



# Mikrosimulering av trafikk

Teori og praksis

**Atle Kristian Svela**

Master i veg og jernbane

Innlevert: mai 2015

Hovedveileder: Arvid Aakre, BAT

Medveileder: Tarjei Østmoe, Statens vegvesen

Norges teknisk-naturvitenskapelige universitet  
Institutt for bygg, anlegg og transport







Oppgavens tittel: Mikrosimulering av trafikk – teori og praksis	Dato:2015-05-15		
	Antall sider (inkl. bilag): 92		
	Masteroppgave	x	Prosjektoppgave
Navn: Atle Kristian Svela			
Faglærer/veileder: Arvid Aakre, BAT			
Eventuelle eksterne faglige kontakter/veiledere: Tarjei Østmoe, Statens vegvesen og Sebastian Nerem, Norconsult Sandvika.			

<p>Ekstrakt:</p> <p>Etter hvert som transportbehovet øker og arealene reduseres, er det nødvendig å tenke nytt og kreativt for å komme opp med de beste løsningene for fremtidens transportbehov. For å kunne vurdere konsekvensene av en løsning trenger planleggeren et verktøy hvor han kan teste ut forskjellige tiltak, uten å måtte utføre resurskrevende fullskalaforsøk. Den mest aktuelle løsningen for et slikt verktøy er i dag mikrosimuleringsmodeller.</p> <p>Denne oppgaven belyser de viktigste aspektene med mikrosimuleringsmodeller og de viktigste bestanddelene i en slik modell. I tillegg beskriver den prosessen rundt det å utføre et simuleringsopdrag.</p> <p>Videre er det gitt en tilsvarende beskrivelse av simuleringsverktøyet VISSIM, og hvordan VISSIM kan brukes for å vurdere konsekvensene av tiltak på vegnettet. I denne oppgaven er dette vist med å simulere et før og etter tiltak på bakgrunn av en problemstilling i en gitt case.</p> <p>Casen tar utgangspunkt i Statens vegvesen og Stavanger kommune sitt mål om at bussen skal kunne kjøre fra Byterminalen, gjennom Kannik, til E-39 uten forsinkelse, Problemstillingen vil være hvilke konsekvenser dette vil ha for biltrafikken. Som svar på denne problemstillingen er det i denne oppgaven forslått å innføre midtstilt kollektivfelt med bussprioritering gjennom kryssene.</p>
--

Stikkord:

1. Mikrosimuleringsmodeller
2. VISSIM
3. Midtstilt kollektivfelt
4. Bussprioritering

(sign.)





# Forord

---

Denne masteroppgaven er skrevet som en del av mitt studie ved NTNU-videre og markerer slutten av min videreutdanning. Selv om oppgaven er skrevet ved Institutt for bygg, anlegg og transport ved NTNU i Trondheim, ble oppgaven skrevet hjemme på Klepp høst/vår 2014/2015.

Hovedmålet med oppgaven var å tilegne seg kompetanse innen mikrosimulering og simuleringsprogrammet VISSIM. Samtidig å legge et grunnlag jeg kan bygge videre på som modellbygger. Av PTV, som er utvikleren av VISSIM, har jeg fått tilgang til programmet gjennom en studentlisens. VISSIM er et komplekst program, slik at mye av resursene som er brukt i denne oppgaven har gått med til å lære meg programmet.

Grunnen til at jeg valgte å fordype meg innen dette temaet, er at jeg tror trafikksimulering vil bli stadig viktigere innen samfunnsplanlegging, for å komme opp med de beste løsningene for fremtidens transportbehov. I tillegg har jeg sett at etterspørselen etter trafikksimuleringer har økt de siste årene.

For å kunne gjennomføre denne oppgaven har jeg vært avhengig av hjelp fra flere: Jeg vil takke min arbeidsgiver Norconsult, og mine kollegaer, som har vært behjelpelige og vist forståelse, slik at denne oppgaven kunne gjennomføres. Jeg vil takke Statens vegvesen ved Tarjei Østmo, som har vært min lokale veileder og vært behjelpelig med trafikk tall og bakgrunnsinformasjon.

Jeg vil benytte anledningen til å takke Arvid Aakre for å ha vært med å utforme oppgaven og å guide meg i rett retning.

Jeg vil rette en spesiell takk til Sebastian Nerem i Norconsult som har vært svært avgjørende for min forståelse av VISSIM.

Jeg retter en stor takk til familie og svigerfamilie som har bistått med barnepass og diverse, slik at min kone og jeg fikk hverdagen til å gå rundt. Til slutt vil jeg

takke min kone som har vist tålmodighet og forståelse for alle ukene og kveldene jeg har vært opptatt med studie, og i tillegg lest korrektur på oppgaven.

# Sammendrag

---

Hensikten med denne oppgave er at jeg skal styrke min kunnskap om mikrosimuleringsmodeller, og lære meg de grunnleggende elementene ved mikrosimuleringsprogrammet VISSIM. Dette gjøres ved å simulere en case.

Casen bygger på Statens vegvesen og Stavanger kommune sin målsetning om at bussen skal kunne kjøre fra Byterminalen, gjennom Kannik, til E39 uten forsinkelse. Problemstillingen i oppgaven vil være å foreslå et tiltak som ivaretar dette målet, og vurdere konsekvensene tiltaket vil ha for den ordinære trafikken. Konsekvensene er i denne oppgaven definert som forsinkelse. Jeg har valt å se på forslaget om å anlegge midtstilt kollektivfelt gjennom området, og ved hjelp av signalanlegg, prioritere bussen gjennom kryssene.

For å få en bedre forståelse av mikrosimuleringsmodeller innledet jeg oppgaven med å utføre litteraturstudier om emnet. Litteraturen viser at de fleste mikrosimuleringsmodeller begynte som akademiske eksperimenter for å beskrive trafikken på en matematisk måte. Modellene har gjennom årenes løp utviklet seg fra å være enkle mikromodeller til bruk i forskning, til avanserte kommersielle mikrosimuleringsmodeller. De fleste modellene består av flere delmodeller som Car-following-, Lane-change-, og Gap-acceptance-modellen. Disse delmodellene er utviklet for å gjenskape forskjellige handlinger som utføres i trafikken. For å etterligne de tilfeldighetene man kan observere i trafikken, er de fleste mikrosimuleringsmodellene stokastiske.

Ved å bruke VISSIM til å simulere en før og etter situasjon, har jeg fått ett innblikk i hvilke utfordringer og usikkerheter man møter når man utfører trafikksimuleringer. Jeg har lært meg de grunnleggende elementene for hvordan man bygger opp en simuleringsmodell i VISSIM. Videre har jeg opparbeidet meg en viss erfaring om hvilke parametere og parameterverdier som er avgjørende for resultatet.

Trafikkdataene som brukes i denne oppgaven er data som bygger på registreringer gjort i 2006. Trafikkregistreringer gjort i Bergelandstunellen viser at trafikken i Kannik har vært stabil, og man kan anta at den har ligget på kapasitetsgrensen

siden 1998. Man kan dermed anta at dataene fra 2006 enda er akseptabelt brukbare.

For å kunne vurdere konsekvensene av tiltaket må man vite dagens forsinkelse. Det ble derfor etablert en modell som beskriver 0-alternativet. Resultatene fra simuleringen viser at modellen avviker noe mindre trafikk enn det som er registrert i felt.

Som forventet økte forsinkelsen betraktelig ved å ta to felt fra den ordinære trafikken og gjøre dem om til kollektivfelt. Resultatene fra simuleringen må imidlertid sies å være usikre da det viste seg å være en resurskrevende oppgave å optimalisere de kjøretøystyrte signalanleggene. I tillegg er det store usikkerheter i de fremtidige trafikktallene.

Emnet mikrosimulering har et stort omfang, og jeg hadde i løpet av oppgaven problemer med å begrense meg. Det finnes mange aspekter vedrørende emnet som ikke dekkes i denne oppgaven, men ut fra tidsaspektet på oppgaven er jeg fornøyd med de erfaringer jeg gjorde.

# Summary

---

The purpose of this thesis is to strengthen my knowledge in microsimulation models, and learn the basic elements of the microsimulation software VISSIM. This is done by conducting a simulation of a case.

The case is based on Statens vegvesen and Stavanger municipality's goal that the bus can travel between the Bus terminal, through Kannik, to E39 without delay. The task of this case will be to suggest a measure to forefill this goal and consider the consequences this measure will have on the ordinary traffic. The consequences are in this task defined as delay. I have chosen to place a BRT between the lanes for ordinary traffic through the area, and give the bus priority through the intersection with help from signal lights.

To get a better understanding of microsimulation models, I began this thesis with studying the literature on the subject. The literature showed that most microsimulation models started out as academic experiments to mathematically describe the traffic. The models have over the years evolved from simple micromodels for use in research, to advanced commercial microsimulation models. Most of the models consist of several sub-models, Car-following-, Lane-Change- and Gap acceptance model. These sub-models are designed to imitate different actions carried out in traffic. To replicate the different varieties in driving behaviour one can observe in the traffic, most microsimulation models are made stochastic.

By using VISSIM to simulate a before and after situation, I've been giving an understanding of the challenges and the uncertainties one encounters when conducting a traffic simulation. I learned the basic elements on how to build a simulation model in VISSIM. Furthermore, I have gained some experience of what parameters and parameter values that are essential for the result.

The traffic data used in this thesis are based on registrations in 2006. Registrations from Bergelandtunnelen show that the traffic in Kannik has been stable. One can assume that the roads in Kannik have been at the capacity limit since 1998. One can therefore assume that the traffic data from 2006 is still acceptable for use.

In order to consider the consequences of the measure, one has to compare it with existing delay. It was therefore established a model that describes the 0-option. The results of the simulation show that the model is able to complete less traffic than is recorded in the field.

As expected, the delays increased considerably by taking two lanes from the ordinary traffic, and make them into bus lanes. The results of the simulation must be said to host some uncertainties as it was a difficult task to optimize the signalling control systems. Additionally uncertainties will be found in the future traffic volumes.

The topic microsimulation has a wide range, and I had in this thesis difficulties of limiting myself. There are many aspects of the topic that are not covered in this thesis, but from the aspect of time given, I am satisfied with the experiences I did.





# Innhold

---

<b>FORORD</b> .....	<b>I</b>
<b>SAMMENDRAG</b> .....	<b>III</b>
<b>SUMMARY</b> .....	<b>V</b>
<b>INNHold</b> .....	<b>VIII</b>
<b>BILDE OG FIGURER</b> .....	<b>XII</b>
<b>1 INNLEDNING</b> .....	<b>2</b>
1.1 BAKGRUNN .....	2
1.2 MÅLSETNINGEN OG PROBLEMSTILLING.....	2
1.3 LESERVEILEDNING .....	3
<b>2 METODE</b> .....	<b>4</b>
<b>3 EN INNFØRING I MIKROSIMULERINGSMODELLER</b> .....	<b>6</b>
3.1 INNLEDNING .....	6
3.2 FORDELER MED TRAFIKKSIMULERING .....	8
3.3 UTFORDRINGER MED TRAFIKKSIMULERING .....	8
3.4 NETTVERKSBYGGING .....	9
3.5 INNGANGSDATA .....	10
3.6 CAR-FOLLOWING .....	12
3.6.1 GHR- modeller .....	12
3.6.2 Modell basert på sikkerhetsavstand .....	13
3.6.3 Psykofysiske modeller .....	13
3.6.4 Modeller basert på Fuzzy Logic .....	14
3.7 LANE-CHANGE-MODELL .....	15
3.8 GAP-ACCEPTANCE MODELL.....	18
3.9 UTVELGELSE OG TILFELDIGHETER.....	20
3.10 KVALITETSSIKRING AV MODELLEN .....	21
3.10.1 Verifisering .....	21

3.10.2	<i>Kalibrering</i> .....	21
3.10.3	<i>Validering</i> .....	22
3.11	RESULTATER .....	24
3.12	METODIKK .....	26
<b>4</b>	<b>ET INNBLIKK I VISSIM</b> .....	<b>28</b>
4.1	GENERELT OM VISSIM.....	28
4.2	INFRASTRUKTUR .....	29
4.2.1	<i>Vegen</i> .....	29
4.2.2	<i>Andre elementer</i> .....	29
4.3	TRAFIKK.....	30
4.3.1	<i>Kjøreruter</i> .....	30
4.3.2	<i>VISSIMS atferdstyper</i> .....	31
4.3.3	<i>VISSIMs Car-following modell</i> .....	31
4.3.4	<i>VISSIMs Lane-change modell</i> .....	33
4.3.5	<i>Kjøretøysammensetning</i> .....	35
4.3.6	<i>Trafikkgenerering</i> .....	36
4.4	KONTROLL.....	36
4.4.1	<i>Signalanlegg</i> .....	36
4.4.2	<i>VISSIMS gap acceptance-modell</i> .....	37
4.5	RESULTAT OG OUTPUT .....	41
4.6	SIMULERINGSPARAMETERE .....	44
<b>5</b>	<b>DAGENS SITUASJON</b> .....	<b>46</b>
5.1	BAKGRUNN FOR CASEN .....	46
5.2	TRAFIKKGRUNNLAG .....	47
5.3	DAGENS KOLLEKTIVTILBUD.....	48
<b>6</b>	<b>SIMULERING AV DAGENS SITUASJON</b> .....	<b>50</b>
6.1	INNLEDNING .....	50
6.2	BYGGE MODELLEN .....	50
6.3	SENTRALE PARAMETERE .....	51
6.3.1	<i>Atferdsparametere</i> .....	51
6.3.2	<i>Nettverksparametere</i> .....	51
6.3.3	<i>Kjøretøyparametere</i> .....	51
6.3.4	<i>Simuleringsparametere</i> .....	52
6.4	UTFORDRINGER.....	53

6.4.1	<i>Rundkjøring</i> .....	53
6.4.2	<i>Ruter</i> .....	55
6.5	VERIFISERING, KALIBRERING OG VALIDERING AV MODELLEN.....	56
6.5.1	<i>Verifisering</i> .....	56
6.5.2	<i>Kalibrering</i> .....	56
6.5.3	<i>Validere</i> .....	58
6.6	RESULTAT.....	59
<b>7</b>	<b>TILTAK</b> .....	<b>60</b>
7.1	BAKGRUNN.....	60
7.2	MIDTSTILT VS. SIDESTILT KOLLEKTIVFELT.....	60
7.3	KRYSSLØSNINGER.....	61
7.4	TRAFIKKTALL.....	61
7.5	KOLLEKTIVTRAFIKK.....	62
<b>8</b>	<b>SIMULERING AV TILTAK</b> .....	<b>64</b>
8.1	BYGGE MODELLEN.....	64
8.2	PARAMETERINNSTILLINGER.....	65
8.3	ANDRE ELEMENTER.....	65
8.4	MODELLERE SIGNALANLEGG.....	65
8.4.1	<i>Signalregulert rundkjøring med midtstilt kollektivfelt</i> .....	68
8.4.2	<i>Signalregulerte X-kryss med midtstilt kollektivfeltprioritering</i> .....	70
8.5	VERIFISERING, VALIDERING OG KALIBRERING.....	71
8.6	RESULTAT.....	72
<b>9</b>	<b>RESULTAT OG DISKUSJON</b> .....	<b>74</b>
9.1	RESULTAT FRA CASEN.....	74
9.2	DISKUSJON AV RESULTATENE.....	75
9.3	DISKUSJON OG ERFARINGER MED VISSIM.....	76
<b>10</b>	<b>KONKLUSJON OG VIDERE ARBEID</b> .....	<b>80</b>
10.1	KONKLUSJON.....	80
10.1.1	<i>Målsetningens konklusjon</i> .....	80
10.1.2	<i>Casens konklusjon</i> .....	81
10.2	VIDERE ARBEID.....	81
10.2.1	<i>Mikrosimulering</i> .....	82
10.2.2	<i>Casen</i> .....	82
	<b>REFERANSELISTE</b> .....	<b>84</b>

<b>11</b>	<b>VEDLEGG OG BILAG .....</b>	<b>86</b>
11.1	VEDLEGG .....	86
11.2	BILAG .....	86
11.3	BILAG 1 .....	1
11.4	BILAG 2 .....	5
11.5	BILAG 3 .....	6

# Bilde og figurer

---

## Bilde

Bilde 1-Registrering av reisetid .....	58
Bilde 2-Midsilt Kollektivt i Hillevåg, Foto:Stavanger Aftenblad .....	68

## Figurer

Figur 1-Illustrasjon GAM (Arvid AÅkre) .....	19
Figur 2-Eksempel på resultat fra VISSIM .....	25
Figur 3-Eksempel på resultat fra VISSIM .....	25
Figur 4- Eksempel på en simuleringsprosess (FHWA) .....	27
Figur 5-Hovdblokkene i VISSIM (PTV) .....	28
Figur 6-Eksempel på «dummy connector» .....	29
Figur 7-Predfinetere adferdstyper .....	31
Figur 8-Illustrasjon på VISSIM's Car-following modell .....	32
Figur 9-Wiedeman74	Figur 10-Wiedeman99.....33
Figur 11-Dialogboks, Lane-Change .....	34
Figur 12-Illustrasjon, Conflict area.....	37
Figur 13-Parametertabell, Conflict area.....	38
Figur 14-Dialogboks for datainnsamling .....	42
Figur 15-Offline data .....	43
Figur 16-Online data .....	43
Figur 17-3D-animering .....	43
Figur 18-Dialogboks, Simuleringsparamtere.....	44
Figur 19-Oversiktskart for Kannik .....	46
Figur 20-Illustarsjon, alternativ 1 .....	53

Figur 21-Illustasjon, alternativ 2 .....	54
Figur 22-Ruter.....	55
Figur 23-Feilmeldingsfil .....	56
Figur 24-Strekning for reisetidsregistrering .....	58
Figur 25-Oversiktskart, strekninger .....	59
Figur 26-Resultat, reisetid.....	59
Figur 27-Oversiktskart, kollektivfelt .....	60
Figur 28-Illustrasjon, Midtstillt kollektivfelt.....	64
Figur 29-Flowchart med forklaring .....	66
Figur 30-VisVAP .....	67
Figur 31-Detektroplassering, Rundkjøring .....	69
Figur 32-Illustasjon, kryss 1 .....	69
Figur 33-Illustasjon, kryss 3 .....	70
Figur 34-Oversiktskart, strekning .....	72
Figur 35-Resultat, reisetid.....	72
Figur 36-Oversiktskart, reisetid .....	74
Figur 37-Trafikktall,2006 .....	5
Figur 38-Beregna reisetid .....	6

## **Tabell**

Tabell 1-Sammenstilling av reisetid .....	75
Tabell 2-Reisetidsregistreringer.....	6



# 1 INNLEDNING

---

## 1.1 Bakgrunn

Etter hvert som man opplever stadig større avviklingsproblemer på vegnettet i de større norske byene, begynner politikere og planleggere å innse at man ikke kan bygge seg ut av disse problemene. Løsningen blir da å tenke nytt og kreativt. Da tiltak på vegnettet i de fleste tilfeller medfører store kostnader og ulemper, har man oftest ikke muligheter for å «prøve seg fram» for å finne den beste løsningen. Det ideelle vil være å ha en plattform der planleggeren kan teste ut sine teorier uten å måtte utføre omfattende fullskala forsøk. Den mest aktuelle løsningen for en slik plattform er i dag mikrosimuleringsmodeller. Simulering av trafikktiltak er i dag relativt lite utbredt i Norge sammenlignet med våre naboland. Dette kan skyldes at slike simuleringer ofte har et høyt brukersnitt og er resurskrevende. I tillegg finnes det begrenset med kursing og undervisning innenfor dette fagfeltet. De fleste modellutviklerne jeg har vært i kontakt med, har fått sin kompetanse i utlandet. Da jeg synes trafikksimulering er et interessant tema, og ønsker å styrke kompetansen min innen dette fagfeltet, er dette utgangspunktet for denne masteroppgaven. Samtidig tror jeg at trafikksimulering vil bli et stadig viktigere verktøy innen samfunnsplanlegging.

Den beste måten å tilegne seg kunnskap og erfaring innen trafikksimulering er å ta for seg et trafikkbilde, også kalt case, og simulere denne. I denne oppgaven har jeg fått casen og problemstillingen av Statens vegvesen

## 1.2 Målsetningen og problemstilling

Målsetningen med denne oppgaven er å styrke min kunnskap innen mikrosimuleringsfaget, både når det gjelder å skaffe meg en forståelse av teorien bak mikrosimuleringsmodeller, og hvordan utføre et simuleringsoppdrag i praksis. I tillegg skal jeg lære meg de grunnleggende elementene i det tyske mikrosimuleringsprogrammet VISSIM, og få en bedre forståelse av bruken og teorien bak programmet ved hjelp av å simulere en case.

Problemstillingen tar utgangspunkt i Statens vegvesen og Stavanger kommune sitt mål om at bussen skal kunne kjøre fra Byterminalen, gjennom Kannik, til E-39 uten forsinkelse. Oppgaven vil være å foreslå et tiltak, og vurdere hvilke konsekvenser dette vil ha for biltrafikken.



## 1.3 Leserveiledning

Oppgaven kan deles i 4 deler:

**Del 1:** Her gis det en beskrivelse av mikrosimuleringsmodeller for biltrafikk. Dette inkluderer blant annet bakgrunnen for slike modeller, teoretisk grunnlag, anvendelse, inngangsdata og resultater.

**Del 2:** I denne delen gis det en kort innføring i de mest grunnleggende elementene og parameterne i VISSIM.

**Del 3:** Beskriver dagens situasjon og utfordringer i Kannik. Videre beskrives det hvordan jeg gikk fram for å bygge opp VISSIM-modellen for å beskrive dagens trafikksituasjon, og hvilke parameterverdier som ble endret.

**Del 4:** Beskriver tiltaket som svarer på casens problemstilling. Videre beskrives det hvordan jeg gikk fram for å bygge opp VISSIM-modellen, med hovedvekt på modellering av signalanleggene.

Det meste av litteraturen og termologien innen dette fagfeltet er på engelsk. Jeg har ikke brukt resurser på å oversette faguttrykk og kommandoer fra engelsk til norsk.

## 2 METODE

---

Metoden som brukes i denne oppgaven er todelt.

1: I tillegg til å søke i litteraturen, har jeg som metode for å styrke min kunnskap innen mikrosimulering valgt å simulere en case. Casen er utarbeidet i samarbeid med Statens vegvesen og bygger på en virkelig problemstilling.

2: Metoden for å finne konsekvensene av tiltaket vil i denne oppgaven gjøres ved hjelp av mikrosimulering. Konsekvensene defineres som forsinkelse i reisetid.



# 3 EN INNFORING I MIKROSIMULERINGSMODELLER

---

Forskere har siden 50-tallet prøvd å finne metoder og modeller for å gjenskape trafikken på et akademisk nivå. Det har skjedd en stor utvikling fra den første mikromodellen ble lansert av Louis A. Pipes i 1953, fram til de komplekse mikrosimuleringsmodellene man ser i dag. Mye forskning har blitt gjort innen dette emnet og masse litteratur har blitt publisert. I dette kapitlet har jeg prøvd å summere opp de viktigste elementene i en mikrosimuleringsmodell, og prøvd å gi et innblikk i hvordan de fungerer.

## 3.1 Innledning

En mikrosimuleringsmodell (MSM), er et dataverktøy som trafikkplanleggeren kan bruke for å utføre trafikksimuleringer, for å evaluere ITS-tiltak, konsekvensene av nyskapt trafikk, eller andre tiltak på vegnettet. En MSM oppgave er å gjenskape et så virkelighetstro trafikkbilde som mulig under gitte betingelser. Modellen simuler trafikkstrømmer ved å flytte kjøretøy gjennom et nettverk. Modellen gjør nye beregninger av kjøretøyets posisjon, konflikter og adferd flere ganger i sekundet. MSM har tradisjonelt sett vært forbeholdt det akademiske miljøet innen trafikkplanlegging. Etter hvert som datakraften har blitt bedre, har MSM også fått innpass i kommersiell trafikkplanlegging og blitt gjort mer brukervennlige. MSM brukes i dag av trafikkplanleggere i konsulentfirmaer, ved universiter og statlige instanser over hele verden. Det er også et viktig verktøy for trafikkforskere for å bedre kunne forstå trafikken, samt utprøve nye hypoteser og teorier (Barceló, 2010).

De fleste MSM er matematiske og stokastiske modeller som metodisk beskriver interaksjonen mellom kjøretøy som beveger seg i et nettverk. Modellen beskriver hvert kjøretøy individuelt og tilegner kjøretøyet forskjellig adferd og fysiske egenskaper som trekkes ut fra en sannsynlighetsfordeling. Et uttrykk som er sentralt i arbeidet med MSM er parametere. Parameterne beskriver forskjellige egenskaper hos fører, kjøretøy og nettverk. En parameter angir en målestørrelse hvis variasjoner fører til påvirkning av resultatet. Noen parametere er viktige, mens andre kan ses på som mer generelle (Vejregelrådet, 2010). Et eksempel på en parameter kan være hvor hardt en sjåfør er villig til å retardere for å skifte fil. Denne verdien vil igjen påvirke kapasiteten i modellen. De fleste MSM inneholder oftest standardverdier på parametere. Verdien på disse gjenspeiler ofte trafikkbildet i landet som modellen ble utviklet i. I noen tilfeller må man endre på parameteren slik at modellen klarer å gjenskape trafikkbildet til ønsket situasjon. Denne prosessen kalles kalibrering. Kalibrering og validering er to

sentrale begrep innen trafikksimulering og vil bli bedre forklart senere i dette kapittelet. Det finnes også parametere i forbindelse med selve simuleringsprosessen som kan ha påvirkning på resultatet. En sentral parameter er «seeds». Denne parameteren beskriver hvilke tall som skal genereres utfra en tilfeldighetsgenerator, som igjen beskriver de tilfeldighetene man kan observere ute i trafikken. En annen simuleringsparameter er antall simuleringer. Siden de fleste MSM er stokastiske, vil resultatene variere fra en simulering til en annen. For å få et representativt resultat må det kjøres flere simuleringer for å oppnå et statistisk gjennomsnitt.

En MSM bygges opp så realistisk som mulig ved å lage et nettverk av veger og kryssløsninger. Sjøførens adferd gjennom nettverket beskrives ofte av tre delmodeller:

1. Car-following- modellen beskriver hvordan et kjøretøy forholder seg til bilen framforbi.
2. Lane-change- modellen beskriver når og hvordan et kjøretøy skal skifte felt.
3. Gap-acceptance- modellen beskriver hvordan konflikter mellom to kjøretøy skal løses.

Disse delmodellene beskrives mer inngående senere i dette kapittelet.

Modellene er representert i MSM ved hjelp av komplekse algoritmer. Jeg vil i denne oppgaven ikke gå videre inn i disse algoritmene.

Siden slutten av 90-tallet har trafikksimulering fått stadige større plass i trafikkplanleggingen verden over, fordi datakraft og brukervennlighet har blitt bedre. Det er først de senere årene at trafikksimulering har begynt å bli en del av trafikkplanleggingen i Norge. Internasjonalt finnes det flere simuleringsprogram som for eksempel Quadstone Paramics, Tritone, CORSIM og Cube Dynasim. I Norge brukes hovedsakelig to programmer for trafikksimulering, spanske AIMSUN og tyske VISSIM.

Når man jobber med mikrosimuleringsmodeller er det viktig å huske at disse kun er en forenkling av det virkelige trafikkbildet. Det virkelige trafikkbildet er mye mer komplekst enn hva man klarer å simulere. Det er også viktig å huske at en MSM kun er et verktøy, og at det er trafikkplanleggeren sin jobb å kunne tolke resultatene fra en modell.

## 3.2 Fordeler med trafikksimulering

Til forskjell fra vegbyggingsfaget er ikke trafikkteknikken styrt av de fysiske lovene. Det å finne den rette løsningen kan derfor vise seg å være vanskelig. Ofte har en trafikkplanlegger ideer og hypoteser om hvilke tiltak som bør gjøres, men på grunn av kostnader og ulemper er et fullskalaforsøk i de fleste tilfeller uaktuelt. Trafikksimulering kan derfor være en bra måte å komme fram til den beste løsningen, ved å kunne teste ut og evaluere forskjellige tiltak i kontrollert miljø. Mikrosimuleringsmodeller er kraftige og fleksible verktøy som gir masse muligheter. Her kan man sette opp alle typer betingelser og tiltak, for så å kunne hente ut og evaluere data. Mikrosimuleringsmodellen er av noen også kalt trafikkplanleggerens laboratorium (Barceló, 2010).

## 3.3 Utfordringer med trafikksimulering

Selv om trafikksimulering har store fordeler er det viktig å være klar over de utfordringene som oppstår ved å benytte seg av trafikksimulering. Mikrosimuleringsmodeller har et forholdsvis høyt brukersnitt og det stilles store krav både til personen som utarbeider modellen, og til de som bestiller og evaluerer modellen. Der er viktig at begge parter har god forståelse av oppgavens omfang, og er enige om rammene for oppgaven.

Det er viktig å huske at en mikrosimuleringsmodell er et verktøy. Resultatet og utforming av modellen kan variere ut fra planleggerens kunnskap, erfaring og tilnærming til oppgaven. I tillegg er det forskjell blant de forskjellige simuleringsprogrammene. Dette gjør at resultater fra MSM kan være vanskelig å sammenligne. For å redusere disse forskjellene, er det viktig å ha retningslinjer som standardiserer utformingen og metodikken for bruk av mikrosimuleringsmodeller. Våre naboland har kommet et godt stykke på veg i dette arbeidet. I Norge finnes det i dag ingen standardisering eller dokumentasjonskrav i forbindelse med utarbeidelse eller bestilling av trafikksimuleringer. Dette er en stor ulempe, da det er trolig at større investeringer vil basere seg på resultater fra slike modeller i framtiden.

Alle parameterne som en MSM inneholder er dens styrke, men også dens svakhet. Å vite en parameters betydning og hvilke parametere som påvirker andre parametere kan ofte være vanskelig. Det kan være utfordrende og resurskrevende å framskaffe de rette dataene. Det kan også være en utfordring å oppdage feil i modellen, enten de er gjort uvitende eller med overlegg for å manipulere modellen. Utfra animeringen kan man fort bli overbevist om at modellen stemmer, uten å oppdage feil.

En annen svakhet med MSM er hvor realistisk man klarer å gjengi de vurderinger og beslutninger som tas i forbindelse med kjøring i rundkjøringer. Da rundkjøringer er en sentral del av trafikkbildet i Norge, er det en utfordring å simulere urbane områder på en tilfredsstillende måte.

Oppsummert kan man si at man må bruke MSM samvittighetsfullt, og modellbyggeren må kjenne til bakgrunnen og teorien for modellene.

### **3.4 Nettverksbygging**

De fleste mikrosimuleringsmodellene har en eller annen form for nettverksoppbygging. Med nettverksoppbygging menes de fysiske forholdene i et trafikkbilde, som for eksempel veg, kryssområder, skilt, signalanlegg osv. I tillegg

kommer rutevalg fartsgrenser. Nettverksoppbyggingen til de fleste kommersielle mikrosimuleringsmodellene er ganske lik.

De fleste elementene i nettverksbyggingen inneholder attributter og parameterverdier, slik at elementer får en egenskap. Et eksempel kan være en veglenke. Her oppgir man ofte vegbredde, antall felt, stigning og hvilke kjøretøy som kan kjøre der. Med et ortofont eller kart som bakgrunn, trekkes veglenker slik at de danner et nettverk. Veglenkene knyttes sammen med et element som er tiltenkt denne oppgaven, ofte kalt en node. Disse elementene representerer ofte et kryss eller deler av et kryssområde, og kan gis forskjellige egenskaper ved hjelp av attributter.

Da man i en mikrosimuleringsmodell prøver å gjenskape et trafikkbilde så nøyaktig som mulig, er det viktig å gjengi alle komponentene som kan påvirke trafikkbildet så realistisk som mulig. Dette gjelder også veg-geometrien da dette kan påvirke situasjoner der Gap-acceptance- og Lane-Change-modellene brukes.

I nettverksbyggingen defineres også rutene som kjøretøyene skal følge gjennom modellen. Det fins hovedsakelig to måter å definere disse på. Den første måten bygger på registreringer av svingebevegelser i kryss, og kan defineres om statiske ruter. Her definerer man rutene som en viss andel av bilene skal kjøre, gjennom et kryss eller gjennom en modell. Denne metoden er enkel å bruke, men har problem med å gjenskape realistiske feltskifter, da kjøretøyet ikke klarer å planlegge fremtidige feltskifter.

Den andre måten bruker OD-matriser, og kan defineres som dynamiske ruter, også kalt «dynamic route assignment». OD-matrisen beskriver hvor mange biler som skal genereres inn og ut av modellens forskjellige grensesnitt. På bakgrunn av dette beregner programmet den optimale ruten for hvert kjøretøy gjennom modellen. Denne metoden krever litt mer forarbeid da man må utarbeide OD-matrisen, men gir ofte en mer realistisk simulering.

### 3.5 Inngangsdata

Siden man prøver å gjenskape trafikkbildet så virkelighetstro som mulig, er det viktig med presise inngangsdata. Mengden og nøyaktigheten av inngangsdataene vil variere etter modellens nøyaktighetsgrad, og prosjektets resurser. Da datainnsamling er en resurskrevende prosess, må bestiller og modellbygger bli enige om hva som trengs i den aktuelle modellen, og hvordan disse dataene skal innhentes.

I følge The Federal Highway Administration (FHWA) kan inngangsdataene i en mikrosimuleringsmodell kategoriseres i følgende kategorier: Geometri, Trafikkontroll, Trafikketerspørsel, Kalibreringsdata, Kollektiv og myke trafikanter

Geometri	Er data som hovedsakelig brukes i forbindelse med nettverksbyggingen. I de fleste tilfeller får man tilstrekkelig informasjon gjennom å modellere nettverket oppå et kart eller ortofoto. I noen tilfeller vil det være ønskelig å ha data om stigning på vegen. I tillegg kommer fartsgrense inn under denne kategorien.
Trafikkontroll	Omfatter sikt, skilt, som for eksempel vikeplikt og stoppskilt, ITS-systemer, signalhoder, samt eksisterende og nye signalplaner.
Trafikketerspørsel	Denne kategorien er den mest omfattende. Den omfatter trafikktegn og svingebevegelser evt. en OD-matrise, samt trafikkfordelingen lette/tunge kjøretøy.
Kalibreringsdata	Disse dataene brukes i kalibreringsprosessen, og kan for eksempel være kølengde, reisetid, tidsluker osv.
Kollektiv og myke trafikanter	I noen modeller er en nødt til å legge inn data for myke trafikanter, hovedsakelig i forbindelse med kryssinger. Rutetabeller i forbindelse med buss eller trikk kan være sentrale faktorer i en mikrosimuleringsmodell.



I noen modeller er det ønskelig med en detaljert beskrivelse av kjøretøyegenskapene og føreratferd. Dette kan være å innhente data om kjøretøyenes lengde, vekt, motorkraft, akselerasjon, og å kategorisere kjøretøyene i mer en tunge/lette. Data om føreratferd kan være uoppmerksomhet, aggressivitet osv. Disse dataene er oftest svært resurskrevende å hente inn.

## 3.6 Car-following

En Car-following-modell (CFM) er en av de mest grunnleggende delmodellene i en mikrosimuleringsmodell. Den er en adferdsmodell som prøver å beskrive så naturtro som mulig hvordan en sjåfør reagerer på en handling. En handling defineres som endring i fart, avstand, akselerasjon og retardasjon i forhold til bilen foran seg. Et bakomliggende kjøretøys reaksjon på en handling er i de fleste modellene akselerasjon eller retardasjon. Modellen beskriver også hvordan et kjøretøy reagerer annerledes på en handling ved forskjellige hastigheter og avstander. Modellen baserer seg på en situasjon der man har et ledende og et følgende kjøretøy i et kjørefelt, uten muligheten for forbikjøring. En CFM kalles også en mikromodell. Det betyr at modellen er på mikronivå, hvor alle kjøretøyene blir modellert samtidig og individuelt. Den første prototypen av en CFM ble introdusert i 1953 av Louis A. Pipes, og baserte seg på sikker avstand til kjøretøyet framfor (Hoogendoorn, 2003). Etter hvert begynte forskere over hele verden å eksperimentere og utvikle sine egne CFM. I 1958 startet utviklingen av det som skulle bli den mest kjente CFM, GHR-modellen (Brackstone & McDonald, 1999). Det er vanskelig å si hvor mange Car-following-modeller som er utviklet gjennom årenes løp, men oftest kategoriseres de, da det fleste har sitt utspring i noen få modeller. Under beskrives de kategoriene av CFM som oftest omtales i litteraturen.

### 3.6.1 GHR- modeller

Den mest kjente CFM kalles GHR-modellen, da det var trafikkforskerne Gazis, Herman og Rothery som var de mest sentrale i utviklingen av denne modellen på slutten av 50-tallet og begynnelsen av 60-tallet. Modellen ble utviklet ved General Motors i Detroit og kalles også for GM-modellen (Brackstone & McDonald, 1999).

GHR modellen ble utviklet som en RESPONS-STIMULI-modell, der kjøretøyet responderer i form av akselerasjon eller retardasjon på et stimuli, som er representert ved fartsendring mellom to biler. Dersom det ikke var noen fartsendring, ville heller ikke kjøretøyet reagere (Barceló, 2010). Det ble påpekt at modellen var for lineær og ikke tok hensyn til avstanden på luken mellom bilene. Som et resultat av dette ble det utviklet et sensitivitetsledd i ligningen. Dette leddet var proporsjonalt med størrelsen på lukene, og fikk det følgende kjøretøyet til å respondere mer realistisk. Dersom luken var liten, ville kjøretøyet respondere kraftigere enn om luken var større (WENG & WU, 2001).

Denne modellen har blitt endret og modifisert mange ganger opp gjennom årenes løp, og brukes i modifiserte versjoner i kommersielle mikrosimuleringsmodeller.

### **3.6.2 Modell basert på sikkerhetsavstand**

Den første CFM modellen som baserte seg på sikkerhetsavstand ble introdusert av de japanske trafikkforskerne Eiji Kometani og Tsuna Sasalci i 1958. Modellen er en matematisk modell som baserer seg på Newtons bevegelseslover og går ut på at det følgende kjøretøyet skal holde seg på trygg avstand for å unngå kollisjon med kjøretøyet fremfor seg (Brackstone & McDonald, 1999).

I 1980 utviklet Peter G. Gipps modellen ved blant annet å modifisere modellen til å bestå av to ledd. Et ledd representerte akselerasjon og kjøretøyets ønske om å oppnå en gitt hastighet. Det andre leddet, representert ved retardasjon, beskriver de begrensingene kjøretøyet har som følge av et fremforliggende kjøretøy. I tillegg innførte han en sikkerhetsfaktor som tar bedre hensyn til «trygg avstand» i form av reaksjonstiden til følgekjøretøyet (Barceló, 2010). Modellen kan oppsummeres ved at et følgende kjøretøy skal kunne stoppe helt opp uten å kolliderer med kjøretøyet foran. Dette gjør den ved å ta hensyn til sikkerhetsavstander og retardasjonsverdier.

### **3.6.3 Psykofysiske modeller**

Modellen bygger på empiriske undersøkelser og baserer seg på at et kjøretøy vil utføre en handling i form av retardasjon eller akselerasjon dersom en terskelgrense blir brutt. Den første psykofysiske CFM, også kalt terskelmodell, ble lansert av Richard M Michaels i 1968 og bestod av 3 terskler:

Dersom kjøretøyet ikke krysser den første terskelen vil det bety at den har samme hastighet som kjøretøyet framfor seg, og en stor nok luke til å fortsette upåaktet videre. Dersom kjøretøyet har en større hastighet enn kjøretøyet foran, vil det etterhvert komme til et punkt hvor det krysser «oppdage-tersekelen». Dette punktet representerer en avstand der kjøretøyet vil oppdage et fremforliggende kjøretøy og reagere på dette objektet. Når kjøretøyet krysser denne terskelen, vil det retardere for å tilpasse seg kjøretøyet foran. Etterhvert som kjøretøyet retarder vil det også nærme seg kjøretøyet foran. Når avstanden mellom de to kjøretøyene er blitt redusert til en definert avstand, vil kjøretøyet krysse den andre terskelen. Denne terskelen kalles en avstandsterskel, som definerer et område der lukene mellom bilene er så små at man vil få det man kaller en Car-following-situasjon. I området under denne terskelen, vil det kun være små hastighetsendringer. Endringen er enten i positiv eller negativ retning. Størrelsen på disse hastighetsendringene avgrenses av en tredje terskel. Terskelen er illustrert som to

speilvendte parabler. Bryter kjøretøyet en av parablene betyr det at kjøretøyet må akselerere eller retardere for å kunne opprettholde en Car-following-situasjon (Brackstone & McDonald, 1999). Området mellom de to siste tersklene kan defineres som et likevektsområde, der et kjøretøy gjør små fartsendringer i form av akselerasjon eller retardaasjon for å holde en avstand som representerer at bilen følger etter en annen.

Som med de fleste andre CFM, har også denne modellen blitt utviklet og modifisert. De mest vanlige modifiseringene er at det er lagt til flere terskler og områder som skal beskrive Car-following-situasjonen mer realistisk. Det er vanskelig å utføre systematisk kalibrering av slike modeller. Da terskelverdiene er utviklet ut fra undersøkelser om hvordan folk oppfatter trafikkbildet.

#### **3.6.4 Modeller basert på Fuzzy Logic**

Fuzzy Logic kalles også kunstig intelligens og er hovedsakelig et begrep hentet fra IT-verden. Fuzzy Logic ble introdusert i trafikkmiljøet av den japanske trafikkforskeren Shinya Kikuchi i 1992 da han prøvde å introdusere Fuzzy Logic sammen med GHM-modellen. (Brackstone & McDonald, 1999).

Fuzzy Logic er en metode for å få datamaskiner til å etterligne menneskelig beslutningstaking. Utfra et sett med regler og definisjoner reagerer modellen på en hendelse og gjør handlinger som representerer en menneskelig beslutning (IEEE Computational Intelligence Society , 2011). I en Fuzzy Logic Car-following-modell vil en hendelse være forskjell i tid eller avstand mellom et kjøretøy og dens følgebil. Følgebilten vil da reagere utfra et regelsett og variabler som er programmert inn i modellen.

### 3.7 Lane-change-modell

En Lane-change-modell (LCM) er en av de viktigste komponentene i en mikrosimuleringsmodell. LCM er en adferdsmodell som beskriver hvordan og når et kjøretøy utfører et feltskifte på flerfeltsveger. Den klarer ikke å beskrive en tradisjonell forbikjørings situasjon på 2-feltsveger. Da et feltskifte er å anse som en mer krevende handling enn å følge etter en bil, regnes LCM som en mer komplisert modell enn en CFM (SUN, 2009). En LCM består av flere adferdsparametere og er oftest konstruert for å virke sammen med en CFM. Som for CFM har det blitt utviklet flere LCM, noen til kommersielt bruk og noen til forskning. Det har vært forsket på LCM siden begynnelsen av 1980 tallet, og i 1986 presenterte trafikkforskeren Peter G. Gipps sin regelbaserte LCM. Den har vært utgangspunktet for mange LCM som har blitt utviklet i etterkant. Ettersom mikrosimuleringsmodeller har blitt mer populært de siste årene, har man også jobbet med utviklingen av LCM. De viktigste målene med forskningen er å få komplementert inn de menneskelige trekkene som skjer ved et feltskifte, noen som LCM tradisjonelt sett har vært dårlig på (SUN, 2009).

De fleste LCM bygger på prinsippet tvunget eller ønskelig feltskifte. Tvunget feltskifte vil si at et kjøretøy må skifte felt for å fullføre den ruten det har planlagt. Dette kan være viss kjøretøyet tidligere har lagt seg i feil felt, for så å oppdage at dette feltet vil fungerer som en blindveg. Etter hvert som kjøretøyet nærmer seg «siste mulighet» for å skifte felt, også kalt «emergency stop», vil adferden bli mer «desperat». Dette gjøres ved at minimumsverdiene for aksepterte tidsluker reduseres, og maksimal retardasjon økes. Dersom kjøretøyet ikke har skiftet felt innen det har nådd emergency stop, vil kjøretøyet stoppe helt opp, og vente på ledige luker. I urbane strøk og tett kjøring vil man som regel oppleve denne type feltskifte.

Den andre typen, ønskelig feltskifte, er mer gjeldene for frikjøring på flerfelts motorveger. Et ønskelig feltskifte skjer oftest når et kjøretøy blir hindret i å holde eller oppnå en ønsket hastighet. Modellen sjekker kontinuerlig kjøretøyet sin posisjon i nettverket, for å se om ønsket fart eller rute er oppfylt. Viss kriteriene ikke er oppfylt, tar LCM kontroll over kjøretøyet, og starter prosessen med å flytte det over i et annet felt. Når bilen har kommet inn i det nye feltet overtar CFM kontrollen og justerer posisjonen i henhold til kjøretøyet foran.

I tillegg til de opprinnelige feltskifteprinsippene tvungen og ønskelig, har noen modeller også utviklet det som kan kalles strategisk feltskifte. Da begynner sjåføren tidligere å posisjonere seg mot det ønskede feltet. Denne situasjonen vil kunne ligne på «ønskelig feltskifte» og man

vil da kunne se en roligere adferd i forbindelse med feltskifte. Denne typen feltskifte vil være gjeldene på veier med 3 felt eller mer i samme retning (SUN, 2009) .

Et feltskifte kan gjøres på tre måter. Den første er at kjøretøyet som vil skifte felt benytter seg av ledige tidsluker. Den andre er gjennom samarbeid. Kjøretøyet i nabofeltet, også kalt «destinasjonsfeltet», registrer blinklyset for så å redusere hastigheten slik at luken skal bli stor nok, og lar det andre kjøretøyet slippe inn. Den tredje måten er at kjøretøyet tvinger seg inn. Dette fører igjen til at andre kjøretøy tvinges til å redusere hastigheten (Hidas, 2005). Den tredje måten er mest aktuell i urban kjøring. Kvaliteten på en LCM gjenspeiles i hvor godt den klarer å kombinere og beskrive disse tre metodene.

LCM som benytter seg av tidsluker for å skifte felt, kalles for regelbaserte LCM. Dette er de mest brukte modellene. Her prøver kjøretøyet å benytte seg av eksisterende tidsluker, uten å påvirke trafikken i destinasjonsfeltet. Disse inneholder oftest en algoritme som prøver å gjenskape beslutningsprosessen i forbindelse med et feltskifte så realistisk som mulig. Algoritmen består av flere parametere en sjåfør må ta stilling til når og viss han skal skifte felt. Noen parametere er konstante, mens andre kan justeres, slik at man kan tilpasse modellen til å gjenspeile ønsket situasjon.

De mest sentrale parametere er:

Kritisk tidsluke	Beskriver om et feltskifte er fysisk gjennomførbart. Tar hensyn til avstanden mellom bilene i destinasjonsfeltet, enten i sekunder eller meter.
Retardasjon	Kan bestå av flere retardasjonsparametere, maks retardasjon og ønsket retardasjon.
Posisjon	For å bestemme om kjøretøyet kan fortsette i valgt fil beregnes kjøretøyets posisjon iht. definert rute.
Sikkerhetsfaktor	En tilleggsfaktor som legges til kritisk tidsluke, for hvilke luker som aksepteres av føreren.
Fart	I henhold til gitte grenseverdier sjekker denne parameteren om kjøretøyet vil ha noe fartsmessig gevinst av å skifte felt.
Aggressivitet	Ettersom behovet for feltskifte øker, endres kjøreradferden. Denne parameteren reduserer kjøretøyets krav til tidsluker, sikkerhetsavstander og øker retardasjonsverdien.

Etter hvert som de regelbaserte LCM ble etablert, begynte forskere å videreutvikle algoritmene slik at de bedre kunne modellere realistiske feltskifter. Det viktigste var de menneskelige aspektene.

I situasjoner der biler samarbeider ved et feltskifte, inneholder LCM en «høflighetsparameter». Bilen som vil skifte felt sender ut en forespørsel til kjøretøyene i det andre feltet. Utfra stokastiske utvelgelser blir denne forespørselen akseptert eller avist. Dersom forespørselen blir akseptert, vil kjøretøyet i det andre feltet retardere slik at luken foran blir stor nok til at bilen slipper inn (Hidas, 2002).

Noen LCM klarer også å gjenskape situasjoner der man opplever at biler tvinger seg inn i destinasjonsfeltet. Dette påvirker igjen en rekke med biler i dette feltet, som ufrivillig må redusere farten (Hidas, 2005).

Som de andre adferdsmodellene som brukes i mikrosimuleringsmodeller, er også LCM vanskelig å kalibrere. Dette er fordi et feltskifte er et resultat av en rekke ulike menneskelige beslutninger og vurderinger som er vanskelige å beskrive med algoritmer.

### 3.8 Gap-acceptance modell

Litteraturen opererer med to definisjoner for Gap-acceptance-modell. Den ene definisjonen er forbundet med feltskifte og den andre er forbundet med hvordan to trafikkstrømmer løser konflikter. I denne oppgaven defineres Gap-acceptance-modell som en modell til å løse konflikter.

Der hvor trafikkstrømmer krysser, deles eller møtes, vil det oppstå konflikter mellom kjøretøy. Car-following(CFM)- og Lane-change- modellen(LCM) er i de fleste tilfeller ikke designet til å løse slike konflikter. Til dette er det i de fleste mikrosimuleringsmodellene utviklet en Gap-acceptance-modell (GAM). En GAM er, som navnet tilsier, en modell som avgjør om et vikende kjøretøy er villig til å akseptere en luke i en primær trafikkstrøm, enten for å krysse eller for å bli en del av den. Som for LCM kan detaljeringsnivået på GAM variere. Noen modeller er enkle vikepliktsmodeller hvor det kun er de fysiske aspektene som bestemmer om et kjøretøy skal kjøre eller ikke. Andre GAM er mer omfattende og beskriver i tillegg førerens adferd og vurderinger. Felles for de fleste GAM er at de bygger på veletablerte teorier om kritisk tidsluke, følgetid og sikkerhetsfaktor, hvor et kjøretøy i en sekundær trafikkstrøm ønsker å krysse, eller bli en del av primærstrømmen.

Mikrosimuleringsmodellen beregner alle kjøretøyets posisjoner til en hver tid og vil dermed klare å beregne når to kjøretøy vil møtes i et konfliktpunkt. Ofte blir dette konfliktpunktet representert med et konfliktområde, litteraturen omtaler ofte dette området som en «yellow box». I dette område kan det i de fleste GAM kun være et kjøretøy samtidig. Dette reguleres av de fysiske forholdene, som kritisk tidsluke. I tillegg legges det inn en sikkerhetsfaktor for at kjøretøyet skal få kommet seg opp i fart og ikke påvirke kjøretøy fra primærstrømmen. For å gjenspeile et realistisk trafikkbilde, inneholder de fleste modellene en funksjon som beskriver en reduksjon av denne sikkerhetsfaktoren sett i sammenheng med ventetid. Det betyr at dess lenger et kjøretøy må vente for å slippe ut i primærstrømmen, dess mer reduseres sikkerhetsavstanden mot en minimumsverdi.

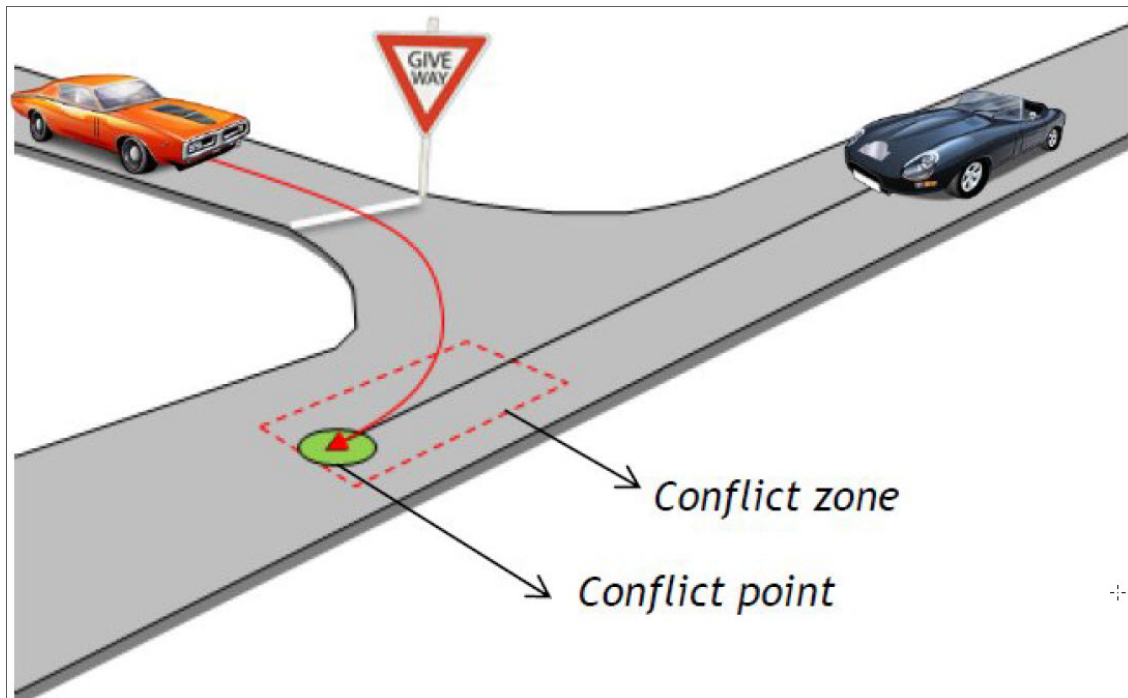
For å unngå blokkeringer i kryss, kan man i noen modeller øke arealet til konfliktområdet. Dette forhindrer at biler fra en sekundær trafikkstrøm kjører ut i krysset og blokkerer primærstrømmen. Dette er spesielt aktuelt i områder hvor det oppstår kø og saktegående trafikk.

Noen GAM beskriver tradisjonell vikeplikt med stopplinje, andre GAM beskriver mer avanserte vikepliktsområder. I en vikepliktsituasjon med stopplinje påvirkes ikke primærstrømmen av biler fra sekundærstrømmen, og bilene slipper inn i primærstrømmen kun



viss lukene er akseptable. I noen av de mer avanserte GAM kan en juster denne påvirkningen for å gjengi et mer realistisk trafikkbilde. Dette kan være nyttig når man skal modellere rundkjøringer. Man kan også justere sekundærkjøretøyets evne til å «lese» trafikken og unngå blokkering i kryss.

Kjøretøyets egenskaper tas også med i GAM sin beregning. Et tungt kjøretøy vil ha dårligere akselerasjon og dermed kreve større kritiske tidsluke og sikkerhetsfaktor.



Figur 1-Illustrasjon GAM (Arvid AÅkre)

### 3.9 Utvelgelse og tilfeldigheter.

I et komplekst trafikkbilde vil det finnes variasjoner blant kjøretøy og sjåførere. Et trafikkbilde en dag, vil sannsynligvis ikke være identisk med dagen før, eller dagen etter. Man kan i teorien beskrive i detalj hvert kjøretøys og sjåførs egenskaper. Dette er hverken hensiktsmessig eller mulig på grunn av ressursene. For at en mikrosimuleringsmodell bedre skal kunne gjenskape disse tilfeldighetene har de fleste modellene en «tilfeldighets-generator». På grunn av denne tilfeldighetsgeneratoren sier vi at de fleste modeller er stokastiske. I denne tilfeldighetsgeneratoren trekkes et sett med verdier som blant annet brukes til å generere forskjellige avvik på gitte parameterverdier. For å få så realistiske resultat som mulig ut fra modellen, bør disse settene med verdier endres ved hver simulering. Dette blir også kalt å endre «seeden» (Shaw & Noyce, 2015).

Til forskjell fra deterministiske modeller, hvor man vil få det samme resultatet ved hver beregning, vil resultatene fra stokastiske modeller variere fra hver beregning, også kalt simulering. Som modellbygger har man delvis kontroll på noen av tilfeldighetene, mens andre er satt av programleverandøren. Et eksempel der man kan ha innvirkning på tilfeldighetene er fart. Mange av parameterne trekkes ut fra en fordeling med et standardavvik, det vanligste er normalfordelingen. Fart er et eksempel på en parameter som trekkes ut fra en normalfordeling. Der fartsgrensen er 60 km/t oppgis 60 som en middelvei med for eksempel 5 som standardavvik. Andre kjøretøy- og adferds parametere som påvirkes av en sannsynlighetsfordeling, er blant annet sjåførens vilje til å benytte seg av tidsluker, kjøretøyets størrelse og egenskaper, samt førerens reaksjonsevne og aggressivitet. Tilfeldigheter man ikke kan påvirke er for eksempel når og hvilke kjøretøy som genereres inn i modellen, samt hvilke føreregenskaper som tilegnes hvilket kjøretøy.

Siden modellen er stokastisk vil inngangsdataene variere. Dette vil igjen føre til at resultatene varierer. Det er derfor viktig å kjøre flere simuleringer for å få gode statistiske resultat som representerer en gjennomsnittlig verdi. Antall repetisjoner pleier å ligge mellom 10-20 (Vejregelrådet, 2010). En detaljert metode for å beregne et minimum antall repetisjoner finnes i den danske mikrosimuleringshåndboken (Vejregelrådet, 2010).

Oppvarming av modellen er en annen ting man må ta i betraktning. Når simuleringen starter vil vegnettet være tomt. Det vil gå litt tid før kjøretøy har kommet seg inn i modellen, og representative data kan innhentes. Det er vanskelig å angi en nøyaktig oppvarmingsperiode, men som en tommelfingerregel bør et kjøretøy ha kjørt gjennom modellens lengste rute før man begynner å samle inn data (Hoogendoorn, 2003).

## **3.10 Kvalitetssikring av modellen**

Verifisering, kalibrering og validering er tre sentrale begreper innen trafikksimulering. Alle modeller bør ha gjennomgått en verifisering-, kalibrering- og valideringsprosess for å kontrollere at modellen er representativ for den situasjonen man prøver å gjenskape.

Verifisering er den første jobben man gjør for å «luke ut» de åpenbare feilene som oppstår i byggingen av en modell.

Kalibreringsprosessen er en interaktiv prosess, hvor man justerer modellens parametere slik at den best mulig klarer å gjenskape virkeligheten.

Validering er en målbar kontroll på om modellen klarer å gjenskape virkeligheten til et visst nivå.

For å utarbeide en realistisk modell, kan man først gjenskape den eksisterende situasjonen, også kalt 0-alternativet. Dette for å enklere kunne kalibrere tiltaket, med de erfaringene en gjør seg i 0-alternativet. Man bør da på forhånd ha kartlagt de eksisterende trafikkforhold så godt som mulig, for å kunne sammenligne 0-alternativet med disse.

### **3.10.1 Verifisering**

Verifiseringsarbeidet er ofte det første man gjør etter at man har bygd en modell. Dette ligger i grenseland mellom kvalitetssikring og modellbygging. Man arbeider her med å fjerne de mest opplagte og synlige feilene i en modell, som feilplassering av lenker, feil bruk av vikeplikt, ruter osv. Verifisering er en interaktiv prosess. Den vanligste måten å gjøre dette på er å kjøre en simulering og studerer animeringen. Feil som blir oppdaget blir korrigert. Denne prosessen gjentas til modellen er på et akseptabel nivå. I denne prosessen er det ikke vanlig å justere på kjøretøy eller adferdsparametere.

### **3.10.2 Kalibrering**

Etter at modellen er validert, begynner man med en «finjustering» av de forskjellige parametere. Denne prosessen kalles for kalibrering. Forskjellige veganelgg krever fokus på forskjellige typer parametere. Et eksempel på dette kan være å kalibrere parametere i forbindelse med en rundkjøring. En rundkjørings utforming, trafikkstrømmer og trafikkmengde vil variere fra kryss til kryss, og parameterverdiene i en rundkjøring stemmer kanskje ikke med en annen. Her må man gjøre individuelle justeringer for å få rundkjøringene til å representere den virkelige situasjonen. Når et kryss eller lignende er ferdig kalibret er

vanskelig å avgjøre. Her må man bruke skjønn og erfaring. Man vil i valideringsprosessen få en indikasjon på om kalibreringsarbeidet var tilfredsstillende.

I en microsimuleringsmodell finnes det et mange av parametere og innstillinger. Det vil være utfordrende og arbeidskrevende å ha kontroll over alle parameterne i en slik modell. Det viktigste er å vite hvilke parametere som er de mest betydningsfulle, og dermed legge kalibreringsprosessen resurser i disse. Antall parametere man har kontroll over øker med erfaringen i bruk av mikrosimuleringsmodeller. Omfanget av kalibreringsprosessen kan ta mye av resursene avsatt i prosjektet om man ikke er strukturert (Vejregelrådet, 2010). De fleste mikrosimuleringsmodellene har en 2D-animering av modellen mens simuleringen kjører. Selv om ingen målbare data kan trekkes ut fra denne visningen, er den til god hjelp i en tidlig fase av kalibreringen. I datainnsamlingsfasen bør en hente inn data som kan sammenlignes med data fra modellen i kalibreringsfasen. Dette kan være kølengde, reisetid og forsinkelse

### **3.10.3 Validering**

Validering av en modell betyr å kontrollere at hele modellen klarer å gjenskape virkeligheten på en akseptabel måte, ikke bare det registrerte trafikkbilde. Til forskjell fra kalibrering som omfatter justering av isolerte parametere. Når man validerer en modell skal man benytte seg av målbare data og sammenligne disse data fra modellen mot virkeligheten. Disse dataene kan være kølengde, reisetid og forsinkelse. For å få en mest mulig korrekt modell skal valideringen utføres uavhengig av kalibreringen. Dette vil si at man ikke skal bruke de samme dataene som ved kalibreringsprosessen eller trafikkdataene som man brukte i oppbygging av modellen. Ved å bruke andre data til valideringen får man en mer robust modell. Data som skal brukes til validering av en modell skal ideelt sett innhentes etter at kalibreringen er ferdig. I de fleste mikrosimuleringsprogrammene kan man hente ut disse dataene. Det er da forholdsvis enkelt å sammenligne mot de registrerte dataene.

En utfordring i valideringsarbeidet vil være å vite hva som er et akseptabelt avvik mellom dataene fra modellen og registrerte data fra felt. Da det finnes få retningslinjer for bruk av mikrosimuleringsmodeller i Norge kan en mulighet være å bruke danske retningslinjer. Det danske veidirektoratet har fått utarbeidet en håndbok for bruk av mikrosimuleringsmodeller. Denne inneholder retningslinjer for validering av modeller og grenseverdier for avvik. Dersom avviket mellom registrerte og simulerte data er for store, må man gå tilbake i modellen å finne grunnen til dette.

En annen utfordring er å validere en modell med dårlige inngangsdata. Dersom trafikk tallene bygger på registrering av en rushperiode vil det være vanskelig å avgjøre om avviket mellom modell og virkelighet skyldes feil i parameterinnstillingene, eller om det skyldes at registreringene ble gjennomført ved et ikke-representativt trafikkbilde.

Etter at man har simulert 0-alternativet vil man ha gjort seg noen erfaringer ang. parameterverdier, nettverksoppbygging osv. Disse erfaringene og innstillingene tar man med seg videre i modellen hvor man skal simulere de nye tiltakene. Kalibreringen av parametere gjøres i samråd med de erfaringene man gjorde seg i 0-alternativet. Valideringen kan i mange tilfeller bli en utfordring da det kan være vanskelig å finne sammenlignbare elementer mellom modell og virkelighet. Her må man bruke erfaring og skjønn, og være åpen ovenfor modellbestiller hvilke betraktninger man har lagt til grunn i valideringsprosessen.

Skal man utføre en trafikksimulering over et område hvor man ikke finner eksisterende trafikk og målbare data, er man som regel overlatt til erfaringstall. Her må man bruke den erfaringen og kunnskapen man har opparbeidet seg gjennom tidligere simuleringsoppdrag. For eksempel vil kalibrering av parameterne i forbindelse med en rundkjøring være mulig ved å sammenligne en lignende rundkjøring man har simulert tidligere. Valideringen kan i noen slike oppdrag vise seg å være umulig. I slike tilfeller er det viktig å opplyse bestiller slik at han kan ta en vurdering rundt usikkerheten som er forbundet med modellen og resultatene.

### 3.11 Resultater

Resultatene fra mikrosimuleringsmodeller deles ofte opp i to kategorier; animerte resultater og numeriske resultater (FHWA, 2004).

Animerte resultater er ikke målbare, men kan i mange tilfeller gi planleggeren den informasjonen han trenger. Ved å studere animeringen kan man observere trafikken i nettverket og identifiser flaskehals, problemområder osv. Denne informasjonen er i noen tilfeller nok.

Numeriske resultater er målbare resultater som generes ut fra modellen i form av tabeller, grafer eller tekstfiler. Det er disse resultatene som sammenlignes, og som ender opp i rapporter. En mikrosimuleringsmodell simuler trafikkstrømmer ved å flytte kjøretøy gjennom et nettverk. Modellen gjør nye beregninger av kjøretøyets posisjon, konflikter og adferd flere ganger i sekundet. Alle kjøretøyene blir modellert samtidig og individuelt. Dette gjør at en mikrosimuleringsmodell er svært fleksibel når det kommer til uttak av resultater. Man kan velge å hente ut store rådatafiler eller definere databehovet mer konkret. Måten man henter inn data på, kan deles i tre kategorier; Punkt, strekning og nettverk.

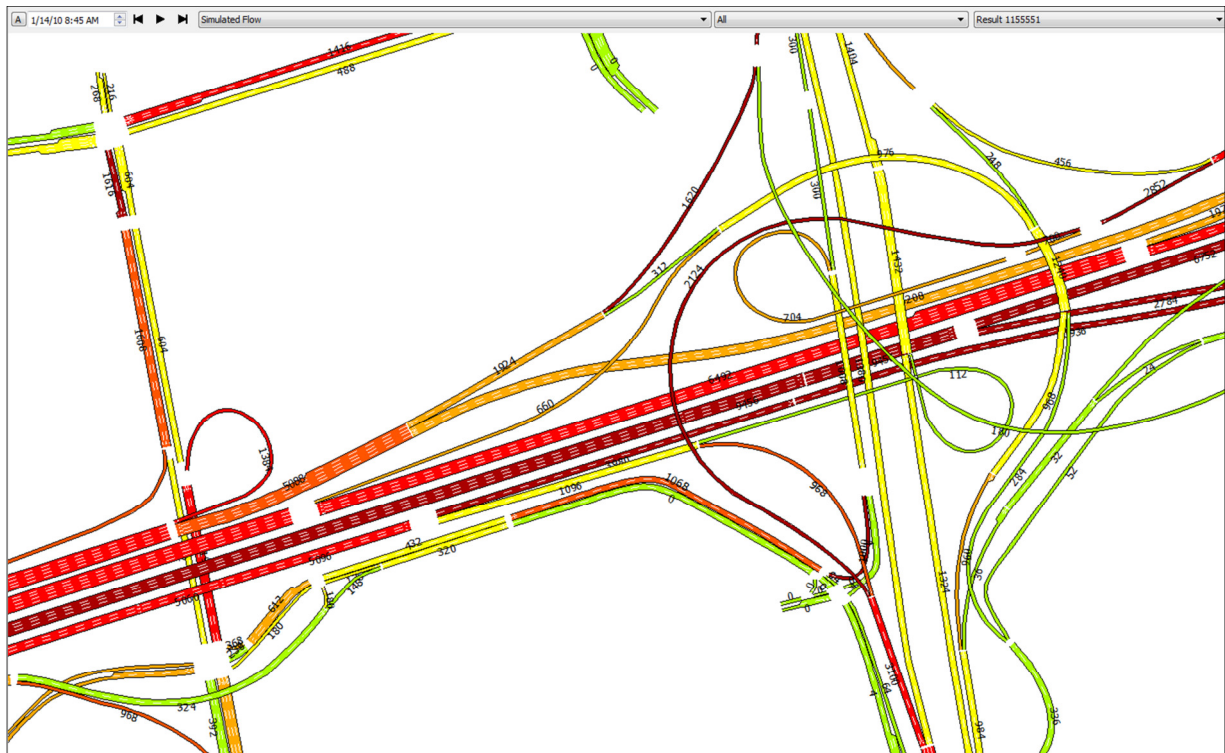
Punktresultater kan sammenlignes med resultater fra en detektor på et vegnett. Detektoren samler inn data for et gitt tverrsnitt av vegen. Her kan man måle punkthastighet, antall biler, antall biler med gitte egenskaper osv.

Når man skal hente inn strekingsdata er det vanlig å definere et start og et slutt punkt for registreringen. Her registres blant annet reisetid, forsinkelse, kølengder, gjennomsnittshastigheter osv. Registreringen kan gjøres for gitte tidspunkt eller for gitte kjøretøy.

Når man skal hente inn data fra et nettverk, menes det å hente inn resultater fra flere lenker. Dette kan være for et gatenett, kryss eller lignende. Nettverket defineres av et polygon eller rektangel, eller man kan definere hvilke lenker som skal være med i resultatet. Her har man ofte mulighet til å hente ut de sammen resultatene som ved strekning.

På grunn av den stokastiske utvelgelsen må man kjøre modellen flere ganger for å få et mest mulig statistisk korrekt resultat. Dess flere simuleringer dess mer reduseres usikkerheten. Resultatene blir som oftest presentert som en gjennomsnittsverdi.

Noen modeller angir i tillegg resultatene for hver simulering, min og maksverdi og standardavvik. Noen modeller, som for eksempel AIMSUN, viser de numeriske resultatene i nettverket. Mange synes at detter er en illustrativ måte å presentere resultatene på.



Figur 2-Eksempel på resultat fra AIMSUN

Count: 8	SimRun	TimeInt	VehicleTravelTimeMeasurement	Vehs(All)	TravTm(All)
1	2	600-3600	1: Strekning Nr.1	361	156.43
2	3	600-3600	1: Strekning Nr.1	361	156.43
3	4	600-3600	1: Strekning Nr.1	326	164.39
4	5	600-3600	1: Strekning Nr.1	344	168.15
5	Average	600-3600	1: Strekning Nr.1	348	161.35
6	Standard deviation	600-3600	1: Strekning Nr.1	17	5.88
7	Minimum	600-3600	1: Strekning Nr.1	326	156.43
8	Maximum	600-3600	1: Strekning Nr.1	361	168.15

Figur 3-Eksempel på resultat fra VISSIM

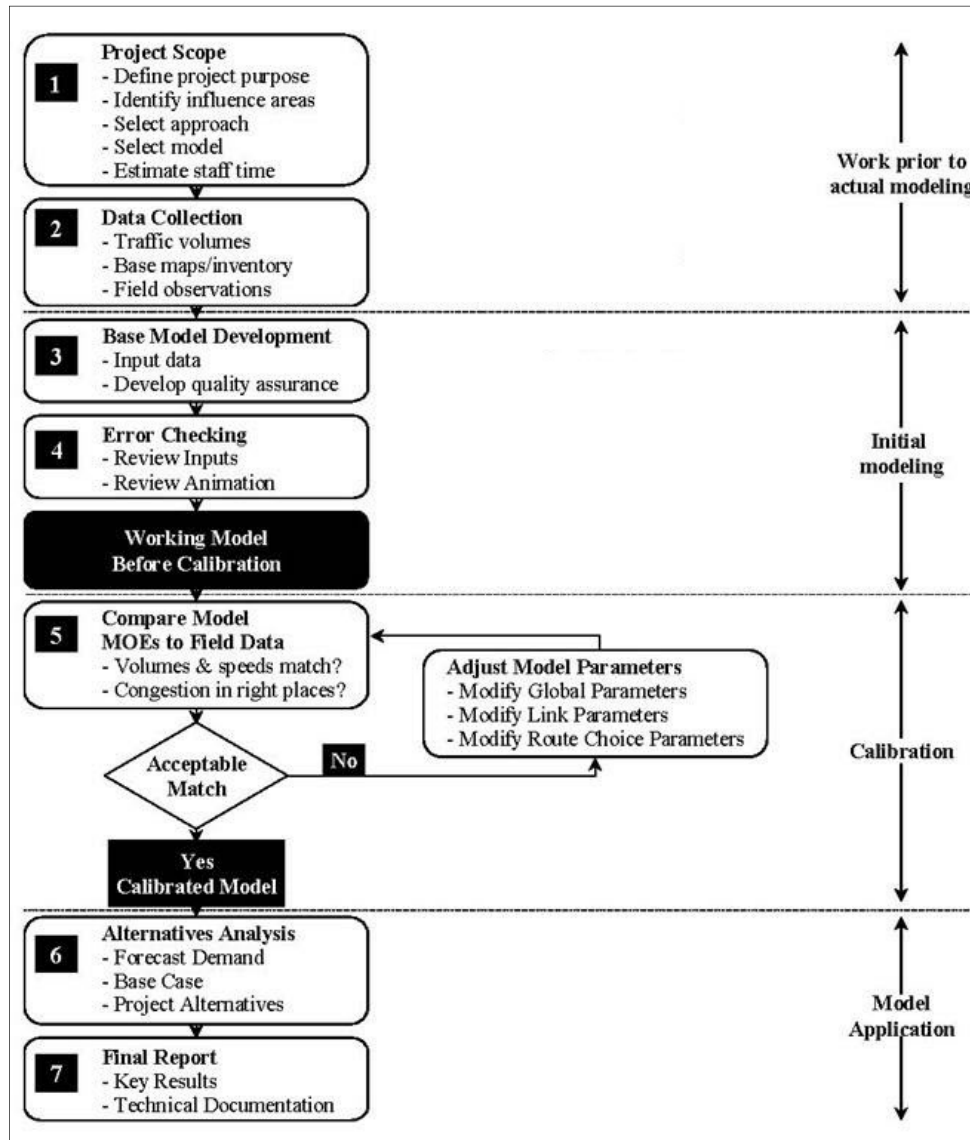
### 3.12 Metodikk

Et av målene med denne masteroppgaven var at jeg skulle opparbeide meg en forståelse av arbeidsmetodikken og prosessen ved å utføre et trafikksimuleringsoppdrag. I følge litteraturen er det flere måter å gjennomføre et oppdrag. For å summere, har jeg valgt å dele arbeidsmetodikken inn i 4 trinn.

- Trinn 1:  
Forberedende arbeider
- I dette trinnet settes rammene for prosjektet og oppgaven delfinnes, ofte sammen med modellbestiller. Her defineres hvilke data som ønskes hentet ut fra modellen.
- Her blir også det meste av grunnlagsdata som skal brukes i modellen hentet inn. For eksempel kartdata, trafikkdata, kjøretøydata og data til bruk for kalibrering.
- Trinn 2:  
Modelloppbygging
- I denne delen av prosessen foregår selve modelloppbyggingen. Det er her en trekker lenker, definerer kryss, legger trafikk på modellen osv. Etter at en har utarbeidet et modellutkast, starter en på verifiseringsarbeidet.
- Det er anbefalt å ha en fungerende modell uten for mange feil før en går videre til det neste trinnet.
- Trinn 3:  
Kvalitetssikring
- Kvalitetssikring av modellen er et viktig trinn i arbeidsprosessen. Det første en gjør er å kalibrere modellen slik at den klarer å gjenskape et realistisk trafikkbilde. Kalibreringen består i å justere parametere og sammenligne modellen med registrerte data fra trinn 1. Etter kalibreringen anbefales det at modellen valideres. I forkant av valideringen bør det hentes inn nye trafikkdata.
- Trinn 4:  
Avsluttende arbeider
- Avsluttende arbeider består i å hente ut de nødvendige dataene fra modellen, og utarbeide en rapport på bakgrunn av definert oppgave, hvor blant annet resultatene presenteres.



The Federal Highway Administration i USA har gitt ut en håndbok for bruk av mikrosimuleringsmodeller. Figur 4 som også beskriver denne arbeidsmetodikken, er hentet fra denne boken. Figuren er et godt eksempel på hvordan man bygger opp et mikrosimuleringsoppdrag.



Figur 4- Eksempel på en simuleringsprosess (FHWA)

# 4 ET INNBLIKK I VISSIM

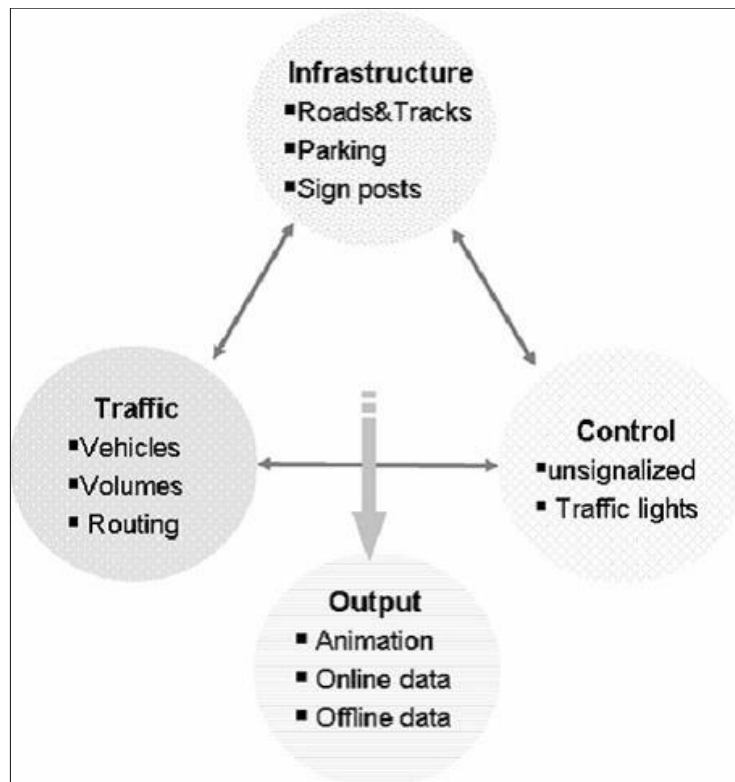
---

I denne oppgaven har jeg valgt å tilegne meg kunnskap om trafikksimulering ved hjelp av mikrosimuleringsmodellen VISSIM. VISSIM er et omfattende mikrosimuleringsprogram, og det vil i denne oppgaven ikke være anledning til å gi en grundig innføring av programmet. Jeg har i dette kapitlet prøvd å gi et overordna bilde av de mest sentrale elementene i programmet.

## 4.1 Generelt om VISSIM

VISSIM er en stokastisk mikrosimuleringsmodell og bygger på registrerte atferder ute i trafikken. Modellen ble påbegynt ved Universitet i Karlsruhe i 1978 og gjort kommersiell i 1994. VISSIM driftes og utvikles i dag av PTV, og er i dag et av de mest brukte mikrosimuleringsprogrammene i verden.

VISSIM er delt opp i 3 hovedblokker, i tillegg til en resultatblokk. Alle blokkene har påvirkning på hverandre, slik figur 5 viser (Fellendorf & Vortisch, 2010). Videre beskrives de forskjellige blokkene nærmere.



Figur 5-Hovdblokkene i VISSIM (PTV)

## 4.2 Infrastruktur

Å bygge nettverket er ofte det første man gjør når man skal gjennomføre en mikrosimulering. I blokken som kalles infrastruktur beskriver man de fysiske aspektene i modellen.

### 4.2.1 Veggen

Vegnettet bygges opp gjennom å trekke «links» og «connectors» oppå et kart eller en DWG-fil. Links representerer selve veggen i modellen. Man angir i dette elementet vegens fysiske egenskaper og kjøreradferd. For eksempel kan man angi at dette elementet er planlagt i et urbant område med påfølgende adferd og fartsgrense. Disse atferdstypene vil bli bedre forklart senere i dette kapitlet. Her beskrives også de fysiske egenskapene som blant annet stigning, antall felt og feltbredde. Man kan også definere hvilke kjøretøy som kan benytte seg av hvilke felt og hvilke feltskifter som er tillat.

Connectors er et element som binder links sammen med hverandre, og har de samme egenskapene som links. I tillegg har det den egenskapen at det kan påvirke feltskifte i modellen dersom man bruker statiske ruter. Man definerer når et kjøretøy skal begynne å planlegge feltskifte og hvor emergency stop er, relativt til den aktuelle connectoren. Ved å plassere en såkalt «dummy connector» oppå en link kan man angi at en rute må innom denne connectoren, og man har da anledning til å styre rutene med tanke på feltbruk og feltskifte, se figur 6. Både links og connectors kan modelleres med kurver slik at de stemmer med virkeligheten.



Figur 6-Eksempel på «dummy connector»

### 4.2.2 Andre elementer

Selv om links og connector er de viktigste elementene i infrastrukturblokken, inneholder den også andre elementer som er viktige for modellen, som skilt, signalhoder og tilhørende

stopplinjer. Ved bruk av kjøretøystyrte signalanlegg bruker man ofte å plassere ut detektorer med forskjellige egenskaper.

VISSIM har utviklet en funksjon som heter «reduced speed area». Dette elementet legges oppå eksisterende links eller connectors og «bremser» kjøretøyene. Denne funksjonen brukes der man vil redusere farten uten å endre fartsgrense. Reduced speed area brukes blant annet i krappe kurver, kryss, tilfarter og sirkulasjonsareal.

I mange urbane områder er kantparkering en viktig del av trafikkbildet. Dette påvirker adferden og kapasiteten. I infrastrukturblokken finnes det funksjoner som lar deg beskrive dette.

Programmet er meget fleksibelt når det kommer til modellering av kollektivtrafikk, enten buss eller trikk. For at kollektivtrafikken kan stoppe trenger de en holdeplass. Kollektivstoppene kan modelleres som busslommer eller som kantsopp. Man kan bestemme hvilke bussruter som stopper på den aktuelle holdeplassen og hvor lenge. Lengden på stoppet kan enten defineres med sekunder, eller programmet kan beregne tiden utfra et antall passasjerer som går av eller på.

## **4.3 Trafikk**

Trafikkblokken er den mest omfattende av blokkene i VISSIM og beskriver egenskapene hos sjåfør og kjøretøy.

### **4.3.1 Kjøreruter**

Som nevnt i kapittel 3.4, kan man også i VISSIM velge om man vil bruke statiske eller dynamiske ruter for å beskrive kjøretøyets veg gjennom modellen. Dette gjøres ved å definere et startpunkt å trekke ruten over de aktuelle links og gjennom kryss. For at kjøretøyene skal få planlegge, og mulighet til å gjennomføre feltskifte før «emergency stop» er det en fordel at rutene begynne så langt borte fra «emergency stop» som mulig. Det angis også hvor mange kjøretøy som skal benytte seg av ruten.

Dersom man vil bruke dynamiske ruter må man gå til anskaffelse av en ekstra modul som heter «dynamic route assignment».

På samme måte som med statiske ruter trekker man bussrutene gjennom modellen, enten på det ordinære vegnettet eller på egne kollektivfelt. For hver rute kan man tilegne en eller flere rutetabeller. Man kan angi hvilken hatighetsfordeling bussen skal ha, og hvor presis den skal være, utfra et standardavvik.

### 4.3.2 VISSIMS atferdstyper

I VISSIM er det predefinert fem atferdstyper: Urban, Right side, Freeway, Footpath og Cycle track – se figur 7. Disse inneholder igjen ulike adferdsmodeller som Car-following og Lane-change. Man står fritt til å lage egendefinerte atferdstyper med egendefinert adferdsmodeller. Den enkleste måten er å kopiere en eksisterende adferdsmodell, gi den et unikt navn og gjøre de nødvendige endringene på denne.

Count: 5	No	Name	ObsrvdVehs	StandDistIsFix	StandDist	CarFollowModType
1	1	Urban (motorized)	4	<input type="checkbox"/>	0.50	Wiedemann 74
2	2	Right-side rule (motorized)	2	<input type="checkbox"/>	0.50	Wiedemann 99
3	3	Freeway (free lane selection)	2	<input type="checkbox"/>	0.50	Wiedemann 99
4	4	Footpath (no interaction)	2	<input type="checkbox"/>	0.50	No interaction
▶ 5	5	Cycle-Track (free overtaking)	2	<input type="checkbox"/>	0.50	Wiedemann 99

Figur 7-Predfinetere atferdstyper

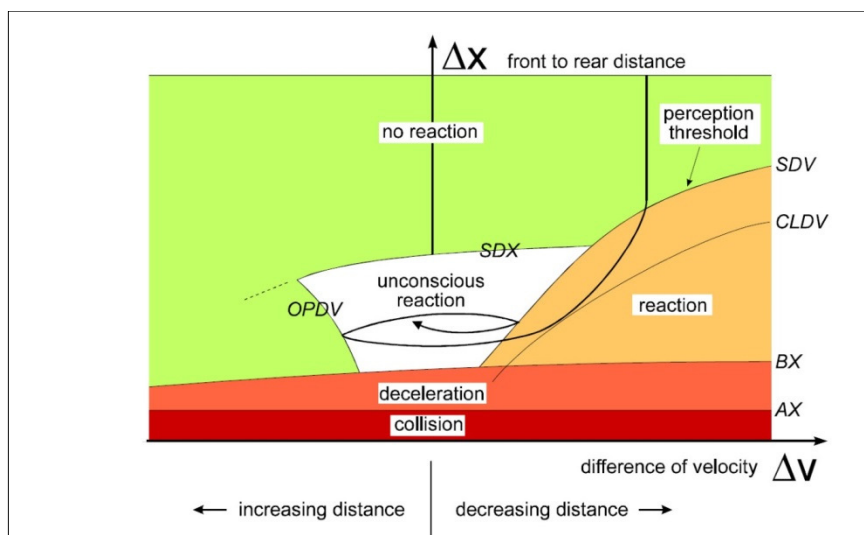
### 4.3.3 VISSIMs Car-following modell

VISSIMs Car-following modell bygger på den tyske trafikkforskeren Rainer Wiedemann sin psykofysiske adferdsmodell. Slike Car-following modeller beskrives nærmere i kapittel 3. De fleste sjåførere oppfører seg annerledes i et bybilde enn ved kjøring på motorveg. I VISSIM kan man velge mellom to forskjellige Car-following modeller, Wiedemann 74 og 99. Wiedemann 74 er en modell som er tilpasset for kjøring i urbane områder, mens Wiedemann 99 er tilpasset kjøring på motorveg.

Som nevnt i kapittel 3 består en psykofysisk Car-following modell av soner som avgrenses av terskler. Disse sonene representerer en type kjøremønster. VISSIMs Car-following modell består av 4 kjøremønstre:

- Fri kjøring
- Innhentning
- Følge
- Bremsing

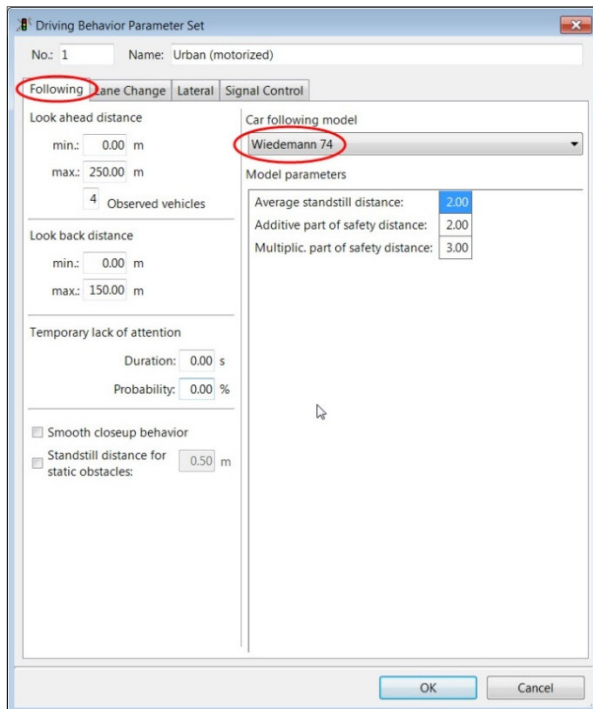
Når disse kjøremønstrene inntreffer avhenger av verdien på tersklene. Man bør være forsiktig med å endre for mye på terskelverdiene, da disse bygger på omfattende forskning og undersøkelser. Det kreves en viss erfaring for å vite hvordan disse verdiene henger sammen med andre parametere, og hvordan de påvirker modellen. I de fleste tilfeller brukes default verdiene.



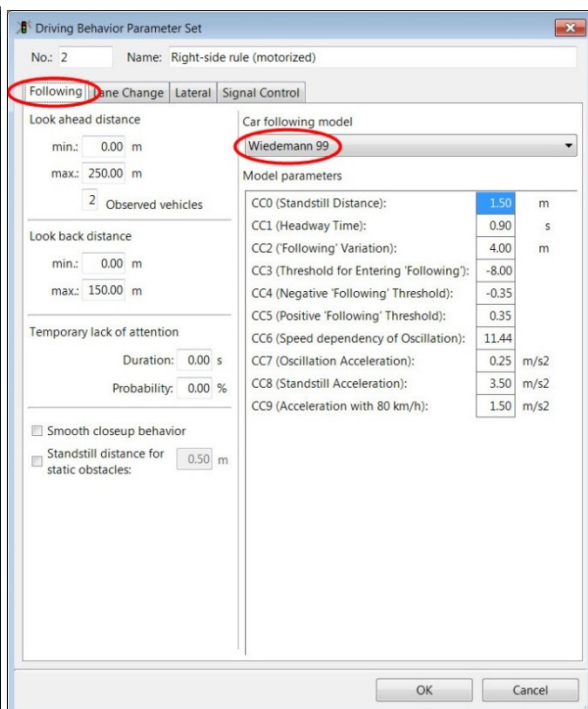
Figur 8-Illustrasjon på VISSIM's Car-following modell

En psykofysisk Car-following modell blir ofte fremstilt som figur 8. Avstanden mellom to kjøretøy vises på Y-aksen og hastighetsforskjellen på X-aksen. Som man kan se utfra figuren er den delt opp i 5 soner. Hver sone representerer et av de forskjellige kjøremønstrene som ble nevnt tidligere. Rød sone representerer kollisjon og defineres ikke som et kjøremønster. Tersklene er beskrevet med forkortelser. Under er det gitt en forklaring på disse forkortelsene.

- SDV** Er terskelen hvor et kjøretøy klarer å registrere at det nærmer seg et framforliggende kjøretøy med lavere hastighet. Når kjøretøyet krysser denne tersklene kommer det inn i sonen for innhenting. Her reagerer kjøretøyet med å retardere.
- SDX** Denne terskelen markerer når et kjøretøy går inn i følgeomodus. Området under denne terskelen kan beskrives som et følgeområde. Her vil kjøretøyet strebe etter å oppnå samme hastighet som kjøretøyet fremfor seg.
- OPDV** Beskriver når et kjøretøy går ut av følgeomodus som følge av at kjøretøyet foran akselerer.
- (A)BX** Minimumsavstanden mellom to kjøretøy ved lav hastighet og liten avstand.
- AX** Minimumsavstanden mellom kjøretøy som står i stillestående kø.



Figur 9-Wiedeman74



Figur 10-Wiedeman99

Figur 9 og 10 viser dialogboksen for innstillingene i VISSIMs Car-following modell, Wiedemann 74 og 99. Parameteren for å endre terskelverdier finner man i dialogboksens høyre kolonne. Som man ser utfra figurene har man mindre muligheter til å endre på terskelverdiene i Wiedemann 74 enn 99.

#### 4.3.4 VISSIMs Lane-change modell

Lane-Change-modellen er en annen viktig adferdsmodell som er plassert i Trafikkblokken. Teorien bak Lane-change-modeller er nærmere beskrevet i kapitel 3.

I VISSIM er og beslutningen om feltskifte basert på nødvendig og ønskelig feltskifte. Med nødvendig feltskifte menes at et kjøretøy må skifte felt for å rekke neste «start» på en rute. Nødvendig feltskifte styres av «emergency stop»- verdiene satt i connector elementet.

Ønskelig feltskifte er basert på ønske om å bedre sin situasjon på vegen. Det kan være ønske om større fart eller større luker rundt seg. Et slikt feltskifte påvirkes av avstanden til kjøretøyet bak seg og sikkerhetsavstanden gitt i Car-following-modellen. Ved et ønsket feltskifte reduseres denne sikkerhetsavstanden. Som default er denne verdien 0.6, som igjen betyr at sikkerhetsavstanden ved et ønsket feltskifte reduseres med 40%. Det finnes få parametere man kan endre på i forbindelse med ønsket feltskifte, og det kan virke som de fleste betingelser ved ønsket feltskifte er gitt av programutvikler.



I kapitel 3 beskrev jeg viktigheten av at en Lane-change-modell klarer å beskrive de menneskelige aspektene med et feltskifte. VISSIMs Lane-change-modeller er i utgangspunktet en regelbasert modell. I tillegg har programutviklerne implementert menneskelig adferd i modellen, ved at bilene samarbeider ved feltskifte og at biler kan tvinge seg inn i feltet. Man kan styre i hvilken grad disse adferdene skal påvirke et feltskifte.

De viktigste parameterne i VISSIMs Lane-change-modell er forbundet med sikkerhetsavstander og førerens aggressivitet i form av akselerasjon og retardasjon.

Driving Behavior Parameter Set

No.: 2 Name: Right-side rule (motorized)

Following | **Lane Change** | Lateral | Signal Control

General behavior: Slow lane rule

Necessary lane change (route)	Own	Trailing vehicle
Maximum deceleration:	-4.00 m/s <sup>2</sup>	-3.00 m/s <sup>2</sup>
- 1 m/s <sup>2</sup> per distance:	200.00 m	200.00 m
Accepted deceleration:	-1.00 m/s <sup>2</sup>	-0.50 m/s <sup>2</sup>

Waiting time before diffusion: 60.00 s

Min. headway (front/rear): 0.50 m

To slower lane if collision time is above: 11.00 s

Safety distance reduction factor: 0.60

Maximum deceleration for cooperative braking: -3.00 m/s<sup>2</sup>

Overtake reduced speed areas

Advanced merging

Consider subsequent static routing decisions

Cooperative lane change

Maximum speed difference: 10.80 km/h

Maximum collision time: 10.00 s

Lateral correction of rear end position

Maximum speed: 3.00 km/h

Active during time period from 1.00 s until 10.00 s after lane change start

OK Cancel

Figur 11-Dialogboks, Lane-Change

Som man kan se utfra figur 11 finnes det parametere som er spesielt knyttet til nødvendig feltskifte, necessary lane change. Disse parameterne beskriver førerens aggressivitet og dens villighet til å tvinge seg inn i nabofeltet. Den beskriver også villigheten til kjøretøyet som påvirkes å godta at et kjøretøy tvinger seg inn.



Under beskrives noen av de forskjellige parameterne som styrer et feltskifte:

Maximum deceleration	Maksimal retardasjon et kjøretøy er villig til å godta ved et feltskifte.
-1m/s <sup>2</sup> per distance	Beskriver hvilken retardasjon kjøretøyet er villig til å bruke ettersom det nærmer seg «emergency stop».
Accepted deceleration	Minimumsverdien for retardasjon, når et kjøretøy skal utføre feltskifte.
Waiting time before diffusion	Maksimumstiden et kjøretøy kan stå og vente ved «emergency stop» før det blir fjernet fra modellen.
Min. headway (front/rear):	Minste avstand mellom to biler som må være tilgjengelig etter feltskifter er gjennomført. Denne verdien må være oppfylt før feltskifte kan gjennomføres.
To slower lane if collision time is above	Er kun relevant hvis «Right side rule» er valgt som kjøreadferd. Parameteren beskriver avstanden i sekunder som et forbikjørende kjøretøy må ha til kjøretøyet framfor seg, før det legger seg tilbake i venstre fil.
Safety distance reduction factor:	Brukes til å redusere sikkerhetsavstanden gitt i Car-following-modellen, ved feltskifte.
Maximum deceleration for cooperative braking	Villigheten et kjøretøy har for å slippe inn. Representert ved den maksimale retardasjonen et kjøretøy vil tillate for å slippe inn ett kjøretøy.

#### 4.3.5 Kjøretøysammensetning

I VISSIM har man mulighet til å lage den kjøretøysammensetningen man ønsker, og tilegne egenskaper til disse.

Kjøretøyene genereres ut i modellen fra noder i modellens grensesnitt. I noden oppgis det hvilken kjøretøykomposisjon som skal genereres ut. En node kan bestå av flere komposisjoner som kan distribueres ut ved forskjellige tider. En komposisjon består av de aktuelle

kjøretøyklassene og % -andelen av disse. For eksempel kan en komposisjon bestå av 2% motorsykler, 89% lette-, og 9% tungekjøretøy.

En kjøretøyklasse kan bestå av en eller flere kjøretøysammensetninger. En kjøretøysammensetning består igjen av et eller flere typer kjøretøy. Disse kjøretøyene er beskrevet med vekt, akselerasjon, retardasjon, motorkraft, akselavstand, osv. Man kan velge kjøretøy utfra VISSIMs bibliotek eller man kan definere egne kjøretøy. Man kan også definere % -andel av en type kjøretøy i en kjøretøysammensetning.

Til sammenligning deler vi i Norge trafikken inn i to kjøretøyklasser, lette og tunge kjøretøy. Lette kjøretøy sin kjøretøysammensetning består av alt fra store SUV'er til små elbiler. Mens tunge kjøretøy kan være alt fra små lastebiler til vogntog.

### **4.3.6 Trafikkgenerering**

Trafikken genereres ut i modellen ved hjelp av «vehicle input» funksjonen. Fra noder, i modellens grensesnitt, generes trafikken ut i modellen. I nodens attributter defineres trafikkvolumet og kjøretøykomposisjonen som er definert på forhånd. Dersom en bruker OD matrise, knyttes denne til de aktuelle nodene.

Hvordan denne noden generer ut kjøretøy, med tanke på størrelse og føreregenskaper, er tilfeldigheter som er gitt av programleverandøren.

## **4.4 Kontroll**

Det fins flere måter å kontrollere og prioritere trafikkstrømmer på i VISSIM. Det gjøres ved hjelp av trafikkskilt som vikeplikt og stoppskilt, hvor en definerer primær og sekundær trafikkstrømmer, samt middelvei på aksepterte tidsluker. En annen måte å kontrollere trafikken på, er ved hjelp signalanlegg.

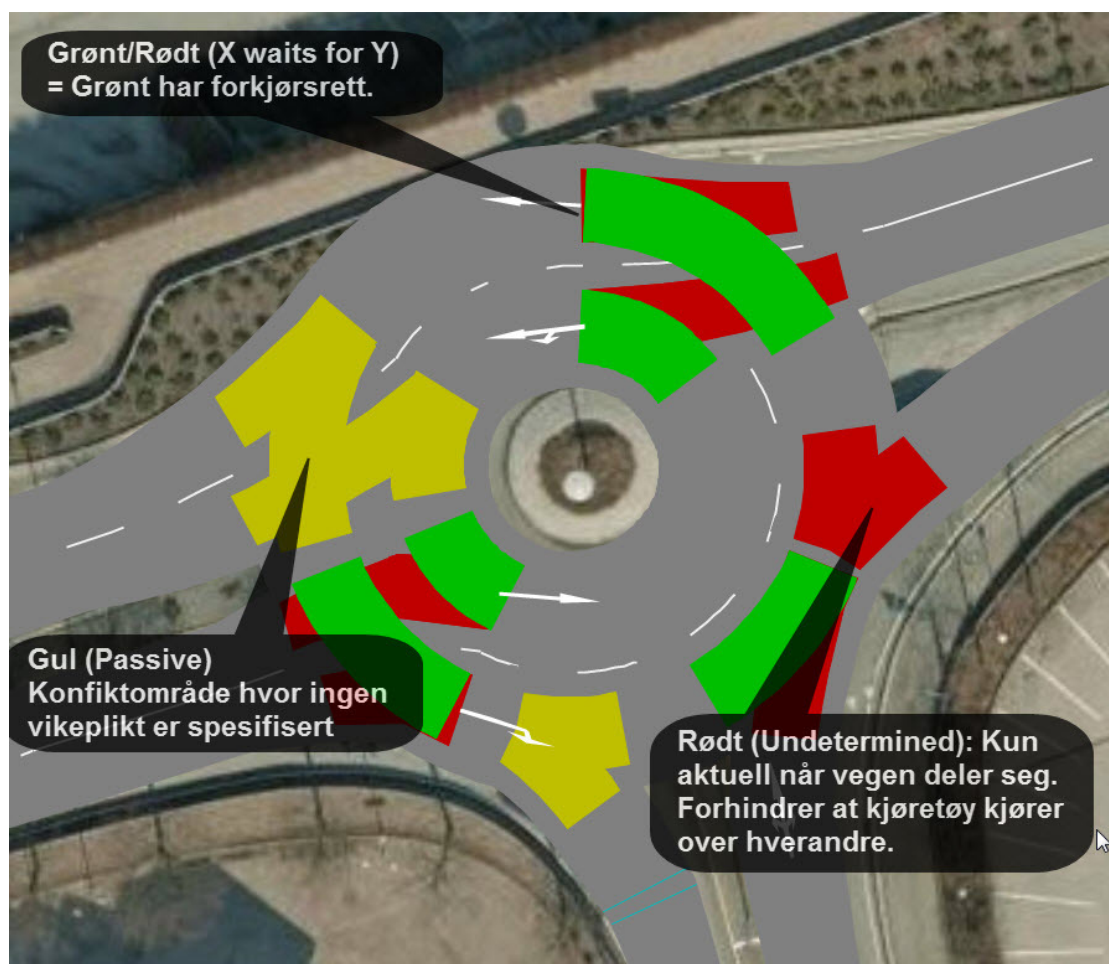
### **4.4.1 Signalanlegg**

Signalanlegg modelleres ved at det settes ut signalhoder som igjen knyttes til en signalgruppe. Signalgruppen styres igjen av en faseplan. Faseplanen må utarbeides på forhånd da VISSIM ikke hjelper deg med dette, i motsetning til for eksempel SIDRA. Det finnes to metoder for å modellere et signalanlegg på. Tidsstyrt eller kjøretøystyrt. Tidsstyrt er den enkleste formen for å modellere signalanlegg, hvor fasene skifter med faste tidsintervall. For å modellere kjøretøystyrte signalanlegg kreves en egen tilleggsmodul. Disse anleggene er svært fleksible og gir mange muligheter med tanke på modellering av ITS-tiltak. Ulempen med kjøretøystyrte

anlegg er at de har et høyt brukersnitt og er resurskrevende å utarbeide. Kjøretøystyrte signalanlegg vil bli bedre forklart i kapittel 8.

#### 4.4.2 VISSIMS gap acceptance-modell

VISSIMS gap acceptance-modell brukes i konfliktområder der biler må løse konflikter. I VISSIM kan man også definere den tradisjonelle vikepliktsituasjonen der man setter ut en stopplinje og konfliktmarkør, angir hvem den skal vike for og minimums gap time (sek) og headway(m). I siste versjon, VISSIM 7, har programmet fått en forbedret utgave av verktøyet «Conflict area». Dette er en metode som programutviklerne anbefaler framfor den tradisjonelle metoden, dersom man skal modellere vikepliktsituasjoner. Den er mer intelligent og beskriver en mer rasjonell oppførsel (Vejregelrådet, 2010). Verktøyet er blant annet til stor hjelp når man skal modellere rundkjøringer, da det i et slikt kryss oppstår flere konflikter. En av hovedfordelene med dette verktøyet, er at det avdekker alle konfliktpunkt automatisk.



Figur 12-Illustrasjon, Conflict area

Count	Link1	VisibLink1	Link2	VisibLink2	Status	FrontGapDef	RearGapDef	SafDistFactDef	AddStopDist	ObsAdjLns	AnticipRout	AvoidBlock
38	10063	100.0	104202: FV446	100.0	Passive	0.5	0.5	1.5	0.0	<input type="checkbox"/>	0.0	100.0
39	32	100.0	10024	100.0	2 waits for 1	0.5	1.4	1.0	0.0	<input type="checkbox"/>	0.0	0.0
40	32	100.0	10066	100.0	Undetermined	0.5	0.5	1.5	0.0	<input type="checkbox"/>	0.0	100.0
41	32	100.0	10090	100.0	2 waits for 1	0.5	0.5	1.5	0.0	<input type="checkbox"/>	0.0	100.0
42	40	100.0	10057	100.0	2 waits for 1	0.5	0.5	1.5	0.0	<input type="checkbox"/>	0.0	100.0
43	10062	100.0	104202: FV446	100.0	Passive	0.5	0.5	1.5	0.0	<input type="checkbox"/>	0.0	100.0
44	10064	100.0	10065	100.0	Undetermined	0.5	0.5	1.5	0.0	<input type="checkbox"/>	0.0	100.0

Figur 13-Parametertabell, Conflict area

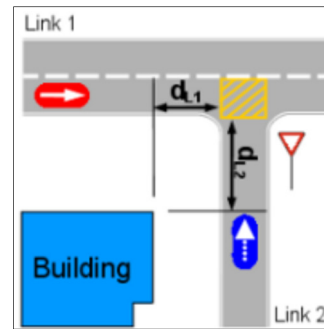
Conflict area en oversiktlig metode for å beskrive konfliktene i for eksempel en rundkjøring, men er forholdsvis krevende å kalibrere, da parameterne i et konfliktområde kan påvirke utfallet av et annet konfliktområde.

I VISSIM har man 3 typer konflikter; når linker krysser, splittes eller slås sammen. VISSIM oppdager disse konfliktene og merker dem med gult, også kalt «passive». Når man har fått definert et område der to trafikkstrømmer vil møtes, må en beskrive hvordan konflikten løses dersom 2 biler vil inn i konfliktområdet på samme tid. Som man ser på figur 12 kan et konfliktområde beskrives på 3 måter, passive, undetermined eller X waits for Y. Når man velger at et konfliktområde skal defineres som X waits for Y, betyr det at man har en sekundærstrøm som må vike for en primerstrøm. Her registrerer et kjøretøy fra sekundærstrømmen kjøretøy med forkjørsrett, og foretar en beslutning om konfliktområdet kan forseres utfra gitte parameterverdier. I tillegg betraktes også situasjonen etter at kjøretøyet har forsert konfliktområdet, det kan oppstå nye konflikter som påvirker beslutningen om å forser konfliktområdet.

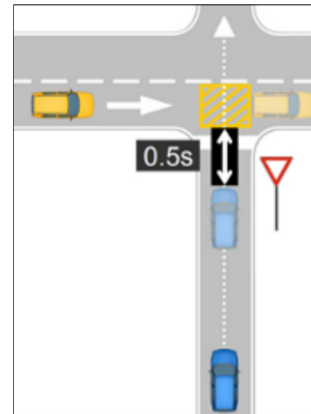
Til forskjell fra vanlig vikepliktsmodellering, kan også primærstrømmen påvirkes av sekundærstrømmen. Hvis for eksempel et vikende kjøretøy ikke klarer å komme seg tidsnok ut av konfliktområdet, vil kjøretøyet med forkjørsrett bremse for å unngå kollisjon. Alle kjøretøy vil i utgangspunktet unngå å kjøre inn i et konfliktområde dersom det vil bli stående der på grunn av tilbakeblokkering i kryss eller lignende (Vejregelrådet, 2010).

Et vikende kjøretøys betraktninger gjøres på grunnlag av gitte parameterverdier. På figur 13 kan en se dialogboksen med de forskjellige parameterne. Videre beskrives de mer inngående.

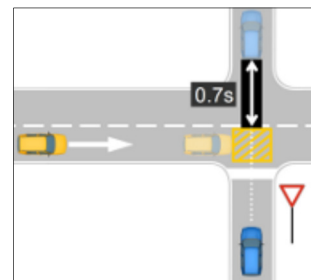
Visibility Beskriver sikten en veg har til en annen i kryssområde.



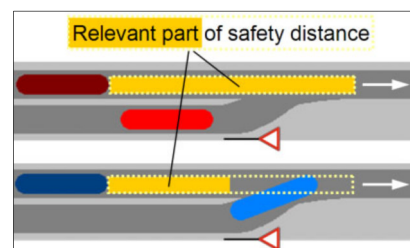
Front Gap Beskriver minste luke i sekunder, når et vikende kjøretøy kan ankomme konfliktområdet etter at kjøretøyet med forkjørsrett har forlatt det.



Rear Gap Minste luke i sekunder, når et kjøretøy med forkjørsrett kan ankomme konfliktområdet, etter at kjøretøyet uten forkjørsrett har forlatt det.



Safety distance factor. En sikkerhetsfaktor i forbindelse med fletting for å sikre en minimums distanse etter at kjøretøyet har kommet inn i trafikkstrømmen. Verdien multipliseres med den opprinnelige sikkerhetsavstanden i Car-following modellen.



Additional stop distance. Beskriver avstand i meter fra en stopplinje til konfliktområde.

Anticipate routes En prosentvis andel av vikende kjøretøy som vil kunne forutse kjøretøy i

primærstrømmen som ikke vil komme inn i konfliktområdet.

Avoid blocking En prosentvis andel av bilene som vil unngå å kjøre inn i konfliktområdet, da de ikke vil kunne kjøre videre uten å så blokkere en annen trafikkstrøm.

## 4.5 Resultat og output

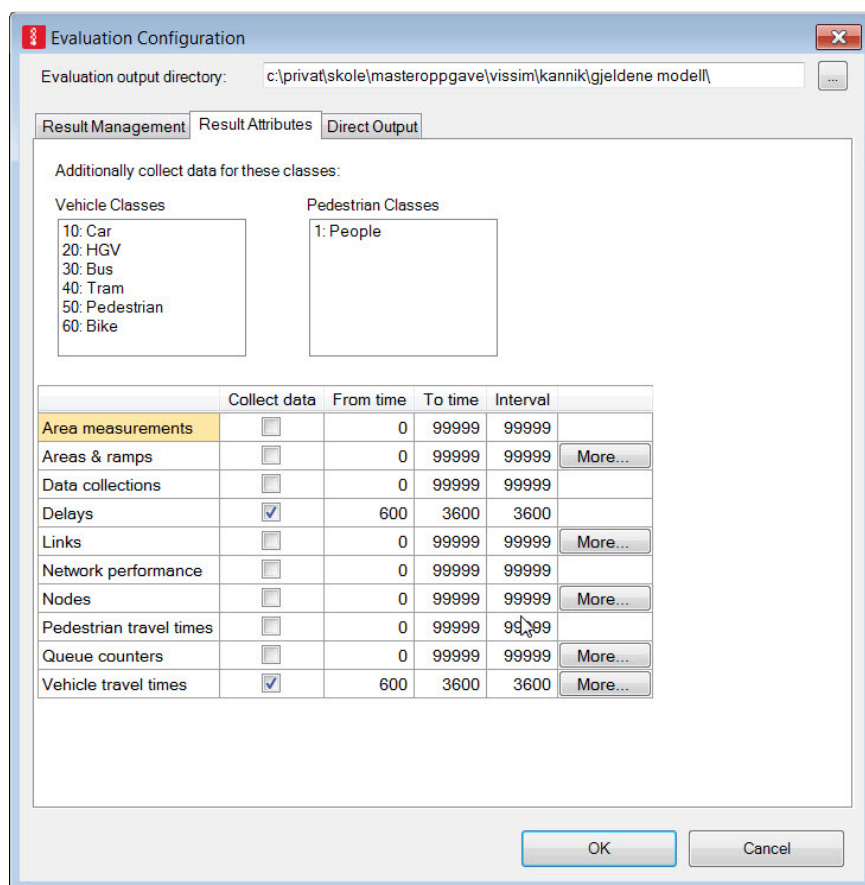
Definisjonen av en mikromodell er at den kontinuerlig beregner hvert kjøretøys posisjon gjennom et nettverk. Dette gir stor fleksibilitet når det kommer til datainnsamling. Utdataene kan deles i 3 kategorier; offline data, online data og animering

Offline data er de konkrete og målbare dataen som blir registret i filer og tabeller. VISSIM har tradisjonelt sett vært svak på dette området, da resultatene i tidligere versjoner har kommet som rådata i tekstfiler. I versjon 6 av programmet hadde dette bedret seg litt, da man fikk litt mer oversiktlige tabeller. Nå i den siste versjonen av programmet, versjon 7 kan man også generere grafer. Jeg mener likevel at denne delen av programmet kunne vært mer brukervennlig.

Man kan enda velge mellom å få ut store detaljert rådatafiler, eller velge å definere det bedre. Det første man gjør er å definere et område, punkt eller en strekning der dataene skal innhentes. Til dette har man forskjellige verktøy:

Nodes	Med node definerer man et kryssområde ved hjelp av et polygon. Man kan i etterkant velge hvilke data man vil hente ut av dette området.
Data collection point	Et registeringspunkt settes ut på en link eller connector.
Vehicle travel time	Ved å definere et start og slutt punkt kan man definere en strekning som skal registreres.
Queue counters	Elementet legges oppå en link eller connector og registrerer maks og gjennomsnittlig kølengde. Plasseres typisk i stopp og vikelinjer.

Etter at områdene for datainnsamling er definert må man definere hvilke data og tidsintervall som skal lagres, og lenke dette opp mot den aktuelle strekning, punkt eller område.



Figur 14-Dialogboks for datainnsamling

Figur 14 viser dialogboksen der man velger hvilke data man ønsker å få registret osv. Ved denne aktuelle simuleringen kan man se at man ønsker at alle typer kjøretøy skal registerets. Videre ser man at man har valgt å samle data om reisetid og forsinkelse.

I denne simuleringen har man valgt et registreringsintervall på 3600 sek, det samme som simuleringsperioden. I dette intervallet skal det registers data fra det har gått 600 sekunder ut i simuleringen, til simuleringens slutt ved 3600. Det betyr at det er lagt inn en oppvarmingsperiode på 600 sekunder.

Figur 15 er et eksempel på en resultattabell fra VISSIM. Tabellen viser resultatene for en reisetidsregistrering på en definert strekning. I «SimRun»-kolonnen vises det hvor mange simuleringer som er gjennomført, i dette eksempelet er det gjennomført 3 simuleringer. Videre i kolonnen ser man gjennomsnitt, standardavvik, maks og minimum for disse simuleringene.

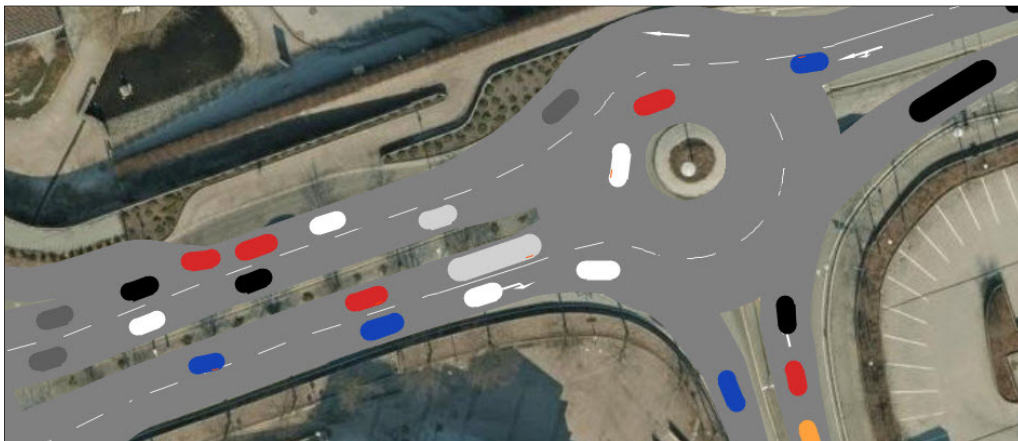


Verdien til disse finner man til høyre i tilhørende rad. Man kan for eksempel lese ut fra rad fire at den gjennomsnittlige reisetiden er 221.24 sekunder. Videre kan man lese at datainnsamlingen startet 600 sekunder etter simuleringstart og stoppet 3600 sekunder etter simuleringstart. Strekingen for datainnsamling var rute 2: Bergeland mot E39 og 342 biler kjørte denne ruten.

Count: 7	SimRun	TimeInt	VehicleTravelTimeMeasurement	Vehs(All)	TravTm(All)
▶ 1	1	600-3600	2: Bergeland mot E39	333	226.99
2	2	600-3600	2: Bergeland mot E39	345	227.00
3	3	600-3600	2: Bergeland mot E39	347	209.73
4	Average	600-3600	2: Bergeland mot E39	342	221.24
5	Standard deviation	600-3600	2: Bergeland mot E39	8	9.97
6	Minimum	600-3600	2: Bergeland mot E39	333	209.73
7	Maximum	600-3600	2: Bergeland mot E39	347	227.00

*Figur 15-Offline data*

Den andre formen for output er online data. Der er dette de fleste forbinder med mikrosimulering, en 2D-animering av kjøretøy i et nettverk, se figur 16. Denne type data gir ingen målbare resultat, men kan være til stor hjelp i en verifisering- og kalibreringsprosess.



*Figur 16-Online data*

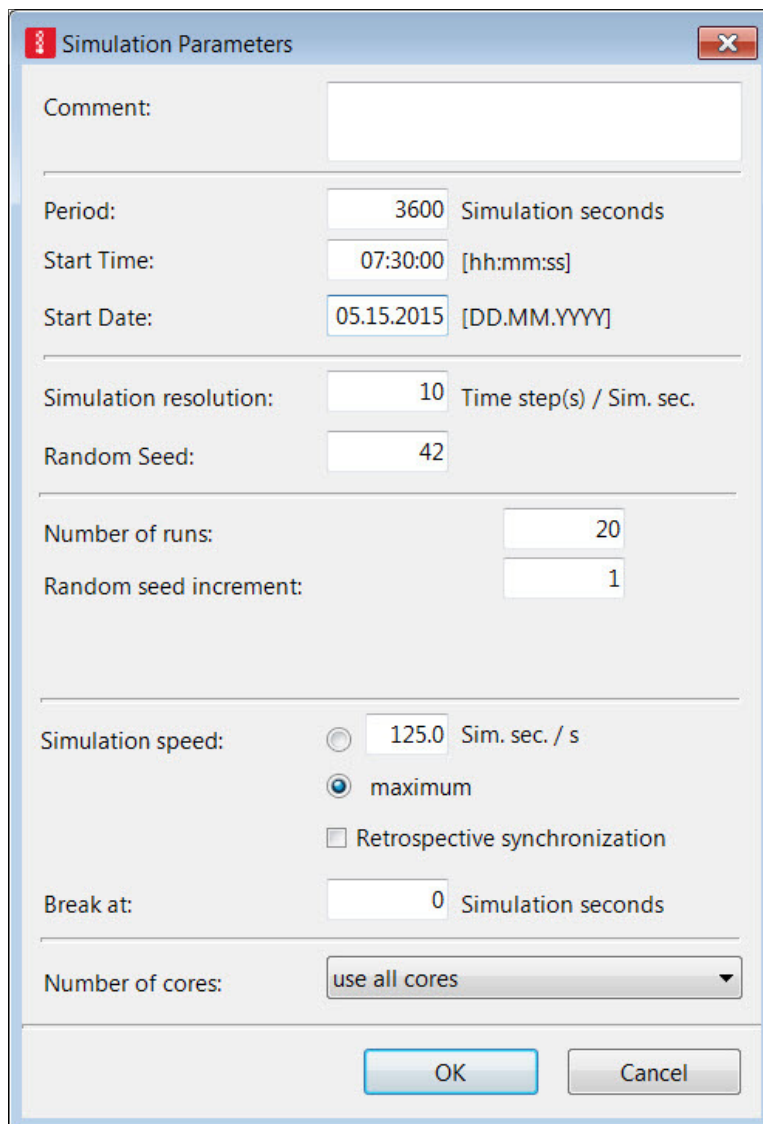
Den tredje formen for Output er 3D-animering, se figur 17. Dette er output beregnet for offentligheten og har ingen resultatmessig betydning



*Figur 17-3D-animering*

## 4.6 Simuleringsparametere

Simuleringsparametere er viktige, og kan ha stor betydning for resultatene.



The screenshot shows a dialog box titled "Simulation Parameters" with the following fields and options:

- Comment: (empty text box)
- Period: 3600 Simulation seconds
- Start Time: 07:30:00 [hh:mm:ss]
- Start Date: 05.15.2015 [DD.MM.YYYY]
- Simulation resolution: 10 Time step(s) / Sim. sec.
- Random Seed: 42
- Number of runs: 20
- Random seed increment: 1
- Simulation speed:  125.0 Sim. sec. / s,  maximum,  Retrospective synchronization
- Break at: 0 Simulation seconds
- Number of cores: use all cores (dropdown menu)
- Buttons: OK, Cancel

Figur 18-Dialogboks, Simuleringsparametere

Videre følger en kort beskrivelse av de viktigste simuleringsparameterne i VISSIM

Period	Lengden på simuleringsperioden i sekunder. Representerer ofte dimensjonerende time.
Simulation resolution	Antall ganger i sekundet programmet beregner kjøretøyets posisjon, valg og beslutninger.

Random Seed	Beskriver hvilke tall som skal genereres utfra tilfeldighetsgeneratoren, disse tallene skaper igjen de tilfeldighetene som gjør modellen stokastisk.
Number of runs	Hvor mange simuleringer som må gjennomføres for å få et statistisk holdbart resultat.
Random seed increment	Beskriver hvor mange mye tallet i Random seed skal økes med for å gjøre modellen stokastisk.

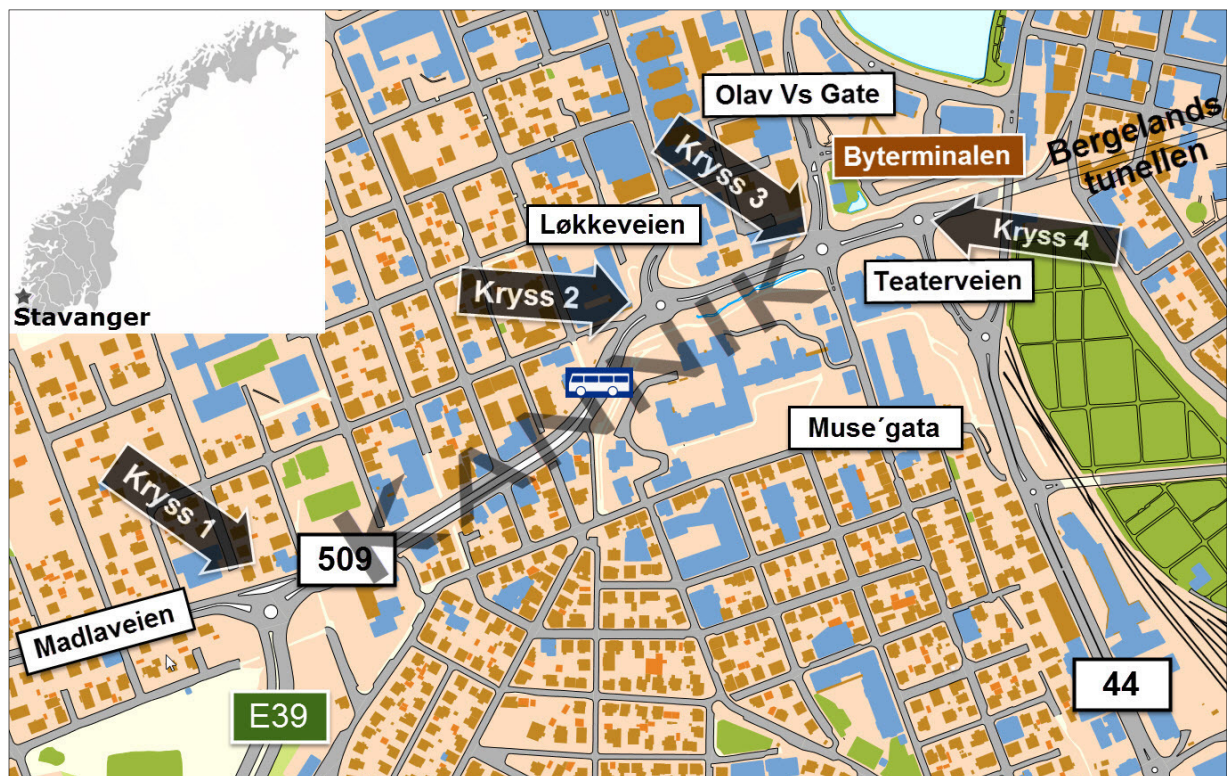
# 5 DAGENS SITUASJON

Som tidligere nevnt, er det en fordel å ha en realistisk case når man skal lære seg mikrosimulering. I dette kapitlet beskrives casen og dagens situasjon, også kalt 0-alternativet.

## 5.1 Bakgrunn for casen

Nord-Jæren er et av Norges raskest voksende områder. Som følge av dette opplever man i dag store avviklingsproblemer på vegnettet, spesielt i morgen- og ettermiddagsrushet. Kannik er av Stavanger Aftenblad regnet som en av de fem verste «flaskehalsene» på Nord-Jæren.

Kannik er et område i Stavanger sentrum, se figur 19. Vegen gjennom Kannik er et trafikknutepunkt og hovedatkomst til Stavangers bykjerne fra E-39, og fra de fleste omliggende bydelene.



Figur 19-Oversiktskart for Kannik

De fleste bussene som skal til og fra Byterminalen må gjennom Kannik. Da det ikke finnes noe kollektivfelt gjennom Kannik, må bussene stå i samme kø som resten av trafikken. Dette skaper forsinkelse og stor frustrasjon hos de som benytter seg av kollektivtilbudet. Det har



lenge vært ytret ønske om kollektivfelt gjennom Kannik, og det har i mange år vært diskutert hvilke kollektivløsning som vil være den beste for byen; bybane på skinner eller et nett av bussveger. Det har i det siste vært et politisk flertall for buss. Samferdselsdepartementet har også, gjennom sin handlingsplan for kollektivtrafikk, gitt støtte til Rogaland fylkeskommunes prosjekt «Bussvei 2020». Målet for Bussvei 2020 er at bussen skal kjøre på egne felt, uhindret av bilkø, samt prioriteres i kryss hvor bil- og kollektivtrafikk krysser.

På bakgrunn av at «Bussvei 2020» er blitt valgt som det fremtidige kollektivtilbudet i Stavangerregionen, er det også naturlig at en slik løsning må implementeres i Kannik. Statens vegvesen har blitt bedt av Stavanger kommune om å utrede konsekvensene en slik utbygging vil medføre. Dette er utgangspunktet for denne casen. Konsekvensene har jeg i denne oppgaven valgt å definere som økning i forsinkelse på definerte strekninger. I samarbeid med Statens vegvesen har jeg valgt å simulere morgenrushet. Grunnen til dette er at det er en enklere oppgave enn å simulere ettermiddagsrushet, som opplever enda større avviklingsproblem.

Vegen gjennom Kannik er fv. 509 og er hovedvegen gjennom modellen. Fv. 509 består av 4 felt med rabatt i midten og fortau på begge sider. Vegen er begrenset av bebyggelse på begge sider samt Kannikparken ved Rogaland teater. Strekingen som simuleres er fra E39 til Bergelandstunellen, en strekning på ca. 1 km. På denne strekingen er det etablert 4 rundkjøringer, i tillegg kommer tilstøtende veger som det er naturlig å ta inn i modellen.

## **5.2 Trafikkgrunnlag**

Å innhente og behandle trafikkdata som trengs til en slik modell vil være for resurskrevende i en slik oppgave. Staten vegvesen har fremskaffet trafikkdataene som brukes i denne oppgaven. Dataene er resultatet av en omfattende registrering som ble utført i 2006, hvor blant annet volum og svingebevegelser i kryss ble registret. Disse dataene kan studeres nærmere i bilag 2. På grunn av omfanget ble det aldri vurdert å bruke andre tall, men det ble gjort en vurdering av hvor gjeldene disse dataene er i dag. Vegvesenet har et tellepunkt i Bergelandstunellen som har registret trafikken siden 1998. Det viser at trafikkmengden har vært stabil med en YDT som har variert mellom 19- og 21 000 mellom 1998 og 2014. Disse tallene kan bety at området har ligget på sin kapasitetsgrense i mange år, noe som igjen viser at dataene fra 2006 ennå er gjeldene.

Tungbilandelen er beregnet ut fra vegvesenets tellepunkt i Bergelandstunellen i 2014, og er på 7 %.

### **5.3 Dagens kollektivtilbud**

I Kannik er det i dag to bussholdeplasser, et for sørgående og et for nordgående busser. Disse er godt betjent med 10 ruter og 2 min frekvens i rushperioden. I tillegg kommer ekspressbussene til/fra Bergen og Kristiansand.



## 6 SIMULERING AV DAGENS SITUASJON

---

Som en del av casen skal jeg vurdere hvordan man skal få bussen gjennom Kannik uten forsinkelse, og konsekvensene av dette. For å kunne vurdere effekten av et tiltak er det avgjørende å ha et sammenligningsgrunnlag. I dette kapittelet beskriver jeg hvordan jeg har kommet fram til resultatet som tiltaket skal vurderes opp mot.

### 6.1 Innledning

Mye tid og resurser er brukt på å sette seg inn i programmet og lære seg de elementære tingene om hvordan man bygger opp en modell som man kan hente ut data fra, med en viss kvalitet. Modellbygging er en interaktiv prosess der man ofte må gå tilbake og endre på elementer etterhvert som man starter på verifiseringsarbeidet. Denne «fram og tilbake» prosessen er ikke beskrevet i denne oppgaven. Videre i dette kapittelet beskriver jeg i korte trekk hvordan jeg gikk fram for å modellere 0-alternativet

### 6.2 Bygge modellen

Det første jeg gjorde var å laste inne et ortofoto og et kart i DWG format. Med hjelp av DWG-filen skalerte jeg ortofotoet slik at det fikk rette proposisjoner.

Etter at ortofotoet var på plass, hadde jeg et godt grunnlag for å trekke links etter vegens geometri og forbinde disse med connectors. Etter dette kom utfordringen med å modellere rundkjøringene. Dette blir bedre beskrevet senere i dette kapittelet.

For å lage en mest mulig realistisk modell bør man legge inn midlertidig fartsreduksjon i krappe kurver, tilfarer og sirkulasjonsområder. I VISSIM kalles denne funksjonen «reduced speed area». Dersom 0-alternativet hadde inneholdt signalregulert styring, hadde jeg prøvd å skaffe meg gjeldende SK-skjema for å modellere signalanlegget mest mulig korrekt. Det neste steget var å løse alle konfliktene som «conflict area»-funksjonen hadde markert.

Når dette var gjort anga jeg hvor mange kjøretøy som skulle genereres inn i modellens grensesnitt. Dette ble gjort med «Vehicle input»-funksjonen, og trafikkdataene fra vegvesenet. Utfra de samme trafikkdataene definerte jeg ruter og svingebevegelser gjennom kryss ved hjelp av «Vehicle Routes (Static)»-funksjonen.

Før jeg begynte å endre på parameterverdier og lignende, kjørte jeg en simulering for å avdekke de mest åpenbare feilene.



## 6.3 Sentrale parametere

Etter at modellen var bygd begynte jeg å eksperimentere med parameterne. I programmets brukermanual advares det mot å endre på parameterne dersom man ikke har erfaring i bruk av programmet. Da hovedhensikten med denne oppgaven var å lære seg VISSIM, og resultatene fra casen kom i andre rekke, var jeg ikke bekymret for å endre for mye eller gjøre feil. Jeg lagde en kopi av alle parameterne jeg endret, slik at jeg alltid kunne hente inn de originale verdiene dersom jeg skulle «gå meg vill» i parameterverdier og dialogbokser.

Siden den danske mikrosimuleringshåndboken er skrevet hovedsakelig for VISSIM brukere, har jeg prøvd å forholde meg til de verdiene som håndboken anbefaler.

### 6.3.1 Atferdsparametere

Det første jeg gjorde var å lage meg en egendefinert atferdstype som var en kopi av Urban(motorized) og gi den et unikt navn, ref. kapittel 4.3 2.

I kalibreringsfasen opplevde jeg problemer med feltskifte. Etter tips fra den danske mikrosimuleringshåndboken og veileder, tilegnet jeg sjåførene en mer aggressiv atferd ved å endre minimum retardasjon for feltskifte fra 1 m/s<sup>2</sup> til 2 m/s<sup>2</sup>. I tillegg endret jeg på villigheten ett kjøretøy har for å slippe inn kjøretøy som ligger i nabofeltet. Dette ble gjort ved å endre parameteren «Maximum deceleration for cooperative braking» fra 3 m/s<sup>2</sup> til 7. Dette er og en verdi som den danske mikrosimuleringshåndboka anbefaler.

Når det kommer til terskelverdiene i Car-following-modellen har jeg ikke grunnlag til å endre på disse. Den danske mikrosimuleringshåndboka anbefaler heller ikke å endre på disse verdiene.

### 6.3.2 Nettverksparametere

Jeg begynte med å tilegne den nye atferdstypen på alle links og connectors. Så gav jeg inn stigning på de links der det var aktuelt. Restriksjoner for feltskrifte i tilfartene ble også angitt der det var aktuelt. Fartsgrensen for nettverket ble angitt i kjøretøykomposisjonen.

### 6.3.3 Kjøretøyparametere

På samme måte som for atferdstyper, definerte jeg en egen kjøretøykomposisjon med kjøretøyfordelingen lette og tunge, samt egendefinerte kjøretøyklasser

I følge den danske mikrosimuleringshåndboken ligger VISSIM's default verdiene for akselerasjon for høyt, sammenlignet med målinger på det danske nettverket. Mens VISSIM

opererer med en maksimal akselerasjon på  $3,5 \text{ m/s}^2$  er det i Danmark målt en gjennomsnittlig maks akselerasjon på  $2,8 \text{ m/s}^2$ . Denne verdien ble endret til den danske verdien.

#### **6.3.4 Simuleringsparametere**

På grunn av at trafikkdataene bygger på registreringer fra 0730-0830, ble simuleringsperioden satt til 3600 sekunder.

Simuleringens oppløsning ble satt til 15 beregninger i sekundet, til sammenligning anbefaler den danske mikrosimuleringshåndboken minimum 10.

I den danske mikrosimuleringshåndboken er det gitt en anbefaling at stokastiske mikrosimuleringsmodeller bør kjøres minimum 10-15 ganger. Jeg har valgt å kjøre 20 simuleringer.

For å finne oppvarmingstiden brukte jeg tommelfingerregelen som er beskrevet i kapittel 3.9, som sier at et kjøretøy bør ha kjørt gjennom modellens lengste rute før man begynner å samle inn data (Hoogendoorn, 2003). Denne tiden fant jeg ved å måle reisetiden på den lengste ruten ved hjelp av traveltime-funksjonen i VISSIM. Etter 5 simuleringer viste resultatet en gjennomsnittsverdi på 221 sekunder. Utfra dette valgte jeg en oppvarmingsperiode på 240 sekunder (4 min).

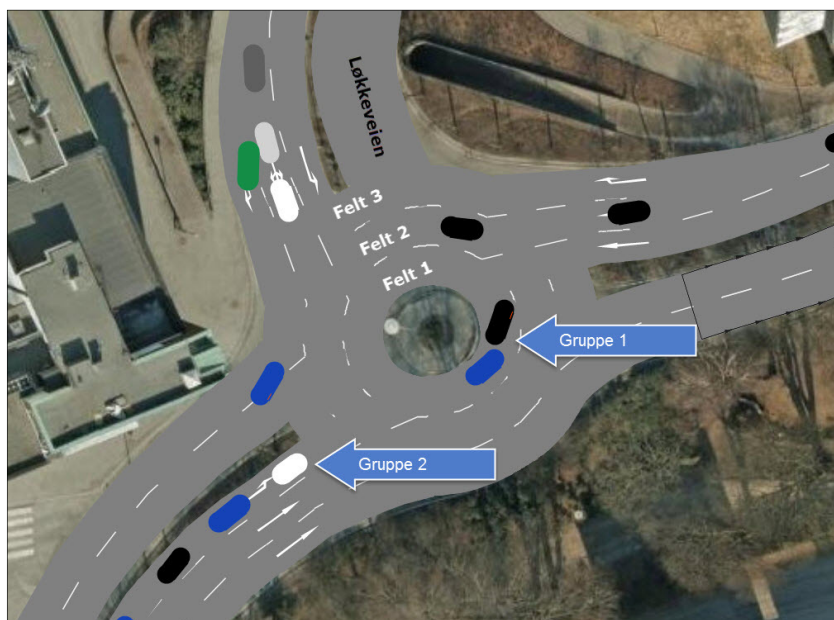
## 6.4 utfordringer

I forbindelse med simuleringen av 0-alternativet, traff jeg på noen utfordringer. Det var spesielt to elementer ved modelleringen som var utfordrende; å modellere trefeltsrundkjøringer som klarte å gjengi en akseptabel avvikling, samt bruken av statiske ruter.

### 6.4.1 Rundkjøring

Som i de fleste andre mikrosimuleringsmodeller er rundkjøring noe av det vanskeligste å modellere i VISSIM. I tillegg hadde tre av rundkjøringene i 0-alternativet 3 felt i sirkulasjonsarealet og 3 felt i tilfarten. Da jeg ikke hadde erfaring med mikrosimulering og VISSIM før jeg startet på denne oppgaven ble det en del prøving og feiling, og jeg endte opp med å prøve ut 3 forskjellige strategier før jeg klarte å modellere en rundkjøring som gav en akseptabel avvikling.

I det første forsøket gikk jeg for en strategi der sirkulasjonsfeltet i rundkjøringen var modellert som en sirkulær link med 3 felt. For at kjøretøyene skulle kunne kjøre sammenhengende i sirkulasjonsarealet ble denne linken forbundet med en så liten connector som mulig. Denne «skjøten» ble plassert midt på rundkjøringens deleøy for ikke å komme i konflikt med connectoern mellom tilfart og sirkulasjonsarealet. Tilfartene ble og modellert som en trefelts link der det var aktuelt. Tilfartene ble knyttet sammen med sirkulasjonsarealet.

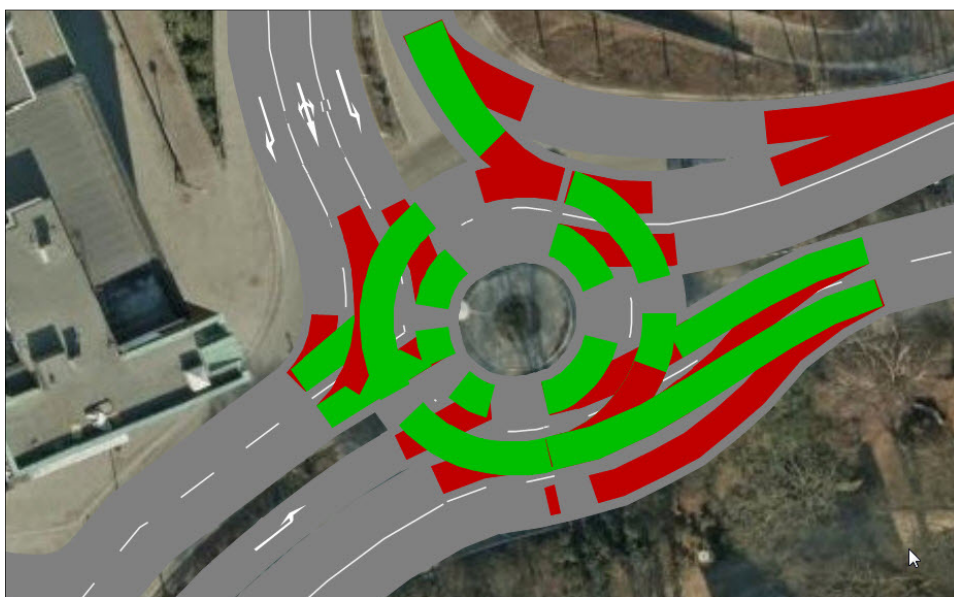


Figur 20-Illustarsjon, alternativ 1

Problemet med dette alternativet, var at de som skulle til venstre i rundkjøringen måtte gjøre tre feltskifter i sirkulasjonsarealet på et forholdvis kort strekke, for å komme til connectoren

som ledet til frafarten. Det en opplevde da, som man kan se utfra figur 20, var at kjøretøy (gruppe 1), ble stående i det innerste feltet i rundkjøringen å vike for vestgående tilfart. Dette førte igjen til at det bygget seg opp kø (gruppe 2), som påvirket avviklingen for de andre feltene i tilfarten. Etter å ha justert på parametere i Lane-change-modellen og «conflict area» bedret avviklingen seg noe, men man kunne stadig oppleve unaturlig adferd. Jeg bestemte meg da for å prøve en annen utforming.

Den andre strategien jeg hadde, var å la sirkulasjonsarealet bestå av to felt, og la det tredje feltet ligge på utsiden av sirkulasjonsarealet. I tillegg bestod tilfartene av flere separate links og connectors.



Figur 21-Illustasjon, alternativ 2

Problemet med dette alternativet var at mange links og connectors ble liggende oppå hverandre, og føre til overlappende konfliktområder, se figur 21. Dette førte til en nølende adferd som igjen førte til en dårligere avvikling i rundkjøringen.

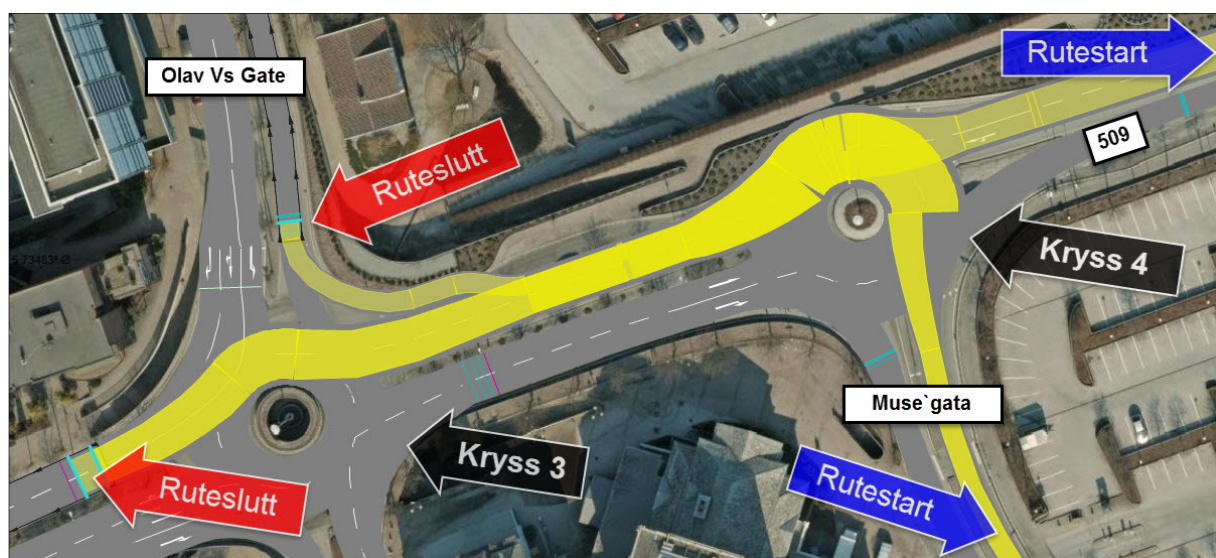
Etter anbefaling fra veileder ble det utarbeidet et tredje utkast som kombinerte de to første alternativene med fokus på å redusere antall konfliktområder. Løsningen ble å dele opp sirkulasjonsarealet i flere sektorer, med to eller tre felt for å unngå problemene som oppstod i det første alternativet. For ikke å oppleve problem med overlappende konfliktområder, slik som i alternativ to, reduserte jeg antall links og connectors i tilfartene der det lot seg gjøre, noe som igjen førte til færre konfliktområder. Dette alternativet viste seg å gi en akseptabel avvikling av trafikken.

Jeg opplevde også utfordringer med atferdsparameterne knyttet til modellering av rundkjøring, dette er nærmere beskrevet senere i kalibreringskapittelet.

## 6.4.2 Ruter

Den danske trafikksimuleringshåndboken kategoriserer simuleringsoppdrag i større og mindre modeller. I følge håndboken kommer denne modellen inn under store modeller. I håndboken anbefales det å bruke «dynamic route assignment» også kalt OD-matrise. Dette er standard i noen mikrosimuleringsprogram, men i VISSIM er dette en tilleggsmodul som må kjøpes. Denne modulen fulgte ikke med studentversjonen som jeg fikk av PTV. Som følge av dette måtte jeg bruke statiske ruter, som er forklart i kapittel 4.

Problemet med statiske ruter er at kjøretøyene ikke klare å planlegge turen videre i nettverket. De klarer dermed ikke å posisjonere seg med tanke på kommende kryss. En rute bør begynne lengst mulig fra krysset den skal gå gjennom, minimum 100-200m, slik at kjøretøyene kan begynne å planlegge eventuelle feltskifer før «emergency stop». Problemet oppstår når det er kort avstand mellom kryss slik at man får korte ruter. Kjøretøy som kommer inn i ny rute, men i feil felt vil ha en kort strekning til å skifte felt, og ender ofte opp med å måtte stoppe helt opp ved «emergency stop». Dette førte til dårlig flyt i trafikken og redusert kapasitet.



Figur 22-Ruter

Dette problemet opplevde jeg mellom kryss 3 og 4. Etter at kjøretøy som kom fra Bergelandstunellen og Muse`gata hadde fullført ruten gjennom kryss 4, kjørte de inn på ruten som ville ta dem gjennom kryss 3. Dersom de kom inn på denne ruten i et felt som ikke var ønskelig, hadde kjøretøyet kun 45 m å skifte felt før «emergency stop». Dette førte til mange stopp og dårlig flyt i trafikken. Som man kan se utfra de gule feltene på figur 22, løste jeg dette problemet med å lage rutene lengre, og å la den gå gjennom både kryss 3 og 4. Dette førte til at kjøretøyene fikk bedre tid til å planlegge feltskifte og det ble bedre flyt i trafikken. Som følge av dette måtte jeg estimere nye trafikk tall for rutene.



## 6.5 Verifisering, kalibrering og validering av modellen

### 6.5.1 Verifisering

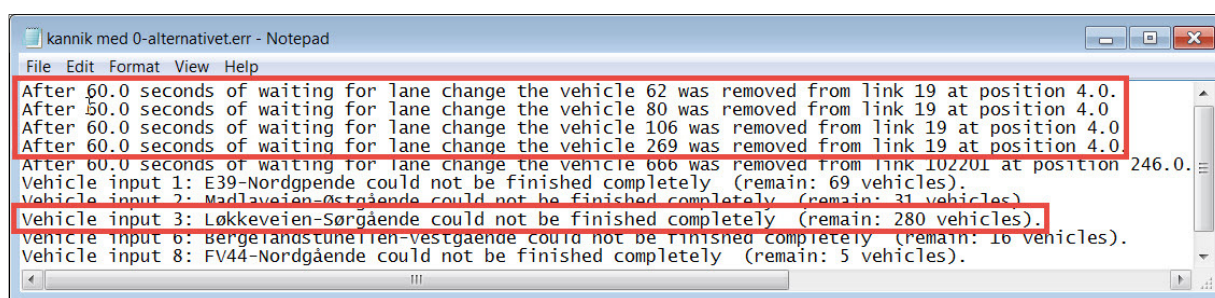
Som nevnt i kapittel 3, er verifisering det første steget i kvalitetssikringsarbeidet. Det er her man luker ut det mest opplagte og synlige feilene. Verifisering er en interaktiv prosess der man kan kjøre en simulering og studerer animeringen, oppdage feil for så å rette feilen. Denne prosessen gjentas til modellen er akseptabel.

På valideringsstadiet fokuserte jeg på å rette opp feil på links og ruter i forbindelse med rundkjøringene. Da jeg ikke hadde noen tidligere erfaring med å bygge nettverk og rundkjøringer ble det en del prøving og feiling før jeg fikk et akseptabelt resultat. I tillegg er kryss 1,2 og 3 utformet som 3 felts rundkjøringer, noe som gjorde det enda mer komplisert.

I tillegg til å bruke animeringen for å oppdage feil i nettverket, finner man også nyttig informasjon i feilmeldingsfilen. Figur 23 viser en feilmeldingsfil jeg fikk i forbindelse med valideringsarbeidet. I tillegg til å observere problem med feltskifte, kan man utfra figuren se at modellen har fjernet biler på link 19, som følge av feltskifteproblemer.

Man kan også se at 280 biler fra Løkkeveien-sørgående ikke har blitt generert inn i modellen i løpet av simuleringsperioden. Noen stokastiske variasjoner vil man kunne oppleve, men 280 biler er for mye. Ideelt sett skal alle bilene genereres inn i modellen ved 0-alternativet, da man vet utfra trafikkregistreringene at området har kapasitet til å avvikle dette.

Feilmeldingen viste at jeg måtte se nærmere på forholdene rundt feltskifte og grunnen til at så mange biler ikke ble generert inn i modellen ved Løkkeveien-sørgående.



Figur 23-Feilmeldingsfil

### 6.5.2 Kalibrering

Da trafikkdataene jeg fikk fra vegvesenet ikke inneholdt kalibreringsdata som kølengde, reisetid osv. fikk jeg pr. definisjon ikke kalibrert modellen. Kalibrering ble derfor en mellomting mellom kalibrering og verifisering, der jeg studerte animeringen og justerte på

parametere hovedsakelig forbundet med adferd. Til dette arbeidet fikk jeg god hjelp av Sebastian Nerm i Norconsult. Som tidligere nevnt i kapittel 6.3.1 endret jeg noen av parameteren i Lane-change-modellen. Dette førte til mindre kødannelse som følge av problemer med feltskifte. I tillegg viste tidligere feilmeldingsfiler at mellom 20 og 30 biler ble tatt ut av modellen etter 60 sekunder på grunn av at de ikke fikk skiftet felt. Dette endret seg til 0 biler etter at jeg endret noen av parameterene i Lane-change-modellen.

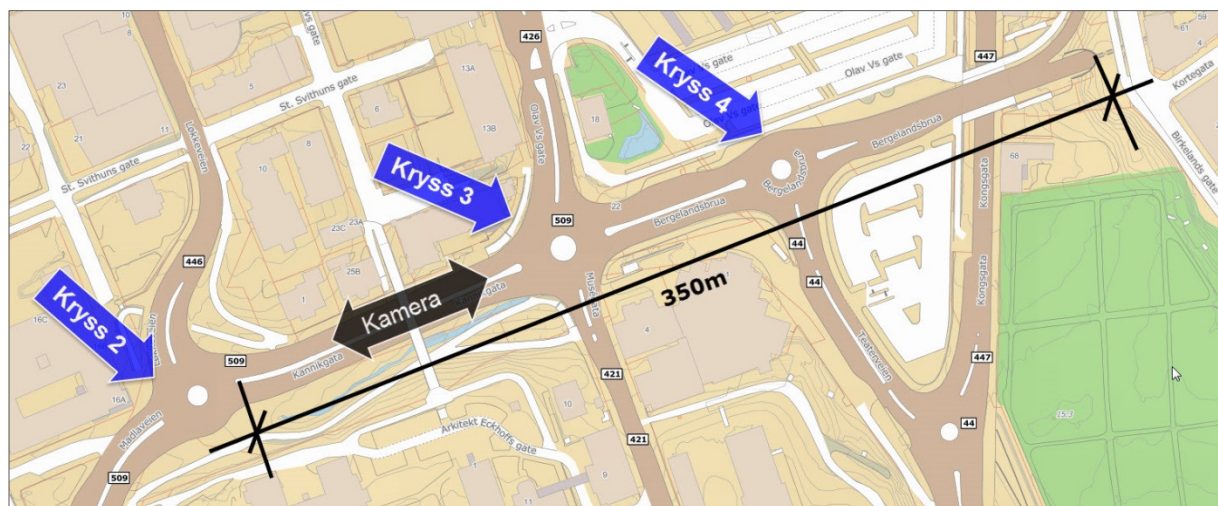
I tillegg til å ha problem med feltskifte, opplevde jeg problemer i forbindelse med avviklingen i rundkjøringene. Her observerte jeg en nølende adferd som skapte unaturlige køer. Et av problemene var at jeg hadde for mange links og connectors i forbindelse med oppbyggingen av rundkjøringen. Dette førte til mange «conflikt areas» som igjen bidro til denne nølende adferden. Etter at jeg hadde endret på geometrien begynte jeg å justere på noen av parameterverdiene på konflikta area.

Det var spesielt parameterverdiene for kryssende konflikter, der et kjøretøy har forkjørsrett som ble justert, ref. kapittel 4.4.2. I følge et eksempel fra PTV anbefales det følgende endringer ved rundkjøring «Front Gap» redusert fra 0,5s til 0,4s., Rear Gap økes fra 0,5s til 1,5s., og «Safety distance factor» reduseres fra 1,5 til 1,0. I tillegg er det to andre parametere i «conflikt area» som har stor betydning for adferden i rundkjøring. Det er «Anticipate routes» og «Avoid blocking». Som beskrevet i kapittel 4.4.2, beskriver «Anticipate routes» andelen av vikende kjøretøy som klarer å forutse kjøretøy i primærstrømmen som ikke vil komme inn i konfliktområdet. Med andre ord er dette en parameter som beskriver hvor mange kjøretøy som klarer å «lese» trafikken. Som default er verdien for denne parameteren 0. I denne oppgaven har jeg variert denne verdien mellom 60 og 90%, som igjen viste seg å ha en god effekt på avviklingen.

Den andre verdien som viste seg å ha stor betydning er «Avoid blocking». Denne parameteren beskriver hvor stor andel av kjøretøyene med forkjørsrett som unngår å blokkere andre trafikkstrømmer ved ikke å kjøre inn i et konfliktområde. Som default er denne verdien 100, det vil si at alle vil unngå å kjøre inn i et konfliktområde der det står kø. Problemene oppstår mellom tilfart og sirkulasjonsarealet når det begynner å bli en del trafikk. Kjøretøy i sirkulasjonsarealet vil ha forkjørsrett framfor de i tilfarten, men vil oppfatte trafikken fra tilfarten som kø. Dette fører igjen til en unormalt defensiv kjøring fra kjøretøy som skal inn i sirkulasjonsarealet, da disse kjøretøyene ikke klare å beregne hvordan de skal forsere konfliktområdet uten å blokkere trafikken fra tilfarten. Dette problemet løses ved å nedjustere denne parameteren. Man kan alltid diskutere parameterverdier, men i denne casen har jeg valgt verdien 0.

### 6.5.3 Validere

I denne oppgaven brukte jeg reisetid som valideringsparameter. Jeg var ute i 2 morgenrush og målte reisetider fra Bergelandstunellen til rundkjøringen mellom Løkkevegen og fv509, en strekning på 350 m, se figur 24.



Figur 24-Strekning for reisetidsregistrering

For å kunne registrert reisetiden, og studere trafikken ved en senere anledning, satte jeg opp to videokamera på GS-broen over fv509 for å filme trafikken. Registreringsperioden varte i en time og det ble gjort målinger av reisetiden for hvert 15 min.

Som man kan se fra resultatene i Bilag 3, er den registrerte reisetiden på 75 sekunder, mens den beregna reisetiden er på 102 sekunder. Dette betyr at modellen ikke klarer å avvikle den trafikkmengden som avvikles i virkeligheten. Iht. den danske håndboken for mikrosimulering anbefales det at avviket mellom registrert data og beregna data er maks 15 %, dersom man skal kunne si at modellen er validert. Avviket i modellen for 0-alternativet er beregna til 36 %.



Bilde 1-Registrering av reisetid



## 6.6 Resultat

Som tidligere nevnt, skal jeg i denne oppgaven bruke «forsinkelse» for å beskrive konsekvensene ved å innføre kollektivtiltak i Kannik. Resultatene fra dette kapittelet skal derfor sammenlignes med resultatene fra kapittel 8.

Som man ser utfra figur 25 har jeg valgt å beregne forsinkelsen for tre strekninger:

- Strekning 1: Fv. 509-Vestgående retning, 810m
- Strekning 2: Løkkeveien-Sørgående retning, 175m
- Strekning 3:Olav Vs gate-Sørgående retning, 210m



Figur 25-Oversiktskart, strekninger

Count: 57	SimRun	TimeInt	DelayMeasurement	StopDelay(All)	Stops(All)	VehDelay(All)
45	15	240-3600	3: Olav Vs gate-Fv509	5.98	0.26	10.71
46	Average	240-3600	1: Bergeland-E39	12.71	3.15	109.89
47	Average	240-3600	2: Løkkeveien-Fv509	10.89	0.80	49.48
48	Average	240-3600	3: Olav Vs gate-Fv509	41.01	1.07	61.92

Figur 26-Resultat, reisetid

Simuleringen ble gjort iht. kapittel 6.3.4 og resultatene vises i figur 26. Som man kan se utfra figuren får vi følgende forsinkelser:

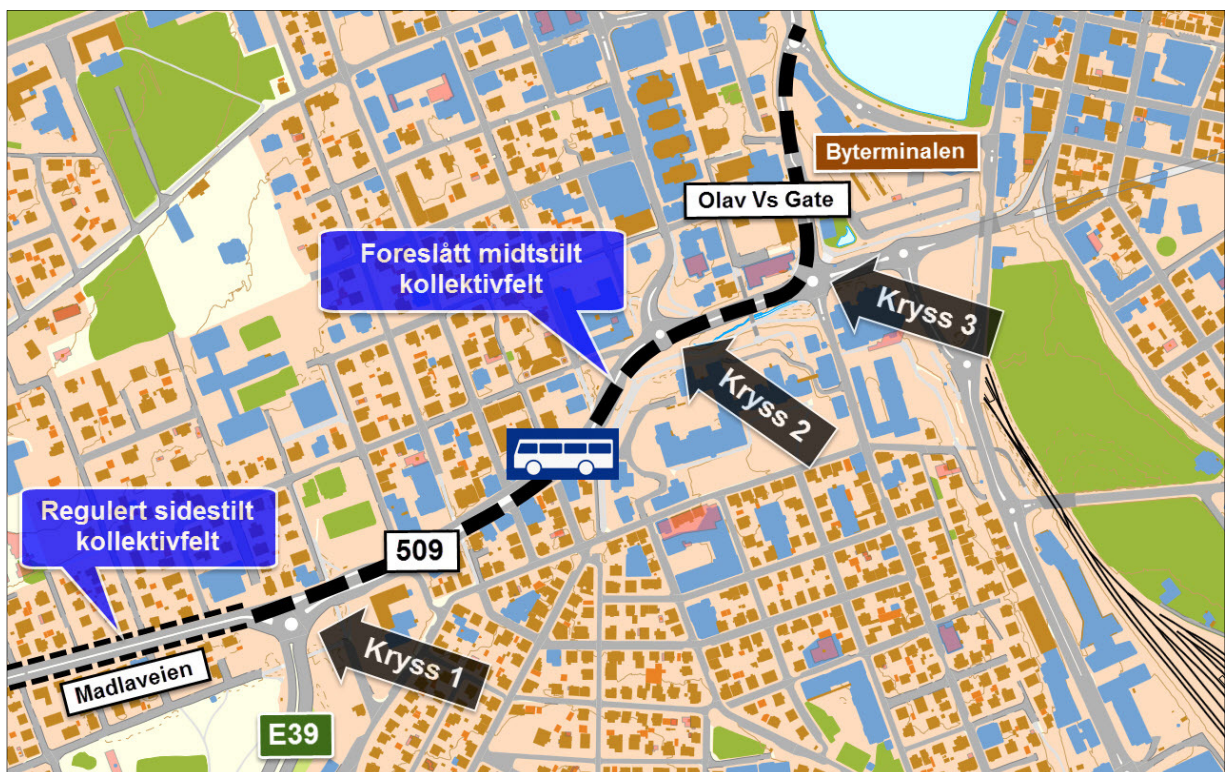
- Strekning 1: Fv. 509-Vestgående retning, 110 sekunder.
- Strekning 2: Løkkeveien-Sørgående retning, 49 sekunder.
- Strekning 3:Olav Vs gate-Sørgående retning, 62 sekunder.

# 7 TILTAK

Dette kapittelet beskriver hvilke tiltak som legges til grunn for casens problemstilling. Oppgaven er å utforme et tiltak som gjør det mulig at bussen kan kjøre fra Byterminalen, gjennom kryss 1 og inn på sidestilt kollektivfelt i Madlavegen uten forsinkelse.

## 7.1 Bakgrunn

Mandatet til vegvesenet som denne casen bygger på/ Mandatet som er gitt vegvesenet i denne casen, er at det ikke skal være noe forsinkelse på kollektivtrafikken. Dette betyr at det må planlegges et tiltak der bussen kan kjøre upåvirket av biltrafikken. Det mest nærliggende er å planlegge egne kollektivfelt der bussen prioriteres gjennom kryssene. Siden det allerede foreligger et gatesnitt, må man ta to felt fra den ordinære trafikken og gjøre om til kollektivfelt, se figur 27.



Figur 27-Oversiktskart, kollektivfelt

## 7.2 Midtstilt vs. sidestilt kollektivfelt

Midtstilt eller sidestilt kollektivfelt? Dette er et spørsmål som stilles i forbindelse med Bussvei2020. Begge løsningene har sine styrker og svakheter. Midtstilt kollektivfelt sine største utfordringer er sikkerhet for myke trafikanter og arealbehov.

For at en myk trafikant skal kunne komme til holdeplassen, må den krysse minimum et felt med ordinær trafikk, det samme må gjentas når passasjeren går av bussen. Til forskjell fra sidestilte kollektivfelt, hvor holdeplassen grenser mot fortauet. Denne situasjonen kan forbedres ved å anlegge signalregulert fotgjengerkryssing, noe som igjen vil føre til nedsatt kapasitet på de ordinære vegene. Ett annet argument mot midtstilt kollektivfelt er arealbehovet. Ved midtstilte kollektivfelt er det oftest en fysisk rabatt mellom ordinær vegtrafikk og kollektivtrafikk. Bredden på denne kan variere, men ved holdeplassene har den en minste bredde på 3,5 m.

Grunnen til jeg har valgt midtstilt kollektivfelt i denne oppgaven er at Kannik kan regnes som Stavanger sentrum, og et midtstilt kollektivfelt gir et mer urbant preg over byen. Området har også noen sidegater som ender ut i fv. 509. Ved å etablere midtstilt kollektivfelt, unngår man eventuelle konflikter et sidestilt kollektivfelt vil skape. Midtstilt kollektivfelt har og en sterkere signaleffekt ovenfor befolkningen om at bussen prioriteres.

### **7.3 Kryssløsninger**

For at bussen skal kunne kjøre uten forsinkelse er det avgjørende at den prioriteres gjennom kryssene. Den mest aktuelle måten for å oppnå dette, er å signalregulere kryssene. Hovedsakelig brukes signalregulering i forbindelse med X og T-kryss, men man kan og signalregulere rundkjøringer.

Siden det allerede er etablert rundkjøringer med midtstilt kollektivfelt i Stavanger, følte jeg det naturlig å videreføre dette i kryss 1 og 2. På grunn av geometrien og en generell vurdering, viste det seg vanskelig å utforme kryss 3 som en rundkjøring. Denne ble derfor utformet som et X-kryss med midtstilt kollektivfelt gjennom.

### **7.4 Trafikktall**

Tallene viser at trafikken har vært stabil siden 1998. Dette kan bety at dagens forsinkelse, representerer grensen for hva de som kjører gjennom området i dag, er villig til å akseptere av forsinkelse.

Med det nye tiltaket vil man sannsynligvis få redusert trafikkmengde. Da det er stor uvisshet rundt de nye trafikkmengdene, har jeg valgt å benytte samme trafikktall som i 0-alternativet. De eneste tallene jeg har endret, er trafikkmengden som kommer fra Bergelandstunellen. Vegvesenet har estimert en reduksjon på ca 10 %, som følge av Ryfast-tunellene.

## 7.5 Kollektivtrafikk

I dette alternativet tok jeg utgangspunkt i dagens rutetabell, med en frekvens på to minutter mellom bussene.

VISSIM kan beregne stopptid ved holdeplass utfra antall passasjerer, eller man kan angi en fast stopptid. Da jeg ikke hadde noe passasjergrunnlag, tok jeg kontakt med Kolumbus, som er bussoperatøren på Nord Jæren. De kunne informere om at de var i ferd med å innføre et nytt billetteringssystem for å redusere tiden på holdeplass som følge av Bussvei 2020. Et av tiltakene var billettering på holdeplass. I deres beregninger la de til grunn 15 s stopp på sentrumsnære holdeplasser. På grunnlag av dette har jeg valgt å bruke 15 s som fast stopptid ved holdeplass.

I forbindelse med Bussvei 2020 har vegvesenet utarbeidet retningslinjer for utforming og plassering av holdeplasser for midtstilte kollektivfelt. Et av kravene er at holdeplasser ikke kan plasseres nærmere en 150 m fra bussprioritert lyskryss. Ved denne avstanden detekteres bussen, signalanlegget skifter fase og krysset får tømt seg.

På den aktuelle strekningen har jeg plassert et busstopp. Jeg ønsket å plassere busstoppet i nærheten av det andre. Siden dette lå for nærme kryss 2, måtte jeg flytte det ca. 100 m vestover.





## 8 SIMULERING AV TILTAK

---

Dette kapitlet bygger på tiltakene som ble beskrevet i kapittel 7, og beskriver hvordan disse kan modelleres i VISSIM.

### 8.1 Bygge modellen

Nettverksoppbyggingen av dette alternativet anser jeg som en enklere oppgave enn 0-alternativet, og jeg opplevde få utfordringer. Grunnen til dette var at dagens rundkjøringer i Kannik er utformet på en måte som gjør de vanskelige å modellere. I dette alternativet kunne jeg derfor utforme rundkjøringene på en måte som gjorde de enkle å modellere.

Siden mandatet var å få bussen fram upåvirket av annen trafikk, ble modellen bygget opp ved at jeg trakk to sammenhengende linker fra kryss 1, gjennom Kannik til enden av Olav Vs. gate, en link for hver kjøreretning. Linkene ble så tilpasset vegen på ortofotoet og gitt den egenskapen at kun busser kunne kjøre på disse. Resten av nettverket ble modellert rundt disse linkene. Se figur 28.



Figur 28-Illustrasjon, Midtstilt kollektivfelt

Det andre jeg gjorde var å trekke de linkene som utgjorde resten av vegnettet. Hovedtanken bak tiltaket er å gjøre to av feltene til den ordinære trafikken om til kollektivfelt. Derfor la jeg en felts link på utsiden av kollektivfeltene, parallelt, med ca. 3 m rabatt.

Rundkjøringene ble modellert som en tofelts link, med to felt i tilfarten og et felt i frafarten, som igjen ble forbudent med connectors.

## 8.2 Parameterinnstillinger

En av grunnene til å modellere et 0-alternativ er at man her skal gjøre seg opp en mening om parameterverdier og kalibrere disse opp mot en «fasit». Jeg har derfor overført parameterverdiene som jeg kalibrete i 0-alternativet over i denne modellen.

## 8.3 Andre elementer

For å kunne sammenligne to simuleringer er det viktig at modellene er så like som mulig. Jeg har derfor overført rutene og kjøretøyegenskapene fra 0-alternativet til denne modellen.

## 8.4 Modellere signalanlegg

Det første man må gjøre er å definere signalanlegget, også kalt signalgruppe. I denne gruppen definerer man antall faser, og hvilken type styring anlegget skal ha. Videre må man sette ut signalhodene i nettverket. Signalhodene plasseres oppå en link eller connector der man ville hatt stopplinjen. Signalhodet gis et nummer, og knyttes til den aktuelle signalgruppen man tidligere har definert.

Det finnes to typer signalanlegg i VISSIM, tidsstyrte og kjøretøystyrte. Tidsstyrte anlegg er de enkleste å bruke. Her kan man relativt fort og enkelt få satt opp signalanlegg med egendefinerte tidsstyrte faser. Det negative med å bruke tidsstyrte anlegg er at de er lite fleksible. VISSIM foreslår heller ingen faseinndelinger, så dette må man ha gjort på forhånd.

Den andre måten å modellere et signalregulert kryss på, er å bruke kjøretøystyrte signalanlegg. I tillegg til signalhoder må man sette ut detektorer. Detektorene legges oppå en link eller en connector, nummereres og knyttet til en signalgruppe. Når man bruker kjøretøystyrte signalanlegg, må man linke to forhåndsprogrammerte informasjonsfiler til den aktuelle signalgruppen, en .pua- og en .vap-fil. Pua-filen beskriver hvilke signalhoder som skal være aktive i hvilke faser, og hvordan fasevekslingen skal foregå. Programmeringen av pua.-filen gjøres i teksteditor.

Vap-filen beskriver signalprogrammets forløp. Denne filen genereres utfra et program som heter VisVAP, som også er utviklet av PTV. I VisVAP beskriver man handlinger utfra oppståtte hendelser ved hjelp av et flowchart. For eksempel, dersom en fotgjenger beveger seg innenfor en detektor, sendes en beskjed til signalanlegget om å skifte fase

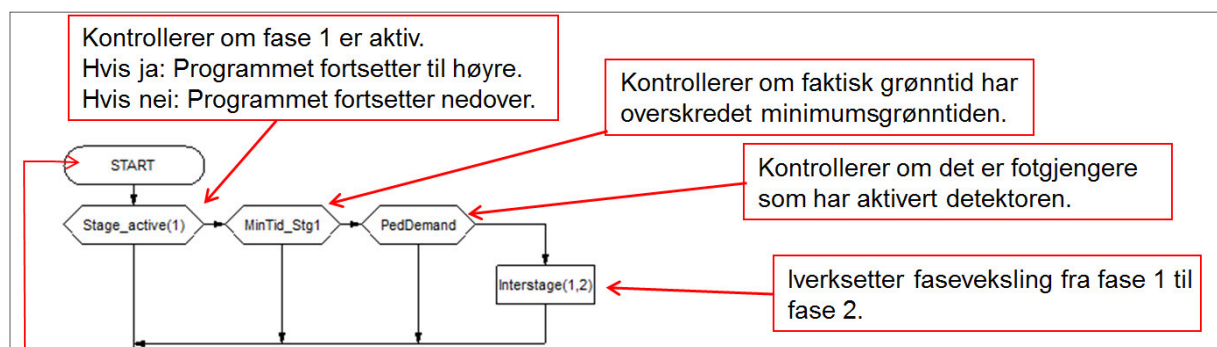
VisVAP er et veldig fleksibelt verktøy, som gjør det svært egnet til å modellere ITS-anlegg og prøve ut alle tenkelige scenarioer. Dette gjør igjen at brukersnittet på programmet er relativt høyt. VisVAP er et tilleggs program som må kjøpes ekstra.

VisVAP er for de fleste vanskelig å lære seg på egenhånd. Til dette arbeidet fikk jeg kyndig veiledning av Sebastian Nerem i Norconsult.

Programmet går ut på at man definerer variabler og handlinger i et flowchart. VISSIM går gjennom flowchartet hver 0.1 s for å se om gitte betingelser er oppfylt. Variablene beskriver en betingelse, og returnerer en verdi på enten 0 eller 1 ut fra om betingelsen er oppfylt eller ei. VISSIM beveger seg gjennom flowchartet nedover eller til høyre på bakgrunn av disse verdiene. En variable kan for eksempel knyttes til en detektor i modellen. Dersom detektoren er aktiv, returnerer den verdien 1 som betyr at programmet må følge flowchartet mot høyre i programmet. Her kan den treffe på en ny variable eller en handling. En handling er oftest knyttet til aktivering av en faseveksling. Dersom en variabel ikke er aktiv returnerer den verdien 0, som betyr at VISSIM skal gå nedover i flowchartet hvor den treffer på nye variabler eller handlinger.

Et eksempel kan være å knytte en køteller til en variabel. Etter at den har talt for eksempel 20 biler, returnerer variabelen verdien 1 og programmet blir ledet mot høyre til en handling som kan være å gi de bilene som står i kø, grønt lys.

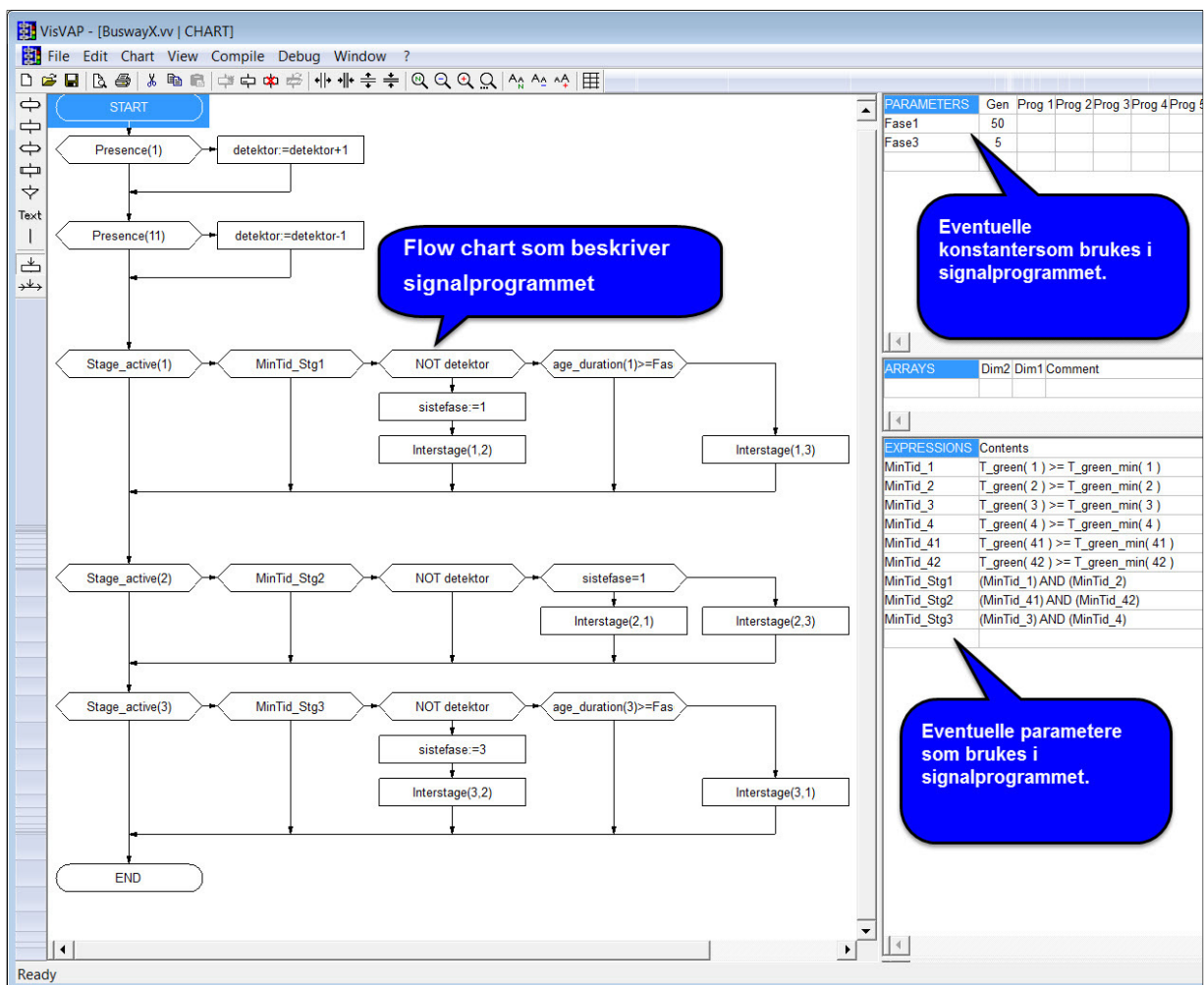
Figur 29 viser et flowchart for et detektorsyrt signalanlegg tilknyttet et gangfelt. Variablene er vist som sekskanta bokser, mens handlingen er vist som firkanta bokser. Detektoren er i dette eksempelet kalt for «PreDemand». Dersom den ikke er aktivert går programmet nedover i Flowchartet og forsetter å gi biltrafikken grønt lys. Dersom den aktiveres går programmet til høyre og treffer den firkanta handlingsboksen som iverksetter fasevekslingen. Varigheten av hver fase, samt vekslings og tømmetider, beskrives i pua-filen.



Figur 29-Flowchart med forklaring-Illustrasjon: Sebastian Nerem



Figur 30 viser VisVAP og signalprogrammet for kryss 3. Som man ser er VisVAP delt inn i tre vinduer. Ett der man beskriver flowchartet, ett for konstanter og ett der man definerer andre parametere. Signalanlegget for kryss 3 er en kombinasjon av tids- og kjøretøystyrt. Det kan man se ved at det er lagt inn verdier for konstanter i tabellen. Utrykkene som står i tabellen for eventuelle parametere refererer til minimums grøntid som er angitt i VISSIM samt faseinndeling og fasevekslingstidene som er angitt i pua filen. Utfra dette genererer VisVAP vap-filen, som linkes til signalanlegget i VISSIM.



Figur 30-VisVAP

### 8.4.1 Signalregulert rundkjøring med midtstilt kollektivfelt

Rundkjøringen fungerer som en vanlig rundkjøring helt til bussen i kollektivfeltet nærmer seg krysset. Bussen kjører over detektoren som sender et signal til signalanlegget, som gir rødt i alle tilfartene. Etter noen sekunder ankommer bussen og kan kjøre uforstyrret gjennom krysset. Bilde 2 viser en slik løsning.



Bilde 2-Midtsilt Kollektivfelt i Hillevåg, Foto:Stavanger Aftenblad

Det er viktig å få justert anlegget slik at rundkjøringen tømmes i det bussen kjører inn i krysset, og at biltrafikken får grønt i det bussen kjører ut.

Et problem som kan oppstå ved slike kryss er at køen står gjennom rundkjøringen ved masse trafikk. Man kan da risikere at den planlagte tømningstiden for rundkjøringen blir for liten, og man risikerer at bussen må vente til den får store nok luker til at den kan slippe igjennom.

Signalanlegget til kryss 1 og 2 er like og har to faser. I første fase har den ordinære trafikken grønt, i andre fase har bussen grønt. Som utgangspunkt har fase en grønt. Når bussen kjører over innsjekksdetektoren, detekteres bussen og gir rødt til fase en. For at rundkjøringen skal få tid til å tømme seg, får bussen først grønt etter 8 sekunder. Denne tiden er justert slik at bussen skal slippe å bremse opp før rundkjøringen. Utsjekksdetektoren er plassert rett på utsiden av rundkjøringens sirkulasjonsareal. Etter at bussen har passert denne, skifter signalanlegget til fase en, og gir grønt til den ordinære trafikken. Figur 31 viser utforming og detektorplassering i kryss 2.



Figur 31-Detektroplassering, Rundkjøring

Ut fra figur 32, kan man se at jeg hadde litt andre utfordringer med kryss 1. Her var utfordringen å gå fra sidestilt til midtstilt og motsatt. I tillegg måtte jeg finne en måte å slippe de bussene som kom fra E39 inn i kollektivfeltet, og slippe ut de bussene som skulle inn på E39. Løsningen ble å trekke stopplinjene lengre bort fra krysset.



Figur 32-Illustrasjon, kryss 1



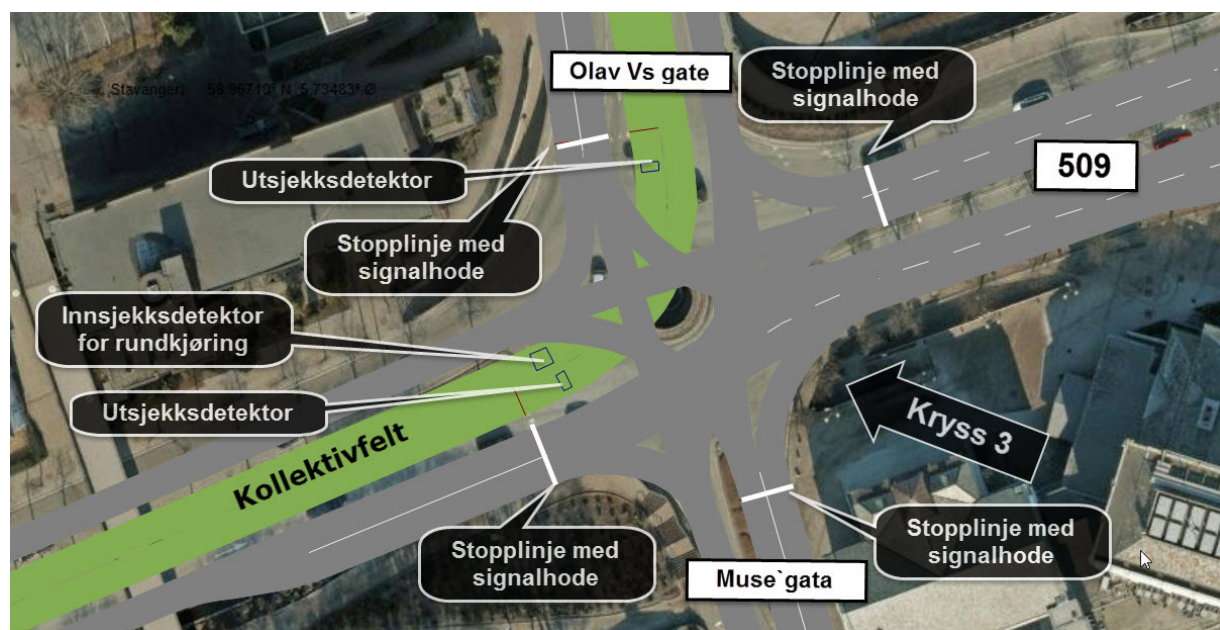
## 8.4.2 Signalregulerte X-kryss med midtstilt kollektivfeltprioritering

Kryss 3 er modellert som et vanlig signalregulert X-kryss, men vil være en kombinasjon mellom kjøretøystyrt og tidsstyrt. Hovedtanken er at krysset skal fungere som et vanlig lysregulert X-kryss, men, som med rundkjøringen, får alle fasene rødt når bussen detekteres. Forskjellen er at det er lagt inn gultid mellom fasevekslingene.

Signalanlegget har tre faser. Fase 1: Fv.509, Fase 2: Olav Vs gate og Muse` gata. Fase 3: Bussveien. Da det er et resurskrevende arbeid å programmere kjøretøystyrte signalanlegg ved hjelp av VisVAP var jeg i denne oppgaven nødt til å gjøre noen forenklinger. Den første forenklingen jeg har gjort er å gjøre signalanlegget delvis tidsstyrt, selv om det i praksis burde vært kjøretøystyrt med tanke på kølengde. I en rushperiode som den jeg skal simulere, vil det imidlertid fungere godt med tidsstyrt, da det alltid vil stå kø bak stopplinjene. Grunnen til at det passer med tidsstyrt i rushperioden er at omløpstiden vil bli brutt av bussen, ca hvert 60 sek. Etter å ha prøvd og feilet litt kom jeg fram til 50 s grønttid på fase 1 og 5 s grønttid for fase 2. For å sikre at fase 2 får min 5 s grønt pr. omløp har jeg programmert signalanlegget slik at signalanlegget starter på fase 2 etter at bussen er blitt detektert ut.

Den største forenklingen, og den som har størst konsekvens for simuleringen, er å gi rødt til all trafikk mens bussen kjører gjennom krysset. I praksis ville det være mulig å utføre noen svingebevegelser mens bussen kjører gjennom. Dette er ikke gjort fordi denne kodingen i VisVAP vil bli for resurskrevende for denne oppgaven.

Figur 33 viser utformingen av kryss 3.



Figur 33-Illustrasjon, kryss 3

## 8.5 Verifisering, validering og kalibrering

Når jeg startet verifiseringsarbeidet bestemte jeg meg for at hovedfokuset måtte ligge i å validere de tekniske tingene vedrørende signalanlegget. Det vil si å sjekke at signalanlegget utførte de operasjoner som jeg hadde programmert. Signalanleggene til kryss 1 og 2 fungerte som planlagt ved første forsøk. De eneste justeringene som ble gjort her var å trekke detektorene litt lenger bort fra krysset for å øke tømmeperioden av rundkjøringen, slik at bussen ikke måtte bremse opp før krysset.

Kryss 3 fungerte delvis slik jeg hadde planlagt. For at fase 2 skulle få nok grøntid, var planen at signalanlegget skulle skifte til fase 2 etter at fase 3 var ferdig. I valideringsfasen så jeg at dette ikke var tilfellet og at fase 2 fikk lite grøntid. På grunn av liten tid fikk jeg ikke rettet opp dette.

Som nevnt tidligere i dette kapittelet hadde jeg få utfordringer med oppbyggingen av nettverket. Jeg brukte masse resurser på validering og kalibrering i 0-alternativet, derfor gikk det lite resurser med på å verifisere og kalibrere selve nettverksoppbyggingen. Da dette alternativet ikke kan sammenlignes med dagens situasjon kan man heller ikke sammenligne trafikkdata for å validere modellen, slik man kunne i 0-alternativet.

Der er og verdt å nevne at feilmeldingsfilen viste et stort antall biler som ikke ble avvirket i løpet av simuleringsperioden.

## 8.6 Resultat

For å kunne sammenligne et før og etter scenario er det viktig at sammenligningsgrunnlaget er likt. Resultatene fra dette kapitlet vil derfor bli framstilt som forsinkelse på de samme strekningene som 0-alternativet, se figur 34



Figur 34-Oversiktskart, strekning

Simuleringen ble gjort iht. kapittel 6.3.4 og resultatene vises i figur 35. Som man ser utfra figuren får man følgende forsinkelser:

Count: 42	SimRun	TimeInt	DelayMeasurement	StopDelay(All)	Stops(All)	VehDelay(All)
30	10	240-3600	3: Olav Vs Gate-fv509	200.16	6.12	281.46
31	Average	240-3600	1: Bergeland-E39	391.89	14.14	556.30
32	Average	240-3600	2: Løkkevein-fv509	20.80	3.27	75.75
33	Average	240-3600	3: Olav Vs Gate-fv509	204.65	6.10	283.09

Figur 35-Resultat, reisetid

- Strekning 1: Fv. 509-Vestgående retning, 556 sekunder.
- Strekning 2: Løkkeveien-Sørgående retning, 76 sekunder.
- Strekning 3:Olav Vs gate-Sørgående retning, 283 sekunder.





## 9 RESULTAT OG DISKUSJON

---

Som nevnt i kapittel 1 er hovedmålet med denne oppgaven å øke min kunnskap rundt mikrosimulering og VISSIM. For å nå dette målet har jeg tatt utgangspunkt i en case med en problemstilling. Resultatene fra denne casen bør derfor betraktes som et resultat av selve metoden, selv om resultatene framstilles som svaret på en konkret oppgave.

Selv om VISSIM og aspektene rundt mikrosimulering er hovedmålet med oppgaven, er aspektene og resultatene fra casen også diskuteres i dette kapitlet.

### 9.1 Resultat fra casen

Resultatene som presenteres, bygger på trafikkdata som er mottatt av Statens vegvesen.

Resultatene fra en mikrosimuleringsmodell kan være mange. I denne oppgaven defineres resultatene som den gjennomsnittlige forsinkelsen på tre definerte strekninger over en rushtidsperiode. På grunn av modellens natur går noe av simuleringsperioden bort til oppvarming av modellen. Registreringsperioden er dermed satt til 3360 sekunder. Måling av forsinkelse er gjort av programmet og blir presentert i form av en gjennomsnittsverdi av flere simuleringer. De aktuelle strekningene hvor forsinkelsen ble beregnet kan man se utfra figur 36.



Figur 36-Oversiktskart, reisetid



Strekning	Lengde	Forsinkelse i sekunder		
		0-alternativet	Alternativ med midtstilt kollektivfelt	Differanse
Strekning 1: Fv. 509-Vestgående retning,	810m	110	556	446
Strekning 2: Løkkeveien-Sørgående retning	175m	49	75	26
Strekning 3:Olav Vs gate-Sørgående retning	210m	62	283	221

Tabell 1-Sammenstilling av reisetid

Resultatene fra 0-alternativet og alternativet med midtstilt kollektivfelt er sammenstilt i tabell-1. Utfra tabellen ser man tydelig at forsinkelsen øker ved å innføre midtstilt kollektivfelt.

## 9.2 Diskusjon av resultatene

Da området i dag opplever avviklingsproblem og forsinkelser, var det på forhånd antatt at forsinkelsen ville øke ytterligere ved å innføre midtstilt kollektivfelt med bussprioritering gjennom kryssene. Dette viser også resultatene som er hentet ut fra modellen. Hvor realistiske disse resultatene er, gjenstår nå å evaluere. For å bedre kunne gjøre det kan man se tilbake på kapittel 6.5.3 som omhandler validering av 0-alternativet. Her ser man at avviket mellom beregna reisetid og observert reisetid er på 36 %. Dette er for høyt til at jeg ville akseptert dette som en validert modell. Man må ta i betraktning at resultatene fra modellen bygger på trafikk tall fra 2006, og valideringen er gjort med trafikkregisteringer fra 2015. Når dette er sagt, har jeg tidligere beskrevet at man anser trafikk tallene fra 2006 for akseptable, da trafikk tellinger i Bergelandstunellen har vist en stabil trafikk mengde siden 1998. Men, 2 uker før denne oppgaven skulle leveres, kom vegvesenet med nye trafikk tall for Kannik. Kryss 1,2,3 og 4 hadde blitt registrert på ny. Registreringene var blitt gjort i den samme perioden som jeg registrert reisetider for bruk til validering. Disse dataene viste at trafikken i Kannik var betydelig redusert. Dette bekreftet også vegvesenet da jeg spurte om dette kunne skyldes lokale variasjoner. Hva denne reduksjonen skyldes er for såvidt irrelevant for oppgaven, men viss jeg skal gjette er det på grunn av Ryfastutbyggingen ved kryss 1. Ettersom jeg mottok disse nye trafikk dataene så kort tid før levering av oppgaven, valgte jeg å ikke bruke resurser på å gjøre nye beregninger basert på disse. Dersom jeg hadde foretatt ny beregninger ville man fått et annerledes resultat fra valideringsprosessen. Dette resultatet ville etter all sannsynlighet vist et mindre avvik mellom beregna og observert reisetid. Hvor mye dette vil utgjøre er uvisst da jeg ikke har fått tid til å studere tallene mer inngående. Utfra denne nye informasjonen vil jeg si at reisetiden fra 0-alternativet antagelig er noe høyere en den man kan observere i området i dag. Hvor mye er litt usikkert, da jeg ikke har kjørt nye beregninger med trafikk data fra 2015.

Resultatene fra alternativet med midtstilt kollektivfelt er derimot forbundet med større usikkerheter. Dette kunne man og observere i kalibreringsfasen. Med usikkerheter mener jeg ikke programmets evne til å gjengi forsinkelse, men usikkerhet i form av å ha funnet den mest optimale løsningen for dette alternativet, og hvor godt alternativet klarer å gjenskape det framtidige trafikkbilde. Et av de største usikkerhetsmomentene er trafikk tallene. Det er lite sannsynlig at de trafikk tallene jeg har brukt vil være gjeldende ved et slikt alternativ. Sannsynligvis vil trafikken reduseres som følge av Ryfast utbyggingen, mindre kapasitet på vegnettet og bedre kollektivtilbud. Ett annet element som skaper usikkerhet i resultatene er utformingen av signalanleggene, og da spesielt anlegget i kryss 3. Som tidligere nevnt gjenstår det en del arbeid med å optimalisere dette. Den største svakheten med dette signalanlegget er at alle felt får rødt når bussen kommer. Modellens kapasitet kunne vært forbedret dersom flere felt fikk grønt samtidig som bussen. Man kan og diskutere hvorvidt fordelingen av grønttid burde vært fordelt annerledes. Etter annet aspekt, som jeg ikke har tatt med i modellen, er hvordan resultatene påvirkes av fotgjengere. I en virkelig situasjon ville det vært fotgjengere til og fra holdeplassene. Dette ville redusert kapasiteten ytterligere, spesielt dersom man hadde gått for en løsning med lysregulering.

Viss man studerer onlinedataene, valideringsresultatene og feilmeldingsfilen, ser man at det er store forsinkelser i nettverket, og at noe av denne forsinkelsen kunne vært redusert ved å bruke mer resurser på å optimalisere signalanleggene. En del av forsinkelsen skyldes også at man fjerner to felt på et allerede overbelasta vegnett. Hvor mye dette utgjør, er vanskelig å si så lenge man ikke har en optimal modell. På grunnlag av dette vil jeg si at resultatene fra tiltaket viser dårligere avvikling enn tiltakets potensial.

### **9.3 Diskusjon og erfaringer med VISSIM**

Etter å ha arbeidet med VISSIM i noen måneder har jeg fått et visst inntrykk av programmet. I begynnelsen syntes jeg det var ganske tungvint å finne fram i menyene, og at programmets oppbygning var lite intuitiv. Det skjedde mange ganger at jeg gikk meg vill i menyene og ikke kunne finne de funksjonene jeg var på jakt etter. Dette har selvsagt mye å gjøre med erfaring, men jeg mener enda at programmet kunne vært mer brukervennlig. Programmet har en oppbygning som gjør at man kan beskrive de fleste elementene på et ganske detaljert nivå. Dette kan og være noe av grunnen til at programmet kan virke litt uoversiktlig.

Det er to elementer ved programmet jeg vil trekke fram. Det første er hvor enkelt og fleksibelt det er å modellere nettverket. Man har kun to elementer å forholde seg til, links og connectors. Disse er svært fleksible og gir deg mulighet til å modellere de fleste løsninger. Et annet

verktøy som jeg hadde stor nytte av var «conflict area». Denne funksjonen markerer alle steder det oppstår en konflikt mellom kjøretøystrømmer, på en tydelig visuell måte. Den setter opp alle konfliktene i en tabell slik som vist i figur 13. Dersom man trykker på en av radene i tabellen markeres konflikten i nettverket, og man kan enkelt justere verdien i tabellen. Med denne funksjonen er det lett å få oversikt og justere konfliktområdets parametere.

Dersom jeg skal trekke fram noe negativt, må det være å styre trafikken ved hjelp av statiske ruter. Det er en operasjon som er ganske resurskrevende og enkel å gjøre feil. I denne oppgaven opplevde jeg redusert kapasitet i modellen som følge av å bruke statiske ruter kombinert med kort avstand mellom kryssene. For å kompensere for dette måtte jeg øke aggressiviteten til sjåførene ved å endre på parameter i Lane change-modellen. Selv om dette reduserte problemet, er det knytte usikkerhet til de nye parameterverdiene og om disse endringene vil påvirke modellen på en måte som jeg ikke har klart å registrere. Disse problemene hadde blitt redusert med bruk av OD-matriser, men dette er tilleggsmoduler i VISSIM som jeg ikke hadde tilgang til i denne oppgaven.

Som fryktet på forhånd gav modellering av rundkjøring meg store utfordringer. Her brukte jeg masse tid og resurser, men skal man drive med trafikksimulering i Norge er det avgjørende å kunne modellere rundkjøringer. I denne oppgaven vil jeg påstå at byggingen av selve rundkjøringen var en vanskeligere jobb enn å kalibrere konfliktområdene i forbindelse med rundkjøringene. Dette skyldes hovedsakelig at jeg ikke hadde erfaring med programmet, og i tillegg er kryss 1, 2 og 3 utformet som 3-felts rundkjøringer med 3-felts tilfarer. Under prosessen med å bygge rundkjøringen fant jeg ut at det finnes flere «veger til målet» og at det ikke alltid finnes en fasit på hvilke og hvor mange elementer en rundkjøring skal bestå av. Det finnes sikkert flere og bedre måter å bygge en rundkjøring på, men etter litt prøving og feiling kom jeg fram til en løsning som gjengav et akseptabelt trafikkbilde og en brukbar avvikling. Jeg erfarte at det kan være en fordel å ha så få elementer som mulig, da flere elementer gir mange konfliktområder, som igjen kan føre til en defensiv kjøreradferd.

Noe jeg kunne brukt mer tid på var å eksperimentere med verdiene på front og rear gap i «conflict area»-verktøyet. Her brukte jeg bare verdier fra et eksempel fra PTV, uten å kjøre en følsomhetsanalyse på disse verdiene. Når jeg jobbet med kalibrering av 0-alternativet observerte jeg problemer med feltskifte i animeringen. I tillegg viste feilmeldingsfilen fra de første simuleringene at 20-30 biler ble fjernet i løpet av simuleringen som følge av at det hadde blitt klemt inne og ikke fått utført feltskifte. I tillegg til parameterinnstillinger, var mye av grunnen til dette at jeg ikke brukte OD-matriser for å styre trafikken, og at det var kort

avstand mellom kryssene. Etter litt prøving og feiling og hjelp fra veileder klarte jeg å komme fram til verdier som gjorde at de fleste av disse problemene forsvant. Feilmeldingsfilen viste nå at programmet ikke hadde fjernet noen biler som følge av feltskifteproblemer.

Som nevnt i det forgående kapittelet viser valideringsdataene et avvik på observert og beregna reisetid på 35 %. Det nevnes også at denne differansen vil reduseres dersom man gjør nye simuleringer med trafikkdata fra 2015. En annen indikasjon på modellens kvalitet er å se i feilmeldingsfilen. Den forteller hvor mange biler modellen ikke klarte å avvikle i løpet av simuleringsperioden, altså hvor mange biler som ikke slapp inn i modellen etter endt simulering. Ved et 0-alternativ skal modellen ideelt sett klare å avvikle alle kjøretøyene, gitt at simuleringsperioden er den samme som registreringsperioden. Noen stokastiske avvik vil forekomme slik at man må foreta individuelle vurderinger. Feilmeldingsfilen til 0-alternativet viste at modellen i noen grensesnitt ikke klarte å avvikle all trafikk. Det er ikke snakk om store verdier, men for store til å være stokastiske avvik. Etter en helhetlig vurdering av onlinedataene, valideringsresultatene og feilmeldingsfilen, vil jeg si at modellen beskriver noe dårligere kapasitet enn i virkeligheten.

Modelleringen av 0-alternativet følte jeg hovedsakelig ble en øvelse i å lære meg å modellere kjøretøystyrte signalanlegg og få en innføring av VisVAP-programmet. VisVAP er et utrolig fleksibelt program som lar deg modellere de fleste ITS tiltak, men til gjengjeld har det et høyt brukersnitt. Hele prosessen med å lage kjøretøystyrte signalanlegg og ITS-tiltak synes jeg er en svært tungvint prosess i VISSIM. For det første er kodingen av pua- og vap-filene en ganske omfattende og en lite brukervennlig operasjon. I tillegg må disse filene være 100 % korrekt med tanke på store og små tegn, riktig antall mellomrom, rett bruk av komma osv. osv. Dette medfører at det går en god del tid på feilsøking og studering av txt-filer og flowcharts dersom man ikke klare det på første forsøk. I denne oppgaven modellerte jeg et forholdsvis enkelt kjøretøystyrt signalanlegg. Det var allikevel det som gav meg størst utfordring i forbindelse med denne oppgaven.

Signalanleggene i forbindelse med rundkjøringene synes jeg ble modellert til et akseptabelt nivå. Disse modelleres mer eller mindre som et detektorstyrt signalanlegg ved et gangfelt. Den største utfordringen ved disse var å justere detektorene slik at bussen ankom rundkjøringen i det den siste bilen kjørte ut av krysset. Noen ganger, spesielt i kryss 1, opplevde jeg at køen stod gjennom rundkjøringen og bussen ble forsinket av dette. Dette kunne vært unngått ved å trekke detektoren lengre bort fra krysset, men da hadde også kapasiteten på vegnettet blitt redusert. I kryss 3 gjenstår det også noe arbeid for å få krysset til å fungere optimalt.

Signalanleggets prinsipp er ganske enkelt. Anlegget består av 3 faser. Fase 1 og 2 styrer den ordinære trafikken, mens fase 3 styrer kollektivtrafikken. Fase 1 og 2 skifter med faste intervall, på henholdsvis 50 og 5 sekunder. Når bussen detekteres får alle felt i fase 1 og 2 rødt, og fase 3 får grønt. Planen var at når fase 3 var ferdig, skulle anlegget begynne på fase 2, for å sikre at fase 2 fikk minimum 5 sekunder grønt pr. omløp. Sannsynligvis på grunn av feilkoding gikk det ikke som planlagt, og man opplever omløp der fase 2 får mindre en 5 sekunder grøntid. Den største jobben som gjenstår er imidlertid å programmere signalanlegget til å gi grønt til trafikkstrømmer som ikke kommer i konflikt med fase 3. Dette vil øke avviklingen i krysset betraktelig. Da jeg begynte å programmere dette signalanlegget innså jeg tidlig at dette var en for resurskrevende jobb for denne oppgaven. Det hender også at det av og til står kø gjennom krysset når fase 3 får grønt. Dette medfører at bussen stopper opp og må vente på ledige luke. Her kunne det blitt gjort noe mer arbeid med å justere på detektorene og gjort noen nettverksjusteringer for å unngå disse situasjonene. Som forventet viser onlinedataene, valideringsresultatene og feilmeldingsfilen at dette alternativet avviker den ordinære trafikken betraktelig dårligere en ved 0-alternativet. Noe av dette skyldes at man reduserer et allerede belasta område med to felt, men også at signalanleggene kunne vært optimalisert med tanke på avvikling. Selv om det gjenstår en del arbeid med å optimalisere signalanleggene føler jeg at jeg har lært det grunnleggende i det å modellere kjøretøystyrte signalanlegg.

For å oppsummere er VISSIM et utrolig fleksibelt program som gir mulighet til å modellere de fleste situasjoner på det detaljnivået man måtte ønske. Der er ingen tvil om at jeg kunne gått grundigere inn i både case og program, men en slik oppgave har begrensa rammer, og det lar seg dessverre ikke gjøre å få gjort alt. Det finnes mange aspekter med programmet jeg ikke har fått utforske, som for eksempel å modellere fotgjengerstrømmer. Det gjenstår en del arbeid før modellene er optimale, men jeg føler allikevel at jeg har lært meg de mest grunnleggende aspektene med programmet og prosessen.

# 10 KONKLUSJON OG VIDERE ARBEID

---

Denne oppgaven er på mange måter todelt. Den består av målsetning og en problemstilling. I dette kapittelet beskrives konklusjonen og videre arbeid for begge to.

## 10.1 Konklusjon

Konklusjonen på denne oppgaven vil være todelt. Det fins en konklusjon på målsetningen og en for casens problemstilling.

### 10.1.1 Målsetningens konklusjon

Hensikten med denne oppgaven var jeg skulle styrke min kunnskap om teorien bak mikrosimuleringsmodeller, lære det grunnleggende i VISSIM, og hvordan man gjennomfører et simuleringsoppdrag.

Gjennom litteratursøk har jeg fått et innblikk i teorien og historien bak mikrosimuleringsmodeller. Kunnskap om hvordan man gjennomfører en simulering i praksis har jeg opparbeidet meg gjennom å studere litteraturen og ved å simulere en problemstilling i en gitt case.

I denne oppgaven har jeg erfart at dersom man skal lære seg VISSIM er det en fordel å lære seg teorien bak programmet, men også bruke programmet aktivt på en problemstilling eller ved å gjenskape dagens situasjon. Ved å simulere 0-alternativet har jeg fått ett innblikk i hvilke utfordringer og usikkerheter man møter når man utfører trafikksimuleringer. Jeg har lært meg de grunnleggende elementene om hvordan man bygger opp en simuleringsmodell i VISSIM. Videre har jeg opparbeidet meg en viss erfaring av hvilke parametere og parameterverdier som er avgjørende for resultatet.

I løpet av oppgaven har jeg og fått erfaring og forståelse rundt prosessen med å kalibrere og validere modeller. Dette er et viktig arbeid som jeg visste lite om da jeg begynte på oppgaven.

I følge litteraturen er det å kunne modellere ITS-tiltak svært sentralt når det kommer til mikrosimulering. Dette var ikke planlagt da jeg startet på oppgaven, men gjennom denne oppgaven har jeg fått en oversikt over hvordan dette fungerer. Ved å modellere et ITS-tiltak i form av et signalanlegg med bussprioritering fikk jeg og et innblikk i VisVAP programmet. Utformingen av signalanlegget kunne vært mer optimal, men ut fra tidsaspektet på oppgaven er jeg fornøyd med de erfaringer jeg gjorde.

Jeg startet på denne oppgaven med et håp om å kunne skaffe meg et godt utgangspunkt for å utvikle meg videre innen trafikksimulering. Gjennom oppgaven har jeg fått en introduksjon til de viktigste aspektene med mikrosimulering, og opparbeidet meg en forståelse av prosessen rundt det å utføre et simuleringsoppdrag. Jeg har og lært meg de grunnleggende elementene i VISSIM, og dermed fått et godt utgangspunkt for min videre utvikling innen faget.

### **10.1.2 Casens konklusjon**

Casens problemstilling var å finne et tiltak som gjorde at bussen kunne kjøre gjennom Kannik uten forsinkelse. Da jeg i denne oppgaven kun har sett på et tiltak, kan man ikke konkludere med at midtstilt kollektivfelt med de foreslåtte kryssløsningene vil være det beste tiltaket for Kannik. Jeg vil likevel konkludere med at det er mulig å oppfylle casens mål med det foreslåtte tiltaket. Dette vil selvfølgelig gå på bekostning av kapasiteten på det øvrige vegnettet, og gi økte ulemper for bilistene. Slik jeg ser det i dag vil det ikke være mulig å få innført et kollektivtiltak der bussen har 100 % prioritet gjennom Kannik, uten å gi biltrafikken økt forsinkelse. Dette er igjen et strategisk spørsmål som jeg ikke vil gå nærmere inn på. Som nevnt tidligere er resultatene fra 0-alternativet ikke så langt unna virkeligheten, men viser sannsynligvis litt for mye forsinkelse. Resultatene jeg har kommet fram til for tiltaket med midtstilt kollektivfelt kan ikke sies å være de endelige. For å få resultater som representerer den optimale versjonen av det foreslåtte tiltaket må det gjøres en del mer arbeid, spesielt med tanke på signalanlegget i kryss 3.

## **10.2 Videre arbeid**

I en slik oppgave er det begrensa tidsrammer og det vil alltid være mer man kunne gjort. I forbindelse med denne oppgaven er det elementer det kan jobbes videre med, både på et personlig plan og et fagmessig plan.

### **10.2.1 Mikrosimulering**

For å kunne gjennomføre trafikksimuleringer som gir pålitelige resultater er det avgjørende å opparbeide seg erfaring og kunnskap. Noe kunnskap skaffes ved litteraturstudie, men erfaringen opparbeider man seg hovedsakelig med bruk av programmet på reelle oppdrag. Et viktig arbeid som kan gjøres for å bli bedre innen trafikksimulering er å utføre følsomhetsanalyser på de forskjellige parameterne i programmet. Dette gjør at man får en føling av parameternes betydning og påvirkning av modellen. Dette fører igjen til at man enklere klarer å forme modellen slik man ønsker.

Et verktøy som jeg fikk utforsket litt i denne oppgaven er VisVAP. Etter min mening er det avgjørende å beherske dette verktøyet dersom man skal kunne utføre gode trafikksimuleringer med VISSIM. Programmet har en enorm fleksibilitet, men også et høyt brukersnitt. Å jobbe mer med dette verktøyet vil derfor være sentralt for å styrke min kompetanse innen trafikksimulering med VISSIM.

Kalibrering og validering av modeller som ikke representerer et 0-alternativ vil alltid være en utfordring. I løpet av denne oppgaven har jeg lest mye litteratur om forskjellige temaer innen mikrosimulering. Et av temaene er kalibrering og valideringsteknikker. Siden dette er to sentrale elementer i det å utføre en trafikksimulering er det også naturlig å studere denne litteraturen mer inngående.

### **10.2.2 Casen**

Dersom man antar at midtstilt kollektivfelt gjennom Kannik er den beste måten å løse casens problemstilling på, er det enda noe arbeid som gjenstår for å få en optimal modell. Det første arbeidet som må gjøres er å beregne nye trafikk tall med tanke på Ryfastutbyggingen, at kapasiteten i Kannik reduseres, og at regionen får et bedre kollektivtilbud.

Det gjenstår også en del arbeid med å få optimalisert signalanlegget i kryss 3. I tillegg kan det gjøres noe mer jobb på å optimalisere utformingen av krysset med tanke på feltinndeling, geometri osv.





# REFERANSELISTE

- Barceló, J., 2010. Models, Traffic Models, Simulation and Traffic Simulation. In: J. Barceló, ed. *Fundamentals of Traffic Simulation*. Barcelona: Springer, p. 450.
- Brackstone, M. & McDonald, M., 1999. Car-following: a historical review. In: *Transportation Research Part F: Traffic Psychology and Behaviour*. s.l.:s.n., pp. 181-196.
- Fellendorf, M. & Vortisch, P., 2010. Microscopic Traffic Flow Simulator VISSIM. In: *Fundamentals of Traffic Simulation*. s.l.:Springer, p. 450.
- FHWA, 2004. Guidelines for Applying Traffic Microsimulation Modeling Software. In: *Traffic Analysis Toolbox Volume III*. s.l.:US. departemt of transportation v/The Federal Highway Administration, p. 148.
- Gipps, P. G., 1986. *A MODEL FOR THE STRUCTURE OF LANE-CHANGING DECISIONS*, Melbourne: s.n.
- Hidas, P., 2002. *Modelling lane changing and merging in microscopic traffic simulation*, s.l.: s.n.
- Hidas, P., 2002. *test*, s.l.: s.n.
- Hidas, P., 2005. *Modelling vehicle interactions in microscopic simulation of merging and weaving*, s.l.: s.n.
- Hoogendoorn, S. P., 2003. *Traffic Flow Theory and Simulation*, s.l.: s.n.
- IEEE Computational Intelligence Society , 2011. *Youtube*. [Online] Available at: <https://www.youtube.com/watch?v=P8wY6mi1vV8> [Accessed 06 04 2015].
- Jeppesen, B., 2010. *Vissimsimulering af bytrafik efter danske forhold*, Ålborg: s.n.
- Jin, S., Wang, D., Tao, P. & Li, P., 2010. Non-lane-based full velocity difference car following model. In: *Physica A: Statistical Mechanics and its Applications*. s.l.:Elsevier, p. 9.
- Shaw, J. W. & Noyce, D. A., 2015. *Are Microsimulation Models Random Enough?*, s.l.: TRB.
- SUN, D., 2009. *A LANE-CHANGING MODEL FOR URBAN ARTERIAL STREETS*, Florida: s.n.
- Vejregelrådet, 2010. *ANVENDELSE AF MIKROSIMULERINGSMODELLER*, s.l.: Vejdirektoratet.
- WENG, Y.-I. & WU, . T.-j., 2001. *Car-following models of vehicular traffic*, s.l.: Springer.



# 11 VEDLEGG OG BILAG

---

## 11.1 Vedlegg

- VISSIM fil: Kannik med 0-alternativet
- VISSIM fil: Kannik med tiltak

## 11.2 Bilag

- 1) Oppgavetekst.
- 2) Trafikkdata.
- 3) Reisetidsregistreringer og resultater.

## 11.3 Bilag 1

### **MASTEROPPGAVE**

(BA6904, masteroppgave studieretning veg)

HØST 2014 / VÅR 2015

for

**Atle Kristian Svela**

Mikrosimulering av trafikk – teori og praksis

#### **BAKGRUNN**

Det er en utfordrende prosess å finne fram til gode transportløsninger i byer og tettsteder. Det stilles økende krav til både effektivitet, sikkerhet, miljø og tilgjengelighet for alle trafikantgrupper. Dette forutsetter at planleggeren har gode verktøy for å vurdere konsekvenser av og sammenligne ulike tiltak.

I de seinere år har det blitt mer vanlig å benytte modeller som baserer seg på mikrosimulering av trafikk. Dette er kompliserte modeller som setter store krav til brukeren.

#### **OPPGAVE**

Målsettingen med oppgaven er at kandidaten skal gjøre seg kjent med teorien bak mikrosimuleringsmodeller og prosessen rundt det å utføre et mikrosimuleringsoppdrag. Dette inkluderer å etablere en simuleringsmodell for et område og benytte denne modellen for å vurdere en aktuell problemstilling.

Statens vegvesen og Stavanger kommune har en målsetting om at bussene skal kunne kjøre fra Byterminalen gjennom Kannikområdet til E-39 med minst mulig forsinkelse. Dette forutsetter en omfattende prioritering av bussen i forhold til annen trafikk. Den aktuelle problemstillingen i oppgaven er å foreslå et tiltak for å prioritere busstrafikken i området og benytte simuleringsmodellen VISSIM til å vurdere konsekvenser av tiltaket for ulike trafikantgrupper.

Oppgaven kan deles i 4 deler:

I den første delen skal kandidaten gi en generell beskrivelse av simuleringmodeller for biltrafikk. Dette inkluderer blant annet bakgrunn for slike modeller, teoretisk grunnlag, anvendelse, inngangsdata og resultater.

I den andre delen skal kandidaten gi en innføring i de mest grunnleggende elementene og parameterne i simuleringmodellen VISSIM.

I den tredje delen skal kandidaten beskrive dagens situasjon og utfordringer i Kannikområdet. Videre skal det bygges opp en VISSIM modell for å beskrive dagens trafikksituasjon. Denne modellen skal i rimelig utstrekning sammenlignes med avviklingen i området.

I den fjerde og siste delen skal kandidaten beskrive et tiltak for å prioritere busstrafikken i området. Dette tiltaket skal modelleres i VISSIM med tanke på å vurdere konsekvensene av tiltaket både for busstrafikken og for andre trafikantergrupper.

## GENERELT

Oppgaveteksten er ment som en ramme for kandidatens arbeid. Justeringer vil kunne skje underveis, når en ser hvordan arbeidet går. Eventuelle justeringer må skje i samråd med faglærer ved instituttet.

Vekt på de ulike delene samt struktur på oppgaven kan vurderes av kandidaten i samråd med faglærer og veileder.

Ved bedømmelsen legges det vekt på grundighet i bearbeidingen og selvstendigheten i vurderinger og konklusjoner, samt at framstillingen er velredigert, klar, entydig og ryddig uten å være unødig voluminøs.

Besvarelsen skal inneholde

- standard rapportforside (automatisk fra DAIM, <http://daim.idi.ntnu.no/>)
- tittelside med ekstrakt og stikkord (mal finnes på siden <http://www.ntnu.no/bat/skjemabank>: 3) Om Masteroppgaven)
- sammendrag på norsk og engelsk (studenter som skriver sin masteroppgave på et ikke-skandinavisk språk og som ikke behersker et skandinavisk språk, trenger ikke å skrive sammendrag av masteroppgaven på norsk)
- hovedteksten
- oppgaveteksten (denne teksten signert av faglærer) legges ved som Vedlegg 1.

Råd og retningslinjer for masteroppgaven finnes på programmets nettsider.

[http://videre.ntnu.no/pages/mastergrader/erfaringsbasert\\_masterprogram\\_i\\_veg\\_og\\_jernbane/priser\\_og\\_betingelser/](http://videre.ntnu.no/pages/mastergrader/erfaringsbasert_masterprogram_i_veg_og_jernbane/priser_og_betingelser/)

### Hva skal innleveres?

Rutiner knyttet til innlevering av masteroppgaven er nærmere beskrevet på <http://daim.idi.ntnu.no/>. Trykking av masteroppgaven bestilles via DAIM direkte til Skipnes Trykkeri som leverer den trykte oppgaven til instituttkontoret 2-4 dager senere. Instituttet betaler for trykkingen, og 1 eksemplar blir sendt til studenten. Ekstra eksemplarer må bekostes av kandidaten/ ekstern samarbeidspartner.

Videre skal kandidaten levere innleveringsskjemaet (fra DAIM) Innleveringsskjema sendes til NTNU VIDERE.

Dokumentasjon som med instituttets støtte er samlet inn under arbeidet med oppgaven skal leveres inn sammen med besvarelsen.

Besvarelsen er etter gjeldende reglement NTNUs eiendom. Eventuell benyttelse av materialet kan bare skje etter godkjenning fra NTNU (og ekstern samarbeidspartner der dette er aktuelt). Instituttet har rett til å bruke resultatene av arbeidet til undervisnings- og forskningsformål som om det var utført av en ansatt. Ved bruk ut over dette, som utgivelse og annen økonomisk utnyttelse, må det inngås særskilt avtale mellom NTNU og kandidaten.



**Helse, miljø og sikkerhet (HMS):**

NTNU legger stor vekt på sikkerheten til den enkelte arbeidstaker og student. Den enkeltes sikkerhet skal komme i første rekke og ingen skal ta unødige sjanser for å få gjennomført arbeidet. Studenten skal derfor ved uttak av masteroppgaven få utdelt brosjyren "Helse, miljø og sikkerhet ved feltarbeid m.m. ved NTNU".

Dersom studenten i arbeidet med masteroppgaven skal delta i feltarbeid, tokt, befaring, feltkurs eller ekskursjoner, skal studenten sette seg inn i "Retningslinje ved feltarbeid m.m.". Dersom studenten i arbeidet med oppgaven skal delta i laboratorie- eller verkstedarbeid skal studenten sette seg inn i og følge reglene i "Laboratorie- og verkstedhåndbok". Disse dokumentene finnes på fakultetets HMS-sider på nettet, se <http://www.ntnu.no/ivt/adm/hms/>. Alle studenter som skal gjennomføre laboratoriearbeid i forbindelse med prosjekt- og masteroppgave skal gjennomføre et web-basert TRAINOR HMS-kurs. Påmelding på kurset skjer til [daniel.erland@ntnu.no](mailto:daniel.erland@ntnu.no)

Studenter har ikke full forsikringsdekning gjennom sitt forhold til NTNU. Dersom en student ønsker samme forsikringsdekning som tilsatte ved universitetet, anbefales det at han/hun tegner reiseforsikring og personskadeforsikring. Mer om forsikringsordninger for studenter finnes under samme lenke som ovenfor.

**Oppstart og innleveringsfrist:**

Frist innlevering masterkontrakt **15. august**, frist innlevering masteroppgaven **15. mai**

**Hovedveileder ved NTNU:** Arvid Aakre

**Lokal veiledere:** Tarjei Østmøe, Statens vegvesen og Sebastian Nerem, Norconsult

Institutt for bygg, anlegg og transport, NTNU

Dato: 15.08.2014 (rev 10.05.2015)

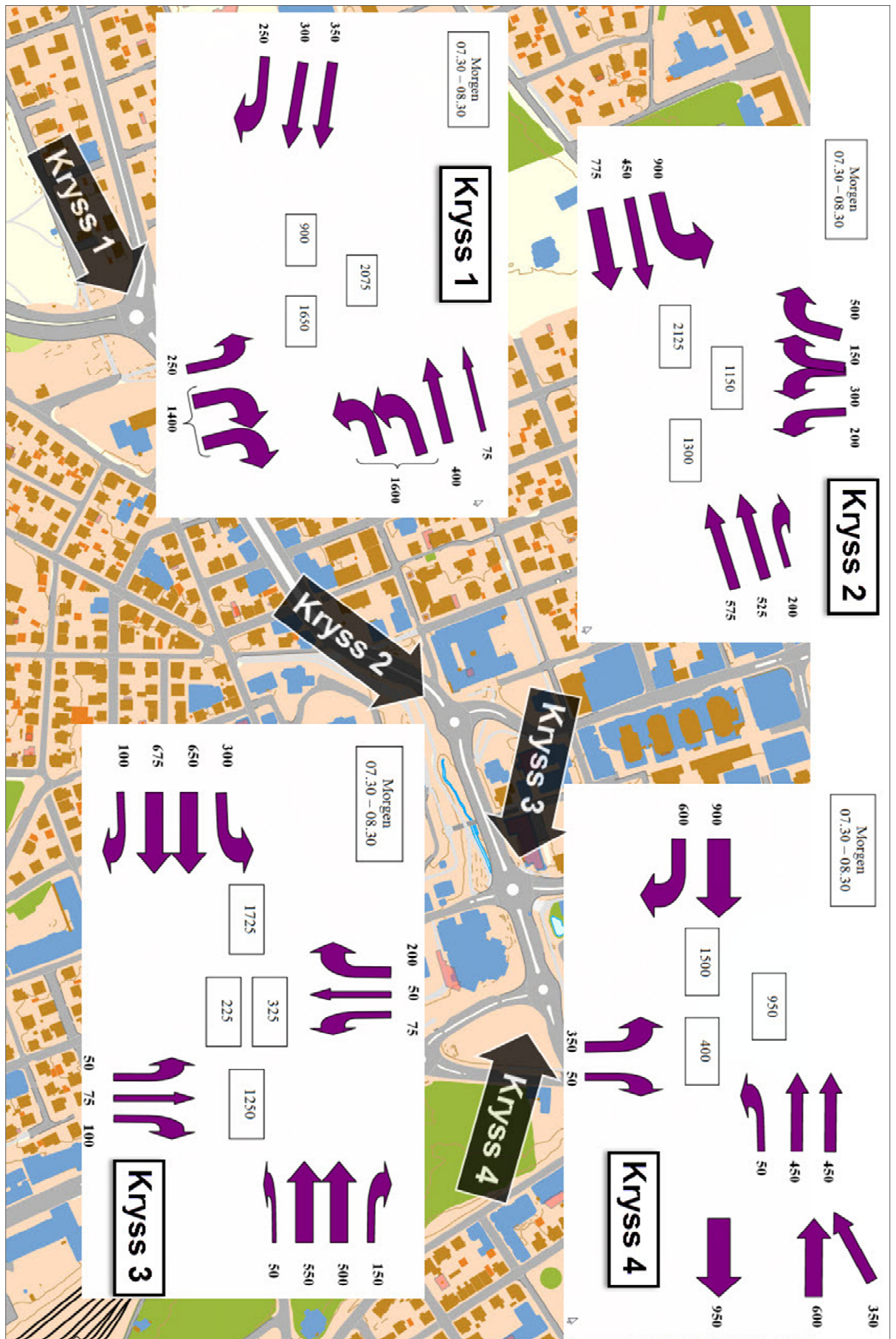
Underskrift



---

Faglærer

# 11.4 Bilag 2



Figur 37-Trafikktall, 2006

## 11.5 Bilag 3

Registrerte reisetider, ref kapittel 6.5.3

Dato for registrering:2015-03-17	
Tidspunkt	Antall sekunder
KI 0730	89
KI 0745	81
KI 0800	60
KI 0815	80
KI 0830	50
Dato for registrering:2015-04-15	
Tidspunkt	Antall sekunder
KI 0730	94
KI 0745	63
KI 0800	78
KI 0815	86
KI 0830	73
Gjennomsnitt	75.4

*Tabell 2-Reisetidsregistreringer*

7	Average	600-3600	4: Bergeland-Løkkeveien	354	101.92
8	Standard deviasi	600-3600	4: Bergeland-Løkkeveien	13	11.03
9	Minimum	600-3600	4: Bergeland-Løkkeveien	339	87.45
10	Maximum	600-3600	4: Bergeland-Løkkeveien	375	112.49

*Figur 38-Beregna reisetid*