

Kjetil Brådland Aamodt

# Modellering av hastighet til tunge kjøretøy i stigninger

Masteroppgave i bygg- og miljøteknikk

Veileder: Arvid Aakre

Juni 2022

**NTNU**  
Norges teknisk-naturvitenskapelige universitet  
Fakultet for ingeniørvitenskap  
Institutt for bygg- og miljøteknikk





Kjetil Brådland Aamodt

# **Modellering av hastighet til tunge kjøretøy i stigninger**

Masteroppgave i bygg- og miljøteknikk

Veileder: Arvid Aakre

Juni 2022

Norges teknisk-naturvitenskapelige universitet

Fakultet for ingeniørvitenskap

Institutt for bygg- og miljøteknikk



Kunnskap for en bedre verden



# Forord

Denne masteroppgaven er skrevet ved Norges teknisk-naturvitenskapelige universitet (NTNU) i Trondheim ved Institutt for bygg- og miljøteknikk. Forarbeid ble gjennomført høsten 2021 i form av en prosjektoppgave, og masteroppgaven er utført våren 2022. Oppgaven tilsvarer 30 studiepoeng i emnet TBA4940 Veg, masteroppgave, og markerer det avsluttende arbeidet for studiet. Hovedveileder for denne oppgaven har vært professor Arvid Aakre, leder for Trafikkteknisk senter ved NTNU.

Jeg vil rette en stor takk til Arvid for god hjelp, veiledning, inspirasjon og godt samarbeid gjennom arbeidet med dette prosjektet. Jeg ønsker også å takke avdelingsingeniør ved Institutt for bygg- og miljøteknikk Jan Erik Molde for god hjelp. Han har vært bindeleddet og lagt til rette for kommunikasjon med transportbransjen og de ulike transportselskapene.

Det hadde ikke vært mulig å gjennomføre denne oppgaven uten transportbransjen. Jeg vil derfor takke Frode Galåen fra Galåen transport, samt sjåførene fra Galåen Transport og Børstad Transport som var med på å legge til rette for arbeidet mitt.

Trondheim, 11. juni 2022



---

Kjetil Brådland Aamodt



# Sammendrag

Denne oppgaven har som formål å teste og videreutvikle en hastighetsmodell for tunge kjøretøy i stigninger. Modellen utarbeider et hastighetsprofil basert på data fra kjøretøyet og utformingen av stigningen. I forbindelse med utarbeidelsen av modellen, ble det gjennomført et begrenset antall forsøk. For å vurdere hvordan modellen fungerer, er det derfor nødvendig å gjennomføre ytterligere testing.

Gjennom oppgaven har det blitt utført flere forsøk med ulike kjøretøy fra Scania. To forskjellige strekninger ble kjørt flere ganger. Kjøretøyene varierte i motorstørrelse og vekt. Under forsøkene ble det brukt utstyr for å samle inn motordata og posisjonsdata. I etterkant ble dataene lagt inn på datamaskinen og brukt i hastighetsmodellen.

Resultatene fra forsøkene viser at hastighetsmodellen produserer et realistisk fartsprofil. Ved en jevnere stigning vil modellen produsere et mer korrekt profil enn ved en mer ujevn stigning. Ved lavere stigningsprosent, er modellen litt optimistisk og øker hastigheten litt for mye. Selv om fartsprofilene fra modellen og fra virkeligheten ikke er identiske kurver, er tidsbruken opp stigningen og den oppnådde minimumshastigheten lik. Det er svært mange faktorer som påvirker hastigheten slik at en perfekt etterligning av virkeligheten ikke er mulig.

Resultatene tilsier at ytterligere arbeid med hastighetsmodellen er aktuelt. Videre er det mulig å gjøre modellen mer generell ved å gjennomføre forsøk med flere kjøretøy fra ulike kjøretøyprodusenter. En fungerende hastighetsmodell kan nyttiggjøres til flere formål. Et potensielt arbeid i fremtiden er å se på et dimensjonerende kjøretøy, og utforming av krabbefelt.

# Abstract

The purpose of this work is to test and further develop a speed model for heavy vehicles in grades. The model makes a speed profile based on data from the vehicle and the design of the grade. In the development of the model, there was conducted a limited amount of experiments. To evaluate the performance of the model, it is necessary to conduct further testing.

Throughout this project, there has been performed several experiments with different vehicles produced by Scania. Two different routes were tested multiple times. The vehicles varied in engine size and weight. During the experiments it was used equipment to gather engine specific data and GPS data. After the experiments, the data was uploaded to the computer and used in the speed model.

The results show that the speed model produces a realistic speed profile. A smooth grade produce a better profile than an uneven grade. At a lower gradient, the model is optimistic and the speed is increasing too much. Even though the speed profiles for the model and from reality are not identical curves, the amount of time used through the grade and the minimum speed achieved is similar. The speed is influenced by a lot of factors, therefore a perfect simulation of the reality is not possible.

Given the results, there is a possibility to continue working with the model. Further it is possible to make a more general model, by conducting more experiments with vehicles from different manufacturers. A functional speed model can be used for several purposes. A future potential is to work on a dimensioning vehicle, and design of crawler lane.





# Innhold

<b>Forord</b> . . . . .	<b>i</b>
<b>Sammendrag</b> . . . . .	<b>iii</b>
<b>Abstract</b> . . . . .	<b>iv</b>
<b>Innhold</b> . . . . .	<b>vi</b>
<b>Figurer</b> . . . . .	<b>ix</b>
<b>Tabeller</b> . . . . .	<b>xi</b>
<b>Akronymer</b> . . . . .	<b>xii</b>
<b>Ordliste</b> . . . . .	<b>xiv</b>
<b>1 Introduksjon</b> . . . . .	<b>1</b>
1.1 Bakgrunn . . . . .	1
1.2 Forskningsspørsmål . . . . .	2
1.3 Forprosjekt . . . . .	3
1.4 Rapportens oppbygning . . . . .	3
<b>2 Teori</b> . . . . .	<b>4</b>
2.1 Fysiske egenskaper . . . . .	4
2.1.1 Motor . . . . .	4
2.1.2 Motstand . . . . .	5
2.2 Dimensjonerende kjøretøy og vegutforming . . . . .	7
2.3 Data og loggerutstyr . . . . .	9
2.3.1 Posisjonsdata med VBOX . . . . .	11
2.3.2 Motordata med CANedge1 . . . . .	12
2.4 Eksisterende arbeider og modeller . . . . .	13
2.4.1 Børnes og Aakre . . . . .	13
2.4.2 Nesdal . . . . .	16

2.4.3	Kogstad og Nygaard . . . . .	18
<b>3</b>	<b>Metode . . . . .</b>	<b>20</b>
3.1	Forprosjekt . . . . .	20
3.2	Pilot med Scania fra Børstad Transport . . . . .	22
3.3	Datainnsamling . . . . .	24
3.3.1	Posisjonsdata . . . . .	24
3.3.2	Motordata . . . . .	27
3.4	Forsøk med Galåen Transport . . . . .	30
3.4.1	Scania R580 . . . . .	30
3.4.2	Scania S650 . . . . .	32
3.4.3	Scania S730 . . . . .	34
3.5	Hastighetsmodellen . . . . .	36
<b>4</b>	<b>Resultat . . . . .</b>	<b>43</b>
4.1	Trondheim-Røros . . . . .	43
4.1.1	Okstad . . . . .	44
4.1.2	Ålen . . . . .	47
4.2	Verdal-Trondheim . . . . .	51
4.2.1	Saltøya . . . . .	51
4.2.2	Sveberg . . . . .	52
4.2.3	Vikhammer . . . . .	54
<b>5</b>	<b>Diskusjon . . . . .</b>	<b>56</b>
5.1	Trondheim-Røros . . . . .	56
5.1.1	Okstad . . . . .	57
5.1.2	Ålen . . . . .	58
5.2	Verdal-Trondheim . . . . .	60
5.2.1	Saltøya . . . . .	60
5.2.2	Sveberg . . . . .	60
5.2.3	Vikhammer . . . . .	61
5.3	Generelt . . . . .	62
5.4	Bruksområde . . . . .	65
5.5	Begrensninger . . . . .	66
5.5.1	Metode . . . . .	66
5.5.2	Modellen . . . . .	68

<b>6</b>	<b>Konklusjon</b>	<b>72</b>
<b>7</b>	<b>Videre arbeid</b>	<b>74</b>
	<b>Referanser</b>	<b>76</b>
	<b>Vedlegg</b>	<b>I</b>
<b>A</b>	<b>Oppskrift posisjonsdata</b>	<b>II</b>
<b>B</b>	<b>Oppskrift motordata</b>	<b>V</b>
<b>C</b>	<b>Hastighetsmodell</b>	<b>VII</b>
<b>D</b>	<b>CSV_File_Reader</b>	<b>XXI</b>
<b>E</b>	<b>NVDB_posisjon</b>	<b>XXVII</b>

# Figurer

2.1	Eksempel på utforming av forbikjøringsfelt i stigning. (Statens vegvesen 2021) . . . . .	8
2.2	Eksempel på PGN. (CSS Electronics 2021d) . . . . .	10
2.3	VBOX Mini. (Racelogic 2020) . . . . .	11
2.4	CANedge1. (CSS Electronics 2021c) . . . . .	12
2.5	Overføring av data til menneskelig leselig form. (CSS Electronics 2021a)	12
2.6	Kalkulert hastighetsprofil. (Børnes og Aakre 2010) . . . . .	13
2.7	Inndeling av stigning i delstrekninger. (Børnes og Aakre 2010) . . . . .	15
2.8	Hastighet, posisjon og giring ved stigning på E6. (Nesdal 2019) . . . . .	17
2.9	Effekt som funksjon av hastighet og gir. (Nesdal 2019) . . . . .	17
2.10	Sammenligning av gammel og ny modell. (Nesdal 2019). . . . .	17
2.11	Forsøk med Scania. (Kogstad og Nygaard 2021) . . . . .	19
2.12	Forsøk med Mercedes-Benz. (Kogstad og Nygaard 2021) . . . . .	19
3.1	Scania R580 fra Børstad Transport. (Foto: Aamodt) . . . . .	23
3.2	Oppsett av VBOX i frontrute. (Foto: Aamodt) . . . . .	24
3.3	Posisjonsdata fra pilotforsøket. . . . .	25
3.4	CSV-filen i Excel. . . . .	26
3.5	Oppsett av CANedge1 i førerhytte. (Foto: Aamodt) . . . . .	27
3.6	Fremgangsmåte for å laste inn konverteringsfil. . . . .	28
3.7	Eksempel på visning i <i>asammdf</i> . . . . .	29
3.8	Scania R580 fra Galåen Transport. (Foto: Aamodt) . . . . .	31
3.9	Scania S650 fra Galåen Transport. (Foto: Aamodt) . . . . .	33
3.10	Scania S730 fra Galåen Transport. (Foto: Aamodt) . . . . .	35

3.11 Alle loggepunkter til venstre og gjennomsnittet av 10 og 10 punkter til høyre. . . . .	37
3.12 Vegkart fra Statens Vegvesen. (Statens vegvesen 2022e) . . . . .	37
3.13 Girutvekslingskurve for Scania R580. (Kogstad og Nygaard 2021) . . .	39
3.14 Dreiemomentkurve for Scania R580. (Kogstad og Nygaard 2021) . . .	39
3.15 Effektkurve for Scania R580. (Kogstad og Nygaard 2021) . . . . .	39
3.16 Høyde gjennom stigningen. . . . .	41
3.17 Stigning ved Saltøya. . . . .	41
3.18 Turtall gjennom stigningen. . . . .	41
3.19 Sammenligning av ny modell, gammel modell og GPS. . . . .	42
4.1 Scania R580 i Okstadbakken. . . . .	44
4.2 Scania S650 i Okstadbakken. . . . .	45
4.3 Scania S650 med start 500 meter senere i Okstadbakken. . . . .	46
4.4 Scania S730 i Okstadbakken. . . . .	46
4.5 Scania R580 gjennom hele stigningen ved Ålen. . . . .	47
4.6 Scania R580 gjennom Ålen_2. . . . .	49
4.7 Scania S650 gjennom Ålen_1. . . . .	49
4.8 Scania S730 gjennom Ålen_4. . . . .	50
4.9 Børstad Transport gjennom stigning ved Saltøya. . . . .	51
4.10 Galåen Transport gjennom stigning ved Saltøya. . . . .	52
4.11 Børstad Transport gjennom stigning ved Sveberg. . . . .	53
4.12 Galåen Transport gjennom stigning ved Sveberg. . . . .	53
4.13 Børstad Transport gjennom stigning ved Vikhammer. . . . .	54
4.14 Galåen Transport gjennom stigning ved Vikhammer. . . . .	55
5.1 Scania R580 i Ålen_1 med bruk av VBOX. . . . .	59
5.2 Scania R580 i Ålen_1 med bruk av CANedge1. . . . .	59
5.3 Scania S730 fra Galåen ved nedkortet stigning i Vikhammer. . . . .	61
5.4 Stigningsprosent og stigningen fra VBOX ved Veret. . . . .	63
5.5 Stigningsprosent fra NVDB ved Veret. . . . .	64
5.6 Ingen vind i Okstadbakken. . . . .	69
5.7 Motvind i Okstadbakken. . . . .	70
5.8 Medvind i Okstadbakken. . . . .	70

# Tabeller

2.1	Kritisk fartsdifferanse mellom tunge og lette kjøretøy. (Statens vegvesen 2021) . . . . .	7
2.2	Parametere for vogntog. (Statens vegvesen 2019) . . . . .	9
3.1	Nøkkelparametere for Scania R580 fra Børstad Transport. (Statens vegvesen 2022d) . . . . .	22
3.2	Nøkkelparametere for Scania R580 fra Galåen Transport. (Statens vegvesen 2022a) . . . . .	30
3.3	Nøkkelparametere for Scania S650 fra Galåen Transport. (Statens vegvesen 2022c) . . . . .	32
3.4	Nøkkelparametere for Scania S730 fra Galåen Transport. (Statens vegvesen 2022b) . . . . .	34
4.1	Sammenligning av modell og virkelighet ved Ålen_1. . . . .	48
4.2	Sammenligning av modell og virkelighet ved Ålen_2. . . . .	48
4.3	Sammenligning av modell og virkelighet ved Ålen_3. . . . .	48
4.4	Sammenligning av modell og virkelighet ved Ålen_4. . . . .	48

# Akronymer

**ASAM** Association for Standardization of Automation and Measuring Systems.

**CAN** Controller Area Network.

**CSV** Comma-separated values.

**DBC** Database Container.

**DBN** Dot bracket format.

**GUI** Graphical user interface.

**IRI** International Roughness Index.

**MDF4** Measurement Data Format version 4.

**MPD** Mean Profile Depth.

**NTNU** Norges teknisk-naturvitenskapelige universitet.

**NVDB** Nasjonal vegdatabank.

**OBD** On-board diagnostics.



**PGN** Parameter Group Number.

**SAE** Society of Automotive Engineers.

**SPN** Suspect Parameter Number.

**ÅDT** Årsdøgnetrafikk.

# Ordliste

**asammdf GUI** Software som kan brukes til å bearbeide motordata fra CANedge1.

**CANedge1** Loggerutstyr fra CSS Electronics.

**Circuit Tools** Programvare som kan brukes til å bearbeide data fra VBOX.

**VBOX** GPS og datalogger fra Racelogic.

**VBOX Test Suite** Programvare som kan brukes til å bearbeide data fra VBOX.

# Kapittel 1

## Introduksjon

Dette kapitlet skal gi en kort innledning til temaet og bakgrunnen for oppgaven. Videre presenteres forskningsspørsmålene som oppgaven skal forsøke å svare på, før oppbygningen av rapporten gjennomgås.

### 1.1 Bakgrunn

Norge er et stort og langstrakt land. I 2019 hadde det offentlige vegnettet en total lengde på 95 000 kilometer, hvorav 55 000 av disse er europaveger, riksveger og fylkesveger mens de resterende er kommunale veger. I tillegg har vi et stort nettverk av private veger som måler nesten samme lengde som det offentlige nettet (Dysvik mfl. 2021). På grunn av begrensninger på jernbanenettet, foregår veldig mye av godstransporten i Norge med trailere og vogntog på vegnettet. Tall fra statistisk sentralbyrå viser at det i 2020 ble fraktet over 20 millioner tonnkilometer med gods på norske veger (Statistisk Sentralbyrå 2021).

På grunn av den unike topografien vi har i landet, med daler og fjell, er store deler av vegnettet to-felts veger med mye både vertikal og horisontal kurvatur. Godstransporten foregår på trailere og vogntog som klassifiseres som tunge kjøretøy. Disse kan veie opp mot 50 tonn, som er grensen på mange av hovedvegene. I tillegg kan modulvogntog, og vogntog hvor avstand mellom første og siste aksel er over 19 meter, som

kjører på visse deler av riksvegnettet veie opptil 60 tonn (Samferdselsdepartementet 2021). Til sammenligning veier en vanlig norsk personbil rundt 2 tonn. Det krever mye mer energi å forflytte og akselerere et tungt kjøretøy kontra en personbil. Vanligvis har et tungt kjøretøy mer krefter og effekt enn en vanlig personbil, men hvis vi ser på vekt per effekt, kg/kW, er de tunge kjøretøyene underlegne, og oppfører seg annerledes i trafikken.

Tunge kjøretøy og stigninger er en dårlig kombinasjon og sammen med svinger og føreforhold sliter flesteparten med å opprettholde fartsgrensen. Dette kan i tillegg til køer, forsinkelser og i verste fall stengte kjørefelt og veger, føre til utålmodige privatbiler og potensielt livsfarlige forbikjøringer. I mange lange og bratte stigninger finnes det derfor forbikjøringsfelt, som kan brukes for å komme forbi tunge kjøretøy. Statens vegvesen har en modell som kan beregne hastighet og brukes blant annet for å utforme forbikjøringsfelt. Den tar utgangspunkt i standard parametere, men har en del begrensninger som gjør at den ikke gjenspeiler virkeligheten på den mest realistiske måten.

## 1.2 Forskningsspørsmål

Tidligere masteroppgaver har sett på hastighetsmodellen som brukes av Statens vegvesen og utviklet en forbedret hastighetsmodell. Det er gjennomført et begrenset antall forsøk for å utvikle modellen, ved å hente inn motordata og posisjonsdata fra tunge kjøretøy. På grunn av den foreløpige begrensede testingen, er det ønskelig med ytterligere forsøk for å teste og videreutvikle modellen, og for å undersøke om den fungerer etter hensikten.

Det er med utgangspunkt i dette utviklet to forskningsspørsmål som skal svares på i forbindelse med dette prosjektet:

1. *Hvor godt gjenspeiler modellen det som skjer i virkeligheten?*
2. *Hva kan en slik hastighetsmodell brukes til?*

## **1.3 Forprosjekt**

I forbindelse med denne masteroppgaven er det gjennomført et forprosjekt i henhold til utdanningsløpet ved studiet. Forprosjektet hadde som mål å finne tema og å utvikle forskningsspørsmål. I tillegg skulle relevant teori innhentes, og en plan og metode for selve masteroppgaven skulle utarbeides.

Forprosjektet og masteroppgaven omhandler det samme temaet, og kan således betraktes som en stor samlet oppgave, delt opp i to forskjellige rapporter. Derfor er mye av innholdet i innledningen, teorien og delvis metoden i masteroppgaven hentet delvis eller direkte fra forprosjektet (Aamodt 2021). Det vil videre ikke henvises til forprosjektet, men forprosjektet ligger som en kilde i referanselisten.

## **1.4 Rapportens oppbygning**

Oppgaven har blitt innledet med en introduksjon til det aktuelle temaet og en presentasjon av forskningsspørsmålene. Videre vil teorikapittelet gå gjennom all relevant teori som behøves for at forskningsspørsmålene kan besvares. Deretter vil metodene og alt som er blitt gjort i løpet av oppgaven bli presentert. Videre vil resultatene presenteres og diskuteres, før det fattes en konklusjon angående forskningsspørsmålene. Til slutt vil eventuelt videre arbeid med temaet og oppgaven presenteres. Oppgaven har produsert mye data og mange resultater. Det som ikke blir presentert i oppgaven eller vedleggene, vil bli lagt ved som digitale vedlegg.

# Kapittel 2

## Teori

I dette kapitlet vil jeg gjennomgå all relevant teori som trengs for å gjennomføre og besvare oppgaven. Jeg starter med elementære fysiske egenskaper knyttet til krefter og motstand for kjøretøy. Videre ser jeg på vegutforming i Statens vegvesen sine håndbøker. Deretter beskriver jeg loggerutstyret og dataprogrammene som brukes før jeg sier litt om tidligere arbeid med hastighetsmodeller.

### 2.1 Fysiske egenskaper

For å kunne modellere hastigheten til et kjøretøy, er det nødvendig å definere et utvalg fysiske størrelser. Disse størrelsene brukes for å forklare hvordan hastighet skapes og hvordan et kjøretøy reagerer ved ytre påvirkning. For å beskrive bevegelsen til et gitt kjøretøy, sees det på hvilke krefter som bidrar til hastighet og hvilke krefter som hindrer hastighet.

#### 2.1.1 Motor

For at et kjøretøy skal kunne skape hastighet behøves det en drivkraft, fra en motor. En motor i et kjøretøy vil ved hjelp av akslinger og reimer overføre energi til en mekanisk bevegelse gjennom dekkene, som gir fremdrift av kjøretøyet (SNL 2019) . Hvis disse kreftene er større enn motstandskreftene, vil kjøretøyet akselerere. Motsatt vil

kjøretøyet miste hastighet hvis motstandskreftene er større, og holde en jevn hastighet hvis kreftene for og imot er like store. Effekten for et kjøretøy oppgis vanligvis i hestekrefter. Det kan også konverteres til SI-enheten som er watt. En hestekraft er det samme som 735,5 watt eller 0,735 kW (Wikipedia 2020).

## 2.1.2 Motstand

Motstandskrefter er de kreftene som virker negativt på fremdrift og er kreftene som må overkommes for at et kjøretøy skal kunne akselerere i fartsretningen. For modellering av kjøretøy er det vanlig å definere at de totale motstandskreftene er summen av rullemotstand,  $F_R$ , luftmotstand,  $F_L$ , og stigningsmotstand,  $F_S$ .

$$F = F_R + F_L + F_S \quad (2.1)$$

- **Rullemotstand**

Rullemotstand er motstanden som oppstår når et kjøretøy ruller eller forsøker å rulle på overflaten. Rullemotstanden eller friksjonen stammer hovedsakelig fra deformasjon (Wikipedia 2021). Når dekket til en bil presses ned mot asfalten, vil gummiene som er mykere enn asfalten forme seg etter teksturen i asfalten og låse seg sammen. Når kjøretøyet er i bevegelse og dekket møter en «topp» i teksturen, må gummiene komprimere seg for å komme over. Når dette skjer lagres det energi i gummiene. Når gummiene har kommet over toppen og går ned igjen, slippes det energi løs. Men den energien som slippes løs er ikke like stor som den energien som lagres. Noe av denne energien forsvinner i form av varme, og det er dette som gir friksjon. Dette kalles hysteresis (Klein-Paste 2021).

Rullemotstandskraften,  $F_R$ , er produktet av rullemotstandskoeffisienten,  $f_r$ , og normalkraften til kjøretøyet,  $N$ :

$$F_R = f_r * N \quad (2.2)$$

- **Luftmotstand**

Luftmotstand er kraften fra luften som virker på kjøretøyet. Størsteparten av kraften skyldes forskjellen i trykk på forsiden og baksiden av kjøretøyet. Det

skapes et overtrykk på forsiden mens det på baksiden oppstår et undertrykk hvor det blir virvler. Det skapes også litt kraft gjennom friksjonen som oppstår når luften glir langsmed kjøretøyets overflate (SNL 2020).

$$F_L = \frac{1}{2} * C_D * \rho * A * v^2 \quad (2.3)$$

Formel 2.3 viser formelen for luftmotstand (Shaddiq 2013) hvor:

- $C_D$ : luftmotstandskoeffisient
- $\rho$ : Tettheten til luft, 1,293 kg/m<sup>3</sup> ved 0°C
- A: Arealet på fronten av kjøretøyet [m<sup>2</sup>]
- v: Relativ hastighet mellom kjøretøy og luft i fartsretning [m/s]

$C_D$  verdien for en personbil ligger mellom 0.20 og 0.30. For et tungt kjøretøy ligger den litt høyere på grunn av det store arealet på fronten, rundt 1.0.

#### • Stigningsmotstand

Stigningsmotstand oppstår når vegen ikke er flat, når vinkelen mellom normalkraften til kjøretøyet og fartsretningen er mer enn 90 grader. Motstanden skyldes altså gravitasjonen som forsøker å dra kjøretøyet nedover. Dette kan forklares med Newtons 2 lov (SNL 2021):

$$F_S = m * g * \sin\alpha \quad (2.4)$$

Hvor m er massen til kjøretøyet, g er gravitasjonskonstanten og  $\alpha$  er vinkelen mellom horisontalplanet og underlaget.

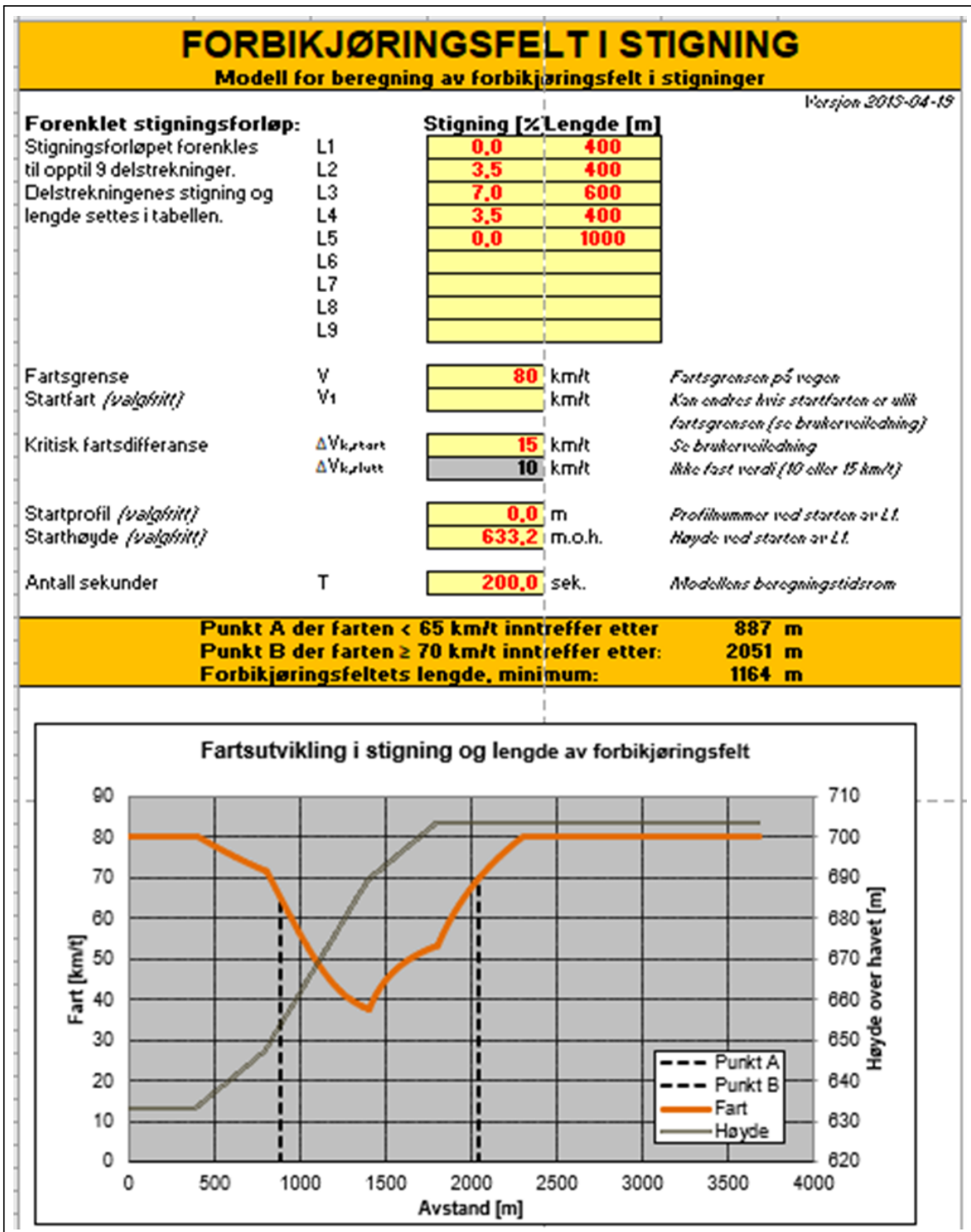


## 2.2 Dimensjonerende kjøretøy og vegutforming

Kapittel 4.5.3 *Forbikjøringsfelt i stigning* i håndbok N100 til Statens vegvesen, sier at det er viktig å vurdere behov for forbikjøringsfelt hvor forskjellen i hastighet mellom tunge og lette kjøretøy kan bli for stor. Dette gjøres for å opprettholde hensynet til et jevnt hastighetsnivå. Håndboka har definerte krav som må følges. På en 2-feltsveg hvor årsdøgntrafikk (ÅDT) er over 4000, skal det være forbikjøringsfelt når en stigning er så lang og bratt at det oppstår en for stor fartsdifferanse. Tabell 2.1 viser kritisk fartsdifferanse mellom tunge og lette kjøretøy. Forbikjøringsfeltet skal i starten ha full bredde senest der hvor den kritiske fartsdifferansen er nådd. For ÅDT-tunge over 400 er grensen satt til 15 km/t. Denne grensen økes til 20 km/t når ÅDT-tunge beveger seg under 400. Ved avslutning av forbikjøringsfelt gjelder det samme som ved starten, det kan ikke snevres inn før fartsdifferansen er under den kritiske verdien. Her er denne differansen satt til 10 og 15 km/t for henholdsvis over og under ÅDT-tunge på 400 (Statens vegvesen 2021). Statens vegvesen har en egen regnemodell i Excel som beregner lengden av forbikjøringsfeltet. Den er utviklet av Børnes og Aakre og vil bli presentert i Kapittel 2.4.1. En illustrasjon av modellen er vist i Figur 2.1.

**Tabell 2.1:** Kritisk fartsdifferanse mellom tunge og lette kjøretøy. (Statens vegvesen 2021)

ÅDT-tunge	Forbikjøringsfeltets start	Forbikjøringsfeltets avslutning
≥ 400	15 km/t	10 km/t
< 400	20 km/t	15 km/t



Figur 2.1: Eksempel på utforming av forbikjøringsfelt i stigning. (Statens vegvesen 2021)

For dimensjonering av forbikjøringsfelt i stigning i Norge er det vogntog som er det dimensjonerende kjøretøyet. Parameterene som inngår i beregningen, er gitt i Statens vegvesen sin *Håndbok V120 Premisser for geometrisk utforming av veier* og vist i Tabell 2.2 (Statens vegvesen 2019).

**Tabell 2.2:** Parametere for vogntog. (Statens vegvesen 2019)

Parameter	Enhet	Verdi
Rullemotstand	-	0.015
Luftmotstand	-	0.6
Areal, front	m <sup>2</sup>	8.0
Masse	kg	40000
Motoreffekt	kW	360

## 2.3 Data og loggerutstyr

Med dagens raskt forbedrede teknologi, inneholder moderne kjøretøy mange elektroniske komponenter som behøver å kommunisere med hverandre. Derfor bruker de fleste kjøretøy Controller Area Network (CAN) for at de ulike komponentene i kjøretøyet og motoren kan kommunisere. Det fungerer slik at en enhet kan sende ut informasjon til alle de andre enhetene, som hver for seg sjekker informasjonen før den enten aksepterer eller ignorerer den. Den mest brukte CAN standarden for det fysiske systemet og de fysiske kablene er CAN High og tillater bit hastigheter fra 40 kbit/s til 1Mbit/s. I tillegg brukes CAN Low. Denne har bare en hastighet på 40 kbit/s til 125 kbit/s, men har den fordelen at den tillater kommunikasjon selv om det er feil på en av to wire. CAN legger hovedsakelig bare fysisk til rette for kommunikasjon, ikke veldig mye mer. Det trengs derfor standardiserte protokoller for å spesifisere hvordan kommunikasjonen mellom enhetene i nettverket foregår (CSS Electronics 2021b).

I de fleste tunge kjøretøy er det standarden SAE J1939 som brukes. Standarden er definert av Society of Automotive Engineers (SAE), og tilbyr en standard metode slik at et felles språk kan brukes av alle produsenter. En J1939 melding består av en CAN ID på 29 bit. 3 bits beskriver prioriteten til meldingen, 18 bits er Parameter Group Number (PGN) mens de resterende 8 er kilden til meldingen. Etter CAN ID kommer CAN dataen på 64 bit. PGN fungerer på den måten at den identifiserer en unik parametergruppe. Videre kan Suspect Parameter Number (SPN) brukes til å identifisere hvilken parameter det er snakk om ut fra den gitte parametergruppen. Et lite eksempel kan være med på å illustrere dette. Hvis du i dokumentasjonen SAE J1939-71 søker opp PGN61444, finner du ut at dette er «Electronic Engine Controller 1». Videre finner du ut at SPN ID 899 tilsvarer «Engine Torque Mode» og SPN ID 190 er «Engine speed». En illustrasjon av dette er vist i Figur 2.2 (CSS Electronics 2021d).

PGN61444 - Electronic Engine Controller 1 - EEC1			
Transmission Repetition Rate: Engine speed dependent			
Data Length: 8 bytes			
Data Page: 0			
PDU Format: 240			
PDU Specific: 4			
Default Priority: 3			
Parameter Group Number: 61444 (0x00F004)			
Bit Start/Byte	Length	SPN ID	SPN Description
1,1	4 bits	899	Engine Torque Mode
2	1 byte	512	Driver's Demand Engine - % Torque
3	1 byte	513	Actual Engine - Percent Torque
4-5	2 bytes	190	Engine Speed
6	1 byte	1483	SA of Controlling Device for Engine Control
7,1	4 bits	1675	Engine Starter Mode
8	1 byte	2432	Engine Demand - Percent Torque

**Figur 2.2:** Eksempel på PGN. (CSS Electronics 2021d)

For å hente ut data som kan brukes i modellen trengs det loggerutstyr. Videre presenteres de valgte loggerene. Posisjon- og hastighetsdata hentes ut ved hjelp av VBOX, mens motordata blir innhentet med CANedge.

### 2.3.1 Posisjonsdata med VBOX

VBOX er en logger som er utviklet av Racelogic. Racelogic er et britisk selskap som utvikler og selger utstyr som måler blant annet hastighet, distanse og posisjonen til kjøretøy (Racelogic 2021). En av loggerene som er utviklet av Racelogic, og som NTNU har gått til innkjøp av er VBOX Mini. Dette er en selvstendig enhet som enkelt kobles opp i kjøretøyets frontrute og som ved hjelp av GPS registrerer hastighet og posisjon ved 10 Hz. For å forsterke signalet kan det kobles til en antenne som festes på taket av kjøretøyet (Racelogic 2020).



Figur 2.3: VBOX Mini. (Racelogic 2020)

Resultatene fra VBOX Mini lagres på et minnekort og kan lastes opp i dataprogrammet VBOX Test Suite som er utviklet av Racelogic. I dette programmet får du opp all informasjonen som ble lagret under turen. Du kan se tid, hastighet og posisjon. Du kan i tillegg koble posisjonen til programmet Google Earth Pro og fremstille ruten som har blitt logget.

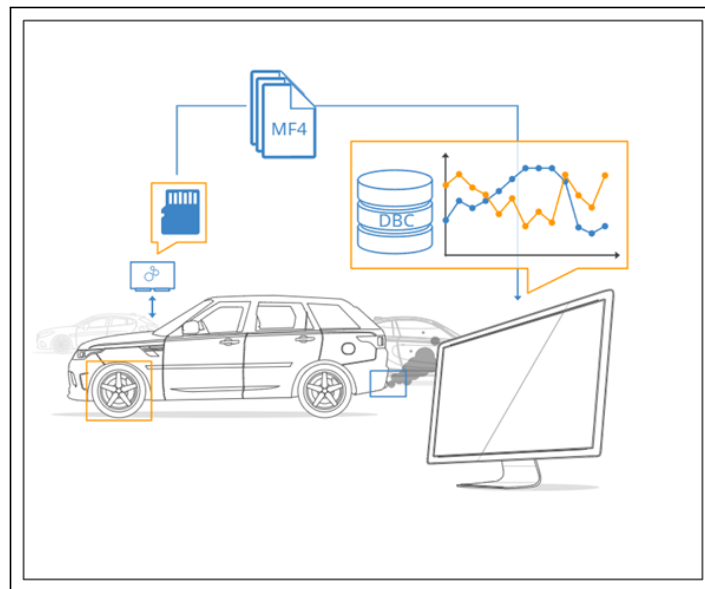
### 2.3.2 Motordata med CANedge1

For å hente ut motordata brukes det loggerutstyr fra CSS Electronics. Dette er et dansk selskap som utvikler og produserer CAN bus dataloggere og software som analyserer dataene. For denne oppgaven brukes loggeren CANedge1, da dette er kjøpt inn og er brukt av tidligere masterstudenter. CANedge1 er en liten boks (52.5 x 81.4 x 24.5 mm) som kobles til CAN bus via en port til On-board diagnostics (OBD), og logger CAN data over på et minnekort (CSS Electronics 2021c).



**Figur 2.4:** CANedge1. (CSS Electronics 2021c)

Når turen er gjennomført, kan minnekortet fjernes og legges inn på en PC og dataene kan hentes ut. Det finnes flere måter å gjøre dette på. Det fokuseres her på programmet *asammdf GUI*. Dataene på minnekortet er lagret på formatet MDF4. I programmet kan man laste inn en J1939 DBC fil som inneholder konverteringsregler, slik at dataene blir omgjort til menneskeleselig form og fysiske verdier som kan plottes (CSS Electronics 2021a). Figur 2.5 illustrerer hvordan dette fungerer.



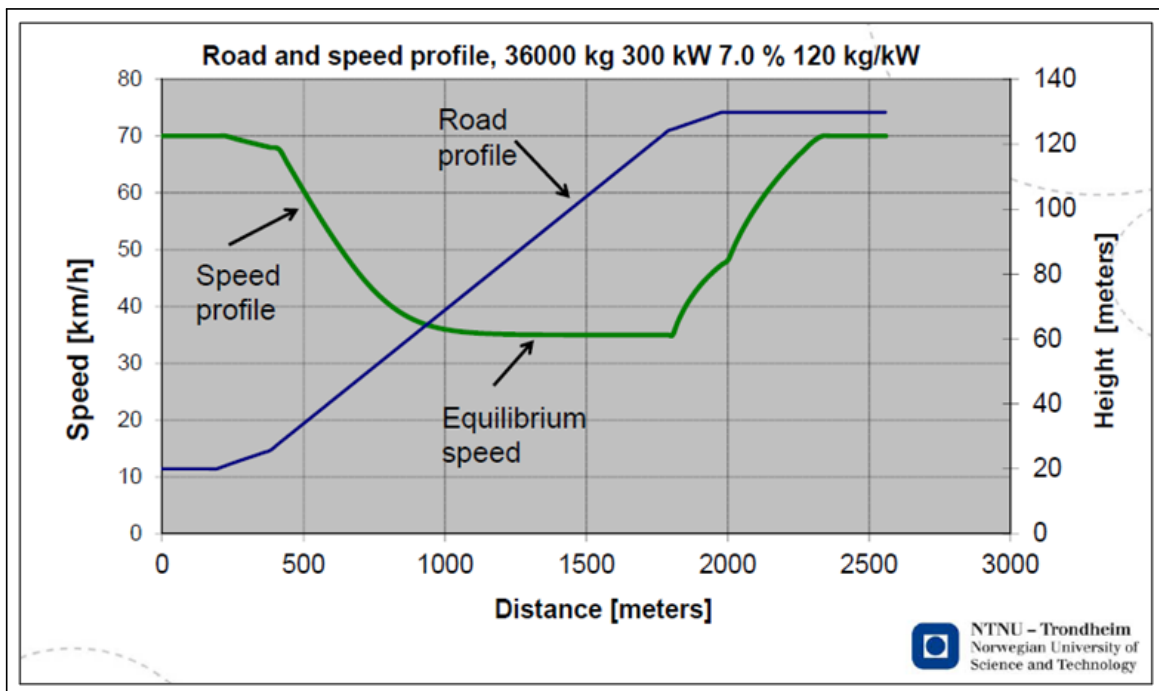
**Figur 2.5:** Overføring av data til menneskelig leselig form. (CSS Electronics 2021a)

## 2.4 Eksisterende arbeider og modeller

Her vil jeg presentere arbeid som har blitt gjort med tidligere modeller samt den eksisterende modellen. Først vil jeg beskrive den opprinnelige modellen som ble konstruert av Børnes og Aakre. Videre vil jeg beskrive masterarbeidet til Nesdal fra 2019, før jeg avslutter med arbeidet som er gjort på den eksisterende modellen til Kogstad og Nygaard.

### 2.4.1 Børnes og Aakre

Børnes og Aakre sin modell forsøker å beskrive en teoretisk hastighetsprofil til tunge kjøretøy i stigninger, basert på parameterne til kjøretøyet og føreren, stigningen og motstandskreftene. Hovedmålet med prosjektet var å utvikle en generell hastighetsmodell som kan brukes til å kalkulere reisetid som en funksjon av fartsgrense, vegens geometri og ÅDT. Et eksempel på en kalkulert hastighetsprofil er vist i Figur 2.6 (Børnes og Aakre 2010).



Figur 2.6: Kalkulert hastighetsprofil. (Børnes og Aakre 2010)

Hastighetsprofilen ble kalkulert ved hjelp av flere parametere.

*Kjøretøy:*

- Massen, kg
- Maksimal effekt, kW
- Utnyttelsesgrad, %. Det antas konstant utnyttelse på 95%
- Starthastighet, m/s

*Stigning:*

- Oppdeling i 5 delstrekninger. Lengde og stigningsprosent for hver delstrekning ble målt.
- Slutthastigheten i den ene delstrekningen er lik som starthastigheten i den neste delstrekningen. Se eksempel i Figur 2.7.

*Rullemotstand:*

- Rullemotstandskoeffisient
- Massen, kg
- Stigningsprosent, %

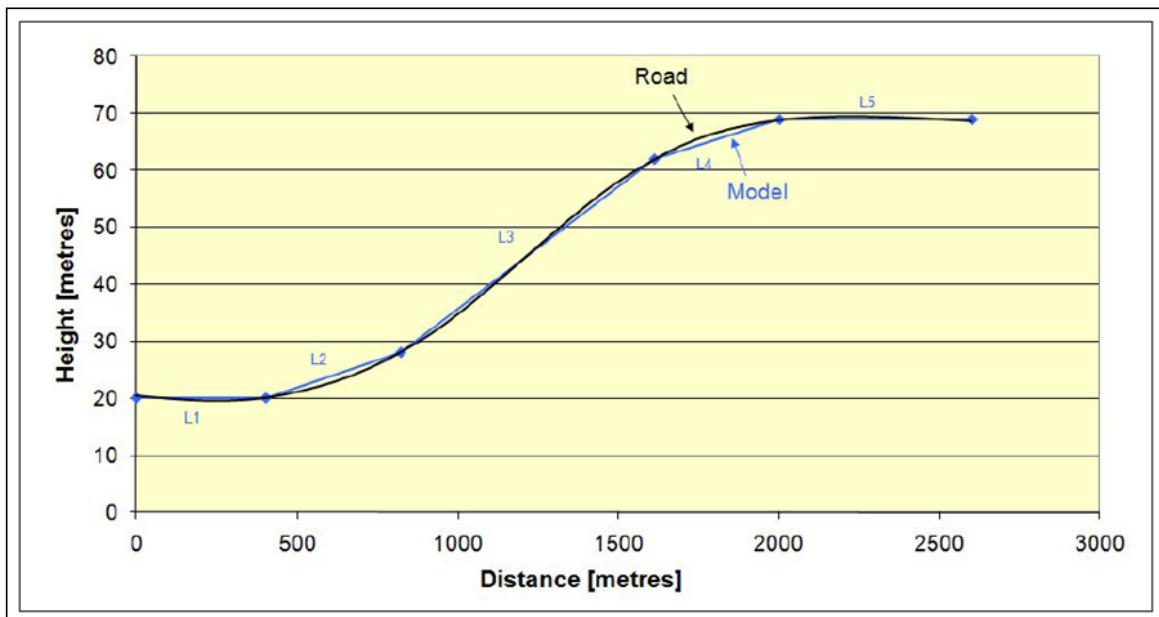
*Luftmotstand:*

- Luftmotstandskoeffisient
- Areal av front,  $m^2$
- Kjøretøyets hastighet, m/s
- Vindhastighet, m/s

*Stigningsmotstand:*

- Massen, kg
- Stigningsprosent, %





**Figur 2.7:** Inndeling av stigning i delstrekninger. (Børnes og Aakre 2010)

Modellen ble kalibrert og validert gjennom flere observasjoner av reell trafikk. En 70 kilometer lang rute ble kjørt flere ganger med en kombinasjon av forskjellige førere, laster og kjøretøy. Ruten har 10 uavhengige stigninger hvor en likevektshastighet skal kunne oppnås. 2 førere og 8 turer ble gjennomført på forskjellige tidspunkt og med forskjellige forhold. Data ble hentet inn med GPS, video og manuelle observasjoner, mens informasjon knyttet til vegen ble hentet fra Nasjonal vegdatabank (NVDB). Den viktigste grunnen til forskjell i hastighet, er weight/power ratio. Det vil si hvor mange kilogram det er per kW. I dette tilfellet varierte dette mellom 35 og 163 kg/kW. Noe av dataen ble brukt til kalibrering mens resten ble brukt for å validere modellen (Børnes og Aakre 2010).

Børnes og Aakre konkluderer med at modellen virker å produsere en realistisk hastighetsprofil for tunge kjøretøy i stigninger. De påpeker imidlertid at modellen er basert på fysiske lover og at den skiller seg fra virkeligheten ved at det er flere faktorer som vil spille inn. Det vil også oppstå støy eller uregelmessigheter knyttet til føre, kjøretøy, trafikk og førerens kjørestil.

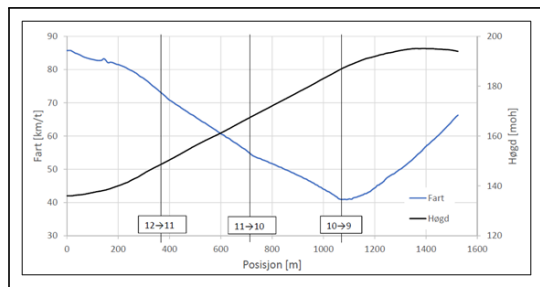
Videre i denne oppgaven vil modellen til Børnes og Aakre bli omtalt som "den gamle modellen".

## 2.4.2 Nesdal

Nesdal sin masteroppgave fra 2019 bygger videre på modellen til Børnes og Aakre. Målet hans med oppgaven var å forbedre modellen ved å implementere variabel motoreffekt. Han påpekte at effekten til et kjøretøy er avhengig av turtallet. Turtallet er avhengig av hastigheten til kjøretøyet og av hvilket gir som er i bruk. Dreiemomentet og effekten varierer ut ifra hvilket turtallsområde motoren opererer med. Derfor så han behovet for å modellere den utnyttede effekten, i motsetning til den eksisterende modellen som brukte en konstant verdi for effekt (Nesdal 2019).

For å implementere den variable effekten er det nødvendig å etablere en sammenheng mellom fart og turtall for hvert gir. Det må også bestemmes for hvilket turtall motoren skifter gir opp og ned. Den faktiske effekten avhenger også av kjøreprogrammet som brukes. En typisk moderne automatgirkasse har gjerne et økonomiprogram, et standardprogram og et mer sporty program. Økonomiprogrammet vil velge gir slik at turtallet holder seg i området hvor dreiemomentet er størst, slik at drivstoffbruken blir minst mulig. Sportsprogrammet søker maks effekt og vil ligge i et høyere turtall. Dette gir litt lavere dreiemoment, men effekten er større og det gir høyere drivstofforbruk. Et standardprogram vil gjerne ligge midt mellom og prøve og balansere effekt mot drivstofforbruk.

For å teste modellen ble det utført et forsøk. Dette ble gjort med en Volvo FH 2019-modell som veide 50 tonn fra Børstad Transport. Det ble gjort målinger med Video VBOX, VBOX Mini og VBOX Sport. Video VBOX hadde tilkoblet to kameraer, en på dashbordet for å filme hastighet og turtall, og en i frontruta for å filme vegen. Etter forsøksturen ble det produsert en graf som viste hastighet, posisjon, giring og stigning til kjøretøyet i en stigning på E6 i Malvik, se Figur 2.8. Ut ifra gir, turtall og hastighet, ble dreiemoment og effekt funnet ved hjelp av kurver gitt av produsenten, se Figur 2.9.

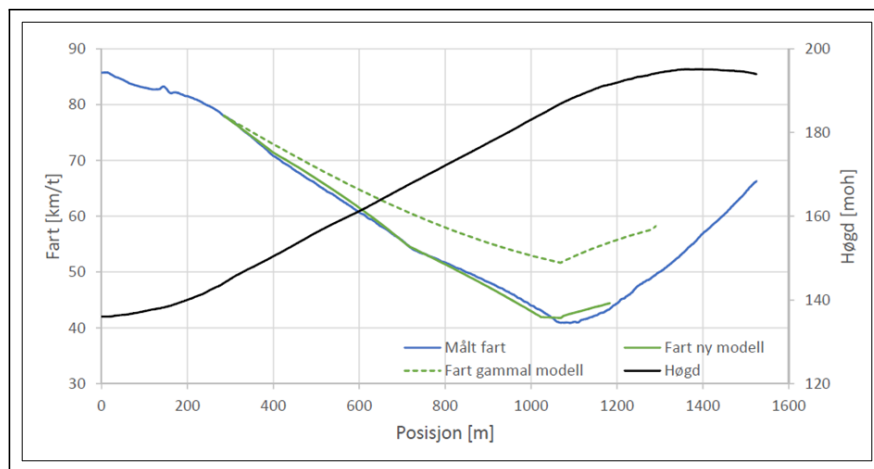


**Figur 2.8:** Hastighet, posisjon og giring ved stigning på E6. (Nesdal 2019)

Fart [km/t]	Gir	Forhold turtal-fart	Turtal	Effekt [kW]
80	12	15,3	1224	334
79	12	15,3	1209	331
78	12	15,3	1193	328
...	12	15,3	...	...
72	12	15,3	1102	309
71	11	19,4	1377	382
70	11	19,4	1358	375
69	11	19,4	1339	368
...	11	19,4	...	...
55	11	19,4	1067	269
54	10	23,9	1291	353
53	10	23,9	1267	347
52	10	23,9	1243	340

**Figur 2.9:** Effekt som funksjon av hastighet og gir. (Nesdal 2019)

Etter at forsøket var gjennomført, ble det gjort beregninger med den nye modellen med variabel effekt. For hvert tidssteg i modellen ble den tilgjengelige effekten hentet fra tabellen. Den gamle modellen ble også kjørt med de samme parameterne, utnyttelsesgraden ble satt til 100%. Figur 2.10 viser den målte hastigheten, hastigheten til den nye modellen og til den gamle modellen.



**Figur 2.10:** Sammenligning av gammel og ny modell. (Nesdal 2019).

Nesdal konkluderer med at den nye modellen gir et profil som stemmer bra med det som er målt, og at den er bedre enn den gamle modellen. Han påpeker imidlertid at dette bare er ett enkelt forsøk med en god del begrensninger og at det med et så lite datagrunnlag ikke kan trekkes noen endelige konklusjoner. Det kan være hensiktsmessig å jobbe videre med temaet for å utvikle og validere en mer generell modell som fungerer på tvers av fabrikanter, føre, trafikk og vegstrekninger.

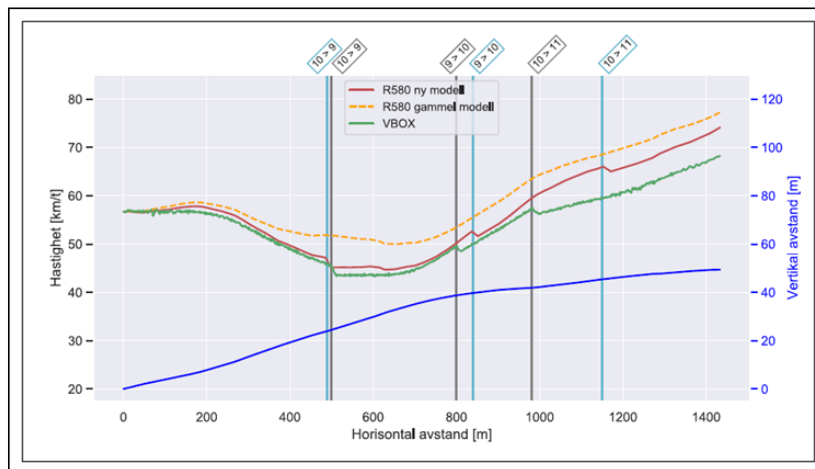
### 2.4.3 Kogstad og Nygaard

Kogstad og Nygaard sin masteroppgave i 2021 hadde som mål å videreutvikle den eksisterende hastighetsmodellen ved å bruke tall fra reelle forsøk med logging av motor-data. Tallene bidrar til en mer dynamisk modell som gir en høyere presisjon i modellen (Kogstad og Nygaard 2021).

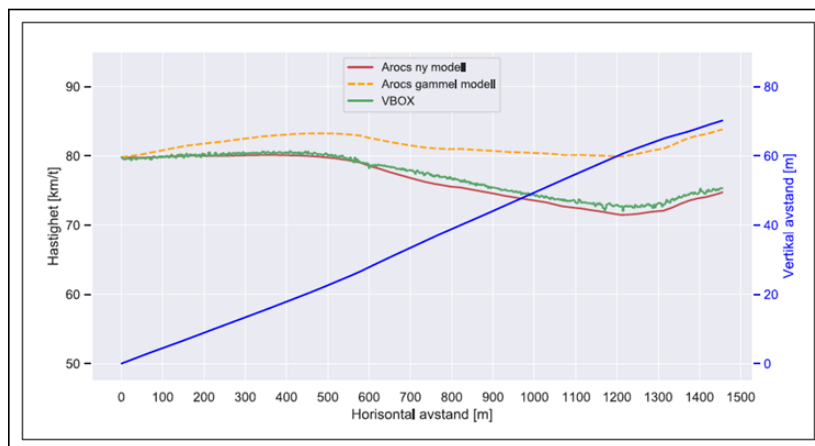
For å samle inn data, ble det gjennomført flere forsøk. Fra Børstad Transport ble det gjort 2 forsøk med en Scania R580 som veide 50 tonn. For posisjonslogging ble VBOX Mini brukt, mens motorloggingen skjedde med CANedge1 gjennom OBD-porten i kjøretøyet. Det ble også gjennomført 2 testturer med en innleid Mercedes-Benz Arocs tippbil. Denne ble kjørt en tur hvor den var halvfull og veide 21 tonn, og en tur hvor den var full og veide 28 tonn. Posisjon ble logget med VBOX Mini mens motorloggingen ble gjort med CANedge1 som ved hjelp av CANCrocodile ble koblet utenpå CAN-High og CAN-Low ledningene i bilen. Et forsøk med en Volvo FH 540 fra Børstad Transport ble også gjennomført. Her var det ikke mulig å koble til CANedge1, så løsningen ble å filme dashbordet og gå gjennom filmen manuelt i etterkant. Det ble også her brukt VBOX Mini for å logge posisjon.

Modellen ble videreutviklet i programspråket Python. Dette ble gjort med fokus på variabel effekt, girskift og motbakkefremstilling. For å få en realistisk modell er det viktig at motbakkene blir fremstilt korrekt. For å modellere stigningene ble det vurdert å enten bruke data fra NVDB eller fra loggerforsøkene med VBOX Mini. Metodene ble sammenlignet med hverandre, og de kom frem til at data fra VBOX ble foretrukket, selv om forskjellen mellom dem hadde liten betydning for modellen. Den variable motoreffekten ble modellert ved hjelp av forholdet mellom turtall og effekt. Beregningsfrekvensen ble satt til hvert hundredels sekund.

Modellen ble brukt til å gjennomføre beregninger av hastighetsprofiler og girskift for alle forsøkene som ble gjort. Det ble også gjort beregninger for den gamle modellen. Disse ble så satt sammen i en graf for å sammenligne mot det som faktisk ble registrert. Figur 2.11 viser den ene bakken kjørt med Scania mens Figur 2.12 viser den ene bakken kjørt med Mercedes-Benz.



**Figur 2.11:** Forsøk med Scania. (Kogstad og Nygaard 2021)



**Figur 2.12:** Forsøk med Mercedes-Benz. (Kogstad og Nygaard 2021)

Kogstad og Nygaard konkluderer med at den nye modellen samsvarer godt med tanke på tidsbruk i en stigning og laveste nådde hastighet. Tidspunkt for giring og girvalg blir ikke like godt fremstilt, da girskift i virkeligheten er kompliserte og avhenger av mange faktorer. De nevner også at det finnes begrensninger ved modellen knyttet til blant annet vegens geometri, kjøremotstand og ytre omgivelser. Til slutt legger de frem forslag til videre utvikling av modellen, som blant annet innebærer ytterligere testing for at modellen skal bli mer generell, innføring av flere parametere og å se på forskjellige kombinasjoner av vekt og motorstørrelse.

Videre i oppgaven vil modellen til Kogstad og Nygaard bli referert til som "den nye modellen", eller bare "modellen".

# Kapittel 3

## Metode

I dette kapittelet blir de aktuelle metodene som er benyttet beskrevet. Først litt informasjon om forprosjektet som ble gjennomført høsten 2021. Videre om piloten som ble gjennomført, før datainnsamling og analyse blir presentert. Til slutt vil den aktuelle hastighetsmodellen bli beskrevet i mer detalj.

Alle figurer uten kilde er egenproduserte.

### 3.1 Forprosjekt

Som nevnt i innledningen, ble det høsten 2021 gjennomført et forprosjekt til denne oppgaven gjennom faget *TBA4541 - Veg, fordypningsprosjekt*. Faget gir en mulighet til å få en grundig innledning til temaet som masteroppgaven skal handle om. I tillegg utarbeides forskningsspørsmål, relevant teori innhentes og det kan legges en plan for hvordan selve masterprosjektet skal gjennomføres. Alt dette bidrar til en raskere og enklere oppstart av selve masteroppgaven ved semesterstart etter nyttår.

Teorien i forprosjektet ble hovedsaklig innhentet ved hjelp av litteratursøk, samt gjennomgang av tidligere masteroppgaver. Veileder hadde også gode tips på hva som kunne være interessant å lese på. Etter å samlet inn teori og lest gjennom tidligere forskning og masteroppgaver, ble det i samråd med veileder bestemt hvilket utstyr som var mest hensiktsmessig å bruke i oppgaven. For å finne ut av hvordan dette utstyret

fungerte, ble det gjennomført to tester, en for hver av dem. I tillegg til testingen, ble masterstudentene fra 2021, Jakob Kogstad og Simen Nygaard, kontaktet for et potensielt møte. Dette var de positive til og gjennom et digitalt møte fortalte de litt om sine erfaringer og hvordan de opplevde å jobbe med oppgaven. Det ble samtidig gitt tips til hvordan gå frem, spesielt med tanke på modellen de hadde utviklet i Python. Etter dette møtet ble det satt av god tid til å sette seg inn i modellen, da denne er en viktig del av masterprosjektet.

GPS-loggeren VBOX Mini ble testet i personbil. Dette viste seg og være ganske rett frem. Loggeren ble koblet til 12V i bilen og dataene fra en kort kjøretur ble lagret på et minnekort. Dette kunne i tur legges inn på datamaskinen og importeres til et dataprogram for å fremstille resultatene. Mer informasjon om hvordan dette utføres finnes i Kapittel 3.3.1.

Motorloggeren CANedge1 ble testet i en MAN lastebil tilhørende Sandvik Transport AS. Disse ble kontaktet gjennom avdelingsingeniør Jan Erik Molde ved NTNU, og var veldig behjelpelige. Grunnet sjåførkort og kjøre- hviletidsbestemmelser ble testturen bare en liten runde på området til Sandvik Transport på Heimdal. Loggeren ble koblet til bak fartsskriveren i førerhytta ved hjelp av krokodilleklemmer over CAN-High og CAN-Low ledningene. Det var ikke like enkelt å produsere resultater for denne loggeren. Det måtte flere forsøk til før resultatene kunne hentes ut. Først ble det brukt et minnekort på 64GB. Dette fungerte ikke da maksimal størrelse som aksepteres av loggeren er 32GB. Deretter var ikke minnekortet riktig konfigurert. I tillegg må minnekortet være tomt når det settes inn. Den siste hindringen var at loggingen ikke ble avsluttet korrekt. Derfor er det viktig å la loggeren avslutte, slik at resultatene blir lagret på minnekortet. Først da kan minnekortet tas ut og lastes opp på en datamaskin. Ytterligere detaljer rundt bruken av CANedge1 beskrives i Kapittel 3.3.2.

Testing av utstyret var både nyttig og lærerrikt. Det ble mulighet for å gjøre seg kjent med utstyret, oppsett og bruken av dataprogrammene. Dette bidro til en enklere oppstart og mer selvtillit når masteroppgaven gikk i gang og piloten ble gjennomført. Forprosjektet inkluderte også utarbeidelse av en fremdriftsplan. Dette er et veldig greit utgangspunkt og legger føringer for når og hva som skal produseres. Samtidig kan den enkelt endres og tilpasses der dette blir nødvendig.

## 3.2 Pilot med Scania fra Børstad Transport

Selv om loggerutstyret ble testet raskt i forbindelse med forprosjektet, ble det sett på som nødvendig å gjennomføre en fullstendig pilot, før videre testing kunne fortsette. Målet med dette var å skaffe ytterligere erfaringer med bruk av systemet og utstyret, og få en fullstendig tur med data, resultater og gjennomgang i modell. Erfaringen fra tidligere var at Scania var den enkleste produsenten å jobbe med, da motorloggeren enkelt kan kobles i OBD-porten i førerhytta. Det var derfor naturlig å bruke en Scania i piloten.

Det ble tidlig tatt kontakt med Børstad Transport gjennom NTNU og Jan Erik Molde. Det ble avtalt at turen skulle gjennomføres med en 2019 modell Scania R580 som kjører for felleskjøpet, se Figur 3.1. Dette på grunn av at det er et kjøretøy med relativt høy vekt, det veide omtrent 47,3 tonn. Ytterligere detaljer om kjøretøyet vises i Tabell 3.1. Det ble gitt telefonnummer til sjåfør for å diskutere ytterligere detaljer om oppmøte og gjennomføring. Turen ble kjørt 31.01 og vogntoget fraktet kraftfor. Ruten gikk fra Felleskjøpet på Steinkjer og sørover på E6 gjennom Trondheim. Det ble bestemt at det enkleste var å ta tog til Steinkjer og hoppe på bilen der. Ettersom kjøretøyet skulle videre fra Trondheim, ble loggingen og turen avsluttet ved bussholdeplassen på E6 ved Moholt. Totalt varte forsøket i 2 timer og det foregikk midt på dagen, på en mandag. Det var ingen trafikk å snakke om, så bilen fikk kjøre uforstyrret. Temperaturen lå rett under 0 grader og vegen var bar på hele strekningen.

**Tabell 3.1:** Nøkkelparametere for Scania R580 fra Børstad Transport. (Statens vegvesen 2022d)

Bil	Scania R580 2019
Reg. Nr.	VJ 17171
Egenvekt	15 000 kg
Totalvekt	47 300 kg
Motorstørrelse	427 kW / 581 HK, 16.4L





**Figur 3.1:** Scania R580 fra Børstad Transport. (Foto: Aamodt)

Forsøket gikk etter planen og anses som vellykket. Det resulterte i tre MF4 filer fra CANedge1 på til sammen 110 MB. To filer var på 50 MB og en på 10 MB, da maks størrelse er 50. I tillegg kom det gode resultater fra GPS-en til VBOX. Selve resultatene blir lagt frem i Kapittel 4, mens fremgangsmåten på hvordan disse blir hentet ut og produsert blir gjennomgått i de neste delkapitlene.

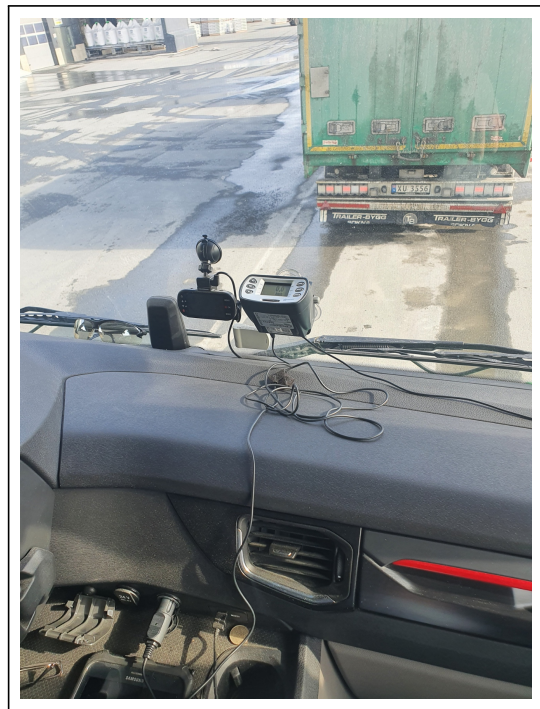
## 3.3 Datainnsamling

Som nevnt tidligere, blir innsamlingen av data gjort med to forskjellige loggere. GPS-koordinater og hastighet blir loggført ved hjelp av VBOX Mini, mens motordata blir hentet inn ved hjelp av CANedge1 fra CSS Electronics. Det mest ideelle hadde vært å bare bruke en logger, men da CANedge1 ikke har posisjonsdata er ikke dette mulig. Resultatene fra loggerene blir videre bearbeidet i hvert sitt program på datamaskinen, før de blir hentet inn i modellen og modellen kan brukes. Disse prosessene vil bli forklart i detalj i de to neste delkapitlene.

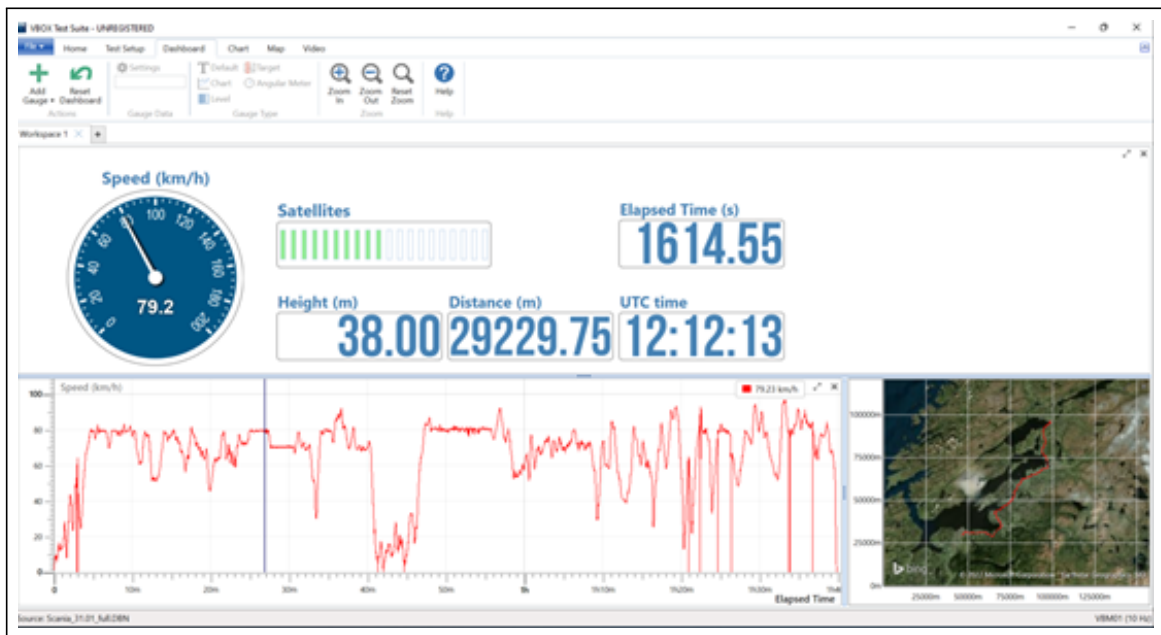
### 3.3.1 Posisjonsdata

Posisjonsdata og hastighet registreres ved hjelp av VBOX Mini fra Racelogic. Under forsøk plasseres enheten i frontruten ved hjelp av sugekopper, helst midt i ruten. Den har en innebygget antenne, men en ekstern antenne som festes på taket av kjøretøyet kan kobles til for bedre signaler. Eksempel på oppsett vises i Figur 3.2. Så lenge det er et minnekort i enheten, kan strømmen kobles til i 12V uttak i kjøretøyet og den vil starte opp og begynne å logge data. Data blir lagret som en DBN-fil. Etter forsøket kan enheten kobles fra og resultatet er lagret på minnekortet.

Når forsøket er over, overføres dataene fra minnekortet inn på datamaskinen. Deretter lastes den loggede filen inn i programmet *VBOX Test Suite*, som er en programvare utviklet av Racelogic. Når filen er lastet inn, kan vinduet se ut slik som vist i Figur 3.3. Dette er filen fra pilotforsøket med Scania fra Børstad Transport.



**Figur 3.2:** Oppsett av VBOX i frontrute. (Foto: Aamodt)



**Figur 3.3:** Posisjonsdata fra pilotforsøket.

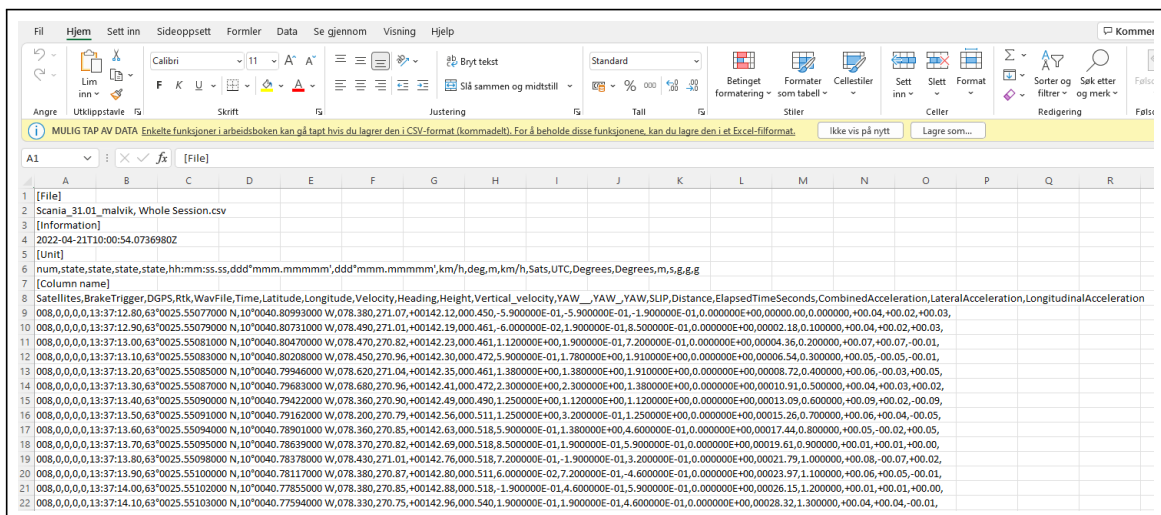
På størsteparten av den nedre delen i vinduet, vises en graf med hastigheten koblet opp mot tiden. Her illustreres altså hastigheten gjennom hele forsøket. Ved å klikke et sted på denne grafen, kommer det opp en del informasjon om akkurat dette punktet. I Figur 3.3 har jeg valgt å vise de mest relevante parameterene, over grafen. Dette inkluderer hastighet, høyde over havet, distansen som er kjørt, den universelle tiden og tiden som er brukt frem til dette valgte punktet. I tillegg er det en parameter som viser hvor god forbindelsen GPS-en har til satellittene.

Til høyre for grafen vises ruten som er kjørt i et kart. Punktet som er valgt i grafen, vises også med et kryss i kartet. Tilsvarende som å velge et punkt på grafen, kan det også velges et punkt i kartet. Da vil dette punktet igjen vises i grafen. Det er også mulig å eksportere ruten i kartet til *Google Earth Pro*. Da vil det vises en linje i programmet som viser den aktuelle ruten som er kjørt. En illustrasjon av dette finnes i Vedlegg A.

Når en del av strekningen skal analyseres, i dette tilfellet en stigning, må denne identifiseres og isoleres i programvaren. Dette gjøres ved å trykke i grafen der stigningen starter. Ved å bruke kutte-funksjonen i programmet, kan alt før dette punktet slettes/tas bort. På samme måte kan slutten av stigningen identifiseres og samme funksjon kan brukes til å fjerne alt etter dette punktet. Når dette er gjort må det lagres som

en ny fil og åpnes i et annet program, *Circuit Tools*. Denne programvaren lastes ned samme sted som *VBOX Test Suite*, og brukes for å eksportere stigningen til en CSV-fil som er en kommadelt fil i Microsoft Excel. I *Circuit Tools* lastes filen inn før den eksporteres ved hjelp av funksjonen *Export laps as CSV from current session*, som ligger under fanen *Session*.

Før filen kan lastes inn i modellen, må den redigeres i Excel. Dette gjøres ved at de syv øverste radene fjernes, se Figur 3.4. Det som da står igjen er den første raden med beskrivelse av parameterene, mens resten er data for de forskjellige parameterene, separert med komma. For denne oppgaven er det *velocity*, *height* og *distance* som er relevant. Nå kan CSV-filen lagres og importeres i modellen i Python. En gjennomgang av selve modellen kommer i kapittel 3.5.



Figur 3.4: CSV-filen i Excel.

For en punktvis gjennomgang av hvordan posisjonsdata blir fremstilt er det laget en oppskrift som kan studeres i Vedlegg A.

### 3.3.2 Motordata

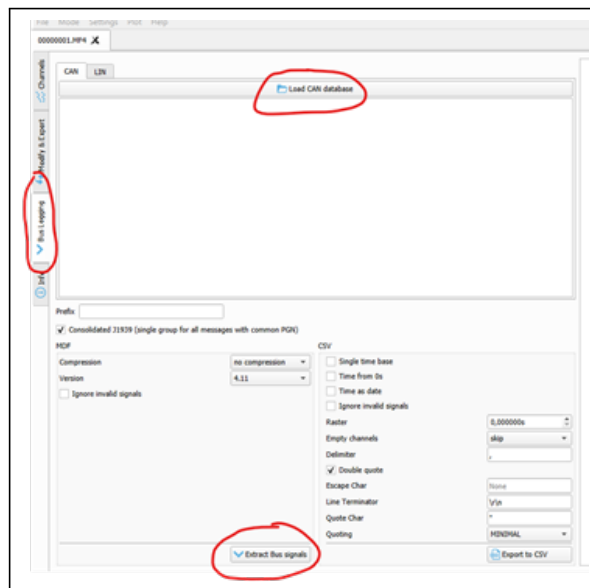
Motorloggerdata fra tunge kjøretøy registreres ved bruk av CANedge1 fra CSS Electronics. Den kan kobles til på flere måter, og er forskjellig på de ulike kjøretøyprodusentene og merkene. Som nevnt i Kapittel 3.1, kan loggeren på et kjøretøy fra MAN kobles til ved hjelp av krokodilleklemmer som plasseres på CAN-High og CAN-Low ledningene. Dette krever litt arbeid, da ledningene først må identifiseres. Det trengs også tilgang til strøm gjennom 12V. For Scania er tilkoblingen enklere, da enheten kan kobles direkte til OBD-porten, *On-board diagnostics*, i førerhytta. Enheten får i tillegg strøm fra porten, så dette er det eneste tilkoblingspunktet som behøves. På nyere kjøretøy fra Scania, 2016 og nyere, finnes porten i fotbrønnen på passasjersiden. Figur 3.5 viser CANedge1 som ligger på dashbordet med ledningen tilkoblet ned i OBD-porten.



**Figur 3.5:** Oppsett av CANedge1 i førerhytte. (Foto: Aamodt)

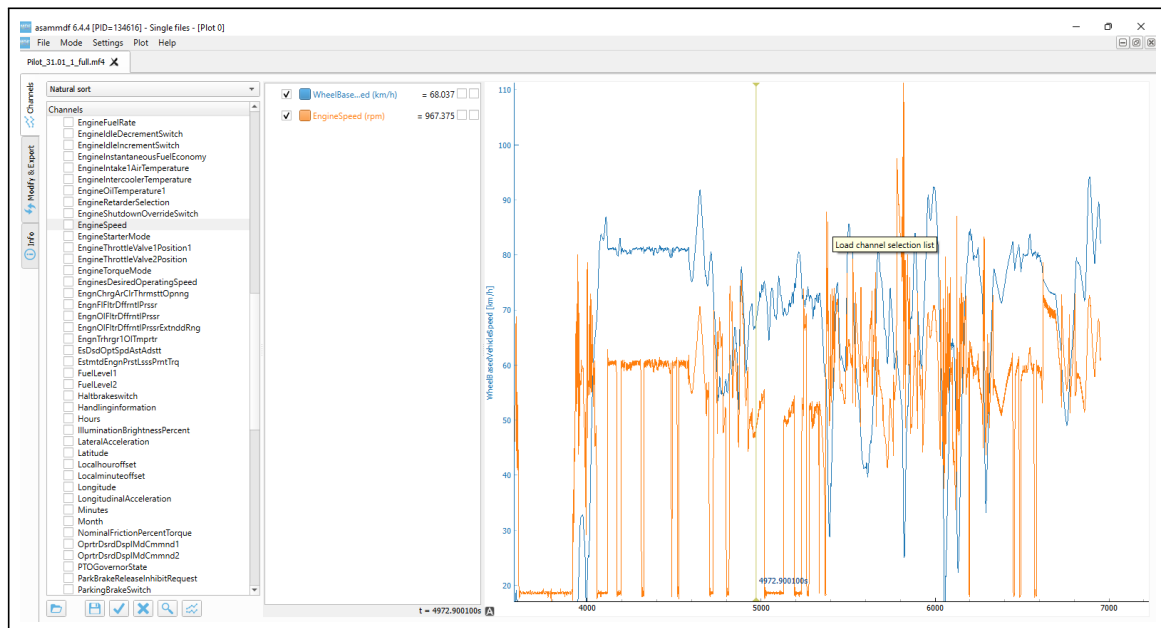
Når loggeren er koblet til og kjøretøyet er startet, vil enheten starte opp. Da vil det lyse grønt der hvor det står *pwr* på loggeren. Hvis det blinker rødt ved *mem*, betyr dette at det er opprettet forbindelse med motoren, og logging til minnekortet er startet. Når forsøket er over, stanses kjøretøyet. Da vil loggingen avsluttes og lagres på minnekortet. Dette tar noen sekunder, og når den er ferdig vil *mem* slutte å blinke. Først da kan minnekortet fjernes og lastes opp på en datamaskin.

Data blir logget på minnekortet som en MF4-fil. Resultatet kan visualiseres og bearbejdes i et åpent verktøy, *asammdf GUI/API*, som kan lastes ned fra nettsidene til CSS Electronics. Når filen er lastet inn, vises en liste med kanaler på venstre siden i programmet. Disse representerer forskjellige data fra forskjellige komponenter i kjøretøyet. Problemet er bare at resultatene ikke er leselig for oss mennesker, og dataene må dermed konverteres. Dette gjøres ved hjelp av en J1939 DBC-fil. Denne lastes inn i programmet slik som vist i Figur 3.6. Først trykker en på *Bus Logging* til venstre i menyen. Her lastes konverteringsfilen inn ved å trykke på *Load CAN database*. Når dette er gjort trykker en på *Extract Bus signals* og lagrer den nye filen under ønsket navn.



**Figur 3.6:** Fremgangsmåte for å laste inn konverteringsfil.

I den nye filen dukker det også opp en liste med kanaler, men i motsetning til før konverteringen, er kanalene nå lesbare. Dette kan for eksempel være *EngineSpeed*, *FuelLevel1* og *EngineOilTemperature1*, se Figur 3.7. Kanalene kan visualiseres i tabell, numerisk eller grafisk. Det kan visualiseres alene eller sammen med andre kanaler. Figur 3.7 viser en grafisk representasjon av turtallet og hastigheten til kjøretøyet gjennom forsøket.



**Figur 3.7:** Eksempel på visning i *asammdf*.

Det er produsert en enkel oppskrift på hvordan en punktvis går frem for å fremstille motordata ved bruken av CANedge1. Denne ligger vedlagt i Vedlegg B.

## 3.4 Forsøk med Galåen Transport

I dette kapittelet vil forsøkene som er utført med Galåen Transport kort beskrives. Galåen Transport er et transportselskap med tilholdssted på Røros. De tilbyr diverse transporttjenester, de kjører blant annet for Tine og Coop, og har derfor hyppige ruter mellom Trondheim og Røros. Høydeforskjellen mellom disse to stedene ligger på rundt 600 meter. Dette betyr at det er en god del stigninger på strekningen Trondheim-Røros, som passer godt å bruke i forbindelse med denne oppgaven. En fordel med Galåen er at selskapet bruker kjøretøy fra Scania. Dette gjør oppsett og gjennomføring av forsøkene enkelt. En annen fordel med å kjøre den samme strekningen flere ganger er at det blir mulig å sammenligne og sette opp forskjellige motorstørrelser og totalvekter opp mot hverandre.

### 3.4.1 Scania R580

Den første turen med Galåen ble gjennomført 2. mars, i en 2016-modell Scania R580 lastebil med henger, illustrert i Figur 3.8. Kjøretøyet var lastet med diverse meierivarer og veide rundt 42,5 tonn. For ytterligere informasjon, se Tabell 3.2.

**Tabell 3.2:** Nøkkelparametere for Scania R580 fra Galåen Transport. (Statens vegvesen 2022a)

Bil	Scania R580 2016
Reg. Nr.	HF 27558
Egenvekt	14 400 kg
Totalvekt	42 500 kg
Motorstørrelse	427 kW / 581 HK, 16.4L

Turen startet på Tine sitt lager på Tunga i Trondheim, og sluttet på hovedkontoret til Galåen Transport rett sør for Røros sentrum. Forsøket startet klokka 14:00, som gjorde at kjøretøyet fikk forlatt Trondheim før ettermiddagsrushet startet. Det var også lite trafikk på resten av ruten som resulterte i at kjøretøyet fikk kjøre fritt i stigningene, uten biler foran seg som forstyrret. Totalt varte turen i 3 timer. Det var ingen nedbør under forsøket, men asfalten var fuktig. Etter å ha tatt av fra E6 ved Støren, over på Fylkesveg 30, blir vegen smalere. Herfra var det også en del snø og is i kanten av vegbanen som påvirket kjøringen, spesielt ved møtende trafikk. Temperaturen startet



på 4 grader og krøp ned til 0 i det turen endte i Røros. Etter at forsøket var fullført ble all data lastet opp på datamaskinen og kontrollert, før turen gikk med tog tilbake til Trondheim.

Da dette kjøretøyet er litt eldre, var OBD-porten plassert et litt annet sted, helt på siden av dashbordet ved føreren. VBOX ble installert som vanlig i midten av frontruten. Selv om kjøretøyet er utstyrt med en ganske kraftig V8 motor, mistet den hastigheten i flere stigninger på vegen på grunn av vekten. Av de litt lenger og brattere stigningene mistet den mye hastighet i Okstadbakken i Trondheim og ved stigningene ved Ålen. I tillegg til disse, var det flere andre litt mindre bratte og kortere stigninger som reduserte hastigheten, slik at forsøket produserte mye god data som kunne brukes.



**Figur 3.8:** Scania R580 fra Galåen Transport. (Foto: Aamodt)

### 3.4.2 Scania S650

Det neste forsøket ble gjort 23. mars i en Scania S650 semitrailer fra 2019, se Figur 3.9. Den var lastet med diverse stykk gods, blant annet meieriprodukter, mineralvann og ventilasjonsrør. Den totale vekten var på cirka 42 tonn.

**Tabell 3.3:** Nøkkelparametere for Scania S650 fra Galåen Transport. (Statens vegvesen 2022c)

Bil	Scania S650 2019
Reg. Nr.	HF 28568
Egenvekt	10 000 kg
Totalvekt	42 000 kg
Motorstørrelse	478 kW / 650 HK, 16.4L

Ruten som ble kjørt under forsøket var den samme som ved forrige forsøk, start ved Tunga og slutt ved Røros. Turen startet klokka 15:30, som gjorde at det var litt trafikk i starten. I tillegg var det en bilbrann på E6 ved Kroppanbrua som skapte en god del kø. Bilbrannen oppstod i midlertid i et nordgående felt, slik at køen løste seg fort opp i den sørgående retningen. Dette resulterte igjen i at semitraileren fikk komme opp i hastighet tidnok før starten på Okstadbakken. Etter dette var trafikken lett og traileren fikk kjøre uforstyrret resten av turen. Temperaturen under forsøket varierte fra 10 grader i Trondheim til 4 grader i Røros. Vegbanen var stort sett tørr. Siden det var flere uker siden forrige forsøk, var snø- og iskanten på Fylkesveg 30 smeltet vekk de fleste stedene. Etter turen, ble resultatene lastet opp og gjennomgått, før togturen tilbake til Trondheim.

Dette kjøretøyet har en enda sterkere motor, en V8 med 650 hestekrefter. Forventningene var at den kom til å ha mindre problemer, og klare flere stigninger uten å miste hastigheten enn ved det forrige forsøket. Dette viste seg også å stemme, men det var allikevel flere plasser hvor stigningen var lang og bratt nok til at også dette kjøretøyet mistet hastigheten. Det er imidlertid interessant å sammenligne de to forsøkene, da vekten er så og si den samme, i tillegg til at ruten var nøyaktig den samme.



**Figur 3.9:** Scania S650 fra Galåen Transport. (Foto: Aamodt)

### 3.4.3 Scania S730

Den 6. april ble det tredje forsøket med Galåen Transport gjennomført. Det skjedde i en 2018 modell Scania S730, vist i Figur 3.10. Den kjører diverse produkter for Felleskjøpet, og ved denne anledningen var den lastet med gjødsel. Den veide totalt rundt 56,7 tonn på denne turen.

**Tabell 3.4:** Nøkkelparametere for Scania S730 fra Galåen Transport. (Statens vegvesen 2022b)

Bil	Scania S730 2018
Reg. Nr.	HF 28354
Egenvekt	24 500 kg
Totalvekt	56 700 kg
Motorstørrelse	537 kW / 730 HK, 16.4L

Startpunktet var fra piren i Trondheim sentrum på morgenen. Derfra gikk turen nordover på E6 til Felleskjøpet på Verdal. Denne strekningen ble ikke logget fordi bilen var tom, da gjødselen skulle hentes på Verdal. Det var like enkelt å sitte på med lastebilen, istedenfor å ta toget. Etter å ha fått ombord lasten, startet tilbaketuren klokka 13:00 til Trondheim og videre til Røros. Trondheim ble passert rundt 14:30 slik at rushen ble unngått. Resten av ruten var heller ikke veldig trafikkert, slik at lastebilen fikk kjøre fritt. Fra starten på loggingen i Verdal til forsøket var over i Røros gikk det totalt 4 timer og 40 minutter, inkludert stopp. Temperaturen startet på 4 grader og holdt seg ganske stabilt gjennom dagen, med et par grader ved Røros ved forsøkets slutt. Vegen var en blanding av fuktig og tørr, med lite eller ingen snø/is i vegkanten på fylkesvegen opp mot Røros. Også denne gangen ble resultatene lastet opp og kontrollert, før turen gikk tilbake til Trondheim med tog.

Selv med veldig stor V8 motor på 730 hestekrefter, slet kjøretøyet ved mange anledninger. Dette skyldes selvsagt den høye vekten på vogntoget. Det ble også en lenger tur, som resulterte i flere stigninger som kan analyseres. For det første kunne turen fra Trondheim og videre til Røros sammenlignes med de to tidligere forsøkene. I tillegg kunne turen fra Verdal til Trondheim sammenlignes med piloten som ble gjort med Børstad Transport fra Steinkjer til Trondheim.



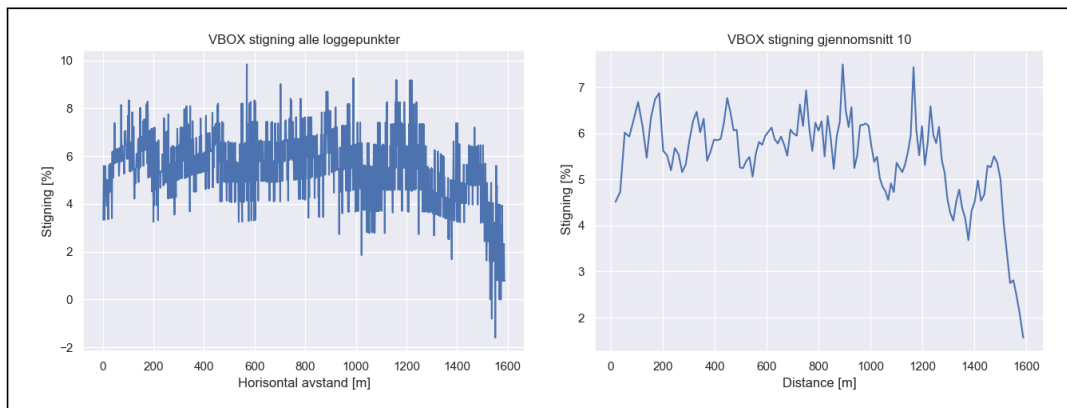
**Figur 3.10:** Scania S730 fra Galåen Transport. (Foto: Aamodt)

## 3.5 Hastighetsmodellen

Med utgangspunkt i arbeidet til Børnes og Aakre, samt Nesdal, utviklet Kogstad og Nygaard hastighetsmodellen i programspråket Python. Hovedpoenget var da å skape en modell som mer realistisk forutser hvordan hastigheten til et tungt kjøretøy endrer seg i en stigning, gitt parameterene til stigningen og kjøretøyet.

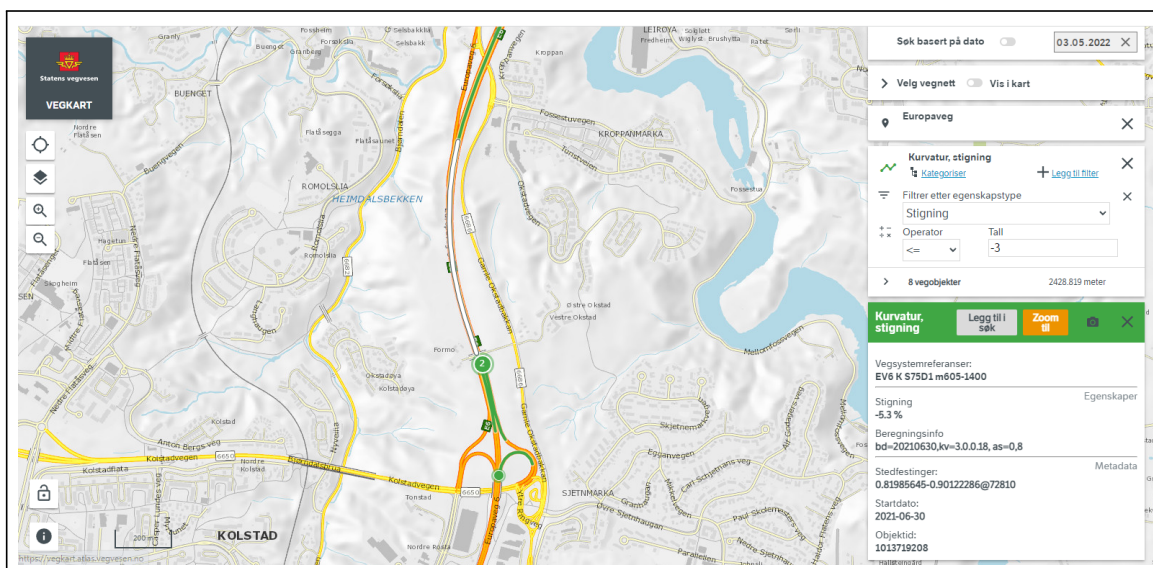
Da modellen er en vital del av oppgaven, er det viktig å vite hvordan den fungerer og hvordan den skal brukes korrekt. Derfor ble det satt av god tid i forprosjektet og i starten av masteroppgaven til å lære seg modellen. Underveis i oppgaven ble det også gjort små endringer og tilpasninger i modellen, der dette ble sett på som hensiktsmessig eller nødvendig. Den nåværende modellen kan sees i Vedlegg C. Videre vil dette delkapittelet gi en grov gjennomgang av både innholdet og bruken av selve modellen.

I innledningen av modellen importeres diverse funksjoner og bibliotek som skal brukes. Dette gjelder blant annet tid, matte og plotting av grafer. I tillegg importeres resultatene fra GPS-en, hastighet og posisjon. Dette gjøres fra en egen Python-fil, `CSV_File_Reader`, se Vedlegg D. I `CSV_File_Reader` blir CSV-filen som ble laget i Kapittel 3.3.1 lastet inn. Videre ”leser” Python gjennom denne CSV-filen og lagrer hastighet, distanse og høyde på tre forskjellige lister. Listen for hastighet for eksempel, inneholder hastigheten for hvert tidels sekund gjennom stigningen. Ved hjelp av distansen og høyden, regnes også stigningen ut. For å unngå for mange punkter og endringer i stigning, tas gjennomsnittet av 10 og 10 punkter. Det resulterer i stigning for hvert sekund. Figur 3.11 illustrerer dette. Her vises stigningen hvor hvert punkt er tatt med til venstre, mens gjennomsnittet av 10 og 10 punkt vises til høyre.



**Figur 3.11:** Alle loggepunkter til venstre og gjennomsnittet av 10 og 10 punkter til høyre.

Hvis posisjonsdata fra GPS-en av en eller annen grunn ikke kan brukes, for eksempel ved dårlig signal, finnes det et alternativ for å utarbeide stigning. Statens Vegvesen har tilgjengelig data for alle veger i Norge via den nasjonale vegdatabanken, NVDB. Et av verktøyene deres er Vegkart. Her er det mulig å bruke søkefunksjonen for å finne det man leter etter, for eksempel stigning. I Figur 3.12 er det søkt på stigning, samt Europaveg, for å fjerne alle de små vegene. I tillegg er det brukt et filter, slik at bare stigninger under -3% vises. Her er det viktig å vite hvilken vei som er positiv, slik at det vises stigninger og ikke nedoverbakker.



**Figur 3.12:** Vegkart fra Statens Vegvesen. (Statens vegvesen 2022e)

Ved å trykke på Objektid nederst i boksen til høyre, åpnes en ny fane med masse tall. Her er alt som står i parenteser etter *LINESTRING Z*, koordinater. Disse kopieres og limes inn i en tom fil i *Windows Notebook*. Her må alle mellomrom fjernes, i tillegg til at alle koordinatene må ha et komma mellom seg. Denne filen lagres og lastes inn i Python-filen *NVDB\_posisjon*, vedlagt i Vedlegg E. Her gjøres dataene om til 3 lister, en av hver for x-, y-, og z-koordinater. Videre brukes dette til å beregne stigningen for hvert punkt.

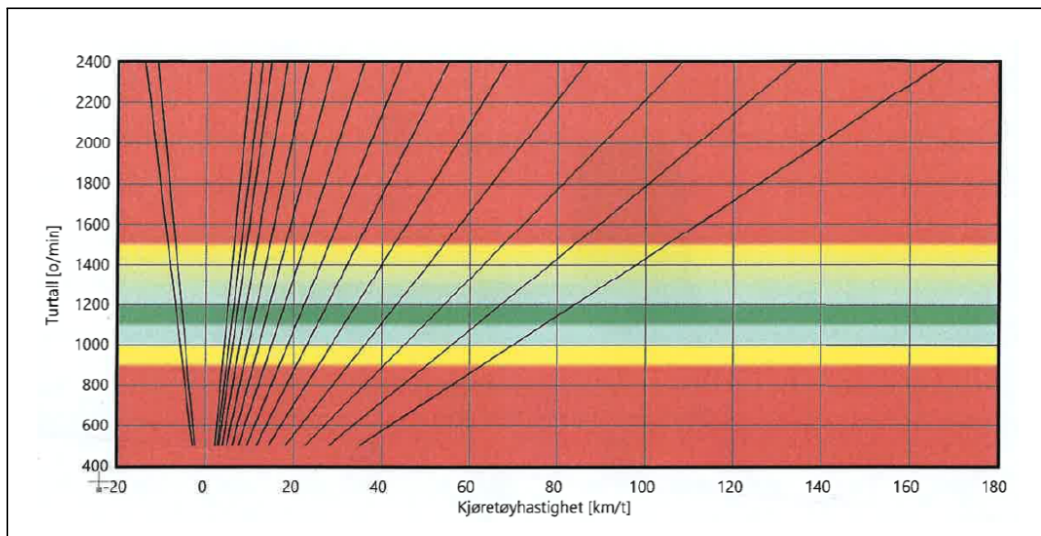
Det ble valgt å bruke stigningen fra GPS-en, da dette ble sett på som enklere, og da begge metodene produserte nesten samme resultat. Hastigheten fra GPS-en måtte hentes ut uansett, så derfor var det enkelt å ta med posisjonsdata samtidig.

Videre i modellen defineres en del funksjoner. Den første funksjonen handler om gir. Ut ifra turtallet i det nåværende giret, bestemmes det om det skal gires opp eller ned, eller om det skal bli værende i det gjeldende giret. Som standard er det slik at det vil gires ned når turtallet er under 1000 omdreininger per minutt, mens det vil gires opp når turtallet overstiger 1400. Disse tallene er valgt basert på motortekniske parametre. Figur 3.13, 3.14 og 3.15 viser henholdsvis girutvekslingskurve, dreiemomentkurve og effektkurve for en Scania R580. Disse figurene er hentet fra oppgaven til Kogstad og Nygaard, som fikk dem tilsendt fra Scania. Dette skjedde i forbindelse med testingen av Scania fra Børstad, den samme som ble brukt under piloten i denne oppgaven. I figurene ser vi at det grønne området er fra 1000-1400, som indikerer ønsket område for turtallet. Figur 3.14 viser også at det er i dette området dreiemomentet er størst. Det er ønskelig og ligge der dreiemomentet er størst mulig fordi det er da drivstofforbruket er lavest. Dette igjen henger tett sammen med økonomi.

I løpet av forsøkene med både Børstad og Galåen, var også erfaringene at bilene stort sett lå mellom 1000 og 1400. Basert på ytterligere erfaringer er det også mulig i modellen å tilpasse når giringen skal skje, både generelt uavhengig av gir, men det kan også være forskjellig ut ifra hvilket gir som er i bruk.

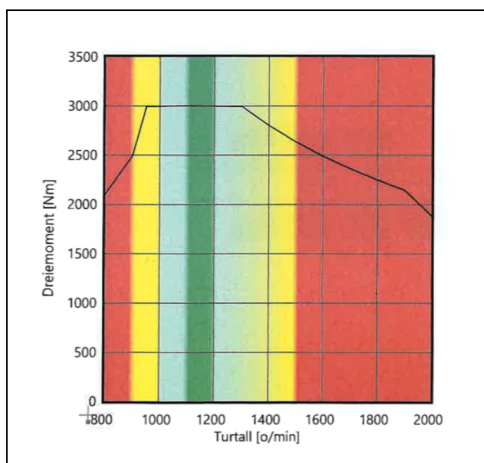
Den neste funksjonen gir en girfaktor, ut ifra det nåværende giret. Er du for eksempel i 12 gir, er girfaktoren 14.1. Figur 3.13 viser girutvekslingen for girene til en Scania R580. Girfaktoren er da stigningstallet til linjene i figuren. Denne faktoren beskriver hvordan turtallet endrer seg i takt med hastigheten, og vice versa, for et gitt gir.



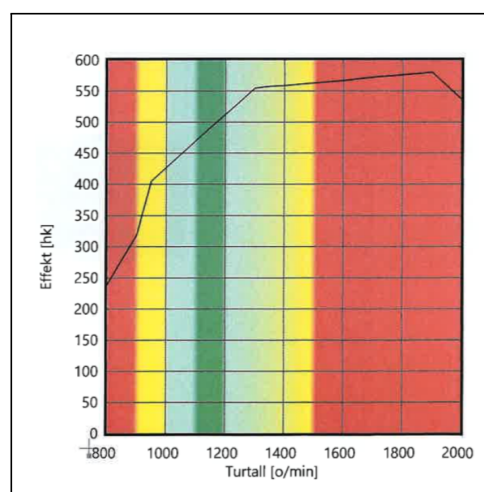


**Figur 3.13:** Girutvekslingskurve for Scania R580. (Kogstad og Nygaard 2021)

Så kommer en funksjon som angir effekten. Denne er basert på linjene i Figur 3.15. Ut ifra det gjeldene turtallet vil denne funksjonen gi den nåværende effekten til motoren. Her ser vi at jo høyere turtallet er, jo større blir effekten. Denne kurven er spesifikk for dette kjøretøyet, og vil se annerledes ut for andre kjøretøy og motorstørrelser.



**Figur 3.14:** Dreiemomentkurve for Scania R580. (Kogstad og Nygaard 2021)

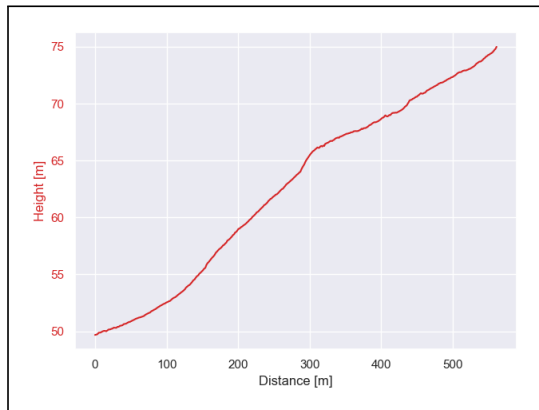


**Figur 3.15:** Effektkurve for Scania R580. (Kogstad og Nygaard 2021)

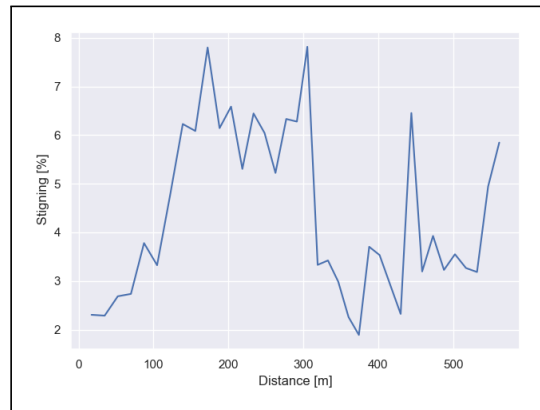
Videre i modellen må alle parametere som skal brukes legges inn. Dette gjelder blant annet startgir, vekt på kjøretøy, vind, tyngdekraft og faktorer i forbindelse med utregning av rulle- og luftmotstand.

Nå som alt det nødvendige er importert, og funksjoner og parametere er definert, kan selve modelleringen starte. Dette gjøres ved bruk av to løkker. Den første løkken er en for-løkke. Denne fungerer slik at man kan gå gjennom stigningen flere ganger. Dette kan være nyttig og i dette tilfellet vil man gå igjennom for den nye modellen, men også for den gamle modellen, slik at de kan sammenlignes. Inne i for-løkken er det en while-løkke. Denne fungerer på den måten at løkken vil bli gjentatt helt til et gitt kriterie er oppfylt. Dette kriteriet er lengden på stigningen. Det vil si at løkken vil bli gjentatt helt til den har vært gjennom hele stigningen. Kort fortalt vil denne løkken starte i bunnen av stigningen og regne ut hastighet, gir, posisjon, høyde, tid, motstand, turtall og effekt for det første punktet. Deretter vil den gå videre til det neste punktet og regne ut de samme parameterene der. Dette vil bli gjentatt for alle punktene, helt til det siste punktet på toppen av stigningen. For hvert punkt i stigningen, lagres hver parameter til en liste. Dette gjør for eksempel at hastigheten gjennom stigningen kan illustreres grafisk. Det er også mulig å finne laveste eller høyeste verdi, for eksempel laveste hastighet, i listen.

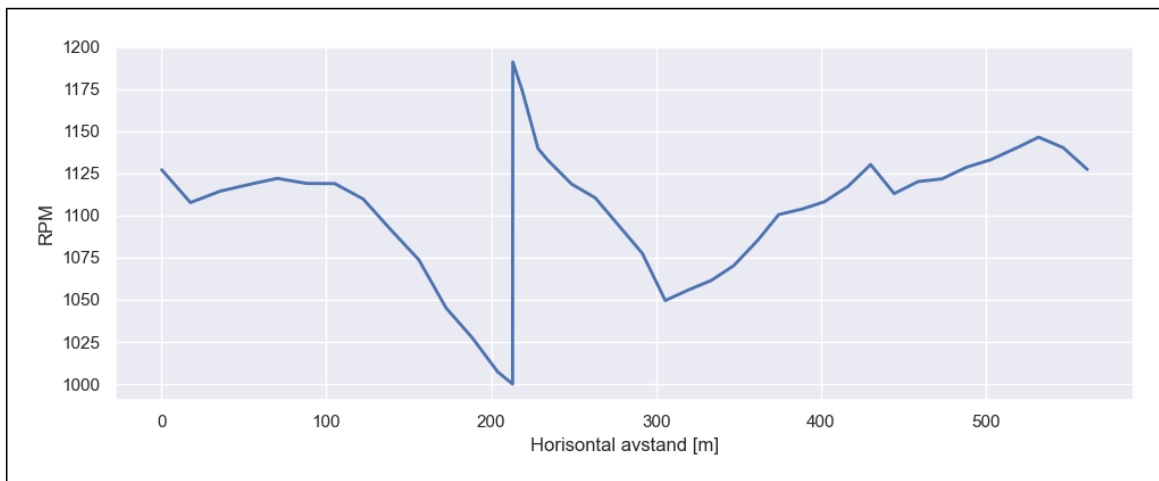
Nå som selve modelleringen er ferdig, kan resultatene i modellen plottes og sammenlignes. Figur 3.16, 3.17, 3.18 og 3.19 viser resultatene fra en stigning på E6 i nærheten av Saltøya, som ligger mellom Stjørdal og Åsen. Figur 3.16 viser høyden, altså hvordan stigningen ser ut. Figur 3.17 viser stigningsprosenten gjennom stigningen, mens Figur 3.18 illustrerer hvordan turtallet endrer seg. Figur 3.19 viser en sammenligning av hastigheten til den nye modellen, 95% av den gamle modellen, 85% av den gamle modellen og resultatet fra GPS-en i VBOX. 95% blir brukt fordi det er denne verdien som legges til grunn for den gamle modellen. 85% brukes fordi dette er verdien Kogstad og Nygaard kom frem til at passet bedre og gav en mer realistisk verdi i forhold til virkeligheten.



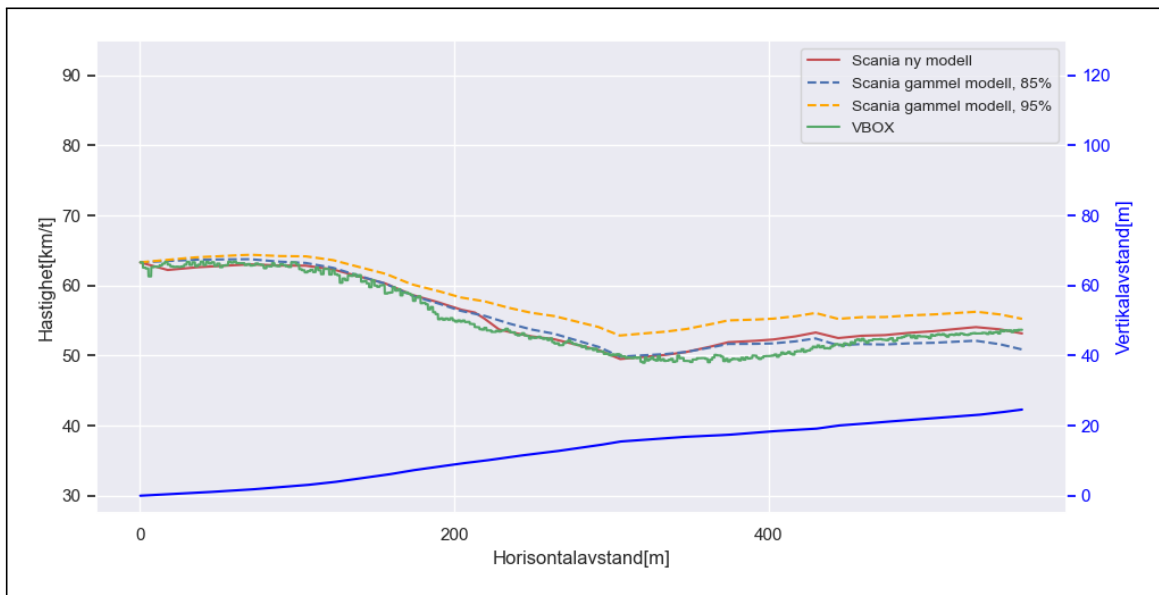
**Figur 3.16:** Høyde gjennom stigningen.



**Figur 3.17:** Stigning ved Saltøya.



**Figur 3.18:** Turtall gjennom stigningen.



**Figur 3.19:** Sammenligning av ny modell, gammel modell og GPS.

Modellen i Python avsluttes med å printe noen parametere. For hver av gjennomkjøringene av stigningen, printes tiden som er brukt opp stigningen, sluthastigheten og minimumshastigheten. Dette gjør at disse parameterene kan sammenlignes opp mot hverandre. Det viktigste er allikevel å sammenligne den nye modellen opp mot den faktiske tiden og hastigheten som ble oppnådd under forsøket. Generelt kan man da si at jo likere tiden opp stigningen er og jo likere minimumshastigheten er, jo bedre stemmer modellen. Det må imidlertid også sees på kurvene, men er tiden og hastigheten lik, er sannsynligheten stor for at kurvene ligner.

I tillegg printes den gjennomsnittlige stigningen, den gjennomsnittlige effekten i den nye og den gamle modellen, samt makseffekten til den nye modellen.

# Kapittel 4

## Resultat

I dette kapittelet vil jeg presentere de viktigste resultatene fra de gjennomførte forsøkene. Forsøkene var forholdsvis lange, som resulterte i veldig mye materiale og resultater. I løpet av forsøkene ble det kjørt mange stigninger, med forskjellige lengde og stigningsprosent. Jeg vil derfor fokusere på de lengste stigningene, stigningene med høyeste prosent og de stigningene hvor hastighetsfallet var størst. De kortere stigningene og de resterende resultatene kan om ønskelig sees i det digitale vedlegget til denne rapporten. Jeg vil starte med å se på stigningene på strekningen mellom Trondheim og Røros. Deretter vil jeg se på strekningen som ble kjørt mellom Verdal og Trondheim.

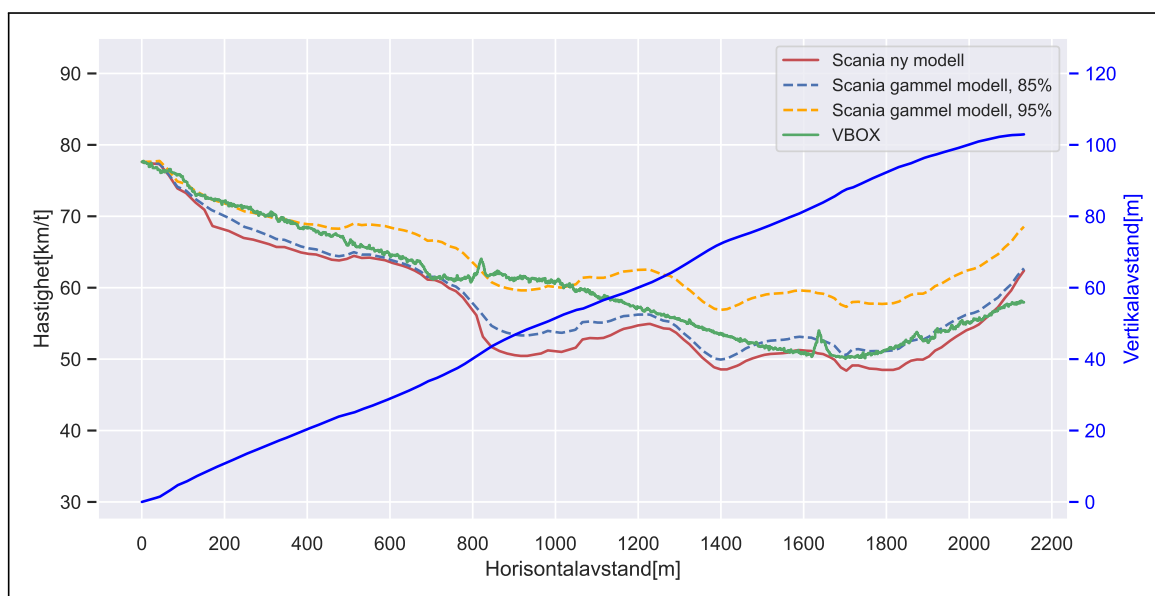
### 4.1 Trondheim-Røros

Strekningen mellom Trondheim og Røros ble kjørt totalt tre ganger, med tre forskjellige kjøretøy fra Galåen Transport. Første gang var med en Scania R580 som veide 42.5 tonn. Andre tur var med en Scania S650 på 42 tonn, mens siste var med en Scania S730 på nesten 57 tonn. De vil videre i dette kapittelet bli referert til ved modellnavnet, R580 osv.

### 4.1.1 Okstad

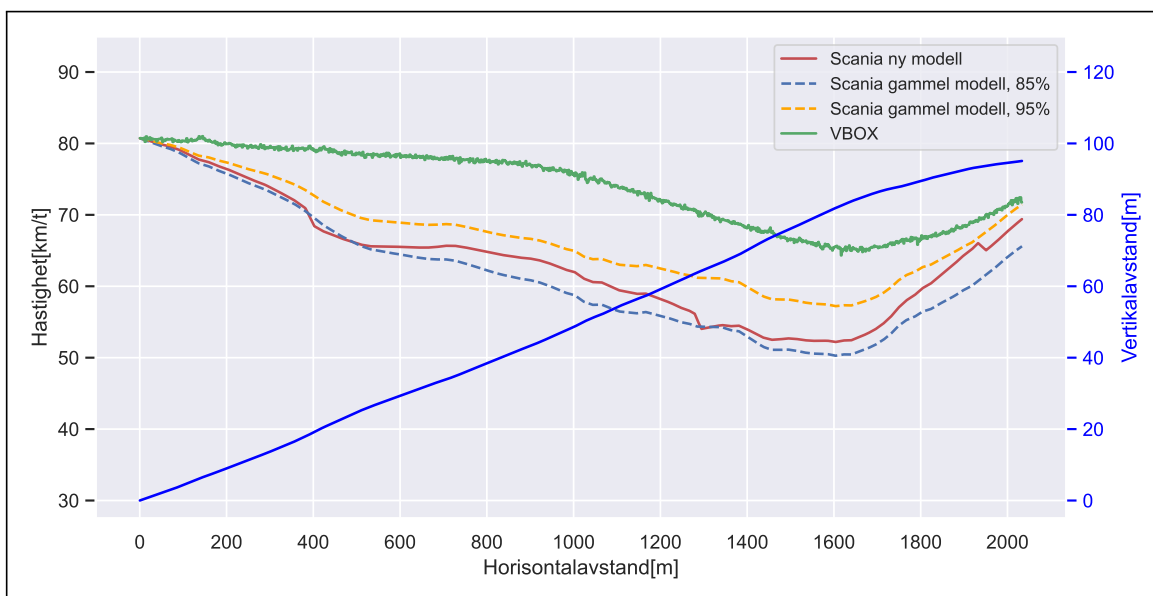
Okstadbakken i Trondheim er over 2 kilometer lang og har en total stigning på rundt 100 høydemeter. Figur 4.1, 4.2 og 4.4 viser hastigheten til henholdsvis R580, S650 og S730. Lengdene på stigningene er litt forskjellige, fordi modelleringen starter der hastigheten begynner å synke og slutter omtrent der stigningen flater ut. Punktene ved start og slutt er satt manuelt på øyemål. I figurene representerer den røde linjen hastigheten til den nye modellen. Den blå stiplede og den gule stiplede linjen illustrerer den gamle modellen ved henholdsvis 85% og 95% av maksimal effekt. Den grønne linjen representerer den faktiske hastigheten som ble målt med GPS-en fra VBOX, mens selve stigningen vises i form av den blå heltrukne linjen.

Figur 4.1 viser resultatet for R580. Her faller hastigheten i modellen kraftig i starten og ligger litt under GPS-en. Etter rundt 800 meter øker stigningsprosenten litt og hastigheten faller kraftig igjen. Deretter øker hastigheten noe før den igjen faller etter 1300 meter. Ved 1800 meter begynner stigningen å flate ut, og hastigheten øker.

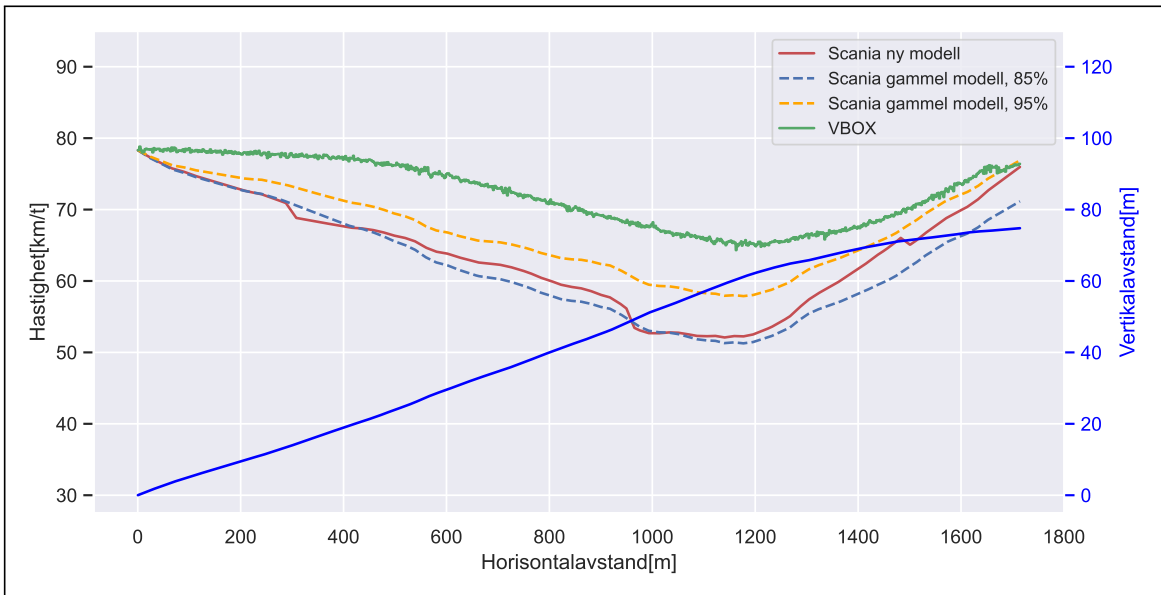


Figur 4.1: Scania R580 i Okstadbakken.

Figur 4.2 viser resultatet for S650. Her faller hastigheten i modellen jevnt frem til cirka 1500 meter før den begynner å øke igjen ved 1700 meter. Som figuren viser er forskjellen mellom GPS og modellen veldig stor. I virkeligheten holdt kjøretøyet hastigheten ganske bra den første kilometeren, før den mistet litt hastighet og så økte igjen. Dette resultatet overrasket litt og modelleringen ble grundig gjennomgått, men det ble ikke funnet feil. Ettersom utviklingen på hastigheten til modellen og GPS-en fra 500 meter og utover lignet på hverandre, ble det forsøkt å starte modelleringen 500 meter senere. Resultatet fra dette er vist i Figur 4.3. Her er resultatet litt av det samme. Hastigheten i modellen synker mens den reelle hastigheten er nokså konstant de første 300 meterene. Etter dette er utviklingen på de to linjene omtrent like, helt frem til slutten der modellen stiger kraftig og slutthastighetene blir omtrent de samme.

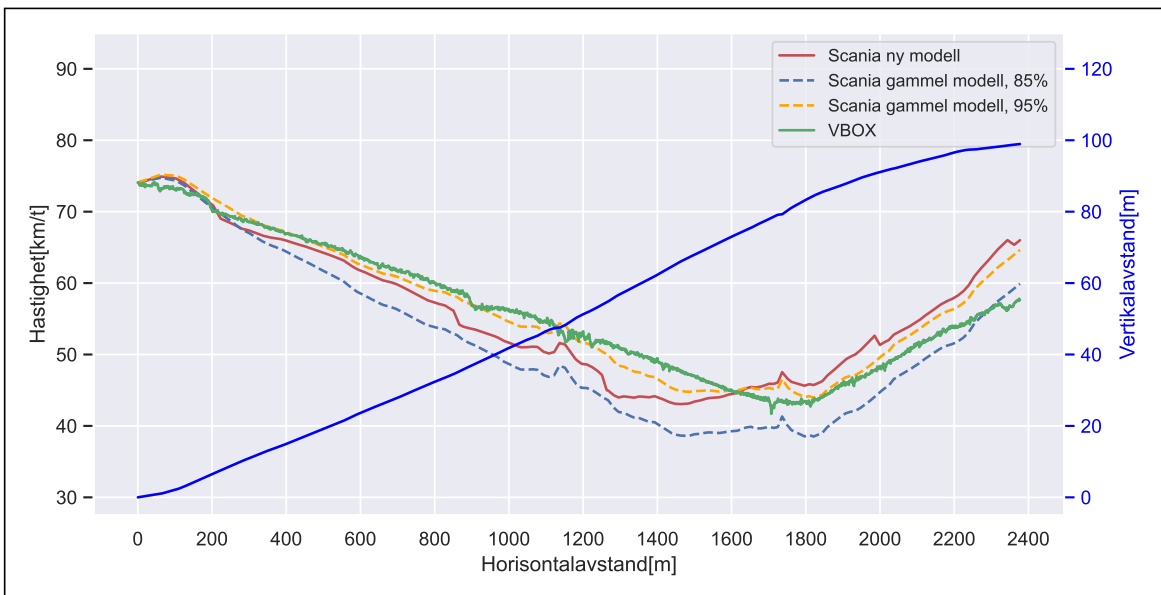


**Figur 4.2:** Scania S650 i Okstadbakken.



**Figur 4.3:** Scania S650 med start 500 meter senere i Okstadbakken.

Figur 4.4 viser resultatet for S730. Her utvikler hastigheten seg ganske likt, men modellen ligger litt lavere den første delen av stigningen. Mot slutten av stigningen når det flater litt mer ut, øker hastigheten mer i modellen enn i virkeligheten, og avslutter litt over.



**Figur 4.4:** Scania S730 i Okstadbakken.



## 4.1.2 Ålen

Høydeforskjellen mellom Trondheim og Røros ligger på i overkant av 600 meter. En stor del av denne stigningen skjer ved Ålen. Over en strekning på 7 kilometer rett etter sentrum, stiger vegen med cirka 250 meter.

Figur 4.5 viser stigningen som ble kjørt med R580. Hastigheten fra forsøket varierer mellom 70 km/t og 50 km/t avhengig av hvor stor stigningen er på de forskjellige punktene. Modellen er hovedsaklig laget for å se på tunge kjøretøy i stigninger. Denne figuren er dermed ikke like relevant, da det innimellom de små stigningene er flate partier eller små nedoverbakker. Resultatet er likevel interessant å se. Hastigheten i modellen ligger stort sett over hastigheten i virkeligheten, men formen på kurvene og partiene hvor hastigheten stiger eller synker, ligner ganske mye på hverandre.



**Figur 4.5:** Scania R580 gjennom hele stigningen ved Ålen.

Stigningen ved Ålen ble delt opp i flere mindre, kontinuerlige stigninger fra forsøkene. Dette resulterte i fire resultater for R580 og S650, mens det for den tyngre S730 ble fem forskjellige stigninger. Tabell 4.1, 4.2, 4.3 og 4.4 gir en oversikt over hvordan modellen gjorde det i forhold til virkeligheten. Det sammenlignes ved å se på laveste hastighet og hvor lang tid stigningen tok. Under tabellene vises en av stigningene fra Ålen for hver av forsøkene.

Tabellene viser at den nye modellen ligger innenfor 1.5 sekund, med unntak av to stig-

ninger, begge med S650. 95% av den gamle modellen ligger generelt litt over både den nye modellen og virkeligheten når det kommer til hastigheten. Den er litt optimistisk. Dette resulterer i at tiden som blir brukt, er litt lavere. 85% av den gamle modellen ligger stort sett ganske nærme den nye modellen, for begge parameterene.

**Tabell 4.1:** Sammenligning av modell og virkelighet ved Ålen\_1.

Ålen_1	R580		S650		S730	
	Tid [s]	V_min	Tid [s]	V_min	Tid [s]	V_min
Virkelighet	40	49,3	40	71,3	38	44,4
Ny modell	41	48,8	48,4	54,3	36,4	38
Gammel modell, 85%	38,8	51,2	49,4	52,7	37	39
Gammel modell, 95%	36,7	55,7	46,5	58,5	35	42,3

**Tabell 4.2:** Sammenligning av modell og virkelighet ved Ålen\_2.

Ålen_2	R580		S650		S730	
	Tid [s]	V_min	Tid [s]	V_min	Tid [s]	V_min
Virkelighet	50	50,5	26	63,4	42	42,2
Ny modell	48,3	50,4	25,6	62,5	40,8	36
Gammel modell, 85%	46,9	54	26,9	57,2	41,3	39
Gammel modell, 95%	44,3	59	26	61,5	38,9	44,1

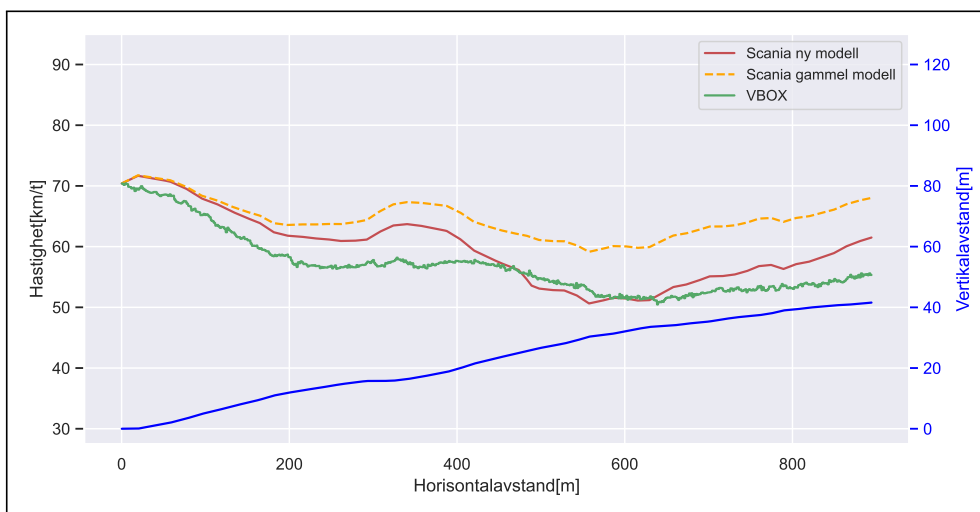
**Tabell 4.3:** Sammenligning av modell og virkelighet ved Ålen\_3.

Ålen_3	R580		S650		S730	
	Tid [s]	V_min	Tid [s]	V_min	Tid [s]	V_min
Virkelig	40	55,3	50	65,9	30	35,8
Ny modell	41,5	49,2	59,3	50,8	29,6	38,7
Gammel modell, 85%	40,8	51,8	61,6	49,5	30,1	38
Gammel modell, 95%	38,6	57	57,4	54,4	28,5	40,7

**Tabell 4.4:** Sammenligning av modell og virkelighet ved Ålen\_4.

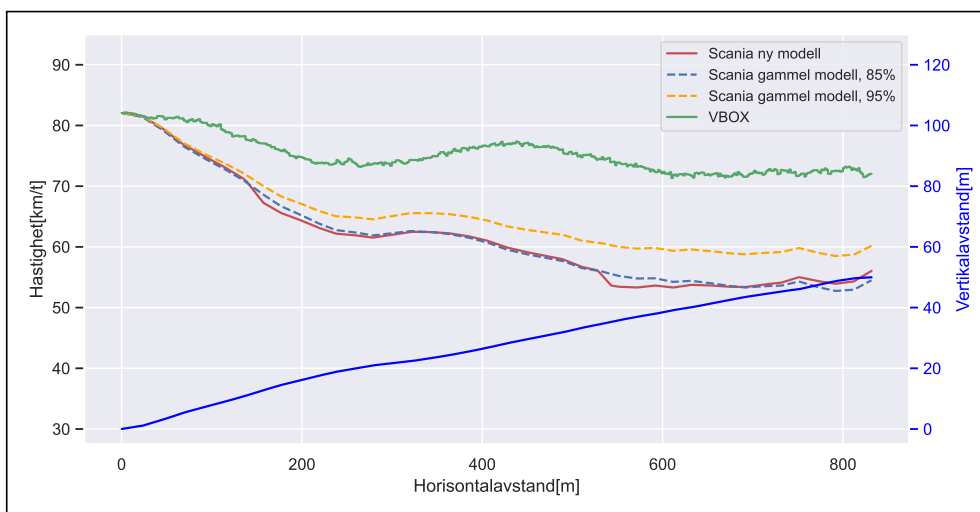
Ålen_4	R580		S650		S730	
	Tid [s]	V_min	Tid [s]	V_min	Tid [s]	V_min
Virkelighet	30	57,3	24	62,6	44	42
Ny modell	30,1	57,8	22,4	67,4	44,1	42,8
Gammel modell, 85%	29,5	59,6	22,8	66,4	48,5	38,4
Gammel modell, 95%	28,4	61,8	22,2	68,5	45,1	42,3

Figur 4.6 viser hastigheten til R580 i den andre av de fire stigningene. Hastigheten i modellen ligger stort sett litt over det som ble registrert i virkeligheten. Av Tabell 4.2 leser man at minimumshastigheten er så og si identisk og at tidsbruken i modellen er halvannet sekund kortere.



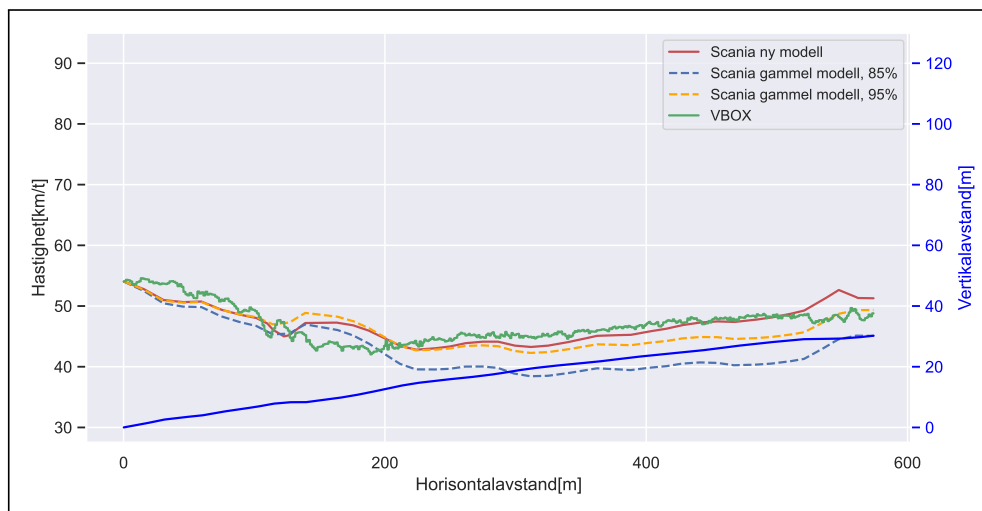
**Figur 4.6:** Scania R580 gjennom Ålen\_2.

Figur 4.7 illustrerer hastigheten til S650 i den første av de fire stigningene. Her ligger hastigheten i modellen vesentlig lavere enn virkeligheten. Tabell 4.1 viser at tidsforskjellen er over 8 sekunder og minimumshastigheten er 17 km/t lavere i modellen.



**Figur 4.7:** Scania S650 gjennom Ålen\_1.

I Figur 4.8 er hastigheten for S730 gjennom den fjerde av fem stigninger vist. Her treffer modellen godt, dette gjenspeiles i tallene i Tabell 4.4. Hastighetsforskjellen er 0.8 km/t mens det bare skiller 0.1 sekund i tidsbruk.



**Figur 4.8:** Scania S730 gjennom Ålen\_4.

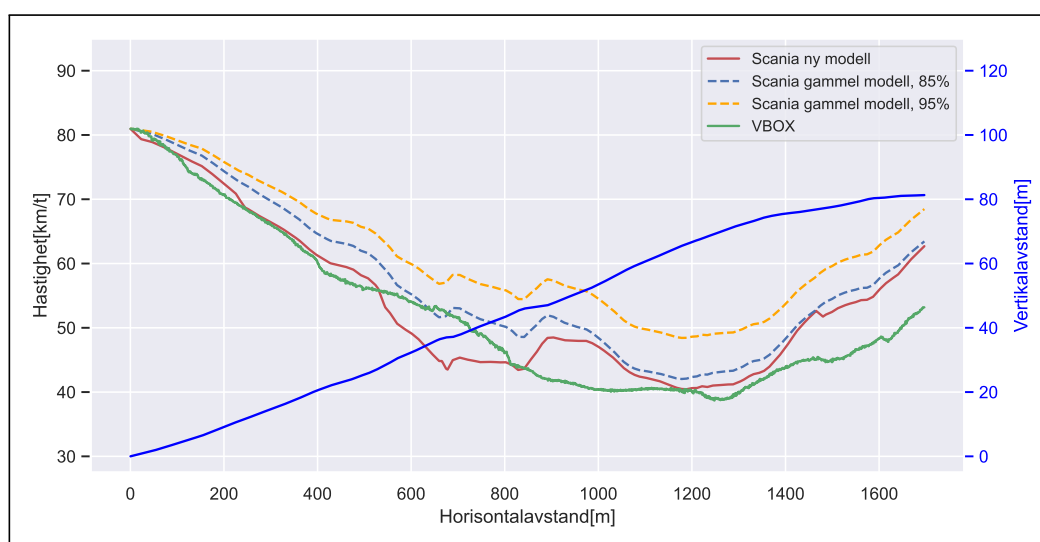
## 4.2 Verdal-Trondheim

Et godt stykke av E6 nord for Trondheim ble kjørt ved to anledninger. Først ble strekningen Steinkjer-Trondheim kjørt med en Scania R580 fra Børstad Transport. Denne bilen kjørte kraftfôr for Felleskjøpet og veide 47.3 tonn. Det andre forsøket var med en Scania S730 fra Galåen Transport, som fraktet gjødsel for Felleskjøpet, og som veide 56.7 tonn. Dette forsøket gikk mellom Verdal og Trondheim. Kjøretøyet gikk også videre til Røros, men resultatene fra den strekningen gjennomgås i Kapittel 4.1.

### 4.2.1 Saltøya

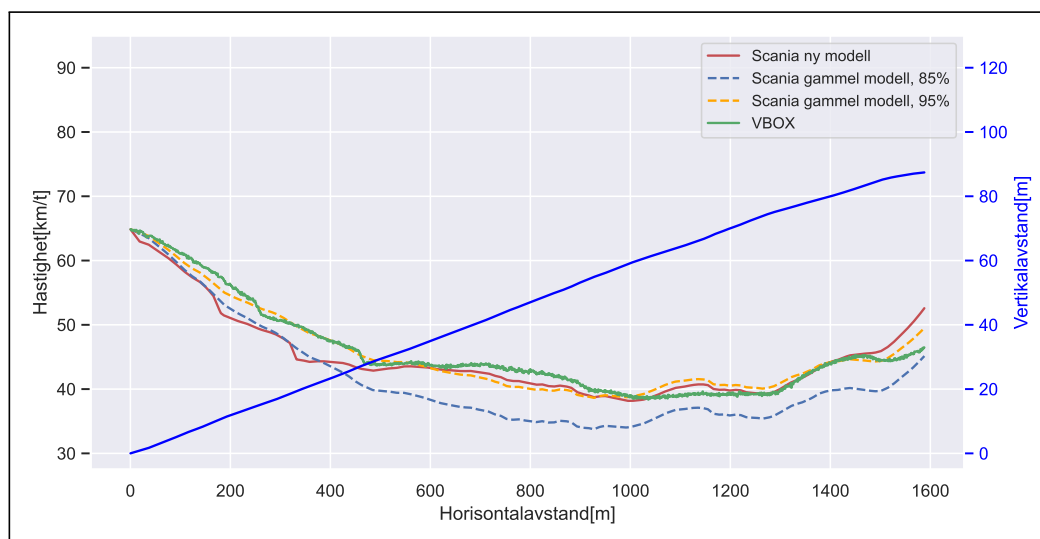
Stigningen på E6 i nærheten av Saltøya, mellom Åsen og Stjørdal, er 1600 meter lang. Den har en gjennomsnittlig stigningsprosent på rundt 4.8% og stiger totalt i overkant av 80 høydemeter. Figur 4.9 og 4.10 viser hastigheten til kjøretøyene for henholdsvis Børstad og Galåen.

Forsøket med Børstad vises i Figur 4.9. I starten av stigningen faller hastigheten i modellen og i virkeligheten ganske likt. Etter 500 meter faller hastigheten i modellen ganske kraftig før den flater ut. Deretter øker den litt før den faller igjen, og sammenfaller med hastigheten i virkeligheten. Når stigningsprosenten faller mot slutten, øker hastigheten for både modellen og virkeligheten, men modellen øker en del mer.



Figur 4.9: Børstad Transport gjennom stigning ved Saltøya.

Figur 4.10 illustrerer hastigheten fra forsøket med Galåen. Hastigheten faller fra 65 km/t til 45 km/t, modellen gjør dette litt raskere enn i virkeligheten. Fra 500 meter og videre, følger kurvene hverandre jevnt nedover til 40 km/t. Etter 1300 meter reduseres stigningsprosenten litt, og hastigheten stiger. De siste 100 meterene øker hastigheten noe mer i modellen enn under forsøket.

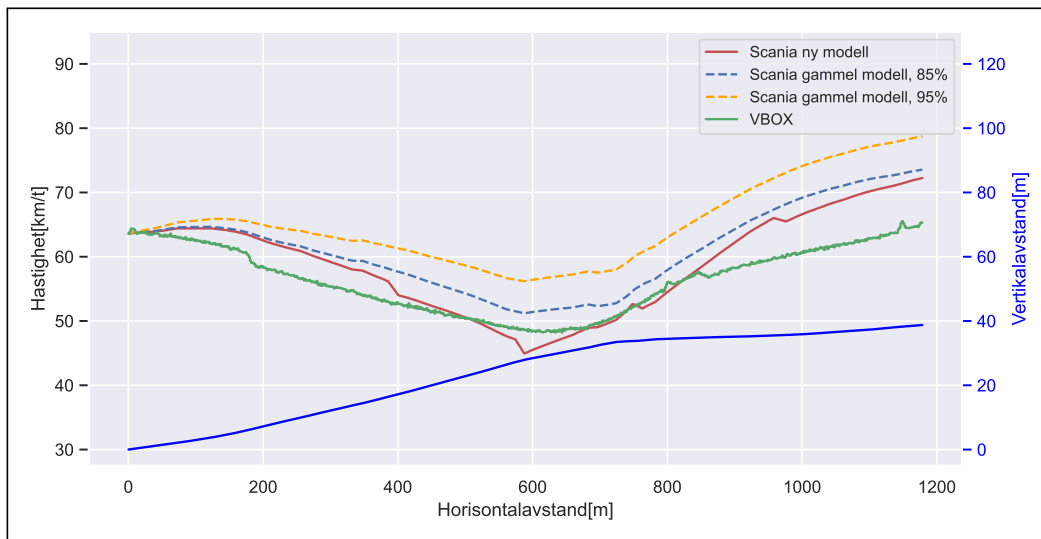


Figur 4.10: Galåen Transport gjennom stigning ved Saltøya.

## 4.2.2 Sveberg

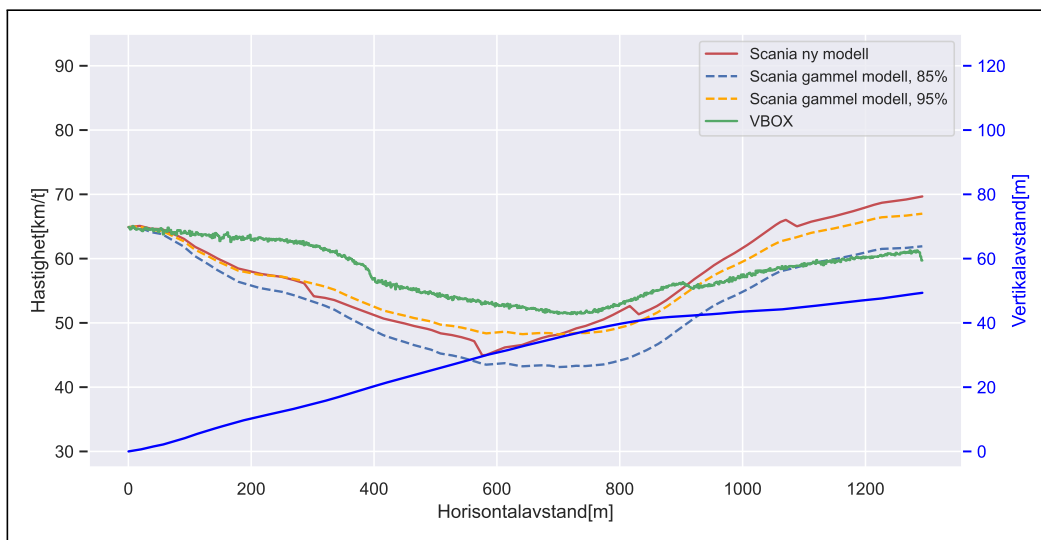
Stigningen på E6 ved Sveberg starter rett etter Stavsjøfjelltunnelen og har en stigningsprosent på cirka 3.4 prosent. Lengden som er målt med Børstad er i underkant av 1200 meter og stiger 40 høydemeter. Lengden som er modellert for Galåen er litt lenger, 1300 meter, og har en stigning på i underkant av 50 meter.

I Figur 4.11 ser man at hastigheten i modellen stiger litt i starten før den synker. Den stemmer ganske bra med det reelle forsøket fra og med 400 meter og frem til 850 meter. Etter dette, stiger hastigheten litt mer i modellen og ender opp 6-7 km/t over virkeligheten.



**Figur 4.11:** Børstad Transport gjennom stigning ved Sveberg.

Figur 4.12 presenterer resultatet fra forsøket med Galåen. Her ligger modellen godt under virkeligheten i begynnelsen og har en laveste hastighet som er ganske mye lavere enn i realiteten. Hastigheten øker etter 600 meter og 900 meter opp i stigningen går den forbi virkeligheten og sluthastigheten er nesten 10 km/t høyere.

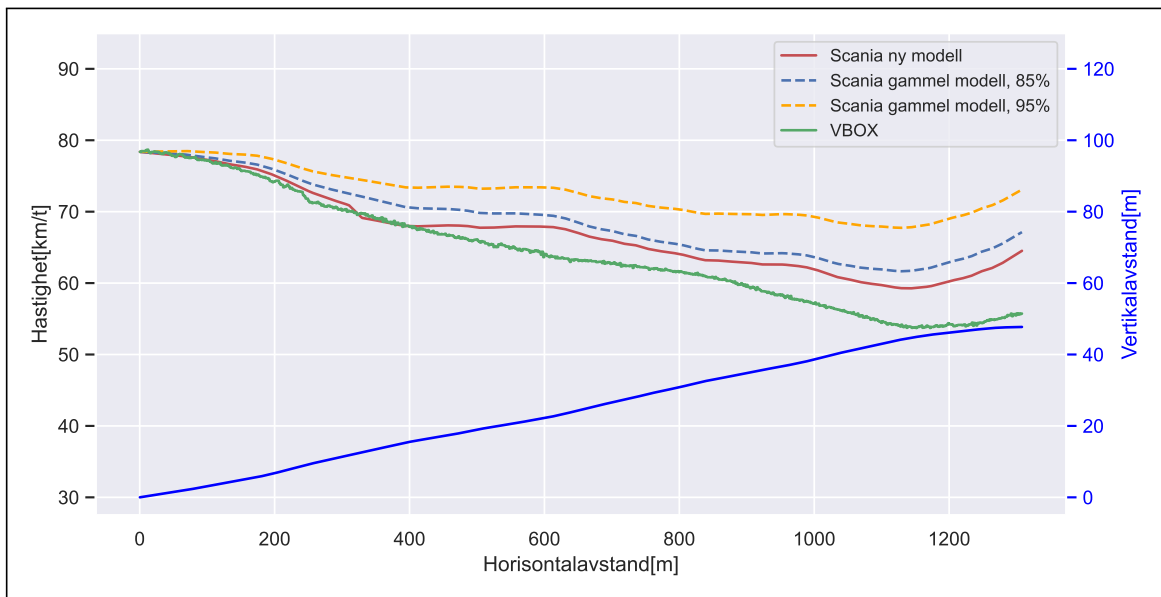


**Figur 4.12:** Galåen Transport gjennom stigning ved Sveberg.

### 4.2.3 Vikhammer

Stigningen ved Vikhammer rett øst for Trondheim har en gjennomsnittlig stigning på 3.6%. Analysen fra forsøket med Børstad ble gjennomført med en lengde på 1.3 kilometer og hadde en økning på 44 meter. Forsøket med Galåen, hvor kjøretøyet veide litt mer, resulterte i en lengde på 2 kilometer og en høydeøkning på omlag 67 meter.

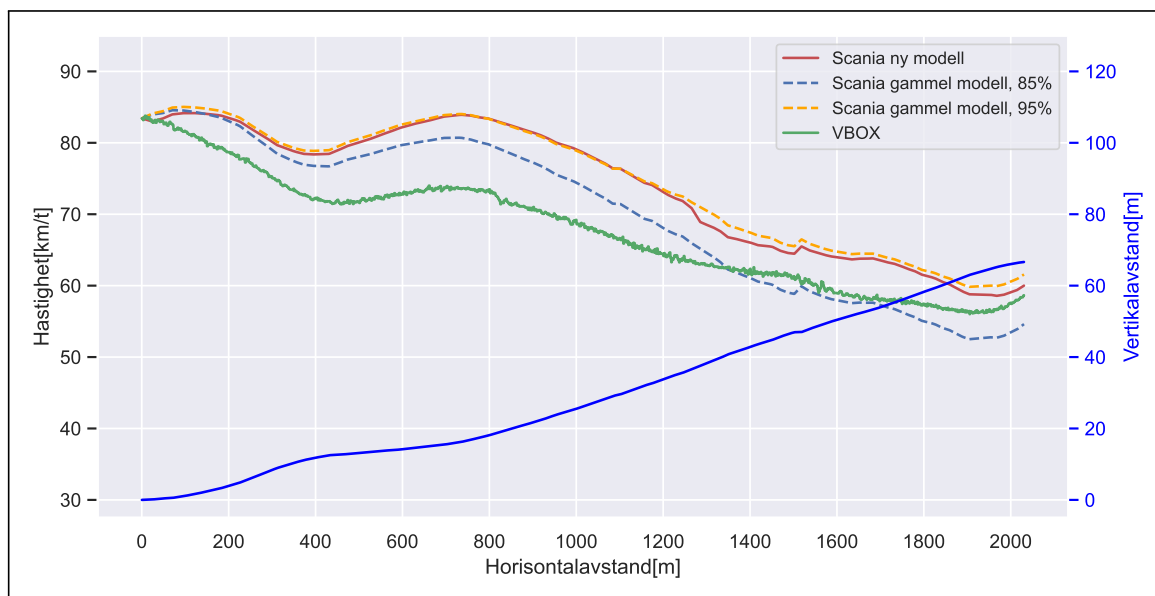
Resultatet fra forsøket med Børstad er illustrert i Figur 4.13. Hastigheten i modellen og i virkeligheten minker, og følger hverandre de første fire hundre meterene. Fra 400 til 600 meter er hastigheten i modellen tilnærmet konstant, mens i virkeligheten fortsetter hastigheten å synke. Videre følger kurvene samme formen fra 600 til 1100 meter, der hastigheten begynner å øke igjen. Slutt hastighetene har en forskjell 8-9 km/t.



Figur 4.13: Børstad Transport gjennom stigning ved Vikhammer.



Figur 4.14 viser resultatet med bilen fra Galåen. Her er målingen startet tidligere enn for Børstad. Det starter å stige før det flater noe ut ved 400 meter, før det igjen stiger litt mer etter 750 meter. Figuren viser at hastigheten i modellen ligger over hastigheten i virkeligheten gjennom hele stigningen, men formen på kurvene er ganske like. Hastighetene nærmer seg hverandre mot slutten av stigningen, og slutt hastighetene er omtrent like.



Figur 4.14: Galåen Transport gjennom stigning ved Vikhammer.

# Kapittel 5

## Diskusjon

I dette kapitlet vil resultatene som er oppnådd diskuteres. Jeg vil starte med å diskutere det som er presentert i resultatdelen, de lengste stigningene. Deretter vil jeg diskutere alle resultatene litt mer generelt. Til slutt drøftes bruksområde, begrensninger tilknyttet metode, samt begrensninger i forbindelse med hastighetsmodellen.

### 5.1 Trondheim-Røros

Selv om strekningen mellom Trondheim og Røros har en stigning på 600 høydemeter, er disse spredt ut over en lang distanse, rundt 150 kilometer. Dette gjør at det er langt mellom de betydelige stigningene. De lengste og bratteste stigningene er hovedsaklig Okstadbakken og stigningene ved Ålen. I tillegg til dette er det mange andre mindre og kortere stigninger langs vegen. Etter å ha tatt av E6 ved Støren blir vegen smalere, mer kupert og svingete. Dette kan hindre kjøretøy i å holde fullt gasspådrag i stigninger som for eksempel i svinger. Det var også is i vegkanten ved det første forsøket, samt delvis i det andre, som påvirker kjøringen.

På grunn av forskjellen i vekt, ble det forskjellig antall resultater for de ulike kjøretøyene. Den tyngste bilen, S730 som var rundt 14 tonn tyngre enn de andre to, slet i vesentlig flere stigninger. Det ble også litt ulike resultater for R580 og S650. Disse veide omtrent det samme, men har en forskjell i motorstørrelse på 70 hestekrefter.

### 5.1.1 Okstad

For R580 ligger hastigheten i modellen litt lavere enn i virkeligheten. Den reagerer også sterkere på variasjoner i stigningsprosenten. Rundt 800 meter inn i stigningen blir det litt brattere, og da faller hastigheten med nesten 10 km/t på kort tid. I virkeligheten er hastigheten mer konstant. Den samme oppstår ved litt slakere partier. Fra 1000 til 1200 meter stiger hastigheten litt i modellen, mens den faller svakt i virkeligheten. Det samme skjer på slutten av stigningen, når det begynner å flate ut. Hastigheten i modellen stiger raskere og mer enn hva tilfellet var i forsøket.

Modellen treffer dårligere for S650. Her ligger hastigheten i modellen godt under hastigheten i virkeligheten. Grunnen kan være den sterke motoren i forhold til vekten. Av Figur 4.2 virker det som om det er de første 500 meterene som avviker mest. Derfor ble det kjørt en ny gjennomgang, med start 500 meter senere. Dette ga litt likt resultat, hastigheten avviker de første 300 meterene, før kurvene er noenlunde like. I tillegg ble det forsøkt å starte modelleringen 10 sekunder tidligere for å se om dette hadde en innvirkning. Det hadde det ikke, og den samme trenden oppsto. For alle disse tilfellene, skjer også det samme som for R580; hastigheten stiger kraftigere når stigningen flater ut på toppen.

Differansen mellom R580 og S650 er også interessant. I virkeligheten skilte det et halvt minutt mellom dem opp Okstadbakken, 130 sekunder mot 100 sekunder. Lengden på stigningen var cirka 100 meter lenger for R580, men 30 sekunder er ganske betydelig. Forskjellen var også stor i forhold til minimum hastighet, 50.2 kontra 64.3.

Resultatet fra S730 viser at her treffer modellen ganske godt. Hastigheten ligger litt under i første del av stigningen, og litt over i andre del av stigningen når det flater ut. Dette resulterte i at det skilte 0.6 sekunder på tiden som ble brukt, mens laveste hastighet hadde en differanse på 1.4 km/t.

En detaljert titt på resultatene fra de tre forsøkene, viser at stigningene ikke er helt like. Stigningen er blitt beregnet for hver gang og for hvert enkelt forsøk ved hjelp av VBOX. Starten og slutten på stigningene ble også valgt individuelt på bakgrunn av fall i hastighet. Det ble vurdert å bruke stigning basert på vegkartet fra NVDB. Dette ble også forsøkt og sammenlignet med VBOX. Resultatet var tilnærmet likt, NVDB traff litt

bedre. Det ble imidlertid sett på som så marginalt at valget var å fortsette med VBOX. Dette er også det enkleste alternativet, da hastigheten allerede hentes fra VBOX. Dette var også den samme konklusjonen Kogstad og Nygaard kom frem til.

### 5.1.2 Ålen

Selv om det stiger mange høydemeter ved Ålen, er det svært varierende stigningsprosent. Det er brattere og slakere partier om hverandre som gjør det vanskeligere å identifisere strekninger som kan analyseres. Figur 4.5 viser hele strekningen med R580. Innimellom stigningene er det her flatere partier, noe modellen egentlig ikke skal brukes til. Hastigheten i modellen ligger over hastigheten i fra forsøket. Men det interessante er at kurvene ligner veldig på hverandre. Dette reiser spørsmålet om modellen kan tilpasses for å modellere større deler av en strekning, inkludert partier uten stigning.

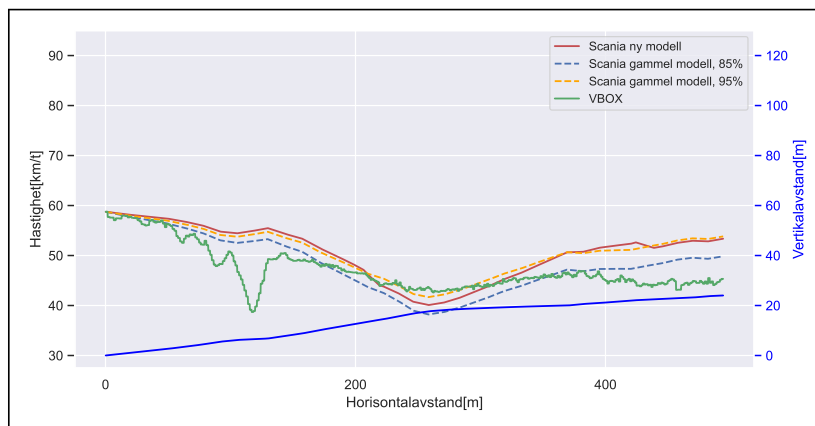
Det ble gjort omtrent 4 modelleringer per kjøretøy, som presentert i tabellene 4.1-4.4 i resultatene. Disse er ikke like for hvert kjøretøy. Ålen\_1 er altså ikke den identiske samme stigningen for alle kjøretøyene, men den første identifiserte stigningen for hver av dem. S650 brukte ikke 50 sekunder på samme stigning som S730 brukte 30 sekunder på ved Ålen\_2, de hadde forskjellig lengde.

Som presentert i resultat-delen, treffer modellen bra på stigningene ved Ålen for R580. Den er innenfor halvannet sekund i forhold til tidsbruken og treffer godt på minimumshastigheten på alle utenom den tredje stigningen. Her er modellen en del lavere en forsøket, men tidsbruken er bare 1.5 sekunder mer.

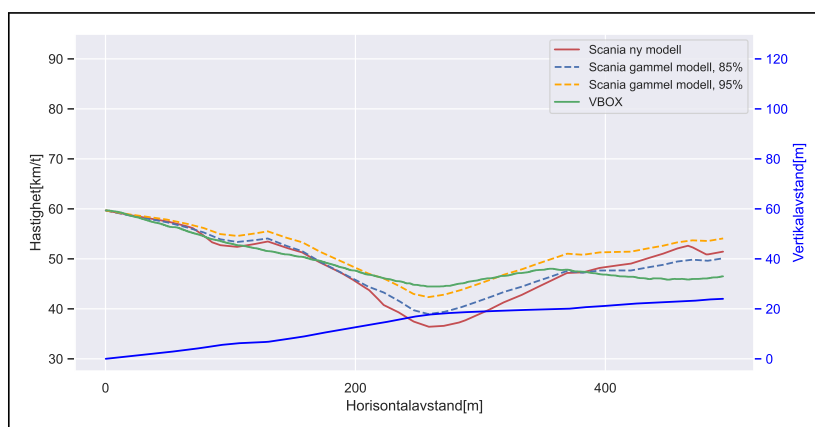
Modellen har litt mer problemer med S650. Den treffer bra på de litt kortere stigningene, den andre og den fjerde, men sliter litt mer med den første og den tredje. Her gir modellen lengre tidsbruk og lavere minimumshastighet. Kurven fra Ålen\_1 som vises i Figur 4.7, har den samme tendensen som oppsto i Okstadbakken. Forskjellen er stor i starten, mens kurvene er likere mot slutten. Her kan problemet ligge i en for sterk motor, men det er samtidig rart at den treffer ganske bra på to av stigningene.

Modellen treffer forholdsvis godt også for S730. Den treffer veldig godt for den fjerde stigningen, som illustrert i Figur 4.8, og er generelt lik angående tidsbruk. Den bommer litt på minimumshastigheten og ligger litt under forsøket på de to første stigningene ved Ålen.

Ved Ålen\_1 ble det et lite hakk i hastigheten fra VBOX, se Figur 5.1. Dette kan skyldes dårlig signal eller andre feil. For å løse dette ble det forsøkt å bruke data fra CANedge1. Dette ble gjort ved å identifisere strekningen og hastigheten i *asammdf*, for så å eksportere denne dataen til en csv-fil, som igjen ble importert inn i modellen. Resultatet av dette vises i Figur 5.2. Her det viktig å merke seg at hastigheten er hakket høyere ved bruken av CANedge1 kontra GPS-en. GPS-en er nok mer nøyaktig da den henter data fra satellitter, mens CANedge1 estimerer hastighet utifra akslingen på kjøretøyet.



Figur 5.1: Scania R580 i Ålen\_1 med bruk av VBOX.



Figur 5.2: Scania R580 i Ålen\_1 med bruk av CANedge1.

## 5.2 Verdal-Trondheim

Strekningen mellom Steinkjer/Verdal og Trondheim foregår på E6. Her er det bred og fin veg, store kurver og delvis 4 felt. Dette gjorde at kjøretøyene kunne kjøre uforstyrret og skulle i prinsippet gi godt grunnlag for modellen. I resultatene, har de tre lengste strekningene blitt illustrert. I tillegg til disse, var det også flere mindre stigninger som gav gode resultater.

### 5.2.1 Saltøya

I stigningen ved Saltøya for kjøretøyet fra Børstad, blir resultat litt av det samme som ble erfart ved stigningene på strekningen mellom Trondheim og Røros. Ved jevn stigning stemmer modellen godt overens med forsøket. Når stigningsprosenten øker eller minker, henholdsvis faller og øker hastigheten mer i modellen enn i virkeligheten. Selv om kurvene ikke følger hverandre hele tiden, er tidsbruken og minimumshastigheten ganske lik. Modellen bruker fire sekunder mindre, 118 kontra 122, og hastighetsforskjellen er på 1.7 km/t.

Resultatet etter forsøket med Galåen stemmer bra. Figur 4.10 viser at kurvene følger hverandre tett hele stigningen. Modellen bruker 1.5 sekunder mer enn i virkeligheten og oppnår nøyaktig samme minimumshastighet. På toppen av stigningen skjer det samme som ved de andre forsøkene, hastigheten øker noe mer i modellen kontra det som skjedde under forsøket.

### 5.2.2 Sveberg

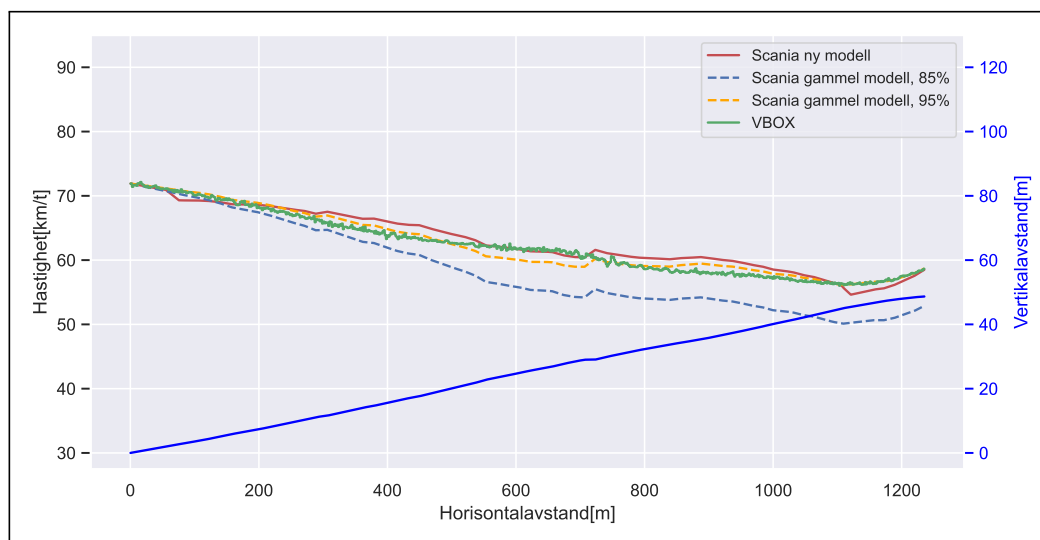
Modellen av forsøket med Børstad i Sveberg stemmer ganske bra med det som skjedde i virkeligheten. Modellen ligger litt over i starten, følger kurven fra forsøket i den midtre delen, før den øker og ligger over igjen i den avsluttende, slakere delen av stigningen.

Den litt tyngre bilen fra Galåen, mister mye hastighet helt i starten og har en betydelig lavere minimumshastighet i modellen enn det som ble oppnådd under forsøket. Når stigningsprosenten synker, øker hastigheten mer enn i virkeligheten og slutthastigheten blir høyere. Tidsbruken mellom modellen og forsøket er relativt lik, 83 mot 81, så på papiret ser det bra ut men, kurvene følger ikke hverandre spesielt bra.

### 5.2.3 Vikhammer

Kjøretøyet fra Børstad treffer bra også i stigningen ved Vikhammer. Kurvene for modellen og fra forsøket følger hverandre tett i starten, før modellen flater litt ut og legger seg litt høyere. Mot slutten skjer det samme som de fleste andre forsøkene igjen, hastigheten øker mer i modellen.

Modellen for bilen fra Galåen treffer ikke like godt. Her faller hastigheten i starten mer i virkeligheten enn i modellen, noe som forplanter seg videre gjennom resten av stigningen. Modellen nærmer seg litt på slutten, slik at minimumshastighetene blir ganske like, men tidsforskjellen er stor, 10 sekunder. Stigningen som er illustrert i Figur 4.14, viser at det først stiger litt. Deretter flater det ut ett lite parti før det øker igjen og er rimelig konstant stigning fra 800 meter og utover. Fra 800 meter og utover ligner kurvene en god del på hverandre. Det ble derfor foretatt en ny modellering, som ble startet 800 meter senere. Figur 5.3 illustrerer resultatet av dette. Her får vi en helt annen utvikling. Kurvene følger hverandre tett hele veien og bruker tilnærmet like lang tid og oppnår samme minimumshastighet. En jevnere stigningsprosent gir altså en mer korrekt modellering av hastigheten i stigningen.



Figur 5.3: Scania S730 fra Galåen ved nedkortet stigning i Vikhammer.

## 5.3 Generelt

De fire forsøkene som har blitt gjennomført, har gitt mange resultater. De lengste stigningene er presentert og diskutert, men det er i tillegg mange andre kortere stigninger som har blitt analysert. Alt dette ligger vedlagt digitalt og kan studeres om ønskelig. Resultatene har generelt sett vært gode, men det er noen ting som går igjen for mange av stigningene.

Modellen virke å treffe best når stigningsprosenten er mer konstant og med mindre svingninger. Hvis stigningsprosenten endrer seg mye i løpet av kort tid er tendensen at modellen reagerer kraftigere og raskere enn det som oppstår i virkeligheten under forsøkene. Et flatere parti i en stigning fører til at hastigheten øker mer i modellen. Dette går igjen mot slutten av stigningene for de aller fleste stigningene som er analysert. Litt av det samme skjer når det blir brattere, hastigheten faller fort. Et girskift vil ytterligere forsterke denne effekten.

Det er også noen resultater som ikke er så gode. For Scania S650 fra Galåen finnes det tilfeller hvor modellen gir en lavere hastighet, blant annet ved Ålen og i Okstadbakken. Dette gir endel lengre tidsbruk opp stigningen, og en lavere minimumshastighet. Dette kan skyldes at vekten er forholdsvis moderat i forhold til motoren som er en ganske sterk V8 motor. Det er imidlertid ikke slik for alle stigningene med denne bilen, noe som setter spørsmålsteget om det er motoren allikevel. Det virker også som om modellen treffer bedre på de kortere stigningene, da de dårlige resultatene produseres for de lengre bakkene.

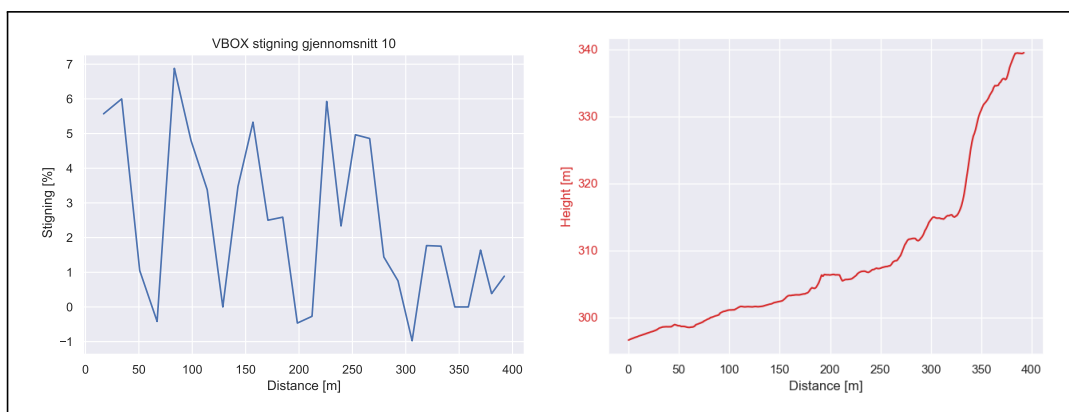
For S730 er tendensen delvis motsatt av S650. I et par av stigningene er modellen litt for optimistisk. Dette gjør at stigningen forseres raskere og minimumshastigheten er høyere. Det skjer imidlertid ikke i like stor grad eller i like mange stigninger.

Selv om ikke modellen treffer perfekt opp mot og i forhold til det som skjer i virkeligheten i forsøkene, betyr ikke dette at modellen ikke er god eller ikke kan brukes. Hvis kurvene for hastighet følger hverandre sånn noenlunde, samt at tidsbruken og minimumshastigheten ligner på hverandre, fungerer modellen etter hensikten. Det er altfor mange elementer og faktorer som spiller inn på hvordan et kjøretøy oppfører seg, til at en modell kan gjenskape en perfekt etterligning. Mange av disse faktorene vil bli diskutert i Kapittel 5.5.2.

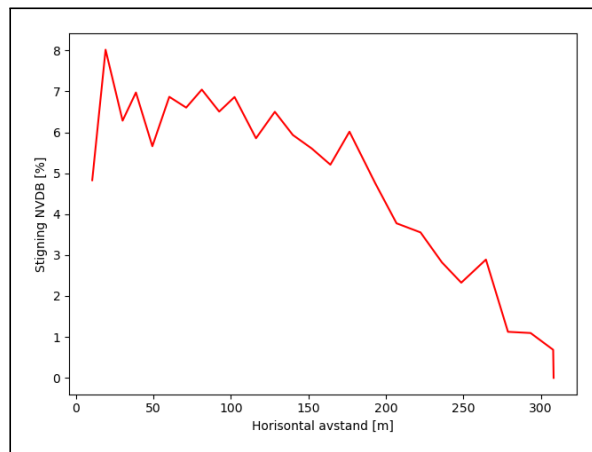


Fremstillingen av stigningene kan gjøres på to ulike måter. Enten ved hjelp av data fra GPS-en i VBOX eller fra vegkartet til vegvesenet sin database, NVDB. Som nevnt tidligere, ble det gjort undersøkelser og metodene ble sammenlignet i forbindelse med modelleringen av Okstadbakken. Forskjellene var små og hadde lite betydning for utfallet av modellen. I tillegg til at fremstillingen av stigningen ved hjelp av data fra GPS-en var relativt enkel på grunn av at hastighetsdata også ble fremstilt, er prosessen med å fremstille stigningen ved hjelp av NVDB relativt omfattende. Dette gjør det til en tidkrevende prosess når det må produseres mange stigninger. I tillegg må starten og slutten på stigningen tilpasses hastigheten, slik at modellen ”treffer” på de samme punktene.

Fordelen med å bruke data fra NVDB er at når en strekning kjøres flere ganger, slik som ble gjort med Trondheim-Røros, trenger stigningen bare å bli fremstilt en gang. En annen fordel er at disse dataene alltid ligger tilgjengelig på vegkartet. Det betyr at hvis GPS-en av en eller annen grunn ikke produserer gode nok resultater, er det alltid mulig å benytte seg av NVDB. Dette skjedde for eksempel i en stigning i nærheten av Veret på Fylkesveg 30 mot Røros. Figur 5.4 viser stigningsprosenten og stigningen som ble utformet ved bruk av data fra VBOX, mens Figur 5.5 viser stigningen produsert ved hjelp av NVDB. Her er det tydelig at det er et eller annet feil med stigningsprosenten fra VBOX som gjør at selve stigningen ser merkelig ut. Hastigheten hadde også en rar dupp akkurat på slutten der. Da gir det mer mening med den jevnere stigningsprosenten som blir fremstilt med hjelp av NVDB.



**Figur 5.4:** Stigningsprosent og stigningen fra VBOX ved Veret.



**Figur 5.5:** Stigningsprosent fra NVDB ved Veret.

I tillegg til hastigheten i virkeligheten og i modellen, viser figurene fra resultatene også to andre hastighetskurver. Dette er resultatene som blir modellert ved bruk av den gamle modellen og den viser resultatene for 95% og 85% av denne modellen. Den gamle modellen antar konstant effekt og bruker 95% av denne i sine beregninger. I tillegg er det valgt å ta med 85%, da dette var en verdi Kogstad og Nygaard kom frem til at var en tilpasning som gav bedre estimater.

Resultatene viser at 95% av maksimal effekt er for mye. Modellen beregner en høyere hastighet og lavere tidsbruk for stigningene. Resultatene viser at 85% stemmer bedre. Den ligner ganske mye på den nye modellen. Det er viktig å nevne at den gamle modellen deler opp stigningen i 5 deler, mens i dette tilfellet er stigningen delt opp i like mange deler som den nye modellen, bare med en konstant effekt.

## 5.4 Bruksområde

En god modell som fungerer etter hensikten har flere potensielle bruksområder. Dette kan for eksempel være et dimensjonerende kjøretøy i håndbøkene til Statens vegvesen. Det nåværende dimensjonerende kjøretøyet ble innført for mange år siden og representerer ikke nødvendigvis det gjennomsnittlige tunge kjøretøyet i dagens kjøretøypark. Hvis det er mulig å skaffe en god oversikt over kjøretøyene som finnes i dag, kan et mer passende kjøretøy bestemmes ved å kjøre denne gjennom modellen.

Et annet aspekt, som henger litt sammen med et dimensjonerende kjøretøy, er utformingen av krabbefelt og påkjøringsfelt. Per dags dato foregår dette på en standardisert måte. Med hjelp av modellen kan et felt tilpasses vegen og hvilke kjøretøy som kjører på strekningen. Dette kan for eksempel være en veg hvor det går mye tømmerbiler som har høy vekt. Da kan dette legges inn i modellen og krabbefeltet vil øke i lengde og passe bedre enn et standardisert utformet felt. På denne måten kan unødvendig kø og farlige forbikjøringer forhåpentligvis fjernes, eller reduseres.

Reisetid er også et tema modellen kan brukes til. Ved å regne på forskjellige kombinasjoner av motorer og forskjellig vekt, kan reisetiden på en gitt strekning modelleres. Modellen er i utgangspunktet ment til å brukes i stigninger, men med litt endringer er det mulig å regne på hele strekninger. Dette kan være nyttig med tanke på valg av kjøretøy. Skal en bedrift gå til anskaffelse av nye kjøretøy, kan modellen hjelpe til å finne det mest fornuftige og økonomiske valget. En større og sterkere bil koster gjerne mer, men vil bruke kortere tid på en strekning. Ut i fra hvor kjøretøyet skal operere kan modellen brukes til å veie fordeler og ulemper for å finne det beste alternativet.

En annen sammenligning som kan gjøres er hvilke kjøreprogram som brukes. Med høye drivstoffpriser er drivstofføkonomi en viktig faktor for enhver bedrift. For forskjellige kjøreprogrammer er det mulig å regne på forskjeller i tidsbruk og drivstofforbruk, for å sammenligne og om mulig finne endringer som kan skape positiv effekt for økonomien.

## 5.5 Begrensninger

I dette kapittelet vil jeg drøfte begrensninger knyttet til den valgte metoden og hva som eventuelt kunne vært gjort annerledes, samt diskutere begrensninger og faktorer i modellen.

### 5.5.1 Metode

Fremgangsmåten og metode som ble bestemt i starten, gikk ut på å gjennomføre en pilot, før ytterligere testing ble planlagt. Det var i utgangspunktet et ønske om å gjøre så mange forsøk som mulig. Da piloten gikk bra og gav gode resultater ble dette sett på som oppnåelig. Sluttantallet var derfor delvis skuffende, da det totalt ble 4 forsøk. Disse forsøkene var imidlertid vellykkede. Lengden på dem både avstandsmessig og tidsmessig gjorde også at det ble mange stigninger og resultater.

Det var flere faktorer som spilte inn og gjorde forsøksutvalget begrenset. Slike forsøk er ganske tidskrevende. Det ble tidlig bestemt at det var helt nødvendig å sitte på og være med på forsøkssturene. Det hadde vært mulig å bare sende med utstyret med en bil, men da er det vanskelig i vite hvilke data som returnerer. Når jeg var med på forsøkene, kunne jeg observere alt som skjedde og notere dette til senere. Opplasting og gjennomgåing av dataene og resultatene i etterkant av forsøkene, ble også mer omfattende og krevende enn først antatt. I tillegg til forberedelser tok dermed ikke et forsøk bare en dag, men ble til flere dagers arbeid.

Forsøkene var også avhengig av transportselskapene. Selv om Jan Erik hadde gode forbindelser med flere selskaper, var det ikke bare å sende en melding for så å gjennomføre forsøket dagen etter. Først og fremst må de være villige til å hjelpe, noe som heldigvis var tilfellet. De kunne ikke tilpasse seg oss, så ut ifra deres timeplan og kjørerute måtte jeg tilpasse meg. Ruten og kjøretøyet kunne heller ikke være hva om helst. For å få en effekt av forsøket måtte ruten som ble kjørt inneholde en viss andel stigninger. Det måtte i tillegg være et kjøretøy som hadde en viss vekt, slik at det fikk hastighetsproblemer i de stigningene som var. En annen utfordring, som på dette tidspunktet var reelt og en faktor å forholde seg til, var den pågående pandemien med Covid-19. Dette skapte utfordringer med sykdom og gjorde planleggingen vanskeligere og mer uforutsigbar.

Et annet mål var å teste ut flere forskjellige lastebilprodusenter og forskjellige konfigurasjoner av motor og vekt. Det ble kjørt med forskjellige motorer og vekt, men bare med det ene merket, Scania. Det ble besluttet å starte med Scania da erfaringer fra tidligere tilsa at dette var den enkleste bilen å få satt opp utstyret i og å få data fra. Dette var også tilfellet under det første forsøket, motorloggeren kunne enkelt kobles til i OBD-porten i førerhytta. Når det videre ble innsett at det ikke kom til å bli like mange forsøk som først planlagt, ble det derfor mest hensiktsmessig å fortsette med Scania på grunn av den enkle oppkoblingen. Erfaring med tilkobling i MAN fra forprosjektet viste at det fungerte, men det blir litt mer jobb. En annen begrunnelse som ble trukket frem var at Scania er, sammen med Volvo, det største merket i Norge. Det ble derfor sett på som meget relevant og et viktig merke.

Sett i ettertid, er det mulig at ting kunne vært gjort annerledes. Jeg kunne forsøkt å kontakte flere selskaper og gjøre dette på et tidligere stadiet av oppgaven. Det kunne også bli vurdert om det var mulig å låne/leie et kjøretøy slik som ble gjort i masteroppgaven fra i fjor. På den måten kunne kjøring blitt gjort på ønsket måte og i ønskede stigninger for å optimalisere tids- og ressursbruk. Selv om jeg ikke ville sende med utstyret uten å være med selv, er dette noe som hadde vært mulig å vurdere. Det er ikke spesielt tidkrevende, og kan gi informasjon om strekningen og/eller kjøretøyet som ble benyttet.

## 5.5.2 Modellen

Hastigheten til et tungt kjøretøy i en stigning avhenger av mange elementer. Modellen tar for seg de viktigste av disse og ser hovedsaklig på stigningsprosent, motortekniske data, vekt, motstand og effekt når den modellerer hastigheten til et tungt kjøretøy i en stigning. Det er imidlertid mange andre faktorer som påvirker hastigheten, i større eller mindre grad. Videre i dette kapittelet vil flere av faktorene diskuteres, både de som er i modellen og de som ikke er og kanskje burde implementeres inn i modellen.

Horisontalkurvaturen er ikke tatt med i modellen, men kan spille en stor rolle. En sving i bunnen av en stigning kan påvirke starthastigheten, som igjen påvirker hastigheten opp stigningen. En sving i selve stigningen kan også tvinge føreren til å senke farten. Hvor krapp og lang svingen er og i hvilken grad dette påvirker hastigheten, kan være interessant å finne ut av.

Bredden på vegen kan også påvirke hastigheten. På en veg med mye trafikk, europaveg eller riksveg, er dette sjelden et problem da vegene er brede og fine. På de mindre vegene, kan en smalere veg påvirke hvordan en stigning blir kjørt. Tunge kjøretøy er også bredere enn vanlige biler og krever mer plass.

Et annet element som er uforutsigbart og vanskelig å modellere er trafikken. Kjøretøy og andre trafikanter rundt deg er med på å påvirke hvordan du kjører og oppfører deg. Blir du liggende bak et annet kjøretøy vil dette påvirke kjøringen i stigningen da hastigheten må tilpasses kjøringen til personen foran. Andre tunge kjøretøy og for eksempel traktorer som kjører sakte, vil tvinge ned hastigheten betydelig. Møtende trafikk kan også skape problemer på de litt smalere vegene, spesielt andre tyngre og brede kjøretøy. Foregår stigningen i et tettbebyggt strøk kan myke trafikanter slik som syklende og gående forstyrre kjøringen. Gangfelt og unger som leker langs vegen er også et element som må hensyn tas i visse områder.

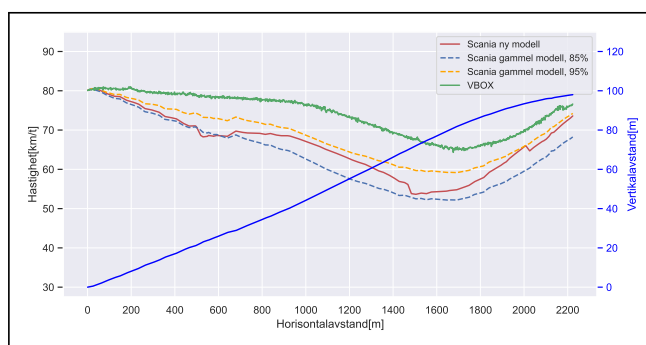
Vær og føre er et annet element som har en innvirkning. Nedbør i form av både regn og snø, gjerne kombinert med vind, fører som regel til litt lavere hastigheter. Snø og is i vegbanen påvirker grepet og friksjonen på underlaget, som videre påvirker kjøringen og hastigheten. Temperaturer rundt 0 grader vil kunne gi glatte veier, som en sjåfør vil ta hensyn til og tilpasse seg etter. Tåke og dårlig sikt er en annen faktor som gjør

kjøring mer krevende og tvinger en fører til å tilpasse seg. Disse elementene er alle viktige, men er vanskeligere å få inn i modellen.

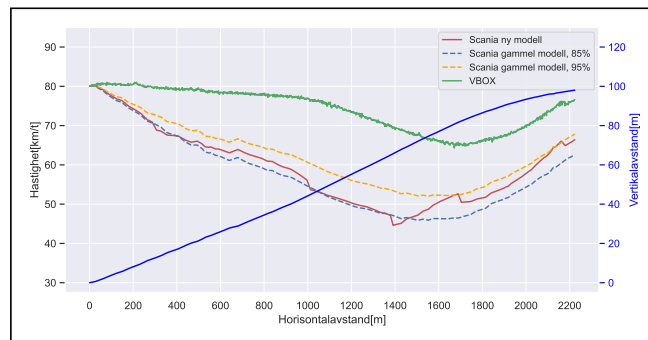
I tillegg til stigningsmotstanden, som tidligere gjennomgått, ser modellen på rulle- motstand og luftmotstand. Rullemotstanden regnes ut ved hjelp av Ligning (2.2) og snittverdier for IRI og MPD. Disse gjennomsnittsverdiene er beregnet på grunnlag av strekningen mellom Klett og Moholt på E6 til henholdsvis 1.2 m/km og 1.3 mm, av Kogstad og Nygaard. Disse verdiene er regnet ut på bakgrunn av data fra Statens vegvesen. Faren ved å bruke disse gjennomsnittsverdiene for andre stigninger enn den de er hentet fra, er avvik fra dette. E6 er en ganske trafikkert og derfor viktig veg. En annen eller mindre veg vil derfor kunne ha andre verdier som avviker fra modellen, som gir utslag. Så fremt vegen ikke er helt ødelagt, vil ikke denne forskjellen utgjøre store endringer, men med mange nok små faktorer kan det til slutt gi større avvik.

Luftmotstanden tar utgangspunkt i standard verdier for overflateareal og luftmotstandscoeffisient. Vindhastighet og retning er vanskelig å måle for hvert enkelt tilfelle, og settes til null i modellen. Figur 5.6, 5.7 og 5.8 viser forskjellen på henholdsvis ingen, mot- og medvind for Scania S650 i Okstadbakken. Det er ganske kraftig vind på 20 sekundmeter medvind og motvind, men vi ser at det blir forskjeller.

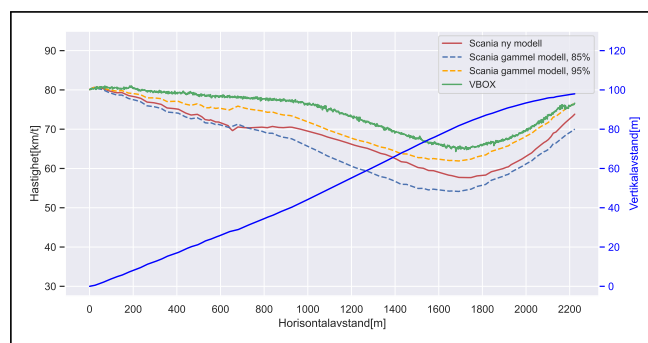
I en stigning vil den desidert største delen av motstandskreftene komme fra stigningen. Det er derfor begrenset resultat å oppnå ved å regne ut nøyaktig luftmotstand for hvert enkelt kjøretøy og stigning. Det vil si at en gjennomsnittlig verdi vil fungere bra.



Figur 5.6: Ingen vind i Okstadbakken.



**Figur 5.7:** Motvind i Okstadbakken.



**Figur 5.8:** Medvind i Okstadbakken.

Det er viktig å kjenne til egenskapene ved selve kjøretøyet, for å få til en god modell. Dette gjelder blant annet motor og vekt. I tillegg til de åpenbare, er det andre elementer som spiller inn, både kollektivt, men også individuelt for hvert kjøretøy. Alderen på bilen kan spille en rolle. Hvor stort er tapet av effekten fra motoren, gjennom akslinger og videre til hjulene. Hvilket kjøreprogram som brukes vil også legge føringer for hvilket turtallsområde som prefereres. Det er heller ikke enkelt å vite hvordan motoren er programert til å gire i de gitte kjøreprogrammene.

Når det gjelder giring, er det fra kjøretøyprodusentene gitt et turtallsområde som kjøretøyet etterstreber å holde seg i. Dette området er ganske vidt, som gjør det vanskelig å modellere nøyaktig. I modellen er det forsøkt å gå inn å endre giringen spesifikt for hvert gir i en stigning, ut ifra data tatt fra denne stigningen. Dette gir naturlig nok bedre resultater enn en generell giring, men avhenger da av data fra akkurat den bilen i akkurat den stigningen. Noe av hensikten er jo å modellere mer generelt, uten å måtte fysisk ut å teste alle kjøretøy.



Dekkene er en annen komponent. Mønsteret og mønsterdybden spiller inn, samt om det er vinterdekk eller sommerdekk. Totalt areal som er i kontakt med underlaget varierer også med antall akslinger og dekk per aksling. Er det tvillinghjul og/eller boggi, hva er størrelsen på hjulene og hva er dekktrykket. Hvor mange akslinger det er drift på er en annen ting som kan være med å påvirke, samt vekten på de ulike akslingene.

En siste viktig del av en slik modellering er føreren. Ulike førere har ulike kjørestiler, en erfaren fører vil mest sannsynlig kjøre annerledes enn en ung og ny sjåfør. En lokalkjent fører vil kjøre på et annet vis enn en fører som ikke er lokal. Dette var blant temaene som kom opp på to av forsøktsturene til Røros. Rett etter en av togovergangene på veg mot Røros er det en sving, og etter denne svingen kommer det en bratt og kort stigning. På innsiden av denne svingen står det et hus, slik at stigningen ikke kommer til syne før dette huset er passert. På vinteren kan denne stigningen skape problemer. De lokalkjente sjåførene sørger for å akselerere raskt etter togovergangen, for å sikre nødvendig hastighet. Begge førerne jeg satt på med påpekte at tidligere vintersesonger hadde utenlandske vogntog satt seg fast her. Årsaken er trolig en kombinasjon av for lav hastighet inn i stigningen, samt et kjøretøy utstyrt med under middels gode dekk.

Elektriske kjøretøy blir mer og mer populære og vanlige i Norge. Dette gjelder først og fremst personbiler, men det jobbes iherdig med de tyngre kjøretøyene også. Foreløpig er rekkevidde en begrensende faktor, slik at det fokuseres på mindre lastebiler til distribusjon og lignende i byområder. I fremtiden er det imidlertid stor sannsynlighet for at teknologien muliggjør også langtransport med elektriske vogntog. En elektrisk motor er veldig annerledes enn en diesel- og bensinmotor, og har ingen tradisjonelle gir. Den har også ofte høyere dreiemoment og vil kunne akselerere hurtigere enn en tradisjonell lastebil. Dette vil kreve litt tilpasning i modellen for å få til.

# Kapittel 6

## Konklusjon

Arbeidet med denne masteroppgaven har hatt som formål å teste og videreutvikle en hastighetsmodell for tunge kjøretøy i stigninger. Dette skal gjennomføres ved hjelp av fysiske forsøkturer og påfølgende modellering av forsøkene. På bakgrunn av dette, har følgende to forskningsspørsmål blitt formulert og undersøkt:

1. *Hvor godt gjenspeiler modellen det som skjer i virkeligheten?*
2. *Hva kan en slik hastighetsmodell brukes til?*

Modellen gjenspeiler virkeligheten generelt på en god måte. Det er umulig å få modellen til å bli 100% korrekt, da det finnes veldig mange faktorer som påvirker hastigheten til et tungt kjøretøy. Resultatene er oppnådd ved bruk av forskjellige kjøretøymodeller fra Scania, og viser at hastigheten i modellen sammenfaller og følger den samme utviklingen som hastigheten i virkeligheten. Denne observasjonen underbygges av tiden som blir brukt opp stigningen, samt likevektshastigheten, den laveste hastigheten som oppnås under stigningen. Disse to parameterene gir en ekstra indikasjon på at modellen fungerer etter hensikten. I de tilfellene hastighetsprofilen i modellen hadde delvis avvik fra det som skjedde under forsøkene, var tidsbruken og minimum hastighet ofte like.

Modellen treffer best når stigningsprosenten er jevn og konstant. Ved endringer i stigningsprosenten innad i en stigning, har modellen en tendens til å reagere noe kraftigere enn det som skjer i virkeligheten. Når prosenten i stigningen øker, vil hastigheten synke raskere i modellen enn i virkeligheten. Det omvendte vil også skje i partier med lavere stigningsprosent, hastigheten øker mer enn hva som er tilfellet i virkeligheten. Dette kommer spesielt frem på slutten av de fleste stigningene. Når bakken begynner å flate ut, vil hastigheten i modellen økes gradvis litt mer.

En fungerende hastighetsmodell for tunge kjøretøy i stigninger har flere potensielle bruksområder. Den kan brukes til å definere et dimensjonerende kjøretøy til bruk i Statens vegvesen sine normaler og håndbøker. Dagens dimensjonerende kjøretøy er basert på standard verdier som ble satt for lenge siden og ikke nødvendigvis er representativt for dagens kjøretøypark. Modellen kan også brukes i dimensjoneringen av krabbefelt og påkjøringsramper. En annen mulighet er å bruke modellen til å regne på reisetid. Sammenligning av forskjellige kjøretøy, motorer og vekt kan være med i beslutningsgrunnlaget når nye kjøretøy skal velges eller gåes til innkjøp av.

# Kapittel 7

## Videre arbeid

I denne oppgaven har det blitt utført multiple forsøk for å teste modellen, men det er fortsatt mulig å jobbe videre med den. Ytterligere testing med flere merker og forsøk vil være med på å forsterke datagrunnlaget og gi modellen mer kredibilitet. Videre undersøkelse av hvordan vekten påvirker hastigheten for ulike motorer er også et interessant tema. Også den motsatte sammenhengen burde utforskes; hva har motorstørrelsen å si for hastigheten for en gitt vekt.

Et alternativ til å kjøre forsøk på ”vanlige” ruter, kan være å kjøre samme stigningen med det samme kjøretøyet i flere omganger. Å gjøre dette på så identisk måte som mulig, for så å analysere og se om resultatene blir like, eller om det finnes individuelle variasjoner.

Undersøkelser rundt et nytt dimensjonerende kjøretøy er et annet element som kan være verdt å belyse. Samtaler og samarbeid med Statens vegvesen kan være nyttig, og et slikt dimensjonerende kjøretøy kan implementeres i håndbøker og normaler.

Et annet aspekt som kan vurderes er å se på utvidelse av hastighetsmodellen. En utvidelse som inkluderer flere eller hele partier av en strekning kan være med på å gi informasjon om spesifikke strekninger og reisetid for ulike kjøretøy.

Som nevnt tidligere i oppgaven, er det svært vanskelig å forutse akkurat ved hvilke tidspunkt og turtall kjøretøyet skifter gir, da det er mange faktorer som spiller inn. Stigning, kjøreprogram, vekt og andre forhold er med på å bestemme giret. En videre utforskning av dette temaet er relevant for modellen. Dette inkluderer også gasspådraget. Dette varierer fra fører til fører, og det varierer også for cruisekontroll. Ved bratte stigninger er det gjerne fullt gasspådrag, men i andre tilfeller og på grunn av andre faktorer som spiller inn, er gasspådraget mindre enn 100%. Derfor kan prosentandel gasspådrag vurderes å bli implementert i hastighetsmodellen.

Med dagens fokus på klima og elektrifisering, vil elektriske lastebiler være interessante for modellen. Disse har ikke tradisjonelle gir og har tilgang til høyt dreiemoment for alle turtall. Fremgang i teknologi fremmer også spørsmål rundt autonomi og førerløse kjøretøy. Dette er også noe som på sikt kan sees på i forbindelse med hastighetsmodellen. Er det for eksempel mulig å bruke modellen til å finne et mest mulig bærekraftig hastighetsprofil. Kolonnekjøring med flere kjøretøy er også et spennende aspekt med fremtidens teknologi.

# Referanser

- Børnes, V. & Aakre, A. (2010). *Description, validation and use of a model to estimate speed profile of heavy vehicles in grades* (tekn. rapp.). Norwegian University of Science and Technology, NTNU. Trondheim.
- CSS Electronics, C. (2021a). asammdf GUI/API [Tilgjengelig fra: <https://www.csselectronics.com/pages/asammdf-gui-api-mdf4-mf4> (Hentet: 9. desember 2021).].
- CSS Electronics, C. (2021b). CAN Bus Explained - A Simple Intro [Tilgjengelig fra: <https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial> (Hentet: 9. desember 2021).].
- CSS Electronics, C. (2021c). CANedge1 [Tilgjengelig fra: <https://www.csselectronics.com/products/can-logger-sd-canedge1> (Hentet: 9. desember 2021).].
- CSS Electronics, C. (2021d). J1939 Explained [Tilgjengelig fra: <https://www.csselectronics.com/pages/j1939-explained-simple-intro-tutorial> (Hentet: 9. desember 2021).].
- Dysvik, E., Homleid, T. & Andreev, L. (2021). Kvaliteten på det norske veinettet [Tilgjengelig fra: <https://s3-eu-west-1.amazonaws.com/opplysningsraadet-for-veitrafikk/images/Kvaliteten-p%C3%A5-det-norske-veinettet.pdf> (Hentet: 7. desember 2021).].
- Klein-Paste, A. (2021). Tire-Pavement-Friction [Tilgjengelig fra: <https://ntnu.blackboard.com> (Hentet: 8. desember 2021).].
- Kogstad, J. & Nygaard, S. W. (2021). *Datadrevet utvikling av hastighetsmodell for tunge kjøretøy i stigning* (Masteroppgave). Norges teknisk-naturvitenskapelige universitet, NTNU.
- Nesdal, A. (2019). *Modellering av fartsprofil til tunge kjøretøy i stigning* (Masteroppgave). Norges teknisk-naturvitenskapelige universitet, NTNU.

- Racelogic. (2020). VBOX Mini [Tilgjengelig fra: [https://www.racelogic.co.uk/\\_downloads/vbox/Datasheets/Data\\_Loggers/RLVBM01\\_DATA.pdf](https://www.racelogic.co.uk/_downloads/vbox/Datasheets/Data_Loggers/RLVBM01_DATA.pdf) (Hentet: 9. desember 2021).].
- Racelogic. (2021). About us [Tilgjengelig fra: <https://vboxautomotive.co.uk/index.php/en/about> (Hentet: 9. desember 2021).].
- Samferdselsdepartementet. (2021). *Forskrift om bruk av kjøretøy* [Tilgjengelig fra: <https://lovdata.no/dokument/SF/forskrift/1990-01-25-92?q=bruk%20av%20kj%C3%B8ret%C3%B8y> (Hentet: 7. desember 2021).].
- Shaddiq, K. (2013). *Kjøretøyparametere i drivlinjen på moderne kjøretøy* (tekn. rapp.). NMBU. Ås.
- SNL, S. n. l. (2019). Drivkraft [Tilgjengelig fra: <https://snl.no/drivkraft> (Hentet: 8. desember 2021).].
- SNL, S. n. l. (2020). Luftmotstand [Tilgjengelig fra: <https://snl.no/luftmotstand> (Hentet: 8. desember 2021).].
- SNL, S. n. l. (2021). Newtons lover [Tilgjengelig fra: [https://snl.no/Newtons\\_lover](https://snl.no/Newtons_lover) (Hentet: 8. desember 2021).].
- Statens vegvesen, S. (2019). Håndbok V120 Premisser for geometrisk utforming av vegger [Tilgjengelig fra: <https://www.vegvesen.no/globalassets/fag/handboker/hb-v120-mai-2019.pdf>].
- Statens vegvesen, S. (2021). N100 Veg- og gateutforming [Tilgjengelig fra: <https://svv-cm-sv-apppublic-prod.azurewebsites.net/product/859922/nb>].
- Statens vegvesen, S. (2022a). Sjekk kjøretøyopplysninger, HF27558 [Tilgjengelig fra: <https://www.vegvesen.no/kjoretoy/kjop-og-salg/kjoretoyopplysninger/sjekk-kjoretoyopplysninger/?registreringsnummer=hf27558> (Hentet: 2. mars 2022).].
- Statens vegvesen, S. (2022b). Sjekk kjøretøyopplysninger, HF28354 [Tilgjengelig fra: <https://www.vegvesen.no/kjoretoy/kjop-og-salg/kjoretoyopplysninger/sjekk-kjoretoyopplysninger/?registreringsnummer=hf28354> (Hentet: 6. april 2022).].
- Statens vegvesen, S. (2022c). Sjekk kjøretøyopplysninger, HF28568 [Tilgjengelig fra: <https://www.vegvesen.no/kjoretoy/kjop-og-salg/kjoretoyopplysninger/sjekk-kjoretoyopplysninger/?registreringsnummer=hf28568> (Hentet: 23. mars 2022).].

- Statens vegvesen, S. (2022d). Sjekk kjøretøyopplysninger, VJ17171 [Tilgjengelig fra: <https://www.vegvesen.no/kjoretoy/kjop-og-salg/kjoretoyopplysninger/sjekk-kjoretoyopplysninger/?registreringsnummer=vj17171> (Hentet: 31. januar 2022).].
- Statens vegvesen, S. (2022e). Vegkart [Tilgjengelig fra: <https://vegkart.atlas.vegvesen.no/>].
- Statistisk Sentralbyrå, S. (2021). Innenlandsk transport [Tilgjengelig fra: <https://www.ssb.no/transport-og-reiseliv/landtransport/statistikk/innenlandsk-transport> (Hentet: 7. desember 2021).].
- Wikipedia. (2020). Hestekraft [Tilgjengelig fra: <https://no.wikipedia.org/wiki/Hestekraft> (Hentet: 8. desember 2021).].
- Wikipedia. (2021). Rolling resistance [Tilgjengelig fra: [https://en.wikipedia.org/wiki/Rolling\\_resistance](https://en.wikipedia.org/wiki/Rolling_resistance) (Hentet: 8. desember 2021).].
- Aamodt, K. (2021). *Modellering av tunge kjøretøy i stigninger* (tekn. rapp.). Norwegian University of Science and Technology, NTNU. Trondheim.



# Vedlegg

Vedlegg A - Oppskrift posisjonsdata

Vedlegg B - Oppskrift motordata

Vedlegg C - Hastighetsmodell

Vedlegg D - CSV\_File\_Reader

Vedlegg E - NVDB\_posisjon

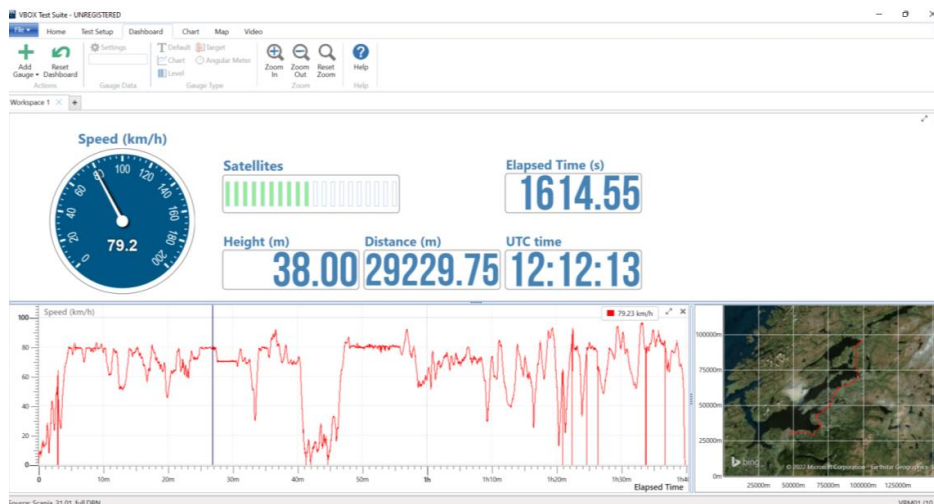
# **Vedlegg A**

## **Oppskrift posisjonsdata**

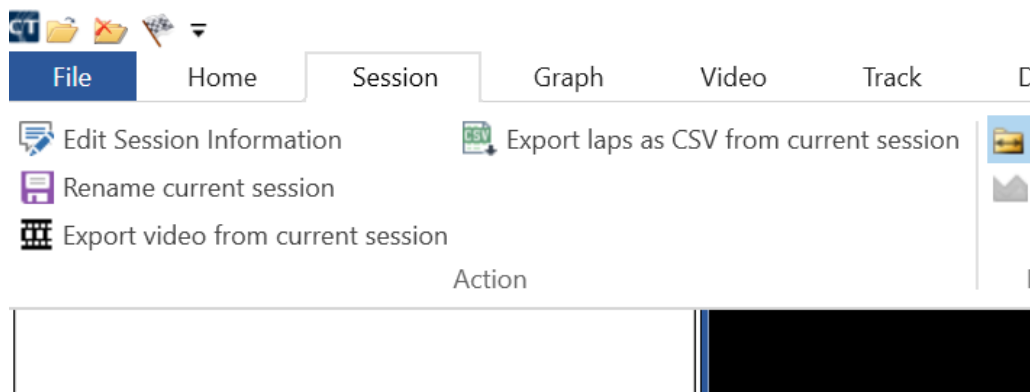
## VEDLEGG A. OPPSKRIFT POSISJONSDATA

### Oppskrift – fremstilling av posisjonsdata

1. Data fra GPS-en VBOX Mini blir logget på et minnekort som en DBN fil
2. Filen lastes inn i et program fra Racelogic som heter *VBOX Test Suite*  
Link til nedlastning: <https://www.vboxautomotive.co.uk/index.php/en/customer-area/software#data-analysis-and-post-processing>
3. Når filen er lastet inn i programmet ser vinduet slik ut:



4. Hvis man vil se på en spesifikk del eller strekning, for eksempel en bakke, kan man klippe i filen. Dette gjøres ved å trykke der du vil starte (blå strek i figur) og under fanen chart bruker funksjonen *Cut – before cursor*. Deretter markerer du hvor du vil stanse og bruker funksjonen *Cut – after cursor*. Da står du igjen med den delen du ønsker å analysere og kan lagre den som en egen fil.
5. For å få posisjon og hastighet inn i modellen i python, må filen eksporteres til en Excel CSV-fil. Dette gjøres ved hjelp av et annet program, *Circuit Tools*, som kan lastes ned fra samme sted som *VBOX Test Suite*.
6. I *Circuit Tools* laster du inn den lagrede filen. Under fanen *Session* brukes funksjonen *Export laps as CSV from current session*.

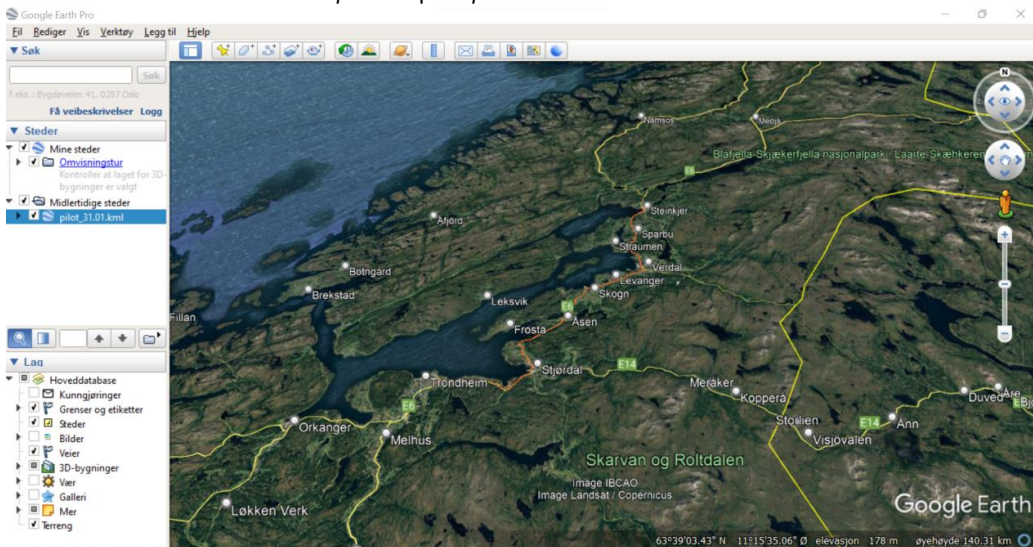


7. Når denne er eksportert, må excel-filen åpnes. Før filen kan lastes inn i python må den ha riktig «oppsett». Dette gjøres i excel ved å fjerne de 7 øverste radene i filen, se figur:

## VEDLEGG A. OPPSKRIFT POSISJONSDATA

num_state	state	state	state	hh:mm:ss.ss	ddd°mm'm.mmmmm'	ddd°mm'm.mmmmm'	km/h	deg	m	km/h	Sats	UTC	Degrees	Degrees	m	s	g	g	
008	0	0	0	13:37:12.80	63°00'25.55077000 N	10°00'40.80993000 W	078.380	271.07	+00142.12	000.450	-5.900000E-01	-5.900000E-01	1.900000E-01	0.000000E+00	0.000000	0.000000	+00.04	+00.02	+00.03

8. Nå kan filen lastes inn i python-modellen som heter: *CSV\_File\_Reader*
9. Ruten som er kjørt kan også lastes inn i programmet Google Earth Pro. Da vil du få en linje i google som viser hvor det er kjørt, se figur under. For å laste inn filen i Google Earth må du i VBOX Test Suite under fanen *map* klikke på *Export to KML*.



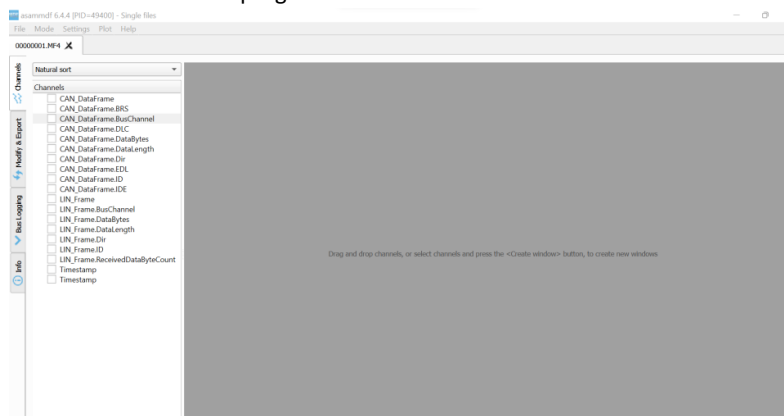
## **Vedlegg B**

### **Oppskrift motordata**

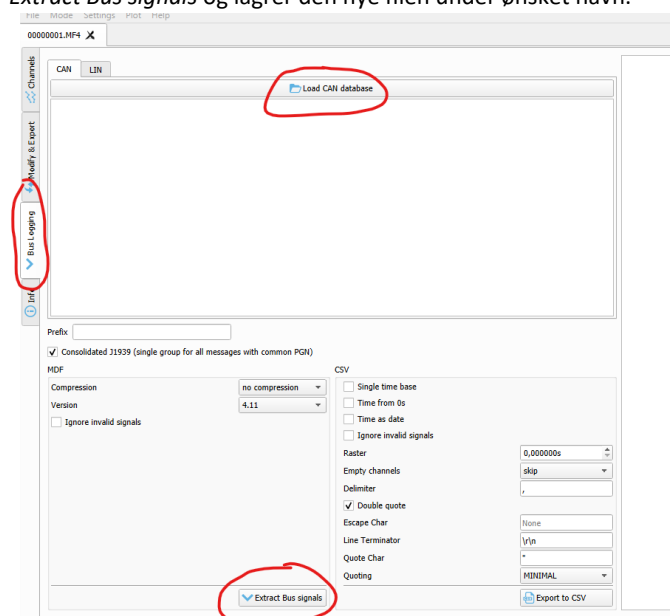
## VEDLEGG B. OPPSKRIFT MOTORDATA

### Oppskrift – fremstilling av motordata

1. Data fra motorloggeren CANedge1 blir logget på minnekortet som en MF4 fil
2. Filen lastes inn i programmet asammdf  
Link til nedlastning av programmet: <https://www.csselectronics.com/pages/asammdf-gui-api-mdf4-mf4#void>
3. Når filen er lastet inn i programmet ser vinduet slik ut:



4. Disse dataene er ikke leselige og trenger å bli konvertert. Dette gjøres ved hjelp av en J1939 DBC fil.
5. For å laste inn denne filen må man gå til *Bus Logging* til venstre i menyen. Her kan du laste inn konverteringsfilen ved å trykke på *Load CAN database*. Når dette er gjort trykker man på *Extract Bus signals* og lagrer den nye filen under ønsket navn.



6. Nå får du opp alle kanalene som er logget og kan dra dem inn i det grå vinduet for å vise dataene enten som en graf, i en tabell eller numerisk. Hvis du vil ha mer info om en spesifikk kanal, kan du dobbeltklikke på den og et nytt vindu med info dukker opp.
7. I tillegg finnes det veldig mye og oversiktlig informasjon på nettsidene til CSS: <https://www.csselectronics.com/>

## **Vedlegg C**

# **Hastighetsmodell**

## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
1 import pandas as pd
2 import math
3 import time
4
5 start = time.time()
6 import matplotlib.pyplot as plt
7
8 plt.rcParams.update({'font.size': 12})
9 import CSV_File_Reader
10 import IRI_MPD
11 # import NVDB_posisjon
12 import numpy as np
13 from statistics import mean
14 import seaborn as sns
15
16 sns.set_theme()
17 # sns.set_style('')
18 import matplotlib.ticker as ticker
19
20 save_results_to = '/Users/kbraa/OneDrive/Dokumenter/SkoLe/NTNU/Semester 4 - MASTER/'
21 import math
22
23 # -----
24
25 # --- Standard ---
26 def get_Gir_standard(Gir):
27     if rpm < 1000 and d == 0:
28         Gir = Gir - 1
29     elif rpm > 1400 and Gir != 12 and d == 0:
30         Gir = Gir + 1
31     return Gir
32
33 # --- Okstad ---
34 # def get_Gir_Okstad(Gir):
35 #     if rpm < 1094 and Gir == 12 and d == 0:
36 #         Gir = Gir - 1
37 #     elif rpm < 1094 and Gir == 11 and d == 0:
38 #         Gir = Gir - 1
39 #     elif rpm < 1015 and Gir == 10 and d == 0:
40 #         Gir = Gir - 1
41 #     elif rpm > 1435 and Gir != 12 and d == 0:
42 #         Gir = Gir + 1
43 #     return Gir
44
45 # --- Scania -----
46
47 def get_rpm_eff(Gir):
48     if Gir == 12:
49         rpm_eff = 14.1
50     elif Gir == 11:
51         rpm_eff = 17.8
52     elif Gir == 10:
53         rpm_eff = 21.2
54     elif Gir == 9:
55         rpm_eff = 26.6
56     elif Gir == 8:
57         rpm_eff = 35.0
58     elif Gir == 7:
59         rpm_eff = 44
60     elif Gir == 6:
61         rpm_eff = 53
62     elif Gir == 5:
63         rpm_eff = 67.5
64     elif Gir == 4:
65         rpm_eff = 82.5
66     return rpm_eff
67
```



## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
68 # R580
69 def get_P_w(rpm):
70     if 800 <= rpm < 950:
71         P_w = ((rpm - 800) * 0.441 * 1000) + 235.3 * 1000
72     elif 950 <= rpm <= 1300:
73         P_w = ((rpm - 950) * 0.315 * 1000) + 301.5 * 1000
74     elif 1300 < rpm < 1900:
75         P_w = ((rpm - 1300) * 0.0245 * 1000) + 411.8 * 1000
76     return P_w
77
78 # S650
79 #def get_P_w(rpm):
80 #     if 800 <= rpm < 950:
81 #         P_w = ((rpm - 800) * 0.441 * 1000) + 235.3 * 1000
82 #     elif 950 <= rpm <= 1350:
83 #         P_w = ((rpm - 950) * 0.373 * 1000) + 332.5 * 1000
84 #     elif 1350 < rpm < 1900:
85 #         P_w = ((rpm - 1350) * 0.023 * 1000) + 472 * 1000
86 #     return P_w
87
88 # S730
89 #def get_P_w(rpm):
90 #     if 800 <= rpm < 1000:
91 #         P_w = ((rpm - 800) * 0.441 * 1000) + 235.3 * 1000
92 #     elif 1000 <= rpm <= 1400:
93 #         P_w = ((rpm - 1000) * 0.373 * 1000) + 372.8 * 1000
94 #     elif 1400 < rpm < 1900:
95 #         P_w = ((rpm - 1400) * 0.045 * 1000) + 522 * 1000
96 #     return P_w
97
98 # -----
99
100 ll = CSV_File_Reader.Acchor10
101 ss = CSV_File_Reader.stigning10
102
103 # ll = NVDB_posisjon.Acchor
104 # ss = NVDB_posisjon.S
105
106 speed_VBOX = CSV_File_Reader.speed_VBOX
107
108 # -----
109
110 # --- Basisparametere -----
111 startgir = 12
112 # --- Vekt ---
113 m = 42500
114 # --- Rullemotstand Alt. 1 ---
115 k_r = 0.001
116 k_s = 1 / 4470
117 # --- Rullemotstand Alt. 2 ---
118 iri = 1.2
119 mpd = 1.3
120 # --- Luftmotstand ---
121 A = 8
122 cw = 0.6
123 rho = 1.2
124 # --- Vindhastighet ---
125 v0 = 0
126 # --- Tyngdekraft ---
127 g = 10
128 # --- Grenseverdier ---
129 v_max = 90
130 a_max = 3.0
131 # --- Starthastighet ---
132 v_start = speed_VBOX[0]
133 # v_start = 60
134 # --- Startpunkt ---
```

## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
135 x1 = 0
136 h1 = 0
137 # --- Tidsteg ---
138 deltaT = 0.01
139 # --- Effekt gammel modell [kW]---
140 eff_gammel = 425
141
142 # Oppsett = [1,1,1,1,1]
143 # Oppsett = [0.7,1,1.138,1.326, 1]
144 Oppsett = [0.96, 1, 1, 1, 1]
145
146 # --- Lister ---
147
148 T = [[] for _ in range(len(Oppsett))]
149 RPM = [[] for _ in range(len(Oppsett))]
150 GIR = [[] for _ in range(len(Oppsett))]
151 X = [[] for _ in range(len(Oppsett))]
152 H = [[] for _ in range(len(Oppsett))]
153 V1 = [[] for _ in range(len(Oppsett))]
154 F_R0 = [[] for _ in range(len(Oppsett))]
155 F_L = [[] for _ in range(len(Oppsett))]
156 F_CL = [[] for _ in range(len(Oppsett))]
157 P_W = [[] for _ in range(len(Oppsett))]
158
159 for k in range(len(Oppsett)):
160     Gir = startgir
161     GIR[k].append(startgir)
162     t = 0
163     stegfaktor = 0.1 / deltaT
164     v1 = v_start
165     stepcounter = 0
166     c = 0
167     u = 0
168     i = 0
169     p = 0
170     d = 0
171     x = x1
172     h = h1
173
174     while x < ll[len(ll) - 1]:
175         s = ss[c]
176         if x > ll[c]:
177             c = c + 1
178             s = ss[c]
179
180         stepcounter = stepcounter + 1
181
182         v1 = v1 / 3.6
183
184         if u == 10 * stegfaktor:
185             u = 0
186
187         if i == 100 * stegfaktor:
188             i = 0
189
190         if d == 100 * stegfaktor:
191             d = 0
192
193         if i != 0:
194             i = i + 1
195
196         if u != 0:
197             u = u + 1
198
199         if d != 0:
200             d = d + 1
201     else:
```

## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
202     rpm = get_rpm_eff(Gir) * v1 * 3.6
203
204     # Gir = get_Gir(Gir)
205
206     Gir = get_Gir_standard(Gir)
207
208     rpm = get_rpm_eff(Gir) * v1 * 3.6
209     P_w = get_P_w(rpm)
210     GIR[k].append(Gir)
211
212     if Gir != GIR[k][stepcounter - 1]:
213         u = u + 1
214         i = i + 1
215         d = d + 1
216
217     # --- Girskift tidsbruk ---
218     if 0 < u <= 10 * stegfaktor:
219         P_w = P_w * 0
220     # --- Redusert effekt etter giring ---
221     if 10 * stegfaktor < i <= 100 * stegfaktor:
222         P_w = P_w * 1
223
224     # if d == 100 * stegfaktor and s < 10:
225     #     d = 0
226
227     # --- Varierende iri og mpd ---
228     # if IRI_MPD.P[p] < x < IRI_MPD.P[len(IRI_MPD.P)-1]:
229     #     p = p + 1
230     #
231     # if p > 71:
232     #     iri = 1.3
233     #     mpd = 1.22
234     #
235     # iri = IRI_MPD.IRI_list[p]
236     # mpd = IRI_MPD.MPD_list[p]
237
238     # --- Motstandskrefter ---
239     f_r = 0.00414 + 0.0000158 * iri * v1 + 0.00102 * mpd
240     # f_r = (k_r + k_s * v1)
241     # f_r = 0.015
242     F_ro = f_r * m * g * 1
243     F_l = 0.5 * rho * A * cw * (v1 + v0) ** 2
244     # F_l = 0.5 * rho * 7.13 * (v1 + v0) ** 2 # Oppgitt av Mercedes at A*cw=7.13
245     F_cl = m * g * 0.01 * s
246
247     F_R0[k].append(F_ro)
248     F_L[k].append(F_l)
249     F_CL[k].append(F_cl)
250
251     # if Gir == 10:
252     #     P_w = P_w*1
253
254     P_w = P_w * Oppsett[k]
255
256     P_w = P_w * 1
257
258     # --- Konstant effekt ---
259     if k == 2:
260         P_w = eff_gammel * 1000 * 0.839
261     if k == 3:
262         P_w = eff_gammel * 1000 * 0.95
263     if k == 4:
264         P_w = eff_gammel * 1000 * 1
265     # -----
266
267     P_W[k].append(P_w)
268
```

## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
269     # --- Sum av krefter ---
270     F_wn = F_ro + F_l + F_cl
271     F_wp = P_w / v1
272     F_ws = F_wp - F_wn
273
274     a = F_ws / m
275
276     t = t + deltaT
277     deltaX = v1 * deltaT + 0.5 * a * deltaT * deltaT
278
279     x = x + deltaX
280     deltaH = (deltaX * s) / 100
281
282     h = h + deltaH
283     v1 = v1 + a * deltaT
284
285     T[k].append(t)
286     X[k].append(x)
287     H[k].append(h)
288
289     RPM[k].append(rpm)
290
291     if v1 >= v_max / 3.6:
292         v1 = v_max / 3.6
293         print('Maks hastighet nådd')
294     else:
295         v1 = v1
296
297     v1 = v1 * 3.6
298     V1[k].append(v1)
299
300 # --- Printing -----
301 print(stepcounter)
302 print(iri)
303 print(mpd)
304 print(p)
305 print('U =', u)
306 print(i)
307 print('Gir', Gir)
308 print('kW', P_w / 1000)
309 print('rpm', rpm)
310 print('Luftmotstand', F_l)
311 print('Stigningsmotstand', F_cl)
312 print('Rullemotstand', F_ro)
313 print('Negative krefter sum', F_wn)
314 print('Positive krefter sum', F_wp)
315 print('Sum krefter', F_ws)
316 print('Tid totalt:', t)
317 print('Akselesasjon:', a, 'm/s^2')
318 print('Distanse:', x, 'm')
319 print('Høyde økning:', deltaH, 'm')
320 print('Høyde:', h, 'moh')
321 print('Hastighet:', v1, 'km/t')
322
323 # --- Utrekning av motstandsgraf -----
324
325 # Motstand_1 = F_CL
326 # Motstand_2 = [x + y for x, y in zip(F_CL, F_RO)]
327 # Motstand_3 = [x + y + z for x, y, z in zip(F_CL, F_RO, F_L)]
328 #
329 #Sum_motstand = [x + y + z for x, y, z in zip(F_CL, F_RO, F_L)]
330 #F_CL_DEL = [x/y for x, y in zip(F_CL, Sum_motstand)]
331 #F_RO_DEL = [x/y for x, y in zip(F_RO, Sum_motstand)]
332 #F_L_DEL = [x/y for x, y in zip(F_L, Sum_motstand)]
333 #
334 #
335 #
```

## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
336 #Motstand_4 = [x * 100 for x in F_CL_DEL]
337 #Motstand_5 = [100 * (x + y) for x, y in zip(F_CL_DEL, F_RO_DEL)]
338 #Motstand_6 = [100* (x + y + z) for x, y, z in zip(F_CL_DEL, F_RO_DEL, F_L_DEL)]
339
340
341 Lengste = max(enumerate(V1), key=lambda tup: len(tup[1]))
342 Lengste_liste = Lengste[0]
343
344 mod_VBOX = len(speed_VBOX) / len(X[Lengste_liste])
345 VS = []
346 V1_eq = [[] for _ in range(len(Oppsett))]
347 X_eq = [[] for _ in range(len(Oppsett))]
348
349 for hastighet in range(len(X[Lengste_liste])):
350     vs = speed_VBOX[math.floor(hastighet * mod_VBOX)]
351     VS.append(vs)
352
353 for b in range(len(Oppsett)):
354     if len(V1[b]) < len(VS):
355         forhoId = len(V1[b]) / len(VS)
356         for r in range(len(VS)):
357             speed = V1[b][math.floor(r * forhoId)]
358             V1_eq[b].append(speed)
359             distance = X[b][math.floor(r * forhoId)]
360             X_eq[b].append(distance)
361     else:
362         forhoId = len(VS) / len(V1[b])
363         for r in range(len(VS)):
364             speed = V1[b][math.floor(r * forhoId)]
365             V1_eq[b].append(speed)
366             distance = X[b][math.floor(r * forhoId)]
367             X_eq[b].append(distance)
368
369 # --- Plotting -----
370 # -----
371
372 #
373 fig, ax1 = plt.subplots()
374
375 color = 'tab:red'
376 ax1.set_xlabel('Horisontal avstand [m]')
377 ax1.set_ylabel('Hastighet modell [km/t]', color=color)
378 ax1.plot(X[0], V1[0], color=color)
379 ax1.tick_params(axis='y', labelcolor=color)
380
381 ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis
382 ax1.set_ylim(min(V1[0] + VS) - 1, max(V1[0] + VS) + 1)
383 ax2.set_ylim(min(V1[0] + VS) - 1, max(V1[0] + VS) + 1)
384 color = 'tab:blue'
385 ax2.set_ylabel('Hastighet VBOX [km/t]', color=color) # we already handled the x-label
    with ax1
386 ax2.plot(X[0], VS, color=color)
387 ax2.tick_params(axis='y', labelcolor=color)
388
389 fig.tight_layout() # otherwise the right y-label is slightly clipped
390
391 plt.grid()
392 plt.show()
393
394 # fig, ax1 = plt.subplots()
395 #
396 # color = 'tab:red'
397 # ax1.set_xlabel('Horisontal avstand [m]')
398 # ax1.set_ylabel('Hastighet modell [km/t]', color=color)
399 # ax1.plot(X[0], V1[0], color=color)
400 # ax1.tick_params(axis='y', labelcolor=color)
401 #
```

## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
402 # ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis
403 #
404 # color = 'tab:blue'
405 # ax2.set_ylabel('Hastighet VBOX', color=color) # we already handled the x-label with
    ax1
406 # ax2.plot(X[0], VS, color=color)
407 # ax2.tick_params(axis='y', labelcolor=color)
408 #
409 # fig.tight_layout() # otherwise the right y-label is slightly clipped
410 # plt.show()
411
412 fig, ax1 = plt.subplots()
413
414 color = 'tab:red'
415 ax1.set_xlabel('Tid [s]')
416 ax1.set_ylabel('Høyde [moh]', color=color)
417 ax1.plot(T[0], H[0], color=color)
418 ax1.tick_params(axis='y', labelcolor=color)
419
420 ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis
421
422 color = 'tab:blue'
423 ax2.set_ylabel('Hastighet', color=color) # we already handled the x-label with ax1
424 ax2.plot(T[0], V1[0], color=color)
425 ax2.tick_params(axis='y', labelcolor=color)
426
427 fig.tight_layout() # otherwise the right y-label is slightly clipped
428 plt.grid()
429 plt.show()
430
431 # fig, ax1 = plt.subplots(figsize=(10,4))
432 # alpha = 1
433 # linewidth = 2
434 # ax1.set_xlabel('Horisontal avstand [m]')
435 # ax1.set_ylabel('GIR')
436 # ax1.plot(X[0], GIR[0][:len(X[0])], alpha=alpha, linewidth=linewidth)
437 # ax1.plot(X[1], GIR[1][:len(X[1])], alpha=alpha, linewidth=linewidth)
438 # ax1.plot(X[2], GIR[2][:len(X[2])], alpha=alpha, linewidth=linewidth)
439 # ax1.plot(X[3], GIR[3][:len(X[3])], alpha=alpha, linewidth=linewidth)
440 # ax1.tick_params(axis='y')
441 #
442 # fig.tight_layout() # otherwise the right y-label is slightly clipped
443 # plt.show()
444 #
445 # fig, ax1 = plt.subplots(figsize=(10,4))
446 # alpha = 1
447 # linewidth = 2
448 # ax1.set_xlabel('Horisontal avstand [m]')
449 # ax1.set_ylabel('RPM')
450 # ax1.plot(X[0], RPM[0], alpha=alpha, linewidth=linewidth)
451 # ax1.plot(X[1], RPM[1], alpha=alpha, linewidth=linewidth)
452 # ax1.plot(X[2], RPM[2], alpha=alpha, linewidth=linewidth)
453 # ax1.plot(X[3], RPM[3], alpha=alpha, linewidth=linewidth)
454 # ax1.tick_params(axis='y')
455 #
456 # fig.tight_layout() # otherwise the right y-label is slightly clipped
457 # plt.show()
458
459 # --- Girskift og turtall -----
460
461 fig, ax1 = plt.subplots(figsize=(10, 4))
462 alpha = 1
463 linewidth = 2
464 ax1.set_xlabel('Horisontal avstand [m]')
465 ax1.set_ylabel('GIR')
466 ax1.plot(X[0], GIR[0][:len(X[0])], alpha=alpha, linewidth=linewidth)
467 ax1.tick_params(axis='y')
```

## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
468
469 fig.tight_layout() # otherwise the right y-label is slightly clipped
470 plt.show()
471
472 fig, ax1 = plt.subplots(figsize=(10, 4))
473 alpha = 1
474 linewidth = 2
475 ax1.set_xlabel('Horisontal avstand [m]')
476 ax1.set_ylabel('RPM')
477 ax1.plot(X[0], RPM[0], alpha=alpha, linewidth=linewidth)
478 ax1.tick_params(axis='y')
479
480 fig.tight_layout() # otherwise the right y-label is slightly clipped
481 plt.show()
482
483 # --- Forskjellig motoreffekt -----
484
485 # fig, ax1 = plt.subplots()
486 #
487 # color = 'tab:pink'
488 # ax1.set_xlabel('Tid [s]')
489 # ax1.set_ylabel('Høyde [moh]', color=color)
490 # ax1.plot(T[0], H[0], color=color)
491 # ax1.tick_params(axis='y', labelcolor=color)
492 #
493 # ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis
494 #
495 # color = 'tab:blue'
496 # ax2.set_ylabel('RPM', color=color) # we already handled the x-label with ax1
497 # ax2.plot(T[0], RPM[0], color=color)
498 # ax2.tick_params(axis='y', labelcolor=color)
499 #
500 # fig.tight_layout() # otherwise the right y-label is slightly clipped
501 # plt.show()
502
503 # Hastighet, Stigning mot Avstand
504
505 # fig, ax1 = plt.subplots(figsize=(10,10))
506 # ax1.set_xlabel('Horisontal avstand [m]')
507 # ax1.set_ylabel('Hastighet [km/t]') # we already handled the x-label with ax1
508 # ax1.plot(X[0], V1[0], label='530')
509 # ax1.plot(X[1], V1[1], label='580')
510 # ax1.plot(X[2], V1[2], label='660')
511 # ax1.plot(X[3], V1[3], label='770')
512 # ax1.plot(X[4], V1[4], label='580 gammel')
513 # ax1.plot(X[4], VS, label='VB0X')
514 #
515 # ax1.set_ylim([27.5, 95])
516 # ax1.minorticks_on()
517 # ax1.tick_params(axis='y')
518 # plt.legend(loc=6, prop={'size': 10})
519 # ax1.set_facecolor('#bfbfbf')
520 #
521 # color = 'blue'
522 # ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis
523 #
524 # ax2.set_ylabel('Høyde [m]', color=color)
525 # ax2.plot(X[0], H[0], color =color)
526 # ax2.tick_params(axis='y', colors=color)
527 # ax2.set_ylim([-5, 130])
528 # ax2.minorticks_on()
529 # ax2.grid(False)
530 #
531 # plt.savefig(save_results_to + 'Hastighet_ulik_effekt.pdf', bbox_inches='tight')
532 # fig.tight_layout() # otherwise the right y-label is slightly clipped
533 # plt.show()
534
```

## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
535 # --- Sammenligning av ny og gammel graf -----
536
537 # fig, ax1 = plt.subplots(figsize=(10, 5))
538 # ax1.set_xlabel('Horisontal avstand [m]')
539 # ax1.set_ylabel('Hastighet [km/t]') # we already handled the x-label with ax1
540
541 # plt.axvline(220, color='grey')
542 # plt.axvline(680, color='grey')
543 # plt.axvline(1120, color='grey')
544 # plt.axvline(2130, color='grey')
545 #
546 # plt.axvline(390, color='c')
547 # plt.axvline(810, color='c')
548 # plt.axvline(1190, color='c')
549 # plt.axvline(1550, color='c')
550 # plt.axvline(2120, color='c')
551 #
552 # plt.text(220, 86, '12 > 11', bbox={'facecolor':'white', 'alpha':1, 'edgecolor':'grey', '
pad':2}, ha='left', va='bottom', rotation=45, size=9)
553 # plt.text(680, 86, '11 > 10', bbox={'facecolor':'white', 'alpha':1, 'edgecolor':'grey', '
pad':2}, ha='left', va='bottom', rotation=45, size=9)
554 # plt.text(1100, 86, '10 > 9', bbox={'facecolor':'white', 'alpha':1, 'edgecolor':'grey', '
pad':2}, ha='left', va='bottom', rotation=45, size=9)
555 # plt.text(2150, 86, '10 > 11', bbox={'facecolor':'white', 'alpha':1, 'edgecolor':'grey', '
pad':2}, ha='left', va='bottom', rotation=45, size=9)
556 #
557 # plt.text(390, 86, '12 > 11', bbox={'facecolor':'white', 'alpha':1, 'edgecolor':'c', 'pad
':2}, ha='left', va='bottom', rotation=45, size=9)
558 # plt.text(810, 86, '11 > 10', bbox={'facecolor':'white', 'alpha':1, 'edgecolor':'c', 'pad
':2}, ha='left', va='bottom', rotation=45, size=9)
559 # plt.text(1190, 86, '10 > 9', bbox={'facecolor':'white', 'alpha':1, 'edgecolor':'c', 'pad
':2}, ha='left', va='bottom', rotation=45, size=9)
560 # plt.text(1550, 86, '9 > 10', bbox={'facecolor':'white', 'alpha':1, 'edgecolor':'c', 'pad
':2}, ha='left', va='bottom', rotation=45, size=9)
561 # plt.text(2050, 86, '9 > 10', bbox={'facecolor':'white', 'alpha':1, 'edgecolor':'c', 'pad
':2}, ha='left', va='bottom', rotation=45, size=9)
562
563 # Forhold = len(V1[4])/len(V1[3])
564 # V3MOD = []
565 #
566 # for hastighet in range(len(V1[4])):
567 #     v3mod = V1[3][math.floor(hastighet*Forhold)]
568 #     V3MOD.append(v3mod)
569
570 # ax1.plot(X_eq[0], VS, label='VB0X', color='g')
571 # ax1.plot(X_eq[1], V1_eq[1], label='R580 standard giring', color='r')
572 # ax1.plot(X_eq[0], V1_eq[0], label='R580 tilpasset giring', color='orange', linewidth=2
)
573 # ax1.plot(X[2], V1[2], label='R580 gammel modell justert', color='orange', linewidth=2)
574
575 # ax1.plot(X_eq[3], V1_eq[3], '--', label='R580 gammel modell 95 %', color='c')
576 # ax1.plot(X[4], V1[4], '--', label='R580 gammel modell 85 %', color='orange')
577 # ax1.plot(X[4], V3MOD, '--', label='R580 gammel modell 80 %', color='orange')
578
579
580 # plt.legend(loc=9, prop={'size': 10})
581
582 # ax1.fill_between(X[4], V3MOD, V1[4], facecolor='orange', alpha=0.15, interpolate=True)
583
584 # ax1.set_ylim([27.5, 95])
585 # ax1.tick_params(axis='y')
586 # plt.xticks(np.arange(0, max(X[0]) + 100, 200))
587 # plt.legend(loc=9, prop={'size': 10})
588
589 # plt.axvline(290, color='c')
590 #
591 # plt.text(290, 86, '11 > 9', bbox={'facecolor':'white', 'alpha':1, 'edgecolor':'c', 'pad':
```



## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
591 2}, ha='left', va='bottom', rotation=45, size=9)
592
593 # color = 'blue'
594 # ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis
595
596 # ax2.set_ylabel('Vertikal avstand [m]', color=color)
597 # ax2.plot(X[0], H[0], color =color)
598 # ax2.tick_params(axis='y', colors=color)
599 # ax2.set_ylim([-5, 130])
600 # ax2.grid(False)
601
602 # plt.savefig(save_results_to + 'Plott_forside.png', bbox_inches='tight')
603 # fig.tight_layout() # otherwise the right y-label is slightly clipped
604 # plt.show()
605
606 # --- Forskjellig motoreffekt -----
607
608 fig, ax1 = plt.subplots(figsize=(10,4))
609 ax1.set_xlabel('Horisontal avstand [m]')
610 ax1.set_ylabel('Hastighet [km/t]') # we already handled the x-label with ax1
611 ax1.plot(X[0], V1[0], label='Scania R580')
612 ax1.plot(X[0], VS, label='VB0X')
613
614 ax1.set_ylim([27.5, 95])
615 ax1.minorticks_on()
616 ax1.tick_params(axis='y')
617 plt.legend(loc=6, prop={'size': 10})
618 ax1.set_facecolor('#bfbfbf')
619
620 color = 'blue'
621 ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis
622
623 ax2.set_ylabel('Høyde [m]', color=color)
624 ax2.plot(X[0], H[0], color =color)
625 ax2.tick_params(axis='y', colors=color)
626 ax2.set_ylim([-5, 130])
627 ax2.minorticks_on()
628 ax2.grid(False)
629
630 # plt.savefig(save_results_to + 'Modell_vs_VB0X.pdf', bbox_inches='tight')
631 fig.tight_layout() # otherwise the right y-label is slightly clipped
632 plt.show()
633
634 # --- Absolutt motstandsgraf -----
635
636 # fig, ax = plt.subplots()
637 #
638 # ax.plot(X, Motstand_1, X, Motstand_2, X, Motstand_3, color='black', linewidth=0.5)
639 # ax.set_xlabel('Horisontal avstand [m]')
640 # ax.fill_between(X, 0, Motstand_1, facecolor='#24a65e', interpolate=True)
641 # ax.fill_between(X, Motstand_1, Motstand_2, facecolor='#246ba6', interpolate=True)
642 # ax.fill_between(X, Motstand_2, Motstand_3, facecolor='#24a6a0', interpolate=True)
643 # ax.set_title('Motstandskrefter')
644 #
645 # plt.savefig(save_results_to + 'Motstandskrefter.pdf', bbox_inches='tight')
646 # ax.grid(axis='y')
647 # plt.show()
648 #
649 # fig10, ax = plt.subplots()
650 #
651 # ax.plot(X, F_CL_DEL, X, F_RO_DEL, X, F_L_DEL, color='black')
652 # ax.fill_between(X, 0, F_CL_DEL, facecolor='#24a65e', interpolate=True)
653 # ax.fill_between(X, F_CL_DEL, F_RO_DEL, facecolor='#246ba6', interpolate=True)
654 # ax.fill_between(X, F_RO_DEL, F_L_DEL, facecolor='#24a6a0', interpolate=True)
655 # ax.set_title('fill between where')
656
657 # plt.show()
```

## VEDLEGG C. HASTIGHETSMODELL

---

```
File C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
658
659 # --- Motstandsgraf -----
660 # --- Laget uten seaborn -----
661
662 # fig, ax = plt.subplots()
663 #
664 #
665 # twin1 = ax.twinx()
666 # twin2 = ax.twinx()
667 #
668 # fig.subplots_adjust(right=0.75)
669 #
670 # twin2.spines["right"].set_position(("axes", 1.2))
671 # twin2.spines.right.set_position(("axes", 1.2))
672 #
673 # ax.set_xlim(0, X[len(X)-1])
674 # ax.set_ylim(0, 100)
675 # twin1.set_ylim(min(V1[0])-5, max(V1[0])+5)
676 # twin2.set_ylim(min(H)-5, max(H)+5)
677 #
678 # ax.plot(X, Motstand_4, color='black', linewidth=0.5)
679 # ax.plot(X, Motstand_5, color='black', linewidth=0.5)
680 # ax.plot(X, Motstand_6, color='black', linewidth=0.5)
681 # # ax.plot(X, Motstand_5, color='black', linewidth=0.5)
682 # # ax.plot(X, Motstand_6, color='black', linewidth=0.5)
683 # twin1.plot(X, V1[0], color="red")
684 # twin2.plot(X, H, color= "blue")
685 #
686 # ax.set_xlabel("Horisontal avstand [m]")
687 # ax.set_ylabel("Andel av total kjøremotstand [%]")
688 # twin1.set_ylabel("Hastighet [km/t]", color="red")
689 # twin2.set_ylabel("Høyde [m]", color="blue")
690 #
691 # ax.fill_between(X, 0, Motstand_4, facecolor='#24a65e', alpha=0.5, interpolate=True)
692 # ax.fill_between(X, Motstand_4, Motstand_5, facecolor='#246ba6', alpha=0.5, interpolate
693 # =True)
694 # ax.fill_between(X, Motstand_5, Motstand_6, facecolor='#24a6a0', alpha=0.5, interpolate
695 # =True)
696 # # ax.set_title('Motstandskrefter andeler')
697 #
698 # # Set axes colors
699 # ax.spines["left"].set_color("black")
700 # twin1.spines["right"].set_color("black")
701 # twin2.spines["right"].set_color("black")
702 #
703 # ax.minorticks_on()
704 # twin1.minorticks_on()
705 # twin2.minorticks_on()
706 # # y_ticks = np.arange(0, 1.0001, 0.1)
707 # # plt.yticks(y_ticks)
708 #
709 # ax.tick_params(direction="in", which="both", colors="black")
710 # twin1.tick_params(direction="in", which="both", colors="red")
711 # twin2.tick_params(direction="in", which="both", colors="blue")
712 #
713 # ax.set_title('Motstandskrefter, 111 kg/kW', fontsize = 12)
714 #
715 # # plt.text(1450, 34, '(a)')
716 # # plt.text(2250, 20, '(b)')
717 # # plt.text(2500, 62, '(c)')
718 # # plt.text(100, 1, '0,0%')
719 # # plt.text(445, 12, '3,5%')
720 # # plt.text(980, 48, '7,0%')
721 # # plt.text(1365, 65, '3,5%')
722 # # plt.text(2100, 71, '0,0%')
723 #
724 # plt.savefig(save_results_to + 'Motstandskrefter_andeler_2.pdf', bbox_inches='tight')
```

## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
723 # # ax.grid(axis='y')
724 # plt.show()
725 # -----
726 fig, ax1=plt.subplots(figsize=(10,5))
727 ax1.set_xlabel('Horisontalavstand[m]')
728 ax1.set_ylabel('Hastighet[km/t]')#wealreadyhandledthex-labelwithax1
729
730 # plt.axvline(110,color='grey')
731 # plt.axvline(400,color='grey')
732 # plt.axvline(810,color='grey')
733 # plt.axvline(1605,color='grey')
734 # plt.axvline(2350,color='grey')
735 # plt.axvline(3280,color='grey')
736
737 # plt.axvline(250,color='c')
738 # plt.axvline(530,color='c')
739 # plt.axvline(640,color='c')
740 # plt.axvline(1460,color='c')
741 # plt.axvline(2550,color='c')
742 # plt.axvline(3040,color='c')
743
744 # plt.text(110,86,'12>11',bbox={'facecolor':'white','alpha':1,'edgecolor':'grey','pad':2
745 },ha='left',va='bottom',rotation=45,size=9)
746 # plt.text(400,86,'11>10',bbox={'facecolor':'white','alpha':1,'edgecolor':'grey','pad':2
747 },ha='left',va='bottom',rotation=45,size=9)
748 # plt.text(810,86,'10>9',bbox={'facecolor':'white','alpha':1,'edgecolor':'grey','pad':2
749 },ha='left',va='bottom',rotation=45,size=9)
750 # plt.text(1605,86,'9 >10',bbox={'facecolor':'white','alpha':1,'edgecolor':'grey','pad':
751 2},ha='left',va='bottom',rotation=45,size=9)
752 # plt.text(2350,86,'10>11',bbox={'facecolor':'white','alpha':1,'edgecolor':'grey','pad':
753 2},ha='left',va='bottom',rotation=45,size=9)
754 # plt.text(3280,86,'11>12',bbox={'facecolor':'white','alpha':1,'edgecolor':'grey','pad':
755 2},ha='left',va='bottom',rotation=45,size=9)
756
757 # plt.text(250,86,'12>11',bbox={'facecolor':'white','alpha':1,'edgecolor':'c','pad':2},
758 ha='left',va='bottom',rotation=45,size=9)
759 # plt.text(530,86,'11>10',bbox={'facecolor':'white','alpha':1,'edgecolor':'c','pad':2},
760 ha='left',va='bottom',rotation=45,size=9)
761 # plt.text(640,86,'10>9',bbox={'facecolor':'white','alpha':1,'edgecolor':'c','pad':2},ha
762 = 'left',va='bottom',rotation=45,size=9)
763 # plt.text(1460,86,'9 >10',bbox={'facecolor':'white','alpha':1,'edgecolor':'c','pad':2},
764 ha='left',va='bottom',rotation=45,size=9)
765 # plt.text(2550,86,'10>11',bbox={'facecolor':'white','alpha':1,'edgecolor':'c','pad':2},
766 ha='left',va='bottom',rotation=45,size=9)
767 # plt.text(3040,86,'11>12',bbox={'facecolor':'white','alpha':1,'edgecolor':'c','pad':2},
768 ha='left',va='bottom',rotation=45,size=9)
769
770 ax1.plot(X[0],V1[0],label='Scania ny modell',color='r')
771 ax1.plot(X[2],V1[2], '--',label='Scania gammel modell, 85%',color='b')
772 ax1.plot(X[3],V1[3], '--',label='Scania gammel modell, 95%',color='orange')
773 ax1.plot(X[0],VS,label='VBOX',color='g')
774 plt.legend(loc=9,prop={'size':10})
775
776 ax1.set_ylim([27.5,95])
777 ax1.tick_params(axis='y')
778 plt.xticks(np.arange(0,max(X[0])+100,200))
779 plt.legend(loc=0,prop={'size':10})
780
781 color='blue'
782 ax2=ax1.twinx()#instantiateasecondaxesthatsharesthesamex-axis
783
784 ax2.set_ylabel('Vertikalavstand[m]',color=color)
785 ax2.plot(X[0],H[0],color=color)
786 ax2.tick_params(axis='y',colors=color)
787 ax2.set_ylim([-5,130])
788 ax2.grid(False)
```

## VEDLEGG C. HASTIGHETSMODELL

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\Modell.py
778
779 plt.savefig(save_results_to+'Ny_og_gammel_modell.pdf',bbox_inches='tight')
780 fig.tight_layout()#otherwisetherighty-labelisslightlyclipped
781 plt.show()
782
783 # -----Diverse print konsoll-----
784
785 end = time.time()
786 print('Tid:', round(T[0][len(T[0]) - 1], 2), 'v_slutt:', round(V1[0][len(V1[0]) - 1], 1
), 'v_max:',
787       round(max(V1[0]), 1), 'v_min:', round(min(V1[0]), 1))
788 print('Tid:', round(T[1][len(T[1]) - 1], 2), 'v_slutt:', round(V1[1][len(V1[1]) - 1], 1
), 'v_max:',
789       round(max(V1[1]), 1), 'v_min:', round(min(V1[1]), 1))
790 print('Tid:', round(T[2][len(T[2])-1],2), 'v_slutt:', round(V1[2][len(V1[2])-1],1), '
v_max:', round(max(V1[2]), 1), 'v_min:', round(min(V1[2]), 1))
791 print('Tid:', round(T[3][len(T[3])-1],2), 'v_slutt:', round(V1[3][len(V1[3])-1],1), '
v_max:', round(max(V1[3]), 1), 'v_min:', round(min(V1[3]), 1))
792 # print('Tid:', round(T[4][len(T[4])-1],2), 'v_slutt:', round(V1[4][len(V1[4])-1],1), '
v_max:', round(max(V1[4]), 1), 'v_min:', round(min(V1[4]), 1))
793
794 average = mean(ss)
795 average_pw1 = mean(P_W[0])
796 average_pw2 = mean(P_W[3])
797 print("Gjennomsnittlig stigning =", round(average, 2), "%")
798 print("Gjennomsnittlig effekt ny modell", round(average_pw1 / 1000, 2))
799 print("Gjennomsnittlig effekt gammel modell", round(average_pw2 / 1000, 2))
800 print("Ny modell prosent av gammel snitt", round(average_pw1 / 10 / 425, 3), "%")
801
802 max_pw1 = max(P_W[0])
803 print("Max effekt ny modell", round(max_pw1 / 1000, 2))
804 print(min(VS))
805 print()
806
807 GEH_list = [[] for _ in range(len(Oppsett))]
808 GEH_snitt = []
809 for w in range(len(Oppsett)):
810     for q in range(len(V1_eq[w])):
811         GEH = (math.sqrt(2 * (V1_eq[w][q] - VS[q]) ** 2) / (V1_eq[w][q] + VS[q]))
812         GEH_list[w].append(GEH)
813     geh_snitt = sum(GEH_list[w]) / len(GEH_list[w])
814     GEH_snitt.append(geh_snitt)
815     print('GEH_snitt', w, '=', GEH_snitt[w])
816
817 # --- Tid koden bruker på å kjøre -----
818 print(end - start)
819
```

**Vedlegg D**

**CSV\_File\_Reader**

## VEDLEGG D. CSV\_FILE\_READER

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProjectModell\CSV_File_Reader.py
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import math
5 import seaborn as sns
6
7 sns.set_theme()
8 save_results_to = '/Users/kbraa/OneDrive/Dokumenter/SkoLe/NTNU/Semester 4 - MASTER/'
9
10 # t = np.arange(0.1,182.4,0.1).tolist()
11 # VBOX_Speed = ['Longitude']
12 # CSS_Speed = ['S1_PID_0D_VehicleSpeed']
13 # data_VBOX = pd.read_csv(r'C:\Users\simen\Desktop\VBOX_062.csv',usecols=VBOX_Speed)
14 # data_CSS = pd.read_csv(r'C:\Users\simen\Desktop\second time change test export.
    ChannelGroup_7_0BD2_0x7E8.csv',usecols=CSS_Speed)
15 #
16 # data_VBOX = pd.read_csv(r'C:\Users\simen\Desktop\15 mars\VBOX_069.csv',usecols=
    VBOX_Speed)
17 # data_CSS = pd.read_csv(r'C:\Users\simen\Desktop\second time change test export.
    ChannelGroup_7_0BD2_0x7E8.csv',usecols=CSS_Speed)
18
19 # list_VBOX = data_VBOX['Longitude'].values.tolist()
20 # list_CSS = data_CSS['S1_PID_0D_VehicleSpeed'].values.tolist()
21 # print(list_VBOX)
22 #
23 # Max = max(list_VBOX + list_CSS)
24 #
25 # diff = []
26 #
27 def round_half_up(n, decimals=0):
28     multiplier = 10 ** decimals
29     return math.floor(n * multiplier + 0.5) / multiplier
30
31
32 #
33 # zip_list = zip(list_VBOX,list_CSS)
34 # for list_VBOX_i, list_CSS_i in zip_list:
35 #     diff.append(round_half_up(abs(list_VBOX_i-list_CSS_i), 2))
36 #
37 #
38 # print(Max)
39 # print(diff)
40 #
41 # fig, ax1 = plt.subplots()
42 #
43 # color = 'tab:red'
44 # ax1.set_xlabel('time (s)')
45 # ax1.set_ylabel('Speed VBOX', color=color)
46 # ax1.plot(t, data_VBOX, color=color)
47 # ax1.tick_params(axis='y', labelcolor=color)
48 #
49 # ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis
50 # ax1.set_ylim(0,Max + 5)
51 # ax2.set_ylim(0,Max + 5)
52 # color = 'tab:blue'
53 # ax2.set_ylabel('Speed CSS', color=color) # we already handled the x-label with ax1
54 # ax2.plot(t, data_CSS, color=color)
55 # ax2.tick_params(axis='y', labelcolor=color)
56 #
57 # fig.tight_layout() # otherwise the right y-label is slightly clipped
58 # plt.show()
59
60 # --- Filnavn -----
61 filename = "\\Scania_06.04_rinnan.csv" #
62 # filename = "\\Scania_06.04_eidsbotn.csv" #
63 # filename = "\\Scania_06.04_gråmyra.csv" #
64 # filename = "\\Scania_06.04_hammervatnet.csv" #
```

## VEDLEGG D. CSV\_FILE\_READER

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\CSV_File_Reader.py
65 # filename = "\\Scania_06.04_åsen.csv" #
66 # filename = "\\Scania_06.04_saltøya_0.csv" #
67 # filename = "\\Scania_06.04_saltøya1.csv" #
68 # filename = "\\Scania_06.04_skatval.csv" #
69 # filename = "\\Scania_06.04_sveberg.csv" #
70 # filename = "\\Scania_06.04_vikhammer.csv" #
71 # filename = "\\Scania_06.04_vikhammer_kortere.csv" #
72 # filename = "\\Scania_06.04_okstad.csv" #
73 # filename = "\\Scania_06.04_fossgrenda.csv" #
74 # filename = "\\Scania_06.04_singsås1.csv" #
75 # filename = "\\Scania_06.04_osan.csv" #
76 # filename = "\\Scania_06.04_almås.csv" #
77 # filename = "\\Scania_06.04_veret.csv" #
78 # filename = "\\Scania_06.04_storlia.csv" #
79 # filename = "\\Scania_06.04_ålen.csv" #
80 # filename = "\\Scania_06.04_ålen1.csv" #
81 # filename = "\\Scania_06.04_ålen1_canedge1_1.csv" #
82 # filename = "\\Scania_06.04_ålen2.csv" #
83 # filename = "\\Scania_06.04_ålen3.csv" #
84 # filename = "\\Scania_06.04_ålen4.csv" #
85 # filename = "\\Scania_06.04_ålen5.csv" #
86 # filename = "\\Scania_06.04_grubba.csv" #
87 # filename = "\\Scania_06.04_orvos.csv" #
88 # filename = "\\Scania_06.04_jamtmoen.csv" #
89
90 # filename = "\\Scania_02.03_okstad.csv" # 12 gir
91 # filename = "\\Scania_02.03_flå.csv" # 11 gir
92 # filename = "\\Scania_02.03_fossgrenda.csv" # 12 gir
93 # filename = "\\Scania_02.03_bones.csv" # 11 gir
94 # filename = "\\Scania_02.03_singsås.csv" # 12 gir
95 # filename = "\\Scania_02.03_osan.csv" # 11 gir
96 # filename = "\\Scania_02.03_veret_1.csv" # 11 gir
97 # filename = "\\Scania_02.03_veret.csv" # 11 gir, bruk NVDB_posisjon!
98 # filename = "\\Scania_02.03_ålen_1_v2.csv" # 11 gir
99 # filename = "\\Scania_02.03_ålen_2_v2.csv" # 11 gir
100 # filename = "\\Scania_02.03_ålen_hele_v2.csv" # 9/10 gir
101 # filename = "\\Scania_02.03_ålen_3_v2.csv" # 11 gir
102 # filename = "\\Scania_02.03_ålen_4_v2.csv" # 11 gir
103 # filename = "\\Scania_02.03_grubba.csv" # 11 gir
104 # filename = "\\Scania_02.03_orvos.csv" # 12 gir
105 # filename = "\\Scania_02.03_jamtmoen.csv" # 12 gir
106 # filename = "\\Scania_02.03_jamtmoen_del2.csv" # 12 gir
107 # filename = "\\Scania_02.03_campingen.csv" # 12 gir
108
109 # filename = "\\Scania_23.03_okstad.csv" # 12 gir
110 # filename = "\\Scania_23.03_singsås.csv" # 11 gir
111 # filename = "\\Scania_23.03_osan.csv" # 11 gir
112 # filename = "\\Scania_23.03_veret.csv" # 12 gir
113 # filename = "\\Scania_23.03_ålen.csv" # 12 gir
114 # filename = "\\Scania_23.03_ålen1.csv" # 12 gir
115 # filename = "\\Scania_23.03_ålen2.csv" # 12 gir
116 # filename = "\\Scania_23.03_ålen3.csv" # 12 gir
117 # filename = "\\Scania_23.03_ålen4.csv" # 12 gir
118 # filename = "\\Scania_23.03_grubba.csv" # 11 gir
119 # filename = "\\Scania_23.03_jamtmoen.csv" # 12 gir
120 # filename = "\\Scania_23.03_campingen.csv" # 12 gir
121
122 # filename = "\\Scania_31.01_vuddu.csv"
123 # filename = "\\Scania_31.01_saltøya.csv"
124 # filename = "\\Scania_31.01_skatval.csv"
125 # filename = "\\Scania_31.01_sveberg.csv"
126 # filename = "\\Scania_31.01_målvik.csv"
127
128 # filename = "\\Tillerbakken.csv"
129 # filename = "\\Scania_Okstadbakken_100.csv"
130 # filename = "\\VBOX_scania_test2.csv"
131 # filename = "\\VBOX_tillerbakken_test.csv"
```

## VEDLEGG D. CSV\_FILE\_READER

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\CSV_File_Reader.py
132 # filename = "\Hallanbakken_2.csv"
133 # filename = "\Volvo_0kstadbakken_2.csv"
134 # filename = "\Sveberg_Scania_100.csv"
135 # filename = "\0kstadbakken_Mercedes_Arocs_Full.csv"
136 # filename = "\Moholtbakken_Mercedes_full.csv"
137 # filename = "\Arocs_Klett_til_Moholt.csv"
138 # filename = "\Klettbakken_Arocs.csv"
139 # filename = "\Bakkeprofil_Tiller_2.csv"
140
141 BASE_PATH = r"C:\Users\kbraa\OneDrive\Dokumenter\SkoLe\NTNU\Semester 4 - MASTER\CSV"
142 READ_FILE = BASE_PATH + filename
143 #READ_FILE_1 = BASE_PATH + filenam
144
145 VBOX_Speed = ['Velocity']
146 data_VBOX = pd.read_csv(READ_FILE, usecols=VBOX_Speed, index_col=False)
147 speed_VBOX = data_VBOX['Velocity'].values.tolist()
148
149 VBOX_distance = ['Distance']
150 VBOX_height = ['Height']
151 distance_VBOX = pd.read_csv(READ_FILE, usecols=VBOX_distance, index_col=False)
152 height_VBOX = pd.read_csv(READ_FILE, usecols=VBOX_height, index_col=False)
153
154 fig, ax1 = plt.subplots()
155
156 color = 'tab:red'
157 ax1.set_xlabel('Distance [m]')
158 ax1.set_ylabel('Height [m]', color=color)
159 ax1.plot(distance_VBOX, height_VBOX, color=color)
160 ax1.tick_params(axis='y', labelcolor=color)
161
162 fig.tight_layout() # otherwise the right y-label is slightly clipped
163 plt.show()
164
165 list_distance = distance_VBOX['Distance'].values.tolist()
166 list_distance = [x - list_distance[0] for x in list_distance]
167 list_height = height_VBOX['Height'].values.tolist()
168
169 # print(list_height)
170
171 acchor = 0
172 deltah = []
173 deltax = []
174 stigning = []
175 Acchor = []
176 i = 0
177
178 for height in range(len(list_height) - 1):
179     dh = list_height[i + 1] - list_height[i]
180     deltah.append(dh)
181     dx = list_distance[i + 1] - list_distance[i]
182     deltax.append(dx)
183     acchor = dx + acchor
184     Acchor.append(acchor)
185     if dx == 0:
186         s = 0
187     else:
188         s = dh / dx * 100
189     if abs(s) > 15:
190         s = 0
191
192     stigning.append(s)
193     i = i + 1
194
195 # Hastighet VBOX -----
196 fig, ax1 = plt.subplots()
197
198 color = 'tab:red'
```



## VEDLEGG D. CSV\_FILE\_READER

---

```
File C:\Users\kbraa\PycharmProjects\pythonProject\Modell\CSV_File_Reader.py
199 ax1.set_xlabel('Horisontal avstand [m]')
200 ax1.set_ylabel('Hastighet [km/t]', color=color)
201 ax1.plot(Acchor, speed_VBOX[1:], color=color)
202 ax1.tick_params(axis='y', labelcolor=color)
203
204 fig.tight_layout() # otherwise the right y-label is slightly clipped
205 plt.show()
206
207 # Stigning -----
208 fig, ax1 = plt.subplots()
209
210 ax1.set_xlabel('Horisontal avstand [m]')
211 ax1.set_ylabel('Stigning [%]')
212 ax1.plot(Acchor, stigning, 'b')
213 ax1.tick_params(axis='y')
214 ax1.set_title('VBOX stigning alle loggepunkter')
215
216 fig.tight_layout() # otherwise the right y-label is slightly clipped
217 # plt.savefig(save_results_to + 'VBOX_stigning_alle_loggepunkter.pdf')
218 plt.show()
219
220 # Gjennomsnitt stigning 10 -----
221 stigning10array = np.mean(np.array(stigning).reshape(-1, 10), axis=1)
222 stigning10 = stigning10array.tolist()
223 Acchor10 = Acchor[9::10]
224
225 fig, ax1 = plt.subplots()
226
227 ax1.set_xlabel('Distance [m]')
228 ax1.set_ylabel(' Stigning [%]')
229 ax1.plot(Acchor10, stigning10, 'b')
230 ax1.tick_params(axis='y')
231 #ax1.set_title('VBOX stigning gjennomsnitt 10')
232 fig.tight_layout() # otherwise the right y-label is slightly clipped
233 plt.savefig(save_results_to + 'Stigning_10_snitt.pdf')
234 plt.show()
235
236 # stigning20array = np.mean(np.array(stigning).reshape(-1,20), axis=1)
237 # stigning20 = stigning20array.tolist()
238 # Acchor20 = Acchor[19::20]
239 #
240 # fig, ax1 = plt.subplots()
241 # ax1.set_xlabel('Distance [m]')
242 # ax1.set_ylabel('Stigning [%]')
243 # ax1.plot(Acchor20, stigning20, 'g')
244 # ax1.tick_params(axis='y')
245 # fig.tight_layout() # otherwise the right y-label is slightly clipped
246 # plt.savefig(save_results_to + 'Stigning_20_snitt.pdf')
247 # plt.show()
248
249 # stigning50array = np.mean(np.array(stigning).reshape(-1,50), axis=1)
250 # stigning50 = stigning50array.tolist()
251 # Acchor50 = Acchor[49::50]
252 #
253 # fig, ax1 = plt.subplots()
254 # ax1.set_xlabel('Distance [m]')
255 # ax1.set_ylabel('Stigning [%]')
256 # ax1.plot(Acchor50, stigning50, 'g')
257 # ax1.tick_params(axis='y')
258 # fig.tight_layout() # otherwise the right y-label is slightly clipped
259 # plt.savefig(save_results_to + 'Stigning_50_snitt.pdf')
260 # plt.show()
261 # #
262 # stigning100array = np.mean(np.array(stigning).reshape(-1,100), axis=1)
263 # stigning100 = stigning100array.tolist()
264 # Acchor100 = Acchor[99::100]
265 #
```

## VEDLEGG D. CSV\_FILE\_READER

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\CSV_File_Reader.py
266 # fig, ax1 = plt.subplots()
267 #
268 # color = 'tab:blue'
269 # ax1.set_xlabel('Distance [m]')
270 # ax1.set_ylabel('Stigning100 dh/dx', color=color)
271 # ax1.plot(Acchor100, stigning100, color=color)
272 # ax1.tick_params(axis='y', labelcolor=color)
273 #
274 # fig.tight_layout() # otherwise the right y-label is slightly clipped
275 # plt.show()
276 #
277 # print(Acchor[math.floor(10.4)])
278
```

## **Vedlegg E**

### **NVDB\_posisjon**

## VEDLEGG E. NVDB\_POSISJON

---

```
File - C:\Users\kbraa\PycharmProjects\pythonProject\Modell\NVDB_posisjon.py
1 import pandas as pd
2 import math
3 import matplotlib.pyplot as plt
4
5 data = pd.read_csv(r'C:\Users\kbraa\OneDrive\Skribebord\Test\veret.txt', header=None)
6
7 liste = data.values.tolist()
8 liste2 = liste[0]
9 # liste2.reverse() #snur listen i de tilfellene man går mot metreringslinjen
10
11 --- Endres til z y x når man går mot metreringslinjen ---
12 x = liste2[0::3]
13 y = liste2[1::3]
14 z = liste2[2::3]
15
16 Hor = []
17 acchor = 0
18 Acchor = []
19 S = []
20 c = 0
21 c2 = 0
22
23 for posisjon in range(len(x) - 1):
24     hor = math.sqrt((x[c + 1] - x[c])**2 + (y[c + 1] - y[c])**2)
25     Hor.append(hor)
26     acchor = hor + acchor
27     Acchor.append(acchor)
28     s = (z[c + 1] - z[c]) / hor * 100
29     S.append(s)
30
31     c = c + 1
32     # print(c)
33     # print(x[c])
34     # print(y[c])
35
36 Q = [0,]
37 L = Q + Acchor
38
39 fig, ax1 = plt.subplots()
40
41 color = 'tab:green'
42 ax1.set_xlabel('Horisontal avstand [m]')
43 ax1.set_ylabel('Høyde [m]', color=color)
44 ax1.plot(L, z, color=color)
45 ax1.tick_params(axis='y', labelcolor=color)
46
47 fig.tight_layout() # otherwise the right y-label is slightly clipped
48 plt.show()
49
50
51 fig, ax1 = plt.subplots()
52
53 color = 'tab:red'
54 ax1.set_xlabel('Horisontal avstand [m]')
55 ax1.set_ylabel('Stigning NVDB [%]')
56 ax1.plot(Acchor, S, 'r')
57 ax1.tick_params(axis='y')
58
59 fig.tight_layout() # otherwise the right y-label is slightly clipped
60
61 plt.show()
```

