



CHALMERS



Program för energioptimering av solbil

Examensarbete inom högskoleingenjörsprogrammet Mekatronik

REBECCA OXELQVIST

FÖRORD

Detta examensarbete är utfört på företaget Broccoli i Göteborg för Institutionen för Elektroteknik på Chalmers tekniska högskola. Arbetet omfattar 15 hp och är utfört av en Mekatronikstudent under totalt 20 veckor. Arbetet har utförts för att vara till nytta för Solarteam som är Chalmers tävlingsbidrag i Bridgestone World solar challenge 2019.

Jag vill tacka min sambo Pontus för att han otröttligt har hjälpt mig att googla då min kunskap har varit otillräcklig. Jag vill också tacka Ludvig, Kimberly och Josefin som har lyssnat och hjälpt till med lösningar till diverse problem i Broccolis fikarum. Jag vill tacka min examinator Veronica som har varit stöttande och gett bra kritik vid skrivandet av detta arbete. Sist vill jag tacka min handledare Björn på Broccoli som gav mig möjligheten att utföra detta arbete.

Rebecca Oxelqvist, Göteborg juni 2019

SAMMANFATTNING

Bridgestone World solar challenge är ett solbilsrace på 3020 km som hålls varje år i Australien. Många av de bilar som tävlar i racet kommer aldrig i mål. En solbil som är med i ett race har framförallt två saker denna ska prestera: Bilen ska köra så snabbt som möjligt och batteriet får inte ta slut innan målgång. Syftet med detta projekt är att utveckla ett program som ska kunna öka en solbils prestation. För att öka prestationen har detta projekt utvecklat en modell för att kunna simulera en solbils förbrukning samt batteriets ackumulation. Projektet har också utvecklat ett allmänt program som behandlar statistik om solenergi per markyta, väderprognoser samt 27 parametrar. Programmet har utvecklats för att kunna simulera olika bilar, olika rutter samt olika väder. I detta projekt har programmet testats och simulerats utifrån data som fanns tillgänglig från en bil som kommer delta i Bridgestone World solar challenge 2019. När programmet körs, genererar det en fil med optimal hastighet för bilen vid GPS-koordinaterna bilen passerar. Programmet är utrustat med sex olika körstrategier som jämförs med varandra för att finna den bästa strategin. Simuleringar visade att programmets optimerade strategi hade en tidsvinst på 5,54 h jämfört med att hålla en konstant hastighet som precis tar slut på batteriet under tävlingen.

ABSTRACT

Bridgestone World solar challenge is a solar car race that is held in Australia. The race is 3020km long. Several of the participants never make it to the finish line. It is desirable for a participant to be able to do two things: first, make it to the finish line without depleting the battery second, drive as fast as possible. The purpose of this project is to develop a program that can help the car to perform better. This was accomplished by developing a model for calculation of the energy consumption and accumulation of the batteries. A program was developed to simulate a model for the entire route. The program was developed so that it could take the sun energy per square meter, weather forecast and 27 other parameters into account. This makes the program capable of simulating different cars at different locations under different weather conditions. The program has been tested based on data from a car that will participate in the Bridgestone World solar challenge 2019 and the competition route. When running, the program generates a file with optimized setpoints for the velocity based on the GPS-coordinates of the car. The program is equipped with six different driving strategies which are compared to each other to select the best strategy. Simulations showed that the program's best strategy gave a 5.54h time gain compared to simulations of the car keeping a constant speed that exactly depleted the battery.

INNEHÅLL

Terminologi.....	1
1 Inledning.....	2
1.1 Bakgrund	2
1.2 Syfte.....	2
1.3 Mål.....	3
1.4 Avgränsningar	3
1.5 Precisering av frågeställning	3
2 Teori	4
2.1 Bridgestone World solar challenge 2019.....	4
2.2 Energimodell	4
2.2.1 Energiförbrukning hos elbilar	4
2.2.2 Faktorer som påverkar effektiviteten hos solceller	5
2.2.3 Bilbatterier för elbilar	6
2.2.4 Generativa bromsar	6
2.3 Energiberäkningar	7
2.3.1 Friktion	7
2.3.2 Luftmotstånd	7
2.3.3 Elevationsenergi	9
2.3.4 Acceleration eller retardation	9
2.3.5 Övrig förbrukning	9
2.3.6 Total konsumtion från hastighet.....	10
2.3.7 Förluster	10
2.4 GPS-Koordinater	11
2.5 Programmering	12
2.5.1 Dynamisk minnesallokering.....	12
2.5.2 Binärsökning	13
2.5.3 Rekursiva funktioner	13
3 Metod	14
4 Funktionsbeskrivning energienhet	16
4.1 Utgående parametrar.....	16
4.2 Ingående parametrar	16
4.2.1 Elevationsfil	16
4.2.2 Solenergifil.....	17
4.2.3 Väderfil.....	17
4.2.4 Konfigurate.h.....	18

5	Funktionsbeskrivning program	24
5.1	Hjälpfunktioner.....	24
5.1.1	Konfiguration och strukturer	25
5.1.2	Fält.....	25
5.1.3	Minne och filhantering	25
5.1.4	Felmeddelande	26
5.1.5	Fysik och Matematik	26
5.1.6	Matematik.....	26
5.2	Algoritmer för att simulera energikonsumtion	26
5.3	Simulera vägstycke.....	28
5.3.1	Avstånd.....	29
5.3.2	Energi	29
5.4	Simulera hela ruten.....	31
5.5	Körstrategier	32
5.5.1	Körsätt	33
5.5.2	Strategier	33
5.6	Optimering av strategier	34
5.7	Konfiguration och datainsamling	35
5.7.1	Solenergi.....	35
5.7.2	Elevation och rutt	36
5.7.3	Konfiguration	37
6	Optimering	40
6.1	Strategi konstant hastighet.....	40
6.1.1	Optimalt resultat för konstant hastighet	42
6.2	Strategi dynamisk energi	43
6.2.1	Optimalt resultat för strategi dynamisk energi	45
6.3	Effekt vid låg hastighet.....	46
6.3.1	Optimalt resultat för effekt vid låg hastighet	48
6.4	Energi från föregående segment	48
6.4.1	Optimalt resultat för energi från föregående segment.....	50
6.5	Batterianpassad hastighet	51
6.5.1	Optimalt resultat för energi från batterianpassad hastighet.....	52
6.6	Dynamisk hastighet	53
6.6.1	Optimalt resultat för Dynamisk, hastighet	54
6.7	Resultat optimering.....	55
6.8	Regn.....	55
6.9	Simulerad tävling.....	59

7	Resultat.....	63
7.1	Måluppfyllelse	63
8	Slutsats och diskussion.....	64
8.1	Vidareutveckling	64
8.2	Slutsats.....	64
	Referenser.....	65
	Bilaga 2- resultat av väderratio verifiering	6

TERMINOLOGI

BMS:

BMS står för *Battery management system*. Vilket är ett reglersystem för batterier.

MPPT:

MPPT, *Maximum Power Point Tracking* används i reglersystem för solceller. [1] En MPPT är en DC till DC konverterare som optimerar spänningen mellan solcellen till batteriet.

SOC:

SOC står för *state of charge* och syftar till ett batteris laddningsnivå i procent.

1 INLEDNING

1.1 Bakgrund

FN har som mål att bekämpa klimatförändringar [2]. En stor del av dessa klimatförändringar kommer från koldioxidutsläpp. Fordon står för en stor del av dessa utsläpp då de använder fossila bränslen. Världen och framförallt fordonsindustrin står därmed inför en utmaning: Hur bör framtidens fordon se ut för att minimera miljöpåverkan? Ett svar på denna fråga skulle kunna vara solbilen. Solbilar använder sig dels av ett batteri och dels av solceller installerade på bilen. Detta gör att bilens utsläpp vid drift är mycket liten.

Utvecklingen av solbilar är i sin linda. Solceller utvecklas ständigt men har fortfarande förhållandevis låg verkningsgrad. Detta gör att solbilen ännu inte nått marknaden i någon större utsträckning. För att skynda på denna utveckling finns det flera universitet runt om i världen som varje år utvecklar sina egna solbilar. Solbilarna används sedan i solbilsrace.

Bridgestone World solar challenge är ett solbilsrace som hålls varje år där olika lag världen över tävlar mot varandra för att komma först längs en planerad rutt. Racet som går av stapeln 2019 hålls i Australien. Chalmers har år 2019 gått med i tävlingen med ett lag som kallas Solarteam. Det är första året solarteam deltar i tävlingen. Laget består av högskoleingenjörstudenter från programmen elektronik, data, mekatronik och maskin. Dessa studenter har under året innan tävlingen designat och byggt en solbil.

Tävlingen startar i Darwin och avslutas i Adelaide, en sträcka på 3020km. [3] Solbilarna är försedda med både batteri och solcell. Bilarna körs på allmänna vägar. Detta gör att bilarna måste kunna köra minst 60km/h. Varje år är det flera lag som går med i tävlingen som inte kommer hela vägen in i mål. Batteriet tar slut eller så kan inte bilen hålla lägsta tillåtna hastighet.

En solbils användning av energi varierar med terrängen samt hastigheten som bilen håller. Energin från solen kan variera både beroende av solstyrka och väder. Detta gör att bilen har en begränsad mängd energi till förfogande som bör användas resurseffektivt, för att säkerställa målgång. Resurseffektiviteten kan säkras upp genom att hastigheten och acceleration begränsas.

Hastigheten ska dock vara så hög som möjligt för att Solarteam ska ha en chans att vinna racet. Denna avvägning mellan hög hastighet och kunna köra hela sträckan utan att batteriet tar slut kan lösas på många olika sätt och med många olika strategier. I detta projekt har ett program utvecklats för att ge stöd i valet av körstrategi.

1.2 Syfte

Syftet med detta projekt är leverera ett program som kan beräkna optimal hastighet för en solbil. Hastigheten ska optimeras så att solbilen med hög sannolikhet kan köra en viss sträcka utan att energin tar slut. Solbilen ska dessutom köras så snabbt som möjligt.

1.3 Mål

Målet med projektet är att utveckla ett program som kan:

- med hög sannolikhet förutspå energiförbrukningen hos en solbil
- med hög sannolikhet beräkna energiackumulationen i batteriet hos en solbil.
- Räkna ut en lämplig körstrategi för solbilen så att den dels kommer i mål och dels kommer i mål så snabbt som möjligt.

1.4 Avgränsningar

- Resultatet av programmet kommer inte jämföras med datainsamling från en verklig solbil vid drift.
- Programmet kommer inte implementeras i bilen i detta projekt.
- Programmet kommer skriva sitt resultat till en fil och ej återge informationen på något annat sätt.

1.5 Precisering av frågeställning

För att med hög sannolikhet kunna beräkna förbrukningen hos en solbil krävs följande samband och modeller:

- Samband mellan solstyrka och laddning
- Samband mellan hastighet och luftmotstånd
- Samband mellan bilens lutning och energiförbrukning
- Bilens tyngd
- Modell för hjulens friktion och värmeutveckling i förhållande till hastighet
- Modell för beräkning av energi som finns till förfogande för resterande del av loppet
- Modell för energiförbrukning för återstående tid av lopp

Följande frågor behöver undersökas och besvaras:

- Vilka parametrar begränsar maxhastigheten för bilen?
- Kan den förväntade momentana energiförbrukningen estimeras baserat på statistik?

2 TEORI

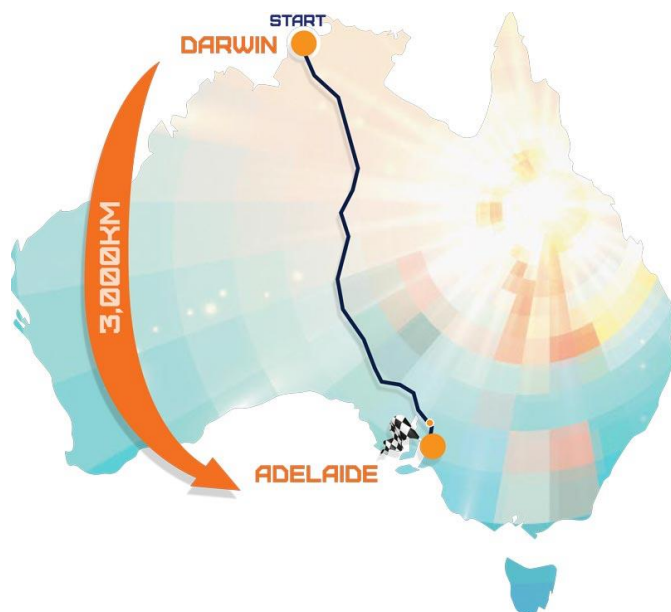
I detta kapitel följer en beskrivning av de teorier och den teknik som använts i projektet.

2.1 Bridgestone World solar challenge 2019

Bridgestone World solar challenge är en solbilstävling som år 2019 går den 3020 km långa sträckan från Darwin till Adelaide i Australien, Figur 2-1 [3]. Solbilarna körs på allmänna vägar vilket innebär att de måste kunna köra minst 60 km/h.

Solarteam kommer att vara med i tävlingen under klassen *Challenger*. Denna tävlingsklass har solbilar designade för att vara enmansfordon. Solbilen körs 8 timmar om dagen med ett mindre stopp vid lunch. Solbilen får enbart laddas under körning. När bilen står stilla kommer batteriet att kopplas ut ur solcellerna för att förhindra laddning.

Skulle bilen köra långsammare än 60km/h eller om batteriet tar slut kommer solbilen att förlora tävlingen.



Figur 2-1 Tävlingsrutten över Australien, Bridgestone World solar challenge 2019. [3]

2.2 Energimodell

De faktorer som påverkar om bilens batteri tar slut kan delas upp i två stora undergrupper: energiförbrukningen hos elbilen samt effektiviteten hos solcellen. Dessa två påverkas av olika faktorer.

2.2.1 Energiförbrukning hos elbilar

Det finns många olika faktorer som påverkar en elbils förbrukning. Dessa kan delas in i fyra olika kategorier. Dessa kategorier är bilens teknik, artificiella miljöfaktorer, naturliga miljöfaktorer samt förarens körstil.

Teknik

Batterisystemet är en essentiell del av en elbil och dess teknik har en stor inverkan på elbilens förbrukning. [4] Detta beror på att batteriet kan designas på flera olika sätt bland annat kan batteritypen, antalet celler och *battery management system*, BMS variera. Dessa faktorer påverkar i sin tur batteriets kapacitet, vikt och energitäthet. Laddningen kan också påverka energiförbrukningen. Detta beror på att en hög laddningsnivå påverkar återladdning vid, t.ex. inbromsning. Höga temperaturer på batteriet påverkar dessutom energiförbrukningen negativt.

Klimatsystemet i bilen är en annan faktor som påverkar förbrukningen. [4] Det är framförallt uppvärmning av bilen som drar energi. Denna faktor har dessutom en stark koppling med förarens preferenser samt det lokala klimatet.

Drivlinan och effektiviteten i motorn påverkar bilens förbrukning till en hög grad. [4] Faktorer som bilens massa, luftmotståndet samt rullningsmotståndet är andra faktorer som starkt påverkar bilens förbrukning.

Andra funktioner som inte är direkt relaterade till bilens drivning så som ljus, HMI, GPS och så vidare har låg inverkan på bilens förbrukning. [4]

Artificiella miljöfaktorer

Till Artificiella miljöfaktorer hör alla de miljöfaktorer som kan relateras till människan. [4] Till dessa hör trafik, trafikljus och vilken typ av körning som görs, t.ex. stadskörning eller landsbygdskörning, trafikens flöde samt medelhastighet. Dessutom ger trafikstockningar en hög förbrukning på grund av den alternerande accelerationen och retardationen.

Naturliga miljöfaktorer

Den naturliga miljön som bilen körs i påverkar bilens förbrukning. Denna faktor påverkas av klimatet bilen befinner sig i, vädret och topografin. [4] En bil som befinner sig i ständig uppförsbacke kommer att konsumera mer energi. Dessutom är både bil och människan känsliga för klimatets påverkan. Det mänskliga beteendet påverkas t.ex. av regn och batteriet påverkas som nämnt av temperatur.

Förarens körstil

En förare som kör aggressivt d.v.s. accelererar och retarderar ofta, gör att bilen konsumerar mer el. [4] Hur aggressivt en förare kör är i sin tur beroende av vilket sinness tillstånd förvararen är i samt klimatet i bilen.

2.2.2 Faktorer som påverkar effektiviteten hos solceller

Det finns flera faktorer som påverkar effektiviteten hos solceller. Utvecklingen av nya högpresterande material som kan användas i solceller går fort framåt. Solceller är dock fortfarande begränsade i sin förmåga att omvandla solenergi till elenergi. [5] Högpresterande solceller kan ha en verkningsgrad från 12 % till 20 %.

De största klimatmässiga faktorerna som starkt påverkar en solcells prestanda är solens strålning, temperatur, vindhastigheten, relativa luftfuktigheten samt damm. [5] Under de flesta förhållanden påverkas solcellens uteffekt mest av solstrålningen och dess temperatur. Detta på grund av att högre solstyrka ger en högre temperatur. En hög temperatur påverkar solceller negativt. Höga vindhastigheter runt solcellen ger en nedkylning. [5] Detta ökar effektivitet hos

solcellen då höga temperaturer påverkar solcellen negativt. De höga vindhastigheterna tar även bort damm som annars påverkar solcellen negativt.

MPPT

MPPT används i reglersystem för solceller. [1] En MPPT är en DC till DC konverterare som optimerar spänningen mellan solcellen till batteriet. Detta sker genom att konvertera den högre spänningen från solcellen till en lägre spänning för batteriet. Optimeringen görs genom att ändra antalet celler som laddas så att varje battericell får optimal spänning. Detta system har vissa förluster och drar därmed energi.

2.2.3 Bilbatterier för elbilar

Det batteri som används i dagens elbilar är litium-jon batteriet. [6] Detta batteri har en hög energitäthet och en lätt vikt jämfört med andra batterialternativ som finns på marknaden.

SOC

Litium-jon batterier påverkas av sin *state of charge*, SOC vilket är ett fönster för vilket batteriet fungerar optimalt. [7] SOC är vanligtvis från 10 % till 100 % av laddningen. Om batteriet laddas över det normala kommer batteriets temperatur att stiga kraftigt. Laddas batteriet långt över sin maximala kapacitet kan batteriet svälla eller till och med sprängas och utgör därmed en säkerhetsrisk. Under fönstret kommer batteriets spänning att stiga exponentiellt tills den kommer till en normal nivå av SOC. Batteriets funktion påverkas därför kraftigt av en låg nivå av SOC.

BMS

Bilbatterier är utrustade med BMS, som används som reglersystem. [6] Systemet övervakar battericellernas funktion och använder givare för att övervaka temperaturer och spänningar. Systemet övervakar också strömmen och stänger av och på celler vid behov. Systemet används dessutom till att reglera temperaturen hos batteriet med hjälp av kylare och värmare.

BMS systemet måste se till att hela tiden hålla batteriets laddning inom SOC för att batteriet ska fungera korrekt. I de fall batteriet under- eller överstiger gränserna för SOC kommer BMS stänga av systemet eller laddningen. [6] BMS har vissa förluster och drar därmed energi.

2.2.4 Generativa bromsar

I elbilar kan motorn användas för bromsning. [8] Detta görs genom att motorn kör baklänges vilket ger en kraft motsatt färdriktningen. Under tiden detta sker alstras elektricitet i motorn som kan användas för att ladda batteriet. Bilar utrustade med generativbromsning har alltid ett sekundärt bromssystem som går igång vid hård inbromsning. Ett vanligt sekundärt bromssystem är en skivbroms. Skivbromsen alstrar värme vilket leder till att en hård inbromsning inte laddar batteriet i samma utsträckning utan att energi går till spillo.

Vid en mjuk inbromsning eller motorbromsning kan motorn ostört fungera som en generator. Detta leder till mycket små förluster och batteriet kan laddas med stor verkningsgrad av denna energi. Vid en hård inbromsning används både den generativa bromsen och sekundärbromsen. Sekundärbromsen har så stora förluster att det förbrukas energi från batteriet. I en elbil som körs på en motorväg brukar denna förbrukning vara i snitt 17% högre än den energi som den generativa bromsen alstrar vid inbromsningsförloppet [8].

2.3 Energiberäkningar

Nedan följer en beskrivning av de formler och samband som använts i projektet för att ta fram en matematisk modell av bilens konsumtion, batteriets laddning/urladdning samt solcellens effekt.

2.3.1 Friktion

Friktion påverkar bilen under körning och är en teknikrelaterad förlust. Denna förlust påverkar bilen starkt [4] och är därför viktig att väga in.

Friktionen som påverkar bilens förbrukning hittas framförallt i motor, drivlina samt hjul och däck. [4] Bidraget från drivning, hjul och däck, kan ses som en konstant, då energin som går åt att övervinna friktionen är proportionellt mot avståndet bilen färdats [9]. Energin som går åt för att övervinna friktionen kan därför ses som:

$$\Delta x = \text{körd sträcka}$$

$$\Delta E_{\text{friktion}} = F_{\text{friktion}} \cdot \Delta x \quad (2.1)$$

$$\text{Där } F_{\text{friktion}} = F_{\text{däck,hjul}} + F_{\text{drivlina}}$$

2.3.2 Luftmotstånd

Luftmotstånd påverkar bilen starkt under körning [4]. Luftmotståndet storlek påverkas starkt av bilens teknik, så som bilens tvärsnittsarea samt luftmotståndskoefficient. Luftmotståndet påverkas dessutom av den naturliga miljön kring bilen, då densiteten på luften är en faktor som påverkas av luftfuktighet samt temperatur.

Då bilen färdas framåt påverkas den av luftmotstånd. Energin som går åt till att övervinna luftmotstånd räknas ut enligt [10]:

$$\Delta E_{\text{luftmotstånd}} = \Delta x \cdot F_d = \frac{1}{2} C_d \rho A v^2 \Delta x$$

Där

ρ = luftens densitet C_d = luftmotståndskoefficient A =bilens tvärsnittsarea

Om ett antagande görs om att luftens densitet förblir den samma över hela resan är $\frac{1}{2} C_d \rho A$ konstant. Detta gör att luftmotståndet kan beräknas enligt:

$$\Rightarrow \Delta E_{\text{luftmotstånd}} = K_{\text{luftmotstånd}} v^2 \Delta x \quad (2.2)$$

$$\text{Där } K_{\text{luftmotstånd}} = \frac{1}{2} C_d \rho A$$

Detta gäller då hastigheten är konstant. Om i stället accelerationen är konstant kommer luftmotståndet påverkas annorlunda.

Antag konstant acceleration samt att sträckan börjar vid noll.

Det ger sambanden:

$$x = a \frac{t^2}{2} \quad (2.3)$$

$$t = \frac{v}{a} \quad (2.4)$$

Som i sin tur leder till:

$$(2.3) \text{ in i } (2.4) \quad \Rightarrow \quad x = \frac{a v^2}{2 a^2} = \frac{v^2}{2a} \quad (2.5)$$

$$\Rightarrow v = \sqrt{2ax} \quad (2.6)$$

$$\Delta E = \int F dx$$

tillsammans med (2.2) fås

$$\Delta E_{\text{luftmotstånd under acceleration}} = \int K_{\text{luftmotstånd}} v^2 dx$$

$$\text{med (2.6)} \Rightarrow \Delta E_{\text{acceleration}} = \int K_{\text{luftmotstånd}} \sqrt{2ax}^2 dx$$

Då ekvationen ska användas med start- och slut-hastighet används $x(v_{\text{start}})$ samt $x(v_{\text{slut}})$ som gränser $\Rightarrow$

$$\begin{aligned} \Delta E_{\text{luftmotstånd under acceleration}} &= \int_{\frac{v_{\text{start}}^2}{2a}}^{\frac{v_{\text{slut}}^2}{2a}} K_{\text{luftmotstånd}} 2ax dx = \\ &= K_{\text{luftmotstånd}} \frac{(v_{\text{slut}}^4 - v_{\text{start}}^4)}{4a_{\text{bil}}} = \end{aligned}$$

$$= K_{\text{luftmotstånd}} \frac{(v_{\text{slut}}^2 + v_{\text{start}}^2)}{2} \frac{(v_{\text{slut}}^2 - v_{\text{start}}^2)}{2a_{\text{bil}}} = K_{\text{luftmotstånd}} \frac{(v_{\text{slut}}^2 + v_{\text{start}}^2)}{2} \Delta x \Rightarrow$$

$$\Delta E_{\text{luftmotstånd under acceleration}} = K_{\text{luftmotstånd}} \frac{(v_{\text{slut}}^2 + v_{\text{start}}^2)}{2} \Delta x \quad (2.7)$$

2.3.3 Elevationsenergi

Bilen påverkas av hur vägen ser ut vilket är en naturlig miljöfaktor [4]. När det är backigt påverkar detta på två olika sätt. Om det är nedförsbacke kan lägesenergin bilen har användas för att öka hastigheten eller ladda batteriet via generativ inbromsning. Om det är uppförsbacke måste bilen förbruka energi för att köra upp för backen. För att kunna simulera bilens förbrukning över loppet behöver därför elevationen räknas in.

Energien för att förflytta bilen uppför eller nerför en lutande väg ges enligt:

$$\Delta E_{\text{elevation}} = M_{\text{bil}} g \cdot \sin(\theta_{\text{lutningväg}}) \Delta x \quad (2.8)$$

2.3.4 Acceleration eller retardation

Acceleration är en av de faktorer som påverkar hur mycket bilen förbrukar. Denna faktor är stark förknippad med förarens körstil [4]. Om föraren accelererar och retarderar ofta kommer detta att öka konsumtionen av energi. När en solbil kör kan detta i vissa fall vara önskvärt, då en lägre hastighet kan resultera i en nettovinst av laddning av batteriet. Om batteriet laddas mer än vad bilen konsumerar samt om retardationen görs med en mjuk inbromsning kan energi ackumuleras i batteriet.

Accelerationen och retardationen är en stor faktor för hur mycket bilen förbrukar och är därmed av intresse att väga in i energimodellen.

Energien som går åt att accelerera eller retardera bilen från en hastighet till en annan fås enligt [10]:

$$\Delta E_{\text{acceleration}} = M_{\text{bil}} \frac{(v_{\text{slut}}^2 - v_{\text{start}}^2)}{2} \quad (2.9)$$

2.3.5 Övrig förbrukning

En elbils förbrukning påverkas av andra sidofunktioner som förbrukar energi. En av dessa sidofunktioner som kan förbruka mycket energi är klimatanläggningen. [4] Andra funktioner så som lampor och mikrokontrollers förbrukning är dock försumbar. Slutsatsen blir att det är av vikt att ta hänsyn till klimatanläggningen om en sådan är installerad.

Eftersom denna förbrukning är proportionell mot tiden behövs ett uttryck för tiden det tar att köra en viss sträcka. Vid konstant acceleration ges då:

$$t = \frac{2\Delta x}{(v_{slut} + v_{start})}$$

vilket leder till att förbrukningen kan uttryckas som:

$$\Delta E_{\text{övrig förbrukning}} = W_{\text{effekt övrig förbrukning}} \cdot \frac{2\Delta x}{(v_{slut} + v_{start})} \quad (2.10)$$

2.3.6 Total konsumtion från hastighet

Den totala energin att föra bilen framåt kan räknas ut genom att addera alla delar som bidrar till förbrukningen.

Den totala energikonsumtionen för att bilen ska röra sig en viss sträcka blir totalt:

$$\begin{aligned} \Delta E_{\text{tot}} = M_{\text{bil}} & \frac{(v_{\text{slut}}^2 - v_{\text{start}}^2)}{2} \\ & + \left(M_{\text{bil}} g \sin(\theta_{\text{lutningväg}}) + K_{\text{luftmotstånd}} \frac{(v_{\text{slut}}^2 + v_{\text{start}}^2)}{2} + F_{\text{friktion}} \right) \\ & \cdot \Delta x + W_{\text{effekt övrig förbrukning}} \cdot \frac{2\Delta x}{(v_{\text{slut}} + v_{\text{start}})} \end{aligned} \quad (2.11)$$

Enligt (2.1), (2.2), (2.7), (2.8), (2.9) samt (2.10)

2.3.7 Förluster

Utöver det bilen konsumerar för att röra sig framåt finns även förluster i de delar som tillför energi.

Motor

Motorn ska övervinna den kraft som krävs att driva bilen samt motorns egen värmeutveckling och friktion. Detta gör att motorn förbrukar energi enligt:

$$\frac{\Delta E_{\text{tot}}}{\eta_{\text{motor}}} = \Delta E_{\text{motor}} \quad (2.12)$$

Detta gäller så länge $0 < \Delta E_{\text{motor}} \leq W_{\text{motor}} \cdot t$

Förbrukar inte bilen någon energi för att röra sig framåt kommer inte motorn att förbruka någon energi.

Motorn har även en maxeffekt, detta gör att om det krävs större energi än maxeffekten att föra bilen framåt i tex en uppförsbacke kommer bilen att retardera istället för att dra ytterligare energi från batteriet.

Generativ broms

Den energi batteriet laddas med vid mjuk inbromsning fås av:

$$\Delta E_{\text{generativ broms}} = \Delta E_{\text{tot}} \cdot \eta_{\text{generativ broms}} \quad (2.13)$$

Detta gäller endast då bilens totala energiförbrukning är mindre än noll, batteriet är i en önskad SOC samt den totala effekten batteriet laddas med är mindre än den maximala effekten batteriet kan laddas med.

Solcell

Energien som solcellen omvandlar från solenergi till elektrisk energi kan ses som:

$$E_{\text{solen}} = \text{Energien från solen} \left[\frac{\text{W}}{\text{m}^2} \right]$$
$$R_{\text{väder}} = \frac{E_{\text{från solen}}(\text{bestämt väder})}{E_{\text{från solen}}(\text{medel})} \quad (2.14)$$

$$\Delta E_{\text{solcell}} = E_{\text{solen}} A_{\text{solcell}} \eta_{\text{solcell}} \eta_{\text{MPPT}} \Delta t R_{\text{väder}} \quad (2.15)$$

Där $E_{\text{från solen}}(\text{medel})$ syftar till den medelenergi som träffar en kvadratmeter markyta per sekund under en bestämd tidsperiod under en månad.

$E_{\text{från solen}}(\text{bestämt väder})$ syftar på den medelenergi som träffar en kvadratmeter markyta per sekund under ett dygn vid ett bestämt väder.

Batteri

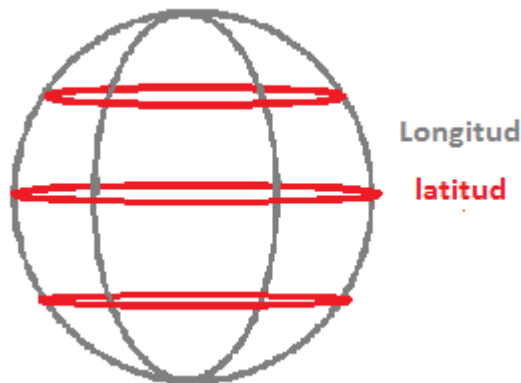
Energien som batteriet kan återge efter laddning kan ges av:

$$E_{\text{batteri}} = \eta_{\text{batteri}} E_{\text{laddning}} \quad (2.16)$$

Detta gäller så länge batteriet är i en önskad nivå av SOC. Är batteriet i en SOC som är större än ca 100% kommer batteriet inte gå att ladda.

2.4 GPS-Koordinater

GPS-koordinater kan beskrivas i longitud och latitud. Longituden har ett spann mellan 0° och 180° och går från det södra halvklotet till det norra. Latituderna sträcker sig rätvinkligt i förhållande till longituden och spänner sig från -90° till 90°, se Figur 2-2.



Figur 2-2 Jordens longitud och latitud ritad schematiskt.

För att räkna ut avståndet, x mellan två koordinater fågelvägen kan följande ekvation användas:

λ =longitud i radianer ϕ =latitud i radianer $\phi_m = (\phi_1 + \phi_2)/2$

$$x_{\text{fågelvägen}} = R_{\text{jorden}} \sqrt{(\Delta\phi)^2 + (\cos(\phi_m)\Delta\lambda)^2} \quad (2.17)$$

Ekvationen kompenserar för det faktum att avståndet som motsvarar 1° latitud är beroende av longituden. Till exempel är 1° skillnad i latitud vid ekvatorn väldigt mycket längre än 1° skillnad i latitud vid polerna.

Ekvation (2.17) kallas för ”spherical earth projected to a plane” Denna approximation av avståndet mellan två GPS-punkter är enkel samt ger ett bra resultat vid korta avstånd. [11]

För att vidare få ett uttryck på den färdade sträckan som även innehåller färd i höjdled kan Pythagoras sats användas tillsammans med (2.17) enligt:

$$x = \sqrt{x_{\text{fågelvägen}}^2 + \Delta h^2} \quad (2.18)$$

2.5 Programmering

Nedan följer de teorier som använts vid programmeringen av energienheten.

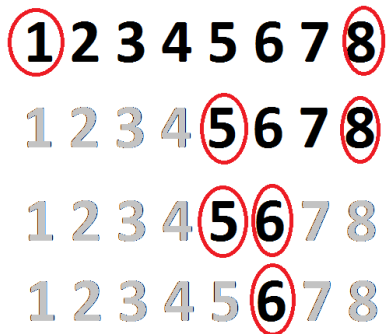
2.5.1 Dynamisk minnesallokering

Interna variabler frisläpps vid slutet av funktioner. [12] Detta gör att minne måste allokeras för att variablers livslängd ska förlängas utanför funktionen där de anropas. Dynamisk minnesallokering är framförallt viktigt för att hålla ner mängden minne som är allokerat vid en viss tidpunkt. Framförallt vid fält där programmet själv räknar ut antalet variabler som ska ingå. Allokeras inte minnet dynamiskt, blir alternativet att i förväg allokeras det maximala minnet som t.ex. ett fält kan tänkas behöva.

2.5.2 Binärsökning

Binärsökning används då en lista eller ett fält är sorterat. [12] Binärsökning är ett mer effektivt sätt att söka igenom sorterade listor eftersom det kräver färre jämförelser än rak sökning som letar igenom en lista från början till slut.

Binärsökning undersöker om första eller sista indexet i ett fält är närmst det sökta elementet. [12] Visar det sig att det sista elementet är närmst, halveras sökarean mot det sista elementet. Visar det sig att det första elementet är närmst, halveras sökarean mot det första elementet. Detta sker upprepade gånger till det sökta elementet är hittat, se Figur 2-3.



Figur 2-3, Illustration av binärsökning efter siffran 6 i ett fält med 8 sorterade siffror.

2.5.3 Rekursiva funktioner

En rekursiv funktion är en funktion som indirekt eller direkt anropar sig själv. [12] När programmet exekveras kan flera upplagor av funktionen finnas samtidigt. Varje upplaga har sina egna variabler samt värden och försvinner först då anropet för upplagan avslutas.

3 METOD

Detta projekt har tagit fram en modell för energiförbrukning för solbilar. Modellen tar hänsyn till väder, solenergi, vägens utformning, körstil samt bilens teknik. Modellen är anpassad för att räkna ut förbrukningen hos en förutbestämd sträcka för en förutbestämd solbil men kan konfigureras för att passa olika bilar och olika sträckor. För att göra detta har flera olika samband tagits fram och 27 parametrar räknats med i modellen.

Samband mellan vägens utformning samt energiförbrukningen hos bilen har tagits fram med hänsyn till elevationen. Vägens kurva har dock inte tagits med i beräkningen vilket kan påverka luftmotståndet bilen utsätts för samt hur snabbt bilen kan framföras.

Samband mellan hastighet samt förbrukning har tagits fram. Detta påverkar förbrukningen vid acceleration och retardation på grund av rörelseenergin samt luftmotståndets storlek. Den relativa vindhastigheten har således inte innefattats av modellen. Accelerationen eller retardationen har antagits konstant.

Modellen som tagits fram kan begränsas efter aktuell bils teknik. Motorns maxeffekt begränsar hur mycket energi som kan konsumeras till bilens drivning. Batteriets SOC, maximal laddningsförmåga och total laddningskapacitet begränsar batteriets laddning. Effektiviteten hos motorn, solcellen, batteriet och BMS används. Även fysiska storheter så som bilens massa, solcells area, bilens luftmotstånd, bilens friktion och övriga funktioners effekt tas också med i beaktande.

Modellen är inte anpassad för att ta hård inbromsning i beaktande och räknar med att denna förbrukning är försumbar. Detta antagande har gjorts utifrån att solbilen framförallt ska framföras på motorväg. Detta gör också modellen mycket mindre exakt vid stadskörning.

Parametrarna som använts för att testa modellen har tagits dels från Solteams solbils specifikationer, Google earth samt Australian Government bureau of Meteorology.

Från Google earth har vägens GPS-koordinater samt vägens elevation hämtats.

Från Australian Government bureau of Meteorology har statistik över solenergi hämtats. Denna information har använts för att göra en approximation över hur många procent solenergi av snittsolenergin som träffar marken vid sol, moln och regn.

Statistiken över solenergi har dessutom hämtats för olika GPS-koordinater runt om vägen varvid bilen kör. Denna statistik har sedan använts för att ta hänsyn till de olika klimat som kan finnas längs vägen. T.ex. kan en bergskedja ge mer regn lokalt.

Modellen har programmerats i ett C program i IDE:n codelite. Programmet använder sig av en konfigureringsfil, fil med GPS-koordinater och elevation, fil med GPS-koordinater och solenergi samt fil med GPS-koordinater och väder efter klockslag. Programmet använder energimodellen för att simulera flera olika körstrategier som sedan vägs mot varandra. Resultatet skrivs till en fil.

Programmet använder sig av dynamisk minnesallokering samt läser och skriver från filer. Detta görs för att hålla ner minnet som finns allokerat i RAM-minnet. Programmet använder sig också av binärsökning för att hålla nere exekveringstiden av programmet.

4 FUNKTIONSBESKRIVNING ENERGIEHET

Detta avsnitt beskriver hur användaren kan bruka programmet samt mer utförligt hur programmet fungerar. Innan användning ska programmet konfigureras. Detta ska göras för varje gång det finns ny data, innan programmet körs. Detta görs via tre textfiler samt en headerfil vid namn konfigure.h. När programmet exekveras svarar den med en fil med resultatet av dess algoritm. Denna fil innehåller en beräknat optimal hastighet för varje segment av sträckan. Nedan följer en beskrivning av vilka parametrar som ska ställas in samt vad slutresultatet av programmet blir.

4.1 Utgående parametrar

Programmet skapar en fil med:

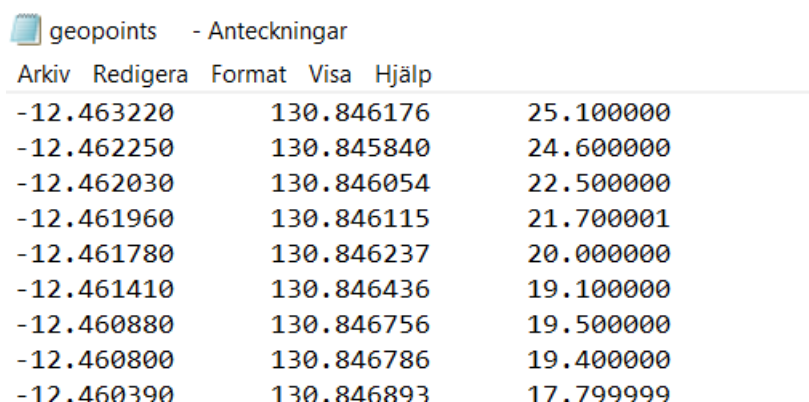
1. GPS-koordinater över rutten
2. GPS-koordinatens elevationsskillnad fram till nästa punkt.
3. Hel timme från start
4. Tid från start i timmar
5. Beräknad laddningsnivå hos batteriet vid GPS-koordinaten
6. Optimal hastighet fram till nästkommande GPS-koordinat

4.2 Ingående parametrar

I konfigure.h ställs 27 parametrar in för att programmet ska fungera så bra som möjligt. I en elevationsfil, solenergifil samt en väderfil ska statistik över körvägen finnas.

4.2.1 Elevationsfil

Elevationsfilen som tillhör programmet ska uppladdas i debug mappen för programmet. Textfilen ska innehålla GPS-koordinater över vägen som ska köras samt elevationen i dessa punkter, se Figur 4-1. Den första kolonnen är data över latituder, den andra kolonnen är data över longituder och den sista kolonnen är elevationen i meter över havet. Dessa parametrar läses in till programmet som flyttal. Om filen är tom eller om datan är av fel typ kommer användaren få ett felmeddelande. Fel format för filen kan t.ex. vara att latituden är över 90 grader eller longituden är negativ.



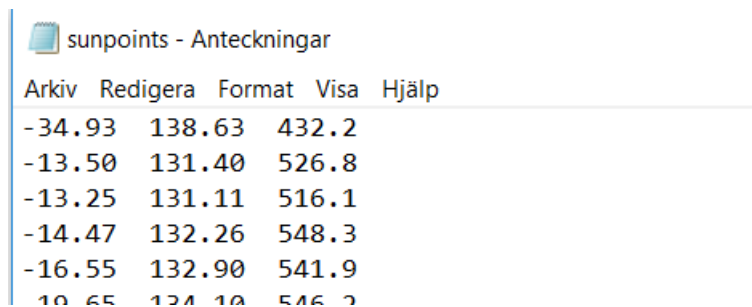
geopoints - Anteckningar

Arkiv	Redigera	Format	Visa	Hjälp
-12.463220		130.846176		25.100000
-12.462250		130.845840		24.600000
-12.462030		130.846054		22.500000
-12.461960		130.846115		21.700001
-12.461780		130.846237		20.000000
-12.461410		130.846436		19.100000
-12.460880		130.846756		19.500000
-12.460800		130.846786		19.400000
-12.460390		130.846893		17.799999

Figur 4-1 Figur som visar hur textfilen med elevationsdata kan se ut.

4.2.2 Solenergifil

Solenergifilen som tillhör programmet ska uppladdas i debug mappen för programmet. Textfilen ska innehålla GPS-koordinater och solenergi per kvadratmeter hos denna punkt, se Figur 4-2. Den första kolonnen är data över latituder, den andra kolonnen är data över longituder och den sista kolonnen är solenergi som träffar en kvadratmeter markyta per sekund i joule. Även här får användare felmeddelande vid fel format eller om filen är tom. Datatypen för filen är åter flyttal.



-34.93	138.63	432.2
-13.50	131.40	526.8
-13.25	131.11	516.1
-14.47	132.26	548.3
-16.55	132.90	541.9
-19.65	134.10	546.2

Figur 4-2 Figur som visar hur textfilen med solenergidata kan se ut.

4.2.3 Väderfil

Väderfilen som tillhör programmet ska uppladdas i debug mappen för programmet. Textfilen ska innehålla GPS-koordinater och information om väder, se Figur 4-3. Den första kolonnen är data över latituder, den andra kolonnen är data över longituder och de sista kolonnerna är data över väder timme för timme. Den första kolonnen är timme ett från det att bilen har börjat köra, nästa är timme två och så vidare. Antalet aktuella timmar kan ställas in i konfigurationsfilen, där de maximala antalet timmar är 16 timmar, se nedan för hur denna parameter ställs in.



-34.93	138.63	1	1	1	1	1	1	1	1
-13.50	131.40	1	1	1	1	2	2	2	2
-13.25	131.11	1	1	1	1	1	1	1	1
-14.47	132.26	0	0	0	1	1	1	1	1
-16.55	132.90	1	1	1	1	1	1	1	1
-19.65	134.10	1	1	1	1	1	1	1	1

Figur 4-3 Figur som visar hur textfilen med väderdata kan se ut.

GPS-koordinaterna representeras som flyttal. Väderdatan representeras som två bitar. Detta gör att väderdatan enbart kan anta ett värde mellan noll och tre. Dessa värden representerar olika väder, se Tabell 4-1. Programmet ger användaren ett felmeddelande om något i filen skulle vara av fel format eller om filen är tom. Programmet varnar också om det finns för många timmar med data.

Tabell 4-1 över värden på väderdata

0	Lätta moln eller ingen väderdata
1	Sol
2	Mycket moln
3	Regn

4.2.4 Konfigurate.h

Konfiguration av konfigurate.h görs via 27 parametrar. Dessa kan delas in i fem kategorier och täcker allt från batteriets teknik till filhantering. Nedan följer först en översikt av de inställbara parametrarna, se Tabell 4-2 och vidare en mer ingående förklaring av dessa.

Tabell 4-2 Inställbara parametrar

Parameternamn	Kort förklaring [enhet]
ENERGYFILE	Filnamn på solenergifil
ELEVATIONFILE	Filnamn på elevationsfil
WEATHERFILE	Filnamn på väderfil
SPEED_SET_POINT_FILE	Filnamn på resultatfil
EARTH_RADIUS	Jordens radie [m]
GRAVITATION	Gravitationen [m/s^2]
FRICTION	Rullmotståndet för hela bilen [F]
MASS	Massan på bilen [kg]
AIRRESISTANCE_KONSTANT	Konstanta luftmotståndet för bilen, se avsnitt 2.3.2 Luftmotstånd
MOTOR_MAX_POWER	Motorns maxeffekt [w]
MOTOR_EFFICIENCY	Motorns verkningsgrad [%/100]
REGENERATIVE_BRAKE_EFFICIENCY	Generativa bromsens verkningsgrad [%/100]
GENERATIVE_BRAKE	Om generativ broms finnes [1-ja/2-nej]
OTHER_FUNCTIONS	Förbrukning i watt för kringfunktioner
BATTERY_START	Batteriets startnivå [%/100]
BATTERY_MAX_WORKING_LEVEL	Batteriets startnivå [%/100]
BATTERY_MINI_WORKING_LEVEL	Batteriets startnivå [%/100]
BATTERY_TOTAL_CAPACITY	Batteriets kapacitet i Joule [J]
BATTERY_EFFICIENCY	Batteriets verkningsgrad [%/100]
BATTERY_CHARGE_MAX	Batteriets maximala laddning per sekund
SUNENERGY_EFFICIENCY	Solcellens verkningsgrad [%/100]
SUNENERGY_AREA	Solcellens area [m^2]
MINIMUM_SPEED	Minimala optimerings hastighet [km/h]
MAXIMUM_SPEED	Maximala optimerings hastighet [km/h]

Krafter

Vidare ställs vissa av krafterna som verkar på bilen in, se Figur 4-6. Det första som ställs in är en approximation på friktionskraften för hela bilen. Friktionen som verkar på bilen från däck, lager, drivning och andra rörliga delar kan ses som konstant i förhållande till sträckan [9].

```
//*****
//****          Teknologi          ****
//*****

//*****Force constants*****

#define FRICTION 13.45 → → → → → // [N]
#define MASS 300 → → → → → // [kg]
#define AIRRESISTANCE_KONSTANT 0.099 → → → → → // Force constant ( $K \cdot v^2 = F$ )
```

Figur 4-6, Bild över hur krafter ställs in i konfigure.h

Massan för bilen representerar hela bilens massa med förare och last inräknat.

Under denna rubrik ställs även `AIR_RESISTANCE_KONSTANT` in. Denna parameter är ekvivalent med $K_{\text{luftmotstånd}}$ i formel (2.2).

Motor

Under rubriken motor ställer användaren in motorns specifikationer, se Figur 4-7. Dessa specifikationer är motorns maximala effekt och motorns verkningsgrad. Det sista som ställs in är vilken verkningsgrad den regenerativa bromsen har och sist kan den generativa bromsen stängas av eller på. Detta gör att programmet kan simulera med eller utan generativa bromsar.

```
//*****Motor*****

#define MOTOR_MAX_POWER 5000 → → → → → // [W]
#define MOTOR EFFICENCY 0.94 → → → → → // [%/100]
#define REGENERATIVE_BRAKE EFFICENCY 0.98 → → → → → // [%/100]
#define GERENETIVE_BRAKE 1 → → → → → // [1-YES, 0-NO]
```

Figur 4-7, Bild över hur motorns specifikationer ställs in i konfigure.h

Övriga funktioner

I det fall bilen har en klimatanläggning som drar mycket energi kan denna parameter ställas in för att ta dess förbrukning i beaktande, se Figur 4-8. Har inte bilen någon klimatanläggning utan enbart lampor och få kringenheter som mikrokontroller etc. kan denna parameter ses som försumbar [4].

```
//*****other functions*****//
#define OTHER_FUNCTIONS 0 → → → → → // [W]
```

Figur 4-8, Bild över hur övriga funktioners förbrukning ställs in i konfigure.h

Batteriet

Batteriets specifikation ställs in under rubriken *Battery*, se Figur 4-9. Det första som ställs in är vilken nivå av laddning som batteriet har vid start. Nästa parameter är övregränsen för SOC. Alltså vid vilken nivå av laddning som BMS stänger av laddning av batteriet. Vidare ställs batteriets lägre SOC gräns in. Om modellen simuleras över bestämd rutt och batteriet kommer under denna nivå räknas det som att bilen måste stanna av säkerhetsskäl.

Den fjärde parametern som ställs in är batteriets totala kapacitet i Joule.

BATTERY_EFFICENCY är batteriets verkningsgrad.

Den sista parametern är batteriets maximala laddningsförmåga i watt.

```
//*****Battery*****

#define BATTERY_START 1 → → → → → // [%/100]
#define BATTERY_MAX_WORKING_LEVEL 0.95 → → → // [%/100]
#define BATTERY_MINI_WORKING_LEVEL 0.1 → → → // [%/100]
#define BATTERY_TOTAL_CAPACITY 1393600 → → → // [J](mAh*nr_of_cells*3,6)
#define BATTERY_EFFICENCY 0.96 → → → → // [%/100]
#define BATTERY_CHARGE_MAX 2926 → → → → // [w]
```

Figur 4-9, Bild över hur batteriets specifikationer ställs in i *konfigure.h*

Solcell

Solcellens inställbara specifikationer är dess verkningsgrad samt dess area, se Figur 4-10. Vid nedsmutsning av solcellen kan denna parameter vara försämrad i förhållande till verkningsgraden som fås av tillverkaren.

```
//*****SUN CELL*****

#define SUNENERGY_EFFICENCY 0.2 → → → → //[%/100]
#define SUNENERGY_AREA 4 → → → → // [m2]
```

Figur 4-10, Bild över hur solcellens specifikationer ställs in i *konfigure.h*

Hastighetsbegränsningar

Hastighetsbegränsning för optimering ställs in under *speed limits*, se Figur 4-11.

Inom dessa hastighetsbegränsningar optimeras programmet, se optimering senare i rapporten. Det gör inte nödvändigtvis att en optimerad rutt inte rekommenderar en hastighet utanför dessa begränsningar vid vissa segment av sträckan, se 5.6 Optimering av strategier.

Skulle det till exempel vara omöjligt för solbilen att köra över minsta hastigheten, *MINIMUM_SPEED* i en backe kommer programmet att simulera körning vid den lägre hastigheten. Detta kommer inte räknas som en förlust.

Högsta hastighet kommer också optimeras utifrån *MAXIMUM_SPEED*. Men skulle det vara fördelaktigt för bilen att köra snabbare i t.ex. en nedförsbacke istället för att använda den generativa bromsen kommer programmet att tillåta detta.

Hastighetsgränserna används i vid optimering. Gränserna används där för att formulera villkor för vilket körsätt bilen ska köra. Det kan till exempel vara så att bilen ska använda maximal effekt från motorn under *MINIMUM_SPEED* och rulla fram utan att konsumera energi från motorn vid *MAXIMUM_SPEED*.

```
//*****
//***                               Speed limits                               ***
//*****
#define MINIMUM_SPEED 20 → → → → → → → → // [km/h]
#define MAXIMUM_SPEED 120 → → → → → → → → // [km/h]
```

Figur 4-11, Bild över hur hastighetsbegränsningar ställs in i *configure.h*

Solenergi och väder

Det sista som användaren kan ställa in i programmet är parametrar associerade med väder och solenergi, se Figur 4-12.

SUN, *CLOUDS* och *RAIN* ställs in enligt formel (2.14) och är en ratio för hur mycket solenergi av snitt-solenergin som träffar markytan vid olika väder. Denna ratio kan se olika ut för olika klimat och är därför konfigurerbar.

```
//*****
//***                               Weather and sun                               ***
//*****
#define SUN 1.10 → → → → → → → → // [%/100]
#define CLOUDS 0.67 → → → → → → → → // [%/100]
#define RAIN 0.22 → → → → → → → → // [%/100]

#define MAX_DISTANCE_FOR_WEATHER_DATA 3000000 → // [m]

#define NUMBER_OF_HOURS_WHIT_WEATHER_DATA 8 → // [h] maximum 16
```

Figur 4-12, Bild över hur väder data används i *configure.h*

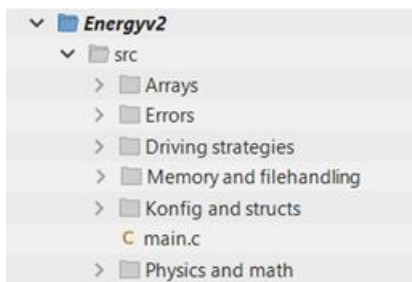
Väderdata anges för en GPS-koordinat utifrån vilken väderdata som är närmast. Om lite data finns att tillgå kan denna punkt vara långt ifrån den aktuella GPS-koordinaten. Väder tenderar att vara mycket varierande både beroende på position samt tidpunkt på dygnet. Därav kan det vara önskvärt att begränsa användning av väderdata utifrån avstånd. Om ingen väderdata finns för en punkt använder programmet sig av snittvärdet för solenergi för det området. *MAX_DISTANCE_FOR_WEATHER_DATA* är en variabel som begränsar det maximala avståndet från en väderstation som en väderprognos fortfarande kan antas relevant.

NUMBER_OF_HOURS_WHIT_WEATHER_DATA är en parameter som beskriver antalet timmar som det finns väderdata för, se avsnittet väderfil. Denna parameter kan maximalt vara

16 h. Detta beror på att en förare inte bör köra längre än 16 h i sträck och om bilen stannar måste programmet köras på nytt för att väderprognoser ska vara aktuella.

5 FUNKTIONSBESKRIVNING PROGRAM

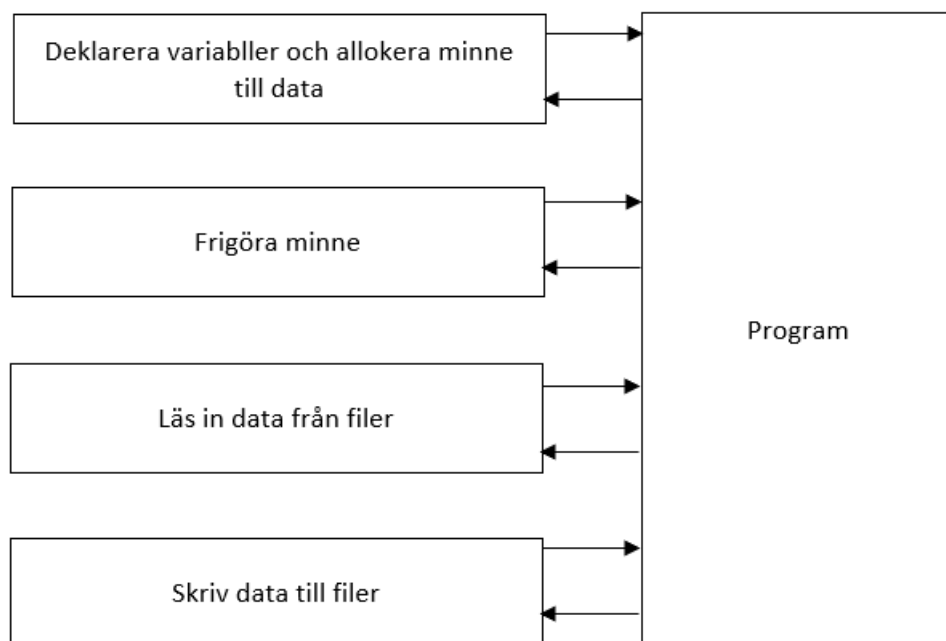
Programmet består av 68 funktioner uppdelade i 11 funktionsbibliotek i 6 olika mappar samt två extra headers med typdeklARATIONER för hela projektet. Mapparna delar upp projektet i mindre delar för att de ska vara enklare att hitta funktioner som söks, se Figur 5-1. Programmet simulerar sex olika körstrategier. Varje körstrategi optimeras. Varje optimerad körstrategi jämförs sedan med de andra för att se vilken optimerad körstrategi som är mest optimal. Programmet returnerar sedan resultatet via en textfil.



Figur 5-1, bild över projektets mappar

5.1 Hjälpfunktioner

För att programmet ska kunna räkna ut vilken hastighet bilen bör köra med behöver programmet tillgång till data. För att programmet ska kunna få tillgång till data behövs hjälpfunktioner som kan utföra enkla matematiska operationer, allokerar minne, frigöra minne samt skriva och läsa från filer, se Figur 5-2. Dessa funktioner anropas på flertalet ställen över hela programmet. Nedan beskrivs dessa funktioners mappar, bibliotek och funktion.



Figur 5-2, Schematiskbild över användandet av hjälpfunktioner

5.1.1 Konfiguration och strukturer

Det finns en mapp inom projektet som kallas *Konfig and structs* i denna mapp finns två stycken headerfiler. Den första är *Konfigurate.h* som skall användas av användaren, vilket beskrivs i avsnitt *konfigurate.h*.

Den andra headerfilen denna mapp innehåller är *allstructs.h*. Detta är en headerfil som deklarerar alla strukturer som används av stora delar av programmet. Alla konstanter som används i flera delar av programmet definieras även här.

Flera av strukturerna som är deklarerade i *allstructs.h* är strukturer som innehåller ett eller flera fält tillsammans med antalet element fälten innehåller, se Figur 5-3. Dessa strukturer kan sedan användas för enklare allokering av minne samt enklare läsning av fälten. Eftersom programmet inte i förväg vet hur många GPS-koordinater som programmet ska behandla är detta en metod för att minimera minnet som behöver allokeras.

```
typedef struct GPS_header_struct{
    →GPS_struct *GPS;
    →int nr_of_points;
}GPS_header_struct;
```

Figur 5-3, Bild på en struktur med ett fält av GPS-strukturer tillsammans med antalet element.

En annan typ av struktur som används av programmet är flera olika variabler som ofta används tillsammans och därför är fördelaktigt att ha i samma struktur, se Figur 5-4. Exempel på en sådan struktur är GPS-struktur som med fördel kan länka ihop longitud och latitud.

```
typedef struct GPS_struct{
    →float longitude;
    →float latitude;
}GPS_struct;
```

Figur 5-4, Bild på en GPS-struktur

5.1.2 Fält

En av mapparna i projektet kallas *Arrays* denna mapp har ett bibliotek med samma namn. Detta bibliotek innehåller två funktioner som kan användas för att sortera fält. Dessa behövs för att programmet ska kunna använda binärsökning.

5.1.3 Minne och filhantering

Memory and filehandling är en mapp som innehåller tre bibliotek *memory*, *filereading* och *filewriting*.

Biblioteket *memory* innehåller 12 funktioner. Tio av dessa funktioner används för allokering och frigörande av minne. Dessa funktioner används för strukturerna i programmet som innehåller fält. För att minnet till dessa enklare ska kunna allokeras och frigöras. De sista två funktionerna används för väderstatistik. Väderstatistiken för en punkt och tidpunkt anges som nämnt av två bitar. Detta gör att två konverteringsfunktioner behövs för att läsa och skriva till en variabel av typen long där bitarna sparas.

Biblioteket *filereading* innehåller 13 funktioner. Två av dessa funktioner används för att dels räkna ut hur många punkter med information en fil innehåller samt en funktion som undersöker om en fil är tom. De andra 11 funktionerna används för att kunna läsa in olika datatyper från en fil. Dessa funktioner allokerar behövt minne, läser från filen samt undersöker om filformatet stämmer.

Biblioteket *filewriting* innehåller tre funktioner som alla kan skriva ut olika strukturer och variabler till en fil.

5.1.4 Felmeddelande

Mappen *errors* innehåller en funktion. Denna funktion skriver ut felmeddelande till användaren då det finns något fel på uppladdade filer.

5.1.5 Fysik och Matematik

Mappen *Physics and Math* innehåller tre bibliotek: *Distance_calc*, *Energy_calc* och *Mathematics*.

5.1.6 Matematik

Mathematics innehåller en funktion som kan räkna ut hypotenusan av en triangel samt en funktion som räknar om grader till radianer.

5.2 Algoritmer för att simulera energikonsumtion

När de behövda hjälpfunktionerna är på plats kan programmet använda sig av dessa för att räkna ut energikonsumtionen samt önskad hastighet.

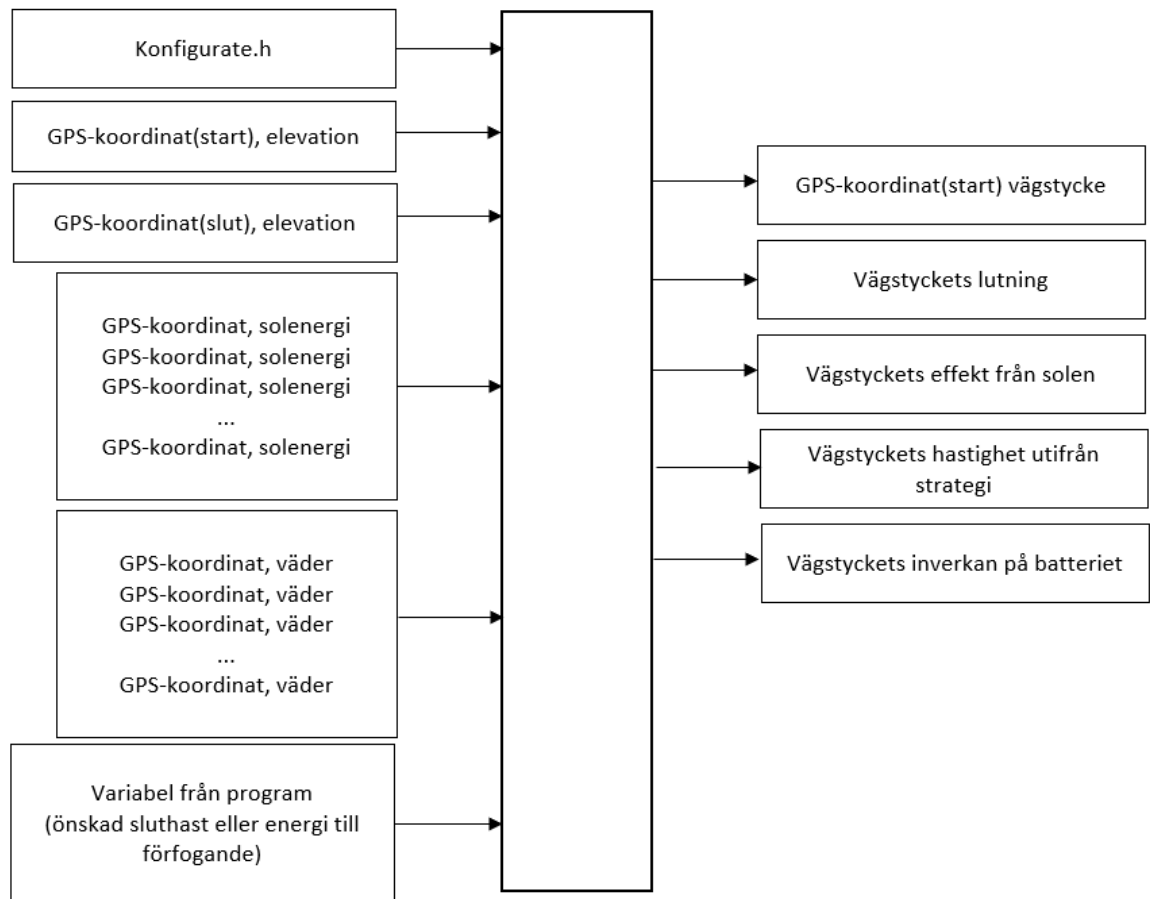
Programmets övergripande struktur kan beskrivas som följande: Programmet jämför olika körstrategier för att finna den bästa strategin. Olika strategier fungerar olika bra beroende på hur väl de är optimerade, därför optimeras varje strategi för sig genom att simulera vägbanan med olika villkor. För att simulera rutten måste det göras beräkningar på varje vägstycke. Alla vägstycken kan sedan adderas med varandra för att skapa en simulerad rutt, se Figur 5-5.



Figur 5-5, Bild på programmets övergripande struktur

5.3 Simulera vägstycke

Hela rutten delas upp i mindre delar och uträkningar görs på varje segment av vägen. Programmet gör detta utifrån de formler som beskrevs i avsnitt 2.2. När ett vägstycke ska simuleras behövs flera olika variabler och parametrar, se Figur 5-6. Denna data behandlas sedan av ett antal funktioner som finns i biblioteken *energy_calc* och *distance_calc*.



Figur 5-6, Schematisk bild över insignaler och utsignaler då programmet simulerar ett vägstycke. Den tomma rutan symboliserar programmet som använder sig av flera olika funktioner för att uppnå sitt mål.

5.3.1 Avstånd

Distance_calc innehåller 8 funktioner för att räkna ut avstånd samt konvertera hastigheten mellan m/s och km/h, se Figur 5-7.

```
//*****funktioner*****  
  
//*****distance*****  
int closest_GPS(GPS_struct thepoint, GPS_struct closepoint1, GPS_struct closepoint2);  
float distance(GPS_struct point1, GPS_struct point2);  
int find_closest_point(GPS_struct *points_with_data, int nr_of_points, GPS_struct point);  
double distance_laws_of_cousine(GPS_struct point1, GPS_struct point2);  
float delta_elevation(float elevation, float last_elevation);  
float calculate_slope(float elevation, float distance_between_points);  
  
//*****svt*****  
float m_sec(float kmperh);  
float km_h(float meterspersec);
```

Figur 5-7, funktionsdeklarationen i "Distance_calc.h"

Funktionen *closest_GPS* tar två GPS-koordinater samt en ensam GPS-koordinat sedan returnerar funktionen vilken av de två som den ensamma GPS-koordinaten var närmast.

Funktionen *find_closest_point* tar ett fält med GPS-koordinater och en ensam GPS-koordinat och returnerar indexet för den närmaste GPS-koordinaten i fältet till den ensamma GPS-koordinaten.

Dessa två funktioner är användbara för att t.ex. hitta den närmsta väderprognosen för ett vägstycke.

5.3.2 Energi

Biblioteket *Energy_calc* innehåller 12 funktioner som underlättar uträknandet av energin som används av bilen, se Figur 5-8.

```
float energy_from_sun(GPS_struct GPS, Data_and_gps_struct *sun_energy_in);  
float energy_from_sun_with_weather(GPS_struct GPS, float sun_in,  
    —>Weather_and_gps_struct* weather_points, race_struct *race);  
double delta_battery_set_point(race_struct *race);  
float energy_to_move_forward(distance_struct *point);  
float speed_from_energy(distance_struct *point, float energy);  
float energy_to_keep_konstant_speed(distance_struct *point);  
float speed_from_motor_energy(distance_struct *point, float energy);  
float speed_from_max_effekt(distance_struct *point);  
float speed_from_slope(distance_struct *point);  
float energy_to_accelrate_to_speed(distance_struct *point, float speed);  
float time_taken(distance_struct *point);  
char what_time_will_it_be(double sec_from_start);
```

Figur 5-8, Bild på funktionsdeklarationen i *Energy_calc.h*

Energi från solen

När bilen befinner sig vid en bestämd koordinat kommer den att träffas av en viss energi från solen. Från filen *solenergi* finns statistik över flera koordinater nära vägen. Denna statistik är mest relevant till de vägpunkter som är närmast. Funktionen *energy_from_sun* räknar ut vilken av de tillgängliga solenergipunkterna som är närmast en GPS-koordinat på vägen. Funktionen räknar också ut hur mycket av denna energi som är batteriet tillgodo genom att räkna med de förluster som uppstår mellan solcell och batteri.

Vädrets inverkan

När bilen kör över rutten kommer olika väder råda vid olika tidpunkter. Funktionen *energy_from_sun_with_weather* räknar ut den mest aktuella väderprognosen för en GPS-koordinat och returnerar solenergin för punkten med ett väderfilter. En väderprognos är beroende av två saker: dels hur nära i tiden prognosen är samt hur nära prognosens ursprung bilen befinner sig. Därför tar funktionen dessa två delar i beaktande. Om bilen befinner sig långt från närmsta prognos ursprung kommer inget väderfilter räknas med. Väderfilter refererar till ration som nämns i formel (2.14). Energin som träffar markytan vid t.ex. moln delat med snitt-energin som träffar markytan.

Funktionen *what_time_will_it_be* returnerar vilken timme från start som den aktuella GPS-koordinaten kommer nås. Detta används för att funktionen *energy_from_sun_with_weather* ska veta vilken kronologisk prognos som ska användas.

Batteriets laddningsnivå

Det är essentiellt att batteriet i bilen inte tar slut innan rutten är avslutad. Därför krävs det ett referensvärde för om batteriet är högt eller lågt. Om batteriet ska vara laddat över hela sträckan är det önskvärt att batteriets laddning är någorlunda proportionellt mot avståndet som är kvar. Det innebär att om batteriet är 100% laddat är 100% av avståndet kvar samt om batteriet är 0% laddat är det 0% av rutten kvar. Detta kan användas för att skapa ett referensvärde för om batteriet är högt eller lågt. Funktionen *delta_battery_setpoint* jämför laddningsprocenten hos batteriet med procent-rutt. Funktionen returnerar sedan ett värde. Om detta värdet är under noll har batteriet en låg laddningsnivå, om batteriet är över värdet kan det ses som en hög laddningsnivå.

Energi till bilens rörelse

Funktionen *energy_to_move_forward* räknar ut hur mycket energi som krävs för att köra bilen framåt över en viss sträcka där start och sluthastighet är bestämt. Funktionen använder sig av formel (2.7) som återger den totala energin som förbrukas av bilen då den kör över en bestämd sträcka med bestämd slut- och start-hastighet.

Hastighet från energi

Funktionerna *speed_from_energy*, och *speed_from_max_effekt* fungerar utifrån samma princip, se Figur 5-9, funktionen *speed_from_energy*. Båda funktionerna räknar ut vilken sluthastighet som maximalt kan hållas utifrån en förutbestämd energi.

Funktionerna börjar med att räkna ut energiförbrukningen om sluthastigheten är 0 och jämför förbrukningen med givet värde. Om den förbrukade energin är mindre än det som finns till förfogande höjs hastigheten. Detta görs med hjälp av 10-potenser för att få ett flyttals noggrannhet. Först adderas tio till hastigheten Om hastigheten är för låg adderas tio igen. Om

hastigheten blir för hög subtraheras tio och en lägre tio-potens adderas, i det här fallet ett. Programmet fortsätter på detta sätt tills den funnit en hastighet som ger önskad energikonsumtion. När funktionen sedan hittat sluthastigheten som ger given förbrukning returneras den.

```
float speed_from_energy(distance_struct *point, float energy){
    point->speed2=0;
    float dec[]={10,1,0.1,0.001,0.0001,0.00001,0.000001};
    int i=0;
    for(;i<6;i++){
        while(1){
            if((energy_to_move_forward(point)) < energy){
                point->speed2+=dec[i];
            }else{
                point->speed2-=dec[i];
                break;
            }
        }
    }
    if(point->speed2<0)
        point->speed2=0;
    return point->speed2;
}
```

Figur 5-9, funktionen *speed_from_energy*

Energi

Energy_to_keep_konstant_speed och *energy_to_accelrate_to_speed* är två funktioner som räknar ut hur mycket energi som förbrukas av batteriet om bilens start och sluthastighet är kända. Dessa funktioner lägger också till allmänna förluster från motorn och batteri, till skillnad från funktionen *energy_to_move_forward*.

Motoreffekt

Funktionen *speed_from_motor_energy* räknar ut maximala hastigheten för bilen i det fall att energin kommer från motorn. Eftersom motorn har förluster måste dessa läggas på. Dessutom kan motorn enbart leverera sin maxeffekt vilket begränsar hastigheten som kan fås av motorn.

Elevation

Om bilens motor inte tillför någon effekt och bilen är placerad i en nerförsbacke kommer denne att börja rulla. Funktionen *speed_from_slope* returnerar sluthastigheten vid dessa förhållanden.

5.4 Simulera hela rutten

Under mappen *driving strategies* finns det tre bibliotek: *simulate_rut*, *driving_strategy* och *driving_strategy_opti*. Dessa bibliotek använder sig av de funktioner som nämnts ovan.

```
char calculate_speed_and_energy_from_max_min_speed_and_strategy(char driving_strategy, result_t *result);
void calculate_energy_and_save_point_value_to_file(Data_and_gps_struct point, Data_and_gps_struct *data_and_gps_struct,
Weather_and_gps_struct *weather, race_struct *race, FILE *out);
float calculate_parameters_for_distance_between_points_and_return_battery_level(GPS_struct GPS_struct, result_t *result);
```

Figur 5-10, bild från headerfilen för biblioteket *simulate_rutt*

I bibliotek *simulate_rut* finns tre funktioner som tillsammans arbetar för att simulera bilens resa över vägen, se Figur 5-10, bild från headerfilen för biblioteket *simulate_rutt*. Dessa funktioner gör flera saker. Funktionerna i detta bibliotek måste ha all information som behövs för att simulera körvägen. Det innebär att dessa funktioner behöver ha en maximal och minimal körhastighet samt en strategi att köra efter.

Funktionen *calculate_driving_strategy* fungerar som en överordnande funktion. Denna funktion hanterar all data som behövs för att simulera sträckan. Den läser in väderfilen samt solenergifilen till minnet och allokerar variabler som kan användas för beräkningar. Funktionen läser sedan in en punkt i taget från elevationsfilen och anropar *calculate_energy_and_save_point_to_value_file* för varje punkt i ordning.

Funktionen *calculate_energy_and_save_point_to_value_* tar två stycken GPS-punkter från ruten samt informationen från filerna. Funktionen räknar sedan ut avståndet mellan punkterna, sluthastighet, elevation, solenergi in och batterinivå. Detta gör funktionen genom att tillkalla funktioner i andra bibliotek som nämnts tidigare samt tillkalla funktionen *calculate_parameters_for_distance_between_points_and_return_battery_level*. Dessa variabler skrivs sedan till en fil.

När *calculate_parameters_for_distance_between_points_and_return_battery_level* anropas har den för uppgift att räkna ut önskad sluthastighet. Funktionen har information om hur varje punkt ska optimeras, utifrån minimal och maximal hastighet samt strategi.

5.5 Körstrategier

I denna rapport avser en körstrategi, ett sätt att köra som är olika sätt vid olika tillfällen till exempel rulla om hastigheten är hög och använd maximal effekt från motorn om hastigheten är låg. I denna rapport avser ett körsätt ett enda sätt att framföra bilen, till exempel rulla. I biblioteket *driving_strategy* finns funktioner för att kunna simulera sex olika körstrategier, se Figur 5-11, bild på biblioteket *driving_strategies*. Dessa strategier har hjälp av tre körsätt som kan tillämpas av de övriga funktionerna. Det finns även en funktion som uppdaterar batteriets laddning samt tid från start utifrån strategierna.

```

void strategy_keep_konstant_low_speed_deaccelrate_with_generative_brakes(distance_struct *point, race_struct *race, float sun_in);
void strategy_motor_power_only_under_minimum_speed(distance_struct *point, race_struct *race, float sun_in);
void strategy_dynamic_drive_two(distance_struct *point, race_struct *race, float sun_in);
void strategy_power_from_delta_energy(distance_struct *point, race_struct *race, float sun_in);
void strategy_speed_high_if_high_battery_speed_low_if_low_battery(distance_struct *point, race_struct *race, float sun_in);
void strategy_dynamic_drive_one(distance_struct *point, race_struct *race, float sun_in);

void energy_or_max_power(distance_struct *point, race_struct *race, float sun_in);
void accelrate_to_speed_or_max_power(distance_struct *point, race_struct *race, float sun_in);
void update_battery_sun_and_time(distance_struct *point, race_struct *race, float sun_in);
void roll(distance_struct *point, race_struct *race, float sun_in);

```

Figur 5-11, bild på biblioteket *driving_strategies*

5.5.1 Körsätt

En bil kan rulla. Detta innebär att bilen inte har igång sin motor och enbart rullar fram. Funktionen *roll* simulerar denna typ av körning. Funktionen räknar ut vad sluthastigheten för bilen blir samt hur mycket batteriet laddas under tiden.

Ett annat körsätt är att försöka köra enligt en given hastighet. Om den givna hastigheten inte går att uppnå är det önskvärt att fortsätta att accelerera efter den givna sträckan. Funktionen *accelrate_to_speed_or_max_power* får bilen att köra i den givna sluthastigheten eller maximalt köra med motorns maxeffekt.

Det sista definierade körsättet i biblioteket är att köra utifrån en förutbestämd förbrukning. Funktionen *energy_or_max_power* simulerar detta.

5.5.2 Strategier

I biblioteket *driving_strategy* finns som nämnt sex stycken framtagna strategier. Nedan följer beskrivningar av dessa.

Konstant hastighet

Den första strategin i detta bibliotek är *strategy_keep_konstant_speed_and_deaccelrate_with_generative_brakes*. Denna strategi håller en konstant hastighet så långt som är möjligt. Påverkas bilen av t.ex. av en nedförsbacke kommer den extra energin sparas i batteriet genom den generativa bromsen. Detta sker i det fall att den generativa bromsen är på i konfigurate.h. I det fall att den generativa bromsen är avstängd kommer inte denna energi att sparas i batteriet. Om bilen påverkas av en uppförsbacke kommer bilen försöka accelerera med hjälp av motorns maxeffekt.

Motoreffekt vid minimal hastighet

Strategin, Motoreffekt vid minimal hastighet gör att bilen antingen accelererar till minsta tillåtna hastighet eller rullar fram. Detta simulerar en typ av tvålägesreglering.

Dynamisk reglering

Den dynamiska regleringen använder sig av tre olika villkor. Om hastigheten är låg kommer de dynamiska strategierna att accelerera till minihastigheten. Om hastigheten är hög kommer de att rulla. Om hastigheten ligger mellan ytterligheterna kör de båda dynamiska strategierna olika.

Strategy_dynamic_drive_one kommer att hålla en konstant hastighet om hastigheten är inom det önskade spannet.

Strategy_dynamic_drive_two kommer att hålla en konstant energi utifrån vad batteriet laddas med från solcellen.

Hastighet baserat på energin till förfogande

Strategierna *strategy_speed_high_if_high_battery_speed_low_if_low_battery* och *strategy_power_from_delta_energy* simulerar hastigheterna utifrån batteriets nivå. Den första funktionen kommer att accelerera till max hastighet om batteriet är hög och rulla om batteriet är lågt. Den andra funktionen kommer att enbart använda den energin som bilen laddades med föregående vägsegment.

5.6 Optimering av strategier

I biblioteket *driving_strategy_opti* finns funktioner som kan optimera de ovan nämna strategierna. Funktionsbiblioteket innehåller två stycken funktioner, se Figur 5-12, funktiondeklarationerna tillhörande *driving_strategy_opti*. Optimering görs genom att varje körstrategi optimeras utifrån dess inparametrar. Funktionen *optimize_driving_strategy* har två olika sätt att optimera strategier, vilka beskrivs nedan.

```
void optimize_driving_strategy(char strategy, result_struct *best);  
void calculate_best_driving_strategy_and_save_in_file_SAVED(void);
```

Figur 5-12, funktiondeklarationerna tillhörande *driving_strategy_opti*

De funktioner som använder sig av maximal och minimal hastighet optimeras genom att sträckan simuleras för varje tänkbar kombination av max och min hastighet. Detta görs genom att först simuleras vägen för om både övre och undregränsen sätts till *MINIMUM_SPEED*. Sedan adderas *OPTI_RES* till båda gränserna och sedan simuleras rutten åter, se Figur 5-13.

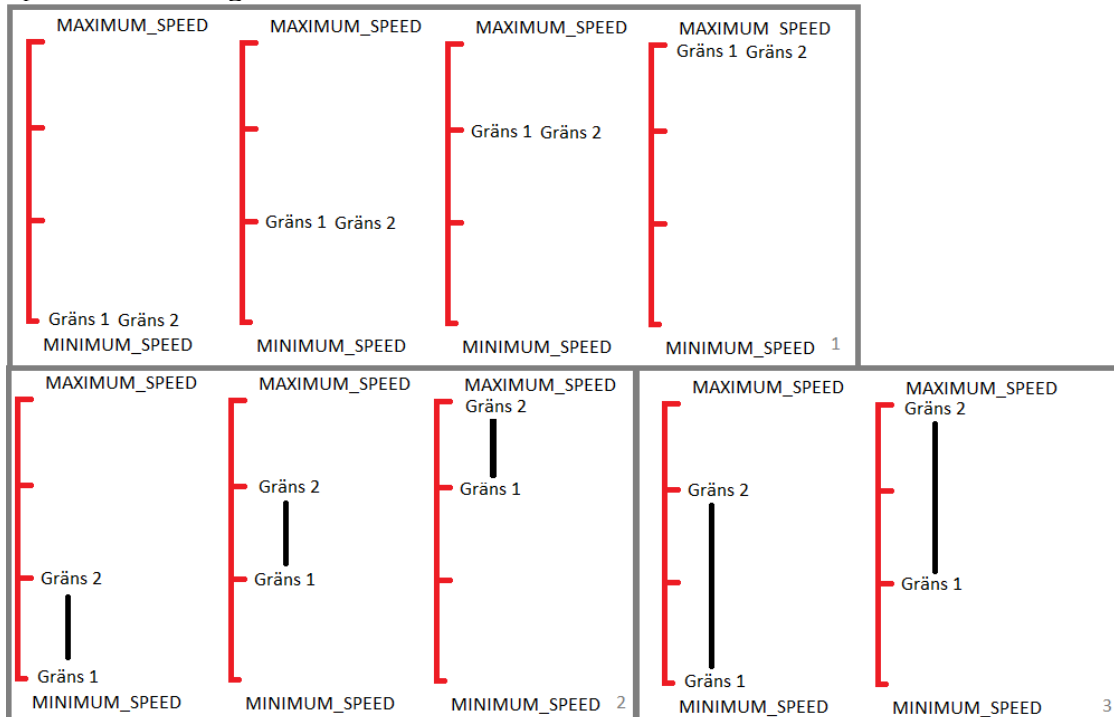
OPTI_RES adderas till båda gränserna tills det att den övre gränsen nått *MAXIMUM_SPEED*. Därefter sätts den lägre gränsen till *MINIMUM_SPEED* och den övre gränsen ökar sitt avstånd till den lägre gränsen med *RES_OPTI*. Detta görs tills alla olika kombinationer av gränserna har simulerats av programmet.

OPTI_RES är en intern variabel för biblioteket som bestämmer hur stort hastighetsskillnad det ska vara mellan de hastigheter som simuleras.

Om en strategi enbart har en hastighetsgräns kommer programmet att simulera rutten för alla hastigheter mellan *MINIMUM_SPEED* och *MAXIMUM_SPEED* med stegavstånd på *RES_OPTI*.

Funktionerna i detta bibliotek simulerar alla sex strategier och optimerar dessa för att hitta den bästa strategin för körväg av alla strategier och sparar till sist denna

optimerade strategi till resultatfilen.



Figur 5-13, schematisk bild över hur optimering av strategier med två hastighetsgränser sker.

5.7 Konfiguration och datainsamling

Under projektets gång har programmet testats med data om Bridgestone World solar challenge och solarteams tävlingsbil. Data som inte fanns att tillgå har approximerats. Nedan följer en beskrivning om datainsamlingsprocessen.

5.7.1 Solenergi

Statistiken som används för att approximera solenergin som träffar markytan har inhämtats hos Australian Bureau of Meteorology. Statistiken som finns tillgänglig hos institutet är given på följande sätt:

Det finns väderstationer över hela landet. Dessa inhämtar data över hur många MJ/m² som träffar markytan vid stationen över ett dygn. Det finns också ett snittvärde per månad för denna parameter. Detta snittvärde har inräknat alla år väderstationen mätt denna parameter.

Till programmet beskrivet i denna rapport ska solenergistatistiken vara angiven i effekt. Solbilracet går dessutom av stapeln i oktober månad. Utifrån detta införskaffades dels solenergi statistik över oktober samt dess soltimmar. Detta gjordes för 12 stationer i närheten av rutten.

Statistiken användes för att först få ut en snittenergi för varje väderstation. Väderstationernas GPS-koordinater samt snittenergi per sekund skrevs sedan in i solenergifilen som användes i detta projekt.

Utifrån solenergidatan gjordes också approximationer över hur mycket solenergi som träffar markytan vid olika väder. Detta gjordes genom att statistik över oktober månad. På institutet fanns även statistik över lägsta snittvärde för en dag samt högsta.

Det lägsta snittvärdet delat med totala snittvärdet räknades ut för alla stationer. Samma sak med det högsta snittvärdet. Vidare användes det lägsta värdet för alla stationer för att definiera en procentsats för hur mycket energi som släpps genom vid regn. Det högsta värdet användes för sol.

Mörka moln antogs vara ett värde någonstans emellan. Dessa värden jämfördes sedan med statistik som fanns tillgänglig för oktober 2018. Där även vädret för dagarna fanns tillgängligt. Ration multiplicerat med snittvärdet jämfördes med de verkliga värdena för regn, sol och moln. Vid jämförelsen gav approximation med ration en felmarginal på 6% över hela månaden. Approximation med enbart medelvärde gav en felmarginal på 13% över hela månaden, se Bilaga 2.

Båda sätten att approximera väder var i vissa fall undermåliga för en dag. Approximation med ratio gav som sämst ett fel på 221 % och näst sämst ett fel på 52 %. Denna felmarginal var störst då vädret representerades som halvklart. Detta tros bero på att halvklart väder kan ge stor spridning i hur mycket sol som träffar markytan. Beroende på om det är mycket eller lite moln. Denna strategi gav även större spridning då vädret var olika under dagen. Detta beror på att väderstatistiken som var tillgänglig enbart hade väderstatistik för två klockslag per dag. I snitt gav approximation med ratio ett fel på 1 %.

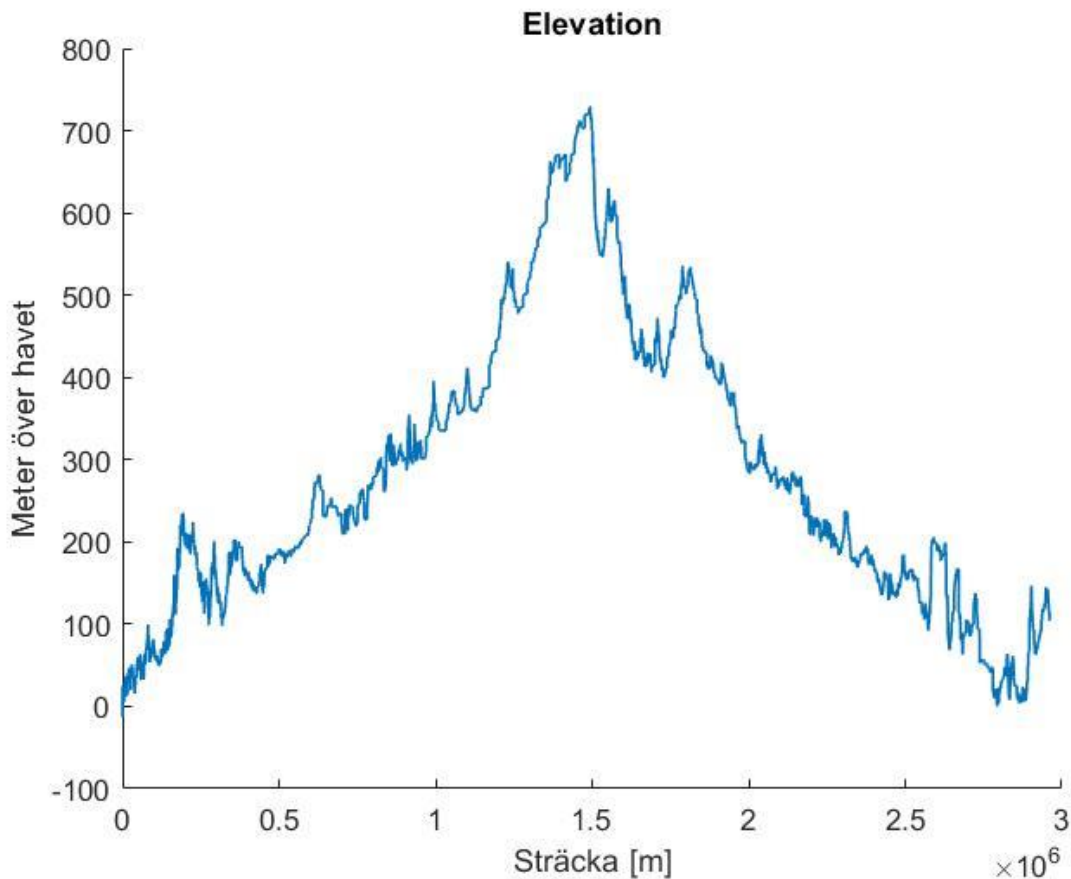
Medelvärde gav som sämst ett fel på 435% och näst sämst på 319%. Approximation med medelvärde gav ett fel på i snitt 44%. Som störst var felet vid regn. Detta tros bero på att de soliga dagarna är fler till antal i Oktober månad i Australien.

Approximation med ratio ses som ett bättre sätt att approximera väder än med medelvärde.

Under försöket fanns det enbart väderprognoser för klockan 6:00 och klockan 12:00. Under lopp och vid framtida användning av väderprognoser kommer väder timme för timme finnas tillgängligt. Detta gör att approximationens tillförlitlighet kan komma att ändras.

5.7.2 Elevation och rutt

Elevationsfilen ställdes in med data över rutten som inhämtades från Google earth, se Figur 5-14, elevation i förhållande till sträcka. Körvägen simulerades i Matlab och jämfördes med en graf över tävlingsvägen framtagen av en solbil som kört racet som tillhandahölls av solarteam.



Figur 5-14, elevation i förhållande till sträcka

5.7.3 Konfiguration

Nedan följer en beskrivning av hur parametrarna för programmet ställdes in vid simulering och optimering av loppet, se Tabell 5-1. Det följer även en förklaring hur dessa värden tagits fram.

Tabell 5-1, Tabell över inställda värden i *configure.h*

Parameternamn	Värde	Enhet	Källa
EARTH_RADIUS	6371000	m	
GRAVITATION	9.81	m/s ²	
FRICTION	10,37	N	Se rubrik om Friktion nedan.
MASS	300	kg	Solarteam
AIRRESISTANCE_KONSTANT	0.099		Approximation utifrån experiment av solarteam
MOTOR_MAX_POWER	5000	W	Motorspecifikation [13]
MOTOR_EFFICIENCY	0.95	%/100	Motorspecifikation [13]
REGENERATIVE_BRAKE_EFFICIENCY	0.64	%/100	Rapport om generativa bromsar [14]
GENERATIVE_BRAKE	1	SANT/FALSKT	Solarteam

Parameternamn	Värde	Enhet	Källa
OTHER_FUNCTIONS	0	W	Approximation från rapport [4]
BATTERY_START	1	%/100	Solarteam [15]
BATTERY_MAX_WORKING_LEVEL	0.95	%/100	Approximation från rapport [7]
BATTERY_MINI_WORKING_LEVEL	0.1	%/100	Approximation från rapport [7]
BATTERY_TOTAL_CAPACITY	1393600	Joule	Approximation från datablad [16]
BATTERY_EFFICENCY	0.95	%/100	Från bok [6]
BATTERY_CHARGE_MAX	2926	w	Datablad [16]
SUNENERGY_EFFICENCY	0.2	%/100	Solarteam
SUNENERGY_AREA	4	m ²	Solarteam
MINIMUM_SPEED	20	km/h	Se rubrik, optimering nedan
SUN	1.10	%/100	Approximation, se rubrik om solenergi ovan
CLOUDS	0.67	%/100	Approximation, se rubrik om solenergi ovan
RAIN	0.22	%/100	Approximation, se rubrik om solenergi ovan
MAX_DISTANCE_FOR_WEATHER_DATA	3000000	m	Inställt för att kunna simulera ett väder för hela sträckan

Friktion

Friktionen bör tas fram via experiment. Detta har inte varit möjligt då solbilen ännu inte är färdig. Det värde som använt i detta projekt har approximerats utifrån elbilars konsumtion av energi. En elbil konsumerar ca. 22% av sin energi till rullmotståndet och 45% av sin totala konsumtion till luftmotståndet då den kör på en motorväg [8]. Detta samband användes sedan för att approximera friktionen genom att räkna ut kraften som verkar på bilen av luftmotståndet vid 100 km/h och använda uttrycket ovan för att få en approximerad friktion. 100km/h användes då detta är en vanlig hastighet att köra i vid motorvägskörning. Denna siffra anses vara något hög. Då en konventionell elbil har bildäck och lager med större friktion jämfört med tävlingsbilen. Solbilen kommer att köra med motorcykeldäck och keramiklager.

Inför tävlingen behöver experiment göras för att beräkna den aktuella bilens friktion. Friktionsvärdet som har använts här är bra för att verifiera programmets funktion och visa möjliga simuleringsresultat. Men för att få korrekta simuleringsresultat för Solarteams bil krävs friktionsvärde för den bilen.

Optimering

MAXIMUM_SPEED och *MINIMUM_SPEED* är två parametrar som används för att optimera bilens hastighet. Dessa valdes för ytterligheterna. 120 km/h är den ungefärliga maxhastigheten för bilen, uträknat vid simulering av programmet. 120km/h är även den övre hastighetsgränsen för allmänna vägar i Australien. Därför är det en naturlig maxgräns för optimering. Simulering över 120km/h hade inte gett ett bättre resultat då bilen ej kan köra över denna hastighet. 20km/h valdes som lägsta hastighet för att det gav fler strategier med möjlighet att komma i mål, vilket visade sig under simulering.

6 OPTIMERING

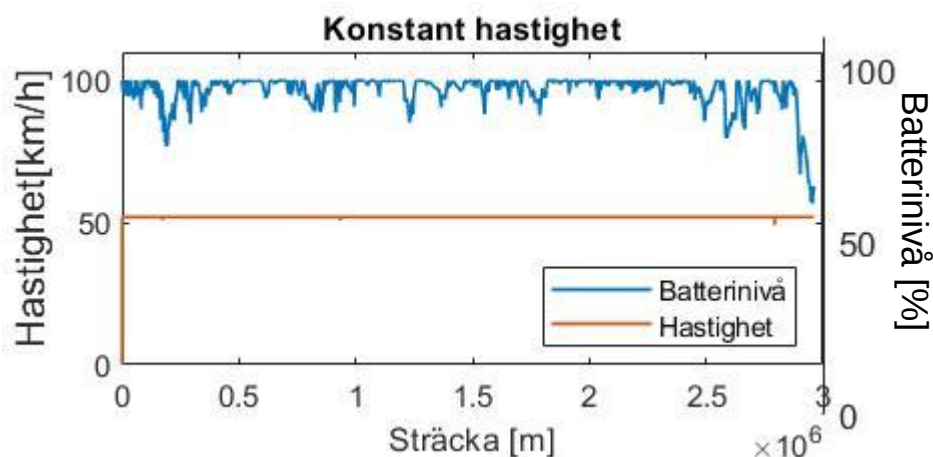
Projektet resulterade i ett program som kan simulera olika körstrategier för att hitta den bästa av dessa. Strategierna optimeras innan de jämförs med varandra. I kapitel Optimering visas grafer över den simulerade tävlingsruten i Australien vid olika villkor. Dessa grafer är gjorda i Matlab, se Bilaga 1, Matlab kod.

Avsnittet om optimering beskriver varje strategi var för sig. Alla simuleringar gjorda i detta avsnitt har gjorts med sol över hela sträckan. Simuleringen har gjorts för tävlingssträckan i Australien med de inställda värdena beskrivna i avsnitt 5.7 som konfiguration.

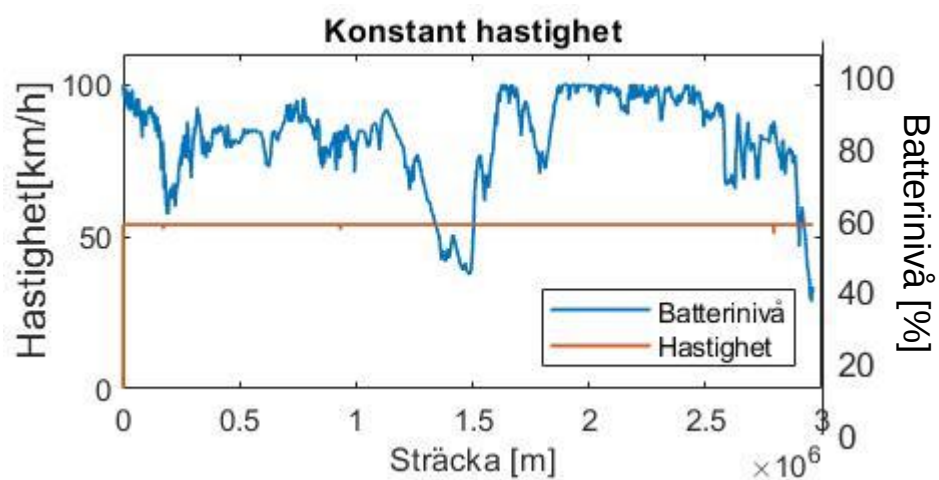
6.1 Strategi konstant hastighet

Strategin konstant hastighet har en hastighetsgräns som den optimeras mot. Nedan har några av de hastigheter som testas vid optimering simulerats i grafer. Den första grafen, Figur 6-1 som visas har simulerats utifrån 52 km/h. Denna konstanta hastighet gör att batteriet är mer eller mindre fullt under hela sträckan. Batteriet utnyttjas därmed föga.

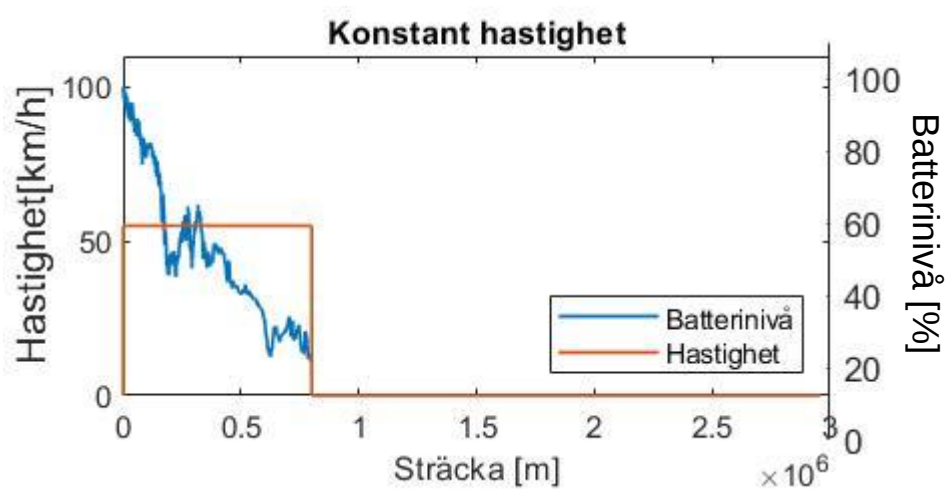
Om den konstanta hastigheten sätts till 54 km/h kommer batteriets kapacitet att utnyttjas bättre än om den konstanta hastigheten sätts till 52 km/h, se Figur 6-2. Detta gör att bilen kan komma i mål snabbare. Jämför sluttiden för de två som är 56,95 h respektive 54,84 h. Batteriet har dock inte använts till fullo om hastigheten sätts till 54 km/h. Batterinivån vid målgång är 40%. Höjs hastigheten till 55 km/h kommer dock batteriet ta slut efter 800618m, se Figur 6-3. Detta beror på att batteriet inte är tillräckligt laddat vid första uppförsbacken, se Figur 5-14 för höjdförhållanden under loppet.



Figur 6-1 Konstant hastighet: 52 km/h, sluttid: 56,95 h



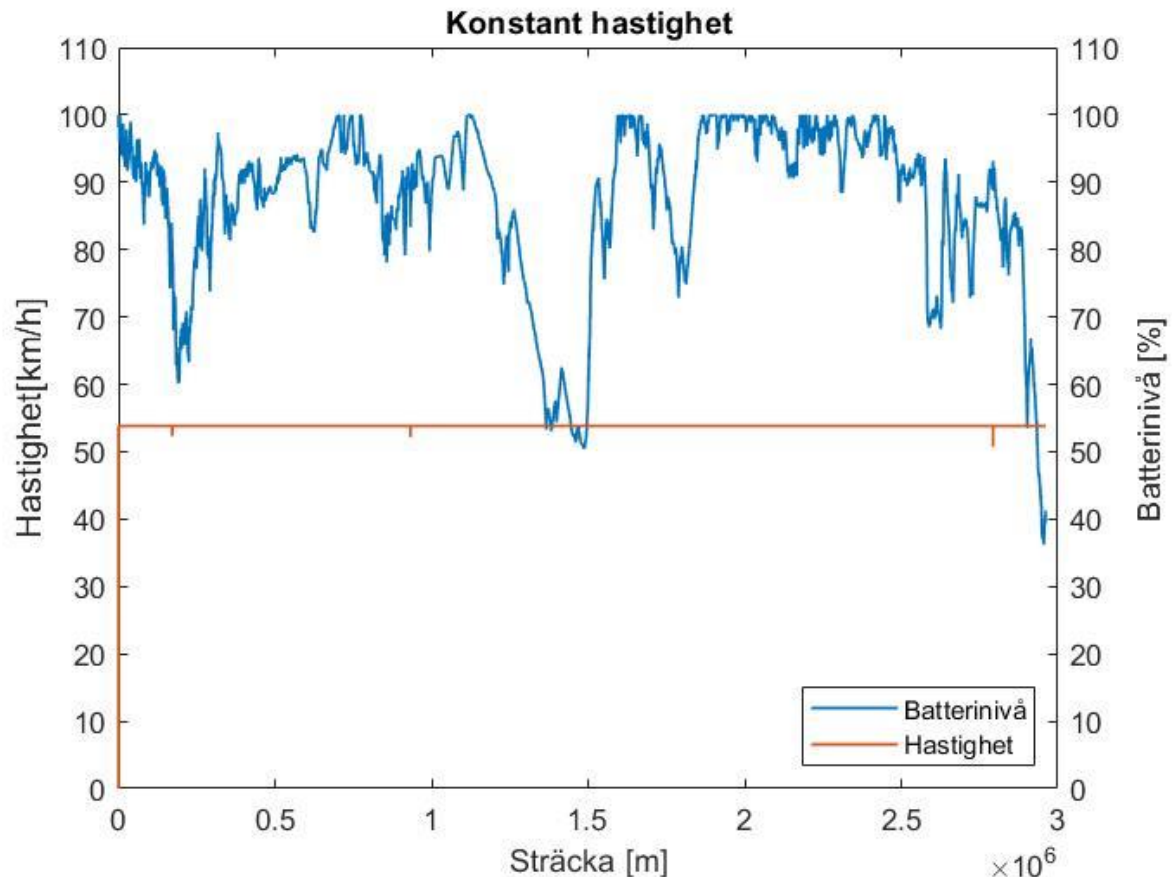
Figur 6-2.konstant hastighet 54 km/h, sluttid;54,84 h



Figur 6-3 Konstant hastighet 55 km/h, batteriet slut efter 800618m

6.1.1 Optimalt resultat för konstant hastighet

Optimalt resultat då bilen håller konstant hastighet är då bilen kör i en hastighet av 52,4 km/h, se Figur 6-4. Batteriet är då inte helt slut. Sluttiden är 56,51 h. Detta beror på att den sista uppförsbacke behöver en ganska hög batterinivå för att solbilen ska klara att köra förbi den. När hastigheten sänks en aning kommer det finnas mycket mer energi till batteriet. Eftersom batteriet är väldigt litet i förhållande till all den energi som förbrukas och ackumuleras under loppet kommer strategin för konstant hastighet inte gå att optimera så att batteriet precis tar slut i slutet av racet. Batteriet kommer helt enkelt antingen vara för laddat i slutet eller ta slut vid någon av uppförsbackarna.



Figur 6-4, Det bästa resultatet av strategin konstant hastighet.
Konstant hastighet: 54,39 km/h, sluttid:56,51 h

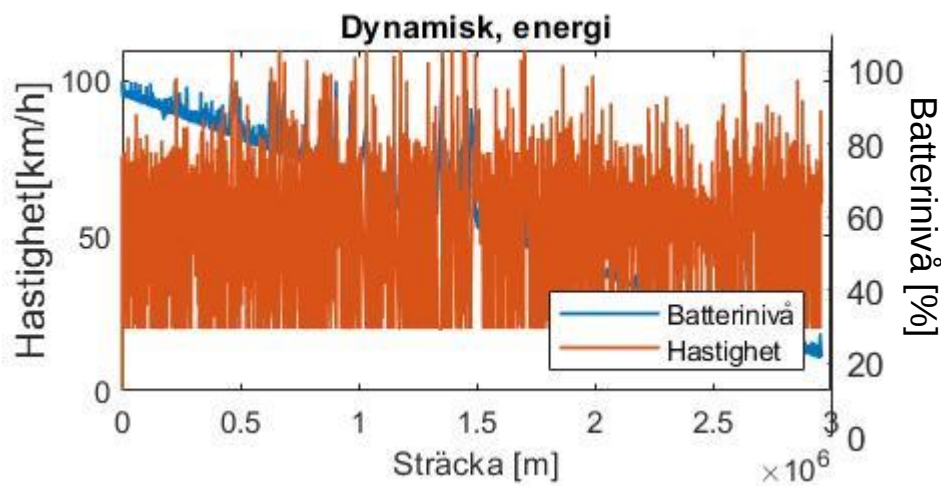
6.2 Strategi dynamisk energi

Resultatet för strategin dynamisk energi visas nedan. Denna strategi använder den energi som behövs för att hålla bilen över en undre hastighetsgräns. Om hastighet från elevation kan utnyttjas används denna för att accelerera bilen. Om hastigheten är över den övre gränsen kommer bilen att rulla. Om hastigheten är inom spannet för hastighet kommer energi förbrukas utifrån batterinivån. T.ex. om batteriet är 60% fullt och det är 40 % kvar av vägen kommer 20% av batteriet användas för att accelerera bilen.

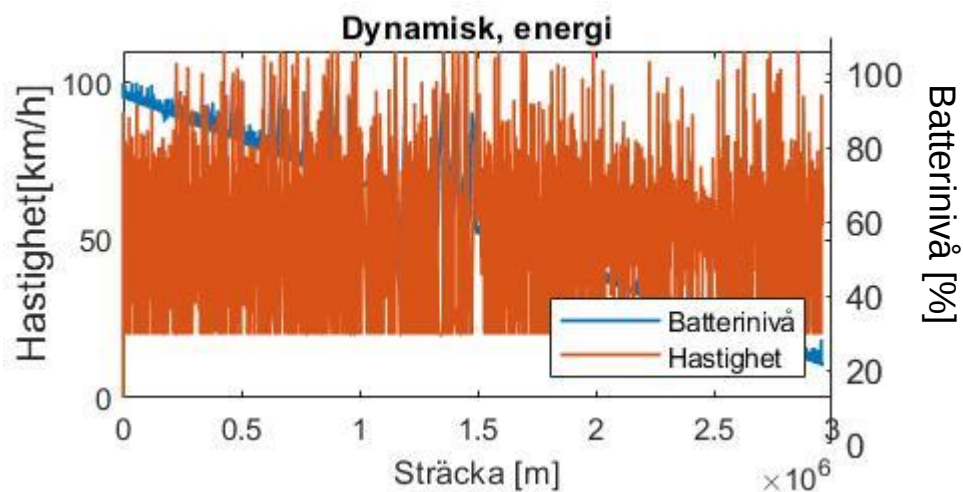
Denna strategi tar väl tillvara på hela batteriets kapacitet. Strategin byter ofta hastighet. När bilen kör snabbare kommer mer energi att gå åt för att framföra bilen. Men när bilen kör för en nedförsbacke är det ofta önskvärt att använda denna energi till att accelerera. Detta gör att bilen i slutändan kommer längre än om lägesenergin hade bromsats bort genom att bilen håller konstant hastighet.

Nedan visas några av de hastighetsbegränsningar som programmet simulerat. Den första visar det fall då hastighetsgränserna sätts till 20 respektive 100 km/h, se Figur 6-5. Denna körstrategi ger en sluttid på 61,99 h. Denna strategi tar längre tid än om hastighetsgränserna sätts till 20 respektive 60 km/h, se Figur 6-6. Detta beror på att en lägre övre gräns accelererar oftare med hjälp av elevationen medan en högre gräns kommer att använda energi från batteriet oftare för att accelerera. Det gör att den lägre gränsen kan använda batteriets kapacitet mer effektivt när det behövs.

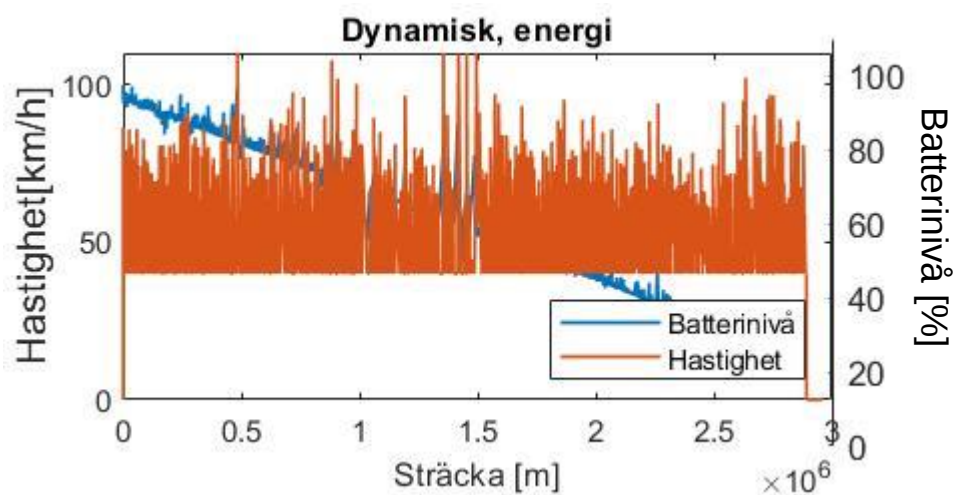
Då hastighetsgränserna sätts till 40 respektive 80 km/h kommer inte bilen klara av att gå i mål, se Figur 6-7. Detta beror på att 40 km/h är en för hög gräns för bilen att vara tvungen att köra över. Detta gör att batteriet tar slut i sista backen.



Figur 6-5, Dynamisk energi, hastighetsbegränsningar 20 respektive 100 km/h, sluttid: 61,99h



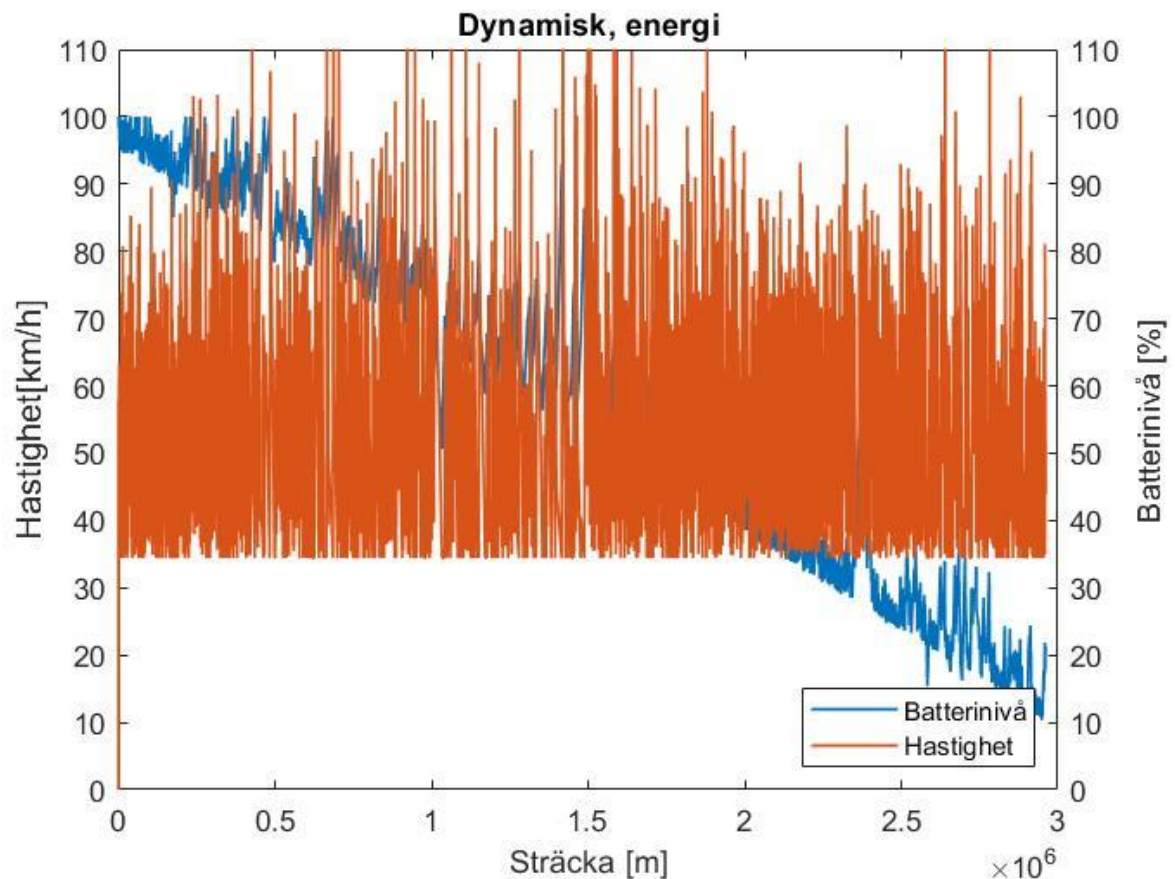
Figur 6-6, Dynamisk energi, hastighetsbegränsningar 20 respektive 60 km/h, sluttid: 60,62h



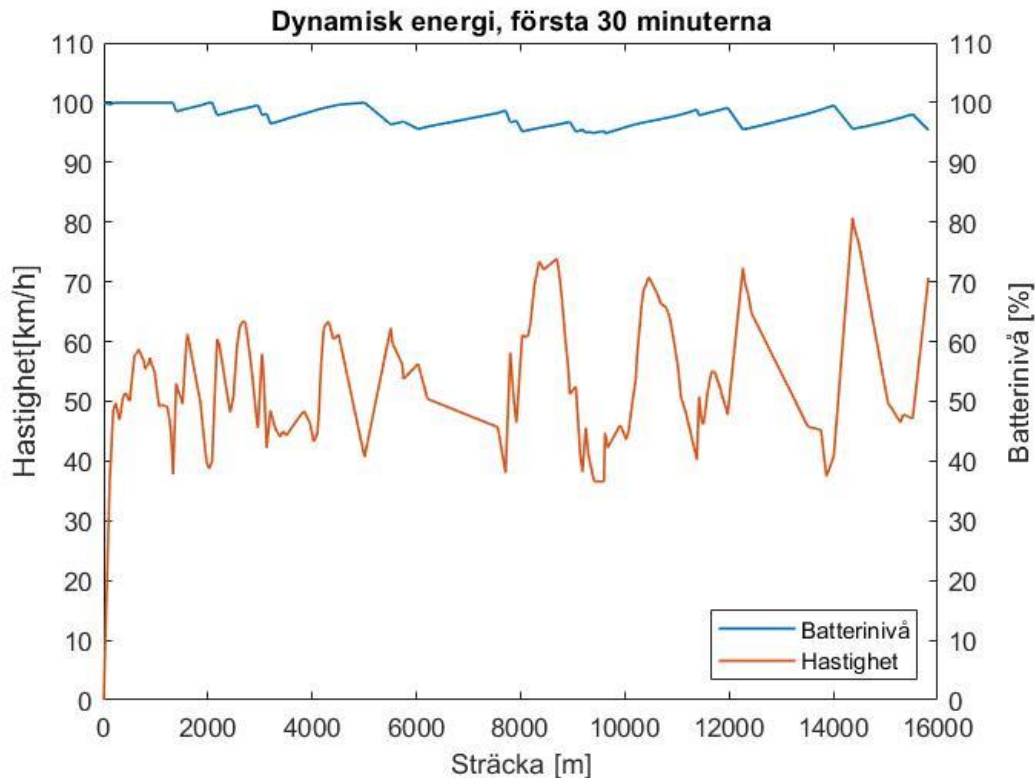
Figur 6-7, Dynamisk energi, hastighetsbegränsningar 40 respektive 80 km/h, Batteri slut efter 2892504 m

6.2.1 Optimalt resultat för strategi dynamisk energi

Denna strategi tar precis slut på batteriet vid målgång. Då batterinivån ej ska gå under 10% laddning, se avsnitt 2.2.3. Hastigheten fluktuerar mycket under rutten, se Figur 6-8. Hastigheten fluktuerar dock inte mer än att det är rimligt att hålla hastigheten. Om ett kortare spann simuleras, se Figur 6-9 på ungefär 30 min, ses tydligt att accelerationen och retardationen kan uppnås av en bil.



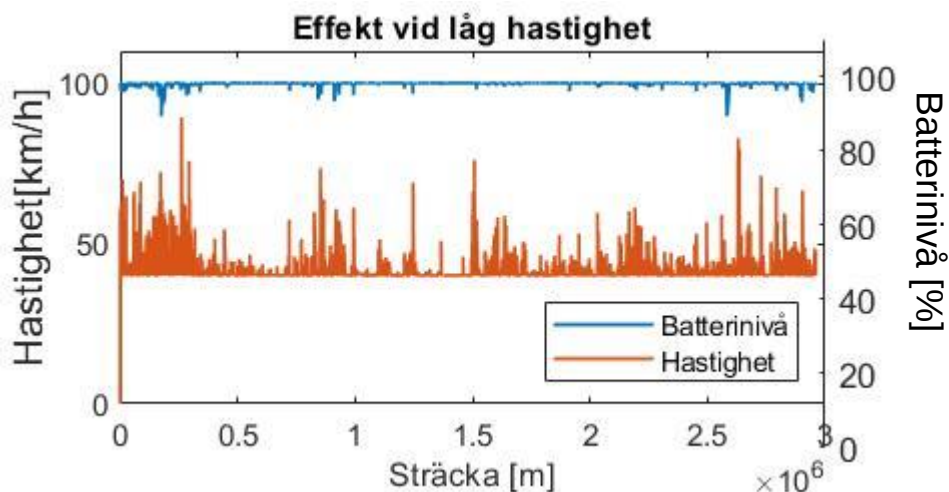
Figur 6-8 Bästa resultat för strategin dynamisk, energi. Hastighetsbegränsningarna 34.40 respektive 34.76 km/h. sluttid:58.32 h



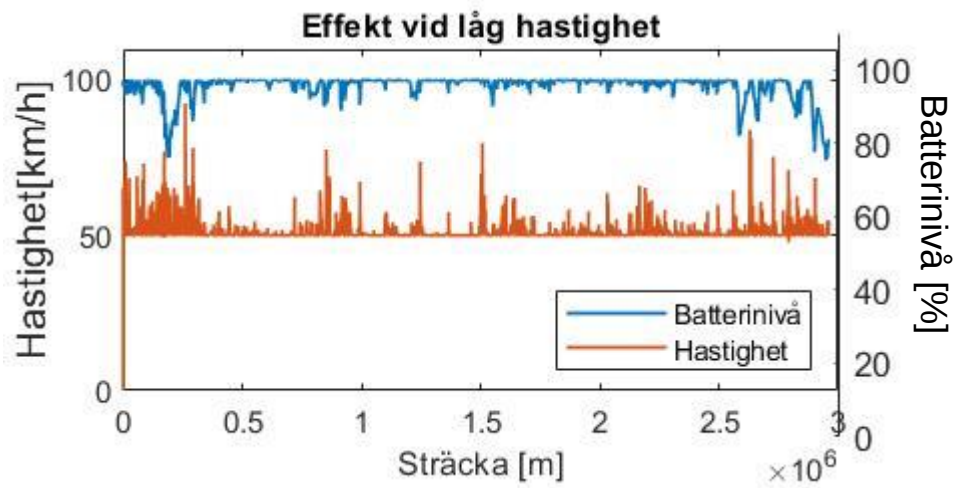
Figur 6-9, De första 30 minuterna av bästa resultatet av strategin dynamisk energi

6.3 Effekt vid låg hastighet

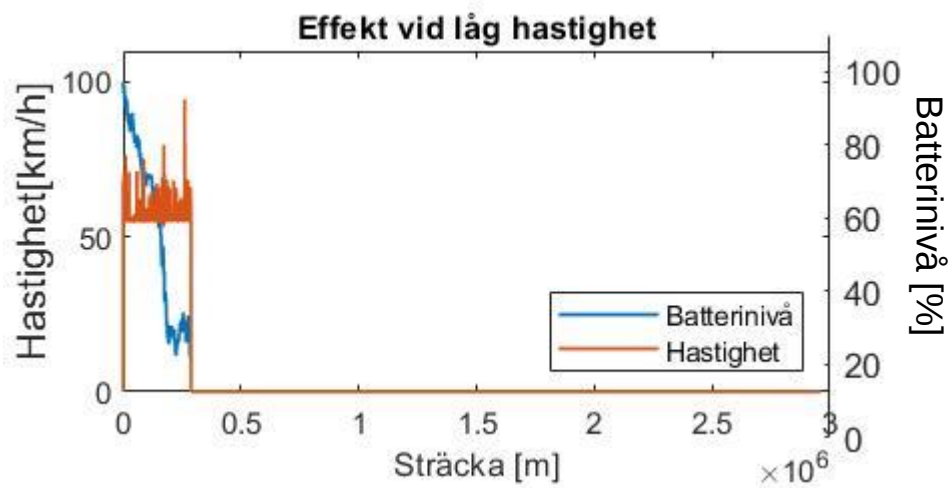
Effekt vid låg hastighet är en strategi som rullar då hastigheten är över en hastighetsgräns och ger energi till motorn endast om hastigheten är under denna gräns. Strategin optimeras utifrån denna parameter. Nedan visas exempel på några av de fall som simuleras vid optimering. När hastighetsgränsen är satt till 40 km/h kommer batteriet vara helt fullt vid målgång, se Figur 6-10. Detta gör att batteriet skulle kunna användas mer effektivt. När hastighetsgränsen är satt till 50 km/h kommer bilen komma snabbare i mål. Batteriet är dock 80 % fullt vid målgång, se Figur 6-11. Höjs hastighetsgränsen ytterligare till 55 km/h kommer batteriet ta slut i första uppförsbacken, Figur 6-12.



Figur 6-10, effekt vid låg hastighet, hastighetsgräns 40 km/h, sluttid: 72.29 h



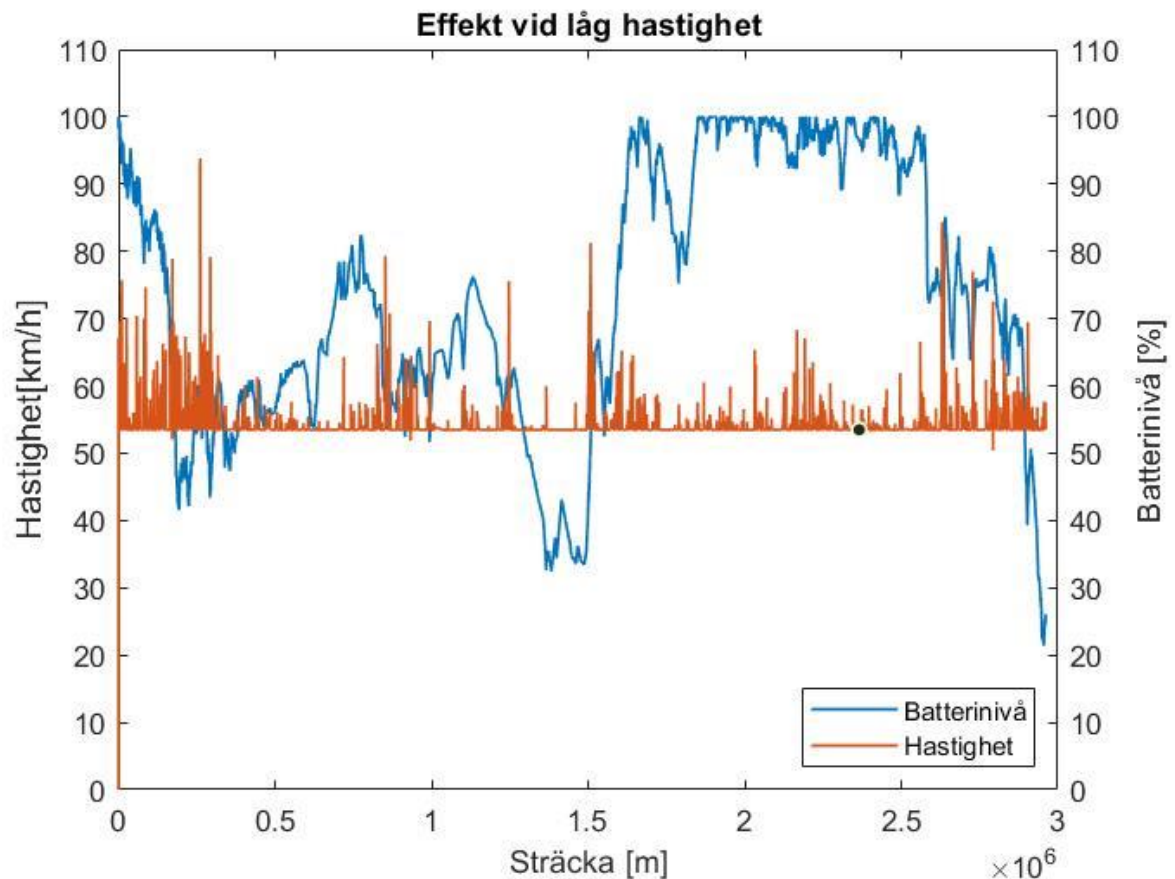
Figur 6-11, effekt vid låg hastighet, hastighetsgräns 50 km/h, sluttid: 58,58 h



Figur 6-12, effekt vid låg hastighet, hastighetsgräns 55 km/h, batteri slut efter 288428 m

6.3.1 Optimalt resultat för effekt vid låg hastighet

Det optimala resultatet för strategin effekt vid låg hastighet ges av en hastighetsbegränsning på 55,99 km/h. Detta gör att batteriet används mer optimalt än i exemplen ovan. Batteriet är nästan helt slut vid målgång, det är endast 25 % av batteriet kvar. Sluttiden är också låg 54,90 h, vilket är det näst bästa resultatet för alla strategier som simulerats vid soligt väder.



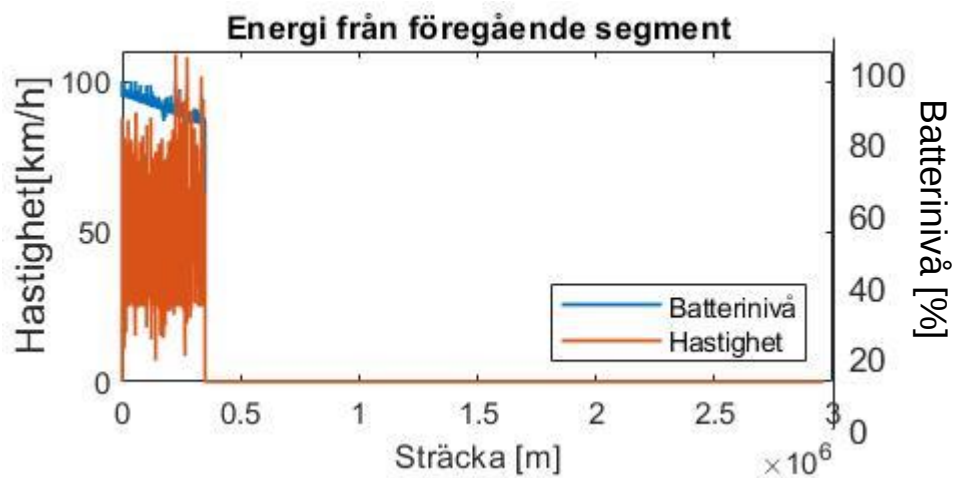
Figur 6-13, Bästa resultat för strategin: Effekt vid låg hastighet.
Hastighetsgräns: 55,99 km/h, sluttid: 54,90 h

6.4 Energi från föregående segment

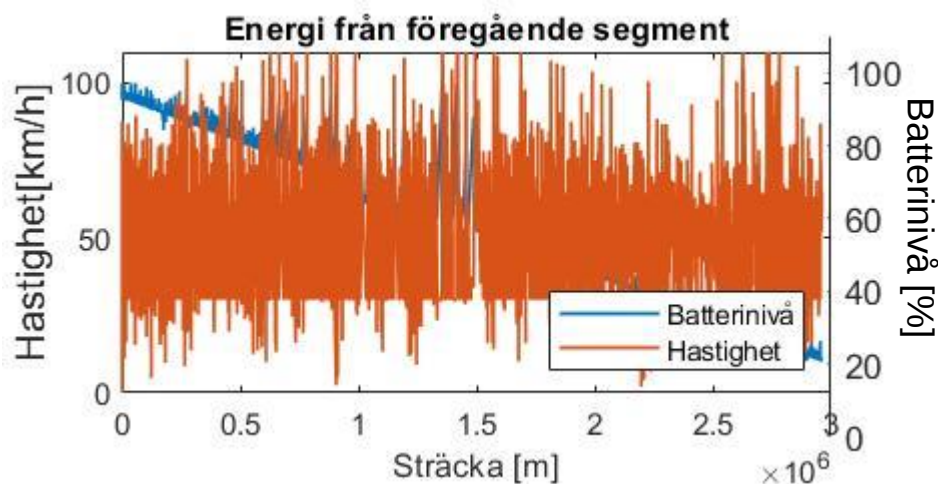
Resultat av strategin som använder energin ackumulerad i batteriet från föregående vägsegment. Denna strategi använder hastighetsgränserna genom att hastigheten ökas till den undre hastighetsgränsen om bilen annars stannat. Denna strategi fluktuerar mycket i hastighet och använder batteriet väl då strategin optimerats.

Denna strategi kommer inte fram både om hastigheten är för låg eller för hög, se Figur 6-14 samt Figur 6-15. Då den är för låg kan inte bilen komma över första backen. Detta beror på att den extra lägesenergin som fås av en högre nedre gräns uteblir. Om den lägre gränsen är för hög kommer däremot bilen dra för mycket energi och batteriet tar i stället slut lägre upp i första backen. Dessa två scenarion sker vid 25 respektive 35 km/h. Om hastighetsbegränsningen sätts

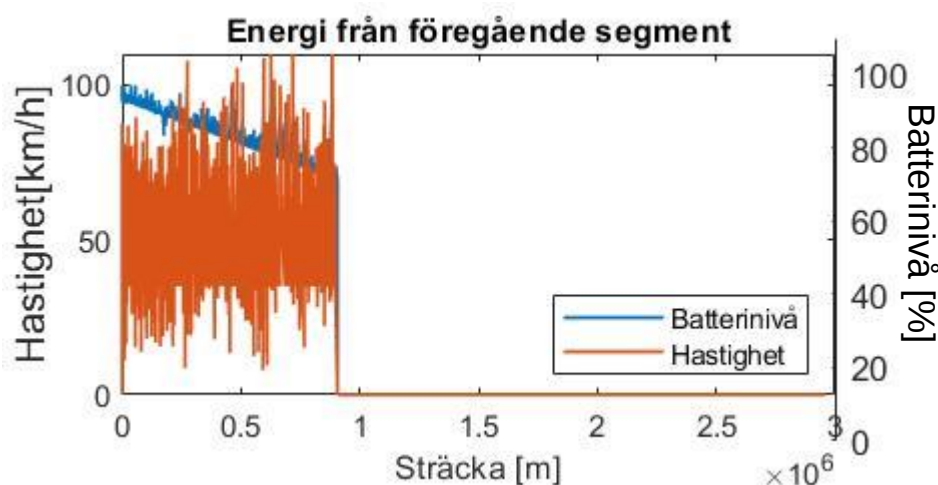
till 30 km/h kommer bilen däremot att komma fram i mål om ändå förhållandevis långsamt, se Figur 6-15.



Figur 6-14, Energi från föregående segment, Hastighetsbegränsning: 25 km/h, Batteriet tog slut efter 350112 m



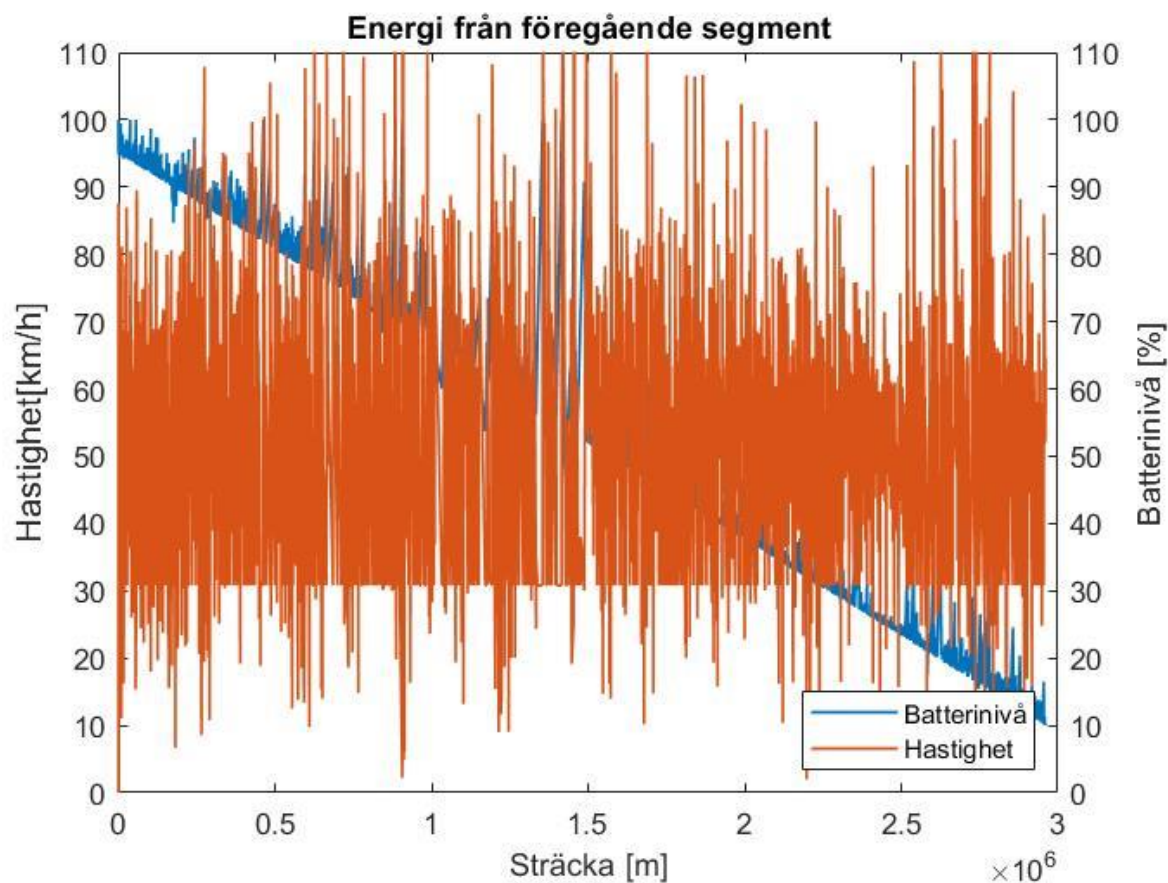
Figur 6-15 Energi från föregående segment, Hastighetsbegränsning: 30 km/h, sluttid: 61,65 h



Figur 6-16, Energi från föregående segment, Hastighetsbegränsning: 35 km/h, Batteriet tog slut efter 906570 m

6.4.1 Optimalt resultat för energi från föregående segment

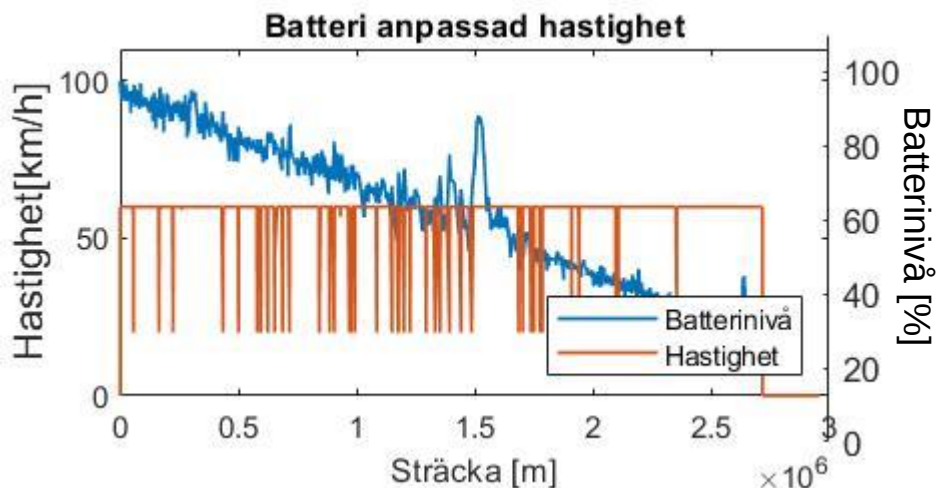
Det optimala resultatet för energi från föregående segment ger en sluttid på 61,54 h. Detta är förhållandevis långsamt jämfört med de andra strategierna. Denna strategi tar helt slut på batteriet men kör väldigt långsamt stundtals. Hastigheten fluktuerar mest jämfört med de andra strategierna.



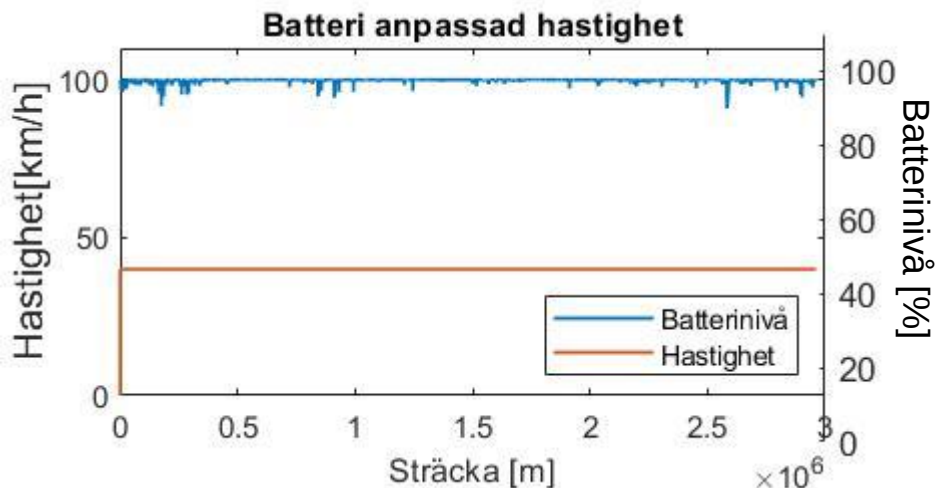
Figur 6-17 Bästa resultat av strategi Energi från föregående segment.
Hastighetsbegränsning 30,79 km/h, sluttid: 61,54 h

6.5 Batterianpassad hastighet

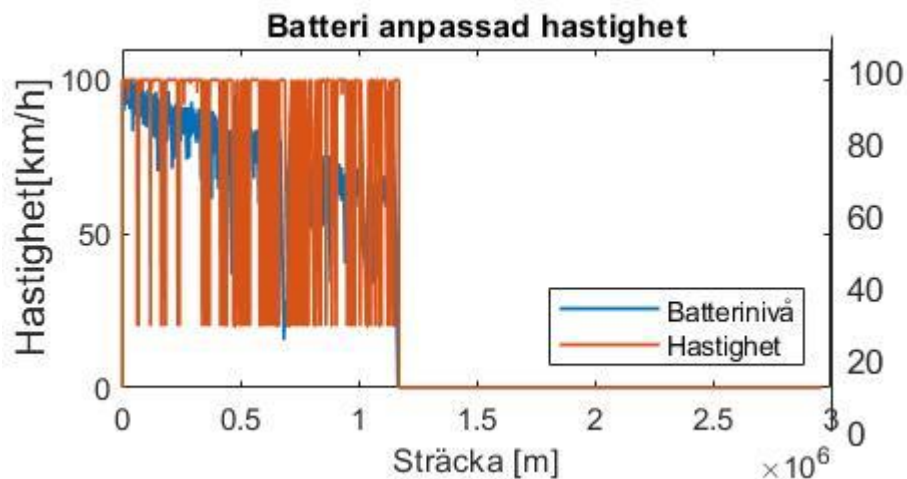
Strategin batterianpassad hastighet håller hög hastighet då batteriets laddning är högt och håller låg hastighet om batteriets laddning är lågt. Denna strategi optimeras utifrån två hastighetsgränser. Om den övre gränsen är mycket låg kommer bilen enbart köra i den högsta hastigheten, se Figur 6-19. Detta gör att batteriets kapacitet används dåligt och att resan tar lång tid. Om hastighetsgränserna har ett väldigt stort spann mellan sig, speciellt om den övre är väldigt hög, kommer batteriet att ta slut innan bilen kommer fram, se Figur 6-20 och Figur 6-19. Även om den övre gränsen sätts till 60 och den lägre 20 km/h klarar inte bilen att köra förbi sista backen, se Figur 6-18. Dessa gränser behöver vara förhållandevis snäva, se optimerad version Figur 6-21.



Figur 6-18 Batterianpassad hastighet, hastighetsbegränsningar 20 respektive 60 km/h. Batteri slut efter 2718464 m



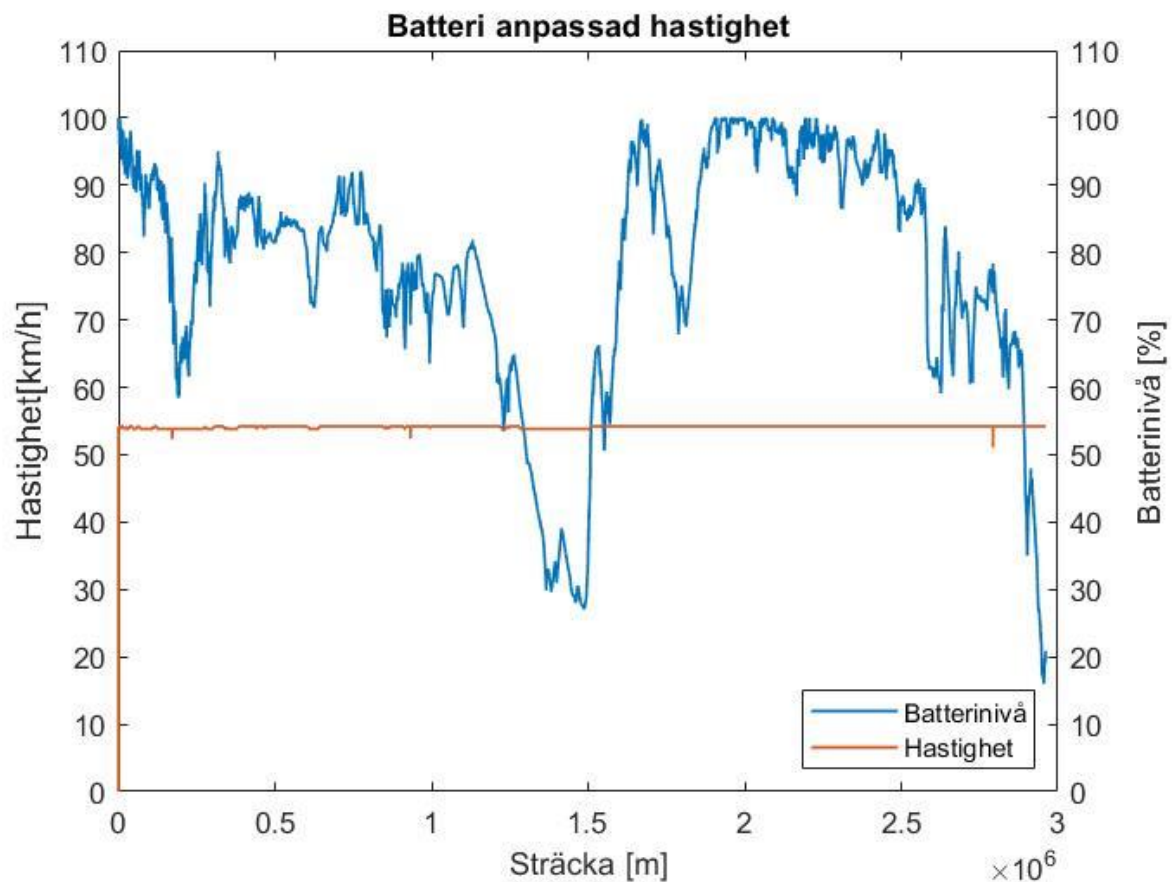
Figur 6-19 Batterianpassad hastighet, hastighetsbegränsningar 20 respektive 40 km/h. sluttid: 74.03 h



Figur 6-20 Batteri anpassad hastighet, hastighetsbegränsningar 20 respektive 100 km/h. Batteri slut efter 1171540 m

6.5.1 Optimalt resultat för energi från batteri anpassad hastighet

Det optimala resultatet för batteri anpassad hastighet gav när hastighetsgränserna var 53,82 respektive 54,2 km/h, se Figur 6-21. Detta gav en sluttid på 54,71 h vilket är det bästa resultatet av alla strategier. Denna strategi utnyttjar batteriets kapacitet på ett bra sätt. Hastigheten är dessutom i princip konstant vilket gör att lite energi går åt till acceleration.

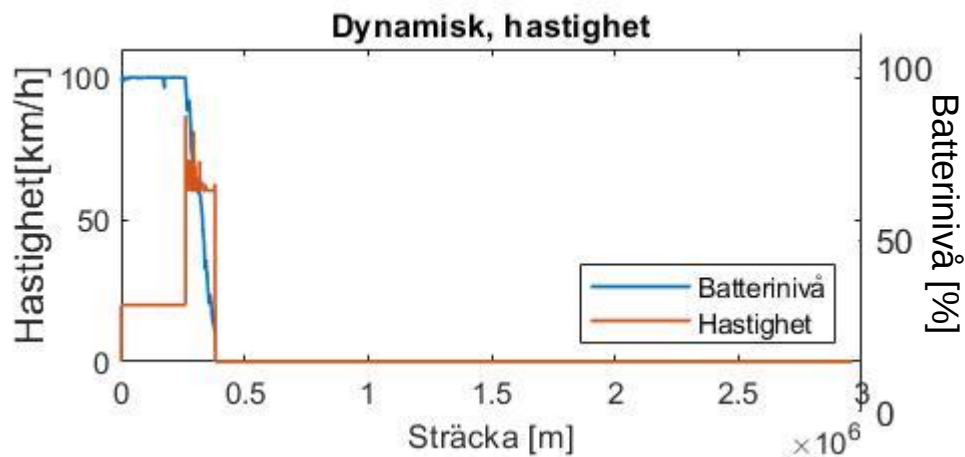


Figur 6-21 Bästa resultat för strategin Batteri anpassad hastighet, Hastighetsbegränsningar 53,82 respektive 54,2 sluttid: 54,71 h

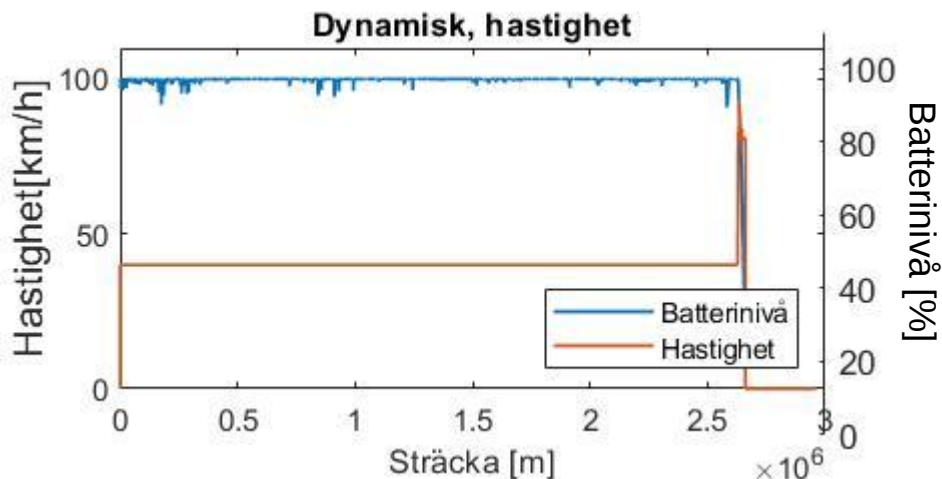
6.6 Dynamisk hastighet

Resultat av strategi som håller en konstant hastighet då hastigheten är inom det tillåtna spannet. Rullar om bilen kan komma upp i det högre hastighetsspannet och håller annars en konstant lägre hastighet. Pondera att hastighetsgränserna är 30 respektive 40 km/h. Om bilen kör i 30 km/h och kör i en backe som eventuellt skulle kunna ge bilen en acceleration till 35 km/h, då kommer bilen behålla sin hastighet på 30 km/h. Om bilen däremot kan få en hastighet på 45 km/h från backen kommer bilen att accelerera till denna hastighet. Detta gör att denna strategi inte fluktuerar lika mycket som strategin dynamisk energi men också att strategin blir långsammare.

Nedan visas exempel på två av de fall som simuleras vid optimering, se Figur 6-22 samt Figur 6-23. Hastighetsgränserna 20 respektive 60 km/h som kan ses i första grafen är förhållandevis låga hastighetsbegränsningar. Dock kommer dessa ta slut på batteriet i första backen. Då hastighetsgränserna är satta till 40 respektive 80 km/h kommer batteriet ta slut senare. I båda dessa fall är gränserna för långt ifrån varandra för att ge ett bra resultat.



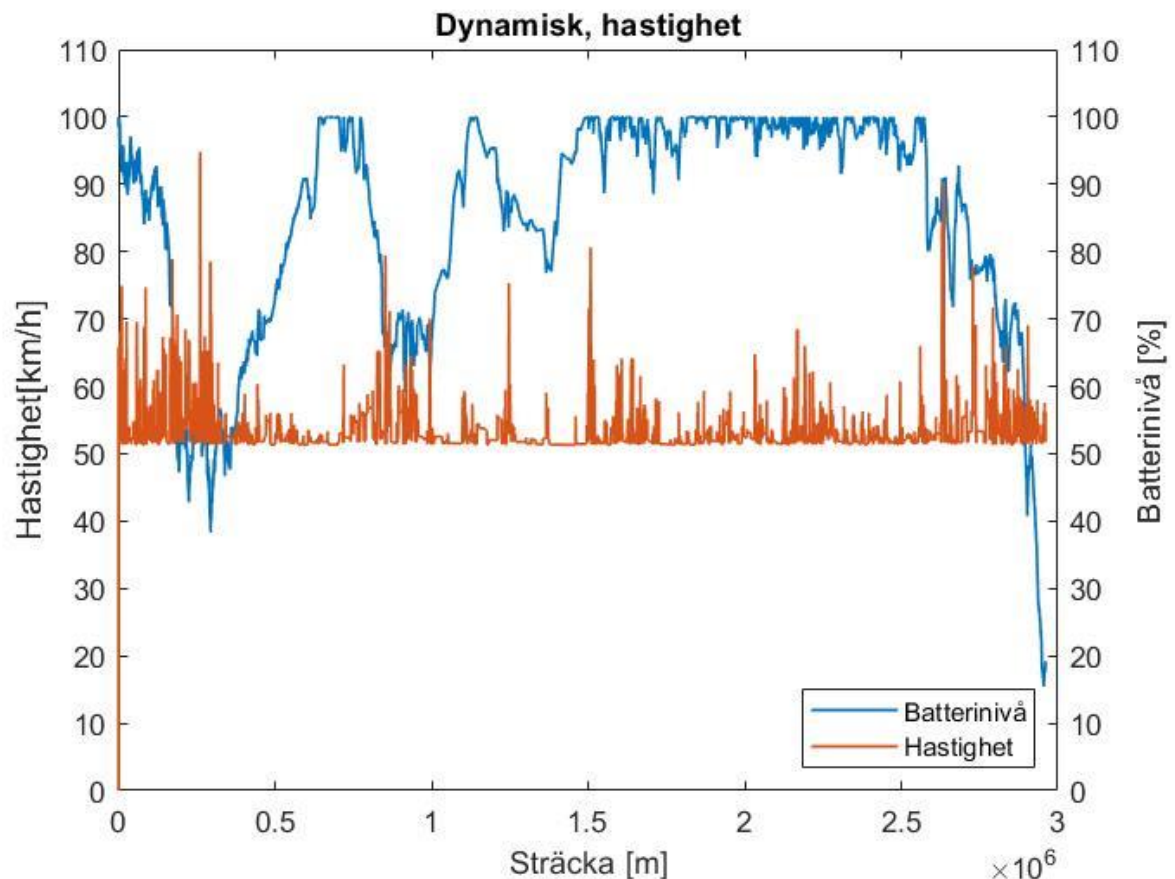
Figur 6-22 Dynamisk, hastighet Hastighetsbegränsningar 20 respektive 60 km/h, batteri slut efter 380432 m



Figur 6-23 Dynamisk, hastighet Hastighetsbegränsningar 40 respektive 80 km/h, batteri slut efter 2892504 m

6.6.1 Optimalt resultat för Dynamisk, hastighet

Det optimala resultatet för dynamisk, hastighet var med hastighetsgränserna 50,91 respektive 51,3 km/h, se Figur 6-24. Som kan ses i grafen nedan kommer batteriet att vara 20 % fullt vid målgång. Batteriet kommer dock också att vara maximalt laddat vid flera tillfällen under loppets gång. Detta är ineffektivt då batteriet inte laddas vid så höga batterinivåer. Denna strategi är utanför SOC mest av alla strategier. Strategin är till trots inte långsammast av strategierna utan har en sluttid på 55,75 h. Jämför detta med strategi dynamisk energi som har en sluttid på 58,32 h men använder batteriet till fullo.



Figur 6-24, bästa resultat för strategin dynamisk hastighet. Hastighetsbegränsningar 50.96 respektive 51,32 sluttid 55.75h

6.7 Resultat optimering

Strategin batterianpassad hastighet var i simuleringen vid sol den bästa strategin. Nedan visas resultatet av de olika strategierna, se Tabell 6-1, Resultat vid sol.

Tabell 6-1, Resultat vid sol

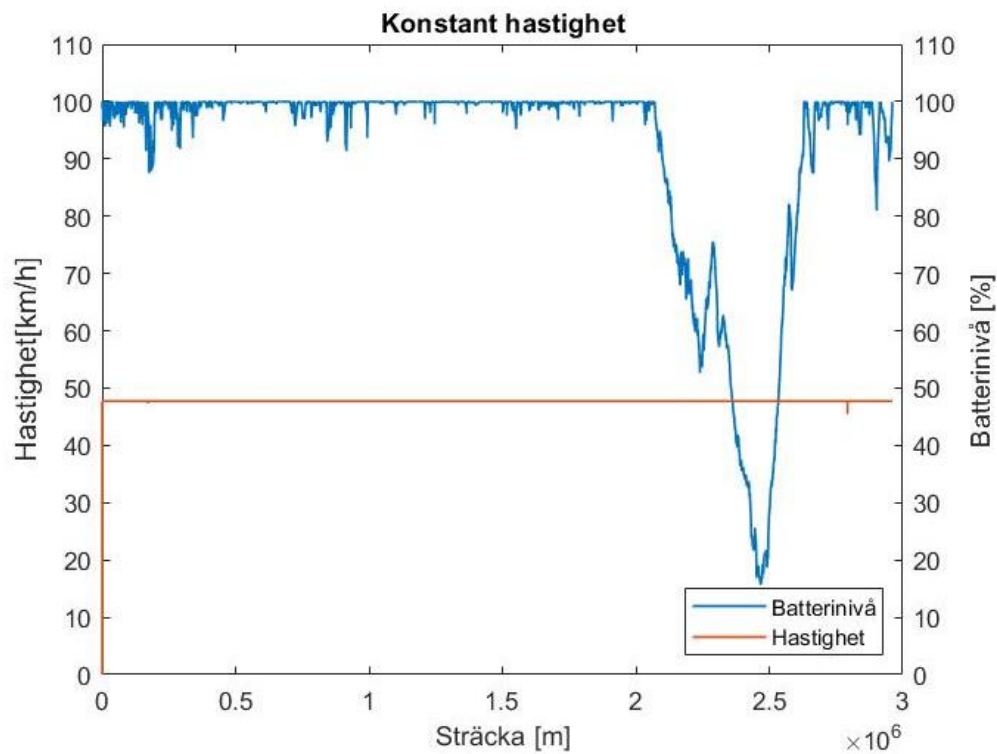
Strategi	Hastighetsbegränsning [km/h]		Sluttid [h]
	Min	max	
Konstant hastighet	52,4	-	56,51
Dynamisk, energi	34,4	34,8	58,32
Effekt vid låg hastighet	56,0	-	54,90
Energi från föregående segment	30,8	-	61,54
Batterianpassad hastighet	53,8	54,2	54,71
Dynamisk, hastighet	51,0	51,3	55,75

6.8 Regn

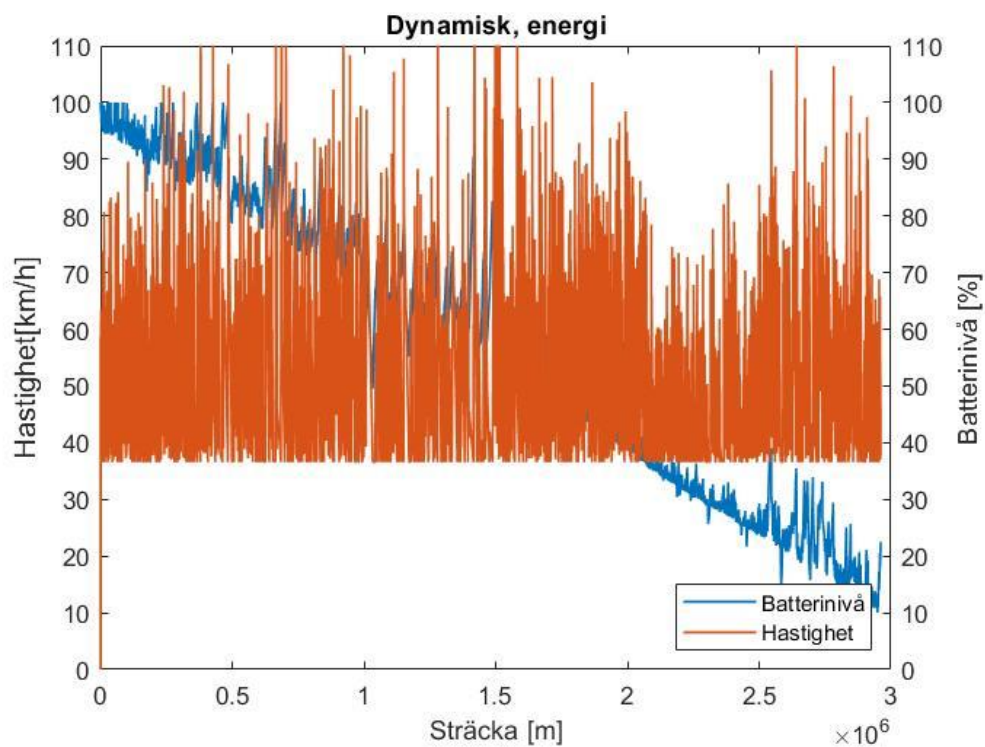
Väder påverkar solcellers förmåga att generera el. Under tävlingen finns en risk att det regnar. Om det skulle regna över en del av ruten påverkas hur bilen kan köra drastiskt. Nedan visas simuleringar av de sex olika strategierna vid regn. I detta fall är regn simulerat för 50mil under 8 timmar av resan. Alla strategier utom dynamisk energi och energi vid föregående vägsegment använder föga batteriernas kapacitet, se Figur 6-25, Figur 6-27, Figur 6-29 och Figur 6-30. Strategi energi vid föregående vägsegment klarar inte att komma i mål utan att batteriet tar slut, se Figur 6-28. Den strategi som är bäst är dynamisk energi, se Figur 6-26. Denna strategi har en sluttid på 59,23 h och använder batteriets kapacitet till fullo. Detta trots att hastighetsgränsningarna för strategin är de näst lägsta för alla strategier under regn, se Tabell 6-2, Resultat vid regn. **Fel! Hittar inte referenskälla..**

Tabell 6-2, Resultat vid regn

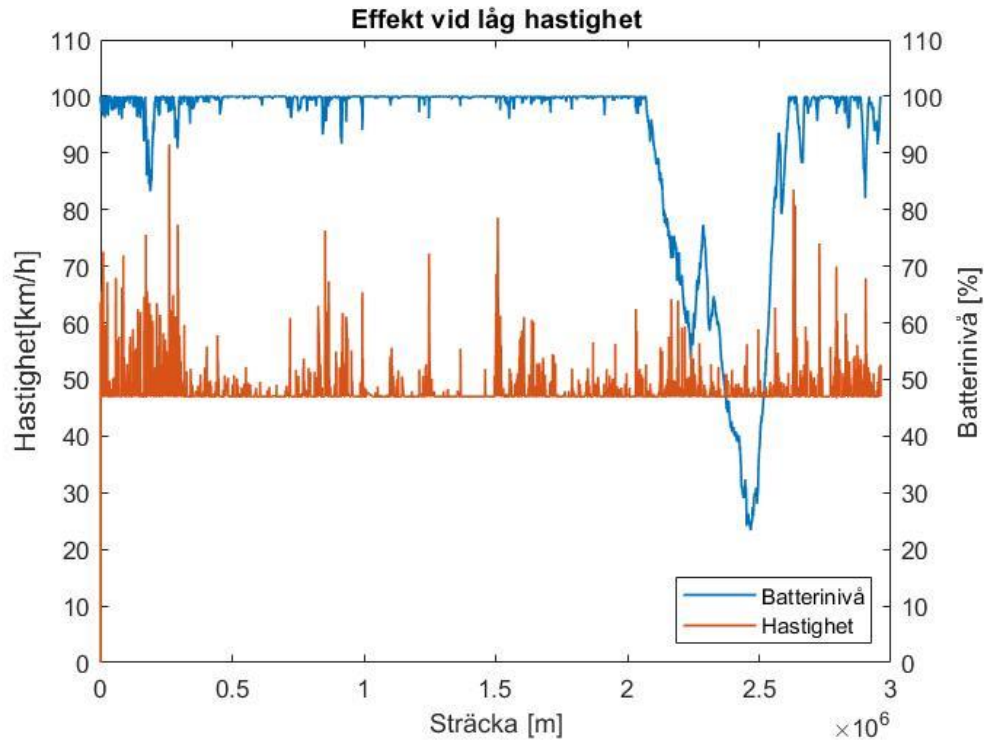
Strategi	Hastighetsbegränsning [km/h]		Sluttid [h]
	Min	max	
Konstant hastighet	47,7	-	62,05
Dynamisk, energi	36,5	36,9	59,23
Effekt vid låg hastighet	47,0	-	62,14
Energi från föregående segment	23,6	-	-
Batterianpassad hastighet	47,4	47,7	62,06
Dynamisk, hastighet	45,6	47,7	61,67



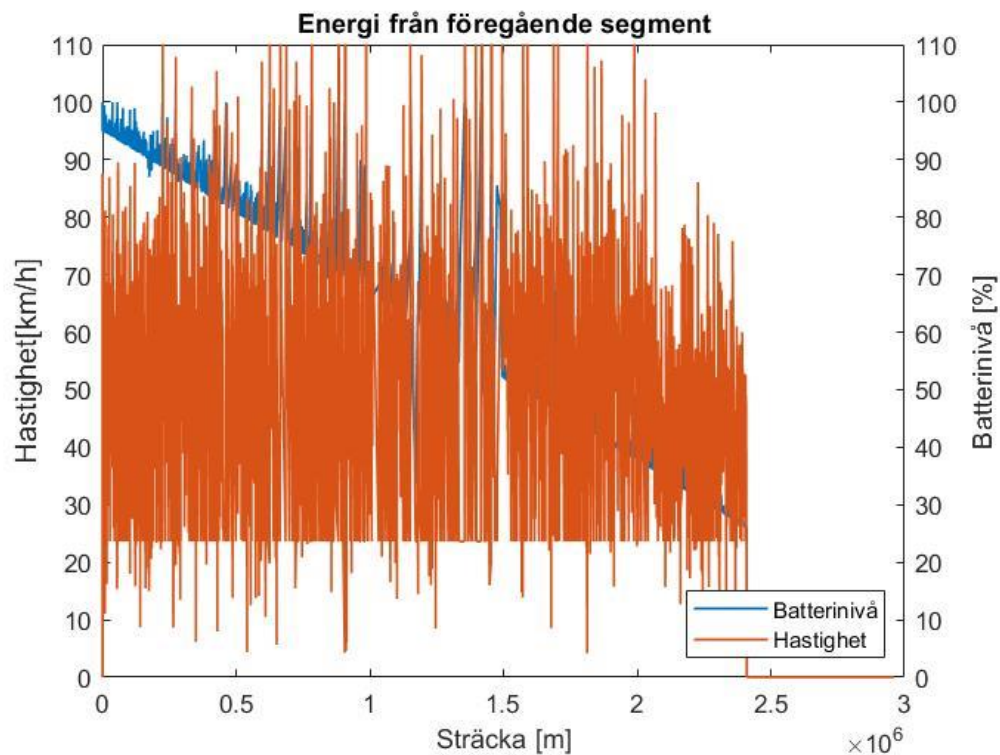
Figur 6-25, Konstant hastighet, Regn över område på 50 mil. Bästa resultat: Hastighetsbegränsning 47.42 sluttid: 62.05h



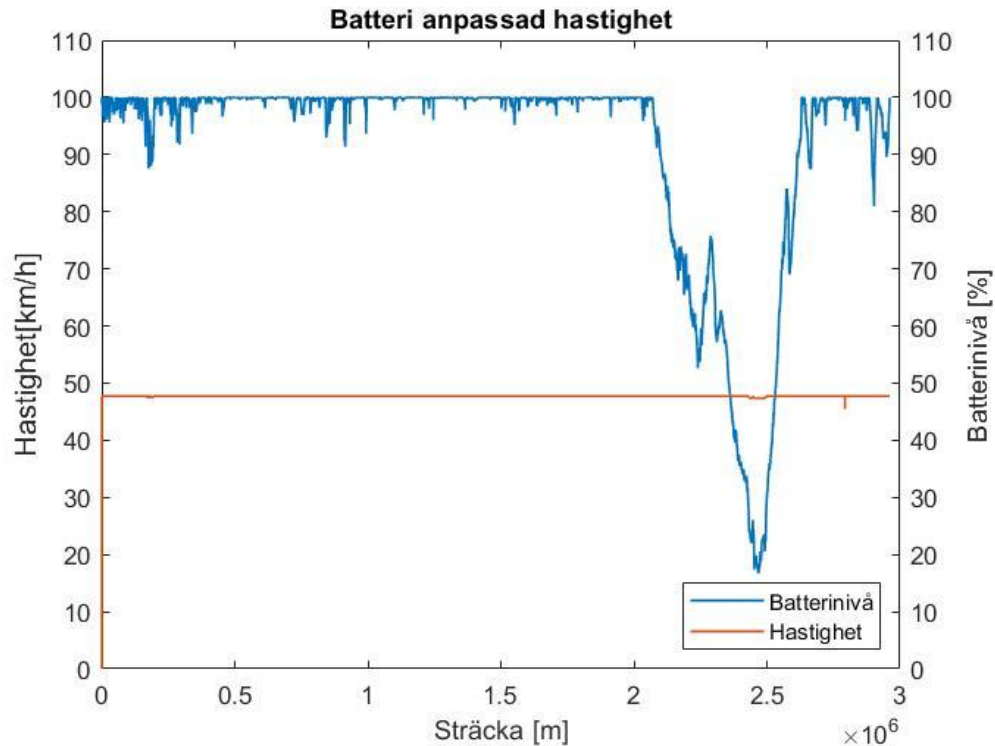
Figur 6-26, Dynamisk energi, Regn över område på 50 mil. Bästa resultat: Hastighetsbegränsningar 36,5 respektive 36,9 km/h sluttid: 59,24 h



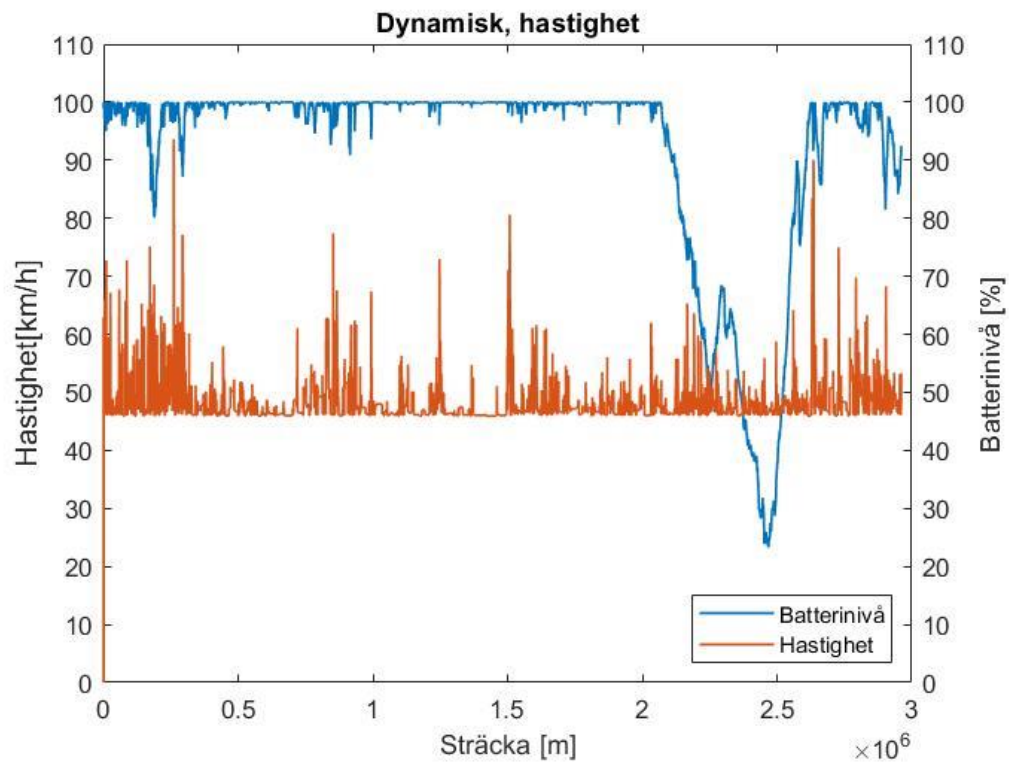
Figur 6-27, Effekt vid låg hastighet Regn på ett område om 50 mil. Bästa resultat: Hastighetsbegränsningar 47 km/h sluttid: 62.14h



Figur 6-28, Energi från föregående segment. Regn över område på 50 mil. Bästa resultat: Hastighetsbegränsningar 23,60 respektive 120 km/h. Batteriet tar slut efter 2410157 m



Figur 6-29, Batterianpassad hastighet, Regn över område på 50 mil. Bästa resultat: Hastighetsbegränsningar 47,36 respektive 47,7 km/h sluttid: 62,07 h



Figur 6-30, Dynamisk, hastighet. Regn över område på 50 mil. Bästa resultat: Hastighetsbegränsningar 45,56 respektive 45,92 km/h sluttid: 61,67 h

6.9 Simulerad tävling

Detta program kommer att användas som ett hjälpmedel för tävlingen i Bridgestone World solar challenge. Nedan visas resultatet av programmet då det används på det sättet som programmet är ämnat till att göra. Programmet ska uppdateras varje morgon för att få den mest aktuella väderdatan samt korrigering kan göras för batterinivå så att programmet kan förutspå kvarvarande energiförbrukning samt batteriet ackumulation mer exakt. Då bilen endast körs 8 timmar om dagen kommer programmet att uppdateras efter 8 timmar av körning.

Nedan visas ett simulerat race. I denna simulation har programmet fått uppdatera sina värden en gång varje morgon. Detta har gjort att programmet har kunnat använda energin till förfogande bättre. Simuleringen har gjorts utifrån samma data som använts i avsnittet 6.8 regn. Nedan kan resultatet av varje dags optimerade strategi ses, se Figur 6-32, Figur 6-33, Figur 6-34, Figur 6-35, Figur 6-36, Figur 6-37 och Figur 6-38. Dessa grafer visar att strategin långsamt kan öka hastigheten.

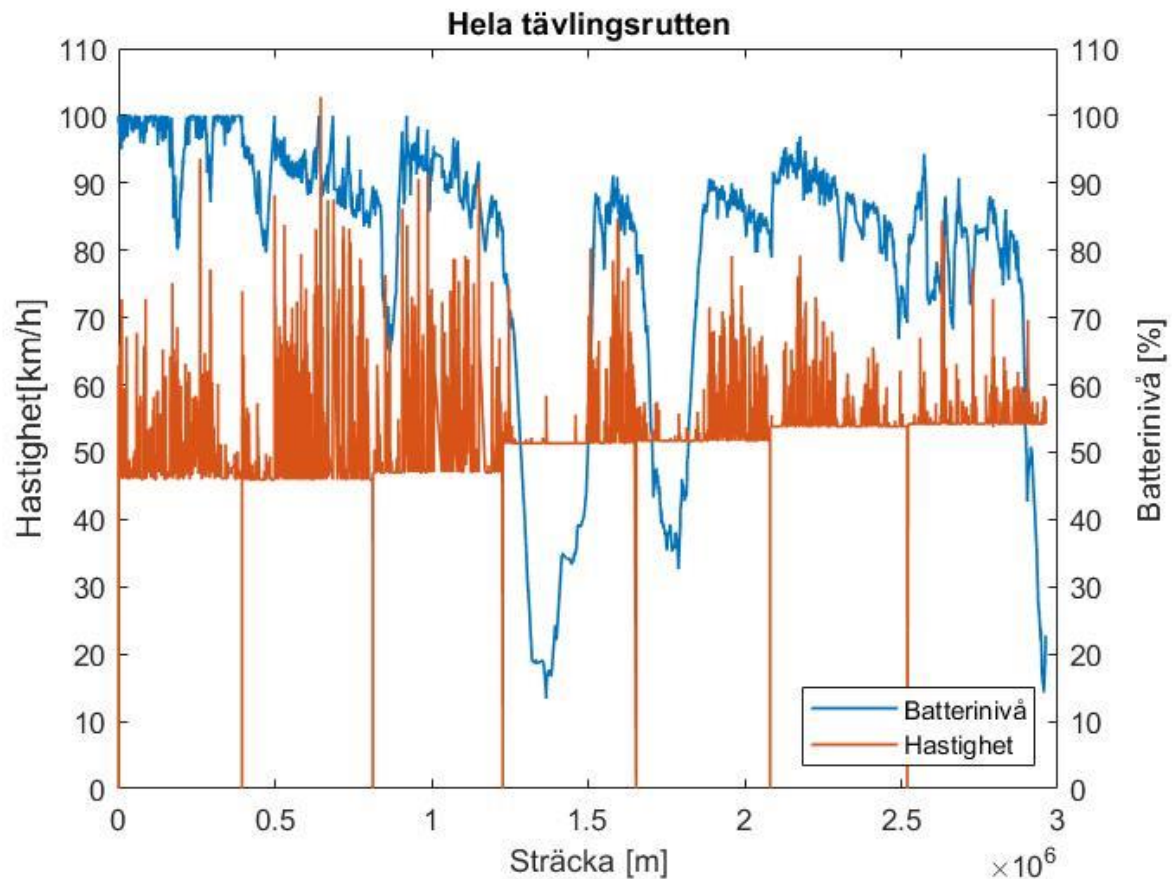
Dels består tävlingssträckan av först en backe och sedan en nedförsbacke, se avsnitt 5.7.2 och dels uppdateras den optimala strategin varje morgon. Dessa två faktorer gör att hastigheten för bilen successivt kan höjas, se Figur 6-31, Resultatet av optimerad strategi med ny uträkning inför varje dag av loppet, sluttid: 56,52 h. Denna optimering med uppdatering ger en sluttid på 56,52 h. Detta är mycket kortare än sluttiden på 59,24 h som kalkyleras om enbart en optimering körs. Det är dessutom en bättre tid än tre av de andra strategierna vid strålende sol.

En konventionell farthållare kan liknas vid strategin konstant hastighet. Om denna använts under loppet där hastigheten räknats ut en gång skulle solbilen komma i mål efter 62,05 h under regn. Om däremot strategin dynamisk energi används med uppdatering av parametrar och väder varje dag blir sluttiden 56,52 h. Detta gör att tidsvinsten blir 5,54 h.

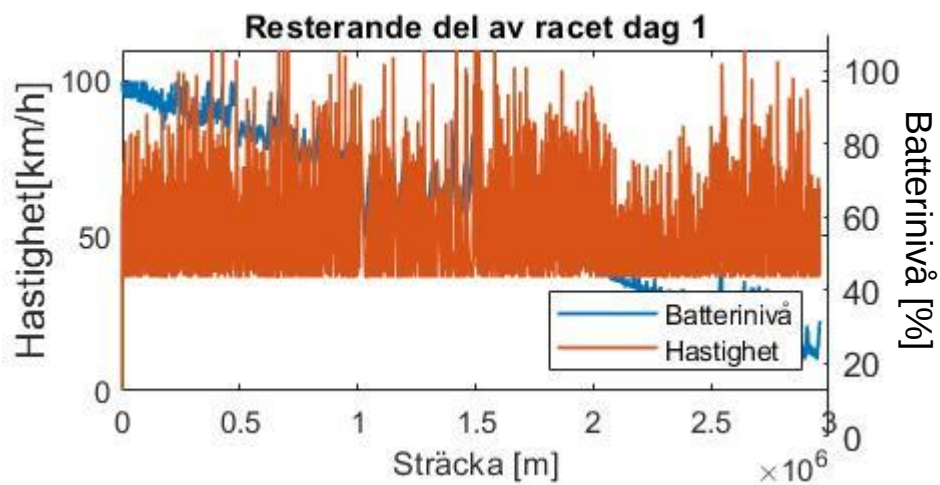
Tabell 6-3, Resultat av beräkningar dag för dag

Dag	Bästa strategi	Hastighetsbegränsning [km/h]	
		Min	max
1	Dynamisk energi	36,5	36,9
2	Dynamisk energi	45,9	46,3
3	Dynamisk energi	47,0	47,4
4	Dynamisk energi	51,3	58,7
5	Dynamisk energi	51,7	52,0
6	Dynamisk energi	53,8	54,2
7	Dynamisk energi	54,2	54,6

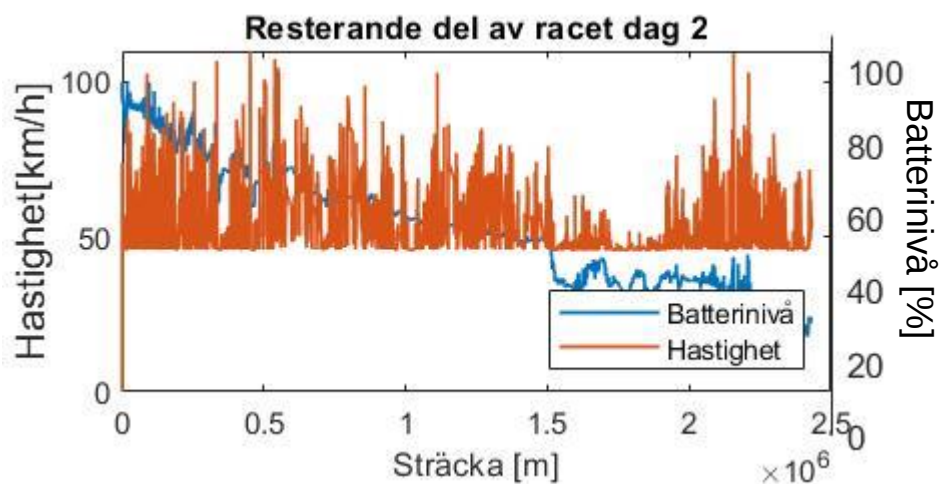
Sluttid för hela loppet: 56,52 h



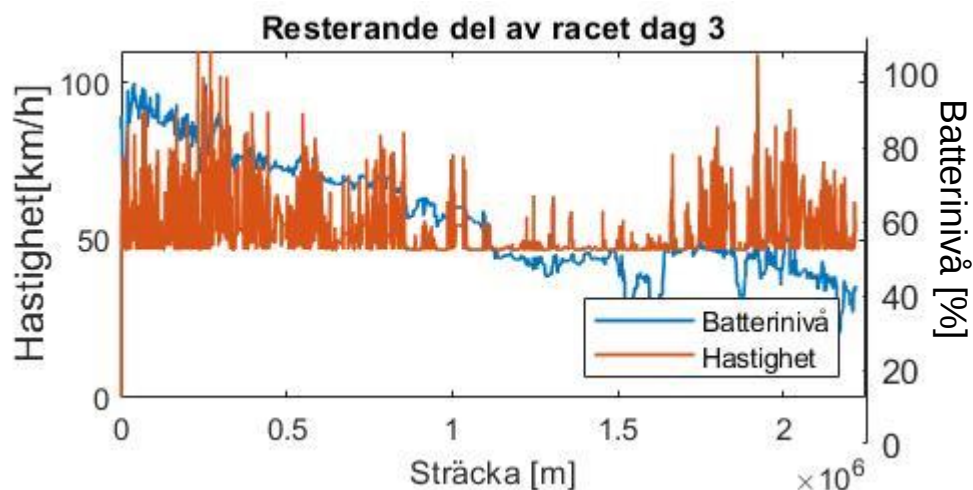
Figur 6-31, Resultatet av optimerad strategi med ny uträkning inför varje dag av loppet, sluttid: 56,52 h



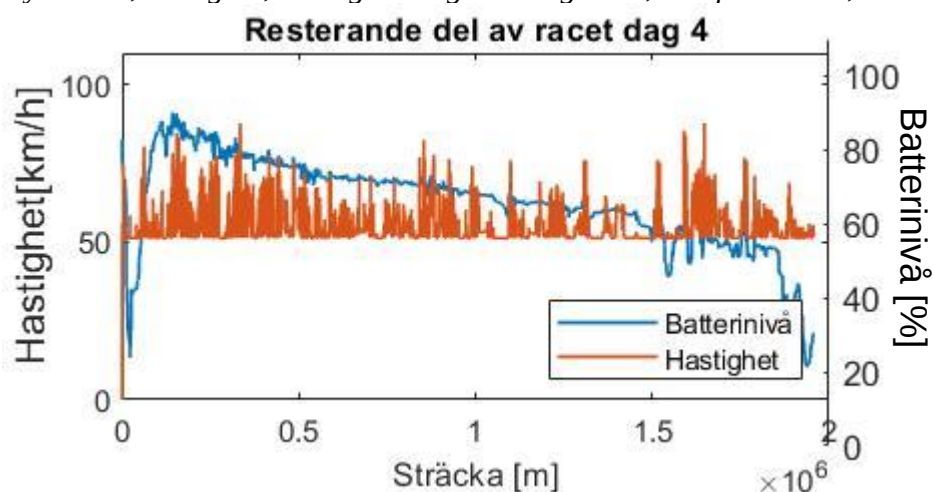
Figur 6-32, Resultat av optimering för resterande del av tävlingen, dag 1. Bästa strategi: Dynamisk, hastighet, Hastighetsbegränsningar 34,4 respektive 34,8 km/h



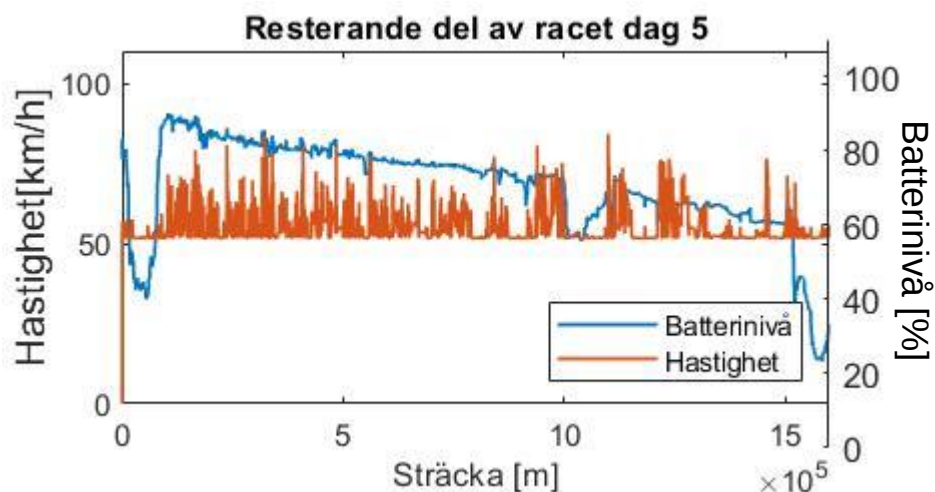
Figur 6-33, Resultat av optimering för resterande del av tävlingen, dag 2. Bästa strategi: Dynamisk, hastighet, Hastighetsbegränsningar 45,92 respektive 46,28 km/h



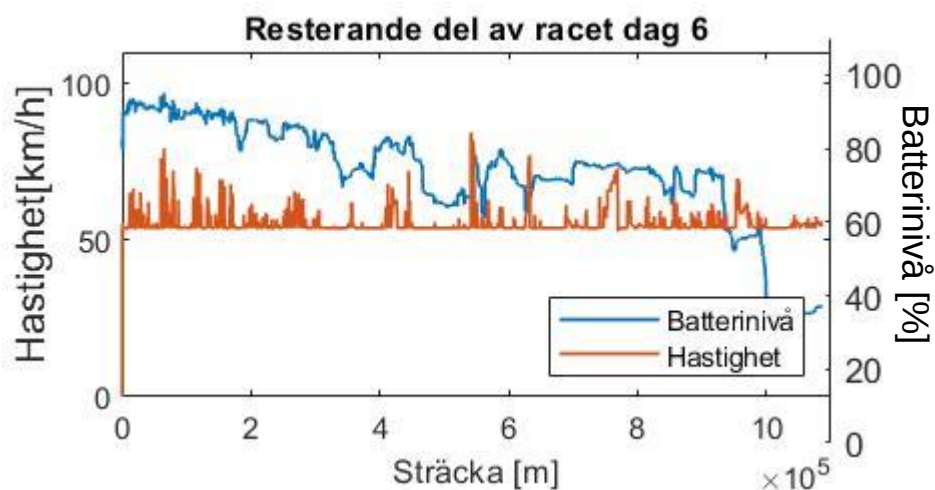
Figur 6-34 Resultat av optimering för resterande del av tävlingen, dag 3. Bästa strategi: Dynamisk, hastighet, Hastighetsbegränsningar 47,0 respektive 47,4 km/h



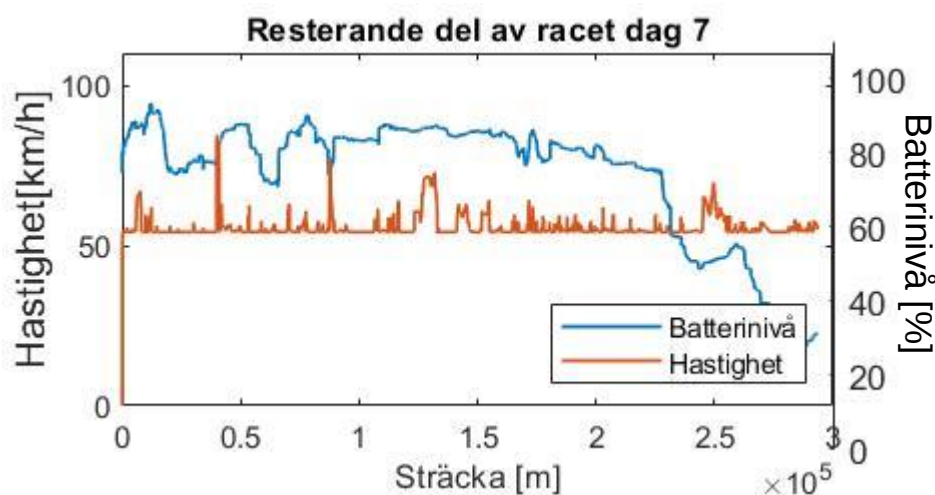
Figur 6-35, Resultat av optimering för resterande del av tävlingen, dag 4. Bästa strategi: Dynamisk, hastighet, Hastighetsbegränsningar 51,3 respektive 51,7 km/h



Figur 6-36, Resultat av optimering för resterande del av tävlingen, dag 5. Bästa strategi: Dynamisk, hastighet, Hastighetsbegränsningar 51,7 respektive 52,0 km/h



Figur 6-37, Resultat av optimering för resterande del av tävlingen, dag 2. Bästa strategi: Dynamisk, hastighet, Hastighetsbegränsningar 53,8 respektive 54,2 km/h



Figur 6-38, Resultat av optimering för resterande del av tävlingen, dag 2. Bästa strategi: Dynamisk, hastighet, Hastighetsbegränsningar 54,2 respektive 54,6 km/h

7 RESULTAT

I kapitlet resultat beskrivs måluppfyllelsen för projektet.

7.1 Måluppfyllelse

Programmet framtaget i detta projekt kan med hjälp av statistik och data ta fram en modell för vilken hastighet bilen optimalt ska ha över en viss rutt utifrån sex strategier.

Utifrån det resultat som visas i avsnitt strategier i detta kapitel finns det god möjlighet att resultatet kan användas för att säkerställa målgång samt öka prestationen i racet. Enligt simuleringar gjorda i detta projekt, se avsnitt 6.9, ger optimeringstekniken en tidsvinst på 5,54 h jämfört med farthållare inställd på en konstant hastighet.

Resultatet som har simulerats över projektets gång visar att bilen inte kan hålla en hastighet på 60km/h utan att behöva stanna innan målgång. Då friktionen och andra parametrar enbart är framtagna teoretiskt visar detta att dessa parametrar behöver ställas in noggrannare för att programmet ska kunna användas i racet.

Programmet behöver vidare testas för att verifiera dess förmåga att förutspå framtida ackumulation av batteriet samt energikonsumtion.

8 SLUTSATS OCH DISKUSSION

Nedan följer ett kapitel som diskuterar vidareutveckling av produkten samt arbetsgången

8.1 Vidareutveckling

I dagsläget kan programmet framtaget i detta projekt enbart användas för att ge en fingervisning om hur solbilar kan köra för att öka sin prestation under en rutt. Används programmet tillsammans med ett system som kan reglera bilen utefter strategierna är programmet ett väldigt enkelt verktyg för att få fram börvärden till reglersystemet.

I framtiden hade ett automatiskt system kunnat användas för att återge körstrategin direkt till bilen. Detta skulle kunna användas med system för att ladda ner väderprognoser samt att föraren skulle kunna skriva in en rutt. Strategin skulle sedan kunna uppdateras flera gånger under loppet för att ge ett mer tillförlitligt resultat. Detta program skulle behöva hålla reda på bilens GPS-koordinater och jämföra dessa med uträknade värden.

Programmet skulle även kunna uppdateras med fler strategier. Detta kan öka solbilens prestation ytterligare.

Konfiguration av programmet sker idag av användaren. Exakta värden för bilens konsumtion är svårt att experimentera fram. Om programmet kunde ta emot statistik från själva bilen hade framtida förutsägelserna om körväg och konsumtion kunnat bli ännu bättre. Friktionen hade t.ex. kunnat räknas ut av programmet och mer exakta värden hade kunnat användas vid förutsägelse.

8.2 Slutsats

Projektet var från början mycket större än det slutgiltiga resultatet. Hypotesen som antogs i början var att en simuleringsbil skulle användas för att få fram en bra körstrategi. Denna simuleringsbil skulle vara lik den slutgiltiga bilen och användas för olika experiment för att komma fram till det slutgiltiga resultatet.

Hypotesen om att simuleringsbilen skulle kunna ge goda resultat övergavs ganska fort. Detta för att solbilen detta projekt utvecklas för ej var specificerad. Det fanns många frågetecken kring hur den slutgiltiga bilen skulle vara. Att konstruera en modellbil att simulera med visade sig därmed opraktiskt.

Projektet tog därmed en annan riktning och gjordes utifrån en teoretisk grund. Programmet blev dessutom allmänt. Vilket resulterade i att det skulle kunna användas både var som helst och till vilken bil som helst.

Avslutningsvis visar projektets resultat att körstrategier för solbilar kan öka avståndet en solbil kan köra samt minska tiden fram till slutdestinationen. Om teorin vidareutvecklas har den mycket goda möjligheter att vara en del av en hållbarare utveckling av fordon.

REFERENSER

- [1] Northern Arizona Wind and sun, "www.solar-electric.com," 2019. [Online]. Available: <https://www.solar-electric.com/mppt-solar-charge-controllers.html>. [Använd 21 04 2019].
- [2] UNDP, "www.globalamalen.se," UNDP, 2015. [Online]. Available: <https://www.globalamalen.se/om-globala-malen/>. [Använd 29 05 2019].
- [3] World solar Challenge, "https://www.worldsolarchallenge.org," 17 October 2018. [Online]. Available: https://www.worldsolarchallenge.org/files/2272_2019_bwsc_regulations_release_version_3.pdf. [Använd 20 December 2018].
- [4] P. S. P. E. S. K. C. H. Wen Li, "Determining the main factors influencing the energy consumption of electric vehicles in the usage phase," 23rd CIRP Conference on Life Cycle Engineering, Joint German-Australian Research Group, 2016.
- [5] G. H. H. M. W. N. A.-A. Syed A.M. Said, "The effect of environmental factors and dust accumulation on photovoltaic modules and dust-accumulation mitigation strategies," *Renewable and Sustainable Energy Reviews*, vol. 82, nr 1, pp. 743-760, Februari 2018.
- [6] R. Korthauer, *Lithium-Ion Batteries*, Berlin: Springer Nature, ISBN 978-3-662-53071-9, 2018.
- [7] D. s. R. X. F. b. L. J. L. M. Ouyanga, "Comparison of the Overcharge Behaviors of Lithium-ion Batteries Under Different Test Conditions," *Energy Procedia*, Beijing, 2019.
- [8] C. M. Gonzales, "What's the Difference Between Friction and Regenerative CAR BRAKES?," *Gonzalez, Carlos M*, p. Vol. 88, 1 mars 2016.
- [9] Michellin, "thetiredigest.michelin.com," 2019. [Online]. Available: <https://thetiredigest.michelin.com/michelin-ultimate-energy-tire>. [Använd 21 04 2019].
- [10] J. C. R. H. T. Yunus A. Cengel, *Fundamentals of Thermal-Fluid Sciences*, New York: McGraw- Hill Education, 2017.
- [11] Wikipedia, "wikipedia.org," 1 november 2018. [Online]. Available: https://en.wikipedia.org/wiki/Geographical_distance. [Använd 22 05 2019].

[12 J. Skansholm, C från början, Lund: Studentlitteraur AB, 2016.
]

[13 Mitsuba, "Mitsuba.co.jp," Mitsuba, 2019. [Online]. Available:
] <https://www.mitsuba.co.jp/scr/products/m2096-%E2%85%A2>. [Använd 25 05 2019].

[14 H. W. D. C. Chen Lv, "Chapter 8 - Brake-Blending Control of EVs," i *Modeling,
] Dynamics and Control of Electrified Vehicles*, Bedford, WoodHead Publishing, 2017,
pp. 275-308.

[15 L. Brodén, Interviewee, *Teknikansvarig Solarteam*. [Intervju]. 25 03 2019.
]

[16 Panasonic, "Specifikations for NCR18650GA, 4J14Z5UTAS," Panasonic, 2019-05-25.
]

BILAGA 1, MATLAB KOD

RACE.M

```
%Läser in speed_setpoint för att göra grafer

fileID = fopen('speed_setpoint_sl_bit.txt','r');
formatSpec = '%f';
sizeA= [6 inf];
S1 = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);

t=size(S1);

z=t(1,2);

Yzero=zeros(2,z);
X1=S_EL(1,1:z);
YMatrix1=[[S1(5,:);S1(6,:)]];

%CREATEFIGURE(X1, YMatrix1)
% X1: vector of x data
% YMATRIX1: matrix of y data

% Auto-generated by MATLAB on 26-May-2019 10:55:27

% Create figure
figure1 = figure('Name','Körd strategi hela tävlingen');

% Create axes
axes1 = axes('Parent',figure1);
hold(axes1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(X1,YMatrix1,'Parent',axes1,'LineWidth',1);
set(plot1(1),'DisplayName','Batterinivå');
set(plot1(2),'DisplayName','Hastighet','MarkerSize',1);

% Create ylabel
ylabel('Hastighet[km/h]','HorizontalAlignment','center','fontsize',12);

% Create xlabel
xlabel('Sträcka [m]');
```

```

% Create title
title({'Dynamisk energi, första 30
minuterna'}, 'HorizontalAlignment', 'center', ...
      'FontWeight', 'bold');

% Uncomment the following line to preserve the Y-limits of
the axes
ylim(axes1, [0 110]);
box(axes1, 'on');
% Create legend
legend1 = legend(axes1, 'show');
set(legend1, 'Location', 'southeast');

% Create axes
axes2 = axes('Parent', figure1, ...
            'Position', [0.904166666666667 0.109803921568627
0.001302083333333337 0.815686274509804]);

% Create ylabel
ylabel('Batterinivå
[%]', 'HorizontalAlignment', 'center', 'fontsize', 12, ...
      'Interpreter', 'none');

% Uncomment the following line to preserve the Y-limits of
the axes
ylim(axes2, [0 110]);
xlim(axes1, [0 16000]);
% Set the remaining axes properties
set(axes2, 'YAxisLocation', 'right');
set(axes2, 'fontsize', 10);

```

GET_ELEVATION_DATA.M

```
%%läser in altitud data

fileID = fopen('copy.txt','r');
formatSpec = '%f';
sizeA= [4 inf];
GPS_EL = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);

%%räknar ut avståndet mellan punkterna
%%initerar S_EL med avstånd, altitude

S_EL=[0;GPS_EL(1,3)];
S=0;

for i=1:1:9856

    %%Longituderna till radianer
    long1=deg2rad(GPS_EL(2,i));
    long2=deg2rad(GPS_EL(2,(i+1)));

    % %%latituderna till radianer
    lat1=deg2rad(GPS_EL(1,i));
    lat2=deg2rad(GPS_EL(1,(i+1)));

    dlong=long1-long2;
    dlat=lat1-lat2;

    %%avståndsformel mellan punkterna fågelvägen
    %%ty sin för små vinklar = sin
    deg=sqrt(dlong.^2*cos(dlat/2)+dlat.^2)
    b_dist=deg*6371000;

    %%avstånd mellan punkterna med altitud inräknat
    y_dist=GPS_EL(3,i)-GPS_EL(3,(i+1));
    dist=sqrt((b_dist.^2)+(y_dist.^2));

    %%Avstånd från start
    S=dist+S;

    %%avstånd från start
    S_EL(1,(i+1))=S;
    %%meter över havet
    S_EL(2,(i+1))=GPS_EL(3,i);
    %%höjdskillnad från förra punkten
    S_EL(3,(i+1))=y_dist;
```

```
    %%Avstånd från förra punkten
    S_EL(4,(i+1))=b_dist;

end

var=0;
for i=1:1:9856
    var=var+S_EL(4,i);
end
```

BILAGA 2- RESULTAT AV VÄDERRATIO VERIFIERING

Sydney okt-18

Verkligt värde		Väder under dagen		dag
joule/dag*m2	medel watt/m2	första halvan	andra halvan	
21,9	481,3	sol	sol	1
23,1	507,7	sol	sol	2
17,3	380,2	sol	sol	3
4,6	101,1	regn	regn	4
6	131,9	regn	regn	5
14,3	314,3	regn	sol	6
10,8	237,4	mycket moln	regn	7
14	307,7	regn	lätta moln	8
18,7	411,0	sol	lätta moln	9
3,6	79,1	lätta moln	regn	10
12,3	270,3	mycket moln	lätta moln	11
8,9	195,6	regn	regn	12
17,6	386,8	mycket moln	regn	13
19	417,6	regn	sol	14
16,6	364,8	regn	mycket moln	15
22,2	487,9	sol	lätta moln	16
12,7	279,1	lätta moln	lätta moln	17
14,2	312,1	lätta moln	regn	18
23,6	518,7	lätta moln	sol	19
18,1	397,8	sol	lätta moln	20
16,3	358,2	lätta moln	lätta moln	21
21,2	465,9	lätta moln	sol	22
26	571,4	sol	lätta moln	23
12,3	270,3	sol	lätta moln	24
21,8	479,1	lätta moln	sol	25
19,2	422,0	lätta moln	sol	26
27,1	595,6	sol	sol	27
15,2	334,0	lätta moln	lätta moln	28
15,2	334,0	lätta moln	lätta moln	29
27,7	608,8	sol	sol	30
27,1	595,6	sol	sol	31

11617,0

metod:	Approximering med procentsatser			
dag	Beräknad medel watt/m2		tot under dagen	Avvikelse från verkligt värde
	första halvan	andra halvan		
1	232,9	232,9	465,8	97%
2	232,9	232,9	465,8	92%
3	232,9	232,9	465,8	123%
4	42,4	42,4	84,8	84%
5	42,4	42,4	84,8	64%
6	42,4	232,9	275,3	88%
7	141,9	42,4	184,3	78%
8	42,4	211,8	254,2	83%
9	232,9	211,8	444,7	108%
10	211,8	42,4	254,2	321%
11	141,9	211,8	353,7	131%
12	42,4	42,4	84,8	43%
13	141,9	42,4	184,3	48%
14	42,4	232,9	275,3	66%
15	42,4	141,9	184,3	51%
16	232,9	211,8	444,7	91%
17	211,8	211,8	423,6	152%
18	211,8	42,4	254,2	81%
19	211,8	232,9	444,7	86%
20	232,9	211,8	444,7	112%
21	211,8	211,8	423,6	118%
22	211,8	232,9	444,7	95%
23	232,9	211,8	444,7	78%
24	232,9	211,8	444,7	165%
25	211,8	232,9	444,7	93%
26	211,8	232,9	444,7	105%
27	232,9	232,9	465,8	78%
28	211,8	211,8	423,6	127%
29	211,8	211,8	423,6	127%
30	232,9	232,9	465,8	77%
31	232,9	232,9	465,8	78%
			10965,7	94%

metod:	Approximering med medelvärde för månaden	
dag	Medelvärde för perioden	Avvikelse från verkligt värde
1	423,6	88%
2	423,6	83%
3	423,6	111%
4	423,6	419%
5	423,6	321%
6	423,6	135%
7	423,6	178%
8	423,6	138%
9	423,6	103%
10	423,6	535%
11	423,6	157%
12	423,6	217%
13	423,6	110%
14	423,6	101%
15	423,6	116%
16	423,6	87%
17	423,6	152%
18	423,6	136%
19	423,6	82%
20	423,6	106%
21	423,6	118%
22	423,6	91%
23	423,6	74%
24	423,6	157%
25	423,6	88%
26	423,6	100%
27	423,6	71%
28	423,6	127%
29	423,6	127%
30	423,6	70%
31	423,6	71%
	13131,6	113%