Capacity estimation of an intelligent turbo-roundabout

Development of a microscopic traffic simulation model in VISSIM

J. Andries

TU Delft

Capacity estimation of an intelligent turbo-roundabout

Development of a microscopic traffic simulation model in VISSIM

by

J. Andries

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on 09/04/2024

Student number: 4575121

Project duration: December 12, 2022 – April 9, 2024

Thesis committee:

Dr. V.L. Knoop,	TU Delft, Chair
Dr. ir. A.M. Salomons,	TU Delft, Daily Supervisor
Dr. ir. I. Martinez Josemaria,	TU Delft, Supervisor
Dr. ir. L.G.H. Fortuijn,	External Supervisor

TU Delft

Preface

The past year was one of the most challenging periods of my life so far. It was intense and stressful, but I learned much and noticed a steep personal development. The last five months, in which I embarked on another MSc adventure next to this one, were especially revealing and led me to find the limits of how much I can handle at a time.

Now that I am finishing this report, I can confidently say that I learned much about Connected and Automated Vehicles and vehicle behaviour on turbo-roundabouts. This thesis, titled *"Capacity estimation of an intelligent turbo-roundabout; Development of a microscopic traffic simulation model in VISSIM"* is a first attempt at determining the achievable capacity increase for automated vehicles on turbo-roundabouts using a microscopic traffic simulation model.

Without Dr. Ir. A.M. Salomons and Dr. Ir. LGH Fortuijn, I would not have had the opportunity to learn as much as I did. Their bimonthly expertise proved invaluable and allowed me to write my thesis about a topic where my initial knowledge came short. Even though the meetings with Dr. V.L. Knoop and Dr. Ir. I. Martinez Josemaria were much more scarce, their feedback and opinions were also instrumental in determining the right direction for my thesis.

Furthermore, I would like to thank my friends and family for their unconditional support during this challenging year. Bram and Sean, who were in a similar situation and dragged me through rough motivational stretches, deserve special attention. Of course, my mother and father also deserve a special mention here. Without their belief and confidence in my abilities and giving me the freedom to do what I wanted, I would not be in such a fortunate position as I am now.

Dear reader, I hope you enjoy reading the culmination of my efforts of the last year and may find inspiration for future research in this report.

J. Andries
Delft, April 2024

Summary

The field of automated vehicles gains more and more attention each year as the effective implementation of such vehicles in real life is imminent. The increasing amount of scientific studies addressing the impact of automated vehicles on the traffic network shows the same trend. While much research focuses on regular intersections, only a few investigate the impact of turbo-roundabouts. In particular, studies researching the obtainable capacity increase on turbo-roundabouts under fully automated conditions are scarce. A turbo-roundabout is a multi-lane roundabout with separated lanes that require vehicles to choose their desired destination before entering the turbo-roundabout, effectively eliminating most weaving conflicts on the roundabout itself. Only a single paper researches how much additional merging capacity can be obtained by automated vehicles on a turbo-roundabout compared to human-driven vehicles (Fortuijn and Salomons, 2020). This study takes a theoretical approach and uses existing theory about turbo-roundabout capacity to generate five graphs depicting the merging capacity of the left lane of a minor access branch for varying circulating traffic volumes on the turbo-roundabout: four models for differing automated vehicle settings and one human-driven vehicle model. These four automated vehicle models differ in complexity, where the most complex model features gap synchronisation, headway optimisation, and path guidance. Each model, therefore, has different following distances, vehicle speeds, critical gaps, etc. The results of the theoretical models are capacity curves depicting the merging traffic capacity over the circulating traffic volumes.
This thesis investigates whether the capacity increase obtained from the most simplified automated vehicle model of the theoretical study can be obtained through microscopic traffic simulation in VISSIM. What must be noted is that the most simplified model of this theoretical study features communication between vehicles, which is not the case in the simulation models of this thesis. Therefore, higher follow-up times must be maintained in the simulation models, whereas the theoretical model uses minimal follow-up times to obtain a capacity increase of 58%. Initially, this thesis aimed to determine the achievable capacity increase for Connected and Automated Vehicles. However, due to limitations in VISSIM, the capacity increase could only be obtained for Automated Vehicles as the software cannot model communication between vehicles. Therefore, this report does not obtain the highest potential capacity increase for automated vehicles but serves as a stepping stone for more complex AV models. It determines the additional merging capacity of automated vehicles on turbo-roundabouts, only considering key parameters such as the car-following behaviour, critical gap and follow-up times. Gap synchronisation, an important aspect of the study of Fortuijn and Salomons, is not included in the simulation models due to shortcomings in VISSIM.

Before making the AV models, this thesis aims to calibrate the VISSIM model for human drivers with real traffic data at a turbo-roundabout in Nieuwerkerk aan den IJssel, the Netherlands. Only when the VISSIM model is capable of accurately modelling human behaviour can it be used to determine the capacity increase through AVs. The thesis starts with developing a human-driven vehicle model to determine the merging capacity of the turbo-roundabout in current traffic conditions. The calibration is conducted on the traffic flow level through an iterative process where the critical gap and follow-up times are adjusted. The root-mean-normalised-squared error (RMNSE) determines the difference between regression lines of the merging capacities of the simulation model and the real traffic measurements. Based on the calculated differences between both lines, the critical gap and follow-up time parameters are adjusted.

For the left lane, the iteration process reduced the RMNSE from 0.427 to 0.072, equaling an improvement of 72.5%. For the right lane, the iteration procedure gave an unsatisfactory result. The RMNSE increased from 0.408 to 0.427, which signifies an increase of 4.7%. The final capacity curve after seven iterations was unrealistic. It was concluded that the VISSIM model performs adequately at simulating realistic behaviour on the left lane but that the same accuracy could not be achieved for the right lane. The calibration procedure is only applied to the minor branch of the turbo-roundabout as the data for the major branch could not be used for calibration.

Subsequently, the calibrated model serves as the basis for developing three automated vehicle models: *cautious, normal*, and *aggressive*, each representing a different aggression level of automated vehicles. Each of these models features a different car-following model to simulate the various levels of aggression. Furthermore, the critical gap times of the human model are changed to model Avs properly, and for each of the three models, a penetration rate of 100% of AVs is applied. For each of the three automated vehicle models, this thesis measures the increase in capacity. It compares it to the theoretically obtained capacity increase of 58% for the left lane of the minor access branch of the turbo-roundabout. This thesis finds capacity increases of 7.47%, 38.62%, and 24.32% for the cautious, normal, and aggressive automated vehicle models, respectively. For the right lane, the obtained capacity increases were not calculated as the final HDV model is inaccurate for this lane. An unexpected result is that the normal automated vehicle model outperforms the aggressive model for the left lane, even though the parameter settings should result in a higher capacity for the aggressive model.

Comparing the resulting capacity increases to the 58% increase of Fortuijn and Salomons, one can conclude that the simulation model does not reach the same increase in capacity as the theoretical model. This is not an unexpected result, as a theoretical simulation model has fewer interactions, and the theoretical model assumes communication between vehicles, whereas the simulation models do not. Therefore, concluding that the microscopic traffic simulation models in VISSIM cannot accurately model the behaviour of automated vehicles is premature. The models leave room for improvement by fine-tuning various parameters that were ignored in this thesis.
This thesis concludes that, while VISSIM is adequate at simulating human-driven vehicles, it has various limitations for modelling automated vehicles. First of all, gap synchronisation and headway optimisation cannot be modelled, while these are the most critical elements to unlock the full potential of automated vehicles on turbo-roundabouts. If one wants to incorporate these into VISSIM, an external mathematical software package is necessary. Considering other software packages for studying (C)AVs on turbo-roundabouts is advised. This thesis is a foundation for future research on turbo-roundabouts, both for the calibration process and for simulating automated vehicles.

Contents

List of Figures

List of Tables

List of abbreviations

t_c critical gap. 15

t_f follow-up time. 15

AVs Autonomous Vehicles. 1

CACC Connected Adaptive Cruise Control. 10

CAVs Connected and Automated Vehicles. 1

HDVs Human-driven Vehicles. 5

OD Origin-Destination. 30

pce passenger-car-equivalent. 25

RMNSE Root-Mean-Normalised-Squared Error. 6

V2I vehicle-to-infrastructure. 3

V2V Vehicle-to-vehicle. 11

W74 Wiedemann 74 car-following model. 17

W99 Wiedemann 99 car-following model. 17

1

Introduction

Since the beginning of the 21st century, automated vehicles (AVs) have garnered more and more attention each year. By now, a substantial proportion of the population has probably had its first experience with automated driving or an automated driving system such as adaptive cruise control. In the 1920s, researchers conducted the first self-driving car experiments in the United States. Since the computer was not invented, the term *self-driving* should be taken lightly as someone remotely controlled these cars from another vehicle (LaFrance, 2016). The idea of vehicles driving entirely autonomously, without human intervention, was seen as science fiction.
In the 1960s, when the first computers were being implemented in society, the fictitious ideas from the twenties suddenly seemed less absurd. They had already established that autonomous vehicles should be able to sense the environment, process this information, and react accordingly. Sadly, the computers at that time could only carry out the final two steps (Weber, 2014). In 1977, the first standalone autonomous vehicle was introduced in Japan. However, it was only in the 90s that major advancements in autonomous vehicles were made. Car brands started to spend an increasing amount of money and energy to develop autonomous driving technologies.
The first autonomous vehicles, such as mining vehicles, combine harvesters and warehouse vehicles, were used for particular purposes. Implementation for these purposes was relatively easy since these vehicles did not have to deal with changing driving conditions and did not encounter much traffic. Due to the increasing capabilities of autonomous vehicles, their application area also started to widen. In 1999, *ParkShuttle*, introduced in Rotterdam, was the world's first publicly accessible autonomous vehicle (Sadler, 2016).
As of 2013, various car manufacturers offer systems such as adaptive cruise control and lane- and parking assistance on some models. Currently, automated vehicles capable of driving with level three (L3) automation are available on the market. L3 implies that a car can drive autonomously but requires human override in certain situations. The vehicle cannot yet navigate itself under all circumstances. These levels of automation are developed by the Society of Automobile Engineers (SAE, 2021; Synopsis, n.d.). While the higher L4 and L5 levels of automation are currently not attainable (partly due to technological limitations, partly due to legislation issues), it is only a matter of time before vehicles of these higher levels of automation will be available to the broader public and completely automated driving will become an embedded part of society. These vehicles will also communicate with one another, hence the name Connected and Automated Vehicles (CAVs).

The technological advancements in automated driving have spurred many academics to research its impacts on society. Many researchers believe that the broad deployment of automated vehicles, from L3 automation and up, has a lot of potential benefits for society, such as safety improvements, increased road capacity, improved social equality, and decreased emissions.
Automated vehicles can potentially increase road capacity, which is the most exciting benefit for this thesis research. The expected capacity increase with automated vehicles is highly dependent on the location in the network and the type of infrastructure; the capacity gain on a single-lane rural road will differ from that of a five-lane highway. Locations with multiple conflicting traffic streams could benefit most from automated vehicles (Goodall et al., 2013; Li et al., 2013). This thesis only focuses on the obtainable capacity increase through automated driving at turbo-roundabouts.
A turbo-roundabout is a multi-lane roundabout developed by Dr. Ir. L.G.H. Fortuijn in the late 1990s. It is considered a safer and more efficient alternative to regular multi-lane roundabouts. It has a few distinctive characteristics, which are:

1. From at least one access branch, priority has to be given to more than one lane on the roundabout.
2. All branches must prioritise at most two lanes on the roundabout.
3. Weaving/crossing conflicts are absent due to consequently applied spiral lines.
4. Unnecessary steering movements are limited due to a smooth course of the spiral lines
5. A designated lane enables drivers to exit the roundabout or proceed towards the next exit at each roundabout section.
6. At least two egress branches consist of two lanes.
7. Surmountable raised lane separators prevent corner cutting.
8. A small roundabout diameter.
9. The branches join the roundabout radially.
10. Surmountable pavement offers sufficient space for long vehicles to traverse the roundabout.

The first four conditions are mandatory and must all be satisfied for a roundabout to classify as a turbo-roundabout. If conditions five and six are unsatisfied, the roundabout is only a partial turbo-roundabout. The remaining conditions are essential for the safety of a turbo-roundabout but are not required to receive the label 'turbo-roundabout' (Fortuijn, 2013). Figure 1.1 provides an aerial overview of a turbo-roundabout.

The dichotomy between researchers makes investigating the impact of AVs on traffic performance at turbo-roundabouts so exciting and relevant. Some researchers believe that (turbo-)roundabouts might become obsolete due to the introduction of AVs. As a possible argument, they mention that one of the most significant advantages of roundabouts, higher safety, may diminish because automated vehicles have a lower risk of crashing. The economic benefits of roundabouts could thus diminish (Lawson, 2018). This group believes that regular intersections could replace all roundabouts, considering a fully automated future.
On the other side of the spectrum, some researchers still believe in the validity of roundabouts. Where the sceptics primarily focus on the safety side of the story, the believers pay more attention to other aspects. One of the advantages of turbo-roundabouts over regular intersections is the reduced number of conflict points. Where an intersection has 24 conflict points, a roundabout only has four (SWOV, 2022). For two-lane intersections and roundabouts, the amount of conflict points decreases from 43 to 16. A turbo-roundabout decreases this number even further to 10. Figure 1.2 shows a schematic comparison of the conflict points of a roundabout and intersection.
Furthermore, vehicles drive at a lower speed on the roundabout and have a lower risk of side-by-side accidents (Guerrieri et al., 2012).

Figure 1.1: Aerial overview of a turbo-roundabout in Nieuwerkerk aan den IJssel, the Netherlands

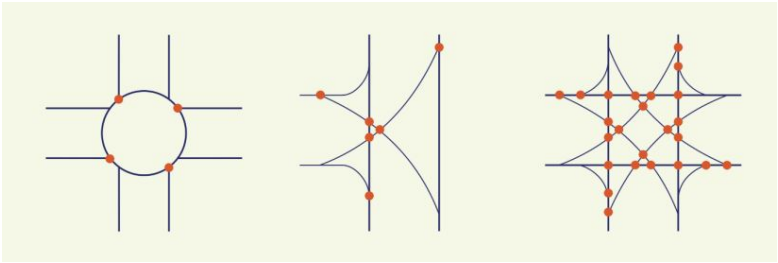


Figure 1.2: Comparison of the number of conflict points, indicated by the red dots, for roundabouts, three-legged and four-legged intersections (SWOV, 2022)

The combination of a lower number of conflict points at a turbo-roundabout, and a higher spatial separation between those points could be advantageous in a fully automated future. After all, fewer conflict points mean fewer conflicting traffic streams must be synchronised, potentially leading to higher capacities under fully automated driving conditions. According to the literature, turbo-roundabout capacity could increase more than regular intersections. Various papers state that intersection capacity under fully automated conditions can be increased by 127, 99.5, and 138%, respectively, compared to current intersections (Amouzadi et al., 2022; Lefkowitz, 2019; Sun et al., 2018). On the other hand, a different paper addressing turbo-roundabout capacity analytically determines that, by only using vehicle-to-infrastructure (V2I) communication, the capacity increase on a turbo-roundabout could be as high as 150% (Fortuijn and Salomons, 2020). This capacity increase is higher than any of the increases found for regular intersections. According to the authors, lower speeds on the turbo-roundabout simplify the gap estimation, and a higher spatial separation of conflict areas allows vehicles from conflicting directions to enter the roundabout simultaneously. With this, they infer that the obtainable capacity improvement of turbo-roundabouts could be higher than regular intersections.

This thesis investigates whether the theoretical capacity increase of 58.3% obtained by Fortuijn and Salomons can be approached through simulation. After all, it is interesting to research the potential capacity increases with the current infrastructure rather than replacing all roundabouts with intersections. Section 1.1 describes the relevance of this topic, Section 1.2 describes the research questions and contents of this thesis report, and Section 1.3 gives an outline of the complete report.

1.1. Thesis relevance

1.1.1. Research gap

While the amount of researchers working on AVs is ever-increasing, the amount of conducted research focusing on turbo-roundabouts is limited. There are quite a few papers addressing the behaviour at single-lane roundabouts. However, only very few papers describe the impact of automated vehicles on traffic at turbo-roundabouts. The potential traffic capacities under fully automated conditions are usually not the main point of interest for papers focusing on turbo-roundabouts. Their primary focus is often on safety under mixed-traffic conditions rather than capacity effects.
Only a single paper addresses the possible impact of automated vehicles on the capacity of a turbo-roundabout (Fortuijn and Salomons, 2020). The authors analytically calculate the achievable capacity increase, leading to theoretical values. No research paper tries to determine the achievable capacity gains of automated vehicles on turbo-roundabouts through simulation.
The research gap extracted from the literature is the need to investigate whether turbo-roundabouts' theoretical capacity increase can be obtained (or approached) through traffic simulation. Furthermore, no papers use field-collected data, eliminating the possibility of calibrating the simulation results. These should be considered with more care than calibrated models. Thus, a calibrated model is also missing from the literature. Lastly, the existing research papers mostly look at the transition phase, where human-driven- and automated vehicles use the network simultaneously. Papers looking at fully automated conditions are scarce.
To summarise, a possible addition to the current literature is a traffic simulation model that can be calibrated on real data of human-driven vehicles and a model that only considers fully automated driving conditions.

1.1.2. Relevance

There are several reasons why the contents and results of this thesis are relevant. The first reason is that, at this point, it is inevitable that society will have a future with automated vehicles. Car manufacturers are at a stage where vehicles can drive autonomously in most traffic conditions. Soon, automated vehicles will thus be driving on turbo-roundabouts, which makes research on their behaviour at turbo-roundabouts to achieve the highest possible merging capacity desirable. Since only a few papers touch on this topic, there is plenty to investigate. This thesis fills part of the existing research gap by developing a calibrated simulation model of an existing turbo-roundabout and investigating the merging capacity increase under fully automated driving conditions. The initial goal of this thesis was to investigate the capacity of CAVs, but due to limitations in the used software package, this could not be achieved, Therefore, the focus shifted towards AVs and their respective capacity increase on a turbo-roundabout.
This report can serve as a stepping board for more advanced research in this field and aid future researchers in choosing an appropriate modelling approach in VISSIM. Based on findings from this report, they can make better decisions in the modelling process. It is the first step in determining the optimal control of automated vehicles on turbo-roundabouts to obtain the highest achievable capacity. The capacity outcomes of this report need optimisation, leaving room for improvement. This report can also be an example of model calibration based on traffic volumes. Usually, model calibration happens through queue lengths, vehicle speeds, or average delays. Due to limited data, this report explores other ways of calibrating a microscopic traffic simulation model in VISSIM.

This report is meant for researchers planning to research the capacity of turbo-roundabouts through microscopic traffic simulation in VISSIM. The limitations of VISSIM are explored and elaborated in Chapter 6; the lessons learned in this report are valuable for future research using the microscopic traffic simulation software of PTV VISSIM. Furthermore, this report is directed to people with a keen interest in the effects of automated vehicles on traffic performance at turbo-roundabouts.

1.2. Research questions and thesis contents

This section presents the research questions and elaborates on the contents of this research thesis in more detail.

1.2.1. Research questions

The research in this thesis consists of two major parts. The first part of the paper discusses the development of a calibrated model for human-driven vehicles, while the second part of the research focuses on developing three models for automated vehicles on a turbo-roundabout. The second part answers the main research question, which is formulated as follows:

What capacity increase can be achieved by AVs on turbo-roundabouts through microscopic simulation, only considering automated vehicle car-following- and merging behaviour?

Due to the complexity of this research question, an additional subquestion is formulated to help in answering the main research question:

- *What are the key parameters defining human driving behaviour at a turbo-roundabout in a microscopic traffic simulation capacity model, and how can these be used for model calibration, ensuring an accurate representation of the simulation to real-world data?*

1.2.2. Contents

The majority of this thesis revolves around a microscopic simulation model in VISSIM. Microscopic simulation models mimic human behaviour for individual vehicles rather than aggregate behaviour on a network level. Applying a microscopic simulation model makes sense since individual vehicle interactions are essential for traffic performance at a turbo-roundabout. Furthermore, such models can also simulate the behaviour of AVs, making them suitable for answering the main research question. VISSIM, however, does not allow the modelling of CAVs without interference of an external program, coupled through the COM interface. It can only simulate AVs.

In the first part of the research, a microscopic simulation model representing the current traffic conditions of human-driven vehicles (HDVs) is constructed in VISSIM, a computer program designed explicitly for microscopic traffic simulation. From now on, this model is referred to as the HDV model.
An existing turbo-roundabout in Nieuwerkerk aan den IJssel, the Netherlands, is the research subject of this thesis, as Dr. Ir. L.G.H. Fortuijn collected real traffic data measurements for this turbo-roundabout. For the major and minor access branches, Fortuijn measured traffic volumes for entering and circulating traffic lanes, allowing calibration of the model.
Before conducting the calibration, the model needs to be constructed. An extensive literature review is done to mimic the behaviour of vehicles on the turbo-roundabout properly. From the literature, the most critical behavioural parameters that define human driving behaviour are obtained. These parameters relate to car-following behaviour, merging behaviour, and speed and are programmed into the microscopic simulation model. Next to human behaviour, the roundabout is modelled to represent reality best. Due to the limitations of the software, modelling the roundabout just like real life introduces unrealistic behaviour. Some specific measures are taken to prevent this unrealistic behaviour. These are described later. Once the model is set up, the iterative calibration process starts. The HDV model is calibrated based on the measured traffic conditions at the turbo-roundabout. The calibration process is the most important step of this thesis and, therefore, takes up a large part of the report.

Regression curves are made for the simulation results and the data from the turbo-roundabout in Nieuwerkerk-aan den IJssel (from now on referred to as the Nieuwerkerk data). The Root-Mean-Normalised-Squared Error (RMNSE) measures the deviation between real-life data and the simulation model. By tweaking the follow-up and critical gap times through an iterative process, an attempt is made to minimise the RMNSE and improve the fit between the simulation model and the Nieuwerkerk data. This iterative process is conducted separately for the left and right lanes.

Once the model sufficiently represents reality, the automated vehicle models simulate the behaviour of AVs on the turbo-roundabout. VISSIM already contains specific functions to simulate the car-following behaviour of automated vehicles. These depend on the aggression of the automated vehicles and range from defensive to aggressive -their parameters are based on research of the European CoExist project. These various car-following behaviours and other vital parameters from the literature are used to obtain the capacity of the turbo-roundabout for various aggression levels of automated vehicles. The results are then compared to the results of the calibrated model to formulate conclusions about the potential capacity increases of AVs on turbo-roundabouts, only considering the most essential parameters for automated driving.
The resulting capacity increases do not portray the ultimate potential of automated vehicles. Vehicles should be connected to use the potential of AVs to the fullest extent, hence CAV. This enables gap synchronisation and headway optimisation, further increasing the merging capacity at a turbo-roundabout. By communicating with other CAVs, vehicles can adjust their arrival times at the turbo-roundabout based on those of other vehicles approaching it. This allows them to merge into the gaps of other traffic flows, leading to the highest potential merging capacity. VISSIM cannot model these features; coupled to VISSIM through the COM interface, an external controller is necessary to program this behaviour. This is the next step in obtaining the potential capacity improvements for turbo-roundabouts and is left for future research. This report only investigates the capacity increase for AVs obtained solely through automated car-following behaviour, critical gap and follow-up times. This report also recommends further research and synthesises the lessons learned during this thesis.

1.2.3. Scope

The scope of this report consists of developing a calibrated microscopic traffic simulation model for HDVs in VISSIM, upon which the AV model builds to predict the capacity increase of AVs on turbo-roundabouts. It is important to note that each model only features one vehicle category. The calibrated model only contains human-driven vehicles, whereas the models investigating the capacity increase contain AVs only. It eliminates interactions between human-driven vehicles and AVs, for which there is no consensus in the literature about how this will be. This report only considers regular person cars; vehicles such as trucks, buses, and motorbikes are not part of the models. The impact of other road users, such as cyclists and pedestrians, on the capacity is also out of the scope of this research.

As mentioned in the previous subsection, this report does not take all necessary research steps to obtain the optimal roundabout capacity under automated conditions. VISSIM cannot model communication between automated vehicles without external mathematical software, and programming this external software is outside the scope of this thesis.

Furthermore, the research does not include the lateral behaviour of vehicles in VISSIM. The lateral behaviour of vehicles on the turbo-roundabout has no impact on the merging capacity and is, therefore, not considered. It is included in the VISSIM software package, but its parameters have remained the same as standard.

Lastly, this research does not consider the sham conflict. The sham conflict occurs between vehicles exiting the roundabout at the same branch as vehicles trying to merge. As the merging vehicle is often unsure about the intentions of the vehicle on the turbo-roundabout, it will be more hesitant to merge onto the roundabout and leave a larger space. Including the sham conflict would decrease the merging capacity, but this thesis disregards it as it is unclear how this can be accurately modelled in VISSIM.

1.3. Outline of the thesis report

This master thesis is structured as follows. Chapter 2 conducts an extensive literature review; in the first part of the literature review, the literature describing how to determine the capacity of a turbo-roundabout is given. Then, the literature describes the VISSIM elements with the most significant impact on the capacity. Next, the literature about the behaviour of HDVs on turbo-roundabouts is elaborated. Lastly, literature regarding the behaviour of AVs at (turbo-)roundabouts is described.
In Chapter 3, the methodology of this thesis explains the most critical steps in detail:

1. A description of the general preparations
2. The construction of the HDV model, followed by an extensive section about the calibration of said model.
3. The construction of the AV models, followed by a short description of how the capacity increase of the AV models is measured compared to the HDV model.

Chapter 4 describes the settings of the HDV model in more detail. The initial simulation results, before calibration, are also shown in this chapter. Then, the iterative calibration procedure is shown, followed by the parameter values and results of the calibrated HDV model. The chapter concludes with an extensive analysis of the calibration procedure and the obtained calibrated model.
Chapter 5 describes the AV models in more detail. Just like in Chapter 4, the parameters of the models are first depicted, followed by the simulation outcomes for each model. These are then compared to one another, and the capacity increases of all AV models are calculated. The chapter concludes with an extensive analysis of the capacity curves of the models and obtained capacity increases for each model.
Finally, Chapter 6 concludes this thesis and formulates an answer to the research questions. Furthermore, it contains the limitations of this thesis and makes recommendations for future research.

2

Literature review

This chapter provides the reader with an extensive overview of all relevant literature for this thesis. It extracts and summarises the most important and relevant findings from each paper. This chapter is divided into multiple sections, each describing a different part of the literature. Section 2.1 provides an overview of the literature that defines the research gap and gives a more thorough description. Next, Section 2.2 elaborates on the definition of *'roundabout capacity'* and summarises the parameters having the most impact on the outcome of microscopic traffic simulations of (turbo-)roundabouts.
Once it is known which elements need to be fine-tuned in VISSIM, Section 2.3 shows the literature describing the magnitude of these parameters. Section 2.4 shows the literature describing the validation/calibration of a microscopic traffic simulation model. Lastly, Section 2.5 shows the literature that describes the behaviour of automated vehicles on turbo-roundabouts.

2.1. Current literature and research gap

The literature assessing the capacity of turbo-roundabouts with AVs is minimal. Only a few papers address car behaviour at turbo-roundabouts, probably because turbo-roundabouts are a relatively recent invention and are not (yet) widespread across the globe. In 2023, there were 777 turbo-roundabouts worldwide and 415 in the Netherlands (De Baan, 2023). The papers addressing the impact of automated vehicles on regular roundabouts are more present and relevant for turbo-roundabouts; some aspects can be used for both roundabouts. Therefore, these are also considered in this literature review. The literature is divided into two groups, one for single-lane roundabouts and one for turbo-roundabouts.

2.1.1. Literature for single-lane roundabouts

Only one paper investigates the impact of automated vehicles using actual traffic data from an existing roundabout in their research (Boualam et al., 2022). While this report does not determine the capacity, it evaluates what penetration rates of AVs are necessary to offset a given traffic volume increase. They do this using a microscopic traffic simulation model constructed in VISSIM. The authors investigate the impacts of AVs on the roundabout performance based on the queue lengths of the access branches. A queue length of 25 metres is considered an acceptable level of service. The paper concluded that for volume increases of 10, 20 and 30%, a respective penetration rate of 40,60 and 80% is necessary to keep the queue length within service boundaries.

However, the paper only considers Connected Adaptive Cruise Control (CACC) for automated vehicles, therefore only correctly modelling the car-following behaviour. They state that the modelling of CACC in VISSIM is accurate but that other automated driving behaviours cannot be modelled. Furthermore, they state that the gap times they used in their research might not be acceptable for human passengers. While it may be technologically possible, occupants may not accept the small gaps. When people are in control, they accept smaller gaps than when they are not.

A second paper implements a microscopic traffic simulation model to investigate the impact of CACC on roundabout efficiency in terms of delay (Tumminello et al., 2022). The automated vehicles are implemented in the network through adjustment factors for AVs (taken from the Highway Capacity Manual), applied to the entry capacity equations. They are thus not directly modelled but rather simulated by changing the entrance capacities of the roundabout. Their results showed that introducing AVs in traffic can create a general performance improvement for roundabouts. They determined the entry capacities for single-lane entries and the left lane of two-lane branches based on the traffic volume of the circulating traffic flow on the roundabout.

Another paper investigates the implications of coordinating AVs under mixed-traffic conditions to achieve a smooth traffic flow (Zhao et al., 2018). They simulated their experiment using different penetration rates of AVs. They concluded that the total travel time and fuel consumption could be reduced by as much as 51% and 35%, respectively, assuming a penetration rate of 100%. An interesting finding from their study was that the presence of HDVs has a significant impact on the overall performance of the roundabout. Even at high penetration rates of automated vehicles (80%), the impact of HDVs is significant. Only at full penetration can the benefits of automated vehicles reach their full potential.

Another paper, also using a simulation-based approach, features a calibrated car-following model of a single-lane roundabout and varying penetration rates of AVs while using a central controller in the simulation for communication (Mohebifard and Hajbabaie, 2020). The results show that for a penetration rate of 100%, the traffic operations in both under- and semi-saturated conditions improve. However, in oversaturated conditions, AVs lead to an increase in delays. After optimising the AV trajectories, they found a great improvement in the roundabout in general and a significant decrease in delays.

Bakibillah et al. wrote a paper focusing on the design of a control system for automated vehicles on roundabouts (Bakibillah et al., 2021). The proposed system works at two levels to minimise the time for all approaching vehicles to approach a roundabout. Vehicles approaching the roundabout are joined into clusters at the higher level, and the lower level determines the optimal arrival time of these clusters through a receding horizon control approach. They concluded that the control system improved fuel consumption by 35%, travel time by 51% and total delay by 100% at a penetration rate of AVs of 100%. An interesting comment in this paper is that delays in communication between vehicles can have a significant impact on the performance of the system.
A second paper on this bi-level control system states that the clustering zone is 200-60 m upstream of the merging point and that the merging execution zone starts 60m upstream and ends at the merging point (Bakibillah, Kamal, Susilawati, and Tan, 2019). After reading the two similar papers, how the system works still needs to be clarified. Other than a broad description of the system, its details are unknown.

A third paper proposes a similar control method but does not use platooning (Danesh et al., 2021). Instead, they use a mixed-integer programming model that optimises arrival times at the roundabout in the first control stage and the vehicle trajectories on the roundabout in the second stage. He used the parameters of driving behaviour, as mentioned in the CoExist project.

In their results, the authors stated that the roundabout's throughput and delay improved compared to a non-controlled roundabout. They applied their research to a synthetic roundabout with no actual traffic data.

The next paper uses a central controller, just like the previous papers (Martin-Gasulla and Elefteriadou, 2021). However, its use is different, managing conflicts on the roundabout. It aims to improve traffic performance by safely and efficiently managing conflicts. The central controller prioritises vehicles on the access branches based on strategies dependent on the demand. In total, the paper looks at seven different strategies to process the vehicles and assesses the effectiveness of each one. The performance measures they used for their research are the throughput and control delay. They simulated their strategies using an actual roundabout with synthetic traffic volumes. Irrespective of the scenario, the designed controller increases the throughput of the roundabout. Under fully automated conditions, the traffic flow is increased by 58 to 73%. Prioritising the vehicles on the circulatory roadway for the longest time is considered the best coordination of vehicles on the roundabout. However, these results should be handled with care since the data used in the experiment was manipulated because all vehicles were given enough space before entering the network, and no vehicles were too close to one another. In reality, this will seldom occur, so the results must be considered less realistic. Whereas previous papers worked with V2V communication exclusively, the authors of this paper state that it is possible to achieve the desired outcome using either Vehicle-to-vehicle (V2V) or V2I communication.

Whereas most research papers opt for a simulation approach, the next one does not and opts for an optimisation framework (Mohebifard and Hajbabaie, 2021). Through objective functions, they optimise the behaviour of AVs on single-lane roundabouts. The decision variables are the acceleration, and the state variables are the speed and position of the CAVs. In the objective function, the travelled distance and acceleration magnitudes are minimised. The problem is divided into two sub-problems; the first sub-problem includes the vehicle dynamics constraints, and the second one uses the solutions of the first problem to create a collision-free region. The paper uses an actual roundabout and studies it under different demand levels. The results show that total travel times decreased by 9.1% to 36.8%. In contrast, the average delays decreased by 95.8% to 98.5% compared to a base case with human-driven vehicles. The average speed of the vehicles also shows a significant increase of 12.8% to 100.5%.

The following paper focuses more on the behaviour of automated vehicles on roundabouts while keeping in mind safety aspects instead of looking into achievable capacity increases. It presents a steering method for automated vehicles, just as in previous papers, but aims to make AVs behave more like humans (Rodrigues et al., 2018). The authors propose an adaptive tactical behaviour planner (ATBP) to plan human-like behaviour for automated vehicles by combining naturalistic behaviour and a tactical decision-making algorithm. They conducted a driving simulator test to gather the necessary data for naturalistic driving behaviour. Their results showed that automated vehicles outperformed most human driver participants on average travel time and had better control of the longitudinal acceleration and, thus, more comfort.

2.1.2. Literature about turbo-roundabouts

Only three papers investigate the effects of automated vehicles on turbo-roundabouts. Two of these papers primarily focus on the safety aspects of a turbo-roundabout. The first paper investigates the performance of a turbo-roundabout regarding safety in a mixed-traffic environment (Giuffrè et al., 2021). They conduct the research through a microscopic simulation in VISSIM, where the vehicle trajectories are analysed. They concluded that conflicts on a turbo-roundabout can be decreased by 83% when AVs account for 25% of total traffic.

The second paper also uses a microscopic simulation approach for mixed-traffic conditions to investigate the performance of turbo-roundabouts in terms of safety (Severino et al., 2021). The paper tries to find the traffic conditions determining maximum performance if the vehicle flow consists of conventional vehicles for 75% and connected automated vehicles for the remaining 25%. Their research used a calibrated VISSIM model and concluded that the turbo-roundabout performed best when the traffic is 25% crossing, 60% right turning and 15% left turning.

The last paper investigates the capacity gain achieved under fully automated conditions (Fortuijn and Salomons, 2020). This paper is the most relevant for this thesis, as it analytically determines what capacity increases are achievable on both roundabouts and turbo-roundabouts by utilising V2I communication. They investigate the obtainable capacity gain by applying two merging strategies: clustering and zipping. They concluded that zipping is the most promising merging strategy for a single-lane roundabout. An optimal capacity increase can be obtained if the arrival times of vehicles at the roundabout are automatically adjusted so that the headway distribution is optimal for zipping. For turbo-roundabouts, however, the paper states that zipping will not suffice since vehicles using the inner lane must cross the outer roundabout lane, thus creating the need for gap extension. For synchronisation, they state that the distances between two successive conflict points must be covered simultaneously on the inner and outer lanes.

2.1.3. Research gap

While there has been much research on the impact of automated vehicles, the literature still leaves a few specific gaps to explore. From Table 2.1, one can notice that each paper lacks a few elements. By combining the missing elements, a few observations can be made. First, most papers use synthetic traffic data, thus making it impossible to calibrate those models. Results from a non-calibrated model need more careful consideration than those from a validated model. Therefore, calibration is a precious part to include in research. The paper that does use actual traffic data uses it for a purpose other than calibration. Therefore, no paper features a calibrated microsimulation model of a (turbo-)roundabout.

A second remark about the research papers is that only very few make statements about the capacity increase that AVs can achieve on roundabouts. There are a few papers that state that roundabout performance can increase. However, they do not make claims about the magnitude of these performance improvements (Bakibillah et al., 2021; Bakibillah, Kamal, Susilawati, and Tan, 2019; Boualam et al., 2022; Giuffrè et al., 2021; Mohebifard and Hajbabaie, 2020; Tumminello et al., 2022; Zhao et al., 2018). The papers that make claims about the obtainable capacity only consider CACC (Giuffrè et al., 2021; Martin-Gasulla and Elefteriadou, 2021; Mohebifard and Hajbabaie, 2021; Rodrigues et al., 2018). They, therefore, do not unlock the full potential of automated driving on roundabouts.
The only paper investigating the obtainable capacity increase on a turbo-roundabout does so using an analytical approach (Fortuijn and Salomons, 2020). Turbo-roundabouts are underexposed in literature and the expected capacity increases even more under automated conditions. The available literature on turbo-roundabouts mostly focuses on behavioural parameters and theoretical capacity estimations. No research papers investigate the potential capacity increase on turbo-roundabouts through microscopic simulation.
Lastly, most of the discussed papers consider mixed-traffic conditions. The disadvantage of this approach is the high uncertainty about the interaction between HDVs and AVs. Therefore, several assumptions have to be made that could significantly influence the final results. A model with only AVs or HDVs needs fewer assumptions that generate more trustworthy outcomes.

Combining these observations, one can formulate the following clear research gap. What currently needs to be added to the literature is a paper addressing the capacity increase AVs can achieve on turbo-roundabouts by designing a calibrated microscopic simulation model.

Table 2.1: Overview of research gaps per paper

Title	Research gap
Simulation-Based Analysis of "What-If" Scenarios with Connected and Automated Vehicles Navigating Roundabouts	Only entry capacity is changed to model CAVs, only CACC included, single-lane roundabout, no capacity calculations nor model calibration
Safety Evaluation of Turbo-Roundabouts with and without Internal Traffic Separations Considering Autonomous Vehicles Operation	Do not determine the capacity and look at mixed-traffic conditions on single-lane roundabout
Impact of Autonomous Vehicles on Roundabout Capacity	Mixed traffic conditions, only consider CACC and do not look at capacity increases, single-lane roundabout
Effects of Automated Vehicles on Traffic Operations at Roundabouts	No capacity calculations, mixed traffic conditions, fictitious traffic and roundabout
Optimal Control of Connected and Automated Vehicles at Roundabouts: An Investigation in a Mixed-Traffic Environment	No capacity calculations, fictional roundabout and traffic, single-lane roundabout
Bi-level coordinated merging of connected and automated vehicles at roundabouts	Single-lane roundabout, no capacity estimation, fictitious traffic
The Optimal Coordination of Connected and Automated Vehicles at Roundabouts	Single-lane roundabout, no capacity estimation, fictitious traffic, bad explanation of control system
Optimal roundabout control under fully connected and automated vehicle environment	Fictitious single-lane roundabout and traffic, no capacity estimations or quantifiable conclusions
Traffic Management With Autonomous and Connected Vehicles at Single-lane Roundabouts	Fictitious, manipulated traffic data, single-lane roundabout, short simulation periods
Connected automated vehicle control in single lane roundabouts	Fictitious traffic volumes, single-lane roundabout
Autonomous Navigation in Interaction-Based Environments—A Case of Non-Signalized Roundabouts	Mixed-traffic environment, fictitious roundabout, no capacity calculations
Safety Evaluation of Turbo-Roundabout Considering Autonomous Vehicles Operation	No capacity determination but only optimal traffic distribution for maximum safety, mixed-traffic conditions
Capacity Increase through connectivity for the i-Roundabout and i-Turbo roundabout	Analytical approach instead of micro-simulation, only V2I communication considered

2.2. Capacity of a turbo-roundabout

Before being able to make any claims about the capacity increase with AVs on turbo-roundabouts, it is essential to have a clear definition of the phrase 'capacity of a turbo-roundabout'.

A first paper considers the Tanner equation, adjusted by Brilon, the NCHRP method for the right-turn lane, and the Harders and NCHRP method for the through- and left-turn lanes (Guerrieri et al., 2012). Their paper does not mention whether they apply to the major or minor access branch. However, judging from the context, it is believed to be applied to both branches. They state that the NCHRP model overestimates the capacity when the circulating flow is high. The paper states that the entry capacities of individual lanes cannot simply be added to one another to calculate the total entry capacity of a turbo-roundabout. It can only be done when the degrees of saturation on both lanes are equal. They calculate the effective entry capacity through a formula if they are unequal.
A second paper uses the equations of Tanner, adjusted by Brilon, for the capacity of the right-turn lane and the Harder model for through- and left-turn lanes of the minor roundabout branch (Guerrieri and Granà, 2009). For both lanes of the major branches, they propose the theoretical method of Brilon-Wu. The right-turn lane shows a linear relation between the entry and circulating capacities. In contrast, the through and left-turn lanes show a quadratic relation. As in the previous paper, the entry capacity is not obtained by adding the capacity of both lanes. They state, "If saturation is reached at one of the lanes, operational conditions typical of capacity occur at the entry, thus causing queuing at either one or both entry lanes due to excess demand". The formulas they use are the same as the previous paper.

L.G.H. Fortuijn wrote two papers on the theoretical capacity estimations of access branches on a turbo-roundabout. These papers are the most promising and valuable for this thesis. In the first paper, he distinguishes between conflict stream and gap acceptance models (Fortuijn, 2009). He considers the Bovy model for the first group model while considering Tanner, Siegloch, Troutbeck, and Fisk in the second group. The Troutbeck model is an expansion on the Tanner model, whereas the Hagring model is an expansion on the Fisk model.
For turbo-roundabouts, he states that the gap acceptance models of Hagring and Fisk take the difference between both circulatory lanes into account and that this is important for the accuracy of capacity models. However, most capacity models at the time did not consider sham conflicts that occur at the branches of the roundabout. The sham conflict occurs between vehicles exiting the roundabout at the same branch as vehicles trying to merge. When the merging vehicle has the impression that the exiting vehicle will continue its journey on the roundabout, it will give priority to a vehicle that will not be in the merging section. Fortuijn chose to adapt the capacity models to consider this sham conflict.
In his PhD dissertation, he explains this process in more detail (Fortuijn, 2013). After applying several approaches, he noticed that the impact of the sham conflict is dependent on the origin of the vehicle exiting the roundabout. A vehicle from the previous branch negatively influences the merging capacity.
In contrast, a vehicle from the opposite branch or further positively impacts the entrance capacity. The merging vehicle then knows that the car will exit the roundabout, thus creating a merging gap. He noticed that since all models would need to be changed to consider the sham conflict, the Troutbeck and Hagring model would be the easiest to adapt. The models of Tanner and Fisk can be obtained by inserting a specific value into his adapted formulas. He used the Troutbeck method for the major access branch and the Hagring model (different variations for each lane) for the lanes on the minor access branches. The capacity models were calibrated based on actual data, making them reliable. Table 2.2 gives an overview of the literature and their used capacity models.

From the papers, one can extract that the capacity of a turbo-roundabout depends on the traffic on the turbo-roundabout's circulatory lanes. It is the maximum flow rate that can enter a turbo-roundabout from one of the access branches. The capacity of a roundabout is thus a function with the maximal entrance capacity as the dependent variable and the circulating traffic volume as the dependent variable. The capacity is generally calculated for individual branches because there is a difference between the major and minor access branches. For example, the left lane on a minor access branch only has a crossing conflict, unlike a crossing and merging conflict on a major branch. When looking at one access branch, the individual lane capacities are also different. Since each lane serves different driving directions and encounters different conflicts (in the main direction of the turbo-roundabout, the left lane has a crossing and merging conflict while the right lane only has a merging conflict), the capacities of these lanes are different. Therefore, the left lane is decisive in determining the merging capacity of a turbo-roundabout.
According to some papers, the merging capacities of both lanes of the same branch cannot be added to one another to obtain the total merging capacity of that branch (Guerrieri et al., 2012; Guerrieri and Granà, 2009). They state that each lane's saturation rate should be considered. Only when it is the same can the values be added to one another to obtain the entrance capacity.
The various theories described in this section are not used in the remainder of this thesis, they are only used to obtain an understanding about how the capacity of a turbo-roundabout is determined.

Table 2.2: Overview of the various capacity models for the access branches, obtained from the literature

Lane	Guiffre	Guerrieri	Fortuijn
Major branch			
Right-turn	Brilon-Wu	Tanner, Brilon-Wu	Adapted Troutbeck
Through and left-turn	Brilon-Wu	Harder model	Adapted Troutbeck
Minor branch			
Right-turn	Tanner, Brilon-Wu	Tanner, Brilon-Wu	Hagring^{CS}
Through and left-turn	Harder model	Harder model	Hagring^{CKHR}

2.2.1. VISSIM elements with the biggest impact on simulation outcome

Now that it is clear how one can measure the capacity of a turbo-roundabout, it is essential to know which model parameters ultimately define this capacity. According to a paper, the critical gap and follow-up times are the most critical parameters for the performance of a turbo-roundabout capacity model (Boualam et al., 2022). The critical gap (t_c) is defined as the smallest time gap [s] between the back of the first vehicle and the front of a following vehicle on a major road that a merging vehicle from a minor road deems acceptable to merge between two vehicles. The follow-up time (t_f) is the average time gap [s] between two successive vehicles from a minor traffic stream merging into a major road. A small change in these values could have a substantial impact on the capacity of the roundabout. Since critical gap and follow-up time are both parameters associated with the conflict area in VISSIM, one can conclude that the conflict areas are the most critical part of the microscopic traffic simulation model.

A second paper also states the same (Li et al., 2013). This paper mentions that the operational performance of roundabout simulation models mostly depends on the settings of the following elements: priority rules/conflict areas, reduced speed areas, and the Wiedemann car-following models. A third paper confirms that the essential part of modelling roundabouts in VISSIM is properly formulating the conflict areas (Gallelli et al., 2017). A last paper also states that critical features of VISSIM to set for a correct simulation are vehicle speeds, priority rules, traffic assignment, and driver behaviour (Vaiana et al., 2013).

When combining the findings of the papers, it is evident that these major elements in VISSIM have the largest impact on the capacity results of the simulation: conflict areas, reduced-speed areas, and car-following behaviour, and it appears that the conflict areas are the most crucial element. Relevant literature on these elements is described in Section 2.3.

2.3. Human Behaviour at Turbo-Roundabouts

This section describes essential simulation elements that model human behaviour on turbo-roundabouts. The section is divided into three subsections, each describing a different parameter.

2.3.1. Merging behaviour

The merging behaviour applies to vehicles in conflict areas, defined as the areas where vehicles from conflicting traffic directions cross paths. For turbo-roundabouts, these are the locations where merging vehicles could potentially collide with vehicles on the roundabout itself. In Figure 2.1, green and red areas depict the conflict areas. The critical gap t_c and the follow-up time t_f are the most important parameters determining the merging behaviour. Numerous papers investigate the values of the critical gap and the follow-up time. A summary of their findings is given before an overview of the critical gap and follow-up times of the various papers are depicted in Tables 2.3 and 2.4.

The first paper observed traffic movements at a turbo-roundabout in Maribor, Slovenia, using cameras to analyse traffic and obtain the critical gap and follow-up times. They first looked at the kinematic parameters like speed and acceleration on each branch and the circulatory lanes (Guerrieri et al., 2018). Since one cannot directly observe the critical gap time, they estimate them based on the largest rejected and accepted gaps. Logically, the critical gap lies between both values. Given the calculated critical gaps, they noticed that an Erlang probability density function best describes the data.
The follow-up times were directly obtained from the traffic data and showed an inverse Gaussian distribution. They determined the follow-up and critical gap times for the four access lanes: major right and left and minor right and left. Table 2.4 depicts the average values for the critical gap for all lanes. One should note that the turbo-roundabout has a different shape than other turbo-roundabouts. The driving radius is smaller in its main direction, allowing higher driving speeds. The obtained values for the critical gap and follow-up time should thus be handled with care since they can differ from a circular turbo-roundabout. The critical gap and follow-up time are likely higher on a regular turbo-roundabout.

A second paper describes the validation process of a VISSIM model (Li et al., 2013). They validated their model by comparing the measured critical gap and follow-up times with Brilon's theoretical values. The paper states that parameter values based on field data and quantitative calibration guidelines are more helpful in facilitating fast and accurate modelling. They investigated a roundabout in Wisconsin with video cameras to capture arrivals, entries and exits on the roundabout. With the maximum-likelihood method, they derived the critical gap and follow-up headways. The values they obtained apply to the left lane of the minor branches only.

In a fifth Rural Roads Design Meeting presentation, Bondzio states the follow-up time and critical gap determined by Brilon (Bondzio, n.d.). Lastly, Fortuijn determined the values of the critical gap and follow-up times by fitting a function through an actual data set, which results in calibrated values for t_f and t_c. Once again, the tables contain the obtained values from these papers.

As seen in Tables 2.3 and 2.4, the follow-up times are fairly similar in all research papers. The critical gaps, however, show bigger differences between the different papers. While most papers estimate the critical gap at around 4-4.5 s, Fortuijn's values are lower.

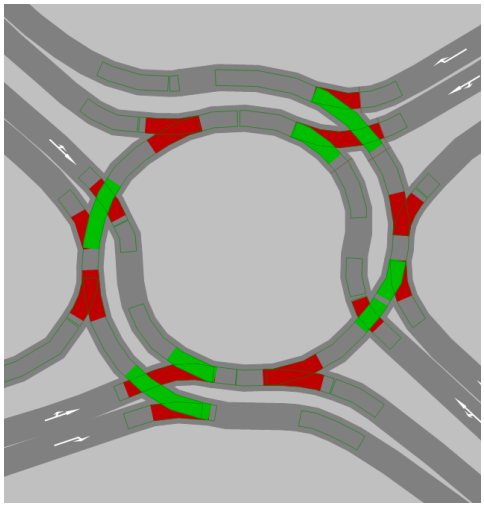


Figure 2.1: Overview of conflict areas on a turbo-roundabout. The green colour indicates which lanes have priority over the red lanes.

Table 2.3: Different values of critical gap t_c [s] of several research papers

Lane	Guerrieri	Brilon	Li	Fortuijn
Minor-left	4.41	4.0 (in), 4.5 (out)	4.3	3.94
Major-left	5.48	4.5	/	3.71
Minor-right	4.03	4.5	/	3.65
Major-right	4.18	4.5	/	3.41

Table 2.4: Different values of follow-up times t_f [s] of several research papers

Lane	Guerrieri	Brilon	Li	Fortuijn
Minor-left	2.53	2.6	3.1	2.25
Major-left	2.56	2.5	/	2.68
Minor-right	2.71	2.7	/	2.82
Major-right	2.6	2.5	/	3.08

2.3.2. Car-following behaviour

VISSIM has two options for car-following models: the Wiedemann 74 car-following model (W74) and Wiedemann 99 car-following model (W99). W74 is especially useful in urban environments, whereas W99 is useful for modelling freeway traffic without merging movements (in Transportation Engineering, n.d.). The car-following behaviour of human-driven vehicles can be described more adequately with the W74 model than with the W99 model (Mohebifard and Hajbabaie, 2020). On the other hand, W99 car-following models are better used to define the following behaviour of autonomous vehicles. This paper is the only one describing the used parameters for the Wiedemann model and uses the standard defined parameters in VISSIM.

Fortuijn and Salomons do not propose parameters for fine-tuning the Wiedemann 74 model but state that the distance between vehicles on the turbo-roundabout can be as close as three meters at driving speeds of approximately 36 km/h. (Fortuijn and Salomons, 2020).

2.3.3. Velocity of vehicles

As Section 2.2.1 mentions, the reduced speed areas are also important for turbo-roundabout performance. Li et al. recommend applying reduced speed areas on the access branches to the roundabout to make approaching vehicles slow down before entering the roundabout (Li et al., 2013).

Fortuijn does not touch upon the setting of reduced speed areas but on the general driving speeds on a roundabout (Fortuijn, 2018). He calculated the driving speeds on roundabouts in the Netherlands. This country applies a negative cant of 2% towards the outside for water runoff. In the paper, he presents a formula that calculates the driving velocity given a certain curve radius. The formula reads:

$$v = 9.2282 * R^{0.4185} \tag{2.1}$$

Figure 2.2 depicts the found relation between both variables. It shows that at low velocities, the impact of an increasing radius is greater than it is for higher values. Once the curve radii on the turbo-roundabout are known, the found relation determines the driving speeds in the reduced speed areas. However, it should be noted that this formula entails the fastest possible speed and therefore, the speeds in the simulation model will be lower.

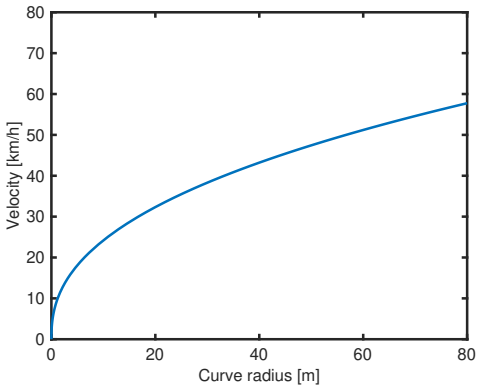


Figure 2.2: Graph depicting the relation between the curve radius and driving speed

2.4. Calibration of microscopic traffic simulation models

One of the most essential parts of this thesis is calibrating the microscopic traffic simulation model. This section describes the literature relevant to the calibration process.

A first paper describes a calibration and validation process to improve the simulation results of a VISSIM model of a regular, single-lane roundabout (Gallelli et al., 2017). They state that, for microscopic simulation models, sensitivity analysis and a trial-and-error method are helpful to calibrate the model automatically, but these procedures are very time-consuming. Instead, they apply a two-stage calibration/validation procedure. In the first stage, they insert the following macroscopic traffic flow variables in VISSIM: observed traffic volumes and speed distributions. In the second step, they use the W99 car-following model to enhance the correlation between field-measured- and simulated queue lengths. Their objective is to find the configuration of VISSIM that ensures a good level of correlation between the two. They conducted their research for four different scenarios, calculating the RMNSE for each scenario to measure the model performance compared to reality. In the RMNSE, model and reality queue lengths are compared. Using the standard W99 car-following parameters of VISSIM, they obtained an RMNSE of 1.29, which they deemed too high.

By finding the optimal configuration of the W99 car-following model parameters, they obtained an RMNSE of 0.59, which is a considerable improvement and indicates an acceptable level of transferability. Through an ANOVA test, they determined which of the W99 parameters had a statistically significant impact on the queue lengths in the roundabout. Furthermore, they stated that the W99 model with its standard parameters returns unrealistic driving behaviours for HDVs.
The authors also wrote another paper, comparing simulated and observed queue lengths (Gallelli et al., 2016). They considered the O/D matrix, the speed distributions for approaching vehicles, reduced speed areas, circulatory- and exiting speed, minimum gap and headways for the conflict areas, and W74 driver behaviour. Instead of using the RMNSE, they calculated the percentage deviation on each entry.

Another paper that uses queue lengths to calibrate a microsimulation model also uses the travel time as a measure for calibration (Otkovic et al., 2013). They use a neural network approach to calibrate and make claims about the validation procedure's strength by comparing their calibrated and uncalibrated model to actual data.

The next paper uses a different variable to validate its model (Arroju et al., 2015). Instead of using the queue lengths, the weaving speeds are extracted from their VISSIM simulation and compared to real-life ones. The mean average percentage error (MAPE) calculates the difference between the two. Other papers use location, time, headway, and speed (Benekohal, n.d.), traffic flows at sensor locations, point-to-point travel times, and queue lengths (Toledo et al., 2003). This last paper used the RMNSE to compare the traffic flow (aggregated for 15 minutes) of the simulation results to actual data. It came within 5 to 17 % of the actual data. They stated that these values indicated that their simulated flows corresponded well to the measurements.

There are various ways to validate microscopic traffic simulation models. Most papers use the queue length as their validation subject, while fewer use vehicle speeds, headways, and traffic flow elements. The only papers that state how they calculate the difference between their simulation and actual data uses the RMNSE method(Gallelli et al., 2017) or MAPE (Arroju et al., 2015).

2.5. Behaviour of Automated Vehicles on Turbo-Roundabouts

Before elaborating on the literature describing the driving behaviour parameters for automated vehicles, the capabilities of VISSIM to simulate automated vehicles are explored.

2.5.1. Capabilities of VISSIM to simulate automated vehicles

An employee from the PTV Group, the company that developed VISSIM, wrote a guideline describing how to model automated vehicles in VISSIM (Sukennik, 2020). In this document, he writes about the possibilities of VISSIM when it comes to modelling automated vehicles. In VISSIM itself, the built-in driving behaviours can simulate automated vehicle behaviour. The parameters of these vehicle behaviours are based on research of the European Union's CoExist project. The CoExist project prepares the transition phase during which automated and conventional vehicles co-exist on cities' roads. It bridges the gap between AVs technology, transportation and infrastructure planning by strengthening the capacities of urban road authorities and cities to plan for the effective deployment of AVs (n.a., 2024).
There are three car-following models for automated vehicles, ranging from cautious to aggressive. According to the author, the cautious model adopts a safe behaviour where the vehicle keeps large gaps to other vehicles. The normal model behaves human-like but has the additional capacity to measure the distances and speeds of the surrounding vehicles with its sensors. Lastly, the all-knowing or aggressive model has a larger awareness and predictive capabilities, leading to smaller gaps in all situations.

According to Zeidler et al., the behaviour of automated vehicles communicating with the leading vehicle is reproduced well through these car-following models in VISSIM (Zeidler et al., 2019). VISSIM also offers the possibility to simulate platoons. Various parameters can be set for the platoons, such as the number of vehicles in a platoon, safety distances, and the safety distances between vehicles in the same platoon.

The conclusion drawn from these papers is that automated vehicles can be modelled in VISSIM through car-following behaviour and vehicle platooning. For a more advanced modelling approach, an external controller must be coupled to VISSIM.

2.5.2. Merging behaviour

The merging behaviour of automated vehicles is, just as HDVs, determined at the conflict areas. The critical gap and follow-up times are once again the most important parameters.

The first paper uses a bi-level coordinated merging system for CAVs at single-lane roundabouts (Bakibillah et al., 2021). The critical gap and follow-up times they use in this report are 4 and 2 seconds, respectively.
A second paper uses VISSIM to determine the impact of automated vehicles on the capacity of single-lane roundabouts (Boualam et al., 2022). As stated earlier, this paper uses the standard car-following models of VISSIM. It obtains follow-up times of 1.8 seconds and a critical gap of 2.4 seconds. These values are thus not manually set but rather the result of their simulation.

The last paper states the values for the critical gap and follow-up time used during their theoretical determination of turbo-roundabout capacity under fully automated conditions (Fortuijn and Salomons, 2020). They used various levels of automation through four different automatic vehicle guidance (AVG) variants, giving values of both variables for the four different lanes. These four levels of automation are labelled as *AVG-T1*, *AVG-T2*, *AVG-T3*, and *AVG-T4*. The *AVG-T1* or AVG simplified model is limited to reducing headways and gaps, the *AVG-T2* model introduces gap synchronisation, the *AVG-T3* model uses gap synchronisation and headway optimisation, and the *AVG-T4* model uses gap synchronisation, headway optimisation and course guidance. For the purpose of this thesis, the *AVG-T1* model is the model of interest.
Tables 2.5 and 2.6 depict an overview of the obtained critical gap- and follow-up times. While the literature on HDVs distinguishes between major and minor access branches and left and right entrance lanes, the literature describing the merging behaviour of automated vehicles does not make this distinction.

Table 2.5: Critical gaps t_c [s] for automated vehicles for the several research papers

Lane	Bakibillah	Boualam	Fortuijn			
			AVG-T1	AVG-T2	AVG-T3	AVG-T4
Critical gap	4.0	2.4	2.34 inner 2.79 outer	2.79 inner 2.79 outer	2.79 inner 2.79 outer	2.36 inner 2.36 outer

Table 2.6: Follow-up times t_f [s] of several research papers for automated vehicles

Lane	Bakibillah	Boualam	Fortuijn			
			AVG-T1	AVG-T2	AVG-T3	AVG-T4
Follow-up time	2.0	1.8	1.17	1.32	1.32	1.18

2.5.3. Car-following behaviour

A first paper investigates the effects of various penetration rates of automated vehicles on traffic operations at roundabouts, specifically the average delay (Mohebifard and Hajbabaie, 2020). They use the standard car-following behaviours for automated vehicles programmed in VISSIM to investigate traffic operations. Next, they apply an optimisation-based approach to optimise the vehicle trajectories on the roundabout through an external mathematical program.
Other papers also use the built-in car-following models of VISSIM (Boualam et al., 2022; Giuffrè et al., 2021; Sukennik, 2020). The latter paper also looks at mixed-traffic conditions. However, it changes the standard W99 car-following model parameters through a calibration process to mimic the presence of heavy vehicles.

The earlier described paper of Fortuijn and Salomons depicts the different values used in their analysis of turbo-roundabout capacity (Fortuijn and Salomons, 2020). The different automatic vehicle guidance variants provide values for the following distance and minimum headways for both circulating- and merging traffic. They are summarised in Table 2.6.

One thesis paper uses an entirely different approach (Dijkhuizen, 2020). Its goal was to assess the actual and perceived safety of HDVs and CAVs in traffic. During the experiment, she analysed the preferred time gap between a perpendicular crossing vehicle leaving a conflict area and the subject entering that area. Instead of looking at the possibilities of automated vehicles, she focused on the gap acceptance of humans in those automated vehicles. She concluded that a gap of 2 s would be considered safe by 99% of the population and suggested using a value of 1.5 s for AVs as a start, with an acceptance grade of 93%. Lastly, she mentioned that this time gap could decrease as people gain more trust in automated vehicle technologies.

3

Methodology

This chapter explains the research methodology. Choices are made based on the literature from Chapter 2 and expert knowledge. The research methodology is divided into parts, each representing a specific step in the research process. Section 3.1 describes the general preparations needed to start the research. First, it describes the software selection, followed by the turbo-roundabout selection, and lastly, the data requirements. Once this is established, Section 3.2 elaborates on developing the human-driven model and describes the calibration process. Finally, Section 3.3 describes the model under fully automated conditions, and Section 3.4 elaborates on the method for comparing both models.

3.1. General preparation

3.1.1. Software

As mentioned in the introduction, microscopic simulation is an adequate way to simulate HDVs and AVs on turbo-roundabouts. Due to the importance of interactions between individual vehicles, microscopic simulation is a more suitable approach than simulations on a network level. Numerous software packages are capable of simulating traffic on a turbo-roundabout. Examples are *Aisum Next* (Aisum, 2024), *OpenTrafficSim* (OpenTrafficSim, 2024), *SUMO* (Lopez et al., 2018), and *PTV VISSIM* (Group, 2021). The software used in this research is selected based on the writer's capabilities, adequate support, and the simulation programs used in the literature. Numerous research papers use VISSIM to simulate vehicle behaviour on (turbo-)roundabouts (Arroju et al., 2015; Boualam et al., 2022; Gallelli et al., 2017; Li et al., 2013; Mohebifard and Hajbabaie, 2021; Severino et al., 2021; Vaiana et al., 2013).
A license and extensive manual for the 2021 version of VISSIM is available through the TU Delft. VISSIM also has a detailed user manual explaining every software feature, making it relatively easy to get up to speed with the program. Combining this with the fact that numerous researchers also use VISSIM for their simulations, the research in this thesis also makes use of it. More specifically, the 2021 version of the software is used. *OpenTrafficSim*, a software package developed at the university, offers more freedom in setting up a simulation environment. However, it is much more challenging to learn than VISSIM. That is why, for this thesis, *OpenTrafficSim* is not used.

VISSIM only has a few aspects included in the software for modelling AVs. It can only simulate automated car-following models and the formation of platoons (Sukennik, 2020; Zeidler et al., 2019), while it cannot simulate connections between AVs. This means that the simulation outcomes of VISSIM can only be compared to the results of the AVG-T1 model of Fortuijn and Salomons.

However, a perfect comparison to the AVG-T1 model cannot be made as it uses communication between AVs and the models developed in this thesis do not. Because of the communication, the follow-up times of the simulation models need to be greater than the AVG-T1 model. Without communication, vehicle sensors must first perceive other vehicles before they can respond, causing an additional delay.

3.1.2. Turbo-roundabout selection

The first step in this research is the selection of a real turbo-roundabout. When choosing an appropriate turbo-roundabout as a subject of this thesis, data availability is the most important selection criterion. At the very least, this data should contain measurements for various traffic flow rates on the turbo-roundabout and the access branches, allowing an adequate model calibration. These traffic flow rates must be collected under congested conditions to allow making claims about the capacity of a turbo-roundabout. Traffic on the circulatory lanes of the turbo-roundabout does not need to be congested; only the lanes of the access branches should be. After all, the literature in Section 2.2 states that the capacity of a turbo-roundabout is measured as the maximal traffic flow entering the turbo-roundabout. Only when entry lanes are congested is it possible to accurately determine the maximal capacity of the turbo-roundabout.
It is also possible to collect data for this specific research. However, due to the limited time frame of this master thesis and the high difficulty of conducting measurements under congested conditions, a roundabout with readily available data is chosen.

Furthermore, it benefits the research results if no external factors influence the traffic on the turbo-roundabout. Cyclist and pedestrian crossings on the access branches close to the turbo-roundabout can influence traffic behaviour and, thus, the data. Since this thesis is interested in the capacity of a turbo-roundabout, the interactions with other road users lie outside the scope of the research. A traffic light close to the turbo-roundabout also influences the traffic on the roundabout by grouping traffic with substantial gaps separating two subsequent waves, which is undesirable.

The turbo-roundabout for this research is located in Nieuwerkerk aan den IJssel[1], a municipality on the outskirts of Rotterdam. For this roundabout, all of the abovementioned criteria apply. Traffic flow data is available, and no external factors such as cyclists, pedestrians, or traffic lights exist. Figure 3.1 provides an aerial overview of this turbo-roundabout. The *Schielandweg*, *Europalaan*, and on- and off-ramps of the freeway A20 make up the four branches of the roundabout. The *Schielandweg* is the major branch of the roundabout, while the other two are the minor access branches. There are shortcuts for traffic from the *Schielandweg* for right-turning traffic. Since vehicles using these shortcuts do not enter the turbo-roundabout, they are not modelled in VISSIM. However, these shortcuts do affect the traffic on the turbo-roundabout. Subsection 3.2.4 explains how this effect is appropriately incorporated into the model.

3.1.3. Traffic data collection

The data collected at the turbo-roundabout in Nieuwerkerk aan den IJssel contains traffic flow measurements for various combinations of entering and circulating traffic. The entering times of merging traffic, passing times of circulating traffic at the merging sections, and times of vehicles leaving the turbo-roundabout were measured using traffic cameras on light poles. Traffic measurements were taken over five minutes when the traffic on the left merging lane was always queuing. This implies that, for the left lane, the measurements represent the merging capacity.

[1] $51°58'45"N 4°37'11"E$

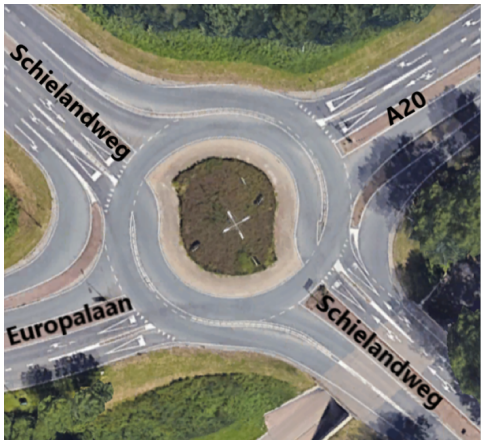


Figure 3.1: Aerial overview of a turbo-roundabout in Nieuwerkerk aan den IJssel, the Netherlands. The street names are depicted in black.

However, because data on the left and right access lanes was collected simultaneously, the data of the right lane is often not made in a congested state and, therefore, does not represent the correct merging capacity for the right lane. After all, the right lane has fewer conflicts than the left lane and reaches its maximum capacity less frequently. 41 traffic measurements were collected, of which only 19 featured congestion on both access lanes. This thesis uses all 41 measurements.[2] Only the data for the minor roundabout branch is used, as the minor branch is decisive for roundabout capacity.

The total flow of passenger cars, buses, and (semi)trucks was measured over (approximate) five-minute intervals, starting immediately after a vehicle entered the turbo-roundabout. After collecting the data, each vehicle type is assigned a passenger-car-equivalent (pce), representing the total number of passenger cars one vehicle from each category takes up in road space. By multiplying the pce by the number of observed vehicles from that category, one obtains the total amount of pce per five-minute interval. This ensures that (semi-)trucks and buses observed during the data collection convert into a certain number of passenger cars.
The collected data is then converted to hourly numbers for the research in this thesis. The resulting traffic volumes of the minor access branch are then used to calibrate the simulation model in VISSIM later in this report. Appendix A contains the Python code that was used for processing the data in the right format.

3.2. HDV model

After completing the preparatory steps, the model construction starts. The HDV model represents the current traffic conditions on the turbo-roundabout and is calibrated by comparing its simulation results to the Nieuwerkerk data. Once calibrated, it serves as a basis for the model simulating AVs on the turbo-roundabout. This section elaborates on the entire process of getting to the calibrated model. First, the turbo-roundabout is modelled, and then, an explanation is given on how traffic is modelled. Subsequently, the parameters are fine-tuned based on findings in the literature and, finally, the iterative calibration procedure is elaborated extensively.

[2] As it was only discovered after all research had been conducted that not all data points for the right lane were collected in congested conditions, this thesis used all 41 points as if they were congested.

3.2.1. Geometrical design of turbo-roundabout

The first step in developing the model is to copy the geometrical structure of the chosen turbo-roundabout into VISSIM. While the software package offers an *OpenStreetMap* overlay, the resolution of this map is insufficient to transfer the exact turbo-roundabout dimensions to the model. Therefore, a scaled aerial photograph of the turbo-roundabout is inserted as a background, allowing easy transferring of the exact dimensions.

The VISSIM software uses links and connectors to construct lanes. For each link, the number of lanes and their width has to be defined. As standard, lanes are 3.5 meters wide, approximately the same as the turbo-roundabout in Nieuwerkerk. The network in this report, therefore, uses the standard lane widths. The lane width has a marginal impact on the behaviour of vehicles, giving it a negligible impact on the performance of the turbo-roundabout. The crossing conflicts are the only network parts where the link width plays a role. Crossing traffic needs slightly more time to clear the conflict area for wider lanes than for narrower lanes.
Their length upstream of the turbo-roundabout is sufficient to allow congestion to form on the access branches. This ensures that enough vehicles are always waiting to enter the roundabout, meaning that merging traffic flow measurements are always done in congestion.

All access branches have two driving lanes, just like reality, with the only difference being that lane changes are not allowed. An exploratory simulation showed that modelling the network like reality generates inaccurate simulation outcomes. Simulation realism is mainly affected when a single lane precedes the two-lane segments. Vehicles choose the correct lane when lanes increase in front of an intersection/roundabout. In the model, however, vehicles do not show this behaviour and often choose the wrong lane, only to change lanes further downstream. This behaviour causes much unnecessary weaving and influences the simulation. Vehicles trying to merge into the other lane wait until they find a gap, often coming to a complete standstill. After a while, they are deleted from the network when the standstill time exceeds a certain threshold. This behaviour leads to an upstream blocking of the lane and causes substantial delays for the vehicles on that lane. As a result, the downstream merging section becomes uncongested, returning inaccurate capacity measurements. The measured entering flows thus portray a lower capacity of the merging sections. While this unwanted behaviour can be optimised in the settings, it is much easier to model the two access lanes separately. This ensures that vehicles approach the turbo-roundabout on the correct link and do not hinder other vehicles. By modelling the links this way, the merging capacity is marginally overestimated, as in real life, some vehicles change lanes after the lane increase.

Connectors connect links and allow vehicles to travel from one link to another. They thus model the merging movements onto the turbo-roundabout. Hence, traffic conflicts in the model always occur on connectors: merging conflicts, crossing conflicts, and weaving conflicts. According to the literature in Section 2.2.1, conflict areas created by the connectors are essential modelling aspects in VISSIM. Figure 3.2 shows the model of the turbo-roundabout in Nieuwerkerk aan den IJssel, where the coloured areas represent the conflict areas. Green denotes a lane with priority, while red denotes lanes that need to yield. The yellow areas depict the reduced speed areas, which are described next.

Once the lanes are modelled, the speed limits of the network are added. In VISSIM, cumulative probability distributions determine the speed input to reflect reality. Where vehicles enter the network, they have a speed of 80 km/h, which matches the speed limit on the access branches of the actual network.
On the turbo-roundabout, reduced-speed areas determine vehicle speeds. Applying this method allows one to influence vehicle speeds for specific sections in the network. When approaching a reduced-speed area, vehicles automatically slow down to obtain the speed specified in the area by the time they enter.

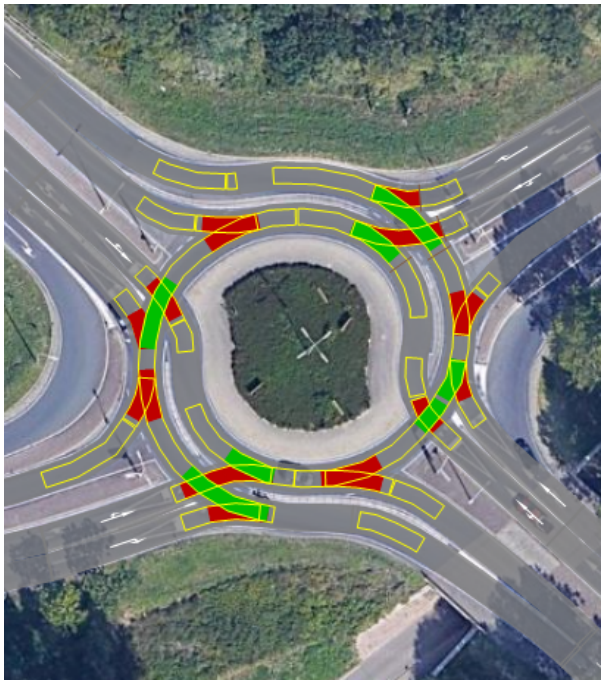


Figure 3.2: Turbo-roundabout in VISSIM. The red areas denote lanes that need to yield for vehicles in the green areas. The yellow areas indicate the reduced speed areas.

When a vehicle leaves the area, it accelerates until it reaches the desired speed of the link. Applying reduced speed areas in curves is necessary since vehicles in VISSIM do not automatically adjust their speed to the curve radius. Their speed is independent of the curve radius of the road. The speeds of the reduced speed areas can be defined by determining the curve radius. Once again, a cumulative probability distribution determines the speed of each vehicle. As mentioned in Section 2.2.1, the reduced speed areas affect the model's outcome. Therefore, much attention is paid to this element.

The curve radii on the turbo-roundabout itself vary between 11 and 25 meters. Using equation 2.1 from Subsection 2.3.3 gives speeds between 25 and 36 km/h. Since most curves have a radius of approximately 18 meters (31 km/h), the 30 km/h speed distribution of VISSIM is used for all reduced speed areas.

3.2.2. Human behaviour modelling

The next step in modelling the turbo-roundabout is the development of accurate human behaviour in the model. This subsection describes the most important parameters for achieving realistic vehicle behaviour. As already mentioned in the literature review, the three most important aspects of the VISSIM model that define human behaviour are the conflict areas, reduced speed areas, and the Wiedemann car-following models. The parameters are chosen according to the findings of the literature in Chapter 2. However, other factors must also be set to have a functioning model. This section describes these as well.

Merging behaviour

The conflict areas determine the merging behaviour, making them the most essential modelling aspect of the turbo-roundabout. The parameters defined for the conflict areas determine the merging capacity of the turbo-roundabout to a large extent. Conflict areas also define the priority rules for conflicting traffic streams.

For each conflict area, a status determines what lane has priority over the other one(s). The status of the conflict areas on the turbo-roundabout is set such that vehicles on the roundabout have priority over all merging traffic. The critical gap t_c is the most important parameter that describes the behaviour in the conflict areas and is easily inserted into VISSIM. The Excel file provided by Fortuijn, which contains the Nieuwerkerk data, also contains calibrated critical gap times. He calibrated using the actual turbo-roundabout data, making them especially useful for this thesis. Therefore, this thesis uses Fortuijn's calibrated critical gap times for the simulations.
The other important factor, the follow-up time t_f, cannot be inserted into the model as quickly since the conflict areas do not contain a parameter for it. Instead, the car-following models determine the follow-up times. Since the follow-up times vary per merging section and lane, unique car-following models must be made for each lane. Therefore, this thesis develops four different car-following models. These models are the same as the W74 car-following model but have different parameter values to change the follow-up times at the merging sections. The Wiedemman model uses a simple formula to specify the distances between vehicles in meters:

$$d = ax + (bx_{add} + bx_{mult} * z) * \sqrt{v} \tag{3.1}$$

Where:

d = distance between vehicles [m]

ax = Average standstill distance [m]. Default 2.0 metres

bx_{add} = Additive part of safety distance [m]. Default 2.0 metres

bx_{mult} = Multiplicative part of safety distance [m]. Default 3.0 metres

z = a random value in the range [0-1], normally distributed around 0.5

v = vehicle speed [m/s]

Only two equation parameters are changed to obtain the desired follow-up times at the different merging sections. Initially, the parameter ax is left untouched since it primarily describes the distance vehicles keep at a standstill. The multiplicative part of the safety distance bx_{mult} is also kept constant since it determines the spread of follow-up times between vehicles. The only parameter changed to obtain a desired follow-up time is the additive part of the safety distance bx_{add}.

Once changes in this parameter no longer influence the follow-up times, the standstill distance ax is changed. In VISSIM, the average follow-up time can be found by counting the number of vehicles under free flow conditions passing a certain point in the network over a certain period. The average follow-up time is obtained by dividing a period by the measured traffic flow. The exact values of the additive part of the safety distance and the standstill distance are iteratively obtained by comparing the desired follow-up time with the calculated follow-up time and tweaking the ax and bx_{add} parameters until both follow-up times match.
After determining the correct values for the parameter, the car-following behaviours are applied to the conflict area and the link preceding that conflict area. Applying the adjusted car-following model only to the conflict area leaves insufficient space for vehicles to slow down and adopt the desired follow-up time to other vehicles on the same link. The desired follow-up times are obtained by applying it on the upstream link.
Like the critical gap times, this thesis also uses Fortuijn's calibrated follow-up times. The exact values of the critical gap- and follow-up times can be found in Table 3.1. The values between brackets indicate the parameter values of bx_{add} that resulted in the desired follow-up times. No changes in the ax parameter were necessary.

Table 3.1: Values of critical gap and follow-up times for different access branches, as determined by Fortuijn.

Lane	Critical gap [s]	Follow-up time [s]
Minor-left	3.9	2.25 (2.87)
Major-left	3.7	2.68 (4.07)
Minor-right	3.7	2.82 (4.70)
Major-right	3.4	3.08 (5.23)

Car-following behaviour

The car-following behaviour is described on the link level of the network. Only the link behaviour type is relevant for this simulation. It describes the behaviour of vehicles on a link, and its relevant parameter is the car following model. The literature uses the W74 model to mimic HDV car-following behaviour on roundabouts(Li et al., 2013; Mohebifard and Hajbabaie, 2020); therefore this thesis also applies the W74 car-following model for HDVs.

3.2.3. Data extraction

Vehicle counts placed in specific locations of the network obtain the right data of the roundabout from VISSIM. While these vehicle counts can gather different types of information, such as speed, average distances, acceleration, vehicle lengths, and delays in the queue, only the amount of vehicles passing these points over a certain period is of interest. The vehicle counts in the model are programmed to merge vehicle counts for five-minute intervals. One obtains the hourly traffic counts by multiplying the obtained values by 12. These can then be compared to the actual traffic data, expressed in the same unit of measurement, and used to formulate conclusions.

Placing the vehicle counts at the correct locations is very important. Having them at the wrong locations can alter the simulation outcome and reduce accuracy. For the placement, the vehicle counts must only detect vehicles entering the conflict area, ensuring that all detected vehicles in the aggregated vehicle count have entered the conflict area. Figure 3.3 shows the locations of the data collection points for the minor access branch in yellow. Each lane has a separate vehicle count to enable the formation of capacity curves at a later stage in the research. A capacity curve depicts the merging capacity for various circulating traffic volumes, ranging from no circulating traffic to a volume where merging is no longer possible.

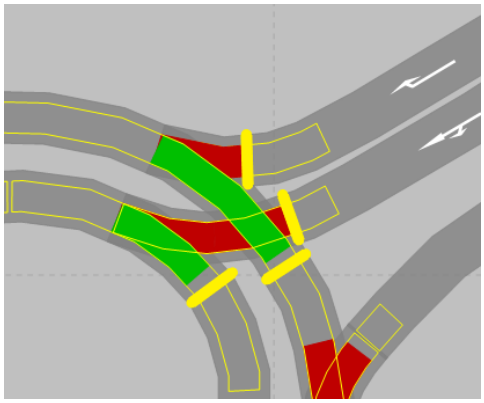


Figure 3.3: Location of data collection points for accurately representing simulation outcomes.

3.2.4. Network loading and model simulation

Now that all physical roundabout attributes and realistic driver behaviour are modelled, the network is loaded with traffic. OD (Origin-Destination) matrices load vehicles into the network, where each roundabout branch serves as an origin and a destination. In VISSIM, origins and destinations are modelled by nodes. The OD matrix gives the total amount of vehicles in each direction. However, it does not consider the vehicle composition, which needs modelling through the vehicle composition tab in VISSIM. As the Nieuwerkerk data was converted to passenger car equivalents, this research only simulates passenger cars.
Each OD matrix loads the network during a specified period. For this research, each matrix runs for ten minutes: the first five minutes serve as a warm-up period to achieve stable traffic conditions on the turbo-roundabout, and in the final five minutes, the model gathers data from the simulation. Between successive OD matrices, the model cools down for 10 minutes, ensuring that the previous OD matrix does not influence subsequent measurements.

By correctly choosing the different OD matrices, the vehicle counts obtain the capacity curves of the turbo-roundabout. Using an OD matrix where traffic originates from all access branches makes obtaining the complete capacity curve impossible. Because the circulatory lanes contain vehicles from different origins, the distance between vehicles is generally greater as the critical gap parameter becomes more decisive. The critical gap times are greater than the follow-up time and become more and more decisive for increasing traffic volumes. The circulating lanes reach their capacity before the merging traffic capacity decreases to zero, leading to incomplete capacity curves.
Therefore, OD matrices are constructed where the traffic originates from the previous branch of the one being tested. The simulation models in this thesis thus do not simulate the turbo-roundabout as a whole but only the merging section of the minor branch. The complete capacity curve can be obtained by gradually increasing the traffic originating from the previous branch. This process is carried out for one of the minor access branches as both have an equal layout and capacity curve in extension.

As Section 3.1.2 mentions, the turbo-roundabout in Nieuwerkerk aan den IJssel features shortcuts for the major traffic branch in both directions. Therefore, right-turning traffic from the major access branches does not enter the turbo-roundabout but instead uses these shortcuts. To achieve the same in the model, no right-turning traffic from the major branches is generated as this would induce unrealistic traffic behaviour on the turbo-roundabout. The effect on the access branch of leaving out these vehicles is negligible.

Simulations in VISSIM contain certain randomness: elements such as speed distributions and vehicle spawn times are all connected to a random seed. Multiple random seed simulations prevent this randomness from influencing the simulation results. In total, the model in this thesis simulates five different random seeds. Due to the long run times for each simulation and the limited time frame of this thesis, no more simulations are carried out. Given a longer time frame, carrying out additional random seed simulations is advised as the spread of the simulation data over five random seeds is low. Adding additional random seeds is unlikely to change the simulation results but would increase the reliability of the results.

3.2.5. Summary of initial simulation parameters

Table 3.2 provides an overview of all parameter values of the initial simulation. Other than these parameters, the standard VISSIM parameters are used.

Table 3.2: Overview of model parameters HDV model

Lane	Critical gap [s]	Follow-up time [s]	Car-following behavior
Minor-left	3.9	2.25 (2.87)	Wiedemann 74
Major-left	3.7	2.68 (4.07)	
Minor-right	3.7	2.82 (4.70)	
Major-right	3.4	3.08 (5.23)	

3.2.6. HDV model calibration

The calibration of the model is a crucial step in this thesis. Using a calibrated model to obtain the capacity curves of turbo-roundabouts under fully automated conditions makes it possible to draw more accurate conclusions about the achievable capacity increases. Calibrating the HDV model improves its credibility by demonstrating its ability to generate realistic results.

The calibration process is only performed for the minor access branch as it is decisive in determining (turbo-)roundabout capacity. It is also possible to calibrate the model using theoretical capacity models, but this thesis opts for calibration based on real data.
The first step in the calibration process is the construction of regression lines. The calibration process compares the simulation outcomes to the Nieuwerkerk data with the simulation data. Both datasets consist of data clouds of different sizes, which are difficult to compare to one another. A linear and nonlinear regression line is drawn for each set of data points. A choice between the linear or nonlinear line is made depending on which of the two has the smallest standard deviation. The linear regression line uses the equation $y = a * x + b$, and the nonlinear regression line uses the equation $y = a * \log(b * x + 1^{-10}) + c$. This function results in convex regression lines, which align with expectations. The equation $y = a * x^2 + b * x + c$ was used for the nonlinear regression line during earlier iterations but resulted in regression line shapes conflicting with traffic flow theory.
At lower circulating traffic volumes, the effect of the critical gap on the capacity is more substantial than that of the follow-up time. With an increasing circulating traffic volume, the importance of the follow-up time increases. At the same time, that of the critical gap decreases. Since the critical gap time is larger than the follow-up times, the capacity curve decreases faster for lower circulating traffic volumes than for higher volumes (hence the convex shape). Furthermore, literature shows that capacity curves are always convex (Fortuijn, 2009; Guerrieri et al., 2012; Guerrieri and Granà, 2009).
A Python program optimises the parameters a, b, and c. It calculates the average deviation between the regression lines and the data clouds. Appendix B contains a more detailed overview of this Python code.

For the Nieuwerkerk data, the regression lines range between circulating traffic volumes of 790 and 1810 pce/h, which is the complete range of the data. However, the regression lines of the simulation results have a much more extensive range. Like the Nieuwerkerk data, the Python code draws linear and nonlinear regression lines, and a choice between both is made the same way as the Nieuwerkerk data.
A vital comment about the regression curves of the data clouds is that approximating a data cloud with a regression line introduces inaccuracies when analysing results. The average deviation of the data clouds to its (non)linear approximation is a vital denominator for the introduced inaccuracy and, therefore, is included in the analysis to improve the accuracy of conclusions.

The last preparatory step, before the actual calibration takes place, is conducting a sensitivity analysis to investigate the effects of changing the most critical model parameters from the literature, the critical gap- and follow-up times, on traffic simulation to deepen the understanding of the model performance and facilitate the calibration process.

By doing several simulation runs while changing a different parameter each time, one can determine the model's sensitivity to these parameters. Then, the resulting percentage difference between the initial and new simulation results is calculated by taking the average percentage difference between the initial and updated simulation datasets. While the amount of measurements of both models is equal, their circulating traffic volume values are slightly different. This prevents one from directly calculating the average percentage deviation for each data point. Therefore, a linear interpolation is applied to one of the two datasets to obtain the same circulating traffic volumes for both simulations. Only then are the percentage differences for each point calculated by dividing the absolute difference between both data points by the value of the initial simulation. The resulting dataset of percentage differences contains some unrealistic values, which are likely the result of the linear interpolation process. Outliers, defined as values more than 1.5 times the interquartile range outside of the first or third quartile, are deleted from the dataset to prevent these from impacting the results. Only then is the average percentage deviation of all remaining points calculated. The calibration occurs after determining the model's sensitivity to the critical gap and follow-up times.

The RMNSE is used to compare the regression lines of the simulation with those of the Nieuwerkerk data, as literature often uses this same method when calculating the difference between two datasets. The RMNSE indicator is used for different types of data, and depending on the type of data and the importance of accuracy of that data, the target value can be different. Weather forecasts, for example, have a higher acceptable deviation than the projected national debt in the next couple of years. One can calculate it using the following formula:

$$RMNSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\frac{(m_n^{sim} - m_n^{obs})^2}{m_n^{obs}}} \tag{3.2}$$

where:

N = the total amount of data points

m_n^{sim} = the simulation data point

m_n^{obs} = the actual data point

This formula returns a single value, displaying the average deviation percentage of the simulation data from the Nieuwerkerk data. The RMNSE is calculated over the range of circulating traffic of the Nieuwerkerk data since this is the only range for which both the simulation and Nieuwerkerk data exist. Low values for the RMNSE represent a better fit, whereas higher values denote a worse fit. Since there is no literature discussing the acceptable lower bound for the accuracy of a microscopic traffic simulation model, the calibration process in this thesis does not use a specific target value. Instead, it performs seven iterations to decrease the RMNSE by as much as possible. The critical gap and follow-up times are fine-tuned for the left and right lanes separately in each iteration. For each iteration, decisions about parameter changes are made based on the results of the previous iteration. By doing it for both lanes individually instead of for the aggregated lane data, the performance of both lanes improves. Furthermore, one better understands how VISSIM performs at simulating realistic merging behaviour at both lanes.
Since the spread of the Nieuwerkerk data is too large to draw an accurate trend line, continuing the calibration iterations until perfection would not add value since the regression of the Nieuwerkerk data has a high standard deviation. This would thus create a false impression of accuracy. After calibrating the HDV model, construction starts on the AV models.

3.3. AV model

This section develops the setup of the AV model. To transition from an HDV model to an AV model, human behaviour is replaced by AV behaviour. No changes need to be made to the turbo-roundabout, only to the vehicles' behaviour.

As Subsection 2.5.1 mentioned, VISSIM cannot accurately model the complete behaviour of AVs. VISSIM only models the car-following- and merging behaviour of AVs correctly, but additional aspects should be modelled to unlock the full potential of AVs. As this is not part of the scope, this research does not model optimal AV behaviour and, therefore, does not contain all the steps to reach an optimal merging traffic capacity. This thesis does not go further than investigating the capacity increase of AVs on turbo-roundabouts, considering only the critical gap times and car-following models. Subsection 3.3.1 describes the modelling of AV merging behaviour, and Subsection 3.3.2 describes the car-following behaviour of AVs.

3.3.1. Merging behaviour

As mentioned earlier, the critical gap t_c and follow-up times t_f are the most important parameters describing the merging behaviour. The critical gap of AVs can be substantially lower than that for humans, as indicated by the summarising Table 2.5 in the literature review.
The critical gap values of Bakibillah et al. is different than those of Boualam et al. and Fortuijn and Salomons and, therefore, disregarded (Bakibillah, Kamal, Susilawati, and Tan, 2019; Boualam et al., 2022; Fortuijn and Salomons, 2020). The critical gap values of Fortuijn are used in this AV model since they better represent the goals of this report. These values do not differentiate between a minor or major access branch. While Boualam obtains the critical gap value using a coordinated merging system, the parameter values of the AVG-T1 model of Fortuijn and Salomons do not use an advanced system. This model only features vehicle communication but does not include gap synchronisation, which is included in the more advanced AVG-T2 through T4 models. These more advanced models are therefore not considered in this thesis. Table 3.3 depicts the critical gap values used in the AV model and compares them to the critical gap times of the HDV model.

Table 3.3: Overview of critical gap parameters for the AV model, compared to the HDV model

Lane	t_c AV model [s]	t_c HDV model [s]
Minor left	2.34	3.9
Major left	2.34	3.7
Minor right	2.79	3.7
Major right	2.79	3.4

3.3.2. Car following behaviour

In contrast to the HDV model, unique car-following models at the conflict areas are not utilised to set specific follow-up times in the AV models. Subsection 2.5.3 of the literature review describes that VISSIM's W99 car-following model performs adequately in simulating the car-following behaviour of AVs (Mohebifard and Hajbabaie, 2020). Therefore, this thesis uses the built-in car-following models of VISSIM. As a result, the network-wide car-following models determine the follow-up times and not the other way around.

VISSIM has three car-following behaviour models for AVs, named *AV cautious*, *AV normal*, and *AV aggressive*. The parameters of these models are based on the outcomes of the European CoExist project (Sukennik, 2020). As the literature mentions, the cautious model adopts safe behaviour by keeping larger distances from others.

AVs showing normal behaviour can measure distances and speeds of surrounding vehicles, thus improving their ability to predict sudden changes and leading to a more homogeneous traffic flow. The aggressive, or all-knowing model, has an increased awareness and predictive capabilities that lead to smaller gaps in all situations.
This thesis constructs three AV models, one for each following behaviour, enabling investigation of the effects of different car-following models on the capacity of the turbo-roundabout.

3.4. Calculation of capacity gain for AVs

The final step in the process is to compare the simulation outcomes of the HDV model with the three AV models. To answer the main research question of this report, *What capacity increase can be achieved by AVs on turbo-roundabouts through microscopic simulation, only considering automated vehicle car-following- and merging behaviour?*, the regression lines of the models are compared to one another.

The report of Fortuijn and Salomons calculates the capacity increase as the difference between the human data and the various automated vehicle models using a graph that plots the total load of the conflict area (sum of half of the circulating traffic and merging traffic of the left lane) over the percentage of circulating traffic of the total load of the conflict area (Fortuijn and Salomons, 2020). Figure 3.4 shows the capacity increase obtained by Fortuijn and Salomons.

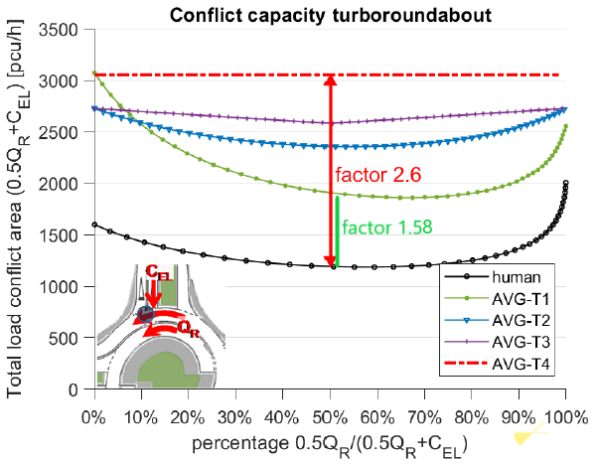


Figure 3.4: Conflict capacity increase for the various automated vehicle models (Fortuijn and Salomons, 2020). The black line represents their human model; the green line represents the AVG-T1 model, and blue, purple, and red represent the AVG-T2 through AVG-T4 models.

The effective capacity increase is calculated as the difference between the HDV and AV curves for the traffic condition where 50% of the total load on the conflict area is circulating traffic, and the other 50% is merging traffic. Fortuijn and Salomons use the same arbitrary traffic condition, which they deem representative of a real-life turbo-roundabout (Fortuijn and Salomons, 2020). One can disregard the factor of 2.6 in the figure since it measures the difference between an HDV model and an AV model with gap synchronisation, headway optimisation, and path guidance. Instead, the increase obtained by the AV models in this thesis is compared to the factor between the human and AVG-T1 models: 1.58. However, as this model features vehicle communication, the obtained capacity increase through simulation is not expected to be as large. This comparison is only possible for the left merging lane as Fortuijn and Salomons do not investigate the right merging lane. According to theory, the capacity increase for the left lane of the minor branch of a turbo-roundabout is always

decisive for merging capacity.
Nonlinear regression lines are made for the three AV models to allow an adequate comparison of simulation results. The calibrated HDV model uses the regression lines obtained during the calibration process. Then, these regression lines are plotted in the same configuration as Figure 3.4 to compare the data of the various simulation models and determine the differences in capacity.

4

Microscopic HDV model

This chapter describes the results of the model simulations with human-driven vehicles only. Section 4.1 shows the results of the first simulation run with the parameters defined in the methodology. Subsequently, Section 4.2 describes the preparatory steps for the calibration process. This section conducts the sensitivity analysis of the VISSIM model, which determines the effects of various parameter changes on model performance. Section 4.3 shows the actual calibration process, and Section 4.4 depicts the simulation results of the calibrated model. Finally, Section 4.5 contains the discussion of the calibration process and the final, calibrated model.

4.1. Initial simulation

The first simulation uses the parameter settings from the literature in Section 2.3. Table 4.1 provides an overview of these parameters. As mentioned in the methodology, other parameters and settings from VISSIM are the same as standard and thus are not included in the table. Table 4.2 provides an overview of the most important parameters, other than the vehicle speeds, and how they are modelled in VISSIM.

Table 4.1: Parameter overview of initial simulation

Category	Parameter	Location	Value
Vehicle speeds	Reduced speed areas	Roundabout+conflict areas	30 km/h
	Desired speed dist.	Access branches	80 km/h
Merging behaviour	Critical gap	Minor-left	3,9 s
		Major-left	3.7 s
		Minor-right	3.7 s
		Major-right	3.4 s
	Follow-up time	Minor-left	2.25 s
		Major-left	2.68 s
		Minor-right	2.82 s
		Major-right	3.08 s
Car-following behaviour	Car-following model	Entire network	W74

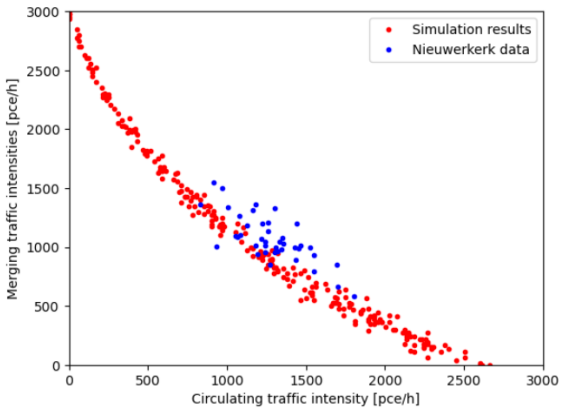
Table 4.2: Overview of the most important parameters for turbo-roundabout capacity and how these are modelled in VISSIM

Parameters	Parameter in VISSIM?	How it is modelled in VISSIM
Critical Gap t_c	Yes	MinGapBlockDef
Follow-up time t_f	No	Car-following models at the conflict areas
Car-following behaviour	Yes	Driving behaviour

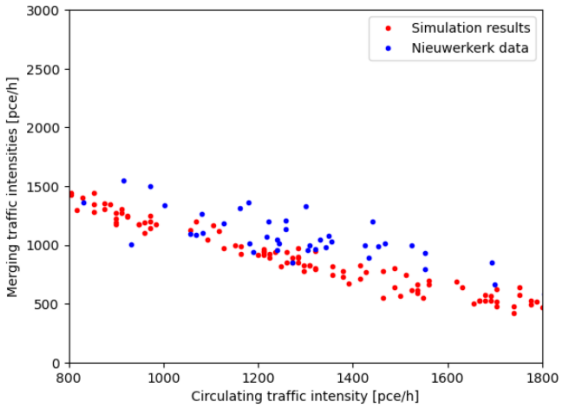
Capacity curves

The capacity curves of the first simulation model are compared to those of the Nieuwerkerk data to get an idea about the performance of the initial HDV simulation model. Figure 4.1 plots the capacity curve for the minor roundabout entrance. The red data dots represent the simulation results, and the blue data dots represent the Nieuwerkerk data. Figure 4.1a depicts the total capacity curve. The circulating traffic on the roundabout (inner and outer lanes combined) is plotted on the horizontal axis, and the entering traffic (left and right lanes combined) is depicted on the vertical axis.

The entering traffic measurements are the maximal volumes for their respective circulating traffic volumes, as the access branch is always congested. As the methodology mentions, the model simulates five random seeds. The figures thus contain the results of five different simulation runs combined into one extensive dataset. The figure shows that the entry capacity is approximately 3000 pce/h with no circulating traffic and decreases almost linearly to 0 at a circulating traffic volume of approximately 2600 pce/h. The Nieuwerkerk data has slightly higher merging traffic volumes than the simulation model for the same circulating traffic volumes, indicating that the base model does not yet accurately represent reality. However, for an initial simulation that still needs fine-tuning, the performance of the simulation model is satisfying. It is clear that the horizontal range of both datasets is substantially different. Therefore, Figure 4.1b shows a close-up of the range of circulating traffic volume in which the measurements of both datasets are present to obtain a clearer view of the difference between both. The close-up figure shows that the difference between both datasets is approximately constant over the entire range of circulating traffic flow when disregarding the data variances.



(a) Comparison of capacity curves of both lanes of the simulation model and Nieuwerkerk data



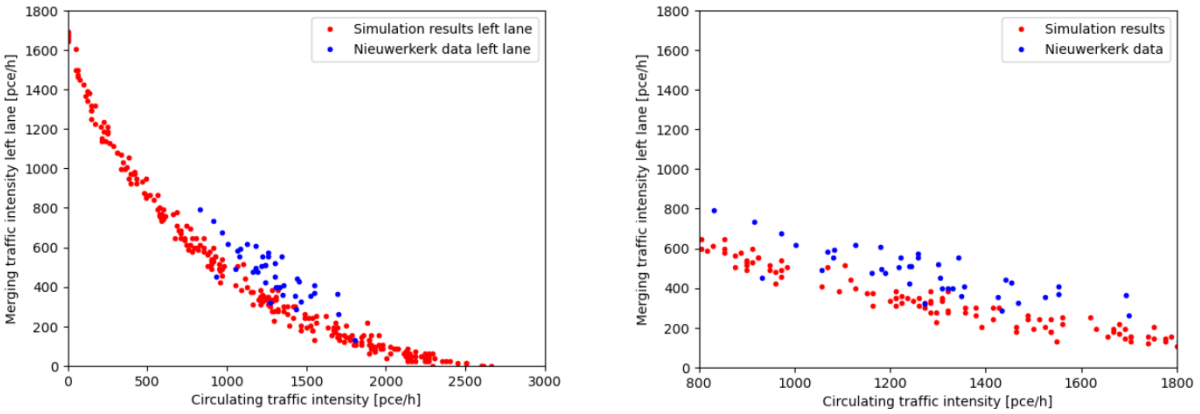
(b) Close-up of overlapping circulating traffic volumes

Figure 4.1: Comparison of the capacity curves of the simulation and Nieuwerkerk data for both entry lanes of the minor roundabout entrance. The red dots indicate the simulation data and the blue dots represent the Nieuwerkerk data.

The graphs contain the aggregated data for the left and right access lanes. Only using these graphs makes some vital information about the model outcome missing. Looking at the capacity curves of the individual lanes can provide more detailed information about the model's accuracy for each lane separately.

Therefore, Figures 4.2 and 4.3 compare the simulation data to the Nieuwerkerk data for the left and right lanes, respectively. For the left lane, merging traffic is plotted against the total circulating traffic. In contrast, only the circulating traffic on the outer roundabout lane is considered for the right merging lane. Both differ because the traffic on the right lane only conflicts with the outer roundabout lane, whereas the left lane conflicts with both circulatory lanes. The entry capacities of both lanes thus depend on different circulatory traffic flows.

When considering the graph of the left access lane in Figure 4.2a, one can notice that the merging capacity is approximately 1700 pce/h for zero circulatory traffic flow and decreases to 0 at a circulating flow of 2600 pce/h. The graph shows a nonlinear relation between the circulating and merging traffic flows. The merging capacity of the Nieuwerkerk data is substantially higher than the simulation data for the same circulating traffic volumes, indicating that the simulated traffic behaviour on the left lane does not represent reality. The close-up in Figure 4.2b shows that the difference between both datasets is once again approximately constant over the entire range of circulating traffic flow.

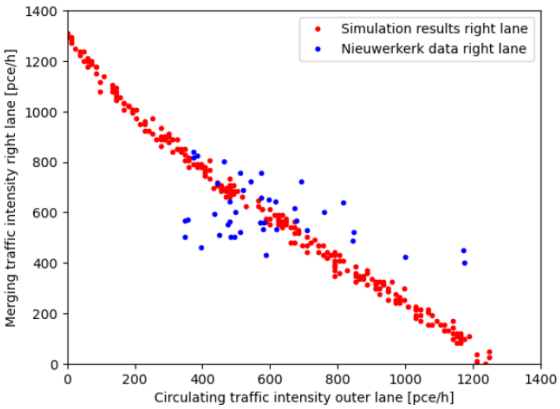


(a) Comparison of capacity curves for the left lane of the simulation model and Nieuwerkerk data

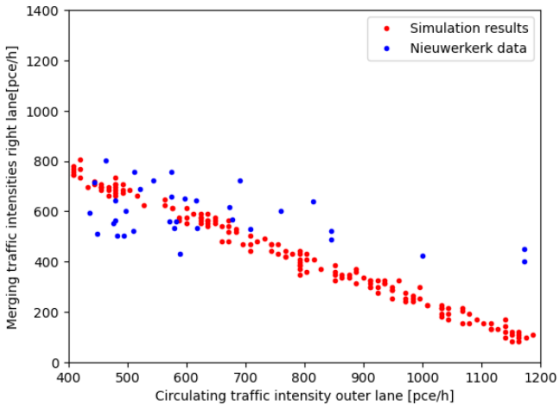
(b) Close-up of overlapping circulating traffic volumes

Figure 4.2: Comparison of the capacity curves of the simulation and Nieuwerkerk data for the left entry lanes of the minor roundabout entrance. Once again, the red dots indicate the simulation data, and the blue dots represent the Nieuwerkerk data.

Figure 4.3a shows the capacity curve for the right access lane. The most obvious difference between the left and right lanes is the type of relation between the circulating and merging traffic. The right lane shows an almost linear relation between the two, whereas the left lane shows a nonlinear relation. In conditions without circulating traffic, the maximal merging traffic flow is approximately the same as the left lane. Figure 4.3b clearly shows the relation between the Nieuwerkerk data and the simulation data. For circulating traffic volumes of approximately 500 pce/h, the merging traffic volumes are the same. However, as the circulating traffic increases in volume, the difference between the Nieuwerkerk and simulation data increases. Here, the simulation data underestimates the entry capacity. While the data for the left lane had a similar slope as the Nieuwerkerk data, the slope was different for the right lane. One can conclude that the right lane does not accurately model realistic traffic behaviour. Tweaking the model parameters is necessary to improve the performance of the model.



(a) Comparison of capacity curves for the right lane of the simulation model and Nieuwerkerk data. Once again, the red dots indicate the simulation data, and the blue dots represent the Nieuwerkerk data.



(b) Close-up of overlapping circulating traffic volumes

Figure 4.3: Comparison of the capacity curves of the simulation and Nieuwerkerk data for the right entry lanes of the minor roundabout entrance

4.2. Preparatory calibration steps

As described in Subsection 3.2.6 of the methodology, preparatory steps are carried out before the calibration. The first step is determining the regression curves that best represent the data, to allow the comparison of multiple datasets. Subsection 4.2.1 constructs these regression curves. Subsequently, Subsection 4.2.2 conducts the sensitivity analysis of the model performance for the critical gap and follow-up times, the most crucial simulation parameters.

4.2.1. Regression of data clouds

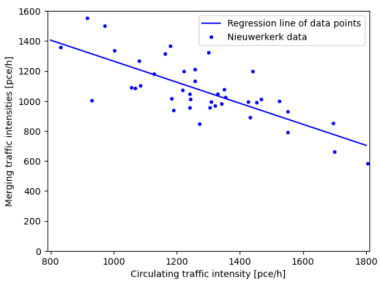
The regression of data clouds is done for both data types and various lane configurations: both entry lanes combined, only the left lane, and only the right entry lane. Carrying out the regression procedure for both separate lanes provides more information than using only the aggregate data. For each regression fit, the choice between a linear and nonlinear regression curve is made based on the standard deviation of the regression line to the data and the theoretically expected trend line.

Nieuwerkerk data

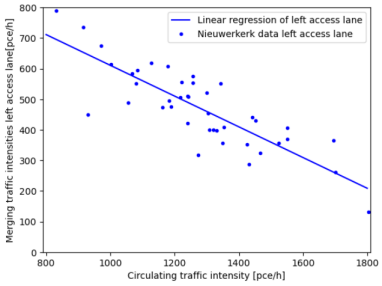
The Nieuwerkerk data is spread, making it more precarious to draw regression lines through the data clouds. This spread is an inherent property of capacity measurements at turbo-roundabouts, as measurements can only be made over short time frames. For the Nieuwerkerk data, the measurement period was five minutes, whereas traffic capacity is expressed in hours. To get hourly data, the Nieuwerkerk data is multiplied by 12, enlarging the differences between individual measurements. Small differences between the five-minute measurements now appear far more substantial. Unfortunately, collecting data over an hour is impossible, as traffic on the access branches would have to be congested during this entire period. Having a dataset with very little variability between individual data points is thus highly unlikely.

Figure 4.4 shows the linear regression results of the aggregated lanes, the left lane and the right lane, in that particular order. Just like previously, the circulating traffic is defined as the sum of the inner and outer lanes for traffic on the left lane and defined as the outer lane volumes for the right lane. Table 4.3 gives the standard deviation for each regression line.

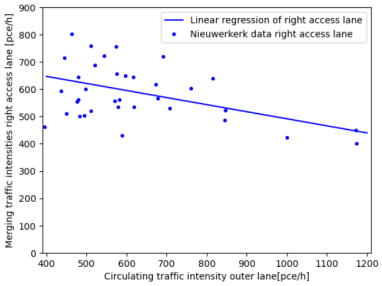
Next, the nonlinear regression is applied to the same data clouds. Once again, the regression is done for the same lane configurations. The results are shown in Figure 4.5. Table 4.3 shows the standard deviations for all regression lines of both the linear and nonlinear regressions.
The best regression line is chosen for the three cases based on the standard deviations. The linear regression method best approximates the Nieuwerkerk data for the aggregated data and right lane individually [1]. For the left lane, the nonlinear regression has a lower standard deviation than the linear regression and is the regression method of choice. The bold standard deviations in Table 4.3 indicate the regression method of choice for the various lane configurations.



(a) Linear regression of the Nieuwerkerk data for the left and right lanes combined

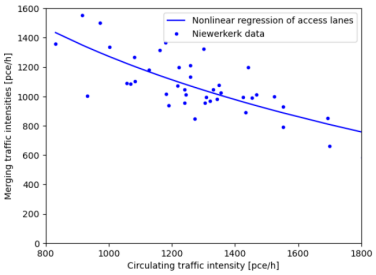


(b) Linear regression of the Nieuwerkerk data for the left lane

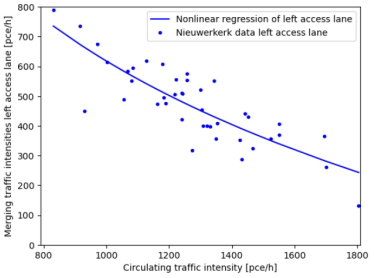


(c) Linear regression of the Nieuwerkerk data for the right lane

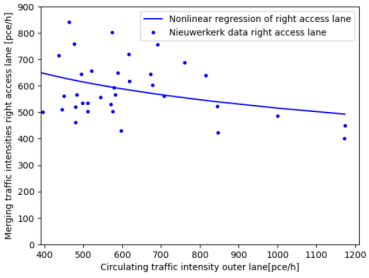
Figure 4.4: Linear regression of various lane configurations of the Nieuwerkerk data



(a) Noninear regression of the Nieuwerkerk data for the left and right lanes combined



(b) nonlinear regression of the Nieuwerkerk data for the left lane



(c) Nonlinear regression of the Nieuwerkerk data for the right lane

Figure 4.5: Nonlinear regression of various lane configurations of the Nieuwerkerk data

Table 4.3: Standard deviations of regression lines Nieuwerkerk data. The bold value indicates the regression line of choice for each lane configuration

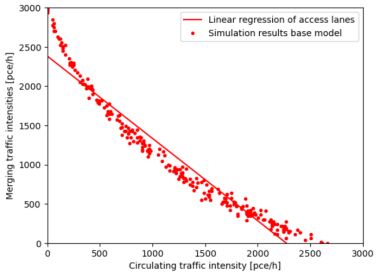
Regression line	Standard deviation [pce]		
	Aggregated	Left lane	Right lane
Linear regression	**107.68**	57.91	**86.73**
Nonlinear regression	109.13	**56.97**	89.52

Simulation data

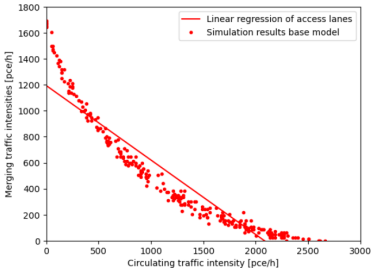
The same process is repeated for the data clouds representing the simulation outcome. Figure 4.6 depicts the linear regression lines, and Figure 4.7 depicts the nonlinear regression curves for all three lane configurations. The regression region is larger for the simulation data than for the Nieuwerkerk data, as the range of circulating traffic obtained from the model is larger.

[1] It should be noted that the right lane data points also include measurements not made under congested conditions. Therefore, the validity of the obtained regression lines is questionable.

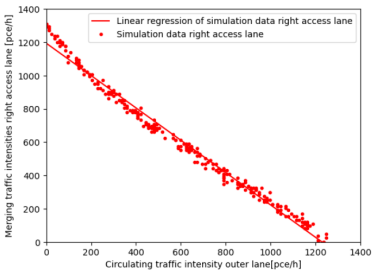
Table 4.3 shows the standard deviations of the various lane configurations, and once again, the standard deviations in bold indicate the chosen regression method for each configuration. Table 4.4 shows substantial differences between the standard deviations of the various regression lines. The standard deviations for the aggregated data regression lines are larger than those for the individual lanes. This aligns with expectations since the merging traffic volumes for the aggregate lane data are substantially higher than the individual lanes. For all three lane configurations, the standard deviations of the nonlinear regressions are smaller than the linear standard deviations. The differences in standard deviations for the aggregate and left lane data are particularly large. Looking at Figures 4.7a, 4.7b, and 4.7c, one notices that the nonlinear regression lines match the data substantially better than the linear ones in Figures 4.6a, 4.6b, and 4.6c. Therefore, the three configurations of the simulation data are approximated through nonlinear regression.



(a) Linear regression of the simulation data for the left and right lanes combined

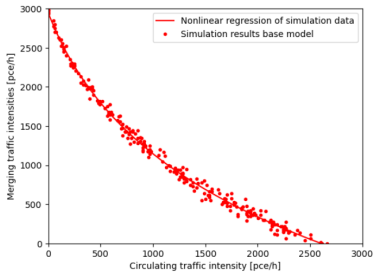


(b) Linear regression of the simulation data for the left lane

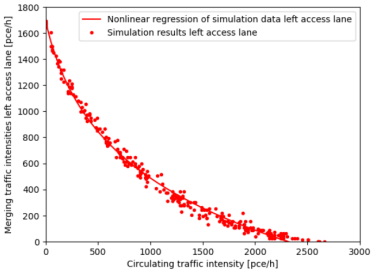


(c) Linear regression of the simulation data for the right lane

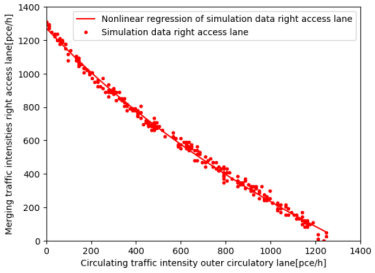
Figure 4.6: Linear regression of various lane configurations of the simulation data



(a) Nonlinear regression of the simulation data for the left and right lanes combined



(b) Nonlinear regression of the simulation data for the left lane



(c) Nonlinear regression of the simulation data for right left lane

Figure 4.7: Nonlinear regression of various lane configurations of the simulation data

Table 4.4: Standard deviations of regression lines simulation data. The bold values indicates the regression line of choice for each lane configuration

Regression line	Standard deviation [pce]		
	Aggregated	Left lane	Right lane
Linear regression	152.70	120.58	31.37
Nonlinear regression	**41.44**	**32.09**	**20.31**

4.2.2. Sensitivity analysis

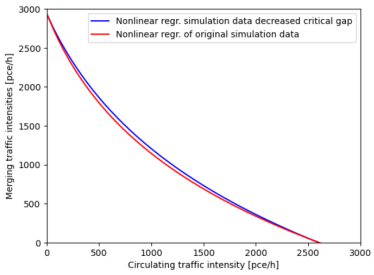
Based on the findings of Section 2.2.1 of the literature review, the conflict areas, reduced-speed areas, and car-following behaviour are the most influential model parameters that determine the capacity. The conflict areas have the largest influence on roundabout capacity and are used to calibrate the model (Boualam et al., 2022).

In this thesis, the calibration process does not change the parameters of the reduced-speed areas and car-following behaviour. The critical gap and follow-up times are the most influential parameters for the conflict areas. This section investigates the impact of a 10% decrease on the performance of the simulation model and the merging traffic flow for both parameters. One obtains the percentage difference between both models by comparing the base simulation to both models with a reduced parameter.

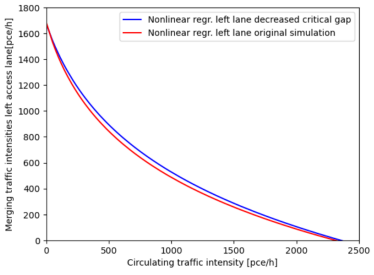
10% decrease in critical gap time

The critical gap time decreases by 10% for all conflict areas to investigate the differences in the simulation. Smaller critical gap times should lead to higher merging traffic volumes for equal circulating traffic volumes since more gaps are accepted. The capacity curve should thus display higher merging capacities than the initial simulation. At low circulating traffic volumes, the critical gap has a negligible impact on the merging traffic volume as the follow-up time is decisive.

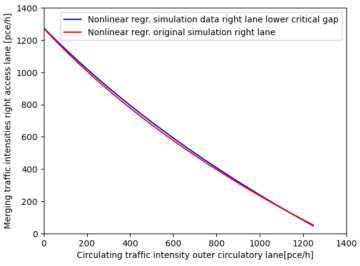
Due to the low circulating traffic volume, the gaps between vehicles on the turbo-roundabout are larger than the critical gap. As the circulating traffic volume increases, the gaps between vehicles on the turbo-roundabout decrease and get closer to the critical gap time. The critical gap becomes a more dominant parameter than the follow-up time for the merging capacity. As the gap times between vehicles on the turbo-roundabout become smaller than the reduced critical gap time, the impact of a reduction in critical gap diminishes. Therefore, a decrease in the critical gap should be most noticeable at higher circulating traffic volumes, where the increase in capacity should be more substantial than at lower circulating traffic volumes. As the circulating traffic volume leads to gap times on the turbo-roundabout smaller than the critical gap, the difference between both regression curves is expected to be minimal.
Figure 4.8 depicts the resulting nonlinear regressions of the capacity curves, and Table 4.5 shows the decreased critical gap parameters. Table 4.6 contains the percentage differences between the original simulation data regression and the decreased critical gap model regression. These percentage differences are calculated as the average difference between both regression models, where the difference is measured for every increase in circulating traffic volume of 25 pce/h. In general, this leads to more than 100 data points per calculation of average percentage change.



(a) Nonlinear regression lines for the aggregated lanes



(b) Nonlinear regression lines for the left lane



(c) Nonlinear regression lines for the right lane

Figure 4.8: Comparison between the nonlinear regression curves of the initial and decreased critical gap simulations. The red lines represent the regressions of the initial simulation model and the blue lines represent the regression lines of the decreased critical gap model.

Table 4.5: Decreased critical gap values

Lane	Value
Minor-left	3.5 s
Minor-right	3.3 s

Table 4.6: Difference between initial simulation and decreased critical gap simulation. Positive values indicate an increase in merging traffic capacity.

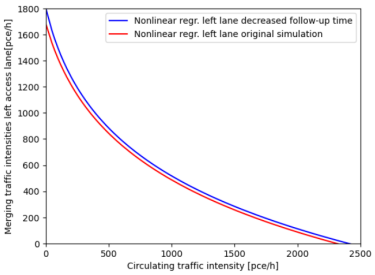
	Left + Right lane	Left lane	Right lane
Difference [%]	4.83	8.26	1.73

One can notice subtle differences between the regression curves in Figures 4.8a, 4.8b, and 4.8c. A 10% reduction of critical gap in VISSIM leads to an average increase in merging traffic capacity of 4.83%. For the left lane, the merging traffic capacity increases the most by 8.26%, while the increase for the right lane of 1.73% is the lowest. Table 4.6 summarises the capacity increase percentages. As expected, the difference in merging capacity is negligible at low circulating traffic volumes and increases with the circulating traffic volume. For the aggregate and left lane data, the biggest difference between the curves is noticeable at a circulating traffic volume of approximately 1000 pce/h. As the circulating traffic volume increases, the differences between the curves decrease. This shows that the critical gap does not impact the merging capacity in case of high circulating traffic intensities. Only at low intensities can substantial differences be noticed.

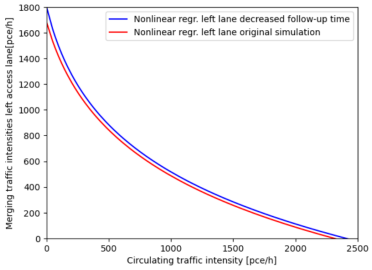
10% decrease in follow-up times

In the sensitivity analysis for the following parameter, the follow-up times at all merging sections decrease by 10%. As mentioned in Subsection 3.3.1, the additive part of the safety distance of the Wiedemann 74 car-following model is changed to obtain the desired values for the follow-up times. By only changing the additive part of the safety distance, one ensures that the distribution of the follow-up distances does not change but that the average follow-up time decreases.

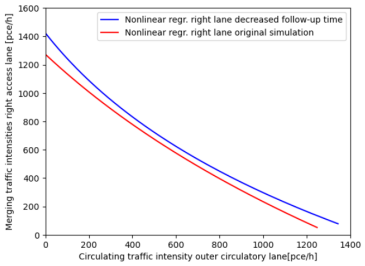
A decrease in follow-up time is expected to result in a higher capacity. The increase should be especially visible at lower volumes of circulating traffic since follow-up times impact merging capacity most at low circulating traffic volumes. Table 4.7 shows the new follow-up times. Figure 4.9 depicts the nonlinear regression curves to compare the difference between the initial simulation model and the model with decreased follow-up times. Table 4.8 contains the percentage differences between the regression lines of both models.



(a) Nonlinear regression lines for the aggregated lanes



(b) Nonlinear regression lines for the left lane



(c) Nonlinear regression lines for the right lane

Figure 4.9: Comparison between the nonlinear regression curves of the initial and decreased follow-up time simulations. The red lines represent the regressions of the initial simulation model and the blue lines represent the regression lines of the decreased follow-up time simulation.

Table 4.7: Decreased follow-up time values

Lane	Value
Minor-left	2.04 s
Minor-right	2.54 s

Table 4.8: Difference between initial simulation and decreased follow-up time simulation. A positive value denotes an increase in merging capacity.

	Left + Right lane	Left lane	Right lane
Difference [%]	10.39	8.36	10.42

The differences between the initial simulation and that with a reduced follow-up time are more noticeable than for the decreased critical gap model. Whereas the decreased critical gap model only showed clear improvements for circulating traffic volumes between 1000-1500 pce/h, the decreased follow-up time model shows improvements over the entire range of circulating traffic volumes. Table 4.8 contains the percentage increases for the various configurations. By reducing the follow-up times by 10%, the capacity of the aggregate lanes increases by 10.39%.

For the left and right lanes, the capacity increases by 8.36% and 10.42%, respectively. Whereas the circulating traffic volume barely increased in the decreased critical gap model, the increase for the right lane is most substantial in the decreased follow-up time model.

Takeaways
Based on the sensitivity analysis results, one can conclude that decreasing the critical gap times by 10% leads to an aggregated improvement of 4.83%. For the left lane, the increase in merging traffic volume is almost double at 8.26%. For the right lane, the capacity increases by only 1.73%. A reduction in follow-up times of 10% leads to a more substantial increase in merging traffic capacity. On aggregate, the merging capacity increases by 10.39%. The increase for the left lane is slightly lower at 8.36%, while it is slightly higher for the right lane at 10.42%. One can thus conclude that the impact of changing the follow-up times is larger than the impact of changing the critical gap times. Now, the calibration is carried out by fine-tuning both parameters over multiple iterations.

4.3. Calibration

This section describes the calibration process based on the information from the sensitivity analysis. The various iterations adjust the critical gap and follow-up times based on the obtained capacity curves from the previous iteration. The iterations in this section are elaborated in the same order as they were carried out. Since the range of circulating traffic volume of the Nieuwerkerk data is substantially smaller than those of the VISSIM simulations, this thesis calibrates for the bandwidth of the Nieuwerkerk data. By comparing the regression lines of the various iterations to the Nieuwerkerk data through the RMNSE, a better fit between the simulation and Nieuwerkerk data is obtained. The choice between linear and nonlinear regression lines is made based on the regression analysis in Section 4.2. This section only shows the calibration procedure for the left lane, even though it was conducted for both access lanes. However, the calibration of the right lane generated unrealistic results, so it was decided to leave it out of the report. The aggregate results of the calibration are also not depicted, as the right lane influenced those.

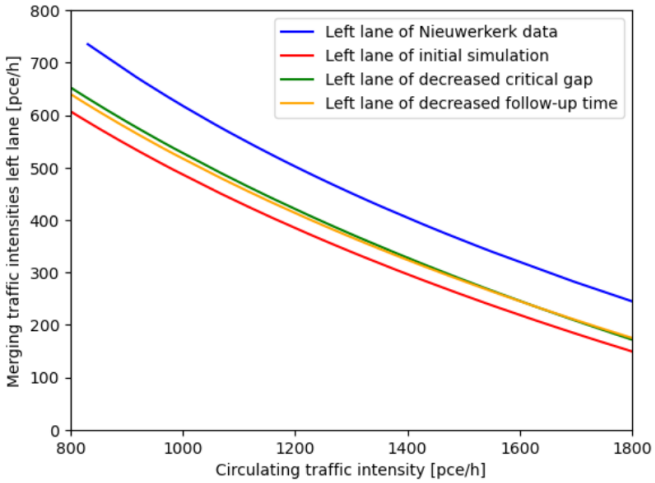


Figure 4.10: Comparison of the regression curves of the initial simulation, the sensitivity analysis simulations and the Nieuwerkerk data for the left lane.

Table 4.9: RMNSE values of initial regression and sensitivity analysis regressions for the left lane

Regression curve	RMNSE
Initial simulation	0.270
Decreased critical gap	0.194
Decreased follow-up time	0.202

Figure 4.10 depicts the nonlinear regression lines of the capacity curves of all models for the left access lane. The blue line represents the Nieuwerkerk data, the red line represents the initial simulation, and the green and orange lines are the decreased critical gap and follow-up time regression lines.

As the figure focuses on a smaller area than Figures 4.8 and 4.9, the differences between the initial and sensitivity analysis simulations are more obvious. The shape and slope of the three simulation models are similar to those of the Nieuwerkerk data. The differences between the simulations and the Nieuwerkerk data are the greatest at low circulating traffic volumes. Where the Nieuwerkerk model has a merging traffic capacity of approximately 740 pce/h, the three simulation models have merging traffic capacities of 600-650 pce/h. As the circulating traffic volume increases, the difference between both curves slightly decreases. The difference is the smallest at a circulating traffic volume of 1800 pce/h. Table 4.9 shows the RMNSEs for the three capacity curves and shows that the decreased critical gap model best matches the Nieuwerkerk data for the left lane.

4.3.1. Iteration 1

Now that the impact of changes in the critical gap and follow-up times on traffic performance in the circulating traffic range of the Nieuwerkerk data is known, the iterative calibration process starts. Table 4.9 and Figure 4.10 show that a 10% reduction in critical gap substantially improves the RMNSE for the left lane. It is expected that a 20% reduction in critical gap time over the initial simulation model decreases the RMNSE even more, which is attempted in this iteration.

Figure 4.11 shows the regression lines resulting from the first iteration and compares them to the initial simulation and the Nieuwerkerk data. Table 4.10 shows the RMNSE values for all three lane configurations of the first iteration and compares them to the RMNSE values of the initial simulation. By combining the results of the figures and the table, necessary parameter changes for the second iteration are determined. From the figure, one can read that a 20% decrease in critical gap leads to the same increase in merging traffic capacity as a 10% critical gap decrease. The RMNSE values of 0.194 and 0.199 confirm this.

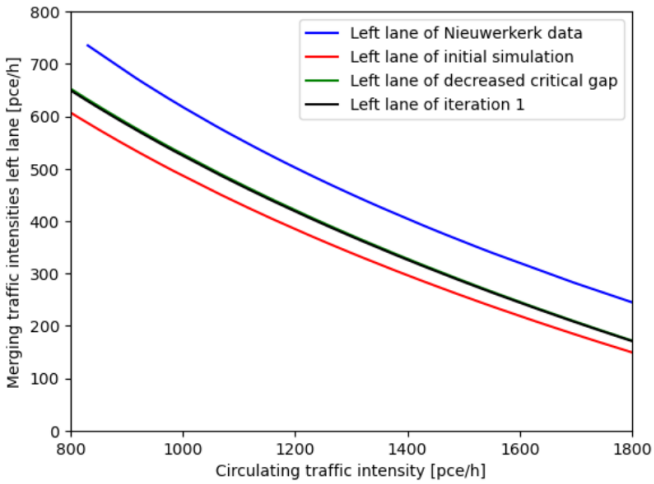


Figure 4.11: Comparison between the nonlinear regression curves of the initial model, the Nieuwerkerk data, and the first iteration of the simulation. The blue lines represent the regression lines for the Nieuwerkerk data, the red lines represent the regression lines of the initial simulation model, and the black line indicates the regression lines of the first iteration.

Table 4.10: RMNSE values of the initial simulation model and the first iteration

Regression curve	Left lane
Initial simulation	0.270
Iteration 1	0.199

Since this is an unexpected result, the reason behind this peculiarity is explored. A deeper dive into VISSIM shows that the effective critical gap of a model is determined by either the *MinGapBlockDef* parameter, which is set to 3.1s in this iteration and represents the time gap between the back of a vehicle on the turbo-roundabout and the front of a vehicle in the merging lane, or by the minimal distance between the back of a vehicle on the roundabout and the front of a merging vehicle.

This distance is calculated by multiplying the normal desired safety distance of the car-following model with a safety factor of 1.5. Of those two values, the decisive one is used. This safety factor appears decisive for critical gaps smaller than 3.4s, and the merging traffic capacity does not change with decreasing critical gaps.

Based on these findings, this safety factor is decreased to 0.2 in the next iteration to investigate its impact on the simulation. A value smaller than 1 allows vehicles to merge closely behind a circulating vehicle, which is realistic. Once on the roundabout, the distance between both vehicles increases to the desired following distance. Decreasing this safety factor even further would cause collisions between vehicles in the model, which is undesirable.

4.3.2. Iteration 2

The second iteration investigates the impact of decreasing the *SafDistFactDef*. The parameter is decreased from 1.5 to 0.2, which is expected to make the critical gap parameter *MinGapBlockDef* decisive for all subsequent iterations. Therefore, the difference between the initial simulation and the second iteration should become more obvious. Table 4.11 and Figure 4.12 show the comparison of the RMNSE of the initial simulation and the second iteration and the regression line of the second iteration.

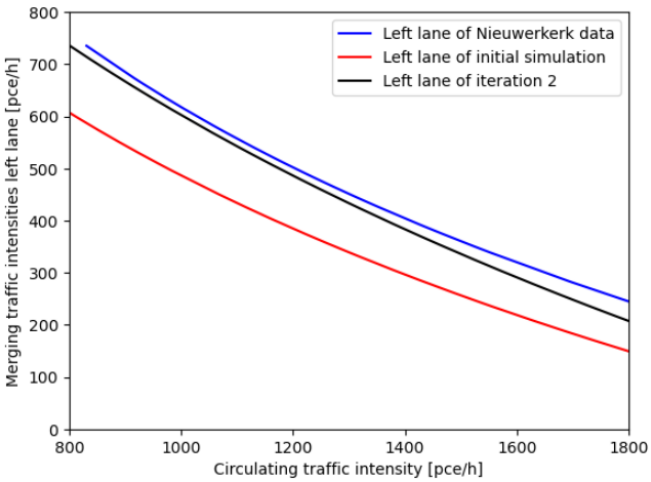


Figure 4.12: Comparison between the nonlinear regression curves of the initial model, the Nieuwerkerk data, and the second iteration of the simulation.

Table 4.11: RMNSE values of the initial simulation model and the second iteration

Regression curve	Left lane
Initial simulation	0.270
Iteration 2	0.062

In the figure, one notices that the reduction in safety factor leads to a substantial improvement in the model's performance for the left lane: the RMNSE decreased from 0.199 to 0.062. From the results of the right lane, a peculiarity was found in this iteration. The slope of the regression line changed substantially with the reduction of the *SafDistFactDef* parameter, requiring closer investigation.
Looking at the behaviour of vehicles in the simulation, it became obvious that the safety factor reduction leads to unrealistic behaviour. The merging behaviour of vehicles on the access branch can be described as more aggressive, obliging the vehicles on the circulatory lanes to brake to avoid a collision and increase the time gap to the standard safety distance. This behaviour violates the priority rules and, therefore, the resulting RMNSE improvement should be disregarded.
One can conclude that reducing the *SafDistFactDef* parameter to 0.2 introduces unrealistic behaviour in the network. Therefore, in the third iteration, the *SafDistFactDef* parameter is changed to 1. This value ensures that vehicles on the circulatory roadway do not slow down for merging vehicles and that the realism of traffic on the turbo-roundabout improves.

4.3.3. Iteration 3

In this iteration, no changes are made to the model except for adjusting the *SafDistFactDef* factor to 1. Table 4.12 and Figure 4.13 show the regression lines of the third iteration and compare the RMNSE of the initial simulation with the third iteration. As expected, the RMNSE of the left lane increased from 0.062 in the second iteration to 0.135 due to the increase in the safety distance factor. The figure shows that the shape of the capacity curve of the third iteration is very close to that of the simulation data but that the simulated merging traffic capacity is, on average, 13.5% too low.

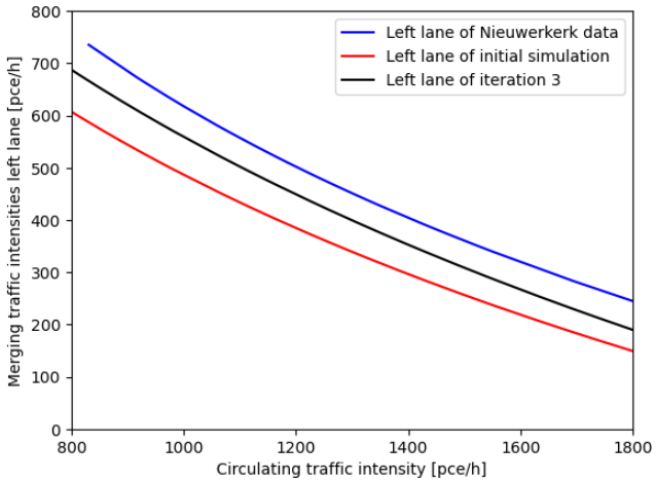


Figure 4.13: Comparison between the nonlinear regression curves of the initial model, the Nieuwerkerk data, and the third iteration of the simulation.

Table 4.12: RMNSE values of the initial simulation model and the third iteration

Regression curve	Left lane
Initial simulation	0.270
Iteration 3	0.135

4.3.4. Iteration 4

In the third iteration, the left lane's performance decreased due to the increasing safety factor. Since the merging traffic capacity was lower for all circulating traffic intensities, the follow-up time for the left lane decreased further by 10%, from 2,25s to 2,02s. This is expected to decrease the distance between the fourth iteration curve and the Nieuwerkerk data.

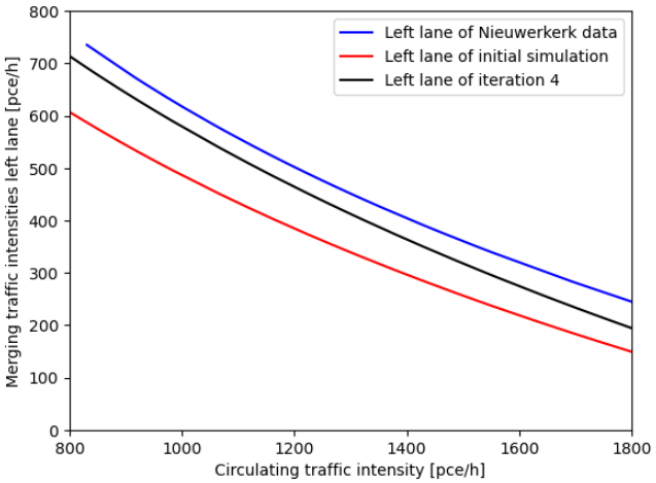


Figure 4.14: Comparison between the nonlinear regression curves of the initial model, the Nieuwerkerk data, and the fourth iteration of the simulation.

Table 4.13: RMNSE values of the initial simulation model and the fourth iteration

Regression curve	Left lane
Initial simulation	0.270
Iteration 4	0.110

Table 4.13 and Figure 4.14 show the regression lines of the fourth iteration and compare the RMNSE of the initial simulation with the fourth iteration. The figure shows that the regression line of the fourth iteration is closer to the Nieuwerkerk data than in Figure 4.14. The reduction of RMNSE from 0.135 to 0.110 confirms this observation. The biggest improvement can be noticed for low circulating traffic intensities.

4.3.5. Iteration 5

The fourth iteration showed slight improvements for the left lane compared to the third iteration. It still simulated a lower merging traffic capacity than the Nieuwerkerk data and, therefore, requires an additional upward shift. The critical gap is further reduced by 10%, from 3.12s to 2.8s. Iteration 1 showed that a decrease in the critical gap does not necessarily translate to an increase in merging traffic capacity due to the presence of the *SafDistFactDef* parameter. However, it is uncertain whether the lowest achievable critical gap was obtained in previous iterations. Therefore, an additional attempt to further decrease the critical gap is made.

Table 4.14 and Figure 4.15 show the regression curves of the fifth iteration and the Nieuwerkerk data, and the RMNSE values of both regression lines. The figure shows no clear difference compared to Subfigure 4.14, which is confirmed by the negligible difference in RMNSE (0.106 vs 0.110). One can conclude that a further reduction in critical gap does not influence the capacity curves.

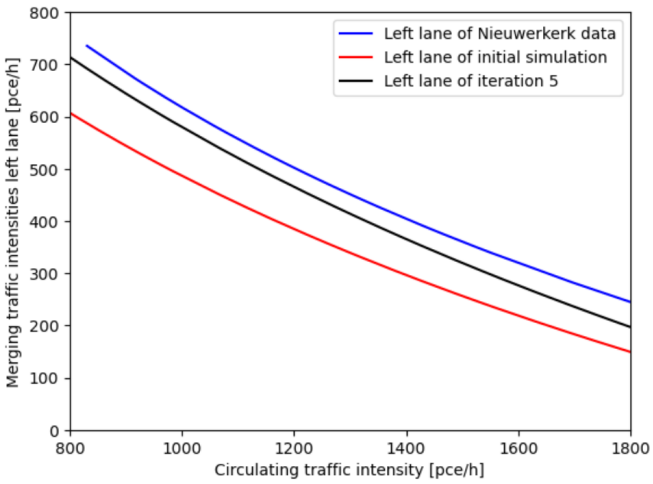


Figure 4.15: Comparison between the nonlinear regression curves of the initial model, the Nieuwerkerk data, and the fifth iteration of the simulation.

Table 4.14: RMNSE values of the initial simulation model and the fifth iteration

Regression curve	Left lane
Initial simulation	0.270
Iteration 5	0.106

4.3.6. Iteration 6

Since the critical gap is at a minimum and the merging traffic capacity is still lower than the Nieuwerkerk data, a further reduction in follow-up time is investigated. During the parameter fine-tuning, it was noticed that the follow-up time of 2.02s for the left lane could not be decreased any further through the bx_{add} parameter. To allow a further decrease in follow-up time, the average standstill distance ax between vehicles on the left lane is decreased from 2m to 1.5m. At low circulating traffic volumes, this is expected to result in a higher merging capacity since the effective follow-up time is further decreased. The effective follow-up time decreases because merging vehicles have to travel a shorter distance when departing from a standstill and can, therefore, merge closer to the previous merging vehicle. At higher circulating traffic volumes, the merging capacity increase is expected to be negligible due to the limited amount of vehicles being able to merge onto the turbo-roundabout.

Figure 4.16 and Table 4.15 show the regression curves of the sixth iteration and the Nieuwerkerk data, and the RMNSE values of both regression lines. The figure shows that the regression line of the sixth iteration better aligns with the Nieuwerkerk regression line, which is especially visible for lower circulating traffic intensities. The decrease in RMNSE from 0.106 to 0.085 confirms this observation.

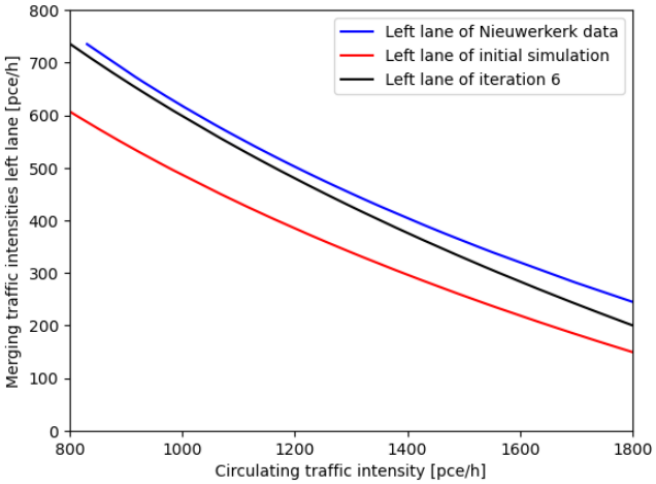


Figure 4.16: Comparison between the nonlinear regression curves of the initial model, the Nieuwerkerk data, and the sixth iteration of the simulation.

Table 4.15: RMNSE values of the initial simulation model and the sixth iteration

Regression curve	Left lane
Initial simulation	0.270
Iteration 6	0.085

4.3.7. Iteration 7

For the seventh iteration, the standstill distance of the left lane is decreased further from 1.5s to 1.0s, as this is expected to improve the fit between the Nieuwerkerk data and the simulation outcomes even more. Figure 4.17 and Table 4.16 show the regression curves of the seventh iteration and the Nieuwerkerk data, and the RMNSE values of both regression lines. In the figure, one can observe that the distance between the Nieuwerkerk data and the regression line of the seventh iteration decreased further. A reduction in RMNSE from 0.085 to 0.074 confirms this observation.

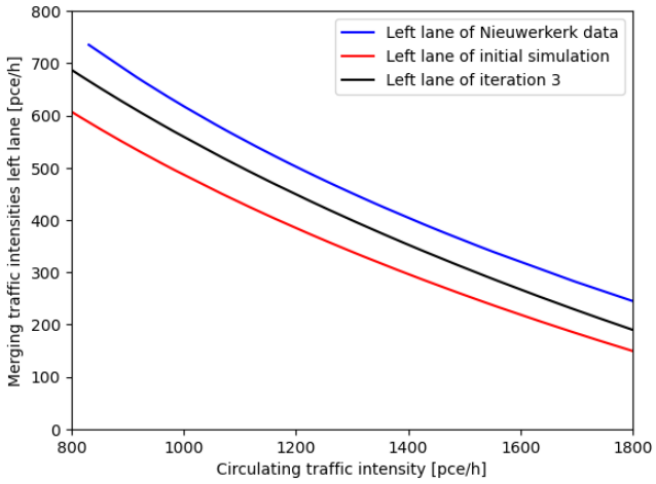


Figure 4.17: Comparison between the nonlinear regression curves of the initial model, the Nieuwerkerk data, and the seventh iteration of the simulation.

Table 4.16: RMNSE values of the initial simulation model and the seventh iteration

Regression curve	Left lane
Initial simulation	0.270
Iteration 7	0.074

The iteration process is terminated after seven iterations, as the maximal obtainable accuracy by only changing the critical gap and follow-up times is achieved. For the left lane, the merging traffic capacity at lower circulating traffic volume closely approaches the Nieuwerkerk data. At higher circulating traffic volumes, the distance between the Nieuwerkerk data and the simulation outcomes is higher. Unfortunately, for higher circulating traffic volumes, the distance between the capacity curves could not be decreased by fine-tuning the critical gap and follow-up times.
For the right lane, the iteration process resulted in highly inaccurate capacity curves, so the model could only be calibrated for the left lane.

The large standard deviations of the Nieuwerkerk data, as shown in Table 4.3, indicate that the regression lines of the Nieuwerkerk data are not a highly accurate representation of reality. Therefore, reaching an RMNSE closer to zero does not necessarily translate to a more realistic model. The seventh iteration is therefore used as the 'calibrated' model in the remainder of this thesis.

4.3.8. Wrap-up of the iteration process

Through seven iterations, various changes to the initial model have been made. For each iteration, a different parameter was changed for the left and right access lanes of the turbo-roundabout, based on the outcomes of the previous iteration. Table 4.17 contains an overview of the RMNSE values of all iterations and compares it to the initial simulation model. Furthermore, it also shows the critical gap and follow-up times for each iteration. For the left lane, a major improvement of 72.6% in RMNSE from 0.27 to 0.074 is obtained. Disregarding iteration 2, which resulted in unrealistic vehicle behaviour, the RMNSE of the final iteration is the lowest of all iterations. Combining a good fit between the regression lines of the seventh iteration and the initial simulation and a low RMNSE, one can conclude that the seventh iteration of the simulation model performs well at simulating traffic on the left merging lane of a minor turbo-roundabout branch.
For the right lane, the model performs insufficiently well. Section 4.5 explores possible reasons for this bad performance.

Table 4.17: RMNSE values, and critical gap and follow-up time parameters of the initial simulation model and all iterations

Regression curve	Left lane
Initial simulation	
Critical gap [s]	3.9
Follow-up time [s]	2.25
SafDistFactDef	1.5
RMNSE	0.270
Iteration 1	
Critical gap [s]	3.12
Follow-up time [s]	2.25
SafDistFactDef	1.5
RMNSE	0.199
Iteration 2	
Critical gap [s]	3.12
Follow-up time [s]	2.25
SafDistFactDef	0.2
RMNSE	0.062
Iteration 3	
Critical gap [s]	3.12
Follow-up time [s]	2.25
SafDistFactDef	1.0
RMNSE	0.135
Iteration 4	
Critical gap [s]	3.12
Follow-up time [s]	2.02
SafDistFactDef	1.0
RMNSE	0.110
Iteration 5	
Critical gap [s]	2.8
Follow-up time [s]	2.02
SafDistFactDef	1.0
RMNSE	0.106
Iteration 6	
Critical gap [s]	2.8
Follow-up time [s]	2.02
SafDistFactDef	1.0
ax [m]	1.5
RMNSE	0.085
Iteration 7	
Critical gap [s]	2.8
Follow-up time [s]	2.02
SafDistFactDef	1.0
ax [m]	1.0
RMNSE	0.074

4.4. Calibrated Model

Table 4.18 provides an overview of the final model parameters of the seventh iteration. The parameters of the right lane are not included, as the final iteration is inaccurate. However, the obtained capacity curves for the right lane are shown in subsequent subsections to provide the reader with a view of the poor performance. The specific car-following model parameters of Table 4.19 are applied to obtain the follow-up times for both minor merging lanes. Parameters not shown in the table are the standard values of the W74 model in VISSIM.

Table 4.18: Overview of parameters of the calibrated simulation model

Category	Parameter	Location	Value
Vehicle speeds	Reduced speed areas	Roundabout+conflict areas	30 km/h
	Desired speed dist.	Access branches	80 km/h
Merging behaviour	Critical gap	Minor-left	3,1 s
		Minor-right	4.1 s
	Follow-up time	Minor-left	2,05 s
		Minor-right	4.1 s
Car-following behaviour	Car-following model	Entire network	W74

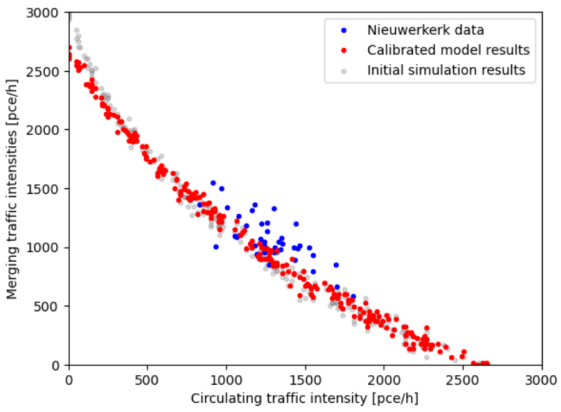
Table 4.19: Overview of the parameter values of the specific car-following models of the merging sections of the minor access branch

Lane	Standstill distance [m]	bx_add [-]
Minor left	1	2.30
Minor right	2	8.00

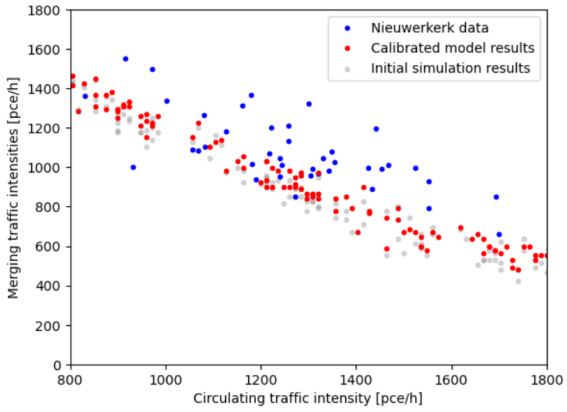
4.4.1. Capacity curves

The capacity curves of the calibrated simulation model are compared to those of the Nieuwerkerk data to get a sense of the improvement of the fit compared to the initial simulation model. Figure 4.18 plots the capacity curve of both merging traffic lanes for the minor roundabout entrance. The red dots represent the calibrated simulation results, the blue dots represent the Nieuwerkerk data, and the transparent grey dots represent the initial simulation data. Subfigure 4.18a shows the total capacity curve. One notices that the merging capacity of the calibrated model is approximately 2700 pce/h with no circulating traffic volume and decreases to 0 at a circulating traffic volume of approximately 2700 pce/h. Comparing this to the initial simulation results with a merging traffic capacity of 3000 pce/h, one can conclude that the merging capacity at low circulating traffic volumes is slightly lower. The differences between both simulation models decrease as the circulating traffic volume increases. From a circulating traffic volume of 1000 pce/h onwards, the merging capacities of both models are approximately equal. This aligns with the findings of the iteration process, which showed that the capacity curves at high circulating traffic volumes barely change with differing critical gaps and follow-up times.

Subfigure 4.18b provides a close-up view of the range of circulating traffic volume in which the measurements of both datasets overlap. It shows that the differences between the calibrated simulation model and the Nieuwerkerk data are smaller than those between the initial simulation and Nieuwerkerk data when disregarding the data variances in the different datasets.



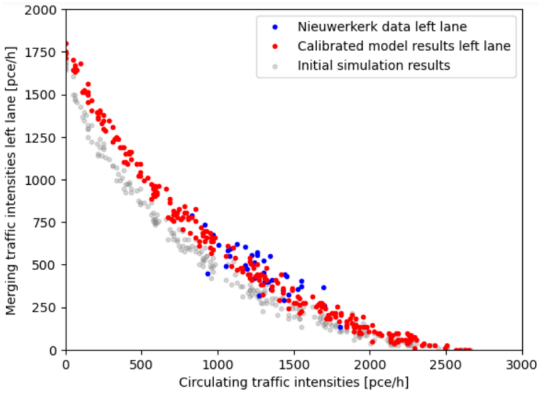
(a) Comparison of capacity curves of the calibrated model and Nieuwerkerk data, with initial simulation data as reference



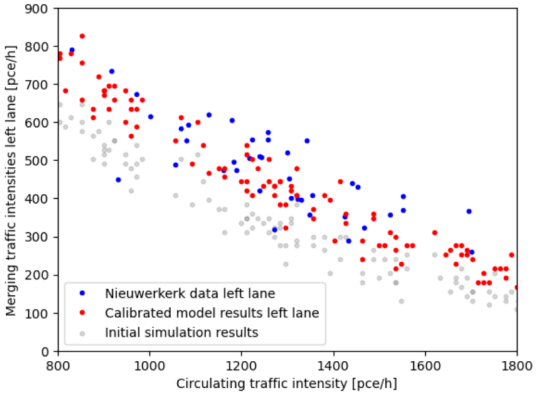
(b) Close-up of overlapping circulating traffic volumes

Figure 4.18: Comparison of the capacity curves of the calibrated simulation and the Nieuwerkerk data with the initial simulation data as a reference, for both entry lanes of the minor roundabout entrance. The red dots represent the calibrated simulation results, the blue dots represent the Nieuwerkerk data, and the transparent grey dots represent the initial simulation data.

Figures 4.19 and 4.20 compare the calibrated simulation data to the Nieuwerkerk data for the left and right lanes, respectively. Subfigure 4.19a shows that the merging capacity of the calibrated simulation model is approximately 1750 pce/h for no circulating traffic, which is higher than the 1500 pce/h of the initial simulation model. It decreases to zero at a circulating traffic volume of approximately 2600 pce/h. The graph shows a nonlinear relation between the merging traffic capacity and the circulating traffic volumes. When comparing it to the initial simulation model, one notices that the merging traffic capacity of the calibrated model increases the most for low circulating traffic volumes and that the difference between both decreases for increasing circulating traffic flows. Considering the close-up in Subfigure 4.19b, one sees that the calibrated simulation data approaches the Nieuwerkerk data better than the initial simulation model. This was also the conclusion from the RMNSEs in Table 4.16.



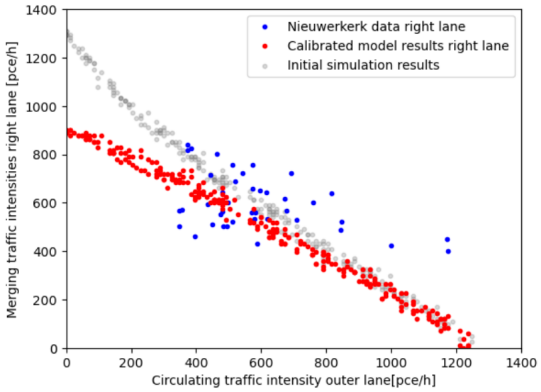
(a) Comparison of capacity curves of the left lanes of the calibrated model and Nieuwerkerk data, with initial simulation data as reference



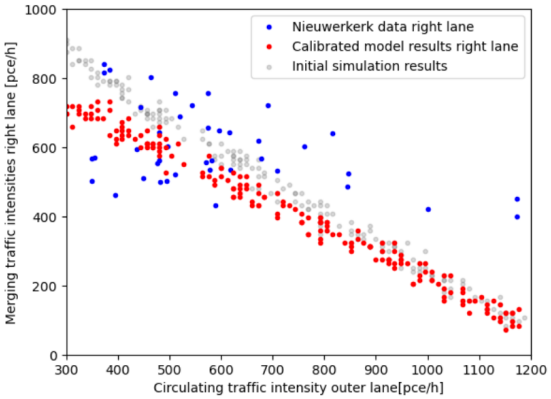
(b) Close-up of overlapping circulating traffic volumes

Figure 4.19: Comparison of the capacity curves of the calibrated simulation and the Nieuwerkerk data with the initial simulation data as a reference, for the left entry lane of the minor roundabout entrance. The red dots represent the calibrated simulation results, the blue dots represent the Nieuwerkerk data, and the transparent grey dots represent the initial simulation data.

Lastly, the resulting simulation outcome for the right lane after seven iterations is profoundly different. As mentioned previously, the calibration procedure resulted in inaccurate capacity curves. Subfigure 4.20a shows that the merging capacity of the calibrated model is approximately 900 pce/h for no circulating traffic volume and decreases to zero at a circulating traffic volume of 1200 pce/h. When comparing it to the initial simulation, one notices that the merging capacity with no circulating traffic decreased substantially from 1300 pce/h to 900 pce/h. In contrast to Figures 4.18 and 4.19, the calibrated model shows a concave nonlinear relation between the circulating traffic flow and the merging traffic capacity. This clashes with all literature regarding turbo-roundabout capacity, which shows a convex relation between the circulating and merging traffic flow.
At circulating traffic volumes greater than 500 pce/h, the difference between the initial and calibrated simulation outcomes decreases. Both models are almost indistinguishable at circulating traffic volumes greater than 1000 pce/h. Considering the circulating traffic volumes of the Nieuwerkerk data, as shown in Subfigure 4.20b, one can observe that the differences between both simulation models are less pronounced. At circulating traffic volumes of approximately 400 pce/h, the calibrated model better matches the Nieuwerkerk data. At higher circulating traffic volumes, both models are once again indistinguishable.



(a) Comparison of capacity curves of the right lanes of the calibrated model and Nieuwerkerk data, with initial simulation data as reference



(b) Close-up of overlapping circulating traffic volumes

Figure 4.20: Comparison of the capacity curves of the calibrated simulation and the Nieuwerkerk data with the initial simulation data as a reference, for the right entry lane of the minor roundabout entrance. The red dots represent the calibrated simulation results, the blue dots represent the Nieuwerkerk data, and the transparent grey dots represent the initial simulation data.

Even though the calibration regression lines already showed that the calibrated simulation model improved the fit between the simulation model and the Nieuwerkerk data for the left lane and slightly decreased for the right lane, plotting the capacity curves gives additional insights into the performance of both simulation models. After all, the circulating traffic volume range is substantially larger for the capacity curves than for the calibration regression lines. From the capacity curves of the calibrated HDV model, one can conclude that the fit between simulation and data improves for the left lane. However, the capacity curves of the right lane show a clear decrease in performance, and the final capacity curves are physically impossible.

4.5. Analysis of calibration procedure and calibrated model

This section critically analyses the calibration procedure and the capacity curves of the calibrated HDV model. Furthermore, a comparison is made with the theoretical capacity curves from the paper of Fortuijn and Salomons (Fortuijn and Salomons, 2020).

4.5.1. Calibration procedure

In the calibration procedure, a perfect fit between the Nieuwerkerk data and the simulation model was not obtained. While the VISSIM simulation model improved by 72.6% for the left lane, the calibration procedure for the right lane resulted in a 4.7% decrease in accuracy, as well as unrealistic capacity curves. Considering the aggregated lane data, an improvement of 16% is observed.
A common shortcoming of the calibration procedure for both separate lanes is the insensitivity of the simulation outcomes to changes in the critical gap and follow-up times at higher circulating traffic volumes. However, at low circulating traffic volumes, the impact of changes in either of the two parameters results in an observable simulation difference.

For the left lane, the final RMNSE of 0.074 indicates an adequate fit between the simulation and the Nieuwerkerk data. At a circulating traffic volume of 900 pce/h, the difference between both merging traffic capacities is only 0.73%, whereas it increases to 16.71% at a circulating traffic volume of 1800 pce/h. The iteration procedure of the left lane thus resulted in a model that performs well at simulating real-life conditions.

For the right lane, however, the iterative procedure did not improve the fit between the Nieuwerkerk data and the simulation outcomes. As mentioned, the simulation proved insensitive to changes in the critical gap and follow-up times for higher circulating traffic volumes. Due to a large difference in the slope of the Nieuwerkerk regression line and the initial simulation caused by major differences in the merging traffic capacity at high circulating traffic volumes, the fit between both could barely be improved. Therefore, the iterations were conducted to improve the fit between the Nieuwerkerk data and the simulation for lower circulating traffic volumes. Less attention was paid to the RMNSE. In hindsight, this proved to be the wrong strategy. The final RMNSE of 0.427 shows that the simulation does not particularly match the Nieuwerkerk data. The fourth iteration has the lowest RMNSE at 0.361, which is an 11.5% improvement over the initial simulation. Subsection 4.5.2 dives into more depth and describes the implication of this decision.
However, the iterations also have a few positives. First, the difference in slope between the simulation model and the Nieuwerkerk data has decreased. Next, the merging traffic capacities at a circulating traffic volume of 400 pce/h are almost equal. The difference between both is only 1.42%. For comparison, the difference between the two curves at a circulating traffic volume of 1200 pce/h is nearly 83%.

There are two reasons why the VISSIM model accurately represents the left access lane but fails to do so for the right lane. As mentioned earlier, the Nieuwerkerk data was simultaneously measured for the left and right lanes. Measurements were done when traffic on the left merging lane was congested for five minutes. However, as the right merging lane has fewer conflicts than the left lane, its merging capacity is greater than that of the right lane. This signifies that, while there was congestion on the left lane, some data points for the right lane do not represent congested traffic conditions. Of the 41 data points, only 19 were made in congested conditions on both access lanes.
It was only after the research that this was discovered, meaning that uncongested traffic data measurements were used in the calibration procedure for the right lane. This data was also used for the regression lines of the right lane of the Nieuwerkerk data, implying that these are also inaccurate. This is likely also why there is a substantial difference in slope between the regression lines of the left and right lanes of the Nieuwerkerk data.

Furthermore, the wrong strategy was applied in the calibration procedure. Once it was noticed that changes in the critical gap and follow-up times did not impact the capacity curves at higher circulating traffic intensities, the calibration procedure shifted towards achieving a good fit between the simulation and the Nieuwerkerk data for low circulating traffic intensities.

Given that the regression line for the right lane of the Nieuwerkerk data was inaccurate in the first place, this strategy might have resulted in an even less accurate final regression line.
To conclude, the calibration procedure for the right lane is done using the wrong regression line. This likely explains why the slope differs substantially between the simulated regression lines and the regression line of the Nieuwerkerk right-lane data. Therefore, no academic validity is given to the calibration process of the right lane.

4.5.2. Calibrated model

This subsection analyses the calibrated simulation model's obtained capacity curves. Furthermore, it compares the capacity curves of the left lane obtained from the calibrated HDV model with Fortuijn and Salomons's theoretical capacity curve. A comparison cannot be made for the right lane, as no theoretical capacity curves are available and the resulting capacity curve of the simulation is inaccurate.

Figures 4.18, 4.19, and 4.20 in Subsection 4.4.1 depict the capacity curves of the combined lanes, and of the left and right lanes separated. While the aggregate and left lane capacity curves show a realistic convex shape, the capacity curve for the right lane shows an unrealistic concave shape.
To obtain realistic values for the merging capacity of the left lane, the obtained critical gap and follow-up times from the literature had to be decreased by 28% and 10%, respectively. Figure 4.21 compares the theoretical human model of Fortuijn and Salomons with the capacity curve of the left lane of the calibrated HDV model. The black line represents the theoretical 'human' model, whereas the red dots indicate the simulation outcomes. The theoretical curves are used as a background for the calibrated model.

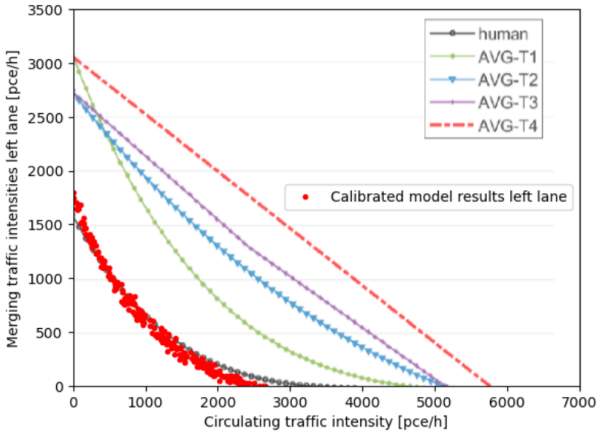


Figure 4.21: Comparison of simulation outcomes with theoretical 'human' model of Fortuijn and Salomons (Fortuijn and Salomons, 2020). The red dots indicate the calibrated model and are plotted over the theoretical curves of Fortuijn and Salomons.

The figure shows that the theoretical model and the VISSIM simulation return similar capacity curves. At low circulating traffic volumes, the capacity of the simulation model is slightly higher than the theoretical model. This aligns with expectations, as the follow-up time of the VISSIM simulation is lower than that of the theoretical model. At a circulating traffic volume of approximately 500 pce/h, the merging traffic capacity of the theoretical model becomes greater than the simulation model. Only at high circulating volumes do the differences between both capacity curves increase. Whereas the VISSIM model reaches an effective merging capacity of 0 at 2600 pce/h, the merging capacity of the theoretical model is approximately 125 pce/h. It is only at a circulating traffic volume of 4100 pce/h that it reaches zero.

The larger difference at high circulating traffic volumes was expected as the iteration process showed that the capacity curves of the VISSIM model at high circulating traffic volumes could not be changed through the critical gap and follow-up times. Therefore, the VISSIM model performs best at lower circulating traffic volumes. Given that the final RMNSE is only 0.076 and that there is a good match with the theoretical model of Fortuijn and Salomons, one can conclude that the VISSIM model is adequate at simulating realistic HDV behaviour at the left lane of a turbo-roundabout.

Figure 4.18 shows the capacity curve of the right lane. Due to incorrect data usage in the calibration procedure, the model was wrongfully calibrated. The resulting capacity curves are, therefore, invalid and cannot be used for further research. However, some remarks about the obtained capacity curve are made.
By only focusing on the circulating traffic volume range of the Nieuwerkerk data in the iteration procedure, the impact of changes in the critical gap and follow-up time on other regions of the capacity curve was not observed. At low circulating traffic volumes, the merging capacity of the calibrated model is much lower than the initial simulation. Instead of 1300 pce/h, the merging capacity without circulating traffic is only 900 pce/h.
The biggest issue with the capacity curve is its concave shape, which does not align with the theory. This was the first sign of a mistake in the calibration process.

To conclude, this chapter proved that VISSIM can accurately model the behaviour of HDVs at the conflict area for the left lane of a minor turbo-roundabout branch. Through an iterative calibration procedure, a final RMNSE of 0.074 was achieved. Furthermore, only subtle differences were observed when comparing the final capacity curve with the theoretical curves of Fortuijn and Salomons.
For the right lane, a different story applies. Some mistakes in the calibration process proved the resulting capacity curve unrealistic. Because of this mistake, one cannot conclude whether VISSIM can accurately model the behaviour of HDVs in the conflict area of the right lane. Carrying out the calibration process again using only data points at capacity could answer this question. Furthermore, it might be beneficial to consider more parameters during the calibration procedure instead of only the critical gap and follow-up times.

5

Microscopic AV model

This chapter describes the development of the three AV models, which ultimately determine if the theoretical claims about the achievable capacity increase, as stated in the paper of Fortuijn and Salomons, can be approached in a microscopic simulation model. As Subsection 3.3.2 mentions, VISSIM has three integrated car-following models for different levels of aggression for AVs. For each of these three car-following models, the merging behaviour and, more specifically, the critical gap times are the same. Sections 5.1, 5.2, and 5.3 describe the simulation results of the cautious, normal, and aggressive AV models, respectively. Section 5.4 compares the three models to one another, and Section 5.5 determines the capacity increases of the various AV models over the calibrated HDV model. Finally, Section 5.6 critically analyses the results of the AV models.

5.1. Cautious AV model

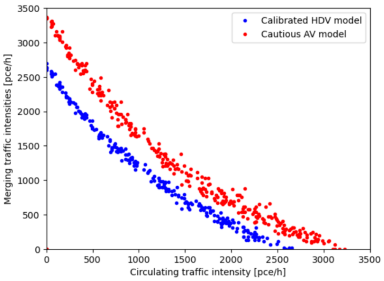
The cautious model simulates AVs with cautious car-following behaviour. These keep a more considerable safety distance from other vehicles than human-driven vehicles would keep in reality, as explained in Subsection 3.3.2. As Subsection 3.3.1 states, the follow-up times do not serve as input for the model. Instead, these are deducted from the simulation data. By dividing the time of one measurement period by the merging traffic volumes under free flow but at maximal capacity, one obtains the average follow-up times between vehicles. Table 5.1 contains these obtained follow-up times and all other changed parameters for the cautious AV model. Where the literature on HDVs stated critical gap times for the left and right lanes, the literature on AVs states critical gaps for the inner and outer circulatory roadway.

Table 5.1: Overview of cautious AV model parameters

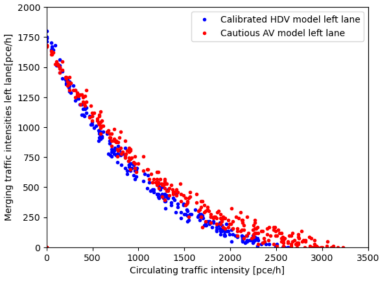
Category	Parameter	Location	Value
Vehicle speeds	Reduced speed areas	Roundabout+conflict areas	30 km/h
	Desired speed dist.	Access branches	80 km/h
Merging behaviour	Critical gap	Inner lane	2.34 s
		outer lane	2.79 s
	Follow-up time	Minor-left	2.16 s
		Minor-right	2.14 s
Car-following behaviour	Car-following model	Entire network	AV cautious

Capacity curves

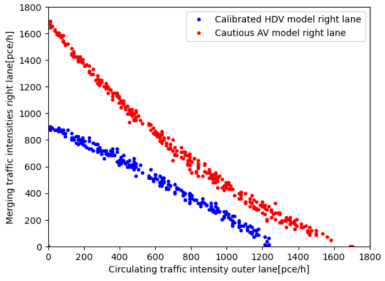
The capacity curves of the cautious AV model are compared to those of the calibrated HDV model to show the difference in capacity between both models. Figure 5.1 plots the capacity curves for the cautious AV model. The red dots represent the AV model results, and the blue dots indicate the calibrated HDV model results. Subfigure 5.1a shows the capacity curve of both merging lanes of the minor roundabout branch. One can observe noticeable differences between the data of the calibrated HDV model and the cautious AV model. With no circulating traffic, the merging traffic capacity of the cautious AV model is approximately 26% higher than the calibrated HDV model. Where the calibrated model has a merging capacity of 2650 pce/h, the cautious AV model has a merging capacity of 3350 pce/h. This aligns with expectations, as the follow-up times for the right lane are larger in the calibrated HDV model, and the follow-up times for the left lane are approximately equal. Where the calibrated HDV model has follow-up times of 2.05 and 4.1 s for the left and right lanes, the cautious AV model has follow-up times of 2.16 and 2.14 s. With no circulating traffic, the follow-up time is the only determining factor for the merging capacity, causing the two models to have differing merging capacities. As the circulating traffic volume increases, the differences in merging traffic capacity between both models decrease. At low circulating traffic volumes, the merging traffic capacity of the cautious AV model decreases faster than the calibrated HDV model. At a circulating traffic volume of 2600 pce/h, the merging capacity of the calibrated HDV model decreases to zero. On the contrary, the cautious AV model has a remaining merging traffic capacity of approximately 250 pce/h. It only reaches a merging capacity of zero at a circulating traffic volume of 3200 pce/h.



(a) Comparison of the capacity curves of the calibrated HDV model and the cautious AV model of both entry lanes



(b) Comparison of the capacity curves of the calibrated HDV model and the cautious AV model of the left lane



(c) Comparison of the capacity curves of the calibrated HDV model and the cautious AV model of the right lane

Figure 5.1: Comparison of the capacity curves of the calibrated HDV model and the cautious AV model for the minor roundabout entrance. The blue dots indicate the calibrated HDV model results, and the red dots represent the simulation data of the cautious AV model.

Subfigure 5.1b shows the capacity curve of the left lane, and Subfigure 5.1c shows the capacity curve of the right entrance lane. Even though the simulation did not result in realistic capacity curves for the right lane, the comparison provides an overview of the performance of AVs on the right access lane. The differences between both capacity curves for the left lane are minor. Whereas the calibrated HDV model has a merging traffic capacity of 1750 pce/h with no circulating traffic, the cautious AV model has a merging capacity of 1680 pce/h. This aligns with expectations, as the follow-up times of the calibrated model are slightly lower than those of the cautious AV model (2.05 vs 2.16s). At a circulating traffic volume of approximately 200 pce/h, the merging capacities of both models are approximately equal. For greater circulating traffic volumes, the merging traffic capacity of the cautious AV model is greater. At a circulating traffic volume of 2600 pce/h, the merging capacity of the AV model is approximately 125 pce/h, whereas it has already decreased to zero for the calibrated HDV model. The cautious AV model's merging capacity reaches zero at a circulating traffic volume of 3100 pce/h.

One can conclude that the merging traffic capacity of the calibrated HDV model decreases faster than that of the cautious AV model, which aligns with expectations as the critical gap of the former is greater than that of the latter. For the right lane, the capacity curve of the cautious AV model shows a more linear course than that of the left lane. This is expected, as the right lane only conflicts with one traffic stream, whereas the left lane conflicts with two traffic streams. The merging traffic capacity without circulating traffic is approximately 100 pce/h lower than for the right lane. The maximum circulating traffic volume for which traffic can merge onto the turbo-roundabout is approximately 1650 pce/h.

5.2. Normal AV model

The normal model simulates AVs with normal car-following behaviour. These vehicles can measure the distances to and speeds of surrounding vehicles, improving their ability to predict sudden changes and creating a more homogeneous traffic flow. Table 5.2 contains the model parameters of the normal AV model. The follow-up times of the normal AV model decreased from 2.16 and 2.14s for the cautious AV model to 1.59 and 1.56s, respectively. Other than the follow-up times, there are no differences between the cautious and normal AV model parameters.

Table 5.2: Overview of normal AV model parameters

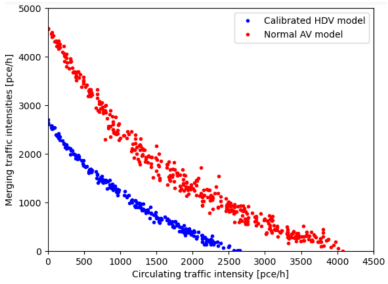
Category	Parameter	Location	Value
Vehicle speeds	Reduced speed areas	Roundabout+conflict areas	30 km/h
	Desired speed dist.	Access branches	80 km/h
Merging behaviour	Critical gap	Inner lane	2.34 s
		outer lane	2.79 s
	Follow-up time	Minor-left	1.59 s
		Minor-right	1.56 s
Car-following behaviour	Car-following model	Entire network	AV normal

Capacity curves

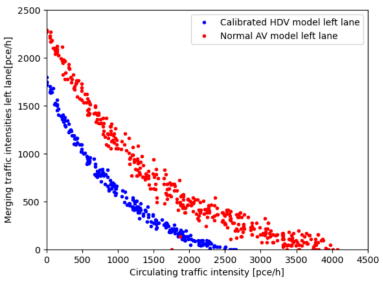
The capacity curves of the normal AV model are also compared to those of the calibrated HDV model. Figure 5.2 plots the capacity curves for the normal AV and calibrated HDV model. Subfigure 5.2a shows the capacity curves of both merging lanes of the minor roundabout branch. The difference between both capacity curves is greater than for the cautious AV model. With no circulating traffic volume, the merging traffic capacity of the normal AV model is 4600 pce/h, which is almost 74% higher than the calibrated HDV model. This aligns with expectations since the follow-up times of the normal AV model are substantially smaller than those of the calibrated HDV model: 1.59 and 1.56s versus 2.05 and 4.1. For the entire range of circulating traffic volumes, the merging capacity of the normal AV model is substantially higher than that of the calibrated HDV model. Where the maximum circulating traffic volume for which traffic can merge onto the turbo-roundabout is 2500 pce/h for the HDV model, it approaches 4000 pce/h for the normal AV model. At a circulating traffic volume of 2500 pce/h, the merging traffic capacity of the normal AV model is still 1000 pce/h.

Subfigure 5.2b shows the capacity curves for the left lane. In contrast to the capacity curves of the previous comparison, the differences between both capacity curves are larger. The merging capacity is higher, no matter the circulating traffic volume. With no circulating traffic, the merging traffic capacity of 2270 pce/h is substantially higher than the 1750 pce/h for the calibrated HDV model. Furthermore, the maximum circulating traffic volume for which traffic can merge onto the turbo-roundabout is also higher at 4100 pce/h compared to 3000 pce/h. This also aligns with expectations.

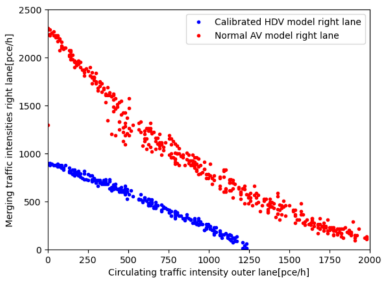
For the capacity curves of the right lane in Subfigure 5.2c, one can observe a similar merging capacity for the normal AV model with no circulating traffic volume as in the left lane. The merging traffic capacity reaches zero at a circulating traffic volume of approximately 2000 pce/h. One peculiar observation in the capacity curve for the normal AV model is that the data spread at a circulating traffic volume of approximately 500 pce/h is larger than it is for other circulating traffic volumes. This peculiarity is described in more detail in Subsection 5.6.1 of the analysis.



(a) Comparison of the capacity curves of the calibrated HDV model and the normal AV model of both entry lanes



(b) Comparison of the capacity curves of the calibrated HDV model and the normal AV model of the left lane



(c) Comparison of the capacity curves of the calibrated HDV model and the normal AV model of the right lane

Figure 5.2: Comparison of the capacity curves of the calibrated HDV model and the normal AV model for the minor roundabout entrance. The blue dots indicate the calibrated HDV model results and the red dots represent the simulation data of the normal AV model.

5.3. Aggressive AV model

The aggressive AV model simulates AVs with aggressive car-following behaviour. These vehicles are all-knowing and have increased awareness and predictive capabilities, leading to smaller gaps in all situations. Table 5.3 contains the model parameters of the aggressive AV model. Compared to the cautious AV model, the follow-up times decreased from 2.16 and 2.14 s to 1.33 and 1.34 s, respectively. Once again, no other changes are made to the model.

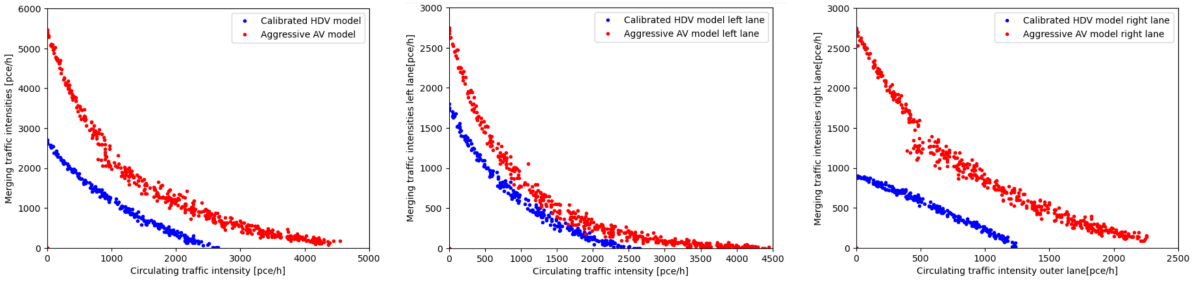
Table 5.3: Overview of parameters used in AV aggressive model

Category	Parameter	Location	Value
Vehicle speeds	Reduced speed areas	Roundabout+conflict areas	30 km/h
	Desired speed dist.	Access branches	80 km/h
Merging behaviour	Critical gap	Inner lane	2.34 s
		outer lane	2.79 s
	Follow-up time	Minor-left	1.33 s
		Minor-right	1.34 s
Car-following behaviour	Car-following model	Entire network	AV aggressive

Capacity curves

Figure 5.3 plots the capacity curves for the aggressive AV car-following and calibrated HDV model. Subfigure 5.3a shows the capacity curves of the left and right merging lanes for the minor roundabout branch. Surprisingly, the difference between the aggressive AV model and the HDV model seems less substantial than for the normal AV and HDV models. At low and high circulating traffic volumes, the merging traffic capacities of the aggressive AV model are higher than the normal AV model, but not for circulating traffic volumes in between. Section 5.6.1 describes this in more detail. With zero circulating traffic, the merging traffic capacity is approximately 2800 pce/h larger than the calibrated HDV model at 5400 pce/h. This aligns with expectations since the follow-up times of the aggressive AV model are substantially smaller at 1.33 and 1.34 s versus 2.05 and 4.1s.

The merging traffic capacity decreases faster than the calibrated HDV model for low circulating traffic volumes, and the maximal circulating traffic volume for which traffic can merge onto the turbo-roundabout is 4500 pce/h, which is 1950 pce/h more than the calibrated HDV model.
Subfigure 5.3b shows the capacity curves for the left lane. Without circulating traffic, the merging traffic capacity of the aggressive AV model is approximately 2600 pce/h, substantially larger than the merging traffic capacity of 1750 pce/h of the calibrated HDV model. However, as the circulating traffic volume increases, the differences in merging capacity between the AV and HDV models decrease. This is a peculiar result since the critical gap times of 2.34 and 2.79s are smaller than those of the calibrated HDV model. Thus, one would not expect the AV model's merging capacity to decrease faster than the calibrated HDV model, especially since the aggressive AV model is presumably all-knowing. Another peculiarity in the capacity curve is the long tail at high circulating traffic volumes. The merging traffic capacity is nearly zero at a circulating traffic volume of 3000 pce/h. However, it reaches zero for a circulating traffic capacity of 4500 pce/h. Thus, it seems that merging vehicles can find gaps for very high circulating traffic volumes.
Subfigure 5.3c depicts the capacity curves for the right lane. For no circulating traffic, the merging traffic capacity of the aggressive AV model is approximately 2600 pce/h. A break in the simulation data appears at a circulating traffic volume of 500 pce/h. This is an unexpected result, but it is present at the same location in the capacity curve as for the normal AV model. Traffic flow at a circulating traffic volume of 500 pce/h seems more heterogeneous and disturbed for the normal and aggressive AV models.



(a) Comparison of the capacity curves of the calibrated HDV model and the aggressive AV model of both entry lanes

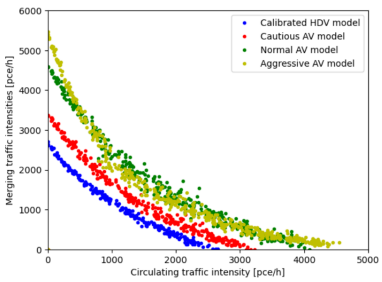
(b) Comparison of the capacity curves of the calibrated HDV model and the aggressive AV model of the left lane

(c) Comparison of the capacity curves of the calibrated HDV model and the aggressive AV model of the right lane

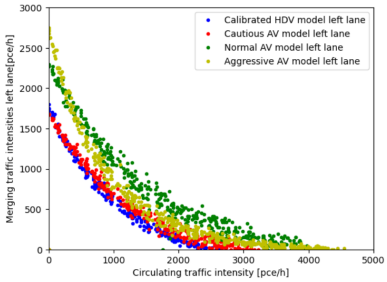
Figure 5.3: Comparison of the capacity curves of the calibrated HDV model and the aggressive AV model for the minor roundabout entrance. The blue dots indicate the calibrated HDV model results, and the red dots represent the simulation data of the aggressive AV model.

5.4. Comparison of AV models

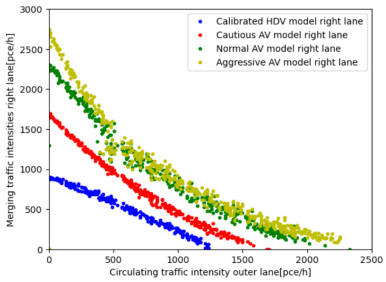
Figure 5.4 plots the capacity curves of the three AV models against one another and the calibrated HDV model. The blue dots represent the calibrated HDV model, the red dots represent the cautious AV model, the dark green dots represent the normal AV model, and the light green dots represent the aggressive AV model. Subfigure 5.4a shows the combined capacity curves for both entry lanes of the minor branch of the turbo-roundabout. One observes substantial differences between the capacity curves of the various AV models, some of which are unexpected. It aligns with expectations that the aggressive AV model has the highest merging traffic capacity when there is no circulating traffic on the roundabout because its follow-up times are the smallest of all AV models. The normal and aggressive AV models perform better than the cautious AV model, which has a similar capacity curve to the calibrated HDV model. At a circulating traffic volume of approximately 500 pce/h, the course of the capacity curves no longer aligns with expectations. Here, the merging capacity of the normal AV model is greater than that of the aggressive AV model.



(a) Comparison of capacity curves of the three AV models for the left and right lanes



(b) Comparison of capacity curves of the three AV models for the left lane



(c) Comparison of capacity curves of the three AV models for the right lane

Figure 5.4: Comparison of the capacity curves of the various AV models and the calibrated HDV model for the minor roundabout entrance. The blue dots represent the calibrated HDV model, the red dots represent the cautious AV model outcome, the green dots represent the normal AV model outcomes, and the light green dots represent the aggressive AV model outcomes.

Given that the critical gap times of both models are the same and that the follow-up times of the aggressive AV model are lower than those of the normal AV model, one would not expect the merging capacity of the normal AV model to be higher than that of the aggressive AV model. Only at the extreme values of circulating traffic flow does the aggressive AV model outperform the normal AV model. Subsection Subfigure 5.4b, depicting the capacity curves for the left lane, shows the same peculiarity, whereas Subfigure 5.4c, displaying the capacities for the right lane, does not. The peculiarity thus originates from the left entrance lane. To explain this, a deep dive into the simulation parameters of the VISSIM models is necessary, which is elaborated in Subsection 5.6.1. At circulating traffic volumes greater than 3300 pce/h, the merging capacity of the left lane of the normal AV model decreases to zero. In contrast, the aggressive AV model only reaches zero at a circulating traffic flow of approximately 4600 pce/h.

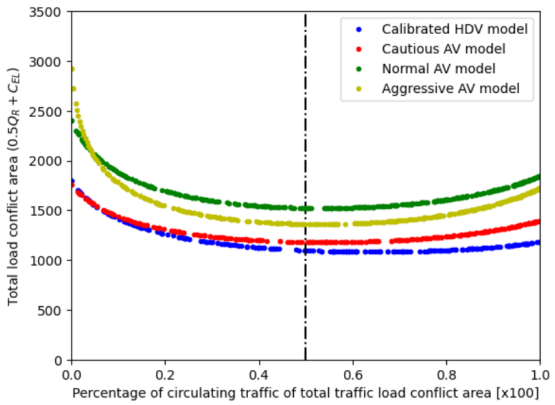
The aggressive AV model has a higher merging traffic capacity for the right merging lane than the cautious and normal AV models for all circulating traffic volumes. However, even though the capacity curves of the normal and aggressive AV models of the right lane are more in line with expectations, they also show a peculiar result. For both models, a disruption in the capacity curves can be noticed for a circulating traffic volume of approximately 500 pce/h. Whereas all capacity curves in this report are relatively smooth, the capacity curves for the right lane of the normal and aggressive AV models show a disturbance. There is an apparent and temporary dip in merging traffic capacity, which disappears at a circulating traffic volume of 600 pce/h and higher. Subsection 5.6.1 analyses this peculiarity.

5.5. Obtainable capacity increase of a turbo-roundabout through microscopic simulation

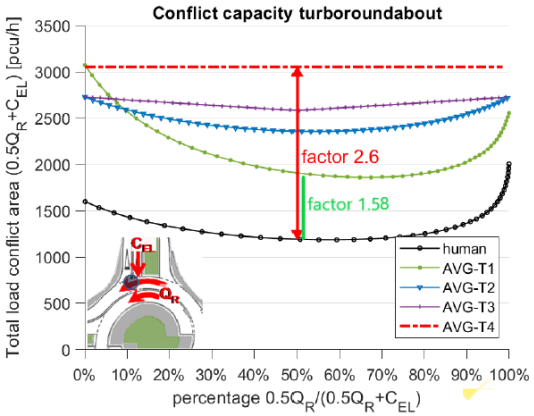
This section describes the last step in obtaining the necessary results that answer the main research question of this thesis. As Section 3.4 explains, the capacity increase of a turbo-roundabout considering AVs can be determined by calculating the difference between the regression lines of the HDV model and those of the AV models. The capacity increases are calculated for traffic conditions where the circulating traffic volume equals the merging traffic flow. Since the left lane is decisive for the capacity of a turbo-roundabout, the capacity increase for the turbo-roundabout of the microscopic simulation models is based on the capacity increase for the left lane. Furthermore, the capacity curves for the right lane in this thesis are inaccurate and cannot be used to formulate conclusions.

Figure 5.5 depicts the theoretical regression lines of Fortuijn and Salomons compared to the calibrated HDV and AV models for the left lane of the minor branch of the turbo-roundabout. Subfigure 5.5a shows the regression lines for the left lane of the microscopic simulation models developed in this report, where the blue line represents the calibrated HDV model, the red line represents the cautious AV model, the dark green line represents the normal AV model, and the light green line represents the aggressive AV model. On the y-axis, the figure displays the total traffic load of the conflict area, which is calculated as the sum of traffic on the outer circulatory lane and the merging traffic from the left lane. The x-axis plots the percentage of the total traffic load of circulating traffic on the outer lane. The vertical dashed line represents the fraction of circulating traffic for which the capacity increases are calculated.

For a circulating traffic volume representing 50% of the total traffic in the conflict area, one notices that the normal AV model reaches the highest total traffic load, followed by the aggressive and cautious AV models. The capacity of the conflict area in the normal AV model thus increases the most compared to the calibrated HDV model. Table 5.4 depicts the exact values of the total loads for the four simulation models at a 50-50 traffic distribution and the percentage differences between the calibrated HDV model and the three AV models. The results from the table show that the maximum capacity of the cautious model increases by 7.47%, that the capacity of the aggressive model increases by 24.32%, and that the capacity of the normal AV model increases by nearly 39%. As the capacity curves in Figure 5.4 already showed, the normal AV model outperforms the aggressive AV model. The percentage increase in traffic volume on the conflict area for the normal AV model is almost twice as high than that of the aggressive AV model.



(a) Depiction of microscopic simulation regression lines for the left lane of the calibrated HDV and AV models. The blue line represents the calibrated HDV model, whereas the red, green and light green lines represent the cautious, normal, and aggressive AV models.



(b) Depiction of theoretical regression lines from the paper of Fortuijn and Salomons that determine the increased capacity for the various vehicle models (Fortuijn and Salomons, 2020).

Figure 5.5: Comparison of capacity curves of the theoretical model of Fortuijn and Salomons and the capacity curves of the microscopic AV models

Table 5.4: Total traffic load of the conflict area of the left lane for 50% circulating traffic for the various simulation models

Model	Total load [pce/h]	Improvement [%]
Calibrated HDV	1098	-
AV cautious	1180	7.47
AV normal	1522	38.62
AV aggressive	1365	24.32

Subfigure 5.5b depicts the theoretical regression lines of Fortuijn and Salomons and shows that a total capacity increase of 260% for AVs is achievable. However, they obtain this increase in capacity for a model with gap synchronisation, headway optimisation and course guidance. Instead, the increase in capacity between the green AVG-T1 model regression line and the black human model regression lines should be compared since the AVG-T1 model best matches the simulation models of this thesis. As mentioned, the AVG-T1 model does feature intra-vehicle communication, implying that the increase in capacity for the simulation models will not be as high. The total load of the conflict area for the human model at a 50-50 distribution is 1200 pce/h, whereas the total load for the AVG-T1 model is 1900 pce/h. Percentage-wise, this is an increase of 58.3%. The theoretically obtained capacity increase of vehicles on the minor branch of a turbo-roundabout is approximately 20% higher than the simulations in this report.

Due to the poor quality of the regression curves obtained in Section 4.4, the capacity increase of the right lane is not calculated. While it is possible to determine the percentage improvements for the three AV models, the resulting increases are unrealistic. Nevertheless, the regression lines of the three AV models are depicted in Figure 5.6. In contrast to Figure 5.5a, the aggressive AV model performs better than the normal AV model, leading to a higher maximum traffic load in the conflict area. Disregarding the 'calibrated' HDV model, the simulation outcomes align with expectations for the right lane.

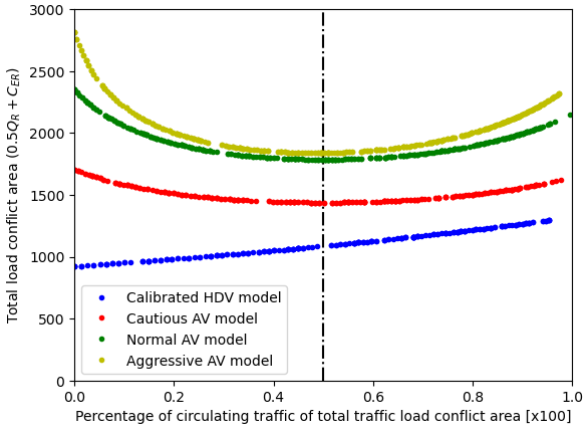


Figure 5.6: Depiction of microscopic simulation regression lines for the right lane of the calibrated HDV and AV models. The blue line represents the calibrated HDV model, whereas the red, green and light green lines represent the cautious, normal, and aggressive AV models.

5.6. Analysis of capacity curves and obtained capacity increases of the AV models and obtained capacity increases for the various AV models

This section analyses the results of the AV models in more detail and touches upon some of the peculiarities encountered in the results. The first major peculiarity encountered is the out-performance of the normal AV model over the aggressive model. This is explained in Subsection 5.6.1. Subsequently, Subsection 5.6.2 critically analyses the obtained capacity increases for the three AV models.

5.6.1. Normal AV model outperforming the aggressive AV model for the left lane

Figure 5.4, displaying the capacity curves of the three AV models for the three different lane configurations, shows some unexpected results. As stated in Section 5.4, the normal AV model surprisingly outperforms the aggressive AV model over a large range of circulating traffic on the left merging lane.

Since the follow-up times of the aggressive AV model are smaller than the normal AV model, and all other parameters are equal, one would expect that the merging capacity of the aggressive model is higher than the normal AV model for all circulating traffic volumes. However, the capacity curves show this is only for low circulating traffic volumes.
To find a cause for this peculiarity, a deeper dive into the settings of both simulation models is required. Since the only difference between the aggressive and normal AV models is the car-following behaviour, the mutual differences between both are considered. The first noticeable differences are the number of interaction objects and vehicles and the look-ahead distances. The normal AV model features two interaction objects and one interaction vehicle for a look-ahead distance of 250 metres, indicating that the sensors of the automated vehicles can only observe the next downstream vehicle within a distance of 250 metres. The interaction vehicles are part of the interaction objects, meaning an AV can only observe one other object next to a downstream vehicle. Interaction objects include network items such as reduced speed areas, red signal heads, priority rules, and stop signs. Conflict areas are not considered interaction objects.
In contrast, the aggressive AV model features ten interaction objects and eight interaction vehicles over a look-ahead distance of 300 metres. An aggressive AV vehicle thus takes more downstream vehicles into account than the normal AV car-following behaviour. A second difference between the normal and aggressive AV models is the car-following parameters. The aggressive AV model features shorter standstill distances, shorter following gap times, and higher acceleration rates.

Possibly, the number of interaction vehicles or objects limits the merging traffic capacity of the aggressive AV model. More specifically, the number of interaction vehicles might be the cause since the only interaction objects in the VISSIM model are the reduced speed areas. It is unknown whether the number of interaction vehicles also includes vehicles on other network links, but given the capacity curves, this report assumes that it does. This would also explain why this peculiarity is only observed for the left entry lane, not the right lane. As mentioned earlier in this report, vehicles on the left merging lane conflict with both circulatory lanes, whereas vehicles on the right merging lane only conflict with the outer roundabout lane. Since a reduced-speed area is situated in the conflict areas, cautious and normal AV models can only observe one downstream vehicle. This implies that vehicles in the left merging lane can only observe a vehicle on one of the two circulatory lanes at once. At higher circulating traffic volumes, left merging vehicles thus observe one vehicle on the outer lane and do not observe any vehicles on the inner roundabout lane. They only observe them once the vehicle on the outer lane has left their observation area. This makes the merging vehicles blind to what happens on the inner roundabout lane, making them more likely to merge onto the roundabout.
On the other hand, an aggressive AV can simultaneously observe seven other vehicles on both circulatory lanes. It changes its merging behaviour accordingly, making an aggressive vehicle less likely to merge onto the turbo-roundabout at higher circulating traffic volumes since it observes more potential conflicts. At a turbo-roundabout, observing multiple upstream vehicles per circulatory lane does not add any value since only the first upstream vehicles influence the merging behaviour of vehicles. Therefore, the number of interaction vehicles of the aggressive model could be reduced to those of the other two AV models. This will likely result in a higher capacity for the aggressive AV model but is left for future research.
This phenomenon does not occur in traffic conditions where there is no circulating traffic on the turbo-roundabout. Therefore, the aggressive AV model outperforms the normal AV model in these traffic conditions since its follow-up times are smaller than those of the normal AV model: 1.33 and 1.34s versus 1.59 and 1.56s.

In the right merging lane, the aggressive AV model has a higher merging traffic capacity than the normal AV model for all circulating traffic volumes. The right lane does not show the same peculiarity as the left lane since merging vehicles only have to consider one circulatory lane. Observing one or seven vehicles downstream makes no difference since only the first vehicle is essential. This leads to higher merging traffic capacities for the aggressive AVs. However, even though the capacity curves of the normal and aggressive AV models are more in line with expectations, they also show a peculiarity. Figure 5.4c shows a disruption in the capacity curves of the normal and aggressive AV models for a circulating traffic volume of approximately 500 pce/h. As the circulating traffic volume increases, this disruption disappears, and the models perform as expected. The reasons behind this deviation are unknown.

As mentioned earlier, the achieved automation of the AV models in this thesis matches best with the AVG-T1 model of Fortuijn and Salomons. Considering the follow-up times of the three AV models in this report, the aggressive car-following model was expected to match best with the AVG-T1 model as its follow-up times (1.33 and 1.34s) are closest to that of the AVG-T1 model (1.17s). However, due to the poorer performance of the aggressive AV model compared to the normal AV model, the obtained capacity increase of the normal AV model, with follow-up times of 1.59 and 1.56s, is considered the decisive one.

Figure 5.7 depicts the nonlinear regression lines of the capacity curves for the left entrance lanes of the three AV models, plotted over the theoretical curves of Fortuijn and Salomons. This gives a better understanding of the relative performance of each simulated AV model. Once again, only the AVG-T1 model is of interest. As expected, the cautious AV model performs worse than the theoretical AVG-T1 model. Due to the higher follow-up times, the merging capacity without circulating traffic is only half that of the theoretical model. At higher circulating traffic volumes, the difference in merging traffic flow decreases. The regression line of the normal AV model best matches with the theoretical AVG-T1 model but shows some substantial differences nonetheless.
First, the merging capacity for no circulating traffic is smaller (2500 pce/h vs 3000). Given that the follow-up times of the normal AV model of 1.59 and 1.56 s are larger than those of the AVG-T1 model (1.17s), this aligns with expectations. Given that the AVG-T1 model features communication between vehicles, it is evident that the follow-up times of this model are shorter than those of the three AV models. Since the critical gap times of both models are the same, the general shape of both capacity curves is approximately equal. However, the normal AV model shows a maximal circulating traffic volume of 3900 pce/h, whereas the AVG-T1 model shows a maximum of 5000 pce/h. At high circulating traffic volumes, the merging capacity of the simulation model decreases faster than the theoretical model. This peculiar result is probably explained by the difference in follow-up times. Given that the merging capacity of the AVG-T1 model is small for circulating traffic volumes greater than 4000 pce/h, the lower follow-up times of this model could lead to a slightly more efficient use of larger gaps in the circulating traffic flow, possibly allowing more vehicles to merge into the turbo-roundabout.
Another potential cause for this difference at high circulating traffic volumes is how traffic on the merging lanes is modelled in the VISSIM model. At all times, the access lanes of the turbo-roundabout are congested. At high circulating traffic volumes, this implies that vehicles always have to merge from a standstill. At a standstill, the gaps between vehicles on the circulatory lanes must be larger to merge onto the turbo-roundabout as the merging vehicle needs to accelerate. Next, the circulatory traffic of the models in this thesis originates only from the previous access branch, causing the traffic flow on the circulatory lanes to be more homogeneous and compact, leaving fewer merging possibilities. Distributing the traffic over all four access branches of the roundabout would lead to a more realistic traffic flow on the circulatory lanes of the roundabout and likely more gaps large enough for merging vehicles.

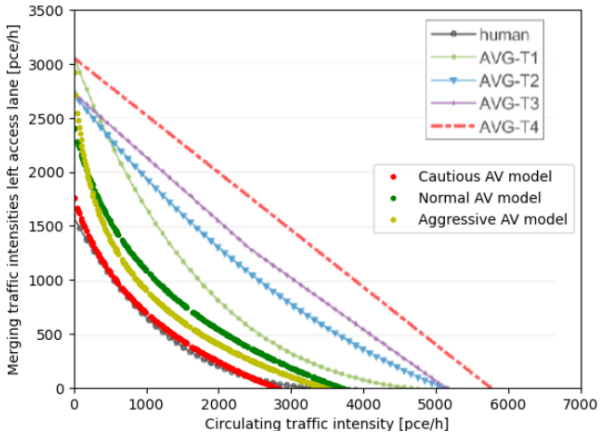


Figure 5.7: Nonlinear regression curves of the left lane of the different AV models

5.6.2. Obtained capacity increase for automated vehicles on a turbo-roundabout

Overall, the additional capacity of 58.3% of the theoretical AVG-T1 model over to the theoretical human model could not be obtained in the simulations of this thesis. The highest achievable capacity increase for the normal AV model is 38.62%, as depicted in Table 5.5.

As stated, it is unexpected that the obtained capacity increase of the normal AV model is higher than the aggressive AV model. Subsection 5.6.1 found the cause of this peculiarity but did not explain the difference between the theoretically determined and simulated capacity increases. The lower increase for the microscopic traffic simulation model is not surprising, as there are a few explanations for this difference.

First, a microscopic traffic simulation model has many more interactions than a theoretical capacity model. Therefore, the capacity increase of a microscopic traffic simulation model is likely to be lower than that of its theoretical counterpart.

Secondly, the follow-up times of the theoretical model can be shorter than the simulation models as the AVG-T1 model features communication between vehicles. They thus need less time to respond to other vehicles than those of the simulation models, which first have to register other vehicles through their sensors before being able to respond. Having shorter follow-up times leads to a higher potential capacity.

Table 5.5: Overview of the achieved capacity increases for the various models

Model	Capacity increase [%]
Theoretical model	58.3
AV cautious	7.47
AV normal	38.62
AV aggressive	24.32

The simulation model still leaves room for improvement. How traffic spawns in the network could impact the simulation outcomes. In the simulations, traffic on the roundabout only originates from the previous access branch to achieve a higher circulating traffic flow for the calibrated HDV model. With traffic coming from all branches of the turbo-roundabout, the maximum traffic volume on the turbo-roundabout was limited, leading to incomplete capacity curves. A sufficient circulating traffic volume could only be achieved by originating traffic only from the previous leg. For the AV models, a change in traffic loading was not considered. This is a limitation of this thesis, which could lead to more realistic results. After all, having traffic only from the previous access branch is unrealistic.

Another consideration entails the critical gap times of the three AV models. In this thesis, all AV models feature the same critical gap times of 2.34 and 2.79s, but one could argue that this is not highly realistic. The standard settings for AVs in VISSIM do not use different critical gap times for the three AV models. For the more aggressive AV models, lower critical gaps than the cautious AV model are logical. After all, they take more risks and are more aware of their environment. They should thus be able to accept lower critical gaps in the circulating traffic without risk.
Furthermore, there are numerous other parameters in VISSIM that can be tweaked and that could potentially influence the simulation results.

6

Conclusions and recommendations

The purpose of this thesis was to develop a calibrated microscopic traffic simulation model for HDVs in VISSIM to investigate whether the VISSIM traffic simulation model performed adequately at simulating HDV behaviour on a turbo-roundabout by comparing its outcomes to real data measurements on a turbo-roundabout. Then, this calibrated model was used to build an AV model upon which claims about the obtainable capacity increase on turbo-roundabouts were formulated. As stated in Section 1.2, this thesis answers one main research question that is aimed at investigating the potential capacity increase of AVs at turbo-roundabouts, as well as one sub-question about the calibration process. Section 6.1 formulates an answer to the sub-question, while Section 6.2 formulates an answer to the main research question. To wrap up this thesis, Section 6.3 touches upon the limitations of this thesis and Section 6.4 makes recommendations for future research.

6.1. Calibration of the HDV model

The sub-research question was formulated as follows:

What are the key parameters defining human driving behaviour at a turbo-roundabout in a microscopic traffic simulation capacity model, and how can these be used for model calibration, ensuring an accurate representation of the simulation to real-world data?

The question consists of different parts, which were answered subsequently. First, an extensive literature review identified that the conflict areas and their critical gap and follow-up time parameters are the most important in determining the performance of a turbo-roundabout capacity model. To a lesser extent, the reduced speed areas and car-following behaviour also impact the capacity. Table 6.1 provides an overview of these parameters and shows the settings of those parameters in VISSIM. The second part of the research question, addressing how they can be used for model calibration, was answered by conducting a sensitivity analysis of the capacity of the turbo-roundabout for the most critical model parameters. Due to the importance of the conflict areas, only the parameters of the conflict areas were considered. Hence, the model's sensitivity to the critical gap and follow-up times was investigated. From this sensitivity analysis, it was concluded that a 10% change in follow-up time and critical gap time has a similar impact for the left merging lane, but a different impact for the right merging lane. For the left lane, the 10% reduction leads to average merging capacity increases of 8.26 and 8.36%.

For the right lane, decreasing the follow-up times by 10% leads to a 10.42% increase in merging capacity, whereas it only increases by 1.73% through a 10% decrease in critical gap. Following this result, the calibration process was carried out by only adjusting the follow-up time values of the merging sections.

Table 6.1: Overview of most important parameters for turbo-roundabout capacity and their settings in the calibrated HDV model

Parameter	VISSIM parameter	Lane	Setting
Critical Gap	MinGapBlockDef [s]	Minor left	2.8
		Minor right	4.1
Follow-up time	$W74bxAdd$ [-]	Minor left	2.3
		Minor right	8
	ax [m]	Minor left	1
		Minor right	2
Reduced speed areas	Speed distribution [km/h]	Minor left	30
		Minor right	30

The fit between the Nieuwerkerk data and the simulation results was improved through seven iterations. With initial RMNSEs of 0.270 and 0.408, the fit between the first simulation results and the Nieuwerkerk data was poor. After the seventh iteration, the obtained RMNSE values were 0.074 and 0.427 for the left and right lanes, respectively. For the left lane, this signifies a substantial improvement of 72.5%. One can thus conclude that the VISSIM model performs well at simulating realistic traffic behaviour at the left entrance lane of the turbo-roundabout.
Unfortunately, the same cannot be said for the right merging lane. Through the seven iterations, the RMNSE increased by 4.7%. There are a few reasons why the calibration procedure performed well for the left merging lane, but not for the right lane. First of all, a mistake was made with the used Nieuwerkerk data. As mentioned previously, the Nieuwerkerk for the right lane also includes measurements that were not made at capacity. As this was noticed too late, these measurements were also included in the research of this thesis. A first implication of this mistake is that these data points were used for the construction of the regression lines that would later on be used in the calibration procedure. Hence, in the calibration, it was attempted to approach an ultimately inaccurate regression line.
A second reason for this bad performance is the applied strategy in the calibration for the right lane. As soon as it was noticed that changes in the critical gap and follow-up times did not impact the merging traffic capacity at high circulating traffic volumes and only had an impact at lower circulating traffic volumes, the focus shifted towards achieving the best possible fit at low circulating traffic intensities. While the calibration succeeded to achieve a difference of only 1.42% between the Nieuwerkerk capacity curve and the simulation results, this decision resulted in even worse capacity curves.
Combining this with inaccurate regression lines for the Nieuwerkerk data, the calibration procedure for the right lane resulted in highly inaccurate and unrealistic capacity curves that clashed with traffic flow theory.

6.2. Obtainable capacity increase of a turbo-roundabout under fully automated driving conditions

The main research question of this report was formulated as follows:

What capacity increase can be achieved by AVs on turbo-roundabouts through microscopic simulation, only considering automated vehicle car-following- and merging behaviour?

This question was answered by building a model to simulate AVs on the turbo-roundabout based on the calibrated HDV model. Some elements in the model had to be changed to change the behaviour of vehicles in the calibrated HDV model to that of AVs. First, the critical gap times of the merging lanes were changed to values found in scientific literature. Furthermore, the car-following behaviour of Wiedemann 74 was replaced by three car-following models based on research of the European CoExist project, each representing a different level of AV aggression. Whereas the follow-up times were manually modelled in the calibrated HDV model, this was not done for the AV models as no scientific papers described the impact of the varying aggression levels on the follow-up times. Rather, the follow-up times were determined by the three VISSIM car-following models. A different model was constructed for each car-following behaviour, and for each model, the increased capacity compared to the HDV model was calculated. For the left lane, a comparison was made with the theoretically obtained capacity increase of Fortuijn and Salomons. Table 6.2 overviews the expected and simulated capacity increases for the left lane. Improvements of 7.47%, 38.62%, and 24.32% were obtained for the cautious, normal, and aggressive AV model. Comparing this to the theoretically obtained value of 58%, one can conclude that the microscopic capacity models in this thesis resulted in a lower capacity increase than that of the theoretical model.

Table 6.2: Overview of the obtained capacity increases for the three AV models, compared with theoretically obtained capacity increase of Salomons and Fortuijn.

	AV cautious	AV normal	AV aggressive
Expected capacity increase [%]		58	
Simulated capacity increase [%]	7.47	38.62	24.32

The difference in capacity increase for the simulation model and the theoretical model of Fortuijn and Salomons aligns with expectations for a number of reasons.
First, the theoretical model assumes communication between vehicles, whereas the simulation model does not. The communication allows for lower follow-up times as vehicles have more information about other vehicles. Using the standard AV car-following models of VISSIM, the follow-up times of all three simulation models were indeed shorter than the 1.17s of Fortuijn and Salomons. These lower follow-up times create a larger merging traffic capacity.
Furthermore, microscopic traffic simulation models feature more interactions than theoretical models and consider more parameters. Therefore, its obtained capacities are likely lower than those of a theoretical model.
Next, the regression lines of the Nieuwerkerk data and the three AV models are not perfect approximations of the actual data. The 'high' spread of the individual Nieuwerkerk data points makes it impossible to draw accurate regression lines that show the actual trend in the data. This implies that the obtained capacity increases are not 100% accurate and could vary slightly.

This is an inherent property of capacity measurements at turbo-roundabouts, as measurements can only be made over short time frames. For the Nieuwerkerk data, the measurement period was five minutes, whereas traffic capacity is expressed in hours. To get hourly data, the Nieuwerkerk data is multiplied by 12, enlarging the differences between individual measurements. Small differences between the five-minute measurements now appear far more substantial. Unfortunately, collecting data over an hour is impossible, as traffic on the access branches would have to be congested during this entire period. Having a dataset with very little variability between individual data points is thus highly unlikely.

For the right lane, the expected capacity increases were not calculated as the final capacity curve of the calibrated model was unrealistic. However, it should be noted that the mistake in the calibration procedure for the right lane did not influence the results of the left lane in any way. The impact of the mistake is thus limited.

The capacity increases found in this thesis do not show the full potential of automated vehicles. Communication between automated vehicles needs to be modelled to unlock the full potential of automated vehicles. This would enable gap synchronisation, headway optimisation, and routing guidance, which has the potential to generate a greater capacity increase than those found in this report. However, VISSIM cannot simulate communication between AVs and thus not offers the ability to investigate the capacity increase of CAVs on turbo-roundabouts. An external mathematical software program must be coupled to the COM interface to model gap synchronisation and headway optimisation. While this allows one to simulate CAVs on turbo-roundabouts in VISSIM, it is worthwhile to explore other potential software programs for modelling CAVs on turbo-roundabouts.

6.3. Limitations of this thesis

6.3.1. General simplifications

The general limitations of this research contain some of the simplifications of the models in this report and some critical thoughts about the results.
The first simplification applied in the model is disregarding the lateral behaviour of vehicles. Since it is unclear how VISSIM simulates lateral behaviour, it was decided not to consider the lateral behaviour of vehicles. A second simplification is that the sham conflict is not considered in the simulations in this thesis, whereas this might have a large impact on the merging behaviour at turbo-roundabouts.
Furthermore, the impact of the speed distributions was not considered. For the reduced speed areas, a speed distribution of 30 km/h was set, which is, in fact, a distribution between 30 and 35 km/h with a linear probability density function. This implies that 0% of vehicles were driving 30 km/h, and half of the vehicles drove faster than 32.5 km/h. For vehicles on the access branches, the desired speed distribution of 80 km/h resulted in only 5% of vehicles driving 80 km/h, while 20% even had a speed greater than 90 km/h. If one had considered these speed distributions more carefully, the obtained merging traffic capacities would likely have differed. In the Wiedemann 74 model, vehicle speed has a high influence on the effective merging capacity if the speeds are low.

A last comment is that the results from this simulation do not take human preferences/demands into account. Even for automated vehicles, the impact of human perception on safety plays a role. As Dijkhuizen suggested, the gap acceptance of individuals inside an automated vehicle should also be considered (Dijkhuizen, 2020). At very small gaps they might feel unsafe if the gaps become too small. This thesis disregards this effect, which could result in minimal merging gap limits in the future.

6.3.2. Calibration of HDV model

The calibration procedure carried out in this thesis was sub-optimal, leaving some room for improvement. While the calibration procedure proved fruitful for the left merging lane, it was sub-optimal for the right lane. The poor performance for the right merging lane has a few causes.
The main cause for the poor performance is the wrong usage of Nieuwerkerk data for the right lane. Because only 19 of the 41 data points were collected under congested traffic conditions on the right lane, the regression curves for the Nieuwerkerk data are inaccurate. Instead, the regression line should have been drawn only for those 19 points. Adding to this, regression lines of data clouds are never a perfect approximation of the data and should therefore be handled with care. The standard deviations for the regression lines of both lanes indicate that the regression lines are no perfect approximation of the Nieuwerkerk data. As earlier stated, this is an inherent property of the data.
Furthermore, the model was calibrated only using the most important VISSIM elements, the conflict areas. The literature review showed that there are also other elements, such as the reduced speed areas and car-following models, that impact the turbo-roundabout capacity in VISSIM. Including these in the calibration procedure could potentially create a better calibration procedure.

6.3.3. HDV model

In the HDV model, the follow-up times were set by applying specific car-following behaviours to the conflict areas. Based on measurements under traffic conditions with no circulating traffic, the parameters were fine-tuned to achieve the desired follow-up times. However, the effect of an increasing circulating traffic volume on the actual follow-up times in the model was not checked. It is thus unknown whether the actual simulation follow-up times were the same as those set beforehand.

6.3.4. AV models

The first limitation of the AV models developed in this thesis is how the car-following models were used. Unlike the calibrated HDV model, where separate car-following models were applied to the conflict areas to reach a desirable follow-up time, the follow-up times in the AV simulations were not specified through specific car-following behaviours applied in the conflict areas. Rather, they were left to the cautious, normal, and aggressive AV car-following models. After the simulations had been carried out, the follow-up times at the conflict areas were obtained by analysing the merging traffic flows under congested conditions without circulating traffic on the turbo-roundabout. The outcomes showed that the follow-up times of all AV models were higher than those found in the literature, implying that the obtained capacity of the three AV models, and hence also the capacity increase, could be improved. Table 6.3 provides an overview of the follow-up times for the three AV models and compares them to the theoretical follow-up time determined by Salomons and Fortuijn. While it is logical that the follow up-times of the theoretical model featuring communication between vehicles are lower, it is unknown whether the simulated follow-up times are realistic.
This report also did not consider changing the speed for automated vehicles, as compared to the calibrated HDV model. In reality, one could predict that the speeds of automated vehicles can be higher because they have more information about the vehicles surrounding them and, therefore, can drive more proactively.

Table 6.3: Comparison of the follow-up times [s] of the three AV models compared to those of Fortuijn and Salomons

Lane	Theoretical	AV cautious	AV normal	AV aggressive
Minor left	1.17	2.16	1.59	1.33
Minor right	1.17	2.14	1.56	1.34

6.3.5. Limitations of VISSIM

During this thesis, it was discovered that VISSIM might not be the optimal way to simulate automated vehicles on turbo-roundabouts. First, VISSIM has an expansive list of parameters, functions, and settings, making it highly cluttered. While the user manual has almost 1400 pages, how some parameters and settings impact the simulation is still unclear. Since it is impossible to fine-tune all parameters, the simulation outcomes of a VISSIM model should be handled with care. This thesis only considered the most important parameters of VISSIM for turbo-roundabout capacity, as found in the literature. Even then, it takes a lot of effort to construct a proper model.
An example of a confusing setting in VISSIM is the lateral behaviour of vehicles. For this thesis, the lateral behaviour of vehicles is not highly impactful, but for models utilising gap synchronisation, this lateral behaviour is much more important. Since it is unknown how VISSIM models the lateral behaviour of vehicles in its models, the impact of lateral behaviour on turbo-roundabout capacity cannot be determined. The only parameter that describes the lateral behaviour is in what position of the lane a vehicle should drive, but how much they deviate from this location is unknown.
Another major limitation of VISSIM is that it cannot correctly model connected and automated vehicles without the interference of an external mathematical controller. It cannot simulate gap synchronisation and headway optimisation, the two most important features to unlock the full potential of CAVs.

Recommending future researchers not to use VISSIM for their research is a bridge too far, as the results of this thesis still leave room for improvement. However, the results of this thesis serve as a first indication that VISSIM is not an ideal tool to simulate the behaviour of vehicles on turbo-roundabouts. Due to the large amount of settings, one can easily get lost in the details without adequately understanding the impact of individual parameters and how these interplay. Furthermore, since VISSIM does not offer the possibility to model gap synchronisation, headway optimisation, and path guidance, it cannot simulate automated vehicles with all their features. Besides changing the car-following behaviour, critical gap times, and platooning behaviour, VISSIM must be coupled with an external controller through the COM interface to simulate CAVs properly. VISSIM needs to be improved to simulate realistic (C)AV behaviour.

6.4. Recommendations for future research

This section elaborates on the recommendations for future research. They are grouped according to their overarching topic, some following from the limitations of this thesis. A general recommendation for further research is to make a simulation model featuring traffic from all roundabout branches. In this thesis, traffic was only generated on the minor roundabout branch of interest, and the first upstream branch. The following recommendations relate to specific aspects of this thesis.

6.4.1. Calibration procedure

As mentioned in Subsection 6.3.2, the calibration procedure in this thesis has its flaws. Therefore, carrying out the calibration procedure once more but only using data collected under congestion is recommended. However, using the Nieuwerkerk dataset leaves only 19 data points for the right access lane. Therefore, it is recommended that the calibration procedure is done with a different dataset that features more data points.
Furthermore, the calibration procedure should also be done for the major branch of a turbo-roundabout. Afterwards, this can be combined with the calibrated minor branch to investigate whether a microscopic traffic simulation model can perform well on all access branches of a turbo-roundabout simultaneously.

6.4.2. HDV model

This thesis does not consider the sham conflict. Fortuijn describes that the sham conflict is often not considered in theoretical capacity models but can impact the merging capacity at turbo-roundabouts (Fortuijn, 2009). He states this could lead to a 10% reduction in merging traffic capacity. Future research should thus be conducted to investigate what impact the sham conflict has on the merging capacity in microscopic traffic simulation models. The sham conflict can be modelled at the conflict areas in VISSIM but was not modelled in this thesis. It only applies to human-driven vehicles, as automated vehicles can inform other vehicles about their route, eliminating the sham conflict.

6.4.3. AV models

In the simulations in this thesis, the critical gaps of the three AV models with different aggression levels were the same. However, one could argue that more aggressive vehicles would likely accept a lower critical gap in the circulating traffic flow than those of less aggressive ones. No literature could be found describing the different acceptable critical gaps for various aggression levels of AVs, thus leaving an interesting research gap to be explored.

A second recommendation of future research entails the comment in Chapter 5 about the number of interaction objects and vehicles impacting the performance of the aggressive AV model for the left merging lane of the turbo-roundabout. As stated, the normal AV model unexpectedly outperforms the aggressive AV model, and a likely explanation for this result is the difference between the number of interaction vehicles/objects of the aggressive and normal AV models. In 'reality', the aggressive AV model should outperform the normal AV model since its follow-up times are lower and vehicles have more information about other vehicles in the model.
However, the fact that aggressive vehicles possess more information about other vehicles leads to a lower performance for aggressive AV vehicles on a turbo-roundabout. An interesting research opportunity is to determine the right way to model the aggressive AV model in VISSIM to outperform the normal AV model while ensuring it remains more knowledgeable about other vehicles in the network than the normal AV model. Does an aggressive AV need to observe multiple upstream vehicles per circulatory lane?

A third, and perhaps the most interesting recommendation for future research, is the simulation of the capacity of a turbo-roundabout where the full potential of automated vehicles is unlocked. As mentioned, VISSIM does not offer the possibility to incorporate gap synchronisation and headway optimisation, which creates the most considerable capacity increase for automated vehicles compared to HDVs. It can only simulate automated vehicles through changed critical gap times, car-following behaviours and platooning. Adding gap synchronisation, headway optimisation and routing guidance was the initial goal of this thesis, but the scope was adjusted due to the complexity of modelling it through the COM interface. Future research should focus on modelling this behaviour using the COM interface in VISSIM or any other microscopic traffic simulation model. The results of such research can then be compared to the theoretical AVG-T4 model of Fortuijn and Salomons.

Other potential interesting research opportunities lie in simulating a turbo-roundabout in traffic conditions with automated cars and other vehicles. None of the papers in the literature review of this thesis considers vehicles such as trucks, buses, and semi-trucks. However, these vehicles will still be part of the future transportation network, making it especially interesting to investigate what would happen with the turbo-roundabout capacity if other vehicle types were also in the network.
Furthermore, the vehicle speeds of automated vehicles are also interesting. In this thesis, the HDV and AV models use the identical speed distributions, but is it possible to allow higher speeds for automated vehicles possessing all the information of other automated vehicles in the network?

Bibliography

Aisum. (2024). *Digital mobility solutions*. https://www.aimsun.com/

Amouzadi, M., Olawumi Orisatoki, M., & M. Dizqah, A. (2022). Capacity analysis of intersections when cavs are crossing in a collaborativeand lane-free order. *Future Transportation*, *2*, 698–710. https://doi.org/https://doi.org/10.3390/futuretransp2030039

Arroju, R., Gaddam, H. K., Vanumu, L. D., & Rao, K. R. (2015). Comparative evaluation of roundabout capacities under heterogeneous traffic conditions. *Journal of Modern Transport*, *23*(4), 310–324. https://doi.org/10.1007/s40534-015-0089-8

Bakibillah, A. S. M., Kamal, M. A. S., Tan, C. P., Susilawati, S., Hayakawa, T., & Imura, J.-i. (2021). Bi-level coordinated merging of connected and automated vehicles at roundabouts. *Sensors*, *21*(19). https://doi.org/10.3390/s21196533

Bakibillah, A., Kamal, M., Susilawati, S., & Tan, C. (2019). The optimal coordination of connected and automated vehicles at roundabouts. *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 1392–1397. https://doi.org/10.23919/SICE.2019.8860072

Bakibillah, A., Kamal, M., Susilawati, & Tan, C. (2019). The optimal coordination of connected and automated vehicles at roundabouts. *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 1392–1397. https://doi.org/10.23919/SICE.2019.8860072

Benekohal, R. F. (n.d.). Procedure for validation of microscopic traffic flow simulation models. *Transportation Research Record*. https://onlinepubs.trb.org/Onlinepubs/trr/1991/1320/1320-023.pdf

Bondzio, L. (n.d.). Experiences with turbo-roundabouts in germany.

Boualam, O., Borsos, A., Koren, C., & Nagy, V. (2022). Impact of autonomous vehicles on roundabout capacity. *Sustainability*, *14*, 2203. https://doi.org/10.3390/su14042203

Danesh, A., Ma, W., Yu, C., Hao, R., & Ma, X. (2021). Optimal roundabout control under fully connected and automated vehicle environment. *IET Intelligent Transport Systems*, *15*(11), 1440–1453. https://doi.org/https://doi.org/10.1049/itr2.12117

De Baan, D. (2023). Turborotondes: Soorten, aantal en waar te vinden? https://www.dirkdebaan.nl/turborotondes.html

Dijkhuizen, S. (2020). Safety margin development for a space-time reservation traffic control scheme. https://repository.tudelft.nl/islandora/object/uuid:0ebac80a-d005-4c17-9528-f62c303e6ac6?collection=education

Fortuijn, L. (2009). Turbo roundabouts: Estimation of capacity. *Transportation Research Record: Journal of the Transportation Research Board*, (2130), 83–92. https://doi.org/10.3141/2130-11

Fortuijn, L. (2013). Turborotonde en turboplein: ontwerp, capaciteit en veiligheid, 46, 86.

Fortuijn, L. (2018). Vergelijking methoden ter bepaling van doorrijsnelheden op rotondes.

Fortuijn, L., & Salomons, A. (2020). Capacity increase through connectivity for the i-roundabout and i-turbo roundabout. *2020 Forum on Integrated and Sustainable Transportation Systems (FISTS), 94*, 19–31. https://doi.org/10.1109/FISTS46898.2020.9264897

Gallelli, V., Iuele, T., & Vaiana, R. (2016). Coversion of a semi-two lanes roundabout into a turbo-roundabout: A performance comparison. *Procedia Computer Science, 83*, 393–400. https://doi.org/1016/j-procs.2016.04.201

Gallelli, V., Iuele, T., Vaiana, R., & Vitale, A. (2017). Investigating the transferability of calibrated microsimulation parameters for operational performance analysis in roundabouts. *Journal of Advanced Transportation, 2017*. https://doi.org/10.1155/2017/3078063

Giuffrè, T., Granà, A., & Trubia, S. (2021). Safety evaluation of turbo-roundabouts with and without internal traffic separations considering autonomous vehicles operation. *Suistainability, 13*(16). https://doi.org/10.3390/su13168810

Goodall, N. J., Smith, B. L., & Park, B. ((2013). Traffic signal control with connected vehicles. *Transportation Research Record, 2381*(1), 65–72. https://doi.org/10.3141/2381-08

Group, P. (2021). *Multimodal traffic simulation software*. https://www.ptvgroup.com/en/products/ptv-vissim

Guerrieri, M., Corriere, F., & Ticali, D. (2012). Turbo-roundabouts: A model to evaluate capacity, delays, queues and level of service. *European Journal of Scientific Research, 92*, 67–282. https://doi.org/https://doi.org/10.1016/j.trc.2017.12.006

Guerrieri, M., & Granà, A. (2009). Evaluating capacity and efficiency of turbo-roundabouts.

Guerrieri, M., Mauro, R., Parla, G., & Tollazzi, T. (2018). Analysis of kinematic parameters and driver behavior at turbo roundabouts. *Journal of Transportation Engineering, Part A: Systems, 144*(6). https://doi.org/https://doi.org/10.1061/JTEPBS.0000129

in Transportation Engineering, A. L. (n.d.). Adjusting driver behavior. https://www.activetbooks.com/adjusting-driving-behavior#:~:text=Wiedemann%5C%2074%5C%20is%5C%20suitable%5C%20for,some%5C%20or%5C%20all%5C%20links%5C%2Fconnectors.

LaFrance, A. (2016). Your grandmother's driverless car. *The Atlantic*. https://www.theatlantic.com/technology/archive/2016/06/beep-beep/489029/

Lawson, S. (2018). *New report tackles the transition to automated vehicles on roads that cars can read*. https://irap.org/2018/06/new-report-tackles-the-transition-to-automated-vehicles-on-roads-that-cars-can-read/

Lefkowitz, M. (2019). Smart intersections could reduce autonomous car congestion. *Cornell Chronicle*. https://news.cornell.edu/stories/2019/12/smart-intersections-could-reduce-autonomous-car-congestion#:~:text=The%5C%20researchers'%5C%20model%5C%20allows%5C%20groups,systems%5C%2C%5C%20according%5C%20to%5C%20the%5C%20study

Li, Z., DeAmico, M., Chitturi, M., Bill, A. R., & Noyce, D. A. (2013). Calibrating vissim roundabout model using a critical gap and follow-up headway approach. *16h Road Safety on Four Continents Conference*.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wießner, E. (2018). Microscopic traffic simulation using sumo. *The 21st IEEE International Conference on Intelligent Transportation Systems*. https://elib.dlr.de/124092/

Martin-Gasulla, M., & Elefteriadou, L. (2021). Traffic management with autonomous and connected vehicles at single-lane roundabouts. *Transportation Research Part C: Emerging Technologies, 125*, 102964. https://doi.org/https://doi.org/10.1016/j.trc.2021.102964

Mohebifard, R., & Hajbabaie, A. (2020). Effects of automated vehicles on traffic operations at roundabouts. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 1–6. https://doi.org/10.1109/ITSC45102.2020.9294563

Mohebifard, R., & Hajbabaie, A. (2021). Connected automated vehicle control in single lane roundabouts. *Transportation Research Part C: Emerging Technologies, 131*, 103308. https://doi.org/https://doi.org/10.1016/j.trc.2021.103308

n.a. (2024). *What is coexist?*

OpenTrafficSim. (2024). *What is opentrafficsim?* https://opentrafficsim.org/manual/

Otkovic, I. I., Tollazzi, T., & Sraml, M. (2013). Calibration of microsimulation traffic model using neural network approach. *Expert Systems with Applications, 40*(15), 5965–5974. https://doi.org/10.1016/j-eswa.2013.05.003

Rodrigues, M., McGordon, A., Gest, G., & Marco, J. (2018). Autonomous navigation in interaction-based environments—a case of non-signalized roundabouts. *IEEE Transactions on Intelligent Vehicles, 3*(4), 425–438. https://doi.org/10.1109/TIV.2018.2873916

Sadler, K. (2016). Plans to expand driverless parkshuttle in the netherlands revealed. *Intelligent Transport*. https://www.intelligenttransport.com/transport-news/21142/driverless-parkshuttle-netherlands/

SAE. (2021). Sae levels of driving automation refined for clarity and international audience. *Society of Automotive Engineers*. https://www.sae.org/blog/sae-j3016-update

Severino, A., Pappalard, G., & Trubia, S. (2021). Safety evaluation of turbo roundabout considering autonomous vehicles operation. *Warsaw University of Technology, 67*(1). https://doi.org/10.3390/su13168810

Sukennik, P. (2020). Micro-simulation guide for autonomous vehicles. *CoExist*.

Sun, W., Zheng, J., & Liu, H. X. (2018). A capacity maximization scheme for intersection management with automated vehicles. *Transportation Research Part C: Emerging Technologies, 94*, 19–31. https://doi.org/https://doi.org/10.1016/j.trc.2017.12.006

SWOV. (2022). *Roundabouts and other intersections: Swov fact sheet*.

Synopsis. (n.d.). The 6 levels of vehicle automation explained. https://www.synopsys.com/automotive/autonomous-driving-levels.html

Toledo, T., Koutsopoulos, H. N., Davol, A., Ben-Akiva, M. E., Burghout, W., Adreasson, I., Johansson, T., & Lundin, C. (2003). Calibration and validation of microscopic traffic simulation tools: Stockholm case study. *Transportation Research Record Journal of the Transportation Research Board, 1831*(3-2189), 65–75. https://doi.org/10.3141/1831-08

Tumminello, M. L., Macioszek, E., Granà, A., & Giuffrè, T. (2022). Simulation-based analysis of "what-if" scenarios with connected and automated vehicles navigating roundabouts. *Sensors, 22*(17), 6670. https://doi.org/10.3390/s22176670

Vaiana, R., Gallelli, V., & Iuele, T. (2013). Sensitivity analysis in traffic microscopic simulation model for roundabouts. *The Baltic Journal of Road and Bridge Engineering, 8*, 174–183. https://doi.org/10.3846/bjrbe.2013.22

Weber, M. (2014). Where to? a history of autonomous vehicles. *Computer History Museum*. https://computerhistory.org/blog/where-to-a-history-of-autonomous-vehicles/?key=where-to-a-history-of-autonomous-vehicles

Zeidler, V., Buck, S., Kautzsch, L., Vortisch, P., & Weyland, C. (2019). Simulation of autonomous vehicles based on wiedemann's car following model in ptv vissim.

Zhao, L., Malikopoulos, A., & Rios-Torres, J. (2018). Optimal control of connected and automated vehicles at roundabouts: An investigation in a mixed-traffic environment. *IFAC-PapersOnLine*, *51*(9), 73–78. https://doi.org/https://doi.org/10.1016/j.ifacol.2018.07.013

A

Python Code of the data processing and the thereby obtained capacity curves

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import scipy as sc
from scipy.optimize import curve_fit
%clear
```

In [2]:
```
x = pd.read_excel("VISSIM east vorige tak.xlsx");
x.head
tijd = np.array(x['Tijd begin'])
tijd_comp = np.array(x['Tijd'])
index = np.array(x['Index'])
iteratie = np.array(x['Iteratie'])
verkeer = np.array(x['Verkeer'])
verkeercrit = np.array(x['Verkeercrit-10'])
verkeerfol = np.array(x['VerkeerFollowUp-10'])
verkeerit1 = np.array(x['It1'])
verkeerit2 = np.array(x['It2'])
verkeerit3 = np.array(x['It3'])
verkeerit4 = np.array(x['It4'])
verkeerit5 = np.array(x['It5'])
verkeerit6 = np.array(x['It6'])
verkeerit7 = np.array(x['It7'])
verkeerval = np.array(x['Val'])

val = pd.read_excel("East validated.xlsx")
tijd_val = np.array(val['Tijd begin'])
tijd_comp_val = np.array(val['Tijd'])
index_val = np.array(val['Index'])
iteratie_val = np.array(val['Iteratie'])
verkeer_val = np.array(val['Val'])

cau = pd.read_excel("East av cautious.xlsx")
tijd_cau = np.array(cau['Tijd begin'])
tijd_comp_cau = np.array(cau['Tijd'])
index_cau = np.array(cau['Index'])
iteratie_cau = np.array(cau['Iteratie'])
verkeer_cau = np.array(cau['cautious'])

norm = pd.read_excel("East av normal.xlsx")
tijd_norm = np.array(norm['Tijd begin'])
tijd_comp_norm = np.array(norm['Tijd'])
index_norm = np.array(norm['Index'])
iteratie_norm = np.array(norm['Iteratie'])
verkeer_norm = np.array(norm['Normal'])

agg = pd.read_excel("East av aggressive.xlsx")
tijd_agg = np.array(agg['Tijd begin'])
tijd_comp_agg = np.array(agg['Tijd'])
index_agg = np.array(agg['Index'])
iteratie_agg = np.array(agg['Iteratie'])
verkeer_agg = np.array(agg['aggressive'])
```

In [3]:
```
#Base case simulation
Left = np.zeros((5,51))
Right = np.zeros((5,51))
In = np.zeros((5,51))
Out = np.zeros((5,51))

#Critical gap 10% lower
Left2 = np.zeros((5,51))
Right2 = np.zeros((5,51))
In2 = np.zeros((5,51))
Out2 = np.zeros((5,51))

#Follow-up time 10% lower
Left3 = np.zeros((5,51))
Right3 = np.zeros((5,51))
In3 = np.zeros((5,51))
Out3 = np.zeros((5,51))

#Iteration 1
Left_it1 = np.zeros((5,51))
Right_it1 = np.zeros((5,51))
In_it1 = np.zeros((5,51))
Out_it1 = np.zeros((5,51))

#Iteration 2
Left_it2 = np.zeros((5,51))
Right_it2 = np.zeros((5,51))
In_it2 = np.zeros((5,51))
Out_it2 = np.zeros((5,51))

#Iteration 3
Left_it3 = np.zeros((5,51))
Right_it3 = np.zeros((5,51))
In_it3 = np.zeros((5,51))
Out_it3 = np.zeros((5,51))

#Iteration 4
Left_it4 = np.zeros((5,51))
Right_it4 = np.zeros((5,51))
In_it4 = np.zeros((5,51))
Out_it4 = np.zeros((5,51))

#Iteration 5
Left_it5 = np.zeros((5,51))
Right_it5 = np.zeros((5,51))
In_it5 = np.zeros((5,51))
Out_it5 = np.zeros((5,51))

#Iteration 6
Left_it6 = np.zeros((5,51))
Right_it6 = np.zeros((5,51))
In_it6 = np.zeros((5,51))
Out_it6 = np.zeros((5,51))

#Iteration 7
Left_it7 = np.zeros((5,51))
Right_it7 = np.zeros((5,51))
In_it7 = np.zeros((5,51))
Out_it7 = np.zeros((5,51))

#Validation
Leftval = np.zeros((5,51))
Rightval = np.zeros((5,51))
Inval = np.zeros((5,51))
Outval = np.zeros((5,51))

#Validated model
Left_val = np.zeros((5,61))
Right_val = np.zeros((5,61))
In_val = np.zeros((5,61))
Out_val = np.zeros((5,61))

#Av cautious
Left_av_caut = np.zeros((5,65))
Right_av_caut = np.zeros((5,65))
In_av_caut = np.zeros((5,65))
Out_av_caut = np.zeros((5,65))

#AV normal
Left_av_norm = np.zeros((5,81))
Right_av_norm = np.zeros((5,81))
In_av_norm = np.zeros((5,81))
Out_av_norm = np.zeros((5,81))

#AV aggressive
Left_av_agg = np.zeros((5,92))
Right_av_agg = np.zeros((5,92))
In_av_agg = np.zeros((5,92))
Out_av_agg = np.zeros((5,92))
```

In [4]:
```
#Base case simulation
for i in range(len(tijd)):
    for j in range(51):
        for k in range(5):
            if iteratie[i] == k+1:
                    if tijd[i] == tijd_comp[j] and index[i] == 2:
                        Left[k,j] = verkeer[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 5:
                        Right[k,j] = verkeer[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 4:
                        In[k,j] = verkeer[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 3:
                        Out[k,j] = verkeer[i]
                    else:
                        pass
```

In [5]:
```
#Critical gap 10% lower
for i in range(len(tijd)):
    for j in range(51):
        for k in range(5):
            if iteratie[i] == k+1:
                    if tijd[i] == tijd_comp[j] and index[i] == 2:
                        Left2[k,j] = verkeercrit[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 5:
                        Right2[k,j] = verkeercrit[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 4:
                        In2[k,j] = verkeercrit[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 3:
                        Out2[k,j] = verkeercrit[i]
                    else:
                        pass
```

In [6]:
```
#Follow-up time 10% lower
for i in range(len(tijd)):
    for j in range(51):
        for k in range(5):
            if iteratie[i] == k+1:
                    if tijd[i] == tijd_comp[j] and index[i] == 2:
                        Left3[k,j] = verkeerfol[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 5:
                        Right3[k,j] = verkeerfol[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 4:
                        In3[k,j] = verkeerfol[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 3:
                        Out3[k,j] = verkeerfol[i]
                    else:
                        pass
```

In [7]:
```
#Iteration 1
for i in range(len(tijd)):
    for j in range(51):
        for k in range(5):
            if iteratie[i] == k+1:
                    if tijd[i] == tijd_comp[j] and index[i] == 2:
                        Left_it1[k,j] = verkeerit1[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 5:
                        Right_it1[k,j] = verkeerit1[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 4:
                        In_it1[k,j] = verkeerit1[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 3:
                        Out_it1[k,j] = verkeerit1[i]
                    else:
                        pass
```

In [8]:
```
#Iteration 2
for i in range(len(tijd)):
    for j in range(51):
        for k in range(5):
            if iteratie[i] == k+1:
                    if tijd[i] == tijd_comp[j] and index[i] == 2:
                        Left_it2[k,j] = verkeerit2[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 5:
                        Right_it2[k,j] = verkeerit2[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 4:
                        In_it2[k,j] = verkeerit2[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 3:
                        Out_it2[k,j] = verkeerit2[i]
                    else:
                        pass
```

In [9]:
```
#Iteration 3
for i in range(len(tijd)):
    for j in range(51):
        for k in range(5):
            if iteratie[i] == k+1:
                    if tijd[i] == tijd_comp[j] and index[i] == 2:
                        Left_it3[k,j] = verkeerit3[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 5:
                        Right_it3[k,j] = verkeerit3[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 4:
                        In_it3[k,j] = verkeerit3[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 3:
                        Out_it3[k,j] = verkeerit3[i]
                    else:
                        pass
```

In [10]:
```
#Iteration 4
for i in range(len(tijd)):
    for j in range(51):
        for k in range(5):
            if iteratie[i] == k+1:
                    if tijd[i] == tijd_comp[j] and index[i] == 2:
                        Left_it4[k,j] = verkeerit4[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 5:
                        Right_it4[k,j] = verkeerit4[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 4:
                        In_it4[k,j] = verkeerit4[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 3:
                        Out_it4[k,j] = verkeerit4[i]
                    else:
                        pass
```

In [11]:
```
#Iteration 5
for i in range(len(tijd)):
    for j in range(51):
        for k in range(5):
            if iteratie[i] == k+1:
                    if tijd[i] == tijd_comp[j] and index[i] == 2:
                        Left_it5[k,j] = verkeerit5[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 5:
                        Right_it5[k,j] = verkeerit5[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 4:
                        In_it5[k,j] = verkeerit5[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 3:
                        Out_it5[k,j] = verkeerit5[i]
                    else:
                        pass
```

In [12]:
```
#Iteration 6
for i in range(len(tijd)):
    for j in range(51):
        for k in range(5):
            if iteratie[i] == k+1:
                    if tijd[i] == tijd_comp[j] and index[i] == 2:
                        Left_it6[k,j] = verkeerit6[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 5:
                        Right_it6[k,j] = verkeerit6[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 4:
                        In_it6[k,j] = verkeerit6[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 3:
                        Out_it6[k,j] = verkeerit6[i]
                    else:
                        pass
```

In [13]:
```
#Iteration 7
for i in range(len(tijd)):
    for j in range(51):
        for k in range(5):
            if iteratie[i] == k+1:
                    if tijd[i] == tijd_comp[j] and index[i] == 2:
                        Left_it7[k,j] = verkeerit7[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 5:
                        Right_it7[k,j] = verkeerit7[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 4:
                        In_it7[k,j] = verkeerit7[i]
                    elif tijd[i] == tijd_comp[j] and index[i] == 3:
                        Out_it7[k,j] = verkeerit7[i]
                    else:
                        pass
```

In [14]:
```
#Validated model
for i in range(len(tijd_val)):
    for j in range(61):
        for k in range(5):
            if iteratie_val[i] == k+1:
                    if tijd_val[i] == tijd_comp_val[j] and index_val[i] == 2
                        Left_val[k,j] = verkeer_val[i]
                    elif tijd_val[i] == tijd_comp_val[j] and index_val[i] ==
                        Right_val[k,j] = verkeer_val[i]
                    elif tijd_val[i] == tijd_comp_val[j] and index_val[i] ==
                        In_val[k,j] = verkeer_val[i]
                    elif tijd_val[i] == tijd_comp_val[j] and index_val[i] ==
                        Out_val[k,j] = verkeer_val[i]
                    else:
                        pass
```

In [15]:
```
#AV cautious
for i in range(len(tijd_cau)):
    for j in range(65):
        for k in range(5):
            if iteratie_cau[i] == k+1:
                    if tijd_cau[i] == tijd_comp_cau[j] and index_cau[i] == 2
                        Left_av_caut[k,j] = verkeer_cau[i]
                    elif tijd_cau[i] == tijd_comp_cau[j] and index_cau[i] ==
                        Right_av_caut[k,j] = verkeer_cau[i]
                    elif tijd_cau[i] == tijd_comp_cau[j] and index_cau[i] ==
                        In_av_caut[k,j] = verkeer_cau[i]
                    elif tijd_cau[i] == tijd_comp_cau[j] and index_cau[i] ==
                        Out_av_caut[k,j] = verkeer_cau[i]
                    else:
                        pass
```

In [16]:
```
#Av normal
for i in range(len(tijd_norm)):
    for j in range(81):
        for k in range(5):
            if iteratie_norm[i] == k+1:
                    if tijd_norm[i] == tijd_comp_norm[j] and index_norm[i] =
                        Left_av_norm[k,j] = verkeer_norm[i]
                    elif tijd_norm[i] == tijd_comp_norm[j] and index_norm[i]
                        Right_av_norm[k,j] = verkeer_norm[i]
                    elif tijd_norm[i] == tijd_comp_norm[j] and index_norm[i]
                        In_av_norm[k,j] = verkeer_norm[i]
                    elif tijd_norm[i] == tijd_comp_norm[j] and index_norm[i]
                        Out_av_norm[k,j] = verkeer_norm[i]
                    else:
                        pass
```

In [17]:

```
#Av aggressive
for i in range(len(tijd_agg)):
    for j in range(92):
        for k in range(5):
            if iteratie_agg[i] == k+1:
                    if tijd_agg[i] == tijd_comp_agg[j] and index_agg[i] == 2
                        Left_av_agg[k,j] = verkeer_agg[i]
                    elif tijd_agg[i] == tijd_comp_agg[j] and index_agg[i] ==
                        Right_av_agg[k,j] = verkeer_agg[i]
                    elif tijd_agg[i] == tijd_comp_agg[j] and index_agg[i] ==
                        In_av_agg[k,j] = verkeer_agg[i]
                    elif tijd_agg[i] == tijd_comp_agg[j] and index_agg[i] ==
                        Out_av_agg[k,j] = verkeer_agg[i]
                    else:
                        pass
```

In [18]:

```
#Base case model
LeftH=Left*12
RightH=Right*12
InH=In*12
OutH=Out*12
lefth=np.array(LeftH).flatten()
righth=np.array(RightH).flatten()
inh = np.array(InH).flatten()
outh = np.array(OutH).flatten()

entering = LeftH+RightH
Entering = np.array(entering).flatten()
circulating = InH+OutH
Circulating = np.array(circulating).flatten()

#Critical gap 10% lower
LeftH2=Left2*12
RightH2=Right2*12
InH2=In2*12
OutH2=Out2*12
lefth2=np.array(LeftH2).flatten()
righth2=np.array(RightH2).flatten()
inh2 = np.array(InH2).flatten()
outh2 = np.array(OutH2).flatten()

entering2 = LeftH2+RightH2
Entering2 = np.array(entering2).flatten()
circulating2 = InH2+OutH2
Circulating2 = np.array(circulating2).flatten()

#Follow-up time 10% lower
LeftH3=Left3*12
RightH3=Right3*12
InH3=In3*12
OutH3=Out3*12
lefth3=np.array(LeftH3).flatten()
righth3=np.array(RightH3).flatten()
inh3 = np.array(InH3).flatten()
outh3 = np.array(OutH3).flatten()

entering3 = LeftH3+RightH3
Entering3 = np.array(entering3).flatten()
circulating3 = InH3+OutH3
Circulating3 = np.array(circulating3).flatten()

#Iteration 1
LeftH_it1=Left_it1*12
RightH_it1=Right_it1*12
InH_it1=In_it1*12
OutH_it1=Out_it1*12
lefth_it1=np.array(LeftH_it1).flatten()
righth_it1=np.array(RightH_it1).flatten()
inh_it1 = np.array(InH_it1).flatten()
outh_it1 = np.array(OutH_it1).flatten()

entering_it1 = LeftH_it1+RightH_it1
Entering_it1 = np.array(entering_it1).flatten()
circulating_it1 = InH_it1+OutH_it1
Circulating_it1 = np.array(circulating_it1).flatten()

#Iteration 2
LeftH_it2=Left_it2*12
RightH_it2=Right_it2*12
InH_it2=In_it2*12
OutH_it2=Out_it2*12
lefth_it2=np.array(LeftH_it2).flatten()
righth_it2=np.array(RightH_it2).flatten()
inh_it2 = np.array(InH_it2).flatten()
outh_it2 = np.array(OutH_it2).flatten()

entering_it2 = LeftH_it2+RightH_it2
Entering_it2 = np.array(entering_it2).flatten()
circulating_it2 = InH_it2+OutH_it2
Circulating_it2 = np.array(circulating_it2).flatten()

#Iteration 3
LeftH_it3=Left_it3*12
RightH_it3=Right_it3*12
InH_it3=In_it3*12
OutH_it3=Out_it3*12
lefth_it3=np.array(LeftH_it3).flatten()
righth_it3=np.array(RightH_it3).flatten()
inh_it3 = np.array(InH_it3).flatten()
outh_it3 = np.array(OutH_it3).flatten()

entering_it3 = LeftH_it3+RightH_it3
Entering_it3 = np.array(entering_it3).flatten()
circulating_it3 = InH_it3+OutH_it3
Circulating_it3 = np.array(circulating_it3).flatten()

#Iteration 4
LeftH_it4=Left_it4*12
RightH_it4=Right_it4*12
InH_it4=In_it4*12
OutH_it4=Out_it4*12
lefth_it4=np.array(LeftH_it4).flatten()
righth_it4=np.array(RightH_it4).flatten()
inh_it4 = np.array(InH_it4).flatten()
outh_it4 = np.array(OutH_it4).flatten()

entering_it4 = LeftH_it4+RightH_it4
Entering_it4 = np.array(entering_it4).flatten()
circulating_it4 = InH_it4+OutH_it4
Circulating_it4 = np.array(circulating_it4).flatten()

#Iteration 5
LeftH_it5=Left_it5*12
RightH_it5=Right_it5*12
InH_it5=In_it5*12
OutH_it5=Out_it5*12
lefth_it5=np.array(LeftH_it5).flatten()
righth_it5=np.array(RightH_it5).flatten()
inh_it5 = np.array(InH_it5).flatten()
outh_it5 = np.array(OutH_it5).flatten()

entering_it5 = LeftH_it5+RightH_it5
Entering_it5 = np.array(entering_it5).flatten()
circulating_it5 = InH_it5+OutH_it5
Circulating_it5 = np.array(circulating_it5).flatten()

#Iteration 6
LeftH_it6=Left_it6*12
```

```
RightH_it6=Right_it6*12
InH_it6=In_it6*12
OutH_it6=Out_it6*12
lefth_it6=np.array(LeftH_it6).flatten()
righth_it6=np.array(RightH_it6).flatten()
inh_it6 = np.array(InH_it6).flatten()
outh_it6 = np.array(OutH_it6).flatten()

entering_it6 = LeftH_it6+RightH_it6
Entering_it6 = np.array(entering_it6).flatten()
circulating_it6 = InH_it6+OutH_it6
Circulating_it6 = np.array(circulating_it6).flatten()

#Iteration 7
LeftH_it7=Left_it7*12
RightH_it7=Right_it7*12
InH_it7=In_it7*12
OutH_it7=Out_it7*12
lefth_it7=np.array(LeftH_it7).flatten()
righth_it7=np.array(RightH_it7).flatten()
inh_it7 = np.array(InH_it7).flatten()
outh_it7 = np.array(OutH_it7).flatten()

entering_it7 = LeftH_it7+RightH_it7
Entering_it7 = np.array(entering_it7).flatten()
circulating_it7 = InH_it7+OutH_it7
Circulating_it7 = np.array(circulating_it7).flatten()

#Validated model
LeftH_val=Left_val*12
RightH_val=Right_val*12
InH_val=In_val*12
OutH_val=Out_val*12
lefth_val=np.array(LeftH_val).flatten()
righth_val=np.array(RightH_val).flatten()
inh_val = np.array(InH_val).flatten()
outh_val = np.array(OutH_val).flatten()

entering_val = LeftH_val+RightH_val
Entering_val = np.array(entering_val).flatten()
circulating_val = InH_val+OutH_val
Circulating_val = np.array(circulating_val).flatten()

#Av cautious model
LeftH_av_caut=Left_av_caut*12
RightH_av_caut=Right_av_caut*12
InH_av_caut=In_av_caut*12
OutH_av_caut=Out_av_caut*12
lefth_av_caut=np.array(LeftH_av_caut).flatten('C')
righth_av_caut=np.array(RightH_av_caut).flatten()
inh_av_caut = np.array(InH_av_caut).flatten()
outh_av_caut = np.array(OutH_av_caut).flatten()

entering_av_caut = LeftH_av_caut+RightH_av_caut
Entering_av_caut = np.array(entering_av_caut).flatten()
circulating_av_caut = InH_av_caut+OutH_av_caut
Circulating_av_caut = np.array(circulating_av_caut).flatten()

#AV normal
LeftH_av_norm=Left_av_norm*12
RightH_av_norm=Right_av_norm*12
InH_av_norm=In_av_norm*12
OutH_av_norm=Out_av_norm*12
lefth_av_norm=np.array(LeftH_av_norm).flatten()
righth_av_norm=np.array(RightH_av_norm).flatten()
inh_av_norm = np.array(InH_av_norm).flatten()
outh_av_norm = np.array(OutH_av_norm).flatten()

entering_av_norm = LeftH_av_norm+RightH_av_norm
Entering_av_norm = np.array(entering_av_norm).flatten()
circulating_av_norm = InH_av_norm+OutH_av_norm
Circulating_av_norm = np.array(circulating_av_norm).flatten()

#Av aggressive
LeftH_av_agg=Left_av_agg*12
RightH_av_agg=Right_av_agg*12
InH_av_agg=In_av_agg*12
OutH_av_agg=Out_av_agg*12
lefth_av_agg=np.array(LeftH_av_agg).flatten()
righth_av_agg=np.array(RightH_av_agg).flatten()
inh_av_agg = np.array(InH_av_agg).flatten()
outh_av_agg = np.array(OutH_av_agg).flatten()

entering_av_agg = LeftH_av_agg+RightH_av_agg
Entering_av_agg = np.array(entering_av_agg).flatten()
circulating_av_agg = InH_av_agg+OutH_av_agg
Circulating_av_agg = np.array(circulating_av_agg).flatten()
```

In [19]:
```
#Base case simulation
pd.DataFrame(lefth).to_clipboard()
```

In [20]:
```
pd.DataFrame(righth).to_clipboard()
```

In [21]:
```
pd.DataFrame(Entering).to_clipboard()
```

In [22]:
```
pd.DataFrame(outh).to_clipboard()
```

In [23]:
```
pd.DataFrame(Circulating).to_clipboard()
```

In [ ]:
```

```

In [24]:
```
#Decreased critical gap
pd.DataFrame(lefth2).to_clipboard()
```

In [25]:
```
pd.DataFrame(righth2).to_clipboard()
```

In [26]:
```
pd.DataFrame(Entering2).to_clipboard()
```

In [27]:
```
pd.DataFrame(outh2).to_clipboard()
```

In [28]:
```
pd.DataFrame(Circulating2).to_clipboard()
```

In [ ]:

In [29]:
```
#Decreased follow-up times
pd.DataFrame(lefth3).to_clipboard()
```

In [30]:
```
pd.DataFrame(righth3).to_clipboard()
```

In [31]:
```
pd.DataFrame(Entering3).to_clipboard()
```

In [32]:
```
pd.DataFrame(outh3).to_clipboard()
```

In [33]:
```
pd.DataFrame(Circulating3).to_clipboard()
```

In [ ]:

In [34]:
```
#Iteration 1
```

In [35]:
```
pd.DataFrame(lefth_it1).to_clipboard()
```

In [36]:
```
pd.DataFrame(righth_it1).to_clipboard()
```

In [37]:
```
pd.DataFrame(Entering_it1).to_clipboard()
```

In [38]:
```
pd.DataFrame(outh_it1).to_clipboard()
```

In [39]:
```
pd.DataFrame(Circulating_it1).to_clipboard()
```

In [ ]:

In [40]:
```
#Iteration 2
```

In [41]:
```
pd.DataFrame(lefth_it2).to_clipboard()
```

In [42]:
```
pd.DataFrame(righth_it2).to_clipboard()
```

In [43]:
```
pd.DataFrame(Entering_it2).to_clipboard()
```

In [44]:
```
pd.DataFrame(outh_it2).to_clipboard()
```

In [45]:
```
pd.DataFrame(Circulating_it2).to_clipboard()
```

In [ ]:

In [46]:
```
#Iteration 3
```

In [47]:
```
pd.DataFrame(lefth_it3).to_clipboard()
```

In [48]:
```
pd.DataFrame(righth_it3).to_clipboard()
```

In [49]:
```
pd.DataFrame(Entering_it3).to_clipboard()
```

In [50]:
```
pd.DataFrame(outh_it3).to_clipboard()
```

In [51]:
```
pd.DataFrame(Circulating_it3).to_clipboard()
```

In [ ]:

In [52]:
```
#Iteration 4
```

In [53]:
```
pd.DataFrame(lefth_it4).to_clipboard()
```

In [54]:
```
pd.DataFrame(righth_it4).to_clipboard()
```

In [55]:
```
pd.DataFrame(Entering_it4).to_clipboard()
```

In [56]:
```
pd.DataFrame(outh_it4).to_clipboard()
```

In [57]:
```
pd.DataFrame(Circulating_it4).to_clipboard()
```

In [ ]:

In [58]:
```
#Iteration 5
```

In [59]:
```
pd.DataFrame(lefth_it5).to_clipboard()
```

In [60]:
```
pd.DataFrame(righth_it5).to_clipboard()
```

In [61]:
```
pd.DataFrame(Entering_it5).to_clipboard()
```

In [62]:
```
pd.DataFrame(outh_it5).to_clipboard()
```

In [63]:
```
pd.DataFrame(Circulating_it5).to_clipboard()
```

In [ ]:
```

```

In [64]:
```
#Iteration 6
```

In [65]:
```
pd.DataFrame(lefth_it6).to_clipboard()
```

In [66]:
```
pd.DataFrame(righth_it6).to_clipboard()
```

In [67]:
```
pd.DataFrame(Entering_it6).to_clipboard()
```

In [68]:
```
pd.DataFrame(outh_it6).to_clipboard()
```

In [69]:
```
pd.DataFrame(Circulating_it6).to_clipboard()
```

In [ ]:
```

```

In [70]:
```
#Iteration 7
```

In [71]:
```
pd.DataFrame(lefth_it7).to_clipboard()
```

In [72]:
```
pd.DataFrame(righth_it7).to_clipboard()
```

In [73]:
```
pd.DataFrame(Entering_it7).to_clipboard()
```

In [74]:
```
pd.DataFrame(outh_it7).to_clipboard()
```

In [75]:
```
pd.DataFrame(Circulating_it7).to_clipboard()
```

In [ ]:
```

```

In [76]:
```
#Validated model
pd.DataFrame(lefth_val).to_clipboard()
```

In [77]:
```
pd.DataFrame(righth_val).to_clipboard()
```

In [78]:
```
pd.DataFrame(Entering_val).to_clipboard()
```

In [79]:
```
pd.DataFrame(outh_val).to_clipboard()
```

In [80]:
```
pd.DataFrame(Circulating_val).to_clipboard()
```

In [ ]:
```

```

In [81]:
```
#AV cautious
pd.DataFrame(lefth_av_caut).to_clipboard()
```

In [82]:
```
pd.DataFrame(righth_av_caut).to_clipboard()
```

In [83]:
```
pd.DataFrame(Entering_av_caut).to_clipboard()
```

In [84]:
```
pd.DataFrame(outh_av_caut).to_clipboard()
```

In [85]:
```
pd.DataFrame(Circulating_av_caut).to_clipboard()
```

In [ ]:
```

```

In [86]:
```
#Av normal
pd.DataFrame(lefth_av_norm).to_clipboard()
```

In [87]:
```
pd.DataFrame(righth_av_norm).to_clipboard()
```

In [88]:
```
pd.DataFrame(Entering_av_norm).to_clipboard()
```

In [89]:
```
pd.DataFrame(outh_av_norm).to_clipboard()
```

In [90]:
```
pd.DataFrame(Circulating_av_norm).to_clipboard()
```

In [ ]:
```

```

In [91]:
```
#AV aggressive
```

In [92]:
```
pd.DataFrame(lefth_av_agg).to_clipboard()
```

In [93]:
```
pd.DataFrame(righth_av_agg).to_clipboard()
```

In [94]:
```
pd.DataFrame(Entering_av_agg).to_clipboard()
```

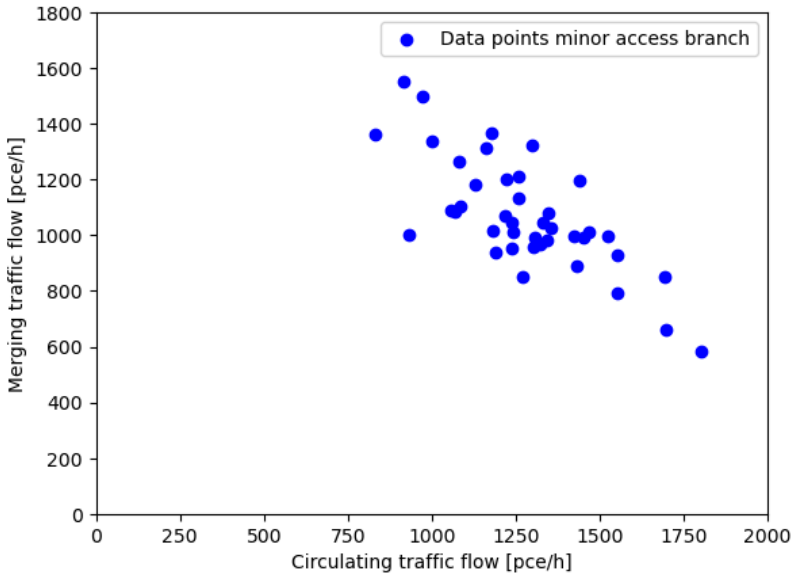
In [95]:
```
pd.DataFrame(outh_av_agg).to_clipboard()
```

In [96]:
```
pd.DataFrame(Circulating_av_agg).to_clipboard()
```

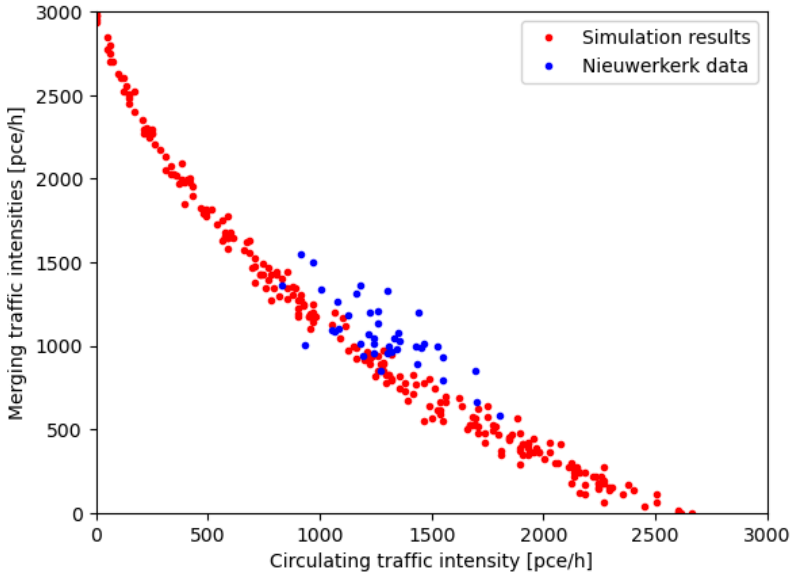
In [97]:
```
data = pd.read_excel('Metingen Oost.xlsx')
Entering_dat = np.array(data['Entering'])
Inner_dat = np.array(data['Inner'])
Outer_dat = np.array(data['Outer'])
Circulating_dat = np.array(data['Circulating'])
Left_dat = np.array(data['Left'])
Right_dat = np.array(data['Right'])
```

In [98]:
```
plt.scatter(Circulating_dat, Entering_dat, color='blue', label = 'Data poin
plt.xlim([0,2000])
plt.ylim([0,1800])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic flow [pce/h]')
plt.ylabel('Merging traffic flow [pce/h]')
```

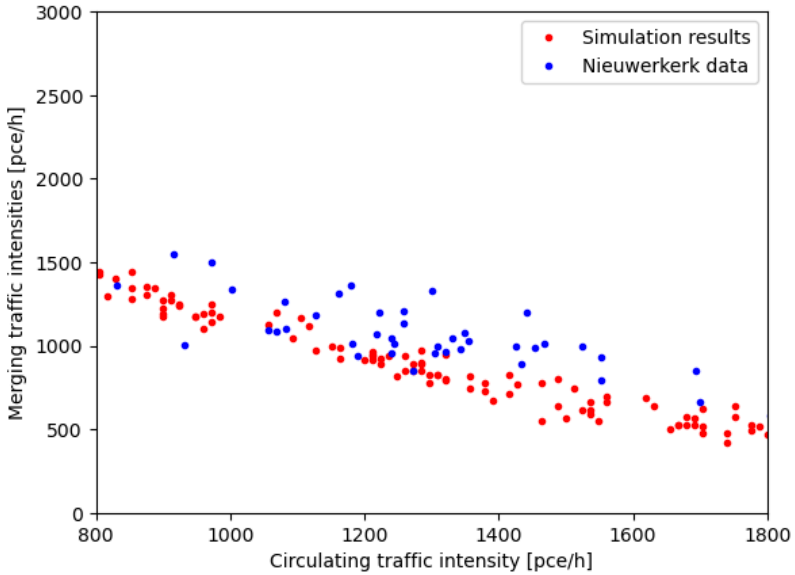
Out[98]:
```
Text(0, 0.5, 'Merging traffic flow [pce/h]')
```



In [99]:
```
plt.plot(Circulating, Entering,'r.',label = 'Simulation results')
plt.plot(Circulating_dat,Entering_dat,'b.', label = 'Nieuwerkerk data')
plt.legend(loc='best');
plt.ylim([0,3000])
plt.xlim([0,3000])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

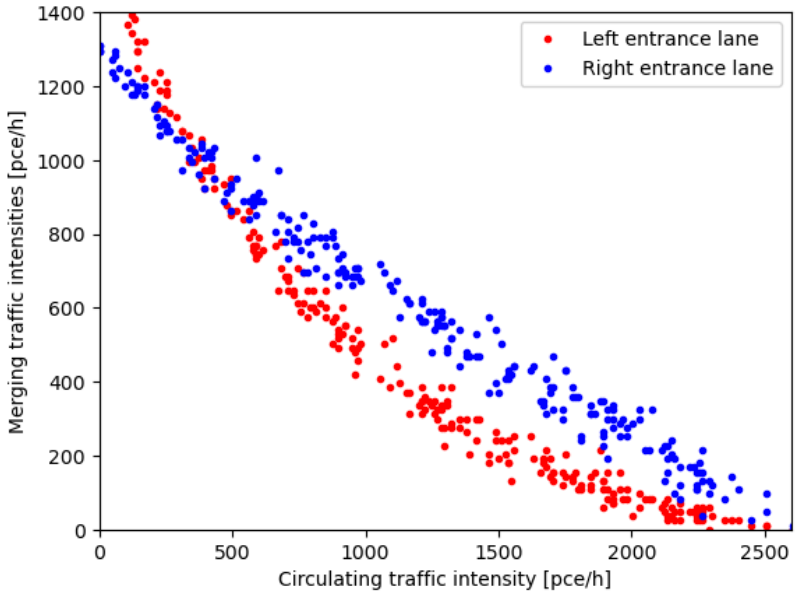


In [100]:
```
plt.plot(Circulating, Entering,'r.',label = 'Simulation results')
plt.plot(Circulating_dat,Entering_dat,'b.', label = 'Nieuwerkerk data')
plt.legend(loc='best');
plt.ylim([0,3000])
plt.xlim([800,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

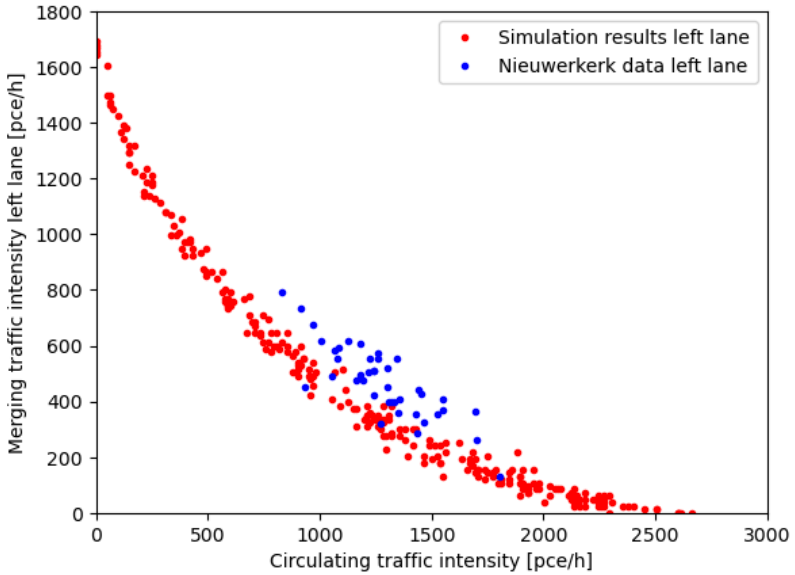


In [101]:
```
lefth=np.array(LeftH).flatten()
righth=np.array(RightH).flatten()

lefth2=np.array(LeftH2).flatten()
righth2=np.array(RightH2).flatten()
```

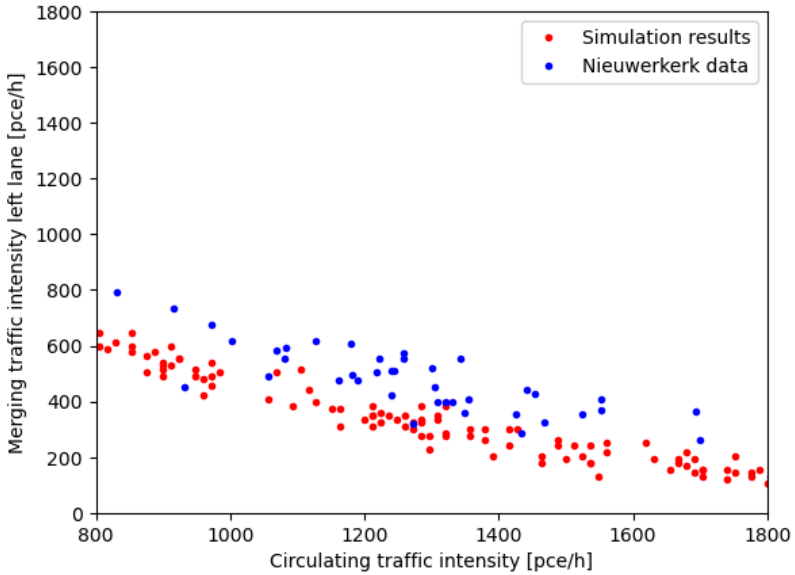
In [102]:
```
plt.plot(Circulating, lefth, 'r.', label = 'Left entrance lane')
plt.plot(Circulating, righth,'b.', label = 'Right entrance lane')
plt.xlim([0,2600])
plt.ylim([0,1400])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```



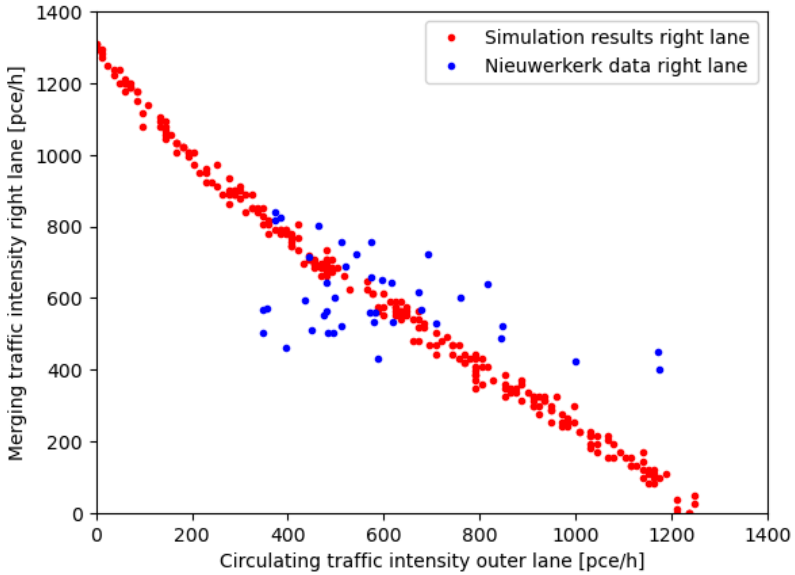
In [103]:
```
plt.plot(Circulating, lefth, 'r.', label = 'Simulation results left lane')
plt.plot(Circulating_dat, Left_dat,'b.', label = 'Nieuwerkerk data left lan
plt.legend(loc = 'best')
plt.xlim([0,3000])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensity left lane [pce/h]');
```



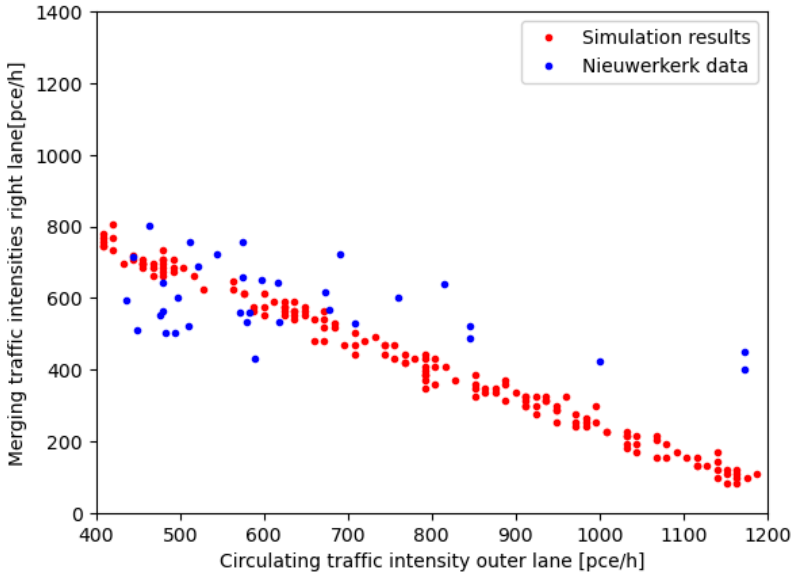
In [104]:
```
plt.plot(Circulating, lefth, 'r.', label = 'Simulation results')
plt.plot(Circulating_dat, Left_dat,'b.', label = 'Nieuwerkerk data')
plt.legend(loc = 'best')
plt.ylim([0,1800])
plt.xlim([800,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensity left lane [pce/h]');
```



In [105]:
```
plt.plot(outh, righth, 'r.', label = 'Simulation results right lane')
plt.plot(Outer_dat, Right_dat,'b.', label = 'Nieuwerkerk data right lane')
plt.legend(loc = 'best')
plt.xlim([0,1400])
plt.ylim([0,1400])
plt.xlabel('Circulating traffic intensity outer lane [pce/h]')
plt.ylabel('Merging traffic intensity right lane [pce/h]');
```

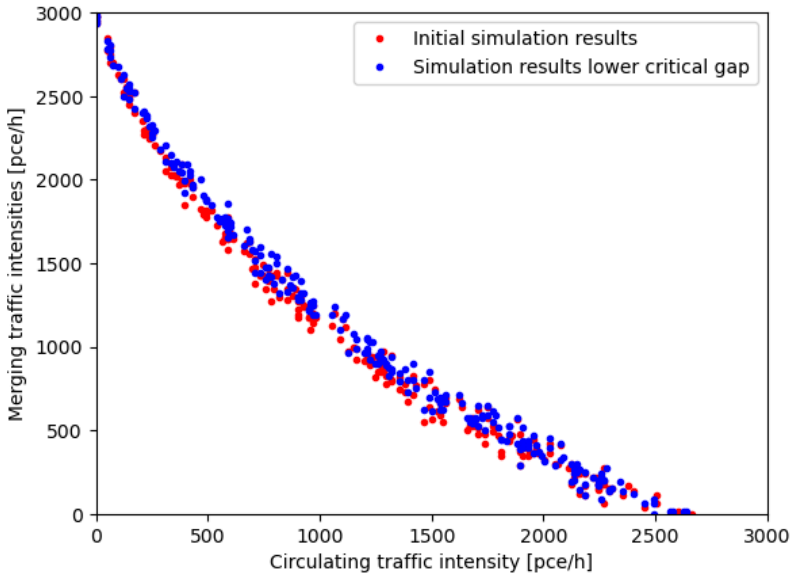


In [106]:
```
plt.plot(outh, righth, 'r.', label = 'Simulation results')
plt.plot(Outer_dat, Right_dat,'b.', label = 'Nieuwerkerk data')
plt.legend(loc = 'best')
plt.ylim([0,1400])
plt.xlim([400,1200])
plt.xlabel('Circulating traffic intensity outer lane [pce/h]')
plt.ylabel('Merging traffic intensities right lane[pce/h]');
```

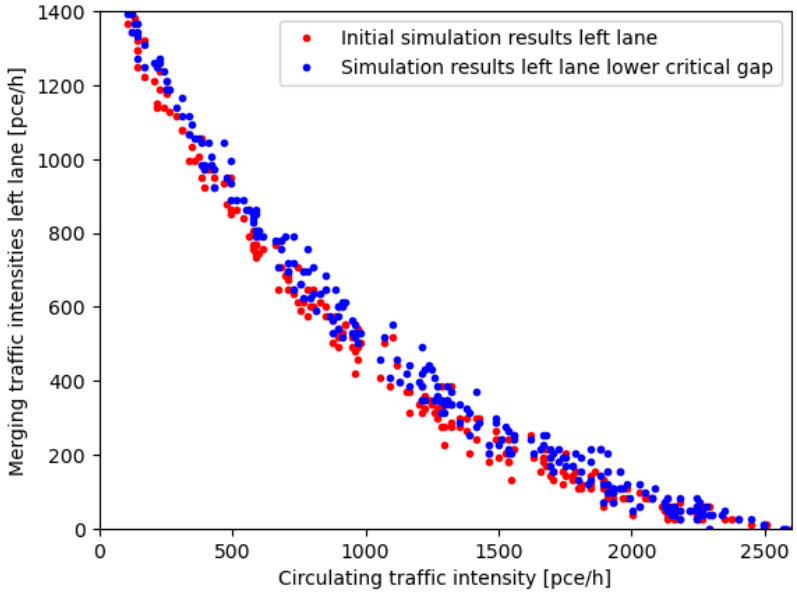


10 percent decrease critical gap

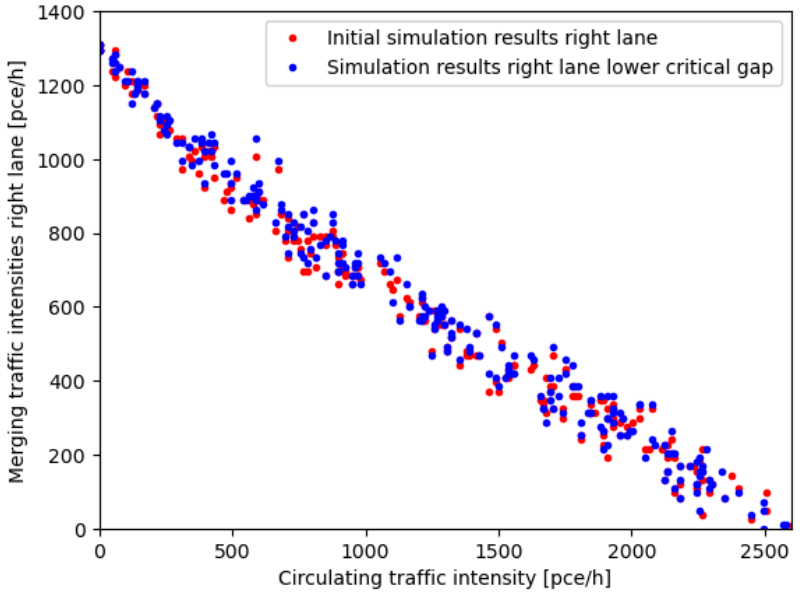
In [107]:
```
plt.plot(Circulating, Entering,'r.',label = 'Initial simulation results')
plt.plot(Circulating2, Entering2,'b.',label = 'Simulation results lower cri
plt.legend(loc='best')
plt.xlim([0,3000])
plt.ylim([0,3000])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```



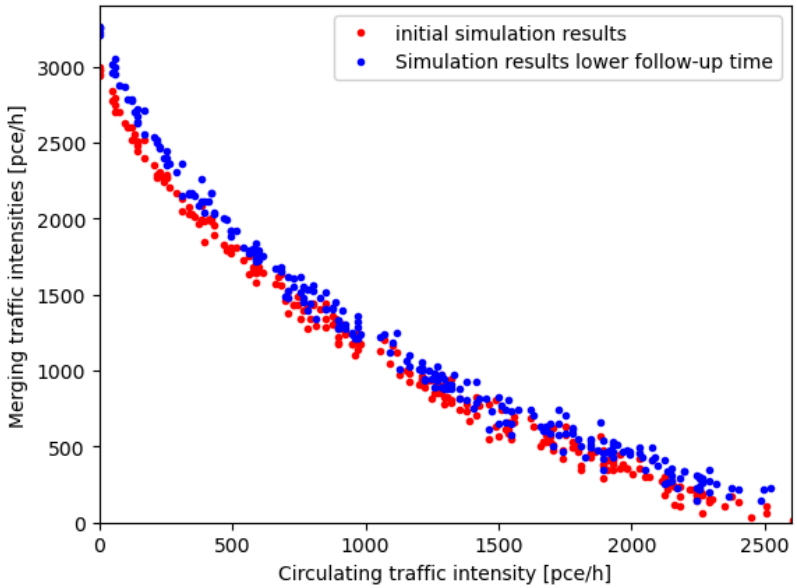
In [108]:
```
plt.plot(Circulating,lefth,'r.', label = 'Initial simulation results left l
plt.plot(Circulating2,lefth2, 'b.', label = 'Simulation results left lane l
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,1400])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left lane [pce/h]');
```



In [109]:
```
plt.plot(Circulating,righth,'r.', label = 'Initial simulation results right
plt.plot(Circulating2,righth2, 'b.', label = 'Simulation results right lane
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,1400])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities right lane [pce/h]');
```

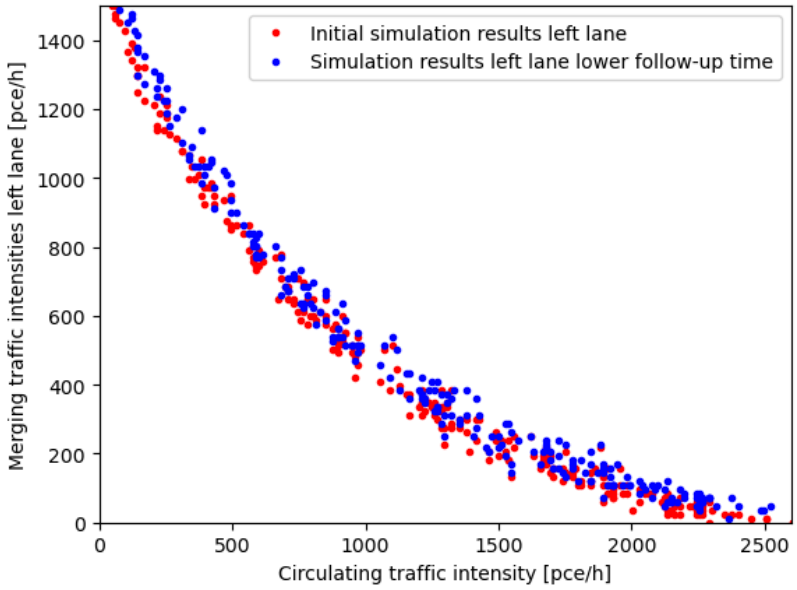


In [110]:
```
plt.plot(Circulating, Entering,'r.',label = 'initial simulation results')
plt.plot(Circulating3, Entering3,'b.',label = 'Simulation results lower fol
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,3400])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

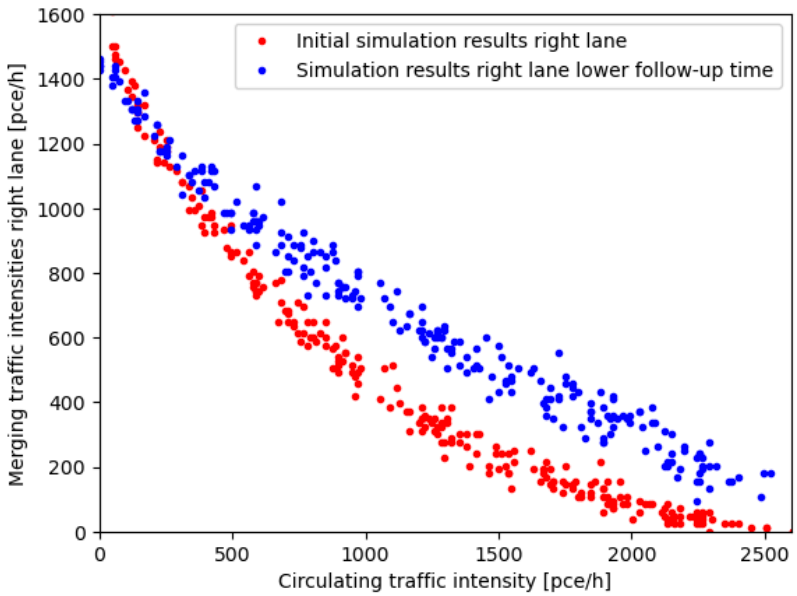


10 percent decrease follow-up time

In [111]:
```
plt.plot(Circulating, lefth,'r.',label = 'Initial simulation results left l
plt.plot(Circulating3, lefth3,'b.',label = 'Simulation results left lane lo
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,1500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left lane [pce/h]');
```

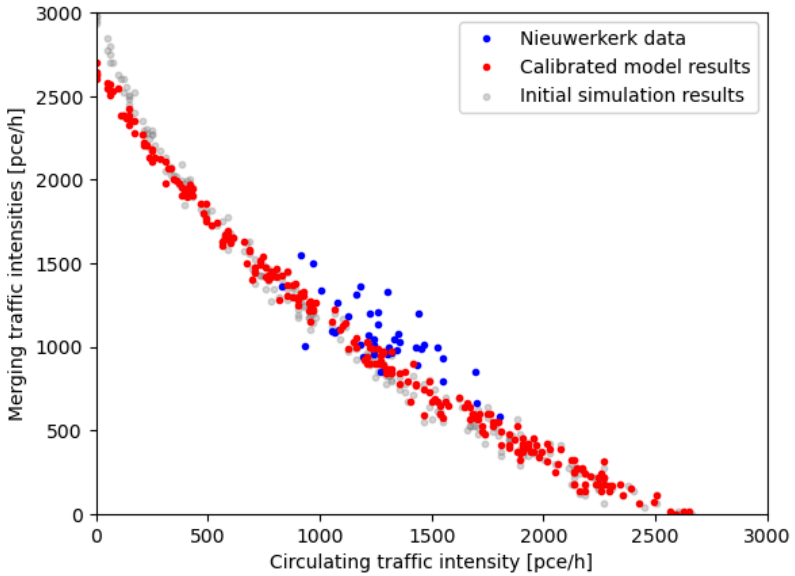


In [112]:
```
plt.plot(Circulating, lefth,'r.',label = 'Initial simulation results right
plt.plot(Circulating3, righth3,'b.',label = 'Simulation results right lane
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,1600])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities right lane [pce/h]');
```

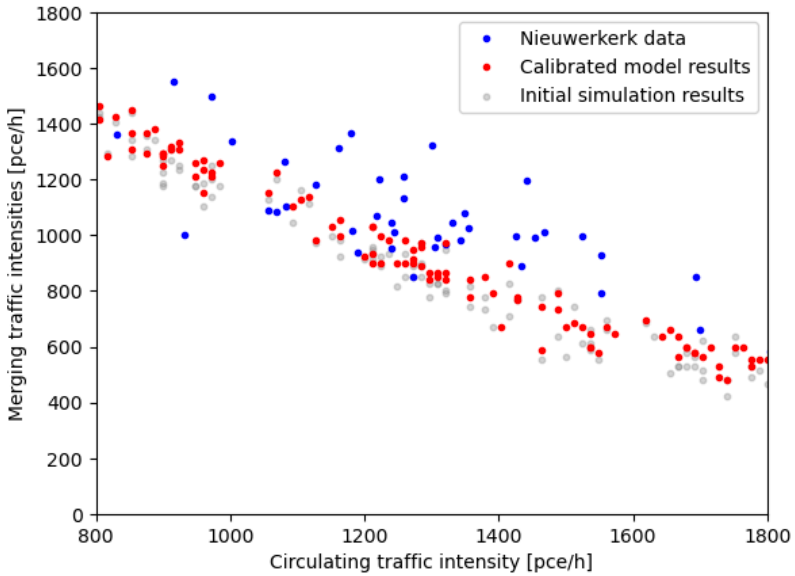


Validated model

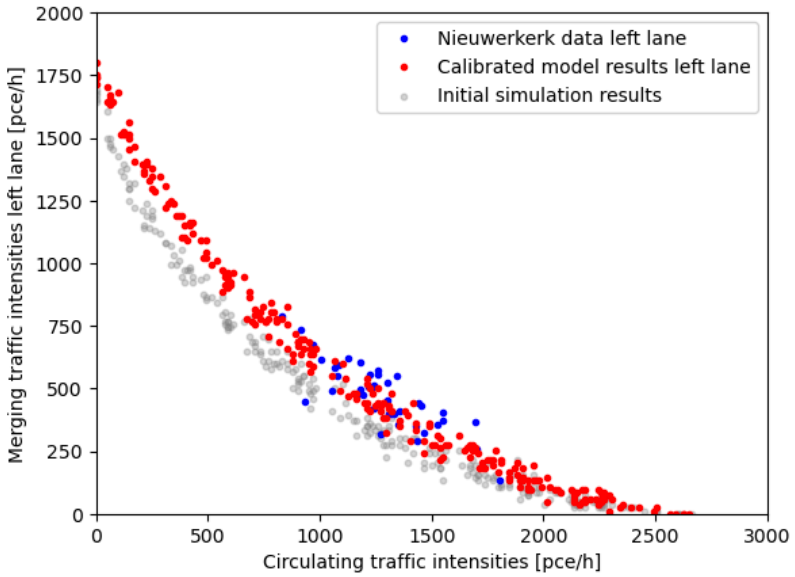
In [113]:
```
plt.plot(Circulating_dat,Entering_dat,'b.', label = 'Nieuwerkerk data')
plt.plot(Circulating_it7, Entering_it7, 'r.', label = 'Calibrated model res
plt.scatter(Circulating, Entering,color='grey',s=10,alpha=0.3,label = 'Init
plt.legend(loc = 'best')
plt.ylim([0,3000])
plt.xlim([0,3000])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```



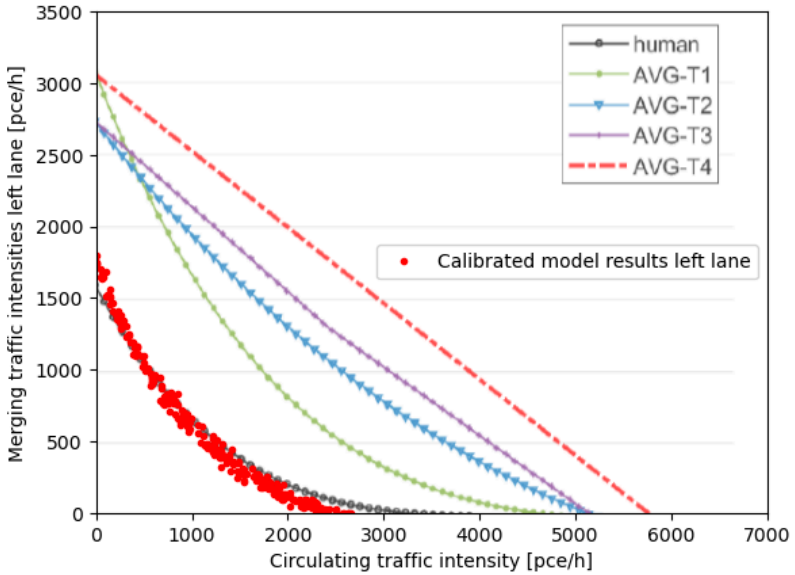
In [114]:
```
plt.plot(Circulating_dat,Entering_dat,'b.', label = 'Nieuwerkerk data')
plt.plot(Circulating_it7, Entering_it7, 'r.', label = 'Calibrated model res
plt.scatter(Circulating, Entering,color='grey',s=10,alpha=0.3,label = 'Init
plt.legend(loc = 'best')
plt.ylim([0,1800])
plt.xlim([800,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```



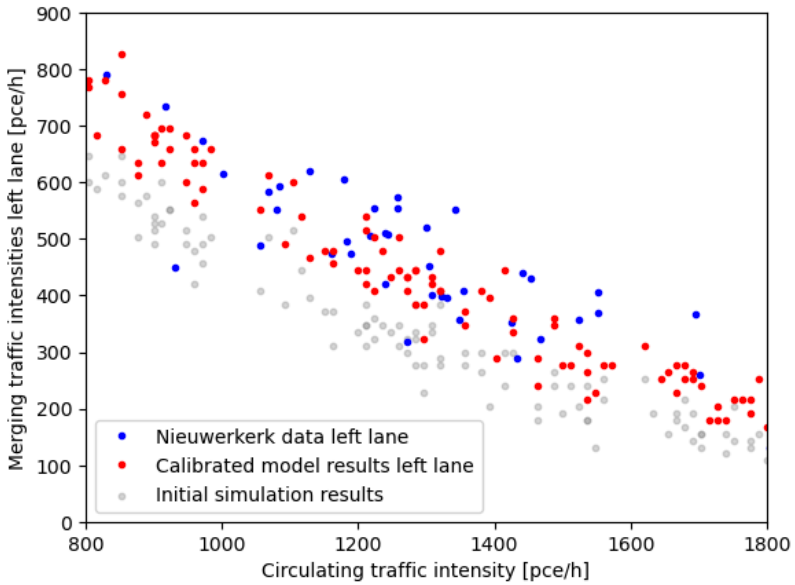
In [115]:
```
plt.plot(Circulating_dat, Left_dat,'b.',label = 'Nieuwerkerk data left lane
plt.plot(Circulating_it7, lefth_it7, 'r.', label = 'Calibrated model result
plt.scatter(Circulating, lefth,color='grey',s=10,alpha=0.3,label = 'Initial
plt.legend(loc = 'best')
plt.ylim([0,2000])
plt.xlim([0,3000])
plt.xlabel('Circulating traffic intensities [pce/h]')
plt.ylabel('Merging traffic intensities left lane [pce/h]');
```



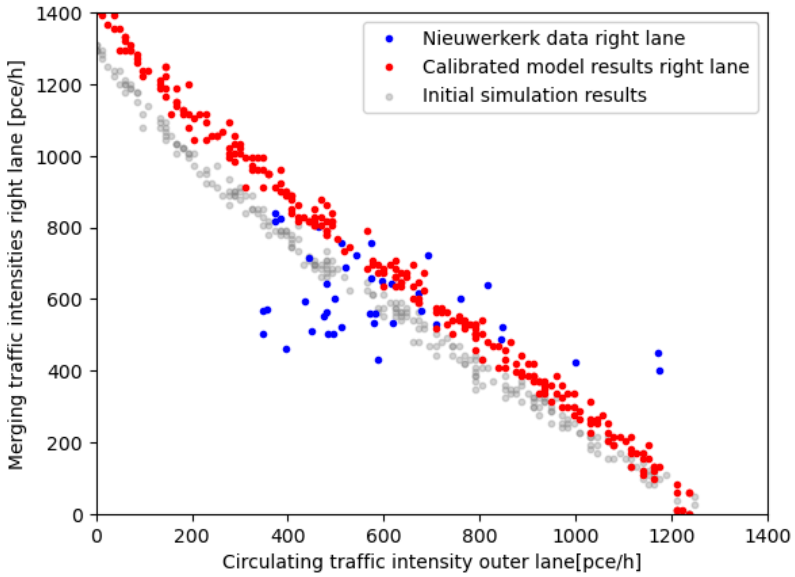
In [116]:
```
im = plt.imread('Fortuijn3.PNG')
plt.imshow(im, extent=[0, 7000, 0, 3500], aspect='auto', alpha=0.7)

xhuman = np.array([0,111,210,315,421,526,605,710,815,921,1000,1105,1210,131
yhuman = np.array([1633,1466.66,1367,1283,1195,1105,1016,925,835,768,708,65

#plt.plot(Circulating_dat, Left_dat,'b.',label = 'Nieuwerkerk data left lan
plt.plot(Circulating_it7, lefth_it7, 'r.', label = 'Calibrated model result
#plt.scatter(Circulating, lefth,color='grey',s=10,alpha=0.3,label = 'Initia
#plt.legend(loc = 'best')
plt.ylim([0,3500])
plt.xlim([0,7000])
plt.legend(loc=7)
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left lane [pce/h]');
```



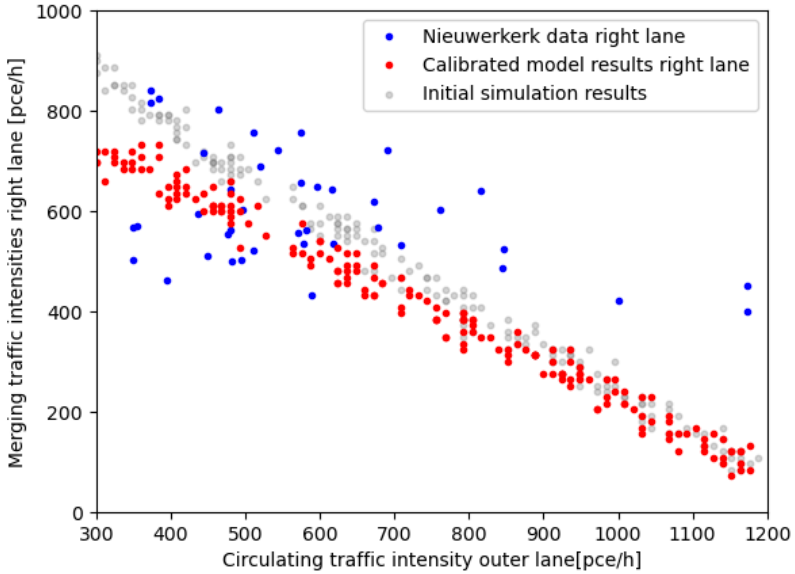
In [117]:
```
plt.plot(Circulating_dat, Left_dat,'b.',label = 'Nieuwerkerk data left lane
plt.plot(Circulating_it7, lefth_it7, 'r.', label = 'Calibrated model result
plt.scatter(Circulating, lefth,color='grey',s=10,alpha=0.3,label = 'Initial
plt.legend(loc = 'best')
plt.ylim([0,900])
plt.xlim([800,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left lane [pce/h]');
```



In [132]:
```
plt.plot(Outer_dat, Right_dat,'b.',label = 'Nieuwerkerk data right lane')
plt.plot(outh_it3, righth_it3, 'r.', label = 'Calibrated model results righ
plt.scatter(outh, righth,color='grey',s=10,alpha=0.3,label = 'Initial simul
plt.legend(loc = 'best')
plt.ylim([0,1400])
plt.xlim([0,1400])
plt.xlabel('Circulating traffic intensity outer lane[pce/h]')
plt.ylabel('Merging traffic intensities right lane [pce/h]');
```

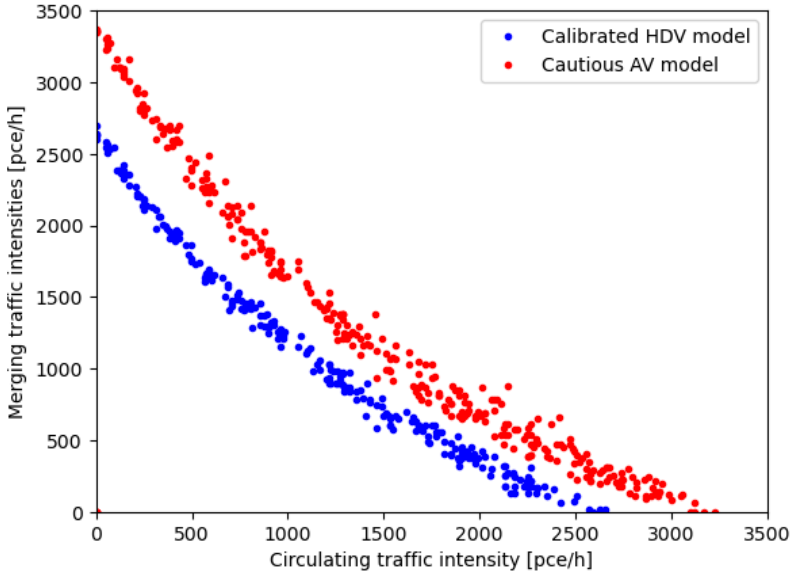


In [119]:
```
plt.plot(Outer_dat, Right_dat,'b.',label = 'Nieuwerkerk data right lane')
plt.plot(outh_it7, righth_it7, 'r.', label = 'Calibrated model results righ
plt.scatter(outh, righth,color='grey',s=10,alpha=0.3,label = 'Initial simul
plt.legend(loc = 'best')
plt.ylim([0,1000])
plt.xlim([300,1200])
plt.xlabel('Circulating traffic intensity outer lane[pce/h]')
plt.ylabel('Merging traffic intensities right lane [pce/h]');
```

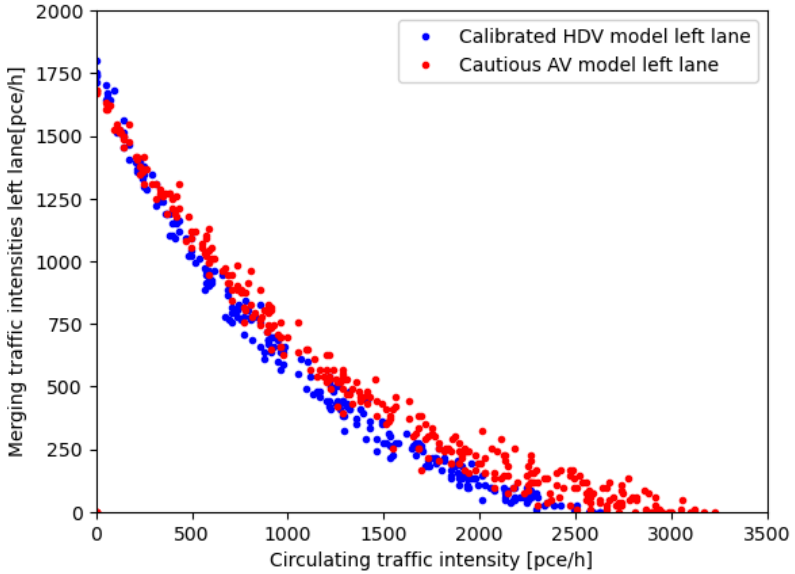


Comparison validated model and av cautious

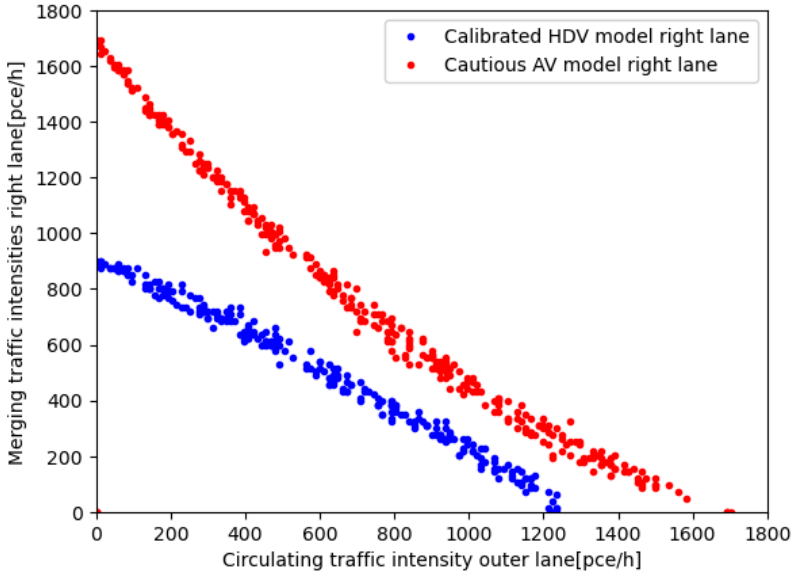
In [120]:
```
plt.plot(Circulating_it7, Entering_it7, 'b.', label = 'Calibrated HDV model
plt.plot(Circulating_av_caut, Entering_av_caut, 'r.', label = 'Cautious AV
plt.legend(loc = 'best')
plt.ylim([0,3500])
plt.xlim([0,3500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```



In [121]:
```
plt.plot(Circulating_it7, lefth_it7, 'b.', label = 'Calibrated HDV model le
plt.plot(Circulating_av_caut, lefth_av_caut, 'r.', label = 'Cautious AV mod
plt.legend(loc = 'best')
plt.ylim([0,2000])
plt.xlim([0,3500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left lane[pce/h]');
```

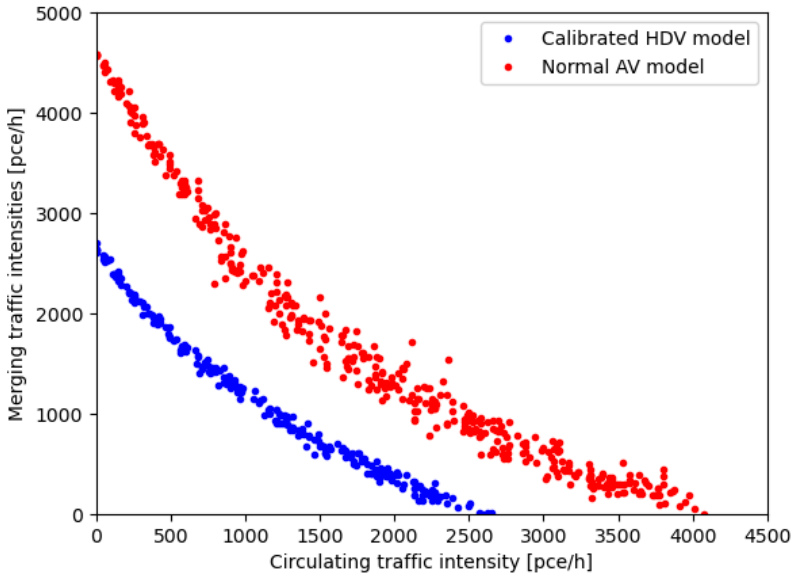


In [122]:
```
plt.plot(outh_it7, righth_it7, 'b.', label = 'Calibrated HDV model right la
plt.plot(outh_av_caut, righth_av_caut, 'r.', label = 'Cautious AV model rig
plt.legend(loc = 'best')
plt.ylim([0,1800])
plt.xlim([0,1800])
plt.xlabel('Circulating traffic intensity outer lane[pce/h]')
plt.ylabel('Merging traffic intensities right lane[pce/h]');
```

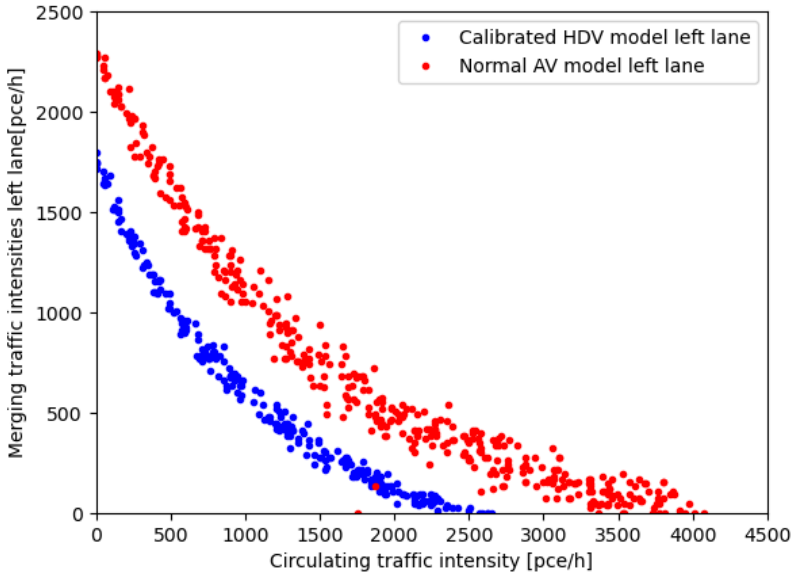


Comparison validated model and AV normal

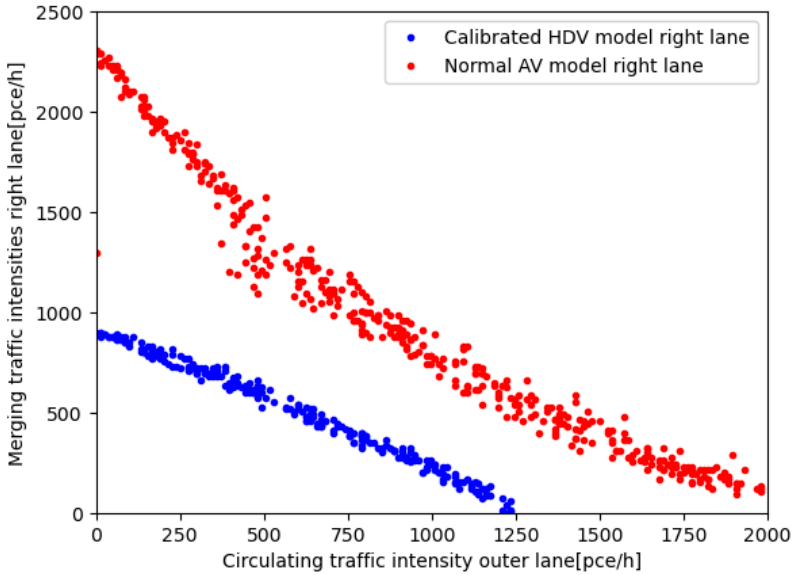
In [123]:
```
plt.plot(Circulating_it7, Entering_it7, 'b.', label = 'Calibrated HDV model
plt.plot(Circulating_av_norm, Entering_av_norm, 'r.', label = 'Normal AV mo
plt.legend(loc = 'best')
plt.ylim([0,5000])
plt.xlim([0,4500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```



In [124]:
```
plt.plot(Circulating_it7, lefth_it7, 'b.', label = 'Calibrated HDV model le
plt.plot(Circulating_av_norm, lefth_av_norm, 'r.', label = 'Normal AV model
plt.legend(loc = 'best')
plt.ylim([0,2500])
plt.xlim([0,4500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left lane[pce/h]');
```

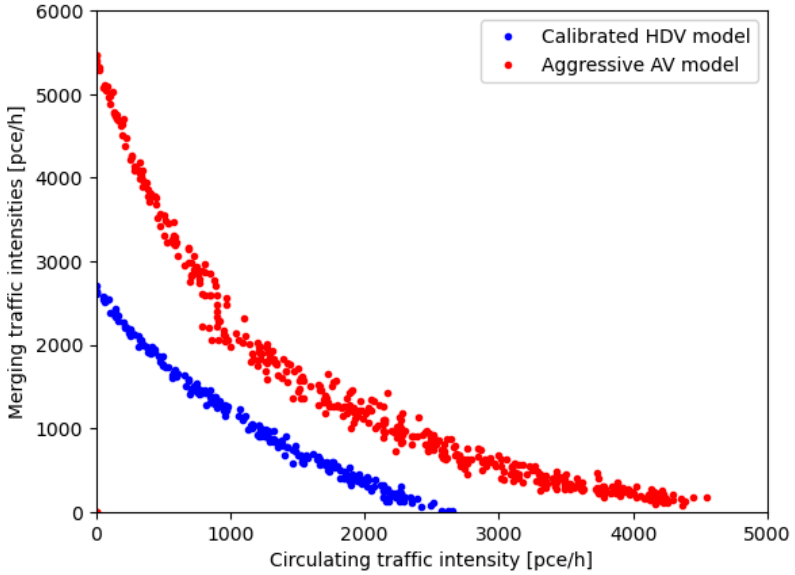


In [125]:
```
plt.plot(outh_it7, righth_it7, 'b.', label = 'Calibrated HDV model right la
plt.plot(outh_av_norm, righth_av_norm, 'r.', label = 'Normal AV model right
plt.legend(loc = 'best')
plt.ylim([0,2500])
plt.xlim([0,2000])
plt.xlabel('Circulating traffic intensity outer lane[pce/h]')
plt.ylabel('Merging traffic intensities right lane[pce/h]');
```

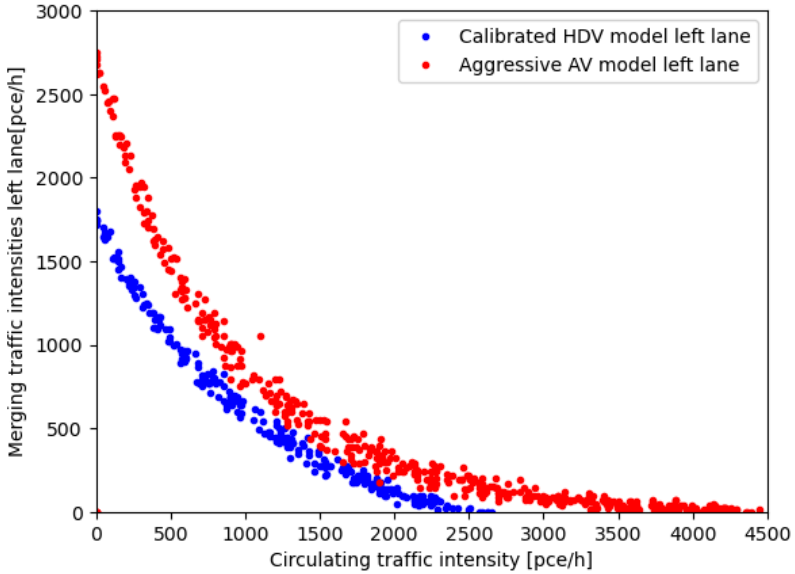


Comparison validated model and AV aggressive

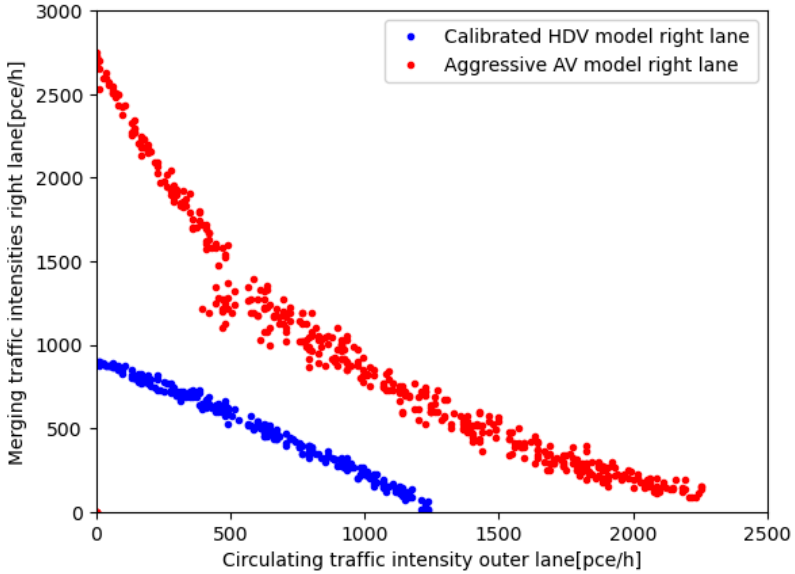
In [126]:
```
plt.plot(Circulating_it7, Entering_it7, 'b.', label = 'Calibrated HDV model
plt.plot(Circulating_av_agg, Entering_av_agg, 'r.', label = 'Aggressive AV
plt.legend(loc = 'best')
plt.ylim([0,6000])
plt.xlim([0,5000])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```



In [127]:
```
plt.plot(Circulating_it7, lefth_it7, 'b.', label = 'Calibrated HDV model le
plt.plot(Circulating_av_agg, lefth_av_agg, 'r.', label = 'Aggressive AV mod
plt.legend(loc = 'best')
plt.ylim([0,3000])
plt.xlim([0,4500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left lane[pce/h]');
```

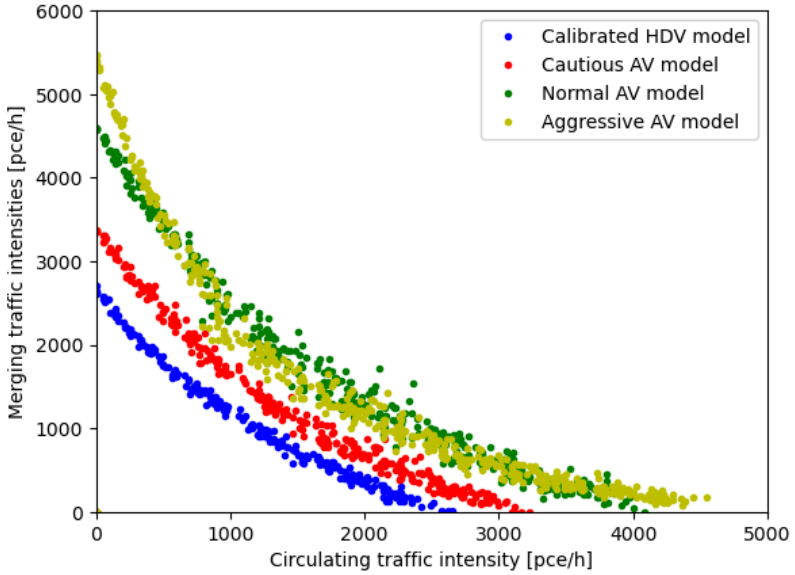


In [128]:
```
plt.plot(outh_it7, righth_it7, 'b.', label = 'Calibrated HDV model right la
plt.plot(outh_av_agg, righth_av_agg, 'r.', label = 'Aggressive AV model rig
plt.legend(loc = 'best')
plt.ylim([0,3000])
plt.xlim([0,2500])
plt.xlabel('Circulating traffic intensity outer lane[pce/h]')
plt.ylabel('Merging traffic intensities right lane[pce/h]');
```

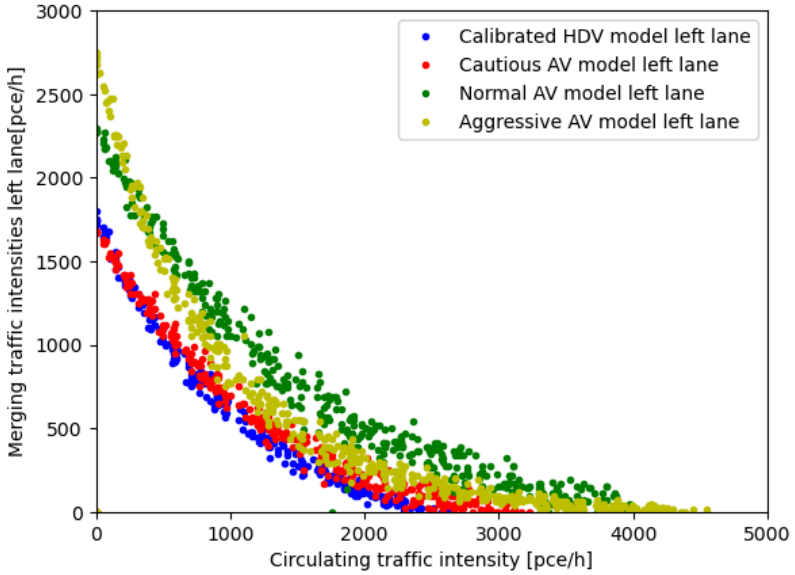


All CAV models plus HDV model validated

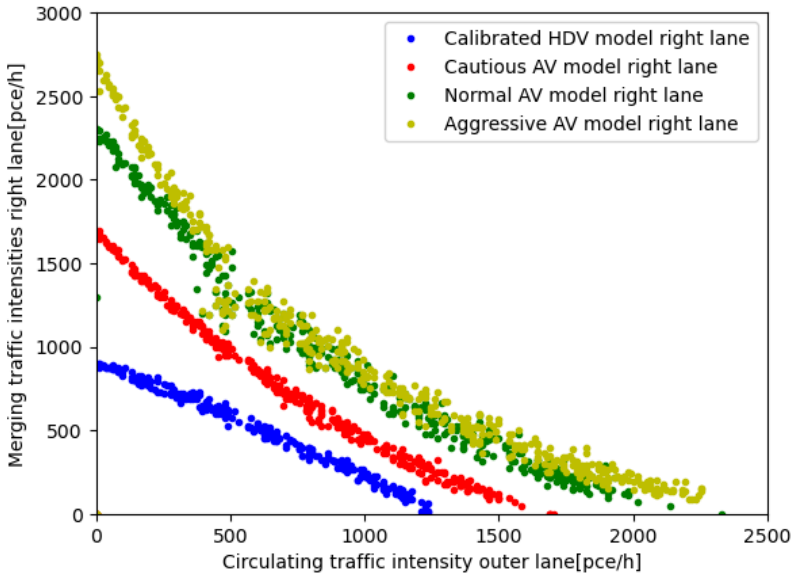
In [129]:
```
plt.plot(Circulating_it7, Entering_it7, 'b.', label = 'Calibrated HDV model
plt.plot(Circulating_av_caut, Entering_av_caut, 'r.', label = 'Cautious AV
plt.plot(Circulating_av_norm, Entering_av_norm, 'g.', label = 'Normal AV mo
plt.plot(Circulating_av_agg, Entering_av_agg, 'y.', label = 'Aggressive AV
plt.legend(loc = 'best')
plt.ylim([0,6000])
plt.xlim([0,5000])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```



In [130]:
```
plt.plot(Circulating_it7, lefth_it7, 'b.', label = 'Calibrated HDV model le
plt.plot(Circulating_av_caut, lefth_av_caut, 'r.', label = 'Cautious AV mod
plt.plot(Circulating_av_norm, lefth_av_norm, 'g.', label = 'Normal AV model
plt.plot(Circulating_av_agg, lefth_av_agg, 'y.', label = 'Aggressive AV mod
plt.legend(loc = 'best')
plt.ylim([0,3000])
plt.xlim([0,5000])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left lane[pce/h]');
```



In [131]:
```
plt.plot(outh_it7, righth_it7, 'b.', label = 'Calibrated HDV model right la
plt.plot(outh_av_caut, righth_av_caut, 'r.', label = 'Cautious AV model rig
plt.plot(outh_av_norm, righth_av_norm, 'g.', label = 'Normal AV model right
plt.plot(outh_av_agg, righth_av_agg, 'y.', label = 'Aggressive AV model rig
plt.legend(loc = 'best')
plt.ylim([0,3000])
plt.xlim([0,2500])
plt.xlabel('Circulating traffic intensity outer lane[pce/h]')
plt.ylabel('Merging traffic intensities right lane[pce/h]');
```



In [ ]:

B

Python Code of the regression methods and calculations of the obtained capacity increases

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import scipy as sc
from scipy.optimize import curve_fit
%clear
```

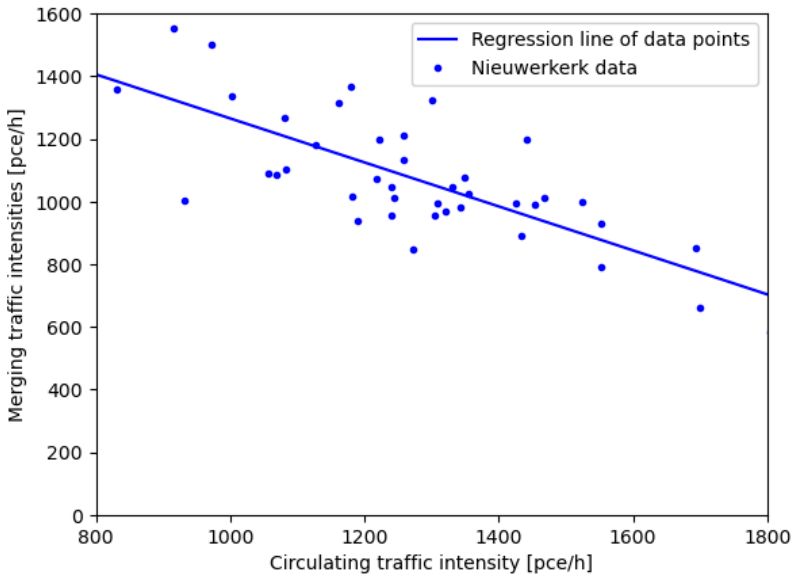
In [2]:

```
#Base simulation
x = pd.read_excel("Validatie Oost 2.xlsx");
Enter_vol = np.array(x['Enter_vol'])[0:255]
Outer_vol = np.array(x["Outer_vol"])[0:255]
Circul_vol = np.array(x['Circul_vol'])[0:255]
Left_vol = np.array(x['Left_vol'])[0:255]
Right_vol = np.array(x['Right_vol'])[0:255]
Right_vol_sorted = np.array(x['Right_vol_sorted'])[0:255]

#Nieuwerkerk Data
Entering_dat = np.array(x['MetingenEntering'])[0:41]
Circulating_dat = np.array(x['MetingenCirculating'])[0:41]
Left_dat = np.array(x['MetingenLeft'])[0:41]
Right_dat = np.array(x['MetingenRight'])[0:41]
Right_dat_sorted = np.array(x['MetingenRightSorted'])[0:41]
Inner_dat = np.array(x['MetingenInner'])[0:41]
Outer_dat = np.array(x['MetingenOuter'])[0:41]
Outer_dat_sorted = np.array(x['MetingenOuterSorted'])[0:41]

#Decreased critical gap
Enter_crit_vol = np.array(x['Enter_crit_vol'])[0:255]
Circul_crit_vol = np.array(x['Circul_crit_vol'])[0:255]
Outer_crit_vol = np.array(x["Outer_crit_vol"])[0:255]
Left_crit_vol = np.array(x['Left_crit_vol'])[0:255]
Right_crit_vol = np.array(x['Right_crit_vol'])[0:255]
Right_crit_vol_sorted = np.array(x['Right_crit_vol_sorted'])[0:255]

#Decreased follow-up time
Enter_fol_vol = np.array(x['Enter_fol_vol'])[0:255]
Circul_fol_vol = np.array(x['Circul_fol_vol'])[0:255]
Outer_fol_vol = np.array(x["Outer_fol_vol"])[0:255]
Left_fol_vol = np.array(x['Left_fol_vol'])[0:255]
Right_fol_vol = np.array(x['Right_fol_vol'])[0:255]
Right_fol_vol_sorted = np.array(x['Right_fol_vol_sorted'])[0:255]

#Iteration 1
Enter_it1 = np.array(x['Enter_it1'])[0:255]
Circul_it1 = np.array(x['Circul_it1'])[0:255]
Outer_it1 = np.array(x["Outer_it1"])[0:255]
Left_it1 = np.array(x['Left_it1'])[0:255]
Right_it1 = np.array(x['Right_it1'])[0:255]
Right_it1_sorted = np.array(x['Right_it1_sorted'])[0:255]

#Iteration 2
Enter_it2 = np.array(x['Enter_it2'])[0:255]
Circul_it2 = np.array(x['Circul_it2'])[0:255]
Outer_it2 = np.array(x["Outer_it2"])[0:255]
Left_it2 = np.array(x['Left_it2'])[0:255]
Right_it2 = np.array(x['Right_it2'])[0:255]
Right_it2_sorted = np.array(x['Right_it2_sorted'])[0:255]

#Iteration 3
Enter_it3 = np.array(x['Enter_it3'])[0:255]
Circul_it3 = np.array(x['Circul_it3'])[0:255]
Outer_it3 = np.array(x["Outer_it3"])[0:255]
Left_it3 = np.array(x['Left_it3'])[0:255]
Right_it3 = np.array(x['Right_it3'])[0:255]
Right_it3_sorted = np.array(x['Right_it3_sorted'])[0:255]

#Iteration 4
Enter_it4 = np.array(x['Enter_it4'])[0:255]
Circul_it4 = np.array(x['Circul_it4'])[0:255]
Outer_it4 = np.array(x["Outer_it4"])[0:255]
Left_it4 = np.array(x['Left_it4'])[0:255]
Right_it4 = np.array(x['Right_it4'])[0:255]
Right_it4_sorted = np.array(x['Right_it4_sorted'])[0:255]

#Iteration 5
Enter_it5 = np.array(x['Enter_it5'])[0:255]
Circul_it5 = np.array(x['Circul_it5'])[0:255]
Outer_it5 = np.array(x["Outer_it5"])[0:255]
Left_it5 = np.array(x['Left_it5'])[0:255]
Right_it5 = np.array(x['Right_it5'])[0:255]
Right_it5_sorted = np.array(x['Right_it5_sorted'])[0:255]

#Iteration 6
Enter_it6 = np.array(x['Enter_it6'])[0:255]
Circul_it6 = np.array(x['Circul_it6'])[0:255]
Outer_it6 = np.array(x["Outer_it6"])[0:255]
Left_it6 = np.array(x['Left_it6'])[0:255]
Right_it6 = np.array(x['Right_it6'])[0:255]
Right_it6_sorted = np.array(x['Right_it6_sorted'])[0:255]

#Iteration 7
Enter_it7 = np.array(x['Enter_it7'])[0:255]
Circul_it7 = np.array(x['Circul_it7'])[0:255]
Outer_it7 = np.array(x['Outer_it7'])[0:255]
Left_it7 = np.array(x['Left_it7'])[0:255]
Right_it7 = np.array(x['Right_it7'])[0:255]
Right_it7_sorted = np.array(x['Right_it7_sorted'])[0:255]

#Validated model
Enter_val_vol = np.array(x['Enter_val_vol'])[0:305]
Circul_val_vol= np.array(x['Circul_val_vol'])[0:305]
Outer_val_vol = np.array(x['Outer_val_vol'])[0:305]
Left_val_vol = np.array(x['Left_val_vol'])[0:305]
Right_val_vol = np.array(x['Right_val_vol'])[0:305]
Right_val_vol_sorted = np.array(x['Right_val_vol_sorted'])[0:305]

#Av Cautious
Enter_cau_vol = np.array(x['Enter_cau'])[0:320]
Circul_cau_vol = np.array(x['Circul_Cau'])[0:320]
Outer_cau_vol = np.array(x['Outer_cau_sorted'])[0:320]
Left_cau_vol = np.array(x['Left_cau'])[0:320]
Right_cau_vol = np.array(x['Right_cau'])[0:320]
Right_cau_vol_sorted = np.array(x['Right_cau_sorted'])[0:320]

#Av normal
Enter_norm_vol = np.array(x['Enter_norm'])[0:405]
Circul_norm_vol = np.array(x['Circul_norm'])[0:405]
Outer_norm_vol = np.array(x['Outer_norm_sorted'])[0:405]
Left_norm_vol = np.array(x['Left_norm'])[0:405]
Right_norm_vol = np.array(x['Right_norm'])[0:405]
Right_norm_vol_sorted = np.array(x['Right_norm_sorted'])[0:405]

#Av aggressive
Enter_agg_vol = np.array(x['Enter_agg'])[0:455]
Circul_agg_vol = np.array(x['Circul_agg'])[0:455]
Outer_agg_vol = np.array(x['Outer_agg_sorted'])[0:455]
Left_agg_vol = np.array(x['Left_agg'])[0:455]
Right_agg_vol = np.array(x['Right_agg'])[0:455]
Right_agg_vol_sorted = np.array(x['Right_agg_sorted'])[0:455]
```

Existing data

In [3]:
```
rdat = sc.stats.linregress(Circulating_dat, Entering_dat,alternative = 'les
slope_dat = rdat.slope
intercept_dat = rdat.intercept
xreg_dat = np.linspace(800,1800,41)
yreg_dat = intercept_dat + slope_dat*xreg_dat

plt.plot(xreg_dat,yreg_dat, color = 'blue', label = 'Regression line of dat
plt.plot(Circulating_dat,Entering_dat, 'b.', label = 'Nieuwerkerk data')
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
plt.xlim([800,1800])
plt.ylim([0,1600])
plt.legend(loc = 'best')

GOFdatlin = np.zeros(len(Entering_dat))
for i in range(len(Entering_dat)):
    GOFdatlin[i] = Entering_dat[i] - (intercept_dat + slope_dat *Circulatin

GOFdatlin_pos = np.sqrt((GOFdatlin)**2)

GOFdatlin_tot = 0
for i in range (len(GOFdatlin_pos)):
    GOFdatlin_tot = GOFdatlin_tot + GOFdatlin_pos[i]

GOFdatlin_def = GOFdatlin_tot/(len(GOFdatlin_pos))
print('The goodness of fit for the linear regression of the simulated data
```

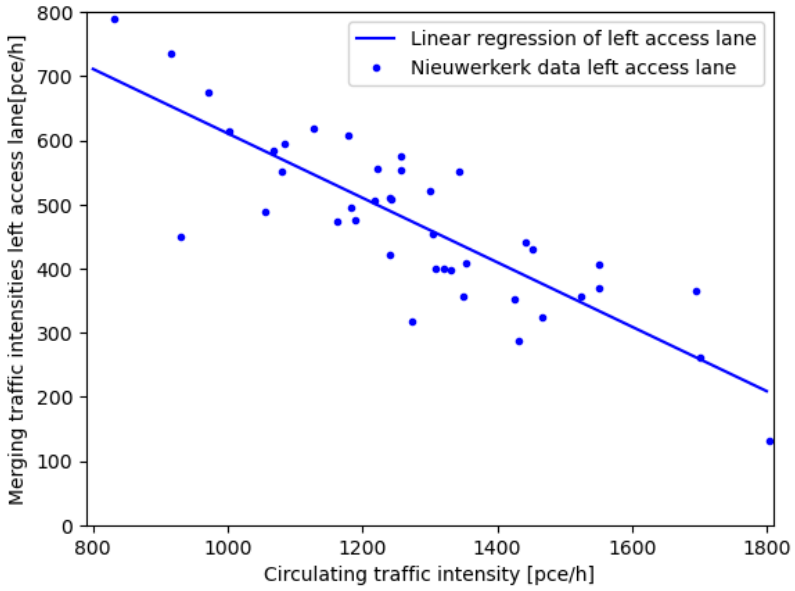
```
The goodness of fit for the linear regression of the simulated data is 10
7.68239461564617
```



Linear regression existing data left lane

In [4]:
```
rdat_Left = sc.stats.linregress(Circulating_dat, Left_dat,alternative = 'le
slope_dat_Left = rdat_Left.slope
intercept_dat_Left = rdat_Left.intercept
xreg_dat_Left = np.linspace(800,1800,41)
yreg_dat_Left = intercept_dat_Left + slope_dat_Left*xreg_dat_Left

plt.plot(xreg_dat_Left,yreg_dat_Left, color = 'blue', label = 'Linear regre
plt.plot(Circulating_dat,Left_dat, 'b.', label = 'Nieuwerkerk data left acc
plt.legend(loc = 'best')
plt.ylim([0,800])
plt.xlim([790,1810])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane[pce/h]');

GOFdatlin_Left = np.zeros(len(Left_dat))
for i in range(len(Left_dat)):
    GOFdatlin_Left[i] = Left_dat[i] - (intercept_dat_Left + slope_dat_Left

GOFdatlin_pos_Left = np.sqrt((GOFdatlin_Left)**2)

GOFdatlin_tot_Left = 0
for i in range (len(GOFdatlin_pos_Left)):
    GOFdatlin_tot_Left = GOFdatlin_tot_Left + GOFdatlin_pos_Left[i]

GOFdatlin_def_Left = GOFdatlin_tot_Left/(len(GOFdatlin_pos_Left))
print('The goodness of fit for the linear regression of the simulated data
```

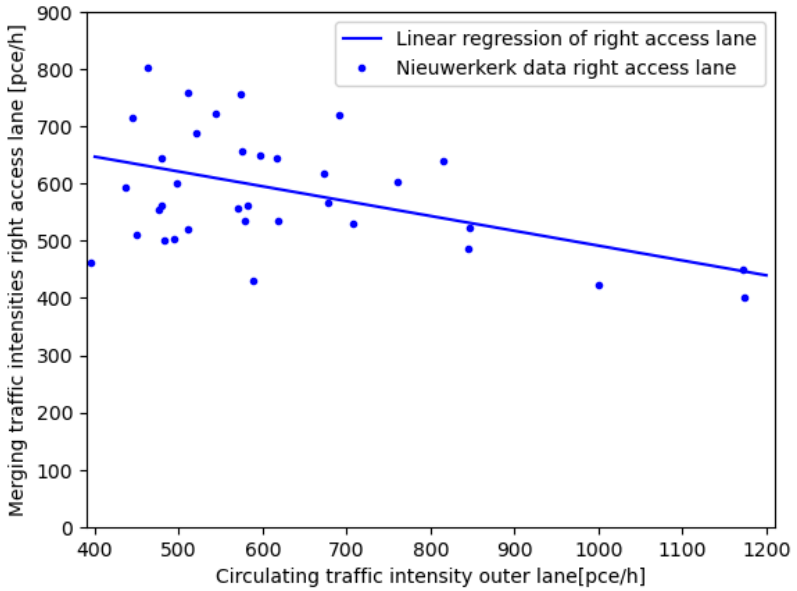
```
The goodness of fit for the linear regression of the simulated data is 57.
91031947786108
```



Linear regression existing data right lane

In [5]:
```
rdat_Right = sc.stats.linregress(Outer_dat, Right_dat,alternative = 'less')
slope_dat_Right = rdat_Right.slope
intercept_dat_Right = rdat_Right.intercept
xreg_dat_Right = np.linspace(400,1200,41)
yreg_dat_Right = intercept_dat_Right + slope_dat_Right*xreg_dat_Right

plt.plot(xreg_dat_Right,yreg_dat_Right, color = 'blue', label = 'Linear reg
plt.plot(Outer_dat,Right_dat, 'b.', label = 'Nieuwerkerk data right access
plt.legend(loc = 'best')
plt.ylim([0,900])
plt.xlim([390,1210])
plt.xlabel('Circulating traffic intensity outer lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');

GOFdatlin_Right = np.zeros(len(Right_dat))
for i in range(len(Right_dat)):
    GOFdatlin_Right[i] = Right_dat[i] - (intercept_dat_Right + slope_dat_Ri

GOFdatlin_pos_Right = np.sqrt((GOFdatlin_Right)**2)

GOFdatlin_tot_Right = 0
for i in range (len(GOFdatlin_pos_Right)):
    GOFdatlin_tot_Right = GOFdatlin_tot_Right + GOFdatlin_pos_Right[i]

GOFdatlin_def_Right = GOFdatlin_tot_Right/(len(GOFdatlin_pos_Right))
print('The goodness of fit for the linear regression of the simulated data
```

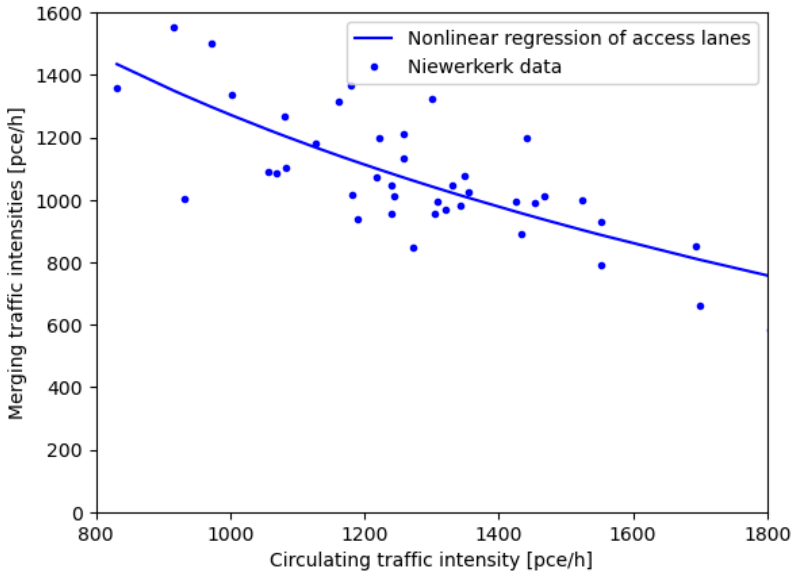
```
The goodness of fit for the linear regression of the simulated data is 86.
73106832684685
```



Nonlinear regression

In [6]:
```
def nonlinear (x,a,b,c):
    return a * (x**2) + b *x + c
def nonlinear2 (x,a,b,c):
    return a/(x/b + 1) + c
def nonlinear3(x, a, b, c):
    return a * np.log(b*x+1e-10) + c

reg_dat , pcov_dat = sc.optimize.curve_fit(nonlinear3, Circulating_dat,Ente
aopt_dat, bopt_dat, copt_dat = reg_dat

y_fit_dat = nonlinear3(Circulating_dat, aopt_dat, bopt_dat, copt_dat)

plt.plot(Circulating_dat, y_fit_dat, color = 'blue', label = 'Nonlinear reg
plt.plot(Circulating_dat, Entering_dat, 'b.', label = 'Niewerkerk data')
plt.legend(loc = 'best')
plt.ylim([0,1600])
plt.xlim([800,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');

GOFdatnon = np.zeros(len(Entering_dat))
for i in range(len(Entering_dat)):
    GOFdatnon[i] = Entering_dat[i] - y_fit_dat[i]

GOFdatnon_pos = np.sqrt((GOFdatnon)**2)

GOFdatnon_tot = 0
for i in range (len(GOFdatnon_pos)):
    GOFdatnon_tot = GOFdatnon_tot + GOFdatnon_pos[i]

GOFdatnon_def = GOFdatnon_tot/(len(GOFdatnon_pos))
print('The goodness of fit for the linear regression of the simulated data
```

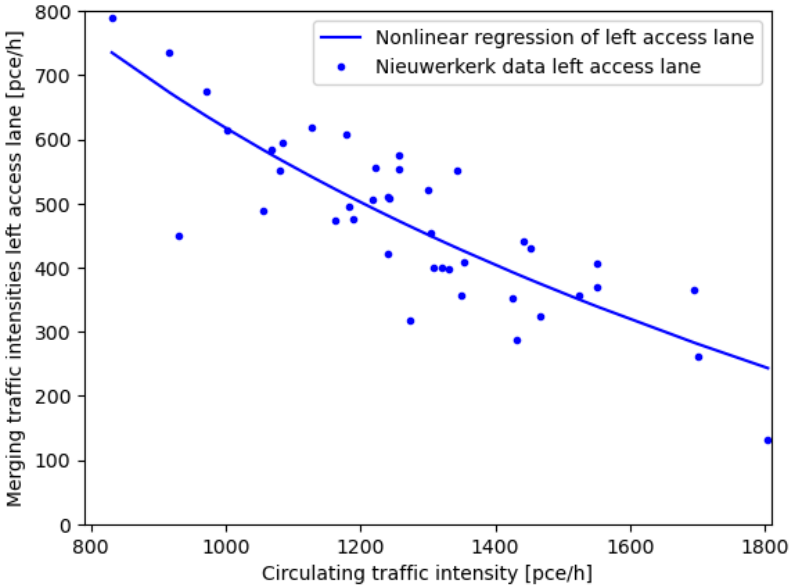
```
The goodness of fit for the linear regression of the simulated data is 10
9.12668266700472
```



Existing data left lane

In [7]:
```
reg_dat_Left , pcov_dat_Left = sc.optimize.curve_fit(nonlinear3, Circulatin
aopt_dat_Left, bopt_dat_Left, copt_dat_Left = reg_dat_Left

y_fit_dat_Left = nonlinear3(Circulating_dat, aopt_dat_Left, bopt_dat_Left,

plt.plot(Circulating_dat, y_fit_dat_Left, color = 'blue', label = 'Nonlinea
plt.plot(Circulating_dat, Left_dat, 'b.', label = 'Nieuwerkerk data left ac
plt.legend(loc = 'best')
plt.ylim([0,800])
plt.xlim([790,1810])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane [pce/h]');

GOFdatnon_Left = np.zeros(len(Left_dat))
for i in range(len(Left_dat)):
    GOFdatnon_Left[i] = Left_dat[i] - y_fit_dat_Left[i]

GOFdatnon_pos_Left = np.sqrt((GOFdatnon_Left)**2)

GOFdatnon_tot_Left = 0
for i in range (len(GOFdatnon_pos_Left)):
    GOFdatnon_tot_Left = GOFdatnon_tot_Left + GOFdatnon_pos_Left[i]

GOFdatnon_def_Left = GOFdatnon_tot_Left/(len(GOFdatnon_pos_Left))
print('The goodness of fit for the linear regression of the simulated data
```

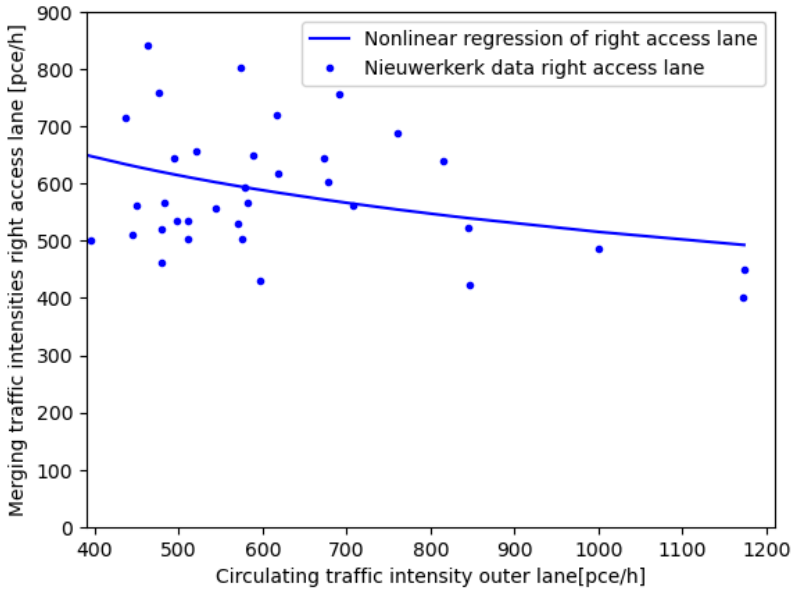
```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 56.
97414256517324
```



Existing data right lane

In [8]:
```
reg_dat_Right , pcov_dat_Right = sc.optimize.curve_fit(nonlinear3, Outer_da
aopt_dat_Right, bopt_dat_Right, copt_dat_Right = reg_dat_Right

y_fit_dat_Right = nonlinear3(Outer_dat_sorted, aopt_dat_Right, bopt_dat_Rig


plt.plot(Outer_dat_sorted, y_fit_dat_Right, color = 'blue', label = 'Nonlin
plt.plot(Outer_dat_sorted, Right_dat, 'b.', label = 'Nieuwerkerk data right
plt.legend(loc = 'best')
plt.xlim([390,1210])
plt.ylim([0,900])
plt.xlabel('Circulating traffic intensity outer lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');

GOFdatnon_Right = np.zeros(len(Right_dat_sorted))
for i in range(len(Right_dat_sorted)):
    GOFdatnon_Right[i] = Right_dat_sorted[i] - y_fit_dat_Right[i]

GOFdatnon_pos_Right = np.sqrt((GOFdatnon_Right)**2)

GOFdatnon_tot_Right = 0
for i in range (len(GOFdatnon_pos_Right)):
    GOFdatnon_tot_Right = GOFdatnon_tot_Right + GOFdatnon_pos_Right[i]

GOFdatnon_def_Right = GOFdatnon_tot_Right/(len(GOFdatnon_pos_Right))
print('The goodness of fit for the linear regression of the simulated data
```

```
The goodness of fit for the linear regression of the simulated data is 89.
51654458025757
```

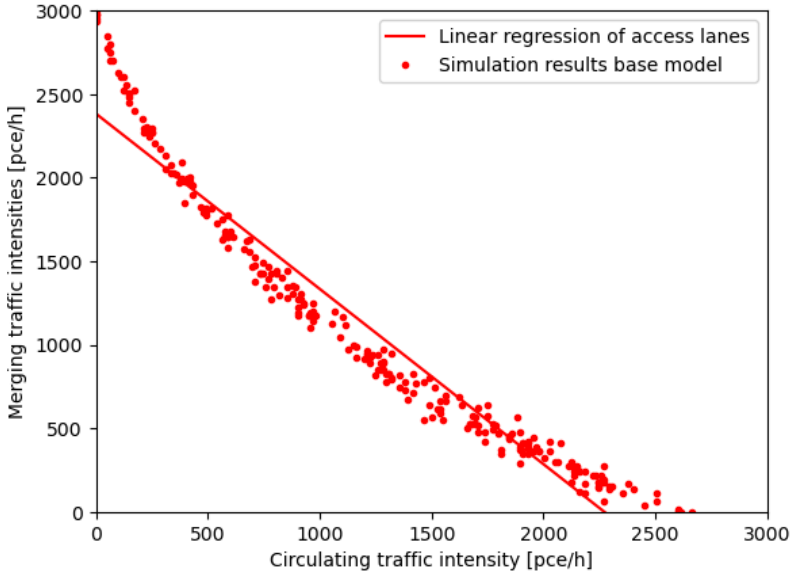


Data both lanes aggregated

Linear regression

In [9]:
```
rsim_vol = sc.stats.linregress(Circul_vol, Enter_vol,alternative = 'less')
slope_sim_vol = rsim_vol.slope
intercept_sim_vol = rsim_vol.intercept
xreg_sim_vol = np.linspace(0,2650,255)
yreg_sim_vol = intercept_sim_vol + slope_sim_vol*xreg_sim_vol

plt.plot(xreg_sim_vol,yreg_sim_vol,color = 'red', label = 'Linear regressio
plt.plot(Circul_vol, Enter_vol, 'r.', label = 'Simulation results base mode
plt.legend(loc = 'best')
plt.xlim([0,3000])
plt.ylim([0,3000])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimlin = np.zeros(len(Enter_vol))
for i in range(len(Enter_vol)):
    GOFsimlin[i] = Enter_vol[i] - (intercept_sim_vol + slope_sim_vol *Circu

GOFsimlin_pos = np.sqrt((GOFsimlin)**2)

GOFsimlin_tot = 0
for i in range (len(GOFsimlin_pos)):
    GOFsimlin_tot = GOFsimlin_tot + GOFsimlin_pos[i]

GOFsimlin_def = GOFsimlin_tot/(len(GOFsimlin_pos))
print('The goodness of fit for the linear regression of the simulated data
```

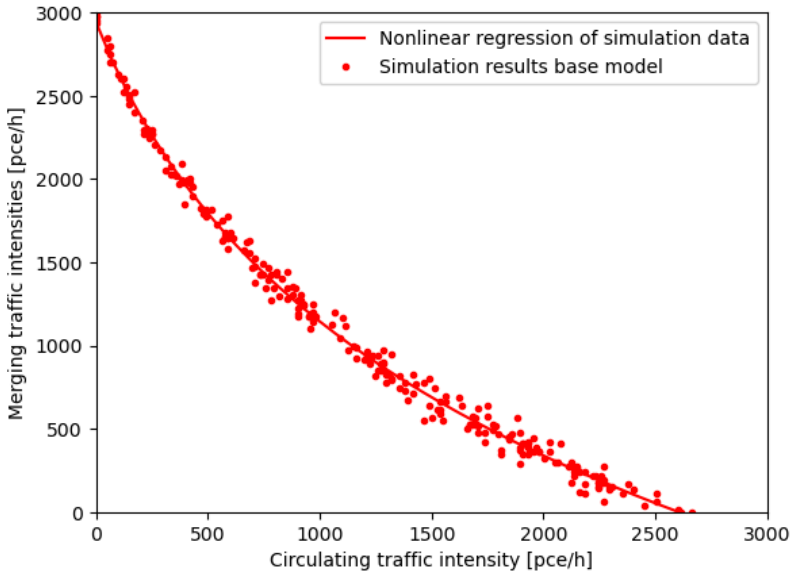
```
The goodness of fit for the linear regression of the simulated data is 15
2.70180301646064
```



Nonlinear regression

In [10]:
```
reg_sim_vol , pcov_sim_vol = sc.optimize.curve_fit(nonlinear3, Circul_vol,E
aopt_sim_vol, bopt_sim_vol, copt_sim_vol = reg_sim_vol

y_fit_sim_vol = nonlinear3(Circul_vol, aopt_sim_vol, bopt_sim_vol, copt_sim

plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Nonlinear regre
plt.plot(Circul_vol, Enter_vol, 'r.', label = 'Simulation results base mode
plt.legend(loc = 'best')
plt.xlim([0,3000])
plt.ylim([0,3000])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimnon = np.zeros(len(Enter_vol))
for i in range(len(Enter_vol)):
    GOFsimnon[i] = Enter_vol[i] - y_fit_sim_vol[i]

GOFsimnon_pos = np.sqrt((GOFsimnon)**2)

GOFsimnon_tot = 0
for i in range (len(GOFsimnon_pos)):
    GOFsimnon_tot = GOFsimnon_tot + GOFsimnon_pos[i]

GOFsimnon_def = GOFsimnon_tot/(len(GOFsimnon_pos))
print('The goodness of fit for the linear regression of the simulated data
```

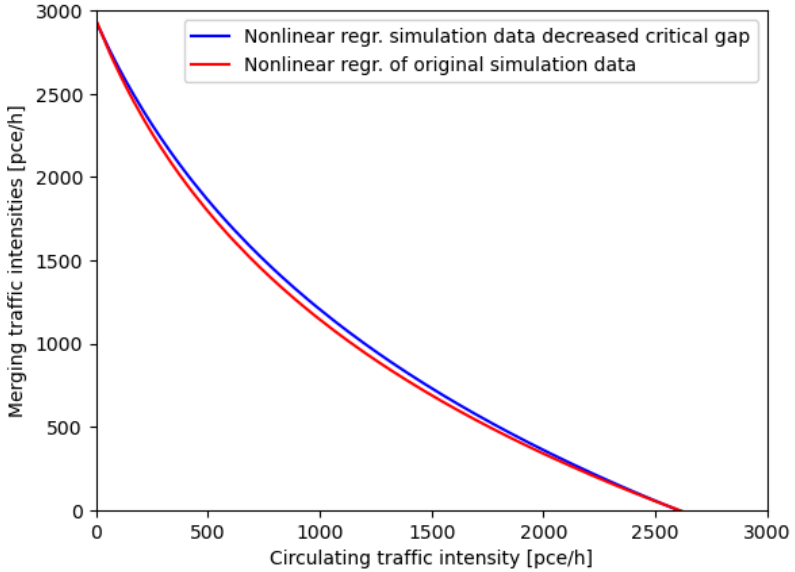
```
The goodness of fit for the linear regression of the simulated data is 41.
444650636123384

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



Decreased critical gap

In [11]:
```
reg_sim_crit_vol , pcov_sim_crit_vol = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_crit_vol, bopt_sim_crit_vol, copt_sim_crit_vol = reg_sim_crit_vol

y_fit_sim_crit_vol = nonlinear3(Circul_crit_vol, aopt_sim_crit_vol, bopt_si

GOFsimnon_crit_vol = np.zeros(len(Enter_crit_vol))
for i in range(len(Enter_crit_vol)):
    GOFsimnon_crit_vol[i] = Enter_crit_vol[i] - y_fit_sim_crit_vol[i]

GOFsimnon_pos_crit_vol = np.sqrt((GOFsimnon_crit_vol)**2)

GOFsimnon_tot_crit_vol = 0
for i in range (len(GOFsimnon_pos_crit_vol)):
    GOFsimnon_tot_crit_vol = GOFsimnon_tot_crit_vol + GOFsimnon_pos_crit_vo

GOFsimnon_def_crit_vol = GOFsimnon_tot_crit_vol/(len(GOFsimnon_pos_crit_vol
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_crit_vol, y_fit_sim_crit_vol, color = 'blue', label = 'Nonl
plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Nonlinear regr.
plt.legend(loc='best')
plt.xlim([0,3000])
plt.ylim([0,3000])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 40.
19820856146249
```



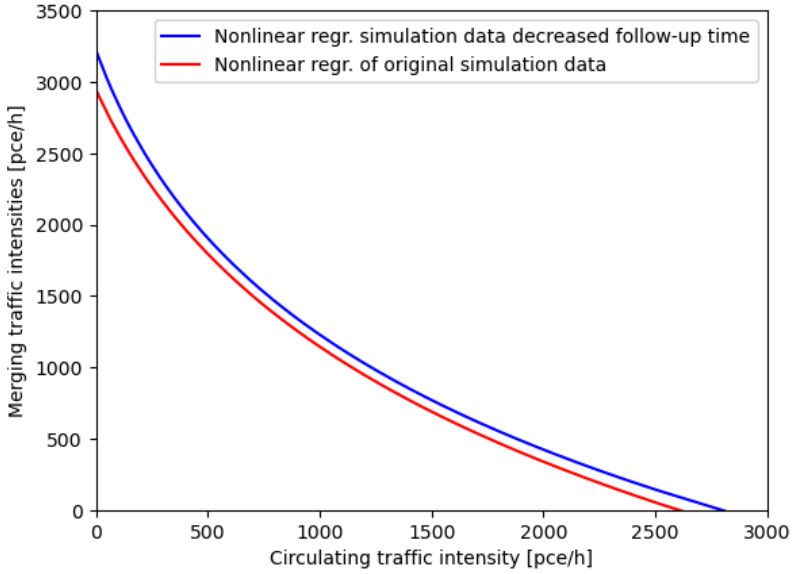
In [12]:
```
y2_interp_crit = np.interp(Circul_vol,Circul_crit_vol,y_fit_sim_crit_vol)
absolute_crit = np.abs(y2_interp_crit - y_fit_sim_vol)
average_crit = (y2_interp_crit)
percentage_crit = (absolute_crit/average_crit)*100

q1_crit=np.percentile(percentage_crit,25)
q3_crit=np.percentile(percentage_crit,75)
iqr_crit = q3_crit-q1_crit
outliers_crit = (percentage_crit < q1_crit-1.5*iqr_crit) |  (percentage_cri
percentage_crit_filt = percentage_crit[~outliers_crit]
print('The difference between the regular simulation and the critical simul
```

```
The difference between the regular simulation and the critical simulation
is 4.8303854900184575
```

Decreased follow-up time

In [13]:
```
reg_sim_fol_vol , pcov_sim_fol_vol = sc.optimize.curve_fit(nonlinear3, Circ
aopt_sim_fol_vol, bopt_sim_fol_vol, copt_sim_fol_vol = reg_sim_fol_vol

y_fit_sim_fol_vol = nonlinear3(Circul_fol_vol, aopt_sim_fol_vol, bopt_sim_f


#plt.plot(Circul_fol_vol, y_fit_sim_fol_vol, color = 'red', label = 'Nonlin
#plt.plot(Circul_fol_vol, Enter_fol_vol, 'r.', label = 'Simulation results
#plt.legend(loc = 'best')
#plt.xlim([0,2600])
#plt.ylim([0,2800])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimnon_fol_vol = np.zeros(len(Enter_fol_vol))
for i in range(len(Enter_fol_vol)):
    GOFsimnon_fol_vol[i] = Enter_fol_vol[i] - y_fit_sim_fol_vol[i]

GOFsimnon_pos_fol_vol = np.sqrt((GOFsimnon_fol_vol)**2)

GOFsimnon_tot_fol_vol = 0
for i in range (len(GOFsimnon_pos_fol_vol)):
    GOFsimnon_tot_fol_vol = GOFsimnon_tot_fol_vol + GOFsimnon_pos_fol_vol[i

GOFsimnon_def_fol_vol = GOFsimnon_tot_fol_vol/(len(GOFsimnon_pos_fol_vol))
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_fol_vol, y_fit_sim_fol_vol, color = 'blue', label = 'Nonlin
plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Nonlinear regr.
plt.legend(loc='best')
plt.xlim([0,3000])
plt.ylim([0,3500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

```
The goodness of fit for the linear regression of the simulated data is 44.
42722517657883

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



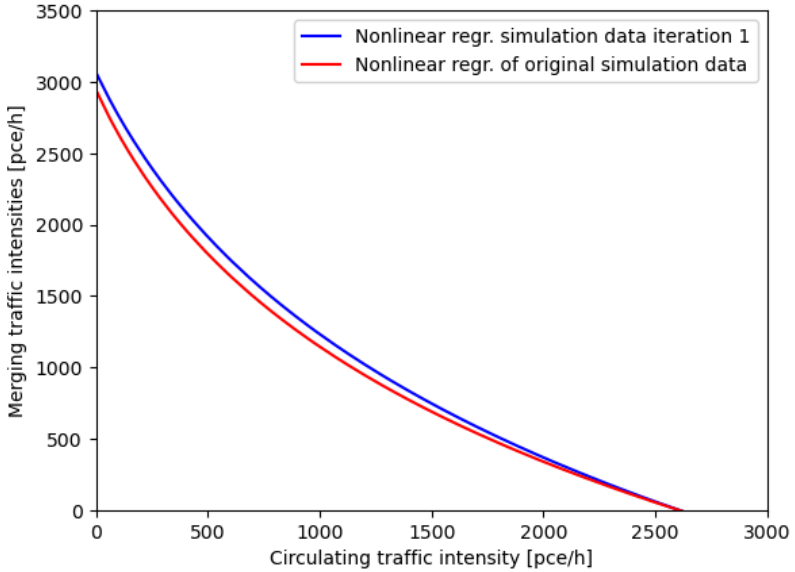
In [14]:
```
x_RMNSE_vol = np.linspace(0,2650,107)
y_fit_sim_fol_RMNSE_vol = nonlinear3(x_RMNSE_vol,aopt_sim_fol_vol, bopt_sim
y_fit_sim_RMNSE_vol = nonlinear3(x_RMNSE_vol,aopt_sim_vol, bopt_sim_vol, co

diff_non_fol_vol = (y_fit_sim_fol_RMNSE_vol - y_fit_sim_RMNSE_vol)/y_fit_si
diff_non_fol_vol_sum = 0
for i in range(0,len(diff_non_fol_vol)):
    diff_non_fol_vol_sum = diff_non_fol_vol_sum + diff_non_fol_vol[i]

#print('The difference between the regular simulation and the critical simu

y2_interp_fol = np.interp(Circul_vol,Circul_fol_vol,y_fit_sim_fol_vol)
absolute_fol = np.abs(y2_interp_fol - y_fit_sim_vol)
average_fol = (y2_interp_fol)
percentage_fol = (absolute_fol/y2_interp_fol)*100

q1_fol = np.percentile(percentage_fol,25)
q3_fol = np.percentile(percentage_fol,75)
iqr_fol=q3_fol-q1_fol
rule_fol = iqr_fol*1.5
outliers_fol = (percentage_fol < q1_fol-1.5*iqr_fol) |  (percentage_fol > q
percentage_fol_filt = percentage_fol[~outliers_fol]
print('The difference between the regular simulation and the critical simul
```

```
The difference between the regular simulation and the critical simulation
is 10.391057336812809
```

Iteration 1

In [15]:
```
reg_sim_it1 , pcov_sim_it1 = sc.optimize.curve_fit(nonlinear3, Circul_it1,E
aopt_sim_it1, bopt_sim_it1, copt_sim_it1 = reg_sim_it1

y_fit_sim_it1 = nonlinear3(Circul_it1, aopt_sim_it1, bopt_sim_it1, copt_sim


#plt.plot(Circul_it1, y_fit_sim_it1, color = 'red', label = 'Nonlinear regr
#plt.plot(Circul_it1, Enter_it1, 'r.', label = 'Simulation results decrease
#plt.legend(loc = 'best')
#plt.xlim([0,2600])
#plt.ylim([0,2800])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimnon_it1 = np.zeros(len(Enter_it1))
for i in range(len(Enter_it1)):
    GOFsimnon_it1[i] = Enter_it1[i] - y_fit_sim_it1[i]

GOFsimnon_pos_it1 = np.sqrt((GOFsimnon_it1)**2)

GOFsimnon_tot_it1 = 0
for i in range (len(GOFsimnon_pos_it1)):
    GOFsimnon_tot_it1 = GOFsimnon_tot_it1 + GOFsimnon_pos_it1[i]

GOFsimnon_def_it1 = GOFsimnon_tot_it1/(len(GOFsimnon_pos_it1))
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it1, y_fit_sim_it1, color = 'blue', label = 'Nonlinear regr
plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Nonlinear regr.
plt.legend(loc='best')
plt.xlim([0,3000])
plt.ylim([0,3500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

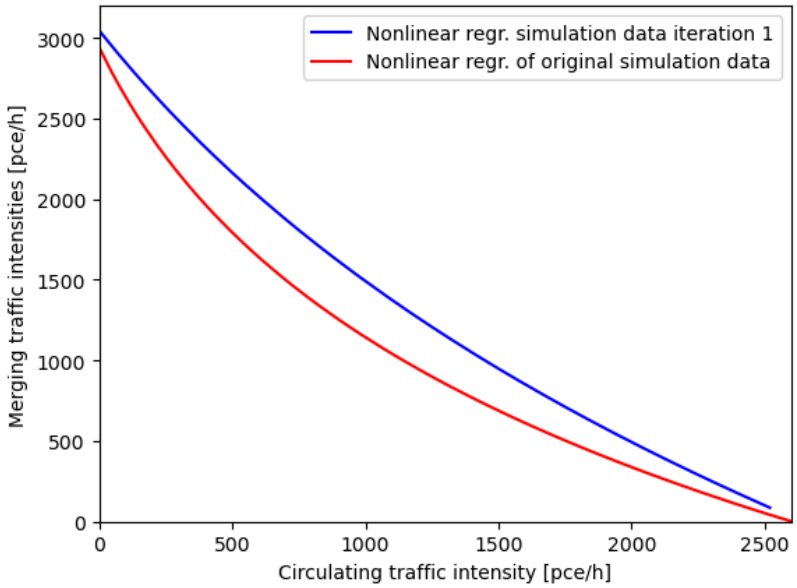
```
The goodness of fit for the linear regression of the simulated data is 42.
95698127202557

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



Iteration 2

In [16]:
```
reg_sim_it2 , pcov_sim_it2 = sc.optimize.curve_fit(nonlinear3, Circul_it2,E
aopt_sim_it2, bopt_sim_it2, copt_sim_it2 = reg_sim_it2

y_fit_sim_it2 = nonlinear3(Circul_it2, aopt_sim_it2, bopt_sim_it2, copt_sim


#plt.plot(Circul_it1, y_fit_sim_it1, color = 'red', label = 'Nonlinear regr
#plt.plot(Circul_it1, Enter_it1, 'r.', label = 'Simulation results decrease
#plt.legend(loc = 'best')
#plt.xlim([0,2600])
#plt.ylim([0,2800])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimnon_it2 = np.zeros(len(Enter_it2))
for i in range(len(Enter_it2)):
    GOFsimnon_it2[i] = Enter_it2[i] - y_fit_sim_it2[i]

GOFsimnon_pos_it2 = np.sqrt((GOFsimnon_it2)**2)

GOFsimnon_tot_it2 = 0
for i in range (len(GOFsimnon_pos_it2)):
    GOFsimnon_tot_it2 = GOFsimnon_tot_it2 + GOFsimnon_pos_it2[i]

GOFsimnon_def_it2 = GOFsimnon_tot_it2/(len(GOFsimnon_pos_it2))
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it2, y_fit_sim_it2, color = 'blue', label = 'Nonlinear regr
plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Nonlinear regr.
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,3200])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

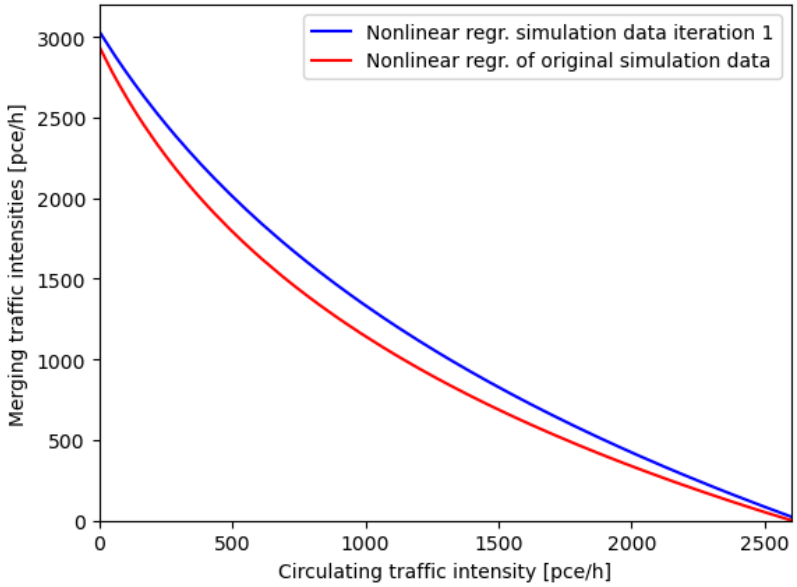
```
The goodness of fit for the linear regression of the simulated data is 58.
63853836709054

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



Iteration 3

In [17]:
```
reg_sim_it3 , pcov_sim_it3 = sc.optimize.curve_fit(nonlinear3, Circul_it3,E
aopt_sim_it3, bopt_sim_it3, copt_sim_it3 = reg_sim_it3

y_fit_sim_it3 = nonlinear3(Circul_it3, aopt_sim_it3, bopt_sim_it3, copt_sim


#plt.plot(Circul_it1, y_fit_sim_it1, color = 'red', label = 'Nonlinear regr
#plt.plot(Circul_it1, Enter_it1, 'r.', label = 'Simulation results decrease
#plt.legend(loc = 'best')
#plt.xlim([0,2600])
#plt.ylim([0,2800])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimnon_it3 = np.zeros(len(Enter_it3))
for i in range(len(Enter_it3)):
    GOFsimnon_it3[i] = Enter_it3[i] - y_fit_sim_it3[i]

GOFsimnon_pos_it3 = np.sqrt((GOFsimnon_it3)**2)

GOFsimnon_tot_it3 = 0
for i in range (len(GOFsimnon_pos_it3)):
    GOFsimnon_tot_it3 = GOFsimnon_tot_it3 + GOFsimnon_pos_it3[i]

GOFsimnon_def_it3 = GOFsimnon_tot_it3/(len(GOFsimnon_pos_it3))
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it3, y_fit_sim_it3, color = 'blue', label = 'Nonlinear regr
plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Nonlinear regr.
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,3200])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

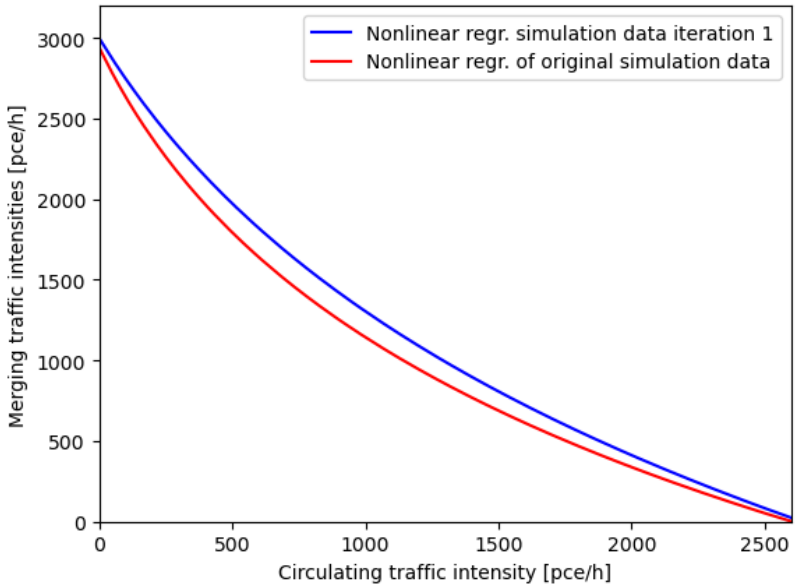
```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 48.
135438898492744
```



Iteration 4

In [18]:
```
reg_sim_it4 , pcov_sim_it4 = sc.optimize.curve_fit(nonlinear3, Circul_it4,E
aopt_sim_it4, bopt_sim_it4, copt_sim_it4 = reg_sim_it4

y_fit_sim_it4 = nonlinear3(Circul_it4, aopt_sim_it4, bopt_sim_it4, copt_sim


#plt.plot(Circul_it1, y_fit_sim_it1, color = 'red', label = 'Nonlinear regr
#plt.plot(Circul_it1, Enter_it1, 'r.', label = 'Simulation results decrease
#plt.legend(loc = 'best')
#plt.xlim([0,2600])
#plt.ylim([0,2800])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimnon_it4 = np.zeros(len(Enter_it4))
for i in range(len(Enter_it4)):
    GOFsimnon_it4[i] = Enter_it4[i] - y_fit_sim_it4[i]

GOFsimnon_pos_it4 = np.sqrt((GOFsimnon_it4)**2)

GOFsimnon_tot_it4 = 0
for i in range (len(GOFsimnon_pos_it4)):
    GOFsimnon_tot_it4 = GOFsimnon_tot_it4 + GOFsimnon_pos_it4[i]

GOFsimnon_def_it4 = GOFsimnon_tot_it4/(len(GOFsimnon_pos_it4))
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it4, y_fit_sim_it4, color = 'blue', label = 'Nonlinear regr
plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Nonlinear regr.
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,3200])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

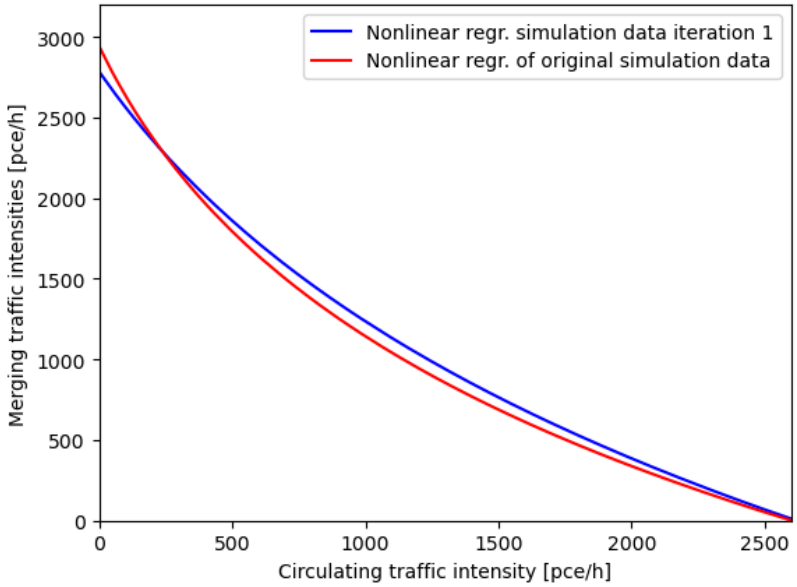
```
The goodness of fit for the linear regression of the simulated data is 46.
938451048944

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



Iteration 5

In [19]:
```
reg_sim_it5 , pcov_sim_it5 = sc.optimize.curve_fit(nonlinear3, Circul_it5,E
aopt_sim_it5, bopt_sim_it5, copt_sim_it5 = reg_sim_it5

y_fit_sim_it5 = nonlinear3(Circul_it5, aopt_sim_it5, bopt_sim_it5, copt_sim


#plt.plot(Circul_it1, y_fit_sim_it1, color = 'red', label = 'Nonlinear regr
#plt.plot(Circul_it1, Enter_it1, 'r.', label = 'Simulation results decrease
#plt.legend(loc = 'best')
#plt.xlim([0,2600])
#plt.ylim([0,2800])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimnon_it5 = np.zeros(len(Enter_it5))
for i in range(len(Enter_it5)):
    GOFsimnon_it5[i] = Enter_it5[i] - y_fit_sim_it5[i]

GOFsimnon_pos_it5 = np.sqrt((GOFsimnon_it5)**2)

GOFsimnon_tot_it5 = 0
for i in range (len(GOFsimnon_pos_it5)):
    GOFsimnon_tot_it5 = GOFsimnon_tot_it5 + GOFsimnon_pos_it5[i]

GOFsimnon_def_it5 = GOFsimnon_tot_it5/(len(GOFsimnon_pos_it5))
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it5, y_fit_sim_it5, color = 'blue', label = 'Nonlinear regr
plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Nonlinear regr.
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,3200])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

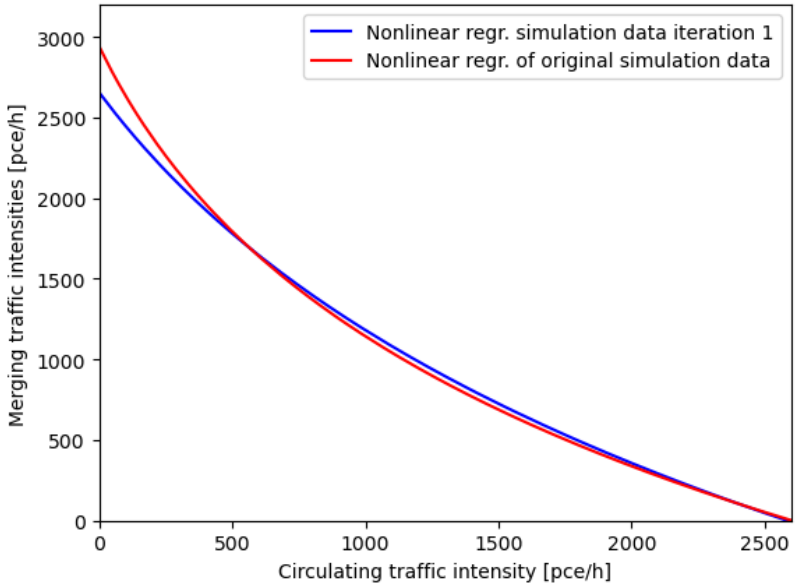
```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 42.
38387430458887
```



Iteration 6

In [20]:
```
reg_sim_it6 , pcov_sim_it6 = sc.optimize.curve_fit(nonlinear3, Circul_it6,E
aopt_sim_it6, bopt_sim_it6, copt_sim_it6 = reg_sim_it6

y_fit_sim_it6 = nonlinear3(Circul_it6, aopt_sim_it6, bopt_sim_it6, copt_sim


#plt.plot(Circul_it1, y_fit_sim_it1, color = 'red', label = 'Nonlinear regr
#plt.plot(Circul_it1, Enter_it1, 'r.', label = 'Simulation results decrease
#plt.legend(loc = 'best')
#plt.xlim([0,2600])
#plt.ylim([0,2800])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimnon_it6 = np.zeros(len(Enter_it6))
for i in range(len(Enter_it6)):
    GOFsimnon_it6[i] = Enter_it6[i] - y_fit_sim_it6[i]

GOFsimnon_pos_it6 = np.sqrt((GOFsimnon_it6)**2)

GOFsimnon_tot_it6 = 0
for i in range (len(GOFsimnon_pos_it6)):
    GOFsimnon_tot_it6 = GOFsimnon_tot_it6 + GOFsimnon_pos_it6[i]

GOFsimnon_def_it6 = GOFsimnon_tot_it6/(len(GOFsimnon_pos_it6))
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it6, y_fit_sim_it6, color = 'blue', label = 'Nonlinear regr
plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Nonlinear regr.
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,3200])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

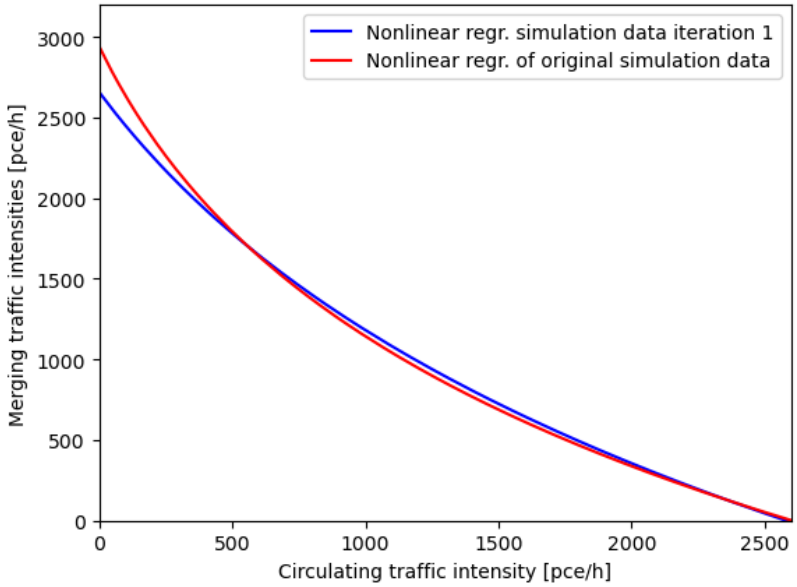
```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 39.
41667139844261
```



Iteration 7

In [21]:
```
reg_sim_it7 , pcov_sim_it7 = sc.optimize.curve_fit(nonlinear3, Circul_it7,E
aopt_sim_it7, bopt_sim_it7, copt_sim_it7 = reg_sim_it7

y_fit_sim_it7 = nonlinear3(Circul_it7, aopt_sim_it7, bopt_sim_it7, copt_sim


#plt.plot(Circul_it1, y_fit_sim_it1, color = 'red', label = 'Nonlinear regr
#plt.plot(Circul_it1, Enter_it1, 'r.', label = 'Simulation results decrease
#plt.legend(loc = 'best')
#plt.xlim([0,2600])
#plt.ylim([0,2800])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimnon_it7 = np.zeros(len(Enter_it7))
for i in range(len(Enter_it7)):
    GOFsimnon_it7[i] = Enter_it7[i] - y_fit_sim_it7[i]

GOFsimnon_pos_it7 = np.sqrt((GOFsimnon_it7)**2)

GOFsimnon_tot_it7 = 0
for i in range (len(GOFsimnon_pos_it7)):
    GOFsimnon_tot_it7 = GOFsimnon_tot_it7 + GOFsimnon_pos_it7[i]

GOFsimnon_def_it7 = GOFsimnon_tot_it7/(len(GOFsimnon_pos_it7))
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it7, y_fit_sim_it7, color = 'blue', label = 'Nonlinear regr
plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Nonlinear regr.
plt.legend(loc='best')
plt.xlim([0,2600])
plt.ylim([0,3200])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```

```
The goodness of fit for the linear regression of the simulated data is 38.
22279087060038

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```

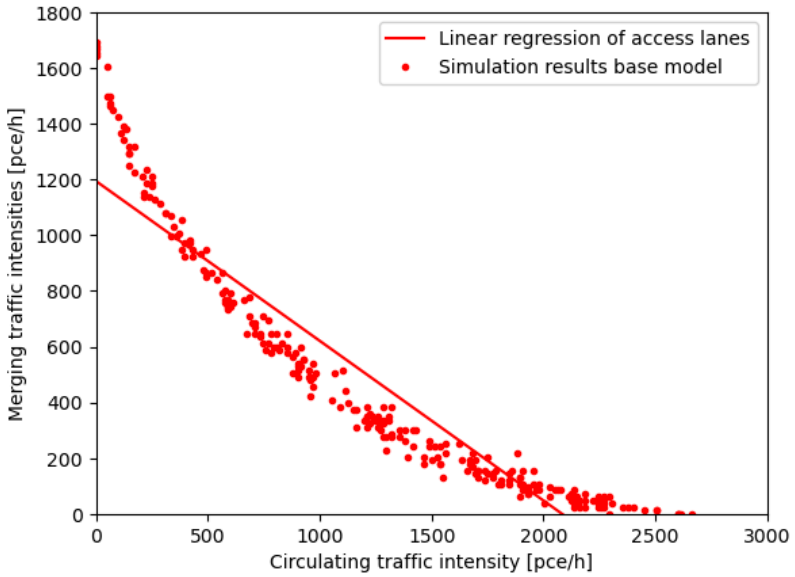


Simulation data left lane

Linear regression

In [22]:
```
rsim_vol_left = sc.stats.linregress(Circul_vol, Left_vol,alternative = 'les
slope_sim_vol_left = rsim_vol_left.slope
intercept_sim_vol_left = rsim_vol_left.intercept
xreg_sim_vol_left = np.linspace(0,2650,255)
yreg_sim_vol_left = intercept_sim_vol_left + slope_sim_vol_left*xreg_sim_vo

plt.plot(xreg_sim_vol_left,yreg_sim_vol_left,color = 'red', label = 'Linear
plt.plot(Circul_vol, Left_vol, 'r.', label = 'Simulation results base model
plt.legend(loc = 'best')
plt.xlim([0,3000])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');

GOFsimlin_left = np.zeros(len(Left_vol))
for i in range(len(Left_vol)):
    GOFsimlin_left[i] = Left_vol[i] - (intercept_sim_vol_left + slope_sim_v

GOFsimlin_pos_left = np.sqrt((GOFsimlin_left)**2)

GOFsimlin_tot_left = 0
for i in range (len(GOFsimlin_pos_left)):
    GOFsimlin_tot_left = GOFsimlin_tot_left + GOFsimlin_pos_left[i]

GOFsimlin_def_left = GOFsimlin_tot_left/(len(GOFsimlin_pos_left))
print('The goodness of fit for the linear regression of the simulated data
```

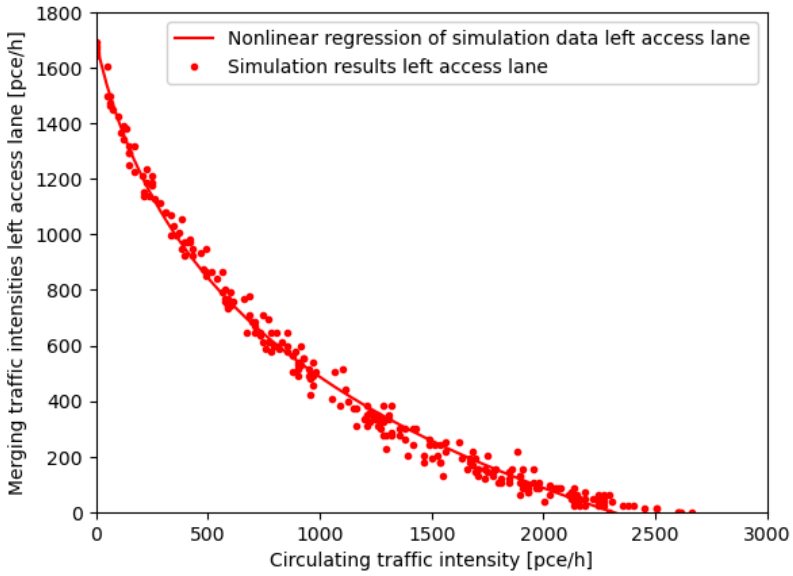
```
The goodness of fit for the linear regression of the simulated data is 12
0.58181919413113
```



Nonlinear regression

In [23]:
```
reg_sim_Left_vol , pcov_sim_Left_vol = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_Left_vol, bopt_sim_Left_vol, copt_sim_Left_vol = reg_sim_Left_vol

y_fit_sim_Left_vol = nonlinear3(Circul_vol, aopt_sim_Left_vol, bopt_sim_Lef

plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Nonlinear
plt.plot(Circul_vol, Left_vol, 'r.', label = 'Simulation results left acces
plt.legend(loc = 'best')
plt.xlim([0,3000])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane [pce/h]');

GOFsimnon = np.zeros(len(Enter_vol))
for i in range(len(Enter_vol)):
    GOFsimnon[i] = Left_vol[i] - y_fit_sim_Left_vol[i]

GOFsimnon_pos = np.sqrt((GOFsimnon)**2)

GOFsimnon_tot = 0
for i in range (len(GOFsimnon_pos)):
    GOFsimnon_tot = GOFsimnon_tot + GOFsimnon_pos[i]

GOFsimnon_def = GOFsimnon_tot/(len(GOFsimnon_pos))
print('The goodness of fit for the linear regression of the simulated data
```

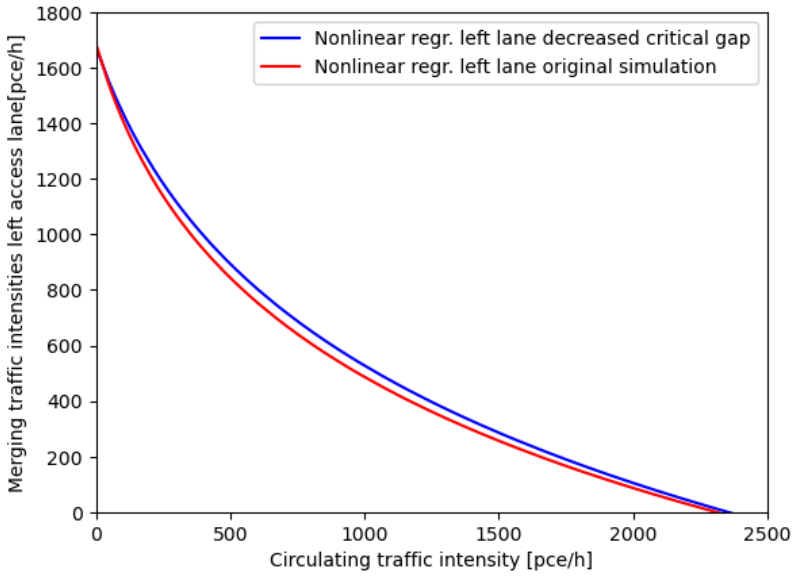
```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 32.
08676847366416
```



Decreased critical gap

In [24]:
```
reg_sim_Left_crit_vol , pcov_sim_Left_crit_vol = sc.optimize.curve_fit(nonl
aopt_sim_Left_crit_vol, bopt_sim_Left_crit_vol, copt_sim_Left_crit_vol = re

y_fit_sim_Left_crit_vol = nonlinear3(Circul_crit_vol, aopt_sim_Left_crit_vo


GOFsimnon_Left_crit_vol = np.zeros(len(Left_crit_vol))
for i in range(len(Left_crit_vol)):
    GOFsimnon_Left_crit_vol[i] = Left_crit_vol[i] - y_fit_sim_Left_crit_vol

GOFsimnon_pos_Left_crit = np.sqrt((GOFsimnon_Left_crit_vol)**2)

GOFsimnon_tot_Left_crit = 0
for i in range (len(GOFsimnon_pos_Left_crit)):
    GOFsimnon_tot_Left_crit = GOFsimnon_tot_Left_crit + GOFsimnon_pos_Left_

GOFsimnon_def_Left_crit = GOFsimnon_tot_Left_crit/(len(GOFsimnon_pos_Left_c
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_crit_vol, y_fit_sim_Left_crit_vol, color = 'blue', label =
plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Nonlinear
plt.legend(loc='best')
plt.xlim([0,2500])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane[pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 30.
22457707328368
```



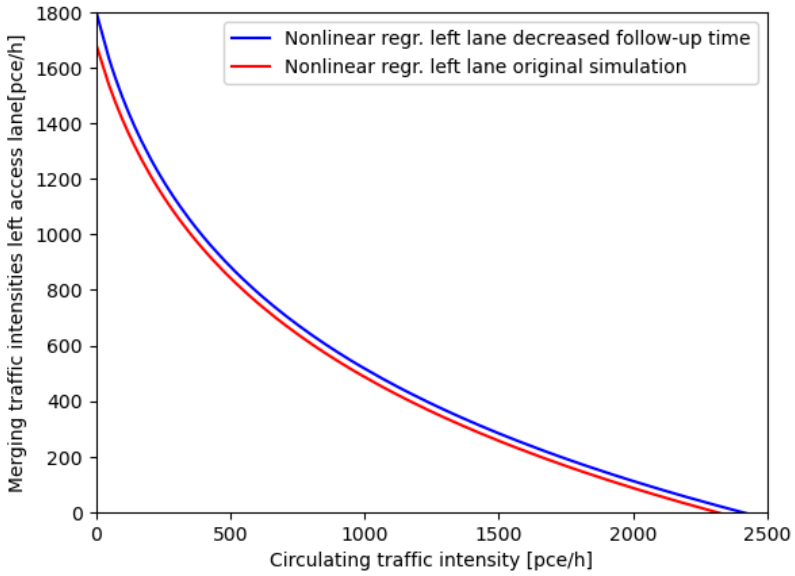
In [25]:
```
x_RMNSE_Left_vol = np.linspace(0,2500,101)
y_fit_sim_crit_RMNSE_Left_vol = nonlinear(x_RMNSE_Left_vol, aopt_sim_Left_c
y_fit_sim_RMNSE_Left_vol = nonlinear(x_RMNSE_Left_vol, aopt_sim_Left_vol, b

diff_non_crit_vol_Left = (y_fit_sim_crit_RMNSE_Left_vol - y_fit_sim_RMNSE_L
diff_non_crit_vol_Left_sum = 0
for i in range(0,len(diff_non_crit_vol_Left)):
    diff_non_crit_vol_Left_sum = diff_non_crit_vol_Left_sum + diff_non_crit

#print('The difference between the regular simulation and the critical simu

y2_interp_crit_left = np.interp(Circul_vol,Circul_crit_vol,y_fit_sim_Left_c
absolute_crit_left = np.abs(y2_interp_crit_left - y_fit_sim_Left_vol)
percentage_crit_left = (absolute_crit_left/y2_interp_crit_left)*100

q1_crit_left=np.percentile(percentage_crit_left,25)
q3_crit_left=np.percentile(percentage_crit_left,75)
iqr_crit_left = q3_crit_left-q1_crit_left
outliers_crit_left = (percentage_crit_left < q1_crit_left-1.5*iqr_crit_left
percentage_crit_left_filt = percentage_crit_left[~outliers_crit_left]
print('The difference between the regular simulation and the critical simul
```

```
The difference between the regular simulation and the critical simulation
is 8.260317890360392
```

Decreased follow-up time

In [26]:
```
reg_sim_Left_fol_vol , pcov_sim_Left_fol_vol = sc.optimize.curve_fit(nonlin
aopt_sim_Left_fol_vol, bopt_sim_Left_fol_vol, copt_sim_Left_fol_vol = reg_s

y_fit_sim_Left_fol_vol = nonlinear3(Circul_fol_vol, aopt_sim_Left_fol_vol,

GOFsimnon_Left_fol_vol = np.zeros(len(Left_fol_vol))
for i in range(len(Left_fol_vol)):
    GOFsimnon_Left_fol_vol[i] = Left_fol_vol[i] - y_fit_sim_Left_fol_vol[i]

GOFsimnon_pos_Left_fol_vol = np.sqrt((GOFsimnon_Left_fol_vol)**2)

GOFsimnon_tot_Left_fol_vol = 0
for i in range (len(GOFsimnon_pos_Left_fol_vol)):
    GOFsimnon_tot_Left_fol_vol = GOFsimnon_tot_Left_fol_vol + GOFsimnon_pos

GOFsimnon_def_Left_fol_vol = GOFsimnon_tot_Left_fol_vol/(len(GOFsimnon_pos_
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_fol_vol, y_fit_sim_Left_fol_vol, color = 'blue', label = 'N
plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Nonlinear
plt.legend(loc='best')
plt.xlim([0,2500])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane[pce/h]');
```

```
The goodness of fit for the linear regression of the simulated data is 34.
317801865988095

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



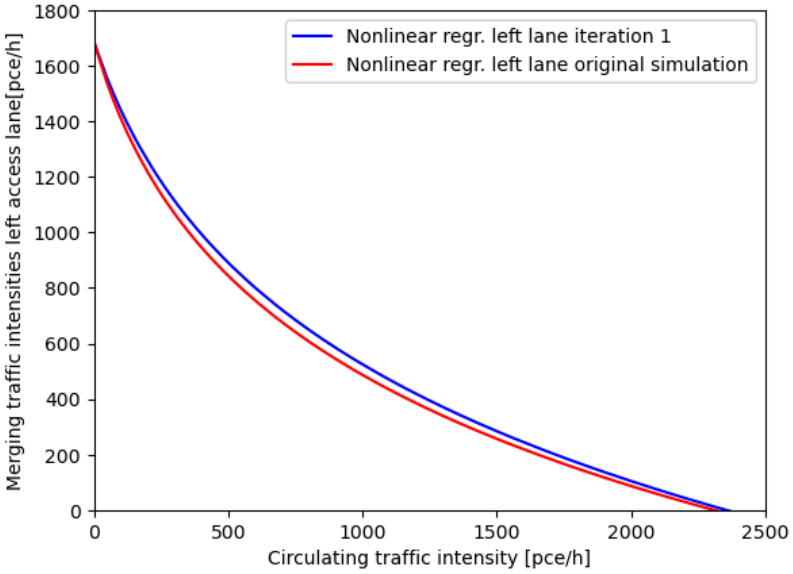
In [27]:
```
x_RMNSE_Left_vol = np.linspace(0,2500,101)
y_fit_sim_fol_RMNSE_Left_vol = nonlinear(x_RMNSE_Left_vol, aopt_sim_Left_fo
y_fit_sim_RMNSE_Left_vol = nonlinear(x_RMNSE_Left_vol, aopt_sim_Left_vol, b

diff_non_fol_vol_Left = (y_fit_sim_fol_RMNSE_Left_vol - y_fit_sim_RMNSE_Lef
diff_non_fol_vol_Left_sum = 0
for i in range(0,len(diff_non_fol_vol_Left)):
    diff_non_fol_vol_Left_sum = diff_non_fol_vol_Left_sum + diff_non_fol_vo

#print('The difference between the regular simulation and the critical simu

y2_interp_fol_left = np.interp(Circul_vol,Circul_fol_vol,y_fit_sim_Left_fol
absolute_fol_left = np.abs(y2_interp_fol_left - y_fit_sim_Left_vol)
percentage_fol_left = (absolute_fol_left/y2_interp_fol_left)*100

q1_fol_left=np.percentile(percentage_fol_left,25)
q3_fol_left=np.percentile(percentage_fol_left,75)
iqr_fol_left = q3_fol_left-q1_fol_left
outliers_fol_left = (percentage_fol_left < q1_fol_left-1.5*iqr_fol_left) |
percentage_fol_left_filt = percentage_fol_left[~outliers_fol_left]
print('The difference between the regular simulation and the critical simul
```

```
The difference between the regular simulation and the critical simulation
is 8.364685222248404
```

Iteration 1

In [28]:
```
reg_sim_Left_it1 , pcov_sim_Left_it1 = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_Left_it1, bopt_sim_Left_it1, copt_sim_Left_it1 = reg_sim_Left_it1

y_fit_sim_Left_it1 = nonlinear3(Circul_it1, aopt_sim_Left_it1, bopt_sim_Lef

GOFsimnon_Left_it1 = np.zeros(len(Left_it1))
for i in range(len(Left_it1)):
    GOFsimnon_Left_it1[i] = Left_it1[i] - y_fit_sim_Left_it1[i]

GOFsimnon_pos_Left_it1 = np.sqrt((GOFsimnon_Left_it1)**2)

GOFsimnon_tot_Left_it1 = 0
for i in range (len(GOFsimnon_pos_Left_it1)):
    GOFsimnon_tot_Left_it1 = GOFsimnon_tot_Left_it1 + GOFsimnon_pos_Left_it

GOFsimnon_def_Left_it1 = GOFsimnon_tot_Left_it1/(len(GOFsimnon_pos_Left_it1
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it1, y_fit_sim_Left_it1, color = 'blue', label = 'Nonlinear
plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Nonlinear
plt.legend(loc='best')
plt.xlim([0,2500])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane[pce/h]');
```

```
The goodness of fit for the linear regression of the simulated data is 31.
588847331743082

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```

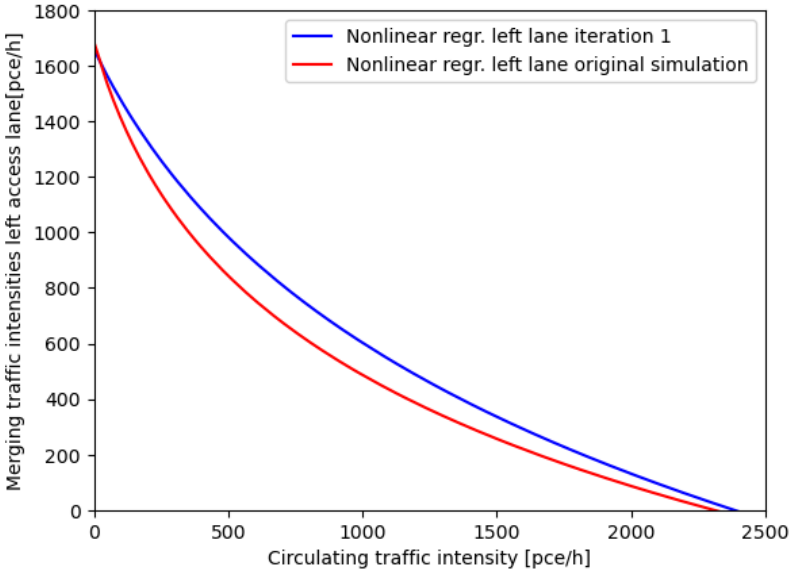


Iteration 2

In [29]:

```
reg_sim_Left_it2 , pcov_sim_Left_it2 = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_Left_it2, bopt_sim_Left_it2, copt_sim_Left_it2 = reg_sim_Left_it2

y_fit_sim_Left_it2 = nonlinear3(Circul_it2, aopt_sim_Left_it2, bopt_sim_Lef

GOFsimnon_Left_it2 = np.zeros(len(Left_it2))
for i in range(len(Left_it2)):
    GOFsimnon_Left_it2[i] = Left_it2[i] - y_fit_sim_Left_it2[i]

GOFsimnon_pos_Left_it2 = np.sqrt((GOFsimnon_Left_it2)**2)

GOFsimnon_tot_Left_it2 = 0
for i in range (len(GOFsimnon_pos_Left_it2)):
    GOFsimnon_tot_Left_it2 = GOFsimnon_tot_Left_it2 + GOFsimnon_pos_Left_it

GOFsimnon_def_Left_it2 = GOFsimnon_tot_Left_it2/(len(GOFsimnon_pos_Left_it2
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it2, y_fit_sim_Left_it2, color = 'blue', label = 'Nonlinear
plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Nonlinear
plt.legend(loc='best')
plt.xlim([0,2500])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane[pce/h]');
```

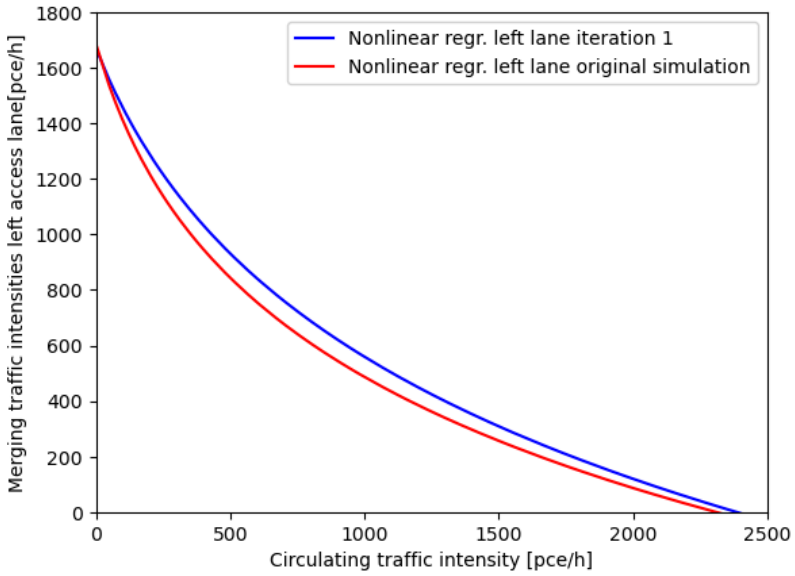
```
The goodness of fit for the linear regression of the simulated data is 27.
90743770329527

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



Iteration 3

In [30]:
```
reg_sim_Left_it3 , pcov_sim_Left_it3 = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_Left_it3, bopt_sim_Left_it3, copt_sim_Left_it3 = reg_sim_Left_it3

y_fit_sim_Left_it3 = nonlinear3(Circul_it3, aopt_sim_Left_it3, bopt_sim_Lef

GOFsimnon_Left_it3 = np.zeros(len(Left_it3))
for i in range(len(Left_it3)):
    GOFsimnon_Left_it3[i] = Left_it3[i] - y_fit_sim_Left_it3[i]

GOFsimnon_pos_Left_it3 = np.sqrt((GOFsimnon_Left_it3)**2)

GOFsimnon_tot_Left_it3 = 0
for i in range (len(GOFsimnon_pos_Left_it3)):
    GOFsimnon_tot_Left_it3 = GOFsimnon_tot_Left_it3 + GOFsimnon_pos_Left_it

GOFsimnon_def_Left_it3 = GOFsimnon_tot_Left_it3/(len(GOFsimnon_pos_Left_it3
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it3, y_fit_sim_Left_it3, color = 'blue', label = 'Nonlinear
plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Nonlinear
plt.legend(loc='best')
plt.xlim([0,2500])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane[pce/h]');
```

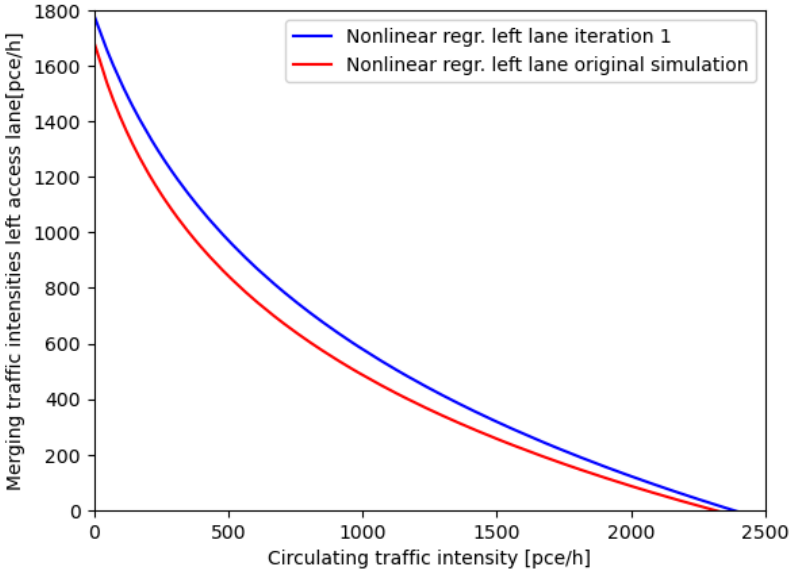
```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 30.
361327903939422
```



Iteration 4

In [31]:
```
reg_sim_Left_it4 , pcov_sim_Left_it4 = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_Left_it4, bopt_sim_Left_it4, copt_sim_Left_it4 = reg_sim_Left_it4

y_fit_sim_Left_it4 = nonlinear3(Circul_it4, aopt_sim_Left_it4, bopt_sim_Lef

GOFsimnon_Left_it4 = np.zeros(len(Left_it4))
for i in range(len(Left_it4)):
    GOFsimnon_Left_it4[i] = Left_it4[i] - y_fit_sim_Left_it4[i]

GOFsimnon_pos_Left_it4 = np.sqrt((GOFsimnon_Left_it4)**2)

GOFsimnon_tot_Left_it4 = 0
for i in range (len(GOFsimnon_pos_Left_it4)):
    GOFsimnon_tot_Left_it4 = GOFsimnon_tot_Left_it4 + GOFsimnon_pos_Left_it

GOFsimnon_def_Left_it4 = GOFsimnon_tot_Left_it4/(len(GOFsimnon_pos_Left_it4
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it4, y_fit_sim_Left_it4, color = 'blue', label = 'Nonlinear
plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Nonlinear
plt.legend(loc='best')
plt.xlim([0,2500])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane[pce/h]');
```

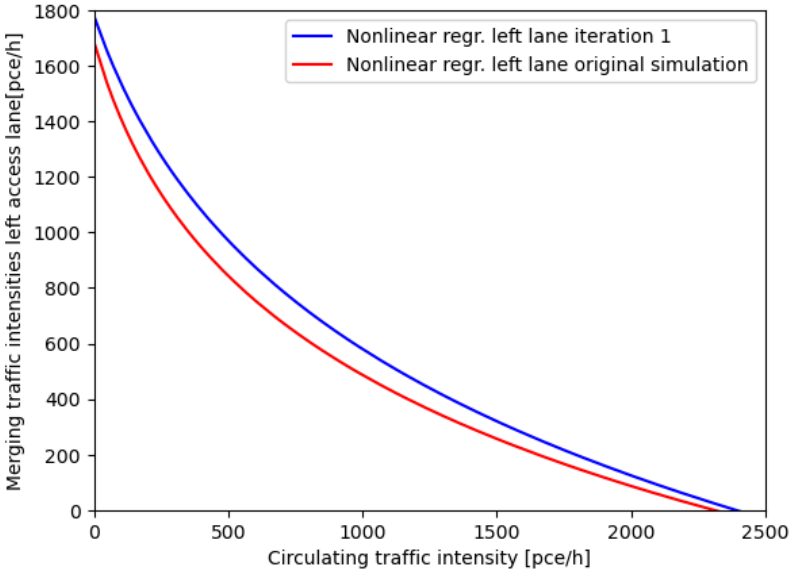
```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 32.
72430876217139
```



Iteration 5

In [32]:
```
reg_sim_Left_it5 , pcov_sim_Left_it5 = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_Left_it5, bopt_sim_Left_it5, copt_sim_Left_it5 = reg_sim_Left_it5

y_fit_sim_Left_it5 = nonlinear3(Circul_it5, aopt_sim_Left_it5, bopt_sim_Lef

GOFsimnon_Left_it5 = np.zeros(len(Left_it5))
for i in range(len(Left_it5)):
    GOFsimnon_Left_it5[i] = Left_it5[i] - y_fit_sim_Left_it5[i]

GOFsimnon_pos_Left_it5 = np.sqrt((GOFsimnon_Left_it5)**2)

GOFsimnon_tot_Left_it5 = 0
for i in range (len(GOFsimnon_pos_Left_it5)):
    GOFsimnon_tot_Left_it5 = GOFsimnon_tot_Left_it5 + GOFsimnon_pos_Left_it

GOFsimnon_def_Left_it5 = GOFsimnon_tot_Left_it5/(len(GOFsimnon_pos_Left_it5
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it5, y_fit_sim_Left_it5, color = 'blue', label = 'Nonlinear
plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Nonlinear
plt.legend(loc='best')
plt.xlim([0,2500])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane[pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 31.
47120725873459
```

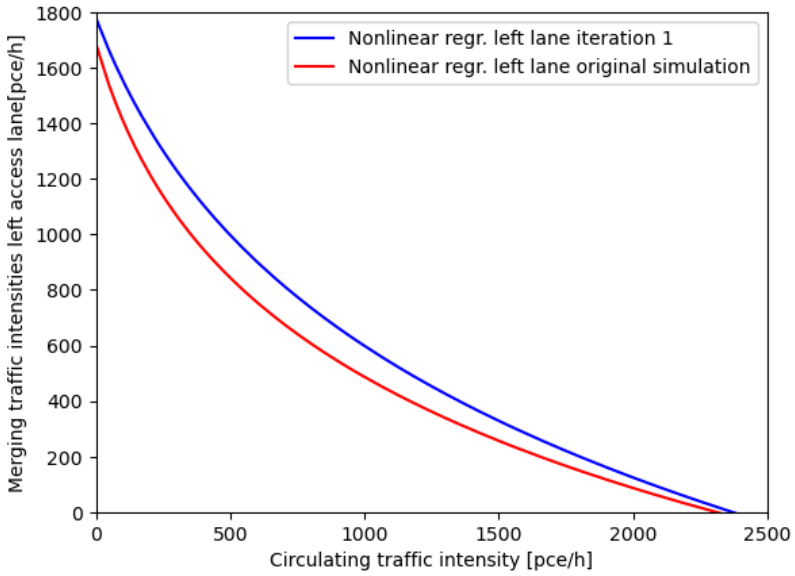


Iteration 6

In [33]:

```
reg_sim_Left_it6 , pcov_sim_Left_it6 = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_Left_it6, bopt_sim_Left_it6, copt_sim_Left_it6 = reg_sim_Left_it6

y_fit_sim_Left_it6 = nonlinear3(Circul_it6, aopt_sim_Left_it6, bopt_sim_Lef

GOFsimnon_Left_it6 = np.zeros(len(Left_it6))
for i in range(len(Left_it6)):
    GOFsimnon_Left_it6[i] = Left_it6[i] - y_fit_sim_Left_it6[i]

GOFsimnon_pos_Left_it6 = np.sqrt((GOFsimnon_Left_it6)**2)

GOFsimnon_tot_Left_it6 = 0
for i in range (len(GOFsimnon_pos_Left_it6)):
    GOFsimnon_tot_Left_it6 = GOFsimnon_tot_Left_it6 + GOFsimnon_pos_Left_it

GOFsimnon_def_Left_it6 = GOFsimnon_tot_Left_it6/(len(GOFsimnon_pos_Left_it6
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it6, y_fit_sim_Left_it6, color = 'blue', label = 'Nonlinear
plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Nonlinear
plt.legend(loc='best')
plt.xlim([0,2500])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane[pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 32.
893223804286706
```

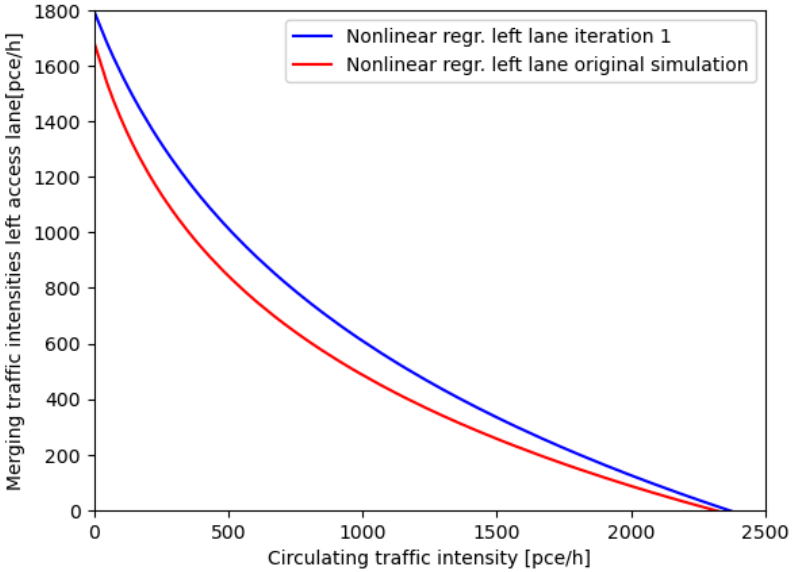


Iteration 7

In [34]:

```
reg_sim_Left_it7 , pcov_sim_Left_it7 = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_Left_it7, bopt_sim_Left_it7, copt_sim_Left_it7 = reg_sim_Left_it7

y_fit_sim_Left_it7 = nonlinear3(Circul_it7, aopt_sim_Left_it7, bopt_sim_Lef

GOFsimnon_Left_it7 = np.zeros(len(Left_it7))
for i in range(len(Left_it7)):
    GOFsimnon_Left_it7[i] = Left_it7[i] - y_fit_sim_Left_it7[i]

GOFsimnon_pos_Left_it7 = np.sqrt((GOFsimnon_Left_it7)**2)

GOFsimnon_tot_Left_it7 = 0
for i in range (len(GOFsimnon_pos_Left_it7)):
    GOFsimnon_tot_Left_it7 = GOFsimnon_tot_Left_it7 + GOFsimnon_pos_Left_it

GOFsimnon_def_Left_it7 = GOFsimnon_tot_Left_it7/(len(GOFsimnon_pos_Left_it7
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Circul_it7, y_fit_sim_Left_it7, color = 'blue', label = 'Nonlinear
plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Nonlinear
plt.legend(loc='best')
plt.xlim([0,2500])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane[pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 33.
24814526853394
```

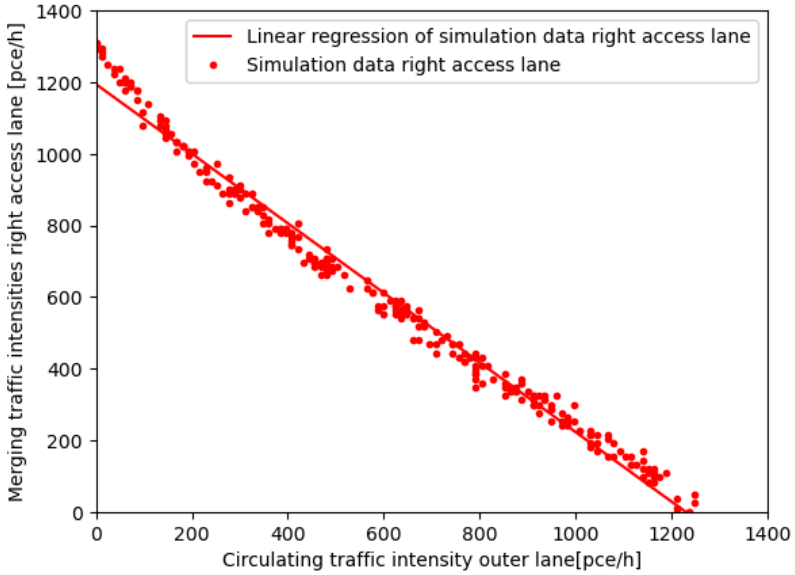


Right access lane simulation

Linear regression

In [35]:
```
rsim_Right = sc.stats.linregress(Outer_vol, Right_vol_sorted,alternative =
slope_sim_Right = rsim_Right.slope
intercept_sim_Right = rsim_Right.intercept
xreg_sim_Right = np.linspace(0,1250,41)
yreg_sim_Right = intercept_sim_Right + slope_sim_Right*xreg_sim_Right

plt.plot(xreg_sim_Right,yreg_sim_Right,color = 'red', label = 'Linear regre
plt.plot(Outer_vol, Right_vol_sorted, 'r.', label = 'Simulation data right
plt.xlim([0,1400])
plt.ylim([0,1400])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');

GOFsimlin_Right = np.zeros(len(Right_vol_sorted))
for i in range(len(Right_vol_sorted)):
    GOFsimlin_Right[i] = Right_vol_sorted[i] - (intercept_sim_Right + slope

GOFsimlin_pos_Right = np.sqrt((GOFsimlin_Right)**2)

GOFsimlin_tot_Right = 0
for i in range (len(GOFsimlin_pos_Right)):
    GOFsimlin_tot_Right = GOFsimlin_tot_Right + GOFsimlin_pos_Right[i]

GOFsimlin_def_Right = GOFsimlin_tot_Right/(len(GOFsimlin_pos_Right))
print('The goodness of fit for the linear regression of the simulated data
```

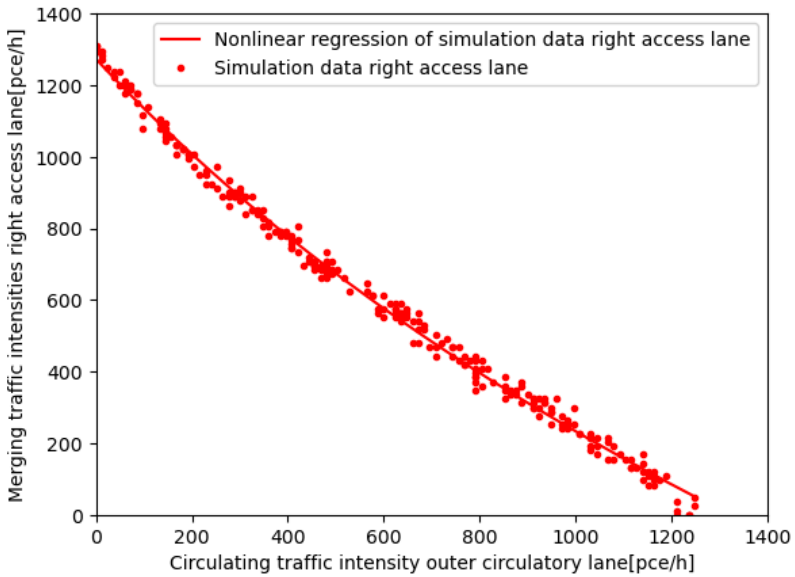
```
The goodness of fit for the linear regression of the simulated data is 31.
36738636680114
```



Non-linear regression

In [36]:
```
reg_sim_Right_vol , pcov_sim_Right_vol = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Right_vol, bopt_sim_Right_vol, copt_sim_Right_vol = reg_sim_Right_

y_fit_sim_Right = nonlinear3(Outer_vol, aopt_sim_Right_vol, bopt_sim_Right_
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Nonlinear regr
plt.plot(Outer_vol, Right_vol_sorted, 'r.', label = 'Simulation data right
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane[pce/h]')
plt.xlim([0,1400])
plt.ylim([0,1400])

GOFsimnon_Right_vol = np.zeros(len(Right_vol_sorted))
for i in range(len(Right_vol_sorted)):
    GOFsimnon_Right_vol[i] = Right_vol_sorted[i] - y_fit_sim_Right[i]

GOFsimnon_pos_Right_vol = np.sqrt((GOFsimnon_Right_vol)**2)

GOFsimnon_tot_Right_vol = 0
for i in range (len(GOFsimnon_pos_Right_vol)):
    GOFsimnon_tot_Right_vol = GOFsimnon_tot_Right_vol + GOFsimnon_pos_Right

GOFsimnon_def_Right_vol = GOFsimnon_tot_Right_vol/(len(GOFsimnon_pos_Right_
print('The goodness of fit for the linear regression of the simulated data
```

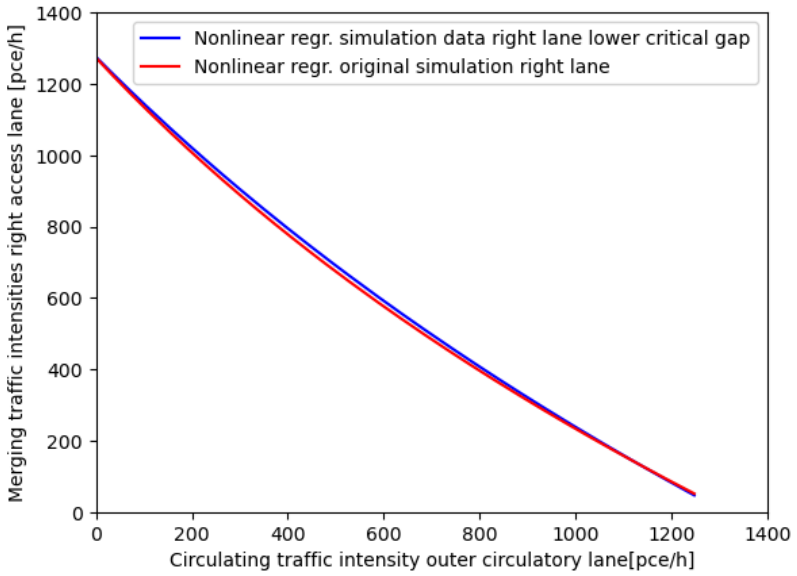
```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 20.
311036293435293
```



Decreased critical gap

In [37]:
```
reg_sim_Right_crit_vol , pcov_sim_Right_crit_vol = sc.optimize.curve_fit(no
aopt_sim_Right_crit_vol, bopt_sim_Right_crit_vol, copt_sim_Right_crit_vol =

y_fit_sim_Right_crit_vol = nonlinear3(Outer_crit_vol, aopt_sim_Right_crit_v

GOFsimnon_Right_crit = np.zeros(len(Right_crit_vol))
for i in range(len(Right_crit_vol)):
    GOFsimnon_Right_crit[i] = Right_crit_vol[i] - y_fit_sim_Right_crit_vol[

GOFsimnon_pos_Right_crit = np.sqrt((GOFsimnon_Right_crit)**2)

GOFsimnon_tot_Right_crit = 0
for i in range (len(GOFsimnon_pos_Right_crit)):
    GOFsimnon_tot_Right_crit = GOFsimnon_tot_Right_crit + GOFsimnon_pos_Rig

GOFsimnon_def_Right_crit = GOFsimnon_tot_Right_crit/(len(GOFsimnon_pos_Righ
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Outer_crit_vol, y_fit_sim_Right_crit_vol, color = 'blue', label =
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Nonlinear regr
plt.xlim([0,1400])
plt.ylim([0,1400])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 36.
18626217898652
```



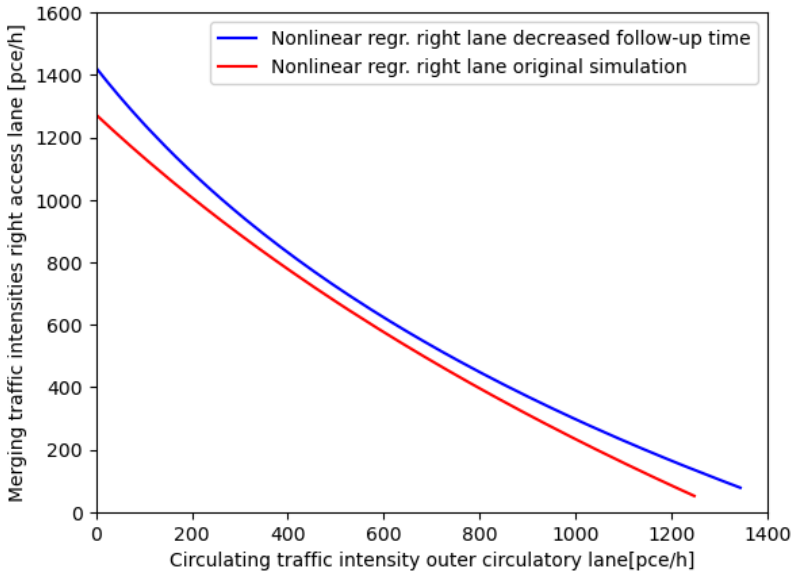
In [38]:
```
x_RMNSE_Right_vol = np.linspace(0,1250,51)
y_fit_sim_crit_RMNSE_Right_vol = nonlinear(x_RMNSE_Right_vol, aopt_sim_Righ
y_fit_sim_RMNSE_Right_vol = nonlinear(x_RMNSE_Right_vol, aopt_sim_Right_vol

diff_non_crit_vol_Right = (y_fit_sim_crit_RMNSE_Right_vol - y_fit_sim_RMNSE
diff_non_crit_vol_Right_sum = 0
for i in range(0,len(diff_non_crit_vol_Right)):
    diff_non_crit_vol_Right_sum = diff_non_crit_vol_Right_sum + diff_non_cr

#print('The difference between the regular simulation and the critical simu

y2_interp_crit_right = np.interp(Circul_vol,Circul_crit_vol,y_fit_sim_Right
absolute_crit_right = np.abs(y2_interp_crit_right - y_fit_sim_Right)
percentage_crit_right = (absolute_crit_right/y2_interp_crit_right)*100

q1_crit_right=np.percentile(percentage_crit_right,25)
q3_crit_right=np.percentile(percentage_crit_right,75)
iqr_crit_right = q3_crit_right-q1_crit_right
outliers_crit_right = (percentage_crit_right < q1_crit_right-1.5*iqr_crit_r
percentage_crit_right_filt = percentage_crit_right[~outliers_crit_right]
print('The difference between the regular simulation and the critical simul
```

```
The difference between the regular simulation and the critical simulation
is 1.7349418086490613
```

Decreased follow-up time

In [39]:
```
reg_sim_Right_fol_vol , pcov_sim_Right_fol_vol = sc.optimize.curve_fit(nonl
aopt_sim_Right_fol_vol, bopt_sim_Right_fol_vol, copt_sim_Right_fol_vol = re

y_fit_sim_Right_fol_vol = nonlinear3(Outer_fol_vol, aopt_sim_Right_fol_vol,

GOFsimnon_Right_fol = np.zeros(len(Right_fol_vol))
for i in range(len(Right_fol_vol)):
    GOFsimnon_Right_fol[i] = Right_fol_vol[i] - y_fit_sim_Right_fol_vol[i]

GOFsimnon_pos_Right_fol = np.sqrt((GOFsimnon_Right_fol)**2)

GOFsimnon_tot_Right_fol = 0
for i in range (len(GOFsimnon_pos_Right_fol)):
    GOFsimnon_tot_Right_fol = GOFsimnon_tot_Right_fol + GOFsimnon_pos_Right

GOFsimnon_def_Right_fol = GOFsimnon_tot_Right_fol/(len(GOFsimnon_pos_Right_
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Outer_fol_vol, y_fit_sim_Right_fol_vol, color = 'blue', label = 'N
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Nonlinear regr
plt.xlim([0,1400])
plt.ylim([0,1600])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');
```

```
The goodness of fit for the linear regression of the simulated data is 37.
89738519296798

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



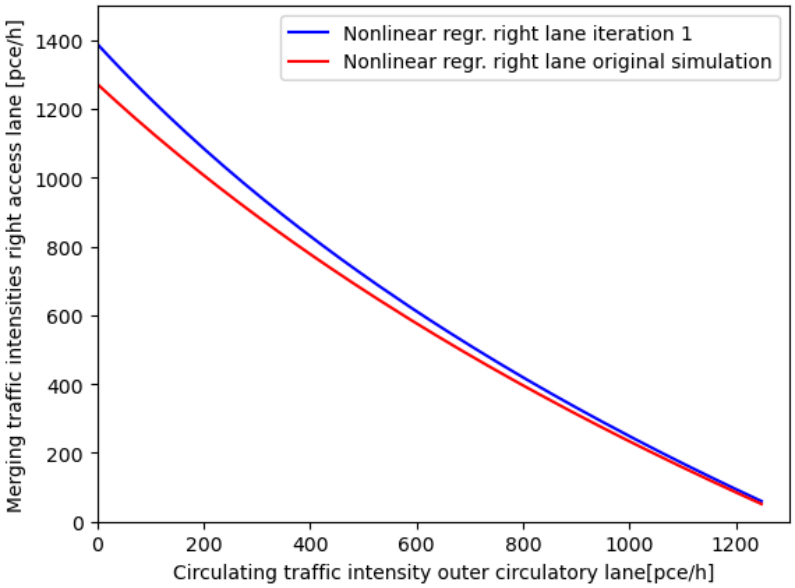
In [40]:
```
x_RMNSE_Right_vol = np.linspace(0,1250,51)
y_fit_sim_fol_RMNSE_Right_vol = nonlinear(x_RMNSE_Right_vol, aopt_sim_Right
y_fit_sim_RMNSE_Right_vol = nonlinear(x_RMNSE_Right_vol, aopt_sim_Right_vol

diff_non_fol_vol_Right = (y_fit_sim_fol_RMNSE_Right_vol - y_fit_sim_RMNSE_R
diff_non_fol_vol_Right_sum = 0
for i in range(0,len(diff_non_fol_vol_Right)):
    diff_non_fol_vol_Right_sum = diff_non_fol_vol_Right_sum + diff_non_fol_
#print('The difference between the regular simulation and the critical simu

y2_interp_fol_right = np.interp(Circul_vol,Circul_fol_vol,y_fit_sim_Right_f
absolute_fol_right = np.abs(y2_interp_fol_right - y_fit_sim_Right)
percentage_fol_right = (absolute_fol_right/y2_interp_fol_right)*100

q1_fol_right=np.percentile(percentage_fol_right,25)
q3_fol_right=np.percentile(percentage_fol_right,75)
iqr_fol_right = q3_fol_right-q1_fol_right
outliers_fol_right = (percentage_fol_right < q1_fol_right-1.5*iqr_fol_right
percentage_fol_right_filt = percentage_fol_right[~outliers_fol_right]
print('The difference between the regular simulation and the critical simul
```

```
The difference between the regular simulation and the critical simulation
is 10.42344605232326
```

Iteration 1

In [41]:
```
reg_sim_Right_it1 , pcov_sim_Right_it1 = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Right_it1, bopt_sim_Right_it1, copt_sim_Right_it1 = reg_sim_Right_

y_fit_sim_Right_it1 = nonlinear3(Outer_it1, aopt_sim_Right_it1, bopt_sim_Ri

#plt.plot(Outer_fol_vol, y_fit_sim_Right_fol_vol, color = 'red', label = 'N
#plt.plot(Outer_fol_vol, Right_fol_vol, 'r.', label = 'Simulation results r
#plt.legend(loc = 'best')
#plt.ylim([0,1500])
#plt.xlim([0,1300])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities right access lane [pce/h]');

GOFsimnon_Right_it1 = np.zeros(len(Right_it1))
for i in range(len(Right_it1)):
    GOFsimnon_Right_it1[i] = Right_it1[i] - y_fit_sim_Right_it1[i]

GOFsimnon_pos_Right_it1 = np.sqrt((GOFsimnon_Right_it1)**2)

GOFsimnon_tot_Right_it1 = 0
for i in range (len(GOFsimnon_pos_Right_it1)):
    GOFsimnon_tot_Right_it1 = GOFsimnon_tot_Right_it1 + GOFsimnon_pos_Right

GOFsimnon_def_Right_it1 = GOFsimnon_tot_Right_it1/(len(GOFsimnon_pos_Right_
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Outer_it1, y_fit_sim_Right_it1, color = 'blue', label = 'Nonlinear
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Nonlinear regr
plt.xlim([0,1300])
plt.ylim([0,1500])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');
```

```
The goodness of fit for the linear regression of the simulated data is 37.
720440963945094

C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```

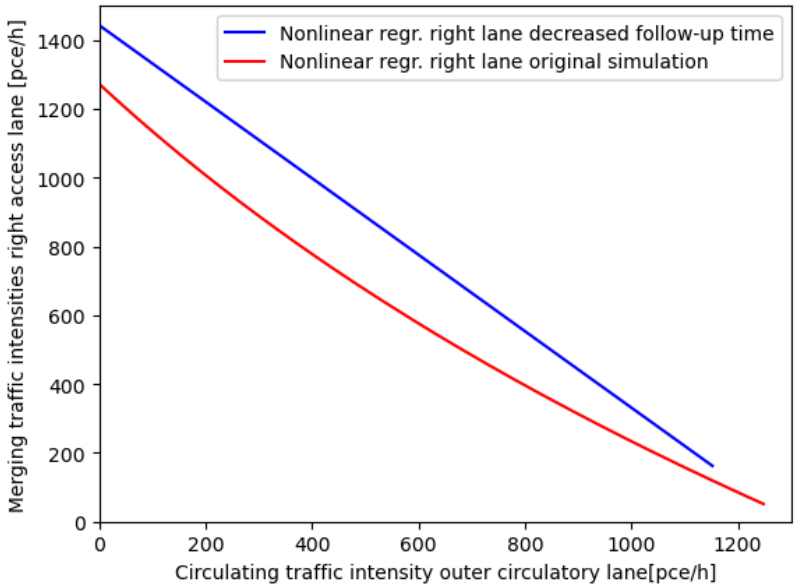


Iteration 2

In [42]:

```
reg_sim_Right_it2 , pcov_sim_Right_it2 = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Right_it2, bopt_sim_Right_it2, copt_sim_Right_it2 = reg_sim_Right_

y_fit_sim_Right_it2 = nonlinear3(Outer_it2, aopt_sim_Right_it2, bopt_sim_Ri

#plt.plot(Outer_fol_vol, y_fit_sim_Right_fol_vol, color = 'red', label = 'N
#plt.plot(Outer_fol_vol, Right_fol_vol, 'r.', label = 'Simulation results r
#plt.legend(loc = 'best')
#plt.ylim([0,1500])
#plt.xlim([0,1300])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities right access lane [pce/h]');

GOFsimnon_Right_it2 = np.zeros(len(Right_it2))
for i in range(len(Right_it2)):
    GOFsimnon_Right_it2[i] = Right_it2[i] - y_fit_sim_Right_it2[i]

GOFsimnon_pos_Right_it2 = np.sqrt((GOFsimnon_Right_it2)**2)

GOFsimnon_tot_Right_it2 = 0
for i in range (len(GOFsimnon_pos_Right_it2)):
    GOFsimnon_tot_Right_it2 = GOFsimnon_tot_Right_it2 + GOFsimnon_pos_Right

GOFsimnon_def_Right_it2 = GOFsimnon_tot_Right_it2/(len(GOFsimnon_pos_Right_
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Outer_it2, y_fit_sim_Right_it2, color = 'blue', label = 'Nonlinear
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Nonlinear regr
plt.xlim([0,1300])
plt.ylim([0,1500])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 45.
06226625439583
```

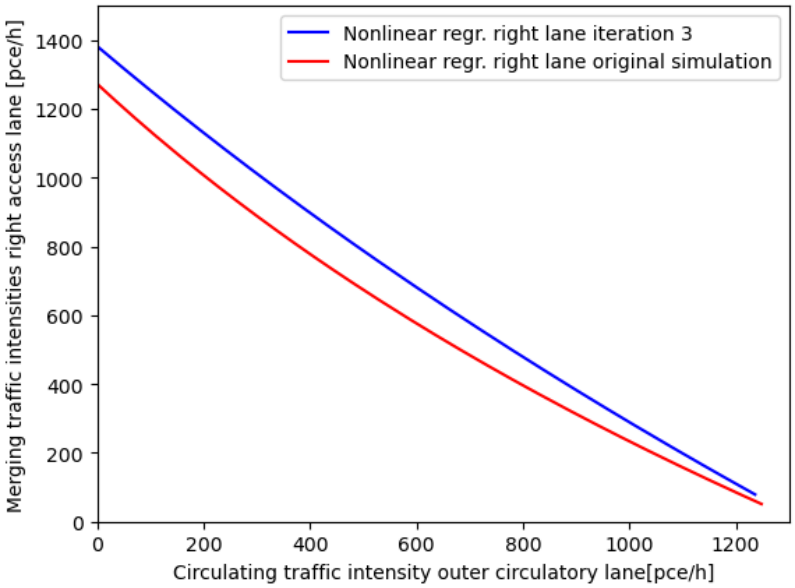


Iteration 3

In [43]:

```
reg_sim_Right_it3 , pcov_sim_Right_it3 = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Right_it3, bopt_sim_Right_it3, copt_sim_Right_it3 = reg_sim_Right_

y_fit_sim_Right_it3 = nonlinear3(Outer_it3, aopt_sim_Right_it3, bopt_sim_Ri

#plt.plot(Outer_fol_vol, y_fit_sim_Right_fol_vol, color = 'red', label = 'N
#plt.plot(Outer_fol_vol, Right_fol_vol, 'r.', label = 'Simulation results r
#plt.legend(loc = 'best')
#plt.ylim([0,1500])
#plt.xlim([0,1300])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities right access lane [pce/h]');

GOFsimnon_Right_it3 = np.zeros(len(Right_it3))
for i in range(len(Right_it3)):
    GOFsimnon_Right_it3[i] = Right_it3[i] - y_fit_sim_Right_it3[i]

GOFsimnon_pos_Right_it3 = np.sqrt((GOFsimnon_Right_it3)**2)

GOFsimnon_tot_Right_it3 = 0
for i in range (len(GOFsimnon_pos_Right_it3)):
    GOFsimnon_tot_Right_it3 = GOFsimnon_tot_Right_it3 + GOFsimnon_pos_Right

GOFsimnon_def_Right_it3 = GOFsimnon_tot_Right_it3/(len(GOFsimnon_pos_Right_
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Outer_it3, y_fit_sim_Right_it3, color = 'blue', label = 'Nonlinear
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Nonlinear regr
plt.xlim([0,1300])
plt.ylim([0,1500])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 39.
33952379557013
```

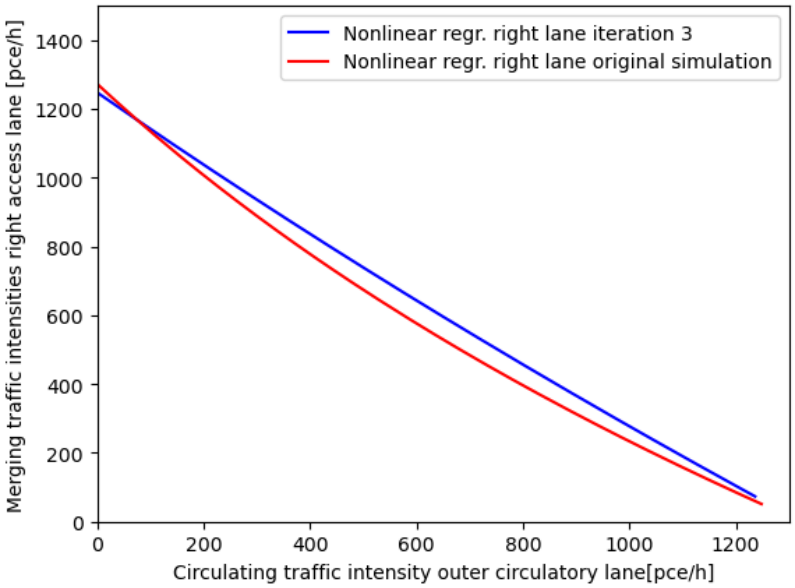


Iteration 4

In [44]:

```
reg_sim_Right_it4 , pcov_sim_Right_it4 = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Right_it4, bopt_sim_Right_it4, copt_sim_Right_it4 = reg_sim_Right_

y_fit_sim_Right_it4 = nonlinear3(Outer_it4, aopt_sim_Right_it4, bopt_sim_Ri

#plt.plot(Outer_fol_vol, y_fit_sim_Right_fol_vol, color = 'red', label = 'N
#plt.plot(Outer_fol_vol, Right_fol_vol, 'r.', label = 'Simulation results r
#plt.legend(loc = 'best')
#plt.ylim([0,1500])
#plt.xlim([0,1300])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities right access lane [pce/h]');

GOFsimnon_Right_it4 = np.zeros(len(Right_it4))
for i in range(len(Right_it4)):
    GOFsimnon_Right_it4[i] = Right_it4[i] - y_fit_sim_Right_it4[i]

GOFsimnon_pos_Right_it4 = np.sqrt((GOFsimnon_Right_it4)**2)

GOFsimnon_tot_Right_it4 = 0
for i in range (len(GOFsimnon_pos_Right_it4)):
    GOFsimnon_tot_Right_it4 = GOFsimnon_tot_Right_it4 + GOFsimnon_pos_Right

GOFsimnon_def_Right_it4 = GOFsimnon_tot_Right_it4/(len(GOFsimnon_pos_Right_
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Outer_it4, y_fit_sim_Right_it4, color = 'blue', label = 'Nonlinear
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Nonlinear regr
plt.xlim([0,1300])
plt.ylim([0,1500])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 38.
364479011698
```

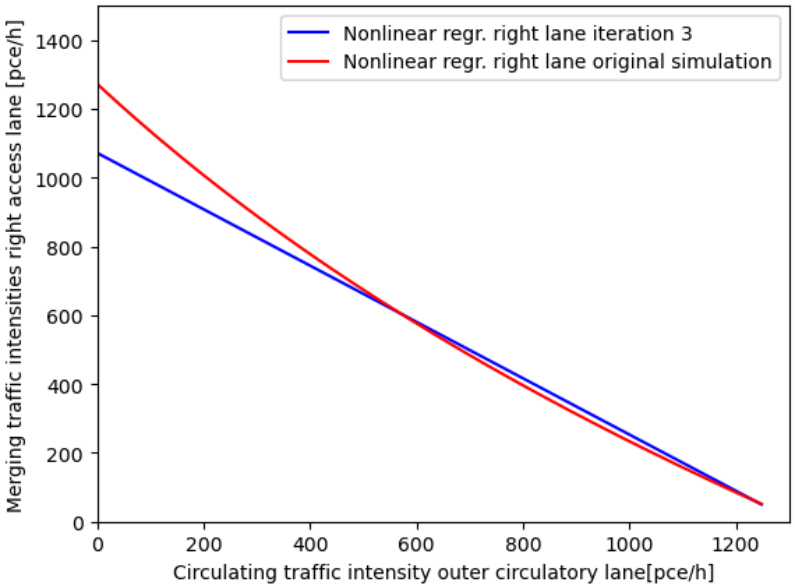


Iteration 5

In [45]:

```
reg_sim_Right_it5 , pcov_sim_Right_it5 = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Right_it5, bopt_sim_Right_it5, copt_sim_Right_it5 = reg_sim_Right_

y_fit_sim_Right_it5 = nonlinear3(Outer_it5, aopt_sim_Right_it5, bopt_sim_Ri

#plt.plot(Outer_fol_vol, y_fit_sim_Right_fol_vol, color = 'red', label = 'N
#plt.plot(Outer_fol_vol, Right_fol_vol, 'r.', label = 'Simulation results r
#plt.legend(loc = 'best')
#plt.ylim([0,1500])
#plt.xlim([0,1300])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities right access lane [pce/h]');

GOFsimnon_Right_it5 = np.zeros(len(Right_it5))
for i in range(len(Right_it5)):
    GOFsimnon_Right_it5[i] = Right_it5[i] - y_fit_sim_Right_it5[i]

GOFsimnon_pos_Right_it5 = np.sqrt((GOFsimnon_Right_it5)**2)

GOFsimnon_tot_Right_it5 = 0
for i in range (len(GOFsimnon_pos_Right_it5)):
    GOFsimnon_tot_Right_it5 = GOFsimnon_tot_Right_it5 + GOFsimnon_pos_Right

GOFsimnon_def_Right_it5 = GOFsimnon_tot_Right_it5/(len(GOFsimnon_pos_Right_
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Outer_it5, y_fit_sim_Right_it5, color = 'blue', label = 'Nonlinear
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Nonlinear regr
plt.xlim([0,1300])
plt.ylim([0,1500])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 32.
293087406361515
```

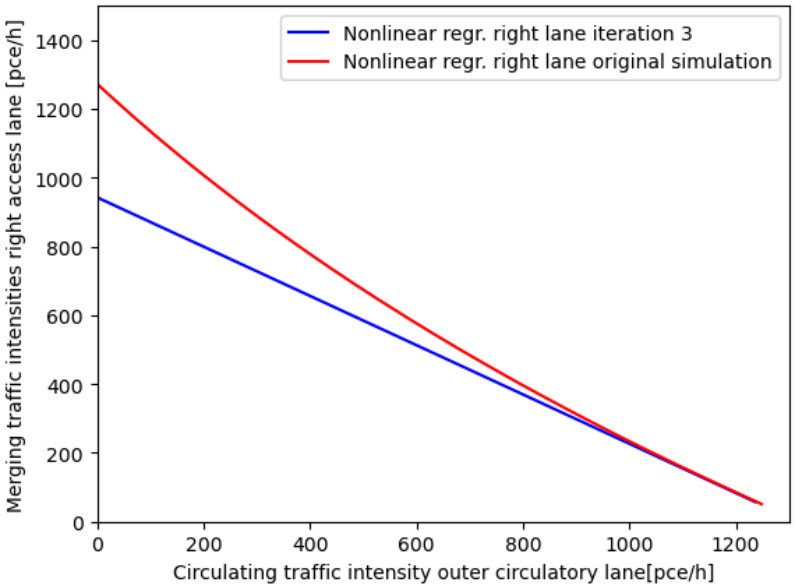


Iteration 6

In [46]:

```
reg_sim_Right_it6 , pcov_sim_Right_it6 = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Right_it6, bopt_sim_Right_it6, copt_sim_Right_it6 = reg_sim_Right_

y_fit_sim_Right_it6 = nonlinear3(Outer_it6, aopt_sim_Right_it6, bopt_sim_Ri

#plt.plot(Outer_fol_vol, y_fit_sim_Right_fol_vol, color = 'red', label = 'N
#plt.plot(Outer_fol_vol, Right_fol_vol, 'r.', label = 'Simulation results r
#plt.legend(loc = 'best')
#plt.ylim([0,1500])
#plt.xlim([0,1300])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities right access lane [pce/h]');

GOFsimnon_Right_it6 = np.zeros(len(Right_it6))
for i in range(len(Right_it6)):
    GOFsimnon_Right_it6[i] = Right_it6[i] - y_fit_sim_Right_it6[i]

GOFsimnon_pos_Right_it6 = np.sqrt((GOFsimnon_Right_it6)**2)

GOFsimnon_tot_Right_it6 = 0
for i in range (len(GOFsimnon_pos_Right_it6)):
    GOFsimnon_tot_Right_it6 = GOFsimnon_tot_Right_it6 + GOFsimnon_pos_Right

GOFsimnon_def_Right_it6 = GOFsimnon_tot_Right_it6/(len(GOFsimnon_pos_Right_
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Outer_it6, y_fit_sim_Right_it6, color = 'blue', label = 'Nonlinear
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Nonlinear regr
plt.xlim([0,1300])
plt.ylim([0,1500])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 31.
314716619618384
```

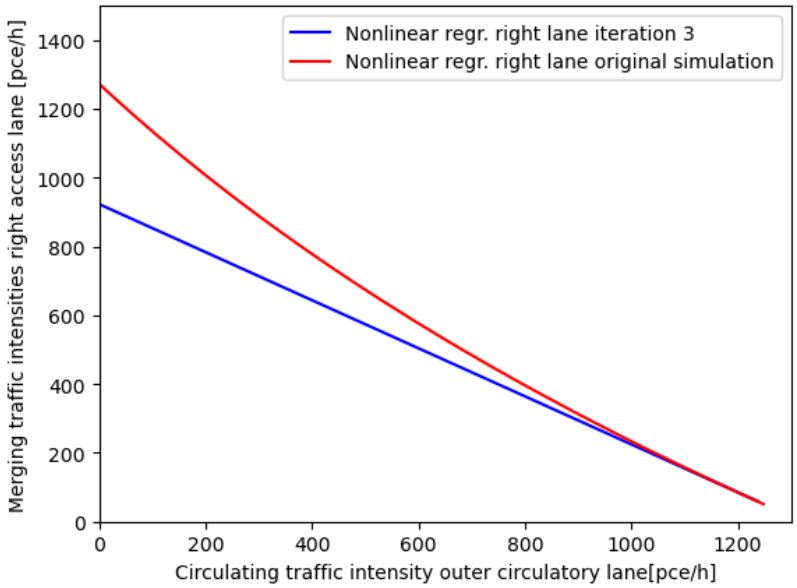


Iteration 7

In [47]:

```
reg_sim_Right_it7 , pcov_sim_Right_it7 = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Right_it7, bopt_sim_Right_it7, copt_sim_Right_it7 = reg_sim_Right_

y_fit_sim_Right_it7 = nonlinear3(Outer_it7, aopt_sim_Right_it7, bopt_sim_Ri

#plt.plot(Outer_fol_vol, y_fit_sim_Right_fol_vol, color = 'red', label = 'N
#plt.plot(Outer_fol_vol, Right_fol_vol, 'r.', label = 'Simulation results r
#plt.legend(loc = 'best')
#plt.ylim([0,1500])
#plt.xlim([0,1300])
#plt.xlabel('Circulating traffic intensity [pce/h]')
#plt.ylabel('Merging traffic intensities right access lane [pce/h]');

GOFsimnon_Right_it7 = np.zeros(len(Right_it7))
for i in range(len(Right_it7)):
    GOFsimnon_Right_it7[i] = Right_it7[i] - y_fit_sim_Right_it7[i]

GOFsimnon_pos_Right_it7 = np.sqrt((GOFsimnon_Right_it7)**2)

GOFsimnon_tot_Right_it7 = 0
for i in range (len(GOFsimnon_pos_Right_it7)):
    GOFsimnon_tot_Right_it7 = GOFsimnon_tot_Right_it7 + GOFsimnon_pos_Right

GOFsimnon_def_Right_it7 = GOFsimnon_tot_Right_it7/(len(GOFsimnon_pos_Right_
print('The goodness of fit for the linear regression of the simulated data

plt.plot(Outer_it7, y_fit_sim_Right_it7, color = 'blue', label = 'Nonlinear
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Nonlinear regr
plt.xlim([0,1300])
plt.ylim([0,1500])
plt.legend(loc = 'best')
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');
```

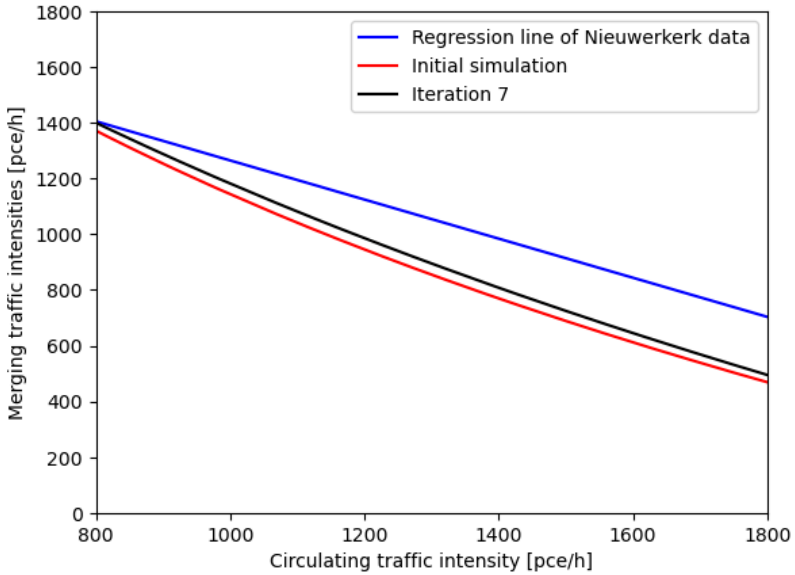
```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c

The goodness of fit for the linear regression of the simulated data is 28.
160694523243343
```



Comparison of graphs for validation

Aggregate

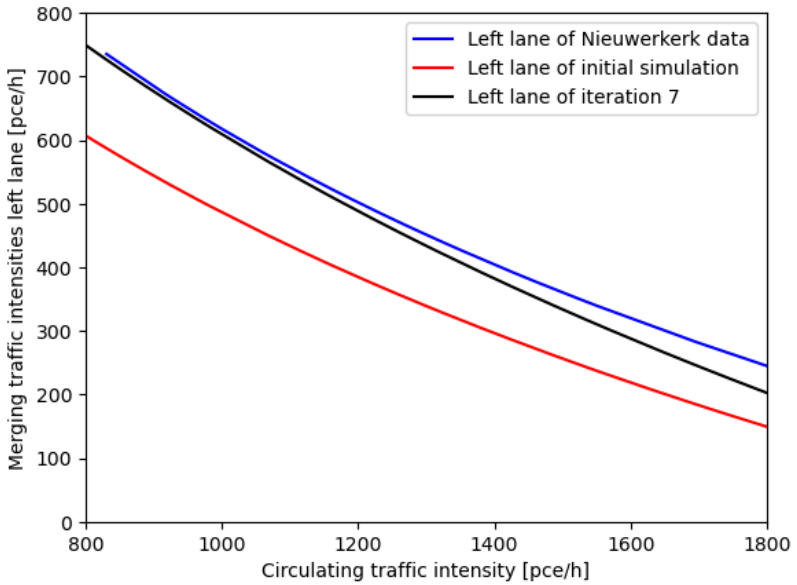
In [48]:
```
plt.plot(xreg_dat,yreg_dat, color = 'blue', label = 'Regression line of Nie
plt.plot(Circul_vol, y_fit_sim_vol, color = 'red', label = 'Initial simulat
#plt.plot(Circul_crit_vol, y_fit_sim_crit_vol, color = 'green', label = 'De
#plt.plot(Circul_fol_vol, y_fit_sim_fol_vol, color = 'orange', label = 'Dec
plt.plot(Circul_it7, y_fit_sim_it7,color = 'black', label = 'Iteration 7')
plt.legend(loc = 'best')
plt.xlim([800,1800])
plt.ylim([0,1800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities [pce/h]');
```



Left lane

In [49]:
```
plt.plot(Circulating_dat, y_fit_dat_Left, color = 'blue', label = 'Left lan
plt.plot(Circul_vol, y_fit_sim_Left_vol, color = 'red', label = 'Left lane
#plt.plot(Circul_crit_vol, y_fit_sim_Left_crit_vol, color = 'green', label
#plt.plot(Circul_fol_vol, y_fit_sim_Left_fol_vol, color = 'orange', label =
plt.plot(Circul_it7,y_fit_sim_Left_it7,color='black', label = 'Left lane of
plt.legend(loc = 'best')
plt.xlim([800,1800])
plt.ylim([0,800])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left lane [pce/h]');
print(y_fit_dat_Left[-1])
print(y_fit_sim_Left_it7[186])
print('The difference between both curves at 900 pce/h is',((y_fit_dat_Left
print('The difference between both curves at 1800 pce/h is',((y_fit_dat_Lef
```

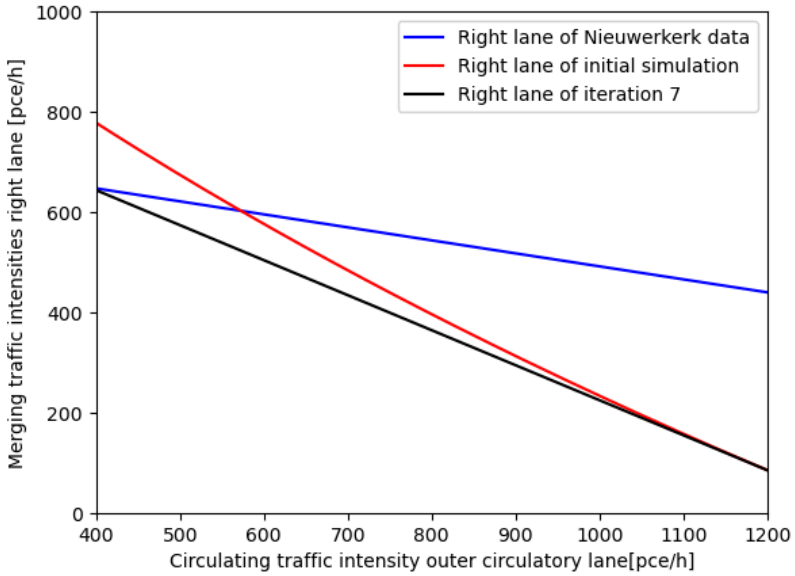
```
243.15041238334015
202.51743448580964
The difference between both curves at 900 pce/h is 0.7340153203494933 perc
ent
The difference between both curves at 1800 pce/h is 16.711046261139117 per
cent
```



Right lane

In [50]:
```
plt.plot(xreg_dat_Right,yreg_dat_Right, color = 'blue', label = 'Right lane
plt.plot(Outer_vol, y_fit_sim_Right, color = 'red', label = 'Right lane of
#plt.plot(Outer_crit_vol, y_fit_sim_Right_crit_vol, color = 'green', label
#plt.plot(Outer_fol_vol, y_fit_sim_Right_fol_vol, color = 'orange', label =
plt.plot(Outer_it7,y_fit_sim_Right_it7,color = 'black', label = 'Right lane
plt.legend(loc='best')
plt.xlim([400,1200])
plt.ylim([0,1000])
plt.xlabel('Circulating traffic intensity outer circulatory lane[pce/h]')
plt.ylabel('Merging traffic intensities right lane [pce/h]');
print(yreg_dat_Right[40])
print(y_fit_sim_Right[247])
print('The difference between both curves at 400 pce/h is',((yreg_dat_Right
print('The difference between both curves at 1200 pce/h is',((yreg_dat_Righ
```

```
439.7283483227423
76.58483227216493
The difference between both curves at 400 pce/h is 1.4240362262488089 perc
ent
The difference between both curves at 1200 pce/h is 82.5836035897429 perce
nt
```



Validation RMNSE

Aggregate

In [51]:
```
x = np.linspace(800,1800,41)
y_sim_agg = nonlinear3(x,aopt_sim_vol,bopt_sim_vol,copt_sim_vol)
y_sim_crit = nonlinear3(x,aopt_sim_crit_vol, bopt_sim_crit_vol, copt_sim_cr
y_sim_fol = nonlinear3(x,aopt_sim_fol_vol, bopt_sim_fol_vol, copt_sim_fol_v
y_sim_it1 = nonlinear3(x,aopt_sim_it1, bopt_sim_it1, copt_sim_it1)
y_sim_it2 = nonlinear3(x,aopt_sim_it2, bopt_sim_it2, copt_sim_it2)
y_sim_it3 = nonlinear3(x,aopt_sim_it3, bopt_sim_it3, copt_sim_it3)
y_sim_it4 = nonlinear3(x,aopt_sim_it4, bopt_sim_it4, copt_sim_it4)
y_sim_it5 = nonlinear3(x,aopt_sim_it5, bopt_sim_it5, copt_sim_it5)
y_sim_it6 = nonlinear3(x,aopt_sim_it6, bopt_sim_it6, copt_sim_it6)
y_sim_it7 = nonlinear3(x,aopt_sim_it7, bopt_sim_it7, copt_sim_it7)

diff_sim_agg = y_sim_agg - yreg_dat
diff_sim_agg_breuk = diff_sim_agg/yreg_dat
diff_sim_agg_pos = (diff_sim_agg_breuk)**2

diff_sim_crit = y_sim_crit - yreg_dat
diff_sim_crit_breuk = diff_sim_crit/yreg_dat
diff_sim_crit_pos = (diff_sim_crit_breuk)**2

diff_sim_fol = y_sim_fol - yreg_dat
diff_sim_fol_breuk = diff_sim_fol/yreg_dat
diff_sim_fol_pos = (diff_sim_fol_breuk)**2

diff_sim_it1 = y_sim_it1 - yreg_dat
diff_sim_it1_breuk = diff_sim_it1/yreg_dat
diff_sim_it1_pos = (diff_sim_it1_breuk)**2

diff_sim_it2 = y_sim_it2 - yreg_dat
diff_sim_it2_breuk = diff_sim_it2/yreg_dat
diff_sim_it2_pos = (diff_sim_it2_breuk)**2

diff_sim_it3 = y_sim_it3 - yreg_dat
diff_sim_it3_breuk = diff_sim_it3/yreg_dat
diff_sim_it3_pos = (diff_sim_it3_breuk)**2

diff_sim_it4 = y_sim_it4 - yreg_dat
diff_sim_it4_breuk = diff_sim_it4/yreg_dat
diff_sim_it4_pos = (diff_sim_it4_breuk)**2

diff_sim_it5 = y_sim_it5 - yreg_dat
diff_sim_it5_breuk = diff_sim_it5/yreg_dat
diff_sim_it5_pos = (diff_sim_it5_breuk)**2

diff_sim_it6 = y_sim_it6 - yreg_dat
diff_sim_it6_breuk = diff_sim_it6/yreg_dat
diff_sim_it6_pos = (diff_sim_it6_breuk)**2

diff_sim_it7 = y_sim_it7 - yreg_dat
diff_sim_it7_breuk = diff_sim_it7/yreg_dat
diff_sim_it7_pos = (diff_sim_it7_breuk)**2

diff_sim_agg_pos_sum = 0
for i in range(0,len(diff_sim_agg_pos)):
    diff_sim_agg_pos_sum =  diff_sim_agg_pos_sum + diff_sim_agg_pos[i]
diff_sim_agg_wortel = diff_sim_agg_pos_sum/len(diff_sim_agg_pos)

diff_sim_crit_pos_sum = 0
for i in range(0,len(diff_sim_crit_pos)):
    diff_sim_crit_pos_sum =  diff_sim_crit_pos_sum + diff_sim_crit_pos[i]
diff_sim_crit_wortel = diff_sim_crit_pos_sum/len(diff_sim_crit_pos)
```

```
diff_sim_fol_pos_sum = 0
for i in range(0,len(diff_sim_fol_pos)):
    diff_sim_fol_pos_sum =  diff_sim_fol_pos_sum + diff_sim_fol_pos[i]
diff_sim_fol_wortel = diff_sim_fol_pos_sum/len(diff_sim_fol_pos)

diff_sim_it1_pos_sum = 0
for i in range(0,len(diff_sim_it1_pos)):
    diff_sim_it1_pos_sum =  diff_sim_it1_pos_sum + diff_sim_it1_pos[i]
diff_sim_it1_wortel = diff_sim_it1_pos_sum/len(diff_sim_it1_pos)

diff_sim_it2_pos_sum = 0
for i in range(0,len(diff_sim_it2_pos)):
    diff_sim_it2_pos_sum =  diff_sim_it2_pos_sum + diff_sim_it2_pos[i]
diff_sim_it2_wortel = diff_sim_it2_pos_sum/len(diff_sim_it2_pos)

diff_sim_it3_pos_sum = 0
for i in range(0,len(diff_sim_it3_pos)):
    diff_sim_it3_pos_sum =  diff_sim_it3_pos_sum + diff_sim_it3_pos[i]
diff_sim_it3_wortel = diff_sim_it3_pos_sum/len(diff_sim_it3_pos)

diff_sim_it4_pos_sum = 0
for i in range(0,len(diff_sim_it4_pos)):
    diff_sim_it4_pos_sum =  diff_sim_it4_pos_sum + diff_sim_it4_pos[i]
diff_sim_it4_wortel = diff_sim_it4_pos_sum/len(diff_sim_it4_pos)

diff_sim_it5_pos_sum = 0
for i in range(0,len(diff_sim_it5_pos)):
    diff_sim_it5_pos_sum =  diff_sim_it5_pos_sum + diff_sim_it5_pos[i]
diff_sim_it5_wortel = diff_sim_it5_pos_sum/len(diff_sim_it5_pos)

diff_sim_it6_pos_sum = 0
for i in range(0,len(diff_sim_it6_pos)):
    diff_sim_it6_pos_sum =  diff_sim_it6_pos_sum + diff_sim_it6_pos[i]
diff_sim_it6_wortel = diff_sim_it6_pos_sum/len(diff_sim_it6_pos)

diff_sim_it7_pos_sum = 0
for i in range(0,len(diff_sim_it7_pos)):
    diff_sim_it7_pos_sum =  diff_sim_it7_pos_sum + diff_sim_it7_pos[i]
diff_sim_it7_wortel = diff_sim_it7_pos_sum/len(diff_sim_it7_pos)


RMNSE_sim_agg = np.sqrt(diff_sim_agg_wortel)
RMNSE_sim_crit = np.sqrt(diff_sim_crit_wortel)
RMNSE_sim_fol = np.sqrt(diff_sim_fol_wortel)
RMNSE_sim_it1 = np.sqrt(diff_sim_it1_wortel)
RMNSE_sim_it2 = np.sqrt(diff_sim_it2_wortel)
RMNSE_sim_it3 = np.sqrt(diff_sim_it3_wortel)
RMNSE_sim_it4 = np.sqrt(diff_sim_it4_wortel)
RMNSE_sim_it5 = np.sqrt(diff_sim_it5_wortel)
RMNSE_sim_it6 = np.sqrt(diff_sim_it6_wortel)
RMNSE_sim_it7 = np.sqrt(diff_sim_it7_wortel)

print('The Root Mean Normalized Square Error of the initial simulation equa
print('The Root Mean Normalized Square Error of the decreased critical gap
print('The Root Mean Normalized Square Error of the decreased follow-up tim
print('The Root Mean Normalized Square Error of first iteration equals', RM
print('The Root Mean Normalized Square Error of second iteration equals', R
print('The Root Mean Normalized Square Error of third iteration equals', RM
print('The Root Mean Normalized Square Error of fourth iteration equals', R
print('The Root Mean Normalized Square Error of fifth iteration equals', RM
```

```
print('The Root Mean Normalized Square Error of sixth iteration equals', RM
print('The Root Mean Normalized Square Error of seventh iteration equals',
```

```
The Root Mean Normalized Square Error of the initial simulation equals 0.2
0607970269641365
The Root Mean Normalized Square Error of the decreased critical gap model
equals 0.1661004092805427
The Root Mean Normalized Square Error of the decreased follow-up time mode
l equals 0.12816405199445052
The Root Mean Normalized Square Error of first iteration equals 0.15297112
678578745
The Root Mean Normalized Square Error of second iteration equals 0.1268396
5019868923
The Root Mean Normalized Square Error of third iteration equals 0.09502884
199760116
The Root Mean Normalized Square Error of fourth iteration equals 0.1045239
2911600061
The Root Mean Normalized Square Error of fifth iteration equals 0.13581721
90522451
The Root Mean Normalized Square Error of sixth iteration equals 0.17229902
48308362
The Root Mean Normalized Square Error of seventh iteration equals 0.173006
58515267955
```

Left lane

In [52]:
```
y_reg_dat_Left = nonlinear3(x,aopt_dat_Left, bopt_dat_Left, copt_dat_Left)
y_reg_sim_Left = nonlinear3(x, aopt_sim_Left_vol, bopt_sim_Left_vol, copt_s
y_reg_crit_Left = nonlinear3(x,aopt_sim_Left_crit_vol, bopt_sim_Left_crit_v
y_reg_fol_Left = nonlinear3(x,aopt_sim_Left_fol_vol, bopt_sim_Left_fol_vol,
y_reg_it1_Left = nonlinear3(x, aopt_sim_Left_it1, bopt_sim_Left_it1, copt_s
y_reg_it2_Left = nonlinear3(x, aopt_sim_Left_it2, bopt_sim_Left_it2, copt_s
y_reg_it3_Left = nonlinear3(x, aopt_sim_Left_it3, bopt_sim_Left_it3, copt_s
y_reg_it4_Left = nonlinear3(x, aopt_sim_Left_it4, bopt_sim_Left_it4, copt_s
y_reg_it5_Left = nonlinear3(x, aopt_sim_Left_it5, bopt_sim_Left_it5, copt_s
y_reg_it6_Left = nonlinear3(x, aopt_sim_Left_it6, bopt_sim_Left_it6, copt_s
y_reg_it7_Left = nonlinear3(x, aopt_sim_Left_it7, bopt_sim_Left_it7, copt_s

diff_sim_left = y_reg_sim_Left - y_reg_dat_Left
diff_sim_left_breuk = diff_sim_left/y_reg_dat_Left
diff_sim_left_pos = (diff_sim_left_breuk)**2

diff_crit_left = y_reg_crit_Left - y_reg_dat_Left
diff_crit_left_breuk = diff_crit_left/y_reg_dat_Left
diff_crit_left_pos = (diff_crit_left_breuk)**2

diff_fol_left = y_reg_fol_Left - y_reg_dat_Left
diff_fol_left_breuk = diff_fol_left/y_reg_dat_Left
diff_fol_left_pos = (diff_fol_left_breuk)**2

diff_it1_left = y_reg_it1_Left - y_reg_dat_Left
diff_it1_left_breuk = diff_it1_left/y_reg_dat_Left
diff_it1_left_pos = (diff_it1_left_breuk)**2

diff_it2_left = y_reg_it2_Left - y_reg_dat_Left
diff_it2_left_breuk = diff_it2_left/y_reg_dat_Left
diff_it2_left_pos = (diff_it2_left_breuk)**2

diff_it3_left = y_reg_it3_Left - y_reg_dat_Left
diff_it3_left_breuk = diff_it3_left/y_reg_dat_Left
diff_it3_left_pos = (diff_it3_left_breuk)**2

diff_it4_left = y_reg_it4_Left - y_reg_dat_Left
diff_it4_left_breuk = diff_it4_left/y_reg_dat_Left
diff_it4_left_pos = (diff_it4_left_breuk)**2

diff_it5_left = y_reg_it5_Left - y_reg_dat_Left
diff_it5_left_breuk = diff_it5_left/y_reg_dat_Left
diff_it5_left_pos = (diff_it5_left_breuk)**2

diff_it6_left = y_reg_it6_Left - y_reg_dat_Left
diff_it6_left_breuk = diff_it6_left/y_reg_dat_Left
diff_it6_left_pos = (diff_it6_left_breuk)**2

diff_it7_left = y_reg_it7_Left - y_reg_dat_Left
diff_it7_left_breuk = diff_it7_left/y_reg_dat_Left
diff_it7_left_pos = (diff_it7_left_breuk)**2

diff_sim_left_pos_sum = 0
for i in range(0,len(diff_sim_left_pos)):
    diff_sim_left_pos_sum =  diff_sim_left_pos_sum + diff_sim_left_pos[i]
diff_sim_left_wortel = diff_sim_left_pos_sum/len(diff_sim_left_pos)

diff_crit_left_pos_sum = 0
for i in range(0,len(diff_crit_left_pos)):
    diff_crit_left_pos_sum =  diff_crit_left_pos_sum + diff_crit_left_pos[i
diff_crit_left_wortel = diff_crit_left_pos_sum/len(diff_crit_left_pos)
```

```
diff_fol_left_pos_sum = 0
for i in range(0,len(diff_fol_left_pos)):
    diff_fol_left_pos_sum =  diff_fol_left_pos_sum + diff_fol_left_pos[i]
diff_fol_left_wortel = diff_fol_left_pos_sum/len(diff_fol_left_pos)

diff_it1_left_pos_sum = 0
for i in range(0,len(diff_it1_left_pos)):
    diff_it1_left_pos_sum =  diff_it1_left_pos_sum + diff_it1_left_pos[i]
diff_it1_left_wortel = diff_it1_left_pos_sum/len(diff_it1_left_pos)

diff_it2_left_pos_sum = 0
for i in range(0,len(diff_it2_left_pos)):
    diff_it2_left_pos_sum =  diff_it2_left_pos_sum + diff_it2_left_pos[i]
diff_it2_left_wortel = diff_it2_left_pos_sum/len(diff_it2_left_pos)

diff_it3_left_pos_sum = 0
for i in range(0,len(diff_it3_left_pos)):
    diff_it3_left_pos_sum =  diff_it3_left_pos_sum + diff_it3_left_pos[i]
diff_it3_left_wortel = diff_it3_left_pos_sum/len(diff_it3_left_pos)

diff_it4_left_pos_sum = 0
for i in range(0,len(diff_it4_left_pos)):
    diff_it4_left_pos_sum =  diff_it4_left_pos_sum + diff_it4_left_pos[i]
diff_it4_left_wortel = diff_it4_left_pos_sum/len(diff_it4_left_pos)

diff_it5_left_pos_sum = 0
for i in range(0,len(diff_it5_left_pos)):
    diff_it5_left_pos_sum =  diff_it5_left_pos_sum + diff_it5_left_pos[i]
diff_it5_left_wortel = diff_it5_left_pos_sum/len(diff_it5_left_pos)

diff_it6_left_pos_sum = 0
for i in range(0,len(diff_it6_left_pos)):
    diff_it6_left_pos_sum =  diff_it6_left_pos_sum + diff_it6_left_pos[i]
diff_it6_left_wortel = diff_it6_left_pos_sum/len(diff_it6_left_pos)

diff_it7_left_pos_sum = 0
for i in range(0,len(diff_it7_left_pos)):
    diff_it7_left_pos_sum =  diff_it7_left_pos_sum + diff_it7_left_pos[i]
diff_it7_left_wortel = diff_it7_left_pos_sum/len(diff_it7_left_pos)

RMNSE_sim_left = np.sqrt(diff_sim_left_wortel)
RMNSE_crit_left = np.sqrt(diff_crit_left_wortel)
RMNSE_fol_left = np.sqrt(diff_fol_left_wortel)
RMNSE_it1_left = np.sqrt(diff_it1_left_wortel)
RMNSE_it2_left = np.sqrt(diff_it2_left_wortel)
RMNSE_it3_left = np.sqrt(diff_it3_left_wortel)
RMNSE_it4_left = np.sqrt(diff_it4_left_wortel)
RMNSE_it5_left = np.sqrt(diff_it5_left_wortel)
RMNSE_it6_left = np.sqrt(diff_it6_left_wortel)
RMNSE_it7_left = np.sqrt(diff_it7_left_wortel)


print('The Root Mean Normalized Square Error of the initial simulation equa
print('The Root Mean Normalized Square Error of the decreased critical gap
print('The Root Mean Normalized Square Error of the decreased follow-up tim
print('The Root Mean Normalized Square Error of the first iteration equals'
print('The Root Mean Normalized Square Error of the second iteration equals
print('The Root Mean Normalized Square Error of the third iteration equals'
print('The Root Mean Normalized Square Error of the fourth iteration equals
print('The Root Mean Normalized Square Error of the fifth iteration equals'
```

```
print('The Root Mean Normalized Square Error of the sixth iteration equals'
print('The Root Mean Normalized Square Error of the seventh iteration equal
```

```
The Root Mean Normalized Square Error of the initial simulation equals 0.2
6973342457095895
The Root Mean Normalized Square Error of the decreased critical gap simula
tion equals 0.19438623657578907
The Root Mean Normalized Square Error of the decreased follow-up time simu
lation equals 0.20163809449854056
The Root Mean Normalized Square Error of the first iteration equals 0.1988
722737852504
The Root Mean Normalized Square Error of the second iteration equals 0.068
25665826392348
The Root Mean Normalized Square Error of the third iteration equals 0.1354
9994380647334
The Root Mean Normalized Square Error of the fourth iteration equals 0.110
45050663786175
The Root Mean Normalized Square Error of the fifth iteration equals 0.1058
2684414960032
The Root Mean Normalized Square Error of the sixth iteration equals 0.0851
2530558116774
The Root Mean Normalized Square Error of the seventh iteration equals 0.07
398053779371021
```

Right lane

In [53]:
```
x_right = np.linspace(400,1200,41)
y_reg_sim_Right = nonlinear3(x_right, aopt_sim_Right_vol, bopt_sim_Right_vo
y_reg_crit_Right = nonlinear3(x_right, aopt_sim_Right_crit_vol, bopt_sim_Ri
y_reg_fol_Right = nonlinear3(x_right, aopt_sim_Right_fol_vol, bopt_sim_Righ
y_reg_it1_Right = nonlinear3(x_right, aopt_sim_Right_it1, bopt_sim_Right_it
y_reg_it2_Right = nonlinear3(x_right, aopt_sim_Right_it2, bopt_sim_Right_it
y_reg_it3_Right = nonlinear3(x_right, aopt_sim_Right_it3, bopt_sim_Right_it
y_reg_it4_Right = nonlinear3(x_right, aopt_sim_Right_it4, bopt_sim_Right_it
y_reg_it5_Right = nonlinear3(x_right, aopt_sim_Right_it5, bopt_sim_Right_it
y_reg_it6_Right = nonlinear3(x_right, aopt_sim_Right_it6, bopt_sim_Right_it
y_reg_it7_Right = nonlinear3(x_right, aopt_sim_Right_it7, bopt_sim_Right_it

diff_sim_right = y_reg_sim_Right - yreg_dat_Right
diff_sim_right_breuk = diff_sim_right/yreg_dat_Right
diff_sim_right_pos = (diff_sim_right_breuk)**2

diff_crit_right = y_reg_crit_Right - yreg_dat_Right
diff_crit_right_breuk = diff_crit_right/yreg_dat_Right
diff_crit_right_pos = (diff_crit_right_breuk)**2

diff_fol_right = y_reg_fol_Right - yreg_dat_Right
diff_fol_right_breuk = diff_fol_right/yreg_dat_Right
diff_fol_right_pos = (diff_fol_right_breuk)**2

diff_it1_right = y_reg_it1_Right - yreg_dat_Right
diff_it1_right_breuk = diff_it1_right/yreg_dat_Right
diff_it1_right_pos = (diff_it1_right_breuk)**2

diff_it2_right = y_reg_it2_Right - yreg_dat_Right
diff_it2_right_breuk = diff_it2_right/yreg_dat_Right
diff_it2_right_pos = (diff_it2_right_breuk)**2

diff_it3_right = y_reg_it3_Right - yreg_dat_Right
diff_it3_right_breuk = diff_it3_right/yreg_dat_Right
diff_it3_right_pos = (diff_it3_right_breuk)**2

diff_it4_right = y_reg_it4_Right - yreg_dat_Right
diff_it4_right_breuk = diff_it4_right/yreg_dat_Right
diff_it4_right_pos = (diff_it4_right_breuk)**2

diff_it5_right = y_reg_it5_Right - yreg_dat_Right
diff_it5_right_breuk = diff_it5_right/yreg_dat_Right
diff_it5_right_pos = (diff_it5_right_breuk)**2

diff_it6_right = y_reg_it6_Right - yreg_dat_Right
diff_it6_right_breuk = diff_it6_right/yreg_dat_Right
diff_it6_right_pos = (diff_it6_right_breuk)**2

diff_it7_right = y_reg_it7_Right - yreg_dat_Right
diff_it7_right_breuk = diff_it7_right/yreg_dat_Right
diff_it7_right_pos = (diff_it7_right_breuk)**2

diff_sim_right_pos_sum = 0
for i in range(0,len(diff_sim_right_pos)):
    diff_sim_right_pos_sum =  diff_sim_right_pos_sum + diff_sim_right_pos[i
diff_sim_right_wortel = diff_sim_right_pos_sum/len(diff_sim_right_pos)

diff_crit_right_pos_sum = 0
for i in range(0,len(diff_crit_right_pos)):
    diff_crit_right_pos_sum =  diff_crit_right_pos_sum + diff_crit_right_po
diff_crit_right_wortel = diff_crit_right_pos_sum/len(diff_crit_right_pos)
```

```
diff_fol_right_pos_sum = 0
for i in range(0,len(diff_fol_right_pos)):
    diff_fol_right_pos_sum =  diff_fol_right_pos_sum + diff_fol_right_pos[i
diff_fol_right_wortel = diff_fol_right_pos_sum/len(diff_fol_right_pos)

diff_it1_right_pos_sum = 0
for i in range(0,len(diff_it1_right_pos)):
    diff_it1_right_pos_sum =  diff_it1_right_pos_sum + diff_it1_right_pos[i
diff_it1_right_wortel = diff_it1_right_pos_sum/len(diff_it1_right_pos)

diff_it2_right_pos_sum = 0
for i in range(0,len(diff_it2_right_pos)):
    diff_it2_right_pos_sum =  diff_it2_right_pos_sum + diff_it2_right_pos[i
diff_it2_right_wortel = diff_it2_right_pos_sum/len(diff_it2_right_pos)

diff_it3_right_pos_sum = 0
for i in range(0,len(diff_it3_right_pos)):
    diff_it3_right_pos_sum =  diff_it3_right_pos_sum + diff_it3_right_pos[i
diff_it3_right_wortel = diff_it3_right_pos_sum/len(diff_it3_right_pos)

diff_it4_right_pos_sum = 0
for i in range(0,len(diff_it4_right_pos)):
    diff_it4_right_pos_sum =  diff_it4_right_pos_sum + diff_it4_right_pos[i
diff_it4_right_wortel = diff_it4_right_pos_sum/len(diff_it4_right_pos)

diff_it5_right_pos_sum = 0
for i in range(0,len(diff_it5_right_pos)):
    diff_it5_right_pos_sum =  diff_it5_right_pos_sum + diff_it5_right_pos[i
diff_it5_right_wortel = diff_it5_right_pos_sum/len(diff_it5_right_pos)

diff_it6_right_pos_sum = 0
for i in range(0,len(diff_it6_right_pos)):
    diff_it6_right_pos_sum =  diff_it6_right_pos_sum + diff_it6_right_pos[i
diff_it6_right_wortel = diff_it6_right_pos_sum/len(diff_it6_right_pos)

diff_it7_right_pos_sum = 0
for i in range(0,len(diff_it7_right_pos)):
    diff_it7_right_pos_sum =  diff_it7_right_pos_sum + diff_it7_right_pos[i
diff_it7_right_wortel = diff_it7_right_pos_sum/len(diff_it7_right_pos)


RMNSE_sim_right = np.sqrt(diff_sim_right_wortel)
RMNSE_crit_right = np.sqrt(diff_crit_right_wortel)
RMNSE_fol_right = np.sqrt(diff_fol_right_wortel)
RMNSE_it1_right = np.sqrt(diff_it1_right_wortel)
RMNSE_it2_right = np.sqrt(diff_it2_right_wortel)
RMNSE_it3_right = np.sqrt(diff_it3_right_wortel)
RMNSE_it4_right = np.sqrt(diff_it4_right_wortel)
RMNSE_it5_right = np.sqrt(diff_it5_right_wortel)
RMNSE_it6_right = np.sqrt(diff_it6_right_wortel)
RMNSE_it7_right = np.sqrt(diff_it7_right_wortel)

print('The Root Mean Normalized Square Error of the initial simulation equa
print('The Root Mean Normalized Square Error of the decreased critical gap
print('The Root Mean Normalized Square Error of the decreased follow-up tim
print('The Root Mean Normalized Square Error of the first iteration equals'
print('The Root Mean Normalized Square Error of the second iteration equals
print('The Root Mean Normalized Square Error of the third iteration equals'
print('The Root Mean Normalized Square Error of the fourth iteration equals
print('The Root Mean Normalized Square Error of the fifth iteration equals'
```

```
print('The Root Mean Normalized Square Error of the sixth iteration equals'
print('The Root Mean Normalized Square Error of the seventh iteration equal
```

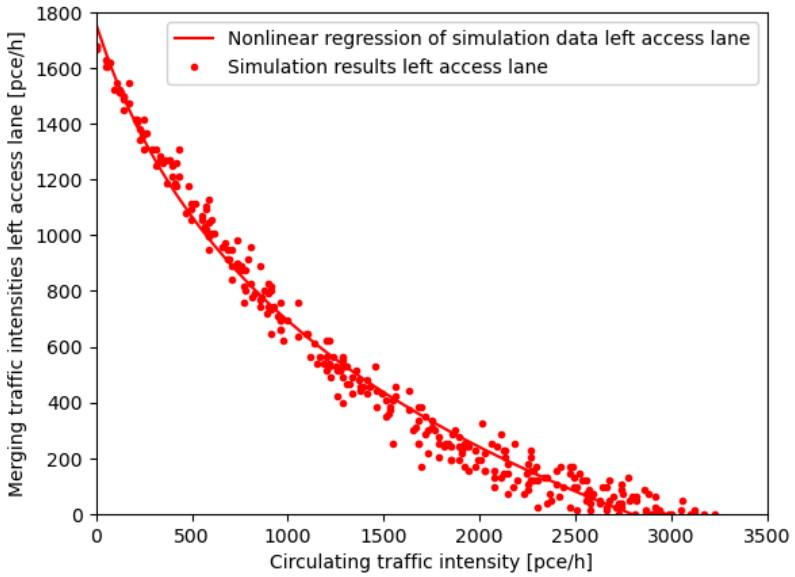
```
The Root Mean Normalized Square Error of the initial simulation equals 0.4
0816252132791897
The Root Mean Normalized Square Error of the decreased critical gap simula
tion equals 0.403858241053405
The Root Mean Normalized Square Error of the decreased follow-up time simu
lation equals 0.31741163876668516
The Root Mean Normalized Square Error of the first iteration equals 0.3916
844476515513
The Root Mean Normalized Square Error of the second iteration equals 0.380
46129158667663
The Root Mean Normalized Square Error of the third iteration equals 0.3616
481165281341
The Root Mean Normalized Square Error of the fourth iteration equals 0.361
32167977217583
The Root Mean Normalized Square Error of the fifth iteration equals 0.3834
0208144989035
The Root Mean Normalized Square Error of the sixth iteration equals 0.4229
7012118569727
The Root Mean Normalized Square Error of the seventh iteration equals 0.42
680490831173185
```

CAV models

Left lane

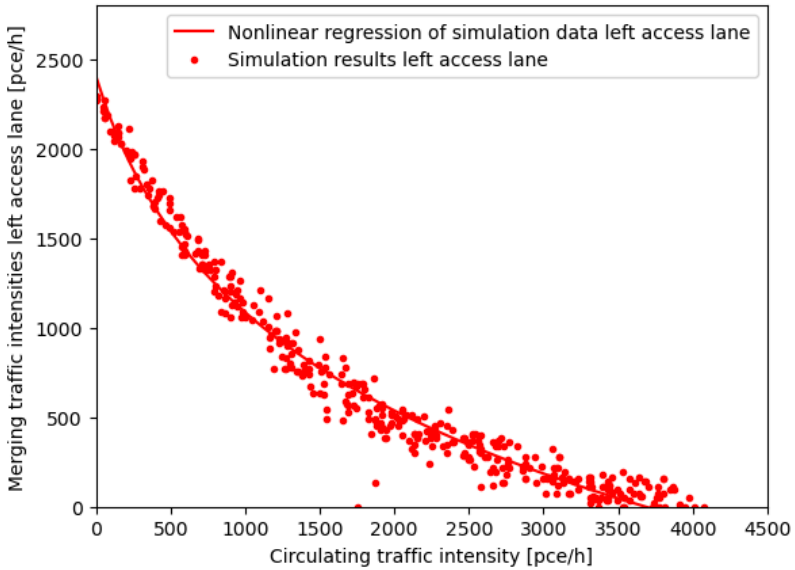
In [54]:
```
reg_sim_Left_cau , pcov_sim_Left_cau = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_Left_cau, bopt_sim_Left_cau, copt_sim_Left_cau = reg_sim_Left_cau

y_fit_sim_Left_cau = nonlinear3(Circul_cau_vol, aopt_sim_Left_cau, bopt_sim

plt.plot(Circul_cau_vol, y_fit_sim_Left_cau, color = 'red', label = 'Nonlin
plt.plot(Circul_cau_vol, Left_cau_vol, 'r.', label = 'Simulation results le
plt.legend(loc = 'best')
plt.ylim([0,1800])
plt.xlim([0,3500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



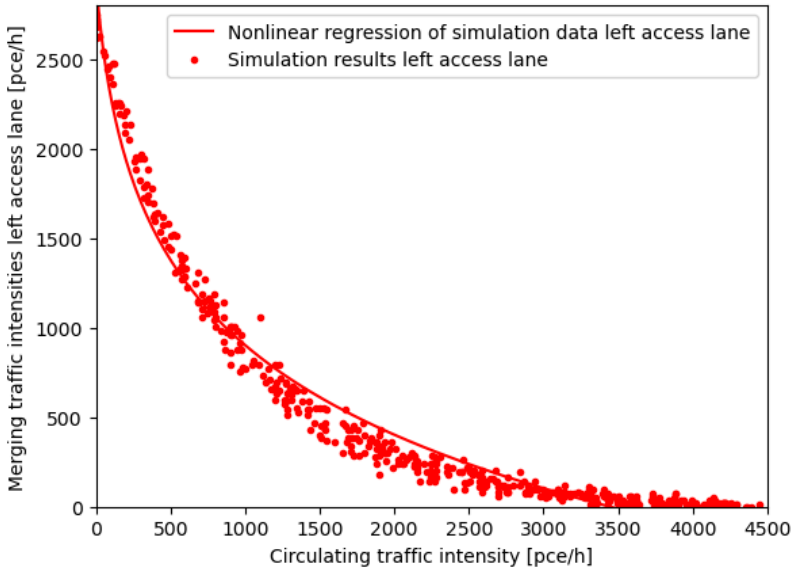
In [55]:
```
reg_sim_Left_norm , pcov_sim_Left_norm = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Left_norm, bopt_sim_Left_norm, copt_sim_Left_norm = reg_sim_Left_n

y_fit_sim_Left_norm = nonlinear3(Circul_norm_vol, aopt_sim_Left_norm, bopt_

plt.plot(Circul_norm_vol, y_fit_sim_Left_norm, color = 'red', label = 'Nonl
plt.plot(Circul_norm_vol, Left_norm_vol, 'r.', label = 'Simulation results
plt.legend(loc = 'best')
plt.ylim([0,2800])
plt.xlim([0,4500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```

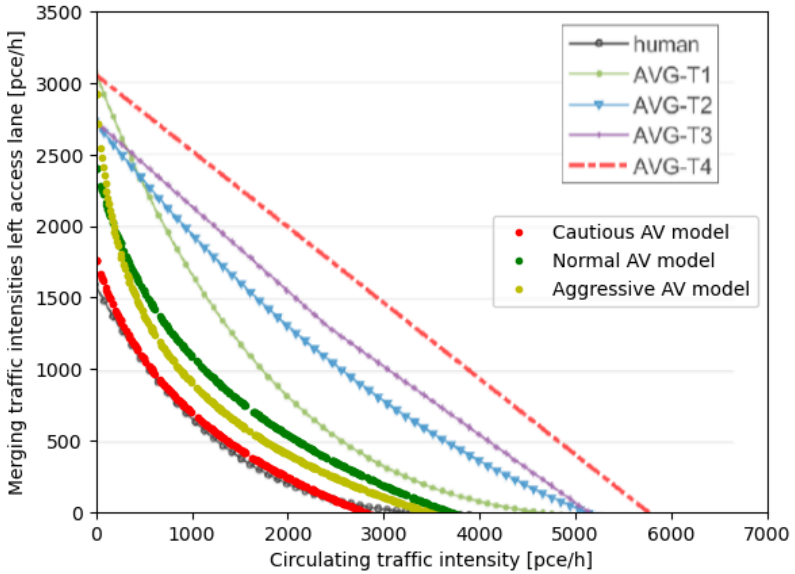


In [56]:
```
reg_sim_Left_agg , pcov_sim_Left_agg = sc.optimize.curve_fit(nonlinear3, Ci
aopt_sim_Left_agg, bopt_sim_Left_agg, copt_sim_Left_agg = reg_sim_Left_agg

y_fit_sim_Left_agg = nonlinear3(Circul_agg_vol, aopt_sim_Left_agg, bopt_sim

plt.plot(Circul_agg_vol, y_fit_sim_Left_agg, color = 'red', label = 'Nonlin
plt.plot(Circul_agg_vol, Left_agg_vol, 'r.', label = 'Simulation results le
plt.legend(loc = 'best')
plt.ylim([0,2800])
plt.xlim([0,4500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



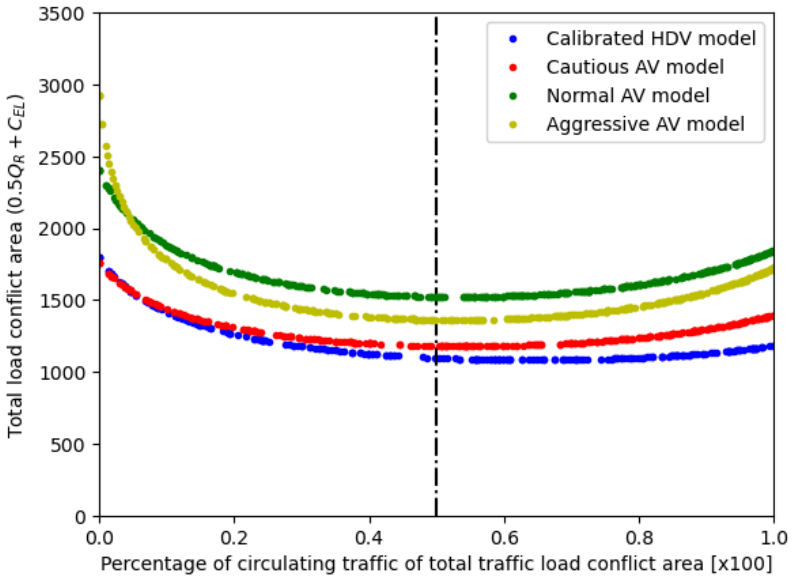
In [57]:
```
im = plt.imread('Fortuijn3.PNG')
plt.imshow(im, extent=[0, 7000, 0, 3500], aspect='auto', alpha=0.7)

plt.plot(Circul_cau_vol, y_fit_sim_Left_cau, 'r.', label = 'Cautious AV mod
plt.plot(Circul_norm_vol, y_fit_sim_Left_norm, 'g.', label = 'Normal AV mod
plt.plot(Circul_agg_vol, y_fit_sim_Left_agg, 'y.', label = 'Aggressive AV m
plt.xlim([0,7000])
plt.ylim([0,3500])
plt.legend(loc=7)
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane [pce/h]');
```



In [64]:

```
ybase = 0.5*Circul_it7 + y_fit_sim_Left_vol
xbase = (0.5*Circul_vol)/(0.5*Circul_vol+y_fit_sim_Left_vol)
print('The merging capacity of the base simulation is',ybase[112],'pce/h.')

yval = 0.5*Circul_it7+y_fit_sim_Left_it7
xval = (0.5*Circul_it7)/(0.5*Circul_it7+y_fit_sim_Left_it7)
print('The merging capacity of the validated HDV model is',(yval[115]+yval[

ycauleft = 0.5*Circul_cau_vol+y_fit_sim_Left_cau
xcauleft = (0.5*Circul_cau_vol)/(0.5*Circul_cau_vol+y_fit_sim_Left_cau)
print('The merging capacity of the cautious CAV model is',ycauleft[121],'pc

ynormleft = 0.5*Circul_norm_vol+y_fit_sim_Left_norm
xnormleft = (0.5*Circul_norm_vol)/(0.5*Circul_norm_vol+y_fit_sim_Left_norm)
print('The merging capacity of the normal CAV model is',ynormleft[160],'pce

yaggleft = 0.5*Circul_agg_vol+y_fit_sim_Left_agg
xaggleft = (0.5*Circul_agg_vol)/(0.5*Circul_agg_vol+y_fit_sim_Left_agg)
print('The merging capacity of the normal CAV model is',yaggleft[140],'pce/

plt.plot(xval,yval,'b.', label = 'Calibrated HDV model')
plt.axvline(0.5,color='black',linestyle='dashdot')
plt.plot(xcauleft,ycauleft,'r.', label = 'Cautious AV model')
plt.plot(xnormleft,ynormleft,'g.',label = 'Normal AV model')
plt.plot(xaggleft,yaggleft,'y.',label= 'Aggressive AV model')
plt.legend(loc='best')
plt.xlabel('Percentage of circulating traffic of total traffic load conflic
plt.ylabel('Total load conflict area ($0.5Q_R + C_{EL}$)')
plt.xlim([0,1])
plt.ylim([0,3500]);
```

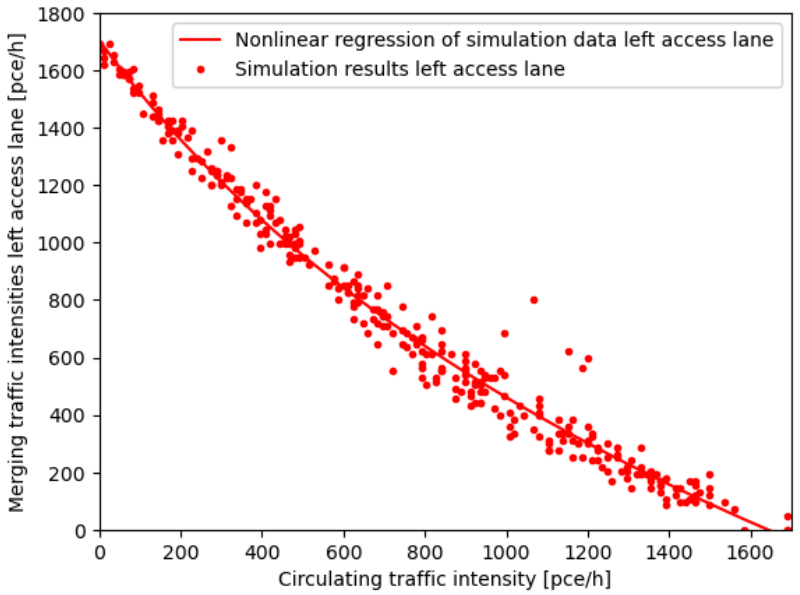
```
The merging capacity of the base simulation is 987.3592243134826 pce/h.
The merging capacity of the validated HDV model is 1098.8296360340137 pce/
h.
The merging capacity of the cautious CAV model is 1180.8266902651703 pce/
h, which is 7.462217212042731 percent higher than the validated HDV model.
The merging capacity of the normal CAV model is 1522.9711637776054 pce/h,
which is 38.59938918961436 percent higher than the validated HDV model.
The merging capacity of the normal CAV model is 1364.798031358976 pce/h, w
hich is 24.204698035349452 percent higher than the validated HDV model.
```



Right lane

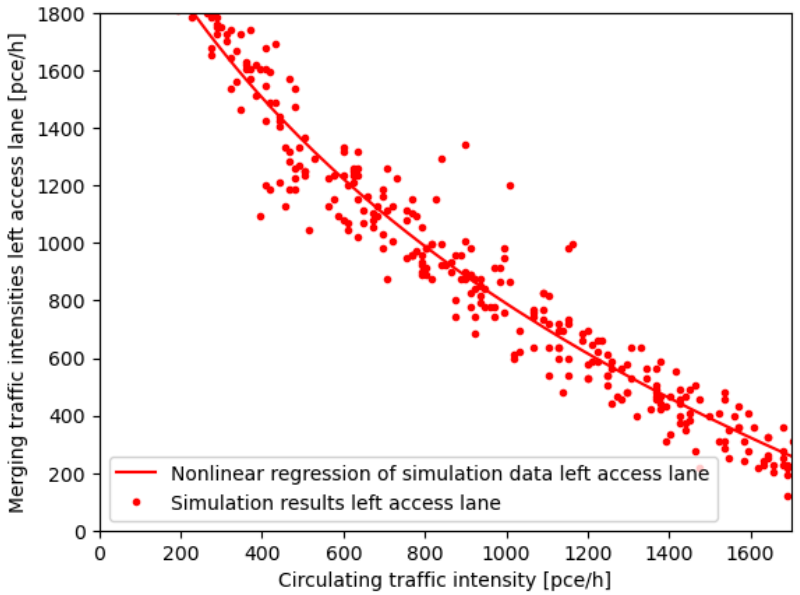
In [59]:
```
reg_sim_Right_cau , pcov_sim_Right_cau = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Right_cau, bopt_sim_Right_cau, copt_sim_Right_cau = reg_sim_Right_

y_fit_sim_Right_cau = nonlinear3(Outer_cau_vol, aopt_sim_Right_cau, bopt_si

plt.plot(Outer_cau_vol, y_fit_sim_Right_cau, color = 'red', label = 'Nonlin
plt.plot(Outer_cau_vol, Right_cau_vol, 'r.', label = 'Simulation results le
plt.legend(loc = 'best')
plt.ylim([0,1800])
plt.xlim([0,1700])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```



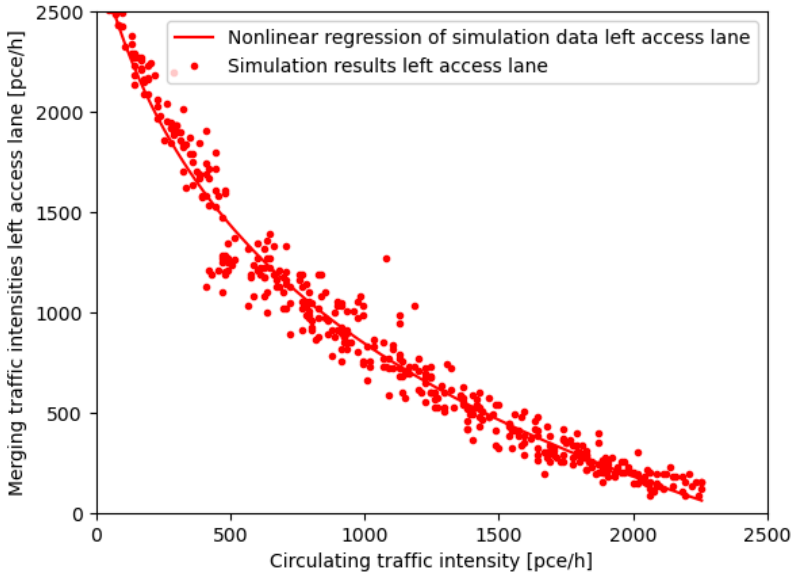
In [60]:
```
reg_sim_Right_norm , pcov_sim_Right_norm = sc.optimize.curve_fit(nonlinear3
aopt_sim_Right_norm, bopt_sim_Right_norm, copt_sim_Right_norm = reg_sim_Rig

y_fit_sim_Right_norm = nonlinear3(Outer_norm_vol, aopt_sim_Right_norm, bopt

plt.plot(Outer_norm_vol, y_fit_sim_Right_norm, color = 'red', label = 'Nonl
plt.plot(Outer_norm_vol, Right_norm_vol, 'r.', label = 'Simulation results
plt.legend(loc = 'best')
plt.ylim([0,1800])
plt.xlim([0,1700])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```

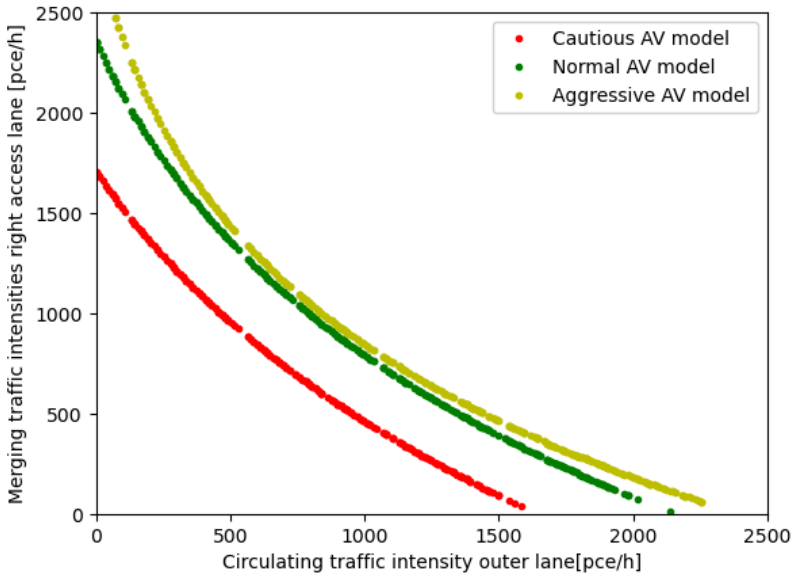


In [61]:
```
reg_sim_Right_agg , pcov_sim_Right_agg = sc.optimize.curve_fit(nonlinear3,
aopt_sim_Right_agg, bopt_sim_Right_agg, copt_sim_Right_agg = reg_sim_Right_

y_fit_sim_Right_agg = nonlinear3(Outer_agg_vol, aopt_sim_Right_agg, bopt_si

plt.plot(Outer_agg_vol, y_fit_sim_Right_agg, color = 'red', label = 'Nonlin
plt.plot(Outer_agg_vol, Right_agg_vol, 'r.', label = 'Simulation results le
plt.legend(loc = 'best')
plt.ylim([0,2500])
plt.xlim([0,2500])
plt.xlabel('Circulating traffic intensity [pce/h]')
plt.ylabel('Merging traffic intensities left access lane [pce/h]');
```

```
C:\Users\jerom\AppData\Local\Temp\ipykernel_11800\4183165999.py:6: Runtime
Warning: invalid value encountered in log
  return a * np.log(b*x+1e-10) + c
```

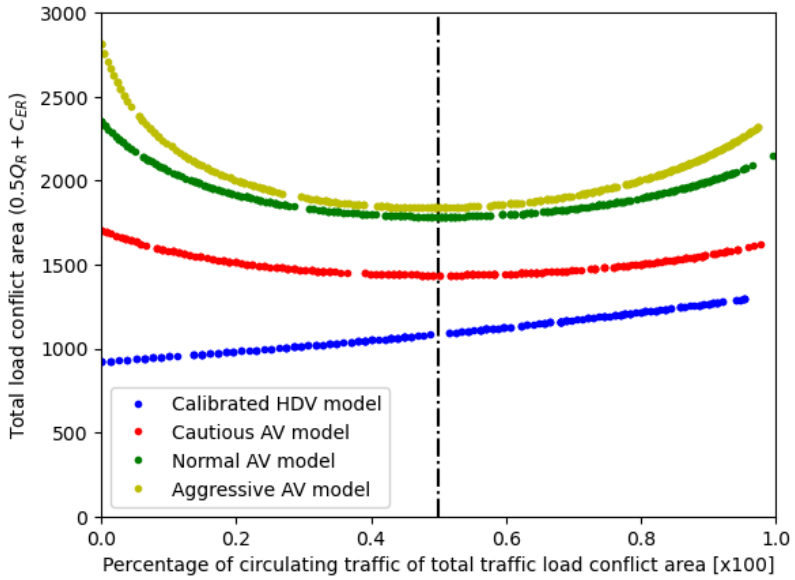


In [62]:
```
plt.plot(Outer_cau_vol, y_fit_sim_Right_cau,'r.', label = 'Cautious AV mode
plt.plot(Outer_norm_vol, y_fit_sim_Right_norm, 'g.', label = 'Normal AV mod
plt.plot(Outer_agg_vol, y_fit_sim_Right_agg, 'y.', label = 'Aggressive AV m
plt.ylim([0,2500])
plt.xlim([0,2500])
plt.legend(loc='best')
plt.xlabel('Circulating traffic intensity outer lane[pce/h]')
plt.ylabel('Merging traffic intensities right access lane [pce/h]');
```



In [65]:
```
yval = Outer_it7+y_fit_sim_Right_it7
xval = Outer_it7/(Outer_it7+y_fit_sim_Right_it7)
print(yval[139])

ycauright = Outer_cau_vol+y_fit_sim_Right_cau
xcauright = (Outer_cau_vol)/(Outer_cau_vol+y_fit_sim_Right_cau)
print(ycauright[155])

ynormright = Outer_norm_vol+y_fit_sim_Right_norm
xnormright = (Outer_norm_vol)/(Outer_norm_vol+y_fit_sim_Right_norm)
print(ynormright[193])

yaggright = Outer_agg_vol+y_fit_sim_Right_agg
xaggright = (Outer_agg_vol)/(Outer_agg_vol+y_fit_sim_Right_agg)
print((yaggright[201]+yaggright[202])/2)

plt.plot(xval,yval,'b.', label = 'Calibrated HDV model')
plt.axvline(0.5,color='black',linestyle='dashdot')
plt.plot(xcauright,ycauright,'r.', label = 'Cautious AV model')
plt.plot(xnormright,ynormright,'g.',label = 'Normal AV model')
plt.plot(xaggright,yaggright,'y.',label= 'Aggressive AV model')
plt.legend(loc='best')
plt.xlabel('Percentage of circulating traffic of total traffic load conflic
plt.ylabel('Total load conflict area ($0.5Q_R + C_{ER}$)')
plt.xlim([0,1])
plt.ylim([0,3000]);
```

```
1118.2718912437558
1437.6632634302878
1785.6947980688055
1839.3565479207064
```



In [ ]: