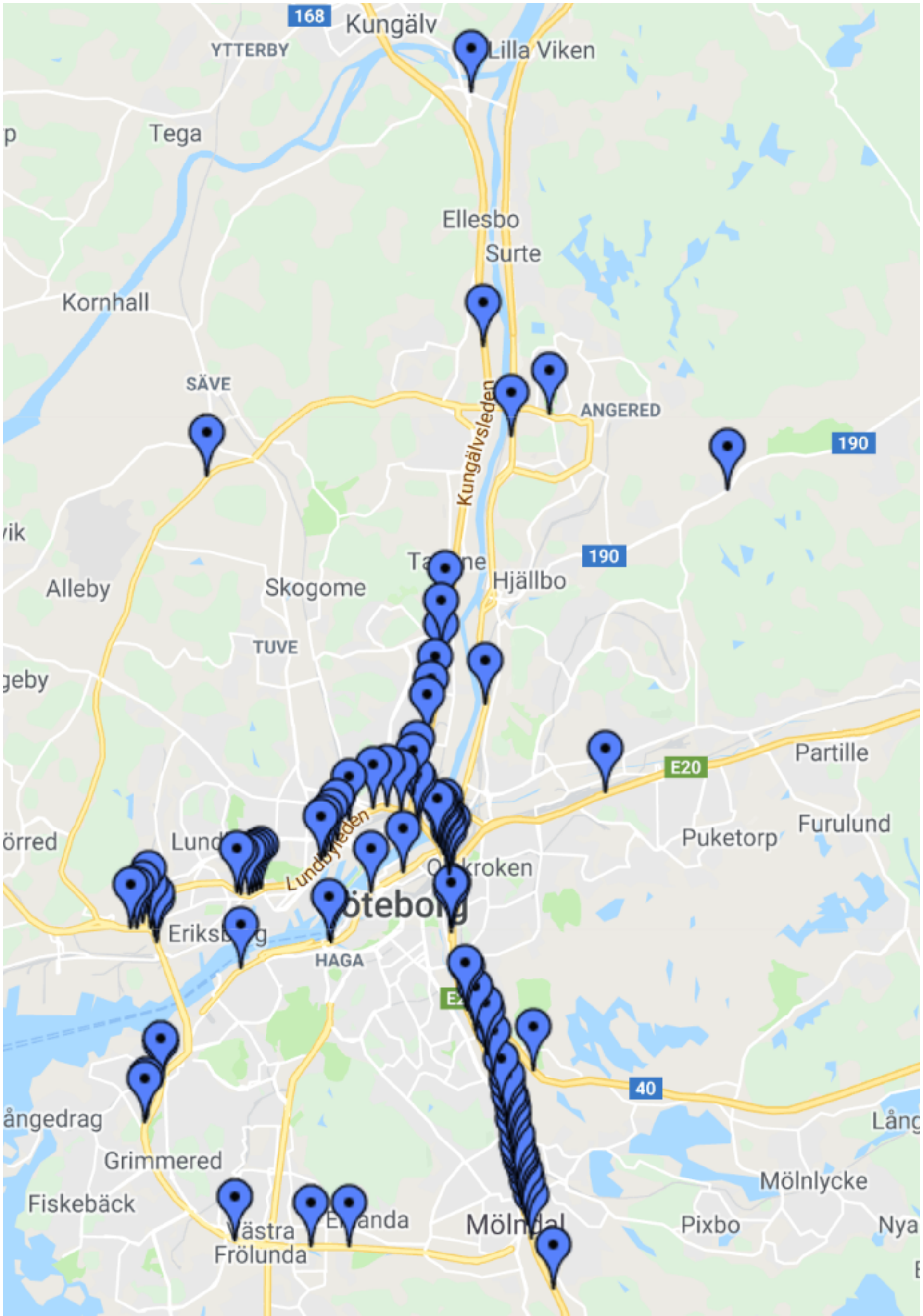


Figure 4.1: All detectors in Gothenburg overlaid on Google Maps.

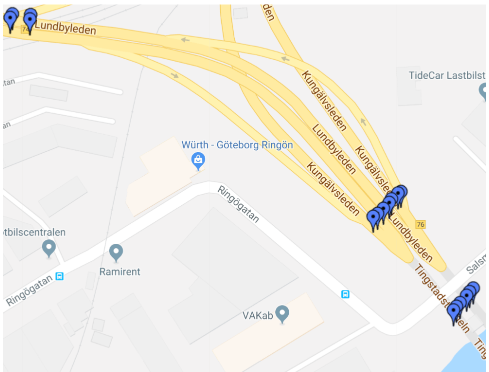


Figure 4.4: Map view of location candidate 2.

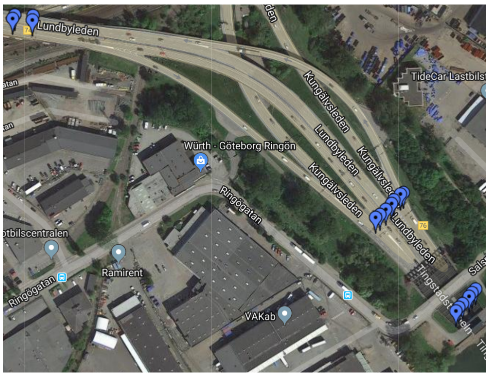


Figure 4.5: Satellite view of location candidate 2.

The third and final location candidate is located on the opposite side of *Tingstadstunneln* along *Kungsbackaleden* and is shown the same way as the previous locations in Figures 4.6 and 4.7.

After inspecting the data available for the detectors in these locations it was concluded that location 1 and 2 did not have sufficient data coverage and that location 3 had almost perfect coverage. The only problem with location 3 was that a detector on one of the lanes would randomly throughout the day stop registering the flow. This was solved by sorting the data for periods when the detector had full coverage.

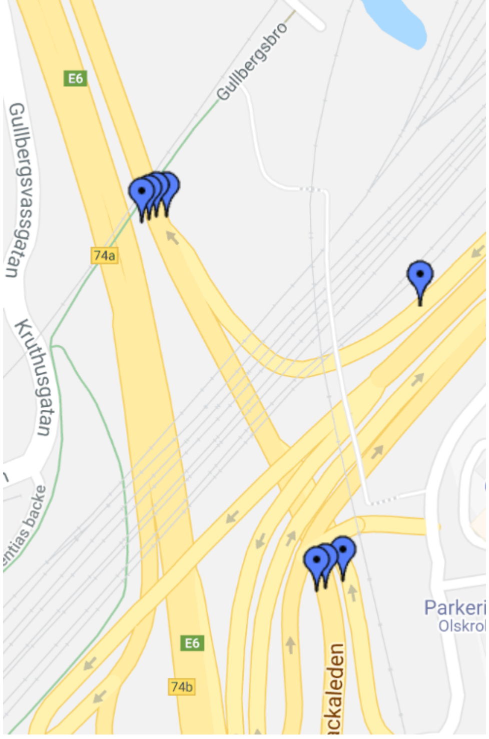


Figure 4.6: Map view of location candidate 3.

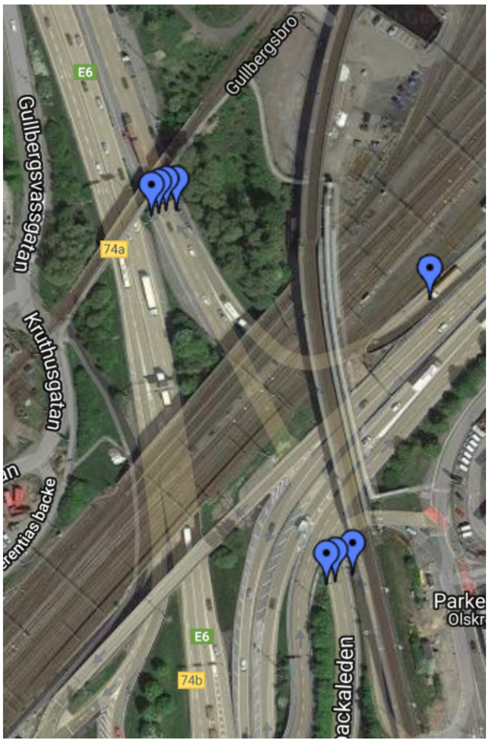


Figure 4.7: Satellite view of location candidate 3.

4.1.1 Location in Sumo

In Sumo, it is possible to import maps directly from OpenStreetMap [23]. After removing all surrounding bike lanes, train tracks, walkways and roads that are not relevant to the simulation, the model location for the simulation is finished, see Figure 4.8.

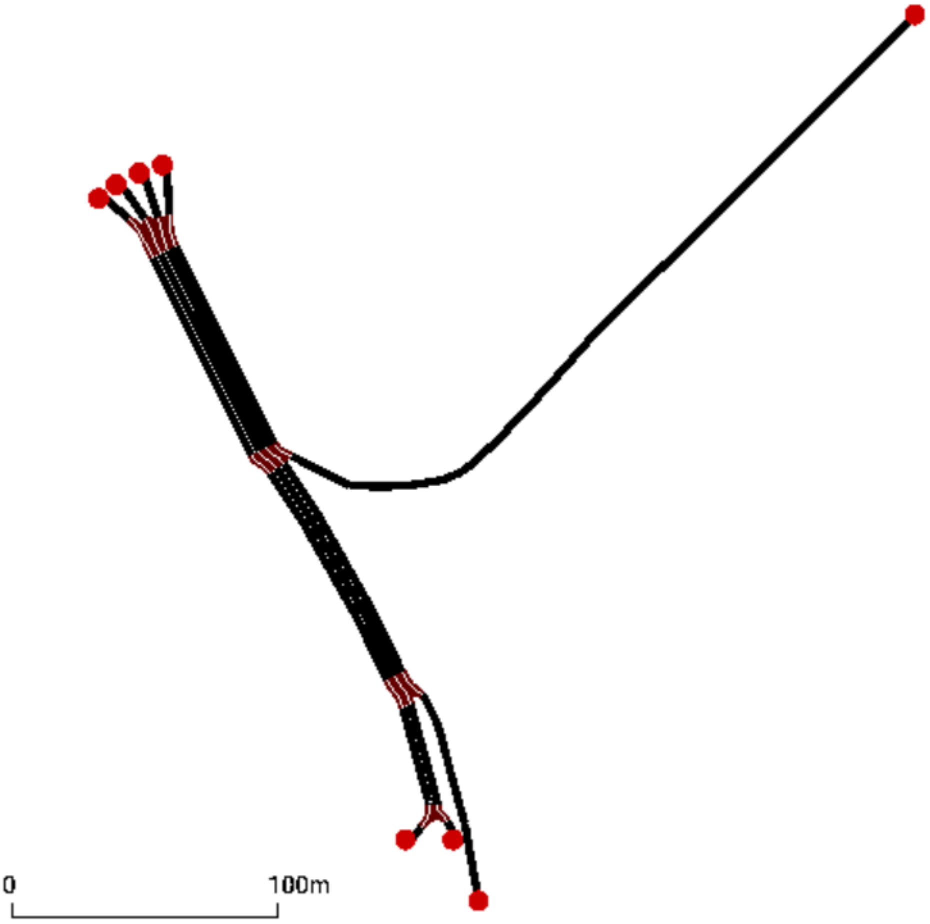


Figure 4.8: Sumo model of Kungsbackaleden.

The lanes are split at the in and outflows to make it more obvious where vehicles are flowing in and out of the system while viewing the simulation graphically. The effect this has one the simulation is completely negligible, and it only serves as visual aid.

An important note about the speed limit is that the real speed limits were not used in the simulation and instead it was adjusted to match the speed that drivers were going in free flow, as collected from detector data. This meant that all the lanes essentially had their true speed limit except for the lane coming in from the far right, which in reality has a speed limit much higher than what drivers choose to drive. There are a few factors for this including the rather sharp turn radius and that there is an overpass that the entering vehicles drive under.

4.2 Dynamic traffic assignemnt

Because the data obtained from Trafikverket for analysis is aggregated every minute it was decided that the DTA should be aggregated every minute as well, to get as close of a match to reality as possible.

Unfortunately, the *flowrouter* script tends to underestimate the flow there is high fluctuation in the data, which often happens when using a small aggregation time like 1 minute. In order to adjust for this a separate script was written that checks and makes sure that all in- and outflows exactly match the collected detector data and assigns routing based on best guesses by looking at the difference between flows in real detectors and simulated detectors.

4.3 Simulation results

The result of the simulations will be presented graphically as fundamental diagram plots (*density-flow plots*, see Section 2.3) where data is aggregated to 10 min intervals, meaning that each point in the plot represents a 10 min interval. The raw data both in the simulation and the real detectors is collected as speed and flow in 1 min aggregated intervals and the speed is further averaged to 10 min by using *time-mean speed* because the raw data from *Trafikverket* used this averaging method. While *space-mean speed* might have been a better representation, it was still more important to remain consistent throughout.

The real and simulated data are overlaid on the same plot with real data being blue and simulated data being red. This holds true for all plots where real and simulated data are plotted against each other.

4.3.1 Default parameter simulation

To give a reference for the improvement that the optimization contributes, a simulation was run with the all default parameters. This simulation still used the correct in- and outflows and the modelled location, but regarding driver behaviour was set to the defaults.

The simulation model is equipped with detectors in the same positions as the real detectors, see Figures 4.6 and 4.7. The detectors that can be seen in the plots are detectors with IDs 35138, 35139, 35140 and 35141 which are the detectors on the four adjacent lanes circled in Figure 4.9, starting from the left.

Now, using the model above, Figures 4.10, 4.11, 4.12 and 4.13. shows the fundamental diagram plot of real and simulated data with default parameters for the four detectors in Figure 4.9. Detector 35141 is the least interesting since that detector is placed right as the inflow meets the rest of the traffic and thus have not had much of a chance to interact with other vehicles. It is still included since the four lanes together cover the entire road segment.

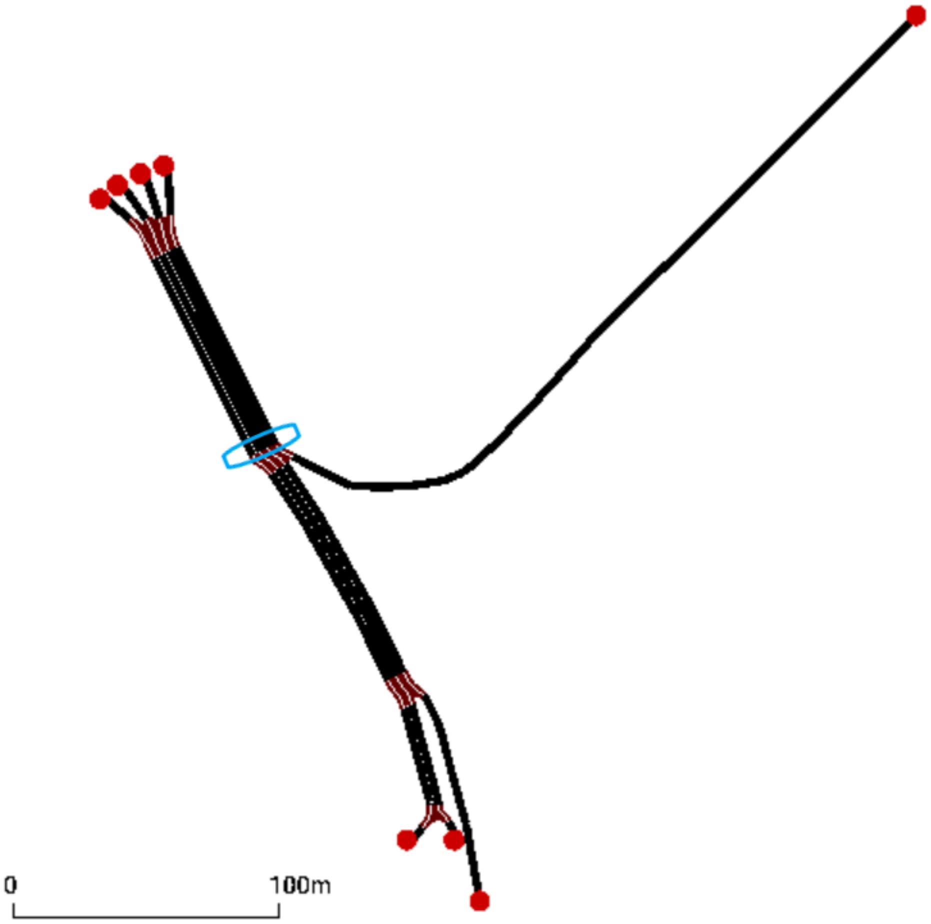


Figure 4.9: Sumo model of Kungsbackaleden with detectors circled.

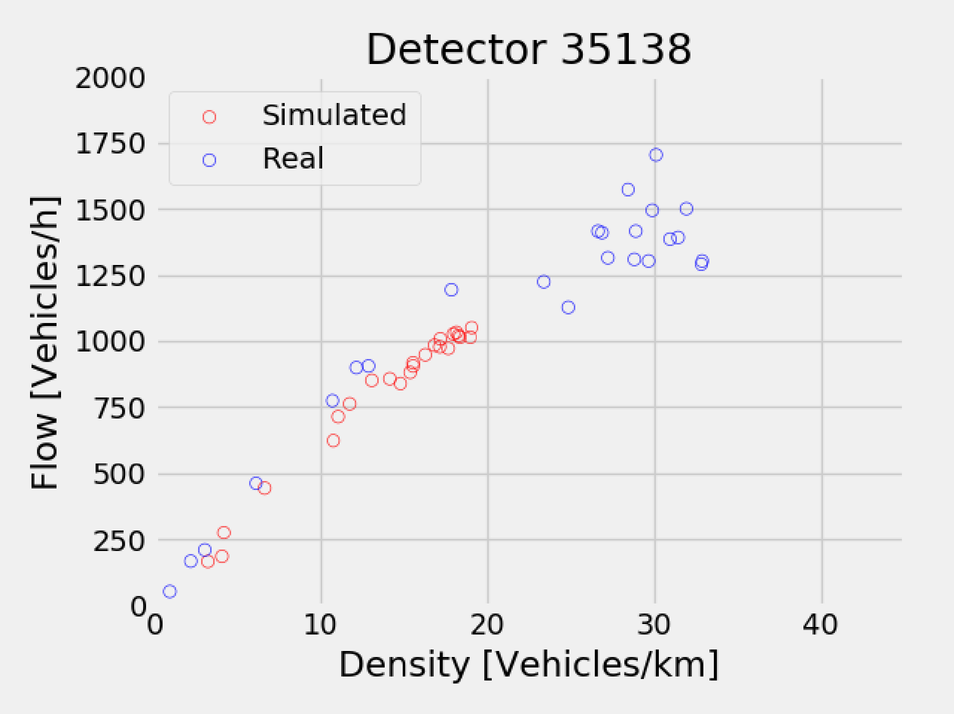


Figure 4.10: Default parameter simulation output for detector 35138.

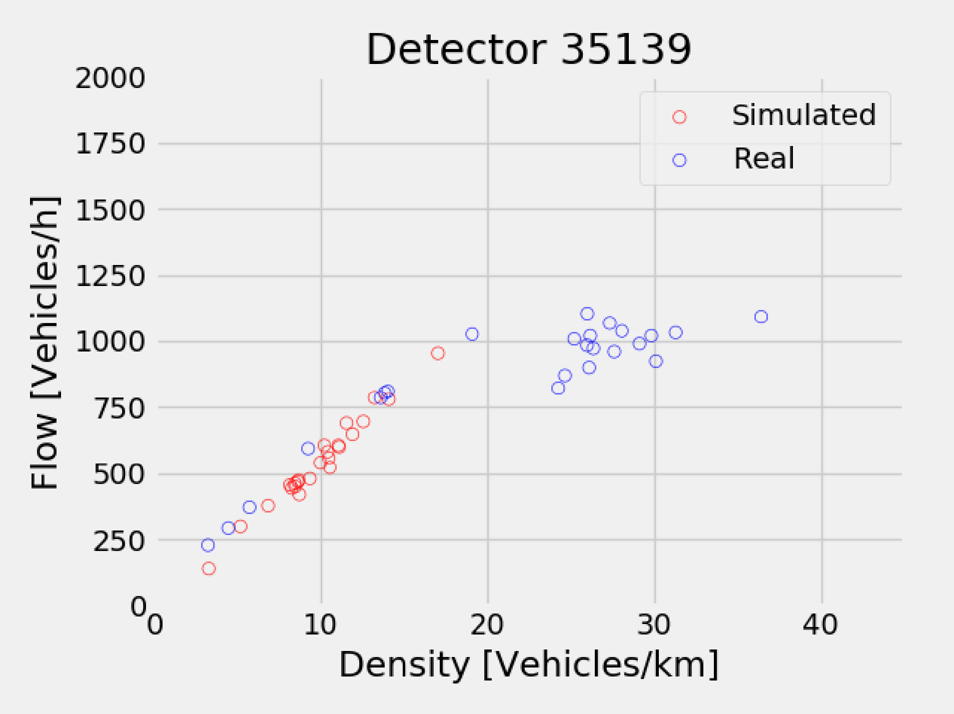


Figure 4.11: Default parameter simulation output for detector 35139.

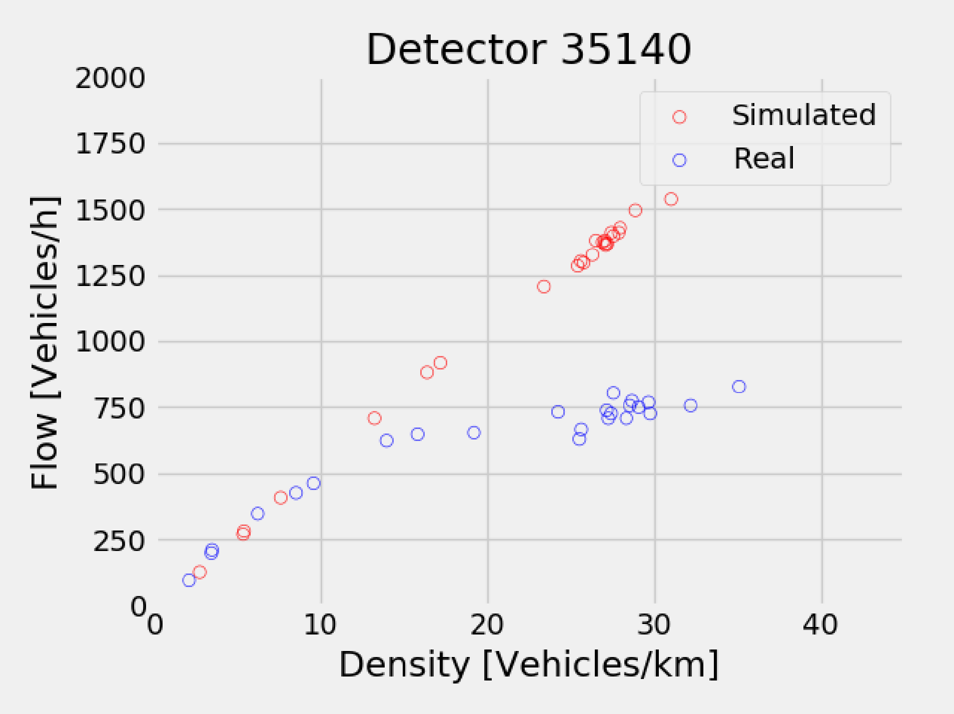


Figure 4.12: Default parameter simulation output for detector 35140.

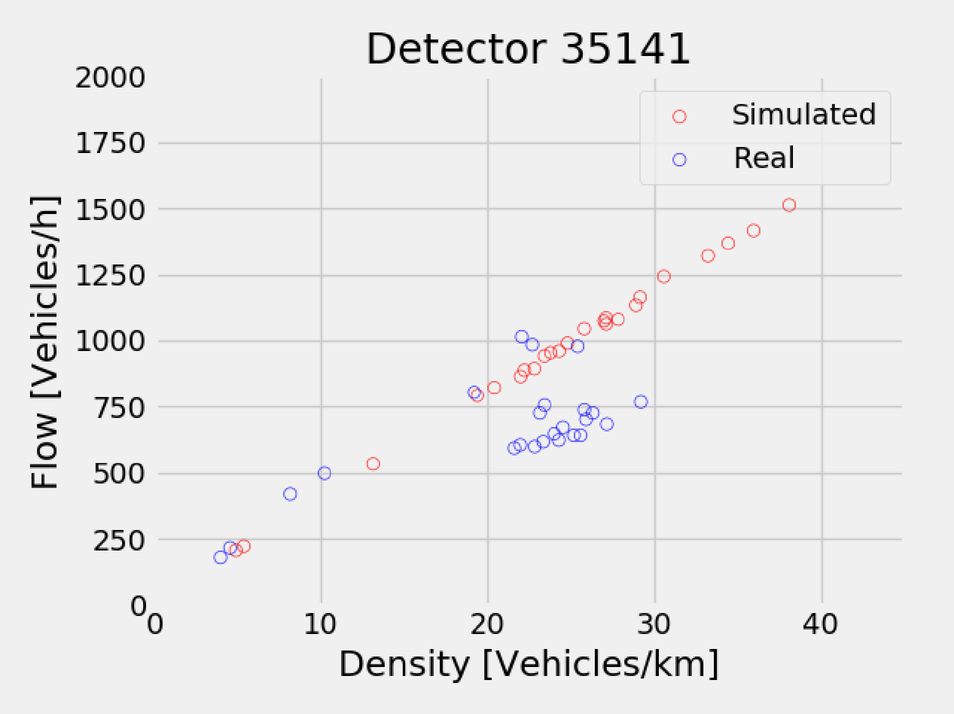


Figure 4.13: Default parameter simulation output for detector 35141.

Each detector (in simulation and reality) collects *flow* and *speed*, which can also be used to calculate the density. These three parameters can be plotted over the four-hour interval to show how simulation and reality line up in a temporal sense.

Figures 4.14, 4.15 and 4.16 shows combined pictures of all detectors, but individual pictures can be seen in Appendix A.

4.4 Optimization for fitting simulation to data

While the default simulation is not a horrible match with reality, it can be improved significantly. This section describes the optimization process and progress.

4.4.1 Optimization setup

The parameters in Table 3.1 were used as the input variables that the optimization can alter. Other parameters were left at the default settings except for the simulation time-step which was changed from the default of 1 second down to 0.25 seconds to avoid collisions when the time gap was below 1 second.

The different combinations of optimization algorithms and global search methods discussed in Section 2.4.3 were tested and it was concluded that *MSRSM* in combination with *Genetic* yielded the best results in this particular case.

4.4.2 Optimization progress

The optimization initializes by trying out several points spread out over the input-space and evaluating them. After this the optimization continuously reports its actions on each iteration back to the user. Figure 4.17 shows the current best value found for the objective function plotted against the iteration number. The red dot directly above the final value shows the value of the objective function evaluated with the default parameters.

Note: The optimization was allowed to run for more than another 300 iterations without finding a new optimum. The optimized parameters were also checked against the afternoon of the same day to make sure they were not overfitted to the particular dataset.

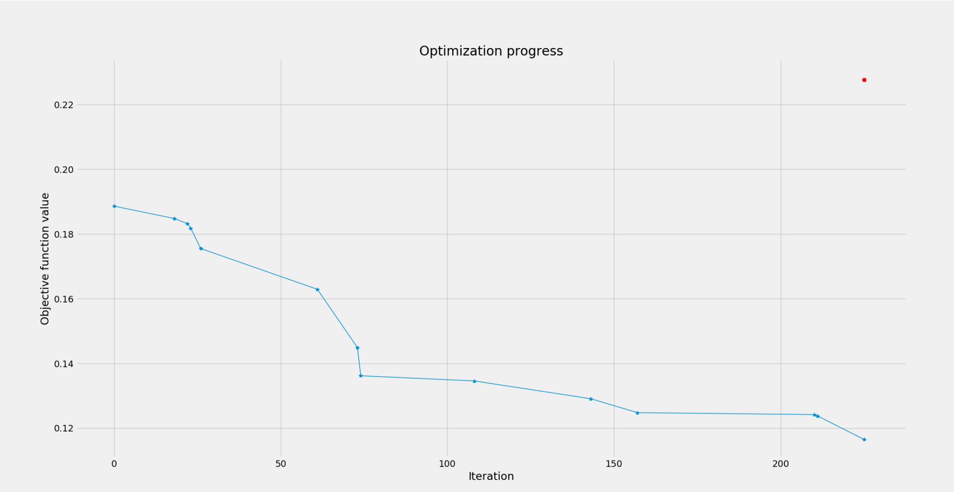


Figure 4.17: Progress of the optimization.

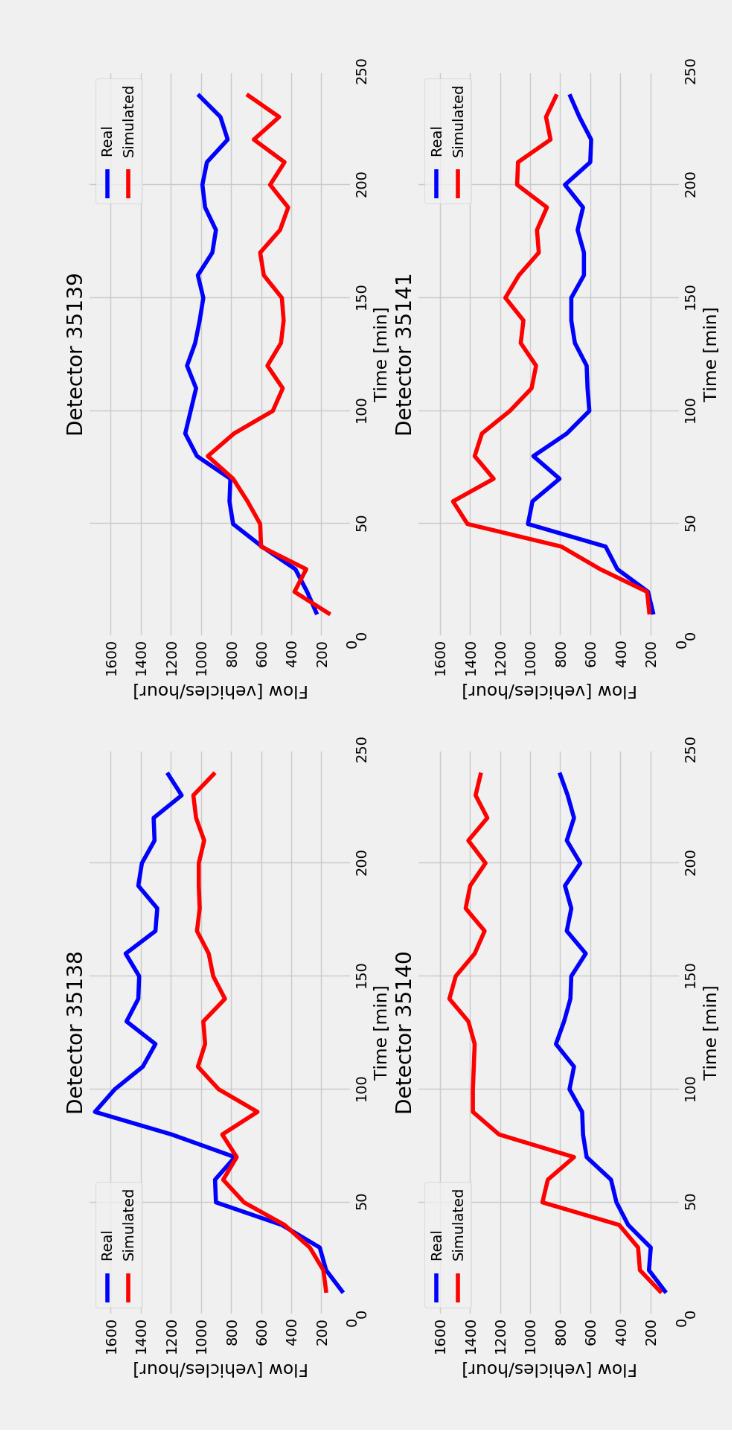


Figure 4.14: Flow over time with default parameters.

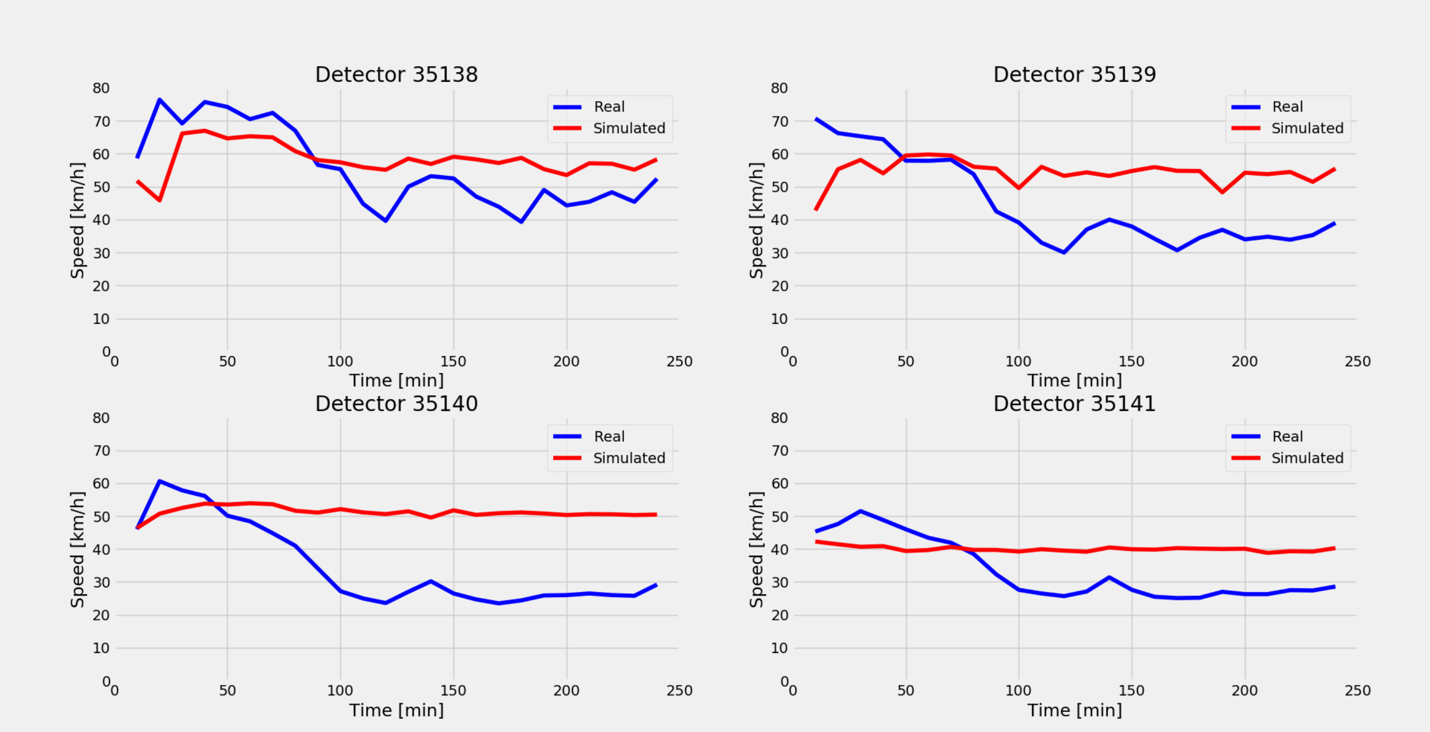


Figure 4.15: Speed over time with default parameters.

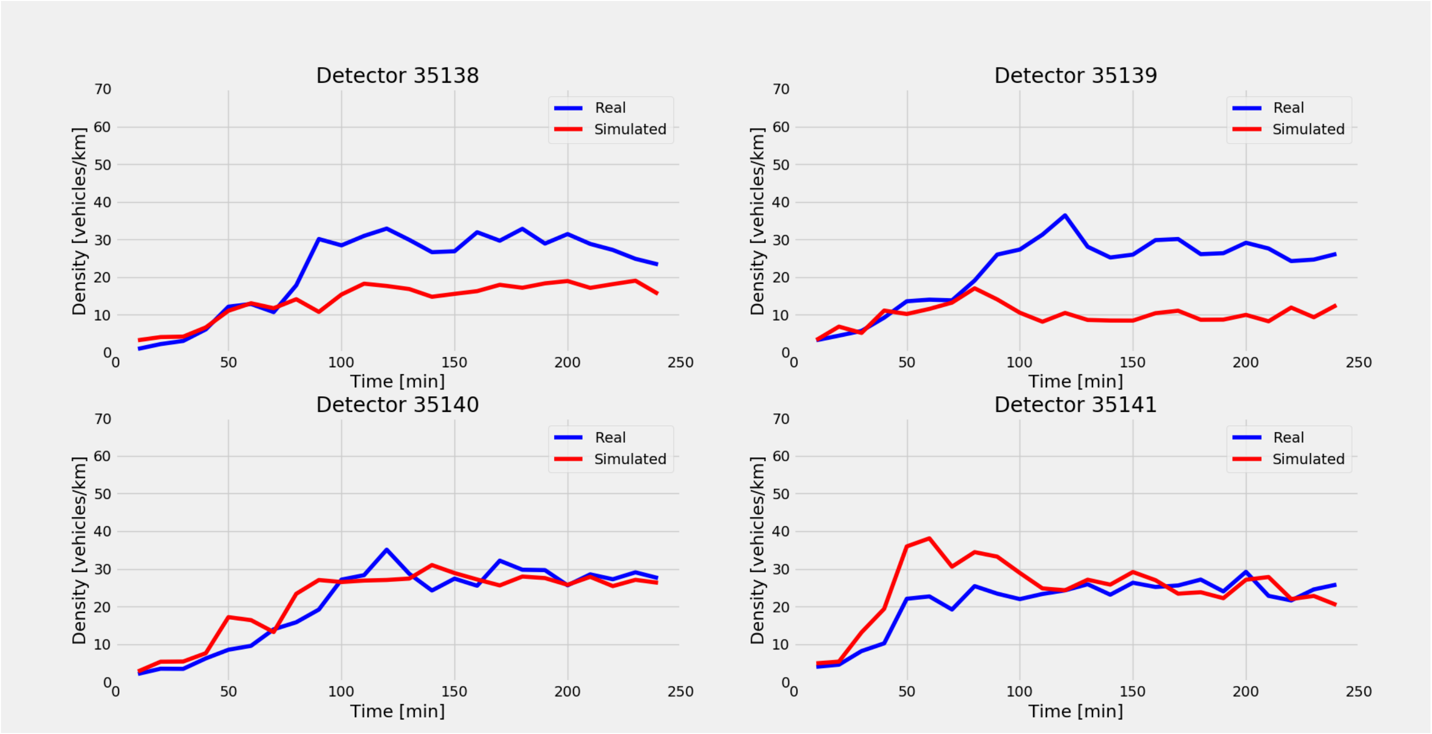


Figure 4.16: Density over time with default parameters.

4.4.3 Parameters optimized to match real data

The best parameters found by the optimization are presented in Table 4.1 (all values rounded to 4 decimal places).

Parameter	**Value**	**Unit**
accel (mean)	1.4976	m/s^2
accel (std)	0.0555	m/s^2
decel (mean)	4.0522	m/s^2
decel (std)	0.9979	m/s^2
sigma (mean)	0.7954	N/A
sigma (std)	0.1615	N/A
tau (alpha)	33.6166	$s*$
tau (beta)	40.6236	$s*$
minGap (mean)	1.5401	m
minGap (std)	0.2188	m
lcStrategic (mean)	0.0122	N/A
lcStrategic (std)	1.6575	N/A
lcCooperative (mean)	0.9978	N/A
lcCooperative (std)	0.1	N/A
speedFactor (mean)	1.2081	N/A
speedFactor (std)	0.1425	N/A
time-to-teleport	181.0	s
lanchange-duration	1.1362	s

Table 4.1: Parameters used in optimzation

4.4.4 Visualizing the result

As can be seen from Figure 4.17, the objective function was more than cut in half. The next step is to see it visually and the Figures 4.18, 4.19, 4.20 and 4.21 show the resulting fundamental diagram plots for detectors 35138, 35139, 35140 and 35141.

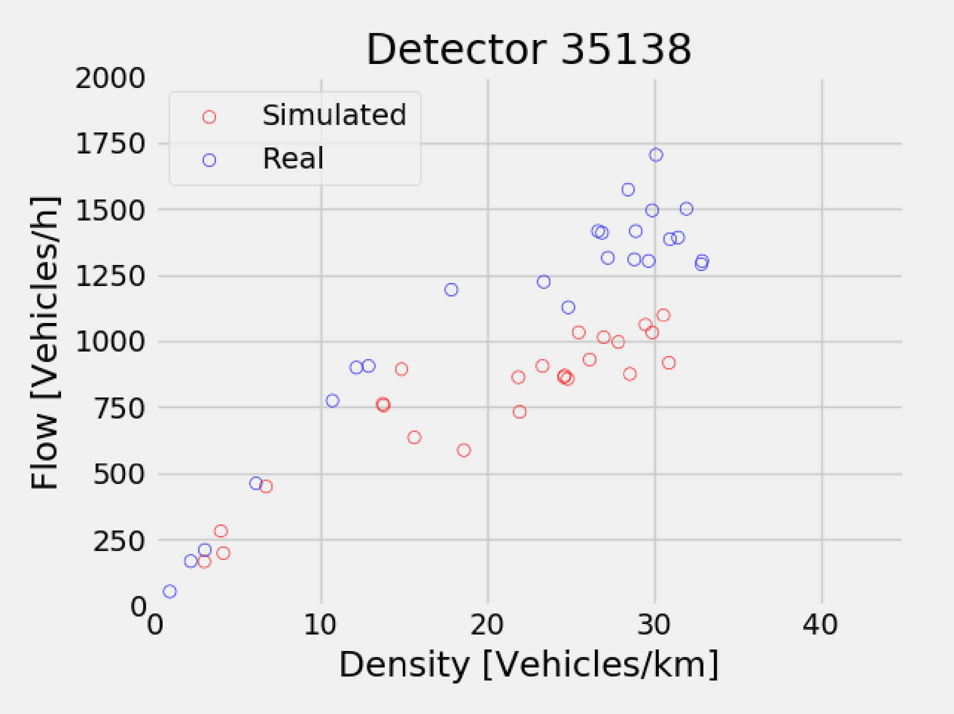


Figure 4.18: Optimized parameter simulation output for detector 35138.

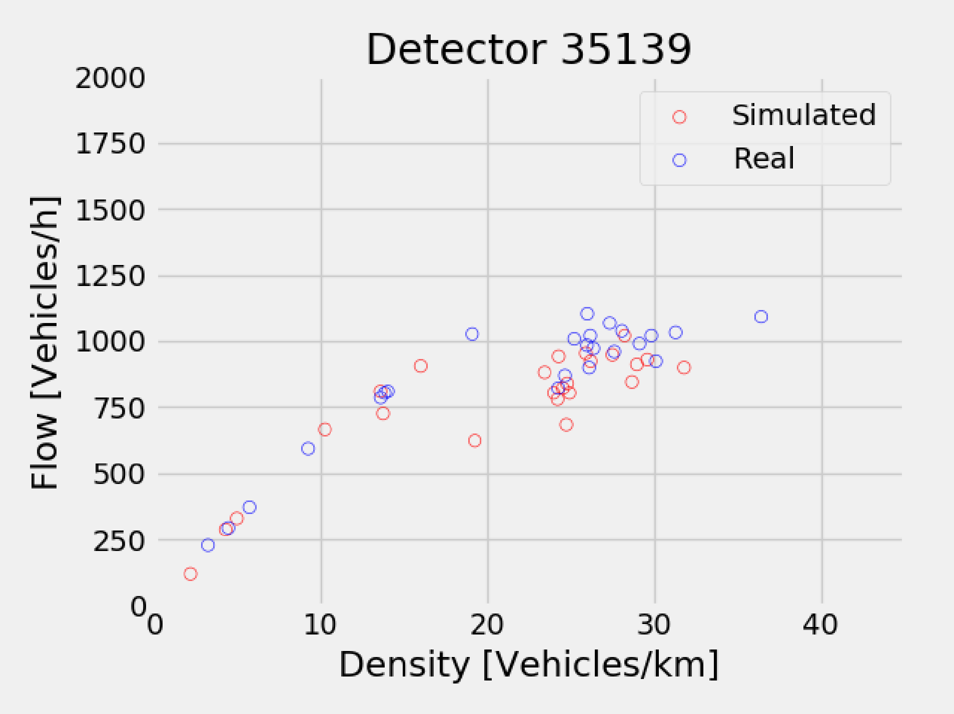


Figure 4.19: Optimized parameter simulation output for detector 35139.

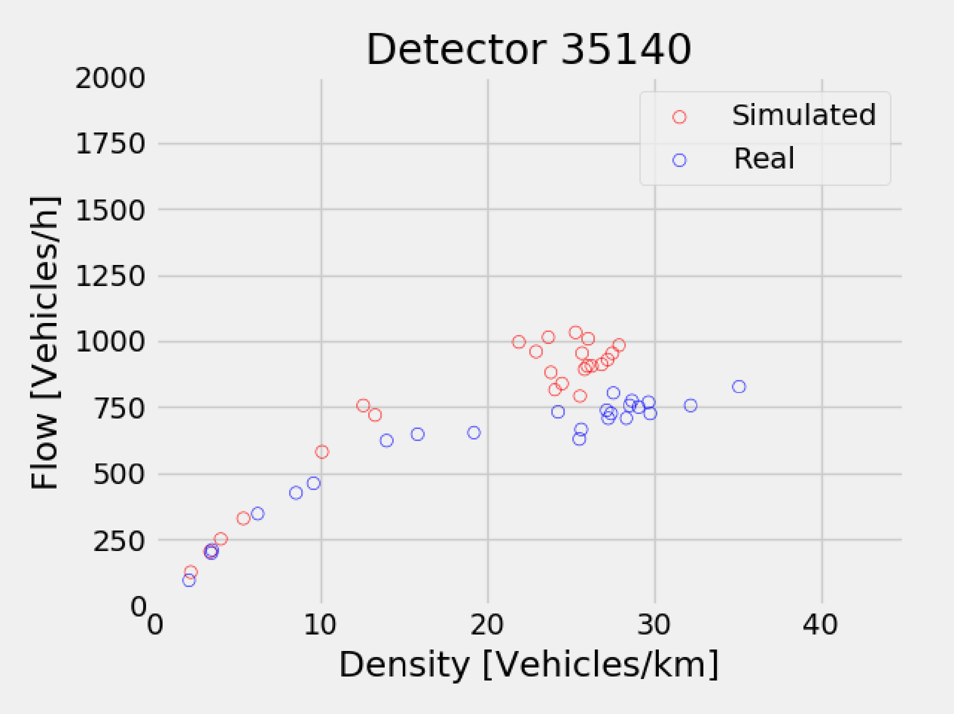


Figure 4.20: Optimized parameter simulation output for detector 35140.

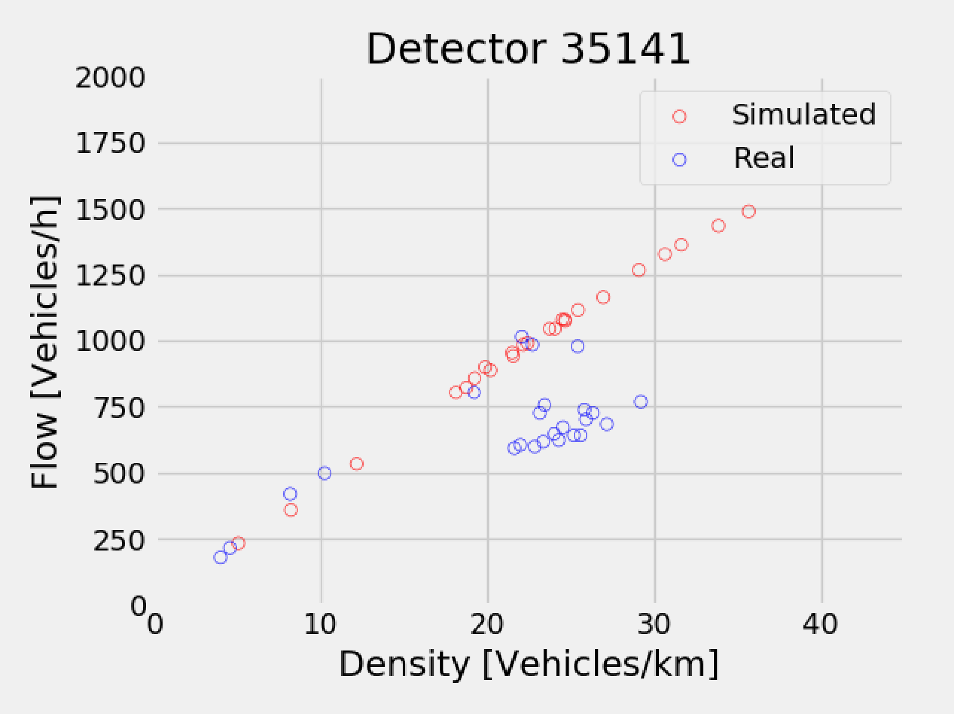


Figure 4.21: Optimized parameter simulation output for detector 35141.

For temporal comparison, data can once again be plotted over time for *flow*, *speed* and *density*, see Figures 4.22, 4.23 and 4.24. For plots for each detector individually, see Appendix B.

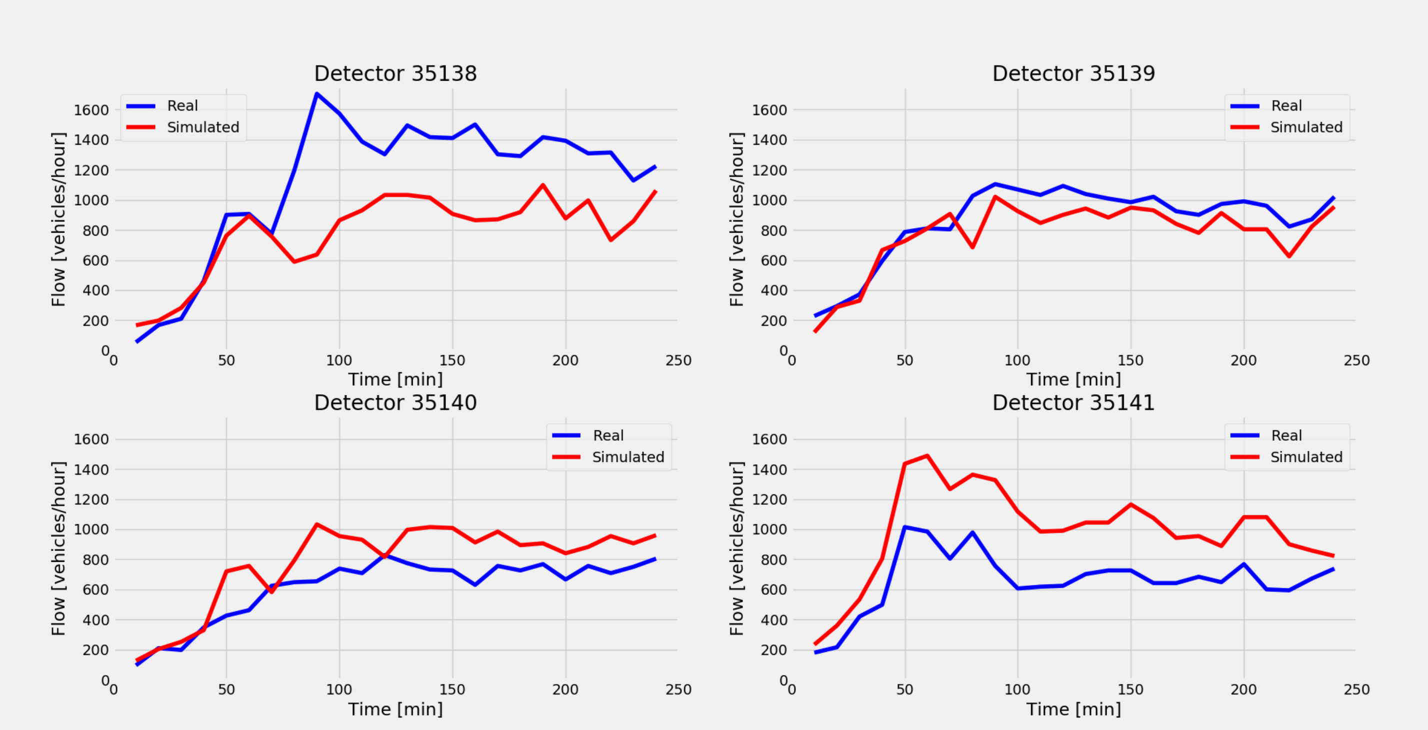


Figure 4.22: Flow over time with optimized parameters.

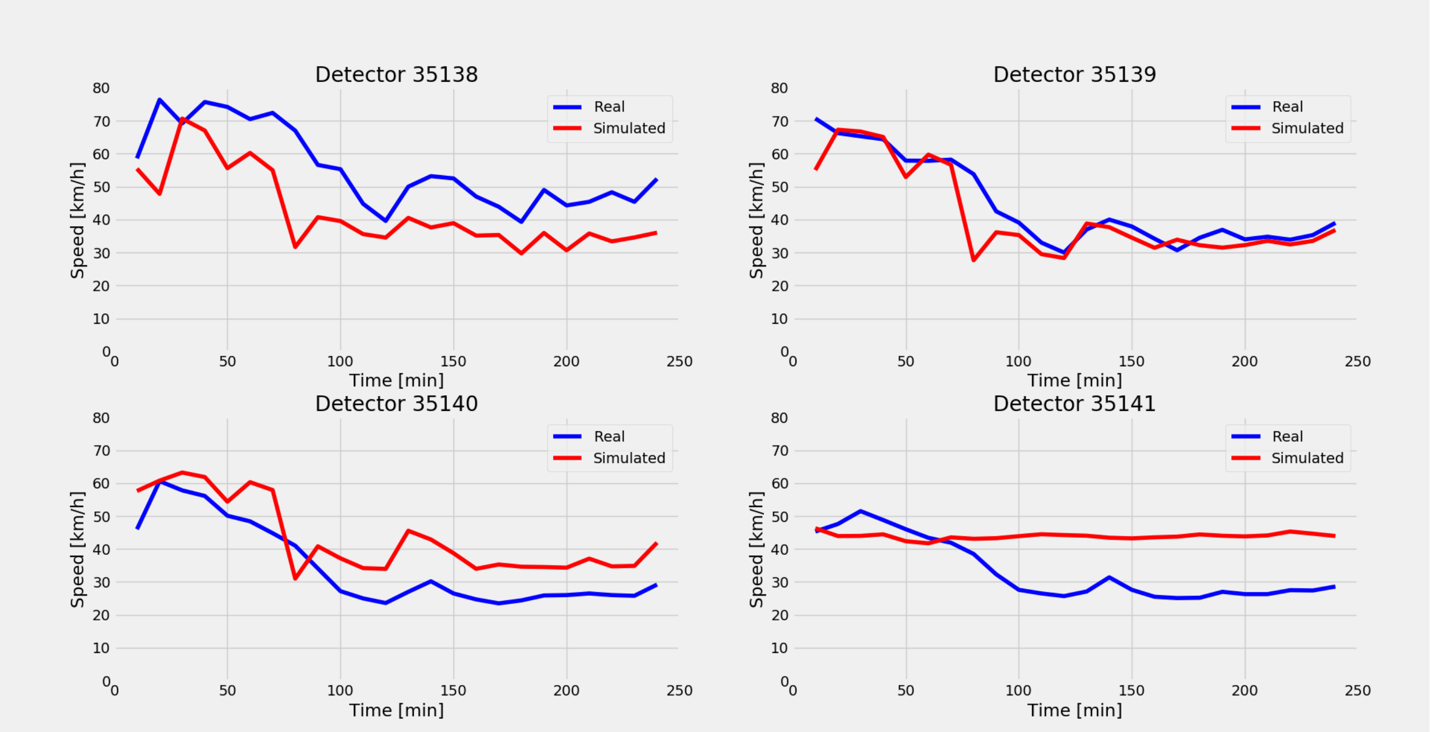


Figure 4.23: Speed over time with optimized parameters.

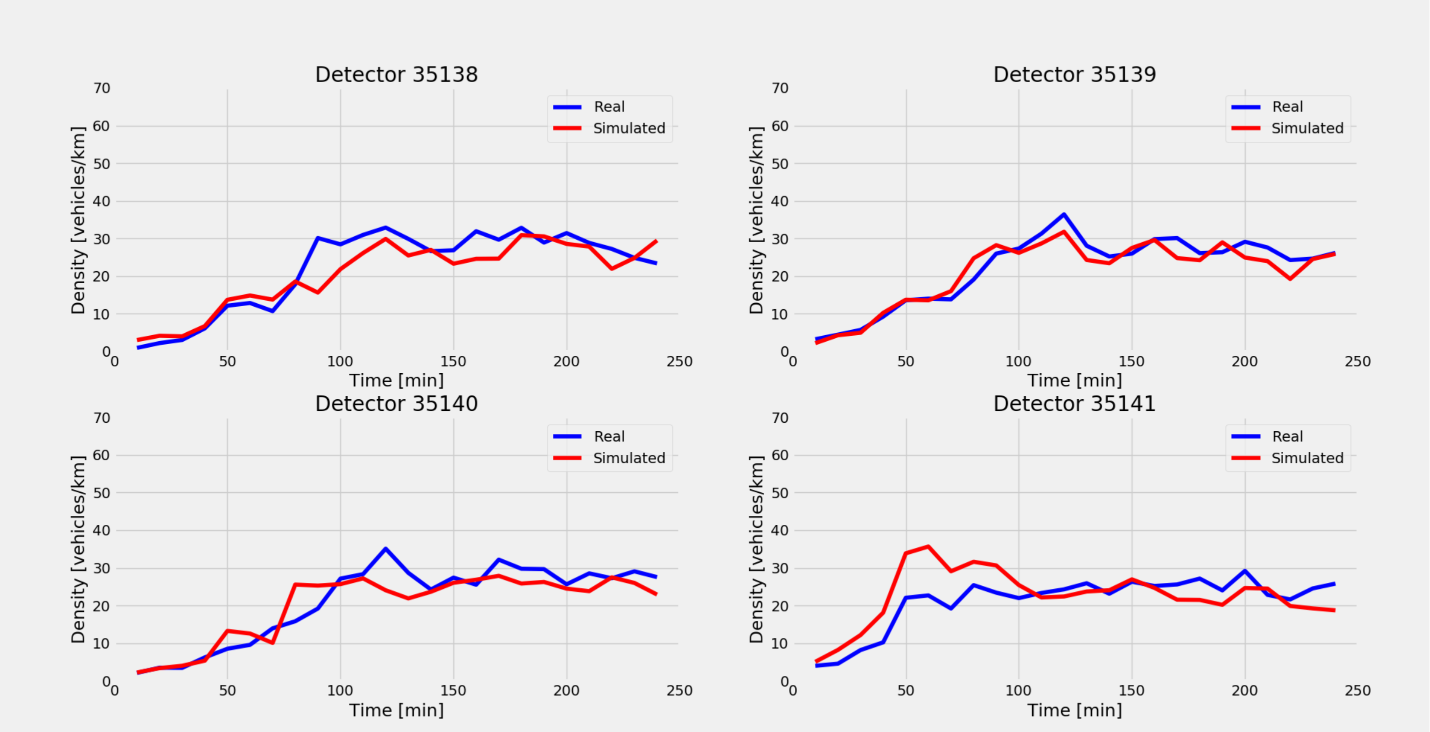


Figure 4.24: Density over time with optimized parameters.

4.5 Optimization for improving traffic

In this section different fractions of autonomous vehicles are introduced and their parameters are optimized to improve traffic. The objective function used in this optimization can be seen in Equation 2.19. For reference the value of the objective function with no AVs is 61.69%. It should be noted that AVs are not allowed to drive faster than the speed limit and that human drivers, when given the chance, go on average 21% over the speed limit in the area. This means that the improvements seen in this section are slightly more impressive than they seem at first glance.

The same in- and outflows as in the real data was used and the AVs were randomly inserted in place of a real vehicles in the appropriate fractions. The optimization parameters and their respective ranges can be seen in Table 3.3.

4.5.1 10% autonomous vehicles

The best value found for 10% AVs was 65.95% (with respect to Equation 2.19) and the parameter values found can be seen in Table 4.2.

Parameter	**Value**	**Unit**
accel	1.0288	m/s^2
decel	4.000	m/s^2
lcCooperative	0.9773	N/A
lcStrategic	9.6796	N/A
minGap	2.8473	m
tau	0.5000	s

Table 4.2: Optimized parameters for 10% AVs.

4.5.2 25% autonomous vehicles

The best value found for 25% AVs was 67.17% (with respect to Equation 2.19) and the parameter values found can be seen in Table 4.3.

Parameter	**Value**	**Unit**
accel	1.3003	m/s^2
decel	5.8798	m/s^2
lcCooperative	1.0000	N/A
lcStrategic	8.7962	N/A
minGap	2.2847	m
tau	0.5000	s

Table 4.3: Optimized parameters for 25% AVs.

4.5.3 50% autonomous vehicles

The best value found for 50% AVs was 68.29% (with respect to Equation 2.19) and the parameter values found can be seen in Table 4.4.

Parameter	**Value**	**Unit**
accel	1.4960	m/s^2
decel	5.6712	m/s^2
lcCooperative	0.9991	N/A
lcStrategic	9.9933	N/A
minGap	3.9989	m
tau	0.5010	s

Table 4.4: Optimized parameters for 50% AVs.

4.5.4 100% autonomous vehicles

The best value found for 100% AVs was 70.58% (with respect to Equation 2.19) and the parameter values found can be seen in Table 4.5.

Parameter	**Value**	**Unit**
accel	1.5000	m/s^2
decel	6.0000	m/s^2
lcCooperative	0.9999	N/A
lcStrategic	10.0	N/A
minGap	1.5014	m
tau	0.5000	s

Table 4.5: Optimized parameters for 100% AVs.

4.5.5 Visualization of the impact of AVs

Since the objective function used to measure how well traffic flows is wholly dependent on the speed, the speed over time is plotted for all fractions of AVs together in Figures 4.25, 4.26, 4.27 and 4.28. It can be clearly seen for detectors 35138, 35139 and 35140 that the more AVs the less likely it is that the average speed will rapidly drop and cause a short stand still. As for detector 35141 the opposite effect can be observed, the more AVs are introduced the lower the average speed. This is simply because 35141 is located at the end of a lane connecting to the other three and is essentially in a permanent state of free flow. This means that the non automated vehicles can go above the speed limit while the AVs still have to obey the speed limit.

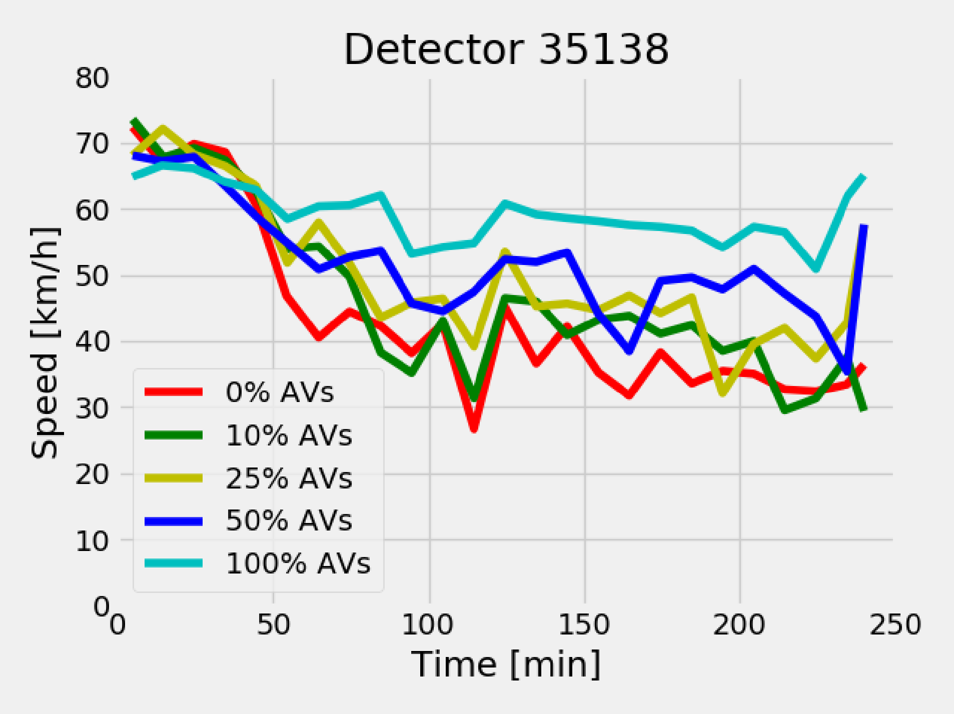


Figure 4.25: Speed over time for different fractions of AVs for detector 35138.

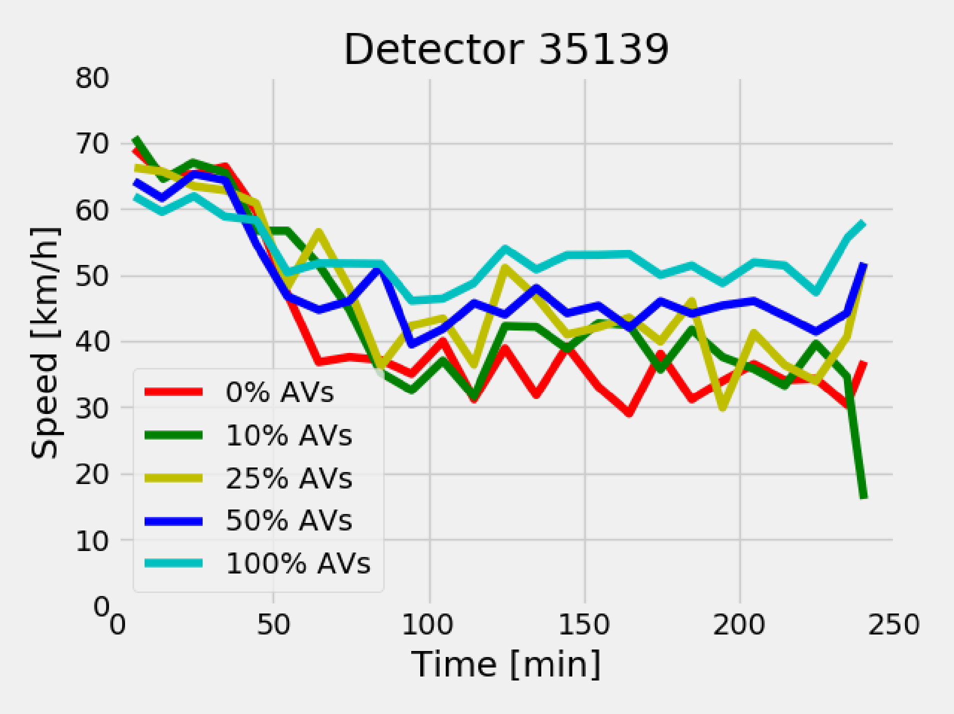


Figure 4.26: Speed over time for different fractions of AVs for detector 35139.

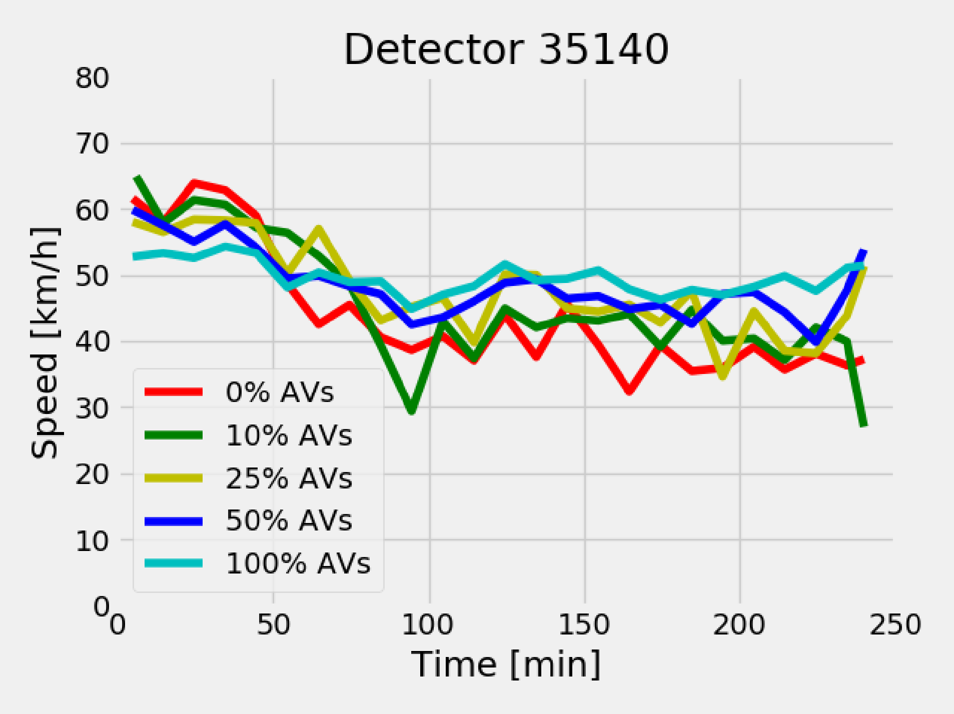


Figure 4.27: Speed over time for different fractions of AVs for detector 35140.

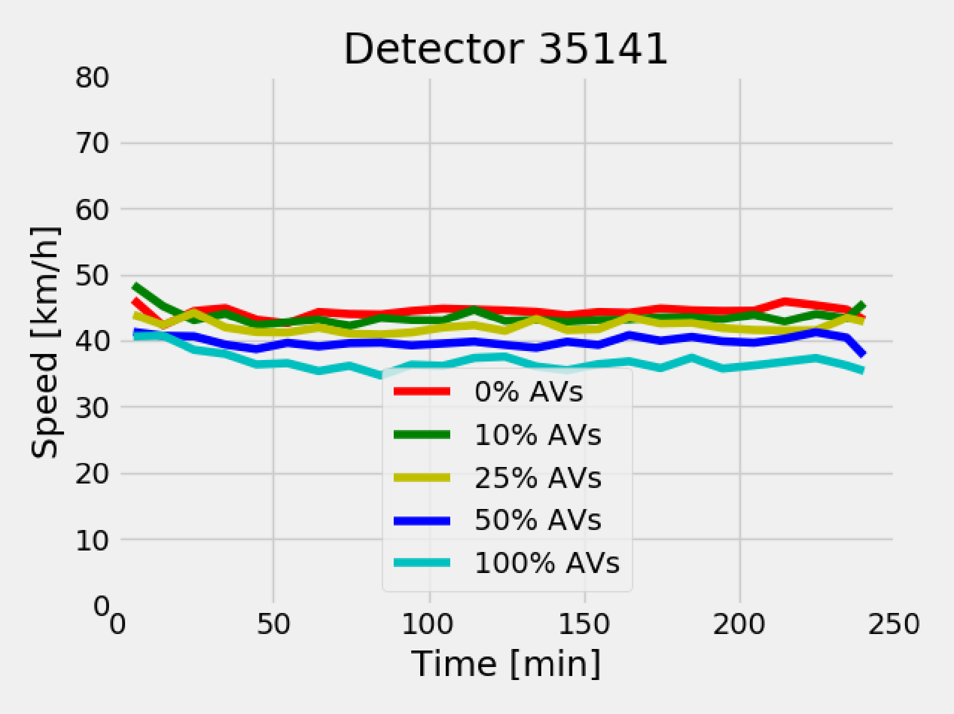


Figure 4.28: Speed over time for different fractions of AVs for detector 35141.

For completeness, the same temporal plots for *Flow* and *Density* can be seen in Appendix C.

4.6 Sensitivity analysis

This section describes the results of the different sensitivity analyses conducted. Because each sensitivity analysis contains four times as many values as variables, it was concluded that only the five largest contributors would be presented.

4.6.1 Sensitivity with respect to fit

Table 4.6 shows the largest contributors in the sensitivity analysis for the objective function in Equation 2.15.

Parameter	**Modifier**	**Objective function**
lcCooperative (mean)	+25%	+32.24%
lcCooperative (mean)	+10%	+27.28%
lcCooperative (mean)	−10%	+25.34%
accel (mean)	+25%	+23.60%
speedFactor (mean)	−25%	+21.55%

Table 4.6: Biggest increases to the objective function found from the sensitivity analysis.

4.6.2 Sensitivity with respect to improving traffic

Table 4.7 shows the largest positive contributors in the sensitivity analysis for the objective function in Equation 2.19.

Parameter	**Modifier**	**Objective function**
speedFactor (mean)	+25%	+6.45%
tau (alpha)	−25%	+5.89%
decel (mean)	+25%	+4.88%
sigma (mean)	−25%	+3.99%
lcCooperative (mean)	+10%	+3.53%

Table 4.7: Biggest positive increases to the objective function found from the sensitivity analysis.

Table 4.8 shows the largest positive contributors in the sensitivity analysis for the objective function in Equation 2.19.

Parameter	**Modifier**	**Objective function**
speedFactor (mean)	−25%	−16.26%
decel (mean)	−25%	−14.54%
lcCooperative (mean)	−25%	−11.95%
lcStrategic (std)	−25%	−10.37%
accel (mean)	−25%	−7.86%

Table 4.8: Biggest negative increases to the objective function found from the sensitivity analysis.

5

Conclusion

This chapter describes the contributions of the thesis, answers the research questions and touches on possible future work.

5.1 Research questions

The thesis set out to answer four research questions, namely:

- What driver behaviour leads to more or less congestion in the area?
- How does introducing different fractions of autonomous vehicles impact traffic in the area?
- How can autonomous vehicles behave to improve traffic?
- What do the findings imply for the future development of autonomous vehicles?

5.1.1 What driver behaviour leads to more or less congestion in the area?

As can be seen from the sensitivity analysis in Section 4.6.2, quite a few different parameters have a big impact on traffic in the area. The biggest improvement is by simply having everyone drive faster in free flow, which is not necessarily a very good idea in terms of safety. The same goes for decreasing the time gap and decelerating more rapidly, since the simulation will not allow collisions to occur it may still not be advantageous in real life.

Other factors that improve traffic according to the sensitivity analysis is reducing the driver imperfection (i.e. driving in a more stable and predictable manner) and being more cooperative with lane changes. These two changes to driver behaviour could more reasonably work out in real world traffic.

In terms of behaviour which makes traffic worse (with respect to Equation 2.19) it seems rather straight forward with parameters such as driving slowly, decelerating slowly (forcing others to adapt), not being cooperative with lane changing, not planning lane changes ahead of time and accelerating slowly. The outcome of changing these parameters is quite self explanatory and actually has a negative effect on traffic.

5.1.2 How does introducing different fractions of autonomous vehicles impact traffic in the area?

Introducing autonomous vehicles did improve traffic flow slightly but even with 100% AVs the improvement was not particularly large. There are a few possible reasons for why this would be the case, for example the AVs were not allowed to exceed the speed limit while human drivers in the area on average driver 21% above the speed limit. This does skew the comparison a bit but also means that the improvement seen by introducing AVs is actually larger than the numbers suggest.

There are other reasons for the rather small improvement brought on by the introduction of AVs, such as road design. It is possible that no matter how vehicles behave they cannot pass through this segment with these levels of flow without having to slow down. Finally, there is also a possibility that other parameters that were not subject to optimization could have further improved the result.

5.1.3 How can autonomous vehicles behave to improve traffic?

The findings indicate that both in small fractions (e.g. 10%) and when there are only AVs on the road it is optimal for them to max out both *lcStrategic* and *lcCooperative* which essentially means that they should change lanes as far in advance as possible and be as cooperative as possible with lane changes.

Another point to take away is that the fewer human drivers are in the system the more aggressively the AVs can accelerate and decelerate. This is most likely because of the effects of reaction time in human drivers, when there are exclusively AVs the upsides of keeping a steady speed go away. The comfortability of the passengers must be considered and accelerating and decelerating too rapidly may be limited due to this factor.

Finally, the *minGap* seems to have no big effect while it is always best to keep as small of a time gap as possible. This may not be true in reality however, since collisions can occur. The takeaway from the time gap is that AVs should keep a time gap that works well with human driver reaction time, which is a study of its own.

5.1.4 What do the findings imply for the future development of autonomous vehicles?

There are a couple of takeaways from the analysis of the behaviour of the AVs, for instance it would make sense to focus on improving/developing sensors and software to see and plan far ahead making it possible for the AVs to plan their actions ahead of time (seeing as lcStrategic is being maxed out by the optimization).

The other most important point is that since it was found that keeping the time gap as small as possible and being as cooperative as possible, it would make sense to study the interaction between humans and AVs further to see how to be as cooperative as possible. Finding the proper time gaps and lane changing procedures is clearly of big importance in creating well functioning AVs.

5.2 Sustainability implications

This thesis looks into the introduction of AVs on a large scale and possible developments that may occur within the foreseeable future. Implementation of fully automated networks, such as the ones investigated in this thesis, are expected to drastically reduce emissions and environmental impact compared to non-automated networks [24].

Because human errors are the overwhelmingly leading cause of vehicle accidents (94%) [25] and injury or even death is common in these types of incidents, it could be argued that the introduction of autonomous vehicles would be a major contributor to public safety.

5.3 Future work

This report should be enough to be able to replicate this study given that similar data is available. There are a few possible improvements that could be made to improve the results further, such as introducing more than just passenger cars (e.g. buses, motorcycles, trucks). This requires at least the knowledge of which proportions of vehicles are of each type, which is information that was gathered for this thesis but never utilized. The reason for this is simple, adding just one more type of vehicle almost doubles the numbers of parameters in the optimization. This can still be worked out but the optimization time would be increased, and the unambiguity of the result becomes even worse. It also important to note that passenger cars account for over 80% of vehicles on the road in Sweden [26] which makes the introduction of other vehicle types much less important than that of passenger cars.

The model could be improved further by taking in other types of data (e.g. time gap) and use this data directly as a parameter instead of letting the optimizer play with parameters that can be found out for certain. This reduces the complexity of the optimization, increases the accuracy of the result and makes the result a little less ambiguous.

Bibliography

[1] M. J. Lighthill and G. B. Whitham, “*On kinematic waves II. A theory of traffic flow on long crowded roads*,” 1955, [Online]. https://doi.org/10.1098/rspa.1955.0089, Date: 12 Feb 2019.

[2] G. F. Newell, “*Applications of Queueing Theory (2nd ed.)*,” 1982.

[3] C. Daganzo, “*The Cell Transmission Model, part II: Network Traffic*,” 1994, [Online]. https://escholarship.org/uc/item/9pz309w7, Date: 12 Feb 2019.

[4] M. Fellendorf, “*VISSIM: A microscopic Simulation Tool to Evaluate Actuated Signal Control including Bus Priority*,” 1994, [Online]. https://www.researchgate.net/profile/Martin_Fellendorf/publication/235358088_VISSIM_A_Microscopic_Simulation_Tool_to_Evaluate_Actuated_Signal_Control_including_Bus_Priority/links/569bfa2708aeeea985a59e74.pdf, Date: 8 Jan 2019.

[5] D. Krajzewicz, G. Hertkorn, P. Wagner, and C. Rössel, “*SUMO (Simulation of Urban MObility) - an open-source traffic simulation*,” 2002, [Online]. https://elib.dlr.de/6661/, Date: 8 Jan 2019.

[6] E. Bjärkvik, F. Fürer, M. Pourabdollah, and B. Lindenberg, “*Simulation and Characterisation of Traffic on Drive Me Route around Gothenburg using SUMO*,” 2017, [Online]. https://www.researchgate.net/publication/318110463_Simulation_and_Characterisation_of_Traffic_on_Drive_Me_Route_around_Gothenburg_using_SUMO, Date: 16 Jan 2019.

[7] S. Krauß, “*Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*,” 1998, [Online]. https://sumo.dlr.de/pdf/KraussDiss.pdf, Date: 11 Mar 2019.

[8] “*Definition of Vehicles, Vehicle Types, and Routes*,” [Online]. https://sumo.dlr.de/wiki/Definition_of_Vehicles,_Vehicle_Types,_and_Routes#Car-Following_Models, Date: 11 Mar 2019.

[9] “*SumoKrauß in C++*,” [Online]. https://github.com/eclipse/sumo/blob/master/src/microsim/cfmodels/MSCFModel_Krauss.cpp, Date: 11 Mar 2019.

[10] K. N. Porfyri, E. Mintsis, and E. Mitsakis, “*Assessment of ACC and CACC systems using SUMO*,” 2018, [Online]. https://doi.org/10.29007/r343, Date: 23 April 2019.

[11] J. Erdmann, “*SUMO’s Lane-Changing Model*,” 2015, [Online]. https://elib.dlr.de/102254/, Date: 23 Jan 2019.

[12] “*Lane-Changing Models*,” [Online]. https://sumo.dlr.de/wiki/Definition_of_Vehicles,_Vehicle_Types,_and_Routes#Lane-Changing_Models, Date: 13 Mar 2019.

[13] T. V. Mathew, "*Fundamental Relations of Traffic Flow*," 2014, [Online]. https://nptel.ac.in/courses/105101008/downloads/cete_02.pdf, Date: 11 Feb 2019.

[14] J. Godfrey, "*The mechanism of a road network*," 1969, [Online]. https://trid.trb.org/view/117139, Date: 7 Feb 2019.

[15] C. F. Daganzo and N. Geroliminis, "*An analytical approximation for the macroscopic fundamental diagram of urban traffic*," 2008, [Online]. https://doi.org/10.1016/j.trb.2008.06.008, Date: 7 Feb 2019.

[16] A. Costa and G. Nannicini, "*RBFOpt: an open-source library for black-box optimization with costly function evaluations*," 2014, [Online]. http://www.optimization-online.org/DB_HTML/2014/09/4538.html, Date: 26 Feb 2019.

[17] F. H. Administration, "*Estimates Of Urban Roadway Congestion - 1990*," 1993, [Online]. https://rosap.ntl.bts.gov/view/dot/14645, Date: 2 May 2019.

[18] F. He, X. Yan, Y. Liu, and L. Ma, "*A Traffic Congestion Assessment Method for Urban Road Networks Based on Speed Performance Index*," 2016, [Online]. https://doi.org/10.1016/j.proeng.2016.01.277, Date: 2 May 2019.

[19] R. G. Regis and C. A. Shoemaker, "*A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions*," 2007, [Online]. https://doi.org/10.1287/ijoc.1060.0182, Date: 24 April 2019.

[20] L. G. Willumsen, "*Estimation of an O-D Matrix from Traffic Counts - A Review*," 1978, [Online]. http://eprints.whiterose.ac.uk/2415/, Date: 22 Feb 2019.

[21] M. Savrasovs and I. Pticina, "*Methodology of OD Matrix Estimation Based on Video Recordings and Traffic Counts*," 2016, [Online]. https://doi.org/10.1016/j.proeng.2017.01.116, Date: 22 Feb 2019.

[22] "*flowrouter.py*," [Online]. https://sumo.dlr.de/wiki/Tools/Detector#flowrouter.py, Date: 22 Feb 2019.

[23] "*OpenStreetMap*," [Online]. https://www.openstreetmap.org, Date: 3 May 2019.

[24] C. Ross and S. Guhathakurta, "*Autonomous Vehicles and Energy Impacts: A Scenario Analysis*," 2017, [Online]. https://doi.org/10.1016/j.egypro.2017.12.646, Date: 29 May 2019.

[25] U. S. department of Transportation, "*Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey*," 2015, [Online]. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115, Date: 11 Jan 2019.

[26] T. Analys, "*Fordon 2017*," 2018, [Online]. https://www.trafa.se/globalassets/statistik/vagtrafik/fordon/2018/fordon_2017_blad.pdf, Date: 22 Jan 2019.

A

Default parameter simulation time plots

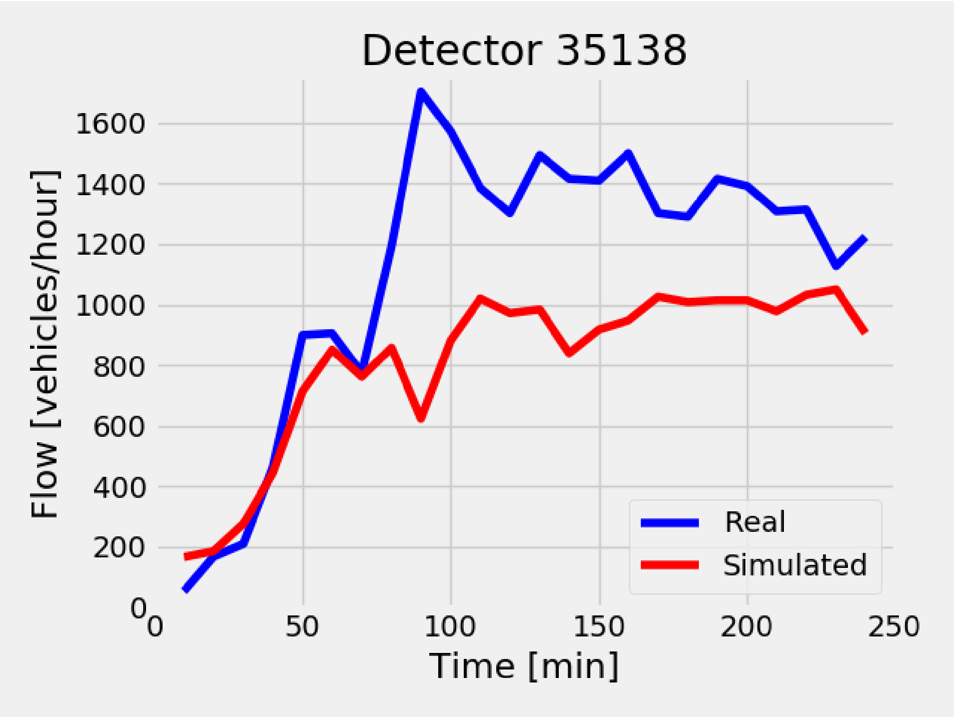


Figure A.1: Default parameter simulation flow output over time for detector 35138.

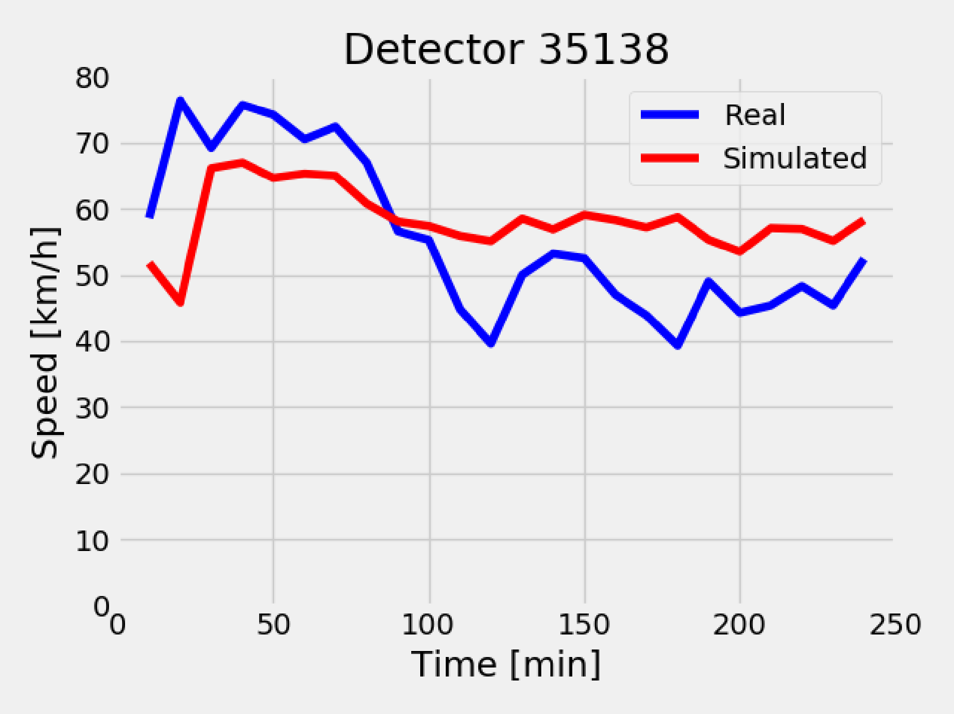


Figure A.2: Default parameter simulation speed output over time for detector 35138.

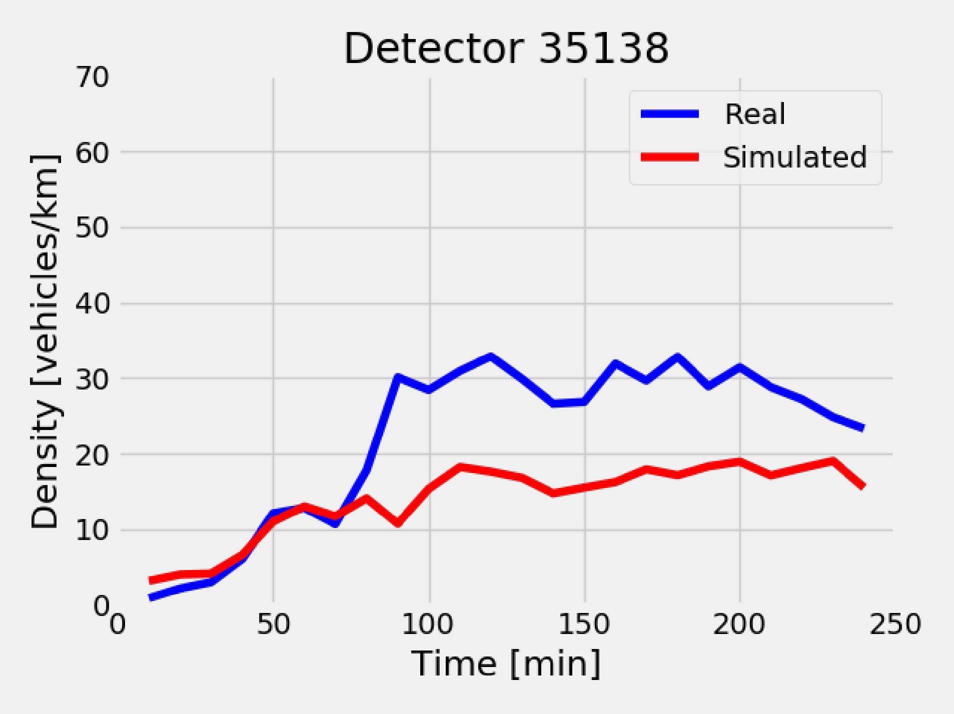


Figure A.3: Default parameter simulation density output over time for detector 35138.

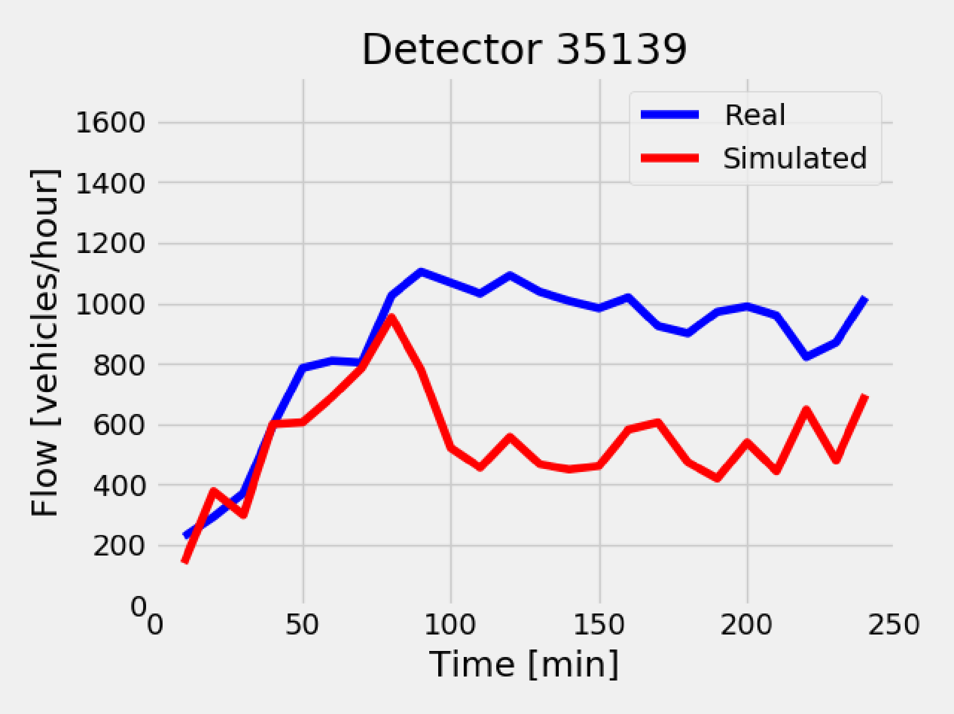


Figure A.4: Default parameter simulation flow output over time for detector 35139.

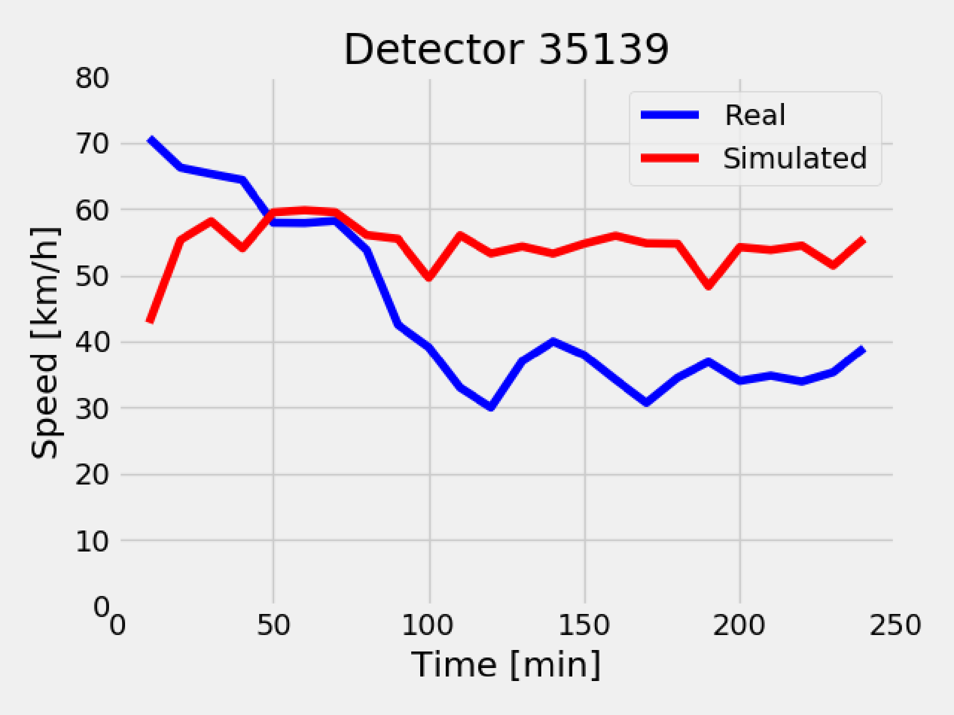


Figure A.5: Default parameter simulation speed output over time for detector 35139.

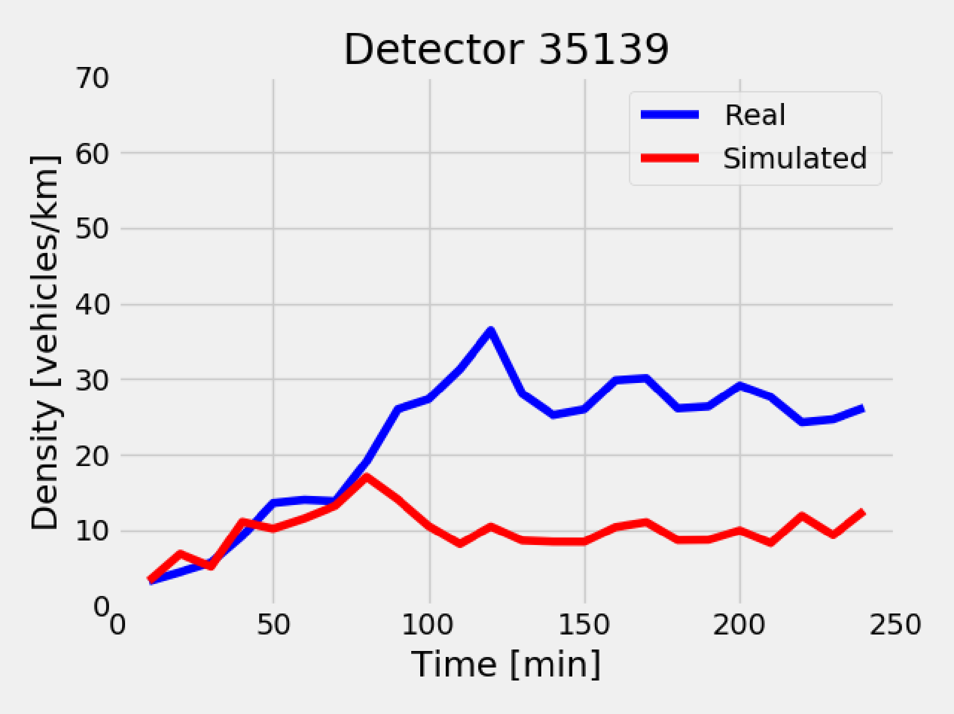


Figure A.6: Default parameter simulation density output over time for detector 35139.

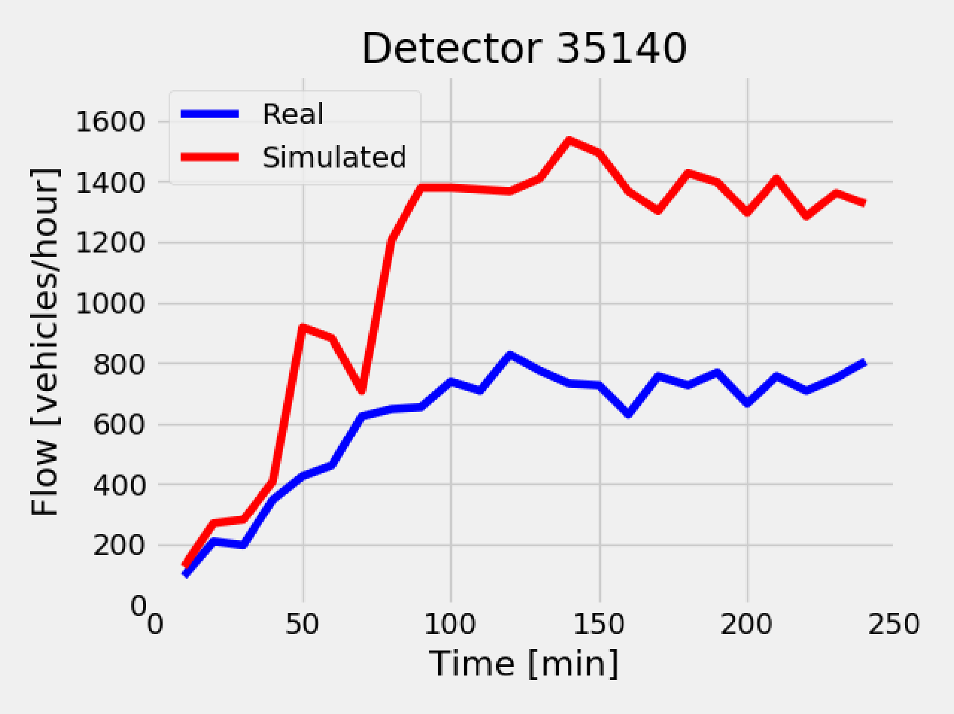


Figure A.7: Default parameter simulation flow output over time for detector 35140.

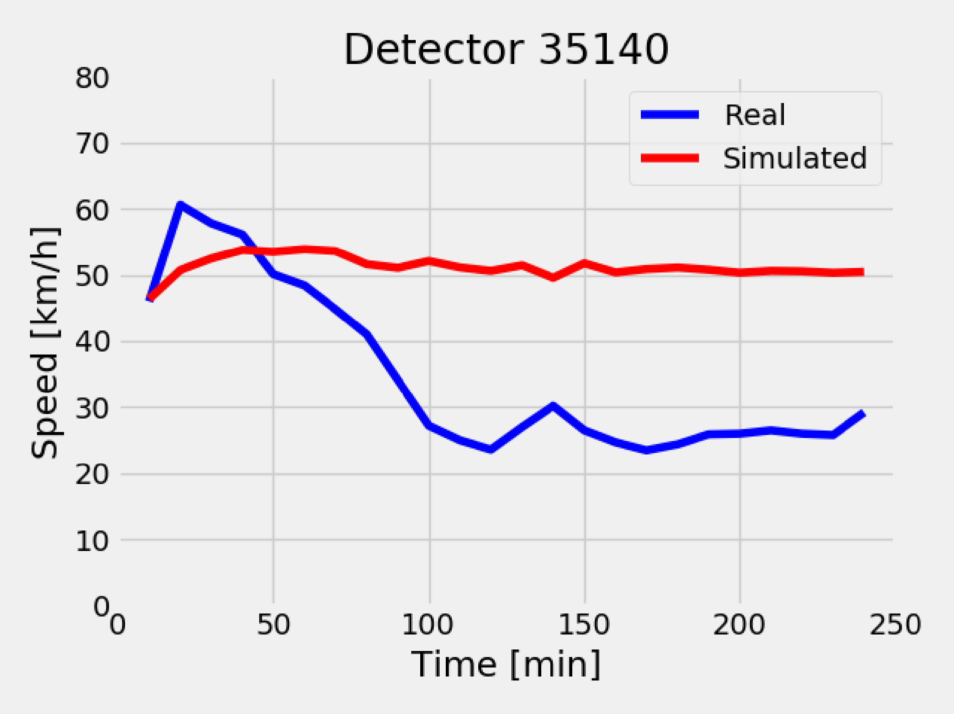


Figure A.8: Default parameter simulation speed output over time for detector 35140.

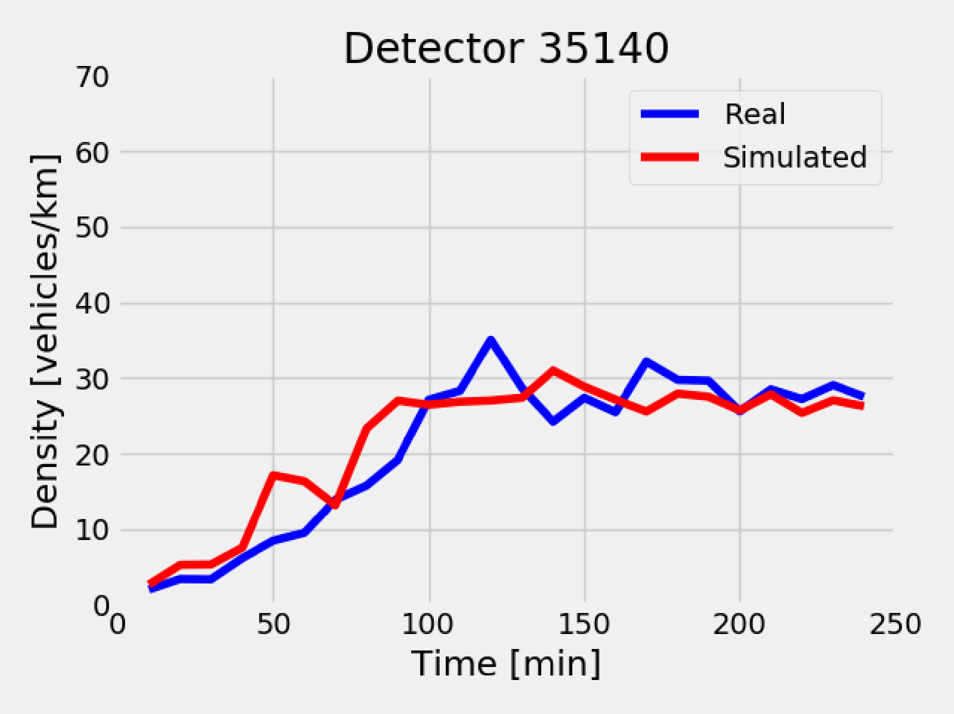


Figure A.9: Default parameter simulation density output over time for detector 35140.

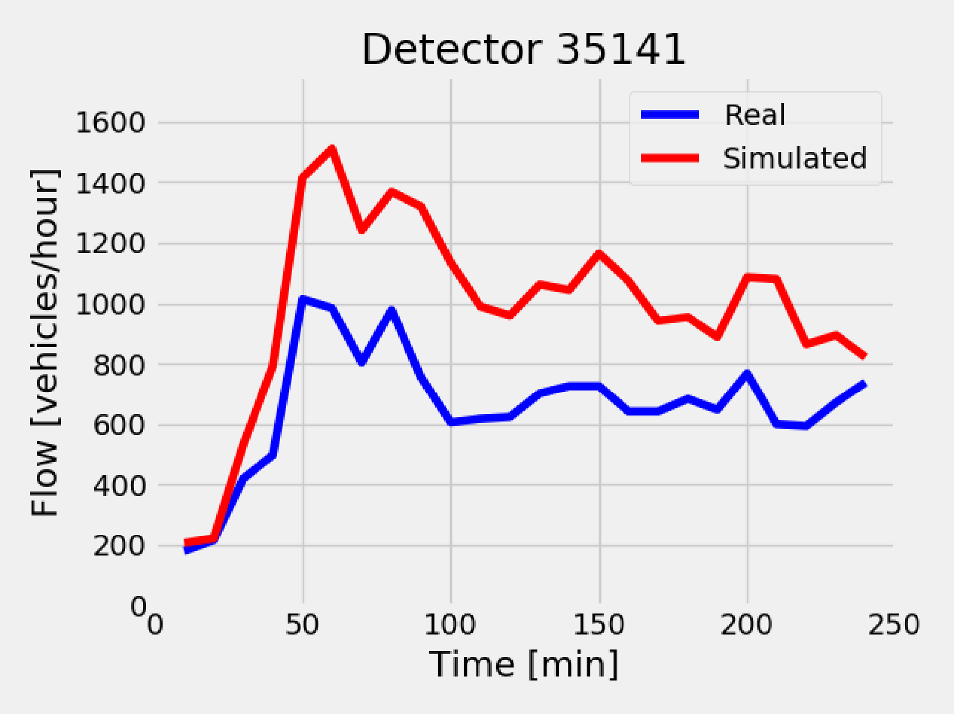


Figure A.10: Default parameter simulation flow output over time for detector 35141.

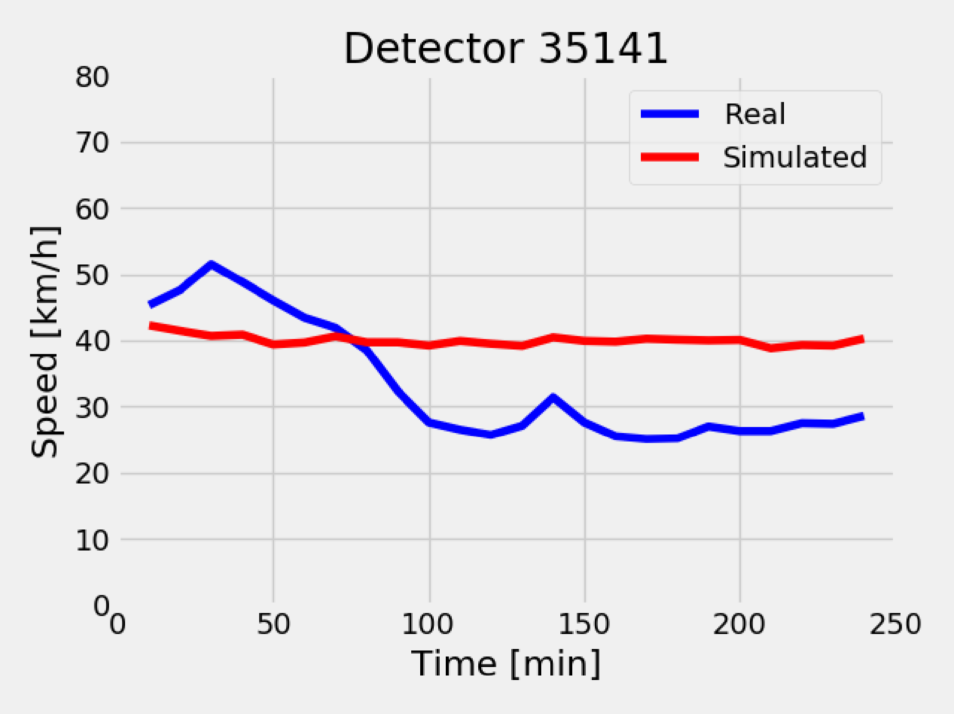


Figure A.11: Default parameter simulation speed output over time for detector 35141.

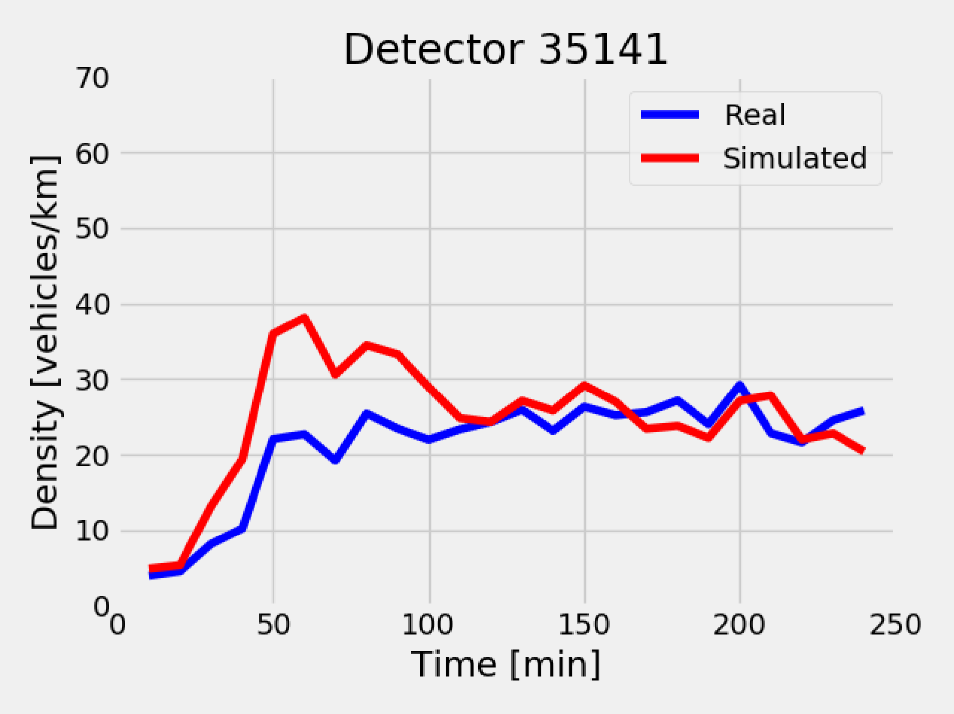


Figure A.12: Default parameter simulation density output over time for detector 35141.

B

Optimized parameter simulation time plots

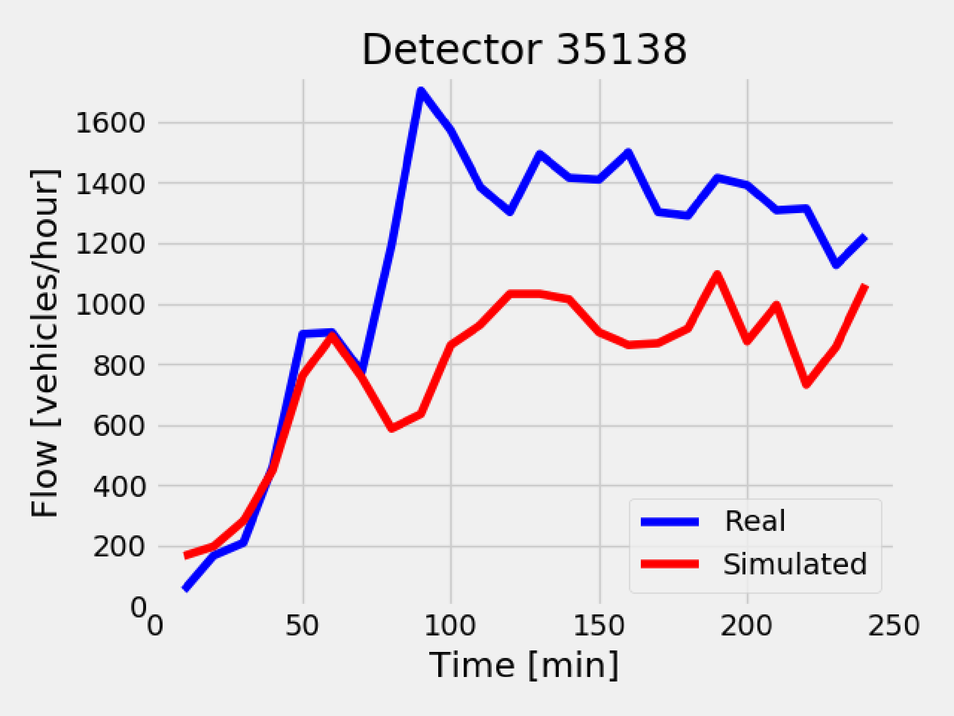


Figure B.1: Optimized parameter simulation flow output over time for detector 35138.

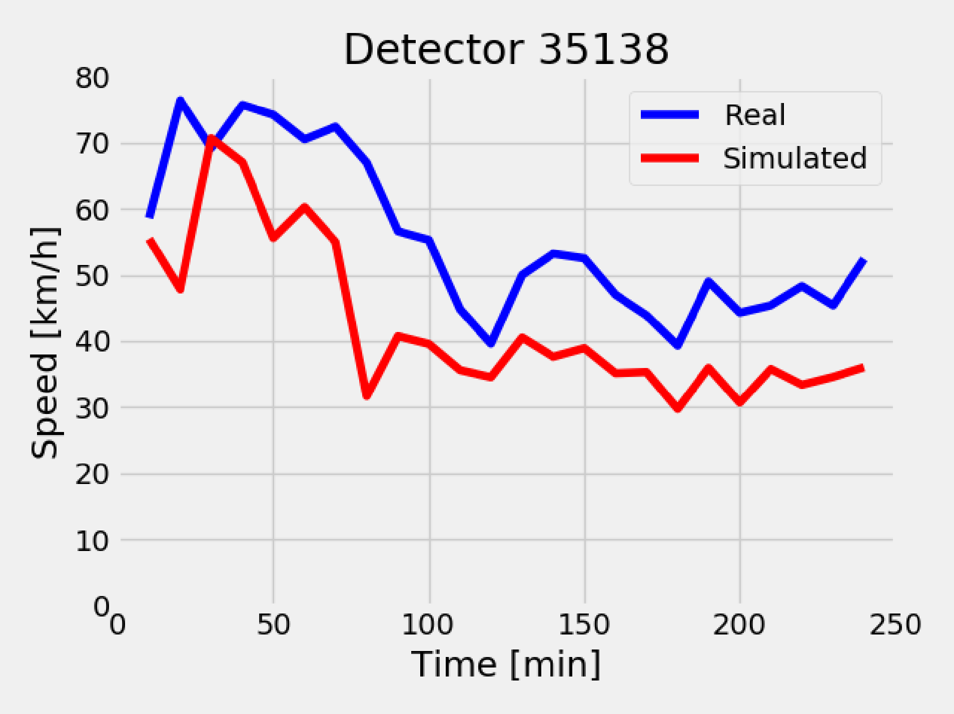


Figure B.2: Optimized parameter simulation speed output over time for detector 35138.

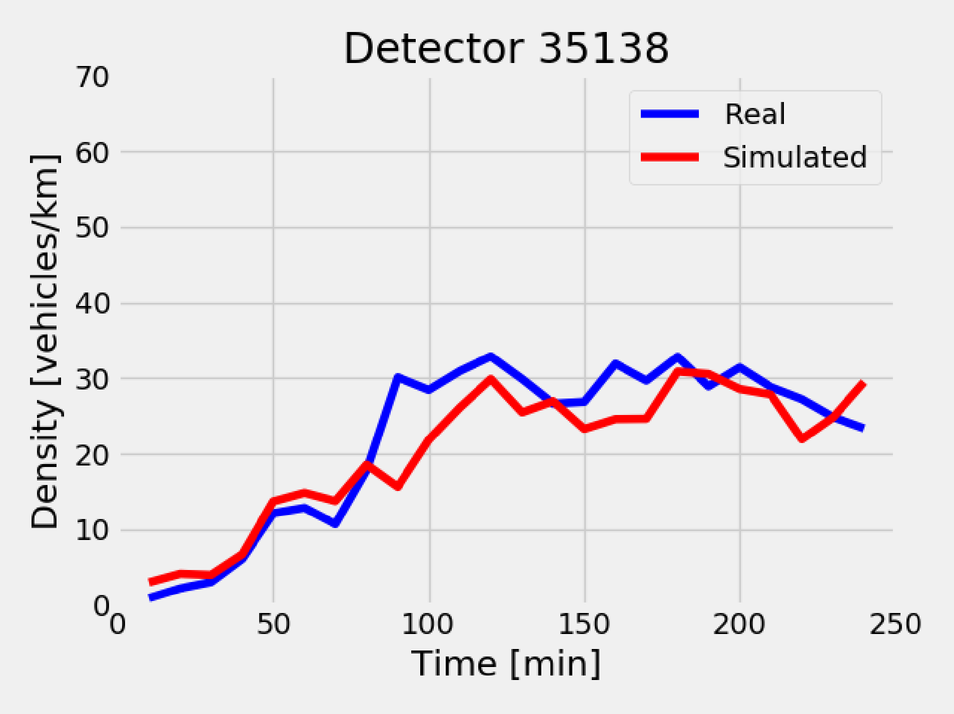


Figure B.3: Optimized parameter simulation density output over time for detector 35138.

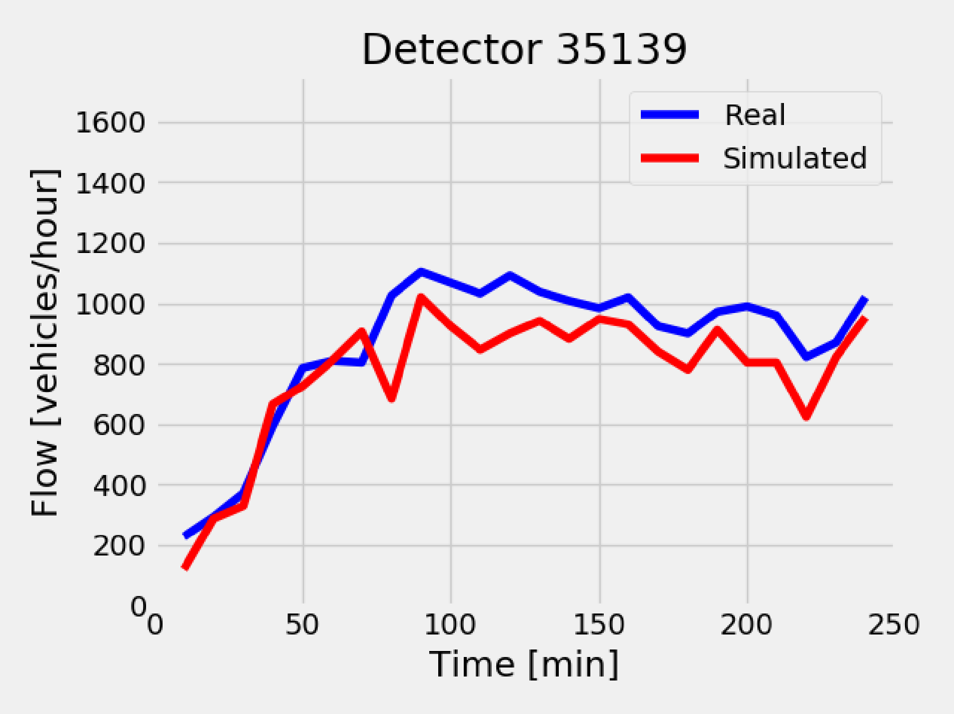


Figure B.4: Optimized parameter simulation flow output over time for detector 35139.

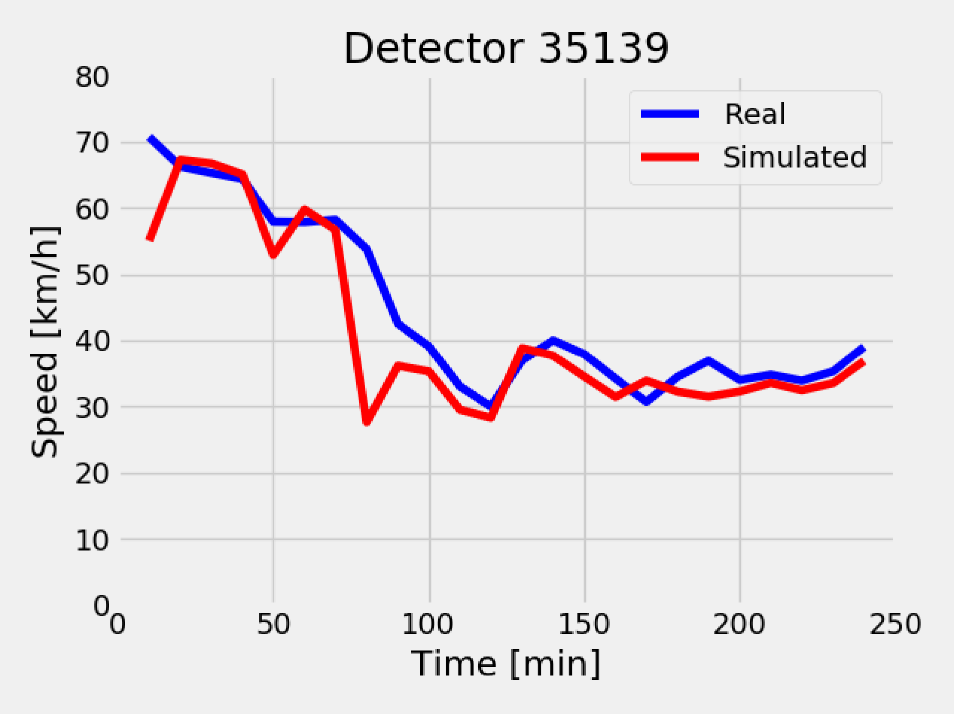


Figure B.5: Optimized parameter simulation speed output over time for detector 35139.

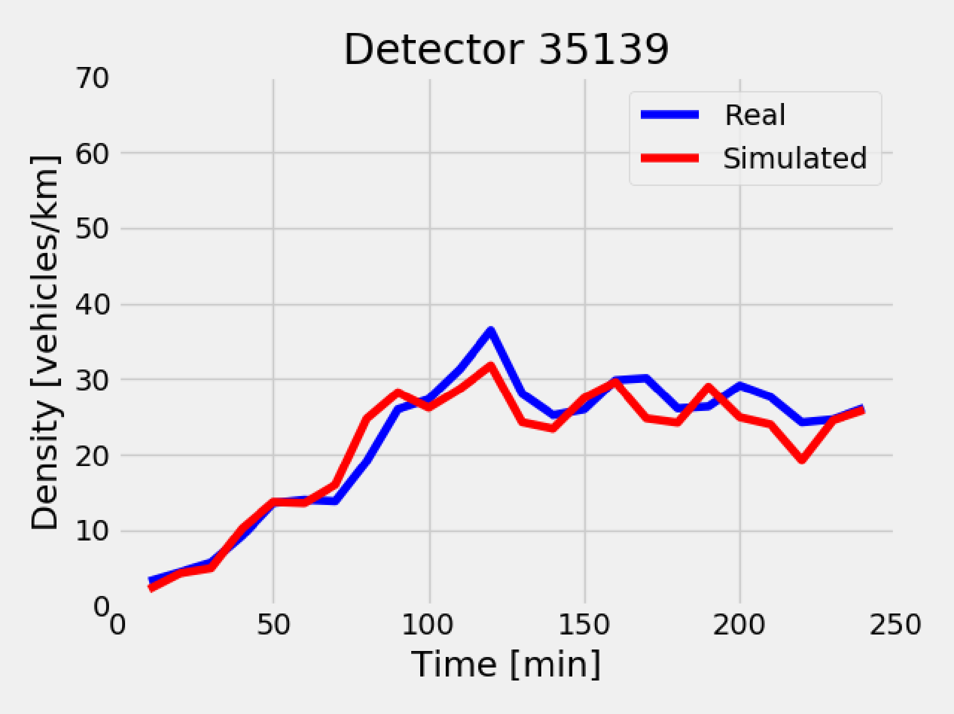


Figure B.6: Optimized parameter simulation density output over time for detector 35139.

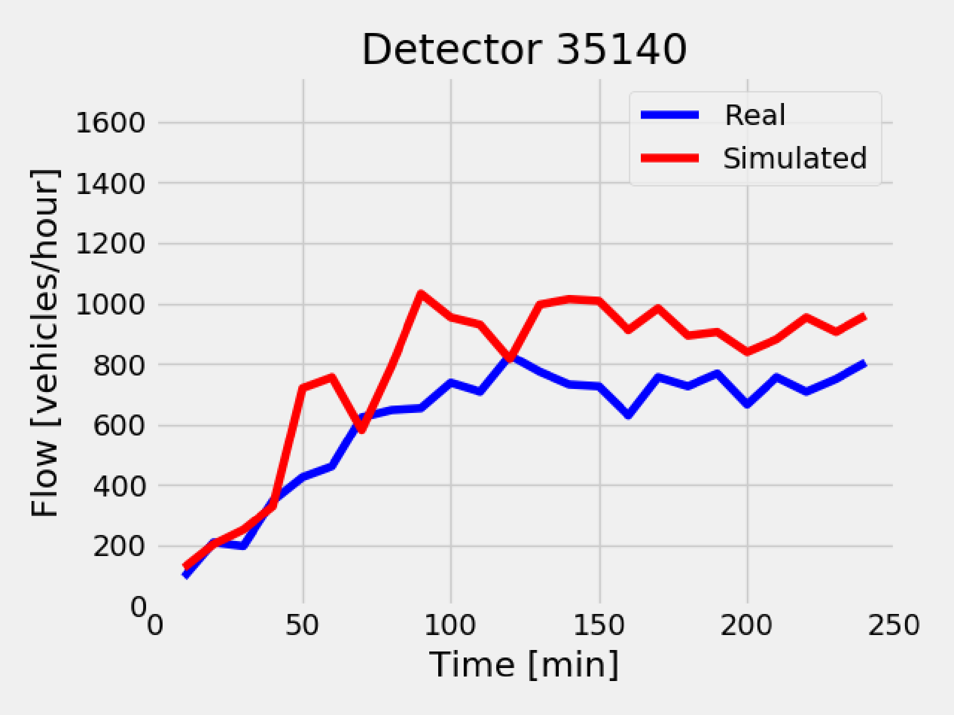


Figure B.7: Optimized parameter simulation flow output over time for detector 35140.

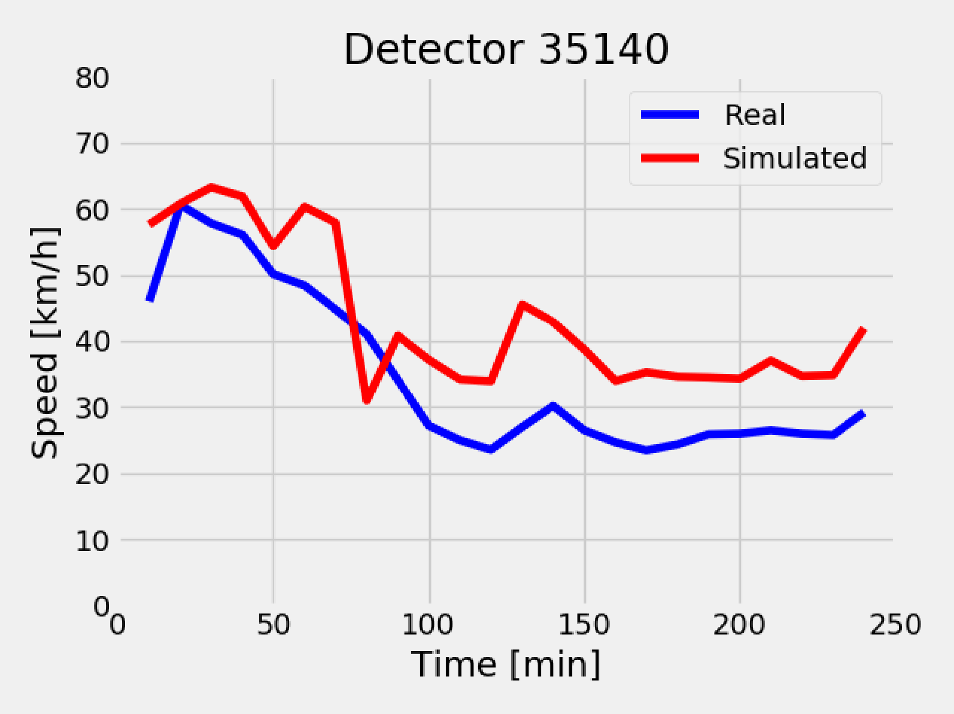


Figure B.8: Optimized parameter simulation speed output over time for detector 35140.

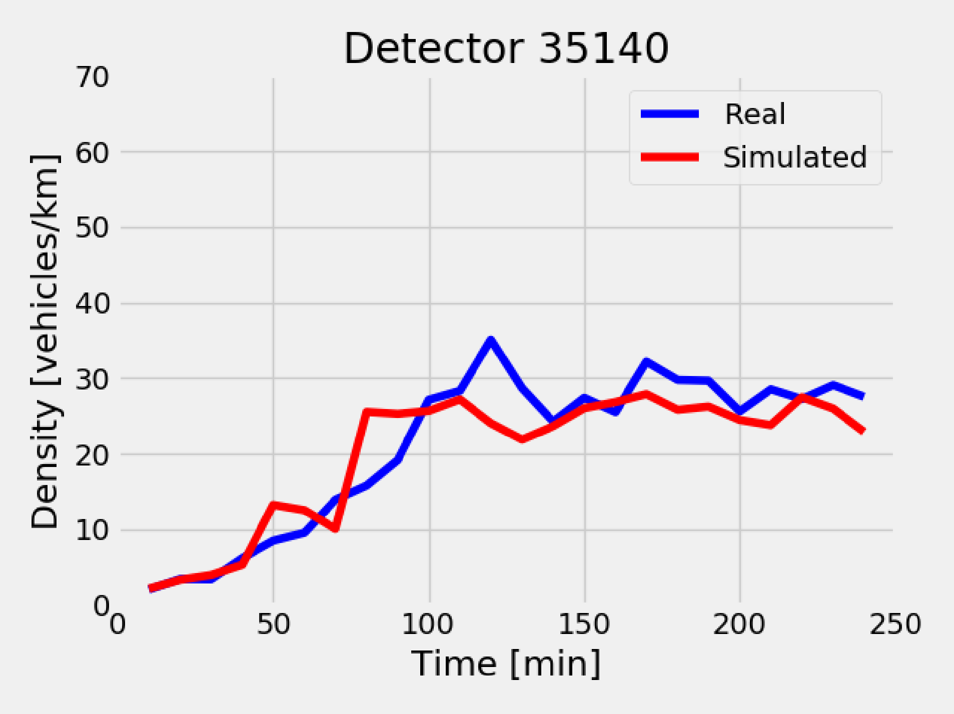


Figure B.9: Optimized parameter simulation density output over time for detector 35140.

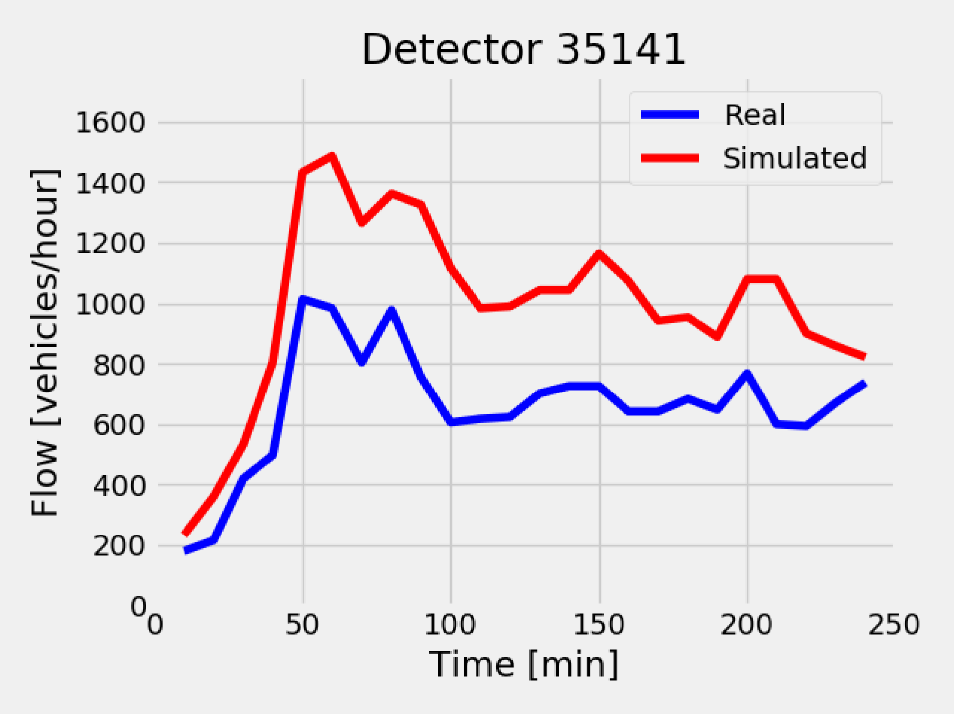


Figure B.10: Optimized parameter simulation flow output over time for detector 35141.

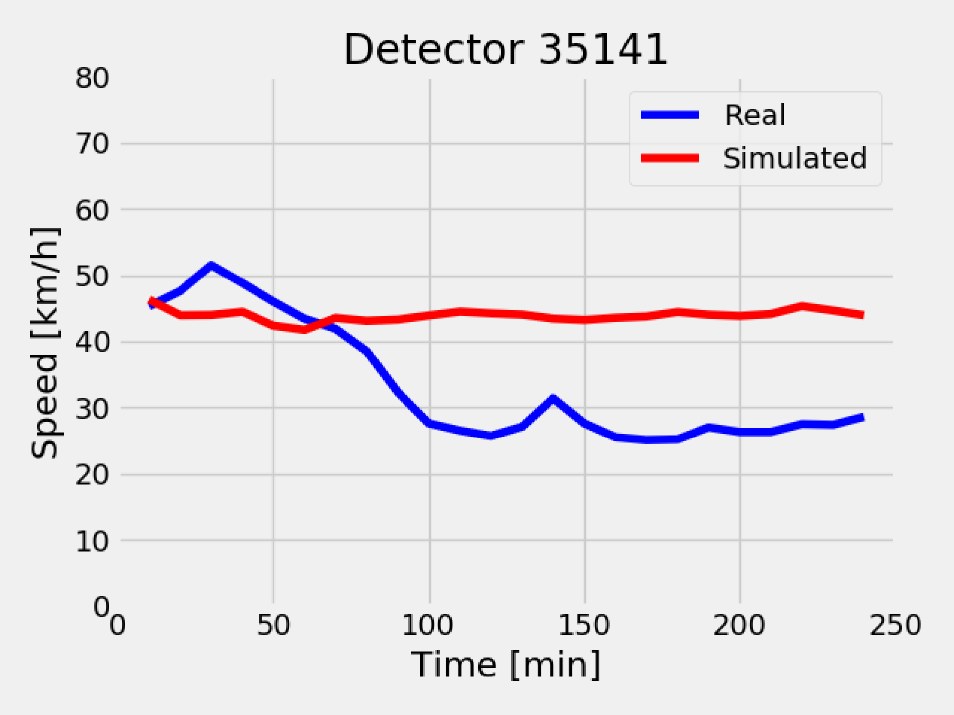


Figure B.11: Optimized parameter simulation speed output over time for detector 35141.

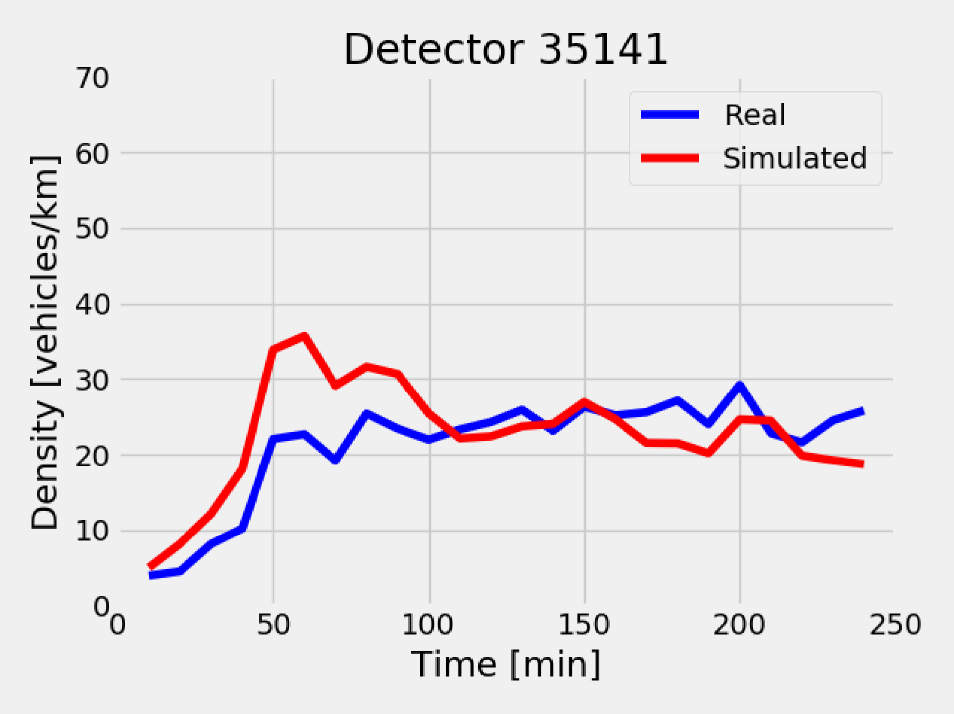


Figure B.12: Optimized parameter simulation density output over time for detector 35141.

C

Optimized parameter simulation time plots

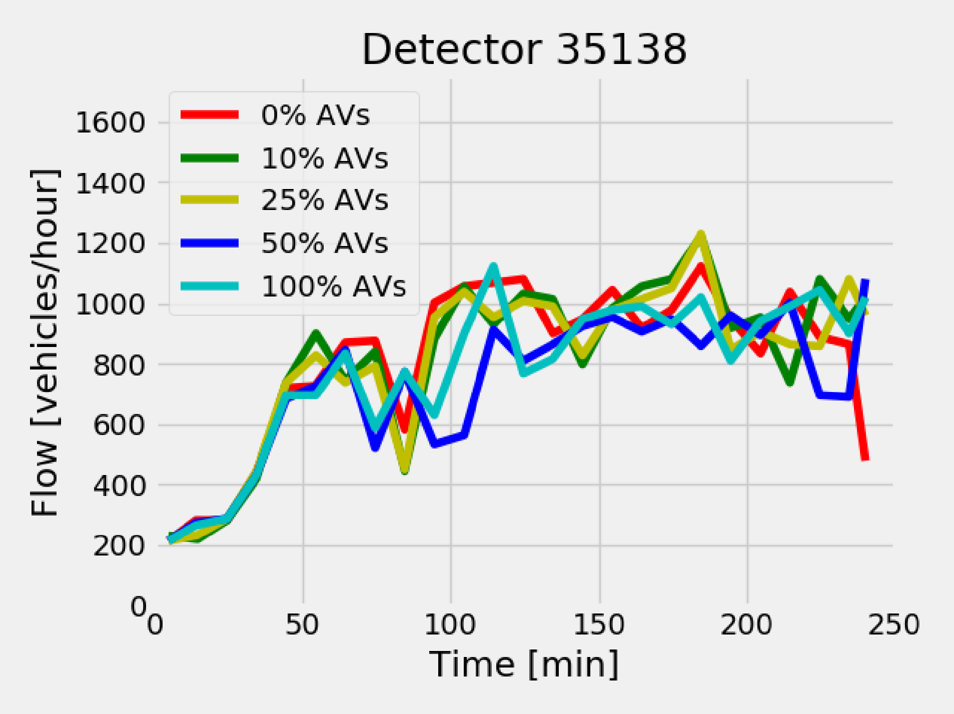


Figure C.1: Flow over time for different fractions of AVs for detector 35138.

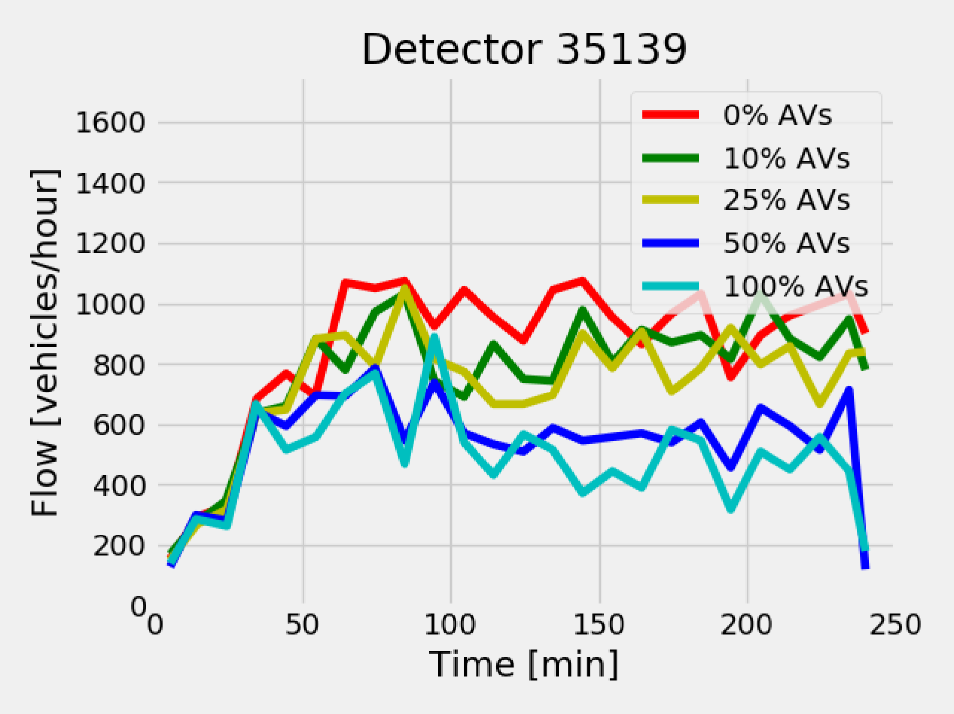


Figure C.2: Flow over time for different fractions of AVs for detector 35139.

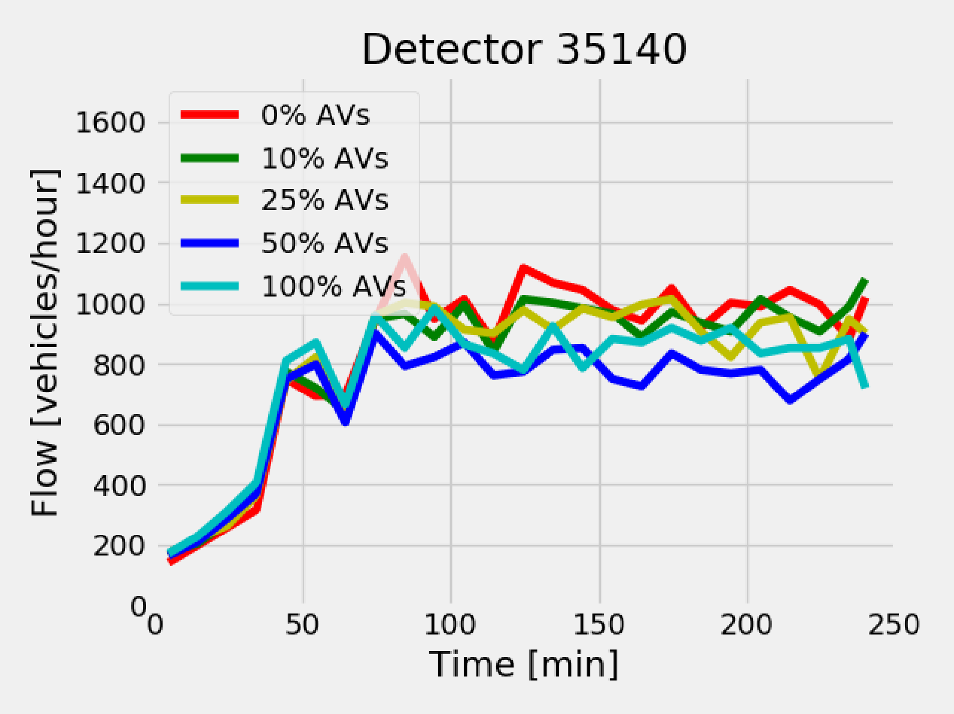


Figure C.3: Flow over time for different fractions of AVs for detector 35140.

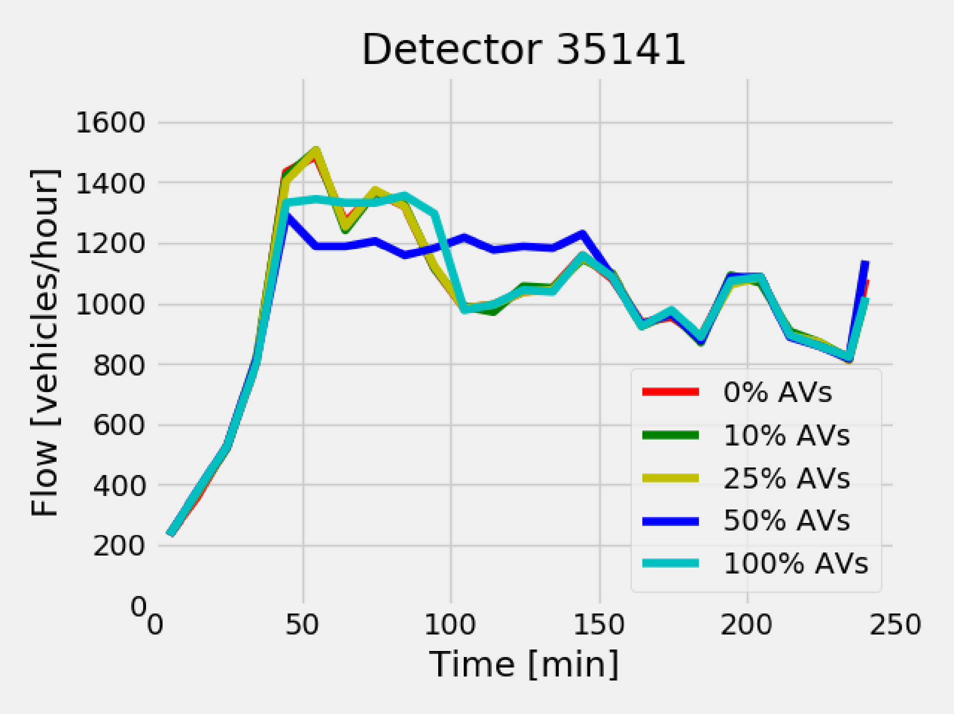


Figure C.4: Flow over time for different fractions of AVs for detector 35141.

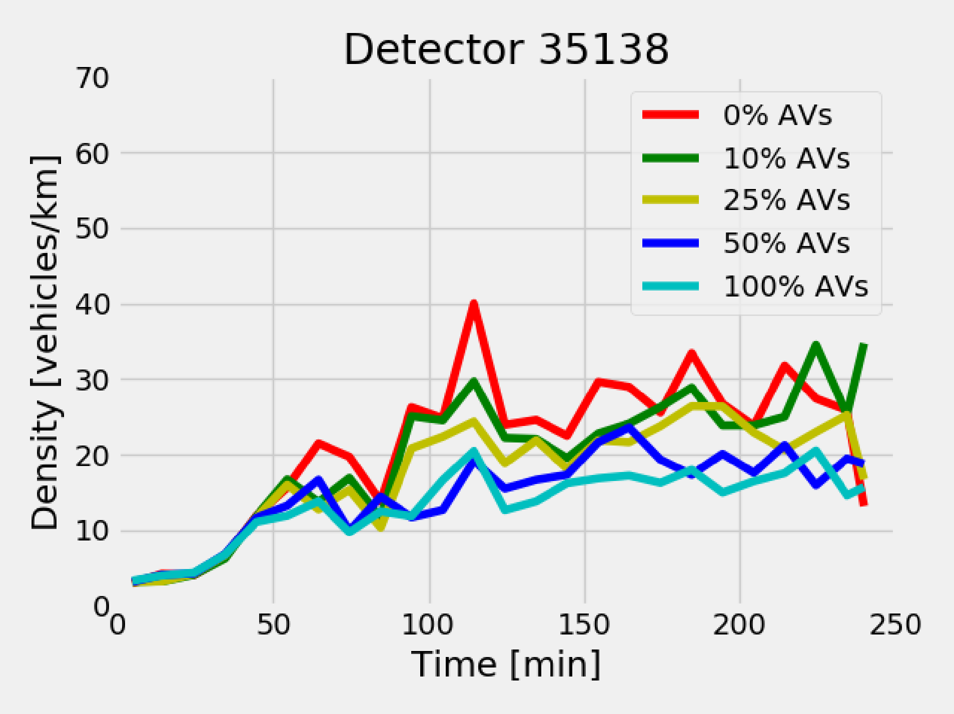


Figure C.5: Density over time for different fractions of AVs for detector 35138.

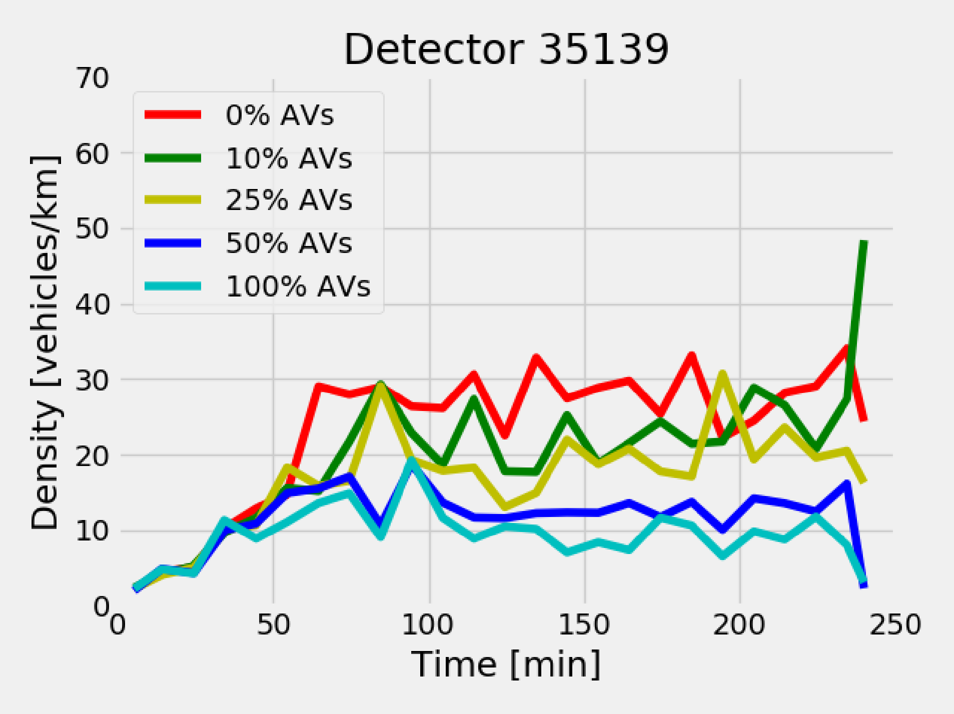


Figure C.6: Density over time for different fractions of AVs for detector 35139.

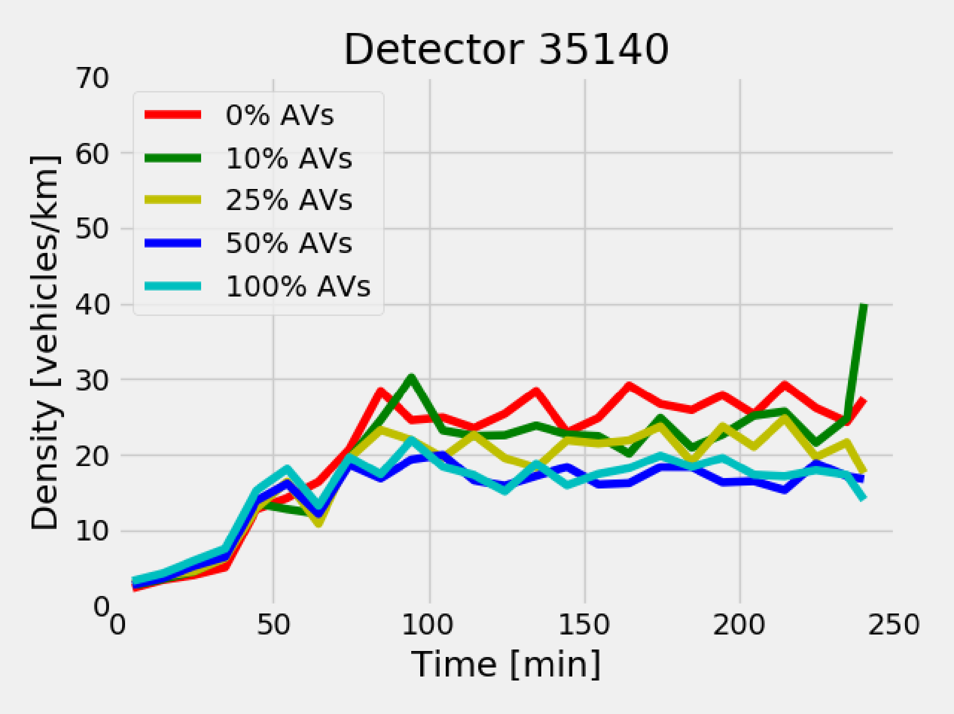


Figure C.7: Density over time for different fractions of AVs for detector 35140.

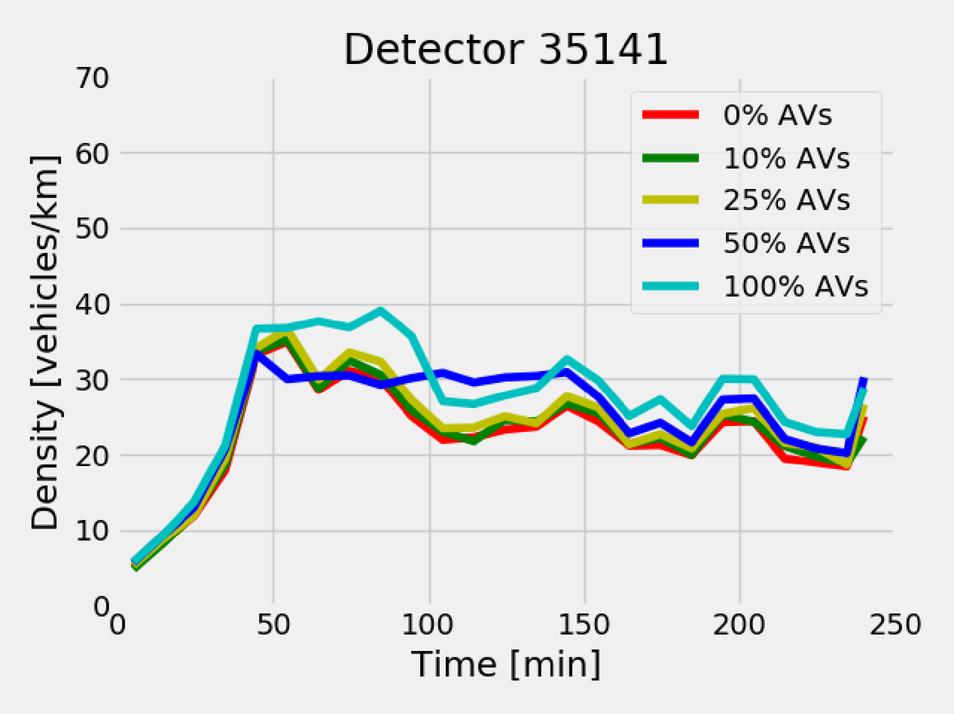


Figure C.8: Density over time for different fractions of AVs for detector 35141.