

# Measuring turbulence distance of acceleration lanes

Marco Hovenga

Delft University of Technology





# Measuring turbulence distance of acceleration lanes

Comparing acceleration lanes and basic freeway segments to measure turbulence

by

**Marco Hovenga**

in partial fulfillment of the requirements for the degree of

**Master of Science**  
in Civil Engineering

at the Delft University of Technology,

Graduation committee:

Professor T&P:	Prof. dr. ir. S. P. Hoogendoorn,	TU Delft
Daily supervisor:	Dr. V. L. Knoop,	TU Delft
Daily supervisor:	Ir. A. S. van Beinum,	Witteveen+Bos
Guest member TBM:	Dr. J. A. Annema,	TU Delft
Thesis coordinator:	Ir. P. B. L. Wiggeraad,	TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.





# Contents

<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>Preface</b>	<b>ix</b>
<b>Summary</b>	<b>xi</b>
<b>Samenvatting</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Definition traffic flow turbulence . . . . .	3
1.3 Research objective . . . . .	4
1.4 Report outline . . . . .	4
<b>2 Theoretical background</b>	<b>7</b>
2.1 Causes of turbulence . . . . .	7
2.2 Driving maneuvers . . . . .	8
2.3 Traffic flow characteristics . . . . .	10
2.4 Conclusions on theoretical background . . . . .	13
<b>3 Methodology</b>	<b>15</b>
3.1 Measuring turbulence distance . . . . .	15
3.2 Case-control study . . . . .	15
3.3 Data collection method . . . . .	18
3.4 Suitable analyses . . . . .	20
3.5 Conclusions on methodology . . . . .	23
<b>4 Data and data processing</b>	<b>25</b>
4.1 Dante . . . . .	25
4.2 Data processing method . . . . .	25
4.3 Location selection . . . . .	26
4.4 Induction loops selection . . . . .	27
4.5 Filtering induction loop data . . . . .	31
4.6 Selected locations and induction loops . . . . .	34
4.7 Selection of interval length . . . . .	35
4.8 Conclusions on data and data processing . . . . .	36
<b>5 Results</b>	<b>39</b>
5.1 Speeds . . . . .	39
5.2 Lane flow distribution . . . . .	45
5.3 Density . . . . .	49
5.4 Turbulence distance estimation . . . . .	52
<b>6 Conclusions and recommendations</b>	<b>57</b>
6.1 Findings . . . . .	57
6.2 Conclusions . . . . .	59
6.3 Recommendations . . . . .	60

---

<b>Appendices</b>	<b>61</b>
<b>Bibliography</b>	<b>63</b>
<b>A Fixed average speed check</b>	<b>67</b>
<b>B Performance maximum likelihood estimations</b>	<b>75</b>
<b>C Filtering results</b>	<b>103</b>
<b>D Description of the selected locations</b>	<b>111</b>
<b>E Used Matlab scripts</b>	<b>125</b>

# List of Figures

1.1	Elements of an acceleration lane . . . . .	2
1.2	Example of turbulent fluid . . . . .	3
1.3	Detailed framework thesis . . . . .	5
2.1	Schematisation of driver subtasks . . . . .	8
2.2	Driving maneuvers on basic freeway segments . . . . .	9
2.3	Driving maneuvers on acceleration lanes . . . . .	10
2.4	Effects in traffic flow characteristics . . . . .	11
2.5	Expected changes in the mean speed . . . . .	12
2.6	Expected changes in the lane flow distrubution . . . . .	12
2.7	Expected changes in the lane density . . . . .	13
3.1	Daganzo's fundamental diagram . . . . .	17
3.2	Example of distribution free headway estimation for site "Doenkade" . . . . .	17
3.3	Positions of the selected induction loops around acceleration lane. . . . .	19
3.4	Selection criteria of the positioning of the induction loops on basic freeway segments . . . . .	20
3.5	Selection criteria of the positioning of the induction loops on acceleration lanes . . . . .	20
3.6	Example of the estimation traffic flow characteristic lane flow distribution with the maximum likelihood on the median lane . . . . .	21
3.7	Example of the comparison of the density on the outside lane of the estimated mean of the basic freeway segment with the estimated distributions around the acceleration lane . . . . .	22
3.8	Example of the lane fitting on the individual induction loops on changes of the lane density on the different lanes . . . . .	24
4.1	Scheme steps data process . . . . .	26
4.2	Lane speeds comparison median lane . . . . .	27
4.3	Lane flow distributions per location outside lane 600 meter influence length criteria . . . . .	28
4.4	Lane flow distributions per location outside lane 900 meter influence length criteria . . . . .	29
4.5	Lane flow distributions per location outside lane 1200 meter influence length criteria . . . . .	30
4.6	Weather behavior April 2013 and 2014 . . . . .	32
4.7	Speeds over densities on the median lane with 130 km/h data on basic freeway segments . . . . .	33
4.8	Example of unexpected output of induction loops filtering, the colors indicate different locations . . . . .	33
4.9	Speeds over densities median lane with A10 induction loops in interval 300-600 meter . . . . .	34
4.10	Positions of the selected induction loops . . . . .	35
5.1	Carriageway mean speed comparison estimated distribution basic freeway segments and acceleration lanes . . . . .	40
5.2	Average speed standard deviation comparison estimated distribution basic freeway segments and acceleration lanes . . . . .	40
5.3	Outside lane speed comparison . . . . .	41
5.4	Outside lane speed standard deviation comparison . . . . .	42
5.5	Center lane speed comparison . . . . .	42
5.6	Center lane speed standard deviation comparison . . . . .	43
5.7	Median lane speed comparison . . . . .	43
5.8	Median lane speed standard deviation comparison . . . . .	44
5.9	Outside lane speed histogram and estimated distribution . . . . .	45
5.10	Outside lane fraction of the flow comparison . . . . .	46
5.11	Outside lane fraction of the flow standard deviation comparison . . . . .	46

5.12 Center lane fraction of the flow comparison . . . . .	47
5.13 Center lane fraction of the flow standard deviation comparison . . . . .	47
5.14 Median lane fraction of the flow comparison . . . . .	48
5.15 Median lane fraction of the flow standard deviation comparison . . . . .	48
5.16 Outside lane density comparison . . . . .	49
5.17 Outside lane density standard deviation comparison . . . . .	50
5.18 Center lane density comparison . . . . .	50
5.19 Center lane density standard deviation comparison . . . . .	51
5.20 Median lane density comparison . . . . .	51
5.21 Median lane density standard deviation comparison . . . . .	52
5.22 Fraction of the flow means of the individual induction loops on the outside lane . . . . .	53
5.23 Estimation start and end point influence acceleration lane . . . . .	54
5.24 Estimation start and end point influence acceleration lane . . . . .	55

# List of Tables

1.1	Turbulence distances in the NOA in meters . . . . .	1
3.1	Applied freeway design characteristics . . . . .	16
4.1	Criteria locations . . . . .	26
4.2	Selection of days as input . . . . .	31
4.3	List of the selected locations on basic freeway segments and acceleration lanes . . . .	34
4.4	Interval determination . . . . .	36
5.1	p-values of the carriageway speed KS-test . . . . .	39
5.2	p-values of the lane mean speed KS-test . . . . .	41
5.3	p-values lane flow distribution KS-test . . . . .	45
5.4	p-values lane density KS-test . . . . .	49



# Preface

This thesis is the final part of my Master Civil Engineering, Transport and Planning at the TU Delft. This master thesis has been carried out at Witteveen+Bos in Almere and Amsterdam, which is an major engineering company in the Netherlands. During my internship at Witteveen+Bos Aries van Beinum interested me for his promotion research on turbulence in traffic flow. This topic was a good fit together with my interest into traffic flow theory, freeway design, programming and big data and a challenge for me to look outside my normal design scope.

First I want to thank thank Aries van Beinum for his support to let me do this topic for my master thesis, his clear comments and providing my work place in Almere and Amsterdam and I hope that he can apply these results of this thesis in his research. Second I want to thank Victor Knoop for his comments and suggestion during our meetings and to do them even by Skype during the period I was sitting at home with my broken ankle. Further I want to thank Bernat Goñi Ros for his support during the period that Victor Knoop was not available and his helpful comments on the structure of my thesis. Further I want to thank the rest of my committee: my professor Serge Hoogendoorn, graduation coordinator Paul Wiggeraad and my external committee member Jan Anne Annema for their comments during the committee meetings.

Further I want to thank Frank Zuurbier and Chris van Hinsbergen for providing the program Dante, without this program the used analyses in this thesis would not have been possible.

Also I like to thank my colleagues at Witteveen+Bos for their support, answering my questions and to discussing many times several topics, especially football.

At last,I want to thank my parents, brother and sister in keeping confidence in my work, in getting the master degree and their support in my period at home with my broken ankle.

*Marco Hovenga  
Amsterdam, December 2014*





# Summary

Turbulence distances are a major design criteria in the design of new freeways and modifications of freeways. The NOA ("*Nieuwe Ontwerprichtlijnen Autosnelwegen*") is the decisive design guideline in the Netherlands to design freeways. Turbulence distances are part of the guideline to determine the minimal distance between on-ramps and off-ramps to enter and to exit the freeway, or to switch from one freeway or the other, and lane drops. In the latest update of the NOA, turbulence distances are identified as one of the topics which need more background. Turbulence distances are already in 1975 part of the freeway design guideline in the Netherlands and the values are not changed in the meanwhile, while the driver and vehicle composition are changed, which can influence the turbulence distance. The current turbulence distances are therefore not representative anymore of reality. Furthermore, the pressure on the traffic system is increasing; budgets are limiting, public acceptance is decreasing and performance and legal requirements are increasing. Rijkswaterstaat (freeway manager in the Netherlands) wants to limit the size of the turbulence distances to reduce costs of new freeway and modifications of freeways. Therefore, the turbulence distances need to be determined again to see if they are still up to date, to provide more background and to see if they can be reduced. In specific, in this thesis the turbulence distances of the acceleration lane are measured in traffic flow.

The first question that is what turbulence in the traffic flow is, since the meaning of turbulence in fluids and gases are well-known, but not turbulence in traffic flows. Traffic flows are sometimes approached as a fluid, and so the lane changes and speed changes are seen as turbulent movements within the traffic flow. However, the design guideline in the Netherlands describes turbulence as effect of the presence of a discontinuity (e.g. acceleration lanes) and the design guideline in the United States (HCM) describes that turbulence is present in normal traffic flow behavior, but that turbulence is increasing around discontinuities. In this thesis turbulence is defined as a quantity to define the amount of lane changes and speed changes in the traffic flow, which is close to the description of turbulence in the HCM and fluid mechanics.

In the HCM two situations can be distinguished, a situation where the turbulence is normal, and a situation where the traffic is subject to the turbulence in traffic flow. The two situations are identified as basic freeway segments (normal turbulence) and acceleration lanes (changed turbulence). In a literature review differences are found between these two situations in the decisions around the acceleration lane. A difference is identified in the set of decisions around the acceleration lane when the driving task by Minderhoud is applied on the traffic. The availability of the strategic route navigation around acceleration lane causes that drivers enter the freeway from the acceleration lane; this driving maneuver results in extra decisions on the freeway by the other driver to drive comfortably and safely. These decisions result mainly in the traffic flow in more driving maneuvers around the acceleration lane to provide gaps for the merging vehicles from the acceleration lane and the transition to the original traffic conditions. Due to an increase in the driving maneuvers the traffic flow characteristics around acceleration lanes are changing, the changed traffic flow characteristics are used to measure the turbulence distance around acceleration lanes.

To measure the turbulence distances two analysis steps are defined. In the first step the changes in the traffic flow characteristics are studied by comparing the traffic flow characteristics around the acceleration lane with the traffic flow characteristics on the basic freeway segment as reference; in this way the changes due to the presence of the acceleration lane become visible, because the changes are the differences between the two situations. Based on the selected data collection method three macroscopic traffic flow characteristics are investigated in this analysis step; the three macroscopic traffic flow characteristics are: lane speed (lane and carriageway), lane flow distribution and lane density to investigate the identified acceleration lane driving maneuvers. In the second step the found changes in the traffic flow characteristics are used to estimate the start and end points of the changed traffic flow characteristics that are related to the presence of the acceleration lane.

To do these analyses, a data set need to be defined. In this thesis macroscopic data is collected with induction loops, induction loops data is widely available; therefore extra data collecting is not necessary, and this method offers an innovative approach to investigate traffic flow behavior over distance, since distance is an important factor to measure the turbulence distance and induction loops are spread over the network with about 500 meter spacings. To increase the accuracy to measure the turbulence distance similar locations are combined to fill the spacings with data from other induction loops.

The available data can not directly used for the analyses, suitable induction loops on suitable locations need to be selected and the data of the suitable induction loops need to be filtered from errors. Therefore, four basic freeway segments are selected with in total fifteen induction loops and eight acceleration lane locations with in total eleven induction loops which are have similar criteria as three lanes and no fixed average speed check and similar geometric design of the acceleration lane. With a coverage of 600 meter upstream and 900 meter downstream of the acceleration lane gore.

Because of the application of the innovative approach to investigate traffic flow characteristics with induction loops, it is important to mention the performance of this approach. Based on the developed selection and filter configuration of the selection of induction loops it is possible to measure the turbulence distance, so this approach with induction loops is working well; however, the amount of selected induction loops is low, especially on the upstream region of the acceleration lane, which increases the possible error in the analyses.

In the analyses to identify changed traffic flow characteristics the means and the standard deviation of the data distributions are investigated. The first traffic flow characteristics investigated are the mean speed on the carriageway and lanes. On the carriageway a decrease in the speed is found just downstream of the acceleration lane, this effect is related to an increase of flow on the outside lane, where the mean speeds are lower than on the other lanes. The standard deviation consist of large changes; these seem to be related to the large difference between combined data of multiple induction loops or due to changes in the traffic flow. In the speeds per lane are not the same changes found, accentually the changes seem to be smaller, only on the outside lane an increase in the mean speed downstream of the acceleration lane is found. No indication of the effects of the driving maneuvers is found on the lane mean speeds. The standard deviation of the lane mean speed increases in most cases, with exception on the outside and center lane just downstream of the acceleration lane where a decrease in noticed. Changes in the traffic flow distributions can be the origin for these results, in the case of that traffic is changing to a more constrained condition, but no a clear relation can be determined between these two traffic flow characteristics.

The changes in the lane flow distributions seem to be larger than in the mean speeds. In the means of the lane flow distribution a clear shift of the flow to the median lane is found upstream of the acceleration lane, this effect can be related to the pre-allocation maneuver; during the acceleration lane the distributions shift to the outside lane to the entrance of traffic flow the outside lane and this effect is larger than the pre-allocation and cooperative lane changes together, since the fraction on the outside lane becomes larger than the basic freeway segment. Further downstream the fraction on the outside lane seems to spread out over the different lanes and the values seem to return to the level of the basic freeway segment. This effect was not expected in the driving maneuver part, but it seem to have a large influence on the duration of the changes turbulence downstream of the acceleration lane. In the standard deviation are also large changes found; however these changes seem to be related to differences between combined induction loops, this is especially the case during the acceleration lane where the differences between the start and end are large.

The lane density has a strong relation with the lane flow distribution; the found results are comparable in the means and standard deviation; therefore, no indications are found of the presence of relaxation in the data downstream of the acceleration lane. It is possible that this effect is further downstream present, since the lane density is still higher than the basic freeway segment within the investigated region of 600 meter upstream and 900 meter downstream of the acceleration lane gore, but this can not measured with the available data.

Based on the found shapes in the means of the lane flow distributions and lane densities a line is fitted on the means of the induction loops. Based on this line fitting, the start of the changed lane flow distribution and lane density is roughly estimated 500 meter upstream and 800 meter downstream of the acceleration lane gore. The estimation of the upstream turbulence distance of 500 meter is very rough based on the availability of only three points to fit the line on. The line fitting on the downstream turbulence distance is more accurate, since 8 induction loops were available in this region.

The turbulence distances that are affected by the presence of the acceleration lanes are empirically measured with macroscopic traffic flow characteristics. The measured traffic flow characteristics are mean speeds, lane flow characteristics and lane densities. The traffic flow characteristics change due to the presence of the acceleration lane. In the mean speeds are small changes found; however no clear shape of these changes is found to estimate the distances; in the lane flow distributions and lane densities are larger changes found with a clear shape. With this shape of the changes in the lane flow distribution and lane density the turbulence distances are roughly estimated. The estimated turbulence distances are 500 meter upstream of the gore and 800 downstream of the gore. Before these distances are incorporated in the NOA, it is wise to validate them.



# Samenvatting

Turbulentieafstanden zijn een belangrijk ontwerpcriteria in the ontwerp van nieuwe snelwegen en aanpassingen aan snelwegen. In Nederland wordt de NOA ("*Nieuwe Ontwerprichtlijnen Autosnelwegen*") gehanteerd als leidraad voor het ontwerp van snelwegen. Turbulentieafstanden zijn nodig om de minimale afstand te bepalen tussen twee opvolgende discontinuïteiten, zoals in- en uitvoeringen. In de laatste herziening zijn er een aantal witte vlekken geïdentificeerd in de NOA die meer achtergrond nodig hebben, een van die onderwerpen zijn de turbulentieafstanden. Turbulentieafstanden zijn al sinds de eerste versie van de NOA in 1975 onderdeel van de richtlijn, maar de waarden in de richtlijn zijn nog niet aangepast op de veranderende bestuurders- en voertuigskarakteristieken. Hierdoor zou het kunnen zijn dat de huidige waarden in de NOA niet meer up to date zijn en geoptimaliseerd kunnen worden. Verder is de druk op infrastructuursystemen aan het groeien. Budgetten krimpen, de publieke acceptatie daalt en hogere prestaties worden verwacht van het systeem. Rijkswaterstaat wil graag de turbulentieafstanden verkleinen om de kosten te drukken van infrastructuurprojecten. Daarom is het doel van deze studie om de turbulentieafstanden te bepalen in de verkeersstroom voor invoeringen.

De eerste vraag die boven komt is: wat is turbulentie in de verkeersstroom? De betekenis van turbulentie in vloeistoffen en gassen is vrij algemeen bekend, maar niet de betekenis van turbulentie in de verkeersstroom. Wanneer de verkeersstroom is benaderd als een vloeistof kan de verkeersstroom worden gezien als een turbulente stroom door de aanwezigheid van rijstrook- en snelheidswisselingen. De NOA omschrijft turbulentie als een specifiek effects van de aanwezigheid van een discontinuïteit, echter de Amerikaanse ontwerprichtlijn (HCM) beschrijft dat turbulentie ook aanwezig is in de normale verkeersstroom zonder de aanwezigheid van een discontinuïteit. In deze studie wordt turbulentie beschouwt als de aantal rijstrook- en snelheidswisselingen in de verkeersstroom, deze benadering is dichtbij de beschrijving van turbulentie in de HCM en in vloeistoffen.

Op basis van de HCM twee situaties kan worden beschouwd: een situatie is waar de turbulentie in de verkeersstroom is normaal en een situatie waar de turbulentie het verkeer meer beïnvloedt door de aanwezigheid van een invoeging. Deze twee situaties worden in deze studie de continue rijbaan en de invoeging genoemd. In de literatuur zijn verschillen gevonden in de keuzes die moeten worden gemaakt tussen de twee situaties. Op basis van de beschrijving van de rijtaak van Minderhoud zijn deze verschillen in de keuzes geïdentificeerd. Rond invoeringen zijn keuzes benodigd om de snelweg op te gaan vanaf de toerit. Deze keuze leidt tot invoegend verkeer vanaf de toerit. Om comfortabel en veilig te rijden op de rijbaan rond invoeringen zijn extra keuzes nodig in de rijtaak door het invoegende verkeer. Deze keuzes leiden tot anticiperende en interactieve rijbewegingen in de verkeersstroom. Hierdoor groeit het aantal rijstrook- en snelheidswisselingen in de verkeersstroom en veranderen ook de verkeersstroomkarakteristieken. De veranderingen in de verkeersstroomkarakteristieken worden gebruikt om de turbulentieafstanden te bepalen.

Om de turbulentieafstanden te bepalen worden er twee stappen gedaan in de analyses. In de eerste stap worden de veranderingen bepaald in de relevante verkeersstroomkarakteristieken door de verkeersstroomkarakteristieken van de continue rijbaan te vergelijken met die rond invoeringen. Op deze manier worden te veranderingen ten opzichte van de referentie in kaart gebracht. Met de bepaalde veranderingen worden de begin- en eindpunten bepaald van de veranderende verkeersstroomkarakteristieken. De afstanden tussen de begin- en eindpunten en het puntstuk van de invoeging wordt gezien als de bepaalde turbulentieafstanden. In deze studie worden de volgende verkeersstroomkarakteristieken vergeleken: gemiddelde snelheden, rijstrookverdelingen en de rijstrookdichtheden. Deze verkeersstroomkarakteristieken zijn bepaald door de verwachte rijstrookbewegingen rond invoeringen en de keuze voor de data aanpak.

Voor het doen van deze analyses is data nodig. In deze studie is data verzameld met inductielussen. Deze lussen liggen wijdverspreid over het snelwegennetwerk in Nederland en de data van deze lussen

in makkelijk te verkrijgen. Verder, de belangrijkste reden voor de keuze van lussen voor het verzamelen van data is de mogelijkheid om een innovatieve methode te onderzoeken voor het analyseren van het gedrag van verkeersstroomkarakteristieken over afstand. Het punt is namelijk dat lussen op locaties ongeveer 500 meter van elkaar liggen en om de turbulentieafstanden beter te bepalen is een nieuwe aanpak nodig. Daarom is ervoor gekozen om lussen van verschillende locaties te combineren om deze afstanden te vullen met lussen van andere, maar gelijkwaardige locaties. Dit is nog niet eerder gedaan en maakt dit een interessante benadering voor het bepalen van de turbulentieafstanden.

Maar eerst is het nodig om een dataset te maken voor de analyses. De beschikbare data bestaat namelijk onder andere uit data met fouten, niet geschikte locaties en lussen. Geschikte locaties en lussen worden geselecteerd, en de data van de lussen is gefilterd. Deze aanpak resulteert in vier continue rijbaan locaties met in totaal vijftien lussen en acht invoeglocaties met in totaal elf lussen op basis van selectie op gelijkwaardige geometrische ontwerpelementen en verkeercondities. De lussen vallen binnen een afstand van 600 meter bovenstrooms and 900 meter benedenstrooms van het puntstuk van de invoeging. Op basis van deze data zijn de analyses gedaan.

Vanwege de innovatieve aanpak voor het verzamelen van data is het belangrijk om even in te gaan op de prestaties van deze aanpak. Met de ontwikkelde selectie- en filterprocedure is het aantal locaties en lussen sterk gefilterd, dit lijkt een positief effect te hebben op het gedrag van de verkeersstroomkarakteristieken over de afstand rond invoegingen en relatie lijken zichtbaar te worden. Echter het overgebleven aantal lussen is laag, dit vergroot de kans op het maken van fouten voor de bepaling van de turbulentieafstand. En daarom is het ook aanbevolen om de gevonden resultaten te valideren met name in de regio's boven- en benedenstrooms van de invoeging waar nog weinig onderzoek is naar gedaan. Met name het bovenstroomse gedeelte waar maar drie lussen geselecteerd zijn vereist verder onderzoek.

Als eerder is beschreven drie verkeersstroomkarakteristieken worden onderzocht naar veranderingen door de aanwezigheid van de invoeging. Als eerste zijn de gemiddelde snelheden onderzocht. In gemiddelde snelheid van de volledige rijbaan blijkt dat de snelheid even afneemt net benedenstrooms van de invoeging, deze verandering lijkt verklaard te kunnen worden door de veranderingen in de rijstrookverdelingen. Ook de standaard deviaties van de gemiddelde rijbaansnelheid is geanalyseerd, hierin zijn ook veranderingen zichtbaar, maar kunnen niet worden gekoppeld aan rijbewegingen in de verkeersstroom en lijken te kunnen worden gekoppeld aan de verschillen per lus. De gemiddelde snelheden per rijstrook lijken constant te blijven en gelijkwaardig met de continue rijbaan. In de standaard deviaties van gemiddelde rijstrooksnelheden lijkt dezelfde conclusie worden getrokken als bij de rijbaansnelheid.

De tweede verkeersstroomkarakteristiek onderzocht zijn de rijstrookverdelingen. Bovenstrooms van de invoeging is er een duidelijke verschuiving gevonden naar de linkerrijstroken. Hieruit blijkt dat er sprake is van voorsorteergedrag bovenstrooms van de invoeging, waardoor de ruimtes toenemen voor het invoegende verkeer. Ter hoogte van de invoeging verschuift de rijstrookverdeling sterk naar rechts, er is zelfs meer verkeer aanwezig op de rechterrijstrook dan op de continue rijbaan. Hieruit blijkt dat het aantal invoegende voertuigen groter is dan het aantal voertuigen dat voorsorteerd en coöperatieve rijstrookwisselingen maken om de invoegingen beter te laten verlopen. Benedenstrooms van de invoeging lijkt de nieuwe verdeling even stand te houden om vervolgens weer richting de normale verdeling te verschuiven. In deze fase verdeelt het verkeer van de invoeging over de verschillende rijstroken van de rijbaan. Ook in de standaard deviaties lijken grote verschillen aanwezig te zijn, deze verschillen lijken net als bij de snelheden te kunnen worden toegeschreven aan de verschillen in de data van de lussen.

De laatste verkeersstroomkarakteristiek is de rijstrookdichtheid. De veranderingen in de rijstrookdichtheid lijken sterk gerelateerd aan de veranderingen in de rijstrookverdelingen. Hierdoor is de verwachte rijbeweging niet echt zichtbaar. Wel kan worden gezegd, aangezien de rijstrookdichtheid op de rechterrijstrook nog niet gelijkwaardig lijkt te zijn aan de continue rijbaan dat het mogelijk is dat deze rijbeweging misschien nog verder benedenstrooms aanwezig is, maar waarschijnlijk is deze aanpak ook niet de beste aanpak om deze rijbeweging te onderzoeken.

Voor de schatting van de turbulentieafstanden zijn de gevonden veranderingen in de rijstrookverdelingen en -dichtheden gebruikt om een lijn te schatten over de data van de lussen, op deze manier zijn de begin- en eindpunten van de veranderingen in deze karakteristieken ruw geschat op 500 meter bovenstrooms en 800 meter benedenstrooms van het puntstuk. Er moet hierbij worden vermeld, dat met name de schatting van de bovenstroomse afstand slechts gebaseerd is op drie lussen wat de kans op een fout in de schatting sterk vergroot. Geadviseerd is om de gevonden afstanden te valideren.

In deze studie zijn turbulentieafstanden rond invoegingen geschat op basis van de gemeten veranderingen in de verkeersstroomkarakteristieken. De onderzochte verkeersstroomkarakteristieken zijn de gemiddelde snelheden, rijstrookverdelingen en rijstrookdichtheden. In de gemiddelde snelheden zijn kleine verschillen waargenomen, maar er is geen duidelijke vorm gevonden die gebruikt kon worden voor het schatten van de turbulentieafstanden. In de rijstrookverdelingen en rijstrookdichtheden zijn grotere veranderingen waargenomen, met name de verschuiving bovenstrooms naar links, naar rechts tijdens de invoegingen en verderop benedenstrooms terug naar links naar de oorspronkelijke verdeling is opvallend. Deze verschuivingen zijn gebruikt om de turbulentieafstanden ruw te schatten op 500 meter bovenstrooms en 800 meter benedenstrooms van het puntstuk van de invoeging. Voordat deze waarden in de NOA gebruikt kunnen worden, wordt er geadviseerd om de gevonden resultaten te valideren.





# 1

## Introduction

### 1.1. Background

New freeways and freeway modifications are designed based on guidelines. The NOA (in Dutch: “*Nieuwe Ontwerprichtlijn Autosnelwegen*” [1]) is the decisive freeway design guideline in the Netherlands. The NOA is revised at this moment, and in this process a few topics were found which need more background than is available now. One of these topics is the turbulence distance. Turbulence distance are used to define the minimal distance between on-ramps and off-ramps to enter and exit the freeway, or to switch from one freeway to the other, and lane drops and lane additions. The NOA defines turbulence distances as distances upstream and downstream of the gore (see figure 1.1) of the discontinuity where the driving behavior is influenced by the presence of the discontinuity. Discontinuity is the category of which on-ramps, off-ramp, lane drops and lane additions are part of in the design of freeways, within the discontinuity category seven types of discontinuities are distinguished. The seven types of discontinuities are: acceleration lanes, deceleration lanes, weaving segments, diverge segments, merge segments, lane drops and lane additions. The different discontinuities have specified turbulence distances; table 1.1 shows the current applied turbulence distances of the NOA of the seven discontinuities.

Table 1.1: Turbulence distances in the NOA in meters [1]

location freeway segment	design speed limit				reference point
	120 km/h	100 km/h	80 km/h	50 km/h	
upstream of acceleration lane	150	130	100	n.a.	end of gore
downstream of acceleration lane	750	600	500	n.a.	end of gore
upstream of merge segment	150	120	100	n.a.	end of gore
downstream of merge segment	375	300	250	n.a.	end of gore or end of the taper
upstream of deceleration lane	750	600	500	n.a.	start of gore or end of taper
downstream of deceleration lane	150	120	100	n.a.	start of gore
upstream of diverge segment	150	120	100	n.a.	begin of block marking
downstream of diverge segment	150	120	100	n.a.	begin of gore
upstream of lane drop	375	300	250	n.a.	begin of gore
downstream of lane drop	150	120	100	n.a.	end of gore

Since 1975 turbulence distances are part of the freeway design guideline in the Netherlands. In 1975 the ROA (in Dutch: “*Richtlijnen voor het ontwerpen van autosnelwegen*” [2]), and a former version of the NOA, was introduced as guideline to design freeways in the Netherlands, and included at that

time already turbulence distances. Turbulence distances are used to limit the negative effects of the discontinuity on the traffic flow safety and possibly also the traffic flow performance. Kraaijeveld found in his master thesis that around acceleration lanes, deceleration lanes and lane drops the probability of traffic accidents increases with a factor 5, compared with other freeway segments [3]. This result indicates the negative impact on the safety due to the presence of the discontinuity.

Since 1975 vehicle characteristics and driver behavior are changed; however, the turbulence distances are not changed in the meanwhile. Therefore, the turbulence distances are possibly not representative anymore for the real traffic flow behavior, which means that the current turbulence distances are possibly out of date and need to be optimized to improve freeway design.

Furthermore, the pressure on infrastructure systems is increasing [4]. The expectations are increasing on the requirements of infrastructure systems, while budgets are limiting for (re-)construction and maintenance. There are four pressures distinguished: limited budgets, less public acceptance, increasing performance requirements and higher legal requirements. It has become more difficult and expensive to plan new infrastructure as people claim a right to the view they always had, and that the landscape should not be disturbed by new infrastructure; while we do not like large travel times to home due to congestion, and being involved of crashes. Further we do not like huge amount of taxes due to an expensive freeway network. Therefore, Rijkswaterstaat, the Dutch freeway manager, is searching for freeway design elements of which the guidelines can be optimized to limit the (re-)construction costs. The search for cost reducing elements is already performed in multiple researches in order of Rijkswaterstaat [5, 6]; however, those researches did not include turbulence distances. Thus, turbulence distances are a possibility for limiting of the cost of infrastructure.

Further some elements of discontinuities are shown in figure 1.1 based on an acceleration lane; these terms of these elements are used in this thesis.

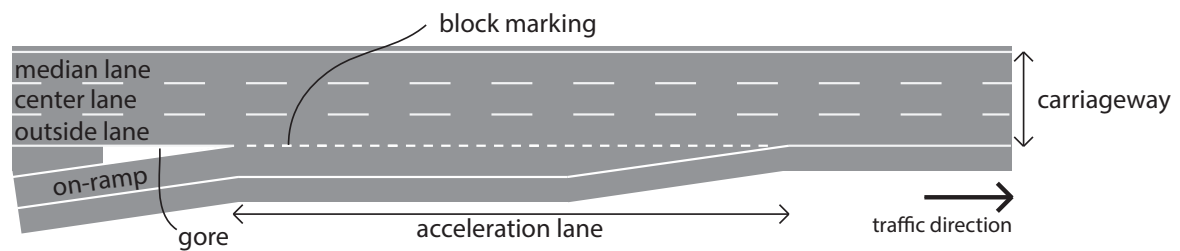


Figure 1.1: Elements of an acceleration lane

## 1.2. Definition traffic flow turbulence

Before turbulence in traffic flow around discontinuities is investigated; the definition of turbulence in traffic flow is determined. Turbulence in traffic flow is not often researched and is thus not well known and defined; however, people are often familiar with turbulence in fluids and gases. Flows are distinguished in laminar and turbulent in fluid and gas mechanics. Laminar flows are characterized by straight flowing in parallel without mixture between the layers of the flow [8], and turbulent flow is characterized by chaotic flow property changes inside the flow. These chaotic property changes consist of variations in flowing directions, flowing in vertexes and changes in pressure and velocity [9]. Figure 1.2 shows an example of turbulent flow. In fluid and gas mechanics turbulence is defined as "sudden, violent movements of gases or fluids" [10].

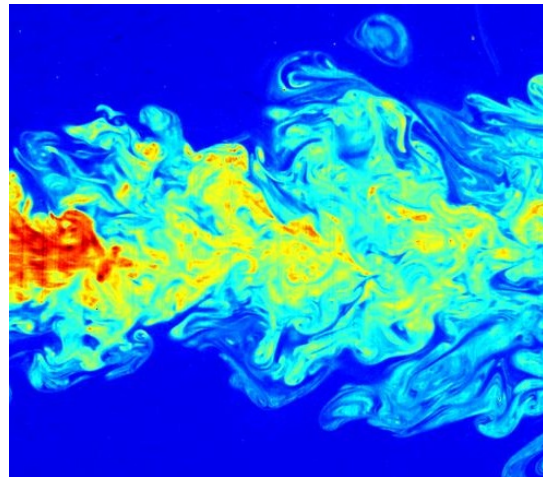


Figure 1.2: Example of turbulent fluid [7]

When this definition is applied on traffic flow, traffic flow can be seen as a turbulent flow often; within the traffic flow drivers are changing lanes and changing their speed. The lane changes and speed changes can be seen as mixture between layers and as changes in pressure and velocity. However, the amount of these turbulent movements in the traffic flow differs in different states of the traffic flow and around discontinuities.

The NOA explains turbulence in traffic flow as: "Turbulence is characterized by deviations in the headways between vehicles and the distribution of the traffic flow over the lanes. Related driving behaviors are e.g. braking actions, avoidance maneuvers or (anticipating) lane changes". The NOA states also that turbulence is present upstream and downstream of a discontinuity in the traffic flow.

In the United States a different design guideline is used to design freeways; this freeway design guideline is the Highway Capacity Manual (HCM) [11]). Turbulence is mentioned a few times in the HCM; however, a clear definition or explanation is not described for turbulence in traffic flow. Based on the sentences close to turbulence in the HCM, turbulence can be explained ambiguous. Turbulence is mentioned as an extra effect in the traffic flow around weaving segments, on-ramps and off-ramps. This approach is close to the description of turbulence in the traffic flow in the NOA. However, the HCM is also stating: "Traffic in a weaving segment is subject to turbulence in excess of that normally is present on basic freeway segment" [11]; based on this statement can be concluded that turbulence is not only present around discontinuities, but also on basic freeway segments in lower amounts and turbulence is seen as a quantity. This is closer to the definition of turbulence in fluids and gases, because on basic freeway segments traffic is changing lanes and changing speeds, which can be seen as turbulence in the traffic flow. With this explanation a division is made between basic freeway segments with normal turbulence and acceleration lane with a larger amount of turbulence in the traffic flow.

In this thesis the description of turbulence in traffic flow in the HCM is applied; in specific, the explanation which is the closest to the explanation of turbulence in fluids and gases. Therefore, the used definition of turbulence in traffic flow is:

---

*"turbulence describes the amount of changes in the traffic flow, based on speed changes and lane changes"*

---

As described in the HCM, turbulence differs around discontinuities and on basic freeway segments; also the state of the traffic flow can affect the amount of lane changes and speed changes in the traffic flow. Two main factors can be distinguished which affects the amount of turbulence: traffic flow characteristics and freeway design characteristics. Due to the first factor, traffic flow characteristics, turbulence is changing without changing the design of the freeway; examples are vehicle population,

driving capabilities and traffic state. Due to second factor, freeway design characteristics, turbulence is changing due to changes in the design of the freeway within the same traffic flow characteristics; examples are acceleration lanes, curves and sags [12]. In this thesis the first factor is held constant, and the second factor differs due to the presence of the discontinuity.

### 1.3. Research objective

Based on section 1.1 and 1.2 is the research objective defined. Also in section 1.1 seven types of discontinuities are distinguished. Acceleration lanes are already topic of many researches and master theses [13–15]; this literature can be used as background for this thesis. Therefore, the turbulence distance of acceleration lanes is investigated in this thesis.

Turbulence around acceleration lanes changes the traffic flow characteristics. The distances upstream and downstream of the gore are the distance where these changes are present. These changes could be empirically measurable in traffic flow. Therefore, the research objective of this thesis is defined as:

---

*“to empirically measure the upstream and downstream distance that is affected by the presence of an acceleration lane”*

---

### 1.4. Report outline

This section describes the outline of this report. Figure 1.3 shows the framework of this thesis; the identified thesis steps are spread over the different chapters.

Chapter 2 describes the background of turbulence in traffic flow. In this chapter the driving task of Minderhoud [16] is used to describe the causes of turbulence in traffic flow and the causes of the increase in turbulence around acceleration lanes. These causes consist decisions in the traffic flow; based on these decisions driver maneuvers in the traffic flow are identified, with these driving maneuvers around acceleration lanes expectations are defined for the changes in the microscopic and macroscopic traffic flow characteristics.

Chapter 3 describes the defined methodology for the analyses. In this chapter first the used analysis approach is described. For the analyses a compatible data collection method/data type and data collection approach is selected for the analyses. With the selected data collection method the applied analyses are defined further and test are defined to test the expectations in the traffic flow characteristics in chapter 2.

Chapter 4 the selected data collection method and approach is used to create the data set for the analyses. The main element in the data processing is the selection of suitable locations for investigation and the filtering of the data from these location. Furthermore, since an innovative data collection method is selected the effects of this method is described in this chapter.

Chapter 5 describes the results of the selected analyses. The first part consist of the description of the changes in the investigated traffic flow characteristics. Based on these changes in the different traffic flow characteristics the turbulence distances are estimated.

This report ends with the main findings, conclusions and recommendations in chapter 6.

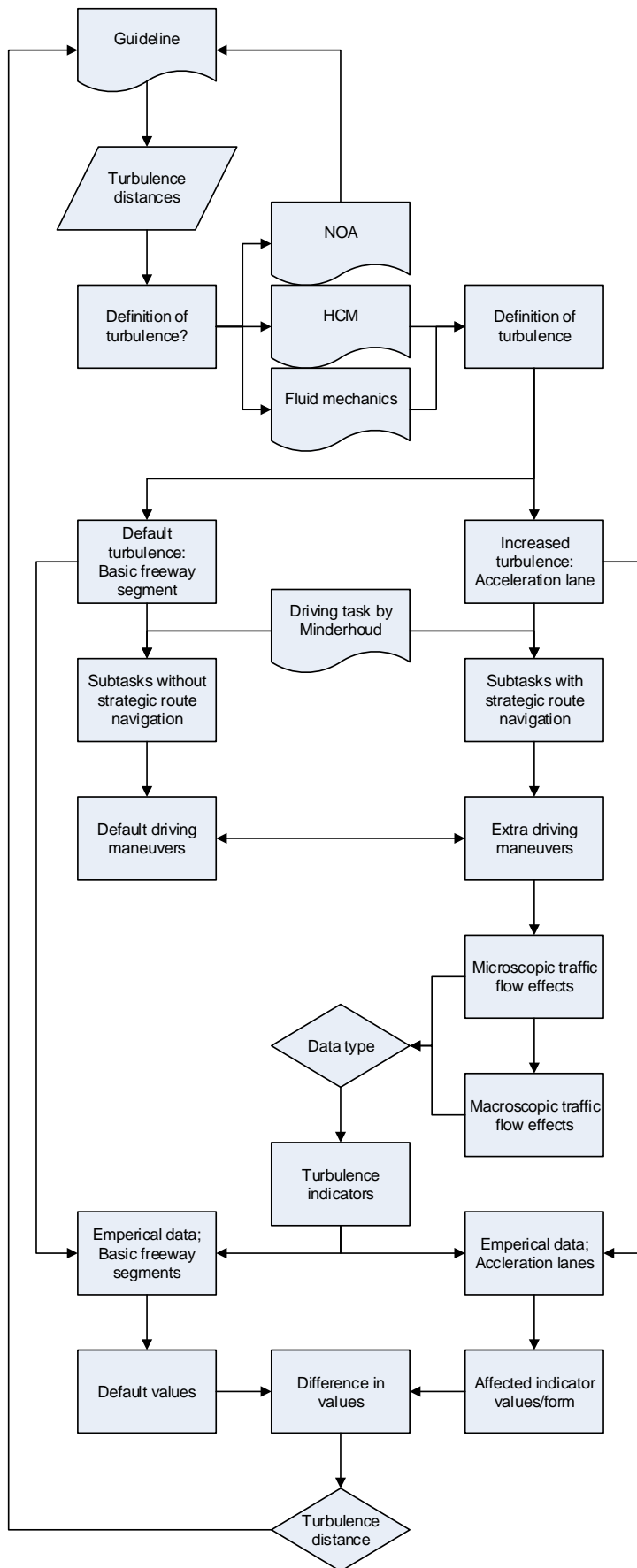


Figure 1.3: Detailed framework thesis



# 2

## Theoretical background

This chapter describes the theoretical background of this thesis. In this chapter the cause of turbulence is described by the driving task of Minderhoud [16]; based on the driving task the driving maneuvers are identified on the basic freeway segment and around the acceleration lane. With the identified driving maneuvers the expected changes in the traffic flow characteristics are defined, these expectations are used to define the suitable analyses in the next chapter, and to be tested in chapter 5.

### 2.1. Causes of turbulence

In the previous chapter turbulence in traffic flow is defined as the amount of changes in the traffic flow, based on speed changes and lane changes. Now the question is rising why are these changes present in traffic flow; especially, why are they normally present in the traffic flow on basic freeway segments, and why is the turbulence changes in the traffic flow around acceleration lanes?

Driving from origin to destination is often described as a driving task. This driving task consists of all tasks which must be executed by the driver to reach his/her destination safely, comfortably and timely. The driving task is classified in multiple ways. Minderhoud [16] proposes an elaborate description of the driving task by explicit distinguishing between subtask dealing with interaction of the driver with the roadway, the interaction between the driver with other traffic participants and the route navigation on strategic level. In total, Minderhoud distinguished five subtasks:

1. *Strategic route navigation*: subtask in which drivers choose their destination, travel direction, route and roads. The choice for a destination or route depends on specific goals and preference of the drivers, and on the traffic conditions.
2. *Lateral roadway subtask*: a collection of decisions of the driver which are needed to guide the vehicle properly and comfortably over the available infrastructure and its elements such as driving lanes, curves and on- and off-ramps, while dealing with the lateral direction.
3. *Longitudinal roadway subtask*: the decisions of the driver which are needed to guide the vehicle properly and comfortably over the available infrastructure and its elements while dealing with the longitudinal direction.
4. *Lateral vehicle interaction*: collection of decisions needed to guide the driver-vehicle unit properly and comfortable around obstacles, vehicles and possible other traffic participants, while focusing on the lateral direction.
5. *Longitudinal vehicle interaction*: decisions needed to guide the driver-vehicle unit around obstacles, vehicles and possible other traffic participants, while focusing on the longitudinal direction.

Figure 2.1 shows a visual representation of the vehicle interaction and the roadway subtasks.

The subtasks are seen as the cause of turbulence in traffic flow; the subtasks consist of a set of decisions to perform lane changes and speed changes. As described before, in this thesis turbulence around acceleration lanes and on basic freeway segments are distinguished. On both types of freeway segments subtasks are applicable; however, a difference is found between acceleration lanes and

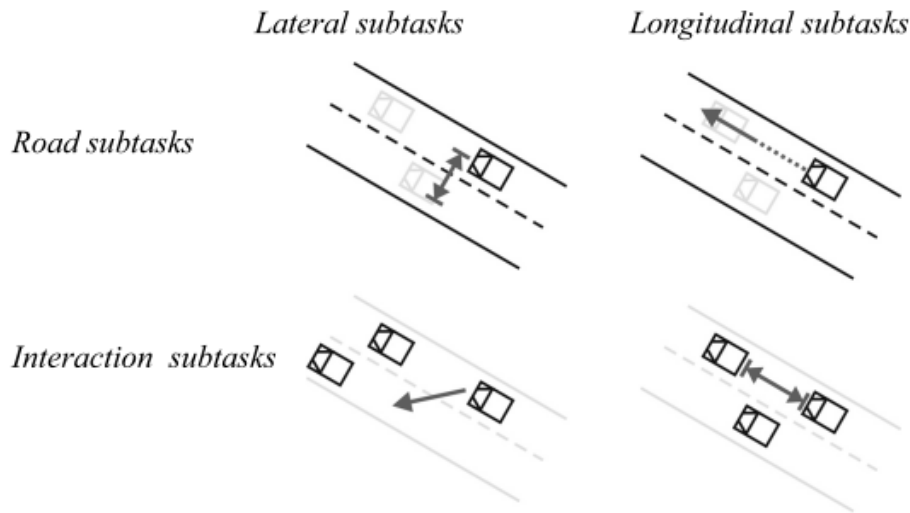


Figure 2.1: Schematisation of driver subtasks [17]

basic freeway segment. Around acceleration lanes traffic enters the freeway from the on-ramp. This means that the strategic route navigation is present on this segment; this subtask consist of the merge maneuver from the acceleration lane to the freeway. While on the basic freeway segment this subtask is not present in the traffic flow and is only present when entering and exit the freeway, and to switch from one freeway to the other.

Due to the presence of the strategic route navigation traffic enters the freeway from the acceleration lane. This presence results in a changed situation on the freeway. The headways between vehicles become smaller, and new speed differences are occurring between vehicles from the acceleration lane and vehicles on the freeway. Therefore, drivers on the freeway need to make new decisions to cope with the entered vehicles; extra interaction and roadway subtasks are arising in the driving task of the drivers to continue safely and comfortably reach their destination around acceleration lanes. These extra subtasks propagate upstream and downstream of the acceleration lane; this distance is seen as the distance that is affected by the presence of the acceleration lane.

## 2.2. Driving maneuvers

As described in the previous section decisions made by the driver to drive safely and comfortably to reach his/hers destination result in lane changes and speed changes in the traffic flow; these decisions are seen as the cause of turbulence in the traffic flow. The decisions can be distinguished in multiple different type of lane changes and speed changes based on the origin of the decision. These lane changes and speed changes are related with driving maneuvers. Multiple researches investigated driving maneuvers to understand traffic flow better, and to improve traffic simulations, like FOSIM. These traffic simulations are often used for decision making in freeway design.

In the next subsections the driving maneuvers on basic freeway segments and around acceleration lanes are described with a literature study. As described in the previous section on basic freeway segment the default or normal set of decisions are present in the traffic flow. The default set of decisions is also present around acceleration lane (image no traffic from the on-ramp); however, due to the entering traffic the set is increased which is arising by the strategic route navigation subtask and more interaction and roadway subtasks. When linked to driving maneuvers, on basic freeway segments a default amount and set of driving maneuvers is available in the traffic flow; around acceleration lanes the default amount and set is available plus a set and amount of extra driving maneuvers due to the higher amount of subtasks. In the next section the extra driving maneuvers around the acceleration lane are used to develop expectations of the effects of the presence of the acceleration lane on the traffic flow which are used to measure the turbulence in the traffic flow.



### 2.2.1. Basic freeway segment

The found driving maneuvers on basic freeway segments are:

- *Overtaking*: lane changing to an adjacent left lane to pass a slower driving vehicle to maintain desired speed [15].
- *Following driver*: encountering a slower driving vehicle without overtaking possibilities. Keeping minimal headway with the slower driving vehicle [18].
- *Cooperative lane changing*: a driver who is (nearly) adjacent to a driver on the right adjacent lane who is starting to change one lane to the left to overtake changes one lane to the left adjacent lane. This driving behavior is only occurring on carriageways with three or more lanes [15, 19–21].
- *Keeping right*: switching back to the rightmost lane because of traffic rules. When the vehicles are overtaken changing to the rightmost lane to follow the traffic rules when the gaps are large enough on the rightmost lane [15].
- *Acceleration*: when driving slower than the desired speed and the headway with the leading vehicle is larger than the desired minimal headway [16, 17, 22].
- *Courtesy yielding*: the courtesy yielding behavior is identified as a deceleration by a driver on the outside lane to enlarge to headway with the leader to provide a gap for a lane changing driver from the right adjacent lane to provide a gap [13, 15, 21].
- *Relaxation*: when driver accept a shorter headway than desired, drivers accept shorter headway than desired only for a short period of time, after this period the headway increased to a more comfortable headway, or the driver will overtake [15, 23–26].

Figure 2.2 shows the found driving maneuvers on the basic freeway segment.

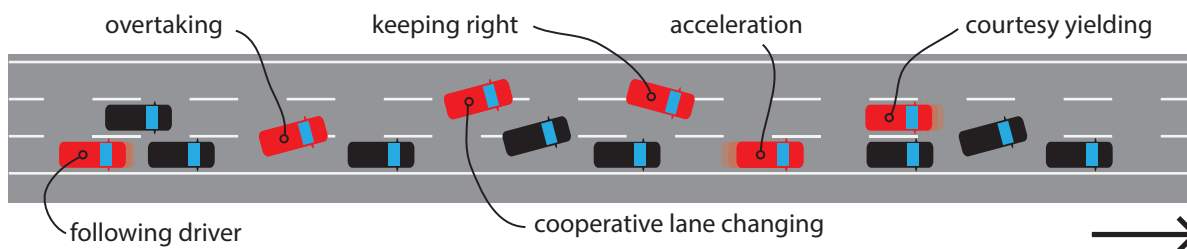


Figure 2.2: Driving maneuvers on basic freeway segments

### 2.2.2. Acceleration lane

The found driving maneuvers around acceleration lanes are:

- *Pre-allocation*: the behavior that drivers upstream of the acceleration lane changes lane to the adjacent left lane to provide gaps on the outside lane without direct interaction with a merging driver [15].
- *Courtesy yielding*: the courtesy yielding behavior is identified as a gentle deceleration by a driver on the outside lane to enlarge to headway with the leader to provide a gap for the merger on the acceleration lane to merge safely on to the outside lane [13, 15, 21].
- *Cooperative lane changing*: a driver who is (nearly) adjacent to the merging vehicle changes one lane to the adjacent left lane [15, 19–21]. Around acceleration lanes this driving maneuver is also present more present on the center lane to the median lane, due to cooperative lane changes from the outside lane.
- *Keeping right*: switching back to the rightmost lane because of traffic rules. When the vehicles are overtaken changing to the rightmost lane to follow the traffic rules when the gaps are large enough on the rightmost lane. Due to the many left lane change maneuvers upstream the expectation is that the number of keeping right maneuvers is increased [15].
- *Merging*: the merging behavior is the mandatory lane change on acceleration lanes and the function of the acceleration lane. The drivers are changing lanes from the acceleration lane to the adjacent left lane, which is the outside lane of the carriageway. The drivers have to perform this action before the end of the acceleration lane [15].

- *Relaxation*: when drivers merge on the outside lane the headways become smaller on the outside lane, often these headways are smaller than the minimal headways and drivers will accept these smaller headways for a short period. After this short period driver will enlarge these short headway to at least the minimal headway [15, 23–26].

Figure 2.3 shows the found driving maneuvers around the acceleration lane.

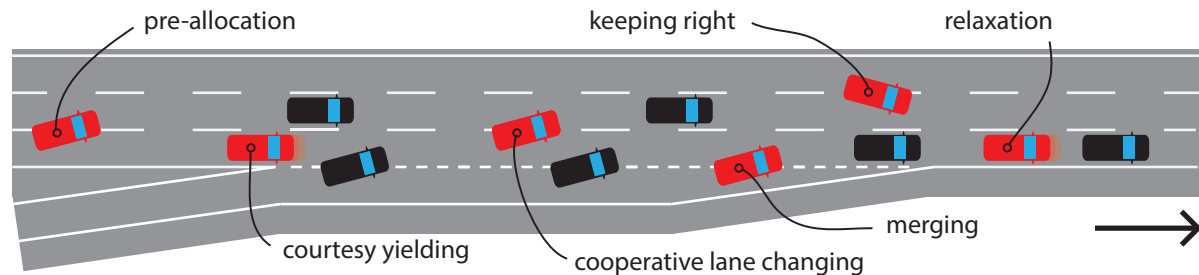


Figure 2.3: Driving maneuvers on acceleration lanes

### 2.3. Traffic flow characteristics

In the last section of this chapter the expectations for the results are determined. The expectations for the changes in the changes in the traffic flow characteristics are determined based on the found driving maneuvers around acceleration lanes, since the turbulence that is affected by the presence of the acceleration lane is investigated to determine the turbulence distance. Based on the expected effects by the presence of the acceleration lane the analysis approach, data collection method and suitable analyses are selected in the next chapter.

Effects in traffic flow are investigated and measured with traffic flow characteristics. Traffic flow characteristics are distinguished in two levels, microscopic and macroscopic traffic flow characteristics. Microscopic traffic flow characteristics relate to the individual driver. Examples of microscopic traffic flow characteristics are time headways and individual speeds. Macroscopic characteristics relate to the properties of the traffic flow as a whole. Examples of macroscopic traffic flow characteristics are flow, time mean speed and density. The traffic flow characteristics are used to measure the effects of turbulence in traffic flow.

The found driving maneuvers in the previous section can be related to changes in the macroscopic and microscopic traffic flow characteristics. To determine the expected effects of the presence of the acceleration lane in the traffic flow three steps are performed. First the driving maneuvers of the acceleration lane are divided on three regions of the acceleration lane, upstream, during and downstream; by this approach the effects by the different driving maneuvers are spread over the distance around acceleration lanes; second, the different driving maneuvers are related to changes in microscopic traffic flow characteristics, since this contains the direct effects of individual driving maneuvers. In last step the changes in the microscopic traffic flow characteristics of the different maneuvers divided in regions are related the effects on macroscopic traffic flow characteristics. The expected changes in the traffic flow characteristics are used to build the expectations for the results.

Figure 2.4 shows the process to determine the changes in the traffic flow characteristic based on the driving maneuvers. Upstream of the gore only the pre-allocation maneuver is expected, which is based on decision to make left lane changes to provide gaps on the outside lane. Therefore, more left changes and a changed lane flow distribution are expected [14]. During the acceleration lane three maneuvers are expected, merging, courtesy yielding and cooperative lane changing. These lane changes are based on left lane changes and speed reductions; therefore, more lane changes and speed changes [13, 15], and a changed lane flow distribution and changes mean speed (on the different lanes and carriageway) are expected. Downstream of the acceleration lane keeping right and relaxation are found; these maneuvers are related with more right lane changes and increasing time headways [15], and changed lane flow distribution and density.

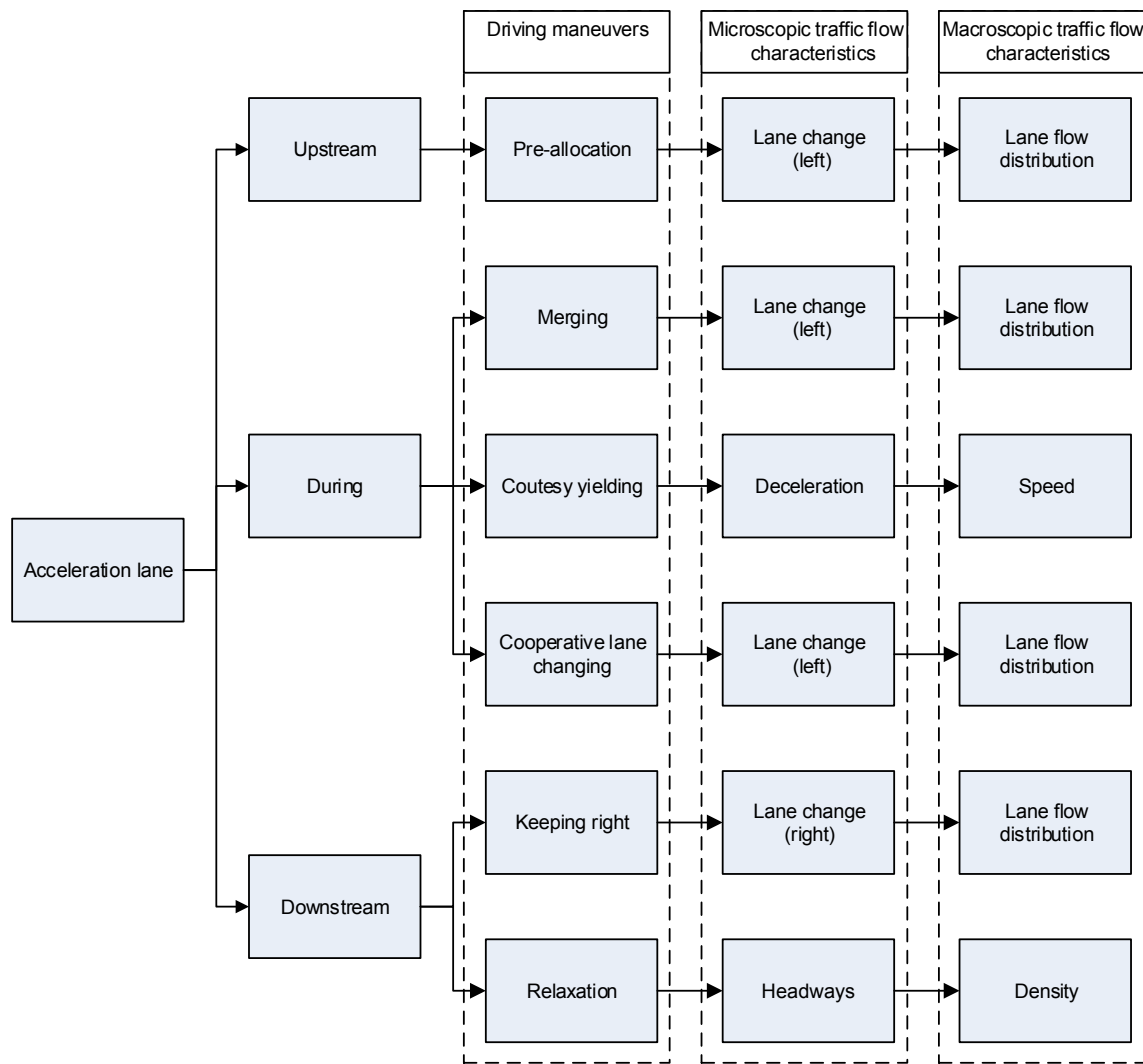


Figure 2.4: Effects in traffic flow characteristics

In the next chapter macroscopic traffic flow characteristics are defined to measure turbulence around acceleration lanes. Therefore, mean speeds, lane flow distributions and densities are identified to be investigated based on the expected effects by the presence of the acceleration lane. Further, the characteristics are investigated on two parameters: the mean and standard deviation based on the defined analyses in the next chapter. Therefore, for the defined changed traffic flow characteristics expectations are defined of the mean and standard deviation.

First the expectations of the mean speeds is defined; mean speeds are determined on two freeway elements, the carriageway speed and the lane of the different lanes. Expected is that the speed is not decreasing upstream of the acceleration lane; during the acceleration lane expected is that the speed is decreasing slightly due to courtesy yielding; downstream of the acceleration the expectation is that within the relaxation maneuver the speed is increasing to the original level. For the standard deviation, upstream no changes are expected, during the acceleration lane expected is that the standard deviation is increasing slightly and stays increased until the relaxation maneuver is performed. These expectations are mainly expected on the outside lane, because the identified speed changing maneuvers are mainly expected to be occurring on the outside lane.

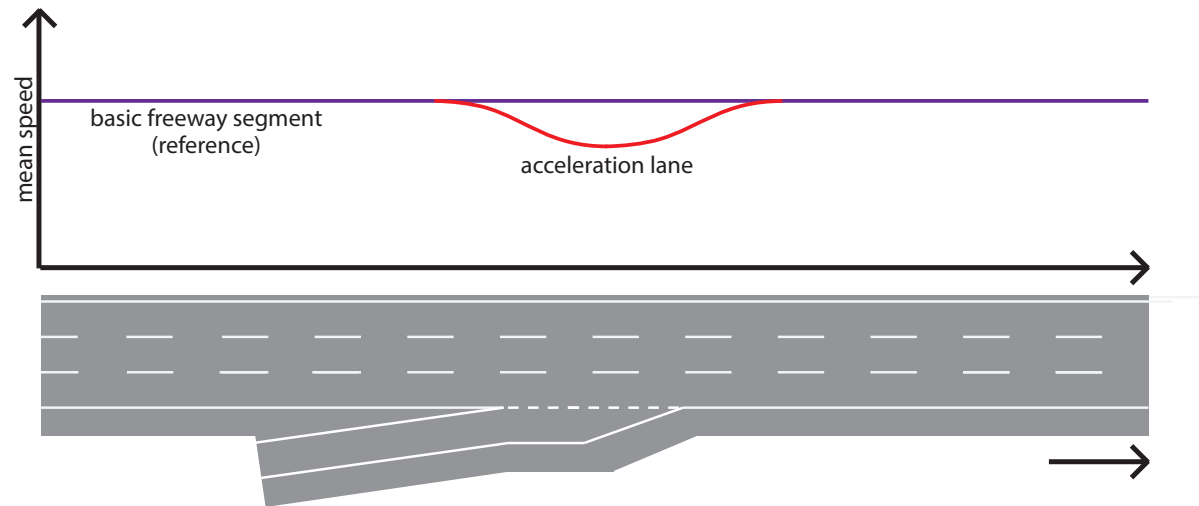


Figure 2.5: Expected changes in the mean speed

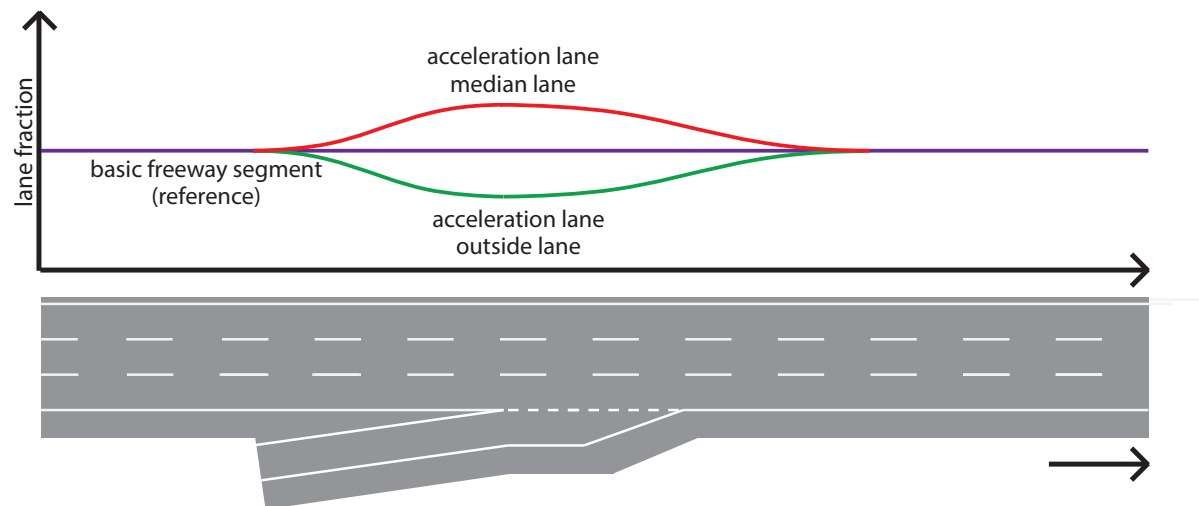


Figure 2.6: Expected changes in the lane flow distribution

In the lane flow distributions the expectation on the outside lane is that upstream the fraction is decreasing (pre-allocation), during the acceleration lane the fraction is increasing due to merging traffic and downstream of the acceleration lane the fraction is increasing more due to keeping right. For the standard deviation the expectation are more difficult to define. On the outside lane the standard deviation is expected to increase due to the extra maneuvers, during the acceleration lane the standard deviation continues increased and decreases slowly downstream of the acceleration lane.

The last macroscopic traffic flow characteristics is the lane density, the lane density is related to the relaxation maneuver. In this process an increase in the density is expected on the outside lane mainly downstream of the acceleration lane. In the analyses the density investigation focuses on the downstream part; however, the effects in the density on the other parts are also investigated. For the standard deviation an increase is expected just downstream of the acceleration lane in the process to increase the headway by relaxation.

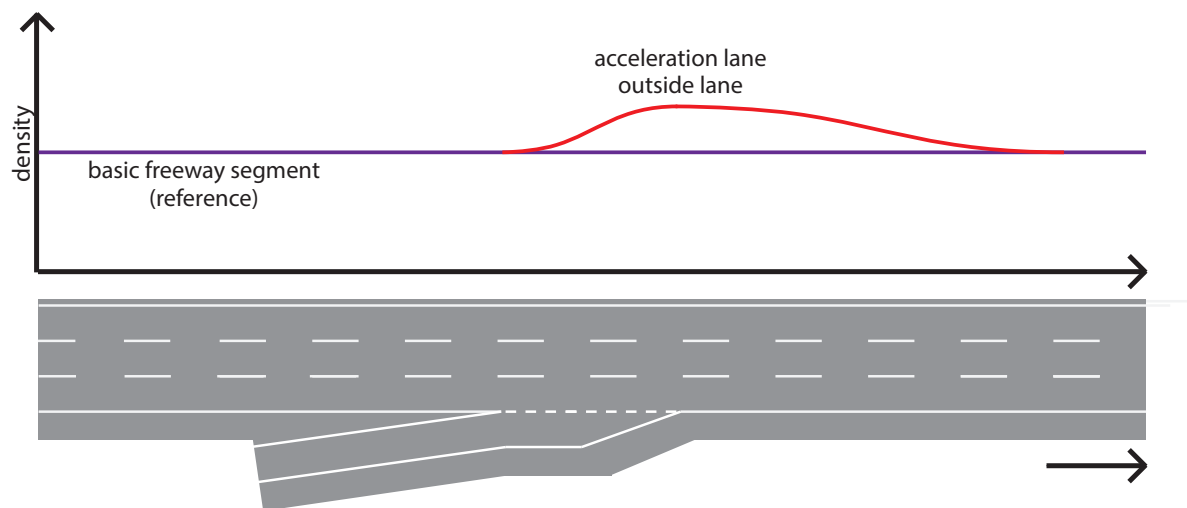


Figure 2.7: Expected changes in the lane density

The last expectation is on the measurement of the turbulence distance. In the NOA the turbulence is defined as a distance of 130 meter upstream and 600 meter downstream of the gore, and it is unknown if a shorter or larger turbulence distance is expected due to the changes in driver and vehicle composition. Therefore, the expectation is to measure a turbulence distance of 130 meter upstream and a turbulence distance of 600 meter downstream.

## 2.4. Conclusions on theoretical background

In this chapter the causes of turbulence in traffic flow and changes around an acceleration lane are investigated. Driving from origin to destination can be described as a driving task; the driving task consists of all tasks executed by the driver to reach his/her destination safely, comfortably and timely. Minderhoud [16] distinguished the driving task in five subtasks: strategic route navigation, lateral roadway subtask, longitudinal roadway subtask, lateral vehicle interaction and longitudinal. These subtasks are seen as the cause of the turbulence in traffic flow; these subtasks result in lane changes and speed changes in the traffic flow. On roadways with no discontinuities the strategic roadway task is not available in the set of tasks, while around acceleration lanes this task is available in the traffic flow; this is a main difference between the acceleration lane and the basic freeway segment. Further, due to the availability of the strategic route navigation subtask, more interaction is arising between other drivers and the merging driver to reach their destinations safely and comfortably. The presence of the strategic route navigation and the increase of more interaction subtasks are the cause of the changed turbulence around acceleration lanes.

These subtasks are related with driving maneuvers in the traffic flow. Based on the route navigation subtask and extra interaction subtasks, multiple types of driving maneuvers are increasing and new types of driving maneuvers are arising around acceleration lanes in the traffic flow. Around acceleration lanes, two new driving maneuvers are arising, pre-allocation and merging. Further, the following types of driving maneuvers increase around acceleration lanes: courtesy yielding, cooperative lane changing, keeping right and relaxation.

The extra driving maneuvers around acceleration lanes are related to changes in traffic flow characteristics to measure the turbulence distances by the presence of the acceleration lanes. The traffic flow characteristics are divided into two levels, macroscopic and microscopic. In relation with the selected data collection method, macroscopic traffic flow characteristics are investigated to find the changes in the mean speeds, lane flow distributions and lane densities to measure the turbulence distances.



# 3

## Methodology

The next step in this thesis is to describe the applied methodology. The methodology is described in three main parts. In the first part the approach is described to measure the turbulence distance of acceleration lanes in the traffic flow. The second part describes the applied method to collect data for the analyses, and the last part describes the used analyses to measure the turbulence distance.

### 3.1. Measuring turbulence distance

Previous chapter showed that extra decisions are required by the drivers in the traffic flow to drive comfortably and safely around acceleration lanes. The extra decisions result in extra lane changes and speed changes in the traffic flow. The changes in the amount of lane changes and speed changes can be measured in the traffic flow characteristics on macroscopic and microscopic level.

In the previous chapter the expected changes in the traffic flow characteristics are described based on the extra driving maneuvers in the traffic flow around acceleration lanes. The changes in the traffic flow characteristics are expected and need to be determined based on empirical measurements. When the changes in the traffic flow characteristics are measured, the changes can be used to determine from where to where the traffic flow characteristics are changed in comparison with the normal situation, the basic freeway segment. The information of the start and end point of the changed traffic flow characteristics around acceleration lane is used to determine the turbulence distances around acceleration lanes. The distances between the start and end points and the gore of the acceleration lanes is seen as the turbulence distance.

Therefore, the approach to measure the turbulence distances consists of two steps. The first step is to measure the changes of the traffic flow characteristics that are affected by the presence of an acceleration lane. The second step is to determine the start and end point of the changes in the traffic flow characteristics to define the turbulence distances.

### 3.2. Case-control study

The performance of both steps is both related to the measurement of the changes in the traffic flow characteristics. Therefore, the changes need to be made visible with the applied measurements. The applied approach to investigate the changes in the traffic flow characteristics is to compare the traffic flow characteristics of the changes situation with the normal situation. By comparing these two situations the differences are investigated in the traffic flow characteristics. These differences are the changes in the traffic flow characteristics by the presence of the acceleration lane and thus the changes in turbulence.

In other research fields this comparison is called a case-control study. Case-control studies are often used to identify factors that may contribute to a medical condition by comparing people who have that condition (the "cases") with people who do not have the condition but are otherwise similar (the "controls") [27, 28]. The control is also defined as the reference, where the default values are

present. In this thesis the turbulence and the traffic flow characteristics are defined around acceleration lanes as the case situation; the basic freeway segment with the default turbulence and the default traffic flow characteristics is defined as the control or reference situation.

For the analyses of the case-control study it is important to compare equal situations, where the only difference is the investigated element. In this research the investigated element is the presence of the acceleration lane. Therefore, all other elements need to be comparable in the two situations. The comparable elements are distinguished in two types: the geometric characteristics and the traffic conditions.

### 3.2.1. Geometric characteristics

Geometric characteristics consist of the freeway design elements of both segments. As described above, the only difference in the geometric characteristics need be the presence of the acceleration lane. The geometric characteristics are also necessary in chapter 4 as criteria to select basic freeway segment locations and acceleration lane locations for the case-control study for the data processing. Main geometric characteristics are the number of lanes and the speed limit. Further in the data processing in chapter 4 rush hour lanes and overtaking restrictions were found available in the data source and used to make the compared segments more comparable. Furthermore, fixed average speed checks ("trajectcontrole") are found to have a large influence on the traffic flow characteristics in appendix A and thus excluded from the comparison. For the main characteristics the number of lanes and the speed limit is set on 3-lanes with a speed limit of 100 km/h. The background of these numbers is the selected data collection method, induction loops. Induction loops are in larger amount present on the network in the West of the Netherlands on freeways with 2x3 lanes with a speed limit of 100 km/h, to increase the amount of possible measurement locations these criteria are selected. Table 4.1 summarizes the applied geometric characteristics in the thesis.

Table 3.1: Applied freeway design characteristics

freeway design characteristic
3-lanes
1-lane acceleration lane
100 km/h speed limit
no rush hour lane
no overtaking restrictions
no fixed average speed check
length acceleration lane 217.5-362.5 meter

### 3.2.2. Traffic condition

The second element is the traffic condition on both situations. In traffic flow theory the traffic condition is related to the position of the traffic flow characteristics on segment in the fundamental diagram. Figure 3.1 shows a simple version of the fundamental diagram defined by Daganzo [29]. In the general applied fundamental diagram two conditions are distinguished: free flow and congested flow. In free flow is more or less unconstrained, in this condition drivers are relative large headways speed differences are large between lanes and relative much lane changes are performed. In congested flow the traffic becomes constrained, the differences in speed between the lanes reduces and the number of lane changes reduces.

In the analyses the free flow condition is studied. In the congested condition less turbulence is considered than in the free flow condition. When a congested traffic flow is approaching the acceleration lane relative low space is available in the traffic flow to interact and anticipate on the entering traffic. Driving maneuvers like pre-allocation and cooperative lane changing are in lesser extend possible than in free flow conditions. In free flow conditions the approaching traffic flow has more space to perform the different extra driving maneuvers around acceleration lanes. To cover as much as possible different changes in the traffic flow characteristics free flow conditions are selected, where more different extra



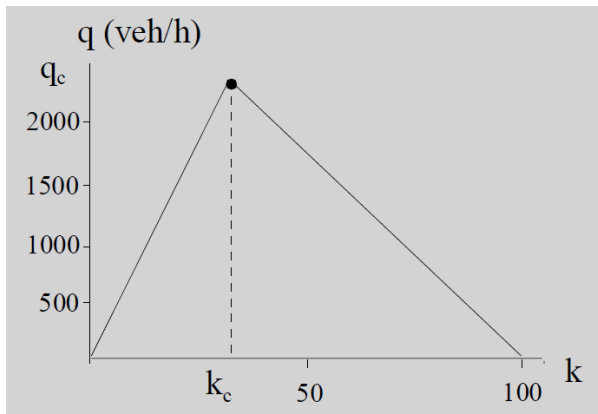


Figure 3.1: Daganzo's fundamental diagram [29]

driving maneuvers are present.

Within the free flow conditions still differences are found; the traffic flow reacts different on the acceleration lane with a low density, than with a high density, and the composition of the traffic flow has influence on the traffic flow characteristics. In the data processing the density is divided in 5 km/h regions, also called density bins. The comparison on the segments is performed in a single density bin, since the effect of density on the turbulence is not the subject of this thesis. For the comparison the most interesting density bin is selected in which is expected that the turbulence is the highest, with this approach the changes in the traffic flow are the most visible.

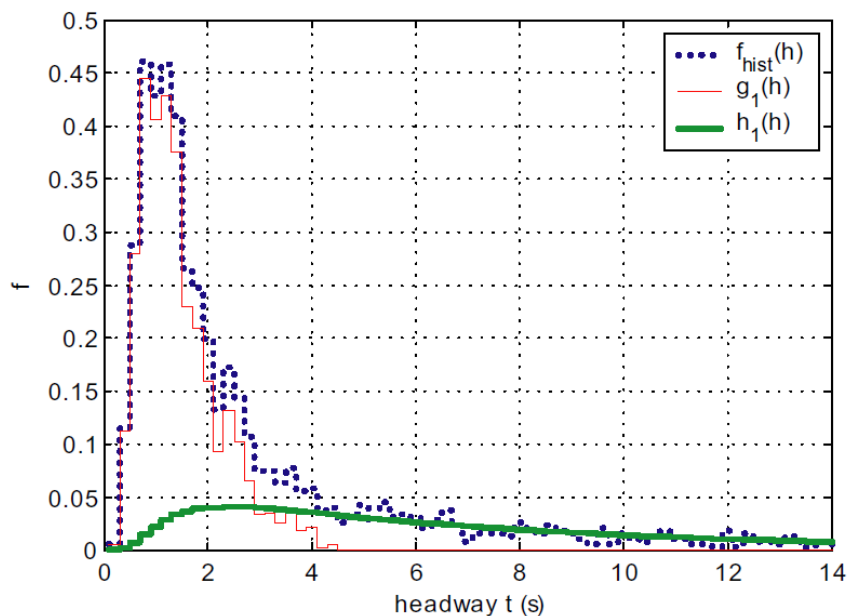


Figure 3.2: Example of distribution free headway estimation for site "Doenkade" [30]

Hoogendoorn [30] on the site "Doenkade" in the headway distribution that around a headway of 3 seconds or lower the frequency of the amount of constrained (or following) headway becomes larger than the amount of unconstrained (or free) headways, see figure 3.2. Therefore, the expectation is that close to an average headway of 3 seconds in the traffic flow the drivers are required to interact or anticipated on an expected decrease of the headway, due to entering drivers from the acceleration lane. In this traffic condition is expected that the interaction and anticipating effects in the traffic flow is the highest and thus also the turbulence and changes traffic flow characteristics. On the outside lane

direct interaction is arising by the entering traffic; by making the interaction the highest on the outside lane drivers on the other lanes are influenced by this interaction, due to like cooperative lane changes, which spread the influence further around the acceleration lane. In the data set, the average headway of 3 seconds is found on the outside lane in the density bin 30-35 veh/km. Appendix ?? shows the effects of the selection of another density bin on the changes in the traffic flow characteristics.

The composition of the traffic flow is expected to have a minor influence on the traffic flow characteristics. The composition of the traffic flow consist of the composition of drivers and vehicles and their capabilities. In section 3.4 data of a longer period is combined to get the average changes in the traffic flow due to the acceleration lanes. To do this well the average traffic flow characteristics of the basic freeway segment and the acceleration lane are determined, and assumed is that because of data over long period the traffic compositions of both segments become comparable.

### 3.3. Data collection method

For the measurement of the turbulence distance data is needed for the analyses. The data for the analyses need to be collected in the traffic flow. Data collecting in traffic flows can be done with multiple methods; these data collection methods have different data types, advantages and disadvantages. Of the possible data collection methods one data collection methods is selected that is the most promising for the measurement of the turbulence distances.

#### 3.3.1. Suitable data collection methods

In literature four data collection methods are found; the data collection methods are imagery, driving simulators, instrumented vehicles and induction loops.

- *Imagery* is recorded traffic flow by a video camera. Recording can be performed by a high positioned camera on buildings and telescopes and on top of the freeway by helicopters and drones equipped with cameras [31, 32]. Voorrips [33], Loot [15] and Kolen [13] used this data collection method in their thesis. Loot and Kolen investigated headways, speeds and lane changes during the acceleration lane on the A12 near Bodegraven; this data set is available for investigation. Imagery data shows a detailed and complete overview of the traffic flow characteristics over time and on the recorded segments. Not only the microscopic traffic flow characteristics can be investigated, with the recorded trajectories also the macroscopic traffic flow characteristics can be investigated. The available data set consists only of data during the acceleration lane and to measure the turbulence distance especially the upstream and downstream region of the acceleration lane are important, and thus the available data set can not be used fore measuring the turbulence distance, and a new data set is required. Further, collecting a new data set with imagery as data collection method is relative expensive.
- *Driving simulators* are mock-ups of vehicles surrounded by screens on which simulated driving environments are displayed. Drivers control their vehicle through vehicle actuators and can therefore drive through the virtual environment [34]. Driving simulators have a high degree of controllability. However, data from only one driver is collected with this method. This aspect increases the difficulty to investigate the changes in turbulence around acceleration lanes, since turbulence consists of multiple driving maneuvers, and these need to be performed by the driver to be measured. Therefore a lot of drivers are required to perform the simulation to find the effects of turbulence completely.
- *Instrumented vehicles* permits quantitative assessments of driver performance in the field, under actual road conditions [35]. Instrumented vehicles are comparable with driving simulators, only driving simulators simulates the environment and by instrumented vehicles actual road conditions are available. Instrumented vehicles have the same disadvantage as driving simulators and lacks the high degree of controllability as driving simulators.
- *Induction loops* are loops which can be found in the asphalt on the freeways. Induction loops are logging the speed and time of passing vehicles [36]. The data of the induction loops can be collected on two levels, one is the detailed data of the induction loops and the other is the mean data of the induction loops. The detailed data consist of the logs of every passing vehicle with

their speed; this data is collected by connecting a computer on site with the induction loop. At the moment a data set is available of the A10 on the West part of the ring of Amsterdam. The mean data of the induction loops is easier available; this data is automatically uploaded to a server, and therefore, this data is available of the whole Dutch freeway network. The mean data of the induction loops consist of the macroscopic traffic flow characteristics flow and time mean speed on different time periods, like 1, 5 or 15 minutes. The density is not measured with induction loops, however there are examples where the density is calculated with the flow and time mean speed data [14, 37]. However, induction loops show not a complete overview of the investigated segment; induction loops measure on fixed locations and induction loops, and induction loops have a spacing of 500 meter on average. This results in that on a single location only data is available of the changes in the traffic flow characteristics over 500 meter. A possibility is to fill these spacings with data of other locations to investigate the changes over short distances than 500 meter; However, this approach is not yet used, which makes investigation with this approach an innovation.

### 3.3.2. Selecting data collection method

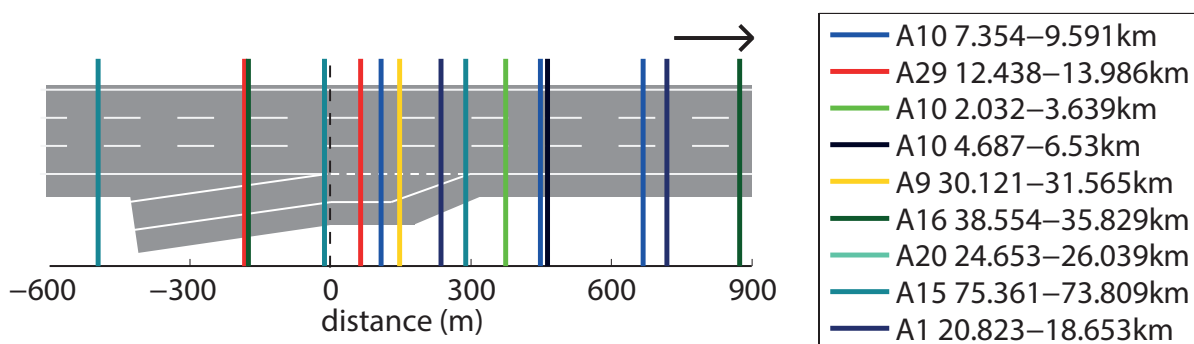


Figure 3.3: Positions of the selected induction loops around acceleration lane.

In the previous subsection four data collection methods are described to collect data to measure the changes in the traffic flow characteristics. The data collection with induction loops with macroscopic data is the one of most promising method to investigate the turbulence distance. Induction loops have the large advantage that the data is directly available, and therefore new data not need to be collected in the field. Further, induction loops have an interesting element, an innovative approach is suggested to reduce the distance accuracy of the changes in traffic flow characteristics. Suggested is to combine induction loops of multiple locations to measure the changes with shorter distances than 500 meter. Figure 3.3 shows how this combining is defined. Induction loops of other similar segments have different positions compared to the gore of the acceleration lane, and the induction loops show the traffic flow characteristics of these positions; with this data the changes on shorter distances become accessible. Therefore, induction loops with macroscopic data is selected as data collection method for the analyses. Based on the selection of induction loops with macroscopic data, the three identified macroscopic traffic flow characteristics are investigated in the analyses, the three macroscopic traffic flow characteristics are mean speed, lane flow distribution and density.

### 3.3.3. Induction loop configuration

Induction loops with mean data is selected as data collection method for the analysis. To investigate the changes in traffic flow characteristics locations more detailed than with spacings of 500 meter locations are combined. In section 3.2 two situations are defined, the basic freeway segment and the acceleration lane. In both situations induction loops are required to investigate the changes, and the data of the induction loops need to be subject to the related segment type. Around acceleration lanes the induction loops need to be influenced by the presence of the acceleration lane and not by other discontinuities to find the changes in the traffic flow characteristics that are related to the acceleration lane. On basic freeway segments only the normal values of traffic flow characteristics need to be found without influence of discontinuities. Therefore, a configuration is defined to select induction loops on both type of segments to find the relevant induction loops on the selected locations.

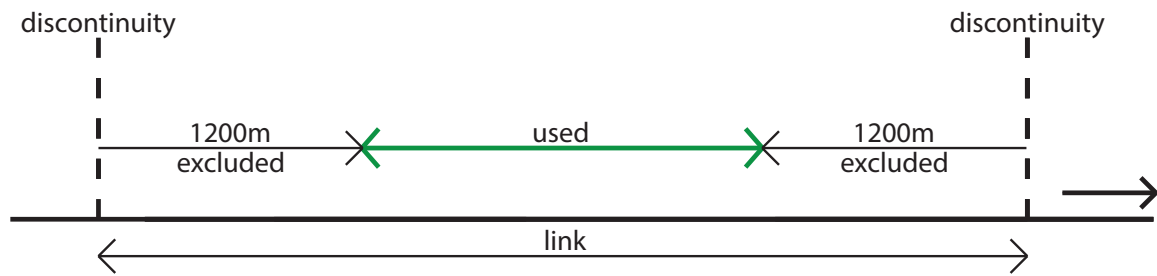


Figure 3.4: Selection criteria of the positioning of the induction loops on basic freeway segments

On basic freeway segments the influence of the upstream and downstream discontinuity need to be not present in the data of the induction loops. To do this, an influence length is defined at the start and end of suitable basic freeway segments. Induction loops positioned within the start and end of the basic freeway segment are considered as affected by the upstream and downstream discontinuity; induction loops between the start and end of the basic freeway segment are not subject to the upstream and downstream discontinuity. In chapter 4 the influence length is defined as 1.200 meter. Figure 3.4 shows the selection configuration of induction loops on basic freeway segments.

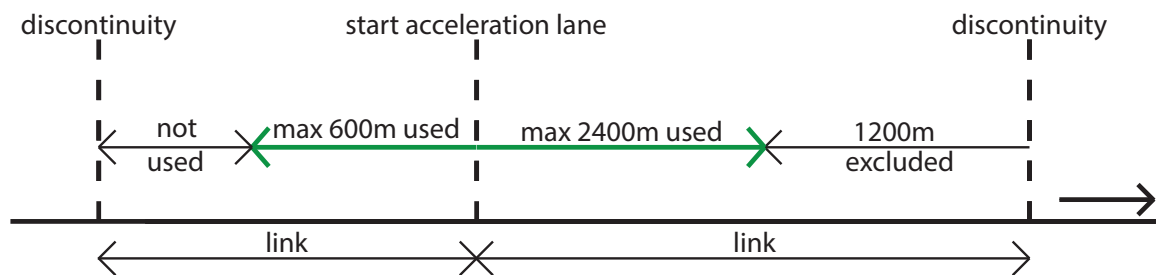


Figure 3.5: Selection criteria of the positioning of the induction loops on acceleration lanes

Around acceleration lanes induction loops need to be also not be influenced by discontinuities, except the acceleration lane. Therefore, for acceleration lanes also a configuration is defined to select suitable induction loops, as presented in figure 3.5. On the upstream region of the acceleration lane a simplified selection procedure is used to select a reasonable amount of detectors, therefore induction loops within 600 meter upstream of the acceleration lane are selected for the analyses. On the downstream region the same approach is used for the downstream discontinuity as for basic freeway segments. The upstream discontinuity is here the acceleration lane and that is case that is investigated and thus induction loops close to the acceleration lane are suitable; to limit the distance induction loops within 2.400 meter of the gore are selected, at this point is expected that the influence of the acceleration lane is not present anymore.

### 3.4. Suitable analyses

In section 3.1 the two steps of the analyses approach are defined. First the changes in traffic flow characteristics around acceleration lanes is investigated and based on the changes the turbulence distance is determined. The two analysis steps are found in the analyses.

#### 3.4.1. Analyzing changes

As described first the changes in traffic flow characteristics are investigated; with the basic freeway segment as reference and acceleration lane as case. The main objective is to identify the changes of the traffic flow characteristics to define the region where these changes are applicable and how the change behave within this region, thus for acceleration lanes is the distance an important factor for investigation. The information on the changes within the region and the possible region are used to determine the turbulence distance.

The analysis of the changes is based on a case-control study, where the default traffic flow char-

acteristics of the basic freeway segment is compared with the distance depended traffic flow characteristics of the acceleration lane. Thus in the analysis the default characteristics of the basic freeway segment and the distance depended traffic flow characteristics need to be measured.

The measurement of the traffic flow characteristics on basic freeway segment is performed by not only investigation a single basic freeway segment, when a single basic freeway segment would be selected to measure the default turbulence the average traffic flow composition of this selected affects the measurements. As described in section 3.2 has not only the density influence on the turbulence in the traffic flow, but also the traffic flow composition, like the share of freight traffic in the traffic flow. These composition are in larger extend related to the investigated locations. Therefore, the analyses are more reliable when data of multiple locations are combined to define the average default traffic flow characteristics.

The data of different locations are seen as subpopulations with their own characteristics of the total data population of the average basic freeway segment. It is likely that in the data processing not all possible data subpopulations are selected for the data analysis, and thus is assumed that there are missing subpopulations. Therefore, the data population of the average basic freeway segment need to be estimated. Verhaeghe citecourse4831 suggested two popular methods to cope with the missing subpopulations. These methods are the maximum likelihood method and the method of moments. He found no reason to choose on method above the other method. However, the data of the induction loops is assumed to be normal distributed; therefore is assumed that the data of the average basic freeway segment is also normally distributed. The parameters of a normal distribution are the mean and standard deviation. Further, in the used software this method is already available and this errors made with implementing another method is reduced. Figure 3.6 shows an example of the application of the maximum likelihood the estimate the lane flow distribution on the median lane. Further the performance of the maximum likelihood on the selected data is shown in appendix B.

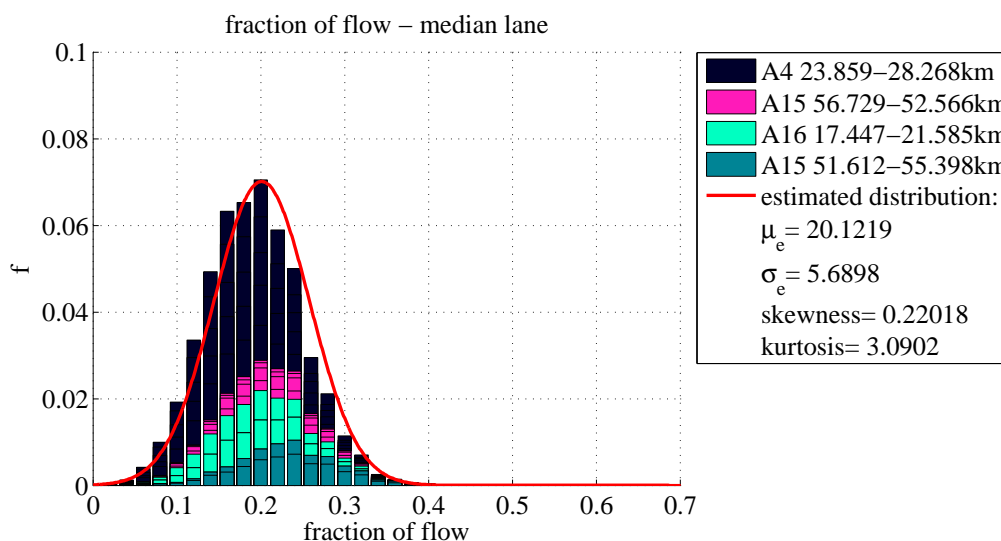


Figure 3.6: Example of the estimation traffic flow characteristic lane flow distribution with the maximum likelihood on the median lane

With the maximum likelihood the average default values of the traffic flow characteristics are estimated. With this method the means and standard deviations of the lane mean speeds, lane flow distributions and lane densities are estimated. These values are used for the comparison with the acceleration lane to identify the changes and determine the turbulence distances.

Now the values of the basic freeway segment can be determined, the traffic flow characteristics around acceleration lanes need to be estimated. In comparison with the basic freeway segment distance is an important factor in the analysis and therefore the data of the induction loops can not

be merged of the whole segment; however, different acceleration lane locations have slightly different traffic compositions and thus averaging is recommended. To fulfill this recommendation induction loops around acceleration lanes are merged within intervals on different distances of the gore. With this approach the averaged traffic flow characteristics are estimated, while still the changes over distance can be investigated. For the averaging process the same method is used as for the basic freeway segment.

To investigate the changes over distance with a higher accuracy the intervals need to be as small as possible, while the number of induction loops is relevant. When only a single induction loop is available in an interval the maximum likelihood only estimates the traffic flow characteristics of the single induction loops, when the number of induction loops is higher than one, the estimation reliability of estimating an average acceleration lane increases. This is important for the selection of the interval length. In this thesis an interval length of 300 meter is selected to have sufficient induction loops in the intervals.

Now the values of the basic freeway segments and the intervals of the acceleration lanes can be determined a comparison is made between these two cases. An example of the comparison is showed in figure 3.7. Compared are the means of the three defined traffic flow characteristics: lane and carriage-way mean speeds, lane flow distribution and lane density. Further the change in turbulence around acceleration lanes can have an effect on the difference between drivers, this can be measured by investigating the standard deviation. Based on the comparisons the changes in traffic flow characteristics is estimated; these are used to determine the turbulence distance.

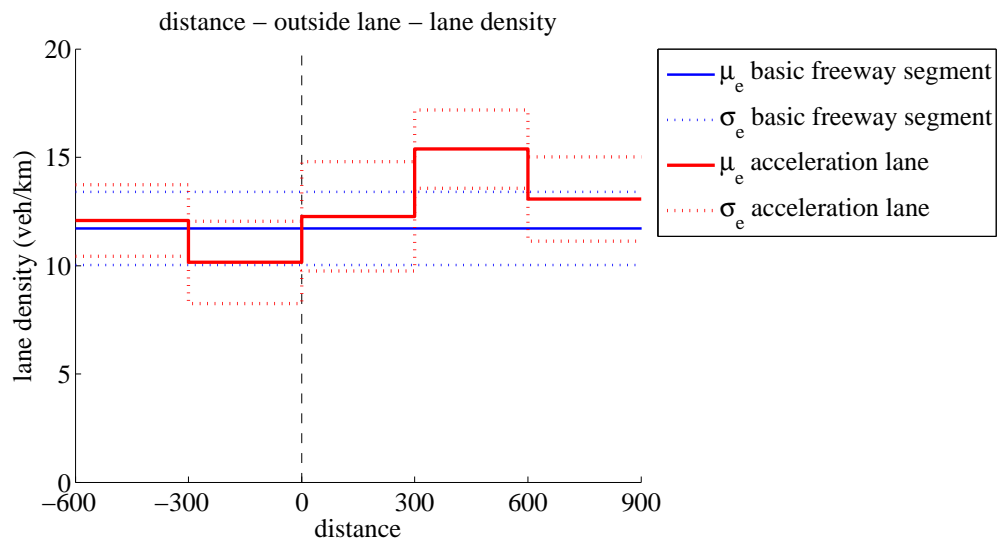


Figure 3.7: Example of the comparison of the density on the outside lane of the estimated mean of the basic freeway segment with the estimated distributions around the acceleration lane

The last step of this analysis is to investigate if the differences between the estimated distribution of the basic freeway segment and the estimated distributions around the acceleration lane can be related to the presence of the acceleration lane. Therefore a significance test is performed on the results. This test is needed to investigate if the difference is based on coincidence or by the presence of the acceleration lane (all other factors are assumed constant). The selected test in this research is the Kolmogorov-Smirnov test (KS-test) [38]. The KS-test will test if the estimated distributions of the acceleration lane are different from the raw data of the basic freeway segment. The choice to test on the raw data of the basic freeway segment is to reduce error, to test on the estimated distribution of the basic freeway segment possible error are appearing in the data set.

In chapter 2 expectations are defined for the changes in the traffic flow characteristics. These expectations are tested with the defined analyses setup above. When the difference is significant and

follows the same changes as expected in the expectations the expectation is confirmed.

### 3.4.2. Analyzing turbulence distance

The second step of the analysis approach is to estimate the start and end points of the changes turbulence and thus changed traffic flow characteristics. In the previous step the changes of the traffic flow characteristics are estimated. Based on the changes the behavior of the traffic flow characteristics can be determined by defining a relation between the distance in comparison with the gore and the value of the traffic flow characteristics. The relation can be used to define the start and end points of the changed turbulence where the turbulence is normal again as a basic freeway segment.

The first part of this analysis consists of a relation is seen in the intervals. To reduce the errors also the means of the single selected induction loops are investigated. The changes in the traffic flow characteristics could be also visible in the individual induction loops, and the individual induction loops lacks the measurement error made by introducing the intervals and maximum likelihood estimation. On the individual means or standard deviation of the induction loops the expected relation is fitted with a line.

When the relation is estimated of the traffic flow characteristics with a line fitting the start and end points are defined when further downstream and upstream the traffic flow characteristics are comparable with the basic freeway segment. Normally a smooth transition is expected between the turbulence behavior of the basic freeway segment to the changed turbulence behavior around acceleration lanes. In the result appears that no data is available in this research to investigate this, and therefore the cross-section of the fitted line on the means of the induction loops of the acceleration lane with the average estimated values in the previous analysis step of the basic freeway segment. These cross-sections are seen as the start and end points of the changed turbulence.

Figure 3.8 shows an example of line fitting and estimation of the start and end point of the changed lane density. To see how well the line fits on the means of the induction loops the coefficient of determination, the  $R^2$  of the estimation is calculated. This coefficient indicates how well the data fits on the estimated line. The closer this coefficient is to 1 the better is the estimation assumed of the line.

In chapter 2 a turbulence distance of 130 meter upstream and 600 meter downstream is expected as result. When the estimated turbulence is close to this value, where close is defined as the half of the used interval length of 150 meter, the expectation is confirmed.

## 3.5. Conclusions on methodology

To measure the turbulence distances of acceleration lanes two steps are performed, first the changes in the macroscopic traffic flow characteristics are investigated by comparing the traffic flow characteristics around acceleration lanes with the reference basic freeway segment; based on these changes in the macroscopic traffic flow characteristics the turbulence distance are determined.

For the measurement of the macroscopic traffic flow characteristics induction loops are used, since this gives to possibility to investigate an innovative method to determine traffic flow changes over distance; since induction loops have an average spacing of 500 meter and a new approach is applied to improve the accuracy of the determination of the turbulence. In the new approach induction loops from multiple locations are combined to fill these spacings with induction loops from other location, and so receive data of the traffic flow characteristics within the 500 meter spacings. This approach combined with a comparison of the acceleration lane and the basic freeway segment is applied to measure the changes in traffic flow characteristics and the turbulence distance.

With the defined data collection approach the analyses are defined in more detail. For the analyses of the changes in the traffic flow distance intervals around acceleration lanes are investigated of the mean and standard deviation to get the average behavior of the traffic flow characteristics, and to reduce the local effects. When the changes are known, these changes are used to find shapes to fit



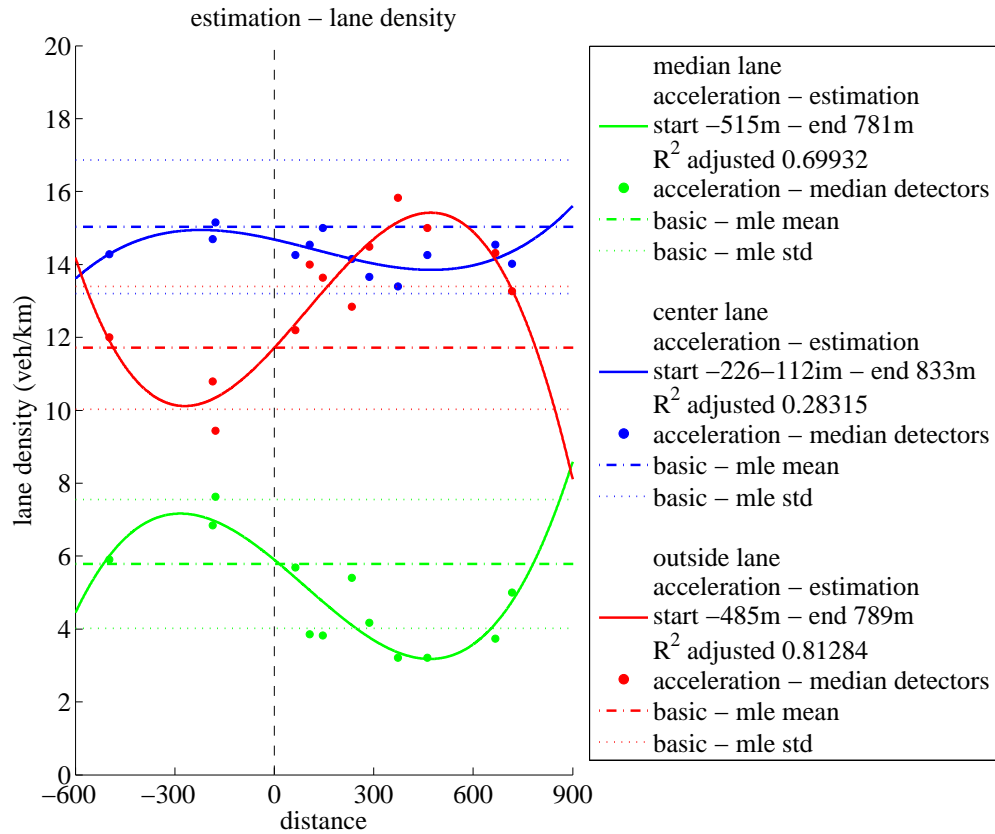


Figure 3.8: Example of the lane fitting on the individual induction loops on changes of the lane density on the different lanes

lines of the means and so estimate the start and end point of the changed turbulence. The distance between the start and end point and the gore of the acceleration lane are the turbulence distances of the acceleration lane.



# 4

## Data and data processing

In the previous chapter the methodology of this thesis is defined. Also in the previous chapter the case-control study and the applied analyses are explained. In this chapter the needed data processing steps and the resulting data is described. This resulting data set is used for the analyses in the next chapter. This chapter consist of a description of the data source, the data processing steps and the resulting locations and induction loops. The data processing steps consist of the selection of basic freeway segments and acceleration lanes and their relevant induction loops. Further errors are filtered from the resulting data set of the induction loops.

### 4.1. Dante

An important step in the data collection is to define the used data input program. In this thesis the program Dante is used as input. Dante is the back software of Fileradar BV which is created by two former PhD students to develop traffic predictions for the freeway network in the Netherlands. To do this they collected all the historic data, developed a model and calibrated this model. During this development of Fileradar Dante is developed as program to deal with large amounts of geographic dynamic data to perform complex analysis tasks. Dante consists of monthly changing traffic networks of the past 10 years, KNMI rain data, one-minute speeds and flows of the induction loops in the network of the past eight years and zip code area combined with social demographic data. With Dante it is possible to i.a. combine induction loop data with road data, correct errors in the data, combine several data inputs and presenting traffic predictions [39].

In this thesis Dante is used in combination with MATLAB (a high-level programming language and interactive environment for numerical computation, visualization and programming [40]) as input for the freeway network characteristics and input for the one-minute induction loop data. In Dante the total freeway network is selected with the used days of analyses (see section 4.5) and exported. The exported files are loaded into MATLAB to analyze the data and selection of the used locations and induction loops. This part is explained in more detailed in the subsequent sections.

### 4.2. Data processing method

As described in the previous section Dante is used as input. Now the steps need to be defined to select the interesting locations and induction loops and to filter the data of the induction loops to create the data set for the analyses in 5. Figure 4.1 shows the steps in the data processing for this creation. The data processing consists of five steps between the export of Dante and the comparison of the segments, which is the analysis. As is shown in the figure the processing steps are: filtering locations, filtering induction loops, filtering data, interval selection and the estimation of the parameters. These steps are performed in MATLAB and are described in appendix E.



Figure 4.1: Scheme steps data process

The data processing can be distinguished in two main steps: the induction loop selection and the data filtering. Before the induction loop selection the freeway network of the Netherlands is available. This available network consists of links, nodes, lines, carriageway detectors and lane detectors and is a digital representation of the actual Dutch freeway network. However, this network is not one-on-one representing the actual network. In this network errors are available, like a wrong number of lanes and missing lane drops. These errors need to be taken into account before the analyses are performed. From this network first basic freeway segments and acceleration lane locations are selected which meet the criteria of section 3.2. On these locations the interesting induction loops are selected which meets the position requirements in section 3.2 and induction loops with empty data sets are also filtered out of the induction loops set.

The second part consists of filtering of the data of the induction loops. Induction loops consist of one-minute data of the whole day (1.440 minutes) of the flow and (time) mean speed. These data set consists of raw data; these sets have one-minute data points with and without errors, and of data of outside the investigated traffic condition of 30-35 vehicles per hour. In the next sections these steps are described in more detail.

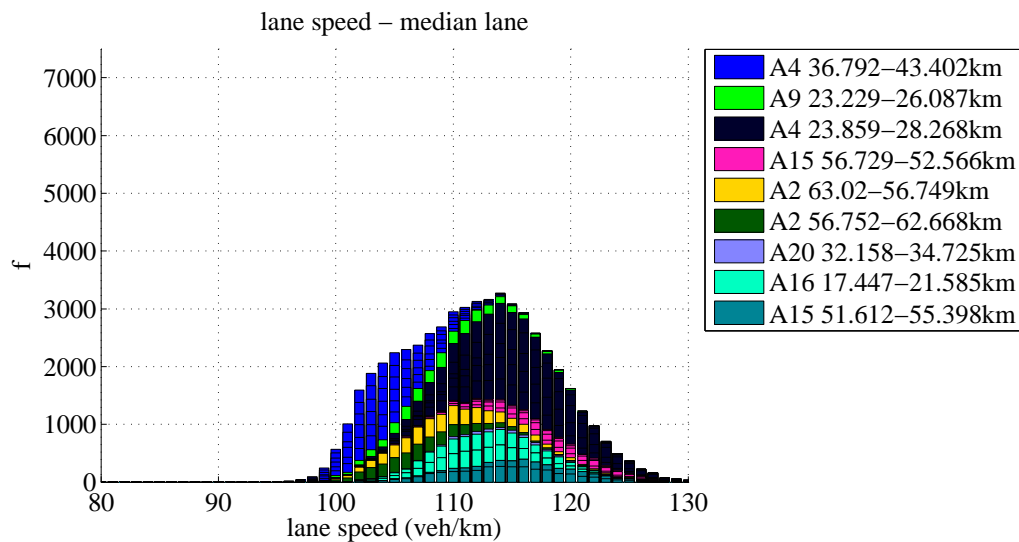
### 4.3. Location selection

The first step is to select the interesting basic freeway segments and acceleration lane locations. To select these locations are in the previous chapter selection criteria developed. The applied selection criteria are shown in table 4.1. Most of these criteria are automatically found in the network of Dante. Only the fixed average speed check are in a later stage of the data processing manual excluded. This characteristic is not available in the network of Dante. In the selection process 15.603 potential basic freeway segments and 1.499 acceleration lane locations were found. After the selection process without the manual exclusion of locations with fixed average speed check 40 acceleration locations and 337 basic freeway segments were selected. Due to exclusion of fixed average speed checks, filtering of locations without induction loops or selected induction loops and filtering of possible false locations only 8 acceleration lane locations and 4 basic freeway segments are selected for the analysis.

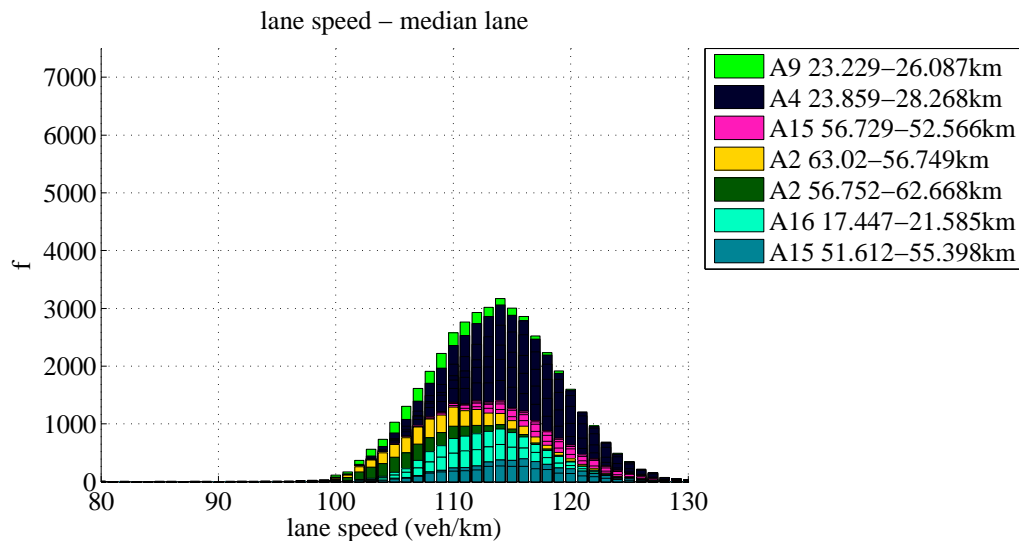
Table 4.1: Criteria locations

criteria
3-lanes
1-lane acceleration lane
100 km/h regime
no rush hour lane
no overtaking restrictions
no fixed average speed check
length acceleration lane 217.5-362.5 meter

Locations with fixed average speed checks (in the Netherlands know as "trajectcontrole") are excluded from the set of locations. The presence of fixed average speed check has a large influence on the data set. Figure 4.2 shows the effect of including locations with fixed average speed checks on the data set of basic freeway segments on the median lane. On this lane the effect of the presence of the fixed average speed check is the most present. On this lane the lane speed data of location A4 36.792-43.402 is located on the left part of the histogram. This results in that the average of the lane speeds is lower than in a data set without fixed average speed check locations. Therefore the decision is made to exclude locations with fixed average speed checks from both the location sets of basic freeway segments and acceleration lanes. More results of the inclusion of fixed average speed check locations are available in appendix A.



(a) Median lane with fixed average speed check



(b) Median lane without fixed average speed check

Figure 4.2: Lane speeds comparison median lane

## 4.4. Induction loops selection

The selection criteria for the position of the detector relative to the gore of the acceleration lane or on the basic freeway are defined in section 3.2. In this section is described that an influence length of 1.200 meter is applied to select not influenced induction loops on basic freeway segments and around acceleration lanes. This length is determined based on a selection of three possible length, 300, 600 and 900 meter, on basic freeway segments. In figures 4.3, 4.4 and 4.5 the effects of the different influence length on the number of detectors and locations is shown based on the lane flow distribution on the outside lane. In these figure is visible that by a small influence length the difference between locations is larger and reduces when the influence length increases. On the largest influence length the difference between the locations with multiple induction loops is small; the locations with single induction loops are identified as outliers. The A9 location is not well presented in Dante, and this leads that on this link in Dante a lane drop is present. A lane drop is a discontinuity, and a lane drop could effect the traffic flow characteristics. This means that the available induction loop on the A9 in the figure is considered as affected, and thus excluded from the set. The other two locations are both on the A2 on the West ring of Utrecht. The presence of these two locations is strange, because both locations are not present in the set of locations by a smaller influence length. While the only difference

between the plots is the smaller influence length. Although, the values of the locations are close to the other locations, the values are not trusted and excluded from the selection set. Based on this analysis a consideration is made between more locations with possible more influence of upstream and downstream discontinuities and less locations with less influence of upstream and downstream locations. In this case less locations with less influence of upstream and downstream locations is chosen to improve the reliability of the results.

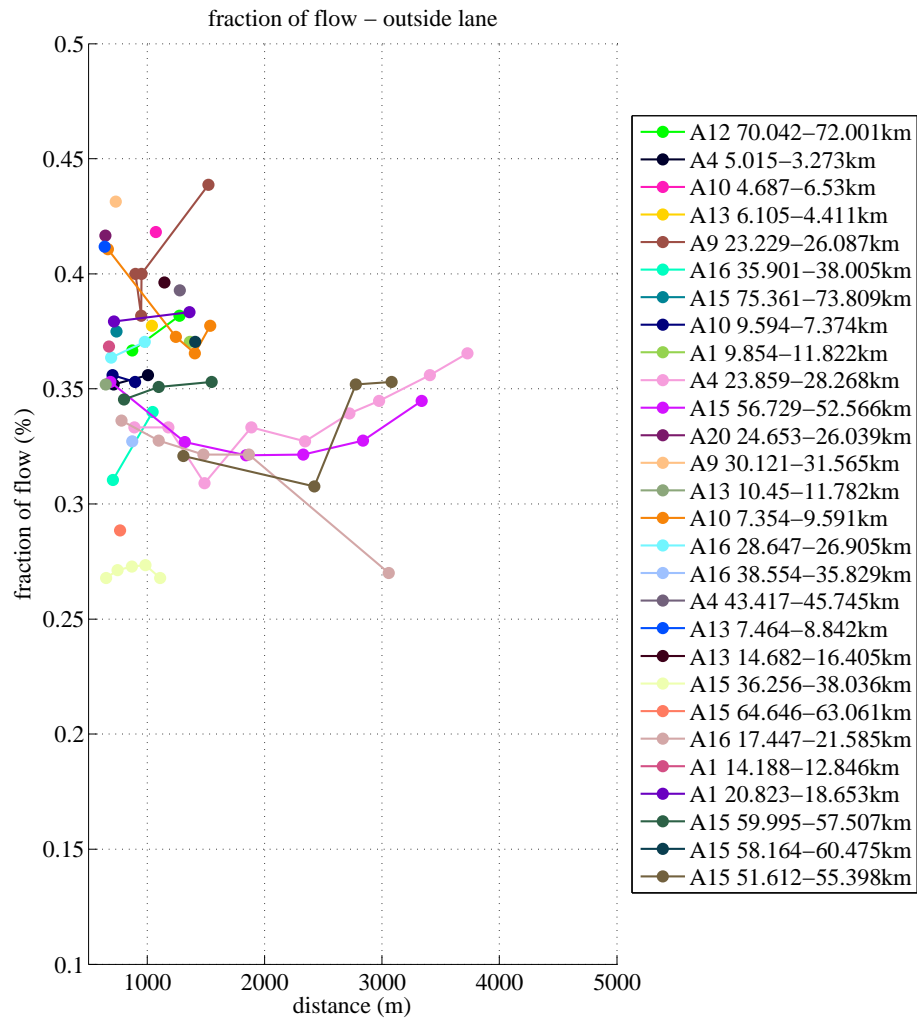


Figure 4.3: Lane flow distributions per location outside lane 600 meter influence length criteria

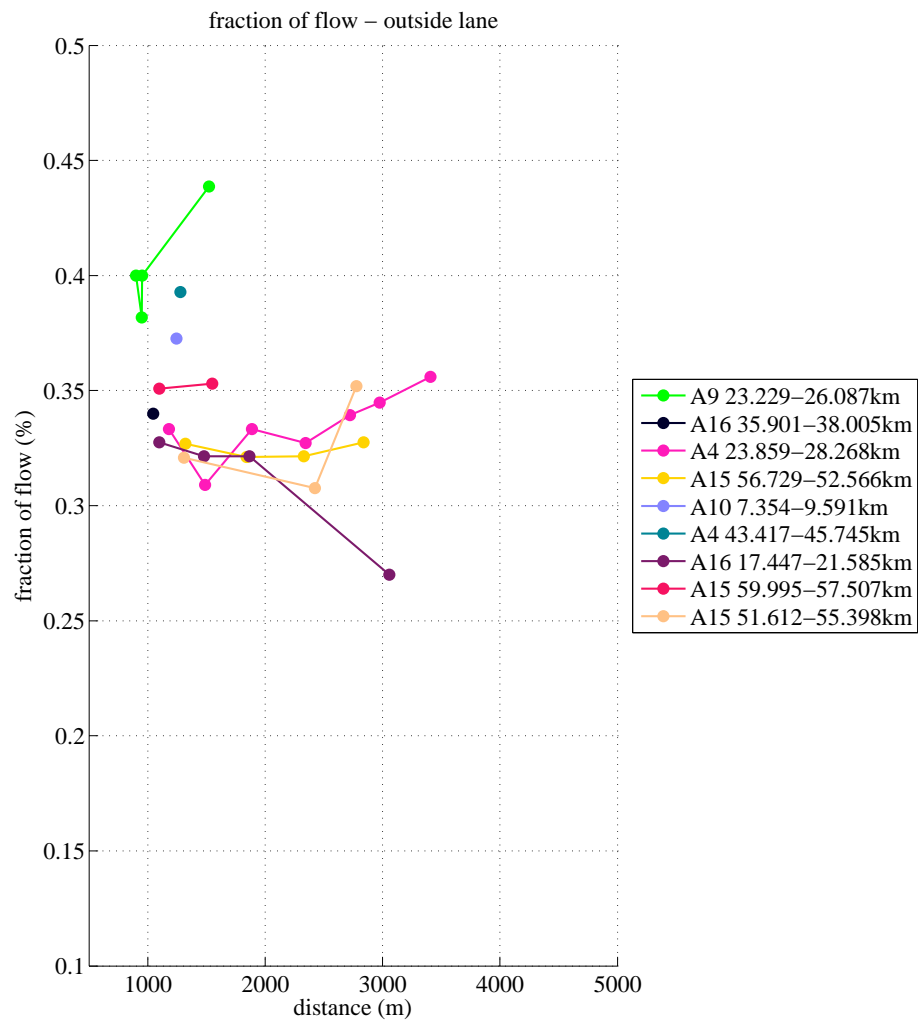


Figure 4.4: Lane flow distributions per location outside lane 900 meter influence length criteria

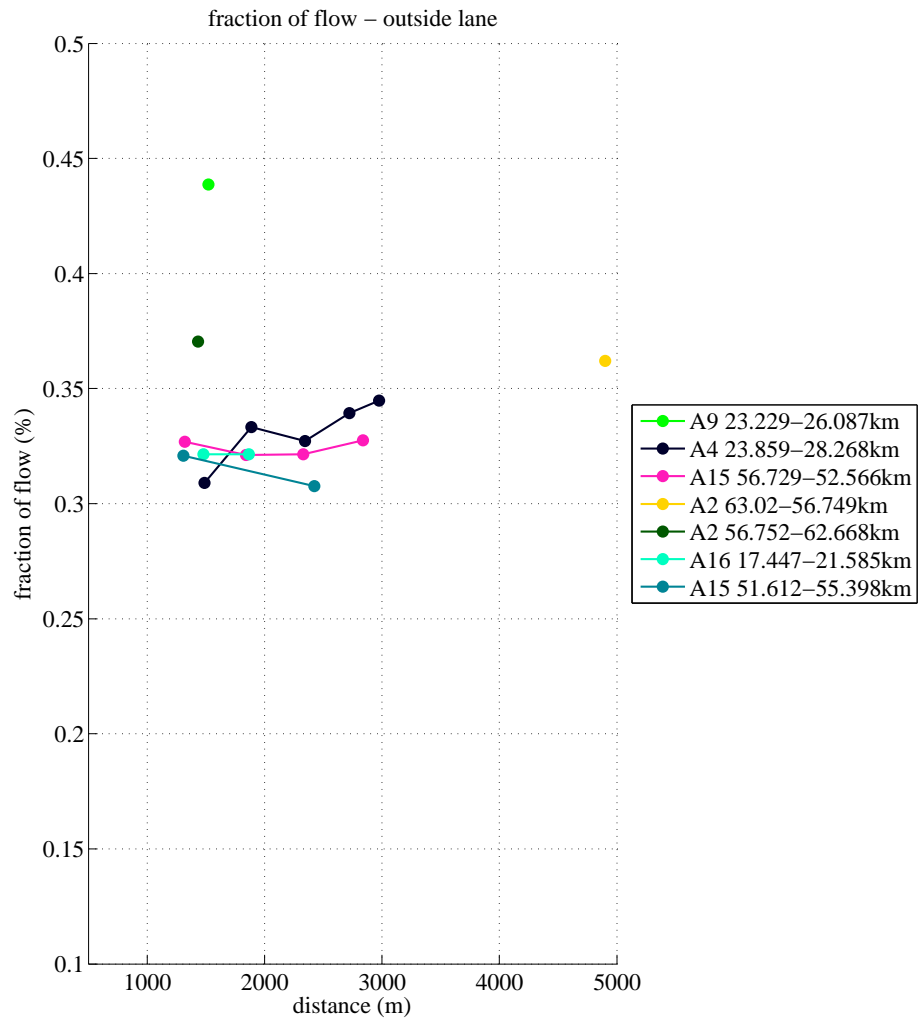


Figure 4.5: Lane flow distributions per location outside lane 1200 meter influence length criteria

## 4.5. Filtering induction loop data

The last filtering is on the data from the selected induction loops. The data of the induction loops can include errors and faults are made in the output of the data. In the data process for the analysis input days, 130 km/h periods and false data are discussed.

The selection of days is divided in three aspects: period of year, the weather, day characteristics. These aspects have influence on the traffic flow characteristics, especially in the vehicle and driver composition and the vehicle and driver capabilities. The influence of weather is already mentioned in the previous chapter.

In this thesis April 2013 and 2014 are selected as input months, both months are nor winter or summer months are changes in the traffic flow characteristics are expected. In the summer more inexperienced driver are presence in the traffic flow due to vacations and people with a day off and in the winter traffic flow conditions are changed due to snow and icing which results in more careful drivers. Both selected months have quite dry and is quite normal of temperature; figure 4.6 shows the characteristics of both months.

The next aspect is the day characteristics. Different days have different driver compositions in the traffic. Major distinctions are made between working and weekend days and the presence of vacation periods [41, 42]. On working days expected are more experienced driver in the traffic flow, which affects their driver capabilities. In example, they accept more often smaller headways than inexperienced drivers. Further, on vacation days the composition is also changing; on these day also more inexperience drivers are expected in the traffic flow, and the vehicle composition is expected to change due to like more car-caravan combinations in the traffic flow. Vacation periods start on Mondays, often people start to go on vacation on the Friday before to extend their vacation as much as possible. Therefore, Fridays before vacation periods are also excluded from the data set. Table 4.2 shows the selected days in April 2013 and 2014.

The last factor is the weather on the exclusion of day from the set. Extreme weather will have an influence on traffic [43]. An example is that drivers are driving slower in heavy rain. Therefore days with a large amount of measured rain are excluded from the set, based on figure 4.6. In table 4.2 the selected days are shown.

Table 4.2: Selection of days as input

Day	April 2013	April 2014
1	Easter	selected
2	selected	selected
3	selected	selected
4	selected	selected
5	selected	weekend
6	weekend	weekend
7	weekend	selected
8	selected	selected
9	selected	selected
10	selected	selected
11	selected	selected
12	selected	weekend
13	weekend	weekend
14	weekend	selected
15	selected	selected
16	selected	selected
17	selected	extreme weather
18	selected	Friday before
19	selected	weekend
20	weekend	weekend
21	weekend	Easter
22	selected	selected
23	selected	selected
24	selected	extreme weather
25	selected	Friday before
26	Friday before	weekend
27	weekend	weekend
28	weekend	May vacation
29	May vacation	May vacation
30	May vacation	May vacation

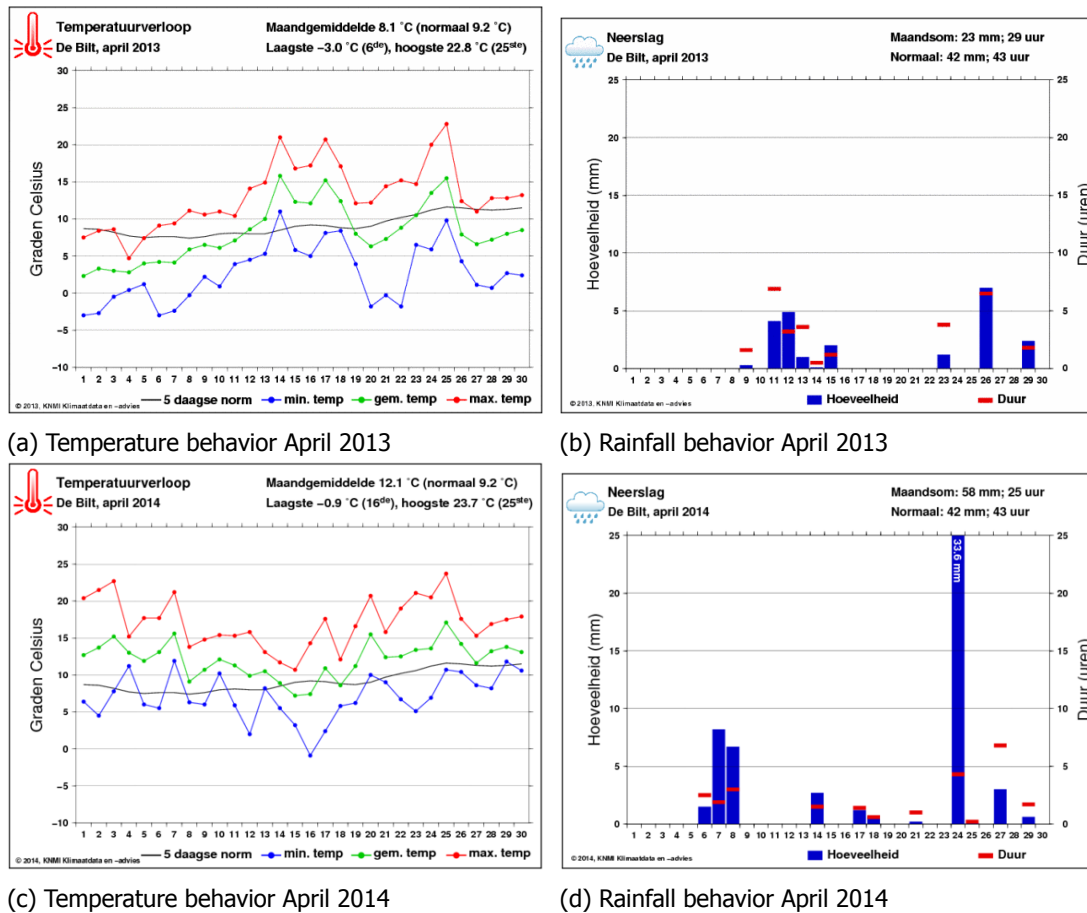


Figure 4.6: Weather behavior April 2013 and 2014 [44, 45]

The next step is to analyze the output data of the induction loops. Based on these analyses false and unexpected results are filtered out. Distribution are drawn based on lane speed, lane flow distributions and lane densities over the carriageway densities. An example of this type of plot is shown in figure 4.7.

In this figure the effect of the introduction of the dynamic speed limits on the Dutch freeway network is found in the data. On some selection locations before 6:00 and after 19:00 the speed limit changes from 100 km/h to 130 km/h. This increase is visible in the plot because these are often the periods that the carriageway density is lower than during the working hours. Therefore is decided to exclude one-minute data before 6:00 and after 19:00 from the data set to exclude effects of the dynamic speed limit from the data set.



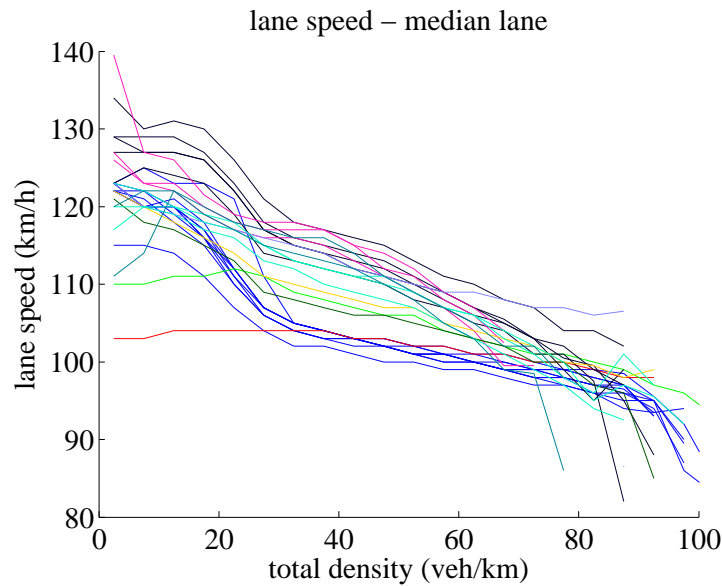


Figure 4.7: Speeds over densities on the median lane with 130 km/h data on basic freeway segments

Other lane speed filters are speeds lower than 80 km/h and higher than 220 km/h. Based on the scope of free flow data to prevent effects on the data due to downstream bottlenecks speeds lower than 80 km/h are excluded from the data set. Speeds higher than 220 km/h are seen as error-ed data and therefore excluded.

The last filters are applied to filter other unexpected data out of the data set. To do this two types of figures are made: line plots to plot the distribution of lane speeds, densities and lane flow distributions over the carriageway density and histograms of the data to find non-normal distributions of the data of the induction loops. A non-normal distribution of a induction loop is seen as unexpected and can be the result of i.g. construction works with a lower speed limit on this location. Figure 4.8 shows an example of the line plot filtering and more is available in appendix C and in figure 4.9 shows an example of a non-linear distribution of the lane speeds on the center and median lane on a selected A10 location.

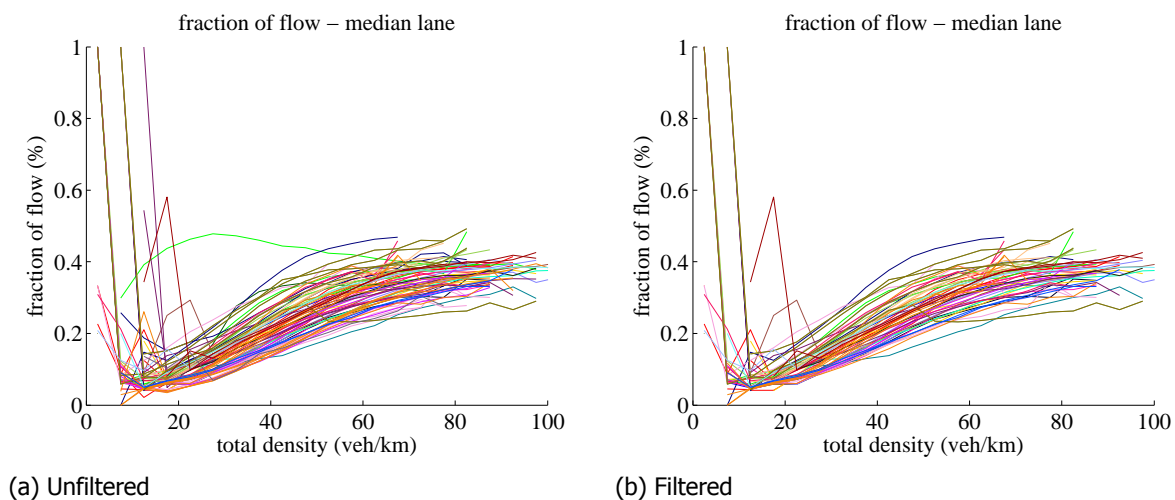


Figure 4.8: Example of unexpected output of induction loops filtering, the colors indicate different locations

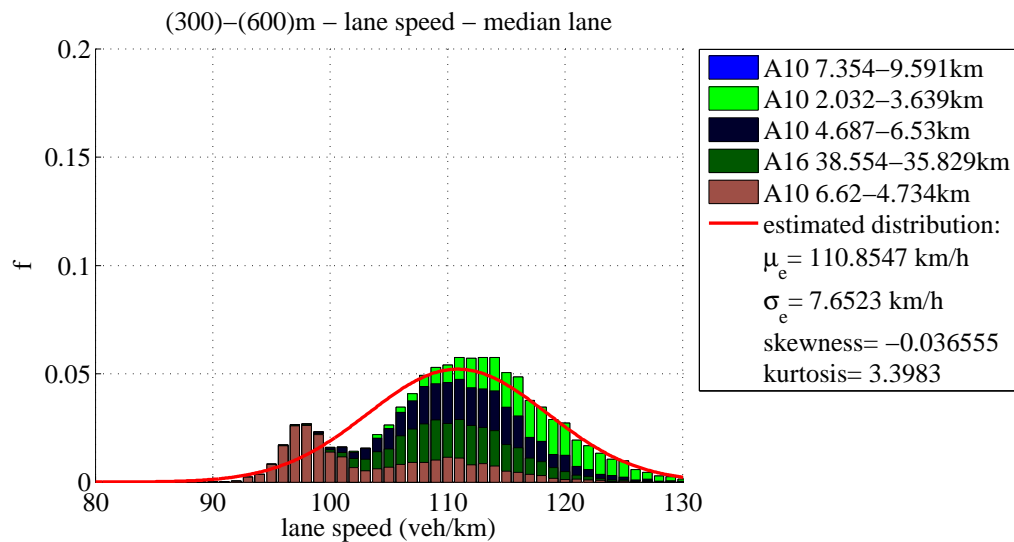


Figure 4.9: Speeds over densities median lane with A10 induction loops in interval 300-600 meter

## 4.6. Selected locations and induction loops

Based on the previous define selection criteria on locations and induction loops and the defined filters on the data locations and induction loops are selected for the analysis. In table 4.3 the selected locations are presented for acceleration lanes and basic freeway segments. Figure 4.10 shows the locations of the selected induction loops on both type of segments. More detailed on the locations is available in appendix D.

Table 4.3: List of the selected locations on basic freeway segments and acceleration lanes

basic freeway segments				acceleration lanes			
roadnumber	from km	to km	#induction loops	roadnumber	from km	to km	#induction loops
A4	23.9	28.3	5	A1	20.8	18.7	2
A15	51.6	55.4	3	A9	30.1	31.6	1
A15	56.7	52.6	4	A10	2.0	3.6	1
A16	17.4	21.6	3	A10	4.7	6.5	1
				A10	7.4	9.6	3
				A15	75.4	73.8	1
				A16	38.6	35.8	1
				A29	12.4	14.0	1

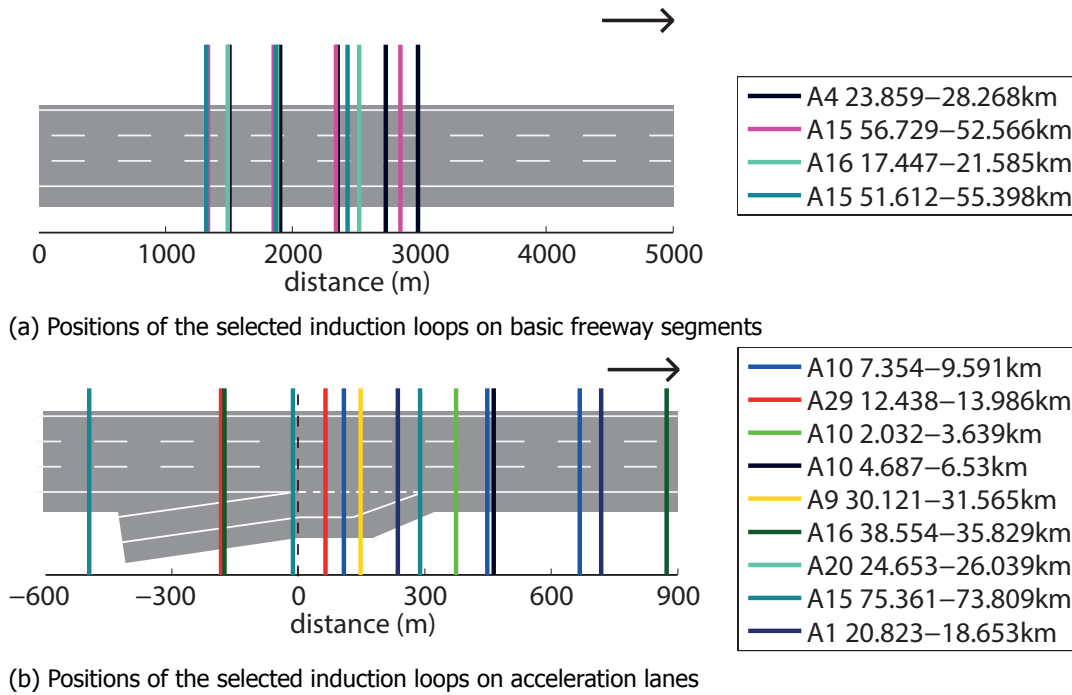


Figure 4.10: Positions of the selected induction loops

The results based on this method seem promising, relations seem to be possible to constructed in the lane flow distributions and lane densities. However, in this thesis, the amount of locations and induction loops is relative low, especially on the upstream region of the acceleration lane. This result in that the size of the error make in the analyses van be relative large.

## 4.7. Selection of interval length

The last step discussed in this chapter is the selection of the interval length. A large part of the analysis consist of a comparison of intervals of induction loops on acceleration lanes. These intervals on the acceleration lanes are compared with the basic freeway segment. The purpose of the intervals is to combine data of induction loops to make an estimation of the distribution of the data in these intervals to gain a general estimation of the distribution of the indicator. The idea is to make these intervals as small as possible to get a much as information as possible about the behavior of the traffic flow characteristics over distance on acceleration lane. However, if the interval length is decreasing the number of induction loops in the interval decreases also and the estimation of the behavior of a single interval becomes less reliable. For the selection of the interval length a consideration need to be made between the quality of the estimation of an interval, more induction loops in an interval, between the quality of estimating the changes in the traffic flow characteristics, low interval length.

In the research three interval lengths are considered: 50 meter, 150 meter and 300 meter. Table 4.4 the number of induction loops in the intervals of the three interval lengths are shown. In the shortest interval length the number of intervals without induction loops is high, this results in that in these intervals the values of the traffic flow characteristics can be estimated. In the 150 meter interval the number of induction loops per interval increases; however, there is still an interval without induction loops and there are also a lot of intervals with only one induction loop. In the intervals with only one induction loop the estimation method only estimates only the behavior of this single loop and not the average behavior of multiple loops. In the largest interval length the number of induction loops increases more. This increase makes it possible to estimate the average behavior of this interval. Only

in the most upstream interval only one induction loop is available. Only in the largest interval length it is possible to estimate the behavior of the intervals well, due to the availability of multiple loops, with exception of the most upstream interval. Therefore, the selected interval length is 300 meter.

Table 4.4: Interval determination

50 meter		150 meter		300 meter	
interval	#induction loops	interval	#induction loops	interval	#induction loops
(-600)-(-550)m	0	(-600)-(-450)m	1	(-600)-(-300)m	1
(-550)-(-500)m	0	(-450)-(-300)m	0	(-300)-(0)m	2
(-500)-(-450)m	1	(-300)-(-150)m	2	(0)-(300)m	5
(-450)-(-400)m	0	(-150)-(0)m	1	(300)-(600)m	3
(-400)-(-350)m	0	(0)-(150)m	3	(600)-(900)m	3
(-350)-(-300)m	0	(150)-(300)m	2		
(-300)-(-250)m	0	(300)-(450)m	2		
(-250)-(-200)m	0	(450)-(600)m	1		
(-200)-(-150)m	2	(600)-(750)m	2		
(-150)-(-100)m	0	(750)-(900)m	1		
(-100)-(-50)m	0				
(-50)-(0)m	1				
(0)-(50)m	0				
(50)-(100)m	1				
(100)-(150)m	2				
(150)-(200)m	0				
(200)-(250)m	1				
(250)-(300)m	1				
(300)-(350)m	0				
(350)-(400)m	1				
(400)-(450)m	1				
(450)-(500)m	1				
(500)-(550)m	0				
(550)-(600)m	0				
(600)-(650)m	0				
(650)-(700)m	1				
(700)-(750)m	1				
(750)-(800)m	0				
(800)-(850)m	0				
(850)-(900)m	1				

## 4.8. Conclusions on data and data processing

For the analyses a data set is created by finding suitable locations and induction loops and filtering inconvenient data from the data set; therefore, criteria and filters are defined to get a convenient data set. Based on the defined criteria and data filters 4 basic freeway segments with in total data of 15 induction loops and 8 acceleration lane locations with in total data of 11 induction loops are selected. During the data process three findings are defined.

First during the location selection fixed average speed checks ("*trajectcontrole*") locations appeared to have a huge impact on the traffic flow characteristics, especially the mean speeds are lower and the lane flow distribution is shifted to the right.

Furthermore, to select convenient induction loops an influence length is defined to select induction loops that are not affected by the upstream or downstream discontinuity. To select a convenient influence length three lengths are compared; the investigated influence lengths are 600, 900 and 1.200 meter. In the 600 and 900 meter influence length the difference between means of the difference induction loops seems to large on the basic freeway segment. In the largest investigated influence length, 1.200 meter, the difference are smaller and the results over distance more continuous; however, three locations are found with a single induction loop, two of these locations were absent in the smaller

influence length, which is rather strange and not logic, and the third location is inconvenient due to discontinuity on the basic freeway segment and this therefore excluded from the suitable locations. For the analyses an influence length of 1.200 meter is selected, which appeared to be the most convenient one. Which can indicate that the turbulence distances are smaller than 1.200 meter.

The last finding on the selected analysis approach is related to the selected locations. Are described above, around acceleration lanes 8 locations are selected with in total 11 induction loops. Divided over the upstream region and downstream region of the acceleration lane are 3 induction loops located in the upstream region and 8 induction loops on the downstream region. The availability of only three induction loops in the upstream region makes the results in this region inconvenient, suggested is to improve the selection procedure in this region and improve reliability of the results and estimation of the turbulence length on this region.



# 5

## Results

In the previous chapter the expected changes in the turbulence are defined on the driving maneuvers and related traffic flow characteristics. The expected changes in the traffic flow characteristics are used to define the required analysis approach to measure the turbulence distance. The analysis approach is divided in two steps to measure the turbulence distance: defining the changes in the traffic flow characteristics and then estimating the start and end points of the changed traffic flow characteristics. With the start and end point the turbulence distances are estimated. For the analyses is in the previous chapter a data set created which fulfill the defined criteria of chapter 3.

As earlier mentioned the analyses consist of two parts, first the changes are investigated, and second the start and end point are estimated all based on the related traffic flow characteristics, mean speeds (both lane and carriageway), lane flow distributions and the lane densities. After the interval comparisons are investigated the start and end points are estimated with a line fitting based on the found changes in the interval comparison. The results of the analyses are compared with the defined expectation in section 2.3.

### 5.1. Speeds

As described in section 3.4 the mean speed are distinguished in two elements, first the mean speed of the carriageway and second the mean speeds of the lanes. The carriageway are investigated on a comparison of the mean and standard deviation between the estimations of the acceleration lane with the estimation of the basic freeway segment.

#### 5.1.1. Carriageway mean speed

Figure 5.1 shows the mean comparison of the carriageway mean speeds, and figure 5.2 shows the standard deviation comparison. The differences between the distributions around the acceleration lanes and the basic freeway segment are tested as significant with a KS-test; the results of the KS-test are shown in table 5.1.

Table 5.1: p-values of the carriageway speed KS-test

Carriageway mean speed	
Interval	Carriageway
(-600)-(-300)m	0.00E+00
(-300)-(0)m	2.79E-281
(0)-(300)m	1.27E-137
(600)-(300)m	0.00E+00
(900)-(600)m	4.08E-105

The carriageway mean speed start higher than the basic freeway segment and decreases until 600

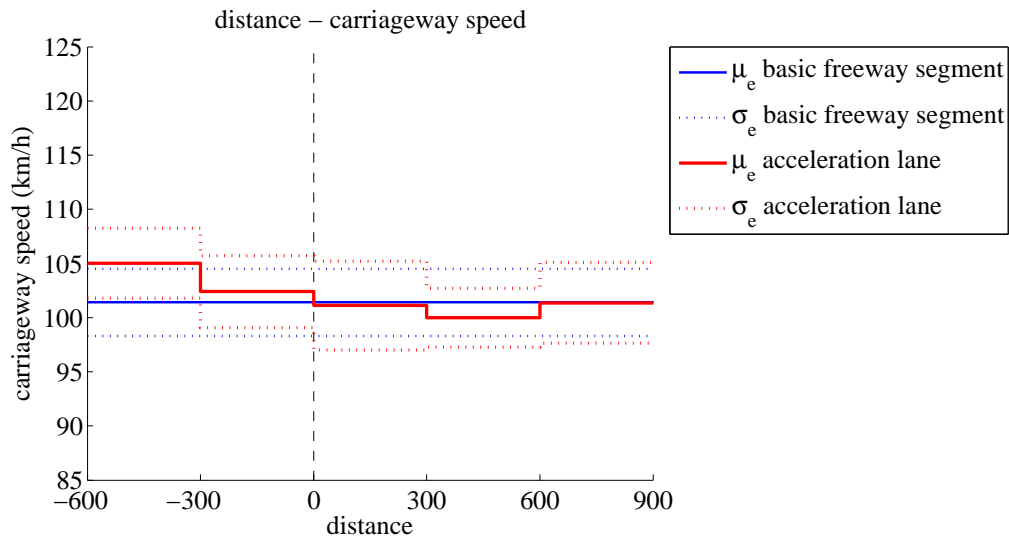


Figure 5.1: Carriageway mean speed comparison estimated distribution basic freeway segments and acceleration lanes

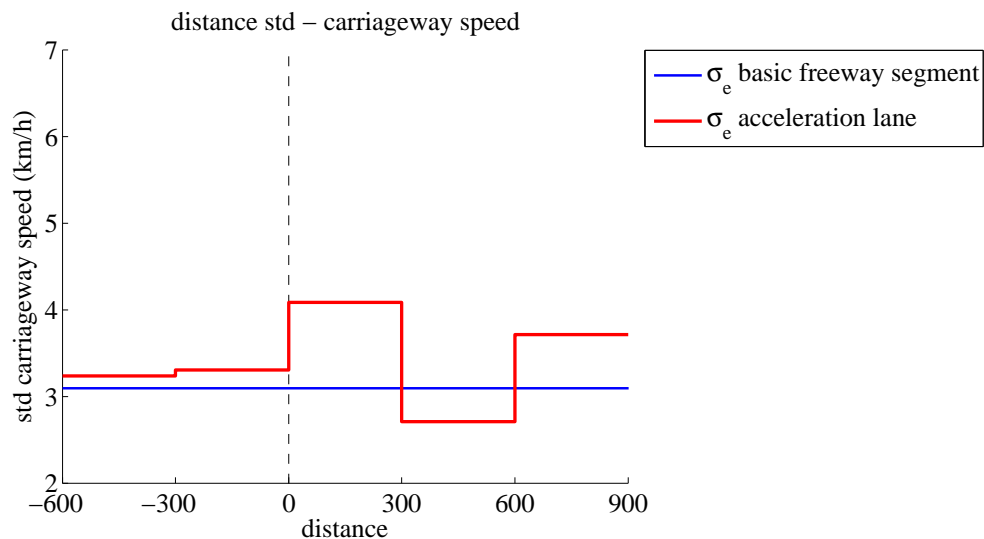


Figure 5.2: Average speed standard deviation comparison estimated distribution basic freeway segments and acceleration lanes

meter downstream of the gore; where the mean speed increases to the level of the basic freeway segment. In the standard deviation, upstream of the gore the standard deviation is more or less comparable with the basic freeway segment; downstream of the gore the standard deviation changes to first larger till 300 meter downstream to smaller between 300 and 600 meter downstream and in the last interval the standard deviation is again larger than the basic freeway segment.

As described in the previous chapter the results upstream of the acceleration lane are inconvenient due to the small number of induction loops. Especially the high speed measured in the first interval can be related to this conclusion. Downstream of the gore the results are more convenient. In this region, a smaller speed is noticed during the acceleration lane and is expected based on the applicable driving maneuvers; however, a decrease in mean speed just downstream of the acceleration was not expected. This decrease in the carriageway mean speed is not found in the lane mean speeds, this decrease in carriageway mean speed can be related with the found lane flow distributions, since more traffic is located on the outside lane the mean speed on the carriageway decreases. Therefore, the merging traffic has an influence on the reduction of the mean speed on the carriageway. A second element found in this interval is the decrease in standard deviation, which can be related to the decrease in



standard deviation on the outside and center lane, where this effect is also found. On these lanes is expected due to an increase in lane changes the share of following/constrained driver increases and therefore the differences in speeds between drivers reduces.

### 5.1.2. Lane mean speed

For the lane mean speeds first the found results are described per lane. After these descriptions the results are discussed on with the expectations. All intervals are found significant with the KS-test; the results are shown in table 5.2.

Table 5.2: p-values of the lane mean speed KS-test

Interval	Lane speed		
	Median lane	Center lane	Outside lane
(-600)-(-300)m	0.00E+00	0.00E+00	0.00E+00
(-300)-(0)m	1.83E-162	1.94E-152	2.79E-281
(0)-(300)m	6.48E-103	6.54E-109	1.27E-121
(600)-(300)m	3.33E-119	2.96E-236	0.00E+00
(900)-(600)m	5.53E-246	6.19E-112	0.00E+00

Figure 5.3 and 5.4 show the changes in the mean and standard deviation of the lane mean speed on the outside lane. The first interval is inconvenient for measurements; between -300 meter upstream and 300 meter downstream of the gore the lane mean speed is increasing and continues to stay on the same level downstream. The lane mean speed starts at -300 meter upstream below the level of the basic freeway segment, and ends above the level of the basic freeway segment. The standard deviation increases from -300 meter upstream till 300 meter downstream of the gore and is above the reference, to decrease dramatically under the reference, and to increase again at 600 meter downstream of the gore above the reference level.

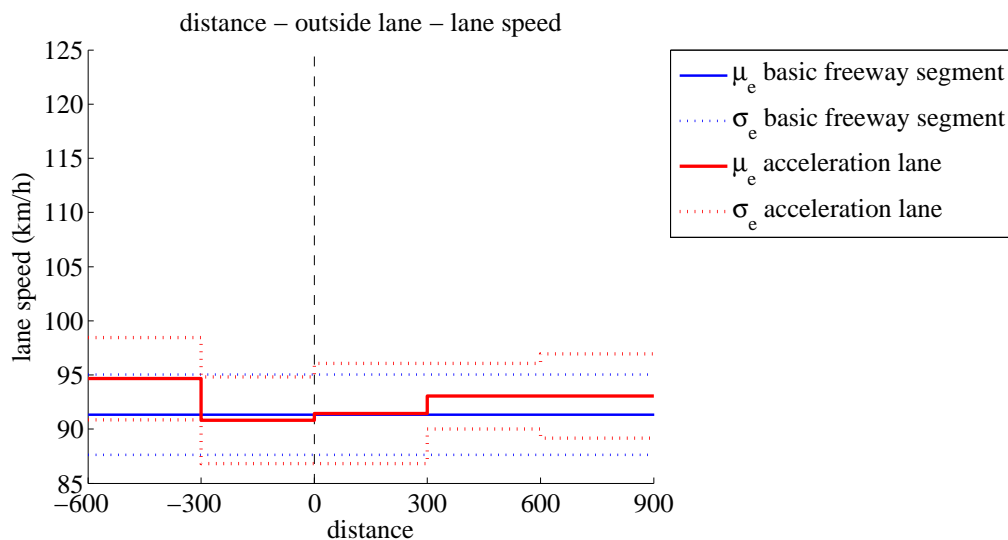


Figure 5.3: Outside lane speed comparison

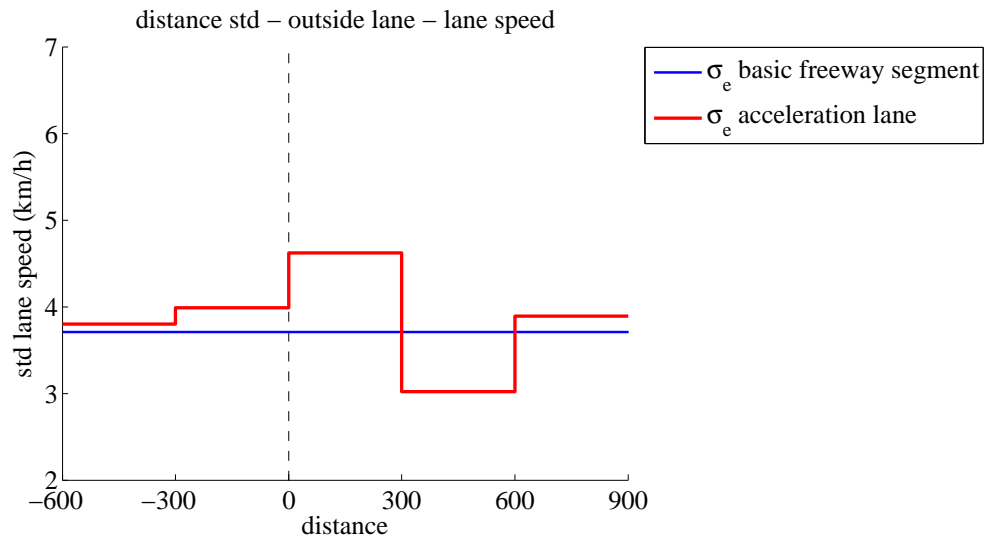


Figure 5.4: Outside lane speed standard deviation comparison

In figure 5.5 and 5.6 are the results on the center lane shown. On the center lane the differences in mean are smaller than on the outside lane, they are more or less equal to the basic freeway segment. Only in the interval 300 till 600 meter downstream a slight increase in mean speed is noticed. In the standard deviations also less differences are found, from -300 till 300 meter downstream the deviation increases slightly above the reference level, as on the outside lane, in the interval just downstream of the acceleration lane the deviation decreases dramatically, to under the reference level; further downstream a comparable level of standard deviation with the reference is found.

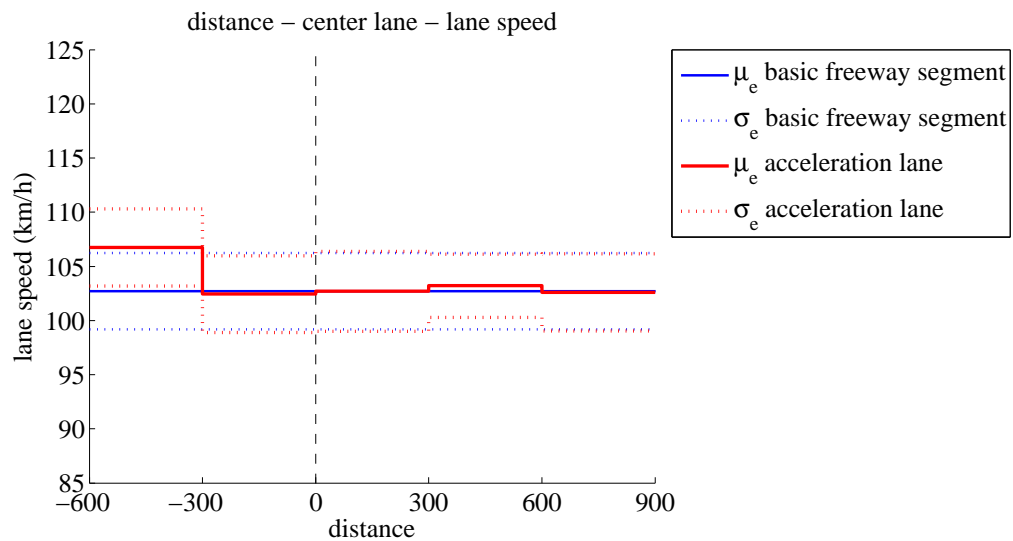


Figure 5.5: Center lane speed comparison

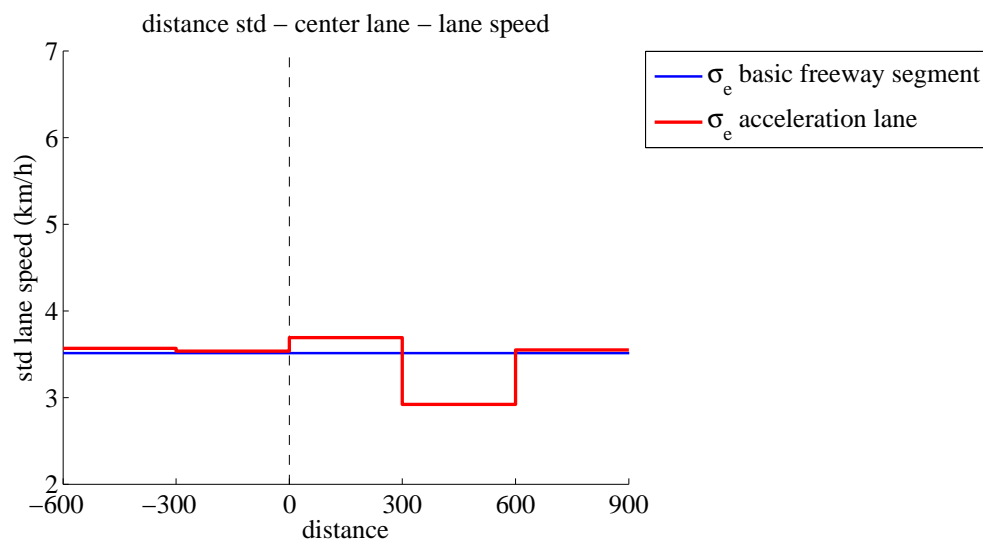


Figure 5.6: Center lane speed standard deviation comparison

The results of the median lane are shown in figures 5.7 and 5.8. At -300 meter the lane mean speed start slightly above the level of the basic freeway segment and start decreasing till 300 meter downstream till just under the level of the reference, at this point the mean speed start to increase to above the level of the reference till the end of the data. The changes in the standard deviation are different than the center and outside lane, on this lane the standard deviation start above the level of the basic freeway segment at -300 meter, from here the standard deviation increases till 600 meter downstream of the acceleration lane, where the deviation start decreasing, continues to stay above the reference level.

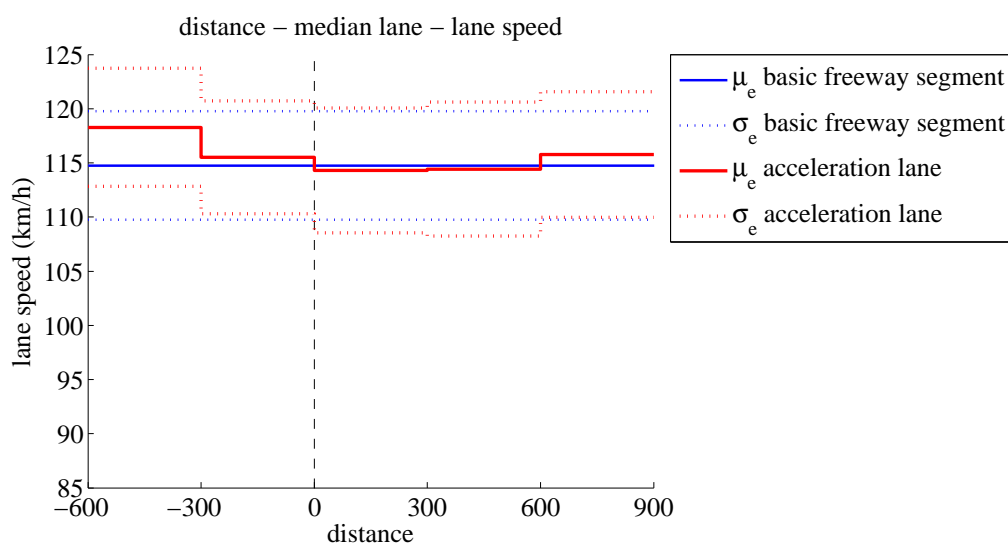


Figure 5.7: Median lane speed comparison

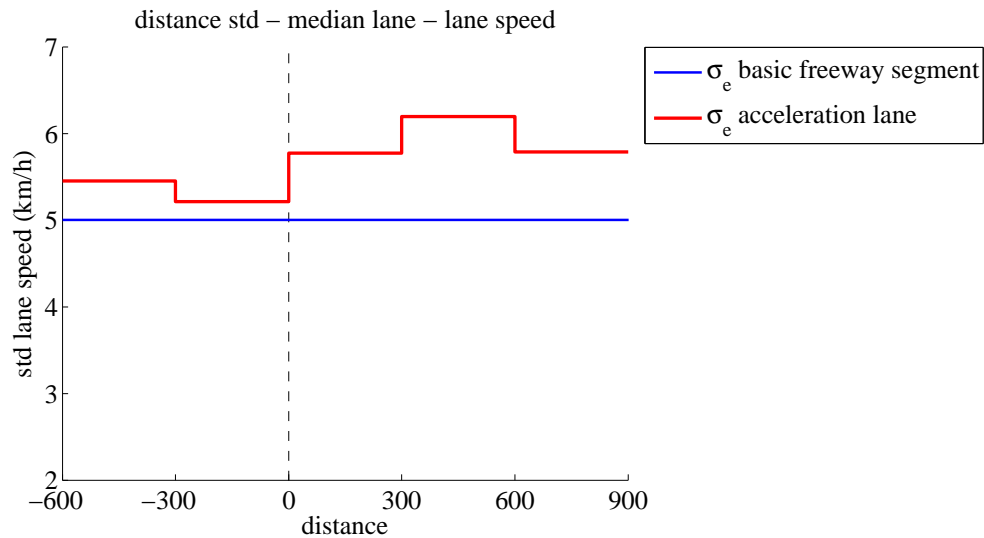


Figure 5.8: Median lane speed standard deviation comparison

The main findings in the lane mean speeds are an increase of the speed downstream of the acceleration lane on the outside lane, on the center and median lane comparable speeds, an increasing standard deviation during the acceleration lane on the outside and center lane followed by a decrease under the reference level in the standard deviation and an increased standard deviation on the median lane until 600 meter where the deviation perhaps start to transform back to the original state. The speed related driving maneuvers seems to have barely effect on the mean speed and standard deviations of the lane mean speeds. The effect of merging drivers seems to have more effect on the lane mean speed. During the acceleration lane increased standard deviations are found, this increase is most likely related to a speed difference in the start and end of this interval as shown in 5.9, where the lower distributions are found in the beginning of the interval and the faster distributions in the end of the interval. This effect can be related to the anticipation of the slower traffic from the acceleration lane by inter alia, pre-allocation, due to the merge of traffic on the outside lane relative more faster vehicles are located on the outside lane which seems to increase the lane mean speed on the outside lane. Further more drivers seem to be constrained according to a decrease in standard deviation on the outside and center lane, an increase in the density by the merging traffic can be the origin of this effect. The last finding is the increase in the standard deviation on the median lane, with a gentle decrease in speed, this effect can be related to the lane changing maneuvers where relative more slower drivers are entering the median lane. Changes in the mean speeds are found to the end of the investigated area, see the outside lane, this could mean that the affected turbulence is exceeding 900 meter downstream and 300 meter upstream, since the more upstream interval is determined on only one induction loop.

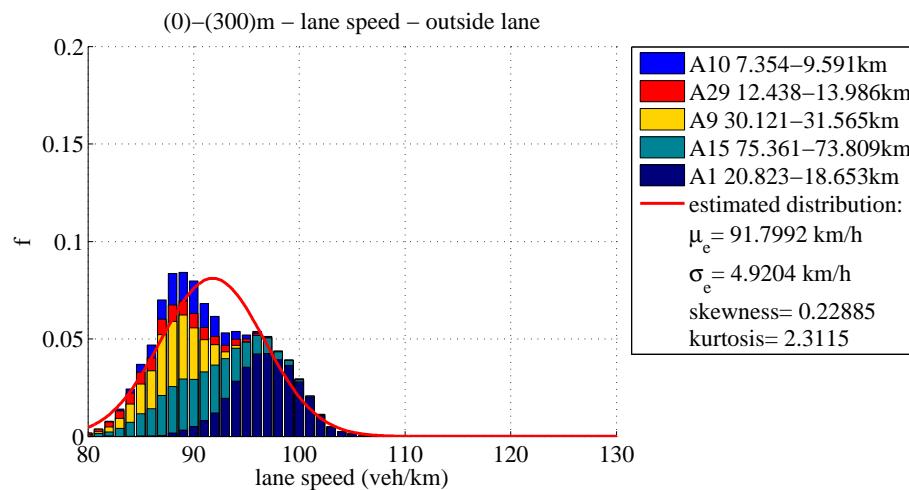


Figure 5.9: Outside lane speed histogram and estimated distribution

## 5.2. Lane flow distribution

The second traffic flow characteristics investigated is the lane flow distribution. As for the lane mean speeds first the results are described and shown per lane, and then the found results are investigated and compared with the expectations of section 2.3. The found results are all found significant with the KS-test, the results of the KS-test are shown in table 5.3.

Table 5.3: p-values lane flow distribution KS-test

Interval	Fraction of flow		
	Median lane	Center lane	Outside lane
(-600)-(-300)m	7.70E-89	0.00E+00	6.42E-238
(-300)-(0)m	0.00E+00	1.95E-15	0.00E+00
(0)-(300)m	2.20E-129	4.72E-172	0.00E+00
(600)-(300)m	0.00E+00	0.00E+00	0.00E+00
(900)-(600)m	0.00E+00	6.19E-242	0.00E+00

Figure 5.10 and 5.11 show the results of the lane flow distribution on the outside lane. On the outside lane start the fraction at -300 meter under the reference level to increase above this level strongly till 600 meter downstream; further downstream the fraction is decreasing, but does not reach the reference level. In this interval the value of the first interval (-600 till -300) is close to the reference level, despite the estimation based on a single induction loop. The standard deviation starts at -300 meter above the reference level to increase strongly till 300 meter to decrease strongly till just above the reference level and increase again to the end of measurements.

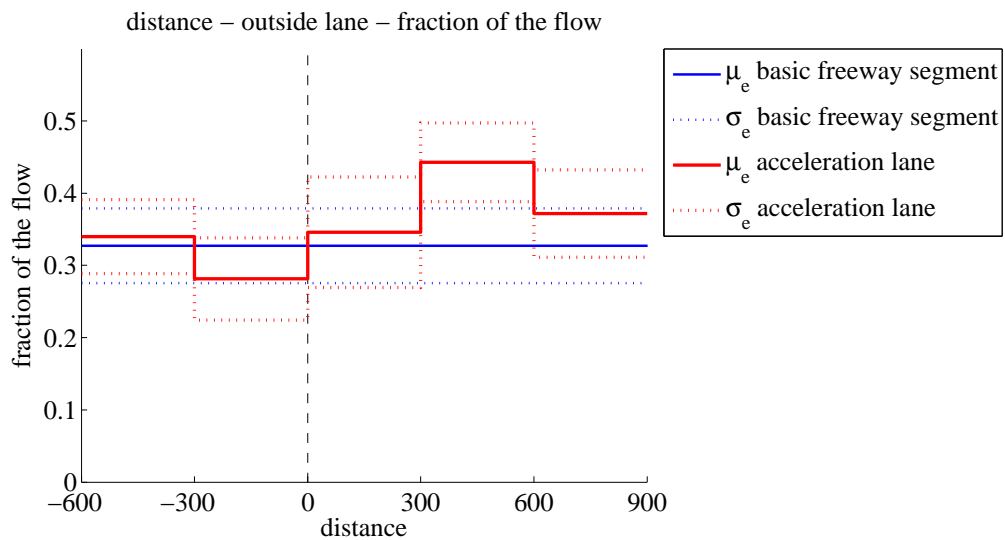


Figure 5.10: Outside lane fraction of the flow comparison

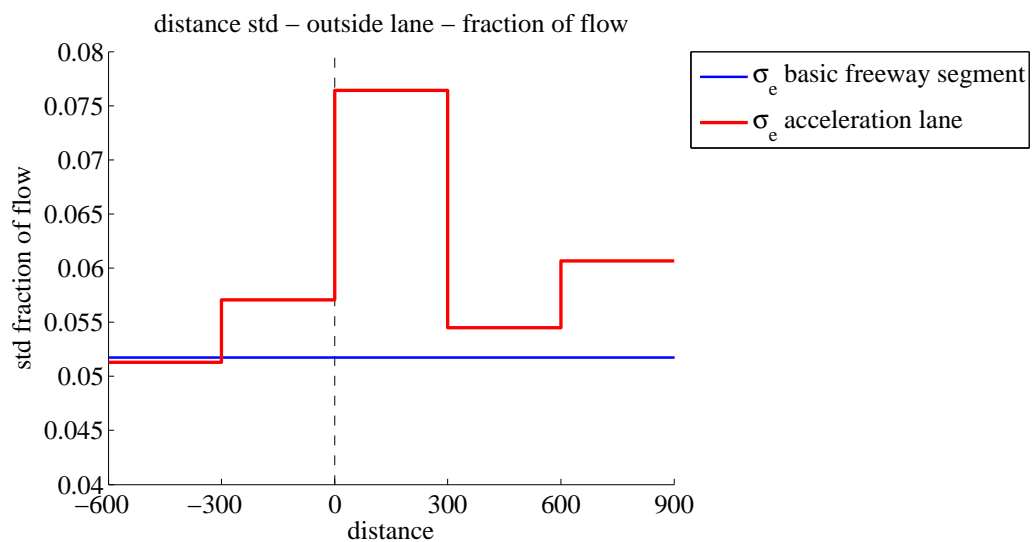


Figure 5.11: Outside lane fraction of the flow standard deviation comparison

Figure 5.12 and 5.13 show the results of the center lane. At -300 meter the fraction of the flow starts almost equal with the acceleration lane; till 600 meter downstream the fraction decreases slightly to under the reference level; after 600 meter downstream the fraction increases slightly. The standard deviation starts also comparable with the reference and increases a bit until 300 meter downstream to above the reference level, at 300 meter downstream the deviation decreases to under the level of the reference to start increasing again.

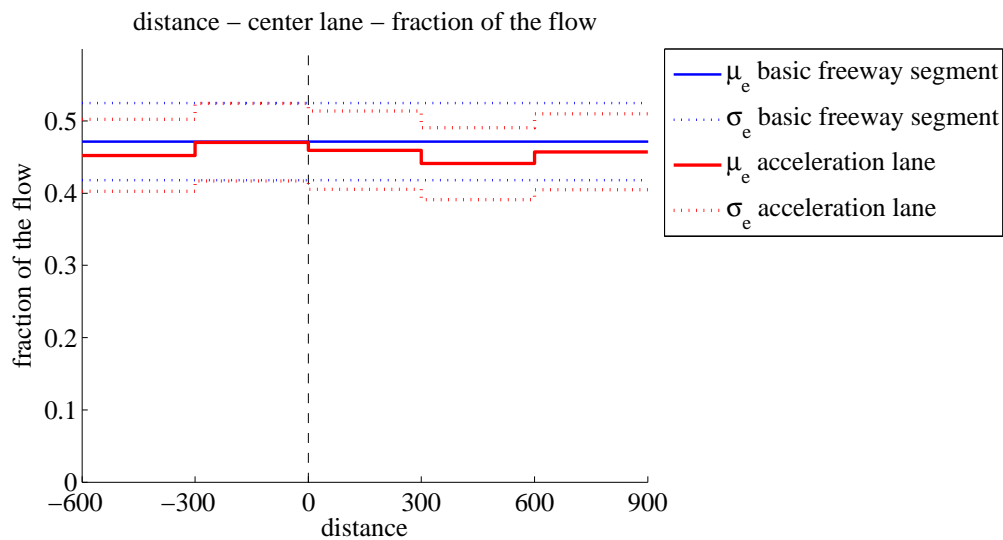


Figure 5.12: Center lane fraction of the flow comparison

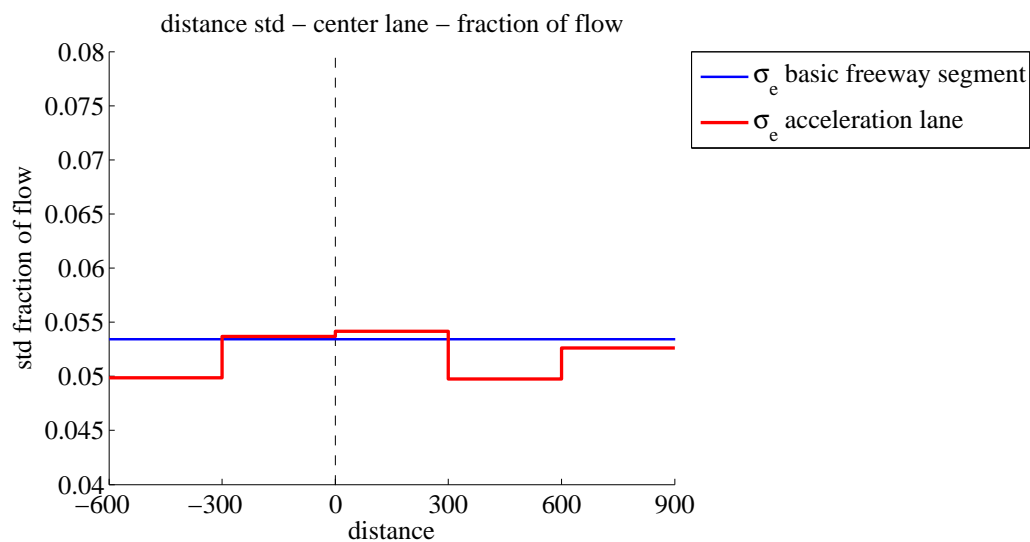


Figure 5.13: Center lane fraction of the flow standard deviation comparison

Figure 5.14 and 5.15 show the results of the lane flow distribution of the median lane. The fraction of the flow start higher than the reference at -300 meter, until 600 meter the fraction decreases to under the reference level, to increase further downstream more towards the reference level. The standard deviation start at -300 meter above the level of the reference to increase even more until 300 meter downstream of the gore. At 300 meter downstream the standard deviation decreases strongly to under the reference level, to increase at 600 meter downstream towards the reference level.

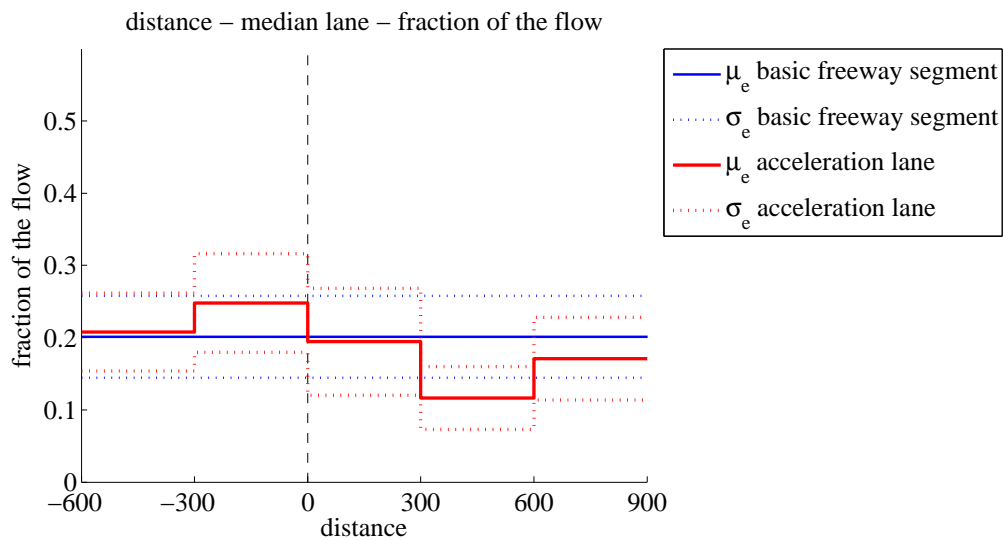


Figure 5.14: Median lane fraction of the flow comparison

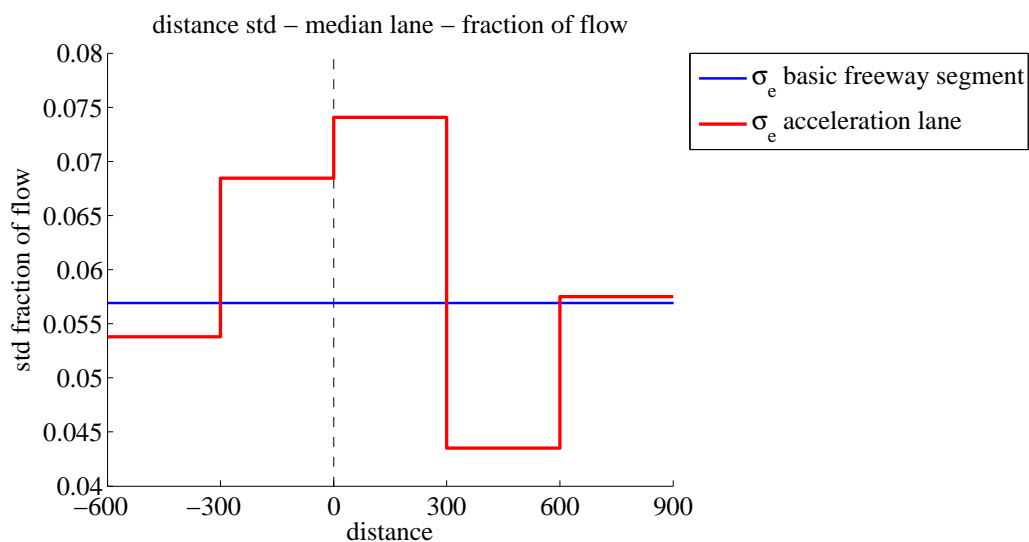


Figure 5.15: Median lane fraction of the flow standard deviation comparison

The results of the lane flow distributions show that the changes in flow fraction are mostly present on the outside and median lane, even the changes in the flow fraction on both lanes are mostly the opposite of each other, where the center lane covers the difference in changes between the outside and median lane. Compared with the reference upstream of the gore the distribution has shift to the left, which indicates the presence of the pre-allocation maneuver. During the acceleration lane the distribution shifts to the right lanes due to the entrance of traffic from the on-ramp. This change is larger than the lane changes by pre-allocation and cooperative lane changes, and thus is in comparison with the basic freeway segment the fraction on the outside lane higher and on the median lane lower. Further downstream is visible that this changes lane flow distribution is returning to the reference lane flow distribution, this indicates that the extra drivers on the outside lane are starting to spread themselves over the lanes to find their preferred lane and speed.

The results of the standard deviation are more difficult to tackle, on the outside lane and median the standard deviation increases, which can be explained by large differences between the results of the induction loops in this interval; however, in the histograms of the lane flow distribution of this interval this behavior is not found, see appendix B; which would be logic due to the location of the



merging traffic. Especially due to the large decrease in the standard deviation just downstream of the acceleration lane on the median lane. The main explanation for this behavior is the smaller difference in the data of this interval. Also the smaller speed differences can be related to this behavior. Traffic could be locked in this interval to their lane.

In the expectations the effect of pre-allocation was predicted by a shift of the lane flow distribution to the left, expected was that the flows would be larger than the effect of merging and that keeping right was a primary change downstream of the acceleration lane. However, the effect of the merging traffic is larger, and therefore the lane flow distribution shift to the right, passing the reference level during the acceleration lane. Downstream of the acceleration lane this change lane flow distribution stays present until 600 meter downstream of the gore and downstream of this point the extra traffic on the outside lane seems to spread out to the other lanes and the lane flow distribution changes back to it's former state as the basic freeway segment. The changes in the traffic flow characteristics seems to exceed the current investigated area and data, this could mean that the affected turbulence is longer than the 900 meter downstream and 300 meter upstream.

### 5.3. Density

The last traffic flow characteristics investigated is the lane density. As the other traffic flow characteristics, the differences between the acceleration lanes and the basic freeway segment appeared to be significant, based on the KS-test shown in table 5.4. Further the results of the lane density are shown in figure 5.16 till 5.21. The results appeared to be strongly comparable with the lane flow distribution. This effect is logic, since the density of the carriageway is constant (30 to 35 veh/h) and the changes in the speeds are small.

Table 5.4: p-values lane density KS-test

Interval	Density		
	Median lane	Center lane	Outside lane
(-600)-(-300)m	3.52E-75	0.00E+00	1.80E-173
(-300)-(0)m	0.00E+00	2.82E-22	0.00E+00
(0)-(300)m	8.98E-116	1.36E-179	0.00E+00
(600)-(300)m	0.00E+00	0.00E+00	0.00E+00
(900)-(600)m	0.00E+00	1.35E-153	0.00E+00

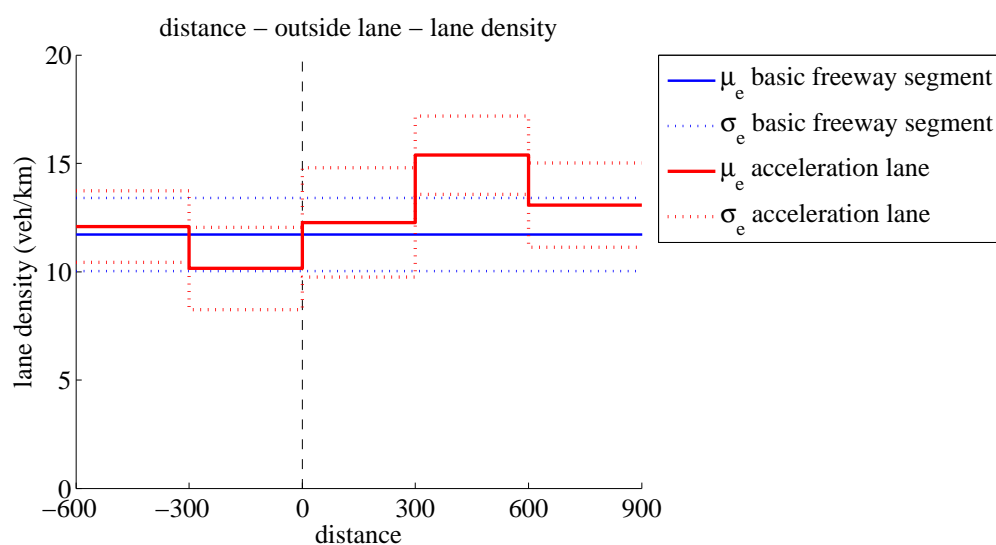


Figure 5.16: Outside lane density comparison

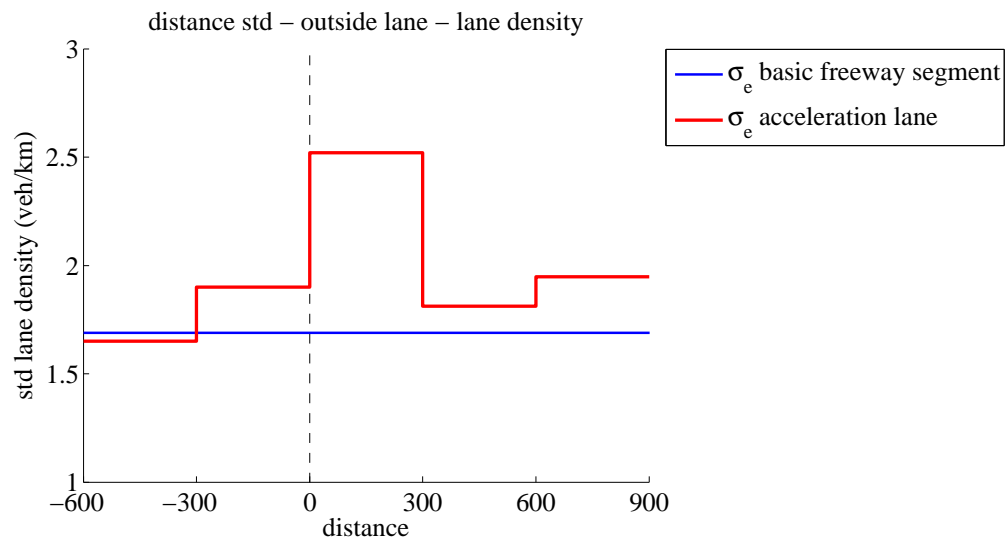


Figure 5.17: Outside lane density standard deviation comparison

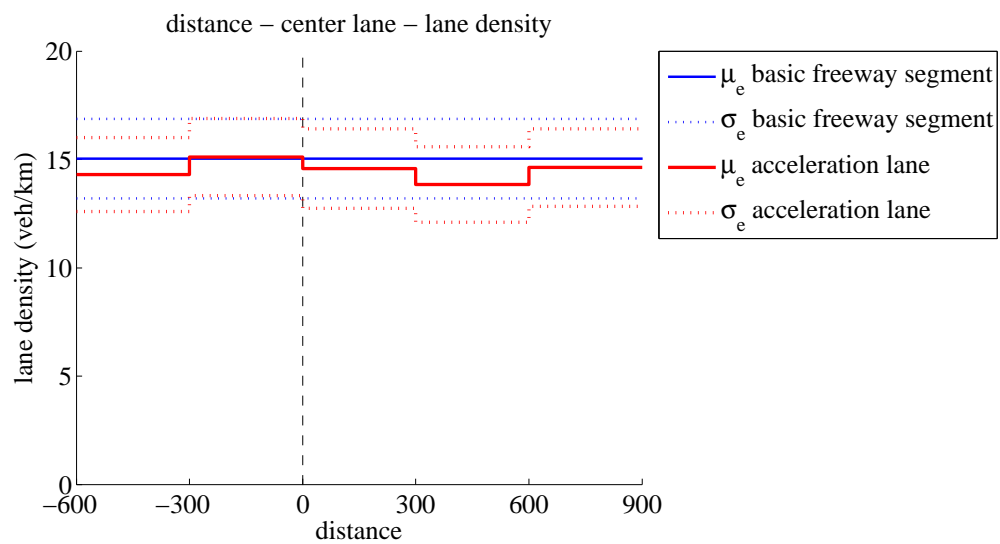


Figure 5.18: Center lane density comparison

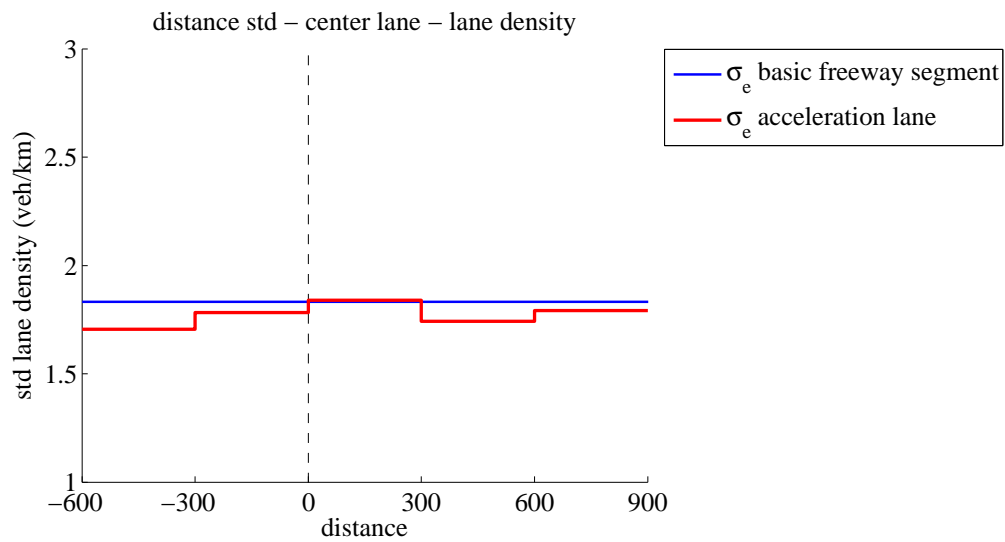


Figure 5.19: Center lane density standard deviation comparison

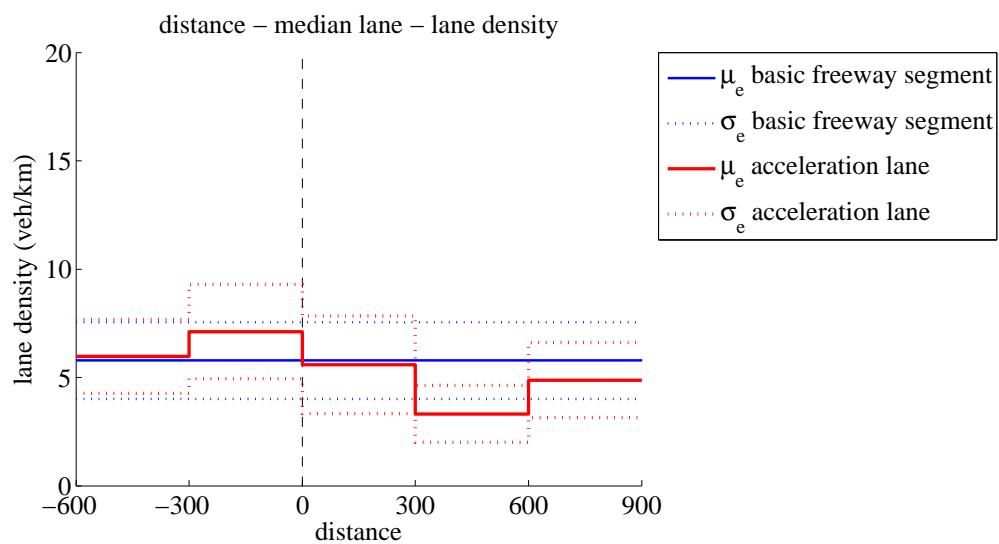


Figure 5.20: Median lane density comparison

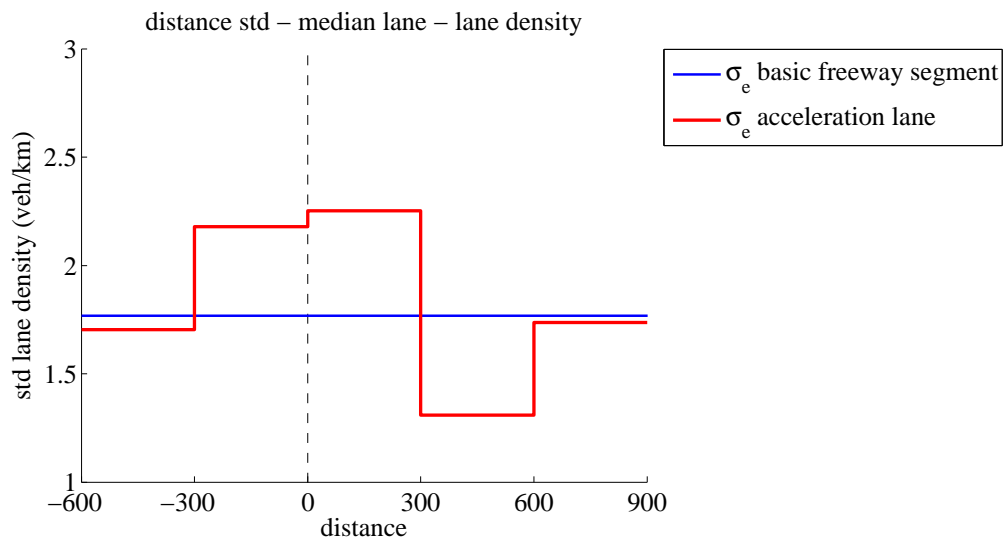


Figure 5.21: Median lane density standard deviation comparison

The background of investigating the density is to find the effects of relaxation, mainly on the outside lane. However, the relaxation maneuver is less present in the traffic flow than the changes appearing by the entrance of traffic from the on-ramp. Therefore an increase is not visible on the outside lane downstream of the acceleration lane. The lane density reaches at the end of the investigated region not the reference level of the basic freeway segment; therefore, it is possible that further downstream relaxation is present in the traffic flow and thus it seems that the turbulence distance could be exceeding the investigated region of -300 till 600 meter.

#### 5.4. Turbulence distance estimation

The second part of the analyses is to estimate with the defined changes in the traffic flow the start and end points of the changed traffic flow characteristics and so the turbulence distance. Based on the previous results the means of the lane flow distribution and the lane density seems to have a quite obvious relation. On the outside lane the fraction of the flow and the density decreases due to pre-allocation, due to the traffic from the acceleration lane the fraction and density increases and becomes larger than the reference, the last change is a decrease in the direction of the basic freeway segment. The center lane seems to act continues over the distance and the median lane seems to act as the opposite of the outside lane.

The behavior of the outside and median lane seems to be similar with a S-shape. Which both are the opposite of each other based on the estimated means. When the individual means of the induction loops are investigated, the same shape is visible. Figure 5.22 shows the means of the individual induction loops around the acceleration lane over distance. Therefore, these means of the individual induction loops are used to estimate the start and end points of the affected turbulence. The S-shape is approached by fitting a third-degree polynomial on the means of the individual loops. As described in chapter 3 the start and end points are determined by calculating the cross-sections between the estimated mean of the basic freeway segment with the fitted line on the means of the induction loops around the acceleration lane.

Figure 5.23 and 5.24 show the results of the line fitting on the individual induction loops and the cross-sections of the estimated mean of the basic freeway segment and the fitted line. The  $R^2$  shows how well the line is fitted on the available means, the closer to 1 the better the line is presenting the shape of the means. In both figures the outside lane the  $R^2$  has a values of 0.8 and on the median lane about 0.7; this shows that the fit on these means is quite well. In the upstream region of the acceleration lane only three induction loops are available, this reduces the reliability of the estimation of the start point. Based on these means a start point of about 500 meter is estimated. Downstream

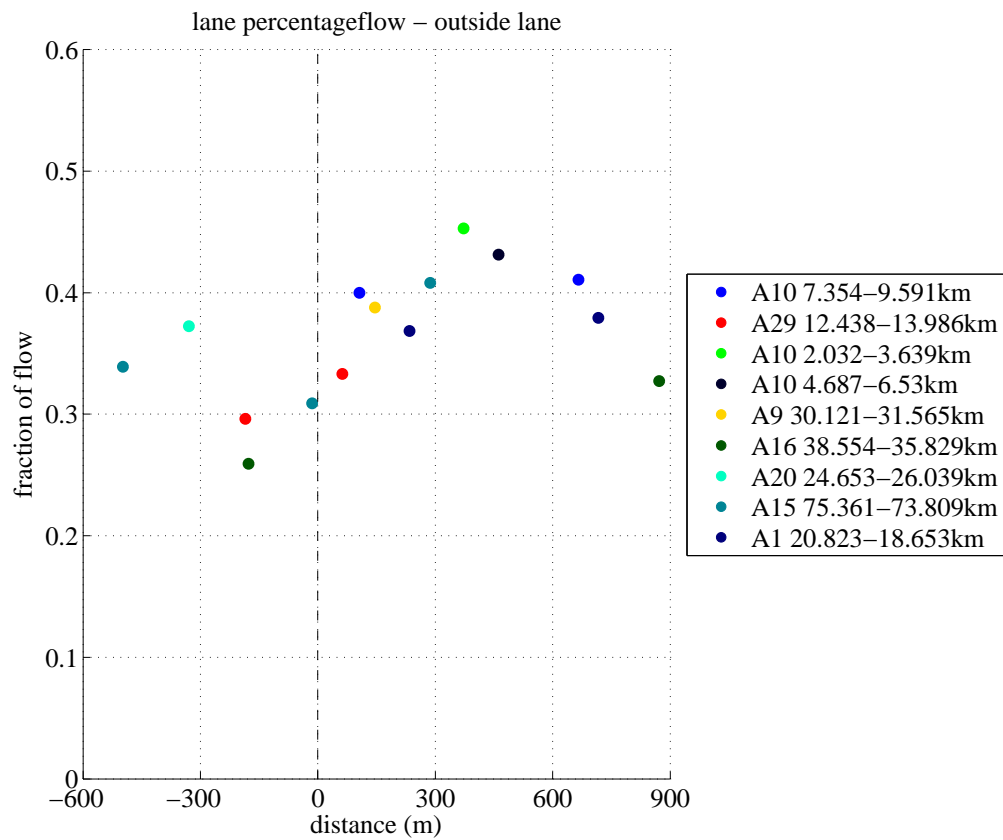


Figure 5.22: Fraction of the flow means of the individual induction loops on the outside lane

the number of induction loops is larger, which increases the reliability and a end point of about 800 meter is estimated. Therefore based on these measurements the turbulence distances are estimated on 500 meter upstream of the gore and 800 meter downstream of the gore.

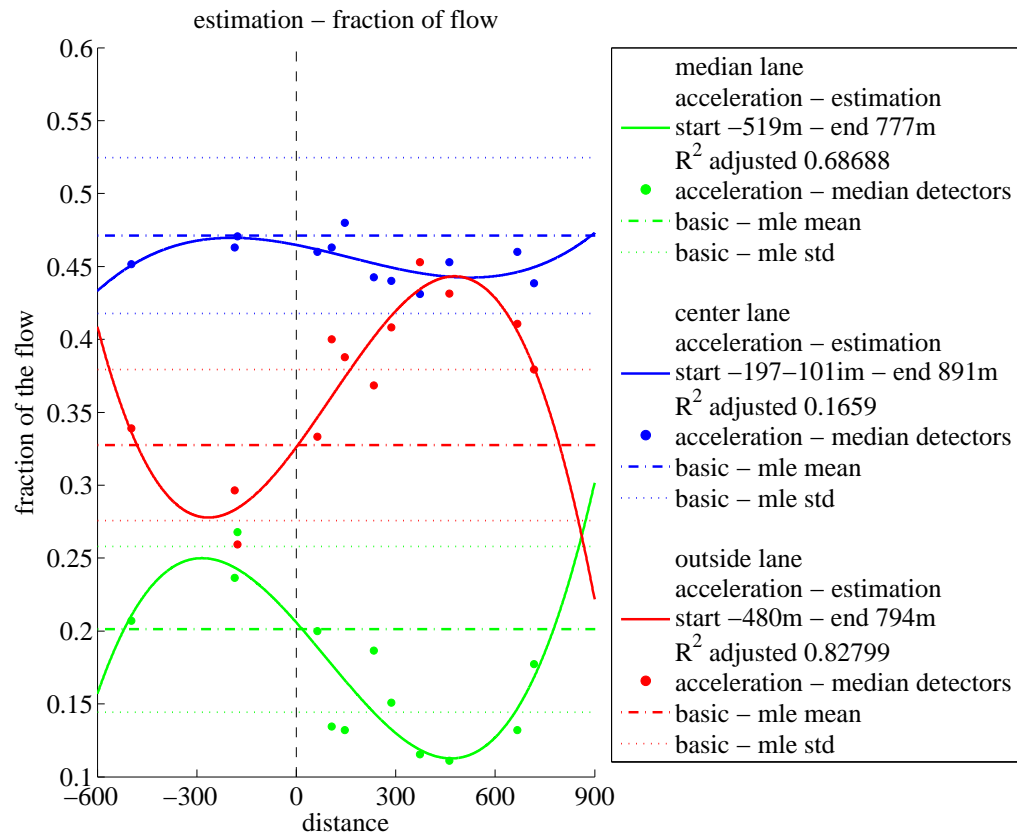


Figure 5.23: Estimation start and end point influence acceleration lane

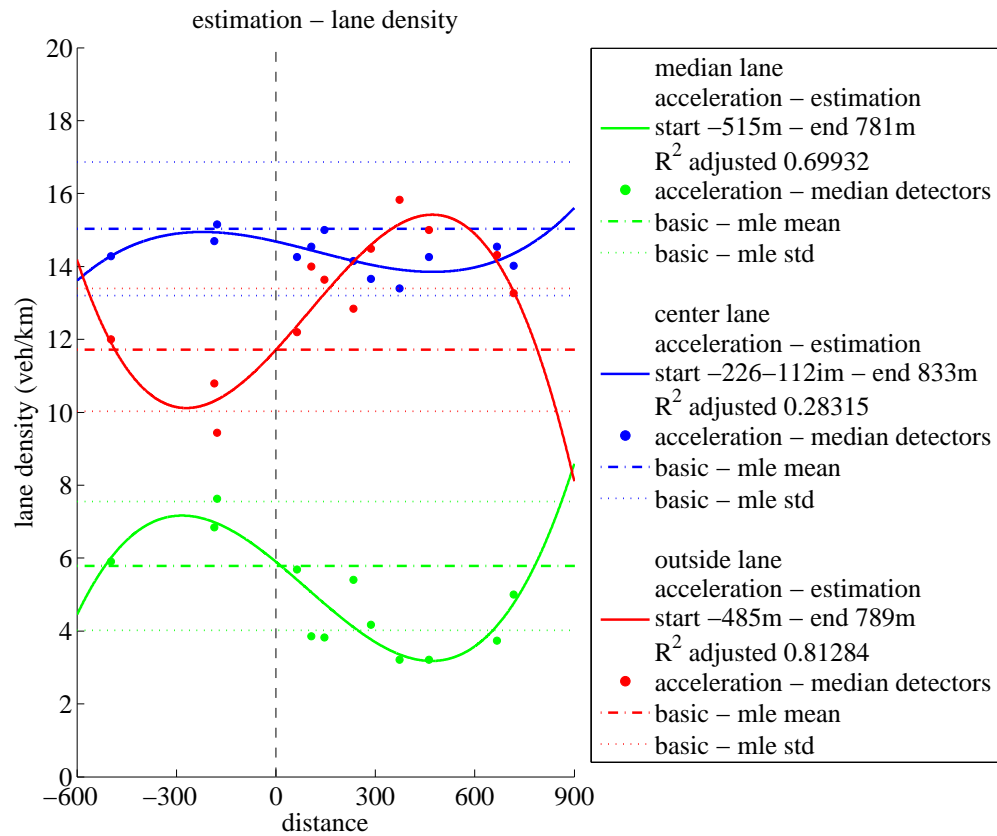


Figure 5.24: Estimation start and end point influence acceleration lane

In section 2.3 the expectation is defined as 130 meter upstream and 600 meter downstream of the gore based on the NOA. These values are smaller than the measured turbulence distances. It is possible that in the period between this thesis and the determination of the turbulence distances of the NOA the turbulence distances increased due to changes in the driver and vehicles compositions; further it is possible that in the NOA turbulence distance is defined differently by in example incorporating the traffic safety effects. Especially the upstream turbulence distances differ; however, in this thesis the selected upstream induction loop selection is identified as not completely reliable, which means that it is possible that the induction loops on the upstream part are possibly affected by the upstream discontinuity. Also the low number of locations and induction loops in total can be increased to offer more reliable results. Furthermore, the found results need to be invalidated due to the low number of induction loops and it is wise to investigate the safety aspects of the affected turbulence to see if the measured turbulence distances can be reduced.



# 6

## Conclusions and recommendations

The last chapter presents the main conclusions of this thesis. The conclusions are drawn from the findings of this research. The last section of this chapter gives recommendations for future research on the turbulence distances and application of the defined data approach. First the main conclusion of this research is defined.

The main conclusion of this thesis is:

---

*"The turbulence distances that are affected by the presence of the acceleration lanes are empirically measured with macroscopic traffic flow characteristics. The measured traffic flow characteristics are mean speeds, lane flow characteristics and lane densities. The traffic flow characteristics change due to the presence of the acceleration lane. In the mean speeds are small changes found; however no clear shape of these changes is found to estimate the distances; in the lane flow distributions and lane densities are larger changes found with a clear shape. With this shape of the changes in the lane flow distribution and lane density the turbulence distances are roughly estimated. The estimated turbulence distances are 500 meter upstream of the gore and 800 downstream of the gore."*

---

### 6.1. Findings

Throughout the thesis, findings are defined based on the research steps performed to find the main conclusion. In this section the main findings of this thesis are summarized.

The first step in investigating turbulence distances around acceleration lane is to define the reasons why turbulence is different around acceleration lanes than on basic freeway segments. Based on the differences the turbulence distance is relevant in freeway design. Driving from origin to destination can be described as a driving task; the driving task consists of all tasks executed by the driver to reach his/her destination safely, comfortably and timely. Minderhoud [16] distinguished the driving task in five subtasks: strategic route navigation, lateral roadway subtask, longitudinal roadway subtask, lateral vehicle interaction and longitudinal. These subtasks are seen as the cause of the turbulence in traffic flow, these subtasks result in lane changes and speed changes in the traffic flow. On roadways with no discontinuities the strategic roadway task is not available in the set of tasks, while around acceleration lanes this task is available in the traffic flow; this is a main difference between the acceleration lane and the basic freeway segment. Further, due to the availability of the strategic route navigation subtask more interaction is arising between other drivers and the merging driver to reach their destinations safely and comfortably. The presence of the strategic route navigation and the increase of more interaction subtasks are the cause of the changed turbulence around acceleration lanes.

These subtasks are related to driving maneuvers in the traffic flow. Based on the route navigation subtask and extra interaction subtasks multiple driving maneuvers increases and new driving maneuvers are present around acceleration lanes in the traffic flow. Around acceleration lanes two new driving

maneuvers are present, pre-allocation and merging. Pre-allocation is a lane change to the left adjacent lane to provide gaps on the outside lane for traffic from the acceleration lane; merging is a lane change from the acceleration lane to the adjacent left lane to enter the freeway from the on-ramp. Further, the following driving maneuvers increase around acceleration lanes: courtesy yielding, cooperative lane changing, keeping right and relaxation. Courtesy yielding is a gentle decrease made by driver on the outside lane to increase the gap let a merging driver merge; cooperative lane change is a lane change to the left adjacent lane to let (nearly) adjacent driver merging to the outside lane; keeping right is a lane change to the right adjacent lane to follow the traffic rules to driver on the right lanes if possible, and relaxation is the phenomenon of accepting a headway shorter than desired by a merging vehicle for a short period and after this period these headways are enlarged.

The extra driving maneuvers around acceleration lanes are related to changes in traffic flow characteristics to measure the turbulence changes by the presence of the acceleration lanes. The traffic flow characteristics are divided in two levels, macroscopic and microscopic. Microscopic traffic flow characteristics are related to the individual driving maneuvers and the macroscopic traffic flow characteristics to the traffic flow. The extra driving maneuvers are related to changes in the number of lane changes, speed changes and changes in the headways. These microscopic effects are related to changes in the lane flow distributions, mean speeds and densities on macroscopic level. Turbulence can be investigated on both levels and in this thesis macroscopic traffic flow characteristics are investigated based on the selected data collection method.

To measure the turbulence distances of acceleration lanes two steps are performed, first the changes in the macroscopic traffic flow characteristics are investigated by comparing the traffic flow characteristics of acceleration lanes with the reference case basic freeway segment; based on these changes in the macroscopic traffic flow characteristics the turbulence distance is determined. For the measurement of the macroscopic traffic flow characteristics induction loops are used; however, induction loops have an average spacing of 500 meter, to improve the accuracy of the determination of the turbulence an innovative approach is applied. In this innovative approach induction loops from multiple locations are combined to fill these spacings with induction loops from other location, and so receive data of the traffic flow characteristics within the 500 meter spacings. This approach combined with a comparison of the acceleration lane and the basic freeway segment is applied to measure the changes in traffic flow characteristics and the turbulence distance.

For the analyses a data set is created by finding suitable locations and induction loops and filtering inconvenient data from the data set; therefore, criteria and filters are defined to get a convenient data set. Based on the defined criteria and data filters 4 basic freeway segments with in total data of 15 induction loops and 8 acceleration lane locations with in total data of 11 induction loops are selected. During the data process three findings are defined.

First during the location selection fixed average speed checks ("*trajectcontrole*") locations appeared to have a huge impact on the traffic flow characteristics, especially the mean speeds are lower and the lane flow distribution is shifted to the right.

Furthermore, to select convenient induction loops an influence length is defined to select induction loops that are not affected by the upstream or downstream discontinuity. To select a convenient influence length three lengths are compared; the investigated influence lengths are 600, 900 and 1.200 meter. In the 600 and 900 meter influence length the difference between means of the difference induction loops seems to large on the basic freeway segment. In the largest investigated influence length, 1.200 meter, the difference are smaller and the results over distance more continuous; however, three locations are found with a single induction loop, two of these locations were absent in the smaller influence length, which is rather strange and not logic, and the third location is inconvenient due to discontinuity on the basic freeway segment and this therefore excluded from the suitable locations. For the analyses an influence length of 1.200 meter is selected, which appeared to be the most convenient one. Which can indicate that the turbulence distances are smaller than 1.200 meter.

The last finding on the selected analysis approach is related to the selected locations. Are described above, around acceleration lanes 8 locations are selected with in total 11 induction loops. Divided over the upstream region and downstream region of the acceleration lane are 3 induction loops located in the upstream region and 8 induction loops on the downstream region. The availability of only three

induction loops in the upstream region makes the results in this region inconvenient, suggested is to improve the selection procedure in this region and improve reliability of the results and estimation of the turbulence length on this region.

The first macroscopic traffic flow characteristic investigated is the mean speed, the mean speed is investigated on the carriageway and on the different lanes. On the center and median lane the mean speed is comparable with the mean speed of the basic freeway segment, on the outside lane a small increase is noticed downstream of the acceleration lane and no return is noticed; on the carriageway a decrease is found just downstream of the acceleration lane and further downstream the mean speed seems to return to its original value. In the standard deviations larger differences are found in comparison with the basic freeway segment, during the acceleration lane the standard deviation is larger, 300 meter downstream of the acceleration lane the deviation is smaller, except on the median lane. 600 meter downstream of the acceleration lane the standard deviation becomes again larger than the basic freeway segment. No relations can be found with the found driving maneuvers identified. The decrease in the speed on the carriageway can be related with the changes in the lane flow distribution where the flow on the slowest lane increases due to the presence of the acceleration lane. The changes in the standard deviation seem to be related to the differences in the data sets of the induction loops within the investigated intervals.

The second macroscopic traffic flow characteristics investigated is the lane flow distribution. The means and the standard deviation are also investigated of the lane flow characteristics. Upstream of the gore the fraction is lower on the outside lane and increases till 600 meter downstream of the gore to above the level of the basic freeway segment and start to decrease. The differences between the acceleration lane and the basic freeway segment are smaller on the center lane, downstream of the gore the fraction decreases slightly till 600 meter downstream to start increasing again. And on the median lane upstream of the gore the fraction is smaller than the reference, till 600 meter downstream the fraction increases strongly to above the reference level, and starts to decrease again.

The changes on the outside lane reflect mainly on the median lane, the changes on these lanes are the more and the less the opposite of each other. The decrease on the outside lane and increase on the median lane upstream of the gore is related to the pre-allocation maneuver, during the acceleration lane the fraction on the outside lane increases due to the merging traffic from the acceleration lane, this effect is larger than the pre-allocation and cooperative lane changes, since the fraction on the outside lane becomes larger than on the basic freeway segment, further downstream the fraction on the outside lane starts to decrease again in the direction of the basic freeway segment, an explanation for these changes is the spreading maneuvers or overtaking maneuvers of the traffic from the acceleration lane to reach their preferred lane and speed.

The last macroscopic traffic flow characteristics investigated is the density per lane. The results found in the lane densities is comparable with the lane flow densities. The main reason for this effect is the fixed carriageway density to make a fair comparison between the acceleration lane and the basic freeway segment and to combine the data of the different induction loops. The expected effect of the relaxation in the data is not found in the lane densities especially on the outside lane.

The last analysis consists the estimation of the turbulence distance. Based on the found changes in the lane flow distribution and the lane density the turbulence distance is estimated with a third degree polynomial which is related to the S-shape found in the earlier analyses. Based on this line fitting the turbulence distance upstream and downstream are estimated on 500 meter upstream and 800 meter downstream.

## 6.2. Conclusions

With the findings summarized in the previous sections, the conclusions are defined for this thesis.

No clear definition of turbulence in traffic flow is found in literature, only descriptions and explanations, especially in the NOA and HCM, the main design manuals in the Netherlands and the United States. Therefore a new definition is determined, mainly based on the descriptions in the HCM and the turbulence in other fields, fluids and gases. The definition of turbulence in traffic flow is: turbulence

describes the amount of the changes in the traffic flow, based on speed changes and lane changes.

Based on the description of Minderhoud of the driving task the cause of turbulence in traffic flow is identified. The main difference between a continuous freeway segment (or basic freeway segment) is the presence of the strategic route navigation, which results in the merging driving maneuver, which is the lane change maneuver from the acceleration lane to the outside lane of the carriageway. Due the strategic route navigation more anticipation decision need to be taken on the freeway by the drivers to drive comfortably and safely, which results in extra driving maneuvers around the acceleration lane. Around the acceleration lane pre-allocation, courtesy yielding, cooperative lane changes, keeping right and relaxation are identified as extra anticipation and interaction decisions to drive comfortably and safely.

The developed data methodology shows to have some possibilities for further researches to investigated traffic flow characteristics over distance around discontinuities; however, in this thesis the amount of locations and induction loops are in the final stage low, this decreases the possibilities to investigate the effects more accurate and possibly increases the error in the results.

One part of the data processing consist of determining the influence length to select induction loops that are not affected by the presence of a discontinuity. Based on this part, it seems that the influence of the presence of a discontinuity is not present anymore further than 1.200 meter away from the discontinuity.

Based on the comparisons of the acceleration lane and the basic freeway segment, the effects of extra speed changing maneuvers around acceleration lanes seems to not be present around acceleration lanes; however, the extra lane changing maneuvers are clearly present around acceleration lanes; even more lane change maneuvers seems to be present, this driving maneuver is the spreading of drivers from the acceleration lane over the lanes of the carriageway. Also pre-allocation is clearly present upstream of the acceleration lane. These effects are seen in the lane flow distribution and lane density data. It seems that the influence of the acceleration lane exceeds the investigated area of 600 meter upstream and 900 downstream of the gore.

Based on the changes in the lane density and lane flow distribution the turbulence distances is estimated on 500 meter upstream and 800 meter downstream of the gore.

### 6.3. Recommendations

The applied methodology seems to give insight in the macroscopic effects of the presence of the acceleration lane on the traffic flow; however, the amount of investigated induction loops is relative low, especially upstream of the gore, to get accurate changes in the traffic flow to estimate the turbulence distances. It is advisable to validate the results with other measurement methods to see if the same changes in the traffic flow characteristics can be found in the new data set. Especially upstream and downstream outside the acceleration lane, which are usually not the investigated regions. When the found changes in this thesis are validated the estimated turbulence distances can be estimated more accurate with the new data set and the developed method in this thesis can be used to estimate turbulence distances around the other discontinuities.

Not all expected driving maneuvers are found in the analyses, one of these driving maneuvers is relaxation. This driving maneuvers can be present outside the investigated region of 900 meter downstream. This driving maneuvers is in literature also not clearly determined. This question is still rising how far the relaxation driving maneuver is present downstream of the acceleration lane; it is advisable to determine the relaxation distance.

In the analyses no large differences where found in the mean speed; however, in the lane flow distributions and lane densities are large differences found upstream and downstream of the gore. Therefore, lane changes has the largest influence on the changes in the traffic flow characteristics. For the determination of the other turbulence distances lane flow distributions are required to be investigated for measurement.

The last recommendation is on the applied data collection method. This method are a newly developed approach to investigate over distance changes in the traffic flow characteristics. The performance of this method appeared in the final analysis to be working quite well and can therefore be used in other researched; however, the error can be quite large due small sets of induction loops.



# Bibliography

- [1] R. Dubbeldam, *NOA : nieuwe ontwerprichtlijn autosnelwegen* (Rijkswaterstaat, Adviesdienst Verkeer en Vervoer, Rotterdam, 2007).
- [2] *Richtlijnen voor het ontwerpen van autosnelwegen* (Rijkswaterstaat, Adviesdienst Verkeer en Vervoer SDU Uitgeverij Koninginnegracht, Rotterdam 's-Gravenhage, 1993).
- [3] R. Kraaijeveld, *Onderzoek naar de identificatie van kwetsbare wegvakken*, Ph.D. thesis, Master's thesis, Delft University of Technology (2008).
- [4] Y. Wijnia and P. Herder, *The state of asset management in the netherlands*, in *Engineering Asset Lifecycle Management* (Springer, 2010) pp. 164–172.
- [5] Ministerie van Verkeer en Waterstaat, Rijkswaterstaat, Adviesdienst Verkeer en Vervoer (RWS, AVV), *Ontwerp autosnelwegen : dwarsprofiel en alignement : mag het ietsje minder zijn? : consequenties van afwijkingen Richtlijnen Ontwerp Autosnelwegen (ROA)*, Tech. Rep. (Ministerie van Verkeer en Waterstaat, Rijkswaterstaat, Adviesdienst Verkeer en Vervoer (RWS, AVV), Rotterdam, 1994).
- [6] Grontmij Verkeer en Infrastructuur, *Beleidsverkenning Basiskwaliteit Autosnelwegen (BkA)*, Report IBW.05-0479 (Rijkswaterstaat Adviesdienst Verkeer en Vervoer, 2005).
- [7] TU Darmstadt, *Stromungsmechanische modellentwicklung*, [http://www.fdy.tu-darmstadt.de/forschung\\_16/forschungsgebiete/forschungsgebiete.de.jsp](http://www.fdy.tu-darmstadt.de/forschung_16/forschungsgebiete/forschungsgebiete.de.jsp), accessed: 2014-09-15.
- [8] G. K. Batchelor, *An introduction to fluid dynamics* (Cambridge university press, 2000).
- [9] G. Falkovich and K. R. Sreenivasan, *Lessons from hydrodynamic turbulence*, Tech. Rep. (Abdus Salam International Centre for Theoretical Physics, Trieste (Italy), 2006).
- [10] Merriam-Webster.com, *Turbulence*, <http://www.merriam-webster.com/dictionary/turbulence>, accessed: 2014-09-29.
- [11] *Highway capacity manual* (Transportation Research Board, National Research Council, Washington, D.C, 2000).
- [12] B. Goni Ros, V. L. Knoop, B. van Arem, and S. Hoogendoorn, *Car-following behavior at sags and its impacts on traffic flow*, in *Transportation Research Board 92nd Annual Meeting*, 13-1907 (2013).
- [13] H. Kolen, *Modelling merging behaviour on freeway on-ramps* (ITS Edulab, 2013).
- [14] V. L. Knoop, A. Duret, C. Buisson, and B. Van Arem, *Lane distribution of traffic near merging zones influence of variable speed limits*, in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on* (IEEE, 2010) pp. 485–490.
- [15] W. Daamen, M. Loot, and S. P. Hoogendoorn, *Empirical analysis of merging behavior at freeway on-ramp*, *Transportation Research Record: Journal of the Transportation Research Board* **2188**, 108 (2010).
- [16] M. M. Minderhoud, *Supported driving: impacts on motorway traffic flow*, (1999).
- [17] S. P. Hoogendoorn, *Traffic flow theory and simulation* (Delft University of Technology, 2013).
- [18] S. Hoogendoorn and P. Bovy, *New estimation technique for vehicle type-specific headway distributions*, *Transportation Research Record* **1646**, 18 (1998).

- [19] C. Choudhury, M. Ben-Akiva, T. Toledo, G. Lee, and A. Rao, *Modeling cooperative lane changing and forced merging behavior*, in *86th Annual Meeting of the Transportation Research Board, Washington, DC* (2007).
- [20] M. Heesen, M. Baumann, J. Kelsch, D. Nause, and M. Friedrich, *Investigation of cooperative driving behaviour during lane change in a multi-driver simulation environment*, *Human Factors: a view from an integrative perspective* (2012).
- [21] J. Wang, *A simulation model for motorway merging behaviour*, *Transportation and traffic theory* **16**, 281 (2005).
- [22] P. G. Gipps, *A behavioural car-following model for computer simulation*, *Transportation Research Part B: Methodological* **15**, 105 (1981).
- [23] W. J. Schakel, V. L. Knoop, and B. van Arem, *Integrated lane change model with relaxation and synchronization*, *Transportation Research Record: Journal of the Transportation Research Board* **2316**, 47 (2012).
- [24] J. Laval and L. Leclercq, *Microscopic modeling of the relaxation phenomenon using a macroscopic lane-changing model*, *Transportation Research Part B* **42**, 511 (2008).
- [25] S. Smith, *Freeway data collection for studying vehicle interaction*, Tech. Rep. FHWA/RD-85/108, FHWA (US, Department of Transportation, 1985).
- [26] B. Sultan, M. Brackstone, B. Waterson, and E. R. Boer, *Modeling the dynamic cut-in situation*, *Transportation Research Record: Journal of the Transportation Research Board* **1803**, 45 (2002).
- [27] B. Forgues, *Sampling on the dependent variable is not always that bad: Quantitative case-control designs for strategic organization research*, *Strategic Organization* **10**, 269 (2012).
- [28] C. J. Mann, *Observational research methods. research design ii: cohort, cross sectional, and case-control studies*, *Emergency Medicine Journal* **20**, 54 (2003).
- [29] C. Daganzo, *Fundamentals of transportation and traffic operations*, Vol. 30 (Pergamon Oxford, 1997).
- [30] S. Hoogendoorn, *Unified approach to estimating free speed distributions*, *Transportation Research Part B: Methodological* **39**, 709 (2005).
- [31] S. Hoogendoorn, H. J. Van Zuylen, M. Schreuder, B. Gorte, and G. Vosselman, *Microscopic traffic data collection by remote sensing*, *Transportation Research Record: Journal of the Transportation Research Board* **1855**, 121 (2003).
- [32] T. Maze, S. D. SCHROCK, and A. KAMYAB, *Capacity of freeway work zone lane closures*, work **6**, 12 (2000).
- [33] R. Voorrips, *Freeway work zone driving behaviour*, .
- [34] R. Hoogendoorn, *State-of-the-art driving theory and modeling in case of adverse conditions*, Tech. Rep. (Internal report, Delft University of Technology, Delft, 2010).
- [35] M. Rizzo, J. Jermeland, and J. Severson, *Instrumented vehicles and driving simulators*, *Geron-technology* **1**, 291 (2002).
- [36] U.S. Department of Transportation - Federal Highway Administration, *Chapter 2, traffic detector handbook: Third edition—volume i*, <http://www.fhwa.dot.gov/publications/research/operations/its/06108/02.cfm>, accessed: 2014-11-19.
- [37] M. M. R.J. Verheaghe and K. Lindveld, *Empirical data analysis* (Delft University of Technology).
- [38] F. J. Massey Jr, *The kolmogorov-smirnov test for goodness of fit*, *Journal of the American statistical Association* **46**, 68 (1951).



- [39] Fileradar BV, *Fileradar*, <http://www.dantesoftware.com/index.php/root/home-nl/fileradar/>, accessed: 2014-10-22.
- [40] The MathWorks, Inc., *Matlab*, <http://www.mathworks.nl/products/matlab/?refresh=true>, accessed: 2014-10-22.
- [41] H. Rakha and M. Van Aerde, *Statistical analysis of day-to-day variations in real-time traffic flow data*, Transportation research record , 26 (1995).
- [42] T. Bellemans, B. De Schutter, and B. De Moor, *Data acquisition, interfacing and pre-processing of highway traffic data*, Proceedings of Telematics Automotive , 4 (2000).
- [43] M. Agarwal, T. H. Maze, and R. Souleyrette, *Impacts of weather on urban freeway traffic flow characteristics and facility capacity*, in *Proceedings of the 2005 mid-continent transportation research symposium* (2005).
- [44] KNMI, *April 2013: koud, droog en vrij zonnig*. [http://www.knmi.nl/klimatologie/maand\\_en\\_seizoensoverzichten/maand/apr13.html](http://www.knmi.nl/klimatologie/maand_en_seizoensoverzichten/maand/apr13.html) (), accessed: 2014-10-22.
- [45] KNMI, *April 2014: Zeer zacht, gemiddeld over het land normale hoeveelheid zon en neerslag*, [http://www.knmi.nl/klimatologie/maand\\_en\\_seizoensoverzichten/maand/apr14.html](http://www.knmi.nl/klimatologie/maand_en_seizoensoverzichten/maand/apr14.html) (), accessed: 2014-10-22.





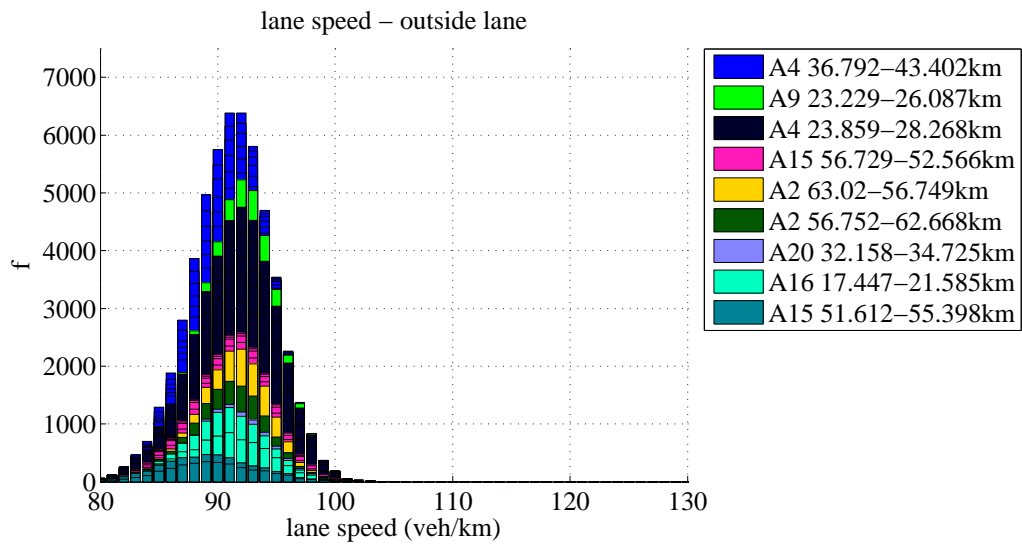
## Fixed average speed check

This appendix shows the effects of fixed average speed check location data. The differences between locations with and without are investigated by comparing the results of the combined data sets on basic freeway segments with and without locations with fixed average speed check included.

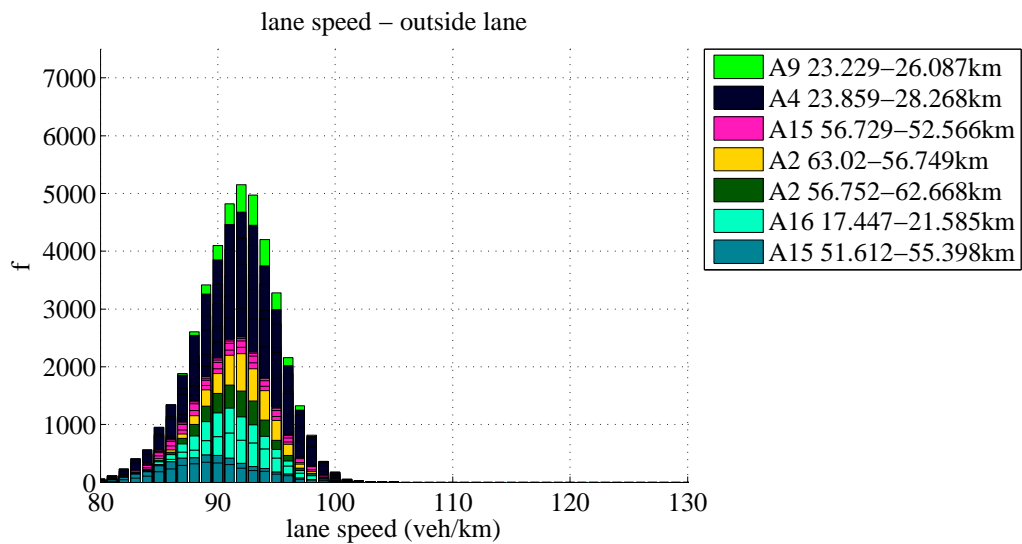
### **A.1.** Lane speed

A large difference on the median lane is found, smaller difference on the center lane and no difference on the outside lane.

### A.1.1. Outside lane



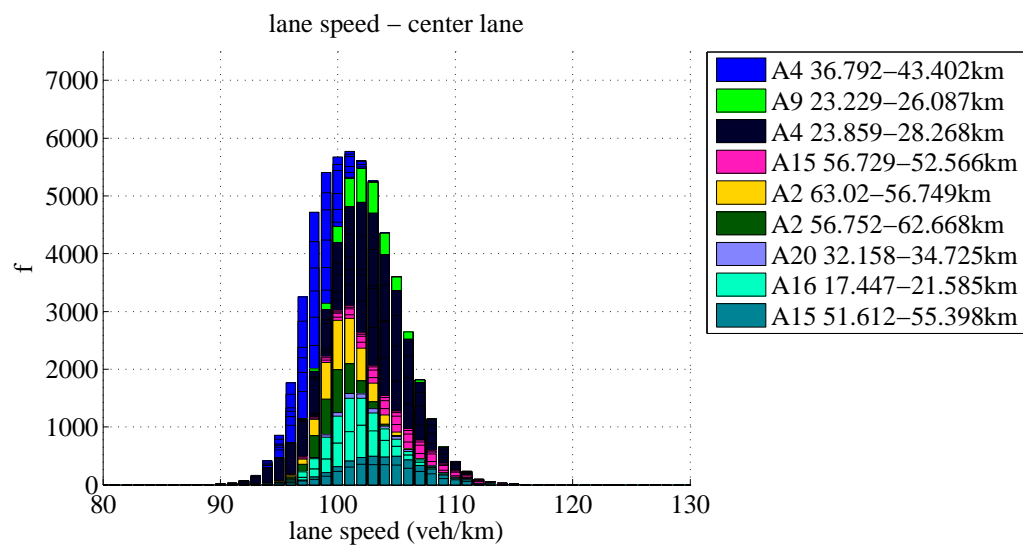
(a) Outside lane with fixed average speed check



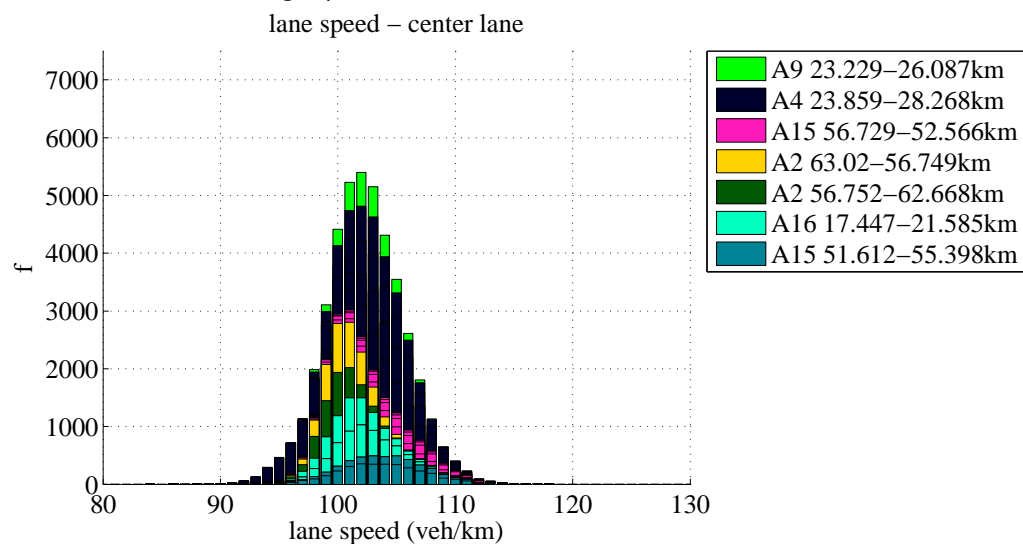
(b) Outside lane without fixed average speed check

Figure A.1: Lane speeds comparison outside lane

## A.1.2. Center lane



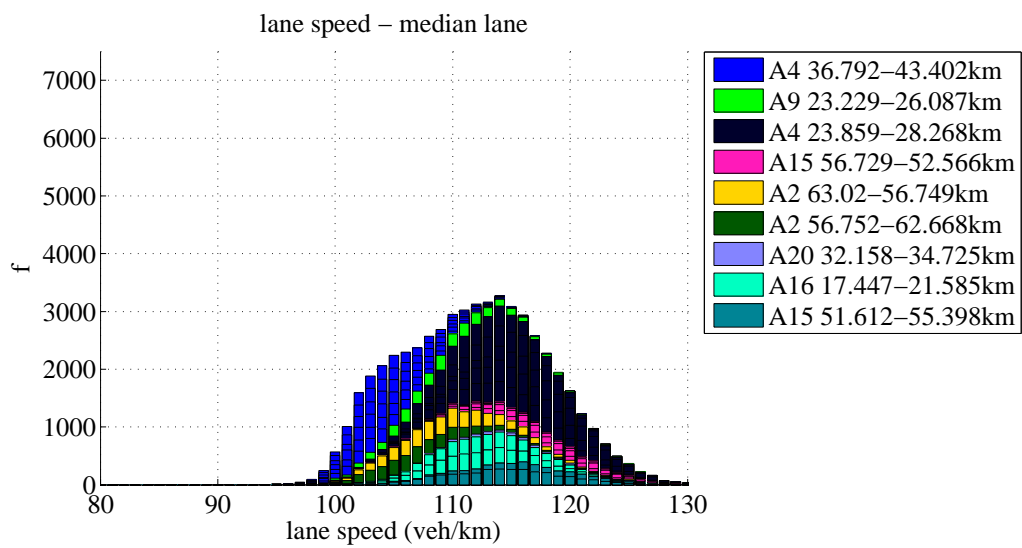
(a) Center lane with fixed average speed check



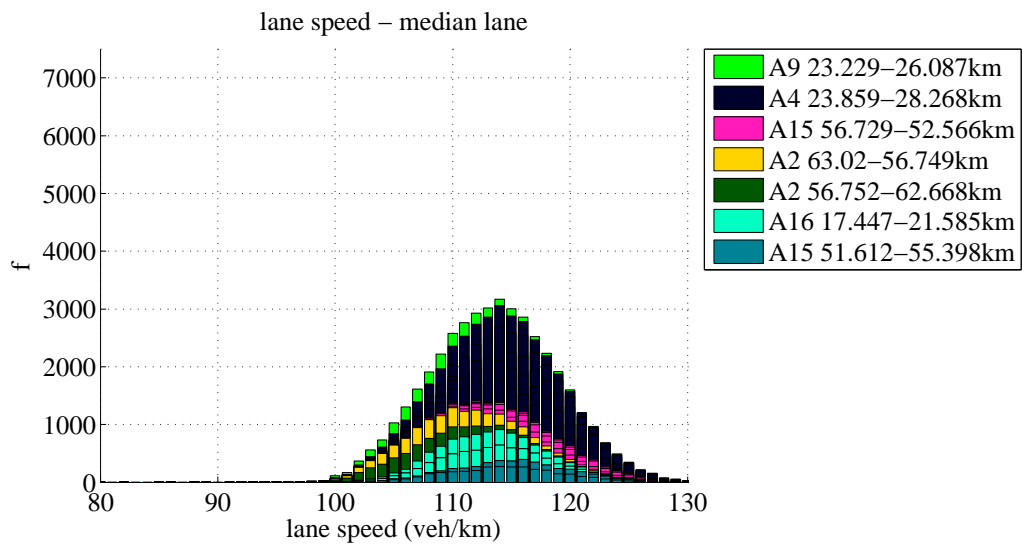
(b) Center lane without fixed average speed check

Figure A.2: Lane speeds comparison center lane

### A.1.3. Median lane



(a) Median lane with fixed average speed check

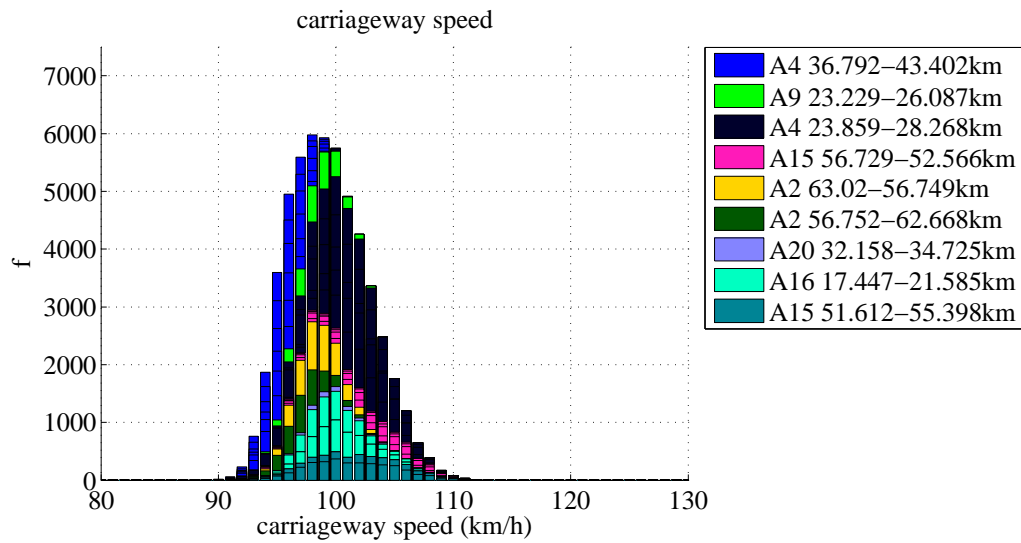


(b) Median lane without fixed average speed check

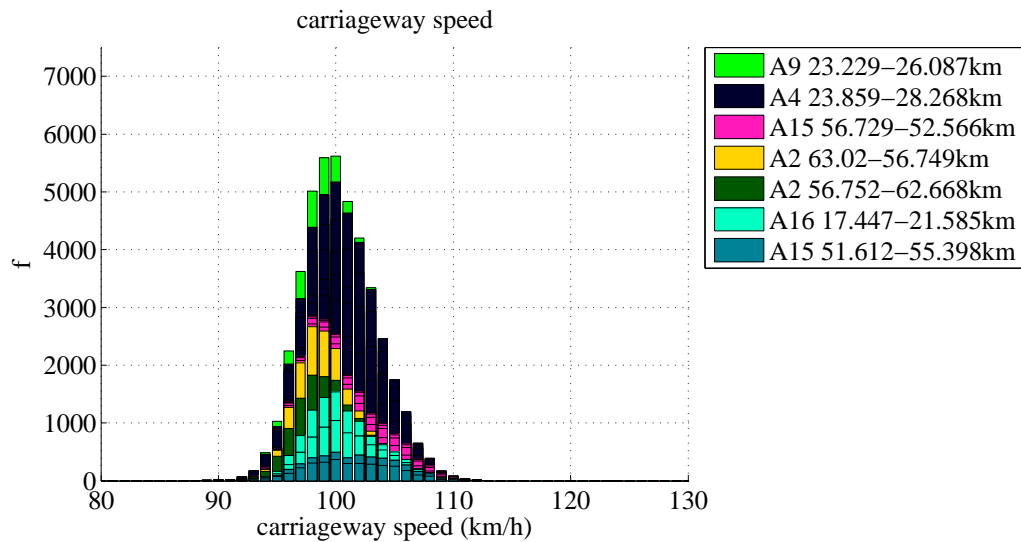
Figure A.3: Lane speeds comparison median lane

## A.2. Average speed

The comparison shows that the average speed on these locations is lower than other locations.



(a) With fixed average speed check



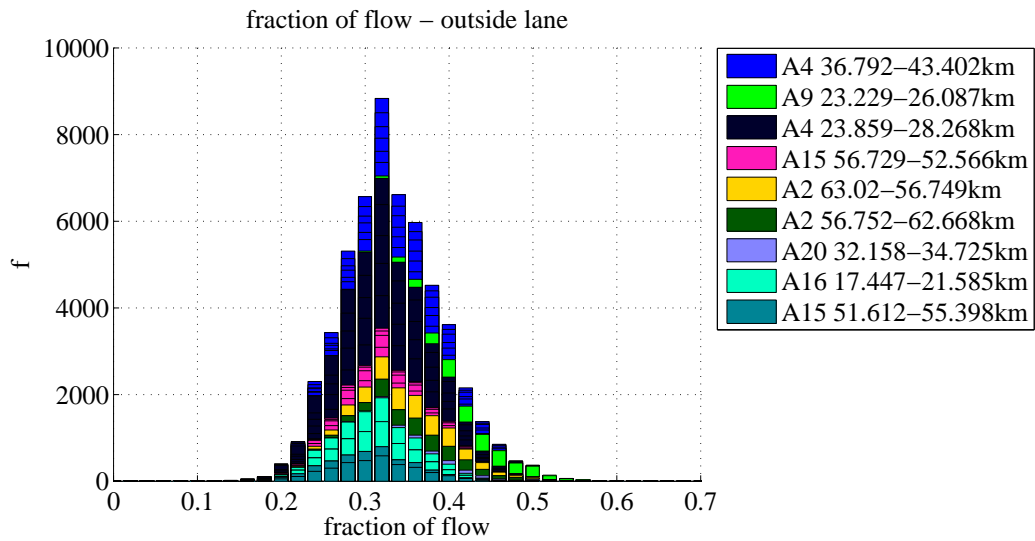
(b) Without fixed average speed check

Figure A.4: Average speeds comparison

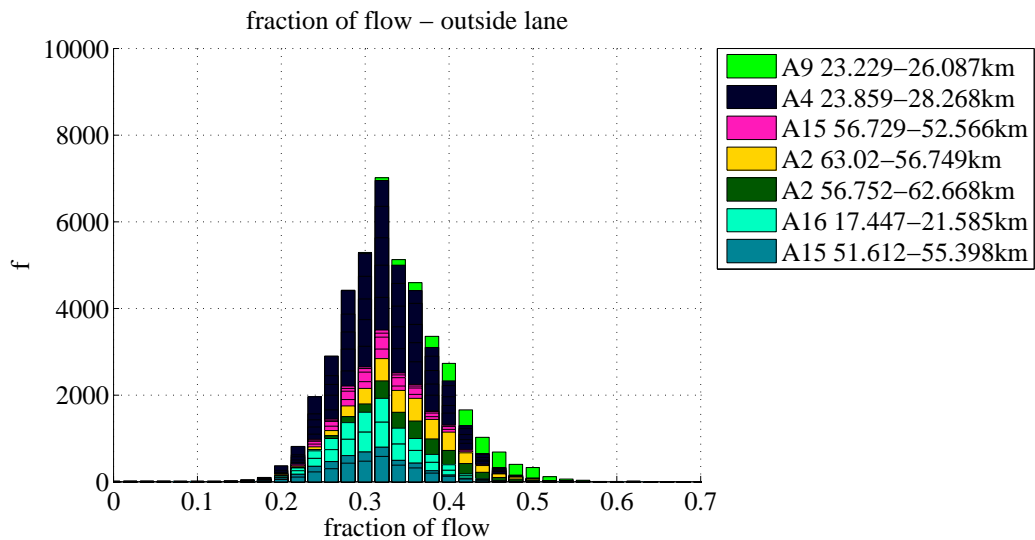
### A.3. Percentage flow

The lane flow distributions shows that more traffic is driving on the right lanes.

#### A.3.1. Outside lane



(a) Outside lane with fixed average speed check

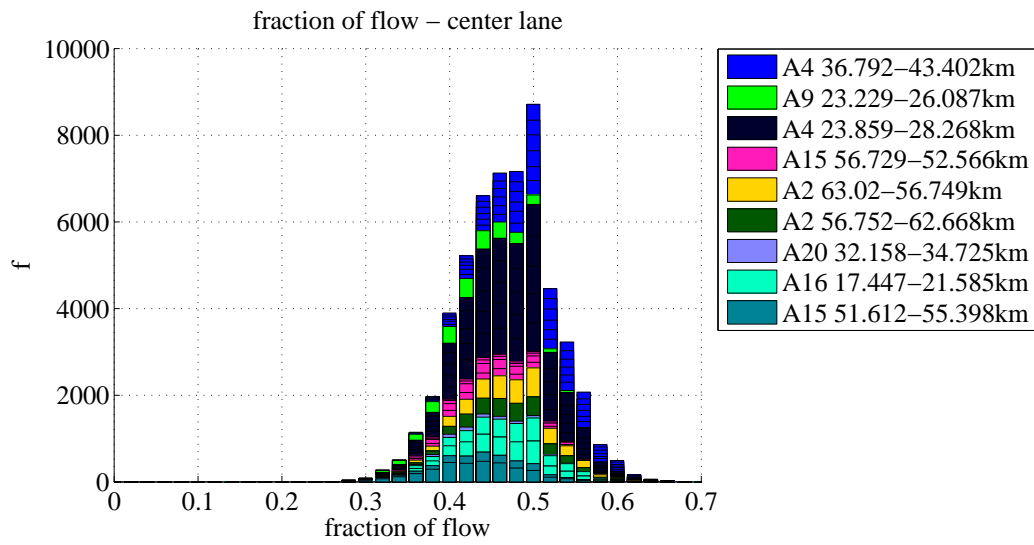


(b) Outside lane without fixed average speed check

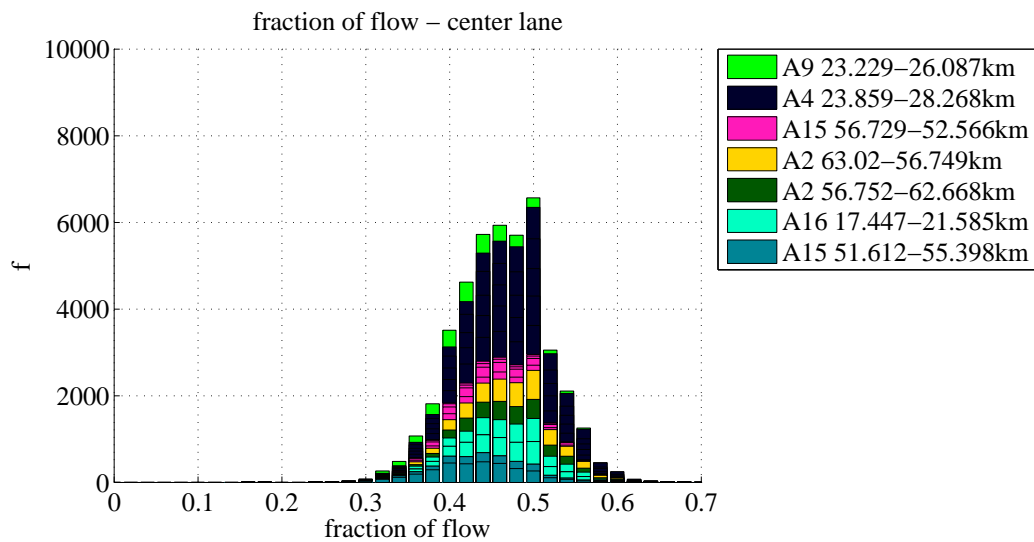
Figure A.5: Fraction of flow comparison outside lane



### A.3.2. Center lane



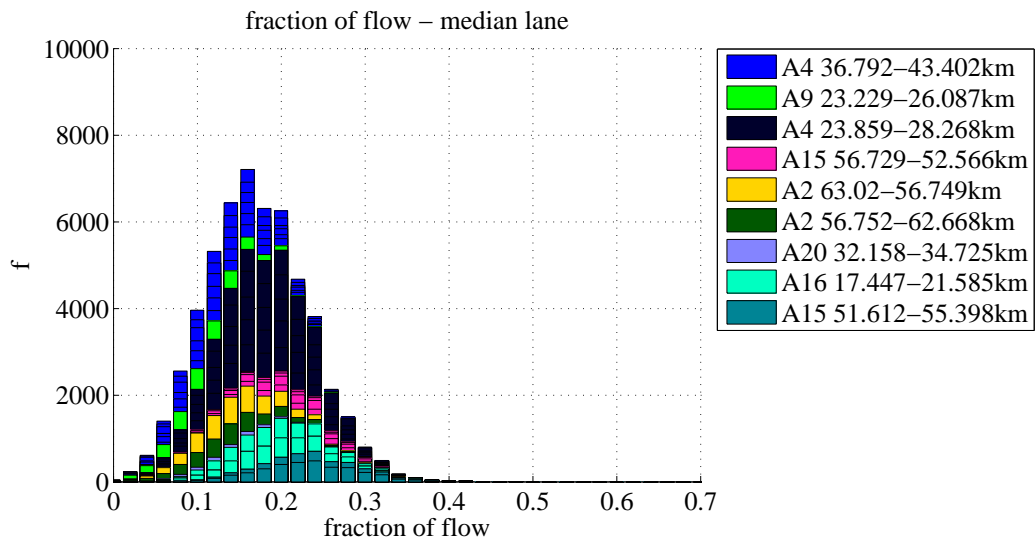
(a) Center lane with fixed average speed check



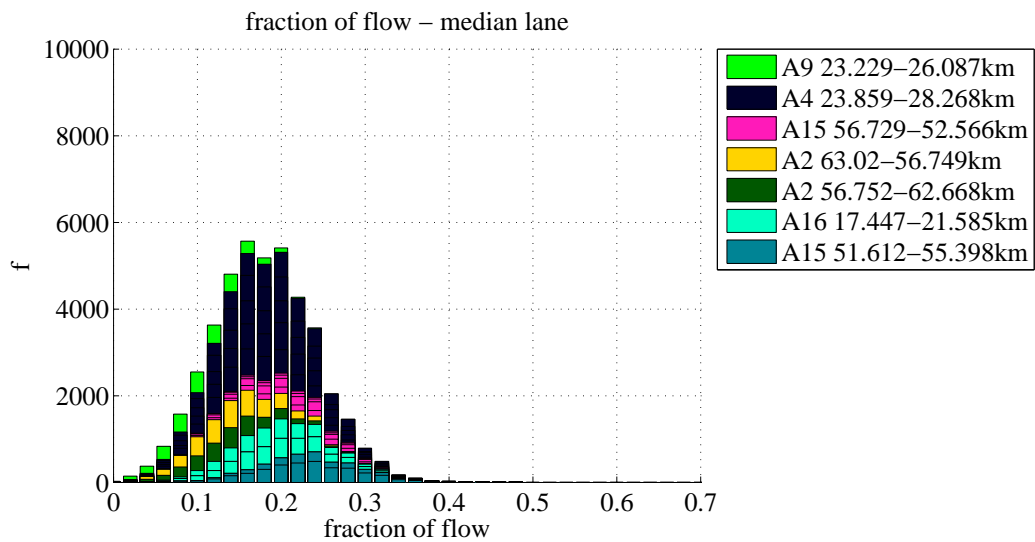
(b) Center lane without fixed average speed check

Figure A.6: Fraction of flow comparison center lane

### A.3.3. Median lane



(a) Median lane with fixed average speed check



(b) Median lane without fixed average speed check

Figure A.7: Fraction of flow comparison median lane

# B

## Performance maximum likelihood estimations

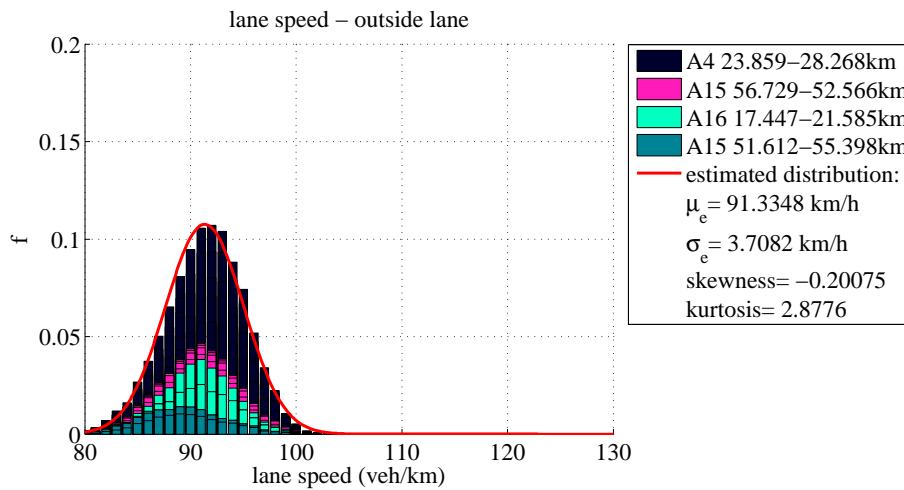
This appendix gives insight into the performance of the maximum likelihood estimation on the acceleration lanes and basic freeway segments. To give insight the estimation with the maximum likelihood is compared with the raw data. The results of this comparison are shown of the identified indicators (lane density, lane speed, average speed and lane flow distribution) and the other analysis, the total flow. For the acceleration lane the comparison is shown of the different intervals.

### B.1. Lane speed

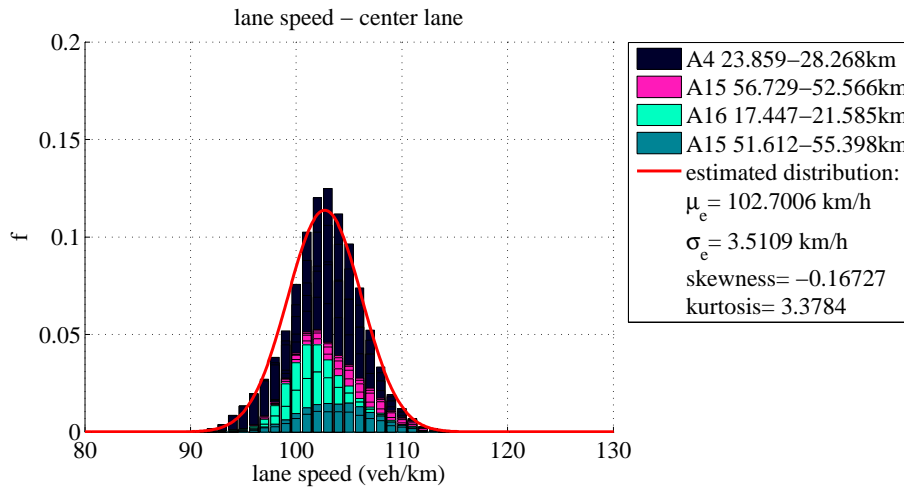
In figures B.1 till B.6 the comparison of the lane speeds are shown for the acceleration lanes and basic freeway segment.

For the basic freeway segment and the acceleration lane both estimations seems to follow the raw data accurate. However, two main findings are found. In the interval (-600)-(-300) meter only one detector is available to estimate the average behavior. This means, due to the availability of only one detector, the estimated distribution is simply the behavior of this one detector and not the average behavior of the detectors in this interval. This need to be taken into account in the analyses. The second finding is the large difference between the estimations on the outside lane and center lane in the interval during the acceleration lane, interval (0)-(300) meter. Without theses exceptions, the estimation with the maximum likelihood are closely related with the raw data.

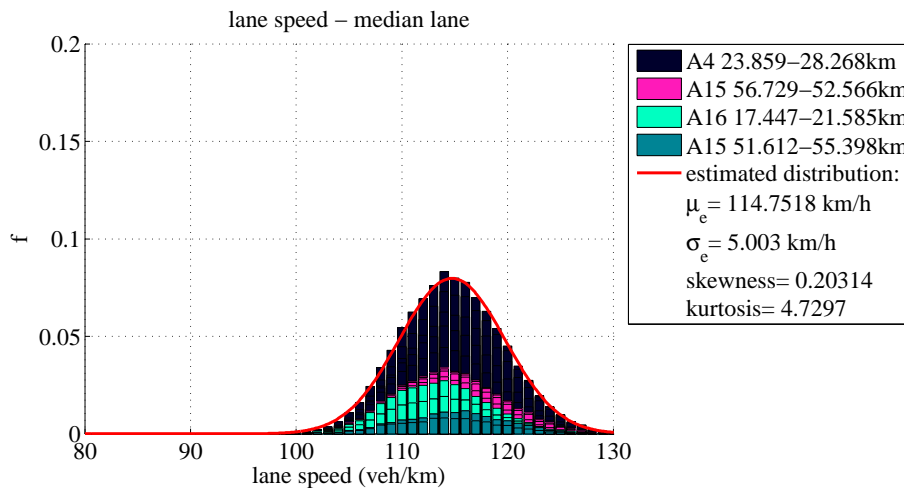
### B.1.1. Basic freeway segment



(a) Outside lane speed histogram and estimated distribution



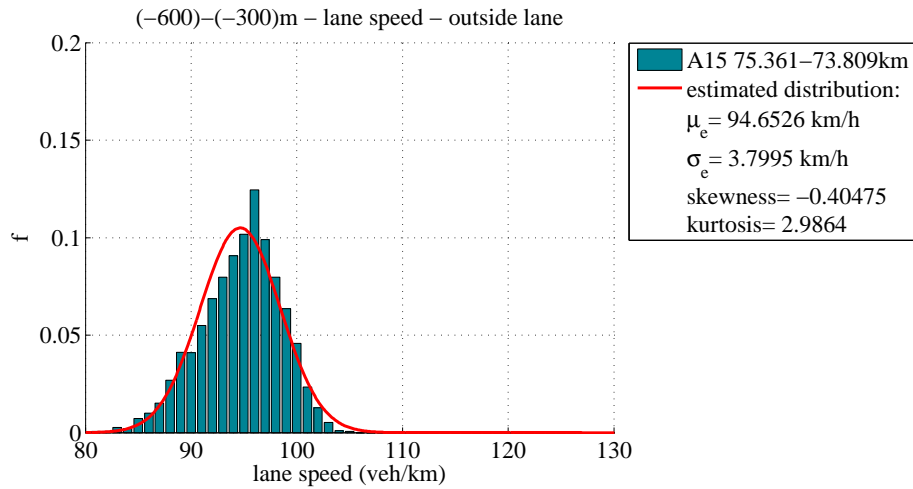
(b) Center lane speed histogram and estimated distribution



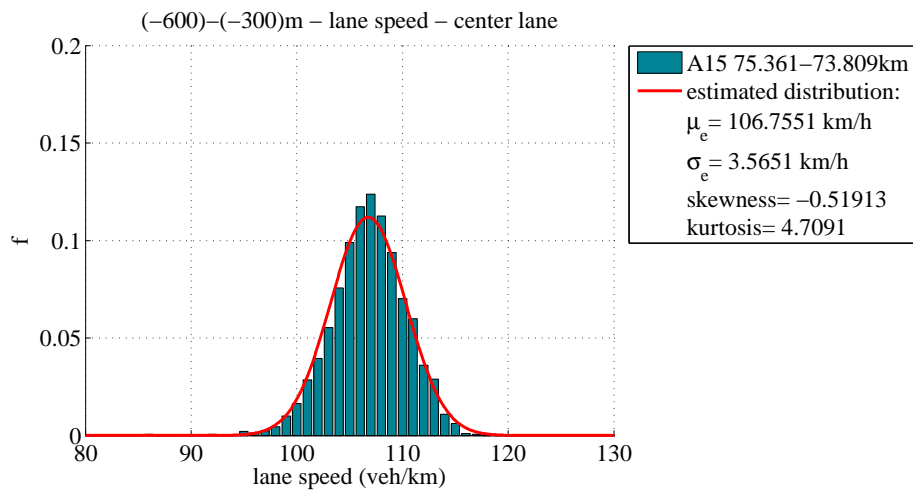
(c) Median lane speed histogram and estimated distribution

Figure B.1: Comparison histogram raw data with the maximum likelihood estimation on the basic freeway segment

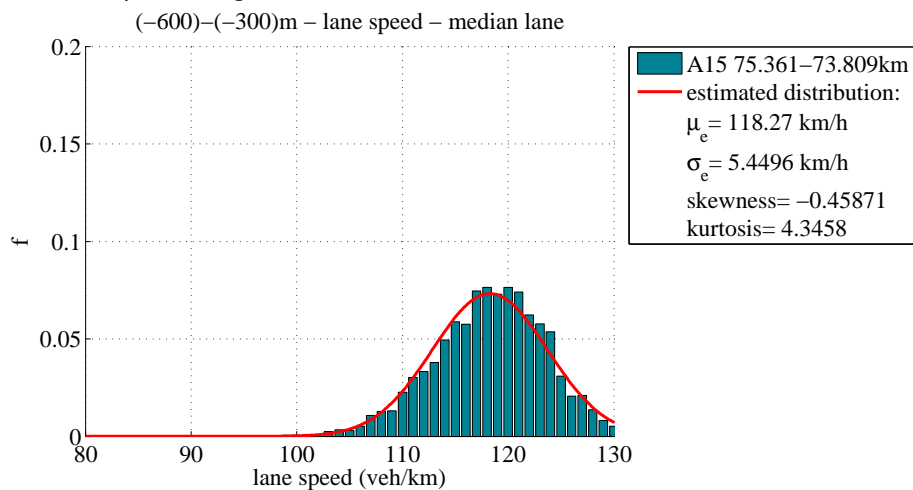
### B.1.2. Acceleration lane (-600)-(-300) meter interval



(a) Outside lane speed histogram and estimated distribution



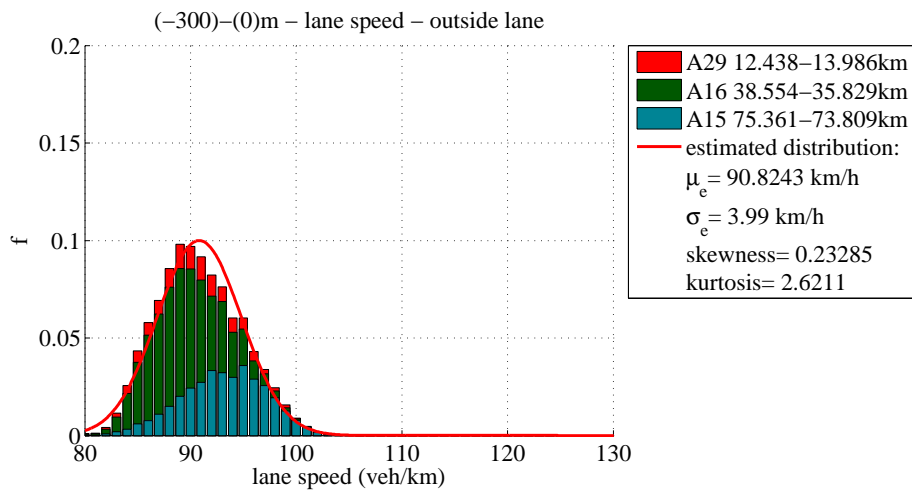
(b) Center lane speed histogram and estimated distribution



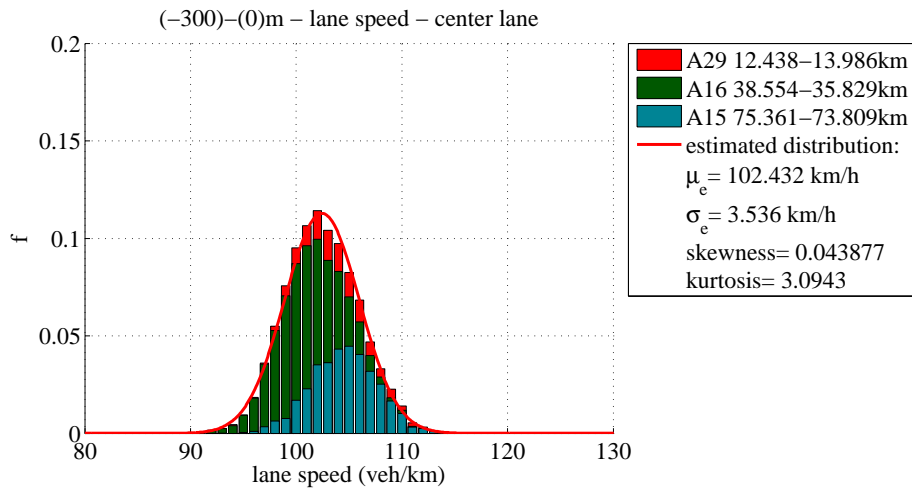
(c) Median lane speed histogram and estimated distribution

Figure B.2: Comparison histogram raw data with the maximum likelihood estimation in interval (-600)-(-300)m

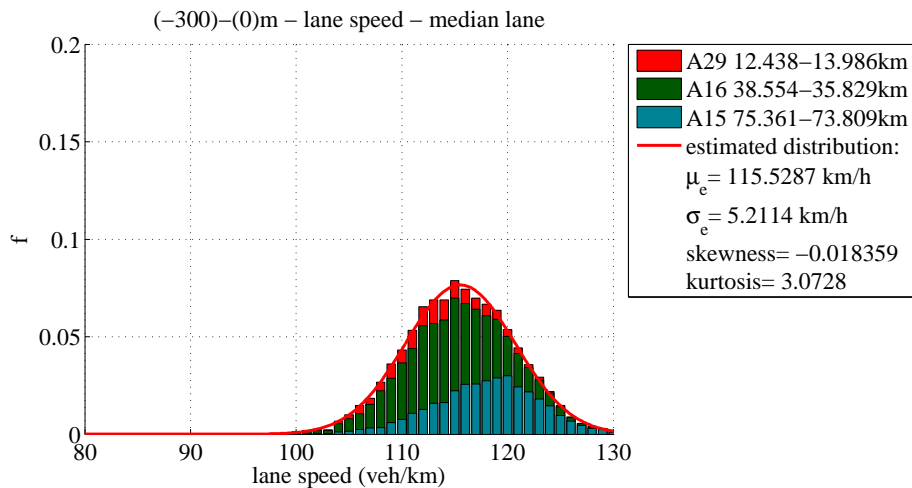
## (-300)-(0) meter interval



(a) Outside lane speed histogram and estimated distribution



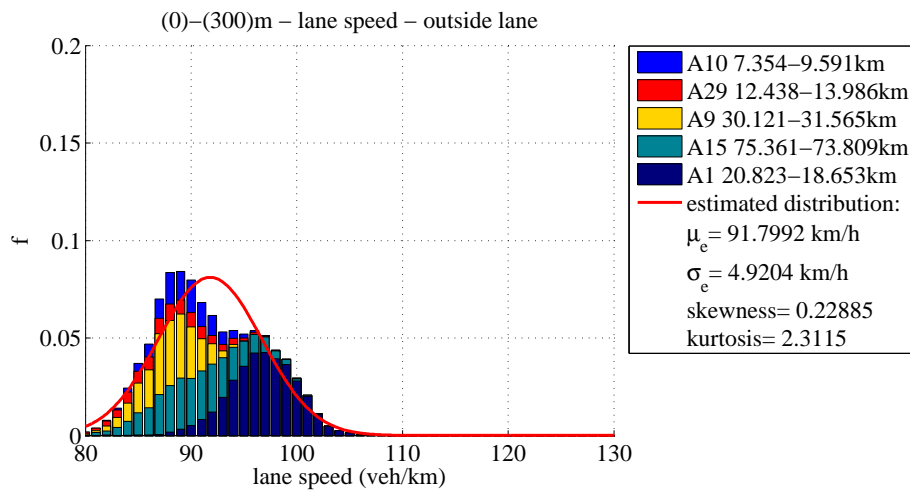
(b) Center lane speed histogram and estimated distribution



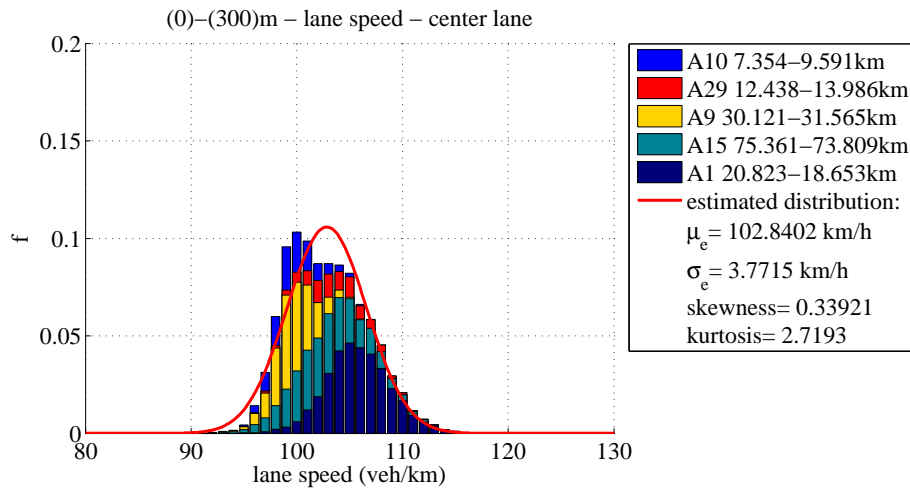
(c) Median lane speed histogram and estimated distribution

Figure B.3: Comparison histogram raw data with the maximum likelihood estimation in interval (-300)-(0)m

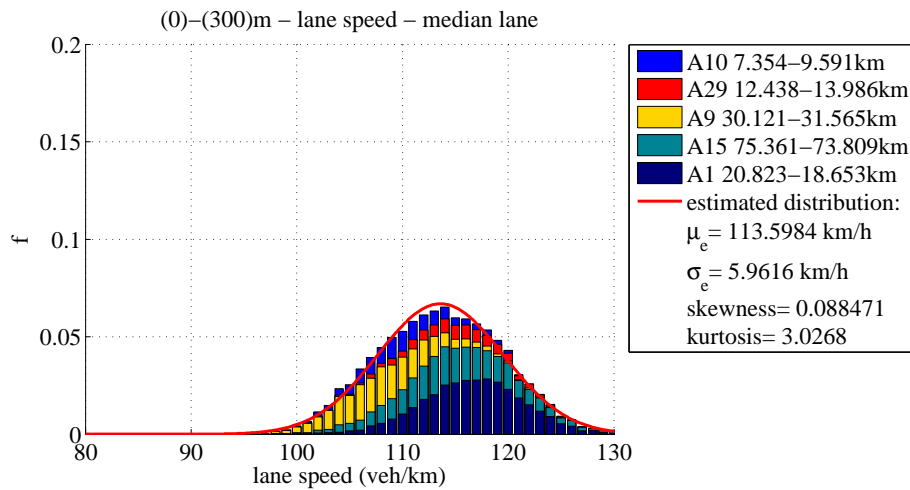
## (0)-(300) meter interval



(a) Outside lane speed histogram and estimated distribution



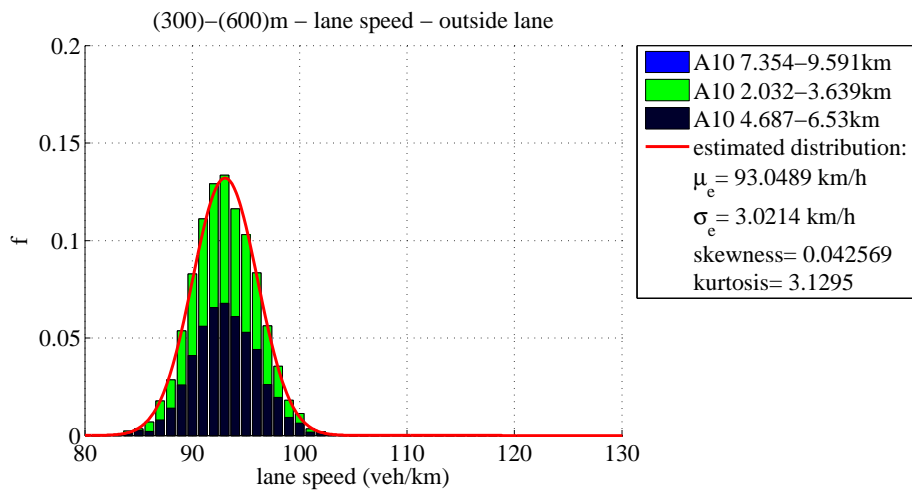
(b) Center lane speed histogram and estimated distribution



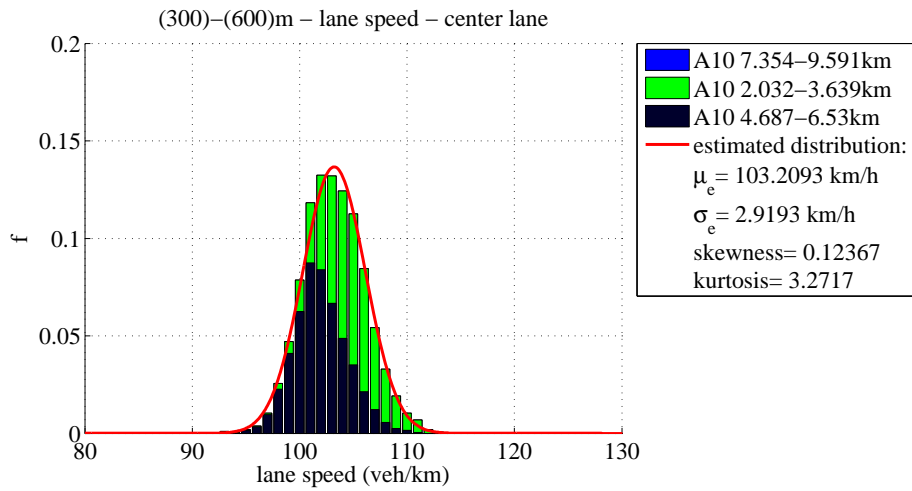
(c) Median lane speed histogram and estimated distribution

Figure B.4: Comparison histogram raw data with the maximum likelihood estimation in interval (0)-(300)m

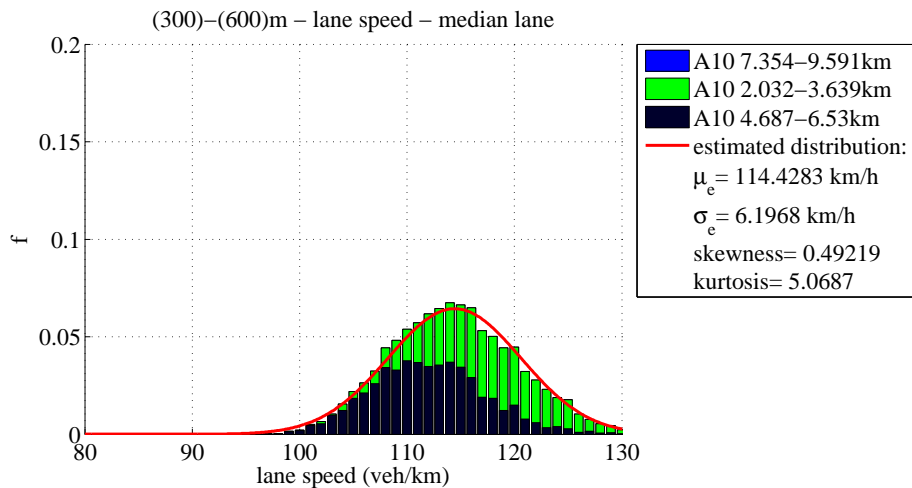
## (300)-(600) meter interval



(a) Outside lane speed histogram and estimated distribution



(b) Center lane speed histogram and estimated distribution

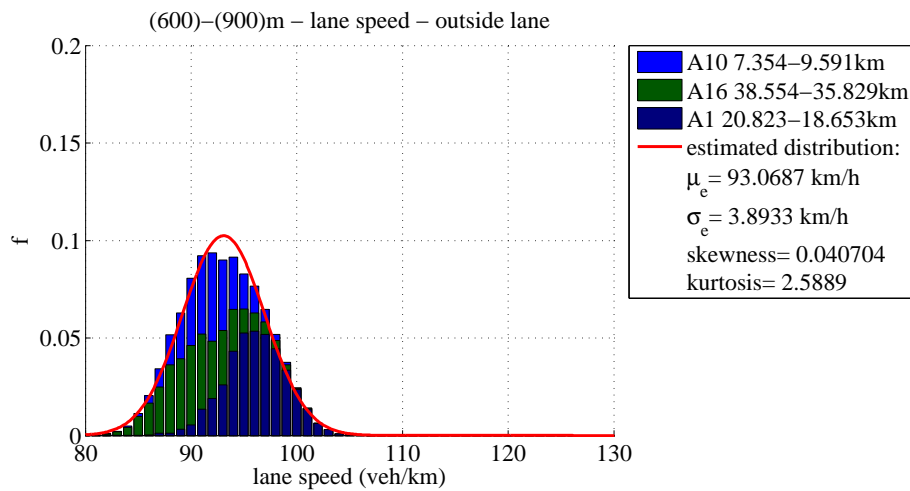


(c) Median lane speed histogram and estimated distribution

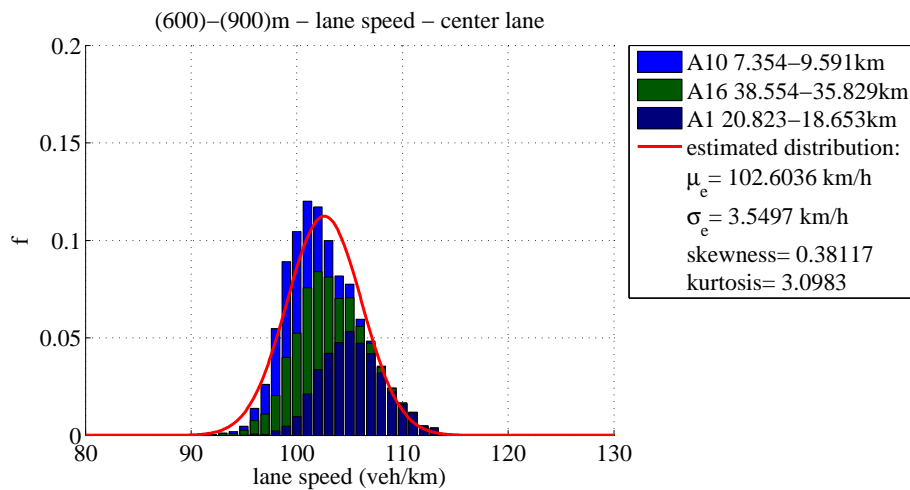
Figure B.5: Comparison histogram raw data with the maximum likelihood estimation in interval (300)-(600)m



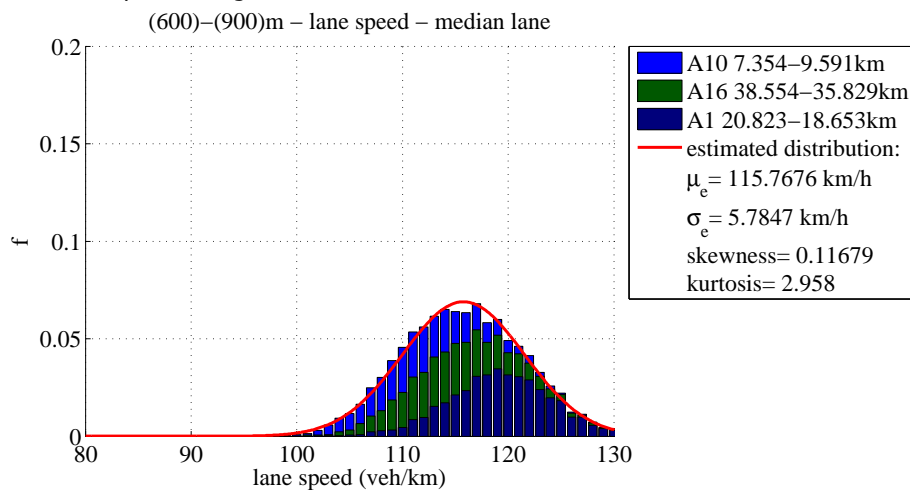
## (600)-(900) meter interval



(a) Outside lane speed histogram and estimated distribution



(b) Center lane speed histogram and estimated distribution



(c) Median lane speed histogram and estimated distribution

Figure B.6: Comparison histogram raw data with the maximum likelihood estimation in interval (900)-(600)m

## B.2. Average speed

Figures B.7 till B.12 the comparisons between the maximum likelihood estimation and the raw data are shown of the average speed indicator. The raw data of the basic freeway segment is slightly skewed to the left. The raw data exist of four locations with multiple detectors per location. The first interval of the acceleration lane is again based on one detector. Further on acceleration lanes the interval during the acceleration lane (0)-(300) meter the raw data consists of a double peak. This difference seems to origin from the position of the detector within this interval (see figure reffig:newpercentageoutside for an impression of the position). This effect was already also noticed in the lane speeds. As last the last interval (600)-(900) meter seems to differ more from the estimation than in the lane speeds. On average, the differences aren't large of the estimation in comparison with the raw data.

### B.2.1. Basic freeway segment

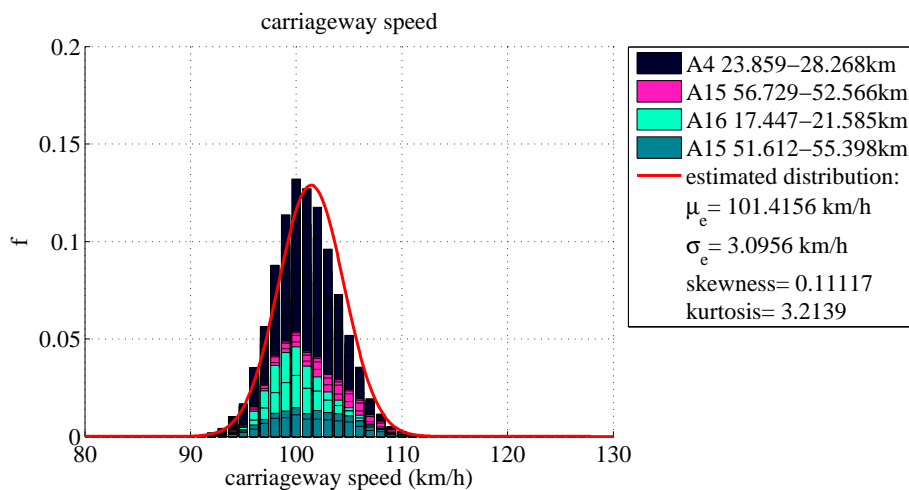


Figure B.7: Comparison histogram raw data with the maximum likelihood estimation of the basic freeway segment

### B.2.2. Acceleration lane (-600)-(-300) meter interval

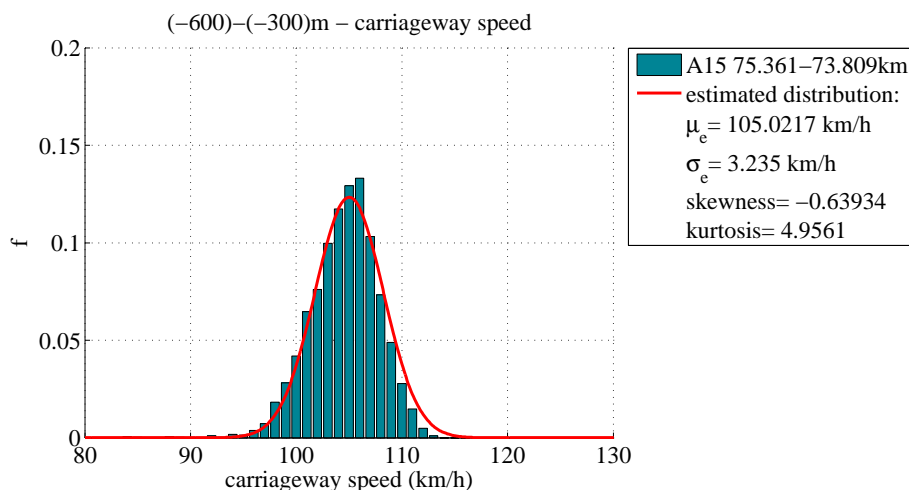


Figure B.8: Comparison histogram raw data with the maximum likelihood estimation in interval (-600)-(-300)m

### (-300)-(0) meter interval

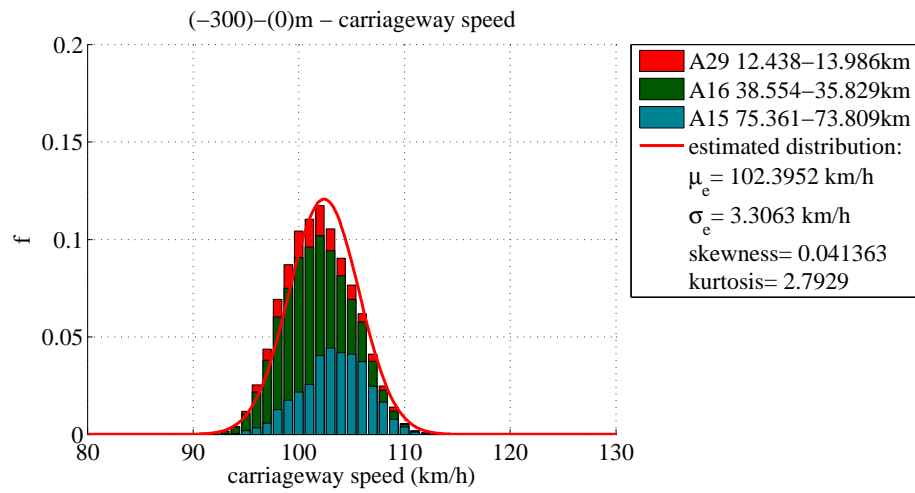


Figure B.9: Comparison histogram raw data with the maximum likelihood estimation in interval (-300)-(0)m

## (0)-(300) meter interval

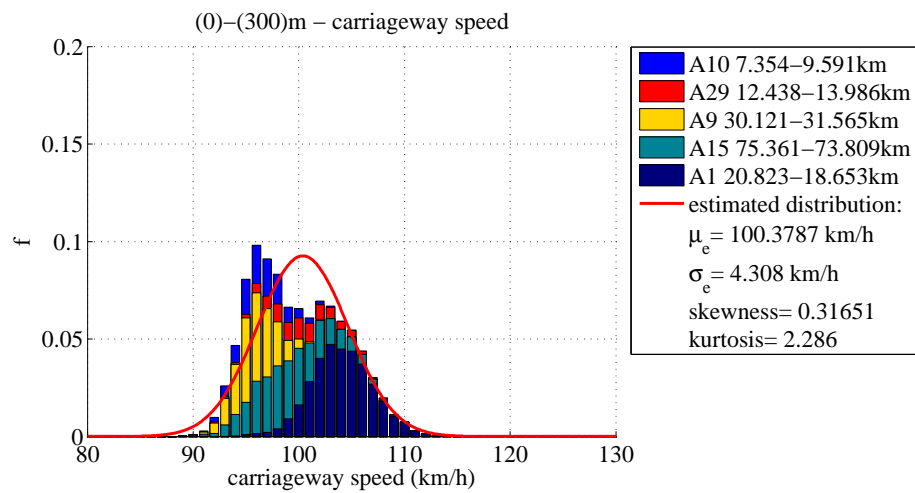


Figure B.10: Comparison histogram raw data with the maximum likelihood estimation in interval (0)-(300)m

## (300)-(600) meter interval

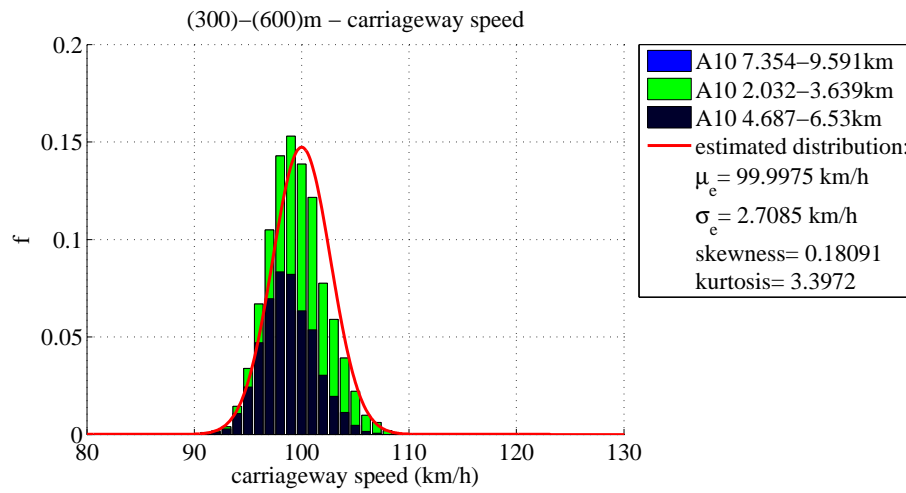


Figure B.11: Comparison histogram raw data with the maximum likelihood estimation in interval (300)–(600)m

## (600)–(900) meter interval

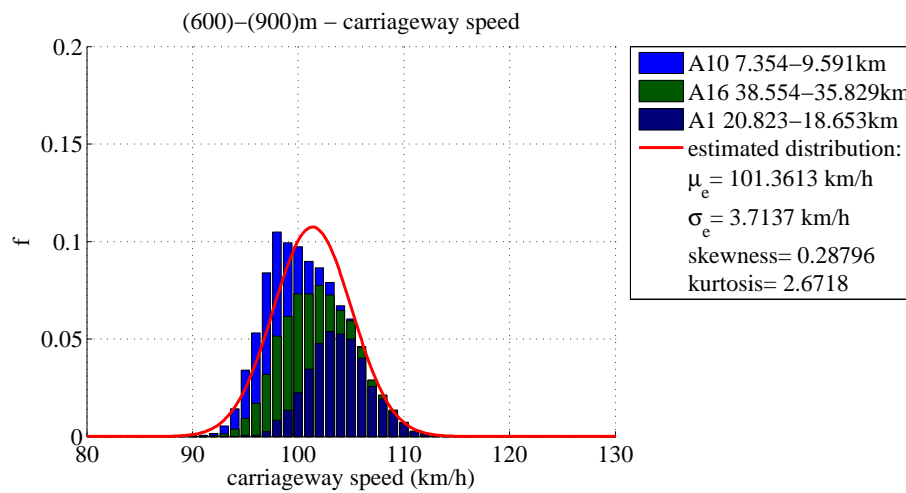
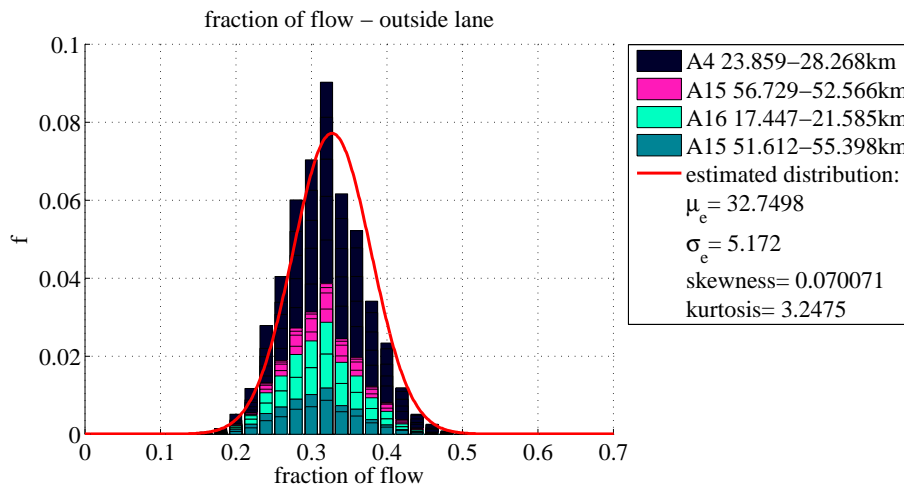


Figure B.12: Comparison histogram raw data with the maximum likelihood estimation in interval (600)–(900)m

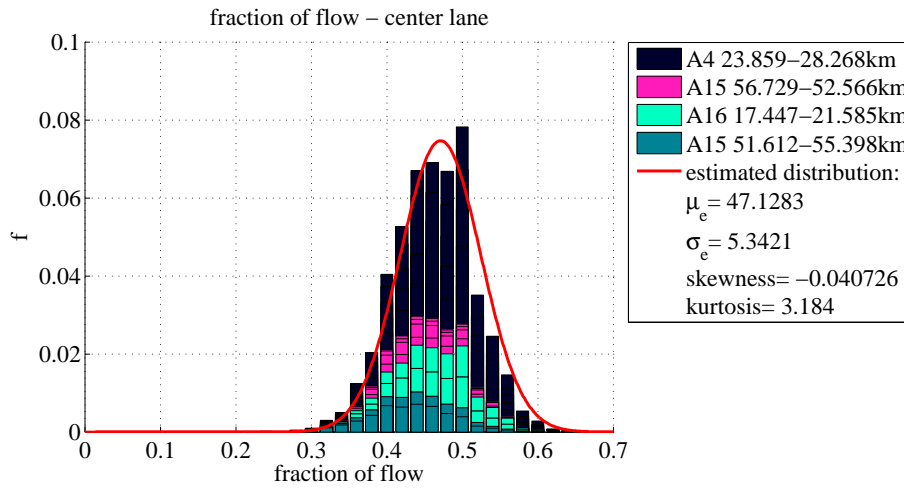
### B.3. Lane flow distributions

In figures B.13 till B.18 the results of the comparison are shown. The raw data of the detectors is less smoothly distributed than of the average or lane speeds. A reason for this effect is the calculation of the flows. Detectors measure the number of vehicles per time period. In this thesis is data per minute used for the analysis. The flow variable is usually presented as the number of vehicles per hour. To achieve the number of vehicles per hour with the used data the number of vehicles passing in a minute is multiplied times sixty. This results in steps in the data by 60 veh/h. In example, when one vehicle was measured in one minute, the resulting flow is 60 veh/h, when two vehicles were measured in one minute, the resulting flow is 120 veh/h. For the presentation of this data in a histogram some bins get more data than others. In total, the estimations with the maximum likelihood are closely related with the raw data.

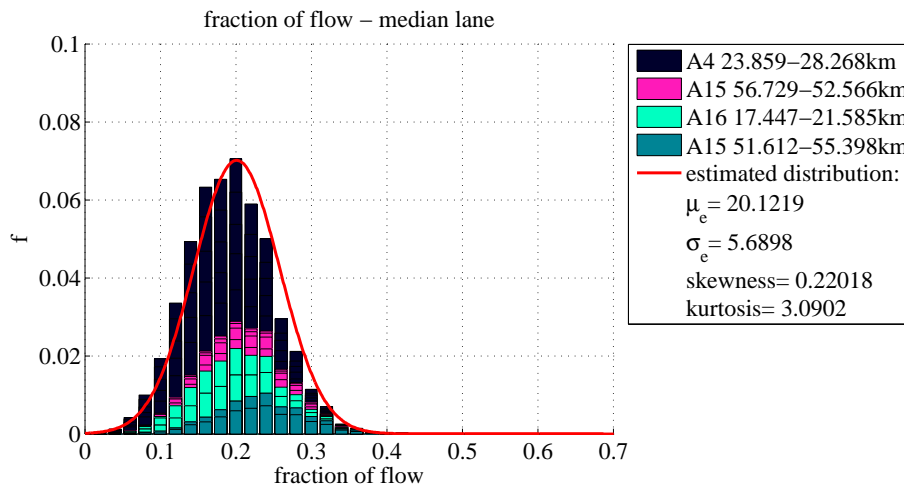
### B.3.1. Basic freeway segment



(a) Outside lane speed histogram and estimated distribution



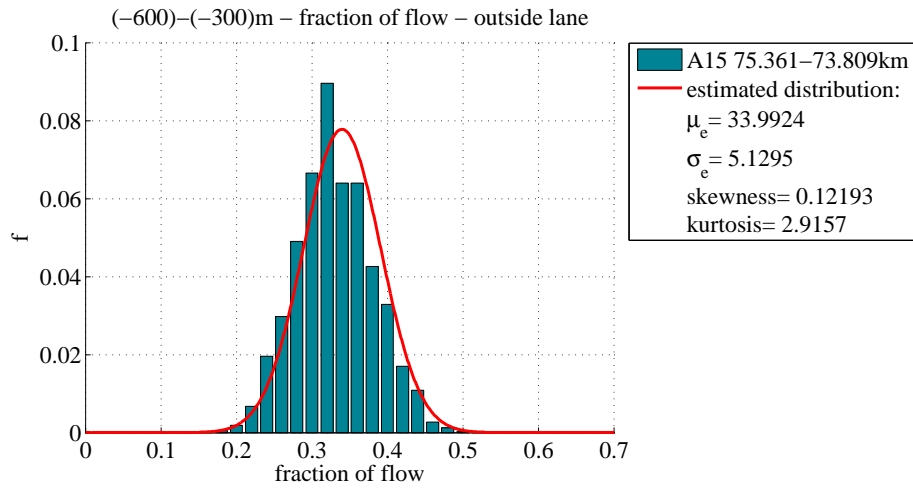
(b) Center lane speed histogram and estimated distribution



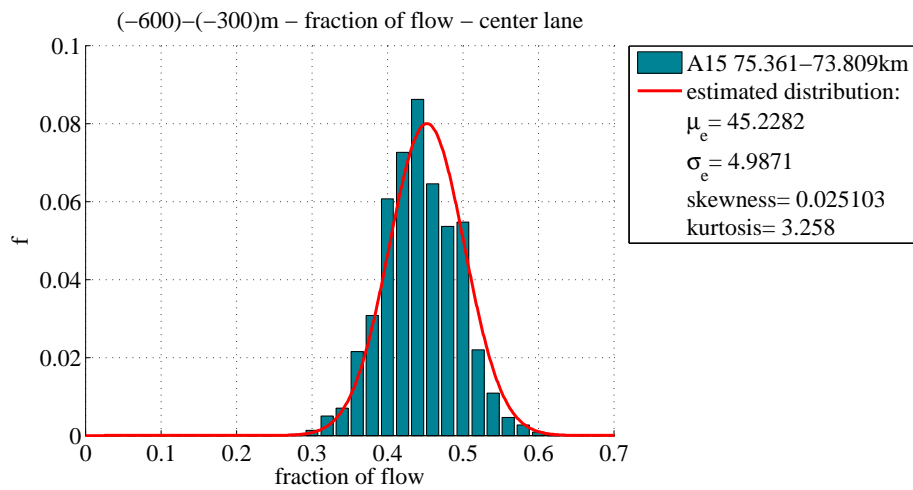
(c) Median lane speed histogram and estimated distribution

Figure B.13: Comparison histogram raw data with the maximum likelihood estimation in basic freeway segment

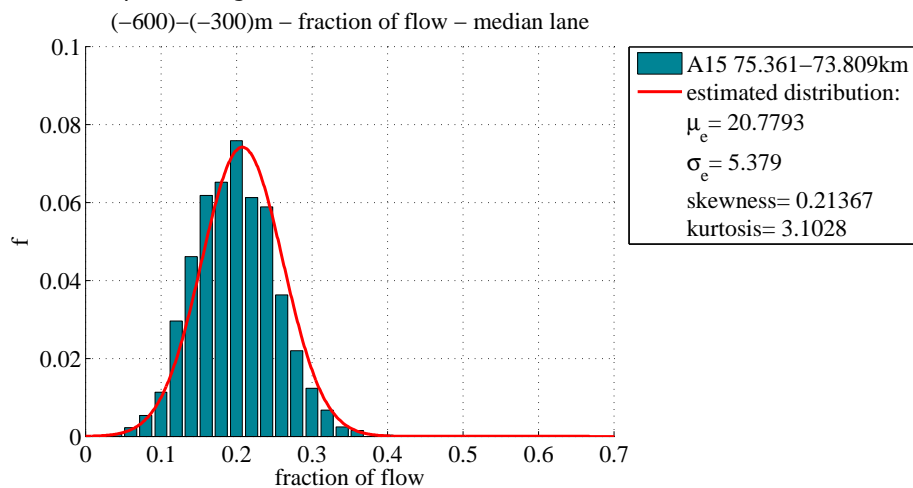
### B.3.2. Acceleration lane (-600)-(-300) meter interval



(a) Outside lane speed histogram and estimated distribution



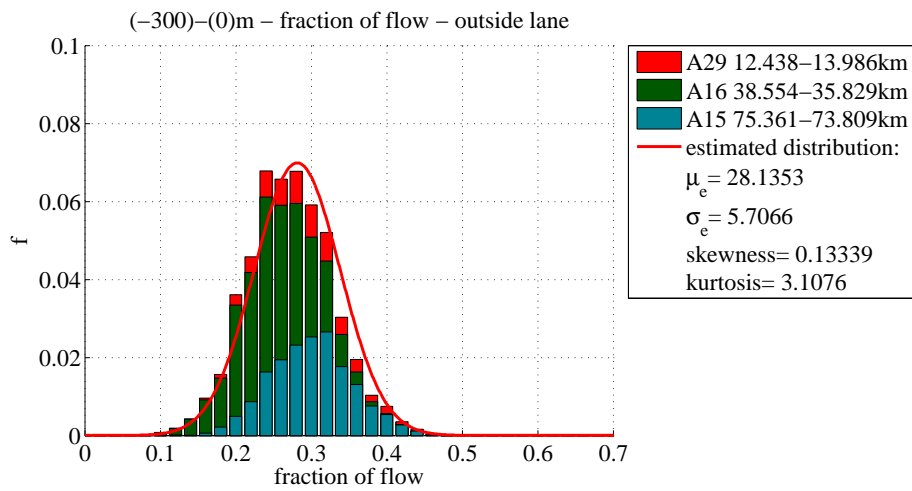
(b) Center lane speed histogram and estimated distribution



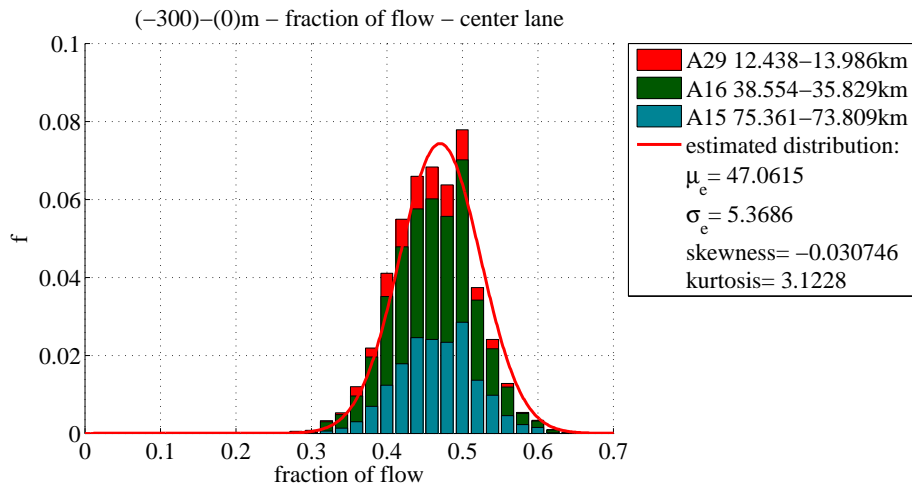
(c) Median lane speed histogram and estimated distribution

Figure B.14: Comparison histogram raw data with the maximum likelihood estimation in interval (-600)-(-300)m

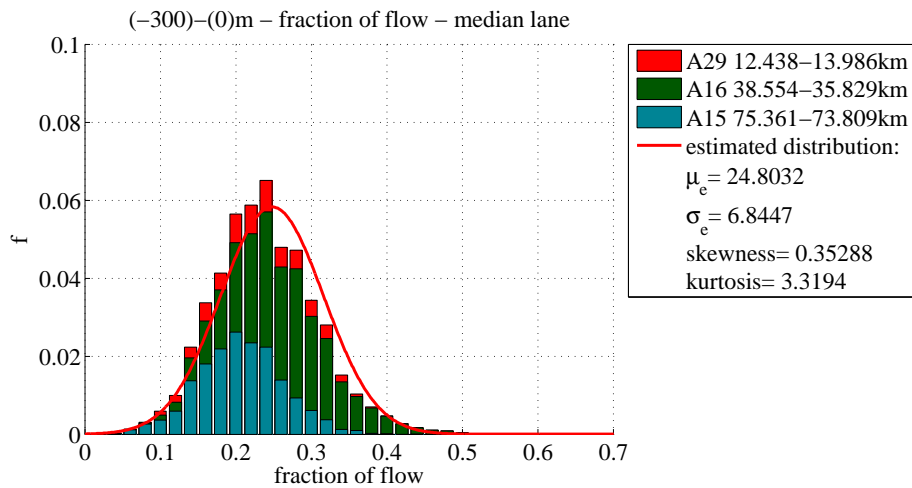
## (-300)-(0) meter interval



(a) Outside lane speed histogram and estimated distribution



(b) Center lane speed histogram and estimated distribution

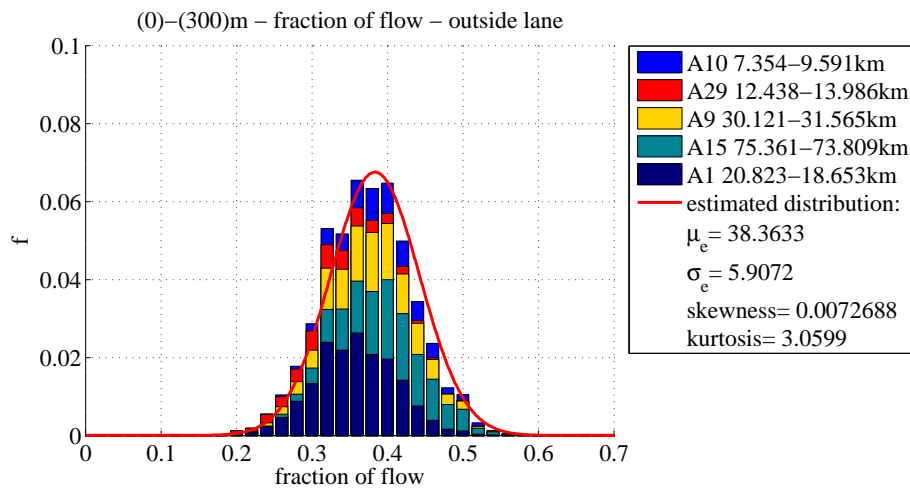


(c) Median lane speed histogram and estimated distribution

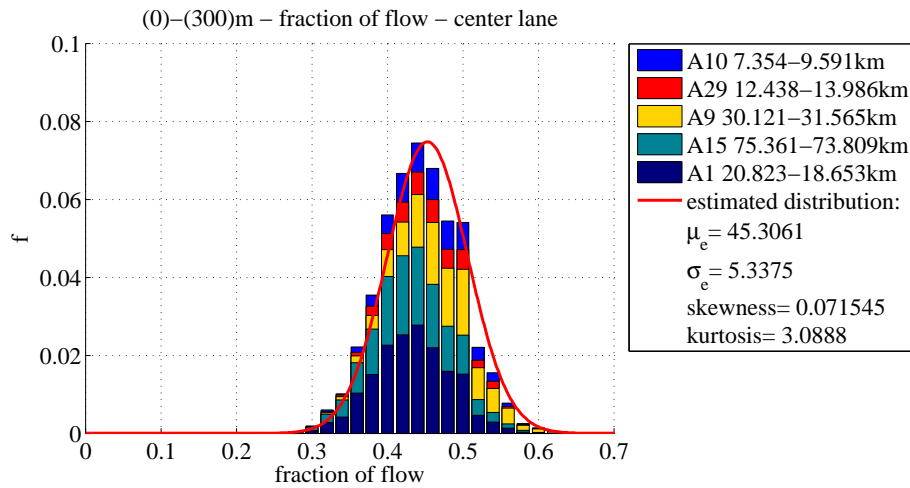
Figure B.15: Comparison histogram raw data with the maximum likelihood estimation in interval (-300)-(0)m



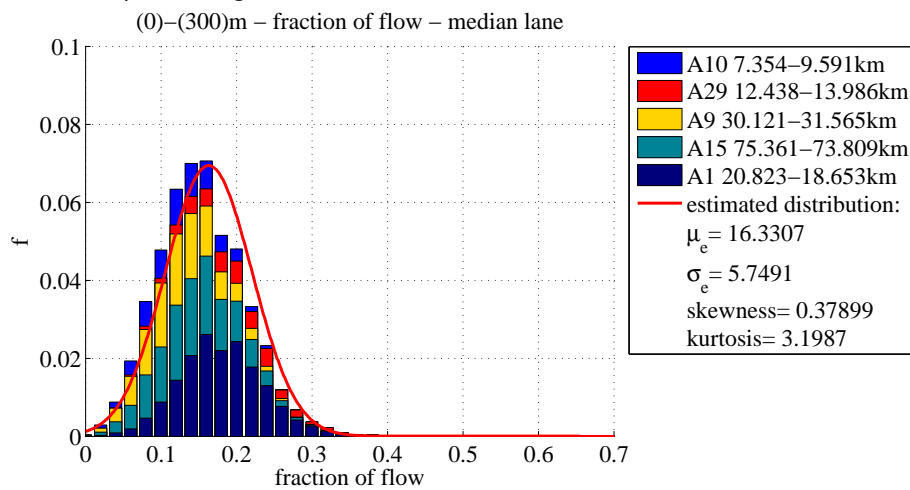
## (0)-(300) meter interval



(a) Outside lane speed histogram and estimated distribution



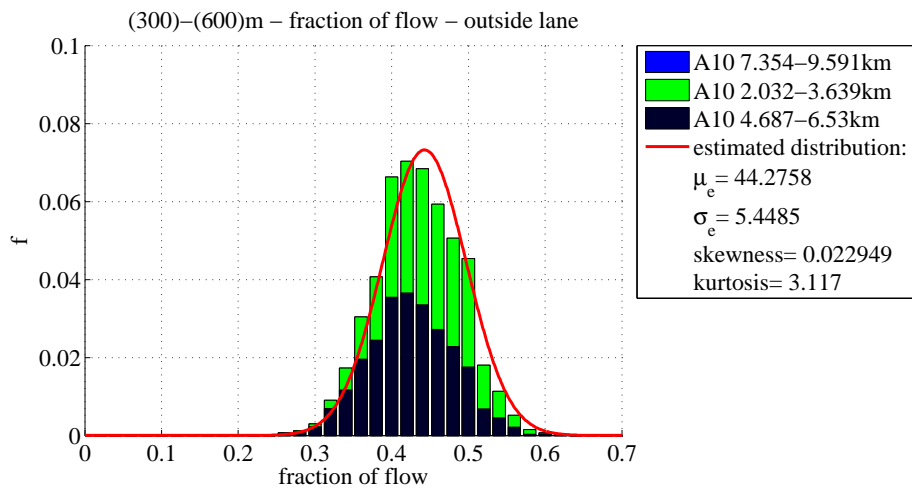
(b) Center lane speed histogram and estimated distribution



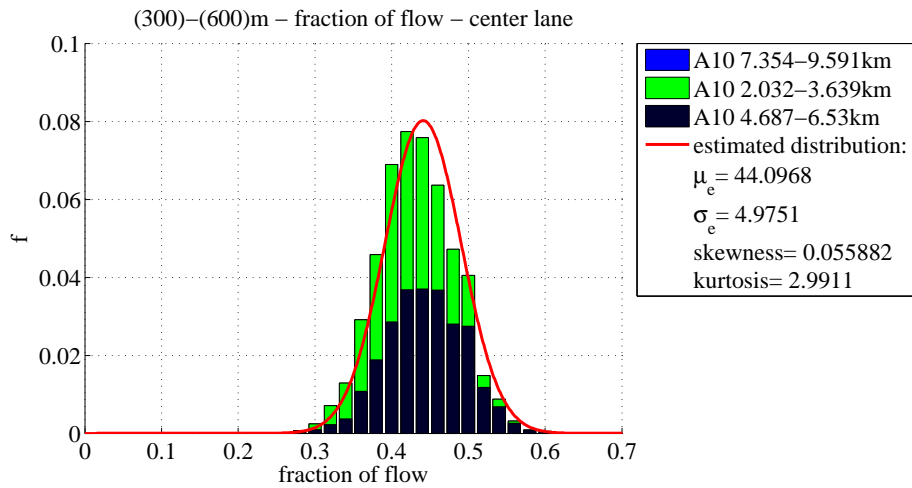
(c) Median lane speed histogram and estimated distribution

Figure B.16: Comparison histogram raw data with the maximum likelihood estimation in interval (0)-(300)m

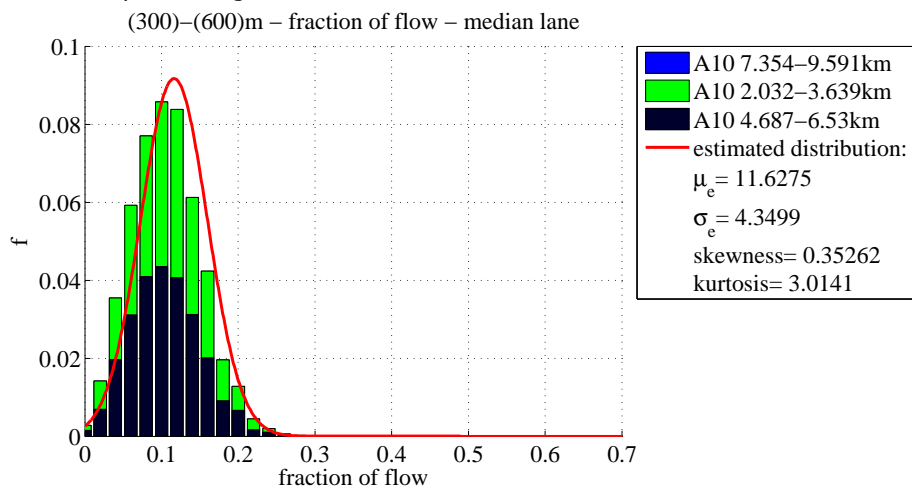
## (300)-(600) meter interval



(a) Outside lane speed histogram and estimated distribution



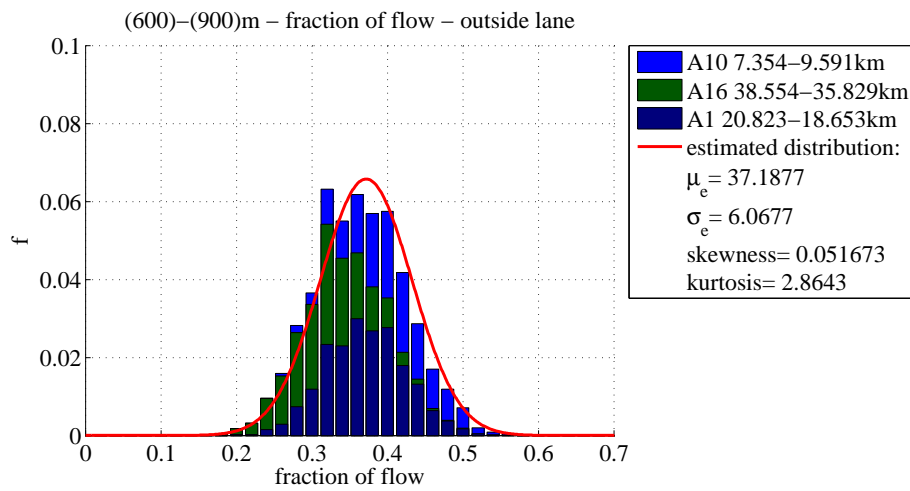
(b) Center lane speed histogram and estimated distribution



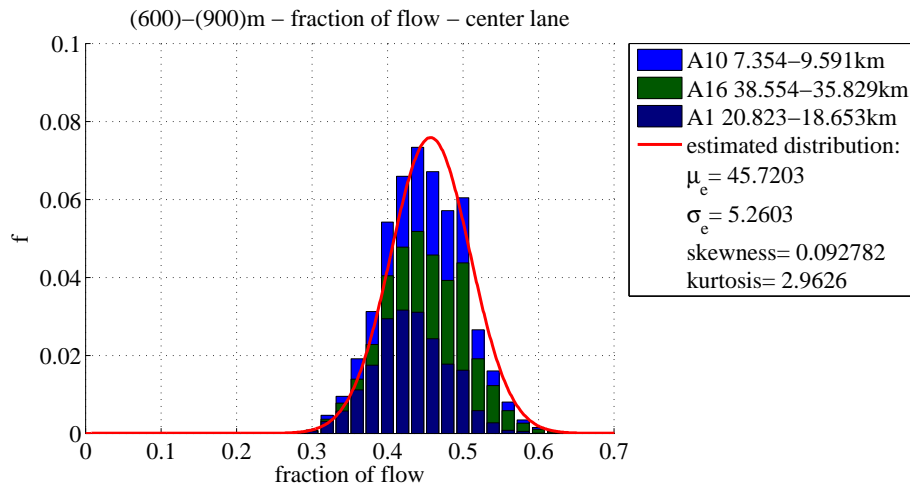
(c) Median lane speed histogram and estimated distribution

Figure B.17: Comparison histogram raw data with the maximum likelihood estimation in interval (300)-(600)m

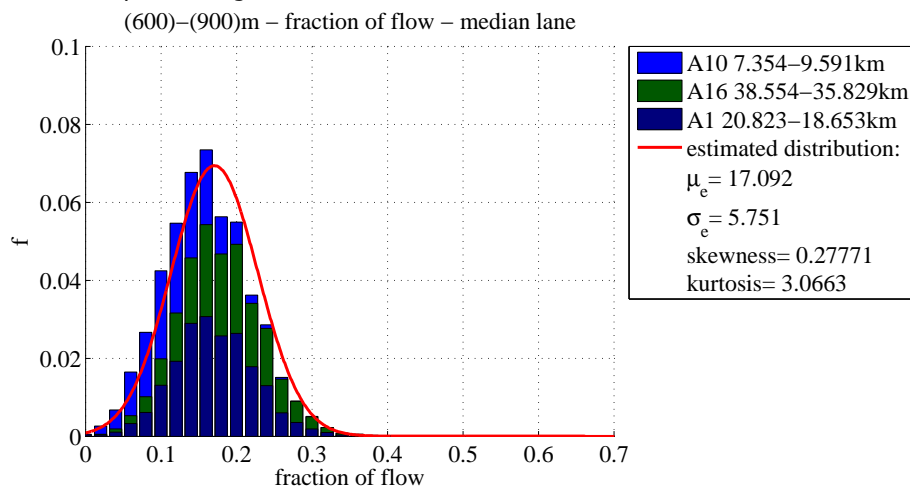
## (600)-(900) meter interval



(a) Outside lane speed histogram and estimated distribution



(b) Center lane speed histogram and estimated distribution



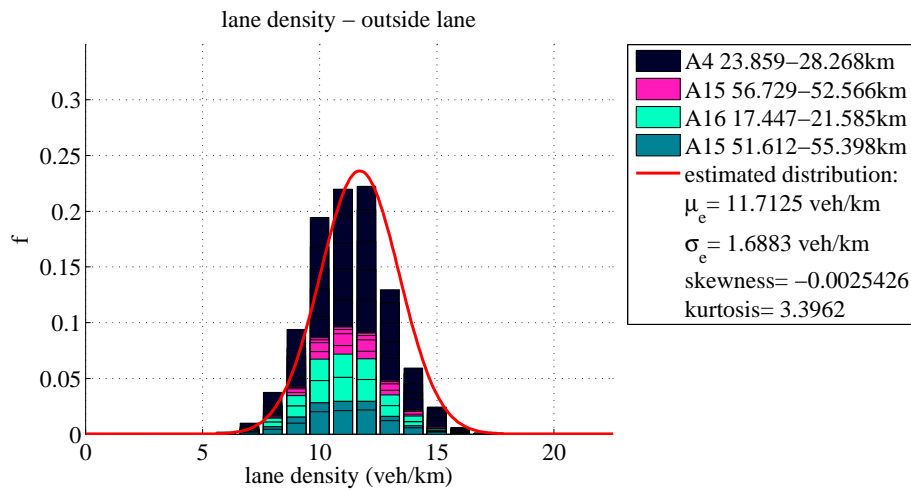
(c) Median lane speed histogram and estimated distribution

Figure B.18: Comparison histogram raw data with the maximum likelihood estimation in interval (600)-(900)m

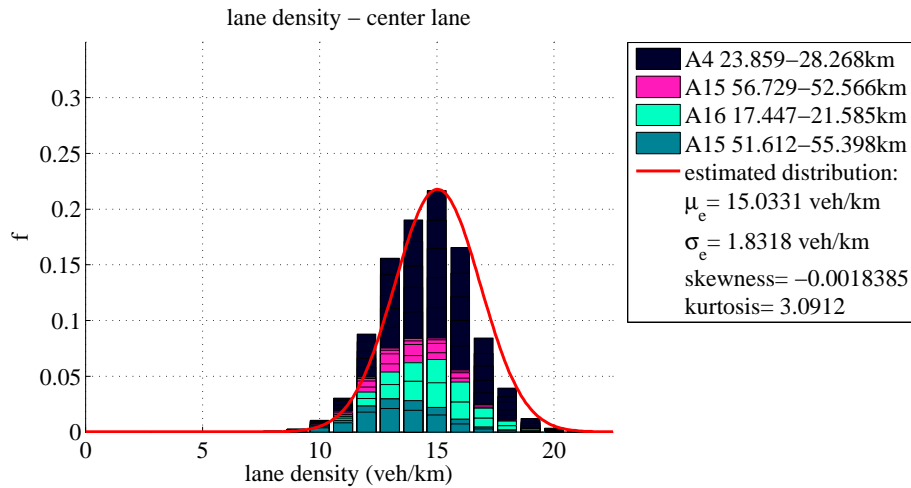
### **B.4. Lane density**

In figures [B.19](#) till [B.24](#) the results of the comparison are shown. In the histograms of the density the effect of the flow calculation with steps of 60 km/h was more present than in the lane flow distribution. To limit the influence of this effect, the bins are slightly increased. Therefore the distribution is more smooth than the lane flow distribution. The lane densities estimations are closely related to the raw data, only the raw data is more skewed to one or the other side.

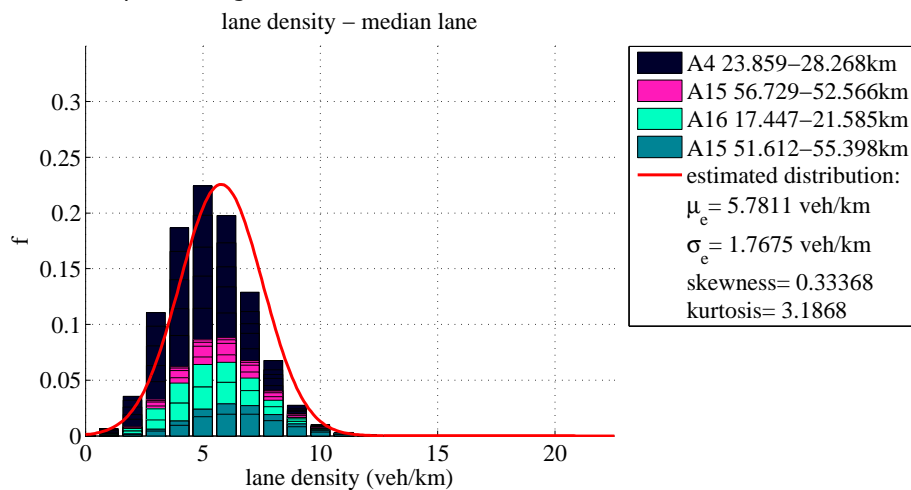
### B.4.1. Basic freeway segment



(a) Outside lane speed histogram and estimated distribution



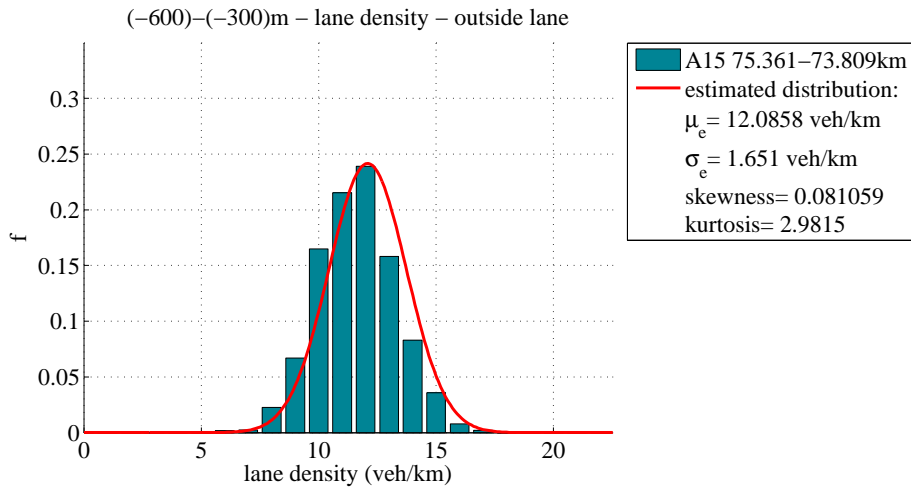
(b) Center lane speed histogram and estimated distribution



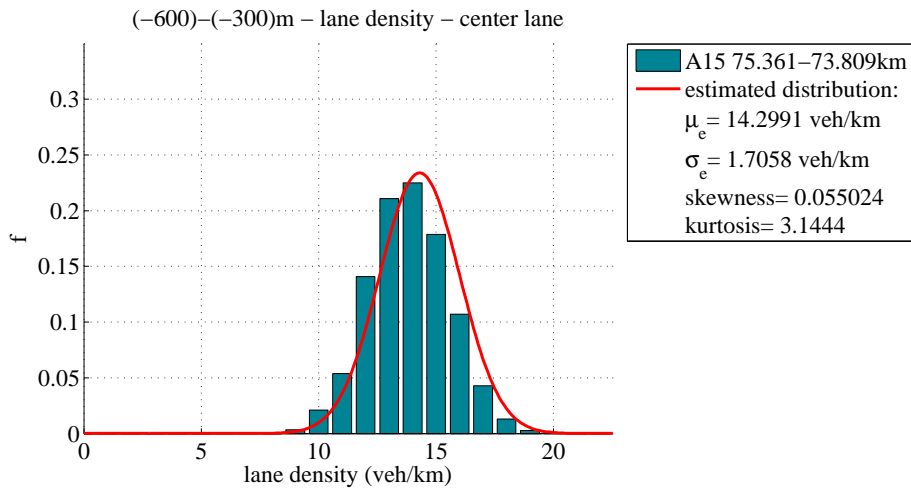
(c) Median lane speed histogram and estimated distribution

Figure B.19: Comparison histogram raw data with the maximum likelihood estimation on basic freeway segment

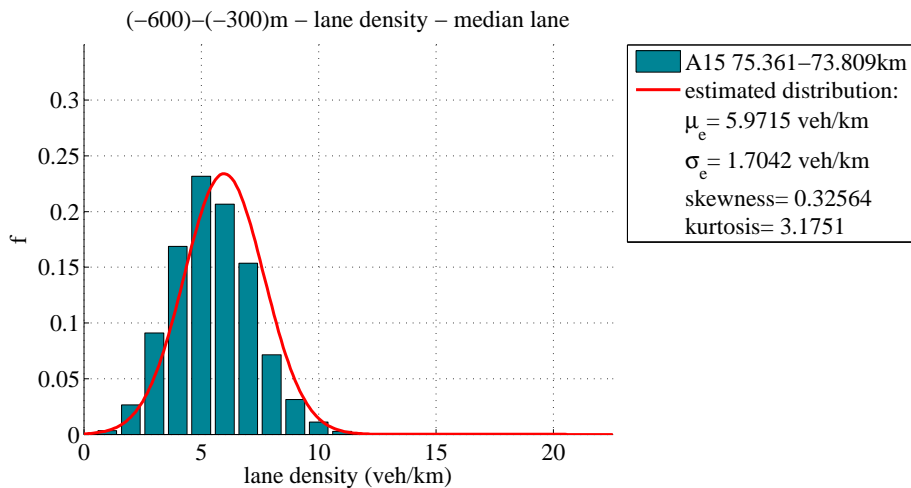
### B.4.2. Acceleration lane (-600)-(-300) meter interval



(a) Outside lane speed histogram and estimated distribution



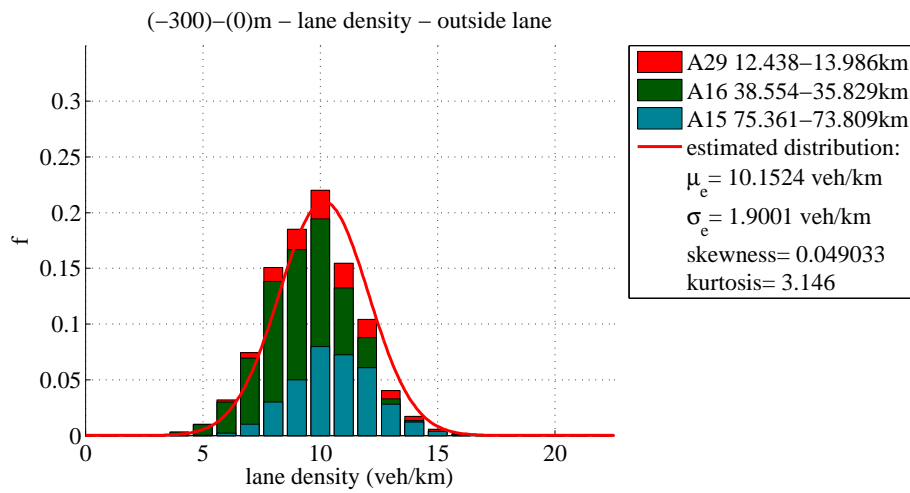
(b) Center lane speed histogram and estimated distribution



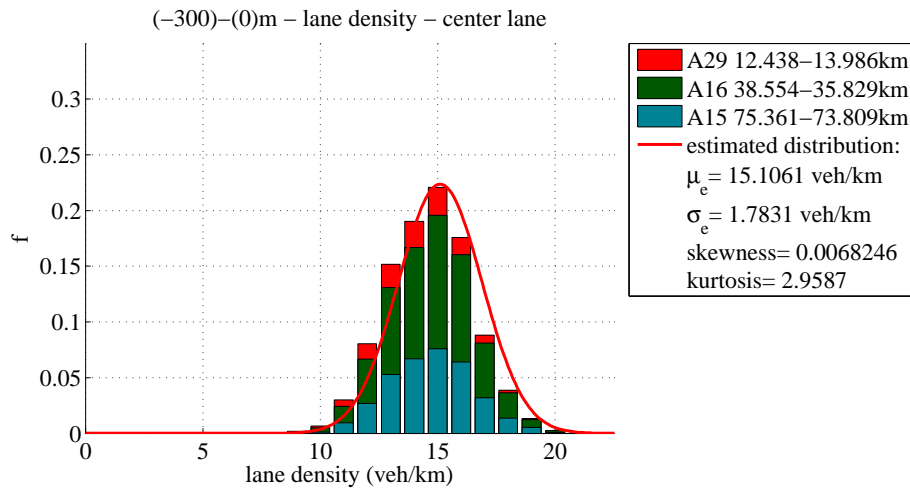
(c) Median lane speed histogram and estimated distribution

Figure B.20: Comparison histogram raw data with the maximum likelihood estimation in interval (-600)-(-300)m

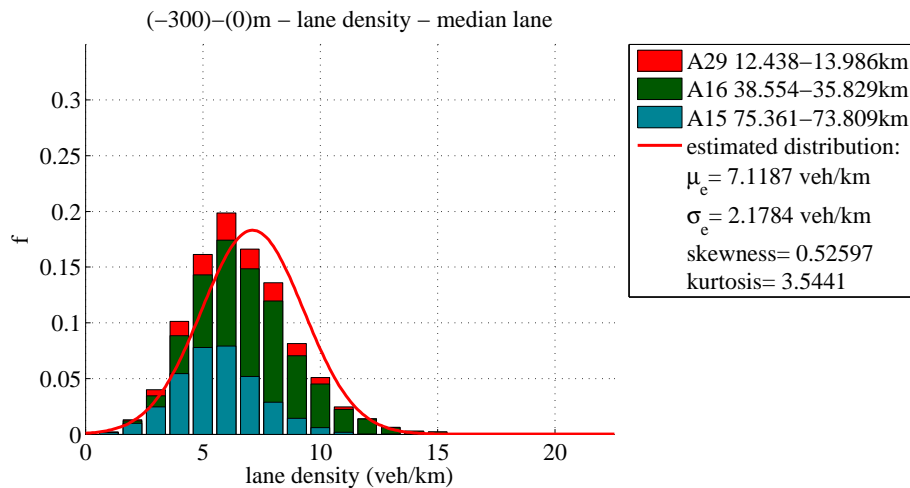
## (-300)-(0) meter interval



(a) Outside lane speed histogram and estimated distribution



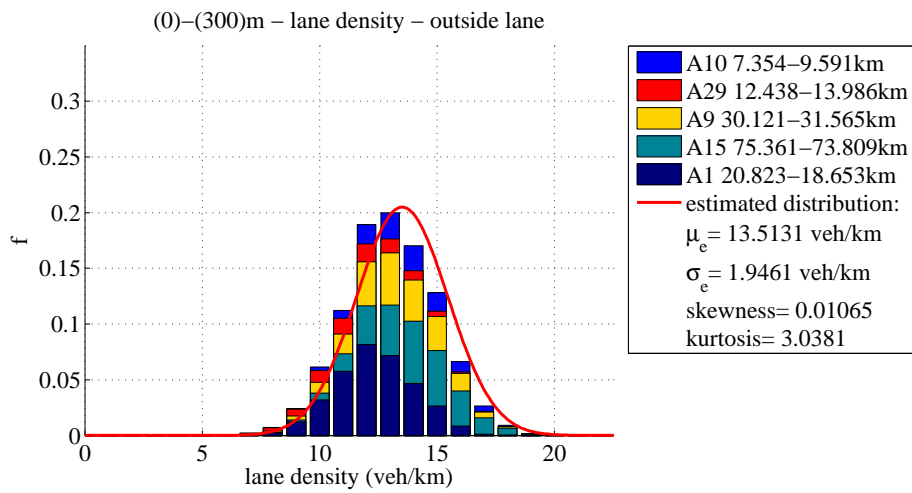
(b) Center lane speed histogram and estimated distribution



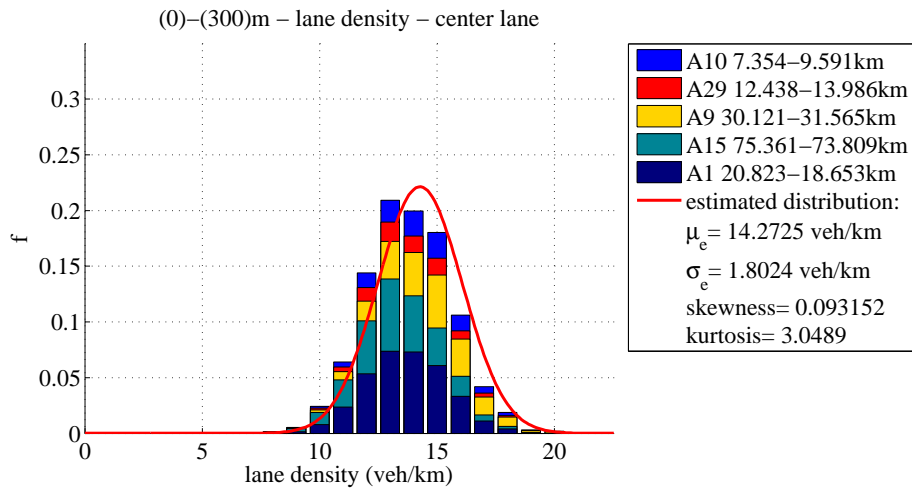
(c) Median lane speed histogram and estimated distribution

Figure B.21: Comparison histogram raw data with the maximum likelihood estimation in interval (-300)-(0)m

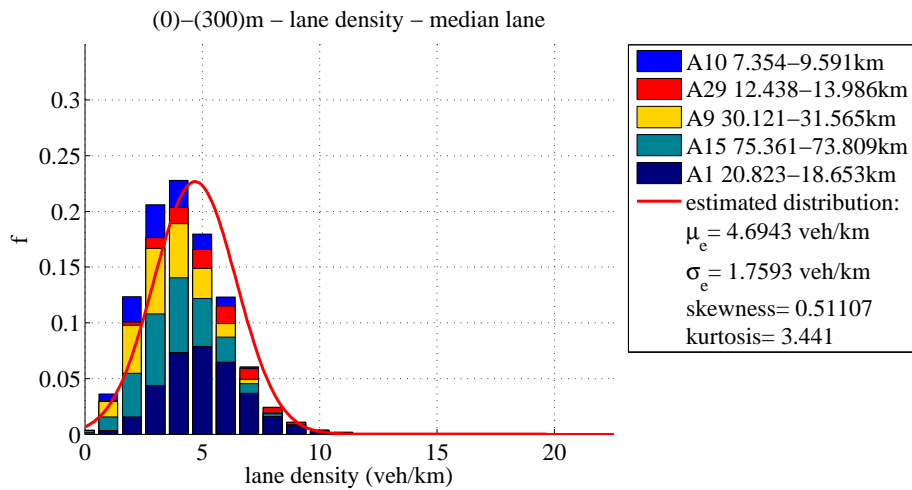
## (0)-(300) meter interval



(a) Outside lane speed histogram and estimated distribution



(b) Center lane speed histogram and estimated distribution

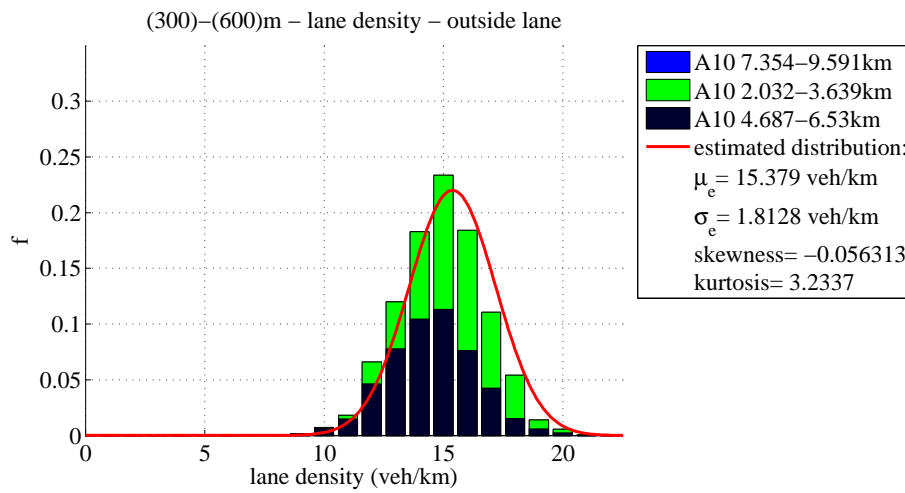


(c) Median lane speed histogram and estimated distribution

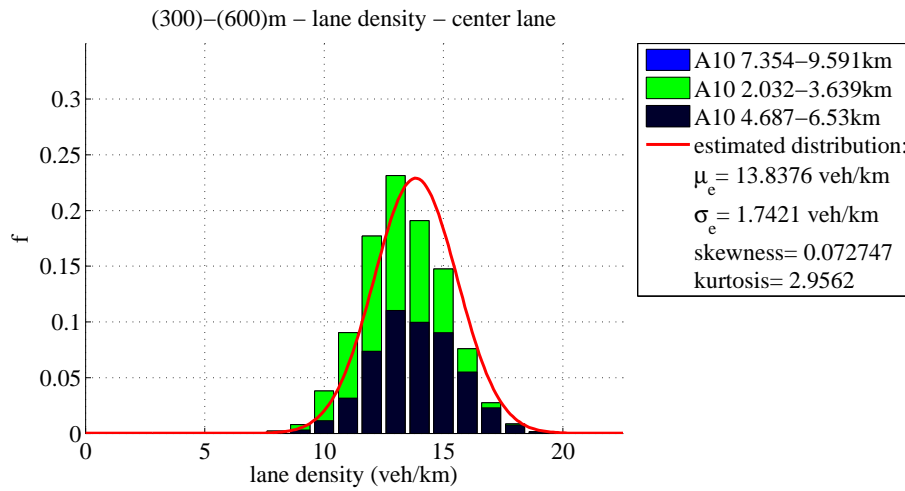
Figure B.22: Comparison histogram raw data with the maximum likelihood estimation in interval (0)-(300)m



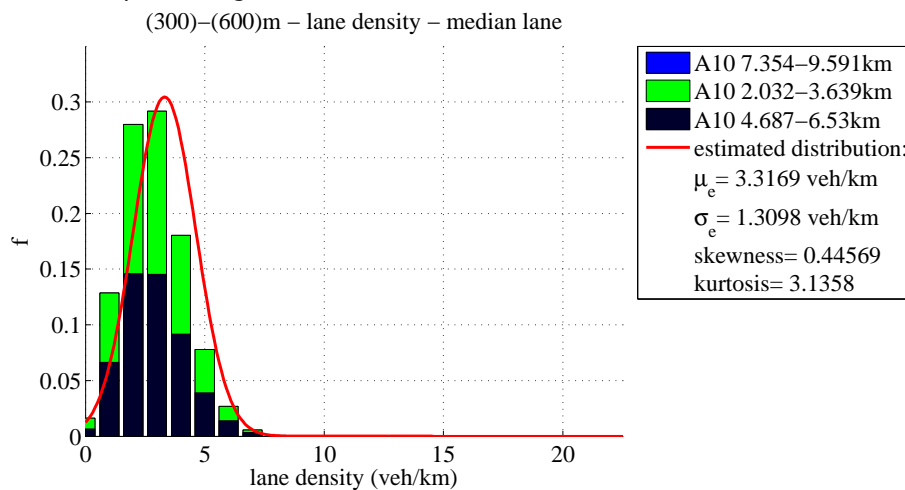
## (300)-(600) meter interval



(a) Outside lane speed histogram and estimated distribution



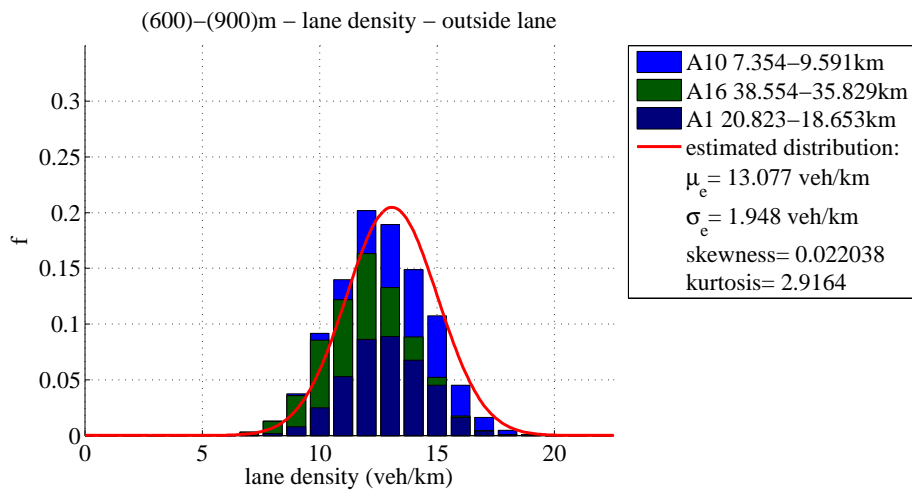
(b) Center lane speed histogram and estimated distribution



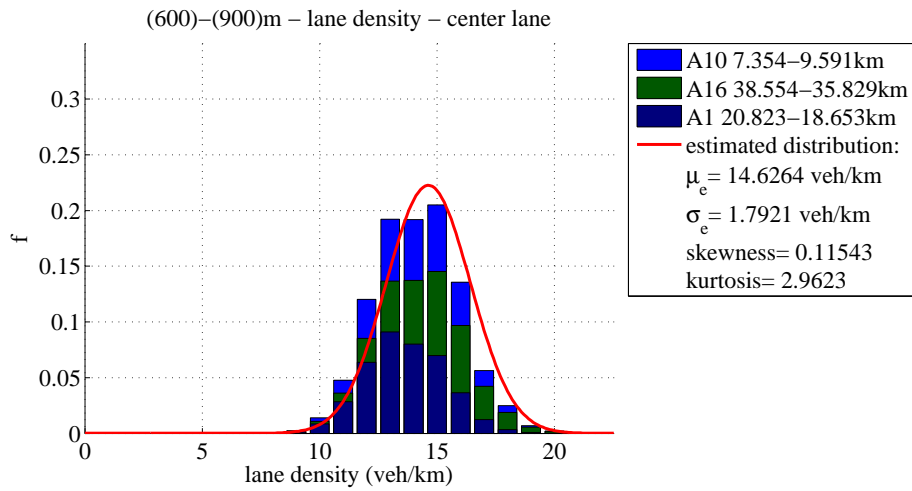
(c) Median lane speed histogram and estimated distribution

Figure B.23: Comparison histogram raw data with the maximum likelihood estimation in interval (300)-(600)m

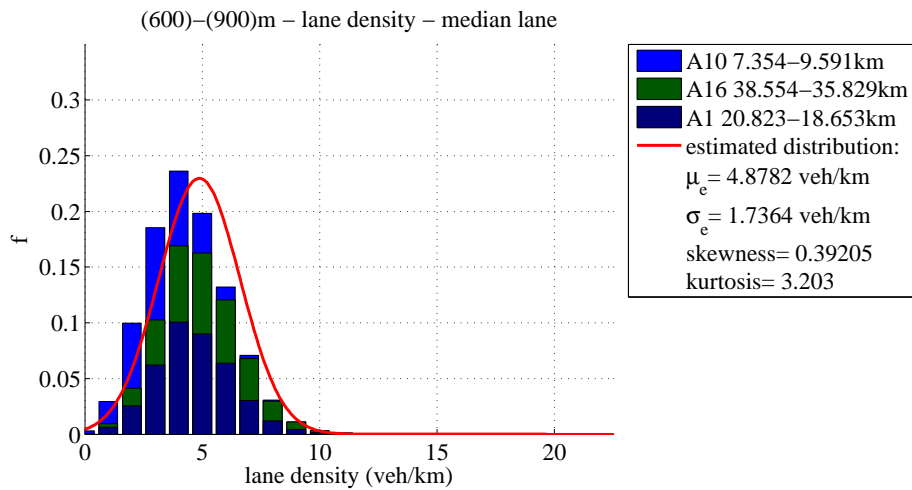
## (600)-(900) meter interval



(a) Outside lane speed histogram and estimated distribution



(b) Center lane speed histogram and estimated distribution



(c) Median lane speed histogram and estimated distribution

Figure B.24: Comparison histogram raw data with the maximum likelihood estimation in interval (600)-(900)m

## B.5. Other analyses - total flow

In figures B.25 till B.30 the results of the comparison are shown of the total flow. Interesting is the form of the raw data. The peak is relative flat compared to the normal distribution. This is the case for the basic freeway segment and also for most of the intervals on the acceleration lane. Therefore the difference is large between the estimation and the raw data. However, the mean seems to be constant.

### B.5.1. Basic freeway segment

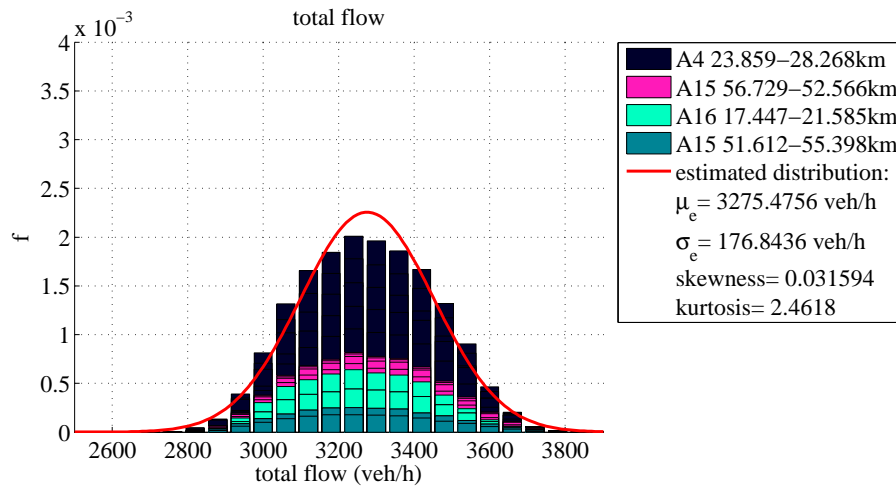


Figure B.25: Comparison histogram raw data with the maximum likelihood estimation on basic freeway segment

### B.5.2. Acceleration lane (-600)-(-300) meter interval

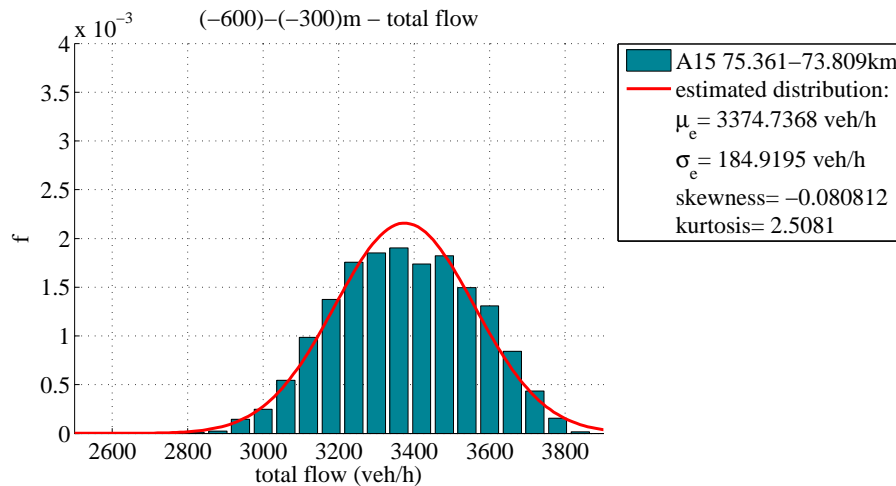


Figure B.26: Comparison histogram raw data with the maximum likelihood estimation in interval (-600)-(-300)m

### (-300)-(0) meter interval

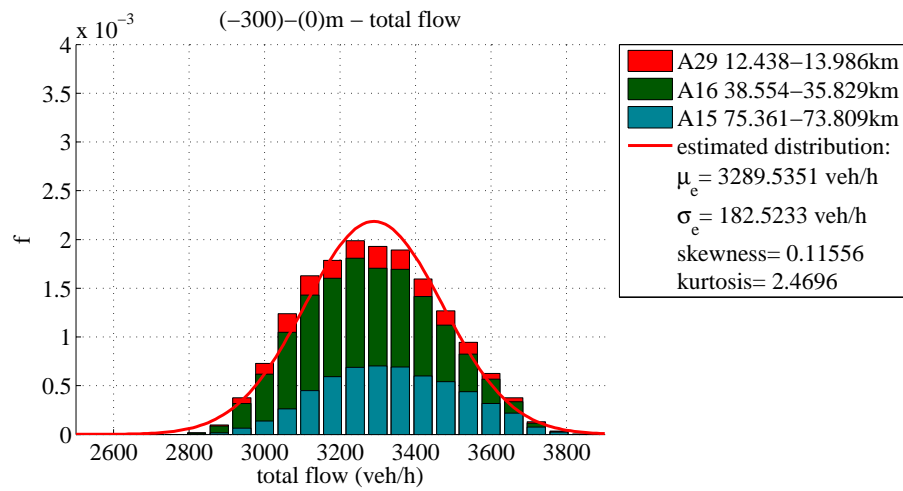


Figure B.27: Comparison histogram raw data with the maximum likelihood estimation in interval (-300)-(0)m

## (0)-(300) meter interval

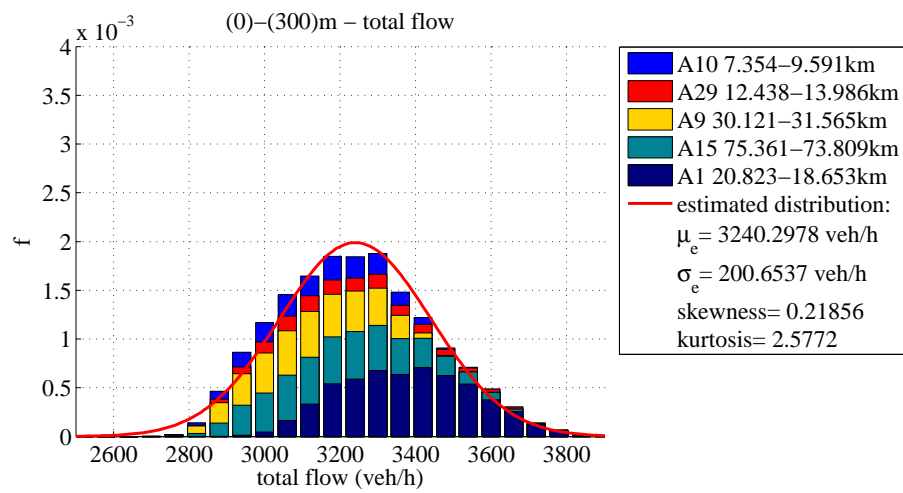


Figure B.28: Comparison histogram raw data with the maximum likelihood estimation in interval (0)-(300)m

## (300)-(600) meter interval

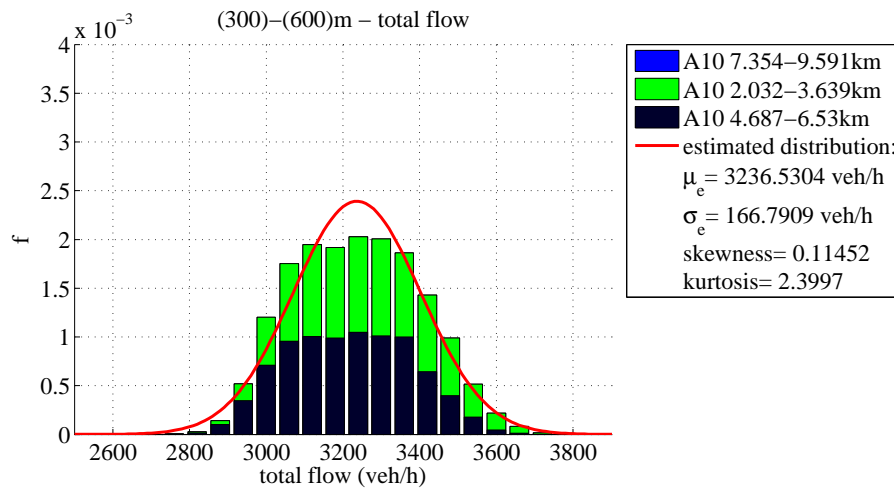


Figure B.29: Comparison histogram raw data with the maximum likelihood estimation in interval (300)-(600)m

## (600)-(900) meter interval

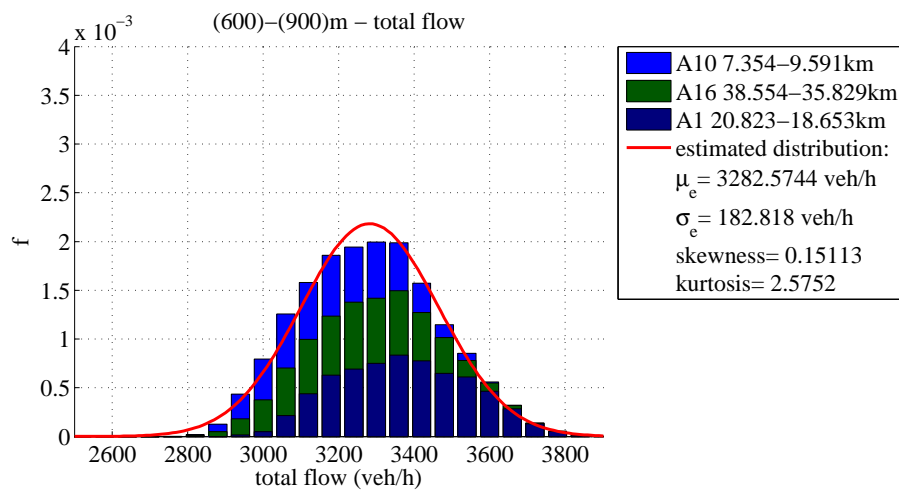


Figure B.30: Comparison histogram raw data with the maximum likelihood estimation in interval (600)-(900)m



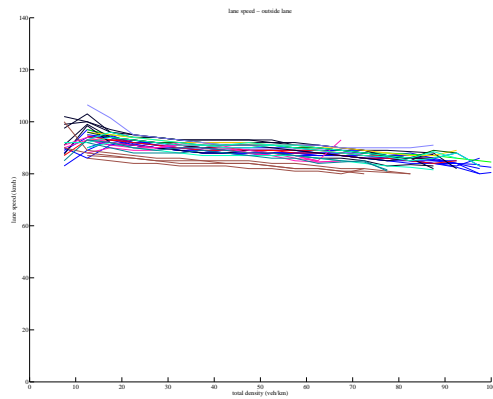
# Filtering **C**

## Filtering results

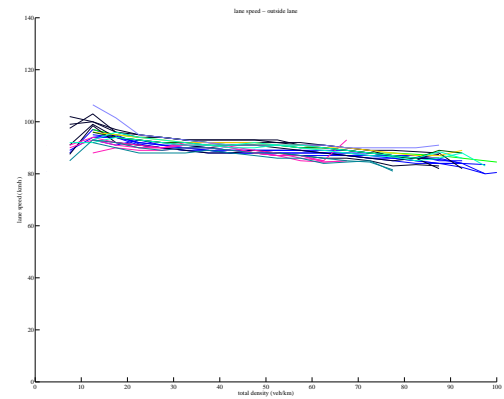
In this appendix the filtering is shown for the basic freeway segment and acceleration lane.

## C.1. Basic freeway segment

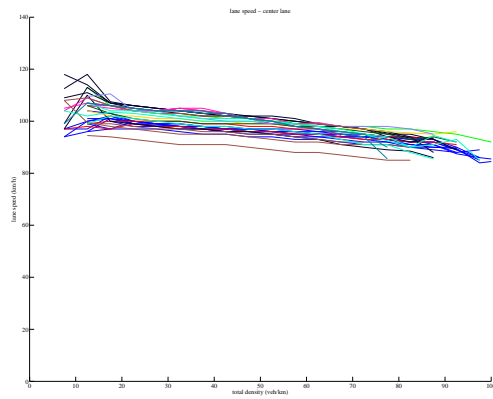
### C.1.1. Lane speed



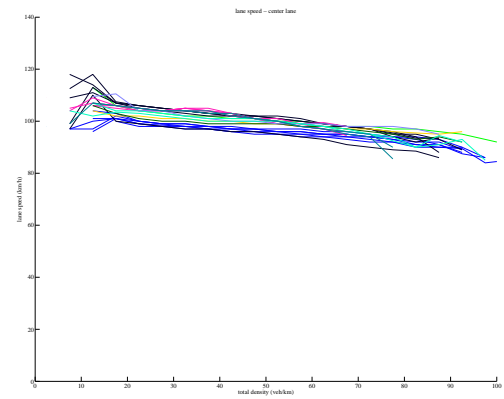
(a) Unfiltered



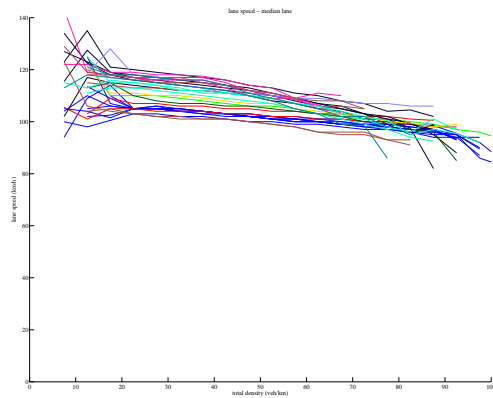
(b) Filtered



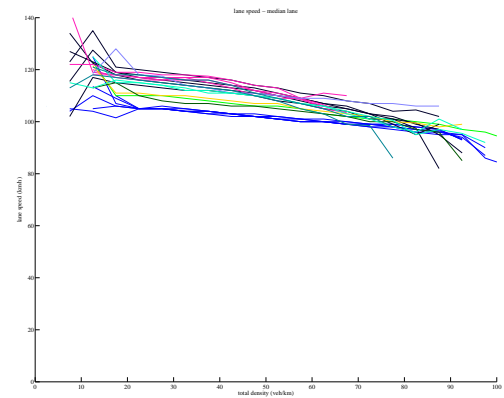
(c) Unfiltered



(d) Filtered



(e) Unfiltered

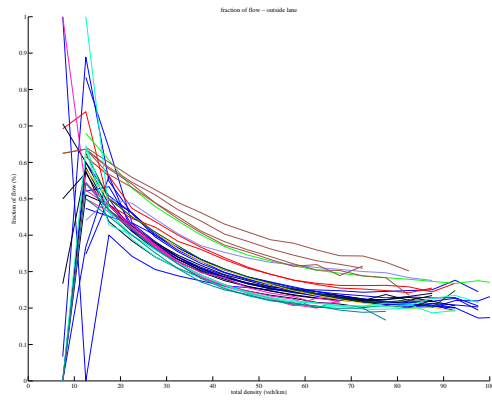


(f) Filtered

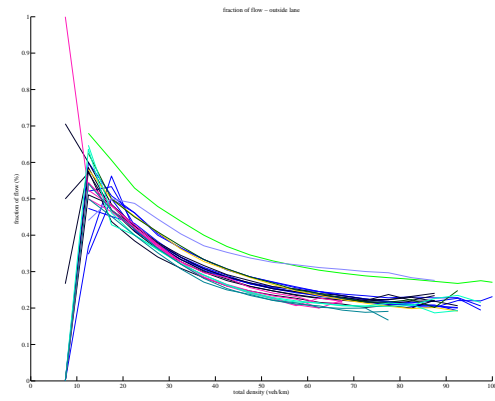
Figure C.1: Lane speed filtering



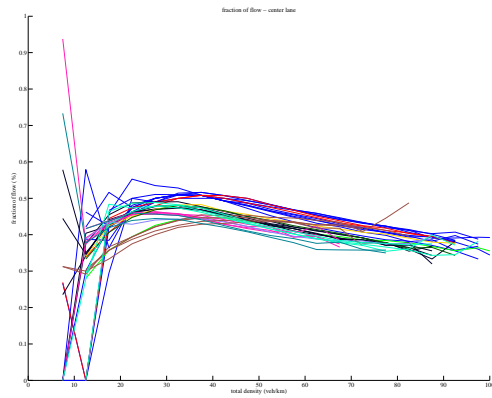
## C.1.2. Lane flow distribution



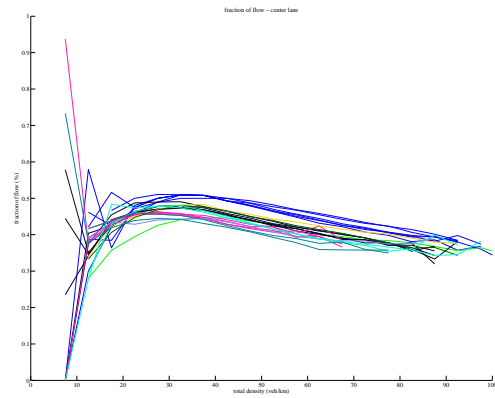
(a) Unfiltered



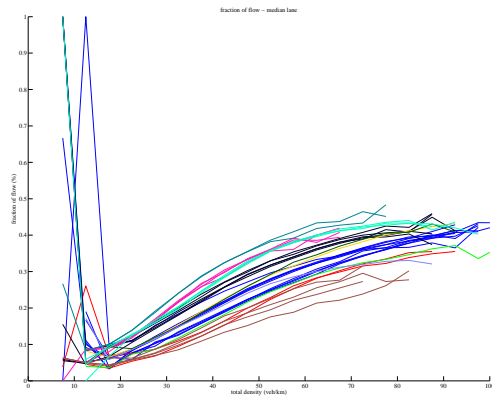
(b) Filtered



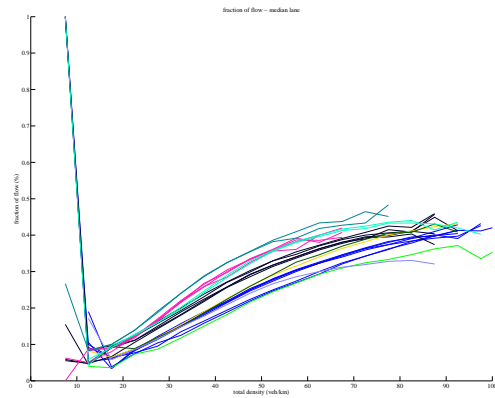
(c) Unfiltered



(d) Filtered



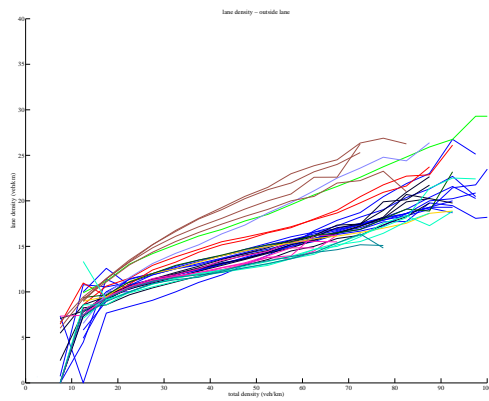
(e) Unfiltered



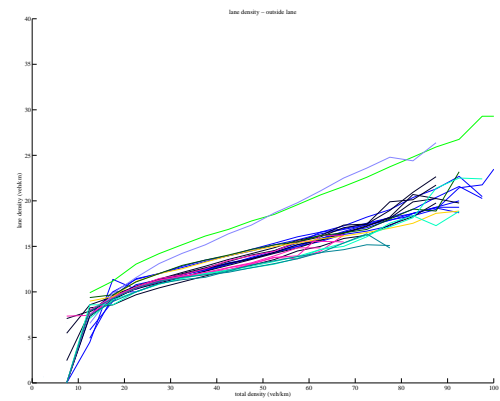
(f) Filtered

Figure C.2: fraction of flow filtering

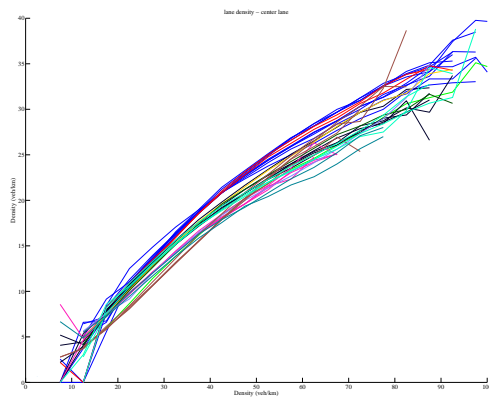
### C.1.3. Lane density



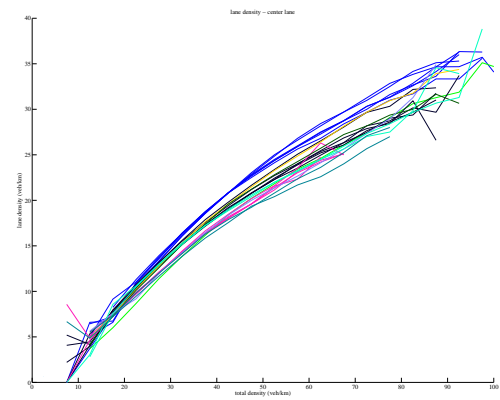
(a) Unfiltered



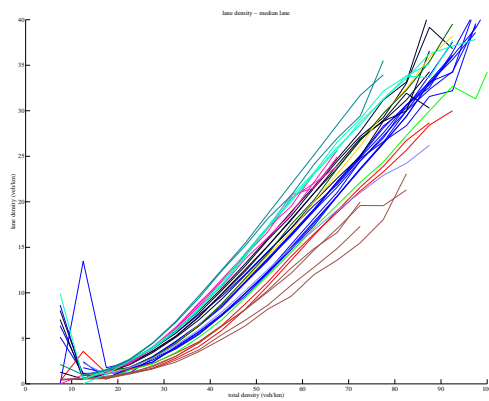
(b) Filtered



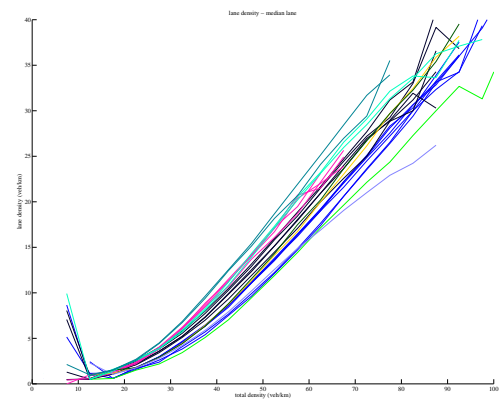
(c) Unfiltered



(d) Filtered



(e) Unfiltered

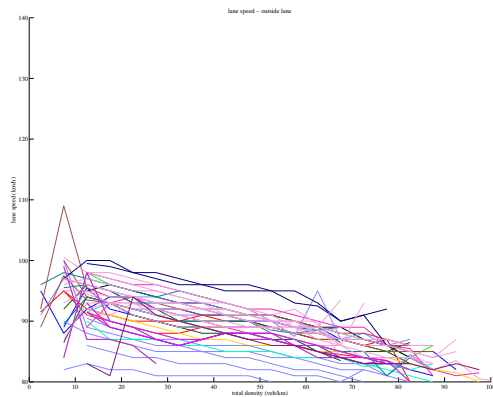


(f) Filtered

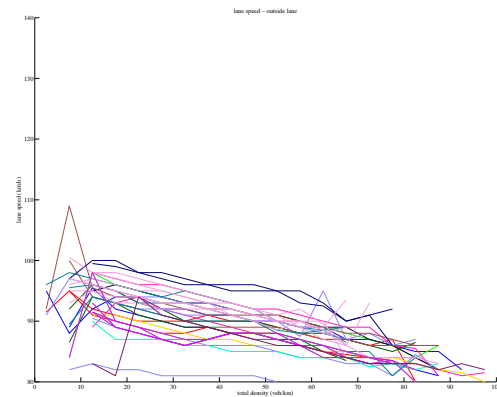
Figure C.3: lane density filtering

## C.2. Acceleration lane

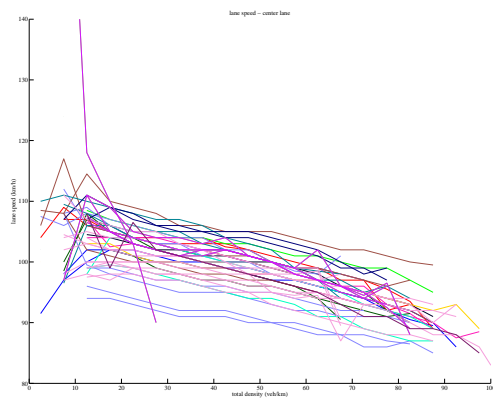
### C.2.1. Lane speed



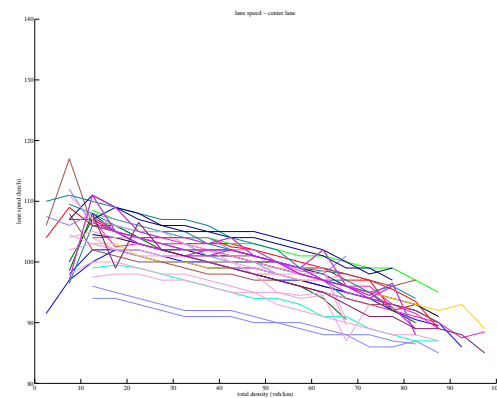
(a) Unfiltered



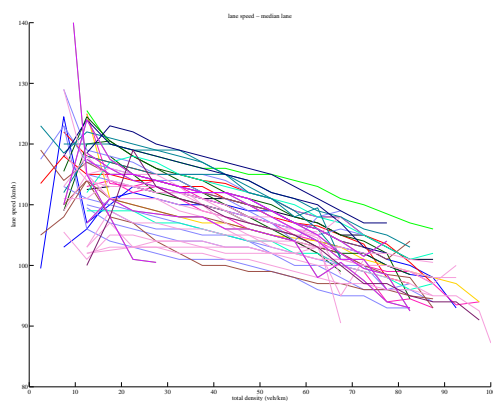
(b) Filtered



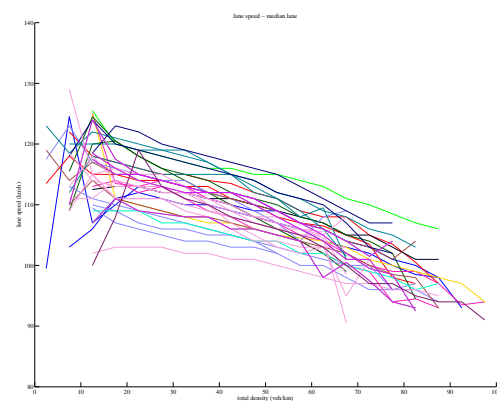
(c) Unfiltered



(d) Filtered



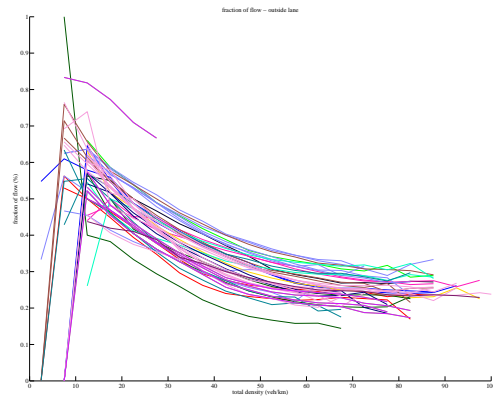
(e) Unfiltered



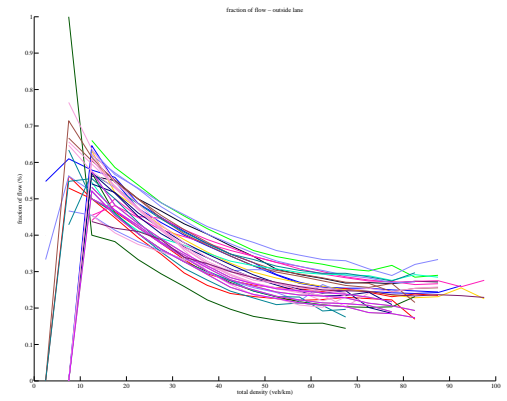
(f) Filtered

Figure C.4: Lane speed filtering

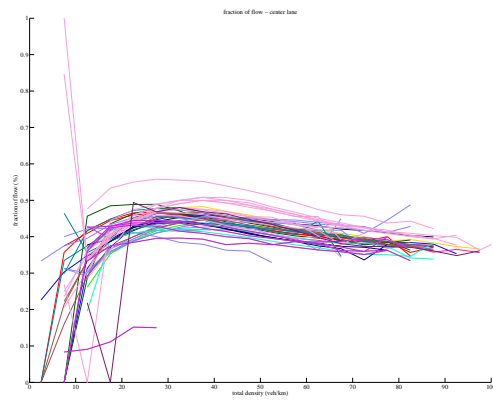
### C.2.2. Lane flow distribution



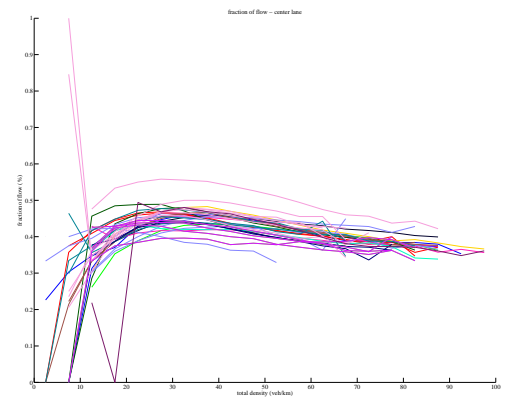
(a) Unfiltered



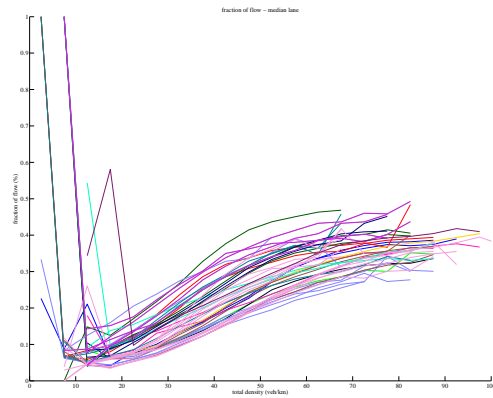
(b) Filtered



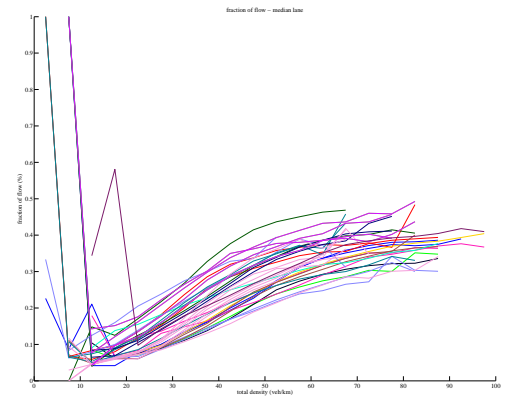
(c) Unfiltered



(d) Filtered



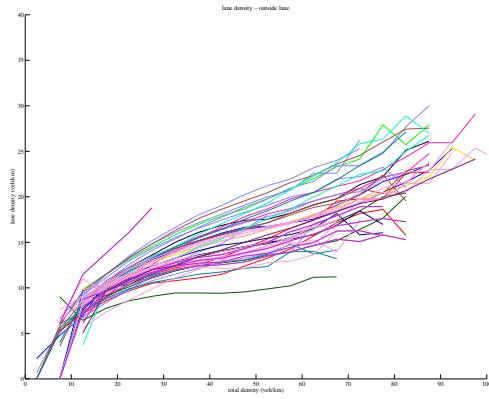
(e) Unfiltered



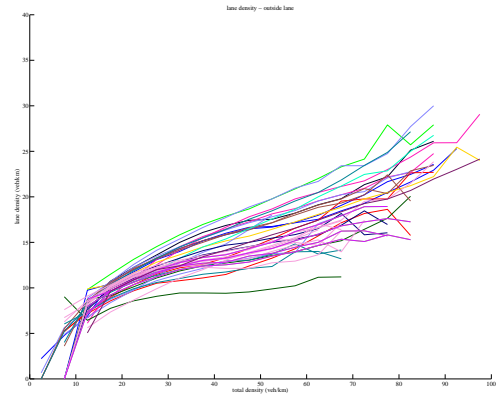
(f) Filtered

Figure C.5: fraction of flow filtering

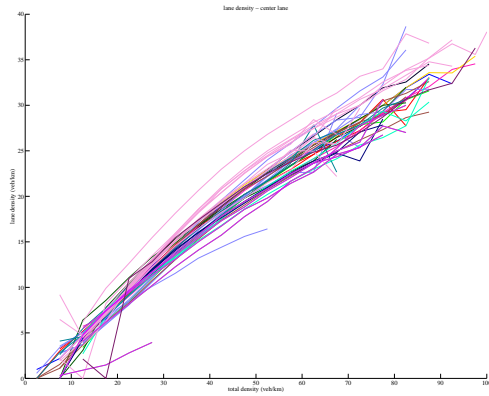
## C.2.3. Lane density



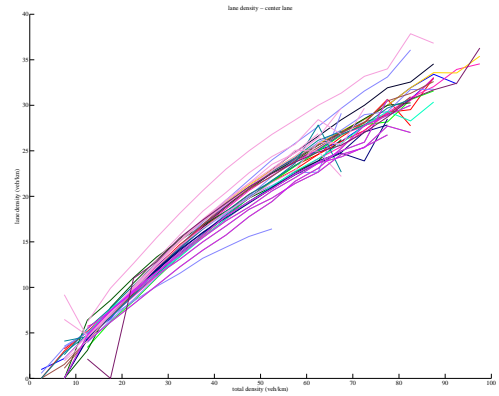
(a) Unfiltered



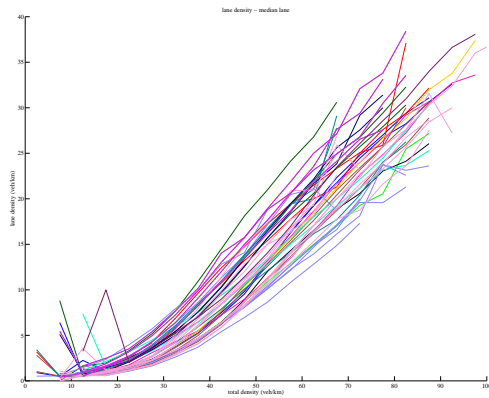
(b) Filtered



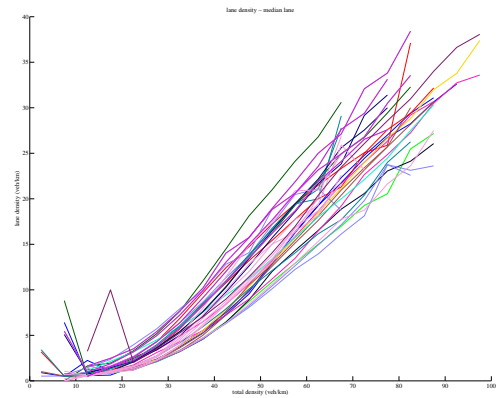
(c) Unfiltered



(d) Filtered



(e) Unfiltered



(f) Filtered

Figure C.6: lane density filtering



# D

## Description of the selected locations

### D.1. Basic freeway segments

road number	from km	to km	#detectors
A4	23.9	28.3	5
A15	51.6	55.4	3
A15	56.7	52.6	4
A16	17.4	21.6	3

**D.1.1. A4 23.9 to 28.3**

Characteristic	Value
Freeway	A4
Location	23.9-28.3
Average week day intensity freeway	49666 vehicle/week day
Average fraction freight traffic	10.8%
Upstream discontinuity	Acceleration lane
Downstream discontinuity	Deceleration lane
Remarks	-



Figure D.1: Situation A4 23.9-28.3 km



**D.1.2. A15 51.6 to 55.4**

Characteristic	Value
Freeway	A15
Location	51.6-55.4 km
Average week day intensity freeway	60809 vehicles/week day
Average fraction freight traffic	22.1%
Upstream discontinuity	Merge segment
Downstream discontinuity	Diverge segment
Remarks	Possible construction site

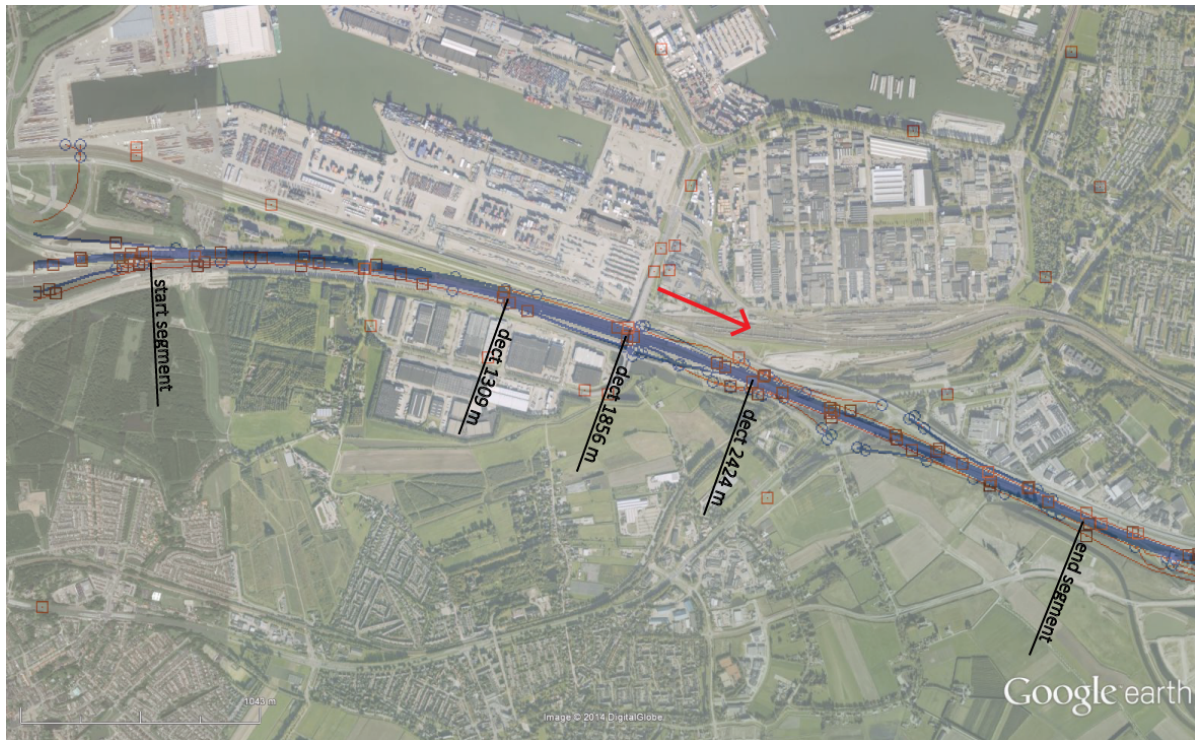


Figure D.2: Situation A15 51.6-55.4 km

**D.1.3. A15 56.7 to 52.6**

Characteristic	Value
Freeway	A15
Location	56.7-52.6 km
Average week day intensity freeway	48548 vehicles/week day
Average fraction freight traffic	19.5%
Upstream discontinuity	Weaving segment
Downstream discontinuity	Weaving segment
Remarks	Possible construction site

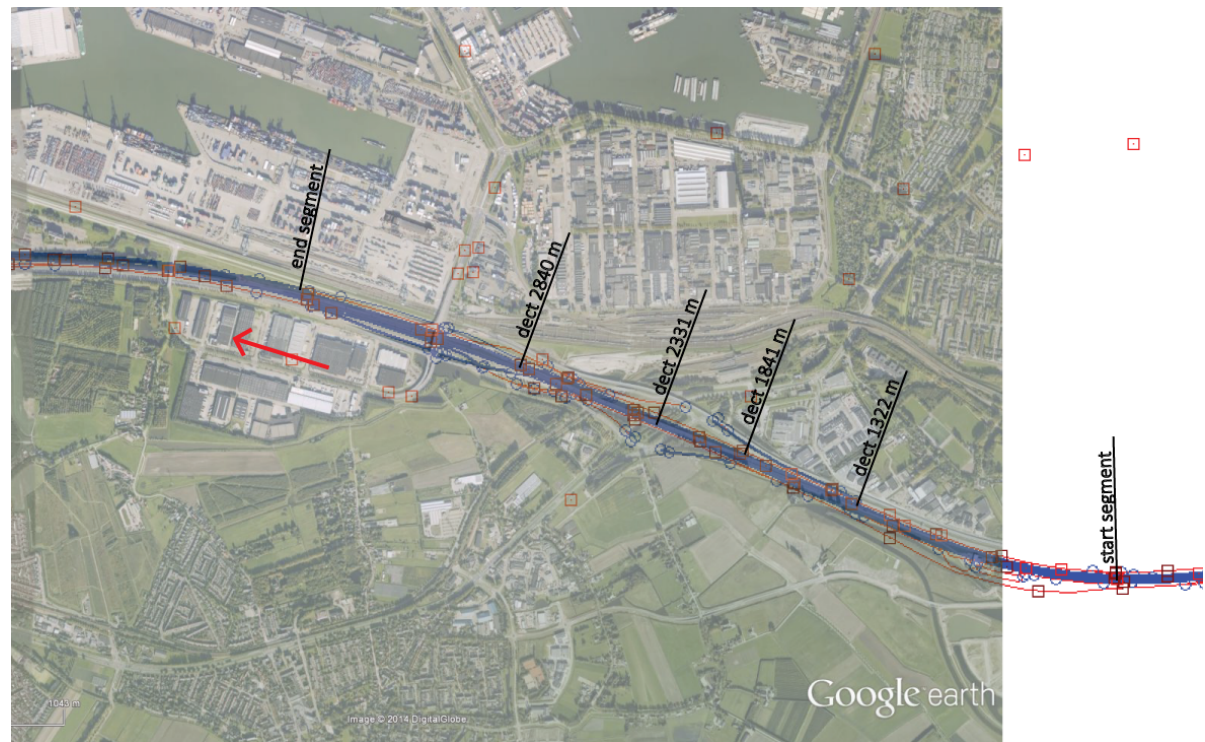


Figure D.3: Situation A15 56.7-52.6 km



**D.1.4. A16 17.4 to 21.6**

Characteristic	Value
Freeway	A16
Location	17.4-21.6 km
Average week day intensity freeway	68740 vehicles/week day
Average fraction freight traffic	9.5%
Upstream discontinuity	Weaving segment
Downstream discontinuity	Weaving segment
Remarks	Bridge



Figure D.4: Situation A16 17.4-21.6 km

**D.2. Acceleration lanes**

roadnumber	from km	to km	#detectors
A1	20.8	18.7	2
A9	30.1	31.6	1
A10	2.0	3.6	1
A10	4.7	6.5	1
A10	7.4	9.6	3
A15	75.4	73.8	1
A16	38.6	35.8	1
A29	12.4	14.0	1

**D.2.1. A1 20.8 to 18.7**

Characteristic	Value
Freeway	A1
Location	20.8-18.7 km
Average week day intensity freeway	49918 veh/week day
Average fraction freight traffic	8.1%
Length acceleration lane	233 meter
Average week day intensity acceleration lane	no data (service area)
Upstream discontinuity	Deceleration lane
Downstream discontinuity	Deceleration lane
Remarks	Service Area



Figure D.5: Situation A1 20.8-18.7 km





**D.2.3. A10 2.0 to 3.6**

Characteristic	Value
Freeway	A10
Location	2.0-3.6 km
Average week day intensity freeway	48531 vehicles/week day
Average fraction freight traffic	8.3%
Length acceleration lane	246 meter
Average week day intensity acceleration lane	5339 vehicles/week day
Upstream discontinuity	Deceleration lane
Downstream discontinuity	Deceleration lane
Remarks	-

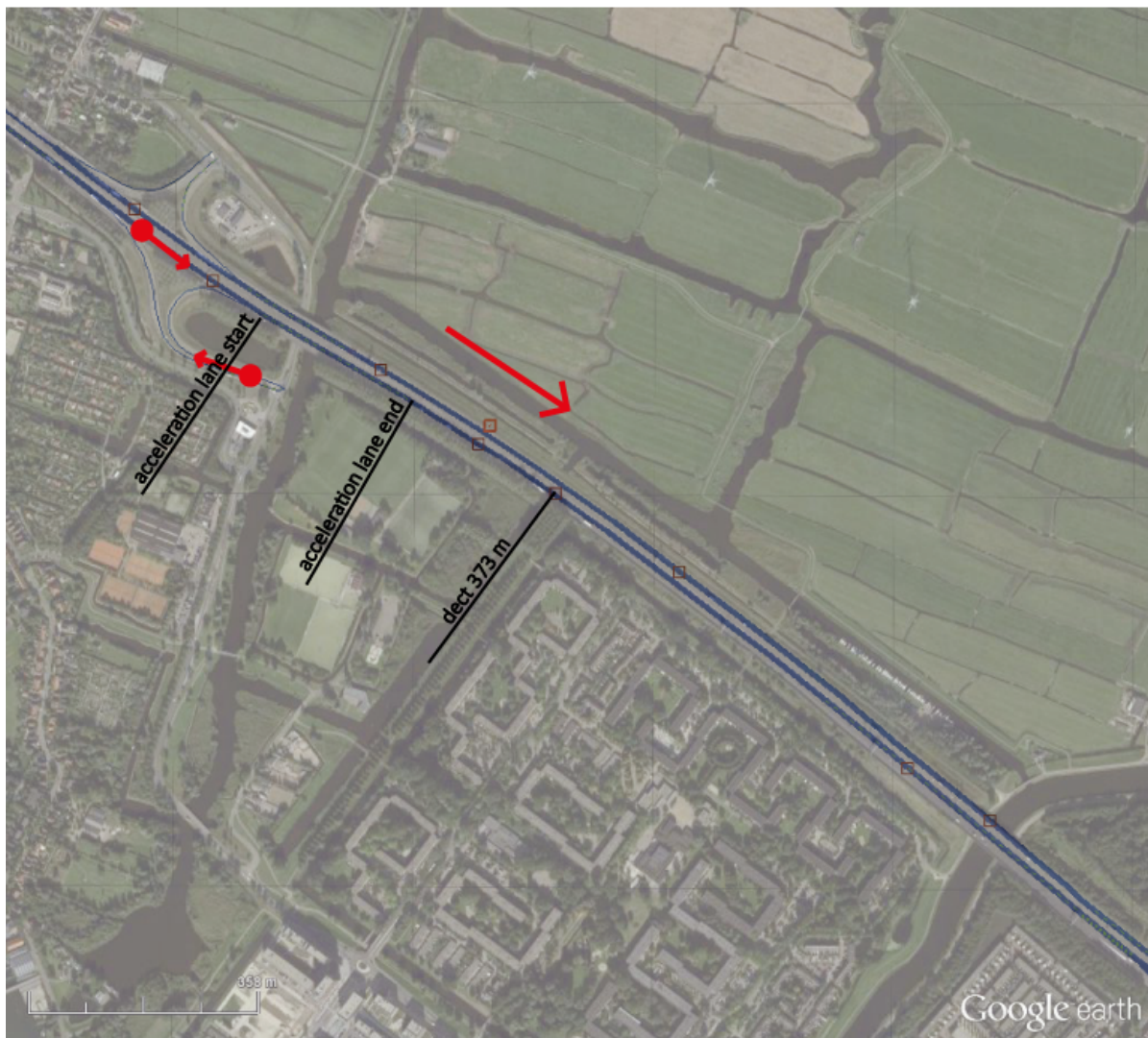


Figure D.7: Situation A10 2.0-3.6 km



**D.2.4. A10 4.7 to 6.5**

Characteristic	Value
Freeway	A10
Location	4.7-6.5 km
Average week day intensity freeway	45941 vehicles/week day
Average fraction freight traffic	8.5%
Length acceleration lane	241 meter
Average week day intensity acceleration lane	13895 vehicles/week day
Upstream discontinuity	Deceleration lane
Downstream discontinuity	Deceleration lane
Remarks	-

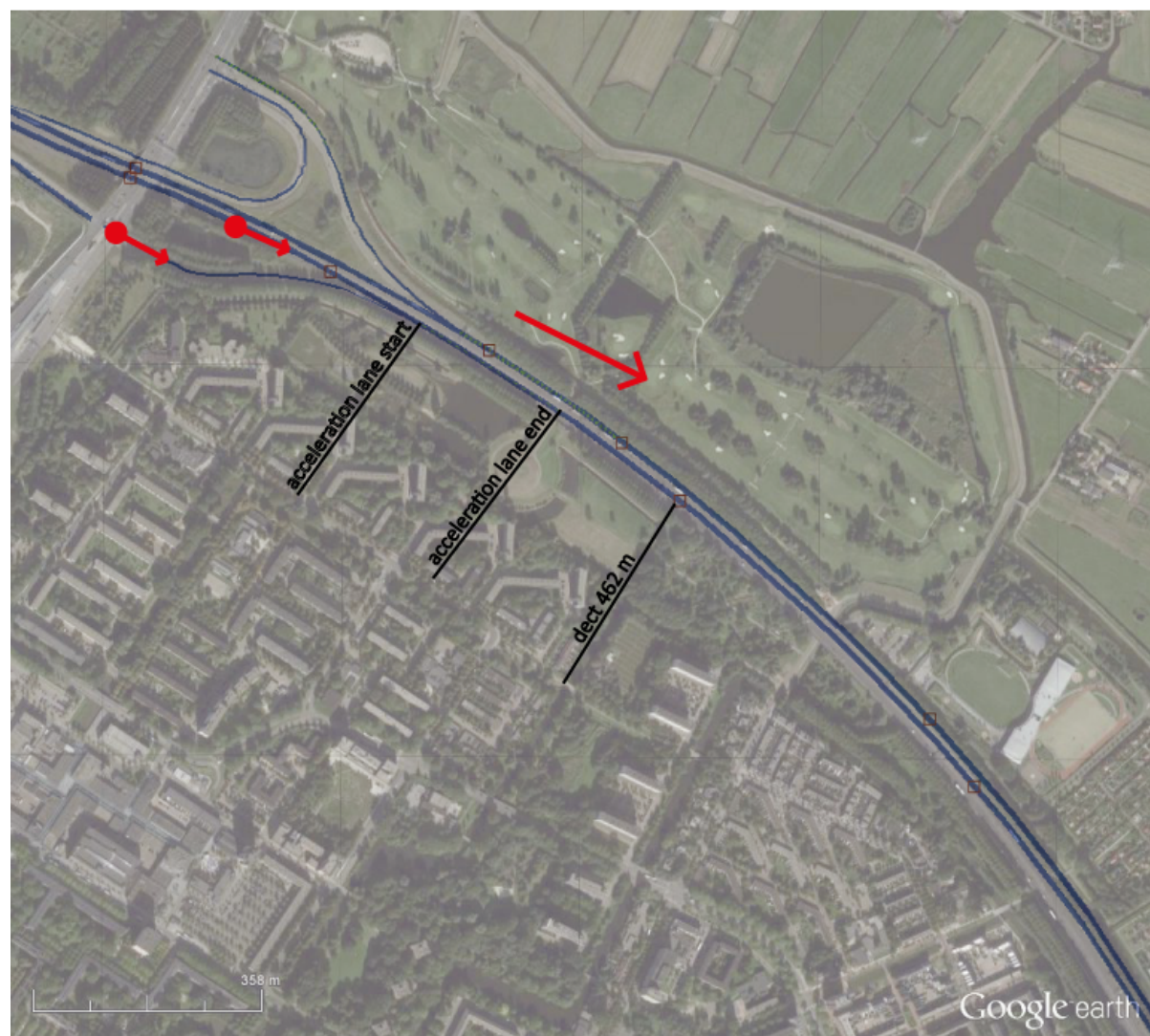


Figure D.8: Situation A10 4.7-6.5 km



**D.2.5. A10 7.4 to 9.6**

Characteristic	Value
Freeway	A10
Location	7.4-9.6 km
Average week day intensity freeway	53178 vehicles/weekday
Average fraction freight traffic	9.9%
Length acceleration lane	260 meter
Average week day intensity acceleration lane	7527 vehicles/week day
Upstream discontinuity	Deceleration lane
Downstream discontinuity	Deceleration lane
Remarks	Tunnel

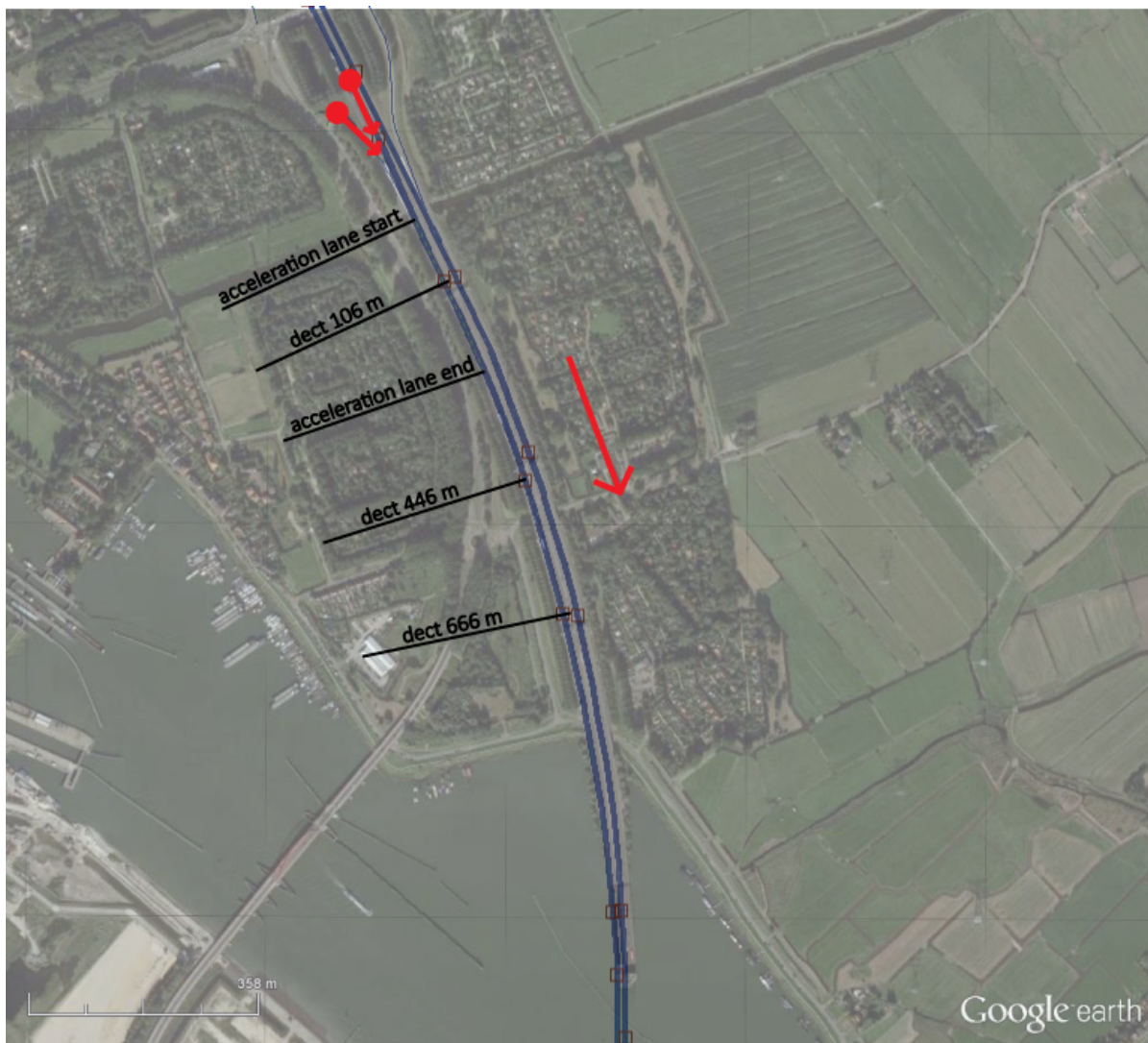


Figure D.9: Situation A10 7.4-9.6 km

**D.2.6. A15 75.4 to 73.8**

Characteristic	Value
Freeway	A15
Location	75.4-73.8 km
Average week day intensity freeway	41223 vehicles/week day
Average fraction freight traffic	13.8%
Length acceleration lane	316 meter
Average week day intensity acceleration lane	no data (service area)
Upstream discontinuity	Deceleration lane
Downstream discontinuity	Deceleration lane
Remarks	Service area with on-ramp

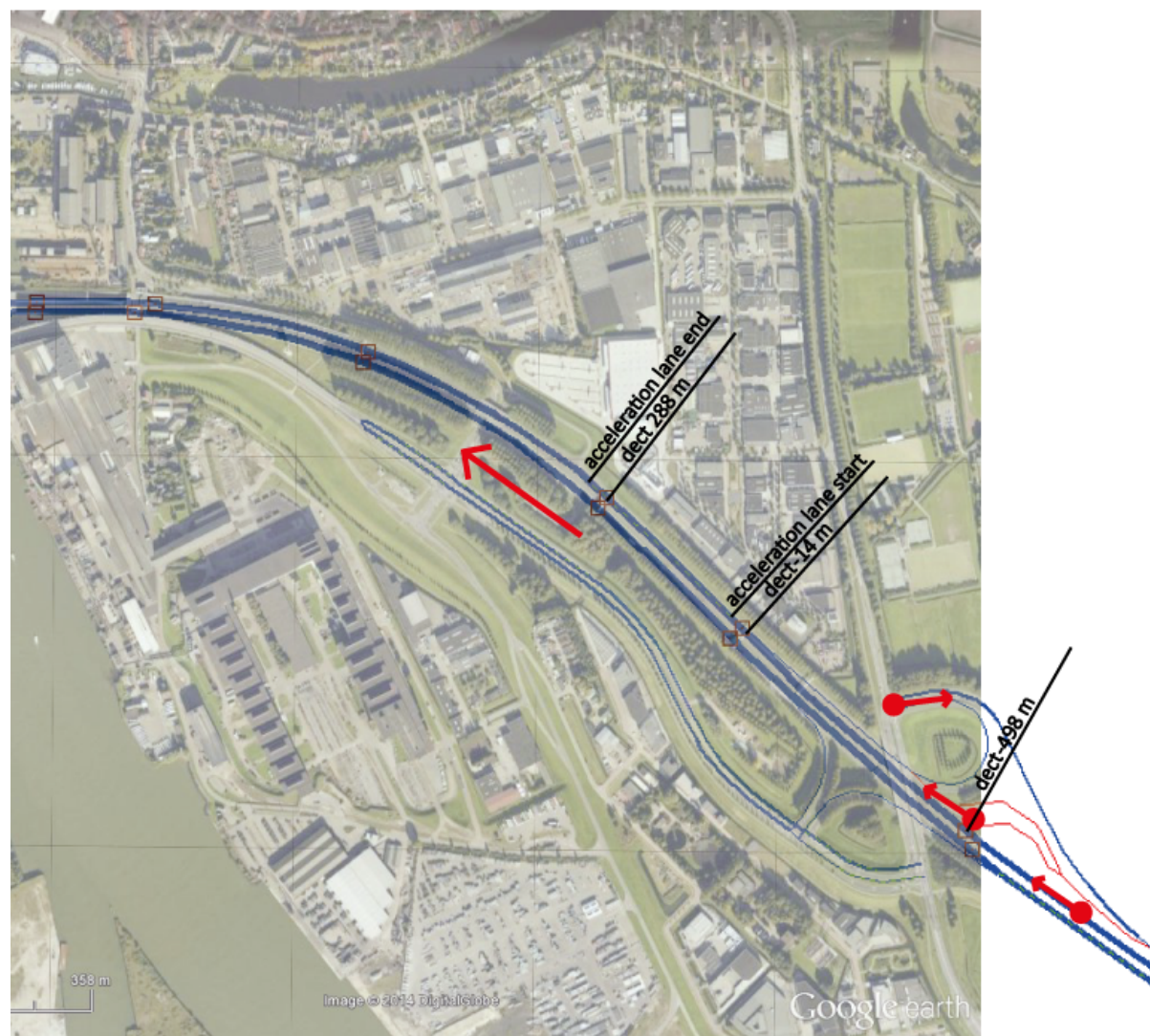


Figure D.10: Situation A15 75.3-73.8 km



**D.2.7. A16 38.6 to 35.8**

Characteristic	Value
Freeway	A16
Location	38.6-35.8 km
Average week day intensity freeway	43657 vehicles/week day
Average fraction freight traffic	16.5%
Length acceleration lane	267 meter
Average week day intensity acceleration lane	5033 vehicles/week day
Upstream discontinuity	Deceleration lane
Downstream discontinuity	Deceleration lane
Remarks	Noise barrier

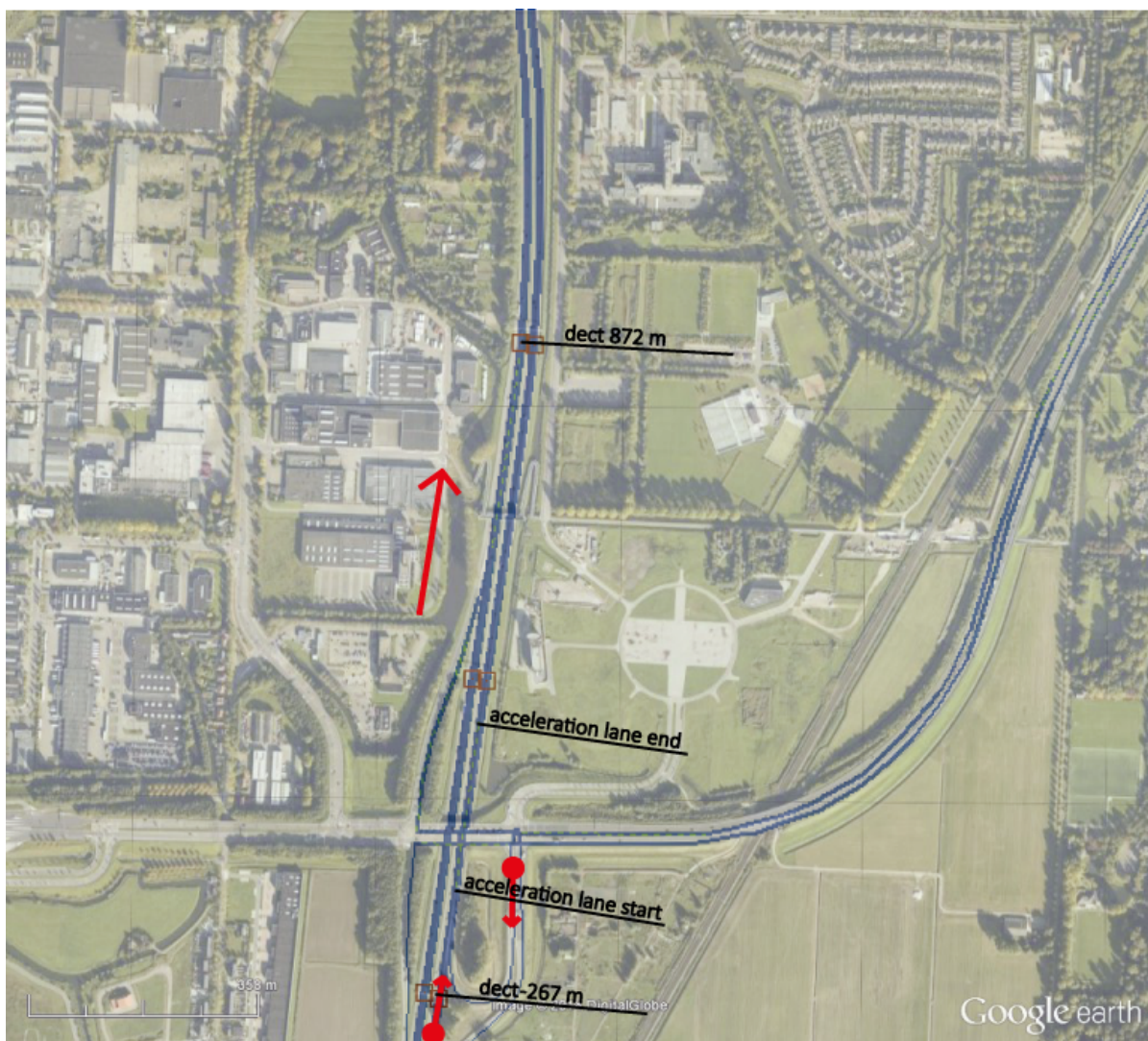


Figure D.11: Situation A16 38.6-35.8 km

**D.2.8. A29 12.4 to 14.0**

Characteristic	Value
Freeway	A29
Location	12.4-14.0
Average week day intensity freeway	40128 vehicles/week day
Average fraction freight traffic	11.0%
Length acceleration lane	258 meter
Average week day intensity acceleration lane	6488 vehicles/week day
Upstream discontinuity	Deceleration lane
Downstream discontinuity	Diverge segment
Remarks	Tunnel

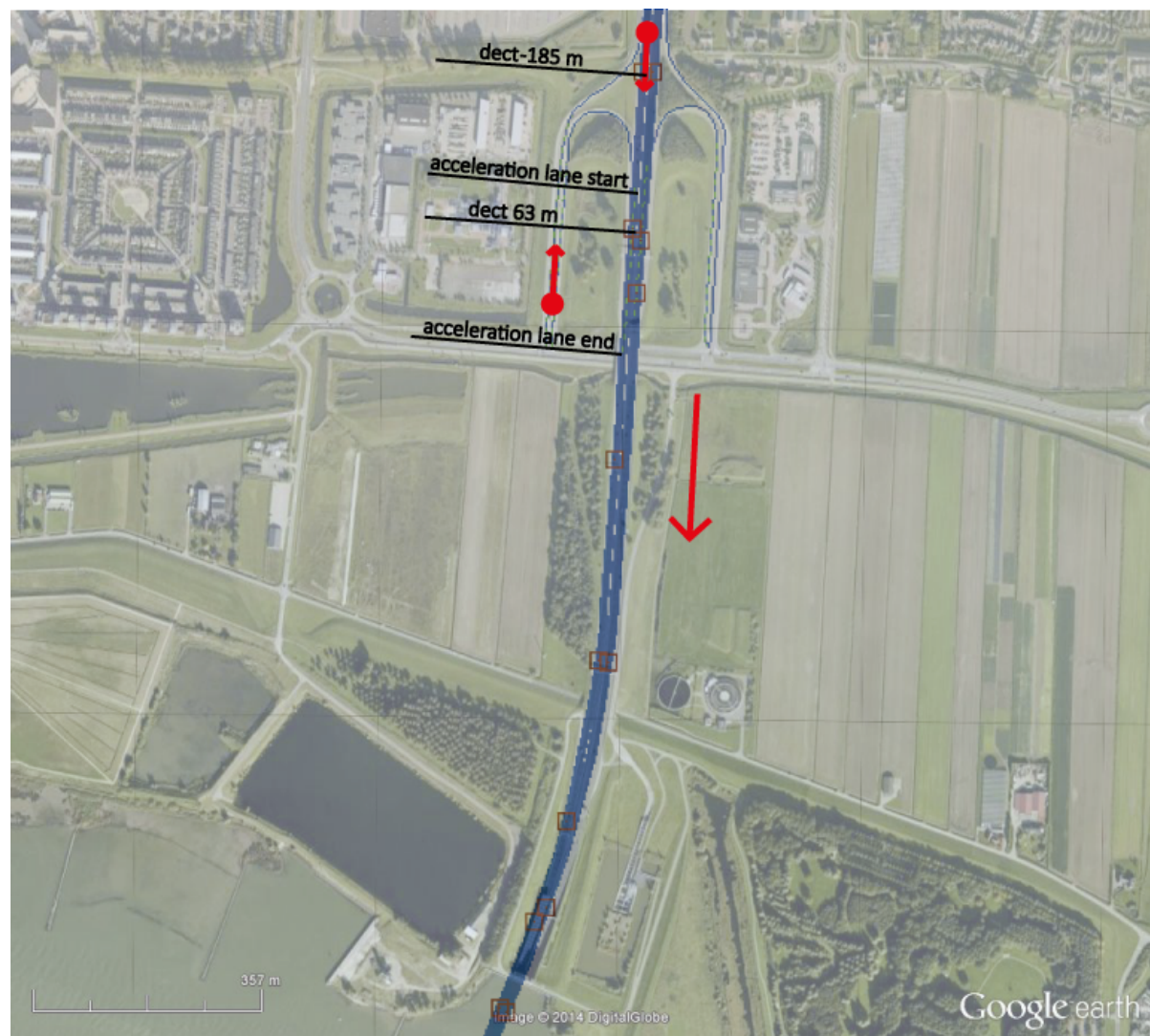
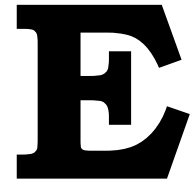


Figure D.12: Situation A29 12.4-14.0 km



## Used Matlab scripts

This appendix consist of the developed scripts in this thesis. This appendix is divided in two main sections. One for the basic freeway segment and the acceleration lane. The main scripts of these two type of segments are quite similar, however, there are some important differences in these scripts. Therefore is chosen to publish the main scripts for both type of segments. The scripts of one type of segment consists of three parts with function scripts. The main scripts is the backbone of the calculations and the function scripts are parts of scripts which are used several times and therefore used as functions.

### E.1. Scripts for basic freeway segments

#### E.1.1. Step 1 - Find interesting locations and detectors and collect data

##### Main script

```
% Version 2014-10-21 by M.J. Hovenga

function locations = ...
    DanteExportZoekRefDect_v20140806(n, speed, rijstroken_hoofdrij)
tic

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SETTING REQUIREMENTS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
requirements.speedlimit = speed;
requirements.lanes_mainroad = rijstroken_hoofdrij;
requirements.type_discontinuity = 'basicfreewaysegment';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRESS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Display used scope as control
disp(['Input: ' requirements.type_discontinuity ' ' ...
    '; ' ...
    num2str(requirements.speedlimit) ' km/h, ' ...
    num2str(requirements.lanes_mainroad) ...
    ' rijstroken hoofdrijbaan'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FINDING LINKS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
location_number = 0;
% Selecting links in the network for investigation
for i = 0:n.nElements - 1
    basicfreewaysegment_el = n.elements.get(i);
    if instanceOf(basicfreewaysegment_el,...
        'nl.fileradar.dante.export.graph.ELink')
```

```

        location_number = location_number + 1;
        % Give ever found link a number for analysis
        locations(location_number).location_number = location_number;
        % save id in network
        locations(location_number).path_mainlink = basicfreewaysegment_el;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRESS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Display process script to see if everything is going well
[dim0 number_of_locations] = size(locations);
disp([num2str(number_of_locations) ...
    ' potentile basicfreewaysegments gevonden in netwerk'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Links with errors deletion %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Links without data deletion, add characteristics as the number of lanes
% and freeway number
[dim0 number_of_locations] = size(locations);
for i = linspace(number_of_locations,1,number_of_locations)
    mainlink = locations(i).path_mainlink;
    try
        locations(i).speedlimit = ...
            mainlink.getProperty('NWBLink.speedLimit').asFloat();
        locations(i).lanes_mainroad = ...
            mainlink.getProperty('NWBLink.nLanes').asInt();
        locations(i).roadnumber = ...
            mainlink.getProperty('NWBLink.roadNumber').asInt();
    catch
        % Delete location without these characteristics
        locations(i) = [];
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SEARCH MAINLINK AND FILTER %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
% Define more characteristics
for i = 1:number_of_locations
    mainlink = locations(i).path_mainlink;
    locations(i).speedlimit = ...
        mainlink.getProperty('NWBLink.speedLimit').asFloat();
    locations(i).lanes_mainroad = ...
        mainlink.getProperty('NWBLink.nLanes').asInt();
    locations(i).roadnumber = ...
        mainlink.getProperty('NWBLink.roadNumber').asInt();
    locations(i).mainlink_length = linkLength(mainlink);
    % Determine fromKM and toKM well, not well implemented in Dante
    if mainlink.getProperty('NWBLink.fromDist').asInt() > 0
        locations(i).fromKM = ...
            mainlink.getProperty('NWBLink.toKm').asFloat();
        locations(i).toKM = ...
            mainlink.getProperty('NWBLink.fromKm').asFloat();
        locations(i).fromDist = ...
            mainlink.getProperty('NWBLink.toDist').asInt();
        locations(i).toDist = ...
            mainlink.getProperty('NWBLink.fromDist').asInt();
    else
        locations(i).fromKM = ...
            mainlink.getProperty('NWBLink.fromKm').asFloat();
        locations(i).toKM = ...
            mainlink.getProperty('NWBLink.toKm').asFloat();
        locations(i).fromDist = ...
            mainlink.getProperty('NWBLink.fromDist').asInt();
        locations(i).toDist = ...
            mainlink.getProperty('NWBLink.toDist').asInt();
    end
end

```



[illegible]

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Find carriageway detectors on selected links
for i = 1:number_of_locations
    main = locations(i).path_mainlink;
    dects = zoekConnection('ECarriageWayDetector',main);
    [r c] = size(dects);
    for j = 1:r
        locations(i).detectors(j).path_detector = dects{j,1}(1,1);
    end
end
% Add characteristics of carriageway detector
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        detector = locations(i).detectors(j).path_detector;
        locations(i).detectors(j).relative_position = detector.getProperty ...
            ('NDWCarriageWayMeasurementPoint.relativePositionOnLink').asFloat();
        locations(i).detectors(j).position_upstream = ...
            locations(i).detectors(j).relative_position * ...
            locations(i).mainlink_length;
        locations(i).detectors(j).position_downstream = ...
            (1 - locations(i).detectors(j).relative_position) * ...
            locations(i).mainlink_length;
    end
end
% Filtering detectors based on position
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = linspace(number_of_detectors,1,number_of_detectors)
        if locations(i).detectors(j).position_downstream < ...
            locations(i).influencelength_downstream_used
            locations(i).detectors(j) = [];
        elseif locations(i).detectors(j).position_upstream < ...
            locations(i).influencelength_upstream_used
            locations(i).detectors(j) = [];
        end
    end
end
% Filtering locations with no carriageway detectors
for i = linspace(number_of_locations,1,number_of_locations)
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        if isempty(locations(i).detectors)
            locations(i) = [];
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OPHALEN LANEDECTS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Find lane detectors of the carriageway detectors
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    number_of_detectors = [];
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = linspace(number_of_detectors,1,number_of_detectors)
        name = [];
        detectors = locations(i).detectors(j).path_detector;
        lane_dect = detectors.connections.get(0).to;
        name = ...
            char(lane_dect.getProperty ...
                ('NDWLaneMeasurementPoint.laneSiteID').asString());
        name = name(1:end-6);
        locations(i).detectors(j).detector_name = name;
        lane_dects = zoekConnection('ELaneDetector',detectors);
        [number_of_lanedetectors dim0] = size(lane_dects);
        % When the number of lane detectors is not correct carriageway
        % detector is not selected
        % This control object can be used to determine the number of lanes
        % better for weaving, diverging and merging sections.
    end
end

```



```

        if number_of_lanedetectors == locations(i).lanes_mainroad
            for k = 1:number_of_lanedetectors
                lanenumber = [];
                path_lanedetector = lane_dects{k,1}(1,1);
                lanetext = ...
                    char(path_lanedetector.getProperty...
                        ('NDWLaneMeasurementPoint.lane').asString());
                lanenumber = str2num(lanetext(end:end));
                locations(i).detectors(j).lane(lanenumber).path_lanedetector ...
                    = path_lanedetector;
                locations(i).detectors(j).lane(lanenumber).lanenumber = ...
                    lanenumber;
            end
        else
            locations(i).detectors(j) = [];
        end
    end
end

% Filtering locations without detectors after the check of the lane
% detectors
for i = linspace(number_of_locations,1,number_of_locations)
    if isempty(locations(i).detectors)
        locations(i) = [];
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRESS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Return status
[dim0 number_of_locations] = size(locations);
disp(['Filtering op detectoren, nog ' ...
    num2str(number_of_locations) ' locaties over...'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SET COLORS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Add colors to the different locations
[dim0 number_of_locations] = size(locations);
cmap = distinguishable_colors(number_of_locations);
for i = 1:number_of_locations
    locations(i).color = cmap(i,:);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PREPARE LOAD DATA %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% preparation input data days
list_raw = dir([cd '\2*.dpnz']);
[number_of_days dim0] = size(list_raw);
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    for j = 1:number_of_days
        locations(i).dates(j).name = list_raw(j).name;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LOAD DATA %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V3 - per dag wegschrijven
[number_of_days dim0] = size(list_raw);
% Empty vector for speed
for loc = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(loc).detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            for day = 1:number_of_days
                locations(loc).detectors(dect).lane(lane).day(day).data.speed ...
                    = [];
            end
        end
    end
end

```

```

        locations(loc).detectors(dect).lane(lane).day(day).data.flow ...
            = [];
    end
end
end
end
% Fill vectors
for day = 1:number_of_days
    % Per Day
    filename = [list_raw(day).name];
    n.loadData(filename)
    for loc = 1:number_of_locations
        % Per Location
        [dim0 number_of_detectors] = size(locations(loc).detectors);
        for dect = 1:number_of_detectors
            % Per Detector
            for lane = 1:locations(loc).lanes_mainroad
                % Per Lanedetector
                flow_array = [];
                speed_array = [];
                lane_detector = ...
                    locations(loc).detectors(dect).lane(lane).path_lanedetector;
                flow = lane_detector.properties.get(1);
                speed = lane_detector.properties.get(2);
                flow_array = double(flow.values);
                flow_array = flow_array(400:1100,1);
                speed_array = double(speed.values);
                speed_array = speed_array(400:1100,1);
                locations(loc).detectors(dect).lane(lane).day(day).data.speed = ...
                    speed_array;
                locations(loc).detectors(dect).lane(lane).day(day).data.flow = ...
                    flow_array;
            end
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CONTAINERS REQUIREMENTS MERGE %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    locations(i).requirements = requirements;
    locations(i).path_mainline = [];
    locations(i).path_mainlink = [];
    locations(i).upstream.path_node = [];
    locations(i).upstream.path_mainlink = [];
    locations(i).downstream.path_node = [];
    locations(i).downstream.path_mainlink = [];
    locations(i).path_node = [];
    locations(i).path_mainlink = [];
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        locations(i).detectors(j).path_detector = [];
        for k = 1:locations(i).lanes_mainroad
            locations(i).detectors(j).lane(k).path_lanedetector = [];
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVE %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
year = locations(1).dates(1).name;
year = year(1:4);
FileNameLoc = ['stap1_ref_' num2str(year) '_detectors_bfs_' ...
    num2str(requirements.speedlimit) '_' ...
    num2str(requirements.lanes_mainroad) '_' date 'mat'];
save(FileNameLoc, 'locations')

format shortG

```

```
t = toc;
disp(datestr(datetime(0,0,0,0,0,t), 'HH:MM:SS'))
clear all
```

### Required function scripts

Script to receive the types of line connected to a link.

```
% Version 2014-10-21 by M.J. Hovenga

function output = zoekType(type, link)
output = [];
line = zoekConnection('ELine', link);
linesize = length(line);
for j = linspace(linesize,1,linesize)
    maintype = getLineType(line{j,1});
    if strcmp(maintype, type)
        line(j,:) = [];
    end
end
if linesize > length(line)
    output = type;
end
```

Script to receive the connections of a link filtered by type.

```
% Version 2014-10-21 by M.J. Hovenga

function connection = zoekConnection(type, main)
% deze functie gaat zoeken naar connections met als filter type, uitkomst is
% een cell array met alle connections met dezelfde type
connect_type = ['nl.fileradar.dante.export.graph.' type];
j = 1;
connection = [];
for i = 0:main.connections.size()-1
    if instanceof(main.connections.get(i).to, connect_type)
        connection{j,1} = main.connections.get(i).to;
        j = j + 1;
    end
end
```

## E.1.2. Step 2 - Merge April 2013 and 2014

### Main script

```
function locations = samenvoegenDATA1314_v20140911(varargin)
% Version 2014-09-11 by M.J. Hovenga
% Upstream eruit gehaald
% 5 veh/km bins
% average speed
% average weighted speed
% Space mean speed deleted

number_of_bins = 30;
size_of_bin = 5;

%%%%%%%%%
% LOAD %
%%%%%%%%%
% Inladen files uit stap 2
[file, path, filter] = ...
    uigetfile('stap1*2013*.mat', 'Please select a mat file (cancel for none)',cd);
dataFile = [path file];
load(dataFile)

locations2013 = locations;
```

```

clear locations file path filter

% Inladen files uit stap 2
[file , path, filter] = ...
    uigetfile('stap1*2014*.mat', 'Please select a mat file (cancel for none)',cd);
dataFile = [path file];
load(dataFile)

locations2014 = locations;
clear locations file path filter

dates = dataFile(end-38:end-4);

tic

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MERGING %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the easy approach, both (April 2013 and 2014) had the same
% locations, in the future, if there are differences adjustment of the code
% is needed.
[dim0,number_of_locations_2013] = size(locations2013);
[dim0,number_of_locations_2014] = size(locations2014);
if number_of_locations_2013 == number_of_locations_2014
    locations = locations2013;
    number_of_locations = number_of_locations_2013;
    for loc = 1:number_of_locations
        if locations2013(loc).toKM == ...
            locations2014(loc).toKM && locations2013(loc).fromKM == ...
            locations2014(loc).fromKM
            [dim0, number_of_detectors] = size(locations(loc).detectors);
            for dect = 1:number_of_detectors
                for lane = 1:locations(loc).lanes_mainroad
                    [dim0, number_of_days] = ...
                        size(locations(loc).detectors(dect).lane(lane).day);
                    [dim0, number_of_days2014] = ...
                        size(locations2014(loc).detectors(dect).lane(lane).day);
                    for days = 1:number_of_days2014
                        totalday = days + number_of_days;
                        locations(loc).detectors(dect).lane(lane).day(totalday).data.speed ...
                            = locations2014(loc).detectors(dect).lane(lane).day(days).data.speed;
                        locations(loc).detectors(dect).lane(lane).day(totalday).data.flow ...
                            = locations2014(loc).detectors(dect).lane(lane).day(days).data.flow;
                    end
                end
            end
        end
    end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CHECK ON EMPTY VECTORS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0, number_of_locations] = size(locations);
for loc = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(loc).detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = ...
                size(locations(loc).detectors(dect).lane(lane).day);
            for days = 1:number_of_days
                if isempty(locations(loc).detectors(dect).lane(lane).day(days).data.speed)
                    locations(loc).detectors(dect).lane(lane).day(days).data.speed ...
                        = ones(701,1) * NaN;
                end
                if isempty(locations(loc).detectors(dect).lane(lane).day(days).data.flow)
                    locations(loc).detectors(dect).lane(lane).day(days).data.flow ...
                        = ones(701,1) * NaN;
                end
            end
        end
    end
end
end

```

```

end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FILTER RAW DATA %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INCORRECTE WAARDEN AANPASSEN
% Speeds: Incorrecte data NaN maken
% Speeds: Nulmetingen (-1) 0 maken
% Speeds: Flows is NaN => Speeds is NaN
% Flows: Speeds is NaN => Flows is NaN
% Flows : Incorrecte data NaN maken
% Speeds: Groter dan 220 NaN maken
% Speeds: Kleiner dan 80 NaN maken
[dim0, number_of_locations] = size(locations);
for loc = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(loc).detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = size(locations(loc).detectors(dect).lane(lane).day);
            for days = 1:number_of_days
                filter = ...
                    locations(loc).detectors(dect).lane(lane).day(days).data.speed ...
                    == 0;
                locations(loc).detectors(dect).lane(lane).day(days).data.speed(filter) ...
                = NaN;
                filter = [];
                filter = ...
                    locations(loc).detectors(dect).lane(lane).day(days).data.speed == -1;
                locations(loc).detectors(dect).lane(lane).day(days).data.speed(filter) ...
                = 0;
                filter = [];
                filter = ...
                    locations(loc).detectors(dect).lane(lane).day(days).data.flow < 0;
                locations(loc).detectors(dect).lane(lane).day(days).data.flow(filter) ...
                = NaN;
                filter = [];
                filter = ...
                    locations(loc).detectors(dect).lane(lane).day(days).data.speed > 220;
                locations(loc).detectors(dect).lane(lane).day(days).data.speed(filter) ...
                = NaN;
                filter = [];
                filter = ...
                    locations(loc).detectors(dect).lane(lane).day(days).data.speed < 80;
                locations(loc).detectors(dect).lane(lane).day(days).data.speed(filter) ...
                = NaN;
                filter = ...
                    isnan(locations(loc).detectors(dect).lane(lane).day(days).data.speed);
                locations(loc).detectors(dect).lane(lane).day(days).data.flow(filter) ...
                = NaN;
                filter = [];
                filter = ...
                    isnan(locations(loc).detectors(dect).lane(lane).day(days).data.flow);
                locations(loc).detectors(dect).lane(lane).day(days).data.speed(filter) ...
                = NaN;
                filter = [];
            end
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DETERMINE DENSITIES %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine Densities
% Density = Flow / Speed
% Speed = Time Mean Speed!
[dim0, number_of_locations] = size(locations);
for loc = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(loc).detectors);

```



[illegible]

```

% Determine percentage flow %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        for k = 1:locations(i).lanes_mainroad
            locations(i).detectors(j).lane(k).data.percentageflow = ...
                locations(i).detectors(j).lane(k).data.flow ./ ...
                locations(i).detectors(j).data.totalflow;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NaN Filtering %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        for k = 1:locations(i).lanes_mainroad
            for l = 1:number_of_bins
                blokken = locations(i).detectors(j).data.bin(l).blokken;
                typedata = locations(i).detectors(j).lane(k).data.flow;
                locations(i).detectors(j).lane(k).data.bin(l).flow = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).lane(k).data.speed;
                locations(i).detectors(j).lane(k).data.bin(l).speed = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).lane(k).data.density;
                locations(i).detectors(j).lane(k).data.bin(l).density = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = ...
                    locations(i).detectors(j).lane(k).data.percentageflow;
                locations(i).detectors(j).lane(k).data.bin(l).percentageflow = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).data.totalflow;
                locations(i).detectors(j).data.bin(l).totalflow = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).data.totaldensity;
                locations(i).detectors(j).data.bin(l).totaldensity = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).data.averagespeed;
                locations(i).detectors(j).data.bin(l).averagespeed = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = ...
                    locations(i).detectors(j).data.averageweightedspeed;
                locations(i).detectors(j).data.bin(l).averageweightedspeed = ...
                    dataBlokkenVerwerking(typedata, blokken);
            end
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Empty containers fill with NaN %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        for k = 1:locations(i).lanes_mainroad
            for l = 1:number_of_bins
                if isempty(locations(i).detectors(j).lane(k).data.bin(l).flow);
                    locations(i).detectors(j).lane(k).data.bin(l).flow = ...
                        ones(2,1) * NaN;
                end
                if isempty(locations(i).detectors(j).lane(k).data.bin(l).speed);
                    locations(i).detectors(j).lane(k).data.bin(l).speed = ...
                        ones(2,1) * NaN;
                end
            end
        end
    end
end

```



```

end
if isempty(locations(i).detectors(j).lane(k).data.bin(1).density);
    locations(i).detectors(j).lane(k).data.bin(1).density = ...
        ones(2,1) * NaN;
end
if isempty(locations(i).detectors(j).lane(k).data.bin(1).percentageflow);
    locations(i).detectors(j).lane(k).data.bin(1).percentageflow = ...
        ones(2,1) * NaN;
end
if isempty(locations(i).detectors(j).data.bin(1).totalflow);
    locations(i).detectors(j).data.bin(1).totalflow = ...
        ones(2,1) * NaN;
end
if isempty(locations(i).detectors(j).data.bin(1).totaldensity);
    locations(i).detectors(j).data.bin(1).totaldensity = ...
        ones(2,1) * NaN;
end
if isempty(locations(i).detectors(j).data.bin(1).averagespeed);
    locations(i).detectors(j).data.bin(1).averagespeed = ...
        ones(2,1) * NaN;
end
if isempty(locations(i).detectors(j).data.bin(1).averageweightedspeed);
    locations(i).detectors(j).data.bin(1).averageweightedspeed = ...
        ones(2,1) * NaN;
end
end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DETERMINING MEDIAN AND STD %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Add color
[dim0 number_of_locations] = size(locations);
cmap = distinguishable_colors(number_of_locations);
for i = 1:number_of_locations
    locations(i).color = cmap(i,:);
end
% medians berekenen
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        for k = 1:locations(i).lanes_mainroad
            for l = 1:number_of_bins
                % Medians
                locations(i).detectors(j).data.bin(l).median.totalflow = ...
                    median(locations(i).detectors(j).data.bin(l).totalflow);
                locations(i).detectors(j).data.bin(l).median.totaldensity = ...
                    median(locations(i).detectors(j).data.bin(l).totaldensity);
                locations(i).detectors(j).lane(k).data.bin(l).median.percentageflow = ...
                    median(locations(i).detectors(j).lane(k).data.bin(l).percentageflow);
                locations(i).detectors(j).lane(k).data.median.percentageflow(1,1) = ...
                    locations(i).detectors(j).lane(k).data.bin(l).median.percentageflow;
                locations(i).detectors(j).lane(k).data.bin(l).median.flow = ...
                    median(locations(i).detectors(j).lane(k).data.bin(l).flow);
                locations(i).detectors(j).lane(k).data.median.flow(1,1) = ...
                    locations(i).detectors(j).lane(k).data.bin(l).median.flow;
                locations(i).detectors(j).lane(k).data.bin(l).median.density = ...
                    median(locations(i).detectors(j).lane(k).data.bin(l).density);
                locations(i).detectors(j).lane(k).data.median.density(1,1) = ...
                    locations(i).detectors(j).lane(k).data.bin(l).median.density;
                locations(i).detectors(j).lane(k).data.bin(l).median.speed = ...
                    median(locations(i).detectors(j).lane(k).data.bin(l).speed);
                locations(i).detectors(j).lane(k).data.median.speed(1,1) = ...
                    locations(i).detectors(j).lane(k).data.bin(l).median.speed;
                locations(i).detectors(j).data.bin(l).median.averagespeed = ...
                    median(locations(i).detectors(j).data.bin(l).averagespeed);
                locations(i).detectors(j).data.bin(l).median.averageweightedspeed = ...
                    median(locations(i).detectors(j).data.bin(l).averageweightedspeed);
            end
        end
    end
end
end

```

```

        locations(i).detectors(j).lane(k).data.median.afstand(1,1) = ...
            locations(i).detectors(j).position_upstream;
        locations(i).detectors(j).lane(k).data.median.bin(1,1) = ...
            1 * size_of_bin - (size_of_bin / 2);
    %Std
    locations(i).detectors(j).data.bin(1).std.totalflow = ...
        std(locations(i).detectors(j).data.bin(1).totalflow);
    locations(i).detectors(j).data.bin(1).std.totaldensity = ...
        std(locations(i).detectors(j).data.bin(1).totaldensity);
    locations(i).detectors(j).lane(k).data.bin(1).std.percentageflow = ...
        std(locations(i).detectors(j).lane(k).data.bin(1).percentageflow);
    locations(i).detectors(j).lane(k).data.std.percentageflow(1,1) = ...
        locations(i).detectors(j).lane(k).data.bin(1).std.percentageflow;
    locations(i).detectors(j).lane(k).data.bin(1).std.flow = ...
        std(locations(i).detectors(j).lane(k).data.bin(1).flow);
    locations(i).detectors(j).lane(k).data.std.flow(1,1) = ...
        locations(i).detectors(j).lane(k).data.bin(1).std.flow;
    locations(i).detectors(j).lane(k).data.bin(1).std.density = ...
        std(locations(i).detectors(j).lane(k).data.bin(1).density);
    locations(i).detectors(j).lane(k).data.std.density(1,1) = ...
        locations(i).detectors(j).lane(k).data.bin(1).std.density;
    locations(i).detectors(j).lane(k).data.std.afstand(1,1) = ...
        locations(i).detectors(j).position_upstream;
    locations(i).detectors(j).lane(k).data.std.bin(1,1) = ...
        1 * size_of_bin - (size_of_bin / 2);
    end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% UNFILTERED PLOTS TO DETERMINE WANTED FILTERS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot2Dpercentageflow_unfiltered(locations)
plot2Ddensity_unfiltered(locations)
plot2Dspeed_unfiltered(locations)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% APPLYING FILTERS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Filtering error-ed results
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    if ~isempty(locations(i).detectors)
        [dim0 number_of_detectors] = size(locations(i).detectors);
        for j = linspace(number_of_detectors,1,number_of_detectors)
            if locations(i).detectors(j).lane(2).data.median.percentageflow(12,1) ...
                < 0.3
                locations(i).detectors(j) = [];
            elseif max(locations(i).detectors(j).lane(2).data.median.speed) ...
                > 120
                locations(i).detectors(j) = [];
            elseif max(locations(i).detectors(j).lane(3).data.median.speed) ...
                > 120
                locations(i).detectors(j) = [];
            elseif locations(i).detectors(j).lane(2).data.median.percentageflow(3,1) ...
                < 0.2
                locations(i).detectors(j) = [];
            elseif locations(i).detectors(j).lane(1).data.median.density(7,1) ...
                < 3.2
                locations(i).detectors(j) = [];
            end
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FILTERED PLOTS TO CONTROL FILTERS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot2Dpercentageflow_filtered(locations)

```

```

plot2Ddensity_filtered(locations)
plot2Dspeed_filtered(locations)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVE %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

FileNameloc = ['stap2_loc_' dates '.mat'];
save(FileNameloc, 'locations', '-v7.3')

format shortG
t = toc;
disp(datestr(datemum(0,0,0,0,t), 'HH:MM:SS'))
clear all

```

### Required function scripts

Script to divide the data into a specific bin.

```

% Version 2014-10-21 by M.J. Hovenga
function output = dataBlokkenVerwerking(typedata, blokken)
blok = typedata .* blokken;
filter = isnan(blok);
blok(filter) = -1;
blok(any(blok == -1,2), :) = [];
output = blok;

```

Script to plot the detector data over all bins.

This script is also used for the filtered versions and for the other detectors, small changes are required for this usage.

```

% Version 2014-10-21 by M.J. Hovenga
function nargin = plot2Dpercentageflow_unfiltered(locations)
% Upstream eruit halen

[dim0 number_of_locations] = size(locations);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Outside lane %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('visible','off')
xlabel('total density (veh/km)')
ylabel('fraction of flow (%)')
hold on
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        X = locations(i).detectors(j).lane(3).data.median.afstand;
        Y = locations(i).detectors(j).lane(3).data.median.bin;
        Z = locations(i).detectors(j).lane(3).data.median.percentageflow;
        color = locations(i).color;
        plot(Y,Z,'Color',color)
    end
end
hold off
ylim([0 1])
xlim([0 100])
drm = ['unfiltered\2DPLOTS\'];
fname = ['fraction of flow - outside lane'];
exportfig(fname,drm)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Center lane %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('visible','off')
xlabel('total density (veh/km)')
ylabel('fraction of flow (%)')
hold on

```

```

for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        X = locations(i).detectors(j).lane(2).data.median.afstand;
        Y = locations(i).detectors(j).lane(2).data.median.bin;
        Z = locations(i).detectors(j).lane(2).data.median.percentageflow;
        color = locations(i).color;
        plot(Y,Z, 'Color',color)
    end
end
hold off
ylim([0 1])
xlim([0 100])
drm = [ 'unfiltered\2DPLOTS\ '];
fname = [ 'fraction of flow - center lane' ];
exportfig(fname,drm)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Median Lane %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('visible','off')
xlabel('total density (veh/km)')
ylabel('fraction of flow (%)')
hold on
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        X = locations(i).detectors(j).lane(1).data.median.afstand;
        Y = locations(i).detectors(j).lane(1).data.median.bin;
        Z = locations(i).detectors(j).lane(1).data.median.percentageflow;
        color = locations(i).color;
        plot(Y,Z, 'Color',color)
    end
end
hold off
ylim([0 1])
xlim([0 100])
drm = [ 'unfiltered\2DPLOTS\ '];
fname = [ 'fraction of flow - median lane' ];
exportfig(fname,drm)

```

```

% EXPORT LIST %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
id = 1;
for i = 1:number_of_locations
    if ~isempty(locations(i).detectors)
        LIJST{id,1} = locations(i).roadnumber;
        LIJST{id,2} = locations(i).fromKM;
        LIJST{id,3} = locations(i).toKM;
        LIJST{id,4} = locations(i).location_number;
        id = id + 1;
    end
end
LIJST = sortrows(LIJST,2);
LIJST = sortrows(LIJST);
HEADERS{1,1} = ['wegnr'];
HEADERS{1,2} = ['vanKM'];
HEADERS{1,3} = ['naarKM'];
HEADERS{1,4} = ['location_number'];
LIJST = [HEADERS; LIJST];
delete('locaties.xls');
xlswrite('locaties.xls',LIJST);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ADDING MANUAL FOUND CHARACTERISTICS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[ndata, text, exceldata] = xlsread('locaties_traject.xls');
[dim0 number_of_locations] = size(locations);
[r c] = size(exceldata);
for i = linspace(number_of_locations,1,number_of_locations)
    location_number = locations(i).location_number;
    for j = 1:r
        if location_number == exceldata{j,4}
            locations(i).characteristics.traject = exceldata{j,5};
        end
    end
end
% Delete unwanted locations
if location_number == 4218 | location_number == 10109 | location_number == 10482
    locations(i) = [];
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% EXPORT LIST TRAJECT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
LIJST = [];
id = 1;
for i = 1:number_of_locations
    if ~isempty(locations(i).detectors)
        LIJST{id,1} = locations(i).roadnumber;
        LIJST{id,2} = ['A' num2str(locations(i).roadnumber)];
        LIJST{id,3} = locations(i).fromKM;
        LIJST{id,4} = locations(i).toKM;
        [dim0 number_of_detectors] = size(locations(i).detectors);
        LIJST{id,5} = number_of_detectors;
        LIJST{id,6} = locations(i).characteristics.traject;
        id = id + 1;
    end
end
LIJST = sortrows(LIJST,3);
LIJST = sortrows(LIJST,1);
LIJST(:,1) = [];
HEADERS{1,1} = ['roadnumber'];
HEADERS{1,2} = ['from KM'];
HEADERS{1,3} = ['to KM'];
HEADERS{1,4} = ['#detectors'];
HEADERS{1,5} = ['fixed average speed check'];
LIJST = [HEADERS; LIJST];
delete('locaties_total.xls');
xlswrite('locaties_total.xls',LIJST);

```

[illegible]

```

for lane = 1:3
    lineplot_percentage_flow_median(locations, lane, bin)
    lineplot_density_median(locations, lane, bin)
    lineplot_speed_median(locations, lane, bin)
end
lineplot_total_flow_median(locations, bin)
lineplot_average_weighted_speed_median(locations, bin)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MAKE DISTRIBUTIONS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for lane = 1:3
    % Density
    BFS_histo_density(locations, lane, bin)
    % Percentage flow
    BFS_histo_percentage_flow(locations, lane, bin)
    % lane speed
    BFS_histo_speed(locations, lane, bin)
end
% Total flow
BFS_histo_total_flow(locations, bin)
% Average speed
BFS_histo_average_weighted_speed(locations, bin)

% Vergelijking Traject
for lane = 1:3
    BFS_histo_density_raw(locations, lane, bin)
    BFS_histo_percentage_flow_raw(locations, lane, bin)
    BFS_histo_speed_raw(locations, lane, bin)
end
BFS_histo_total_flow_raw(locations, bin)
BFS_histo_average_weighted_speed_raw(locations, bin)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Maak plaatje locations %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
combinating_locations_acceleration_lane(locations)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVE %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
FileNameloc = ['stap3_BFS.mat'];
save(FileNameloc, 'locations', '-v7.3')

```

### Required function scripts

#### Plot script line plots

This script is also used for the other indicators in a slightly changed version.

```

function nargin = lineplot_percentage_flow_median(locations, lane, bin)
% Version 2014-10-21 by M.J. Hovenga

close all

YLABEL = 'fraction of flow (%)';
XLABEL = 'distance (m)';
XLIM = [500 5000];
YLM = [0.1 0.5];
if lane == 1
    strlane = 'median lane';
elseif lane == 2
    strlane = 'center lane';
elseif lane == 3
    strlane = 'outside lane';
end

[dim0 number_of_locations] = size(locations);
figure('visible','off','Position',[1 1 1920/2 1080/1])
hold on

```

```

num = 1;
% Validated detectors within region
for i = 1:number_of_locations
    TEMP = [];
    X_distance = [];
    Y_percentage_flow = [];
    COLOR = locations(i).color;
    if ~isempty(locations(i).detectors)
        [dim0 number_of_detectors] = size(locations(i).detectors);
        for j = 1:number_of_detectors
            Y_percentage_flow = ...
                [Y_percentage_flow; ...
                 locations(i).detectors(j).lane(lane).data.bin(bin).median.percentage_flow];
            X_distance = [X_distance; locations(i).detectors(j).position_upstream];
        end
    end
    if ~isempty(X_distance)
        TEMP = [X_distance Y_percentage_flow];
        TEMP = sortrows(TEMP);
        TEMP(any(isnan(TEMP),2),:) = [];
        X_distance = TEMP(:,1);
        Y_percentage_flow = TEMP(:,2);
        plot(X_distance, Y_percentage_flow, ...
             'Color', COLOR, 'LineWidth', 1.4, 'Marker', '.', 'MarkerSize', 25)
    end
    A(num) = plot([-1000 -2000], [-5 -5], ...
                 'Color', COLOR, 'LineWidth', 1.4, 'Marker', '.', 'MarkerSize', 25);
    B(num,1) = locations(i).roadnumber;
    B(num,2) = locations(i).fromKM;
    B(num,3) = locations(i).toKM;
    num = num + 1;
end
length_num = size(A,2);
for i = 1:length_num
    PLOTS(1,i) = A(i);
    TEXTS{1,i} = ['A' num2str(B(i,1)) ' ' ...
                  num2str(B(i,2)) '- ' num2str(B(i,3)) 'km'];
end
line([0 0],[-999999 999999], 'LineStyle', '--', 'Color', 'k')
hold off
legend(PLOTS, TEXTS, 'Location', 'eastoutside')
ylim(YLIM)
xlim(XLIM)
grid on
xlabel(XLABEL)
ylabel(YLABEL)
drm = ['lineplot/median/percentage_flow/'];
fname = ['fraction of flow - ' strlane];
exportfig(fname, drm)
close all

```

### Plot script histograms and estimated distribution mle

This script is also used for the other indicators in a slightly changed version.

```

function nargin = BFS_histo_percentage_flow(locations, lane, bin)

close all

XLABEL = 'fraction of flow';
YLABEL = 'f';
XLIM = [0 0.7];
YLIM = [0 0.1];

figure('visible', 'off', 'Position', [1 1 1920/2 1080/2.5])
binranges = 0:2:100;
bincount = zeros(51,1);
TEMP = [];
[dim0, number_of_locations] = size(locations);
for i = 1:number_of_locations

```



```

[dim0, number_of_detectors] = size(locations(i).detectors);
for j = 1:number_of_detectors
    X = ...
        locations(i).detectors(j).lane(lane).data.bin(bin).percentageflow*100;
    bincount_add = histc(X,binranges);
    bincount = bincount + bincount_add;
    TEMP = [TEMP; ...
        locations(i).detectors(j).lane(lane).data.bin(bin).percentageflow*100];
end
end
AREA = trapz(binranges,bincount);
num = 1;
for i = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        X = locations(i).detectors(j).lane(lane).data.bin(bin).percentageflow*100;
        bincount_min = histc(X,binranges);
        hold on
        bar(binranges/100, bincount/AREA, ...
            'stacked','FaceColor',locations(i).color,'LineWidth',0.25);
        bincount = bincount - bincount_min;
    end
    A(num) = bar(binranges-9999, bincount/AREA, ...
        'stacked','FaceColor',locations(i).color,'LineWidth',0.25);
    B(num,1) = locations(i).roadnumber;
    B(num,2) = locations(i).fromKM;
    B(num,3) = locations(i).toKM;
    num = num + 1;
end
MEDIAN = median(TEMP);
STD = std(TEMP);
SKEW = skewness(TEMP);
KURT = kurtosis(TEMP);
phat = mle(TEMP,'distribution','norm');
MEAN1 = phat(1);
STD1 = phat(2);
binranges = 0:0.1:150;
C = plot(binranges/100,normpdf(binranges,MEAN1,STD1),'r-','LineWidth',2);
C1 = plot(0.999999,'-w');
C2 = plot(0.999999,'-w');
C3 = plot(0.999999,'-w');
C4 = plot(0.999999,'-w');
length_num = size(A,2);
for i = 1:length_num
    PLOTS(1,i) = A(i);
    TEXTS{1,i} = ['A' num2str(B(i,1)) ' ' num2str(B(i,2)) '- ' num2str(B(i,3)) 'km'];
end
PLOTS(1,end+1) = C;
PLOTS(1,end+1) = C1;
PLOTS(1,end+1) = C2;
PLOTS(1,end+1) = C3;
PLOTS(1,end+1) = C4;
% In plaats van losse plots om tekst af te breken in de legenda de functie
% char() had ook gebruikt kunnen worden, verbeterpunt.
TEXTS{1,end+1} = 'estimated distribution: ';
TEXTS{1,end+1} = ['\mu_e= ' num2str(MEAN1)];
TEXTS{1,end+1} = ['\sigma_e= ' num2str(STD1)];
TEXTS{1,end+1} = ['skewness= ' num2str(SKEW)];
TEXTS{1,end+1} = ['kurtosis= ' num2str(KURT)];
legend(PLOTS,TEXTS,'Location','northeastoutside')
xlim(XLIM)
ylim(YLIM)
grid on
hold off
xlabel(XLABEL);
ylabel(YLABEL);
drm = ['distribution/percentageflow/'];
if lane == 1
    strlane = 'median lane';
elseif lane == 2
    strlane = 'center lane';
end

```

```
elseif lane == 3
    strlane = 'outside lane';
end
fname = ['fraction of flow - ' strlane];
exportfig(fname,drm);
close all
```

## E.2. Scripts for acceleration lanes

### E.2.1. Step 1 - Find interesting locations and detectors and collect data

#### Main script

```
function nargin = DanteExportZoekLocDect_v20140821...
    (n, speed, rijstroken_hoofdrij, rijstroken_discontinuity)
% Delete downstream link
% Delete upstream link
% Set downstream influence length at 1200m

tic

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SETTING REQUIREMENTS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
requirements.speedlimit = speed;
requirements.lanes_mainroad = rijstroken_hoofdrij;
requirements.type_discontinuity = 'invoeging';
requirements.direction = 'beneden';
requirements.length_accelerationlane = bepaalafstandinvoeging(requirements);
requirements.length_accelerationlane_minimum = ...
    requirements.length_accelerationlane * 0.75;
requirements.length_accelerationlane_maximum = ...
    requirements.length_accelerationlane * 1.25;
requirements.lanes_accelerationlane = rijstroken_discontinuity;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRESS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Display used scope as control
disp(['Input: ' requirements.type_discontinuity ' ' requirements.direction '; ' ...
    num2str(requirements.speedlimit) ' km/h, ' num2str(requirements.lanes_mainroad) ...
    ' rijstroken hoofdrijbaan, ' num2str(requirements.lanes_accelerationlane) ...
    ' rijstroken '])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FINDING ACCELERATION LANES %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
location_number = 0;
% Zoeken naar invoegingen
for i = 0:n.nElements - 1
    invoeging_el = n.elements.get(i);
    if instanceof(invoeging_el, 'nl.fileradar.dante.export.graph.ELine')
        type = getLineType(invoeging_el);
        % Filter op NWBInvoegstrook
        if strcmp(type, 'NWBInvoegstrook')
            location_number = location_number + 1;
            locations(location_number).location_number = location_number;
            locations(location_number).path_mainline = invoeging_el;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRESS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
disp([num2str(number_of_locations) ...
    ' potentile invoegingen gevonden in netwerk'])
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SEARCH MAINLINK AND FILTER %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Find all mainlinks
for i = 1:number_of_locations
    mainline = locations(i).path_mainlink;
    mainlink = zoekConnection('ELink', mainline);
    locations(i).path_mainlink = mainlink{1};
end
% Define characteristics
for i = 1:number_of_locations
    mainlink = locations(i).path_mainlink;
    mainline = locations(i).path_mainlink;
    locations(i).speedlimit = ...
        mainlink.getProperty('NWBLink.speedLimit').asFloat();
    locations(i).lanes_mainroad = ...
        mainlink.getProperty('NWBLink.nLanes').asInt();
    locations(i).lanes_accelerationlane = ...
        mainline.getProperty('NWBLInvoegStrook.nLanes').asInt();
    locations(i).discontinuity_upstream = ...
        requirements.type_discontinuity;
    locations(i).roadnumber = ...
        mainlink.getProperty('NWBLink.roadNumber').asInt();
    locations(i).mainlink_length = linkLength(mainlink);
    if mainlink.getProperty('NWBLink.fromDist').asInt() > 0
        locations(i).fromKM = mainlink.getProperty('NWBLink.toKm').asFloat();
        locations(i).toKM = mainlink.getProperty('NWBLink.fromKm').asFloat();
        locations(i).fromDist = mainlink.getProperty('NWBLink.toDist').asInt();
        locations(i).toDist = mainlink.getProperty('NWBLink.fromDist').asInt();
    else
        locations(i).fromKM = mainlink.getProperty('NWBLink.fromKm').asFloat();
        locations(i).toKM = mainlink.getProperty('NWBLink.toKm').asFloat();
        locations(i).fromDist = mainlink.getProperty('NWBLink.fromDist').asInt();
        locations(i).toDist = mainlink.getProperty('NWBLink.toDist').asInt();
    end
end
% Filter
for i = linspace(number_of_locations,1,number_of_locations)
    if locations(i).speedlimit ~= requirements.speedlimit
        locations(i) = [];
    elseif locations(i).lanes_mainroad ~= requirements.lanes_mainroad
        locations(i) = [];
    elseif locations(i).lanes_accelerationlane ~= ...
        requirements.lanes_accelerationlane
        locations(i) = [];
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRESS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
disp(['Filtering op rijstroken en snelheid, nog ' ...
    num2str(number_of_locations) ' locaties over...'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FILTER ON RUSH HOUR LANES AND OVERTAKING RULES %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:number_of_locations
    mainlink = locations(i).path_mainlink;
    InhaalVerbod = zoekType('NWBLInhaalVerbod', mainlink);
    SpitsStrook = zoekType('SpitsStrook', mainlink);
    if isempty(InhaalVerbod)
        locations(i).overtaking_rules = ['no'];
    else
        locations(i).overtaking_rules = ['yes'];
    end
    if isempty(SpitsStrook)
        locations(i).rush_hour_lane = ['no'];
    else
        locations(i).rush_hour_lane = ['yes'];
    end
end

```

```

    end
end
% Filter
for i = linspace(number_of_locations,1,number_of_locations)
    if strcmp(locations(i).rush_hour_lane,'yes')
        locations(i) = [];
    elseif strcmp(locations(i).overtaking_rules,'yes')
        locations(i) = [];
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRESS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
disp(['Filtering op spitsstroken en inhaalverboden, nog ' ...
    num2str(number_of_locations) ' locaties over...'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FILTER ON LENGTH ACCELERATION LANE %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine length
for i = 1:number_of_locations
    mainline = locations(i).path_mainline;
    locations(i).length_accelerationlane = linkLength(mainline);
end
% Filter
for i = linspace(number_of_locations,1,number_of_locations)
    if locations(i).length_accelerationlane < ...
        requirements.length_accelerationlane_minimum
        locations(i) = [];
    elseif locations(i).length_accelerationlane > ...
        requirements.length_accelerationlane_maximum
        locations(i) = [];
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRESS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
disp(['Filtering op lengte invoeging, nog ' ...
    num2str(number_of_locations) ' locaties over...'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SEARCHING DETECTORS ON LINKS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:number_of_locations
    main = locations(i).path_mainlink;
    dects = zoekConnection('ECarriageWayDetector',main);
    [r c] = size(dects);
    for j = 1:r
        locations(i).detectors(j).path_detector = dects{j,1}(1,1);
    end
end
end
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        detector = locations(i).detectors(j).path_detector;
        locations(i).detectors(j).relative_position = ...
            detector.getProperty ...
                ('NDWCarriageWayMeasurementPoint.relativePositionOnLink').asFloat();
        locations(i).detectors(j).position_upstream = ...
            locations(i).detectors(j).relative_position * ...
            locations(i).mainlink_length;
        locations(i).detectors(j).position_downstream = ...
            (1 - locations(i).detectors(j).relative_position) * ...
            locations(i).mainlink_length;
    end
end
end
% Filtering detectors

```

```

for i = 1:number_of_locations
    locations(i).influelength_upstream_used = 2400;
    locations(i).influelength_downstream_used = 1200;
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = linspace(number_of_detectors,1,number_of_detectors)
        if locations(i).detectors(j).position_downstream < ...
            locations(i).influelength_downstream_used
            locations(i).detectors(j) = [];
        elseif locations(i).detectors(j).position_upstream > ...
            locations(i).influelength_upstream_used
            locations(i).detectors(j) = [];
        end
    end
end
end
% Filtering locations
[dim0 number_of_locations] = size(locations);
for i = linspace(number_of_locations,1,number_of_locations)
    if isempty(locations(i).detectors);
        locations(i) = [];
    end
end
end
[dim0 number_of_locations] = size(locations);
for i = linspace(number_of_locations,1,number_of_locations)
    [dim0 number_of_detectors] = size(locations(i).detectors);
    if number_of_detectors == 0;
        locations(i) = [];
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OPHALEN LANEDECTS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    number_of_detectors = [];
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = linspace(number_of_detectors,1,number_of_detectors)
        name = [];
        detectors = locations(i).detectors(j).path_detector;
        lane_dect = detectors.connections.get(0).to;
        name = ...
            char(lane_dect.getProperty ...
                ('NDWLaneMeasurementPoint.laneSiteID').asString());
        name = name(1:end-6);
        locations(i).detectors(j).detector_name = name;
        lane_dects = zoekConnection('ELaneDetector',detectors);
        [number_of_lanedetectors dim0] = size(lane_dects);
        if number_of_lanedetectors == locations(i).lanes_mainroad
            for k = 1:number_of_lanedetectors
                lanenumber = [];
                path_lanedetector = lane_dects{k,1}(1,1);
                lanetext = ...
                    char(path_lanedetector.getProperty ...
                        ('NDWLaneMeasurementPoint.lane').asString());
                lanenumber = str2num(lanetext(end:end));
                locations(i).detectors(j).lane(lanenumber).path_lanedetector ...
                    = path_lanedetector;
                locations(i).detectors(j).lane(lanenumber).lanenumber ...
                    = lanenumber;
            end
        else
            locations(i).detectors(j) = [];
        end
    end
end
end
% Filtering
for i = linspace(number_of_locations,1,number_of_locations)
    if isempty(locations(i).detectors)
        locations(i) = [];
    end
end
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIND UPSTREAM LINKS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    mainlink = locations(i).path_mainlink;
    [bovennode links bovenlink bovendiscontinuïteit] = ...
        zoekDiscontinuïteitBoven(mainlink);
    locations(i).upstream.path_mainlink = bovenlink;
    locations(i).upstream.mainlink_length = linkLength(bovenlink);
    dects = zoekConnection('ECarriageWayDetector',bovenlink);
    [r c] = size(dects);
    for j = 1:r
        locations(i).upstream.detectors(j).path_detector = dects{j,1}(1,1);
        detector = locations(i).upstream.detectors(j).path_detector;
        locations(i).upstream.detectors(j).relative_position = ...
            detector.getProperty ...
                ('NDWCarriageWayMeasurementPoint.relativePositionOnLink').asFloat();
        locations(i).upstream.detectors(j).position_upstream = ...
            locations(i).upstream.detectors(j).relative_position * ...
            locations(i).upstream.mainlink_length;
        locations(i).upstream.detectors(j).position_downstream = ...
            (1 - locations(i).upstream.detectors(j).relative_position) * ...
            locations(i).upstream.mainlink_length;
    end
    try
        [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
        for j = linspace(number_of_detectors,1,number_of_detectors)
            name = [];
            detectors = locations(i).upstream.detectors(j).path_detector;
            lane_dect = detectors.connections.get(0).to;
            name = ...
                char(lane_dect.getProperty ...
                    ('NDWLaneMeasurementPoint.laneSiteID').asString());
            name = name(1:end-6);
            locations(i).upstream.detectors(j).detector_name = name;
            lane_dects = zoekConnection('ELaneDetector',detectors);
            [number_of_lanedetectors dim0] = size(lane_dects);
            if number_of_lanedetectors == locations(i).lanes_mainroad
                for k = 1:number_of_lanedetectors
                    lanenumber = [];
                    path_lanedetector = lane_dects{k,1}(1,1);
                    lanetext = char(path_lanedetector.getProperty ...
                        ('NDWLaneMeasurementPoint.lane').asString());
                    lanenumber = str2num(lanetext(end:end));
                    locations(i).upstream.detectors(j).lane(lanenumber).path_lanedetector ...
                        = path_lanedetector;
                    locations(i).upstream.detectors(j).lane(lanenumber).lanenumber ...
                        = lanenumber;
                end
            else
                locations(i).upstream.detectors(j) = [];
            end
        end
    end
    [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
    for j = linspace(number_of_detectors,1,number_of_detectors)
        if locations(i).upstream.detectors(j).position_downstream > 1200
            locations(i).upstream.detectors(j) = [];
        end
    end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRESS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
disp(['Filtering op detectoren, nog ' ...
    num2str(number_of_locations) ' locaties over...'])

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PREPARE LOAD DATA %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
list_raw = dir([cd '\2*.dpnz']);
[number_of_days dim0] = size(list_raw);
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    for j = 1:number_of_days
        locations(i).dates(j).name = list_raw(j).name;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LOAD DATA %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V3 - per dag wegschrijven
[number_of_days dim0] = size(list_raw);
% Empty vectoren maken
for loc = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(loc).detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            for day = 1:number_of_days
                locations(loc).detectors(dect).lane(lane).day(day).data.speed ...
                    = [];
                locations(loc).detectors(dect).lane(lane).day(day).data.flow ...
                    = [];
            end
        end
    end
end
try
    [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            for day = 1:number_of_days
                locations(loc).upstream.detectors(dect).lane(lane).day(day).data.speed ...
                    = [];
                locations(loc).upstream.detectors(dect).lane(lane).day(day).data.flow ...
                    = [];
            end
        end
    end
end
end
% Vectoren invullen
for day = 1:number_of_days
    % Per Day
    filename = [list_raw(day).name];
    n.loadData(filename)
    for loc = 1:number_of_locations
        % Per Location
        [dim0 number_of_detectors] = size(locations(loc).detectors);
        for dect = 1:number_of_detectors
            % Per Detector
            for lane = 1:locations(loc).lanes_mainroad
                % Per Lanedetector
                flow_array = [];
                speed_array = [];
                lane_detector = ...
                    locations(loc).detectors(dect).lane(lane).path_lanedetector;
                flow = lane_detector.properties.get(1);
                speed = lane_detector.properties.get(2);
                flow_array = double(flow.values);
                speed_array = double(speed.values);
                flow_array = flow_array(400:1100,1);
                speed_array = speed_array(400:1100,1);
                locations(loc).detectors(dect).lane(lane).day(day).data.speed ...
                    = speed_array;
                locations(loc).detectors(dect).lane(lane).day(day).data.flow ...
                    = flow_array;
            end
        end
    end
end

```

```

end
try
    [dim0 number_of_detectors] = size(locations(loc).upstream.detectors);
    for dect = 1:number_of_detectors
        % Per Detector
        for lane = 1:locations(loc).lanes_mainroad
            % Per Lanedetector
            flow_array = [];
            speed_array = [];
            lane_detector = ...
                locations(loc).upstream.detectors ...
                (dect).lane(lane).path_lanedetector;
            flow = lane_detector.properties.get(1);
            speed = lane_detector.properties.get(2);
            flow_array = double(flow.values);
            speed_array = double(speed.values);
            flow_array = flow_array(400:1100,1);
            speed_array = speed_array(400:1100,1);
            locations(loc).upstream.detectors(dect).lane(lane).day ...
                (day).data.speed = speed_array;
            locations(loc).upstream.detectors(dect).lane(lane).day ...
                (day).data.flow = flow_array;
        end
    end
end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CONTAINERS REQUIREMENTS MERGE %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    locations(i).requirements = requirements;
    locations(i).path_mainline = [];
    locations(i).path_mainlink = [];
    locations(i).path_node = [];
    locations(i).path_mainlink = [];
    locations(i).upstream.path_mainlink = [];
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        locations(i).detectors(j).path_detector = [];
        for k = 1:locations(i).lanes_mainroad
            locations(i).detectors(j).lane(k).path_lanedetector = [];
        end
    end
end
try
    [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
    for j = 1:number_of_detectors
        locations(i).upstream.detectors(j).path_detector = [];
        for k = 1:locations(i).lanes_mainroad
            locations(i).upstream.detectors(j).lane(k).path_lanedetector ...
                = [];
        end
    end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVE %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
year = locations(1).dates(1).name;
year = year(1:4);
FileNameLoc = ['stap1_loc_' num2str(year) '_detectors_inv_ben_' ...
    num2str(requirements.speedlimit) '_' ...
    num2str(requirements.lanes_mainroad) '_' date '.mat'];
save(FileNameLoc, 'locations')

format shortG
t = toc;
disp(datestr(datenum(0,0,0,0,0,t), 'HH:MM:SS'))

```



```
clear all
```

### Required function scripts

Determine design length acceleration lanes.

```
function richtlijn_length = bepaalafstandinvoeging(requirements)
% Bepaal lengte invoeging
speedlimit = requirements.speedlimit;

% Lengte invoegstrook
if speedlimit == 130 || speedlimit == 120
    richtlijn_length = 250 + 100;
elseif speedlimit == 100
    richtlijn_length = 200 + 90;
elseif speedlimit == 80
    richtlijn_length = 155 + 80;
else
    error(['speedlimit is not correctly defined'])
    richtlijn_length = [];
end
```

Also the mentioned function scripts from the basic freeway segment are applicable.

## E.2.2. Step 2 - Merge April 2013 and 2014

### Main script

```
function nargin = samenvoegenDATA1314_v20140911(varargin)
% Upstream eruit gehaald
% 5 veh/km bins

number_of_bins = 30;
size_of_bin = 5;

%%%%%%%%%
% LOAD %
%%%%%%%%%
% Inladen files uit stap 2
[file, path, filter] = ...
    uigetfile('stap1_loc_2013*.mat', 'Please select a mat file (cancel for none)',cd);
dataFile = [path file];
load(dataFile)

locations2013 = locations;
clear locations file path filter

% Inladen files uit stap 2
[file, path, filter] = ...
    uigetfile('stap1_loc_2014*.mat', 'Please select a mat file (cancel for none)',cd);
dataFile = [path file];
load(dataFile)

locations2014 = locations;
clear locations file path filter

dates = dataFile(end-38:end-4);

tic

%%%%%%%%%
% MERGING %
%%%%%%%%%
% This is the easy approach, both (April 2013 and 2014) had the same
% locations, in the future, if there are differences adjustment of the code
% is needed.
[dim0,number_of_locations_2013] = size(locations2013);
[dim0,number_of_locations_2014] = size(locations2014);
```

```

if number_of_locations_2013 == number_of_locations_2014
    locations = locations2013;
    number_of_locations = number_of_locations_2013;
    for loc = 1:number_of_locations
        if locations2013(loc).toKM == ...
            locations2014(loc).toKM && locations2013(loc).fromKM == ...
                locations2014(loc).fromKM
            [dim0, number_of_detectors] = size(locations(loc).detectors);
            for dect = 1:number_of_detectors
                for lane = 1:locations(loc).lanes_mainroad
                    [dim0, number_of_days] = ...
                        size(locations(loc).detectors(dect).lane(lane).day);
                    [dim0, number_of_days2014] = ...
                        size(locations2014(loc).detectors(dect).lane(lane).day);
                    for days = 1:number_of_days2014
                        totalday = days + number_of_days;
                        locations(loc).detectors(dect).lane(lane).day ...
                            (totalday).data.speed = locations2014(loc).detectors ...
                                (dect).lane(lane).day(days).data.speed;
                        locations(loc).detectors(dect).lane(lane).day ...
                            (totalday).data.flow = locations2014(loc).detectors ...
                                (dect).lane(lane).day(days).data.flow;
                    end
                end
            end
        end
    end
    try
        [dim0, number_of_detectors] = ...
            size(locations(loc).upstream.detectors);
        for dect = 1:number_of_detectors
            for lane = 1:locations(loc).lanes_mainroad
                [dim0, number_of_days] = ...
                    size(locations(loc).upstream.detectors(dect).lane(lane).day);
                [dim0, number_of_days2014] = ...
                    size(locations2014(loc).upstream.detectors ...
                        (dect).lane(lane).day);
                for days = 1:number_of_days2014
                    totalday = days + number_of_days;
                    locations(loc).upstream.detectors ...
                        (dect).lane(lane).day(totalday).data.speed ...
                        = ...
                            locations2014(loc).upstream.detectors(dect).lane(lane).day(days).data.
                    locations(loc).upstream.detectors ...
                        (dect).lane(lane).day(totalday).data.flow ...
                        = locations2014(loc).upstream.detectors ...
                            (dect).lane(lane).day(days).data.flow;
                end
            end
        end
    end
end
end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CHECK ON EMPTY VECTORS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0, number_of_locations] = size(locations);
for loc = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(loc).detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = size(locations(loc).detectors(dect).lane(lane).day);
            for days = 1:number_of_days
                if isempty(locations(loc).detectors(dect).lane(lane).day(days).data.speed)
                    locations(loc).detectors(dect).lane(lane).day(days).data.speed ...
                        = ones(701,1) * NaN;
                end
                if isempty(locations(loc).detectors(dect).lane(lane).day(days).data.flow)
                    locations(loc).detectors(dect).lane(lane).day(days).data.flow ...
                        = ones(701,1) * NaN;
                end
            end
        end
    end
end

```

```

        end
    end
end
try
    [dim0, number_of_detectors] = size(locations(loc).upstream.detectors);
    for dect = 1:number_of_detectors
        for lane = 1:3
            [dim0, number_of_days] = ...
                size(locations(loc).upstream.detectors(dect).lane(lane).day);
            for days = 1:number_of_days
                if isempty(locations(loc).upstream.detectors(dect).lane ...
                    (lane).day(days).data.speed)
                    locations(loc).upstream.detectors(dect).lane(lane).day ...
                        (days).data.speed = ones(701,1) * NaN;
                end
                if isempty(locations(loc).upstream.detectors(dect).lane ...
                    (lane).day(days).data.flow)
                    locations(loc).upstream.detectors(dect).lane(lane).day ...
                        (days).data.flow = ones(701,1) * NaN;
                end
            end
        end
    end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DETERMINE DENSITIES %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INCORRECTE WAARDEN AANPASSEN
% Speeds: Incorrecte data NaN maken
% Speeds: Nulmetingen (-1) 0 maken
% Speeds: Flows is NaN => Speeds is NaN
% Flows: Speeds is NaN => Flows is NaN
% Flows : Incorrecte data NaN maken
% Speeds: Groter dan 220 NaN maken
% Speeds: Kleiner dan 80 NaN maken
[dim0, number_of_locations] = size(locations);
for loc = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(loc).detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = size(locations(loc).detectors(dect).lane(lane).day);
            for days = 1:number_of_days
                filter = locations(loc).detectors(dect).lane ...
                    (lane).day(days).data.speed == 0;
                locations(loc).detectors(dect).lane(lane).day(days).data.speed ...
                    (filter) = NaN;
                filter = [];
                filter = locations(loc).detectors(dect).lane(lane).day ...
                    (days).data.speed == -1;
                locations(loc).detectors(dect).lane(lane).day(days).data.speed ...
                    (filter) = 0;
                filter = [];
                filter = locations(loc).detectors(dect).lane(lane).day ...
                    (days).data.flow < 0;
                locations(loc).detectors(dect).lane(lane).day(days).data.flow ...
                    (filter) = NaN;
                filter = [];
                filter = locations(loc).detectors(dect).lane(lane).day ...
                    (days).data.speed > 220;
                locations(loc).detectors(dect).lane(lane).day(days).data.speed ...
                    (filter) = NaN;
                filter = [];
                filter = locations(loc).detectors(dect).lane(lane).day ...
                    (days).data.speed < 80;
                locations(loc).detectors(dect).lane(lane).day(days).data.speed ...
                    (filter) = NaN;
                filter = isnan(locations(loc).detectors(dect).lane(lane).day ...
                    (days).data.speed);
            end
        end
    end
end

```

```

        locations(loc).detectors(dect).lane(lane).day(days).data.flow ...
            (filter) = NaN;
        filter = [];
        filter = isnan(locations(loc).detectors(dect).lane(lane).day ...
            (days).data.flow);
        locations(loc).detectors(dect).lane(lane).day(days).data.speed ...
            (filter) = NaN;
        filter = [];
    end
end
end
try
    [dim0, number_of_detectors] = size(locations(loc).upstream.detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = ...
                size(locations(loc).upstream.detectors(dect).lane(lane).day);
            for days = 1:number_of_days
                filter = ...
                    locations(loc).upstream.detectors(dect).lane(lane).day ...
                        (days).data.speed == 0;
                locations(loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.speed(filter) = NaN;
                filter = [];
                filter = locations(loc).upstream.detectors(dect).lane ...
                    (lane).day(days).data.speed == -1;
                locations(loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.speed(filter) = 0;
                filter = [];
                filter = locations(loc).upstream.detectors(dect).lane ...
                    (lane).day(days).data.flow < 0;
                locations(loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.flow(filter) = NaN;
                filter = [];
                filter = locations(loc).upstream.detectors(dect).lane ...
                    (lane).day(days).data.speed > 220;
                locations(loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.speed(filter) = NaN;
                filter = [];
                filter = locations(loc).upstream.detectors(dect).lane ...
                    (lane).day(days).data.speed < 80;
                locations(loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.speed(filter) = NaN;
                filter = isnan(locations(loc).upstream.detectors(dect).lane ...
                    (lane).day(days).data.speed);
                locations(loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.flow(filter) = NaN;
                filter = [];
                filter = isnan(locations(loc).upstream.detectors(dect).lane ...
                    (lane).day(days).data.flow);
                locations(loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.speed(filter) = NaN;
                filter = [];
            end
        end
    end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DETERMINE DENSITIES %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine Densities
% Density = Flow / Speed
% Speed = Time Mean Speed!
[dim0, number_of_locations] = size(locations);
for loc = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(loc).detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = size(locations(loc).detectors ...

```

```

        (dect).lane(lane).day);
    for days = 1:number_of_days
        filter = locations(loc).detectors(dect).lane(lane).day...
            (days).data.speed == 0;
        locations(loc).detectors(dect).lane(lane).day...
            (days).data.density = locations(loc).detectors...
            (dect).lane(lane).day(days).data.flow ./ ...
            locations(loc).detectors(dect).lane(lane).day...
            (days).data.speed;
        locations(loc).detectors(dect).lane(lane).day...
            (days).data.density(filter) = 0;
    end
end
end
try
    [dim0, number_of_detectors] = size(locations(loc).upstream.detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = size(locations(loc).upstream.detectors...
                (dect).lane(lane).day);
            for days = 1:number_of_days
                filter = locations(loc).upstream.detectors...
                    (dect).lane(lane).day(days).data.speed == 0;
                locations(loc).upstream.detectors(dect).lane...
                    (lane).day(days).data.density = locations...
                    (loc).upstream.detectors(dect).lane(lane).day(days).data.flow ./ ...
                    locations(loc).upstream.detectors(dect).lane(lane).day...
                    (days).data.speed;
                locations(loc).upstream.detectors(dect).lane(lane).day...
                    (days).data.density(filter) = 0;
            end
        end
    end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MERGING DAYS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Combine days
% Make empty vectors to speed up
[dim0, number_of_locations] = size(locations);
for loc = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(loc).detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = size(locations(loc).detectors(dect).lane(lane).day);
            locations(loc).detectors(dect).lane(lane).data.flow = ...
                ones(number_of_days*701,1) * NaN;
            locations(loc).detectors(dect).lane(lane).data.speed = ...
                ones(number_of_days*701,1) * NaN;
            locations(loc).detectors(dect).lane(lane).data.density = ...
                ones(number_of_days*701,1) * NaN;
            locations(loc).detectors(dect).lane(lane).data.speed = ...
                ones(number_of_days*701,1) * NaN;
        end
    end
end
try
    [dim0, number_of_detectors] = size(locations(loc).upstream.detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = size(locations(loc).upstream.detectors...
                (dect).lane(lane).day);
            locations(loc).upstream.detectors(dect).lane(lane).data.flow...
                = ones(number_of_days*701,1) * NaN;
            locations(loc).upstream.detectors(dect).lane(lane).data.speed...
                = ones(number_of_days*701,1) * NaN;
            locations(loc).upstream.detectors(dect).lane(lane).data.density...
                = ones(number_of_days*701,1) * NaN;
            locations(loc).upstream.detectors(dect).lane(lane).data.speed...
                = ones(number_of_days*701,1) * NaN;
        end
    end
end

```

```

        end
    end
end
[dim0, number_of_locations] = size(locations);
for loc = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(loc).detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = size(locations(loc).detectors ...
                (dect).lane(lane).day);
            speed_array = [];
            flow_array = [];
            density_array = [];
            speed_array = [];
            for days = 1:number_of_days
                start_array = days * 701 + 1 - 701;
                end_array = days * 701;
                speed_array(start_array:end_array,:) = locations(loc).detectors ...
                    (dect).lane(lane).day(days).data.speed;
                flow_array(start_array:end_array,:) = locations(loc).detectors ...
                    (dect).lane(lane).day(days).data.flow;
                density_array(start_array:end_array,:) = locations(loc).detectors ...
                    (dect).lane(lane).day(days).data.density;
                speed_array(start_array:end_array,:) = locations(loc).detectors ...
                    (dect).lane(lane).day(days).data.speed;
            end
            locations(loc).detectors(dect).lane(lane).data.flow = flow_array;
            locations(loc).detectors(dect).lane(lane).data.speed = speed_array;
            locations(loc).detectors(dect).lane(lane).data.density = density_array;
            locations(loc).detectors(dect).lane(lane).data.speed = speed_array;
            speed_array = [];
            flow_array = [];
            density_array = [];
            speed_array = [];
        end
    end
end
try
    [dim0, number_of_detectors] = size(locations(loc).upstream.detectors);
    for dect = 1:number_of_detectors
        for lane = 1:locations(loc).lanes_mainroad
            [dim0, number_of_days] = size(locations(loc).upstream.detectors ...
                (dect).lane(lane).day);
            speed_array = [];
            flow_array = [];
            density_array = [];
            speed_array = [];
            for days = 1:number_of_days
                start_array = days * 701 + 1 - 701;
                end_array = days * 701;
                speed_array(start_array:end_array,:) = locations ...
                    (loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.speed;
                flow_array(start_array:end_array,:) = locations ...
                    (loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.flow;
                density_array(start_array:end_array,:) = locations ...
                    (loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.density;
                speed_array(start_array:end_array,:) = locations ...
                    (loc).upstream.detectors(dect).lane(lane).day ...
                    (days).data.speed;
            end
            locations(loc).upstream.detectors(dect).lane(lane).data.flow ...
                = flow_array;
            locations(loc).upstream.detectors(dect).lane(lane).data.speed ...
                = speed_array;
            locations(loc).upstream.detectors(dect).lane(lane).data.density ...
                = density_array;
            locations(loc).upstream.detectors(dect).lane(lane).data.speed ...
                = speed_array;
        end
    end
end

```

```

        speed_array = [];
        flow_array = [];
        density_array = [];
        speed_array = [];
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DETERMINE TOTALS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Totalen
for i = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        [number_of_minutes, dim0] = size(locations(i).detectors(j).lane(1).data.density);
        totaldensity = [];
        locations(i).detectors(j).data.totaldensity = [];
        locations(i).detectors(j).data.totalflow = [];
        density = [locations(i).detectors(j).lane(1).data.density ...
                    locations(i).detectors(j).lane(2).data.density ...
                    locations(i).detectors(j).lane(3).data.density];
        flow = [locations(i).detectors(j).lane(1).data.flow ...
                locations(i).detectors(j).lane(2).data.flow ...
                locations(i).detectors(j).lane(3).data.flow];
        locations(i).detectors(j).data.totalflow = sum(flow,2);
        locations(i).detectors(j).data.totaldensity = sum(density,2);
    end
    try
        [dim0, number_of_detectors] = size(locations(i).upstream.detectors);
        for j = 1:number_of_detectors
            [number_of_minutes, dim0] = ...
                size(locations(i).upstream.detectors(j).lane(1).data.density);
            totaldensity = [];
            locations(i).upstream.detectors(j).data.totaldensity = [];
            locations(i).upstream.detectors(j).data.totalflow = [];
            density = [locations(i).upstream.detectors(j).lane(1).data.density ...
                        locations(i).upstream.detectors(j).lane(2).data.density ...
                        locations(i).upstream.detectors(j).lane(3).data.density];
            flow = [locations(i).upstream.detectors(j).lane(1).data.flow ...
                    locations(i).upstream.detectors(j).lane(2).data.flow ...
                    locations(i).upstream.detectors(j).lane(3).data.flow];
            locations(i).upstream.detectors(j).data.totalflow = sum(flow,2);
            locations(i).upstream.detectors(j).data.totaldensity = ...
                sum(density,2);
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DETERMINE AVERAGE SPEEDS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:number_of_locations
    [dim0, number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        [number_of_minutes, dim0] = ...
            size(locations(i).detectors(j).lane(1).data.speed);
        totaldensity = [];
        locations(i).detectors(j).data.averagespeed = [];
        locations(i).detectors(j).data.averageweightedspeed = [];
        speed = [locations(i).detectors(j).lane(1).data.speed ...
                  locations(i).detectors(j).lane(2).data.speed ...
                  locations(i).detectors(j).lane(3).data.speed];
        flow = [locations(i).detectors(j).lane(1).data.flow ...
                locations(i).detectors(j).lane(2).data.flow ...
                locations(i).detectors(j).lane(3).data.flow];
        locations(i).detectors(j).data.averagespeed = sum(speed,2) ./ 3;
        locations(i).detectors(j).data.averageweightedspeed = sum(speed .* ...
            flow,2) ./ sum(flow,2);
    end
end

```

```

try
    [dim0, number_of_detectors] = size(locations(i).upstream.detectors);
    for j = 1:number_of_detectors
        [number_of_minutes, dim0] = ...
            size(locations(i).upstream.detectors(j).lane(1).data.speed);
        totaldensity = [];
        locations(i).upstream.detectors(j).data.averagespeed = [];
        locations(i).upstream.detectors(j).data.averageweightedspeed = [];
        speed = [locations(i).upstream.detectors(j).lane(1).data.speed ...
            locations(i).upstream.detectors(j).lane(2).data.speed ...
            locations(i).upstream.detectors(j).lane(3).data.speed];
        flow = [locations(i).upstream.detectors(j).lane(1).data.flow ...
            locations(i).upstream.detectors(j).lane(2).data.flow ...
            locations(i).upstream.detectors(j).lane(3).data.flow];
        locations(i).upstream.detectors(j).data.averagespeed = ...
            sum(speed,2) ./ 3;
        locations(i).upstream.detectors(j).data.averageweightedspeed = ...
            sum(speed .* flow,2) ./ sum(flow,2);
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine Bins %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Number of bins is 15
[dim0, number_of_locations] = size(locations);
for i = 1:number_of_locations
    for k = 1:number_of_bins
        [dim0 number_of_detectors] = size(locations(i).detectors);
        maxleng = 0;
        dect = [];
        for j = 1:number_of_detectors
            start = k * size_of_bin - size_of_bin;
            eind = k * size_of_bin;
            blok = locations(i).detectors(j).data.totaldensity;
            filter1 = blok > start & blok <= eind;
            filter2 = blok <= start | blok > eind;
            blok(filter1) = 1;
            blok(filter2) = NaN;
            locations(i).detectors(j).data.bin(k).blok = blok;
            locations(i).detectors(j).data.bin(k).start = start;
            locations(i).detectors(j).data.bin(k).end = eind;
        end
    end
    try
        [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
        maxleng = 0;
        dect = [];
        for j = 1:number_of_detectors
            start = k * size_of_bin - size_of_bin;
            eind = k * size_of_bin;
            blok = locations(i).upstream.detectors(j).data.totaldensity;
            filter1 = blok > start & blok <= eind;
            filter2 = blok <= start | blok > eind;
            blok(filter1) = 1;
            blok(filter2) = NaN;
            locations(i).upstream.detectors(j).data.bin(k).blok = blok;
            locations(i).upstream.detectors(j).data.bin(k).start = start;
            locations(i).upstream.detectors(j).data.bin(k).end = eind;
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine percentage flow %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors

```



```

        for k = 1:locations(i).lanes_mainroad
            locations(i).detectors(j).lane(k).data.percentageflow = ...
                locations(i).detectors(j).lane(k).data.flow ./ ...
                locations(i).detectors(j).data.totalflow;
        end
    end
end
try
    [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
    for j = 1:number_of_detectors
        for k = 1:locations(i).lanes_mainroad
            locations(i).upstream.detectors(j).lane(k).data.percentageflow ...
                = locations(i).upstream.detectors(j).lane(k).data.flow ./ ...
                locations(i).upstream.detectors(j).data.totalflow;
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NaN Filtering %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        for k = 1:locations(i).lanes_mainroad
            for l = 1:number_of_bins
                blokken = locations(i).detectors(j).data.bin(l).blok;
                typedata = locations(i).detectors(j).lane(k).data.flow;
                locations(i).detectors(j).lane(k).data.bin(l).flow = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).lane(k).data.speed;
                locations(i).detectors(j).lane(k).data.bin(l).speed = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).lane(k).data.density;
                locations(i).detectors(j).lane(k).data.bin(l).density = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).lane(k).data.speed;
                locations(i).detectors(j).lane(k).data.bin(l).speed = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).lane(k).data.percentageflow;
                locations(i).detectors(j).lane(k).data.bin(l).percentageflow ...
                    = dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).data.totalflow;
                locations(i).detectors(j).data.bin(l).totalflow = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).data.totaldensity;
                locations(i).detectors(j).data.bin(l).totaldensity = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).data.averagespeed;
                locations(i).detectors(j).data.bin(l).averagespeed = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).detectors(j).data.averageweighted speed;
                locations(i).detectors(j).data.bin(l).averageweighted speed = ...
                    dataBlokkenVerwerking(typedata, blokken);
            end
        end
    end
end
try
    [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
    for j = 1:number_of_detectors
        for k = 1:locations(i).lanes_mainroad
            for l = 1:number_of_bins
                blokken = locations(i).upstream.detectors(j).data.bin(l).blok;
                typedata = locations(i).upstream.detectors(j).lane(k).data.flow;
                locations(i).upstream.detectors(j).lane(k).data.bin(l).flow = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).upstream.detectors(j).lane(k).data.speed;
                locations(i).upstream.detectors(j).lane(k).data.bin(l).speed = ...
                    dataBlokkenVerwerking(typedata, blokken);
                typedata = locations(i).upstream.detectors(j).lane(k).data.density;

```

```

        locations(i).upstream.detectors(j).lane(k).data.bin(1).density = ...
            dataBlokkenVerwerking(typedata, blokken);
        locations(i).upstream.detectors(j).lane(k).data.speed = ...
            dataBlokkenVerwerking(typedata, blokken);
        typedata = ...
            locations(i).upstream.detectors(j).lane(k).data.percentageflow;
        locations(i).upstream.detectors(j).lane(k).data.bin(1).percentageflow ...
            = ...
            dataBlokkenVerwerking(typedata, blokken);
        typedata = locations(i).upstream.detectors(j).data.totalflow;
        locations(i).upstream.detectors(j).data.bin(1).totalflow = ...
            dataBlokkenVerwerking(typedata, blokken);
        typedata = locations(i).upstream.detectors(j).data.totaldensity;
        locations(i).upstream.detectors(j).data.bin(1).totaldensity = ...
            dataBlokkenVerwerking(typedata, blokken);
        typedata = locations(i).upstream.detectors(j).data.averagespeed;
        locations(i).upstream.detectors(j).data.bin(1).averagespeed = ...
            dataBlokkenVerwerking(typedata, blokken);
        typedata = locations(i).upstream.detectors(j).data.averageweighted speed;
        locations(i).upstream.detectors(j).data.bin(1).averageweighted speed = ...
            dataBlokkenVerwerking(typedata, blokken);
    end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Empty containers fill with NaN %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        for k = 1:locations(i).lanes_mainroad
            for l = 1:number_of_bins
                if isempty(locations(i).detectors(j).lane(k).data.bin(1).flow);
                    locations(i).detectors(j).lane(k).data.bin(1).flow = ...
                        ones(2,1) * NaN;
                end
                if isempty(locations(i).detectors(j).lane(k).data.bin(1).speed);
                    locations(i).detectors(j).lane(k).data.bin(1).speed = ...
                        ones(2,1) * NaN;
                end
                if isempty(locations(i).detectors(j).lane(k).data.bin(1).density);
                    locations(i).detectors(j).lane(k).data.bin(1).density = ...
                        ones(2,1) * NaN;
                end
                if isempty(locations(i).detectors(j).lane(k).data.bin(1).percentageflow);
                    locations(i).detectors(j).lane(k).data.bin(1).percentageflow = ...
                        ones(2,1) * NaN;
                end
                if isempty(locations(i).detectors(j).data.bin(1).totalflow);
                    locations(i).detectors(j).data.bin(1).totalflow ...
                        = ones(2,1) * NaN;
                end
                if isempty(locations(i).detectors(j).data.bin(1).totaldensity);
                    locations(i).detectors(j).data.bin(1).totaldensity ...
                        = ones(2,1) * NaN;
                end
                if isempty(locations(i).detectors(j).data.bin(1).averagespeed);
                    locations(i).detectors(j).data.bin(1).averagespeed ...
                        = ones(2,1) * NaN;
                end
                if isempty(locations(i).detectors(j).data.bin(1).averageweighted speed);
                    locations(i).detectors(j).data.bin(1).averageweighted speed ...

```

```

        = ones(2,1) * NaN;
    end
end
end
try
[dim0 number_of_detectors] = size(locations(i).upstream.detectors);
for j = 1:number_of_detectors
    for k = 1:locations(i).lanes_mainroad
        for l = 1:number_of_bins
            if isempty(locations(i).upstream.detectors(j).lane(k).data.bin ...
                (l).flow);
                locations(i).upstream.detectors(j).lane(k).data.bin(l).flow ...
                    = ones(2,1) * NaN;
            end
            if isempty(locations(i).upstream.detectors(j).lane(k).data.bin ...
                (l).speed);
                locations(i).upstream.detectors(j).lane(k).data.bin(l).speed ...
                    = ones(2,1) * NaN;
            end
            if isempty(locations(i).upstream.detectors(j).lane(k).data.bin ...
                (l).density);
                locations(i).upstream.detectors(j).lane(k).data.bin ...
                    (l).density = ones(2,1) * NaN;
            end
            if isempty(locations(i).upstream.detectors(j).lane ...
                (k).data.bin(l).speed);
                locations(i).upstream.detectors(j).lane(k).data.bin ...
                    (l).speed = ones(2,1) * NaN;
            end
            if isempty(locations(i).upstream.detectors(j).lane ...
                (k).data.bin(l).percentageflow);
                locations(i).upstream.detectors(j).lane(k).data.bin ...
                    (l).percentageflow = ones(2,1) * NaN;
            end
            if isempty(locations(i).upstream.detectors(j).data.bin(l).totalflow);
                locations(i).upstream.detectors(j).data.bin(l).totalflow ...
                    = ones(2,1) * NaN;
            end
            if isempty(locations(i).upstream.detectors(j).data.bin ...
                (l).totaldensity);
                locations(i).upstream.detectors(j).data.bin(l).totaldensity ...
                    = ones(2,1) * NaN;
            end
            if isempty(locations(i).upstream.detectors(j).data.bin ...
                (l).averagespeed);
                locations(i).upstream.detectors(j).data.bin ...
                    (l).averagespeed = ones(2,1) * NaN;
            end
            if isempty(locations(i).upstream.detectors(j).data.bin ...
                (l).averageweightedspeed);
                locations(i).upstream.detectors(j).data.bin ...
                    (l).averagewightedspeed = ones(2,1) * NaN;
            end
        end
    end
end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DETERMINING MEDIAN AND STD %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Add color
[dim0 number_of_locations] = size(locations);
cmap = distinguishable_colors(number_of_locations);
for i = 1:number_of_locations
    locations(i).color = cmap(i,:);
end
% medians berekenen
[dim0 number_of_locations] = size(locations);

```

```

for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        for k = 1:locations(i).lanes_mainroad
            for l = 1:number_of_bins
                % Medians
                locations(i).detectors(j).data.bin(l).median.totalflow = ...
                    median(locations(i).detectors(j).data.bin(l).totalflow);
                locations(i).detectors(j).data.bin(l).median.totaldensity = ...
                    median(locations(i).detectors(j).data.bin(l).totaldensity);
                locations(i).detectors(j).lane(k).data.bin(l).median.percentageflow ...
                    = median(locations(i).detectors(j).lane(k).data.bin(l).percentageflow);
                locations(i).detectors(j).lane(k).data.median.percentageflow(1,1) ...
                    = locations(i).detectors(j).lane(k).data.bin(l).median.percentageflow;
                locations(i).detectors(j).lane(k).data.bin(l).median.flow ...
                    = median(locations(i).detectors(j).lane(k).data.bin(l).flow);
                locations(i).detectors(j).lane(k).data.median.flow(1,1) ...
                    = locations(i).detectors(j).lane(k).data.bin(l).median.flow;
                locations(i).detectors(j).lane(k).data.bin(l).median.speed ...
                    = median(locations(i).detectors(j).lane(k).data.bin(l).speed);
                locations(i).detectors(j).lane(k).data.median.speed(1,1) ...
                    = locations(i).detectors(j).lane(k).data.bin(l).median.speed;
                locations(i).detectors(j).lane(k).data.bin(l).median.density ...
                    = median(locations(i).detectors(j).lane(k).data.bin(l).density);
                locations(i).detectors(j).lane(k).data.median.density(1,1) ...
                    = locations(i).detectors(j).lane(k).data.bin(l).median.density;
                locations(i).detectors(j).lane(k).data.median.afstand(1,1) ...
                    = locations(i).detectors(j).position_upstream;
                locations(i).detectors(j).lane(k).data.median.bin(1,1) = ...
                    l * size_of_bin - (size_of_bin / 2);
                locations(i).detectors(j).data.bin(l).median.averagespeed ...
                    = median(locations(i).detectors(j).data.bin(l).averagespeed);
                locations(i).detectors(j).data.bin(l).median.averageweightspeed ...
                    = median(locations(i).detectors(j).data.bin(l).averageweightspeed);
                %Std
                locations(i).detectors(j).data.bin(l).std.totalflow ...
                    = std(locations(i).detectors(j).data.bin(l).totalflow);
                locations(i).detectors(j).data.bin(l).std.totaldensity ...
                    = std(locations(i).detectors(j).data.bin(l).totaldensity);
                locations(i).detectors(j).lane(k).data.bin(l).std.percentageflow ...
                    = std(locations(i).detectors(j).lane(k).data.bin(l).percentageflow);
                locations(i).detectors(j).lane(k).data.std.percentageflow(1,1) ...
                    = locations(i).detectors(j).lane(k).data.bin(l).std.percentageflow;
                locations(i).detectors(j).lane(k).data.bin(l).std.flow ...
                    = std(locations(i).detectors(j).lane(k).data.bin(l).flow);
                locations(i).detectors(j).lane(k).data.std.flow(1,1) ...
                    = locations(i).detectors(j).lane(k).data.bin(l).std.flow;
                locations(i).detectors(j).lane(k).data.bin(l).std.speed ...
                    = std(locations(i).detectors(j).lane(k).data.bin(l).speed);
                locations(i).detectors(j).lane(k).data.std.speed(1,1) ...
                    = locations(i).detectors(j).lane(k).data.bin(l).std.speed;
                locations(i).detectors(j).lane(k).data.bin(l).std.density ...
                    = std(locations(i).detectors(j).lane(k).data.bin(l).density);
                locations(i).detectors(j).lane(k).data.std.density(1,1) ...
                    = locations(i).detectors(j).lane(k).data.bin(l).std.density;
                locations(i).detectors(j).lane(k).data.std.afstand(1,1) ...
                    = locations(i).detectors(j).position_upstream;
                locations(i).detectors(j).lane(k).data.std.bin(1,1) ...
                    = l * size_of_bin - (size_of_bin / 2);
                locations(i).detectors(j).data.bin(l).std.averageweightspeed ...
                    = std(locations(i).detectors(j).data.bin(l).averageweightspeed);
            end
        end
    end
end
try
    [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
    for j = 1:number_of_detectors
        for k = 1:locations(i).lanes_mainroad
            for l = 1:number_of_bins
                % Medians

```

```

locations(i).upstream.detectors(j).data.bin ...
    (l).median.totalflow ...
    = median(locations(i).upstream.detectors ...
    (j).data.bin(l).totalflow);
locations(i).upstream.detectors(j).data.bin ...
    (l).median.totaldensity ...
    = median(locations(i).upstream.detectors ...
    (j).data.bin(l).totaldensity);
locations(i).upstream.detectors(j).lane(k).data.bin ...
    (l).median.percentagelflow = median(locations ...
    (i).upstream.detectors(j).lane(k).data.bin(l).percentagelflow);
locations(i).upstream.detectors(j).lane ...
    (k).data.median.percentagelflow(l,1) = ...
    locations(i).upstream.detectors(j).lane(k).data.bin ...
    (l).median.percentagelflow;
locations(i).upstream.detectors(j).lane(k).data.bin ...
    (l).median.flow = median(locations(i).upstream.detectors ...
    (j).lane(k).data.bin(l).flow);
locations(i).upstream.detectors(j).lane(k).data.median.flow(l,1) ...
    = locations(i).upstream.detectors(j).lane(k).data.bin ...
    (l).median.flow;
locations(i).upstream.detectors(j).lane(k).data.bin ...
    (l).median.speed = median(locations(i).upstream.detectors ...
    (j).lane(k).data.bin(l).speed);
locations(i).upstream.detectors(j).lane(k).data.median.speed(l,1) ...
    = locations(i).upstream.detectors(j).lane(k).data.bin ...
    (l).median.speed;
locations(i).upstream.detectors(j).lane(k).data.bin ...
    (l).median.density = median(locations(i).upstream.detectors ...
    (j).lane(k).data.bin(l).density);
locations(i).upstream.detectors(j).lane ...
    (k).data.median.density(l,1) = ...
    locations(i).upstream.detectors(j).lane(k).data.bin ...
    (l).median.density;
locations(i).upstream.detectors(j).lane ...
    (k).data.median.afstand(l,1) = ...
    locations(i).upstream.detectors(j).position_upstream;
locations(i).upstream.detectors ...
    (j).lane(k).data.median.bin(l,1) = ...
    l * size_of_bin - (size_of_bin / 2);
locations(i).upstream.detectors ...
    (j).data.bin(l).median.averagespeed ...
    = median(locations(i).upstream.detectors(j).data.bin ...
    (l).averagespeed);
locations(i).upstream.detectors(j).data.bin ...
    (l).median.averageweightedspeed = ...
    median(locations(i).upstream.detectors(j).data.bin ...
    (l).averageweightedspeed);
%Std
locations(i).upstream.detectors(j).data.bin ...
    (l).std.totalflow = std(locations(i).upstream.detectors ...
    (j).data.bin(l).totalflow);
locations(i).upstream.detectors(j).data.bin ...
    (l).std.totaldensity = std(locations(i).upstream.detectors ...
    (j).data.bin(l).totaldensity);
locations(i).upstream.detectors(j).lane ...
    (k).data.bin(l).std.percentagelflow = ...
    std(locations(i).upstream.detectors(j).lane ...
    (k).data.bin(l).percentagelflow);
locations(i).upstream.detectors(j).lane ...
    (k).data.std.percentagelflow(l,1) = ...
    locations(i).upstream.detectors(j).lane ...
    (k).data.bin(l).std.percentagelflow;
locations(i).upstream.detectors(j).lane ...
    (k).data.bin(l).std.flow = std(locations ...
    (i).upstream.detectors(j).lane(k).data.bin(l).flow);
locations(i).upstream.detectors(j).lane ...
    (k).data.std.flow(l,1) = locations(i).upstream.detectors ...
    (j).lane(k).data.bin(l).std.flow;
locations(i).upstream.detectors(j).lane ...
    (k).data.bin(l).std.speed = std(locations ...

```

```

        (i).upstream.detectors(j).lane(k).data.bin(1).speed);
    locations(i).upstream.detectors(j).lane ...
        (k).data.std.speed(1,1) = locations(i).upstream.detectors ...
        (j).lane(k).data.bin(1).std.speed;
    locations(i).upstream.detectors(j).lane(k).data.bin ...
        (1).std.density = std(locations(i).upstream.detectors(j).lane ...
        (k).data.bin(1).density);
    locations(i).upstream.detectors(j).lane(k).data.std.density(1,1) ...
        = locations(i).upstream.detectors(j).lane(k).data.bin ...
        (1).std.density;
    locations(i).upstream.detectors(j).lane(k).data.std.afstand(1,1) = ...
        locations(i).upstream.detectors(j).position_upstream;
    locations(i).upstream.detectors(j).lane(k).data.std.bin(1,1) = ...
        1 * size_of_bin - (size_of_bin / 2);
    locations(i).upstream.detectors(j).data.bin ...
        (1).std.averageweightedspeed = std(locations(i).upstream.detectors ...
        (j).data.bin(1).averageweightedspeed);
    end
end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% UNFILTERED PLOTS TO DETERMINE WANTED FILTERS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot2Dpercentageflow_unfiltered(locations)
plot2Ddensity_unfiltered(locations)
plot2Dspeed_unfiltered(locations)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% APPLYING FILTERS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Filtering error-ed results
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    if ~isempty(locations(i).detectors)
        [dim0 number_of_detectors] = size(locations(i).detectors);
        for j = linspace(number_of_detectors,1,number_of_detectors)
            if locations(i).detectors(j).lane(2).data.median.percentageflow(12,1) ...
                < 0.3
                locations(i).detectors(j) = [];
            elseif max(locations(i).detectors(j).lane(2).data.median.speed) ...
                > 120
                locations(i).detectors(j) = [];
            elseif max(locations(i).detectors(j).lane(3).data.median.speed) ...
                > 120
                locations(i).detectors(j) = [];
            elseif locations(i).detectors(j).lane(2).data.median.percentageflow(3,1) ...
                < 0.2
                locations(i).detectors(j) = [];
            elseif locations(i).detectors(j).lane(1).data.median.density(7,1) ...
                < 3.2
                locations(i).detectors(j) = [];
            end
        end
    end
end
try
    [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
    for j = linspace(number_of_detectors,1,number_of_detectors)
        if locations(i).upstream.detectors(j).lane ...
            (2).data.median.percentageflow(12,1) < 0.3
            locations(i).upstream.detectors(j) = [];
        elseif max(locations(i).upstream.detectors ...
            (j).lane(2).data.median.speed) > 120
            locations(i).upstream.detectors(j) = [];
        elseif max(locations(i).upstream.detectors ...
            (j).lane(3).data.median.speed) > 120
            locations(i).upstream.detectors(j) = [];
        elseif locations(i).upstream.detectors(j).lane ...
            (2).data.median.percentageflow(3,1) < 0.2
            locations(i).upstream.detectors(j) = [];
        end
    end
end

```

```

                elseif locations(i).upstream.detectors(j).lane...
                    (1).data.median.density(7,1) < 3.2
                        locations(i).upstream.detectors(j) = [];
                end
            end
        end
    end
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FILTERED PLOTS TO CONTROL FILTERS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot2Dpercentageflow_filtered(locations)
plot2Ddensity_filtered(locations)
plot2DSpeed_filtered(locations)

%%%%%%%%%%
% SAVE %
%%%%%%%%%%

FileNameloc = ['step2_loc_' dates '.mat'];
save(FileNameloc, 'locations', '-v7.3')

format shortG
t = toc;
disp(datestr(datenum(0,0,0,0,0,t), 'HH:MM:SS'))
clear all

```

### Required function scripts

Most scripts of the basic freeway segment are used here, only adjusted due to the presence of two links per location. Therefore here an example.

```

function nargin = plot2Dpercentageflow_filtered(locations)
% Upstream eruit halen

[dim0 number_of_locations] = size(locations);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Outside lane %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('visible','off')
xlabel('total density (veh/km)')
ylabel('fraction of flow (%)')
hold on
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        X = locations(i).detectors(j).lane(3).data.median.afstand;
        Y = locations(i).detectors(j).lane(3).data.median.bin;
        Z = locations(i).detectors(j).lane(3).data.median.percentageflow;
        color = locations(i).color;
        plot(Y,Z, 'Color', color)
    end
    try
        [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
        for j = 1:number_of_detectors
            X = locations(i).upstream.detectors(j).lane(3).data.median.afstand;
            Y = locations(i).upstream.detectors(j).lane(3).data.median.bin;
            Z = locations(i).upstream.detectors(j).lane(3).data.median.percentageflow;
            color = locations(i).color;
            plot(Y,Z, 'Color', color)
        end
    end
end
hold off
ylim([0 1])
xlim([0 100])
drm = ['filtered\2DPLOTS\'];
fname = ['fraction of flow - outside lane'];

```

```

exportfig(fname,drm)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Center lane %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('visible','off')
xlabel('total density (veh/km)')
ylabel('fraction of flow (%)')
hold on
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        X = locations(i).detectors(j).lane(2).data.median.afstand;
        Y = locations(i).detectors(j).lane(2).data.median.bin;
        Z = locations(i).detectors(j).lane(2).data.median.percentageflow;
        color = locations(i).color;
        plot(Y,Z,'Color',color)
    end
    try
        [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
        for j = 1:number_of_detectors
            X = locations(i).upstream.detectors(j).lane(2).data.median.afstand;
            Y = locations(i).upstream.detectors(j).lane(2).data.median.bin;
            Z = locations(i).upstream.detectors(j).lane(2).data.median.percentageflow;
            color = locations(i).color;
            plot(Y,Z,'Color',color)
        end
    end
end
hold off
ylim([0 1])
xlim([0 100])
drm = ['filtered\2DPLOTS\'];
fname = ['fraction of flow - center lane'];
exportfig(fname,drm)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Median Lane %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('visible','off')
xlabel('total density (veh/km)')
ylabel('fraction of flow (%)')
hold on
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    for j = 1:number_of_detectors
        X = locations(i).detectors(j).lane(1).data.median.afstand;
        Y = locations(i).detectors(j).lane(1).data.median.bin;
        Z = locations(i).detectors(j).lane(1).data.median.percentageflow;
        color = locations(i).color;
        plot(Y,Z,'Color',color)
    end
    try
        [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
        for j = 1:number_of_detectors
            X = locations(i).upstream.detectors(j).lane(1).data.median.afstand;
            Y = locations(i).upstream.detectors(j).lane(1).data.median.bin;
            Z = locations(i).upstream.detectors(j).lane(1).data.median.percentageflow;
            color = locations(i).color;
            plot(Y,Z,'Color',color)
        end
    end
end
hold off
ylim([0 1])
xlim([0 100])
drm = ['filtered\2DPLOTS\'];
fname = ['fraction of flow - median lane'];
exportfig(fname,drm)

```



**E.2.3. Step 3 - Make plots and calculations for analyses**

## Main script

```

function locations = DATAVERWERKING_v20140822(bin)
% Upstream delete
% Boxplot delete

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LOAD %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Inladen files uit stap 2
[file , path, filter] = ...
    uigetfile('stap2_loc*.mat', 'Please select a mat file (cancel for none)',cd);
dataFile = [path file];
load(dataFile)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% EXPORT LIST %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
id = 1;
for i = 1:number_of_locations
    if ~isempty(locations(i).detectors)
        LIJST{id,1} = locations(i).roadnumber;
        LIJST{id,2} = locations(i).fromKM;
        LIJST{id,3} = locations(i).toKM;
        LIJST{id,4} = locations(i).location_number;
        id = id + 1;
    end
end
LIJST = sortrows(LIJST,2);
LIJST = sortrows(LIJST);
HEADERS{1,1} = ['wegnr'];
HEADERS{1,2} = ['vanKM'];
HEADERS{1,3} = ['naarKM'];
HEADERS{1,4} = ['location_number'];
LIJST = [HEADERS; LIJST];
delete('locaties.xls');
xlswrite('locaties.xls',LIJST);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ADDING FOUND CHARACTERISTICS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[ndata, text, exceldata] = xlsread('locaties_traject');
[dim0 number_of_locations] = size(locations);
[r c] = size(exceldata);
for i = 1:number_of_locations
    location_number = locations(i).location_number;
    for j = 1:r
        if location_number == exceldata{j,4}
            locations(i).characteristics_traject = exceldata{j,5};
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% EXPORT LIST TRAJECT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
LIJST = [];
id = 1;
for i = 1:number_of_locations
    if ~isempty(locations(i).detectors)
        LIJST{id,1} = locations(i).roadnumber;
        LIJST{id,2} = ['A' num2str(locations(i).roadnumber)];
        LIJST{id,3} = locations(i).fromKM;
        LIJST{id,4} = locations(i).toKM;
        [dim0 number_of_detectors] = size(locations(i).detectors);
    end
end

```

```

        LIJST{id,5} = number_of_detectors;
    try
        [dim0 number_of_detectors] = ...
            size(locations(i).upstream.detectors);
        LIJST{id,6} = number_of_detectors;
    end
    LIJST{id,7} = locations(i).characteristics.traject;
    id = id + 1;
end
end
LIJST = sortrows(LIJST,3);
LIJST = sortrows(LIJST,1);
LIJST(:,1) = [];
HEADERS{1,1} = ['roadnumber'];
HEADERS{1,2} = ['from KM'];
HEADERS{1,3} = ['to KM'];
HEADERS{1,4} = ['#downstream detectors'];
HEADERS{1,5} = ['#upstream detectors'];
HEADERS{1,6} = ['fixed average speed check'];
LIJST = [HEADERS; LIJST];
delete('locaties_total.xls');
xlswrite('locaties_total.xls',LIJST);
HEADERS = [];
LIJST = [];

% Delete traject and uncertain locations
[dim0 number_of_locations] = size(locations);
for i = linspace(number_of_locations,1,number_of_locations)
    if ~strcmp(locations(i).characteristics.traject,'no')
        locations(i) = [];
    elseif isempty(locations(i).detectors)
        locations(i) = [];
    elseif locations(i).location_number == 729
        locations(i) = [];
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% EXPORT LIST NO TRAJECT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
LIJST = [];
id = 1;
for i = 1:number_of_locations
    if ~isempty(locations(i).detectors)
        LIJST{id,1} = locations(i).roadnumber;
        LIJST{id,2} = ['A' num2str(locations(i).roadnumber)];
        LIJST{id,3} = locations(i).fromKM;
        LIJST{id,4} = locations(i).toKM;
        [dim0 number_of_detectors] = size(locations(i).detectors);
        LIJST{id,5} = number_of_detectors;
        id = id + 1;
    end
end
LIJST = sortrows(LIJST,3);
LIJST = sortrows(LIJST,1);
LIJST(:,1) = [];
HEADERS{1,1} = ['roadnumber'];
HEADERS{1,2} = ['from KM'];
HEADERS{1,3} = ['to KM'];
HEADERS{1,4} = ['#detectors'];
LIJST = [HEADERS; LIJST];
delete('locaties_notraject.xls');
xlswrite('locaties_notraject.xls',LIJST);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LINEPLOTS REPORT MEDIANS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for lane = 1:3
    scatter_percentage_flow_median(locations, lane, bin)
    scatter_speed_median(locations, lane, bin)
end

```

```

        scatter_density_median(locations, lane, bin)
    end

    %%%%%%%%%%%%%%
    % LINEPLOTS %
    %%%%%%%%%%%%%%
    for lane = 1:3
        lineplot_percentage_flow_median(locations, lane, bin)
    end

    %%%%%%%%%%
    % LOAD %
    %%%%%%%%%%
    locations_ACCL = locations;
    clear locations
    % Inladen files uit stap 2
    [file, path, filter] = ...
        uigetfile('stap3_BFS*.mat', ...
            'Please select a mat file (cancel for none)', cd);
    dataFile = [path file];
    load(dataFile)
    locations_BFS = locations;
    clear locations
    locations = locations_ACCL;
    clear locations_ACCL

    %%%%%%%%%%%%%%
    % NaN filtering %
    %%%%%%%%%%%%%%
    [dim0 number_of_locations] = size(locations);
    for i = 1:number_of_locations
        [dim0 number_of_detectors] = size(locations(i).detectors);
        for j = 1:number_of_detectors
            for lane = 1:3
                locations(i).detectors(j).lane(lane).data.bin ...
                    (bin).flow(isnan(locations(i).detectors(j).lane ...
                        (lane).data.bin(bin).flow))) = [];
                locations(i).detectors(j).lane(lane).data.bin ...
                    (bin).speed(isnan(locations(i).detectors(j).lane ...
                        (lane).data.bin(bin).speed))) = [];
                locations(i).detectors(j).lane(lane).data.bin(bin).density ...
                    (isnan(locations(i).detectors(j).lane(lane).data.bin ...
                        (bin).density))) = [];
                locations(i).detectors(j).lane(lane).data.bin ...
                    (bin).percentageflow(isnan(locations(i).detectors(j).lane ...
                        (lane).data.bin(bin).percentageflow))) = [];
            end
            locations(i).detectors(j).data.bin(bin).totalflow ...
                (isnan(locations(i).detectors(j).data.bin(bin).totalflow)) = [];
            locations(i).detectors(j).data.bin(bin).averageweightedspeed ...
                (isnan(locations(i).detectors(j).data.bin(bin).averageweightedspeed)) = [];
        end
    end
    try
        [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
        for j = 1:number_of_detectors
            for lane = 1:3
                locations(i).upstream.detectors(j).lane(lane).data.bin ...
                    (bin).flow(isnan(locations(i).upstream.detectors ...
                        (j).lane(lane).data.bin(bin).flow))) = [];
                locations(i).upstream.detectors(j).lane(lane).data.bin ...
                    (bin).speed(isnan(locations(i).upstream.detectors ...
                        (j).lane(lane).data.bin(bin).speed))) = [];
                locations(i).upstream.detectors(j).lane(lane).data.bin ...
                    (bin).density(isnan(locations(i).upstream.detectors ...
                        (j).lane(lane).data.bin(bin).density))) = [];
                locations(i).upstream.detectors(j).lane(lane).data.bin ...
                    (bin).percentageflow(isnan(locations ...
                        (i).upstream.detectors(j).lane(lane).data.bin ...
                        (bin).percentageflow))) = [];
            end
            locations(i).upstream.detectors(j).data.bin(bin).totalflow ...

```

```

        (isnan(locations(i).upstream.detectors(j).data.bin ...
        (bin).totalflow)) = [];
        locations(i).upstream.detectors(j).data.bin ...
        (bin).averageweightedspeed(isnan(locations ...
        (i).upstream.detectors(j).data.bin ...
        (bin).averageweightedspeed)) = [];
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OPSPLITSEN DATA OVER AFSTAND %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations(i).detectors);
    a = 1;
    b = 1;
    c = 1;
    for j = 1:number_of_detectors
        if locations(i).detectors(j).position_upstream < 300
            locations(i).distance(1).detectors(a) = locations(i).detectors(j);
            a = a + 1;
        elseif locations(i).detectors(j).position_upstream > 300 && ...
            locations(i).detectors(j).position_upstream < 600
            locations(i).distance(2).detectors(b) = locations(i).detectors(j);
            b = b + 1;
        elseif locations(i).detectors(j).position_upstream > 600 && ...
            locations(i).detectors(j).position_upstream < 900
            locations(i).distance(3).detectors(c) = locations(i).detectors(j);
            c = c + 1;
        end
    end
    if a == 1;
        locations(i).distance(1).empty = 'yes';
    else
        locations(i).distance(1).empty = 'no';
    end
    if b == 1;
        locations(i).distance(2).empty = 'yes';
    else
        locations(i).distance(2).empty = 'no';
    end
    if c == 1;
        locations(i).distance(3).empty = 'yes';
    else
        locations(i).distance(3).empty = 'no';
    end
end
% Upstream
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    try
        [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
        a = 1;
        b = 1;
        for j = 1:number_of_detectors
            if locations(i).upstream.detectors(j).position_downstream < 300
                locations(i).upstream.distance(1).detectors(a) = ...
                locations(i).upstream.detectors(j);
                a = a + 1;
            elseif locations(i).upstream.detectors(j).position_downstream ...
                > 300 && ...
                locations(i).upstream.detectors(j).position_downstream ...
                < 600
                locations(i).upstream.distance(2).detectors(b) = ...
                locations(i).upstream.detectors(j);
                b = b + 1;
            end
        end
        if a == 1;

```

```

        locations(i).upstream.distance(1).empty = 'yes';
    else
        locations(i).upstream.distance(1).empty = 'no';
    end
    if b == 1;
        locations(i).upstream.distance(2).empty = 'yes';
    else
        locations(i).upstream.distance(2).empty = 'no';
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TABLE maken met #detectors interval %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
list_interval(locations,300)
list_interval(locations,250)
list_interval(locations,200)
list_interval(locations,150)
list_interval(locations,100)
list_interval(locations,50)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Histogrammen %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for distancebin = 1:3
    for lane = 1:3
        BFS_histo_density(locations ,distancebin ,lane ,bin)
        BFS_histo_percentage(flow(locations ,distancebin ,lane ,bin)
        BFS_histo_speed(locations ,distancebin ,lane ,bin)
    end
    BFS_histo_totalflow(locations ,distancebin ,bin)
    BFS_histo_averageweightspeed(locations ,distancebin ,bin)
end
for distancebin = 1:2
    for lane = 1:3
        BFS_histo_density_up(locations ,distancebin ,lane ,bin)
        BFS_histo_percentage_flow_up(locations ,distancebin ,lane ,bin)
        BFS_histo_speed_up(locations ,distancebin ,lane ,bin)
    end
    BFS_histo_totalflow_up(locations ,distancebin ,bin)
    BFS_histo_averageweightspeed_up(locations ,distancebin ,bin)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot locations detectors maken %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
combinating_locations_acceleration_lane(locations)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Distance plots %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for lane = 1:3
    MLE_distance_density(locations ,locations_BFS ,lane ,bin)
    MLE_distance_density_std(locations ,locations_BFS ,lane ,bin)

    MLE_distance_percentage_flow(locations ,locations_BFS ,lane ,bin)
    MLE_distance_percentage_flow_std(locations ,locations_BFS ,lane ,bin)

    MLE_distance_speed(locations ,locations_BFS ,lane ,bin)
    MLE_distance_speed_std(locations ,locations_BFS ,lane ,bin)
end
MLE_distance_averageweightspeed(locations ,locations_BFS ,bin)
MLE_distance_averageweightspeed_std(locations ,locations_BFS ,bin)

MLE_distance_totalflow(locations ,locations_BFS ,bin)
MLE_distance_totalflow_std(locations ,locations_BFS ,bin)

fitting_percentage_flow(locations ,locations_BFS ,bin)
fitting_density(locations ,locations_BFS ,bin)
fitting_speed(locations ,locations_BFS ,bin)

```

### Required function scripts

Interval length selection. This script shows the number of detectors in the intervals depending on the interval length.

```
% Version 2014-10-21 by M.J. Hovenga
function nargin = list_interval(locations, stepsize)

    [dim0 number_of_locations] = size(locations);
    AFSTANDEN = [];
    for i = 1:number_of_locations
        try
            [dim0 number_of_detectors] = size(locations(i).upstream.detectors);
            for j = 1:number_of_detectors
                AFSTANDEN = [AFSTANDEN locations(i).upstream.detectors ...
                    (j).position_downstream * -1];
            end
        end
        [dim0 number_of_detectors] = size(locations(i).detectors);
        for j = 1:number_of_detectors
            AFSTANDEN = [AFSTANDEN locations(i).detectors(j).position_upstream];
        end
    end
    STEPS = -600:stepsize:900;
    NUMBERSTEPS = size(STEPS,2);
    for i = 1:NUMBERSTEPS-1
        LIJST{i,1} = ['(' num2str(STEPS(i)) ') - (' num2str(STEPS(i+1)) ')m'];
        LIJST{i,2} = ...
            [num2str(size(find(AFSTANDEN>STEPS(i+1)&AFSTANDEN>STEPS(i)),2))];
    end
    HEADERS{1,1} = ['interval'];
    HEADERS{1,2} = ['#detectors'];
    LIJST = [HEADERS; LIJST];
    filename = ['listinterval' num2str(stepsize) '.xls'];
    xlswrite(filename, LIJST);
```

Script to plot the histograms with the estimated distribution. When slightly changed can be applied on other detectors.

```
% Version 2014-10-21 by M.J. Hovenga
function nargin = BFS_histo_percentage_flow(locations, distancebin, lane, bin)

close all

XLABEL = 'fraction of flow';
YLABEL = 'f';
XLIM = [0 0.7];
YLIM = [0 0.1];
if distancebin == 1
    strbin = '(0)-(300)m';
elseif distancebin == 2
    strbin = '(300)-(600)m';
elseif distancebin == 3
    strbin = '(600)-(900)m';
end

figure('visible','off','Position',[1 1 1920/2 1080/2.5])
binranges = 0:2:100;
bincount = zeros(51,1);
TEMP = [];
[dim0, number_of_locations] = size(locations);
for i = 1:number_of_locations
    try
        if strcmp(locations(i).distance(distancebin).empty, 'no')
            [dim0, number_of_detectors] = ...
                size(locations(i).distance(distancebin).detectors);
```

```

        for j = 1:number_of_detectors
            X = locations(i).distance(distancebin).detectors(j).lane ...
                (lane).data.bin(bin).percentageflow*100;
            bincount_add = histc(X,binranges);
            bincount = bincount + bincount_add;
            TEMP = [TEMP; ...
                locations(i).distance(distancebin).detectors(j).lane(lane).data.bin(bin).percentageflow*100]
        end
    end
end
AREA = trapz(binranges,bincount);
num = 1;
for i = 1:number_of_locations
    try
        if strcmp(locations(i).distance(distancebin).empty, 'no')
            [dim0, number_of_detectors] = size(locations(i).distance ...
                (distancebin).detectors);
            for j = 1:number_of_detectors
                X = locations(i).distance(distancebin).detectors(j).lane ...
                    (lane).data.bin(bin).percentageflow*100;
                bincount_min = histc(X,binranges);
                hold on
                A(num) = bar(binranges/100, bincount/AREA, 'stacked', ...
                    'FaceColor', locations(i).color, 'LineWidth', 0.25);
                B(num,1) = locations(i).roadnumber;
                B(num,2) = locations(i).fromKM;
                B(num,3) = locations(i).toKM;
                bincount = bincount - bincount_min;
                num = num + 1;
            end
        end
    end
end
MEDIAN = median(TEMP);
STD = std(TEMP);
SKEW = skewness(TEMP);
KURT = kurtosis(TEMP);
phat = mle(TEMP, 'distribution', 'norm');
MEAN1 = phat(1);
STD1 = phat(2);
binranges = 0:0.1:150;
C = plot(binranges/100,normpdf(binranges,MEAN1,STD1), 'r-', 'LineWidth', 2);
C1 = plot(0.999999, '-w');
C2 = plot(0.999999, '-w');
C3 = plot(0.999999, '-w');
C4 = plot(0.999999, '-w');
length_num = size(A,2);
for i = 1:length_num
    PLOTS(1,i) = A(i);
    TEXTS{1,i} = [ 'A' num2str(B(i,1)) ' ' num2str(B(i,2)) '- ' ...
        num2str(B(i,3)) 'km' ];
end
PLOTS(1,end+1) = C;
PLOTS(1,end+1) = C1;
PLOTS(1,end+1) = C2;
PLOTS(1,end+1) = C3;
PLOTS(1,end+1) = C4;
TEXTS{1,end+1} = 'estimated distribution: ';
TEXTS{1,end+1} = [ '\mu_e= ' num2str(MEAN1) ];
TEXTS{1,end+1} = [ '\sigma_e= ' num2str(STD1) ];
TEXTS{1,end+1} = [ 'skewness= ' num2str(SKEW) ];
TEXTS{1,end+1} = [ 'kurtosis= ' num2str(KURT) ];
legend(PLOTS,TEXTS, 'Location', 'northeastoutside')
xlim(XLIM)
ylim(YLIM)
grid on
hold off
xlabel(XLABEL);
ylabel(YLABEL);
drm = [ 'distribution/percentageflow/' ];

```

```

if lane == 1
    strlane = 'median lane';
elseif lane == 2
    strlane = 'center lane';
elseif lane == 3
    strlane = 'outside lane';
end
fname = [strbin ' - fraction of flow - ' strlane];
exportfig(fname,drm);
close all

```

Script to make the distance-interval plots, for the other indicators changes are required.

```

% Version 2014-10-21 by M.J. Hovenga
function nargin = MLE_distance_percentage_flow(locations, locations_BFS, lane, bin)
close all

figure('visible','off')
TEMP1 = [];
if lane == 3
    lanestr = 'outside lane';
elseif lane == 2
    lanestr = 'center lane';
elseif lane == 1
    lanestr = 'median lane';
end
XLABEL = 'distance';
YLABEL = 'fraction of the flow';
XLIM = [-600 900];
YLIM = [0 0.6];

% Upstream (-600)-(-300)m
TEMP1 = [];
MEAN = [];
STD = [];
distancebin = 2;
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    try
        if strcmp(locations(i).upstream.distance(distancebin).empty, 'no')
            [dim0 number_of_detectors] = ...
                size(locations(i).upstream.distance(distancebin).detectors);
            for j = 1:number_of_detectors
                TEMP1 = [TEMP1; ...
                    locations(i).upstream.distance(distancebin).detectors(j).lane(lane).data.bin(bin).perc
            end
        end
    end
end
[f, x_values] = ecdf(TEMP1);
phat = mle(TEMP1, 'distribution', 'norm');
MEAN = phat(1);
STD = phat(2);
DISTANCE(1).MEAN = MEAN;
DISTANCE(1).STD = STD;
DISTANCE(1).AFSTAND = -600;
DISTANCE(1).SIZE = length(TEMP1);

% Upstream (-300)-(0)m
TEMP1 = [];
MEAN = [];
STD = [];
distancebin = 1;
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    try
        if strcmp(locations(i).upstream.distance(distancebin).empty, 'no')
            [dim0 number_of_detectors] = ...
                size(locations(i).upstream.distance(distancebin).detectors);
            for j = 1:number_of_detectors

```



```

                TEMP1 = [TEMP1; ...
                        locations(i).upstream.distance(distancebin).detectors(j).lane(lane).data.bin(bin).percentageflow];
            end
        end
    end
end
[f,x_values] = ecdf(TEMP1);
phat = mle(TEMP1, 'distribution', 'norm');
MEAN = phat(1);
STD = phat(2);
DISTANCE(2).MEAN = MEAN;
DISTANCE(2).STD = STD;
DISTANCE(2).AFSTAND = -300;
DISTANCE(2).SIZE = length(TEMP1);

% 0-300m
distancebin = 1;
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    if strcmp(locations(i).distance(distancebin).empty, 'no')
        [dim0 number_of_detectors] = size(locations(i).distance(distancebin).detectors);
        for j = 1:number_of_detectors
            TEMP1 = [TEMP1; ...
                    locations(i).distance(distancebin).detectors(j).lane(lane).data.bin(bin).percentageflow];
        end
    end
end
[f,x_values] = ecdf(TEMP1);
phat = mle(TEMP1, 'distribution', 'norm');
MEAN = phat(1);
STD = phat(2);
DISTANCE(3).MEAN = MEAN;
DISTANCE(3).STD = STD;
DISTANCE(3).AFSTAND = 0;
DISTANCE(3).SIZE = length(TEMP1);

% 300-600m
TEMP1 = [];
MEAN = [];
STD = [];
distancebin = 2;
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    if strcmp(locations(i).distance(distancebin).empty, 'no')
        [dim0 number_of_detectors] = size(locations(i).distance(distancebin).detectors);
        for j = 1:number_of_detectors
            TEMP1 = [TEMP1; ...
                    locations(i).distance(distancebin).detectors(j).lane(lane).data.bin(bin).percentageflow];
        end
    end
end
[f,x_values] = ecdf(TEMP1);
phat = mle(TEMP1, 'distribution', 'norm');
MEAN = phat(1);
STD = phat(2);
DISTANCE(4).MEAN = MEAN;
DISTANCE(4).STD = STD;
DISTANCE(4).AFSTAND = 300;
DISTANCE(4).SIZE = length(TEMP1);

% 600-900m
TEMP1 = [];
MEAN = [];
STD = [];
distancebin = 3;
[dim0 number_of_locations] = size(locations);
for i = 1:number_of_locations
    if strcmp(locations(i).distance(distancebin).empty, 'no')
        [dim0 number_of_detectors] = size(locations(i).distance(distancebin).detectors);
        for j = 1:number_of_detectors

```

```

        TEMP1 = [TEMP1; ...
                locations(i).distance(distancebin).detectors(j).lane(lane).data.bin(bin).percentageflow];
    end
end
end
[f,x_values] = ecdf(TEMP1);
phat = mle(TEMP1, 'distribution', 'norm');
MEAN = phat(1);
STD = phat(2);
DISTANCE(5).MEAN = MEAN;
DISTANCE(5).STD = STD;
DISTANCE(5).AFSTAND = 600;
DISTANCE(5).SIZE = length(TEMP1);

% BFS
TEMP1 = [];
[dim0 number_of_locations] = size(locations_BFS);
for i = 1:number_of_locations
    [dim0 number_of_detectors] = size(locations_BFS(i).detectors);
    for j = 1:number_of_detectors
        TEMP1 = [TEMP1; ...
                locations_BFS(i).detectors(j).lane(lane).data.bin(bin).percentageflow];
    end
end
BFS.DATA = TEMP1;
phat = mle(TEMP1, 'distribution', 'norm');
MEAN = phat(1);
STD = phat(2);
BFS.MEAN = MEAN;
BFS.STD = STD;
BFS.SIZE = length(TEMP1);

% KS-test
[intervals] = size(DISTANCE,2);
[lengthdata] = length(BFS.DATA);
for i = 1:intervals
    distancemean(i) = DISTANCE(i).MEAN;
    distancelow(i) = DISTANCE(i).MEAN - DISTANCE(i).STD;
    distancehigh(i) = DISTANCE(i).MEAN + DISTANCE(i).STD;
    afstanden(i) = DISTANCE(i).AFSTAND;

    % ks-test
    DISTANCE(i).ESTIMATED_DATA = random('Normal',DISTANCE(i).MEAN, ...
        DISTANCE(i).STD,[lengthdata,1]);
    [h,p,ksstat] = kstest2(BFS.DATA,DISTANCE(i).ESTIMATED_DATA);
    display(['fraction_of_flow - interval: ' num2str(i) ' lane = ' ...
        num2str(lane) ' h= ' num2str(h) ' p = ' num2str(p)])
    h = [];
end
distancemean(end+1) = distancemean(end);
distancelow(end+1) = distancelow(end);
distancehigh(end+1) = distancehigh(end);
afstanden(end+1) = 900;

close all;
figure('visible','off','Position',[1 1 1920/2 1080/2.5])
hold on
line([0 0],[-1000 10000],'LineStyle','--','Color','k')
A = line([-3000 1900],[BFS.MEAN BFS.MEAN],'LineStyle','-','...',
    'Color','b','LineWidth',1.5);
A1 = line([-3000 1900],[BFS.MEAN-BFS.STD BFS.MEAN-BFS.STD], ...
    'LineStyle',':', 'Color','b','LineWidth',1.5);
line([-3000 1900],[BFS.MEAN+BFS.STD BFS.MEAN+BFS.STD], 'LineStyle',':', ...
    'Color','b','LineWidth',1.5)
B = stairs(afstanden, distancemean, 'r-', 'LineWidth', 2);
B1 = stairs(afstanden, distancelow, 'r:', 'LineWidth', 2);
stairs(afstanden, distancehigh, 'r:', 'LineWidth', 2)
hold off
ylim(YLIM)
xlim(XLIM)
set(gca, 'XTick', -600:300:900)

```

```
xlabel(XLABEL);
ylabel(YLABEL);
text_A = [ '\mu_e basic freeway segment' ];
text_A1 = [ '\sigma_e basic freeway segment' ];
text_B = [ '\mu_e acceleration lane' ];
text_B1 = [ '\sigma_e acceleration lane' ];
legend([A A1 B B1],text_A,text_A1,text_B,text_B1,'Location',...
        'northeastoutside')
drm = [ 'distance/percentageflow/' ];
fname = [ 'distance - ' lanestr ' - fraction of the flow' ];
exportfig(fname,drm)
close all
```