Simulation Project

Milestone Report

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FAKULTÄT FÜR INFORMATIK

Beladi, Seyedbehnam
seyedbehnam.beladi@ovgu.de

Bhavaraju, Sesha Sai Kiran
sesha.bhavaraju@st.ovgu.de

Dartov, Boris
boris.djartov@st.ovgu.de

Kolli, Lalith Sai
lalith.kolli@st.ovgu.de

Klopfer, Bastian
bastian.klopfer@st.ovgu.de

Summer Term 2019

Abstract

This report documents the work done during a simulation project in the Faculty of Informatic of Otto-von-Guericke Universität Magdeburg. Said project is offered to students enrolled in the Master courses "Digital Engineering" and "Data Knowledge Engineering" by the Chair of Simulation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Making changes to an actual system is challenging, time taking and expensive. Simulation is the best approach to model complex systems and to test different approaches before implementation. Our goal is to model the traffic node “Am Krökentor / Erzberger Straße” and then to improve the throughput and decrease waiting time for traffic participants, especially those who use ”Am Krökentor”.

The given intersection is outstandingly busy in peak hours which are in the morning hours at around 8 am and in the afternoon at 4 pm due to inhabitants going to and coming from work. Noteworthy for the street “Am Krökentor” is the large number of vehicles coming from Bundesstraße 1, which usually end up wanting to turn right to get into the city centre. For this specific lane the throughput and waiting time of the traffic participants should be improved.

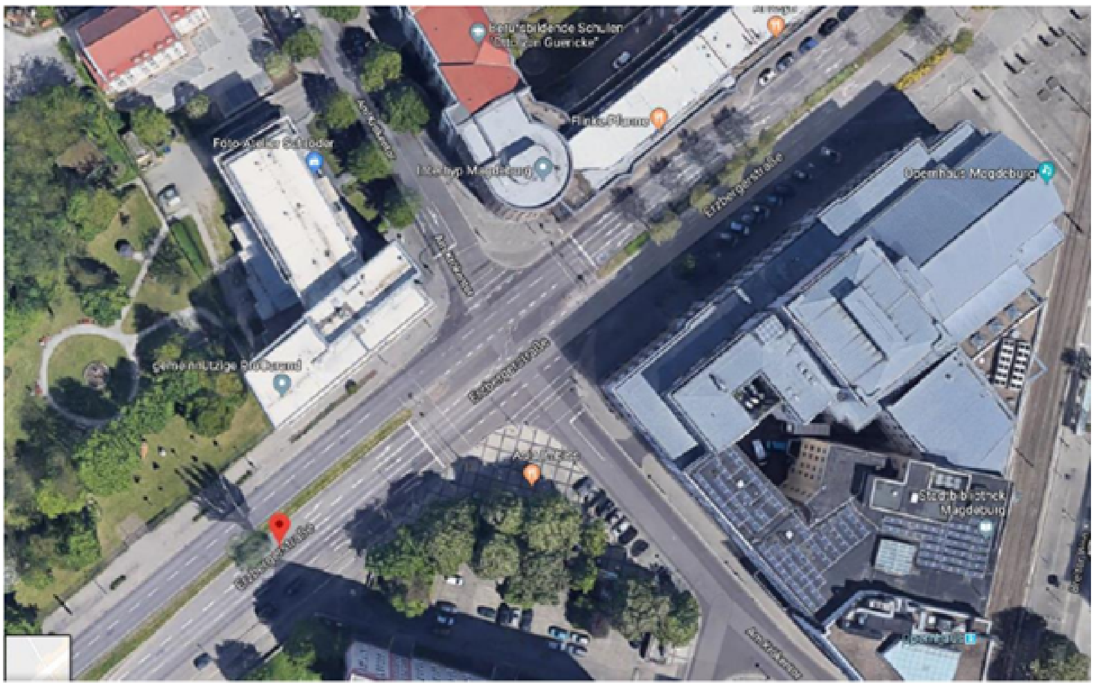


Figure 1.1: Node of interest

The following chapters will describe:

- How a conceptual model is developed
- How data based on the designed model is collected
- How the simulation model using the ANYLOGIC" software is developed.
- How the aforementioned simulation model is validated
- Results from conducted experiments and recommendations

The following section "Team Goal" gives the the overall goal for this team project.

1.1 Team Goal

We are team Beta. Our team goal is to successfully finish the project allocated to us within the given deadline and budgetary constraints. Our individual goals are to gain experience working within a team and to apply the knowledge we have, regarding simulation, in a practical setting. Our project goal is the successful and timely completion of all milestones, while utilizing all the resources, knowledge and skills at our disposal. Furthermore, we also strive to satisfy our client the city of Magdeburg planning office and get a good grade for the completion of the simulation project. Each team member acknowledges the importance of punctuality, aiding a teammate and participation in team tasks. Every member is also prepared to undertake the responsibility for his/her role and to support the team in the envisioned completion of the project.

Chapter 2

Milestones

2.1 Team Formation

This chapter covers the creation of the team along with the roles and responsibilities of each team member. Additionally this chapter describes the quality criteria for the project as well as the team standards.

2.1.1 Team

Team B (beta) is comprised of five members:

- Boris Djartov – Team leader, Validation and quality control
- Bastian Klopfer - Conceptual model
- Lalith Sai Kolli – Data acquisition and analysis
- Seyedbehnam Beladi - Chief software architect
- Sesha Sai Kiran Bhavaraju - Experiments

2.1.2 Quality criteria

For our project we have agreed upon the following quality criteria:

- Timely delivery of the expected end report to the city planning office.
- Respecting the projects budgetary restrictions.
- Satisfaction of the city planning office.
- Usefulness of project and final result to the city planning office.
- Successful completion of milestones (evaluated from both the team members and supervisors).
- Experienced gained and knowledge acquired (how much the team members learned from this project).
- Fun had by team members.

2.1.3 Team Standard

Each team member is expected to be:

- Reliable.
- Punctual for meetings.
- Honest and open.
- Abides by the ideas of team work and comradery.
- Takes responsibility for his/her action.
- Contributes to the project.
- Gives suggestions and possible solutions.
- Does not create unneeded tension within the team or otherwise disrupting the team activates.

2.1.4 Quality criteria for roles

Apart from the general team standard we have identified and agreed upon certain quality standards which correspond to a specific role.

Team leader

- Maintain group cohesion.
- Punctual for meetings.
- Guiding conversation and discussion towards solutions and agreements.
- Keep team members motivated and focused.
- Provide additional assistance.

Data acquisition and analysis

- Good assessment of the needed data.
- Maximizing the use of all of the data on hand.
- Good coordination with other team members (what data is needed).
- Understanding the limitations of the available data and informing other team members.

Conceptual model

- Valid suggestions on how we can simplify the model and how to represent the node of interest.
- Able to accurately represent the situation (taking into account simplifications) and being able to successfully communicate this to the rest of the team.

Chief software architect

- Implementing the Conceptual model to a certain accuracy.
- Fixing the minor details that were not considered in the conceptual model.
- Detect and ignore irrelevant and redundant data in order to have a simple and straightforward model.
- Design an interesting and presentable user interface.

Validation and quality control

- Make sure the models are in line with the situation of interest and that it reflects reality.
- Make sure that the quality of the models are satisfactory.

Experimental Deign

- Suggesting interesting and relevant hypothesis based on available data and modeling capabilities.
- Suggesting experiments that would test hypothesis of interests.

2.2 Project Plan

This chapter will be a brief description of the team plan and budget allocation.

2.2.1 Importance of Planning

It is very crucial to plan ahead before starting a project. There are a lot of benefits from a thorough and precise plan that has been done on time.

There is a better communication between team members and the objective of the team is clear so they will be more effective in allocating the resources. Team members and supervisors have a continuous oversight throughout the project, know where they stand at any given time and therefore there is more transparency and less confusion. All of this can be achieved by a realistic project plan.

2.2.2 Our Planning Strategy

First of all, we had to consider a few important factors to start our planning. Our overall budget was 60000 euro which means 600 hours (100 hours for each member) and we needed to distribute the responsibilities in a way that everyone devotes sufficient and equal amount of work. We had to set priorities and make sure that everyone is involved.

The second thing that we did was to break down the work into smaller work packets, assigned them to team members based on their preferred roles and schedule. Finally, we made our overall plan based on those individual plans.

As you can see in the figure.2.1[1], everybody's tasks are broken down to detailed packets and if they need an assistance from any other member, it is shown with the arrows.

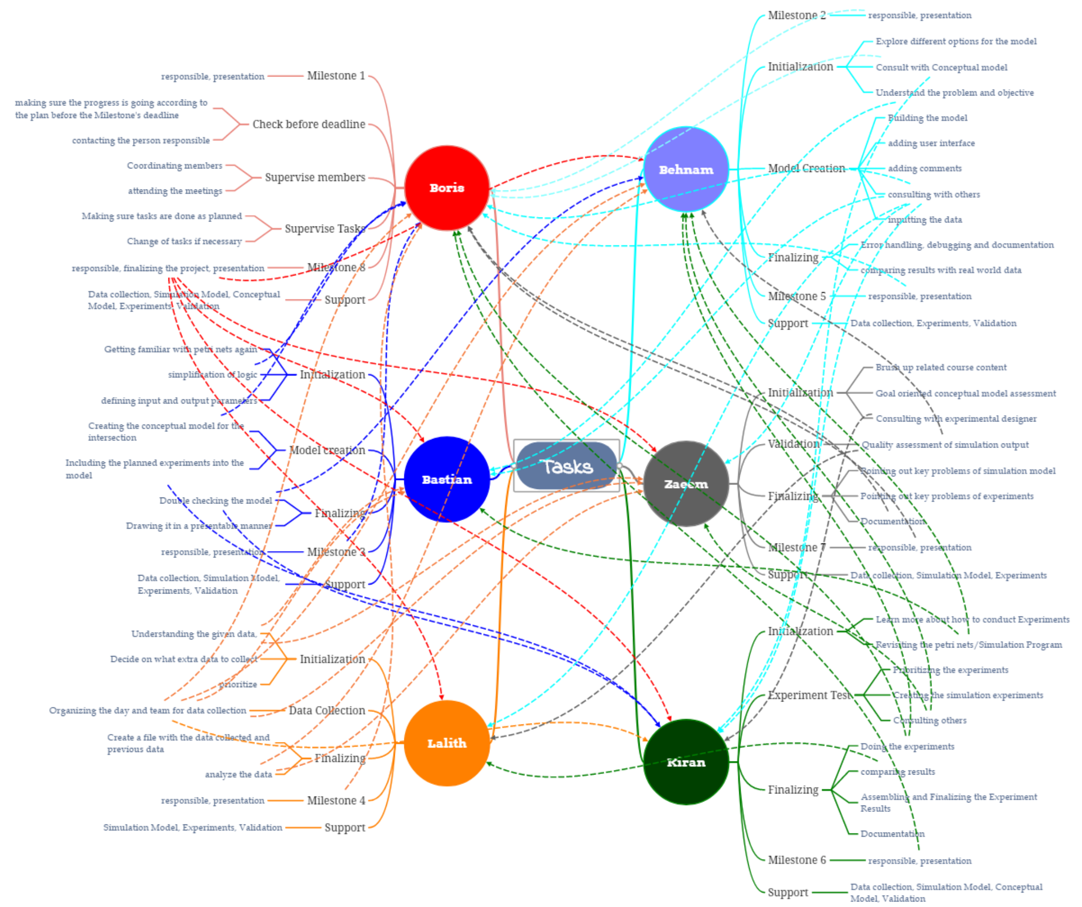


Figure 2.1: Task assignment Mind Map.

Also, our detailed overall plan is shown in figure2.2[2]. Work packets are color coded and connections between work packets indicated dependency and sequence. There are also some symbols to show the status of a work packet (ongoing, finished and etc.)

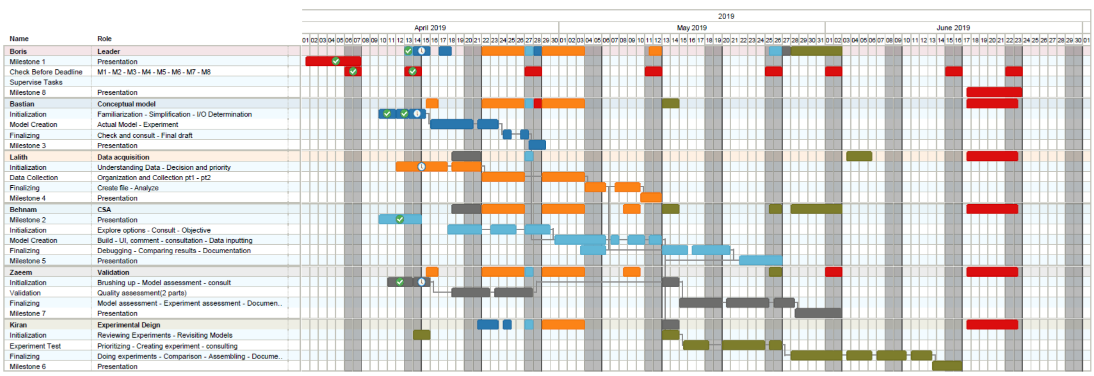


Figure 2.2: Overall plan for the project Gantt chart..

[1]It is created in the XMind software. https://www.xmind.net/

[2]The website used to create the Gantt chart: https://plan.tomsplanner.com

2.2.3 Budget plan

The overall budget, which was 600 working hours in total hat to be allocated for each milestone of the project to indicate an estimate of the required time and workload. As you can see the in the figure 2.3 and figure 2.4 these are our initial estimates of the working hours per milestone.

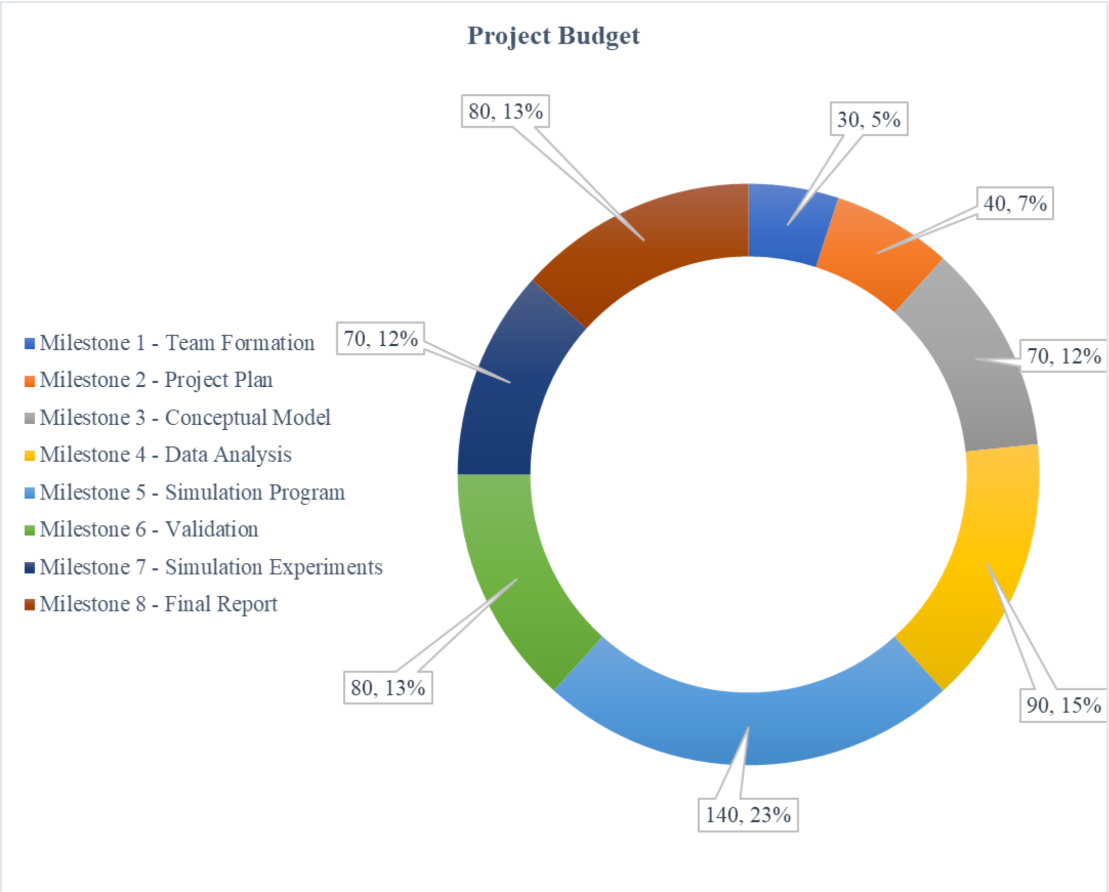


Figure 2.3: Project Budget pie chart [Format of the data labels: number of hours, percentage of the budget]

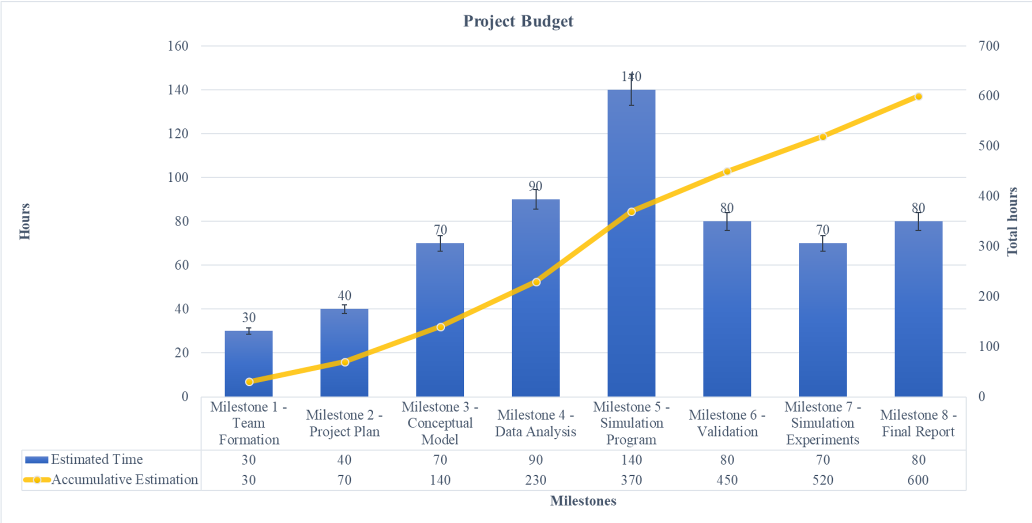


Figure 2.4: Project Budget bar chart with trend line

2.2.4 Problems encountered

- Planning takes a significant amount of time and we did not dedicate enough time for it at the beginning.
- Each person has their own perspective when it comes to planning and combining all these individual plans into an integrated plan proved to be challenging

To solve the above-mentioned problems, it is advisable to start planning as early as possible and to have constant communication between team members.

2.3 Conceptual Model

This chapter covers the creation of the Conceptual Model for the intersection. First, the assumptions made for creating the model and their respective derived simplifications are stated. Also, the justifications of said assumptions are given.

Next, the conceptual model which was created based on those simplifications is shown and described. The conceptual model is represented by a Petri Net. It displays the four different incoming streets and the traffic flow of the intersection.

Additionally, the measures that need to be taken for creating the simulation model out of the conceptual model are shown and described. This description is followed by the explanation of the measures that will be used as validation for the conceptual model to make sure it actually represents the real situation of the intersection.

Finally, the measures that will be used as results for the simulated experiments will be stated.

2.3.1 Simplifications

The following table (2.1) shows the simplifications made for creating the conceptual model. The main goal of the simplifications are the creation of the conceptual model and to check whether the simulation model is feasible within our limited resources. It is wort noting that the conceptual model should represent the actual situation as closely as possible.

Each simplification means loss of information, which will make the model stray further form the real situation. Nevertheless, implementing certain details into the model will over-complicate it and would require a large amount for work for only a small information gain. Implementing such details would decrease the clarity of the model.

The model should be as simple and realistic as possible. Simplifications often have a contrary impact on those two requirements. For each simplification made it was an assessment between said pros and cons. Balancing between these pros and cons is not always a clear decision.

2.3.2 Petri Nets

To describe the conceptual model as a Petri Net, first the events need to be determined. These events will later be shown as transitions in the Petri Net. These events are listed below in 2.2.

Each of the four incoming lanes of our considered intersection has a different lane system. So, for every lane a different conceptual model for the incoming cars had to be designed.

In the following figure 2.7 the petri net represents the conceptual model of vehicles arriving from "Am Krökentor" from North-West direction (2.7). The Petri net of all lanes and the whole intersection can be found in the appendix.

The traffic light in the petri net are simplified and represented by a red transition. Each of those red transitions stand for the mechanism described in 2.5. When the light turns green, which is based on the light phases measured, the transition which allows tokens that represent vehicles to cross the lights will get inhibited. Every light fires tokens based corresponding to the distribution found in the data analysis.

Simplification	**Justification**
All vehicles are considered as the same	• We do not differentiate between Cars/Busses/Trucks/motor cycles since their behaviour does not differ so much. • Considering the difference in behaviour of cars/buses/trucks would make the creation of the conceptual model and the simulation much more difficult without any real benefit. (Also, we only have the data for motor vehicles)
Pedestrians are not considered	• pedestrians are not considered, because there are not so many that they play a role. The intersection should be made better for cars. • As long as the pedestrians keep their light times as they are, it is fine. Our task is to make the intersection better for vehicles • Pedestrians, that hinder cars from crossing are considered. (pedestrians don't get their own tokens)
Only peak hours (early and late) are considered	• Only the critical hours are considered, since the most waiting time is happening there. • When the efficiency of the peak hours is improved, most probably the overall efficiency will be improved.
Only workdays and workweeks (not holidays)	• When the efficiency of the work days is improved, most probably the overall efficiency will be improved.
The time a car needs to start moving and clear the intersection can be averaged.	• The vehicles need a similar time to leave the intersection and therefor it will be averaged. No distribution for leaving times will be considered.
The inductive sensors in front of the traffic lights are not considered	•The observation of the traffic light phases showed, that the phases are fixed and do not rely on the sensors at all.
Unforeseen events are not considered	• Unforeseen events either do not happen often enough (e.g. accidents, based on the data given by the city of Magdeburg) or do not have a big impact onto the traffic flow (e.g.: fire trucks only need a few seconds to cross the intersection).
Bicycles are not considered	• Since there are not as many bicycles, there is almost no waiting time for cyclists. (they only have to wait at the red light, but not in line)

Table 2.1: Simplifications and their Justifications

Event	Type
Traffic light changes from red to green	Primary
Traffic light changes from green to red	Primary
Vehicles arrive at one of the four lanes	Primary
Vehicles pass the lights and start crossing the intersection	Secondary
Vehicles leave the intersection	Secondary
Vehicles blocking the outgoing lane for other vehicles	Secondary
Pedestrian starts crossing	Primary
Pedestrian finishes crossing	Secondary

Table 2.2: Events used and their types

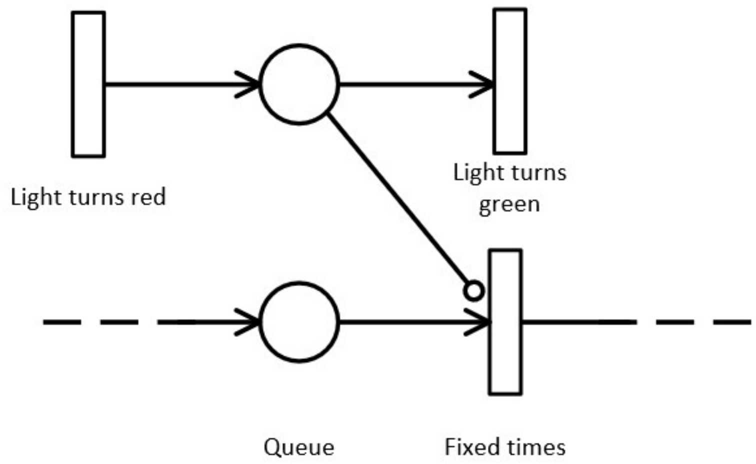


Figure 2.5: Petri Net of a single Traffic Light

For understanding the given petri net, knowledge about the real lane is required:

As one can see in Figure 2.6, the considered lane separates into different lanes right before the traffic lights. One for right turning vehicles only and one from where vehicles can only go straight or turn left. The area marked by the red rectangle shows, where the two separate lanes exist. Behind that the incoming vehicles can wait in only one lane due to parked vehicles on the side of the street. The separate lanes therefore can only take up to maximal seven vehicles.

When one of those lanes contains seven vehicles, vehicles that arrive need to wait behind said queue, regardless of what direction they want to go.

In the petri net shown in Figure 2.7 we find the following places for the tokens, which represent the vehicles: $D1$, $D2$ und $D3$ are the tokens for the vehicles which are currently turning and therefore are on the intersection. D1 is for vehicles turning left, $D2$ for vehicles going straight and $D3$ for vehicles currently turning right.

The place DR is for vehicles wanting to turn right. The places DS and DL are for the one vehicle standing in front of the light for straight and left goers only. DSL is for the vehicles behind that first vehicle in said lane. The place labeled “Queue” is for all the vehicles waiting behind the spot where the lane separates.

The transition named “Vehicles arriving” will create token based on the deter-

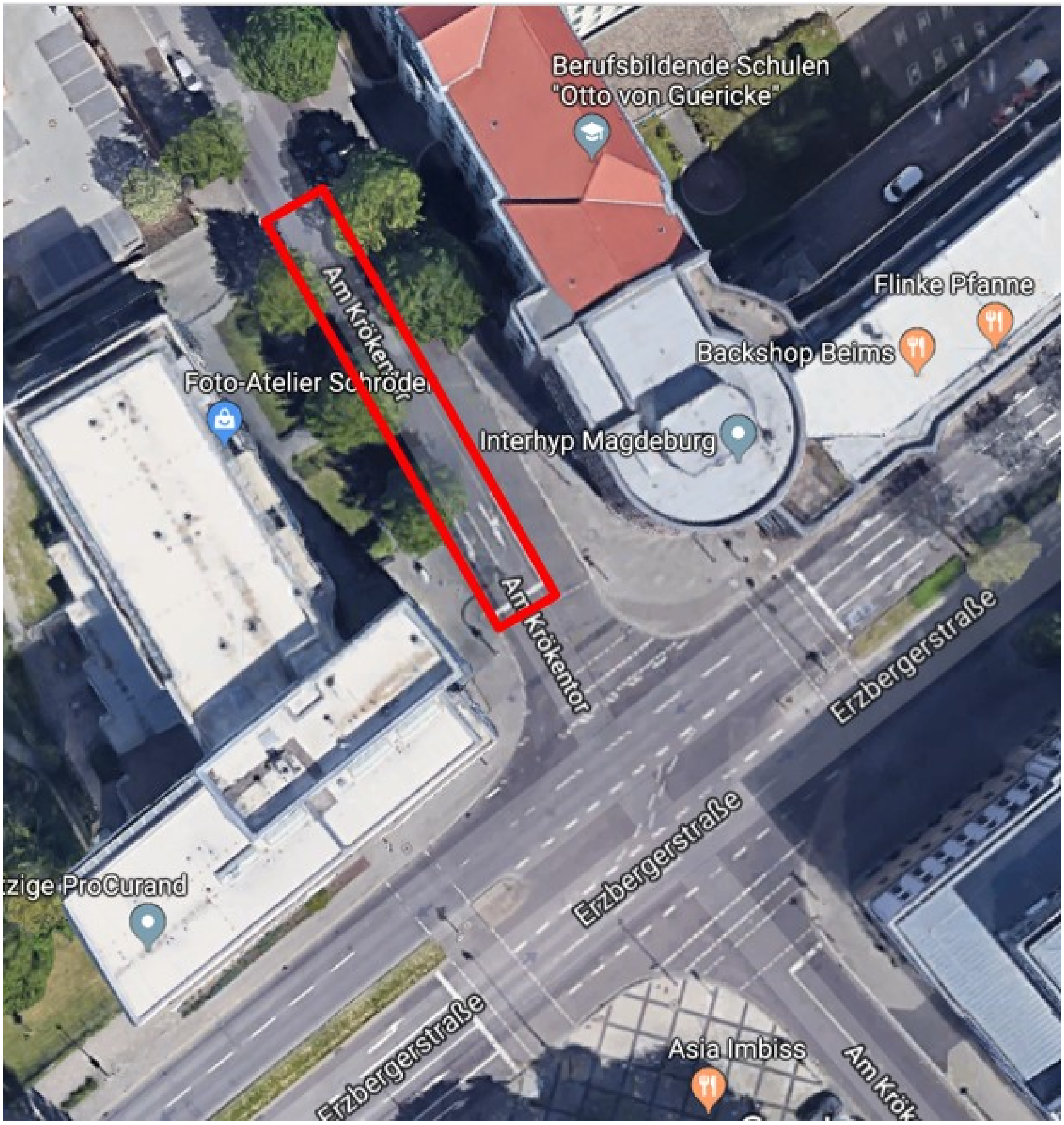


Figure 2.6: The considered intersection, "Am Krökentor" North-West highlighted

mined distribution of inter arrival times for the specific lane.

To make sure that tokens can only then transmit into the places for the separate lanes, after the queue a conditional transition is added $(IFDR < 7AND(DSL + DS + DL) < 7)$. Only when both lanes contain less than seven vehicles, the vehicle coming from the Queue can drive on.

Then based on the probability of going right (Pr) and the summed probabilities of going straight and left $(Ps + Pl)$ the token will go a different path. For the case of going right, it will directly go to DR and wait there in front of the Traffic light (red transition). For the case of going left or straight it will get to the place DSL. Only if there is no token in either DS or DL $(IFDS = 0ANDDL = 0)$, the token will transcend to either DS or DL, based on the respective probabilities for going straight and left $(PsandPl)$. Here the tokens will remain till the transition for the traffic light will start firing.

The traffic light transition will start to fire in a timed manner with a fixed rate. This represents the time needed for vehicles to start driving and actually cross the traffic lights. When having crossed the traffic lights, the tokens will remain in $D1$, $D2$ or $D3$ respectively. The following transitions will also have fixed time rates which represents the time taken for the vehicle to cross. The inhibitor arcs represent the number of vehicles that can be standing on the intersection for each lane. When they inhibit the traffic light transition, too many vehicles are on the intersection and no more vehicles can cross the lights even if they are green.

After the last transitions fire, the token which represents the leaving of the intersection of the vehicle, will be destroyed. These last transitions can be hindered by different events:

Vehicles going right can be hindered by doing so by passing pedestrians ($F7$). The first transition of F7 will fire a token when a pedestrian starts crossing. This token will be destroyed by the second transition which represents the pedestrian finishing crossing.

The same mechanism applies to vehicles trying to turn left with the pedestrian light F8. These left goers also can get hindered by doing so by vehicles that come from the opposite direction going straight ($IFC2 > 0$).

Vehicles trying to go straight can get hindered when the vehicle in front of them tries to go left but gets hindered by vehicles coming from the opposite direction ($IFC2 > 0ANDD1 > 0$).

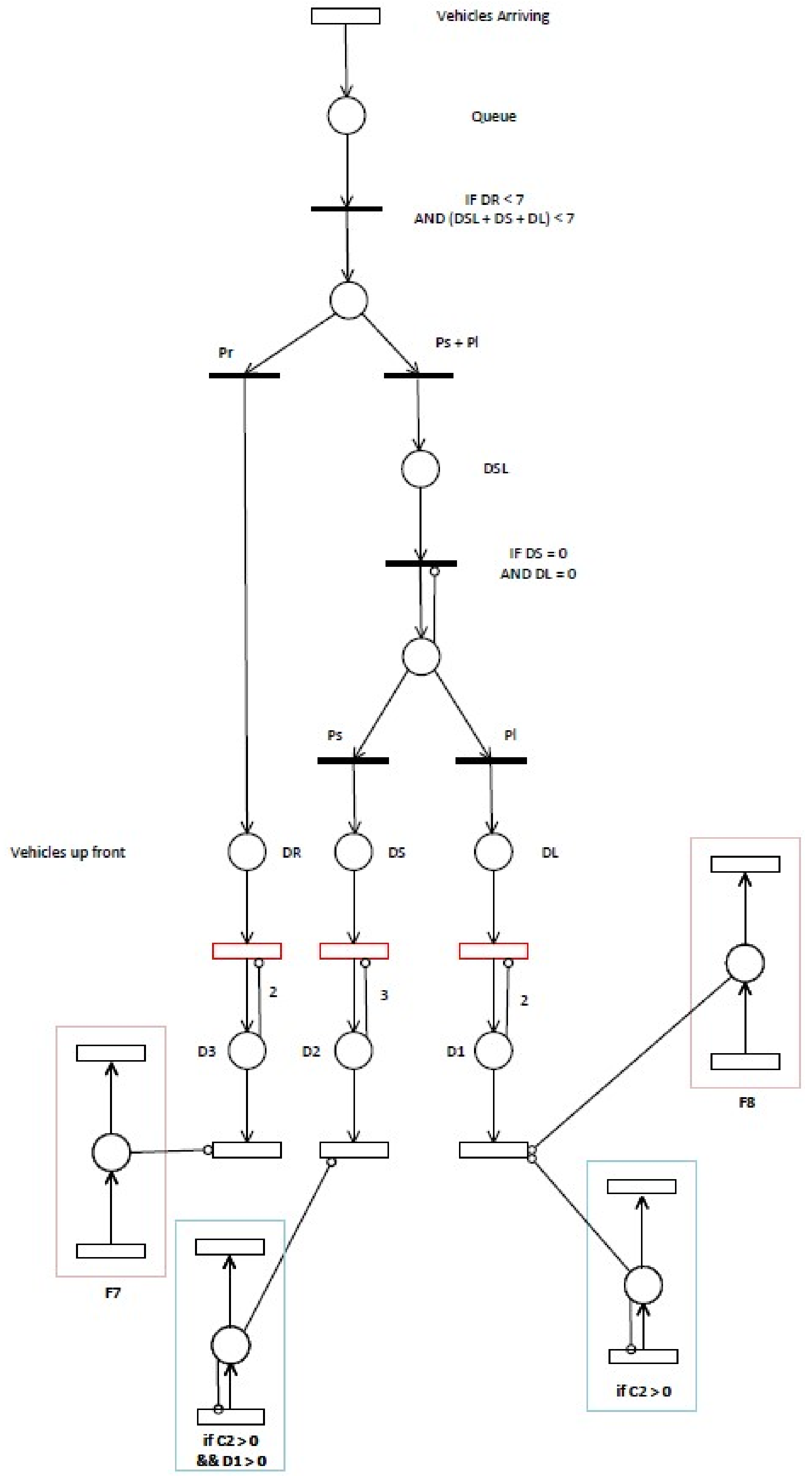


Figure 2.7: Petri net representing incoming vehicles from "Am Krökentor" North-West

2.3.3 Measures Required for Creating the Simulation Model

For transferring the conceptual model into the simulation model and make actual use of it requires different measures. These are:

- Data used from the files given by the city:
  - How many vehicles arrive in the peak hours and where do they go?
  - Light phases.
- Data that needs to be measured at the site:
  - How many times per hour are cars stopped by pedestrians that have green light?
  - How many times do pedestrians cross?
  - How How long does it take them?
  - How How many times per hour are cars stopped by cyclists that have green light?

2.4 Data Analysis

Data analysis is a process of obtaining raw data and analyzing it to answer questions, test hypotheses or disprove theories. Here, data analysis is done for two reasons, to give input to the simulation model and to check the output from the model with the real world data. If the output data from the simulation model represents the real world scenario, then any modifications done to the model would help in predicting the future behavior of the real world system when the modifications are implemented in the real world. This is clearly explained in the Chapters 'Validation' and 'Simulation Experiments'. So, it is really important to be careful while collecting and analyzing the data .The data was collected during high traffic hours (i.e. 16:00-17:00) since higher waiting times were observed during these hours.

2.4.1 Input and Output Variables

Input variables are the variables given as input to the simulation model and output variables are the variables obtained from the simulation model. As discussed before, data is one of the most crucial parts of the simulation. So, it is really important to know which input and output variables to consider and also how to implement them in the simulation program. After some consideration, we came up with the following input and output variables.

Input variables:

- Traffic light phases during rush hour.
- Turning probability at every street.
- Inter-arrival times of vehicles.
- Average time taken by pedestrians to cross the street.

Output variables:

- Queue length.

2.4.2 Traffic Light Phases

The traffic light duration for all the lights are measured using a stopwatch and are represented in an excel sheet as shown in the figure 2.8

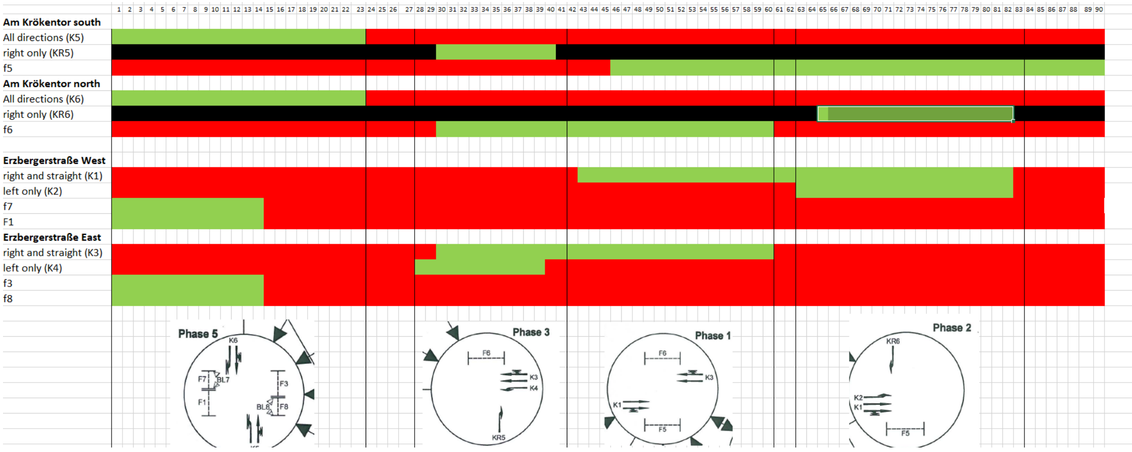


Figure 2.8: Traffic Light Phases

2.4.3 Turning Probabilities

The probability of a vehicle turning or moving from one street to another was measured for all the four streets by counting the cars manually. The calculated turning probabilities are shown in the table 2.3.

Street name	Direction	Probability
Krokentor (NW)	Left	0.032
Krokentor (NW)	Straight	0.262
Krokentor (NW)	Right	0.704
Erzberger (NO)	Left	0.164
Erzberger (NO)	Straight	0.737
Erzberger (NO)	Right	0.097
Krokentor (SO)	left	0.060
Krokentor (SO)	Straight	0.105
Krokentor (SO)	Right	0.839
Erzberger (SW)	Left	0.037
Erzberger (SW)	Straight	0.948
Erzberger (SW)	Right	0.014

Table 2.3: Turning Probabilities

2.4.4 Inter-arrival times of vehicles

The inter arrival times are collected so that we can find out the probability distribution of cars entering the system. These are collected for all the sources i.e. Am Krökentor(NW), Erzbergerstraße(NO) , Am Krökentor(SO), Erzbergerstraße(SW). Based on the collected data histograms were drawn. The purpose of histogram is to

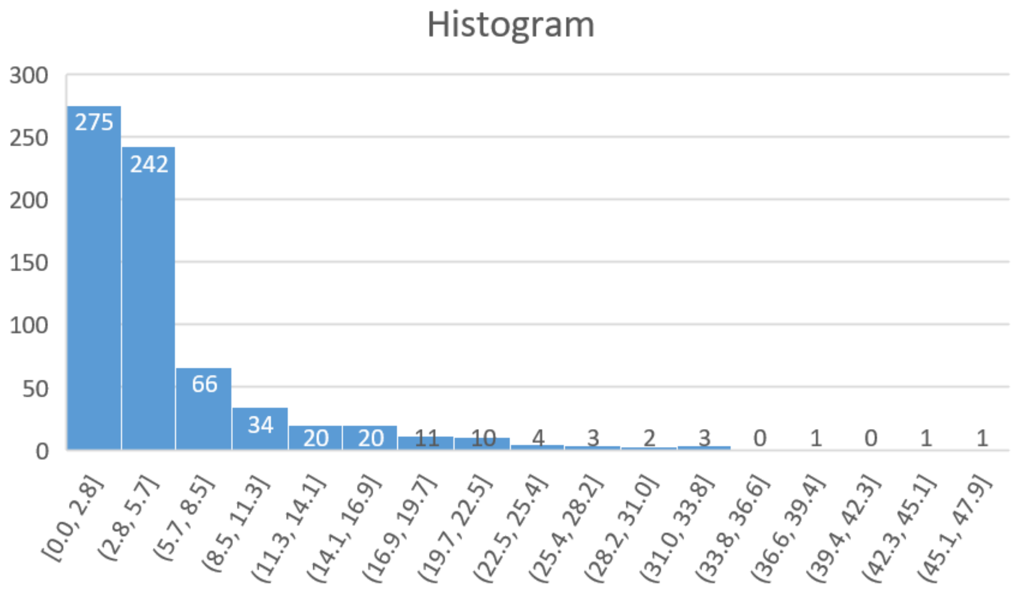


Figure 2.9: Histogram of inter-arrival times of vehicles at Am Krökentor (NW)

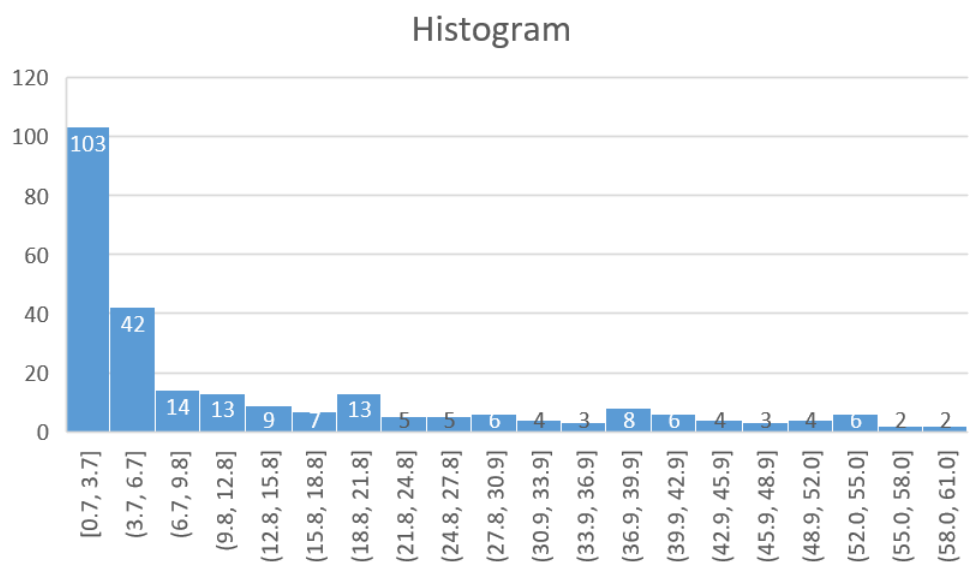


Figure 2.10: Histogram of inter-arrival times of vehicles at Am Krokentor (SO)

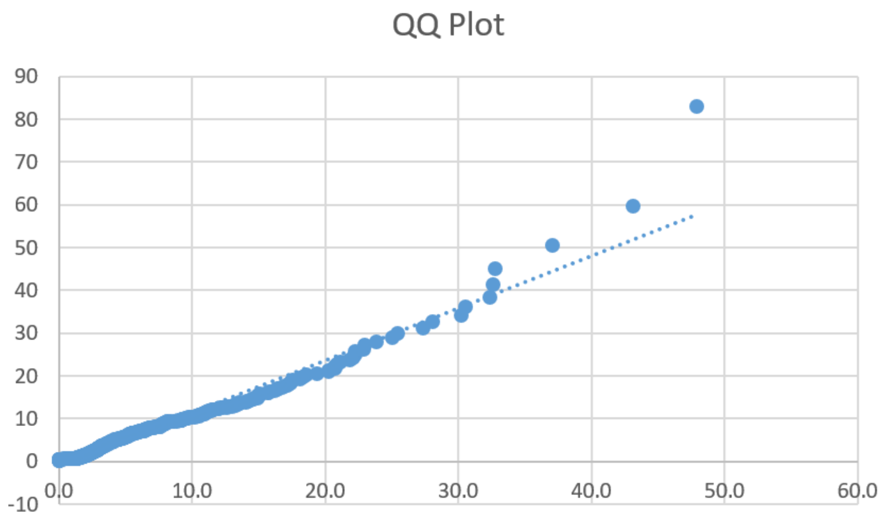


Figure 2.11: Quantile-Quantile (Q-Q) Plot for distribution at Am Krokentor (NW)

help us estimate the type of the distribution. From our histograms the distributions of all sources seamed to be the Log-normal distribution. The histogram for Am Krökentor (NW) is shown in the figure 2.9 and The obtained parameters for all the sources are shown in the table 2.4

Street name	Mean	Standard deviation
Krökentor (NW)	1.12	1.00
Erzberger (NO)	1.70	1.20
Krökentor (SO)	2.13	1.33
Erzberger (SW) lane 1	1.50	1.14
Erzberger (SW) lane 2	1.40	1.20

Table 2.4: Mean and Standard deviation for all the sources

In order to check the correctness of our guess which is Log-normal distribution, we have drawn quantile-quantile plots. Since the plots obtained are almost straight lines and the lines pass through the origin, we can conclude that our guess is sufficiently accurate. The quantile-quantile plot for distribution at Am Krökentor (NW) is shown in the figure 2.11.

The chi-square test is test carried out to compare whether the distributions from our quantile-quantile plots are similar enough to the real data. If a chi-square test accepts the hypothesis it does not mean that the data is correct, it just means we have not been able to disprove it. It also infers that the difference between the two data sets is not large enough for it to be unlikely that they have the same distribution. The expected values and observed are plotted against observed values in figure 2.13 for comparison.

Average time taken by pedestrians to cross the road

We assume that there is always someone wanting to cross the street for every cycle during rush hour.The time taken by pedestrians is measured using stopwatch for 30

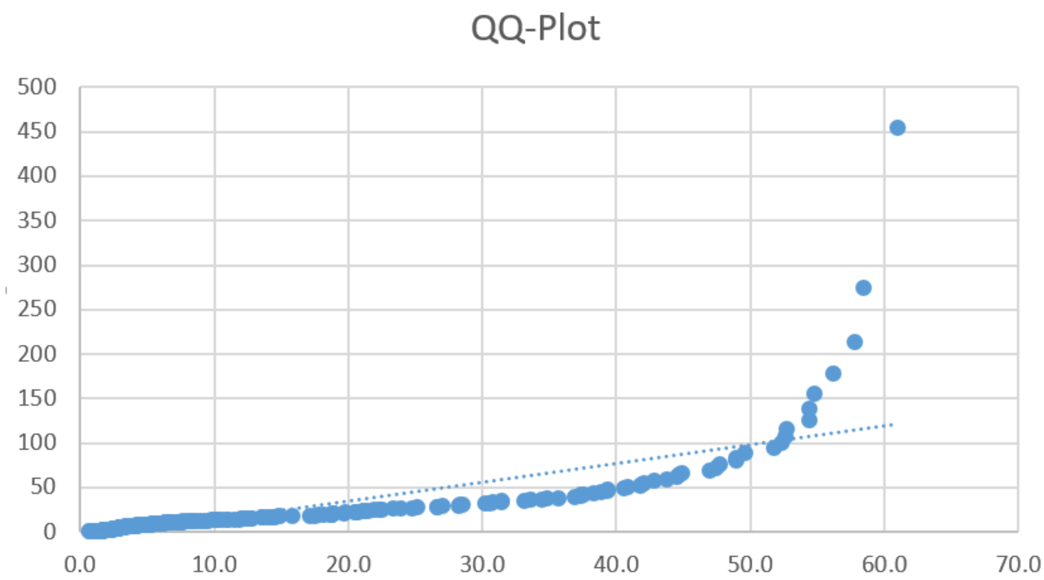


Figure 2.12: Quantile-Quantile (Q-Q) Plot for distribution at Am Krokentor (SO)

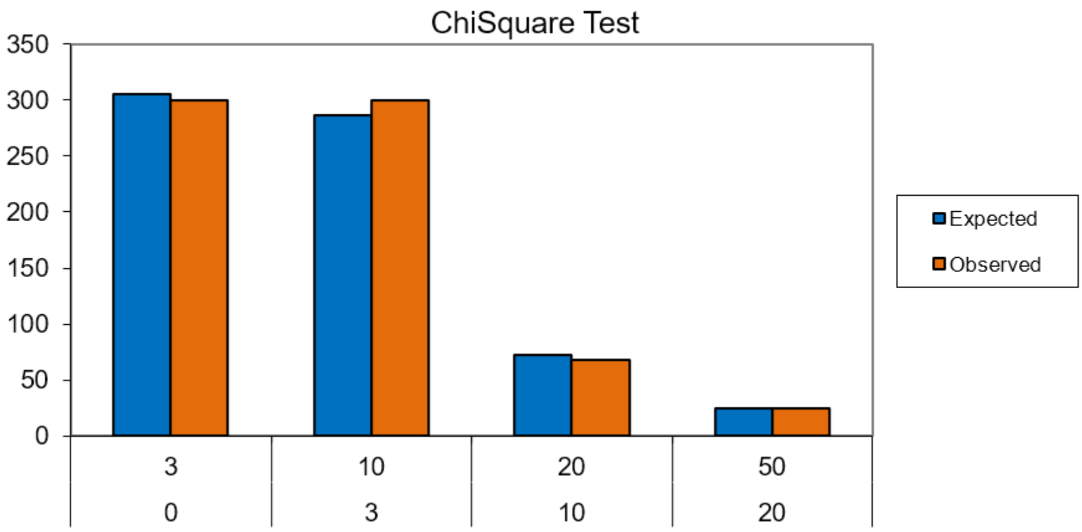


Figure 2.13: Chisquare test - Observed vs Expected values, Krökentor (NW)

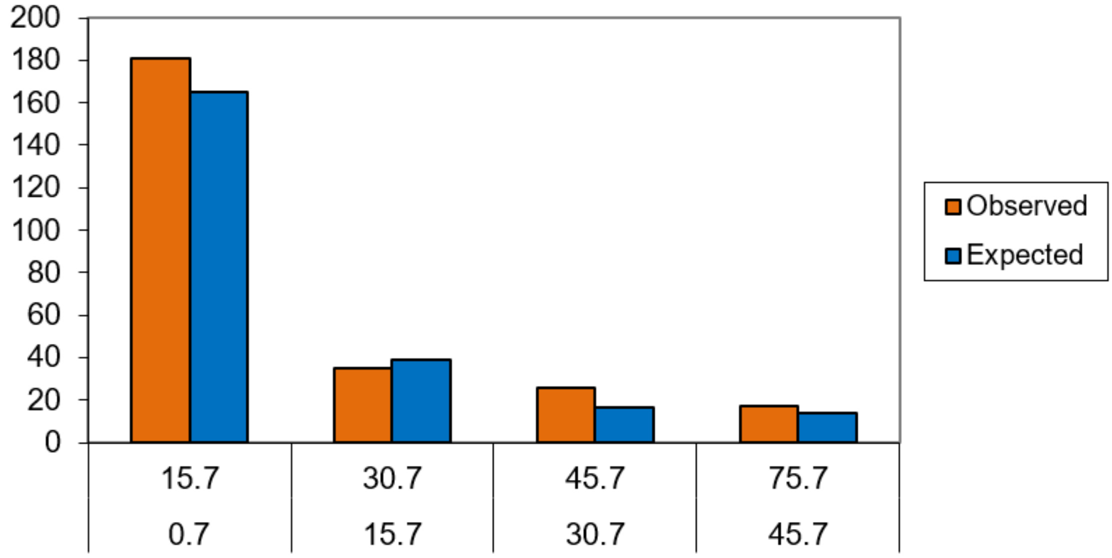


Figure 2.14: Chisquare test - Observed vs Expected values, Krökentor (SO)

cycles and averages are calculated and are shown in the table 2.5.

Street name	Avg. time taken
Krökentor (NW)	12 sec
Erzberger (NO)	15 sec
Krökentor (SO)	9 sec
Erzberger (SW)	18 sec

Table 2.5: Avg. time taken by pedestrians to cross the road

2.4.5 Queue Lengths

Queue Length is used to validate the model. The output from simulation model is compared to the collected queue length to test the validity of the model. The Average queue lengths were measured for all the streets by counting the vehicles waiting at the queue manually for every cycle. The average queue lengths are given in the table 2.6.

Street name	Left lane	Middle lane	Right lane
Krökentor (NW)	N/A	3.10	5.50
Erzberger (NO)	1.10	1.70	1.80
Krökentor (SO)	N/A	4.30	N/A
Erzberger (SW)	0.20	7.20	0.10

Table 2.6: Average Queue Lengths

2.4.6 Difficulties encountered

- Bad weather conditions.
- Distractions while collecting data.
- Lack of time. It was difficult to fix a common time slot to collect the data among team members.

2.4.7 Limitations on accuracy of data

- We only measured the data few times during rush hours. The rush hours were determined from the data given to us by city planning office. So, it might not be the same for entire day.
- We collected the data manually. We did not have any particular equipment to count vehicles, speed etc.. Thus, the obtained data can not be considered to be are 100% accurate.
- We assumed the intersection to be isolated. But in reality, it is effected by other intersections.

2.5 Simulation Program

The next step after drawing a conceptual model is building the simulation model and entering the data into the model. The simulation model is based on the conceptual model ,however many details which are not incorporated in the conceptual model must be considered in the simulation model. In each part of the program these differences will be explained.

The input data is used to for verification of the program and also to determine whether the program is to a certain extend representative of the real world.

In this part, different parts of the simulation model and program will be explained. All the simulation program is designed and implemented with a simulation modeling tool called AnyLogic. Version 8.4.0 was used in this project.

2.5.1 Traffic node design

The first thing that needs to be created is the road network and underlying route. It is worth mentioning that the node is considered to be isolated and connected traffic nodes are not going to be modeled.

First things that were not considered in the conceptual model are details of the road such as length, width, angles and etc. These details were calculated to an adequate precision based on the map of the node and also satellite pictures from Google Maps[3] and Bing Maps[4].

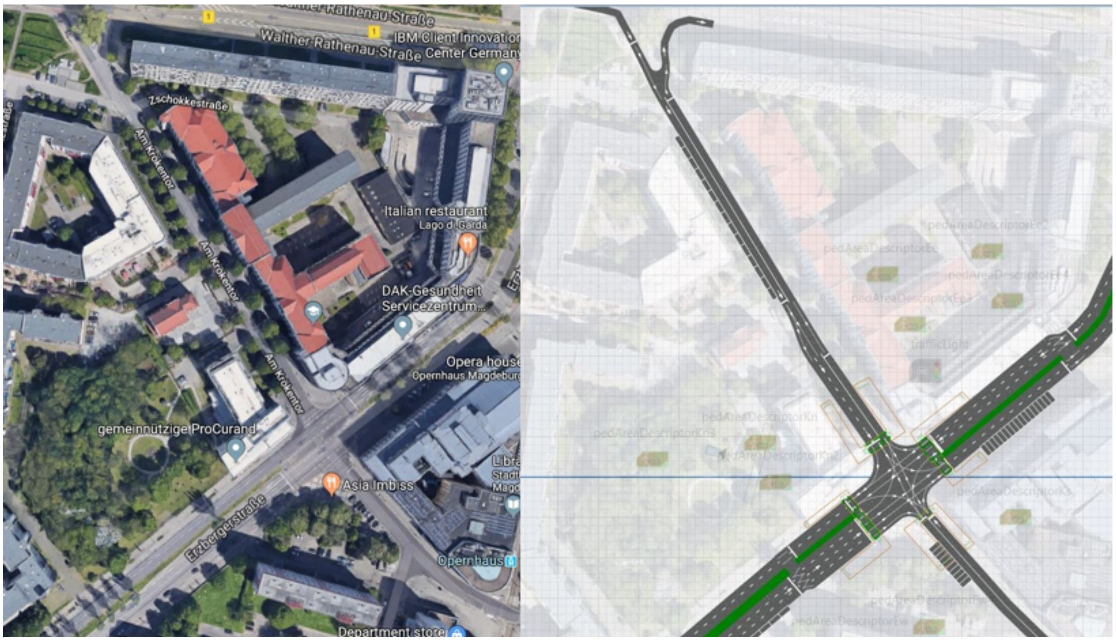


Figure 2.15: a) the satellite picture from Google Maps and b) road design in the simulation model.

The other addition to the simulation model in comparison to the conceptual model are the parking lots. Parking lots on Am Krökentor(SO) and on Erzberger Straße(NO) are actual parking lots that are added to the model.

[3] https://www.google.com/maps
[4] https://www.bing.com/maps

On the other hand, the parking lots on Am Krökentor(NW) and on Erzberger Straße(SW) are actually lanes which are almost always full in the rush hour. Since the model is built for the rush hour, the lanes are replaced by parking lots for simplification.

In the conceptual model lanes and different roads are not accurately represented but they are all carefully added to the model. As you can see in the picture 2.16 the red square shows the parking lot that replaced a lane in the Erzberger Straße and the green square indicates the divergence in from two lanes into three in the other end of Erzberger Straße.

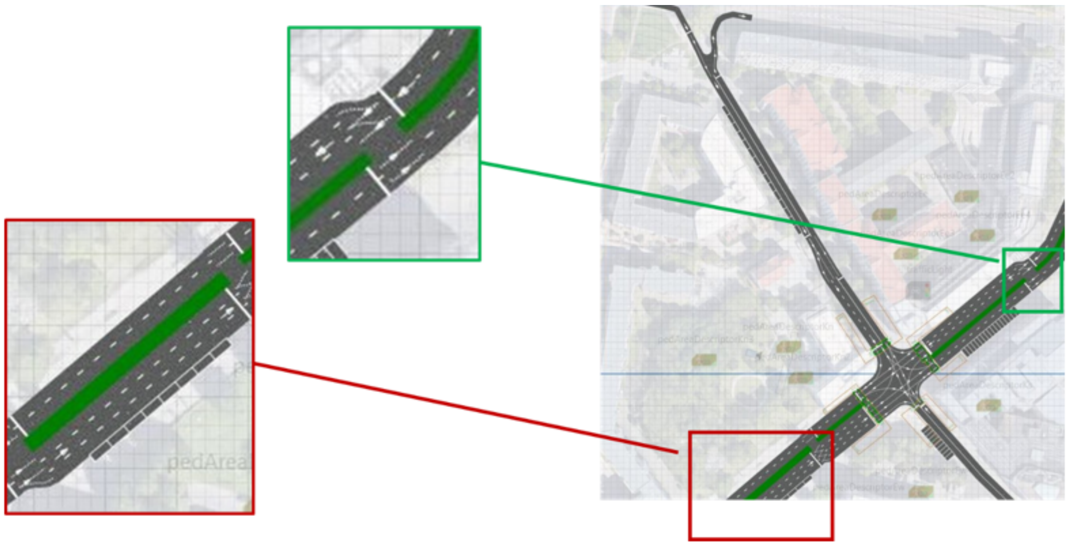


Figure 2.16: Parking lots, lane and street design in the simulation model.

2.5.2 Cars' Logic

As you can see in the picture 2.17, the logic behind the program is relatively simple. Each street (in one case each lane) coming to the intersection has a car source block. In this block, the car specifications, the calculated inter-arrival times and a custom car population are specified.

The select output(selectOutput) blocks are used to guide the cars into different directions based on the probabilities also calculated in the data analysis chapter. After the cars are picked by this block (turn in the real-world sense) they enter the car move to block(carMoveTo) and continue their way to the end of that road. If this way is blocked, not connected or the cars are for any reason not able to follow this path, there will be an error. After reaching their destination they will enter the carDispose block and the car agent will be destroyed (leave the node).

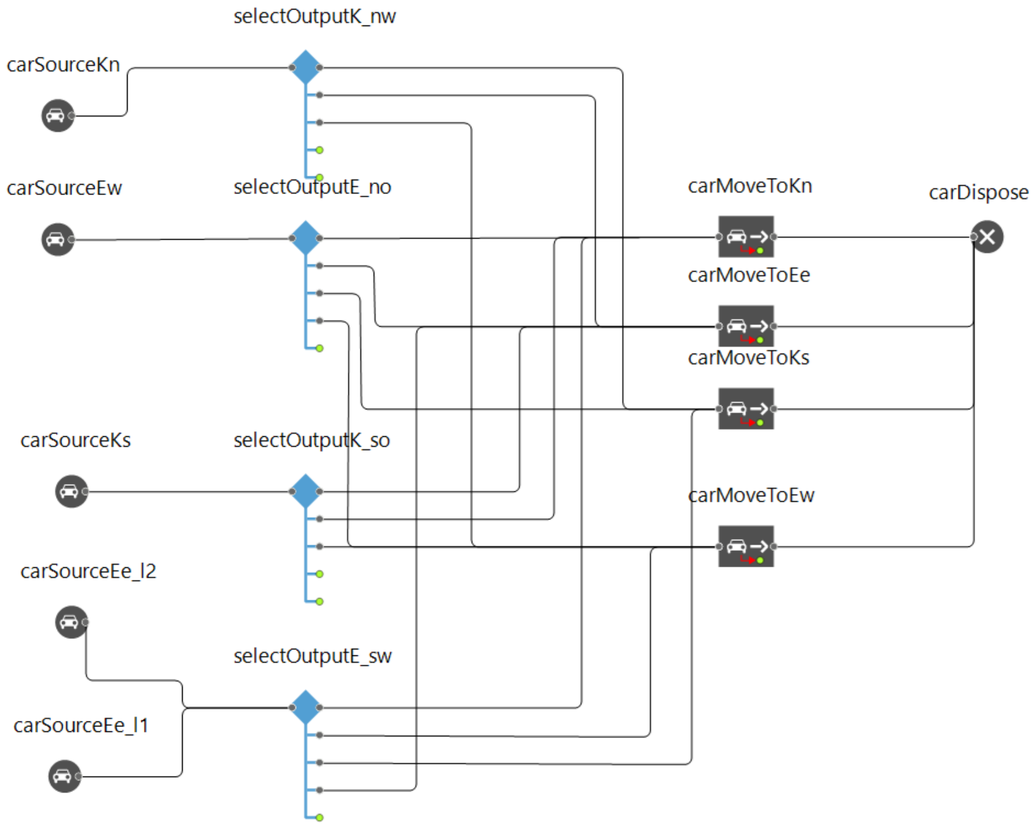


Figure 2.17: Programming logic behind the cars' movement.

2.5.3 Directions and Path

In the actual intersection cars move in 12 major directions and plus two U-turns (which were not considered in the conceptual model). The way cars react in the middle of the intersection and also how they move from one of these directions to another were not considered in the conceptual model either, and these details were designed in a way that is most representative of the model.

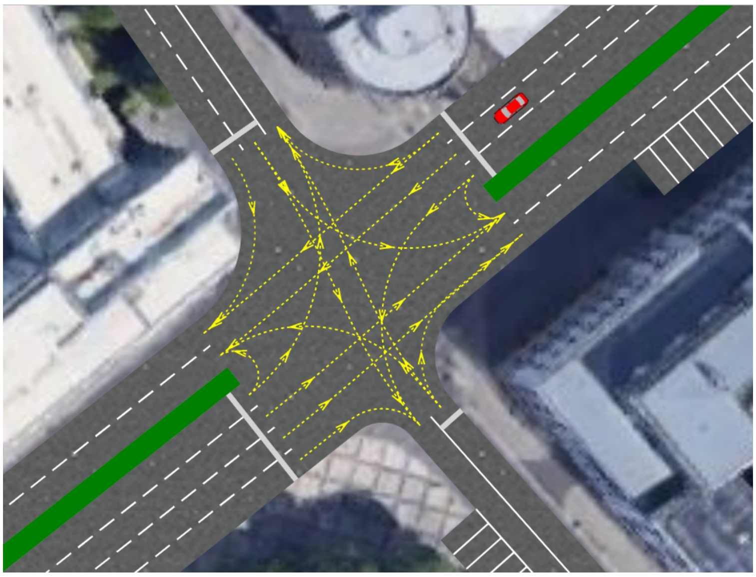


Figure 2.18: All the directions that cars can have in the intersection and also the way they move in the middle of it.

2.5.4 Traffic Lights

The timing for the traffic lights were very complicated, other than the major phases there were short period of time which the lights were not categorized in any of those phases. Also, adding the pedestrian lights made the timings even more detailed. To address this, The whole traffic cycle with duration of 90 sec was broken to 12 phases and also instead of the stop lines, lane connectors were used to control the traffic.

In the picture 2.8, the actual phases are shown. The diagrams below show the major phases and black lines show the minor phases.

In the picture 2.19, the AnyLogic traffic light phases are shown. The pedestrian light phases are coded in the on phase change box in the bottom of the picture.

Figure 2.19: Traffic phases in AnyLogic simulation program.

2.5.5 Pedestrian

There are certain design elements to restrict the movement of the pedestrians in AnyLogic such as Wall, Target Line and Area. These elements were carefully designed in the model in a way that is most representative of the pedestrian movements in the model.

In addition to that, extra stop lines were added to all the roads to make an interaction between cars and pedestrians meaning that when a pedestrian is crossing the car will slow down or stop and let the pedestrian pass and then resumes its movement.

Figure 2.20: Pedestrian design elements.

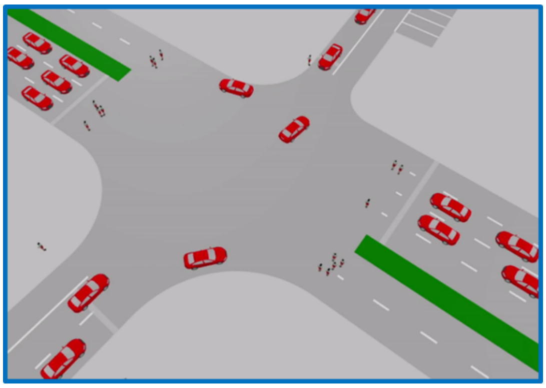


Figure 2.21: Cars stopping when the pedestrians are crossing.

In order to calculate the speed of the pedestrians, the time taken by the pedestrians to cross each street were sampled and based on the distance their movement speed was estimated. The inter-arrival time for pedestrians were estimated in a way that there are always pedestrians ready to cross when the light turns green, but not in a way that makes the node crowded as it is in reality.

2.5.6 Model Verification

The model was modified several times in order to eliminate the errors such as running errors, cars getting stuck in the middle of the model, cars going over each other or pedestrians and so on. The team looked at the model to see if it is representative of the real world and the model was compared to pictures and media taken from the model.

The number of generated cars were compared to the real data. Queue lengths were compared. After considering all the above-mentioned measures, we concluded that the model is verified.

2.5.7 Overview and problems Encountered

The main problem in designing the Simulation model was putting the data that was measured in the data analysis milestone into the model. Initially, there were a lot of irregularity after we introduced the data. Some of the issues were resulted because of the distributions and others were small mistakes in the design of the model. Verifying the data took a lot of extra effort and time.

Another problem that we encountered was that we were not able to work in parallel and participating in the design was usually done by only one or two team members. This was mostly due to programming challenges that we faced for instance which block should contain certain code lines.

In order to solve the issues, we frequently met and discussed the issues and explained each part to other members so that everyone is aware and can participate in solving them. Aside from a lot of time that this took, we were successful to solve the problems.

2.6 Validation

This chapter covers the results of our validation and explains the methodology behind the validation process. Additionally here we also give certain justifications and explain why we have a valid model.

2.6.1 Validation rationale

Validation represents the ultimate test of a model i.e. can it successfully predict the future behavior of the real-world system we are modeling. In other words, the structure of the model should be accurate enough for the model to predict the behavior of the real-world system given the inputs of interest. During the process of validation, the model is essentially viewed as a black box, to which we give certain inputs and evaluate the outputs. In order to validate the model with a greater degree of certainty we will be using input data that was not directly put into the model as input but can be used to compare the model with the real-world system we are modeling. To that end we have chosen to use the queue lengths of the lanes as an input for validation.

2.6.2 Methodology

In order for us to compare the queue lengths obtained from our model with the ones from the real-world system we needed to gather the required real- world data. For our purposes we gathered queue lengths on the 25th of April 2019. The data collection was executed by the entire team standing at certain points of the intersection and manually counting cars, which to their respective counter, seemed to be in a queue.

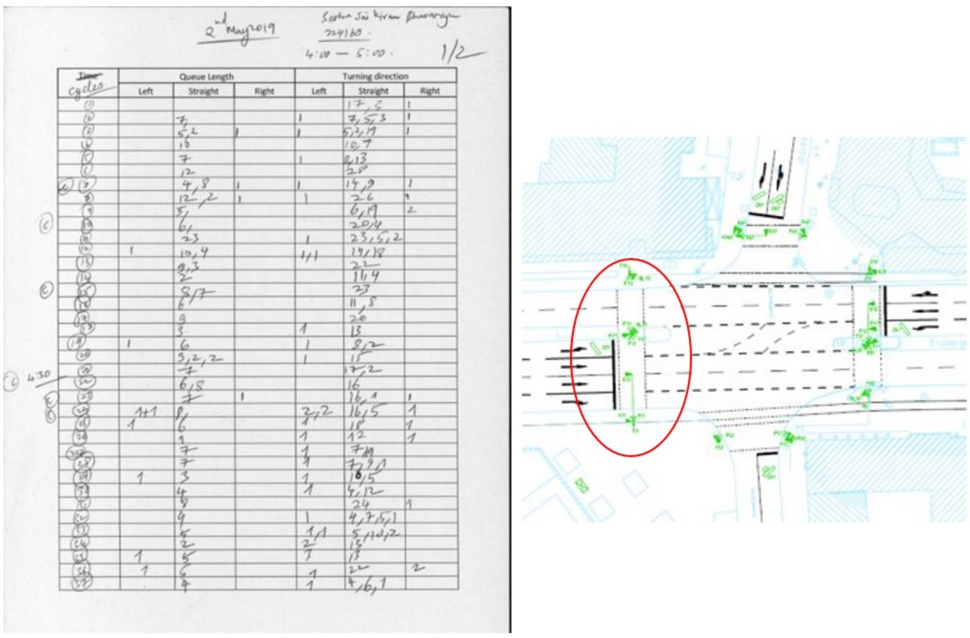


Figure 2.22: Filled form for Erzbergerstraße west

The data was gathered for each lane individually and noted down within a table. On the left-hand side of Figure 2.22 we have one of the forms used in the data collection. The columns represent the individual lanes of the street (on this image we can see the form and data for Erzbergerstraße West) and the rows represent one light change from red to green. The actual process of counting the queue lengths consisted of counting the number of cars that appeared to be in a queue when the light was red. Then when the light changed to green we would note down the number of cars within the respected queue and move to a new row.

2.6.3 Model modifications

In order to obtain the required information from our model as well as to improve its validity certain changes were made to the base model. One of these changes was an addition of a broadcasting function within the car agent, which served as a way to gather and calculate the appropriate queue lengths in our model. Following form this we made certain changes in how we represented the road layout within our model due to difficulties that accorded while trying to extract the necessary information from our model. These changes were implemented on the road representing Erzbergerstraße west (as represented in the technical drawing) as can be seen in the figure 2.23

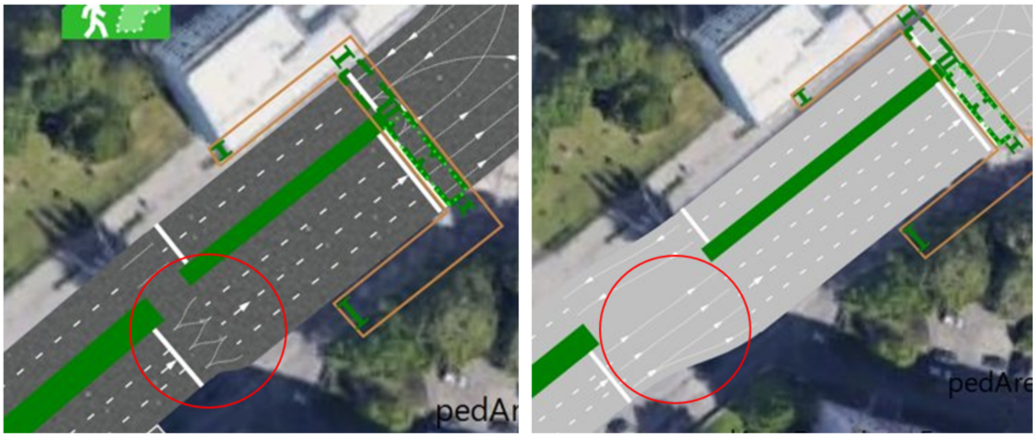


Figure 2.23: Model modifications for the purpose of validation

The image on the left-hand side is taken from our base model while the one on the right shows the modifications that we implemented. The reason for the modification was due to the way Anylogic enables its car agents to change lanes. This element of the simulation software ended up creating a disconnected queue, due to the agents inability to switch lanes in an appropriate manner. This disconnected queue caused problems with our efforts to retrieve the required information for our validation purposes. To mitigate the problem, we made some slight changes in how our car agents could traverse the road. These changes however are still in line with the real-world system and do not make the representation of Erzbergerstraße west in our model any different from the one in the real-world.
Additional changes were made to one of the distributions that was calculated for the inter arrival times of our car agents. Through some trial and error, we found out that changing the distribution of Erzbergerstraße East (as depicted in the technical drawing) from Log-normal (1.1, 1.2, 0.04) to Log-normal (1.5, 1.14, 0.04) improved our models validity i.e. better represents the real-world system.

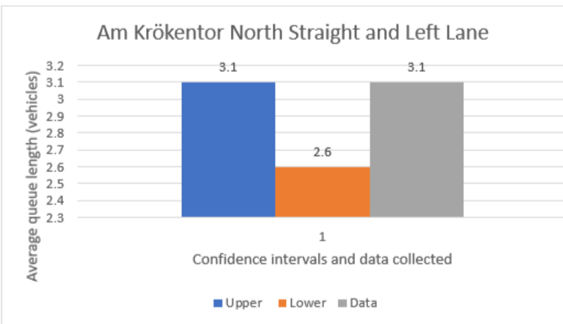
2.6.4 Results of the validation

The following results were obtained by running 101 replications with a level of significance α= 0.1. The model which produced these results has the aforementioned changes implemented. The results are presented in table 2.7.

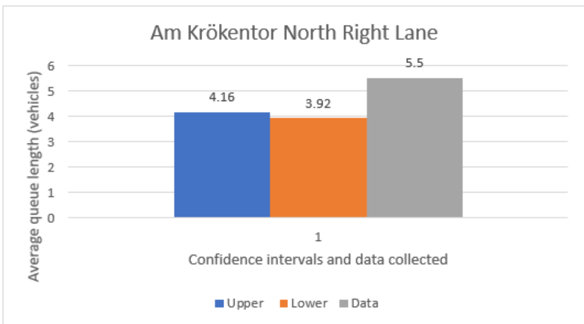
Street	Lane	Model mean	Upper	Lower	Data	Difference
Am Krökentor South	RLS	4.40	5.16	3.92	4.30	/
Am Krökentor North	R	4.04	4.16	3.92	5.50	1.34 less
	SL	2.74	3.10	2.60	3.10	/
Erzbergerstraße East	R	2.76	2.84	2.53	1.80	0.73 more
	S	3.01	3.10	2.90	1.70	1.2 more
	L	2.16	2.84	2.05	1.40	0.6 more
Erzbergerstraße West	R	0.11	0.12	0.09	0.10	/
	S	8.05	8.24	7.85	7.20	0.6 more
	L	0.60	0.62	0.57	0.20	0.39 more

Table 2.7: Results for Validation

Table 2.7 displays the results of our validation for every lane. In the column Data we can see whether the data we have obtained from the real-world system is within the confidence interval from our simulation model (green means within confidence interval, read means out of the confidence interval). The difference column shows the difference of a particular lane from the appropriate confidence interval as well as if it is more or less then what we have observed from the real-world data.
The following charts give a better visual representation of the confidence intervals per lane.



(a) Am Krökentor North Straight and Left



(b) Am Krökentor North Right

Figure 2.24: Confidence intervals for Am Krökentor North

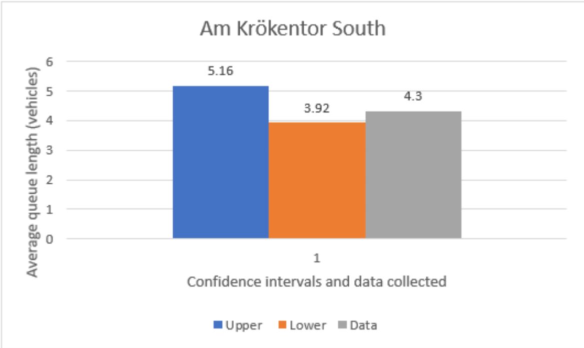
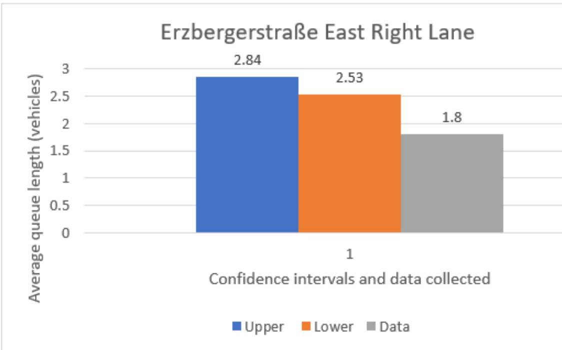
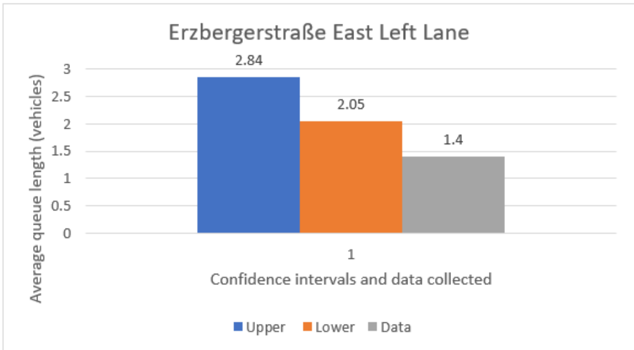


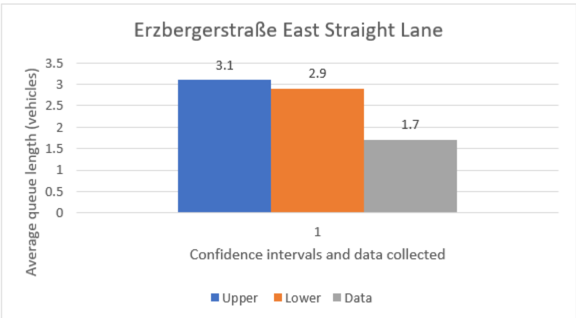
Figure 2.25: Confidence Interval for Am Krökentor South



(a) Erzbergerstraße East Right

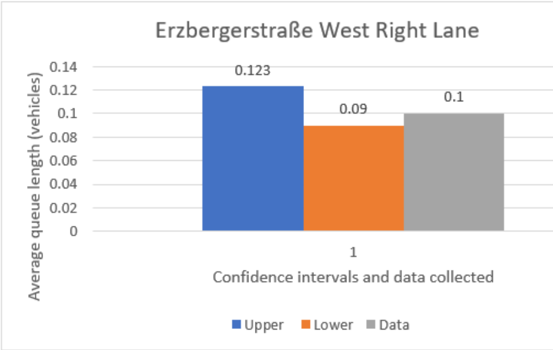


(b) Erzbergerstraße East Left

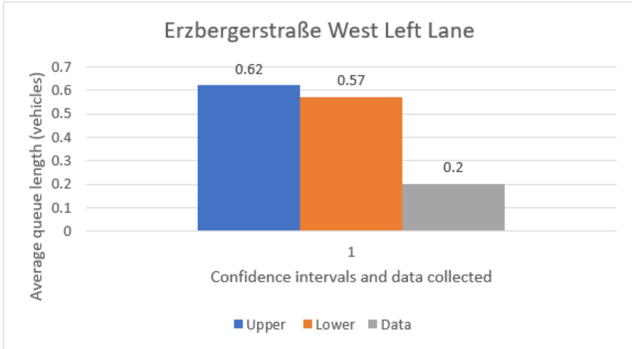


(c) Erzbergerstraße East Straight

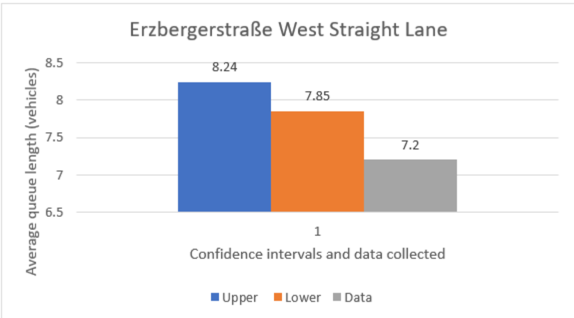
Figure 2.26: Confidence intervals for Erzbergerstraße East



(a) Erzbergerstraße West Right



(b) Erzbergerstraße West Left



(c) Erzbergerstraße West Straight

Figure 2.27: Confidence intervals for Erzbergerstraße West

2.6.5 Justification

From what can be observed in the graphs as well as from the table itself only 3 of our 9 lanes fall within their respective confidence intervals. However, we can also see that the collected data is only slightly out of the intervals (at most by 1.3 cars, while most values do not even show a difference of more than one vehicle). Justifications for the slight deviation of the collected data from the confidence intervals can be attributed to:

- The fact that queue lengths and inter arrival times were not collected on the same day.
- Insufficient amount of data collected due to lack of time and manpower.
- Randomness and unexpected events due to proximity to holidays and the inadequate number of data collection efforts.
- Possible issues with the process of data collection itself due to manual counting and not using any specific tool for measurement (speedometer, sensor etc.).

Special case :
A special case in validation was Am Krökentor North Right Lane. From table 2.7 one can observe that for this particular lane we are not within the confidence interval. However aside from the general justifications which were mentioned, the deviation of this lane from the confidence intervals can be additionally explained.

Figure 2.28: Am Krökentor validation issues

As shown in the image 2.28 we can see that this lane has an adjoining lane which functions as a parking space. Taking this into account, the only way we could have a clear estimate for the queue length for the right lane and combined (straight and left) was up until a line where a car had been parked in the parking lane. This is due to the fact that any car parked in the lane, that is near to the node itself, will cause a convergence of the two separate lanes into one. The problem is compounded by the shifting position of this car (closest to the node) from one data collection run to the other and even within the data collection run itself. We believe this may be an additional reason for the deviation in our data.

2.6.6 Statement of confidence

Given our results and having taken into account the justifications for any slight deviations from the confidence intervals, we can say with a certainty of 90% that we have a valid simulation model which can be used for future experiments.

2.6.7 Difficulties encountered

Some of the difficulties we encountered during the validation process was to figure out how to extract the data from our model. It proved quite time consuming and we had to make certain changes to our model. This was due to the fact that when the model was being build no consideration was given as to how we would retrieve certain data from it. Additionally we truly understood that one should be more thoughtful and diligent when it comes to data collection. It is apparent to us now that sufficient amount of data is vital for successfully validating a model.

2.7 Simulation Experiments

The Experiments were performed by improving the vital parts of the validated model, which were problematic. For the better understanding of the present situation, it is essential to understand the “problem” as defined above. We will also be going through the claims of the system and some of the ideas we implemented in detail.

2.7.1 Claim of the system

“We can assure the Planning Committee of the city of Magdeburg that given the circumstances and the data provided to us, if our recommendations are followed there will most probably be a non-negative impact more often than not” Node assigned: Erzberger / Am Krökentor
Confidence :90 %
P.S. This Remains a constant for all the graphs and visualizations from here on.

2.7.2 Reason leading to optimization / Problem

Damaschkeplatz is closed for construction of a new quad tramway and the project is expected to take a few more years. The only logical route into the city towards Altstadt would be to take the longer route via the Magdeburger ring towards Universitat Platz. We observed the situation and have amounted the extended wait times and long queue lengths to this aspect. The Fig 2.29 below shows the exact place where the closed section is.

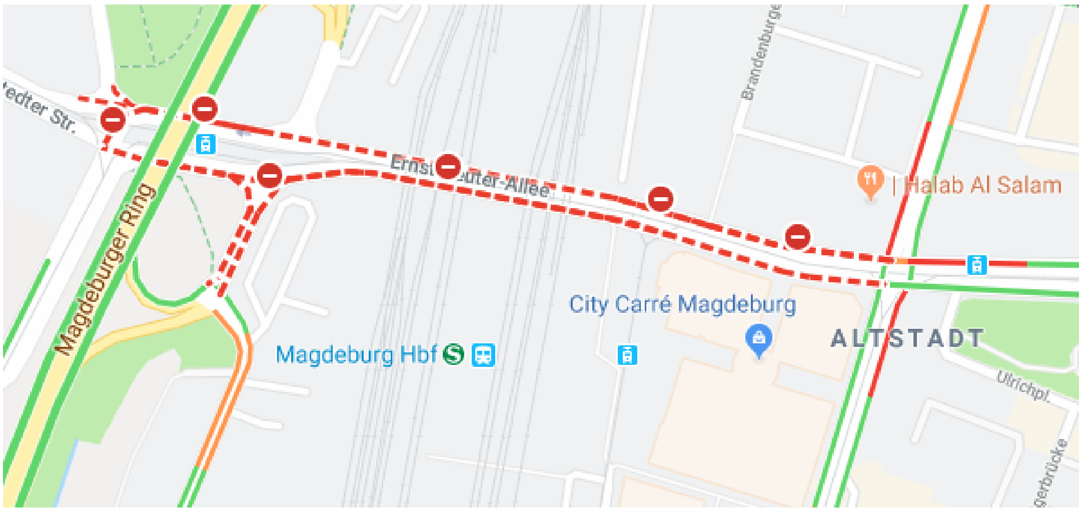


Figure 2.29: Damaschkeplatz Closed area

2.7.3 Experiment Goal

All the Experiments performed henceforth will have two main performance decision parameters, Throughput and the Average Time in the entire node, where in the former should be maximized and the later should be minimized ideally. Experiment

Environment: Each experiment will be performed on a fixed set of traffic loads.
100 – 200 percent in increments of 20.
*100 percent is the current traffic situation as validated in the previous chapter, which will be referred to as “Base Model”.

2.7.4 Experiment's Assumption

Considering the fact that we have performed 6 experiments and each experiment has 6 scenarios with respect to traffic loads, it is essential to optimize the runtime and eliminate the garbage runs as we call them. We have considered the following to tackle the same, A garbage experiment is a simulation experiment which produces worse results as compared to the Base Model, which will be defined by the performance decision parameters discussed above.

2.7.5 Output Parameters

This section further describes how the performance decision parameters discussed above were actually collected and why?
Average Time in the Node was collected as the mean time of every car in the system. This mean is calculated at the end of every run (1 hour) and will be amounted from the individual car agents logging their entry and exit. Each car broadcasts its presence as soon as it is created in the “carSource” block and similarly when it is destroyed at “carDestroy” as displayed in the Fig.2.30 below.

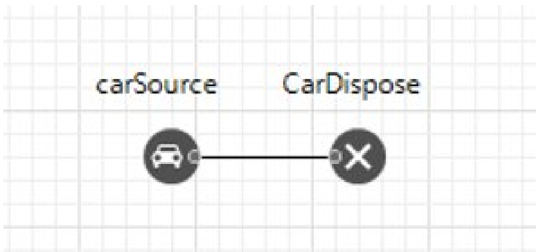


Figure 2.30: Anylogic Screencap

Throughput, the second decision parameter was collected road wise, as soon as a car passed the intersection line at the respective road. For example, as described in the Fig 2.31 car each road will have a variable which will count the number of cars.

Queue Length, an additional decision parameter was collected road wise as well like depicted in the previous Figure 2.31 The longest queue length at a particular phase change will be taken into account as the waiting time is always dependent on the slowest queue. For instance, if there is a queue of 2, 3 and 3 per lane on any road, refer the Fig2.32, the number that would be considered for evaluation would be 3.

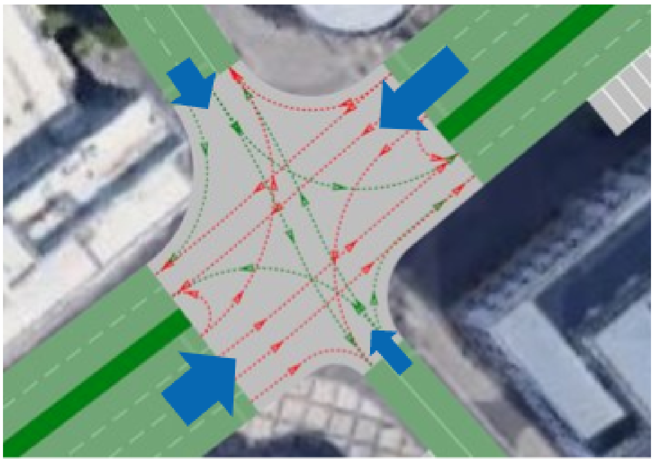


Figure 2.31: Anylogic Screencap for throughput

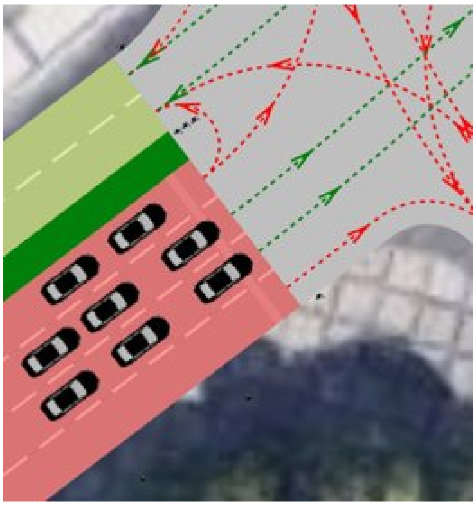


Figure 2.32: Anylogic Screencap of a Road

Upon keen selection these parameters were considered as a good evaluation criterion for any experiment. Queue lengths can be used to compare and analyse the waiting times as well. The prognosis provided to us at the beginning of the project by the Planning Committee is also keenly observed to analyse the traffic load change over time and the recommendations are made.

2.7.6 Experiments Planned / Performed

Initially the experiments were decided to be performed in 4 major phases which are described as follows:

- Adding Lanes
- Restricting turns
- Optimizing Traffic Lights
- Implementing a roundabout / Pedestrian Bridges.
- Mix and Match

2.7.7 Adding Lanes

Adding a lane in Erzberger West without changing Current traffic light phases

This experiment was designed in keeping in mind that there is a lot of congestion inside the intersection at the place where cars from Krökentor North are trying to turn right towards City Carré, which is most of the traffic from Krökentor North as discussed in Problem Analysis. So, there is a possibility of adding a new lane on Erzberger West Backward, for the right turners from Krökentor North. Please Note that this experiment is done without changing the current traffic light phases. Below graphs visualize the outputs compared to the base model.

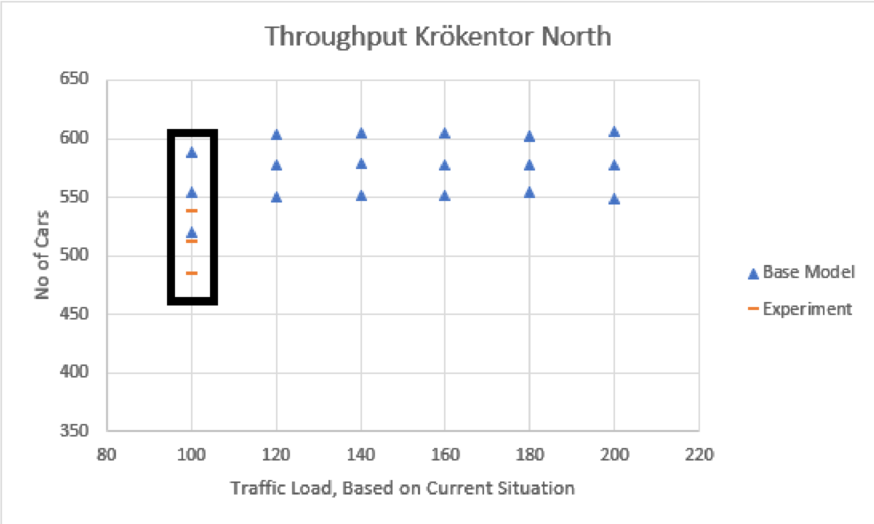


Figure 2.33: Throughput Krökentor North

Figure 2.34: Waiting Time

Adding a Lane in Erzberger West with a Free Right

The same experiment as described earlier, was performed by implementing a custom traffic light phase to allow a free, unrestricted Right Turn. Fig.2.35 describes the altered light situation in AnyLogic.

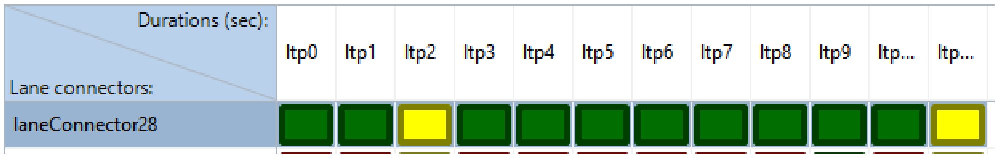


Figure 2.35: AnyLogic ScreenCap of Free Right

Below graphs visualize the outputs compared to the base model.

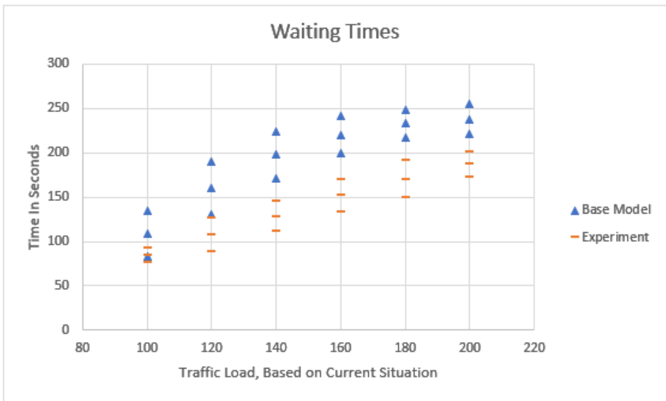


Figure 2.36: Waiting Times

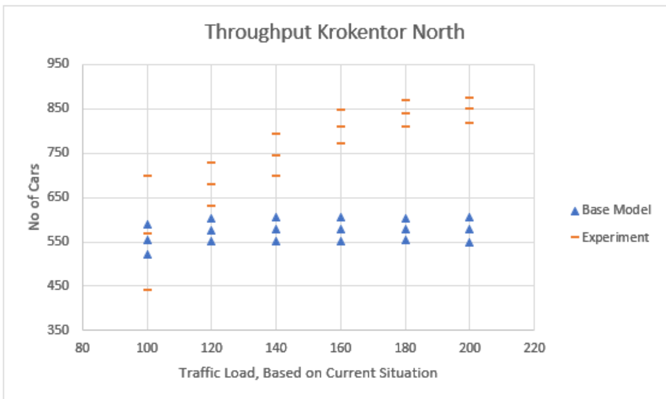


Figure 2.37: Throughputs on Krokentor North

Adding a lane in Krökentor North

We have established the fact that the traffic on Krökentor has been more than the lane can handle. To make matters worse there is a parking on the lane which amounts to more congestion. So, this experiment deals with the removal of the said parking strip and adding an extra lane so as to reduce the load on a single lane.

The same can be visualized in the figure 2.38, please note the new lane is marked with a black-solid line.

Figure 2.38: The Lane added in Black

Below graphs visualize the outputs compared to the base model.

Figure 2.39: Waiting Times

The aim showed promising results when it came to reducing the queue lengths in Krökentor North where we have seen an improved the queue lengths by more than 10 percent. As the Traffic load on Krökentor North increased the road maintained a steady hold of under 20 cars at an average. The same can be observed in the Fig.2.40.

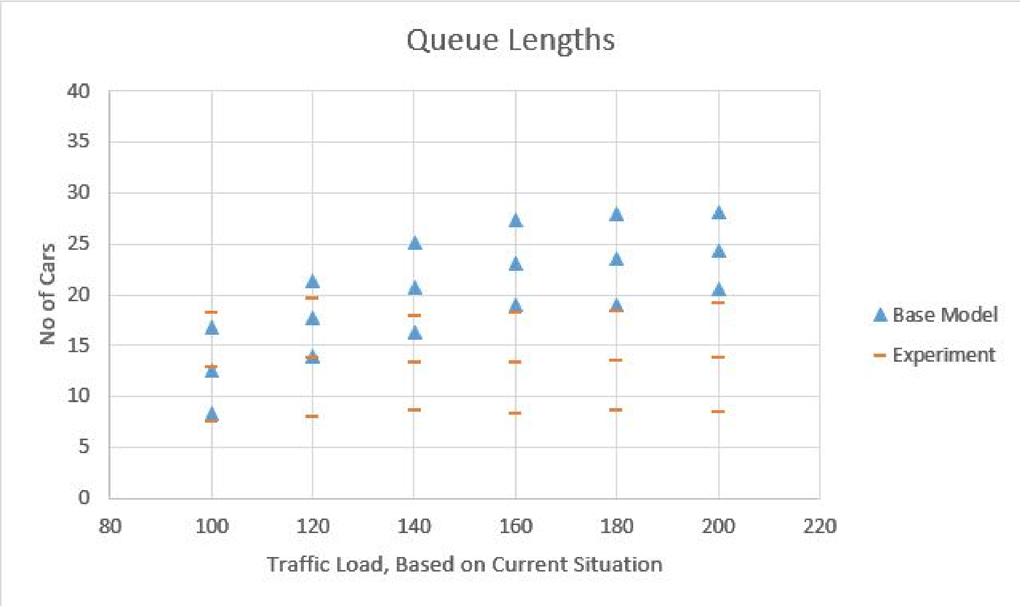


Figure 2.40: Queue Length on Krökentor North

2.7.8 Restricting Turns

This experiment is designed to improve the situation at the centre of the intersection. There are a lot of congestion in the centre as the opposing turns are green at the same time as shown in the Fig.2.41 below.

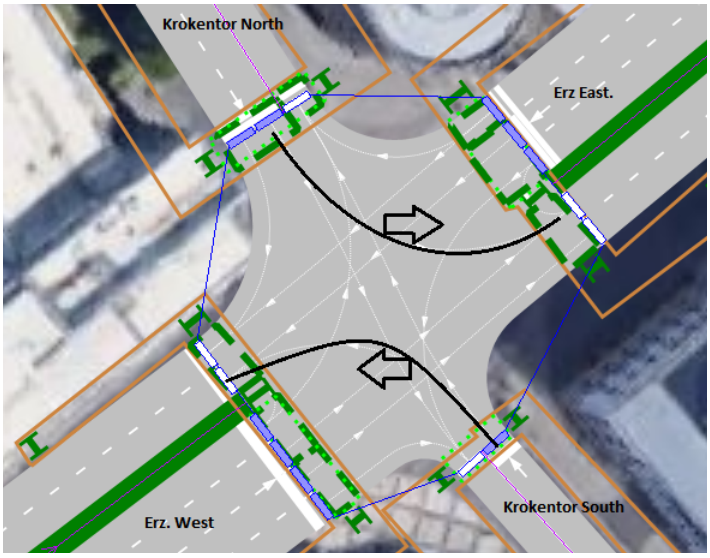


Figure 2.41: Restricting Turns Example

This experiment was discarded before implementation completely because there were a lot of aberration on the node as soon as we unplugged one of the turning directions as shown in the Fig. 2.42. Upon further edits the output decision parameters further deteriorated the system.

Figure 2.42: Restricted Left Turn From Krökentor North

2.7.9 Optimizing Traffic Lights

This experiment focuses on the fact that the traffic lights in the intersection can be further optimized. The phases are phased for 90 sec with varying times for each road. All the yellow lights are for a period of 4 sec and, weren't tampered with. This optimization experiment works on the objective – average time in the node. It focuses on minimizing the same by co-varying the traffic light phases as discreet values. The best solution (solution vector of traffic light times, with the least waiting time) of 100 runs with random seed, is selected as the most optimal solution. The optimal traffic light numbers have been described in the Table2.8.

itp0	10
itp1	06
itp2	04
itp3	02
itp4	07
itp5	01
itp6	02
itp7	13
itp8	02
itp9	14
itp10	04
itp11	04

Table 2.8: Optimal TL Phases (Minor)

The waiting times as compared to the base model with unaltered traffic lights have been visualized below.

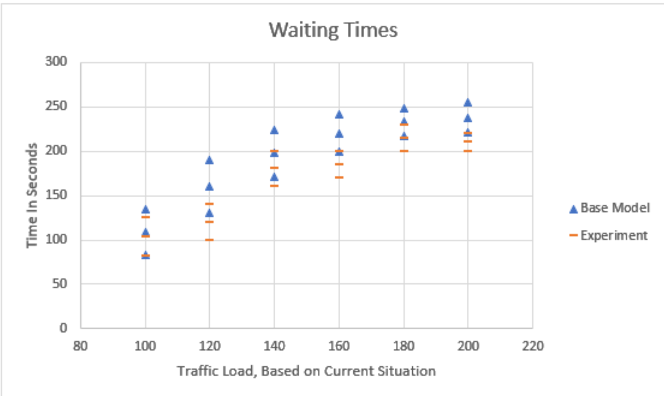


Figure 2.43: Waiting Times

2.7.10 Implementing a Roundabout

A roundabout would generally be a very good consideration for an intersection bugged with waiting time issues. We have observed the conditions at the intersection and the implementation meant increasing wait-times and decreasing through puts. Hence the idea was discarded and was not worked upon, the reasons for which have been described below.

- Roundabouts are generally less effective when there are minor and major roads, especially not if the difference between the said is too much. In our case the roundabout will negatively impact the node as we have 1,2,3,4 lanes on Krökentor South, Krökentor North, Erzberger East and Erzberger West respectively.
- The throughput will be generally less because the traffic throughput in the said roads is very different. The result would be a scenario where the roundabout would be full all the time, negatively impacting the current situation

2.7.11 Implementing a Pedestrian Bridge on all the roads

This experiment deals with the addition of an alternate pedestrian path for people trying to cross the intersection, like a Pedestrian Bridge. This experiment has been agreed upon due to the reason that there is a high volume of people trying to cross the intersection at any given point of time. We have also observed that the reason for such a high load, is due to the fact that Krökentor North is highly approachable for students at Otto Von Guericke University, from the tram stop Opera House. This stop also has a high inflow of trams as Tram-Line 1,2,5,9,10 all pass through the said stop. The results of the experiment were promising and have been described below via the graphs below, visualizing the outputs compared to the base model.

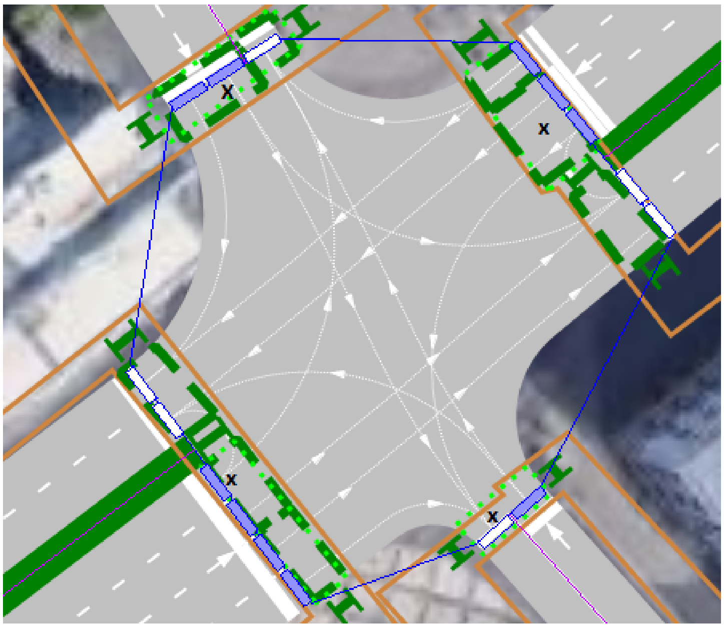


Figure 2.44: x defines Pedestrian = Null

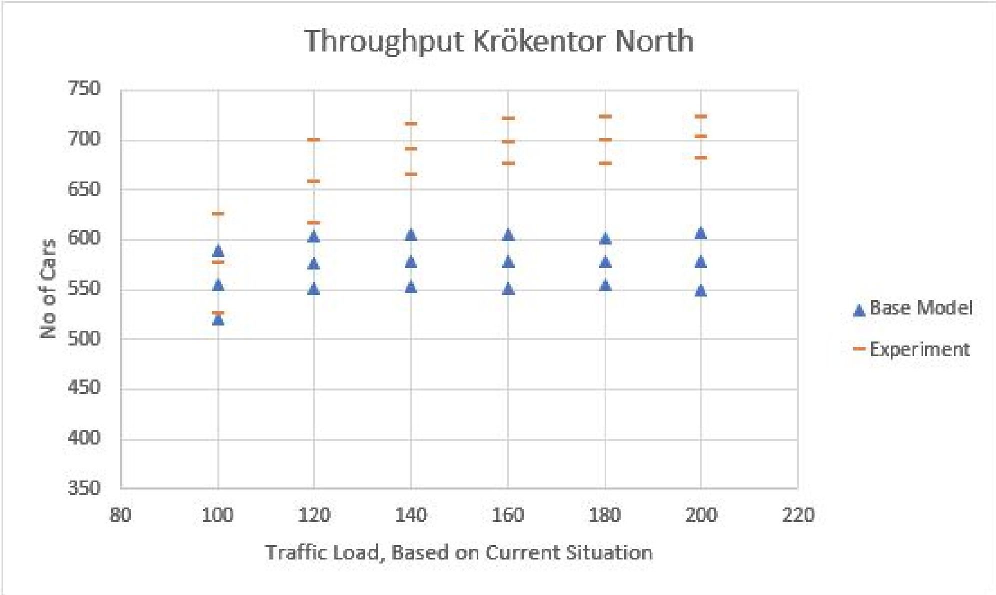


Figure 2.45: Throughput Krökentor North

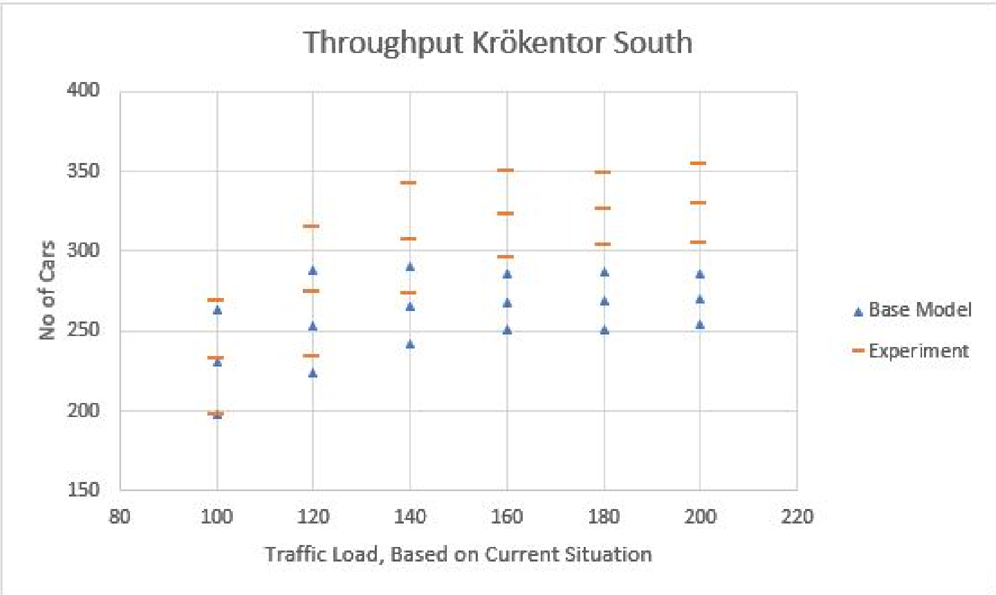


Figure 2.46: Throughput Krökentor South

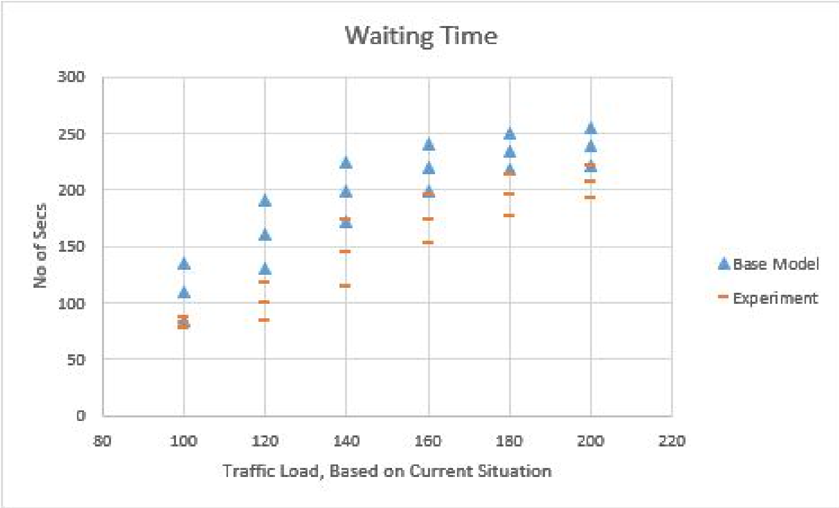


Figure 2.47: Waiting Time visualized

2.7.12 Mix and Match Scenario

This experiment is formulated as a hybrid experiment where in the best improvements from the experiments discussed earlier have been combined resulting in a new scenario. The main aspect of this experiment is adding a lane in Krökentor North for efficient lane handling and a lane on Erzberger West for a free right turn with a modified traffic light phase to allow for a free right turn from Krökentor North. We will also be doing the same with the optimized traffic lights from Experiment 3 and will visualize the same as well. The Fig.2.48 help visualize the same.

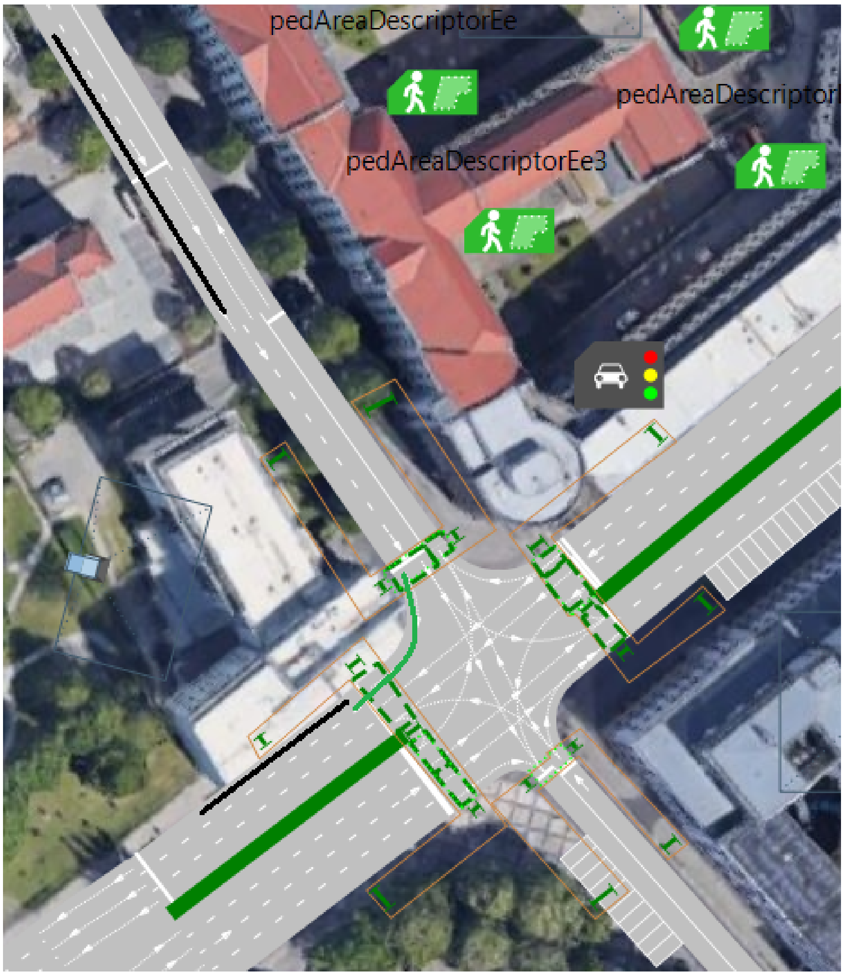


Figure 2.48: Waiting Time visualized

Below graphs visualize the throughput compared to the base model per road.

Figure 2.49: Waiting Time

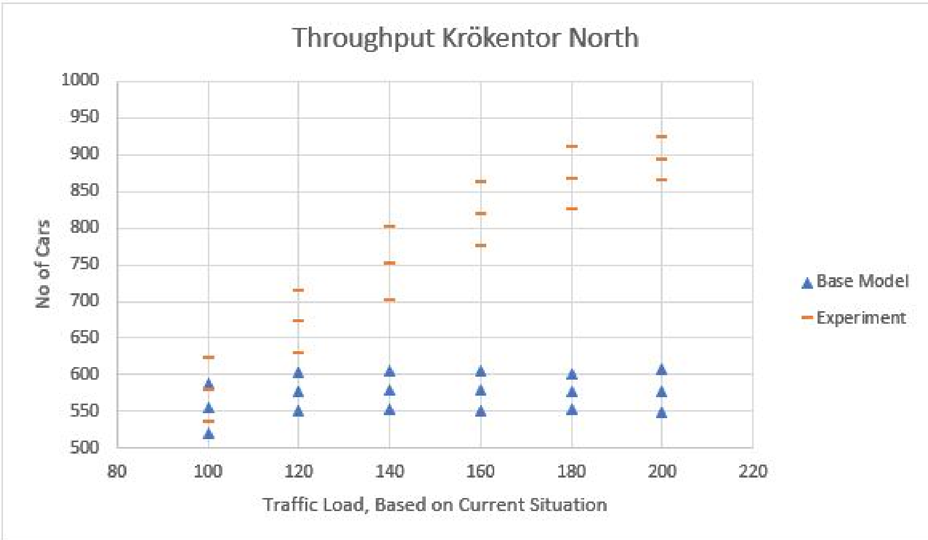


Figure 2.50: Waiting Time

2.7.13 Recommendations

From the aforementioned experiments, all the outputs were meticulously studied to make these following recommendations. The table below describes all of them accurately. The Recommendation is gauged on the respective parameter improvement over the base model.

Suggestion	Avg. Time	Queue Length	Throughput
Add Lanes on Krökentor North	-11.00%	-1.00%	-1.00%
Add Lanes on Krökentor North, Erzberger West, Phase Change	-32.50%	-11.00%	+8.08%
Optimized TL on Recommendation 2	-31.50%	-7.81%	+7.70%

Table 2.9: Recommendations

As seen in table 2.9 the 3 recommendations are:

- Adding Lanes on Krökentor North, which decreased the queue lengths and average waiting time.
- Adding lane on Krökentor North Erzberger West, along with modification to the phases, which decreased waiting time and queue length but also increased the throughput.
- Adding lane on Krökentor North Erzberger West, along with optimization of the traffic lights duration, which resulted in shorter waiting times, smaller queues and greater throughput

2.8 Expenditure and Progress

2.8.1 Budget

In executing the project we are well within the set budgetary constraints. As seen on 2.51 we are have kept our realised budget bellow our planed budget. One notable exception is Milestone 4 Data analysis, where we went slightly over the planned budget. This was due to having some issues planing and successfully executing data collection runs. One of the difficulties encountered was the proximity to the holidays which ended up costing us manpower and hours on an unsuccessful data collection run. Additional challenges for this milestone included bad weather conditions which prolonged and made the data collection process more difficult. This slight deviation from the planed cost can easily be rectified by savings made in other milestones.

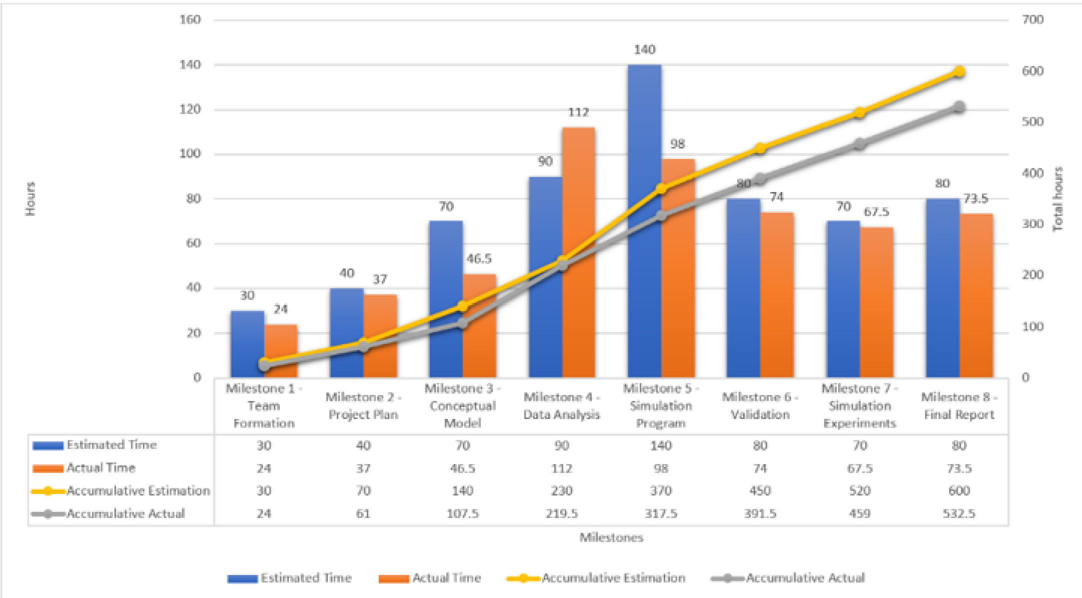


Figure 2.51: Project Expenditure

2.8.2 Progress

Regarding the overall progress, we have achieved our goal of a timely completion of the project. As seen in graph 2.52, we have been successfully in realising our planned actions. Some slight deviations can be noticed around Milestone 4 Data Analysis (weeks 13.05, 20.05, 27.05), where we had to make certain changes to the statistical distributions of our inter arrival times and recheck their validity.

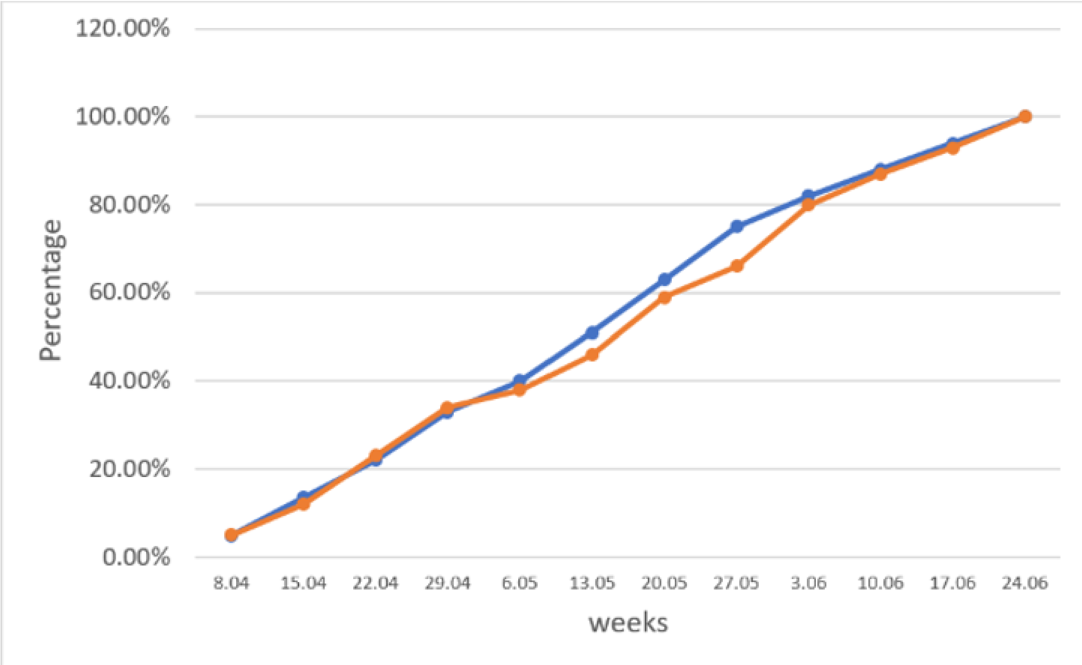


Figure 2.52: Project Progress

In conclusion we are very pleased to have respected our budgetary restrictions as well as our deadlines. Certain slight deviations from what we had planned can be noticed, however we were able to overcome these difficulties and finish the project within the given constraints.

Chapter 3

Conclusion

This project proved to be a exciting challenge for everyone involved in its realisation. Through our hard work and determination we were able to overcome the obstacles in our path and deliver a project which we are proud off. Starting off with forming a team of strangers all the way to more advanced tasks such as project planing, data collection, model building, validation and experiments, each of us gained so much knowledge and experience along the way. We have not only expanded our knowledge regarding simulation, but we have also gained valuable experience as to what it means to work in a team. We believe the complex and difficult nature of the task helped us forge skills needed for our future. Additionally, our team hopes that with our project we can better the city of Magdeburg's transportation network as well as provide insight to possible future improvements.

Appendix A

Petri Nets

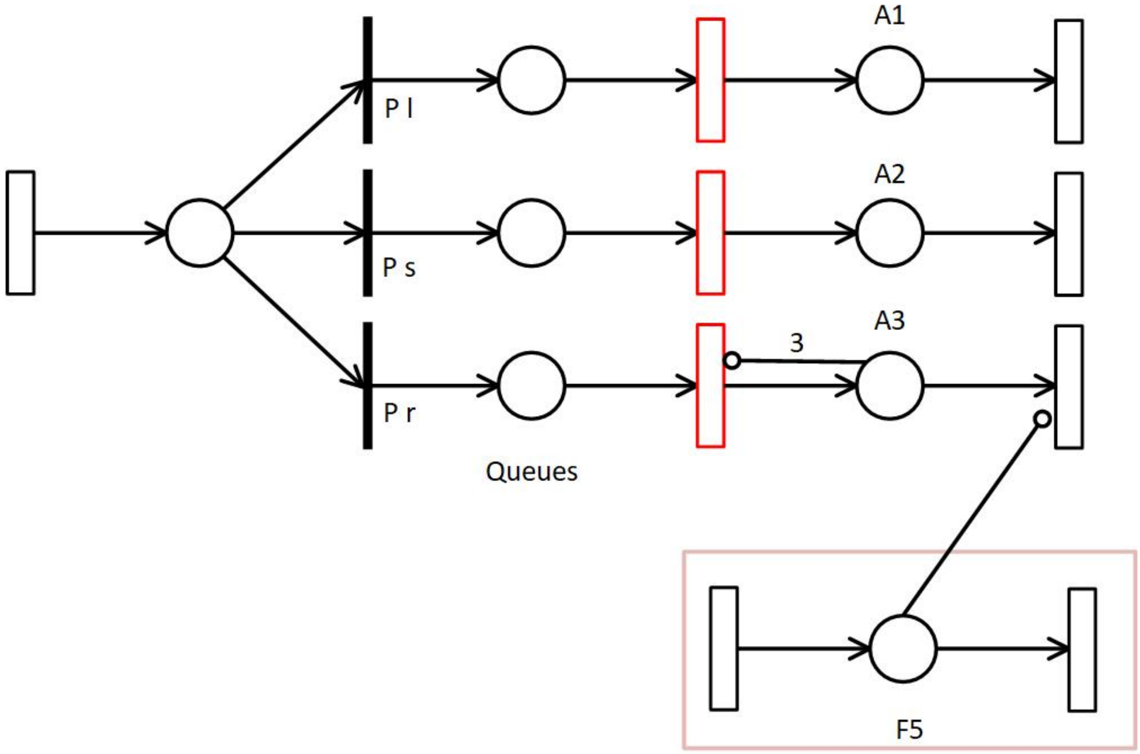


Figure A.1: Petri Net for Erzberger Straße West.

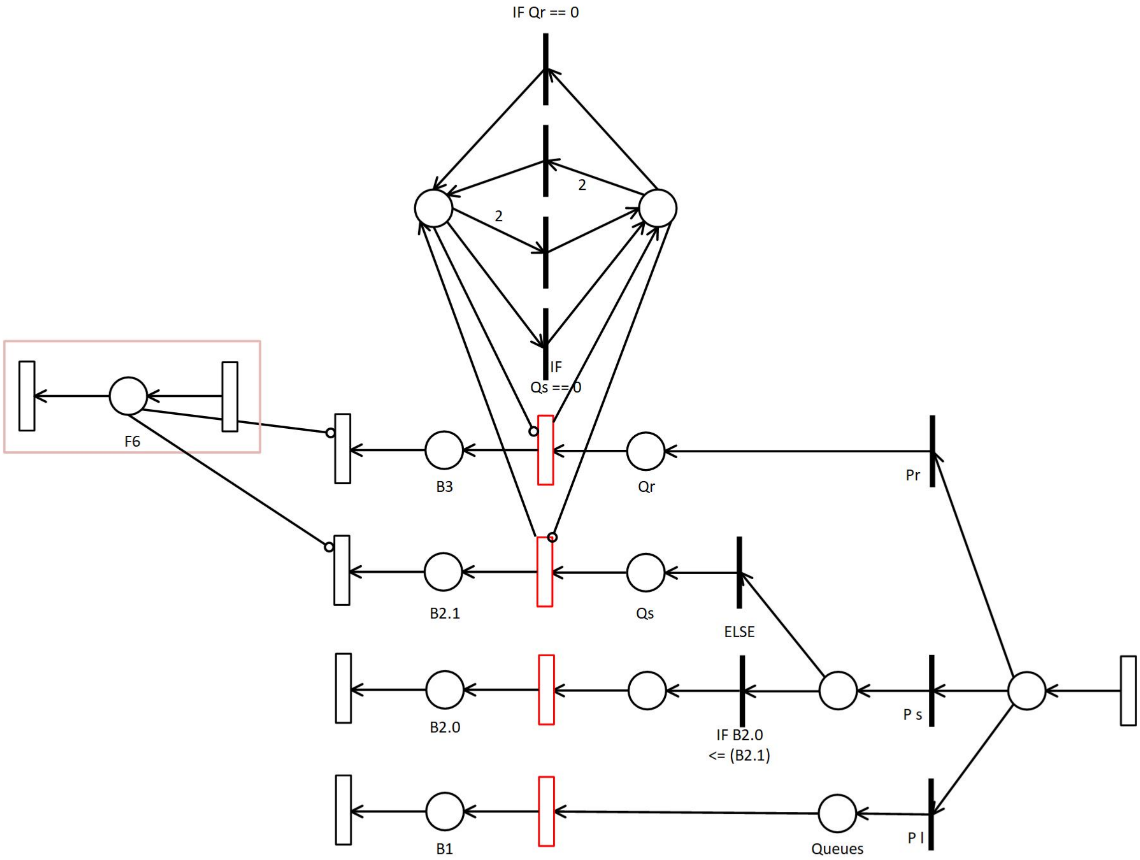


Figure A.2: Petri Net for Erzberger Straße East.

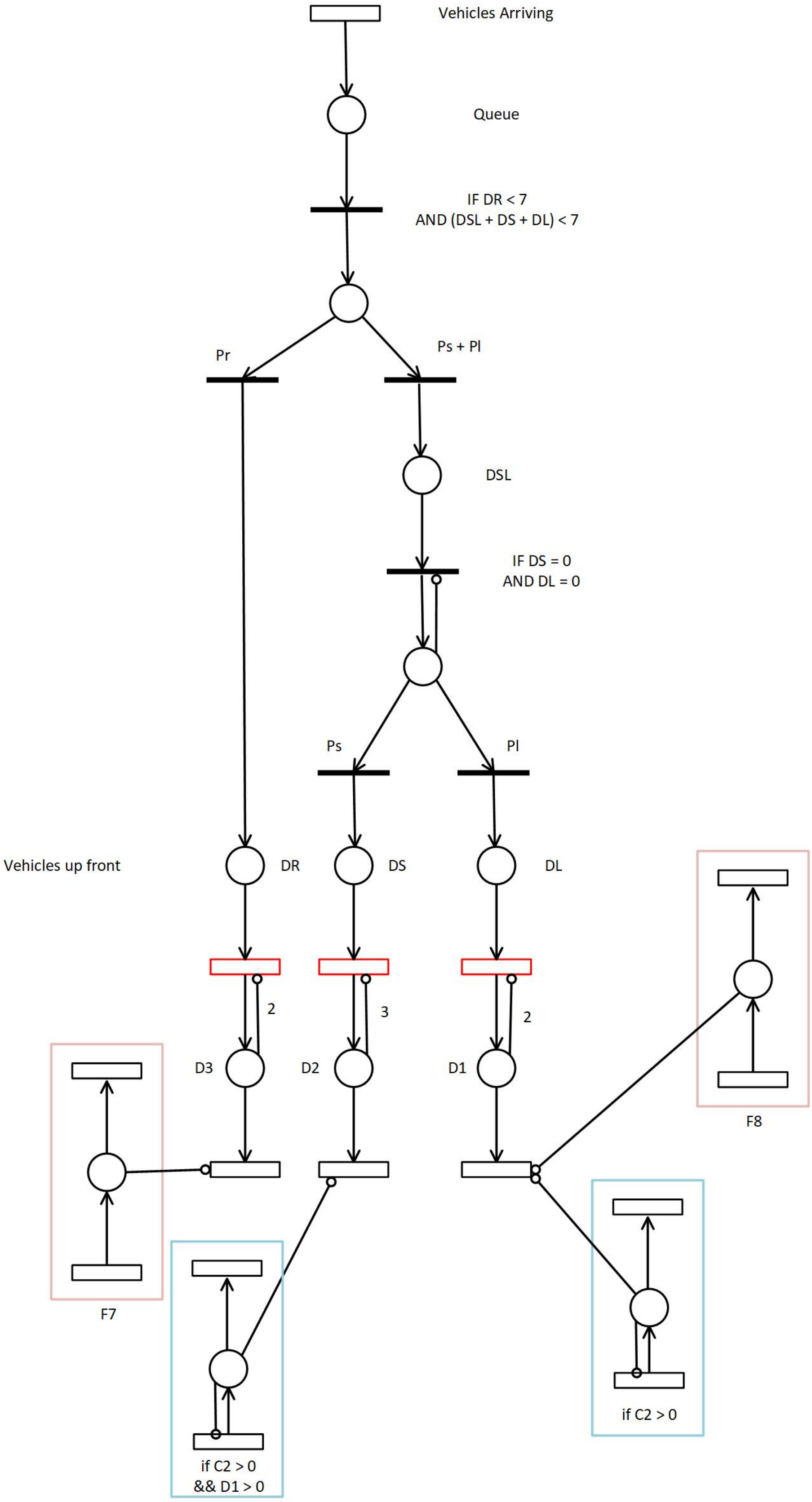


Figure A.3: Petri Net for Am Krökentor North.

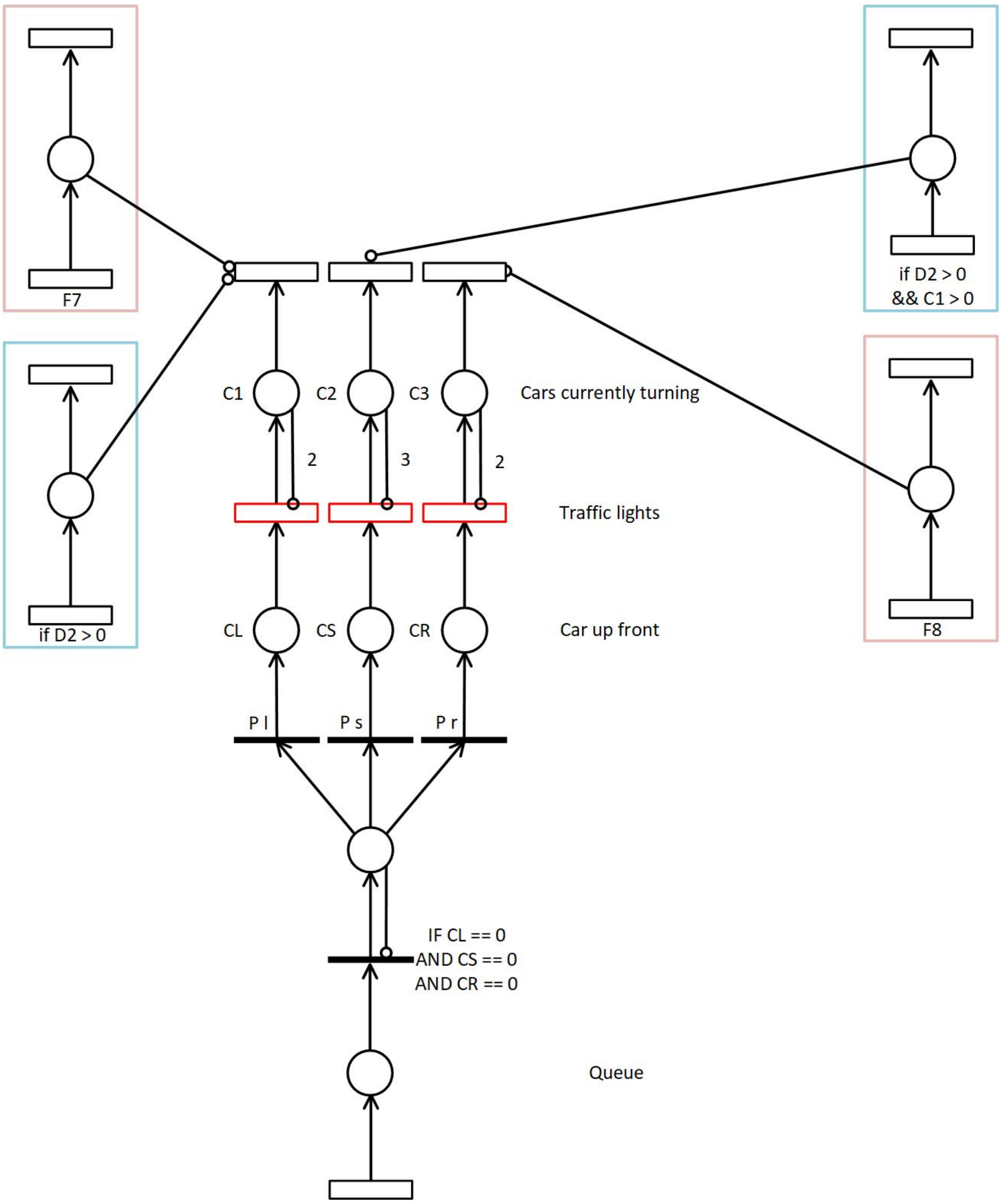


Figure A.4: Petri Net for Am Krökentor South.

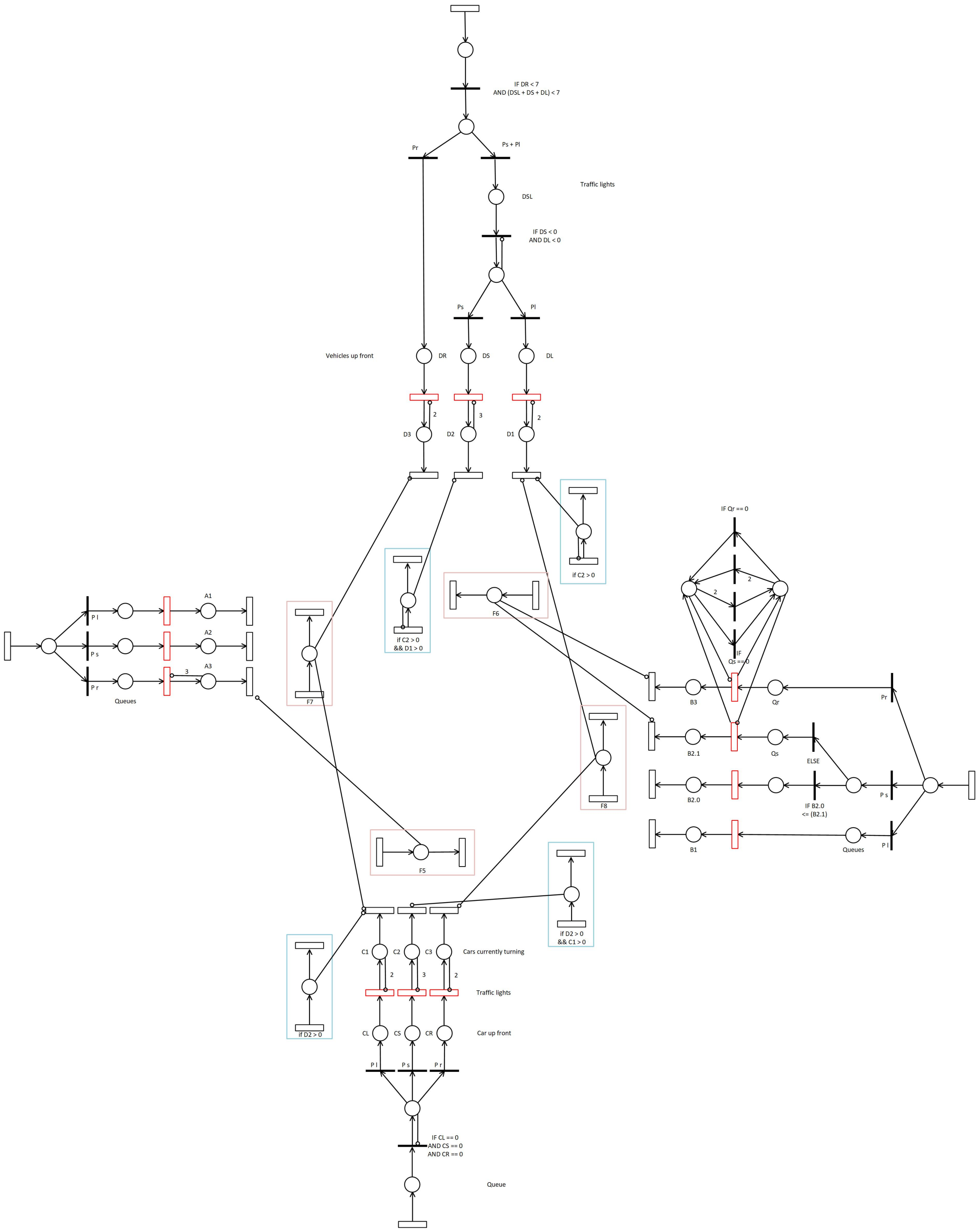


Figure A.5: Petri Net for the Whole Intersection.

Appendix A

Data Analysis

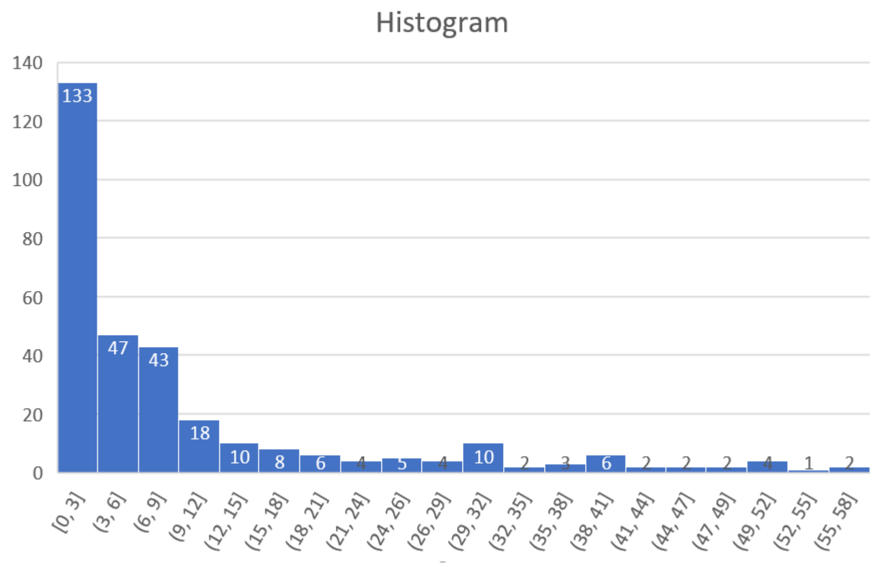


Figure A.1: Histogram of inter-arrival times of vehicles at Erzberger (NO)

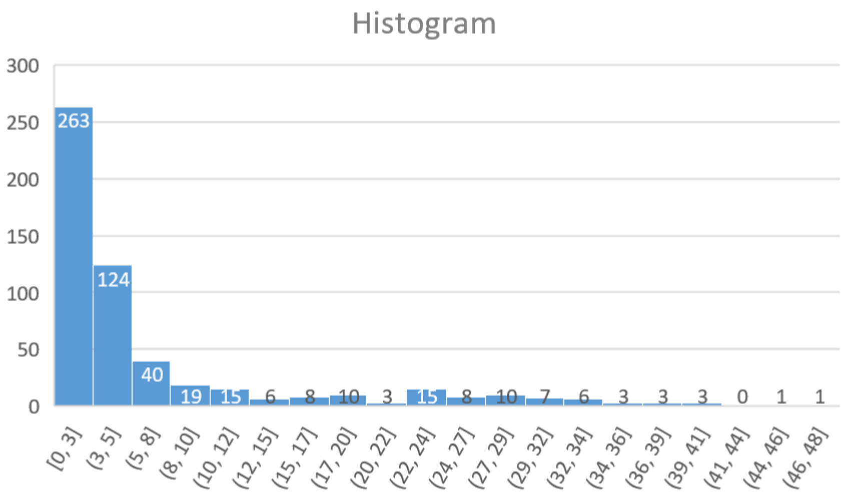


Figure A.2: Histogram of inter-arrival times of vehicles at Erzberger (SW) lane 1

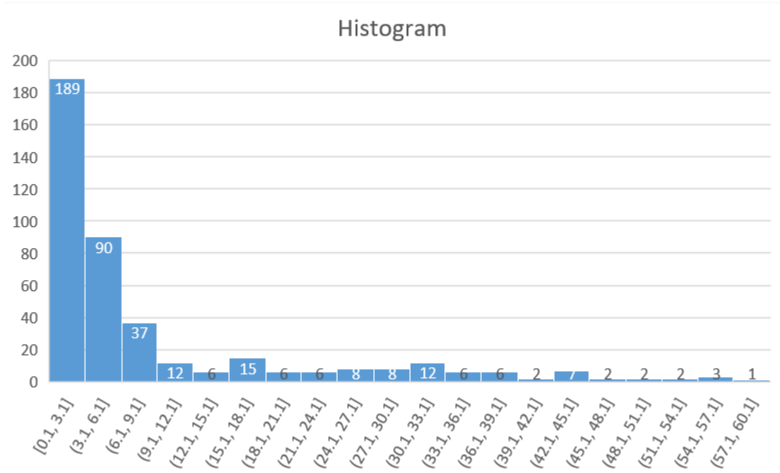


Figure A.3: Histogram of inter-arrival times of vehicles at Erzberger (SW) lane 2

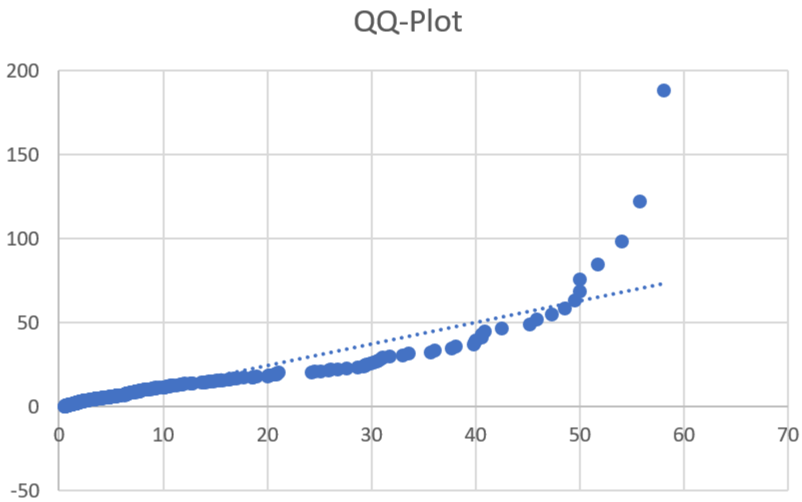


Figure A.4: Quantile-Quantile Plot for distribution at Erzberger (NO)

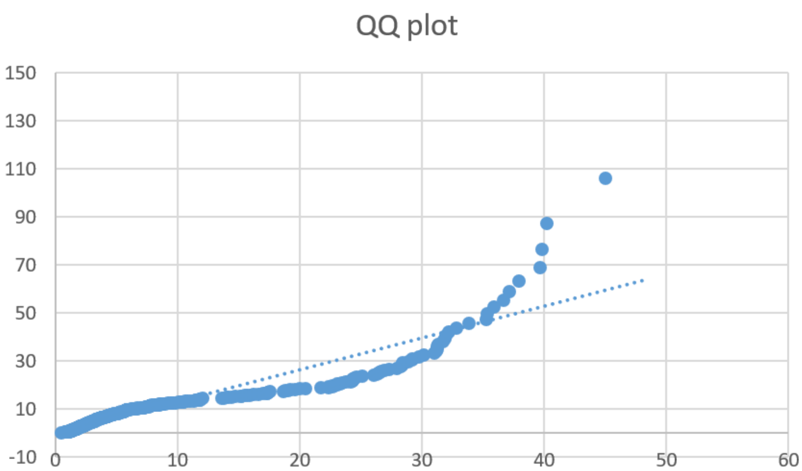


Figure A.5: Quantile-Quantile Plot for distribution at Erzberger (SW) lane 1

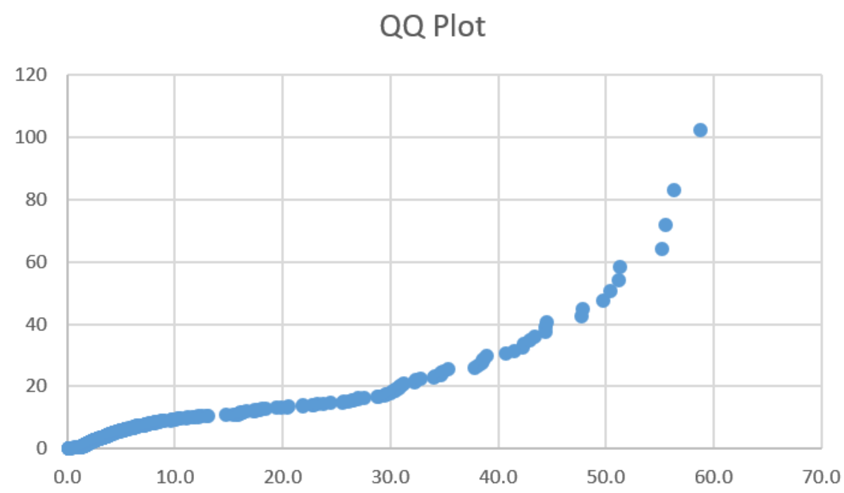


Figure A.6: Quantile-Quantile Plot for distribution at Erzberger (SW) lane 2

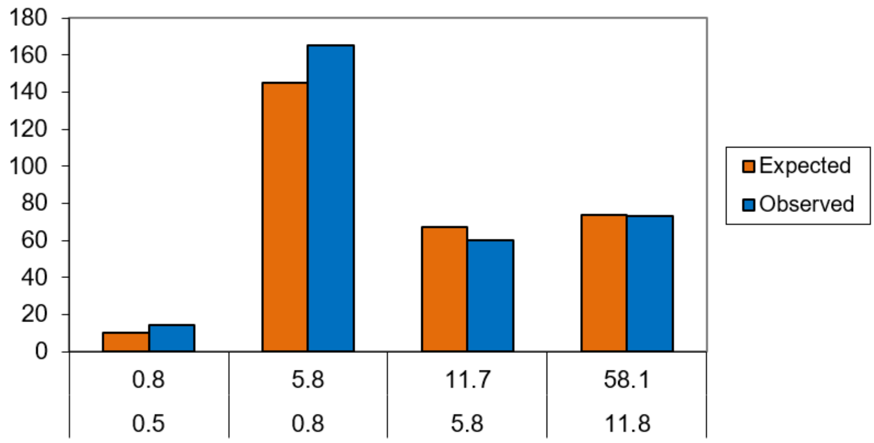


Figure A.7: Chisquare test - Observed vs Expected values, Erzberger (NO)

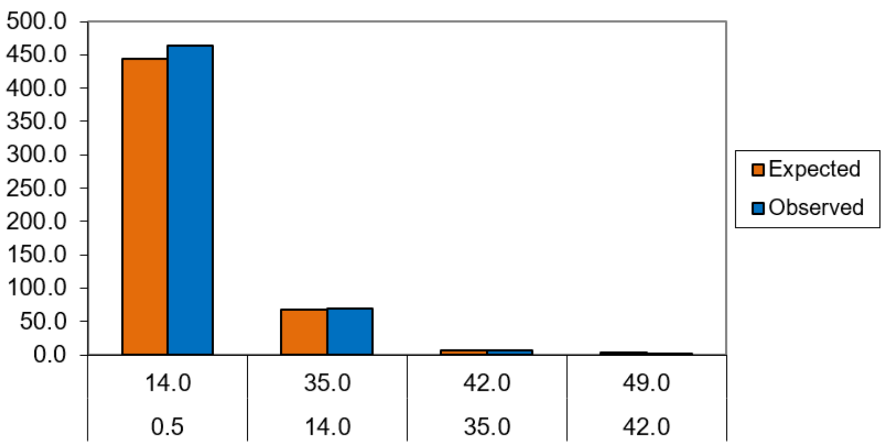


Figure A.8: Chisquare test - Observed vs Expected values, Erzberger (SW) lane 1

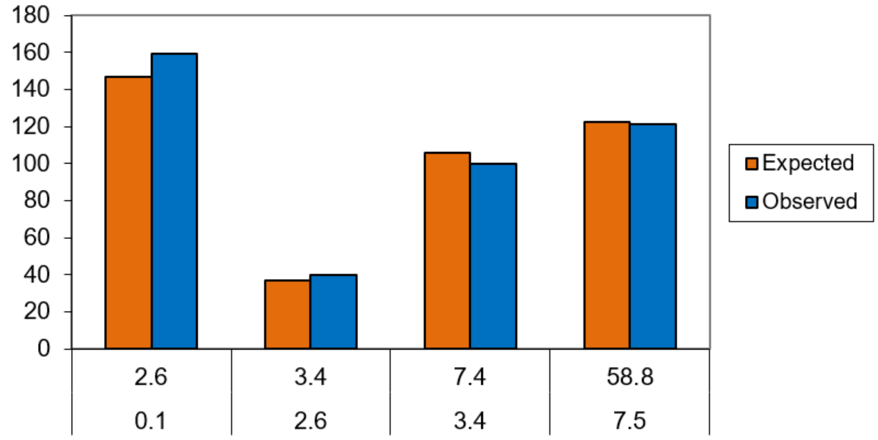


Figure A.9: Chisquare test - Observed vs Expected values, Erzberger (SW) lane 2