NTNU

Norwegian University of
Science and Technology

Range Anxiety Alleviation for Electric Vehicles using a Dual Extended Kalman Filter with a Nonlinear Battery Model

Bjørn Kvaale

Master of Science in Cybernetics and Robotics
Submission date: June 2017
Supervisor: Marta Maria Cabrera Molinas, ITK
Co-supervisor: Ole Jakob Sørdalen, Eltek AS

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

- Use a modular Extended Kalman Filter design for the entire chain of the battery characterization (i.e. battery Parameter Estimation and State of Charge estimation)
- Explore a method for alleviating range anxiety in electric vehicles using vehicle modelling equations
- Validate the range anxiety alleviation method by comparing to real vehicle range data
- Explore situations where the dual Extended Kalman Filter modular design has better State of Charge estimation performance compared with Coulomb Counting
- Discuss the results, conclude and suggest further work

Abstract

The purpose of this thesis was to find a solution to combat range anxiety in electric vehicles (EV). Range anxiety was defined as the fear that the driver will not reach their destination and is a major barrier to increase EV uptake worldwide. Parameter and state of charge (SoC) estimation for the Revolve NTNU battery cell were done using an enhanced self-correcting (ESC) battery model in combination with a dual Extended Kalman Filter (EKF). Next, a method was developed to extract high resolution elevation data from a given Google Maps route. This elevation information was then attached to three different speed profiles and passed as an argument into an algorithm which calculated a SoC estimate. The algorithm used vehicle modelling equations along with the dual EKF to estimate SoC, as well as extrapolated range and time to go until battery pack depletion. Range results comparing the simulation to a real Chevy Volt using the same speed profiles showed a 23% error. The validation of the range algorithm required a different battery cell, which made it impossible to use the dual EKF tuned to the Revolve cell for SoC estimation. Instead, coulomb counting (CC) was implemented in SoC estimation for the validation section. In addition, tests using three different elevation profiles concluded that topography had a major influence on extrapolated range. Lastly, different situations were observed where the EKF had superior SoC estimation performance compared with CC. More focus on speed profile generation will enable future EV users to use their GPS to estimate range more accurately and thereby help alleviate range anxiety.

Sammendrag

Denne oppgaven har som formål å finne en løsning på rekkeviddeangst i elektriske kjøretøy. Rekkeviddeangst er definert som redselen for å ikke nå destinasjonen grunnet at batteriet går tomt for strøm. Dette er en betydelig barriere som må overkommes for å øke prosentandelen av elektriske kjøretøy på veien i mange deler av verden. Parameter og ≪State of Charge≫ (SoC) estimering har blitt gjennomført på en battericelle fra Revolve NTNU ved hjelp av et dual ≪Extended Kalman Filter≫ (EKF) med en ≪Enhanced Self-Correcting≫ (ESC) batteri modell. Videre ble en metode utviklet for å eksportere nøyaktig høydeprofil data fra Google Maps kjøreruter. Denne dataen ble brukt med tre ulike hastighetsprofiler i en algoritme som estimerte SoC, ekstrapolert rekkevidde og tiden til batteripakken var tom. Algoritmen brukte modelleringsligninger til et elektrisk kjøretøy, i tillegg til dual EKF til å estimere SoC. En 23 % feil ble utregnet ved å sammenligne simulert med ekte rekkevidde for en Chevy Volt. Ved validering av algoritmens rekkevidde ble en annen battericelle benyttet. Siden dual EKF er innstilt til Revolve cellen var det ikke mulig å bruke dual EKF til SoC estimering og ≪Coulomb Counting≫ (CC) ble brukt istedenfor. Videre viste flere simuleringer at høydeprofil har stor innflytelse på estimert rekkevidde. Avslutningsvis ble ulike scenarier testet, der en EKF hadde mer nøyaktig SoC estimeringer enn CC. Større fokus på utvikling av hastighetsprofil vil gjøre det mulig for sluttbrukeren å taste inn en destinasjon på GPS og få et nøyaktig estimat av rekkevidde.

Preface

I would like to thank Prof. Marta Molinas, Dr. Ole Jakob Sørdalen from Eltek AS and Dr. Gregory Plett for their immense contributions to this master thesis. In addition, I would like to thank Ørjan Gjengedal and Eivind Klokkehaug for their help with the battery testing. It was a pleasure to be able to discuss results with Ørjan and increase my understanding about batteries. I also want to express my deepest gratitude to my girlfriend Madeleine and my family for their support and feedback throughout the writing of this thesis. Lastly, I greatly appreciate that Prof. Johannes Windeln was able to take the time to read through my thesis and give constructive feedback. It's been an amazing and challenging five years at NTNU.

Bjørn Kvaale
Trondheim, June 2017

Contents

List of Tables

List of Figures

Abbreviations

CC	=	Coulomb Counting
EKF	=	Extended Kalman Filter
ESC	=	Enhanced Self-Correcting
EV	=	Electric Vehicle
HWFET	=	Highway Fuel Economy Test
ICE	=	Internal Combustion Engine
NYCC	=	New York City Cycle
RPM	=	Revolutions per Minute
SoC	=	State of Charge
UDDS	=	Urban Dynamometer Driving Schedule

Chapter 1

Introduction

1.1 Motivation For The Topic & Thesis Objective

Electric vehicles (EVs) can in general refer to hybrid EVs, where an electric motor is used in conjunction with an internal combustion engine (ICE) or battery electric vehicles (BEV), which only uses electric propulsion (Bonges and Lusk, 2016). This report only focuses on BEVs to reduce the vehicle modelling complexity.

With the goal of limiting global temperatures to 2^o Celsius, the Paris Declaration, the Electric Vehicle Initiative and the International Energy Agency see EV uptake as a major solution to limiting CO2 emissions. This assumes an increased uptake in renewable energies in the worldwide electricity grids (Cazzola et al., 2016).

Although EVs currently account for a small percentage of the global vehicle stock, there are positive trends that suggest EVs could become cost-competitive with ICE vehicles as soon as a decade. Policy support for EVs in countries such as Norway and the Netherlands has increased public uptake of EVs. In addition, costs for the battery pack, one of the most expensive components in EVs , have decreased drastically from 1000 USD/kWh in 2008 down to around 220 USD/kWh in 2015. Battery energy densities have more than tripled over the same period, leading to greater EV range for the same battery pack volume (Cazzola et al., 2016).

One of the main reasons that EV uptake has been slow worldwide is the issue of range anxiety. Range anxiety is defined as fear that an EV owner will not be able to reach his/her intended destination without having to recharge the EV battery (King et al., 2015). If it

is difficult for the potential customer to easily recharge the EV, the risks of buying an EV may outweigh the benefits (Bonges and Lusk, 2016). Range anxiety is a much bigger issue for EVs than ICE vehicles because EV range tends to be much lower.

The objective of this thesis is to alleviate range anxiety by using Google Maps to generate an elevation profile for a route and then use different speed profiles to extrapolate an estimated range and time to go until the EV battery pack is depleted. The ultimate goal in future work is for the EV owner to be able to input their destination into their GPS and have an algorithm generate a speed profile based on their driving style and the GPS route elevation profile. Then, this data can be used to estimate extrapolated range and thereby alleviate range anxiety.

1.2 State of the Art

1.2.1 Industry

Tesla, the innovative electric car maker, combats range anxiety through a combination of a big network of fast chargers and innovative software. By using a big network of fast chargers, Tesla owners are able to charge up their vehicles quickly when taking longer trips. According to (Tesla, 2017), Tesla EVs are able to charge from 10% to 80% in as little as 40 minutes. Tesla's range assurance software factors in parameters such as height changes, wind speed, weather and temperature (Tesla, 2015). It is unsure whether Tesla calculates a speed profile for the trip to determine the range of the vehicle, yet from the author's experience there is a final SoC estimate given for the destination. If Tesla does calculate a speed profile for a trip, this would be a very similar technique which is discussed more in this thesis. Given a start position and a destination, the GPS is able to find an optimal route with known speed limits. From this information, it is possible to create a speed profile for the entire trip (see Section 4.4). Tesla has also experimented with battery swapping to combat range anxiety, where the owner's empty battery pack could be swapped for a full battery pack in less time than it takes to fill up a gas tank. This program is not in operation anymore, as there was not enough demand for the service (Korosec, 2015).

1.2.2 Research

(Neaimeh et al., 2013) increases EV range by calculating a minimum energy path between the start point and the destination. This is achieved by using real data collected from EVs to make a linear model (i.e. an energy expenditure equation [kWh/km]). This

equation is then used together with traffic and road altitude data in Dijkstra's graph search algorithm to find a minimum energy route from the start point to the destination.

Many affordable, current generation EVs have less than 150 km of range. This range can become drastically smaller if air conditioning or heating is turned on. Therefore, much of the current research to combat range anxiety has to goal of increasing EV range. This can be achieved by using new batteries with higher energy densities, better vehicle charging or using better energy management systems that minimize energy use in vehicles to increase range (King et al., 2015). (King et al., 2015) focuses on using an on-demand car sharing solution, where the EV owner can loan an internal-combustion engine vehicle for longer trips. According to the study, the costs for such a car sharing solution would be low compared to current subsidies EV makers receive. A similar car-sharing model is being used by Fiat and Volkswagen.

(Fujimoto et al., 2016) proposes taking traffic signal information to optimize the vehicle velocity profile for minimal energy consumption at a traffic signal. This paper is implemented using an autonomous EV and the results show that the optimized velocity profile leads to a 15% reduction in energy consumption compared to a conventional driving profile.

1.3 Main Contributions

The main contributions to this thesis are:

- parameter and SoC estimation using a dual EKF on the Revolve battery cell
- an algorithm to calculate a high resolution elevation/grade angle profile from a Google Maps route
- the extrapolated time to go calculation until the battery pack reaches minimum rated SoC
- validation of extrapolated range algorithm by comparing to real dynamometer range results using a Chevy Volt in electric mode
- highlighting situations where the EKF has better simulated SoC estimation performance compared with CC

1.4 Summary of Previous Work

The enhanced cell correcting cell model combines four elements: the Open Circuit Voltage as a function of SoC, hysteresis, diffusion voltages (i.e. one or multiple RC parallel circuits) and linear polarization (i.e. R_0) to create an accurate battery model.

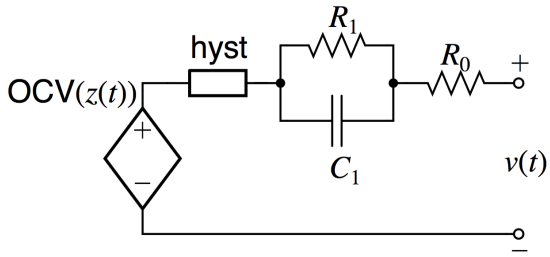


Figure 1.1: The ESC battery model models OCV as a function of SoC, linear polarization, diffusion voltages (i.e. warburg impedance), SoC-varying hysteresis and instantaneous hysteresis. It is referred to as the ESC model (Plett, 2015a).

The state equation is:

$$\underbrace{\begin{bmatrix} i_{R_1,k+1} \\ h_{k+1} \\ z_{k+1} \end{bmatrix}}_{\boldsymbol{x_{k+1}}} = \underbrace{\begin{bmatrix} A_{\mathrm{RC}'} & 0 & 0 \\ 0 & A_{H,k} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} i_{R_1,k} \\ h_k \\ z_k \end{bmatrix}}_{\boldsymbol{x,k}} + \underbrace{\begin{bmatrix} B_{\mathrm{RC}'} & 0 \\ 0 & (A_{H,k}-1) \\ -\dfrac{\Delta t}{Q} & 0 \end{bmatrix}}_{\boldsymbol{B}} \underbrace{\begin{bmatrix} i_k + w_k \\ \mathrm{sgn}(i_k + w_k) \end{bmatrix}}_{\boldsymbol{u_k} \text{ with process noise}} \tag{1.1}$$

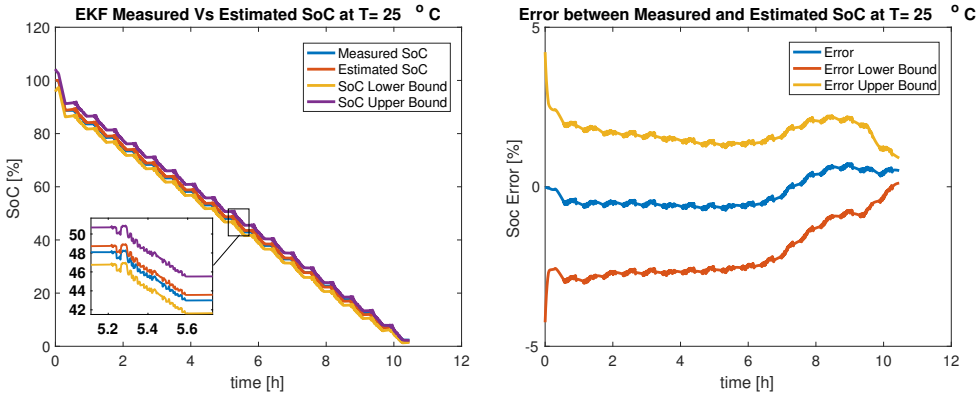
where i_{R_1} is the current over the R_1 resistor, h is a hysteresis voltage, z is the SoC, A_{RC} and B_{RC} are RC parallel circuit time constants, $A_{H,k}$ is an exponential constant and Q refers to the cell capacity. Δt is a time constant and i refers to the terminal voltage current (i.e. the current flowing into or out of the battery cell).

The output equation v(t) can be simplified to:

$$y_k = OCV(z_k, T_k) + M_0 s_k + M h_k - R_1 i_{R1,k} - R_0 i_k \tag{1.2}$$

where OCV(z_k,T_k) relates the open-circuit voltage to the SoC and temperature. The next two elements refer to instantaneous and SoC-varying hysteresis parameters and the last two elements are terms for the diffusion voltage and linear polarization voltage.

Next, two types of battery testing are done. The first test finds the relationship between OCV and SoC and the latter finds the dynamic parameters (i.e. the other parameters in the Figure above). The battery testing phase was completed during the specialization project, yet the parameter estimation phase to find the unknown ESC parameters was not completed. Parameter estimation data from a source was used as the input to an EKF, which calculated the SoC of the battery for the dynamic battery test. Results showed that the EKF could accurately predict SoC with a maximum mean error of 0.78% and a standard deviation of 0.46% (see Figure 1.2). Since parameter estimation was not yet completed for the static (OCV vs SoC) and dynamic values (hysteresis, diffusion voltages, internal resistance) using the Revolve NTNU battery cell, this step is completed during this thesis. The SoC results for the dynamic test cycle using the author's own parameter estimation results for the Revolve cell can be found in Section 4.2.



(a) Simulation results show the measured SoC in blue, the estimated SoC output from the EKF in orange and the predicted lower and upper bounds of the EKF SoC estimate in yellow and purple respectively. A description for finding the measured SoC can be found in (Kvaale, 2016). The bounds are found using (1.3) and magnification is done using (Fernandez-Prim, 2009).

(b) Simulation results show the SoC error in blue with upper and lower error bounds in yellow and orange respectively. The error bounds are found using (1.3).

Figure 1.2: SoC simulation results and error bounds

$$\hat{x}^{+}_{\text{k,Error Bounds}} = \hat{x}^{+}_{k} \pm 3\sqrt{\text{diag}\left(\sum^{+}_{\tilde{x},k}\right)} \tag{1.3}$$

The diagonal elements in the covariance matrix in Equation (1.3) represent the standard deviation values of the estimated states. It is quite certain that the "true" states are within three standard deviations of the estimated states. See (Kvaale, 2016) for more de-

tailed information.

1.5 Structure of Thesis

The thesis first describes how the parameters of the Revolve battery cell model are found using battery testing and an EKF. Then, a second EKF is used to estimate the SoC for a dynamic battery test. Next, an explanation is given for how elevation data from a Google Maps route can be used and attached to different speed profiles. These speed profiles are then inputed as an argument to an algorithm that outputs the SoC and extrapolated range. The second EKF is used to estimate the SoC in the algorithm.

The results section first discusses the Revolve parameter and SoC estimation, before further exploring the elevation profile of a route using two different sets of elevation data. A comparison of different speed profiles is given and how regenerative braking affects extrapolated range. All of the results discussed up until now use the Revolve battery cell along with parameters from a 2013 Chevy Volt. Next, the extrapolated range results are compared to a real 2013 Chevy Volt using the Chevy Volt battery cell. Lastly, three cases are provided which highlight the advantages of the EKF vs CC for SoC estimation.

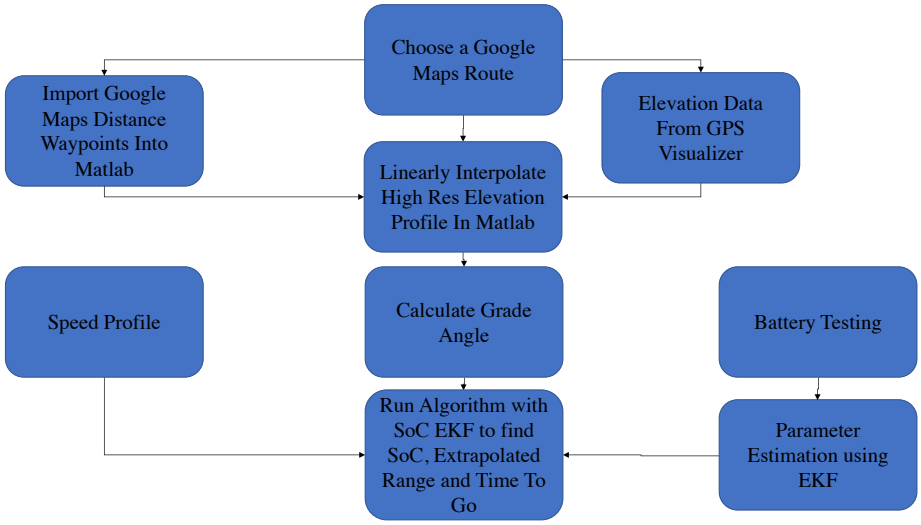


Figure 1.3: Flowchart for estimating extrapolated range

Chapter 2

Battery Model & Parameter Estimation

2.1 Estimating The Unknown Parameters Of The ESC Battery Model With Revolve Battery

The OCV of the battery cell shown as OCV(z(t)) in Figure 1.1 is a static function dependent on SoC and temperature. All of the other unknown parameters of the ESC battery model have a more dynamic relationship. Therefore it is useful to do two separate battery tests: one test to determine the static relationship (i.e OCV as a function of SoC and temperature) and another to determine the other unknown dynamic parameters (Plett, 2015a).

In the author's specialization project (Kvaale, 2016) last semester, parameter estimation was explained in detail. However, the author was not able to complete the parameter estimation phase for the Revolve battery cell. Instead, the estimated parameters for a different battery to the Revolve battery are used to estimate the SoC. Therefore, this section focuses on the main theory for the parameter estimation phase. For a more detailed description, see Reference (Plett, 2015a) and (Kvaale, 2016).

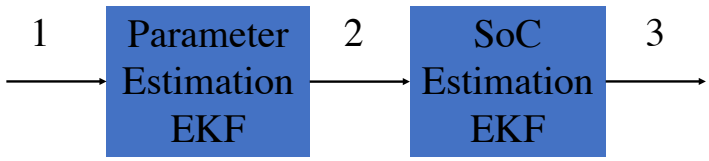


Figure 2.1: The blocks show the two EKFs that are used to estimate the unknown parameters and estimate the SoC of the battery cell. The inputs at point 1 are: voltage and current measurements from the dynamic testing, the OCV vs SoC relationship given as a lookup table and the estimated cell capacity. The main output at 2 is the estimated unknown parameters $\theta = \begin{pmatrix} \tau_{\text{RC}} & \gamma & M & M_0 & R_1 & R_0 \end{pmatrix}$ from the ESC model. The inputs at 2 to the SoC EKF are the coulombic efficiency η, the estimated capacity Q and θ. The output of the SoC EKF at 3 is the estimated SoC and the estimated voltage.

2.1.1 Cell Testing To Find the Dynamic Parameters

Static parameters (i.e. coulombic efficiency, OCV vs SoC relationship,cell capacity) can be found using the static tests described in (Plett, 2015a) or (Kvaale, 2016). For all unknown parameters from the ESC model that cannot be found from the OCV vs SoC relationship, a separate dynamic battery test is run. The current vs time profile of the test represents a similar load current to the final application the battery will be used for. The current profile should excite the entire SoC and temperature range of the cell (Plett, 2015a). A sample profile can be seen in Appendix 5.1. The real profile used is similar to the sample profile. The dual EKF (Plett, 2004) is discussed more thoroughly below, where one EKF estimates the parameters and the other EKF uses these parameters to estimate the SoC.

The dynamic test cycle should be completed at the same test temperatures as the OCV static test. With the battery fully charged to 100% SoC, discharge the battery to around 90% SoC using a constant current discharge. Run the dynamic profile with rest periods in between until the cell SoC is around 10%. Then discharge the battery to 0% SoC and charge the battery at a constant current back up to 100% SoC. This test is only done at 25^o in this report, but can be repeated for different temperatures. A more detailed description including testing scripts can be found in Section 2.11 of Reference (Plett, 2015a).

2.1.2 The EKF Approach For Finding Dynamic Parameters

In the EKF approach, the unknown parameters to be estimated from the ESC model are denoted by θ. In this case $\theta = \begin{bmatrix}\tau_{RC} & \gamma & M & M_0 & R_1 & R_0\end{bmatrix}$. γ, M and M_0 are hysteresis parameters, R_1 and τ_{RC} model diffusion and R_0 models linear polarization. Assuming the parameters change very slowly, they can be modelled as constant with a small deviation, shown in Equation (2.1). r_{k-1} is a white noise input that models the slow variation of the unknown parameters.

$$\theta_k = \theta_{k-1} + r_{k-1} \tag{2.1}$$

The output equation is given in Equation (2.2), where g is the ESC output equation and the white noise e_k models sensor noise and any modelling errors.

$$d_k = g(x_k, u_k, \theta, e_k) \tag{2.2}$$

The EKF algorithm shown in Algorithm 1 for parameter estimation is very similar to a regular EKF. First, the parameter predictions and error covariance matrices are updated. The output prediction refers to the estimated ESC output voltage in this paper (see Equation (1.2)). The linearized output matrix $\hat{C}_k^{\theta}$ requires recursive calculations that are further explained in Appendix 5.2. The $\hat{D}_k^{\theta}$ matrix is equal to one.

Next, the true output d_k is found by linearizing the output equation using a first order Taylor series. Since voltage measurement data from the dynamic parameter estimation test is available, this data is used for d_k instead of the approximated Taylor series. The Kalman gain is then calculated and the aposterior estimate $\hat{\theta}_k^+$ is updated using the Kalman gain and comparing the predicted voltage $\hat{d}_k$ with the measured voltage from the dynamic parameter estimation test. Finally, the error covariance matrix is updated and the algorithm repeats.

The final answer is given by the algorithm output $\hat{\theta}_{k=n}^+$. This answer gives the estimated unknown dynamic parameters of the ESC model. The output from the parameter estimation EKF is then inputed into the SoC EKF that outputs the SoC (see Figure 2.1). (Plett, 2016a) (Plett, 2004).

A full description for the SoC EKF is not given in this thesis, as it has been summarized clearly in (Kvaale, 2016). The general EKF code is appended in Appendix 5.3 for reference.

1: **for** $k = 1, 2, \ldots$ n **do**

2: Parameter prediction time update: $\hat{\theta}_k^- = \hat{\theta}_{k-1}^+$

3: Error covariance time update: $\sum_{\tilde{\theta},k}^- = \sum_{\tilde{\theta},k-1}^+ + \sum_{\tilde{r},k-1}$

4: Output prediction: $\hat{d}_k \approx g(x_k, u_k, \hat{\theta}_k^-, \bar{e}_k)$

5: Define $\hat{C}_k^\theta \triangleq \left.\dfrac{dg(x_k, u_k, \theta, e_k)}{d\theta}\right|_{\theta=\hat{\theta}_k^-}$

6: Define $\hat{D}_k^\theta \triangleq \left.\dfrac{dg(x_k, u_k, \theta, e_k)}{de_k}\right|_{e_k=\bar{e}_k}$

7: $d_k \approx g(x_k, u_k, \hat{\theta}_k^-, \bar{e}_k) + \hat{C}_k^\theta(\theta - \hat{\theta}_k^-) + \hat{D}_k^\theta(e_k - \bar{e}_k)$

8: $\sum_{\tilde{d}_k} = \hat{C}_k^\theta \sum_{\tilde{\theta}_k}^- (\hat{C}_k^\theta)^\top + \hat{D}_k^\theta \sum_{\tilde{e}} (\hat{D}_k^\theta)^\top$

9: Kalman gain: $L_k^\theta = \sum_{\tilde{\theta},k}^- (\hat{C}_k^\theta)^\top \left[\sum_{\tilde{d}_k}\right]^{-1}$

10: Parameter estimate measurement update: $\hat{\theta}_k^+ = \hat{\theta}_k^- + L_k^\theta(d_k - \hat{d}_k)$

11: Error covariance measurement update: $\sum_{\tilde{\theta},k}^+ = \sum_{\tilde{\theta},k}^- - L_k^\theta \sum_{\tilde{d},k} (L_k^\theta)^\top$

12: **end for**

Algorithm 1: General EKF For Parameter Estimation

Chapter 3

Modelling Electric Vehicle (EV) Range Anxiety

This section explains a novel way for SoC and range estimation. First, it is shown how you can download elevation data from a desired route on Google Maps. Then, this elevation data can be attached to three different speed profiles. One speed profile models city driving with stop and go traffic, another models highway driving and the third models city driving with higher speeds than the first and more fluid driving (EPA, 2017). These three speed profiles are then loaded into an algorithm in Section 3.2. The parameter estimation data from Section 2 is used and transferred to the second EKF that outputs the SoC. From this estimated SoC, the range and time to go until the battery is depleted is extrapolated. This essentially assumes that the speed profiles are looped until battery pack depletion.

3.1 Google Maps Elevation Data For A Desired Route

By adding a starting point and endpoint in Google Maps, it is possible to get accurate directions with distance information. Unfortunately, Google Maps does not let you export elevation information, which is important to get a realistic and accurate SoC estimate. By using GPSVisualizer (Schneider, 2016), it is possible to insert the Google Maps URL of a search. GPS Visualizer will then output a text file with information regarding the distance and elevation data. The text file can then be imported into Matlab. To increase distance and elevation data resolution, waypoints from the Google Maps route are added. A sample

waypoint profile can be seen in Figure 3.1, where the waypoints refer to the points where the car changes direction. These waypoints are then printed by pressing the ”print text only” button. This then opens up a new page with directions and the waypoint distances, where distance information is located. By saving this file as a PDF and converting it to a text file using (Marketing, 2017), it is possible to import the distance data into Matlab using a function developed by the author. Then, the sample waypoint distances are linearly interpolated from the GPS Visualizer distance and elevation data to get higher resolution elevation data at each waypoint distance.

The algorithm in Section 3.2 requires grade angle information. To calculate the grade angle, take the arctangent of rise over run over each iteration of the elevation profile (see Figure 3.3 for a sample profile). These grade angle points are then attached to three different speed profiles explained in Sections 3.2.1 and 4.4. The length of the grade angle points is then either shortened or made longer to be equal to the speed profile points. This is necessary to be able to run the algorithm in Section 3.2. If the grade angle points are shortened, some grade angle information is lost. If the length is increased, the grade angle points are repeated in a loop. The algorithm in Section 3.2 then outputs the extrapolated range.

This process is repeated for three different elevation profiles shown in Figure 3.2 to test the validity of this method. Profile 3.2a is referred to as the downhill profile, profile 3.2b as the mixed profile, and 3.2c as the uphill profile.

The distances in the elevation profiles do not matter in this test, as the grade angle information length is changed to the same length as the speed profiles. This is an assumption that is not so realistic, but is currently needed to be able to use the grade angle points in the algorithm to find the extrapolated range. A better method will be constructed in future work.

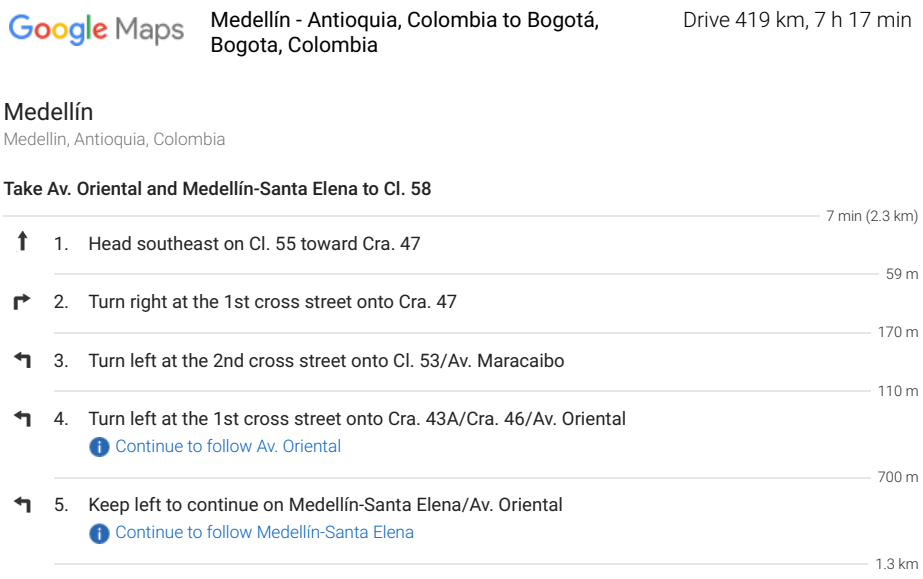
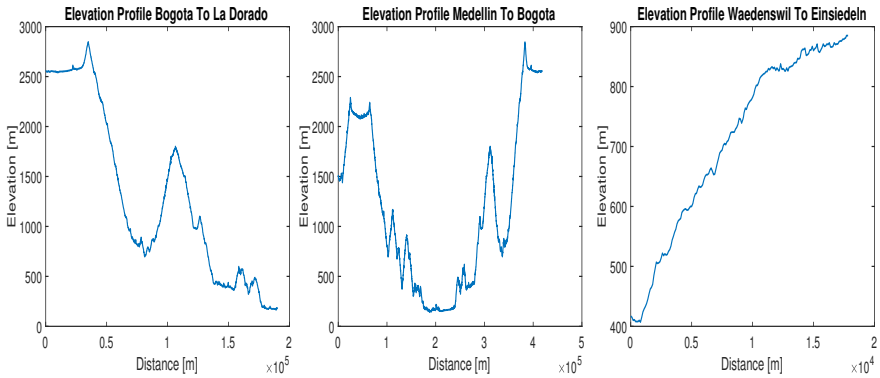
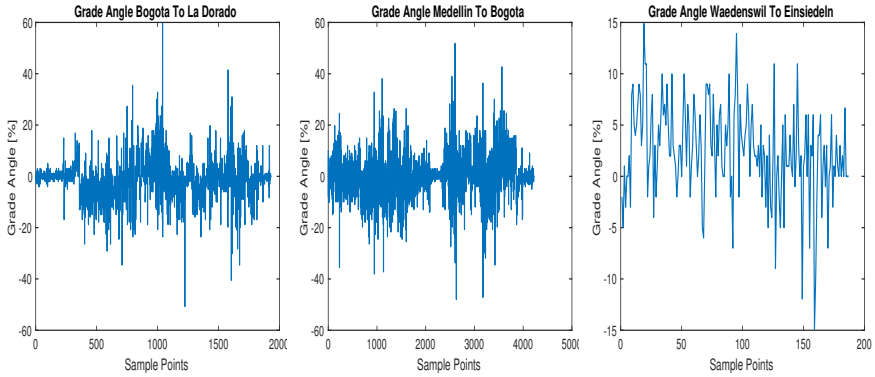


Figure 3.1: This figure shows sample waypoints from the mixed profile in Figure 3.2b.



(a) Downhill **(b)** Mixed **(c)** Uphill

Figure 3.2: The elevation profile for three different routes found using Google Maps is shown. Three different routes with different elevation profiles are chosen to test the validity of this method described in Section 3.1.



(a) Downhill **(b)** Mixed **(c)** Uphill

Figure 3.3: The grade angle information for the three elevation profiles is calculated. A few of the grade angle points are unrealistic, yet the mean and standard deviation values are realistic by comparing the values to the steepest road in the world with a 35% grade angle (Information, 2017).

3.2 Speed Profile To Battery SoC Vehicle Modelling

This chapter is based on sources (Plett, 2016a) and (Gillespie, 1992) and the author's results are compared with results of Gregory Plett given in source (Plett, 2016b). In all of the calculations in Figure 3.5, values for the current time step are calculated using data from the previous time step. These calculations assume that the battery is always able to supply the necessary power for the motor to achieve the limited torque.

For this section, estimated coefficients are assumed to be from a 2013 Chevy Volt. Instead of using the Volt battery cell, the Volt uses the Revolve NTNU battery cell. The estimated capacity for the Revolve battery cell is found in Section 2.1 and shown in Table 3.2. Although the Chevy Volt is a hybrid, only the electric mode is turned on. This makes the Chevy Volt a low-range EV.

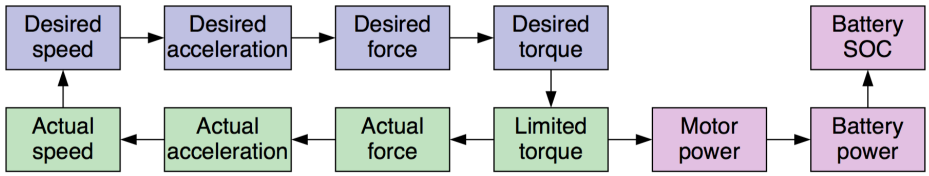


Figure 3.4: This figure shows the simulation flowchart calculations. Section 3.2.1 shows how to get from desired speed to actual speed and Section 3.2.2 shows the calculations from motor power to battery SoC (Plett, 2016a).

3.2.1 Desired Speed To Actual Speed Calculations

Three different speed profiles are used to get from the desired to the actual speed. The New York City Cycle (NYCC) cycle models low speed city driving with stop and go traffic. The Highway Fuel Economy Driving Schedule (HWFET) represents highway driving with a top speed under 96 km/h. Lastly, the Urban Dynamometer Driving Schedule (UDDS) models city driving with higher speeds than the NYCC cycle (EPA, 2017). See Figures 4.6, 4.7 and 4.8 for the real speed profiles.

From the desired drive speed profile shown in Figure 3.5, it is possible to sample the desired speed every second for example. Looking at Equation (3.1), the actual speed is the speed of the EV at the previous timestep. The desired speed is then equal to the current value to be sampled in Figure 3.5. For example, assume a sampling period of one second and a current time of t=6 seconds. The actual speed is then the speed at t=5 seconds and the desired speed is the value sampled from Figure 3.5 at t=6 seconds. Equation (3.1) then gives us the desired acceleration for the current sampling time.

Chevy Volt Coefficients	Value
N	12
Motor Inertia	0.2 kg·m^2
Gearbox Inertia	0.05 kg·m^2
Number of Wheels	4
Wheel Inertia	8 kg·m^2
Wheel Radius	0.35 m
ρ_{air}	1.225 kg/m^3
frontal area	1.84 m^2
C_d	0.22
C_r	0.0111
brake drag	0 N
rated max torque	275 Nm
regen fraction	0.9
wheel radius	0.35 m

Table 3.1: Chevy Volt Coefficient Values for Section 3.2.1. Some of these values are estimated, so they may not be that accurate. This could also influence the total accuracy of the actual speed calculation.(Plett, 2016b)

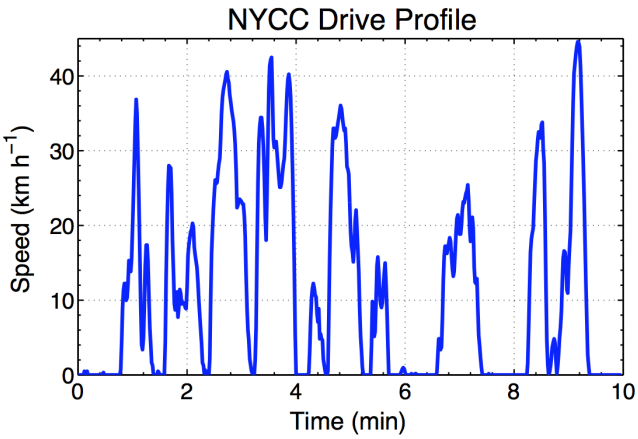


Figure 3.5: This figure shows a sample speed profile that a car should try to follow (Plett, 2016a).

$$\text{desired acceleration}[m/s^2] = \frac{\text{desired speed}[m/s] - \text{actual speed}[m/s]}{\text{sampling period [s]}} \tag{3.1}$$

The rotating equivalent mass can be found using Equation (3.3). It includes the rotating inertias from the motor, the gearbox and the wheels. Inertia refers to the phenomenon

that an object at rest will stay at rest and a moving object will continue moving (QRG, 2017). For example, a rotating wheel on an EV continues rotating unless some counter-acting force (e.g. road friction) slows the rotation. N refers to the gearbox ratio and is a unitless number given by Equation (3.2). RPM refers to the the number of full revolutions per minute.

$$\text{N} = \frac{\text{motor RPM}}{\text{wheel RPM}} \tag{3.2}$$

The motor is connected to the wheels via the gearbox. Since the gearbox inertia is measured at the motor side of the gearbox and not the output side, the motor and gearbox inertia's are multiplied by N^2. The wheel inertia is not multiplied by N^2 because the wheels are on the output side of the gearbox. Since the tire is flattened due to the weight of the car and the contact with the road, this must be taken account in the wheel radius.

$$\begin{aligned}&\text{rotating equivalent mass [kg]}\\ &= \frac{1}{(\text{wheel radius [m]})^2}\big(\big(\text{motor inertia}[\text{kg m}^2] + \text{gearbox inertia}[kg \cdot m^2]\big) * N^2\\ &+ \text{number of wheels} \cdot \text{wheel inertia}[\text{kg m}^2]\big)\end{aligned} \tag{3.3}$$

The equivalent mass is a combination of the maximum vehicle mass and the total rotational mass of the wheels. The maximum vehicle mass is used to see what the worst case performance of the vehicle is. This can be seen in Equation (3.4).

$$\text{equivalent mass [kg]} = \text{max vehicle mass [kg]} + \text{rotating equivalent mass [kg]} \tag{3.4}$$

Using Newton's 2nd law, the total force required by the tires on the road surface can be found from the equivalent mass and desired acceleration shown in Equation (3.5).

$$\text{desired acceleration force [N]} = \text{equivalent mass [kg]} \cdot \text{desired acceleration}[m/s^2] \tag{3.5}$$

The desired acceleration force provided by the motor is not the only force acting on the EV. Other forces present are an aerodynamic drag force, a rolling force, a grade force and brake drag.

The aerodynamic drag force is given by Equation (3.6), where ρ_{air} refers to the air density and C_d is the drag coefficient.

$$\text{aerodynamic force [N]} = \frac{1}{2}\rho_{\text{air}}[\text{kg/m}^3] \cdot \text{frontal area}[\text{m}^2] \cdot C_d \cdot (\text{prior actual speed [m/s]})^2 \tag{3.6}$$

The rolling force is given in Equation (3.7) and only exists if the car is moving. The max vehicle mass is again used to find the worst case performance of the vehicle. C_r is a unitless rolling friction coefficient and $g = 9.81$ m/s^2, the acceleration of gravity.

$$\text{rolling force [N]} = C_r \cdot \text{max vehicle mass [kg]} \cdot g[\text{m/s}^2] \tag{3.7}$$

The grade force refers to the force that gravity enforces on the vehicle if the vehicle is going up or down an incline. The grade angle in Equation (3.8) refers to the angle that the incline makes with the horizontal plane. A positive grade angle refers to an incline and a negative grade angle refers to a descent.

$$\text{grade force [N]} = \text{max vehicle mass [kg]} \cdot g[\text{m/s}^2] \cdot \sin(\text{grade angle [rad]}) \tag{3.8}$$

Brake drag is a constant force that includes frictional losses from the brakes or other losses that slow the vehicle down (see Equation (3.9)).

$$\text{brake drag [N]} = \text{constant value} \tag{3.9}$$

From this, it is possible to compute the demanded motor torque in Equation (3.10). The demanded motor torque is found from $\tau = F \cdot r$, where τ is the torque, F is the total force and r is the wheel radius. The motor torque is divided by N because the forces and the wheel radius are on the output side of the gearbox, whereas the motor torque is on the motor side of the gearbox.

$$\begin{aligned}\text{demanded motor torque [Nm]} = \frac{\text{wheel radius [m]}}{N}(&\text{desired acceleration force [N]} \\ &+ \text{aerodynamic force [N]} + \text{rolling force [N]} \\ &+ \text{grade force [N]} + \text{brake drag [N]})\end{aligned} \tag{3.10}$$

This demanded motor torque must then be compared with the vehicle specific maximum torque of the motor. If the demanded motor torque is positive (i.e. the vehicle is accelerating, motor is spinning in clockwise direction), then the maximum motor torque is found by looking at Figure 3.6.

If the prior actual motor speed is between 0 RPM and the rated RPM, then the maximum motor torque is given by the value of the straight line. In the case of Figure 3.6, the maximum motor torque is roughly 275 Nm. Otherwise, the prior actual motor speed is somewhere between rated RPM and maximum RPM. The actual motor speed is limited in Equations (3.19) and (3.20) to ensure the motor speed cannot exceed the maximum RPM value. Then the curved line for maximum motor torque is approximated using Equation (3.11). The rated max torque refers to the straight line in Figure 3.6 (i.e. 275 Nm), the rated motor speed refers to the rated RPM and the prior actual motor speed is the x-axis.

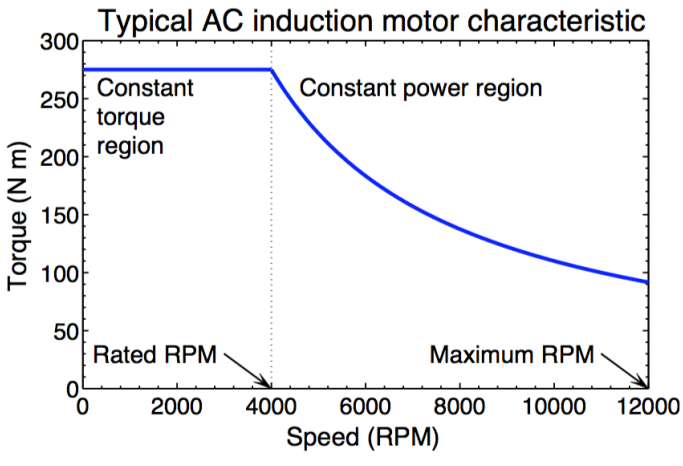


Figure 3.6: This figure shows the torque vs speed characteristics of a typical ideal 3-phase AC induction motor. The region from 0 to rated RPM is the constant torque region and the region from rated RPM to maximum RPM is the constant power region. (Plett, 2016a)

$$\text{max motor torque [Nm]} = \frac{\text{rated max torque [Nm]} \cdot \text{rated motor speed [RPM]}}{\text{prior actual motor speed [RPM]}} \tag{3.11}$$

If the demanded motor torque is negative (i.e. deceleration), the motor will help decelerate until it's maximum negative torque is reached. If more deceleration is required, then the brakes will help to decelerate the vehicle. The motor helps with deceleration because it is able to regenerate (i.e. regen) some of the braking energy and charge up the battery pack again.

The max motor torque must be computed like above using either the straight line or Equation (3.11). Use the absolute value for prior actual motor speed to find the max motor torque. The maximum regen motor torque can be found using Equation (3.12), where the regen fraction refers to an efficiency loss and the rated max torque is the straight line in Figure 3.6. It is important to remember that the max regen motor torque should be negative.

$$\begin{aligned}\text{max regen motor torque [Nm]} = -\ \min(&\text{max motor torque [Nm]},\\ &\text{regen fraction} \times \text{rated max torque [Nm]})\end{aligned} \tag{3.12}$$

Now the maximum motor torques are calculated for both acceleration and deceleration. The limited motor torque is found using Equation (3.13) and (3.14) for demanded motor torque ≥ 0 and < 0 respectively. The max motor torque is positive and the max regen motor torque is negative.

$$\text{limited motor torque} = \min(\text{demanded motor torque,max motor torque}) \tag{3.13}$$

$$\text{limited motor torque} = \max(\text{demanded motor torque,max regen motor torque}) \tag{3.14}$$

Now the actual acceleration force can be calculated using Equation (3.15).

$$\begin{aligned}\text{actual acceleration force [N]} = \text{limited motor torque [Nm]} \cdot \frac{N}{\text{wheel radius [m]}}\\ -\ \text{aerodynamic force [N]} - \text{rolling force [N]} - \text{grade force [N]}\end{aligned} \tag{3.15}$$

From this the actual acceleration can be found (see Equation (3.16)). The equivalent mass is defined in Equation (3.4).

$$\text{actual acceleration}[\mathrm{m/s^2}] = \frac{\text{actual acceleration force [N]}}{\text{equivalent mass [kg]}} \tag{3.16}$$

The calculated actual acceleration may lead to a greater angular velocity than the motor is rated for. Therefore a test speed is first calculated using Equation (3.17).

$$\text{test speed [m/s]} = \text{prior actual speed [m/s]} + \text{actual acceleration}[\text{m/s}^2] \cdot \text{sampling period [s]} \tag{3.17}$$

Then the motor test speed can be calculated using Equation (3.18). By multiplying the test speed on the wheel side by the gearbox ratio N, the test motor speed is found. The 60 [s/min] is a conversion factor and the denominator gives the circumference of the wheel for one revolution.

$$\text{motor test speed [RPM]} = \frac{\text{test speed [m/s]} \cdot N \cdot 60[\text{s/min}]}{2\pi \cdot \text{wheel radius [m]}} \tag{3.18}$$

To find the limited motor speed for motor test speed $>= 0$, use Equation (3.19). Otherwise, use Equation (3.20).

$$\text{limited motor speed [RPM]} = \min(\text{motor test speed [RPM]}, \text{max rated motor speed [RPM]}) \tag{3.19}$$

$$\text{limited motor speed [RPM]} = \max(\text{motor test speed [RPM]}, \text{-max rated motor speed [RPM]}) \tag{3.20}$$

The actual speed can then be calculated using Equation (3.21).

$$\text{actual speed [m/s]} = \frac{\text{limited motor speed [RPM]} \cdot 2\pi \cdot \text{wheel radius [m]}}{60[\text{s/min}] \cdot N} \tag{3.21}$$

From the actual speed, the distance travelled is estimated in Equations (3.22) and (3.23).

$$\Delta\text{x [m]} = \frac{\text{actual speed [m/s]} - \text{previous actual speed [m/s]}}{2} \cdot (\text{current time [s]} - \text{previous time [s]}) \tag{3.22}$$

$$\text{distance [m]} = \text{previous distance [m]} + \Delta x \text{ [m]} \tag{3.23}$$

3.2.2 Motor Power to Battery SoC Calculations

Chevy Volt Coefficients	Value
Overhead Power	0.2 kW
Battery Pack Efficiency	0.96
Inverter Efficiency	0.94
Motor Efficiency	0.95
Gear Efficiency	0.97
Revolve Battery Cell Estimated Capacity	6.8 Ah
max rated battery SoC	75 %
min rated battery SoC	25 %
Num Modules	12
Num Parallel Cells Per Module	3
Num Cells in Series Per Module	8
Num Cells Per Pack	288

Table 3.2: Chevy Volt Coefficient Values for Section 3.2.2. Some of these values are estimated, which could negatively impact the estimated SoC, extrapolated range and time to go until battery pack depletion. (Plett, 2016b)

The motor power can be calculated from the fundamental physics equation given in (3.24), where τ is torque, θ is the angle in Figure 3.7 and t is the time.

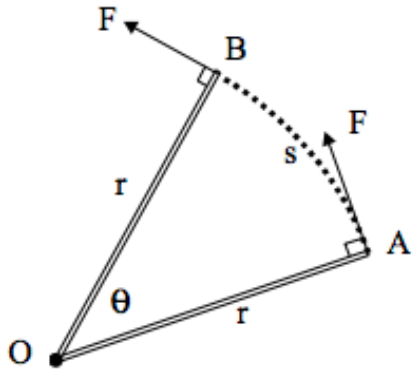


Figure 3.7: This figure shows how the constant force F rotates the lever arm with radius r. s is the circular arc and θ is the angle between two different timesteps as F pushes the lever arm around the axis of rotation. (Sleigh, 2016)

$$\text{power} = \frac{\phi \cdot \tau}{t} \tag{3.24}$$

The instantaneous motor power is computed in Equation (3.26), where $\theta = 2\pi \cdot$ motor speed and $\tau =$ limited torque.

$$\text{average lim motor speed [RPM]} = \frac{\text{previous lim motor speed [RPM]} + \text{lim motor speed [RPM]}}{2} \tag{3.25}$$

$$\text{motor power [kW]} = \frac{2\pi\text{[rad/revolution]} \cdot \text{average lim motor speed [RPM]} \cdot \text{limited torque [Nm]}}{60\text{[s/min]} \cdot 1000\text{[W/kW]}} \tag{3.26}$$

Positive motor power means the vehicle is accelerating and negative motor power means the motor is regenerating energy back to the battery.

For positive motor power, use Equation (3.27), where the overhead power refers to the power needed for other vehicle systems (i.e. air conditioning). Assume overhead power is a constant value. Even though the overhead power will vary in real life, it is assumed that the overhead power is insignificant compared to the motor power required to accelerate the vehicle. This assumption is proven correct by plotting the motor power values. The drivetrain efficiency looks at the efficiency from battery pack to the wheel output and is explained in Equation (3.28). The battery pack efficiency includes I^2R losses and the inverter converts direct current (DC) into alternating current (AC).

$$\text{battery power [kW]} = \text{overhead power [kW]} + \frac{\text{motor power [kW]}}{\text{drivetrain efficiency}} \tag{3.27}$$

$$\text{drivetrain efficiency} = \text{battery pack efficiency} \cdot \text{inverter efficiency} \cdot \text{motor efficiency} \cdot \text{gear efficiency} \tag{3.28}$$

For negative motor power, the equation for battery power is very similar to Equation (3.27), only that motor power is multiplied by drivetrain efficiency (Equation (3.29)). Assume that the drivetrain efficiency is the same value, regardless of whether the motor power is positive or negative.

$$\text{battery power [kW]} = \text{overhead power [kW]} + \text{motor power [kW]} \cdot \text{drivetrain efficiency} \tag{3.29}$$

It is assumed that the battery can always deliver enough power for the motor to produce the limited motor torque. Therefore, no limit is put on the battery power. This assumption is not true in practice, but will hold for these simulations because the battery is never fully discharged. This could become an issue if the speed profile is longer than the vehicle range.

In the parameter estimation phase, the relationship between SoC and Open Circuit Voltage (OCV) is found for the Revolve battery cell. Assuming the battery pack initial SoC is fully charged at 75 %, this value is used to find the OCV for the fully-charged vehicle. Assume then that the OCV is equal to the battery cell voltage. The battery pack voltage is then computed by multiplying the battery cell voltage by the number of cells in series in a battery module and by the number of modules in a battery pack. This is a rough assumption assuming that each cell has an equal voltage and therefore each module also has an equal voltage.

$$\begin{aligned}\text{battery pack voltage [V]} = \text{cell voltage} \cdot \text{num cells in series in one module} \\ \cdot \text{num modules in series in a pack}\end{aligned} \tag{3.30}$$

Then the battery current is found using $I = P/V$. A positive battery current is defined as a current discharge.

$$\text{battery current [A]} = \frac{\text{battery power [kW]} \cdot 1000\text{[W/kW]}}{\text{battery pack voltage [V]}} \tag{3.31}$$

The battery SoC is then computed using the SoC EKF. For a more complete description of how the EKF computes SoC, see Reference (Kvaale, 2016). For all "true" SoC estimates that assume perfect current and voltage measurements in Section 4.10, CC shown below is sufficient. The coulombic efficiency η is assumed to be equal to one.

$$\text{battery SoC [\%]} = \text{prior battery SoC [\%]} - \frac{\text{battery current [A]} \cdot \text{sampling period [s]}}{3600\text{[s/hr]} \cdot \text{battery capacity [Ahr]}} \cdot 100\% \tag{3.32}$$

By extrapolating the driving cycle calculations, the driving range can be found. This calculation loops the speed profile used until the minimum rated battery SoC is found.

$$\begin{aligned}&\text{range [km]} = \\ &\text{total distance simulated [km]} \cdot \frac{\text{max rated battery SoC [\%]} - \text{min rated battery SoC [\%]}}{\text{initial SoC [\%]} - \text{SoC at end of drive cycle [\%]}}\end{aligned} \tag{3.33}$$

The same calculation can be used used to find ”time to go” until the EV reaches the minimum rated SoC. Again, the speed profile is looped like in the range calculation above.

$$\begin{aligned}&\text{time to go [s]} = \\ &\text{total time simulated [km]} \cdot \frac{\text{max rated battery SoC [\%]} - \text{min rated battery SoC [\%]}}{\text{initial SoC [\%]} - \text{SoC at end of drive cycle [\%]}}\end{aligned} \tag{3.34}$$

Chapter 4

Results & Analysis

4.1 Parameter Estimation Results for the Revolve Battery

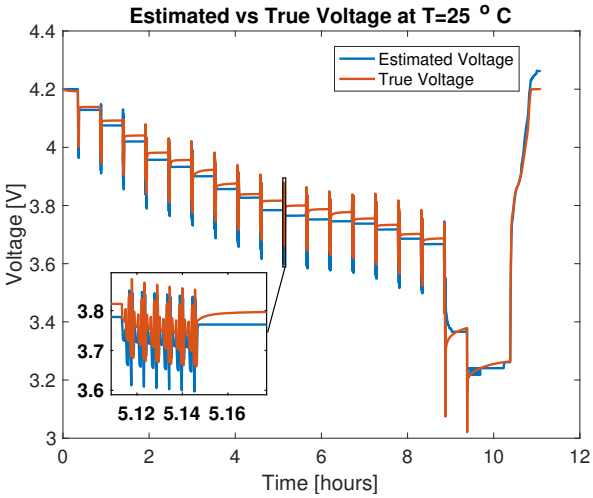


Figure 4.1: This figure shows how good the fit is of the estimated unknown parameters by comparing the measured true voltage with the estimated voltage. Equation (1.2) is used to calculate the estimated voltage in the SoC EKF. The test that is run is the same dynamic test described in Section 2.1.1. The current profile used is shown in Figure 5.1.

The results of the parameter estimation explained in Section 2.1 can be seen clearly in Figures 4.1 and 4.2. The simulated temperature is 25^o C. To test whether the parameters

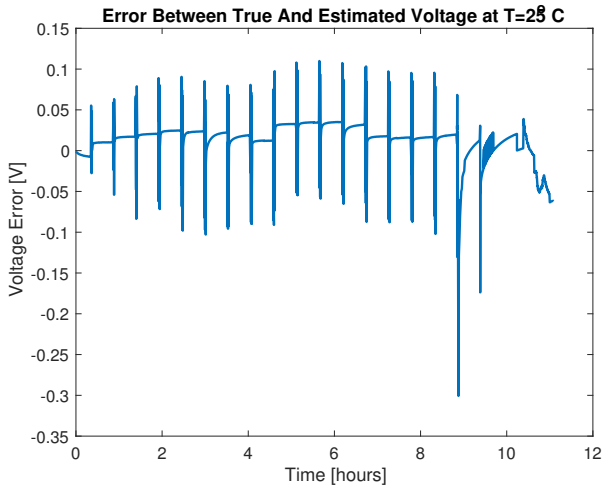


Figure 4.2: This figure shows the error between the true voltage and the estimated voltage shown in Figure 4.1.

	Author's Test	State of the Art Test
Mean Voltage Error	13.1 mV	-1.69 mV
Standard Deviation of Voltage Error	21.8 mV	11.4 mV
RMS Error	25.4 mV	11.5 mV

Table 4.1: The table compares the test run in this section with the current state of the art by looking at the mean voltage error, the standard deviation of the voltage error and the RMS error. Error is defined as the difference between the measured and estimated voltage.

give a good description of the ESC models, two tests are run. The parameter estimation EKF shown in Figure 2.1 returns values for the ESC unknown model parameters θ. These values are then fed into a separate EKF that estimates the battery cell SoC. In addition to outputing the SoC, the second EKF also outputs an estimated voltage for each timestep. This estimated voltage is given by the blue line in Figure 4.1. Assume the sensor and process noises $v_k = w_k = 0$ because the measured current i_k already includes measurement noise. OCV(z_k, T_k) is found using the theory provided in the first battery test and M_0, M, R_1 and R_0 are estimated in θ.

From Table 4.1, it can be concluded that the state of the art test performs better than the author's test. The battery testing repeats the dynamic current test cycle until the minimum voltage of 3 Volts is reached. Then, the battery is charged up to 4.2 Volts again using constant current charging. The true voltage of the last two voltage cycles in Figure 4.2 has a greater voltage span than previous cycles. This makes it more difficult for the EKF to estimate the voltage during this time span. It would be more ideal to have a constant

current discharge from around 10% SoC (i.e. 3.7 V) to the minimum battery voltage, as described in (Plett, 2015a). This could potentially lead to a smaller error between 8.5 and 10 hours.

Although this explains the voltage spike, the mean error in Table 4.1 is still much larger than the state of the art test. This error could potentially be attributed to an error in the author's parameter estimation algorithm that leads to suboptimal parameter values. Looking at Appendix 5.2, it can be seen that the recursive calculations to find the linearized matrix $\hat{C}_k^{\theta}$ is complex and could lead to errors. Another possibility is that the dynamic test run did not excite the dynamic parameters of the ESC model enough (see Appendix 5.1). Both of these possibilities are important for getting good values for the parameter estimation phase (Plett, 2004). Lastly, it could be possible that a sigma-point Kalman filter could give better parameter estimation results (Plett, 2016a).

4.2 ESC SoC Estimation Results With Revolve Battery

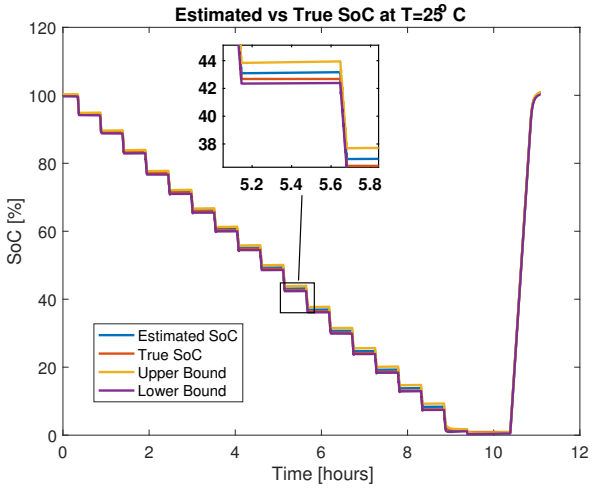


Figure 4.3: Simulation results show the estimated SoC in blue, the measured SoC output from the EKF in orange and the predicted upper and lower bounds of the EKF SoC estimate in yellow and purple respectively at T=25^o C. A description for finding the measured SoC can be found in (Kvaale, 2016). The upper and lower bounds are found using Equation (1.3). Magnification is done using (Fernandez-Prim, 2009).

The main results for SoC estimation are summarized in Figures 4.3, 4.4 & Tables 4.2 and 4.3. The dynamic current test cycle run is shown in Appendix 5.1 and repeated multiple times. The estimated SoC never leaves the upper and lower bounds & stays very close to the true SoC. The same applies for the error & error bounds. The error bounds

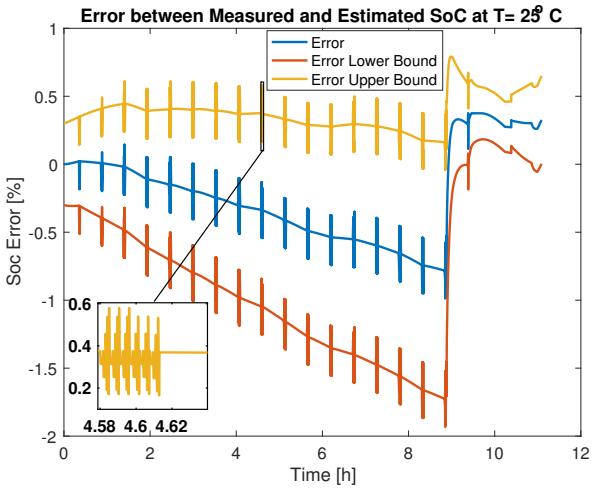


Figure 4.4: Simulation results show the SoC error (i.e. measured - estimated SoC) in blue with upper and lower error bounds in yellow and orange respectively at 25^o C. The error bounds are found using Equation (1.3). Magnification is done using (Fernandez-Prim, 2009).

Test	Temperature	Mean SoC Error	Standard Deviation of SoC Errors
Author's Test	25^oC	-0.22%	0.35 %
State of the Art	25^oC	-0.2 %	0.47 %

Table 4.2: (Dual EKF SoC error compared to state of the art

Real Cell Capacity [Ah]	Estimated Cell Capacity [Ah]
6.55	6.8

Table 4.3: Real vs Estimated Cell Capacity

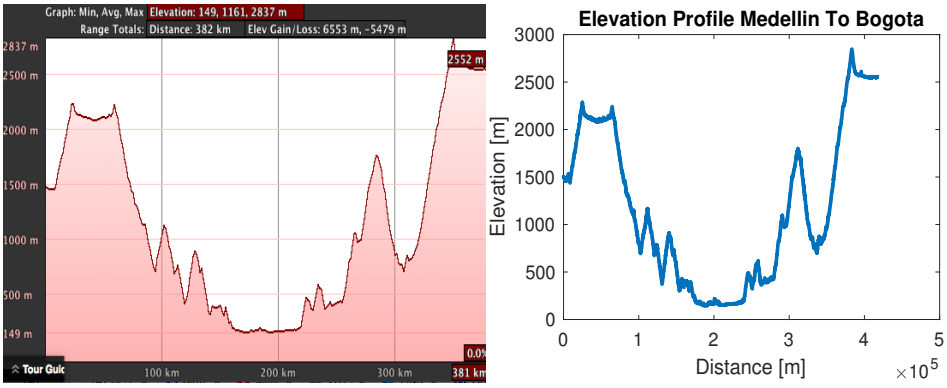
seem to show slight instabilities due to the current stimuli used in the parameter estimation section. Zooming in on the error bounds shows the same current profile shape used for parameter estimation, only inverted. This is because the CC equation used to find SoC in the EKF subtracts the current stimuli to find SoC, thereby inverting the SoC estimate. The pattern is then transferred to the error and error bounds.

Both the mean and standard deviation results in Table 4.2 show that the error bound is small compared to the state of the art. The state of the art test results are found using (Plett, 2015b). Since η and Q are the two main estimated parameters in the CC equation to find SoC, good estimates for these two parameters are found. Table 4.3 shows a 4% error between the real and estimated capacity. The Revolve battery cell data sheet reveals that the capacity must be greater or equal to 6.55 Ah. A constant current discharge from max to min rated cell voltage is used to estimate the cell capacity. The cell current used

to measure cell capacity is smaller in this thesis than the current the battery manufacturer uses. A smaller current discharge in the current discharge test leads to a higher estimated capacity (Dubarry et al., 2009), which validates the cell capacity estimate. In addition, both the SoC estimates and the SoC error estimates remain within the error bounds at all times (see Figures 4.3 and 4.4).

Although the parameter estimation results are not optimal (see Section 4.1), the estimates for η and Q are better. The reason is that the η and Q parameters are not found using the parameter estimation EKF, but rather by measuring accumulated Ah charged and discharged during the SoC vs OCV battery test (see Section 2.1).

4.3 Elevation Profile Results



(a) Google Earth Pro Results **(b)** GPS Visualizer Results

Figure 4.5: This figure shows the elevation profile for the mixed profile (see Figure 3.2) using two different methods.

Dist. GE Pro [km]	Elev. GE Pro [m]	Dist. GPS Visual. [km]	Elev. GPS Visual. [m]
58.8	2232	65	2240
223	476	246	490
284	1771	312	1800
361	2916	383	2846

Table 4.4: Peak elevation points are sampled for the two elevation profiles using Google Earth Pro and Google Maps/GPS Visualizer.

Figure 3.2 and Table 4.4 show the results of two different methods for checking the elevation profile for any route. This case compares the mixed route (see Figure 3.2) from Medellin to Bogota in Colombia. The profile in Figure 4.5a shows the elevation route from Google Earth Pro and compares it to the GPS Visualizer results that are used in this thesis. The Google Earth Pro results are found by inputting the starting point and destination, right clicking on the route and pressing "Show Elevation Profile". Looking at the elevation profiles visually show that the profile is almost identical for both. Taking a closer look at Table 4.4, the Google Earth Pro results do deviate from the GPS Visualizer results. GPS Visualizer uses NASA's Shuttle Radar Topography Mission (SRTM) and the United States Geological Survey National Elevation Dataset to calculate GPS data (Schneider, 2016). Google Earth Pro, on the other hand, uses data from multiple sources (e.g. SIO, NOAA, US. Navy, NGA, GEBCO). The results are very similar for the uphill and downhill elevation profiles.

SRTM data used in GPS Visualizer has an absolute elevation accuracy of 6.2 m comparing accurate ground measurements in South America to the SRTM data (Farr et al.,

2007). (Wing and Frank, 2011) uses GPS receivers to triangulate the elevation to an error of 5 meters in a forest canopy setting. The error between the Google Earth Pro and GPS Visaulizer data presented in Table 4.4 are much greater than these sampled errors and show that there could be an error during the linear interpolation of the elevation profiles. Both routes are inspected visually and it seems the routes used the same roads for most of the profile length. Yet the distance length is different for both elevation profiles (i.e. 381 km for Google Earth Pro, 419 km for Google Maps/GPS Visualizer), leading the author to believe that the best routes chosen by Google Maps and Google Earth Pro are slightly different. Many factors could lead to this deviation, including a different algorithm used for finding the best route or different GPS data. The Google Earth Pro elevation profile results are for some unknown reason not shown for the entire distance length, leading to a limitation in this elevation validation section. This could result in different elevation profile datapoints. Another point to mention is that it is unsure which of the two elevation profiles represent accurate true values, as none of the two profiles have been verified for precision and accuracy. Using mapping-grade GPS receivers or LIDAR connected to an airplane could give the two profiles a reference profile to validate against (Wing and Frank, 2011). A more specific analysis could be analyzed in future work.

4.4 Speed Profiles

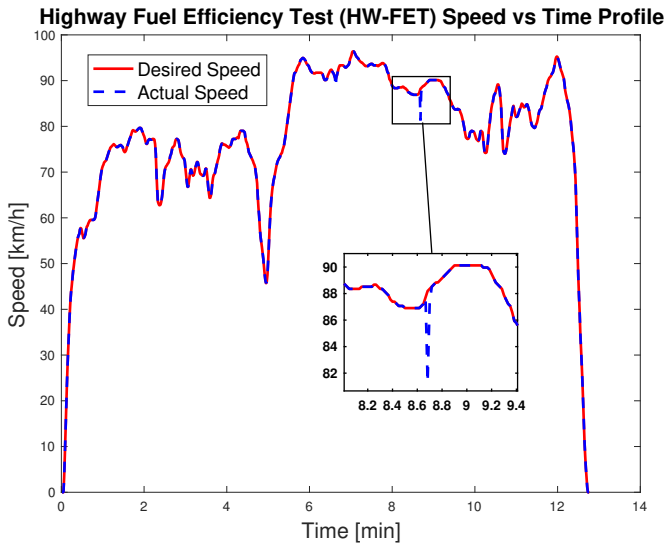


Figure 4.6: Simulation results show that the EV's actual speed is able to follow the desired speed for the HWFET speed cycle. The magnified region shows a case where the actual speed cannot keep up with the desired speed. The downhill elevation profile is used. Magnification is done using (Fernandez-Prim, 2009).

Figures 4.6, 4.7 and 4.8 show the main results for the desired vs actual speed profiles. To best understand the calculations required to get from desired speed to actual speed, see Figure 3.4 and Section 3.2.1. The vehicle that is used for the modelling purposes is a 2013 Chevy Volt with Revolve battery cells (see Tables 3.1 and 3.2 for coefficient values). The Chevy is able to follow the desired speed profiles mostly, except for the magnified region in Figure 4.6. This is due to a high grade angle that leads to a high grade force. The motor is not able to supply enough motor torque to sustain the desired speed in this region. If a flat elevation profile is used, the vehicle is able to follow the desired speed in the magnified region also.

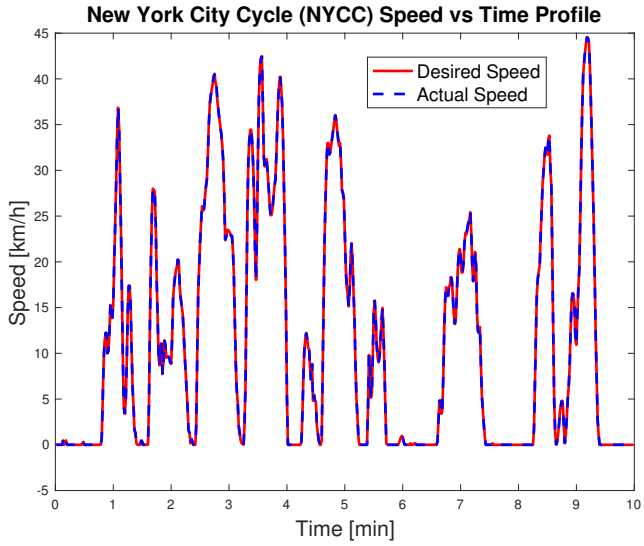


Figure 4.7: Simulation results show that the EV's actual speed is able to follow the desired speed for the New York City cycle. The uphill elevation profile is used.

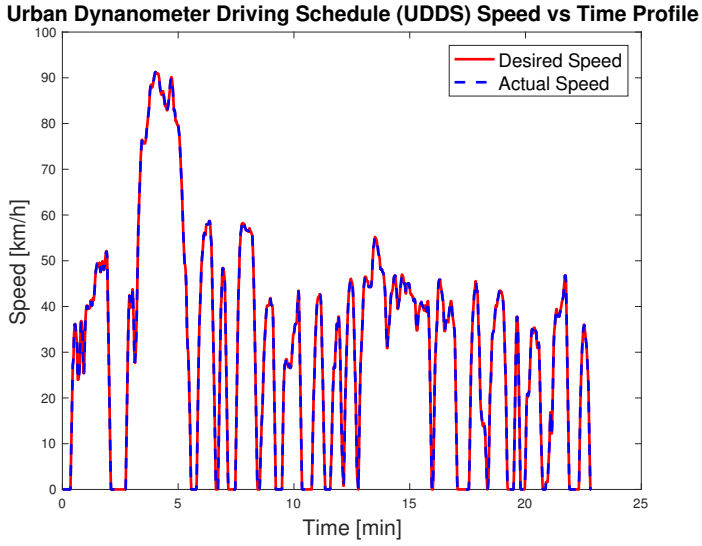
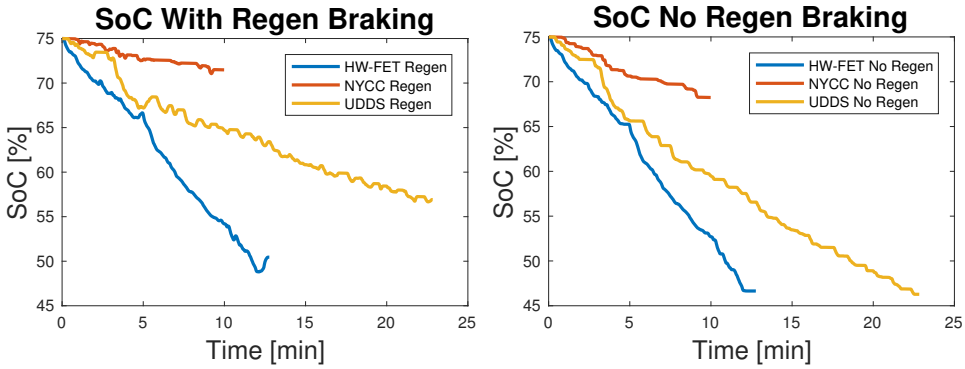


Figure 4.8: Simulation results show that the EV's actual speed is able to follow the desired speed for the UDDS cycle. The mixed elevation profile is used.

4.5 Speed Profiles SoC With And Without Regenerative Braking



(a) Regenerative Braking

(b) No Regenerative Braking

Figure 4.9: The figures shows how the different speed profiles shown in Figures 4.6, 4.7 and 4.8 affect the SoC of the battery pack. The SoC is calculated using the SoC EKF and Revolve parameter estimation data. The SoC increases at times in Figure 4.9a because the EV uses regenerative braking to recharge the battery pack. This simulation assumes a level surface (i.e. grade angle is zero) and the Revolve battery cell is used for SoC estimation.

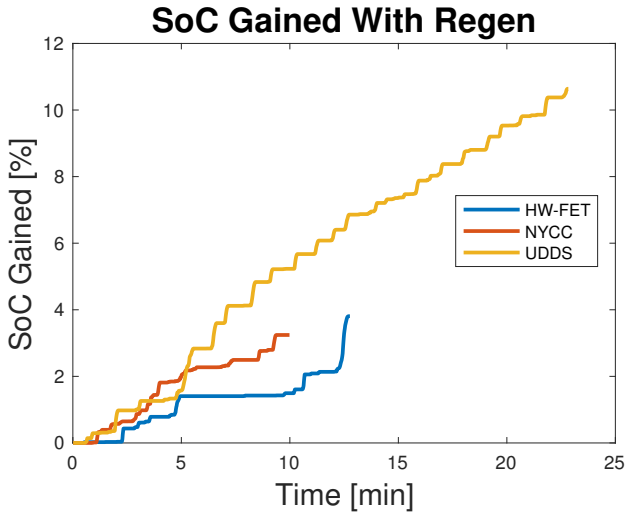


Figure 4.10: This figure plots the ”error” between Figures 4.9a and 4.9b. The plot shows how much SoC can be regenerated for the three different speed profiles using regenerative braking. The Revolve battery cell is used.

This section finds the SoC until the end of each speed profile and compares the effect

Speed Profile	HW-FET	NYCC	UDDS
SoC Lost With Regen [%]	24.5	3.5	18.1
SoC Lost No Regen [%]	28.4	6.8	28.7
SoC Gained With Regen [%]	3.9	3.3	10.6

Table 4.5: This table shows how much SoC is lost for three different speed profiles, assuming the EV has or does not have regenerative braking. The SoC lost is found by comparing the initial with the final values for all of the different scenarios in Figure 4.9. It is assumed that the Revolve battery cell is used in the battery pack.

regenerative braking has on SoC. The battery current is found from the limited torque (see Section 3.2.2). The no regenerative braking case is found by setting all negative battery currents (i.e. recharging currents) equal to zero. In this scenario, the battery is never charged up during deceleration. The main results are summarized in Table 4.5.

From the results presented in Figures 4.9a and 4.9b, one can conclude that the EV can handle the desired speed profiles with much range left to spare. Comparing the different speed profiles, the HW-FET profile discharges the battery the most. This is because the HW-FET profile has a much higher average speed than the others (see Table 4.6). Looking at Equation (3.6), the aerodynamic force is proportional to the actual speed squared. This leads to a much higher aerodynamic drag force for the HW-FET profile compared to the other profiles. Greater motor torque and more battery power are required to achieve the desired vehicle speeds. This requires a greater depth of discharge from the battery.

The NYCC profile discharges the battery the least due to the lowest average speed, the least amount of distance covered and the shortest driving time. The UDDS profile battery discharge is between the HW-FET and NYCC profiles because the values for average speed, distance covered and driving time are between the HW-FET and NYCC profiles.

The speed profiles that profit the most from regenerative braking with a flat elevation profile are the NYCC and the UDDS profiles (see Figure 4.10). Both these profiles have many deceleration sections, which leads to more regenerative braking opportunities. The HW-FET profile has fewer deceleration sections, but has one long deceleration at the very end to decelerate from 90 km/h to a full stop. As can be seen in Figure 4.10, the HW-FET profile regains little SoC during most of the profile and then increases dramatically during the end of the profile.

4.6 Comparison of Three Different Speed Profiles

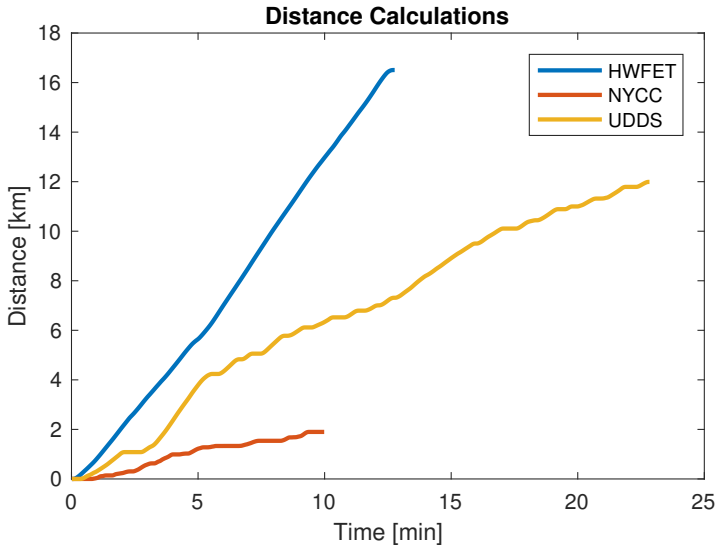


Figure 4.11: The simulated distance travelled is calculated for the three different speed profiles given in Figures 4.6, 4.7 and 4.8 using Equations (3.22) and (3.23).

Speed Profile	Average Speed [km/h]	Distance [km]	Range [km]	Time To Go [min]
HW-FET	77.6	16.51	33.6	26
NYCC	11.4	1.90	27	141.9
UDDS	31.5	11.99	33.2	63.1

Table 4.6: The total distance covered is calculated for the three different speed profiles given in Figures 4.6, 4.7 and 4.8 using Equations (3.22) and (3.23). The distance covered is the final value in all of the scenarios presented in Figure 4.11 and the extrapolated range is found using Equation (3.33). Time to go defines how much time remains until the EV reaches the minimum SoC if the same speed profile repeats itself (see Equation (3.34)). The Revolve battery cell is used in this calculation and a flat elevation profile is chosen (i.e. grade angle is zero).

The main results for this section are presented in Figure 4.11 and Table 4.6. Looking at the speed profiles in Figures 4.6, 4.7 and 4.8 and Table 4.6, it is clear that the HW-FET speed profile covers the greatest distance. The average speed of the HW-FET profile is more than two times greater than the UDDS profile and about seven times greater than the NYCC profile. The UDDS profile covers a greater distance than the NYCC profile due to a higher average speed and driving time.

Looking at the extrapolated range presented in Table 4.6, the HW-FET profile has a slightly longer range prediction than the UDDS profile. The NYCC profile has the lowest extrapolated range of the three speed profiles. Comparing the HW-FET profile with the

UDDS profile, one notices that the former profile has a higher average speed and less deceleration periods than the latter. Equation (3.6) shows that a higher average speed leads to a much greater aerodynamic force and requires more battery SoC to maintain the desired speed. Comparing the speed profiles with the required battery currents, one notices that quick acceleration periods require more battery current than more constant speed profiles.

Although the HW-FET profile has higher aerodynamic forces that act upon the EV than the UDDS profile, the UDDS profile has a greater degree of quick acceleration periods. Both aerodynamic forces and quick acceleration periods require more battery current and SoC to maintain the desired speed. Although deceleration periods regenerate some battery SoC, there are losses in the regenerative braking process. These are two possible reasons that the HW-FET and the UDDS profiles have similar extrapolated ranges. Also, an EV with greater battery capacity would lead to greater extrapolated range and most likely a more pronounced deviation between the three speed profile ranges.

By comparing the actual acceleration plots of the NYCC and UDDS cycles, one notices that the NYCC cycle requires quicker acceleration periods than the UDDS cycle. The HW-FET cycle requires more gradual acceleration periods than the NYCC and UDDS cycles. Since quicker accelerations require more pack current, this depletes more battery SoC. In addition, the NYCC profile regenerates the least amount of SoC of the three profiles. The NYCC profile has the lowest distance covered of the three speed profiles. This distance is used to calculate extrapolated range. Both quick acceleration periods, low regenerative braking capabilities and the lowest distance covered help explain why the NYCC cycle has the lowest extrapolated range of the three speed profiles.

The time to go calculations estimate how much time is remaining until the EV reaches the minimum battery pack SoC, assuming the speed profile repeats itself until the minimum SoC is reached. As can be seen from Table 4.6, the NYCC profile has the greatest time remaining. The UDDS profile has the next highest time to go and the HW-FET profile has the shortest time to go profile. The HW-FET profile length is 12.8 minutes, the NYCC profile is 10 minutes and the UDDS profile is 22.8 minutes. The two biggest components to calculate time to go are the profile length in minutes and the SoC value at the end of the speed profile. Since the NYCC profile loses the least amount of SoC by a factor over more than 5 compared to the other profiles, this helps the NYCC profile achieve the greatest time to go. The UDDS profile has a longer profile length and loses less SoC than the HW-FET profile. This helps explain why the UDDS profile has a longer time to go than the HWFET profile.

One also notices that that the range and average speed affect the time to go calculations by looking at Table 4.6. All three profiles have very similar extrapolated range until min rated SoC is reached. Since the HW-FET profile has more than double the average

speed of the UDDS profile and is a factor of 7 greater than the NYCC profile, it is logical that the HW-FET profile has the lowest time to go. By dividing the extrapolated range by average speed, one can find the time to go.

4.7 Elevation To Speed Profile To SoC

Figures 4.12, 4.13 and 4.14 showcase one of the main results of this thesis. By loading a Google maps profile, it is possible to download elevation data using GPS Visualizer, attach this information to a speed profile (i.e. UDDS,HW-FET,NYCC) and get a SoC estimate with known error bounds. The known error bounds can be calculated using Equation (1.3) and give more certainty that the SoC is within this region. The error bounds are not attached in Figures 4.12, 4.13 and 4.14 because it decreases the readability.

The author makes some crude assumptions to achieve these results. It is assumed that each battery cell in the battery pack has the same parameter estimation properties. Also, the battery cell voltage is found using only the relationship between OCV and SoC and neglecting the other ESC parameters (hysteresis, diffusion voltages & internal resistance)). The tests are done by adding white noise with a standard deviation of 1% to the simulated battery pack voltage and a constant bias of 1% to the simulated battery pack current.

Adding more noise to the EKF leads to instability, where the EKF oscillates between -5% and 105% SoC, similar to Figure 4.19a. The UDDS, HWFET and NYCC speed profiles used in Figures 4.12, 4.13 and 4.14 can be transformed into a current profile using Section 3.2. The reason for this unstable oscillation could be because the ESC model is parametrized using a different current profile than the current profiles used in Figures 4.12, 4.13 and 4.14. One of the limitations of the ESC model is that it is unsure how the cell behaves for different current profiles (Plett, 2015a). This uncertainty can affect the voltage estimate in the EKF, which can lead to instabilities in the SoC estimate. The SoC estimate oscillates between -5% and 105% because an upper and lower bound are set in the SoC EKF to prevent even greater instability.

The SoC results for Figures 4.12, 4.13 and 4.14 are discussed next. The extrapolated range calculations are based upon the estimated SoC from the EKF (see Equation (3.33)). Section 4.9 discusses that the extrapolated range calculations make sense compared to the true vehicle values because a smaller battery pack capacity and potentially less of the pack capacity is used. This hypothesis is verified by increasing the pack capacity. Therefore, if one assumes the extrapolated range is considerably accurate given the battery pack differences, the SoC estimates presented in this section must be reasonably accurate too. It is important to note that the extrapolated range verification using the bigger pack capacity uses CC because the dual EKF is only tuned to work for the Revolve battery.

Another important point to note is that the validation section assumes a flat grade angle is used. A more detailed analysis could be discussed in future work.

Comparing the three cases, it is noticeable after an in depth analysis that the speed and elevation profiles have a profound effect on SoC. As discussed previously in Section 4.6, the profile speed, the amount of acceleration and deceleration periods as well as the profile length influence the SoC estimate. Higher speeds and greater acceleration periods require more current discharge to combat the aerodynamic force and to make the car accelerate quicker. Deceleration periods lead to greater regenerative currents, increasing SoC. Section 4.8 comes to the main conclusion that an uphill elevation profile leads to less extrapolated range (less SoC lost) than a mixed or downhill elevation profile for the same speed profiles. This is due to a greater grade force (see Equation (3.8)) which decelerates the vehicle.

As the SoC is essential for estimating range and time to go until battery depletion, it is important to estimate SoC accurately. As discussed previously, the dual EKF is tuned using a different current profile to the current profiles tested in this vehicle. To verify the SoC results very accurately, one would need to do dynamic battery tests with the Revolve cell using the profile currents in Figures 4.12, 4.13 and 4.14. See Section 2.11 of Reference (Plett, 2015a) for further information on battery testing.

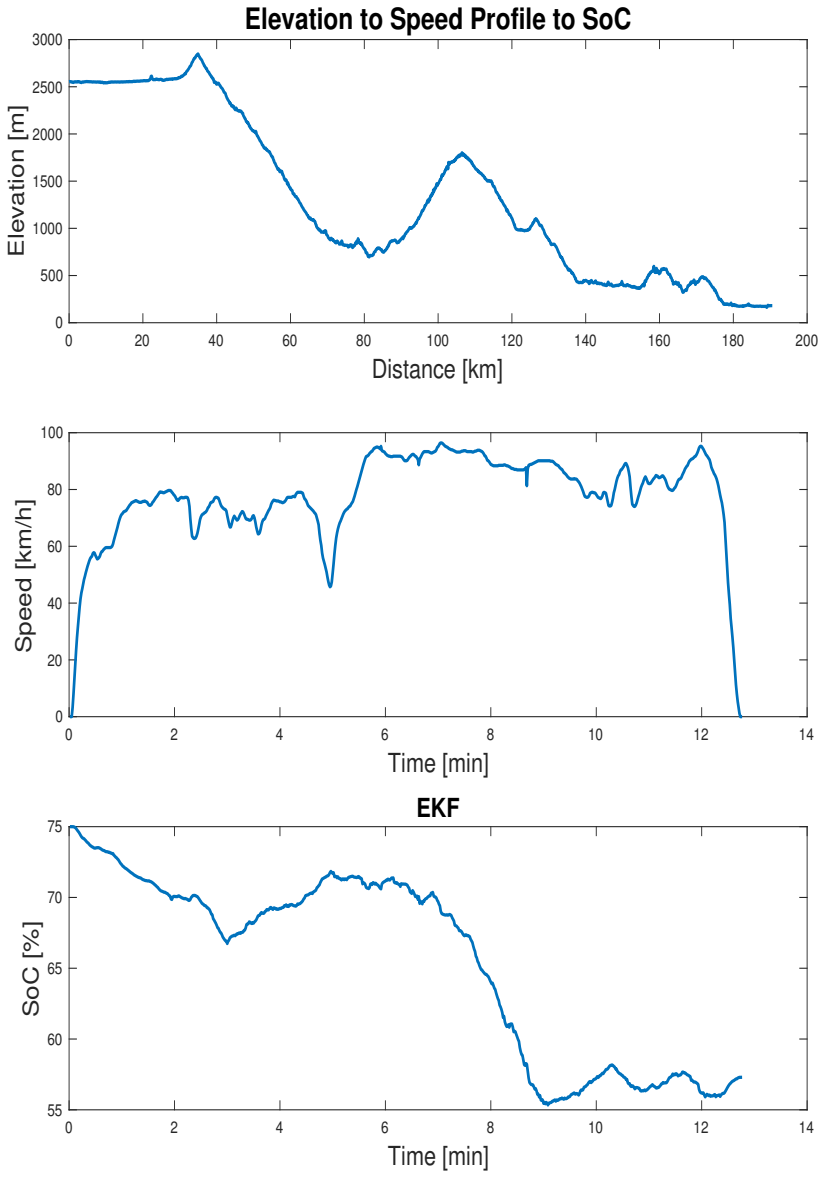


Figure 4.12: The downhill elevation profile grade angle points are attached to the HW-FET speed profile (see Section 3.1 for a description). Using the algorithm in Section 3.2 and running the algorithm through an EKF using the parameter estimation data in Section 4.1 leads to a SoC estimate.

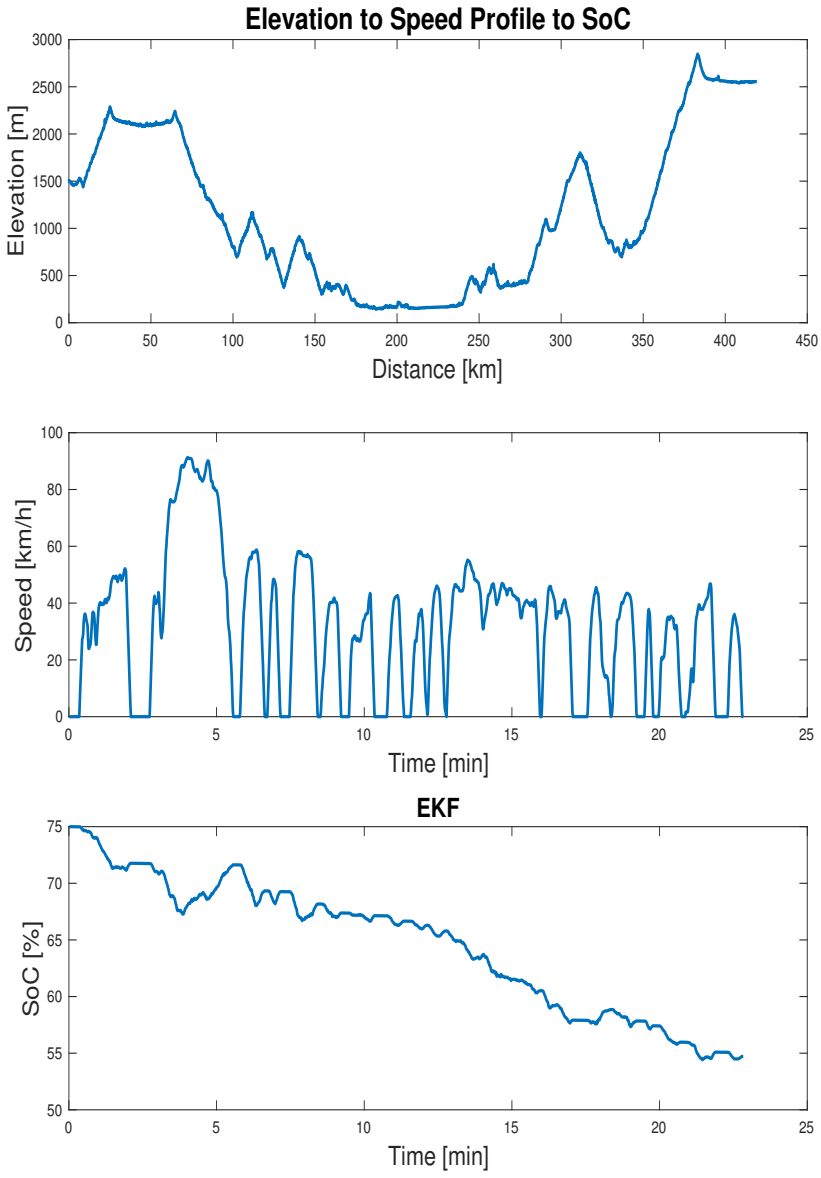


Figure 4.13: The mixed elevation profile grade angle points are attached to the UDDS speed profile (see Section 3.1 for a description). Using the algorithm in Section 3.2 and running the algorithm through an EKF using the parameter estimation data in Section 4.1 leads to a SoC estimate.

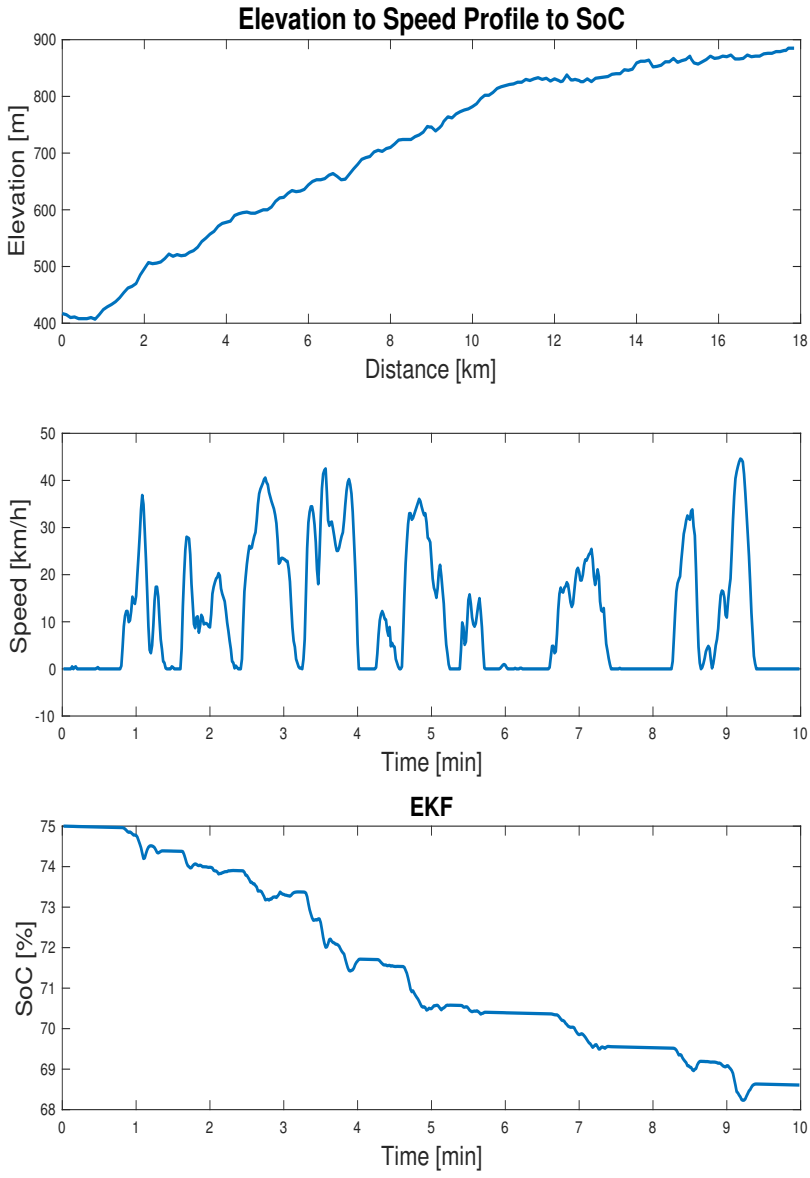


Figure 4.14: The uphill elevation profile grade angle points are attached to the NYCC speed profile (see Section 3.1 for a description). Using the algorithm in Section 3.2 and running the algorithm through an EKF using the parameter estimation data in Section 4.1 leads to a SoC estimate.

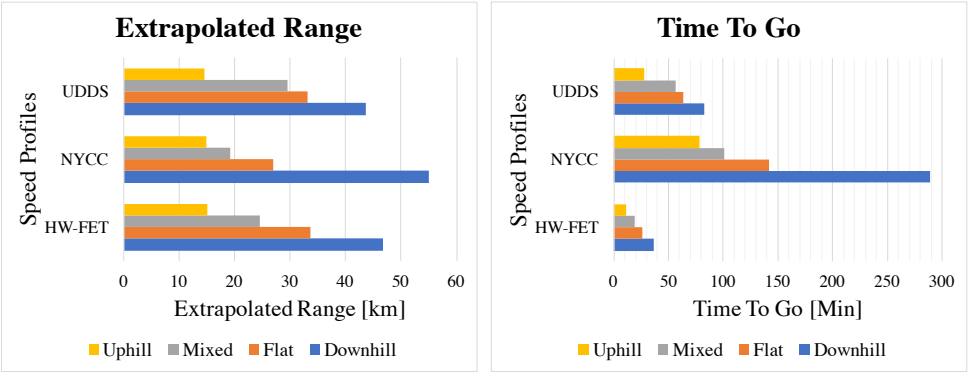
4.8 Speed Profile Extrapolated Range & Time To Go For Different Elevation Profiles

Est. Elevation Profile	HW-FET	NYCC	UDDS
Est. Range Downhill Revolve Cell [km]	46.8	55	43.6
Est. Range Mixed Revolve Cell [km]	24.6	19.2	29.5
Est. Range Uphill Revolve Cell [km]	15	14.8	14.5
Est. Range Flat Revolve Cell [km]	33.6	27	33.2

Table 4.7: This table compares the extrapolated range of three different speed profiles with three different elevation profiles. The EKF is used to estimate the SoC and extrapolated range. Speed profiles can be found in Section 4.4 and the elevation profiles in Section 3.1. The Revolve battery cell in a simulated 2013 Chevy Volt are used to find the extrapolated range. The flat results are the same as found in Table 4.6.

Elevation Profile	HW-FET	NYCC	UDDS
Time To Go Downhill [min]	36.2	289.2	83.1
Time To Go Flat [min]	26	141.9	63.1
Time To Go Mixed [min]	19.1	101	56.2
Time To Go Uphill [min]	11.6	78.1	27.5

Table 4.8: This table compares the extrapolated time to go of three different speed profiles with three different elevation profiles. The speed profiles can be found in Section 4.4 and the elevation profiles in Section 3.1. Time to go is extrapolated using the Revolve battery cell using Equation (3.34).



(a) Extrapolated Range

(b) Time To Go

Figure 4.15: The graphical representation of Tables 4.7 and 4.8 is shown.

From the results in Table 4.7, it is evident that the grade angle (i.e. elevation) has a large influence on driving range. The extrapolated range table above shows that the downhill profile has the longest extrapolated range followed by the flat, mixed and uphill

elevation profiles for all speed profiles. This is due to the fact that an uphill profile requires greater battery currents to be able to follow the three speed profiles, whereas a downhill profile makes great use of the regenerative braking capabilities. The mixed profile requires large discharge currents for the uphill and acceleration sections and allows for regenerative braking. The results for the flat profile in relation to the mixed and uphill profiles show that a lot of range is lost as soon as the car travels uphill. Even though the mixed profile has downhill sections with regenerative braking, this is not enough to increase range more than the flat elevation profile.

Looking more specific at the downhill elevation profile, one can see that the NYCC profile has the greatest extrapolated range compared with the HW-FET and UDDS profiles. This is an interesting result, as Table 4.6 shows that the NYCC profile has the smallest extrapolated range for the flat profile. From Equation 3.33, the main components in extrapolated range are distance travelled and the SoC lost. Since the distance travelled is smallest for the NYCC profile, it must be that the NYCC profile loses the least amount of SoC for the downhill profile. Looking at the simulated SoC gained from regenerative braking shows that both the NYCC and HW-FET profiles regenerate more SoC than the NYCC profile. As discussed in Section 4.5, the low average speed of the NYCC profile leads to a small aerodynamic force and less energy lost compared with the other profiles. Therefore, the NYCC profile does lose the least amount of SoC for the downhill elevation profile.

It is important to note that although the same elevation profiles are used for the different speed profiles, the length of the elevation profiles are adjusted to the length of the speed profiles. Therefore, the elevation profiles are either shortened (lose some grade angle data) or looped. This affects the extrapolated range and time to go calculation results.

Table 4.8 and Graph 4.15b show the main results for time to go. Graph 4.15b shows that the downhill profile has the greatest time to go, followed by the flat, mixed and uphill elevation profiles for all three speed profiles. This is the same pattern that is evident by looking at extrapolated range in Graph 4.15a. As time to go measures a very similar result compared to extrapolated range, it makes sense that this pattern is transferred to the time to go results.

4.9 Validation of Extrapolated Range For Flat Elevation Profile

Comparing this to real results from 2013 Chevy Volt in Table 4.9, the results are encouraging. After an extensive search, the author is unsure whether the real dynamometer testing uses a constant flat elevation profile during the entire test cycle. Therefore, it is assumed

Range [km]	HW-FET	NYCC	UDDS
Est. Range Chevy Volt 2013 [km]	75	60	74
True Range Chevy Volt 2013 Method 1 [km]	101	-	104
True Range Chevy Volt 2013 Method 2 [km]	61	-	61

Table 4.9: This table compares the extrapolated range of a simulated 2013 Chevy Volt with higher capacity battery cells to true dynamometer values using two different methods (DOE, 2013) (DOE, 2017). CC is used to find the estimated extrapolated range because the EKF is only adjusted to the Revolve battery cell. The speed profiles can be found in Section 4.4 and a flat elevation profile is used.

the elevation profile is flat for the whole duration of the test. A dynamometer measures engine power by placing the vehicle on a flat surface with rolls to measure the wheel output power. An explanation for finding the dynamometer and true driving values can be found in Appendix 5.4. Although Table 4.9 shows that the true range for the HW-FET and UDDS profiles are roughly two to three times (i.e. using Method 1 or 2) greater than the estimated range for the flat profile seen in Table 4.7, this is mainly due to a different battery pack capacity. The rated pack capacity for the true vehicle is 45 Ah (DOE, 2013), whereas the simulated Chevy Volt with Revolve cells has half of the pack capacity because the lower capacity Revolve battery cell is used.

To test whether this algorithm makes sense with the true values, the cell capacity is increased to 15 Ah, which increases the pack capacity to 45 Ah. Since each module has an assumed three cells in parallel, the pack capacity is the cell capacity multiplied with the number of parallel cells in a module (Plett, 2016a). Regular CC is used instead of the EKF to measure the SoC for the increased cell capacity. This is because the EKF uses parameter estimation data, which is only valid for the Revolve battery. Since none of the ESC battery model parameters are known apart from capacity for the real Chevy Volt battery cell, CC must be implemented. In addition, white noise on the current is turned off in the extrapolated range algorithm to simulate noise-free current signals. It is important to note that many coefficients in the simulated Chevy Volt vehicle are estimated, which could greatly influence the simulated current. This is discussed further below. Since the current and capacity are the two main components in CC, good current signals with a reliable capacity estimate lead to accurate SoC and extrapolated range estimates.

As can be seen from Table 4.9, the extrapolated range results for the Chevy Volt capacity are in between the true values of the Chevy Volt using Methods 1 and 2. Since Method 1 uses the rated pack energy from the battery manufacturer to calculate the range, it is unsure how accurate this value is (see Appendix 5.4). Method 2 uses range data given in miles, which makes these range values more believable. In addition, the results from Table 4.9 show that the Method 1 range results are 65% greater than the Method 2 range. Because of this, the estimated range will only be compared to the real range using Method 2.

The estimated range values for the HW-FET and UDDS profiles are greater than the real range using Method 2 in Table 4.9. Both estimated range values are approximately 23% greater compared to the true values. There are multiple reasons why this could be the case. Many of the coefficient values in Tables 3.1 and 3.2 are estimated, which can influence the extrapolated range. Two examples that influence range extensively are the min and max rated SoC and the aerodynamic force.

It is unsure what the max and min rated battery SoC values are for the real Chevy Volt (see Table 3.2). These values help prevent over-charge and over-discharge and define when the vehicle is fully charged or fully discharged. These values determine the real utilizable battery capacity and directly influence the true range . By decreasing these values to 30% and 70% for example, the estimated range is the same as the true range using Method 2 (see see Equation (3.33)). Also, the EPA simulates road load forces, such as aerodynamic drag, during the dynamometer testing. Each car manufacturer is allowed to decide how to measure the road-load force themselves (Bunker, 2015). After an extensive search, it is unsure how Chevy determines both the max and min SoC values and the road load forces (i.e. aerodynamic drag). Although there is a level of uncertainty regarding the estimated vehicle values in Tables 3.1 and 3.2, auto manufacturers have access to accurate vehicle parameters. By further improve the modelling equations and using accurate vehicle parameters, auto manufacturers will be able to increase the range estimation accuracy.

If a system similar to this were to be implemented on an EV, it would be very important to estimate a more conservative range. The current results show that the estimated range is greater than the true range, which would most likely result in EV owners being too optimistic about the range of their EV. Thereby, many EV owners not being able to reach their destination because of a depleted battery pack.

4.10 EKF vs CC for HW-FET and NYCC profile

The EKF used in this section is based upon code provided in (Plett, 2015b). The code is updated for increased author readability and understanding.

The parameter estimation data in Section 2 uses a dynamic current test cycle stimuli. The first EKF then finds estimated parameter values for the unknown ESC model values (see Section 1.4). These values are then inputed into another EKF that outputs the SoC (see Figure 2.1). The dual EKF method uses both current and voltage measurements, along with the ESC battery model to generate an estimated voltage. This estimated voltage is compared to the measured voltage in order to update the SoC estimate. CC, on the other hand, only uses current measurements to update the SoC estimate. Therefore, it is expected that the EKF can handle current measurement errors better than CC (Plett, 2016a). (Mastali et al., 2013), for example, shows a situation where the CC SoC estimate diverges drastically from the experimentally measured SoC and the EKF estimated SoC because CC only uses current measurements. This section presents three cases (i.e. a 50% standard deviation current sensor white noise, losing current sensor measurements and a 50% current bias) using two speed profiles where the EKF has better SoC estimation performance compared with CC. The current sensor errors occur after 6 minutes in all profiles. The voltage sensor simulation measurements are assumed perfect without any white noise.

It is assumed that each battery cell has the same behavior as the entire battery pack. Since the focus is on the EKF vs CC behavior and not realistic battery pack modelling, the author assumes this assumption is viable. The author realizes that each cell will have a slightly different behavior and different parameter estimation values in the real world. In addition, the SoC of each cell will be slightly different to one another in practice. It is also assumed that one current sensor is used to measure the battery pack. From the author's understanding, this can be done in practice (Xin et al., 2010).

In all of the figures in this section, it is presumed that the CC line without error has been initialized correctly to 100%. All of the cells in the battery pack are inferred to have the same SoC and each cell in the battery pack discharges equally. To truly initialize the battery pack SoC close to 100%, it would be possible to take take each cell and charge it up fully to T=25 ^oC, since this is often defined as 100% SoC from the battery manufacturer (Plett, 2015a). It is also assumed that CC without any current error has perfect current measurements and the Revolve cell capacity estimate is perfectly correct. Thereby, the CC without error line is defined as the "true" SoC. There will naturally be some errors if these tests are validated using experimental data. This could be a possibility to discuss in future work.

The rest of this section focuses on the results and analysis for the 50% white noise on

the current (Figures 4.17 and 4.18), losing current sensor measurements (Figures 4.19 and 4.20) and a 50% current bias (Figures 4.21 and 4.22).

The current sensor with 50% white noise and perfect voltage sensor measurements results are seen in Figures 4.17 and 4.18. As can be seen, both CC and EKF with the current sensor error are able to follow the true SoC well. The RMS SoC error for the HWFET profile is slightly smaller for CC than EKF (0.3% vs 0.5%). The opposite result can be seen for the NYCC profile (0.3% SoC error for EKF vs 1% SoC error for CC). It is reasonable that the EKF should have a lower error than CC, as the EKF is able to use the perfect voltage measurements to update the SoC estimate. Since the white noise is random, the author has also noticed situations where the EKF has a lower RMS error than CC for the HWFET profile. In the NYCC profile, a bias can be seen in the CC line with the current sensor error that is not noticeable in the HWFET profile. This is most likely because the NYCC is a more aggressive profile that requires higher battery pack currents. The 50% white noise then increases these currents even more compared with the HWFET profile, which results in a noticeable drift away from the true SoC. Looking at Figures 4.17b and 4.18b, the EKF estimated voltage is able to follow the simulated voltage without a problem. The HWFET voltage magnified region shows a drop at 50% SoC that is due to an error in parameter estimation. The simulated voltage is calculated using a lookup table of SoC vs Open-Circuit voltage. This lookup table has a drop very similar to this at 50% SoC .

Figures 4.19 and 4.20 assume a current sensor error at the beginning of the speed profile. Once the error occurs, the sensor outputs 0 A until the end of the test. Therefore, the CC line with the sensor error remains flat once the error occurs. The RMS error of the EKF vs CC SoC results for the HWFET profile is much better at 3.3% compared to 32.4% for CC. The same is true for the NYCC profile (i.e. 0.9% vs. 22%)

The EKF is able to follow the true SoC accurately with little RMS error after a transient response when the current error occurs in both speed profiles. Compared to the CC results, the EKF follows the true SoC more accurately. Figure 4.19a is able to follow the real SoC after 1.5 minutes, whereas Figure 4.20a has a 35 minute transient response to to adjust to the true SoC. Looking at the estimated voltage plots in Figures 4.19b and 4.20b shows that the EKF has an easier time of adjusting to the real voltage of the HWFET than the NYCC profile. This is most likely because the HWFET current profile is smoother and less aggressive than the NYCC profile. Since the current is found using the battery pack voltage, the pack voltage must also be smoother for the HWFET profile. Therefore, it is easier for the EKF to estimate the simulated voltage for the HWFET profile.

In Figure 4.19a, the EKF becomes unstable for a few minutes before converging to the 'true' SoC again. Looking at Figure 4.19b, one can see that the simulated pack voltage has a steep drop in the magnified region at around 50% SoC. This is due to the

same error as explained in the current sensor error with 50% white noise scenario above. The simulated voltage drop is enough to induce instabilities in the EKF voltage and SoC estimates because the current sensor error is so different from the true current. The reason the SoC oscillates between -5% and 105% is because a boundary is set for the estimated SoC value in the EKF to help with stability.

Figures 4.21 and 4.22 depict results for the 3rd case, where a 50% current bias is added to the current measurements. Both positive and negative currents are multiplied by 1.4 and the voltage measurements are still assumed perfect. Both SoC plots show that the EKF is able to follow the true SoC after a transient period. The CC with current error line, on the other, diverges dramatically away from the true SoC. The RMS error for both speed profiles is smaller for the EKF than for CC (16.2% vs 1% for HWFET and 11% vs 1% for NYCC). CC with error diverges downwards compared with the true SoC in both figures because there are more current discharges than regenerative charging portions in both speed profiles. Multiplying this net discharge profile by 1.5 results in a lower SoC compared with the true SoC. The RMS error for the HWFET estimated voltage is slightly smaller at 0.5 V than the NYCC profile at 0.6 V. This is most likely because the NYCC profile requires more aggressive battery currents, which results in more volatile battery voltages. The EKF then has more difficulty adapting to a quicker changing reference signal.

There is no noise added to the simulated voltage measurements, which helps to explain why the EKF is able to reacquire the SoC in Figures 4.17 - 4.22. White noise is added to the voltage measurements in separate tests, and this led to an almost instant instability in the SoC and voltage estimates in the EKF. This can be confirmed by calculating the eigenvalues of the EKF using Equation (4.1). The A and C matrices are the state equation matrices in Equation (1.1) and L refers to the Kalman gain. Looking at the SoC eigenvalue for the HWFET profile using the 0A current sensor error in Figure 4.16 shows that the eigenvalue is equal to one as soon as the current sensor error begins, which means the filter is unstable (Brown and Hwang, 2012). The EKF is also unstable for the NYCC speed profile using the 0A current sensor error. For the other two current error test, the EKF is barely stable for both speed profiles. This is logical, as the situations discussed are quite extreme and test the boundary stability conditions of the EKF. A more detailed description of Kalman filter stability can be further discussed in future work.

$$eig(A_{k-1} - L_k \cdot C_k \cdot A_{k-1}) < 1 \quad \Rightarrow \quad \text{EKF is stable} \tag{4.1}$$

(Brown and Hwang, 2012)

All three cases show extreme situations and thus it is very clear that adding voltage noise in addition to the current sensor error will result in a divergence from the true SoC. The ESC model used to estimate the voltage in the EKF has difficulties predicting accu-

rate voltages for current profiles different to the stimuli used in the parameter estimation testing phase (see Appendix 5.1). Both the NYCC and UDDS profiles are very different to the parameter estimation current stimuli and this could help explain the stability issues mentioned. Physics-based battery models could potentially lead to better results (Plett, 2015a).

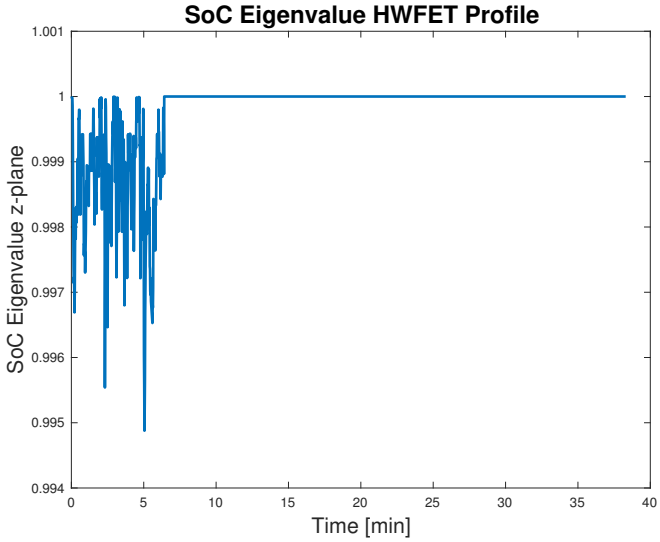
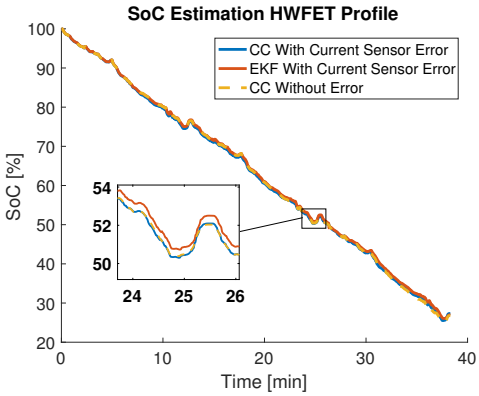
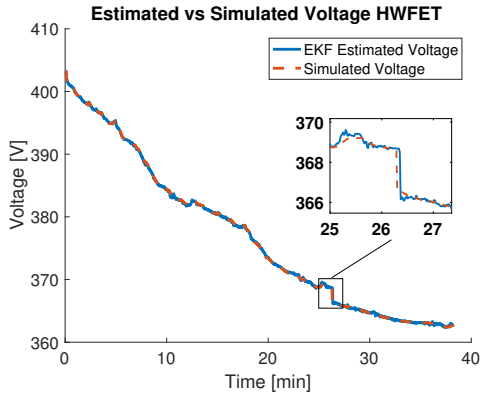


Figure 4.16: The SoC eigenvalue in the z-plane is found for the HWFET SoC profile together with the 0A current error using Equation (4.1). The EKF is unstable if the eigenvalue is ≥ 1 (Brown and Hwang, 2012).

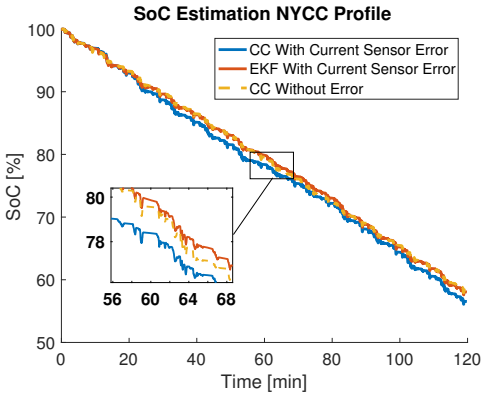


(a) This figure compares CC with and without a current sensor error to an EKF with a current sensor error. The 'true' SoC is the line in yellow. The RMS error is 0.5% for the EKF and 0.3% for CC.

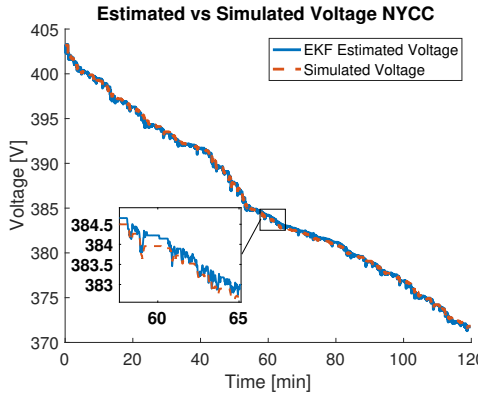


(b) This figure shows the simulated true voltage and EKF estimated voltage. The simulated and estimated voltages are found using Equations (3.30) & (1.2). The RMS error is 0.15 V.

Figure 4.17: Both figures assume a 50% current sensor white noise is added after 6 minutes. The voltage sensor has no measurement errors. A flat elevation profile along with a looped HWFET speed profile are used.

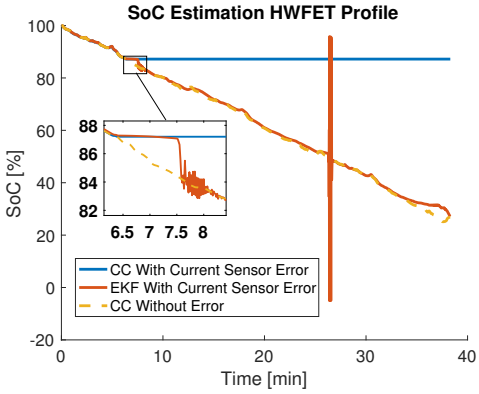


(a) This figure compares CC with and without a current sensor error to an EKF with a current sensor error. The 'true' SoC is the line in yellow. The RMS error is 0.3% for the EKF and 1% for CC.

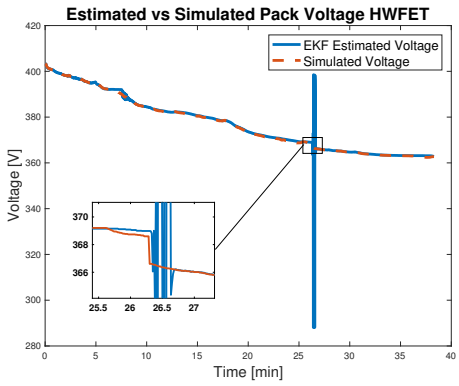


(b) This figure shows the simulated true voltage and EKF estimated voltage. The simulated and estimated voltages are found using Equations (3.30) & (1.2). The RMS error is 0.17 V.

Figure 4.18: Both figures assume a 50% current sensor white noise is added after 6 minutes. The voltage sensor has no measurement errors. A flat elevation profile along with a looped NYCC speed profile are used.

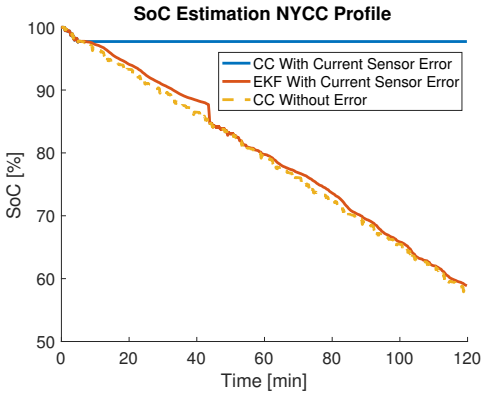


(a) This figure compares CC with and without a current sensor error to an EKF with a current sensor error. The 'true' SoC is the line in yellow. The RMS error is 3.3% for the EKF and 32.4% for CC.

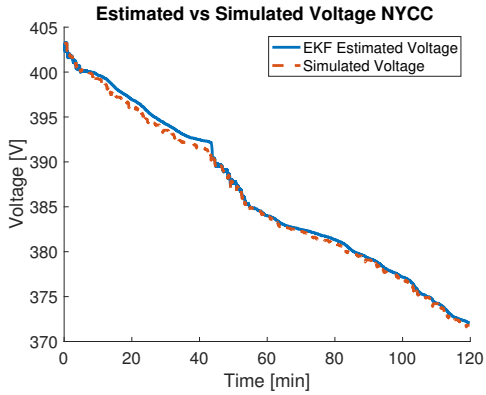


(b) This figure shows the simulated true voltage and EKF estimated voltage. The simulated and estimated voltages are found using Equations (3.30) & (1.2). The RMS error is 2.9 V.

Figure 4.19: Both figures assume the current sensor outputs 0A after 6 minutes. The voltage sensor has no measurement errors. A flat elevation profile along with a looped HWFET speed profile are used.

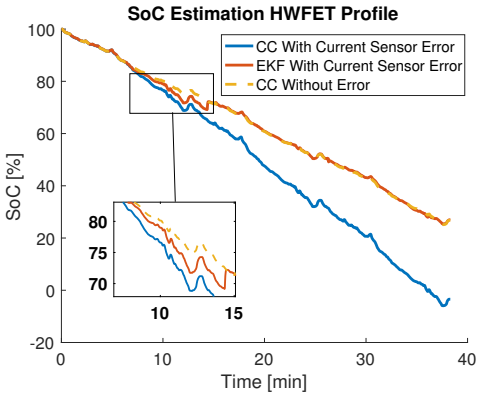


(a) This figure compares CC with and without a current sensor error to an EKF with a current sensor error. The 'true' SoC is the line in yellow. The RMS error is 0.9% for the EKF and 22% for CC.

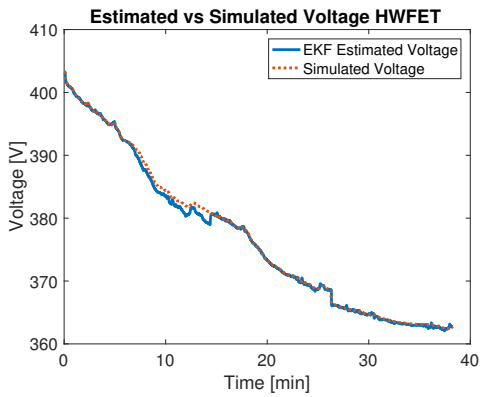


(b) This figure shows the simulated true voltage and EKF estimated voltage. The simulated and estimated voltages are found using Equations (3.30) & (1.2). The RMS error is 0.5 V.

Figure 4.20: Both figures assume the current sensor outputs 0A after 6 minutes. The voltage sensor has no measurement errors. A flat elevation profile along with a looped NYCC speed profile are used.

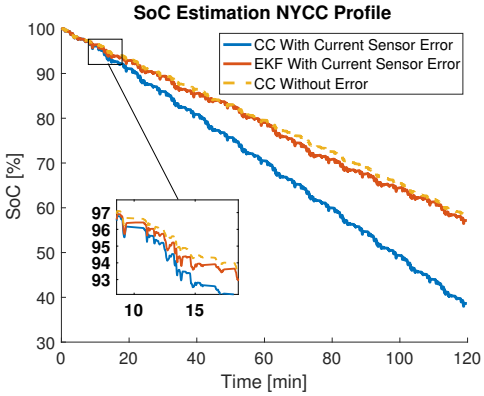


(a) This figure compares CC with and without a current sensor error to an EKF with a current sensor error. The 'true' SoC is the line in yellow. The RMS error is 1% for the EKF and 16.2% for CC.

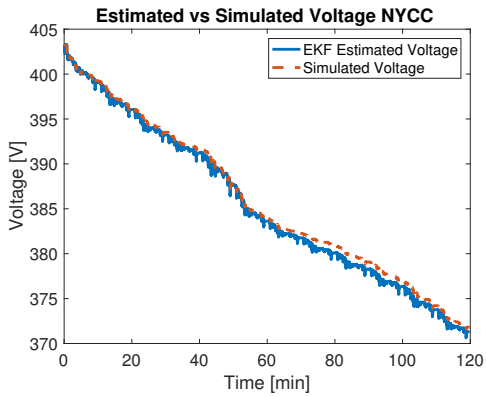


(b) This figure shows the simulated true voltage and EKF estimated voltage. The simulated and estimated voltages are found using Equations (3.30) & (1.2). The RMS error is 0.5 V.

Figure 4.21: Both figures assume the current sensor outputs a 50% current bias after 6 minutes (i.e. current·1.5)). The voltage sensor has no measurement errors. A flat elevation profile along with a looped HWFET speed profile are used.



(a) This figure compares CC with and without a current sensor error to an EKF with a current sensor error. The 'true' SoC is the line in yellow. The RMS error is 1% for the EKF and 11% for CC.



(b) This figure shows the simulated true voltage and EKF estimated voltage. The simulated and estimated voltages are found using Equations (3.30) & (1.2). The RMS error is 0.6 V.

Figure 4.22: Both figures assume the current sensor outputs a 50% current bias after 6 minutes (i.e. current·1.5). The voltage sensor has no measurement errors. A flat elevation profile along with a looped NYCC speed profile are used.

Chapter 5

Conclusion

The goal of this thesis is to find a way to combat range anxiety for EVs. First, the ESC battery model is explained and a method for parameter estimation of the unknown ESC model values is given using two battery tests and an EKF. A second EKF then uses the parameter estimation data, along with voltage and current data from the dynamic battery test, to estimate the SoC and find the estimated voltage. The estimated voltage is then compared to the real voltage to verify the parameter values of the Revolve battery cell.

Next, a method is developed to download elevation and distance data from a random Google Maps distance profile. The Google Maps profile represents the starting point and destination of the EV. The route is loaded into Matlab for accurate distance information at each waypoint. Elevation data from GPS Visualizer is then downloaded for the specific Google Maps distance profile. This elevation data resolution is further increased by interpolating elevation points using the Google Maps distance information. The elevation information is then transformed into grade angle information and attached to three different speed profiles (i.e. HWFET,NYCC,UDDS).

An algorithm then inputs the speed profile and grade angle information to calculate the battery SoC, extrapolated range and time to go until battery depletion. The parameter estimation data from the Revolve battery is used and the SoC is calculated using the SoC EKF described in the first paragraph of the conclusion.

The main result from this thesis shows that the extrapolated range is 23% greater compared to real test data using the 2013 Chevy Volt for two speed profiles assuming a flat elevation profile. The cell capacity is increased to resemble the real Chevy Volt and not a Chevy Volt with Revolve battery cells. Since parameter estimation is not done for the

Chevy Volt cell, CC must be used for SoC estimation. For all other results, the Revolve cells are used in combination with the dual EKF. Test trends using the same speed profiles with three different elevation profiles indicate that the extrapolated range is greatest for a downhill elevation profile, followed by the flat, mixed and uphill profiles. The downhill sections regenerate battery pack SoC, while the uphill sections deplete more SoC. Lastly, three test cases with three different types of current sensor errors and two different speed profiles illustrate situations where the EKF has better SoC estimation performance than CC.

It can be concluded that this methodology for finding the extrapolated range from a Google Maps profile shows promising results for all four elevation profiles. To the best of the author's knowledge, such a system does not yet exist. Tesla has a similar system, yet it is unsure how accurate the SoC estimation is. Many EV parameters in the vehicle modelling section are currently estimated, as this information is not public. Auto manufacturers will be able to use true values to increase the modelling accuracy. Some further improvements will allow the EV owner to input their destination in a GPS, which will then generate a SoC at the destination, as well as extrapolate the range and time to go until battery pack depletion. This information is essential to alleviate range anxiety and increase the uptake of EVs worldwide. It is important to note that the extrapolated range estimate should be conservative when tested in real vehicles to ensure that drivers do not deplete the battery pack fully and thereby cannot reach their destination.

Although the method shows promise, there is still room for improvement in future work. The top priority is to find a better way to model the speed profile given a Google Maps route. The author tried unsuccessfully to use the Google Maps distance waypoints with the elevation information to generate a speed profile, where the car fully stops at every distance waypoint. It might be helpful to use past driving data from the individual driver to generate a speed profile. The second priority is to validate this strategy using a real EV and following a desired route. True speed for the route can be measured and recorded using an app such as SpeedView. This will allow a comparison between the real and generated speed profiles, as well as a comparison between real measured and estimated vehicle SoC at the destination. It is important to note that the real measured vehicle SoC most likely uses CC, so it is unsure how accurate this estimate is. Nonetheless, this validation strategy is a cost-effective first step to further optimize the methodology and increase extrapolated range and time to go accuracy.

Bibliography

Bonges, H., Lusk, A., 1 2016. Addressing ev sales and range anxiety through parking layout, policy and regulation. Elsevier Transportation Research Part A: Policy and Practice 83 (2), 63–64, http://www.sciencedirect.com/.

Brown, R., Hwang, P., 2012. Introduction to Random Signals and Applied Kalman Filtering, 4th Edition. John Wiley, Hoboken, NJ, USA, pg. 197.

Bunker, B., 2 2015. Determination and Use of Vehicle Road-Load Force and Dynamometer Settings. Environmental Protection Agency, Ann Arbor, Michigan, USA, https://iaspub.epa.gov.

Cazzola, P., Gorner, M., et. al., 2016. Global ev outlook 2016. Tech. rep., International Energy Agency, Paris, France, https://www.iea.org.

DOE, 2013. 2013 chevy volt advanced vehicle testing - baseline testing results. Tech. rep., US Department of Energy, Laurel, MD, USA, https://avt.inl.gov.

DOE, 2017. The Official US Government Source for Fuel Economy Information. United States Department of Energy, Laurel, MD, USA, https://www.fueleconomy.gov.

Dubarry, M., Vuillaume, N., Liaw, B. Y., 1 2009. From single cell model to battery pack simulation for li-ion batteries. Elsevier Journal of Power Sources 186 (2), 500–507, http://www.sciencedirect.com/.

EPA, 11 2016. Electric Vehicles - Learn More About the New Label. Environmental Protection Agency, Ann Arbor, Michigan, USA, https://www.epa.gov/.

EPA, 2017. Dynamometer Drive Schedules. United States Environmental Protection Agency, Ann Arbor, Michigan, USA, https://www.epa.gov.

Farr, T., Rosen, P., et. al., 2007. The shuttle radar topography mission. Tech. rep., NASA et. al., Pasadena, CA, USA, https://www2.jpl.nasa.gov.

Fernandez-Prim, D., 2009. On-figure magnifier. https://se.mathworks.com/matlabcentral/.

Fujimoto, H., Yoshida, H., et. al., 2016. Bench test of range extension autonomous driving for electric vehicles based on optimization of velocity profile considering traffic signal information. In: EVTeC and APE Japan. Society of Automotive Engineers of Japan.

Gillespie, T., 1992. Fundamentals of Vehicle Dynamics. Society of Automotive Engineers.

Information, D., 2017. Baldwin street. http://www.dunedin.nz.com.

King, C., Griggs, W., et. al., 2015. Alleviating a form of electric vehicle range anxiety through on-demand vehicle access. International Journal of Control, 717–728Doi: 10.1080/00207179.2014.971521.

Korosec, K., 2015. Tesla's battery swap program is pretty much dead. http://fortune.com.

Kvaale, B., 2016. Extended kalman filtering with a nonlinear enhanced self-correcting li-ion battery cell model for state of charge estimation. Tech. rep., Norwegian University of Science and Technology.

Marketing, M., 2017. Pdf to text. http://pdftotext.com.

Mastali, M., Vazquez-Arenas, J., et. al., 10 2013. Battery state of charge estimation using kalman filtering. Elsevier Journal of Power Sources 239, 294–307, http://www.sciencedirect.com/.

Neaimeh, M., Hill, G., et. al., 2013. Routing systems to extend the driving range of electric vehicles. IET Intelligent Transport Systems 7, 327–336.

Plett, G., 2004. Extended kalman filtering for battery management systems of li-pb-based hev battery packs part 2. Journal of Power Sources, 262–276.

Plett, G., 2015a. Battery Management Systems: Battery Modeling. Vol. 1. Artech House, Norwood, MA, USA, pp. 29–65.

Plett, G., 2015b. Ekf-based battery-cell state estimation (escekf.zip). http://mocha-java.uccs.edu/BMS2/.

Plett, G., 2016a. Battery Management Systems: Equivalent-Circuit Methods. Vol. 2. Artech House, Norwood, MA, USA, pp. 12,25,40–55,70–76,114–129,167–226.

Plett, G., 2016b. Electric-vehicle simulator (evsim.zip). http://mocha-java.uccs.edu/BMS2/.

QRG, 2017. What is inertia? http://www.qrg.northwestern.edu/.

Schneider, A., 2016. Gps visualizer. http://www.gpsvisualizer.com/convert_input.

Sleigh, A., 2016. Work, power and energy. http://www.efm.leeds.ac.uk.

Tesla, 2015. Model s has you covered. https://www.tesla.com/.

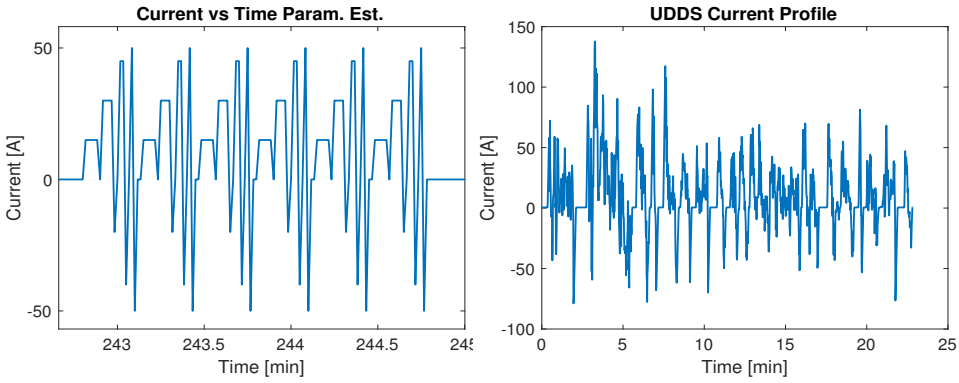
Tesla, 2017. Supercharger. https://www.tesla.com/.

Wing, M., Frank, J., 9 2011. Vertical measurement accuracy and reliability of mapping-grade gps receivers. Elsevier Computers and Electronics in Agriculture 78 (2), 188–194, http://www.sciencedirect.com/.

Xin, D., Chengning, Z., et. al., 2010. State monitor for lithium-ion power battery pack. In: 2010 International Conference on Measuring Technology and Mechatronics Automation. Doi: 10.1109.

Appendix

5.1 Dynamic Test Cycles For Parameter Estimation



(a) Revolve Current Profile For Parameter Estimation

(b) UDDS Current Profile for Parameter Estimation

Figure 5.1: This figure shows two dynamic test cycles that can be run multiple times with rest periods in between from approximately 90% SoC to 10% SoC. These test cycles can be used for estimation of dynamic parameters and for testing the EKF over a sample profile similar to the final application of the battery (Plett, 2015a).

Figure 5.1a shows the current profile used in the parameter estimation section of this thesis. The current profile for the UDDS speed profile is shown in Figure 5.1b. An analysis of both figures shows that the UDDS profile has more varying currents and is more dynamic over the entire profile, whereas the Revolve profile repeats the same sequence over and over. As discussed in the thesis, this could be a major reason why the estimation results of the dynamic ESC parameters are not as good as the state of the art results.

5.2 General EKF Algorithm For Parameter Estimation

To find $\hat{C}_k^{\theta}$ in Algorithm (1), a recursive calculation is done below.

$$\begin{aligned} dg(x_k, u_k, \theta, e_k) = &\frac{\partial g(x_k, u_k, \theta, e_k)}{\partial x_k} dx_k + \frac{\partial g(x_k, u_k, \theta, e_k)}{\partial u_k} du_k \\ &+ \frac{\partial g(x_k, u_k, \theta, e_k)}{\partial \theta} d\theta + \frac{\partial g(x_k, u_k, \theta, e_k)}{\partial e_k} de_k \end{aligned}$$

$$\begin{aligned} \frac{dg(x_k, u_k, \theta, e_k)}{d\theta} &= \frac{\partial g(x_k, u_k, \theta, e_k)}{\partial x_k} \frac{dx_k}{d\theta} + \frac{\partial g(x_k, u_k, \theta, e_k)}{\partial u_k} \underbrace{\frac{du_k}{d\theta}}_{0} \\ &+ \frac{\partial g(x_k, u_k, \theta, e_k)}{\partial \theta} \frac{d\theta}{d\theta} + \frac{\partial g(x_k, u_k, \theta, e_k)}{\partial e_k} \underbrace{\frac{de_k}{d\theta}}_{0} \\ &= \frac{\partial g(x_k, u_k, \theta, e_k)}{\partial \theta} + \frac{\partial g(x_k, u_k, \theta, e_k)}{\partial x_k} \frac{dx_k}{d\theta} \end{aligned}$$

Equation (5.1) is a recursive function. To find the next value of $\frac{dx_k}{d\theta}$, Equation (5.1) goes all the way back to $\frac{dx_0}{d\theta}$, which is initialized to 0 unless a better estimate is known.

$$\frac{dx_k}{d\theta} = \frac{\partial f(x_{k-1}, u_{k-1}, \theta, w_{k-1})}{\partial \theta} + \frac{\partial f(x_{k-1}, u_{k-1}, \theta, w_{k-1})}{\partial x_{k-1}} \frac{dx_{k-1}}{d\theta} \tag{5.1}$$

(Plett, 2016a)

5.3 The General EKF Algorithm

1: **for** $k = 1, 2, \ldots$ n **do**

2: State prediction time update: $\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-} \approx \boldsymbol{f}(\hat{\boldsymbol{x}}_{\boldsymbol{k-1}}^{+}, \boldsymbol{u_{k-1}}, \bar{\boldsymbol{w}}_{\boldsymbol{k-1}})$

3: Error covariance time update: $\tilde{\boldsymbol{x}}_{\boldsymbol{k}}^{-} = \boldsymbol{x_k} - \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-} \approx \left(\hat{\boldsymbol{A}}_{\boldsymbol{k-1}} \tilde{\boldsymbol{x}}_{\boldsymbol{k-1}}^{+} + \hat{\boldsymbol{B}}_{\boldsymbol{k-1}} \tilde{\boldsymbol{w}}_{\boldsymbol{k-1}}\right)$

$$\boldsymbol{\sum}_{\tilde{\boldsymbol{x}},\boldsymbol{k}}^{-} \approx \hat{\boldsymbol{A}}_{\boldsymbol{k-1}} \boldsymbol{\sum}_{\tilde{\boldsymbol{x}},\boldsymbol{k-1}}^{+} \hat{\boldsymbol{A}}_{\boldsymbol{k-1}}^{\top} + \hat{\boldsymbol{B}}_{\boldsymbol{k-1}} \boldsymbol{\sum}_{\tilde{\boldsymbol{w}}} \hat{\boldsymbol{B}}_{\boldsymbol{k-1}}^{\top}$$

4: Output estimate: $\hat{\boldsymbol{y}}_{\boldsymbol{k}} = \mathbb{E}\left[\boldsymbol{g}(\boldsymbol{x}_{\boldsymbol{k}}, \boldsymbol{u}_{\boldsymbol{k}}, \boldsymbol{v}_{\boldsymbol{k}}) | \mathbb{Y}_{k-1}\right] \approx \boldsymbol{g}(\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-}, \boldsymbol{u}_{\boldsymbol{k}}, \bar{\boldsymbol{v}}_{\boldsymbol{k}})$

5: Kalman gain matrix $\boldsymbol{L_k}$: Define $\hat{\boldsymbol{C}}_{\boldsymbol{k}} \triangleq \left.\dfrac{d\boldsymbol{g}(\boldsymbol{x}_{\boldsymbol{k}}, \boldsymbol{u}_{\boldsymbol{k}}, \boldsymbol{v}_{\boldsymbol{k}})}{d\boldsymbol{x}_{\boldsymbol{k}}}\right|_{\boldsymbol{x}_{\boldsymbol{k}}=\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-}}$

Define $\hat{\boldsymbol{D}}_{\boldsymbol{k}} \triangleq \left.\dfrac{d\boldsymbol{g}(\boldsymbol{x}_{\boldsymbol{k}}, \boldsymbol{u}_{\boldsymbol{k}}, \boldsymbol{v}_{\boldsymbol{k}})}{d\boldsymbol{v}_{\boldsymbol{k}}}\right|_{\boldsymbol{v}_{\boldsymbol{k}}=\bar{\boldsymbol{v}}_{\boldsymbol{k}}}$

$$\sum_{\tilde{\boldsymbol{y}},\boldsymbol{k}} \approx \hat{\boldsymbol{C}}_{\boldsymbol{k}} \sum_{\tilde{\boldsymbol{x}},\boldsymbol{k}}^{-} \hat{\boldsymbol{C}}_{\boldsymbol{k}}^{\top} + \hat{\boldsymbol{D}}_{\boldsymbol{k}} \sum_{\tilde{\boldsymbol{v}}} \hat{\boldsymbol{D}}_{\boldsymbol{k}}^{\top}$$

$$\sum_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{y}},\boldsymbol{k}}^{-} \approx \mathbb{E}\left[(\tilde{\boldsymbol{x}}_{\boldsymbol{k}}^{-})(\hat{\boldsymbol{C}}_{\boldsymbol{k}}\tilde{\boldsymbol{x}}_{\boldsymbol{k}}^{-} + \hat{\boldsymbol{D}}_{\boldsymbol{k}}\tilde{\boldsymbol{v}}_{\boldsymbol{k}})^{\top}\right] = \sum_{\tilde{\boldsymbol{x}},\boldsymbol{k}}^{-} \hat{\boldsymbol{C}}_{\boldsymbol{k}}^{\top}$$

$$\boldsymbol{L_k} = \sum_{\tilde{\boldsymbol{x}},\boldsymbol{k}}^{-} \hat{\boldsymbol{C}}_{\boldsymbol{k}}^{\top} \left[\hat{\boldsymbol{C}}_{\boldsymbol{k}} \sum_{\tilde{\boldsymbol{x}},\boldsymbol{k}}^{-} \hat{\boldsymbol{C}}_{\boldsymbol{k}}^{\top} + \hat{\boldsymbol{D}}_{\boldsymbol{k}} \sum_{\tilde{\boldsymbol{v}}} \hat{\boldsymbol{D}}_{\boldsymbol{k}}^{\top}\right]^{-1}$$

6: State estimate measurement update: $\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{+} = \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-} + \boldsymbol{L_k}(\boldsymbol{y_k} - \hat{\boldsymbol{y}}_{\boldsymbol{k}})$

7: Error covariance measurement update: $\sum_{\tilde{\boldsymbol{x}},\boldsymbol{k}}^{+} = \sum_{\tilde{\boldsymbol{x}},\boldsymbol{k}}^{-} + \boldsymbol{L_k} \sum_{\tilde{\boldsymbol{y}},\boldsymbol{k}} \boldsymbol{L}_{\boldsymbol{k}}^{\top}$

8: Compute error bounds: $\hat{\boldsymbol{x}}_{\boldsymbol{k},\text{Error Bounds}}^{+} = \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{+} \pm 3\sqrt{\text{diag}\left(\sum_{\tilde{\boldsymbol{x}},\boldsymbol{k}}^{+}\right)}$

9: **end for**

(Plett, 2016a)

The EKF used for the SoC estimation (the second EKF in Figure 2.1) uses a form of the adaptive EKF, where the SoC index of the process error covariance matrix $\sum_X$ increases in value if there is a bad voltage estimate. In addition, the Joseph form is used to ensure that the process covariance matrix remains symmetric and positive-definite (Plett, 2016a). The Joseph form update is done after $\sum_X$ has been updated using the Kalman gain. These additions help the EKF reacquire in some cases.

5.4 Finding the True Range of 2013 Chevy Volt

The true driving range is found in Table 4.9 using two different methods. The first method finds dynamometer values that are given at 22 degrees Celsius for the UDDS and HWFET profiles in the units Wh/mile (DOE, 2013). Unfortunately, direct range estimation results in miles are not given. To find estimated true values in kilometers, the rated pack energy given in kWh is divided by the values given in Wh/mile. Since the true power expelled during the speed profiles is unknown, this is the best guess. This leaves a result in miles, which is easily converted to km.

To find the true driving values in Method 2, source (DOE, 2017) is consulted. This source gives gives an extrapolated range of 61 km (38 miles) for the 2013 Chevy Volt in the fully electric mode. The range is based on 45% highway driving (i.e. HWFET profile) and 55% city driving (i.e. UDDS profile) (EPA, 2017) (EPA, 2016). From this, it is desirable to get the true extrapolated range values for the HWFET and UDDS profiles. Since we only have one known variable (i.e. 61 km) and two unknown variables (i.e. true HWFET

range and UDDS range), the author makes a simplification that the range consists of 50% HWFET and 50% UDDS cycle results. This is a valid assumption because the HWFET and UDDS profiles have very similar simulated and true ranges for the Revolve and Chevy Volt battery packs (see Tables 4.7 and 4.9). This simplification results in a true range of 61 km for both the HWFET and UDDS cycles.