

Politecnico Di Torino

---

*SCHOOL OF ENGINEERING INFORMATICS*

Department of Control and Computer Engineering

DAUIN

Corso di Laurea Magistrale in Ingegneria Meccatronica

Master Thesis

Real-time battery state of health estimation based on terminal voltage analysis

Supervisor:  
Dr. Alessandro Rizzo

Industry Tutor:  
Ing. Giovanni Guida

Defended By:  
Youssef Saad



December 2019



## i. Abstract

Due to the evolution of automotive industries nowadays, the world's eyes are directed towards power production and power storage. As energy is the most valuable resource for mankind; we as mechatronics engineers not only consider its production or storage, but also, we are concerned about managing power and tailoring it to fit our everyday needs and uses. To be more specific, this thesis addresses batteries in general as the source of power we are concerned about. BATMAN is the name of our product, which originated from BATtery MANagement system.

At this very moment, there is no such product in the market that has the same functionalities. This product took an inner view into the study of batteries. BATMAN is an embedded low-cost device able to perform accurate measurements of the State of Health (SOH) and the State of Charge (SOC) of a battery. Also, it processes data and informs the user through a simple interface such as an application on a smart phone. The outcome of the project will be directly applied for the development and marketing of a wide range of aftermarket products for the monitoring of Lead-Acid batteries. Moreover, the know-how and the technologies developed can be transferred to other types of batteries, such as Lithium Ion (LION), and therefore applicable to the technological development agenda foreseen for the next few years, regarding hybrid/fully electric vehicles (HBEV and BEV) with optimal management of co-generation micro-networks (photovoltaic, wind... etcetera).

Our product has two phases. The first phase is implementing the functionalities mentioned above while using the data generated by both the terminal Voltage and the electric current withdrawn from the battery (aka. Load current). While the second phase is implementing these functionalities using only the terminal Voltage information with no knowledge about the electric current. The aim of this thesis is to fully develop a battery State of Health (SOH) detection algorithm and implementing it in real time, afterwards, connect it to the device under management in order to observe and compile the data, and finally inform the user with various information about the battery under inspection specifically the battery State of Charge and State of Health.

## ii. Acknowledgement

I would like to start by thanking all my professors and teacher assistants at the University of Torino and to all my colleagues in Brain Technologies who worked with me so hard in order to achieve such success. Also, special thanks to Brain Technologies who gave me the opportunity to work on such an interesting and successful project as BATMAN. trusted and believed in me to be a part of this great project. All what have I achieved was thanks to a great supervision and assistance, I was quite lucky to work with a great team that contributed a lot to my personal and professional development.

A special thanks to my leader and mentor Ing.Giovanni Guida for all his support, and his continuous guidance and that he gave me the opportunity to learn from him and improve my skills also for taking keen interest on teaching me the basics of R&D and so much more.

To my great Professor Dr. Alessandro Rizzo, for his support and for always being there when I needed him. Providing me with his expertise and time was something extraordinary.

To my beloved best-friend Ing.Mohamed Salem for always providing me with support and advice along the way.

My lifetime partner Ing.Mohamed Hisham for always consulting me and sharing his expertise despite being in a different continent. Still managing to communicate with me was a huge blessing

Last but not least, to my beloved parents and beloved brothers, for supporting me through my whole life and for their impact of me becoming the person I am today. I will always strive to make them feel proud, I owe them all my love, success and achievements.

Thank you

Youssef Saad

### iii. Table of Contents

MASTER THESIS .....	I
REAL-TIME BATTERY STATE OF HEALTH ESTIMATION BASED ON TERMINAL VOLTAGE ANALYSIS.....	I
I. ABSTRACT.....	III
II. ACKNOWLEDGEMENT .....	IV
III. TABLE OF CONTENTS.....	V
IV. TABLE OF FIGURES.....	VII
V. LIST OF TABLES.....	VIII
<b>1. INTRODUCTION.....</b>	<b>1</b>
1.1. BATTERY STATE OF HEALTH (SOH) .....	1
1.2. BATTERY STATE OF CHARGE (SOC) .....	2
1.3. BATTERY STATE OF HEALTH IMPORTANCE.....	3
1.4. BATTERY STATE OF HEALTH AND STATE OF CHARGE CORRELATION.....	4
<b>2. STATE OF ART AND BATTERY MODEL .....</b>	<b>6</b>
2.1. PATENTED DEVICE (UNITED STATES OF AMERICA BREVETTO N. 8437908 B2, 2013)[4] .....	6
2.2. DEVICE UPGRADES.....	7
2.3. BATTERY MODELLING .....	8
<b>3. MODEL VERIFICATION AND ANALYSIS .....</b>	<b>14</b>
3.1. BATTERY MODEL VERIFICATION.....	14
3.2. REAL VOLTAGE VS. SIMULATED VOLTAGE.....	15
3.3. OUTPUT VOLTAGE ANALYSIS.....	18
3.3.1. <i>Observations</i> .....	18
3.3.2. <i>Box Plots</i> .....	19
<b>4. ALGORITHM KEY PARAMETERS AND CONSTRAINTS.....</b>	<b>21</b>
4.1. SLOPE ANALYSIS .....	21
4.2. PARAMETER 1 ( $\Delta$ ) .....	22
4.3. PARAMETER 2 ( $\Delta$ ) .....	23
4.4. PARAMETER 3 ( $\Sigma^2$ ) .....	27
<b>5. CLASSIFICATION &amp; SUPPORT VECTOR MACHINE (SVM).....</b>	<b>31</b>
5.1. SUPPORT VECTOR MACHINE OVERVIEW .....	31
5.2. INPUT DATA SET.....	36
5.3. RESPONSE DATA SET .....	37
5.4. TRAINING DATA SET.....	38
5.5. VALIDATION DATA SET.....	39
<b>6. SVM IMPLEMENTATION AND TESTING .....</b>	<b>40</b>
6.1. SVM DETAILS.....	40
6.2. MODEL IN THE LOOP TESTING .....	43
<b>7. BAT-MAN IN REAL TIME IMPLEMENTATION AND TESTING.....</b>	<b>47</b>
7.1. REAL TIME MODEL .....	47
7.2. TEST INITIALIZATION.....	48
7.3. TESTING USING REAL INPUT DATA .....	49
7.4. TESTING USING SIMULATED INPUT DATA .....	52
<b>8. CONCLUSION .....</b>	<b>55</b>
<b>APPENDIX .....</b>	<b>56</b>

1.	MATLAB SCRIPTS.....	56
2.	SIMULINK MODELS.....	66
<b>BIBLIOGRAPHY .....</b>		<b>71</b>

## iv. Table of Figures

FIGURE 1-BATTERY/CELL USAGE OVER TIME .....	4
FIGURE 2-BATTERY ASOH AND ASOC DEFINITIONS .....	5
FIGURE 3 - PATENTED DEVICE FLOW CHART.....	6
FIGURE 4-BATMAN FLOW CHART.....	7
FIGURE 5- GENERALIZED LEAD-ACID BATTERIES NYQUIST DIAGRAM RESULTING FROM AND IMPEDANCE SPECTROSCOPY...	9
FIGURE 6 - BATTERY EQUIVALENT CIRCUIT .....	10
FIGURE 7 - COURSE OF VOLTAGE AND CURRENT WHILE DISCHARGING FOLLOWED BY CHARGING .....	11
FIGURE 8 - BATTERY DISCHARGING VOLTAGE AND CURRENT.....	12
FIGURE 9 - BATTERY OPEN CIRCUIT VOLTAGE VS STATE OF CHARGE.....	13
FIGURE 10-BATTERY 1 VERIFICATION.....	15
FIGURE 11-BATTERY 2 VERIFICATION.....	16
FIGURE 12-BATTERY 3 VERIFICATION.....	17
FIGURE 13-BATTERY 3 VERIFICATION.....	17
FIGURE 14-PEAKTOPEAK VOLTAGE .....	18
FIGURE 15-BOXPLOTS REPRESENTATION.....	19
FIGURE 16-BOXPLOTS PERCENTAGES .....	19
FIGURE 17-VOLTAGE DATA BOXPLOT .....	20
FIGURE 18- SLOPE BOXPLOT.....	21
FIGURE 19-SLOPE BOXPLOT PATTERN.....	22
FIGURE 20-PARAMETER 2 PLOT DATA1 .....	23
FIGURE 21-PARAMETER 2 PLOT DATA 2.....	24
FIGURE 22-PARAMETER 2 PLOT SINEWAVE .....	24
FIGURE 23 - PARAMETER 2 WITH DIFFERENT INITIAL SOC .....	25
FIGURE 24 – VARIANCE VS SOH DATA1.....	27
FIGURE 25 - FIGURE 19 – VARIANCE VS SOH DATA2.....	29
FIGURE 26- SVM HYPERPLANE .....	31
FIGURE 27 - SVM REGULARIZATION PARAMETER 1.....	32
FIGURE 28 - SVM REGULARIZATION PARAMETER 2.....	32
FIGURE 29 - SVM REGULARIZATION PARAMETER 3.....	33
FIGURE 30 - HIGH GAMMA.....	34
FIGURE 31 - LOW GAMMA .....	35
FIGURE 32 – MARGINS .....	35
FIGURE 33 - VALIDATION FOLDS .....	41
FIGURE 34 - TRAINING APP DETAILS .....	42
FIGURE 35 - MIL MODEL.....	43
FIGURE 36 - REAL TIME MODEL.....	47
FIGURE 37 – REAL DATA TEST 80% SOHI.....	49
FIGURE 38 - REAL DATA TEST 50% SOHI.....	50
FIGURE 39 - REAL DATA TEST 30% SOHI.....	51
FIGURE 40 - SIMULATED DATA TEST 50% SOHI.....	52
FIGURE 41 - SIMULATED DATA TEST 30% SOHI.....	53
FIGURE 42 - BATTERY SIMULINK MODEL .....	66
FIGURE 43 - INPUT CURRENT SIMULINK MODEL .....	66
FIGURE 44 - SUPPORT VECTOR MACHINE SIMULINK MODEL.....	67
FIGURE 45 - NON REAL TIME TESTING SIMULINK MODEL .....	68
FIGURE 46 - REAL TIME TESTING SIMULINK MODEL.....	69

## v. List of Tables

TABLE 1-LEAD ACID BATTERY STATE OF CHARGE.....	2
TABLE 2 - EMPLOYED LEAD ACID BASED BATTERIES.....	10
TABLE 3 - REAL DATA SET .....	40
TABLE 4 – NON-REAL TIME TEST RESULTS .....	45



# 1. Introduction

As a starting point in order to get deeper into the steps of implementing BATMAN, there are a few aspects to be made clear, which are the definitions of both the State of Health (SOH) and the State of Charge (SOC) of a battery.

## 1.1. Battery State of Health (SOH)

The state of health (SOH) of a battery is a ‘figure of merit’ to describe the battery condition, and how much of useful lifetime is remaining from the battery, thus, if you know the state of health of your battery you would have the ability to know whether your battery needs repairs or even replacement, or is it functioning with the optimum performance. It is calculated in percentage points ‘%’. [1]

A 100% State of health battery means that this battery is brand new (matching the specs of the battery) and was not used before. Accordingly, a used battery has a decreasing state of health over time due to the battery decaying. The battery decaying happens due to chemistry problems.

A healthy battery behaves in a way that the ions flow freely from the anode to the cathode, and when the battery is put to a load aka withdrawing electric current, the ion flow in the reverse direction.

This process tears down the cathode and wears it by time, thus it means reduced battery capacity because less ions are now flowing from the anode to the cathode and vice versa. Nevertheless, battery state of health (SOH) is not provided by battery manufacturers because in its current condition it is known that it has a (100%) state of health, so this concept applies only when a battery is put into use and start withdrawing electric current. Moreover, the state of health of the battery can be applied to each cell of the battery in which in the end gives the overall value. And since every battery has different specifications and behaves differently, a battery state of health is measured differently amongst different batteries because it is a reference to the battery at its optimum performance which are different in reality. So, the factors must contain previous knowledge of the battery states or performance in order to be able to determine the new state of health of a battery.

## 1.2. Battery State of Charge (SOC)

A battery state of charge is the indication of how charged the battery is, like the previous parameter the State of Health, it is measured in percentage points. For example, a (100%) State of charge battery means that the battery is fully charged and there is no possibility to charge the battery more than this limit, whereas the maximum potential for energy storage from this battery is achieved.

While a (0%) state of charge battery will provide no energy at all because the battery is empty and needs to be recharged.

By the usage of the battery and discharging electric current from it, the battery state of charge starts to decrease, indication how many percentage is left, in other words, knowing the state of charge of the battery tells you many important aspects which are, how much energy have you got left stored in your battery to be provided if needed, and when is the optimum time to charge the battery to protect it and maximize its lifetime. Another parameter which is basically linked to the battery state of charge is called the battery Depth of Discharge (DoD), which is the inverse of the battery state of charge, and it tells how discharged your battery is. The battery state of charge is usually addressed when monitoring the state of a battery in its current situation, while the depth of discharged is addressed when monitoring the lifetime of a battery depending on its number of charge and discharge cycle

Below is table interpreting the State of charge of a charge of a lead acid battery under experiment and after resting for 3 hours. Also, the column on the right expresses the Volts per each cell in the battery with respect to the battery State of Charge. [2]

After 3 hour rest

State of Charge	12 Volt battery	Volts per Cell
100%	12.7	2.12
90%	12.5	2.08
80%	12.42	2.07
70%	12.32	2.05
60%	12.2	2.03
50%	12.06	2.01
40%	11.9	1.98
30%	11.75	1.96
20%	11.58	1.93
10%	11.31	1.89
0	10.5	1.75

*Table 1-Lead Acid Battery State of Charge*

### 1.3. Battery State of Health importance

For us electronic engineers, battery State of Health is much more of an importance to be able to calculate or even predict, as the battery state of health is the **battery replacement deciding factor**.

As for a normal battery, a low state of charge is of no concerns in the terms of maintenance and replacement, as the battery will be subjected to the charging process, thus it will be able to provide the energy needed. But, a battery with a low State of health would definitely require repairs or even replacements. As for any battery, there are 3 phases of working, which are:

a. Service zone

Which is the zone where the battery provides its rated capacity, thus providing fully the energy required aka battery at maximum capacity when full charged

b. Replacement zone

The zone where the battery starts degrading and the cathode is wearing out, which means that the battery is not providing its rated capacity. So, a battery in replacement zone will require either maintenance or replacement of the cathode or to be subjected to the necessary type of maintenance in order to retain it back in the Service zone

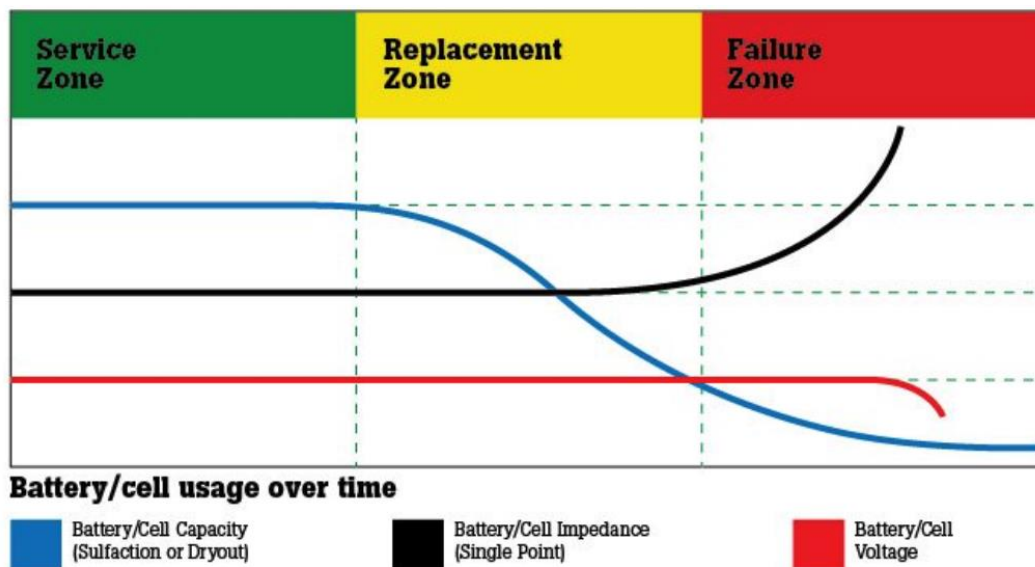
c. Failure zone

Once you enter the failure zone, the battery is known as 'failed'. Which means that the battery behavior cannot be predicted, and the battery is either not functioning or requires replacement.

This phase is very dangerous for many types of applications as they required fixed voltage and current input. So, this phase must be avoided, and the battery should be subjected to direct maintenance or repairs.

[3]

Below is a figure demonstrating the three different zones of a lead acid battery:



*Figure 1-Battery/Cell usage over time*

#### 1.4. Battery State of Health and State of Charge Correlation

In order to understand the final picture of the state of health and state of charge dilemma, there are some definitions to be explained, which are;

- Apparent State of Health (ASoH) - Ah Rating: Which gives an indication of the battery capacity at its maximum, but this does not mean that the battery will provide these ratings, because in reality the battery has degraded over time, and now the real State of Health is different from the apparent one. In other words, the Apparent state of health is no more than the battery rating indication in Ah (ampere hours).
- Apparent State of Charge (ASoC): Which gives indication of the State of Charge of the battery, but in reality, the battery will drain much faster than expected from the normal

behavior, as the battery has already degraded exactly as mentioned before in the (ASoH).[3]

As a result, you can find your battery at full 100% charge but in fact the battery (SOC) may correspond to a different level of charge.

Below is figure indicating the Apparent State of health and the Apparent State of Charge and their meaning compared to the real ones

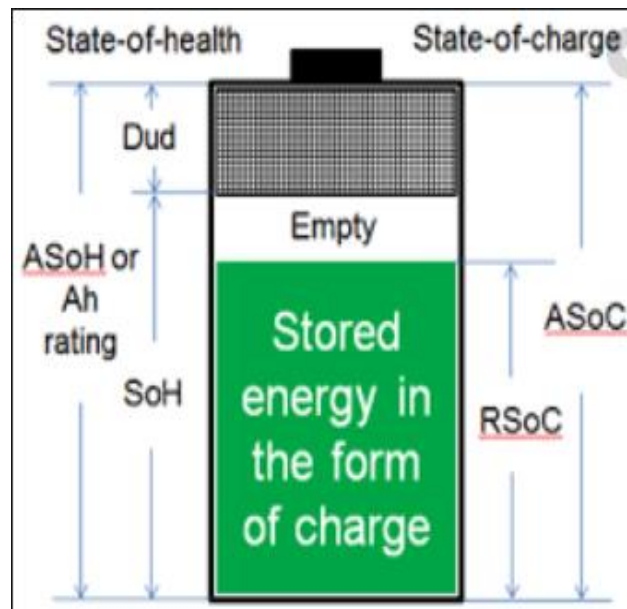


Figure 2-Battery ASoH and ASoC Definitions

## 2. State of Art and Battery Model

### 2.1. Patented Device (United States of America Brevetto n. 8437908 B2, 2013)[4]

The most basic device similar to our product batman, is called a '**BATTERY MONITOR SYSTEM ATTACHED TO VEHICLE WIRING HARNESS**'. It was patented in the United States of America in 2013.

Quoting the Patent

“

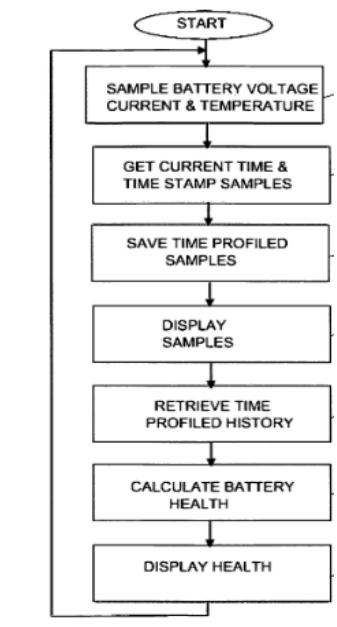
- The focus

It is essential to know when the health of a battery (SOH) has deteriorated sufficiently to signal that a failure is present. In some situations, this information could be lifesaving. Such as when operating in combat zones or under severe weather conditions. It would also be important to know that by merely changing the pattern of a vehicle usage such as combining multiple shopping trips into a single extended one or by knowing when to apply an external battery charger that the life of the battery would be extended, and for failures to be avoided.

- The solution

A computer system that is embedded in the proximity of the vehicle's operator, attaches to the battery power source in the vehicle's wiring harness, gathers information relating to the operational state of the vehicle's battery, calculates the state of health of the battery from the obtained data, provides the health and operational state of the battery to the vehicle's operator and can include the means for supporting non-battery related functions. ”

Figure 2.1 is a flow chart of the outline of the process of this device, it is a similar to our product, but our product has more features that are going to be demonstrated in the rest of this thesis.



*Figure 3 -  
Patented Device  
Flow Chart*

## 2.2. Device Upgrades

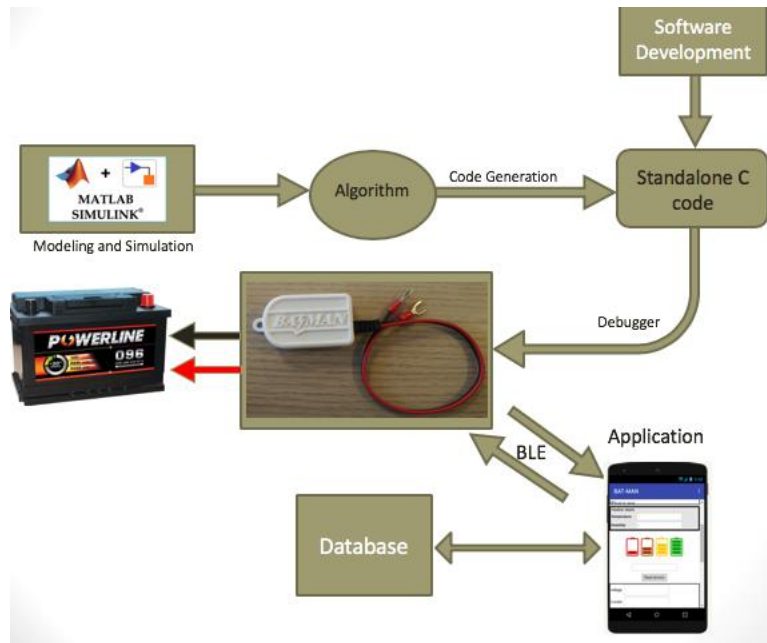


Figure 4-BATMAN Flow Chart

The figure above demonstrates our device's course path starting from developing the software till the application usage. Apparently, our device has more features than the one mentioned above.

- a. BATMAN's algorithm is built on analysis which were obtained only using the battery terminal voltage waveform, unlike any other BMS (Battery Management System) which mainly depends on the electric current waveform analysis aka. withdraw current or load current. Without this additional information it was very challenging (almost impossible) for any other battery management system to understand or give an accurate estimation of the current battery State of Charge or State of health. However, the following chapters will be a guide to a successful State of Health borders estimation using only the Battery Terminal Voltage.

- b. BATMAN is the first online BMS, in the sense that there is direct interaction with the user using our application interface, which gives the user the ability and possibility to monitor and manage his battery from his mobile phone with a simple click.
  
- c. BATMAN proposes to bring significant innovations in technological terms (especially with regard to estimation and diagnostic techniques), realizing at the same time a product idea (creating a prototype) that on the one hand can offer immediate and large-scale feedback on the solutions developed and on the other hand, it can act as a forerunner to a series of applications based on the same technologies, both in areas closely linked to accumulation systems and in areas where advanced diagnostic and estimation techniques can bring significant added value.

### 2.3. Battery Modelling

Lead Acid batteries are commonly used in vehicle applications. They are widely used because they offer a good cost-benefit ratio. Nevertheless, they have plenty of disadvantages which are:

- Low number of cycles compared to other battery technologies
- They require frequent maintenance
- 

However, a good knowledge of the battery behavior is required for any BMS (Battery Management System). As a chemical storage system, the battery has a non-linear behavior. This behavior is affected by:

- Battery State of charge
- Method of charging
- Connected load

As a basic platform, a battery can be modeled by the help of an equivalent circuit diagram that is composed of different circuitry elements.

An impedance spectroscopy (Mauracher and Karden, 1996) can be used as a good alternative of a frequency analysis because it is quite difficult with such a complex and a non-linear behavior, below is a Nyquist diagram which allows us to understand the elements of the equivalent circuit elements. [\[5\]](#)



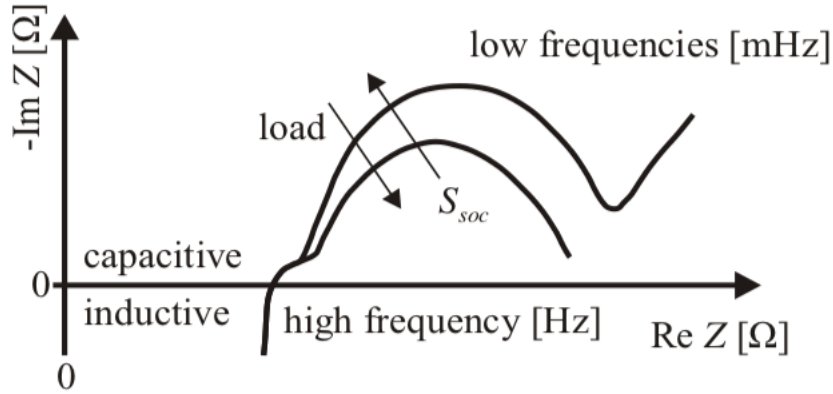


Figure 5-

Lead-Acid Batteries Nyquist Diagram Resulting from and Impedance Spectroscopy

Generalized

The above figure represents an inductive behavior for high frequencies and a capacitive behavior for low frequencies. Therefore, a simpler battery mode can be realized with the internal resistance  $R_i$  to model the battery, along with an inductive impedance  $L$  which is present due to the connection between the battery electrodes and finally a parallel RC circuit which will be represented in the next figure.

Since the battery is an energy source, the core of the model is the  $U_o$  represented in the next figure which is a representation of the battery terminal Voltage which represents the battery open circuit voltage  $U_{oc}$  which is accordingly a function of the battery state of charge.

The load also plays a key factor when modelling a battery, in a way that this this open circuit is overlaid by over-voltages which are described by the rest of the model elements used to build the battery terminal voltage  $U_{bat}$

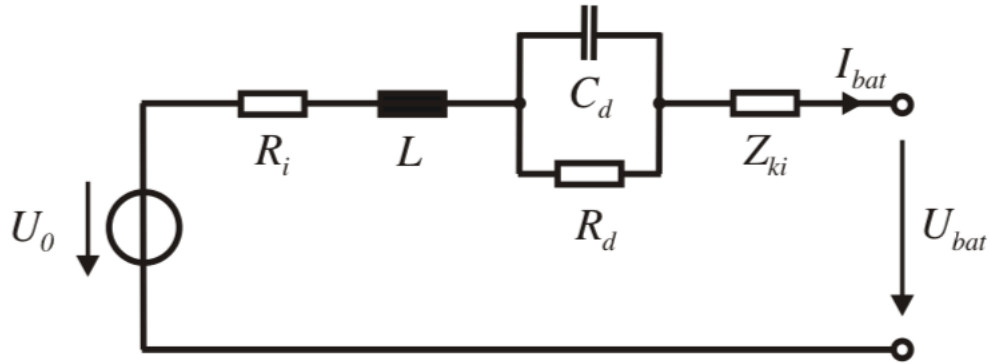


Figure 6 - Battery Equivalent Circuit

Where:

$R_i$  = Battery Internal Resistance

$L$  = Inductive Impedance

$C_d$  = Capacitive Impedance

$U_0$  = Battery open circuit Voltage

$U_{bat}$  = Battery terminal Voltage

$I$  = Battery Op Current

$Z_k$  = Generalized Impedance

Parameters determination of the battery model, the behavior of the battery at its operating conditions (Charging and Discharging) and being exposed to different loads were tested experimentally on different batteries and the following table of parameters were created by experiment.

Manufacturer	Type	Nominal Capacity [Ah]	Nominal Voltage [V]	Cold Cranking Current [A]
Arktis	Lead-Acid	88	12	640
Banner	Absorbent Glass Mat (AGM)	92	12	850
Exide	Gelled Electrolyte	80	12	540

Table 2 - Employed Lead Acid Based Batteries

Moreover, a battery charging/discharging diagram was created in order to monitor the battery voltage and current throughout the experiment. The following results were obtained.

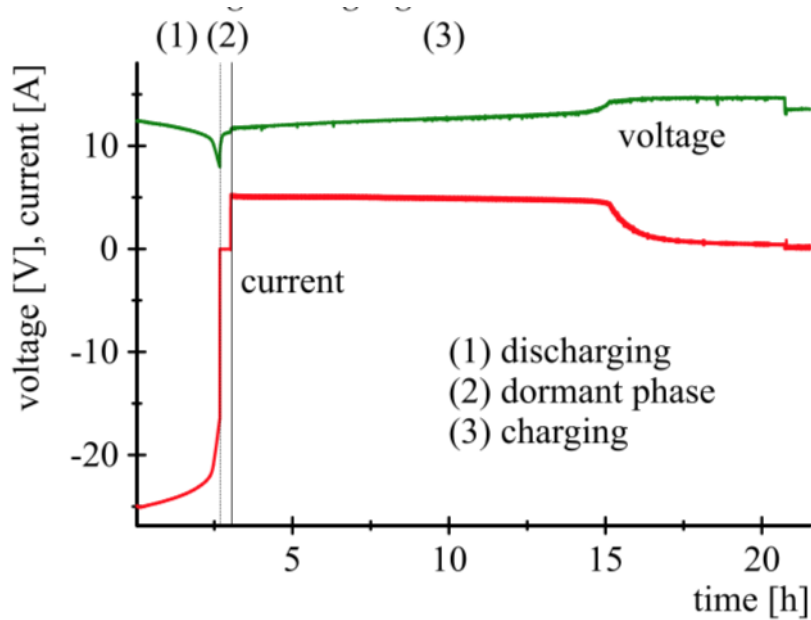
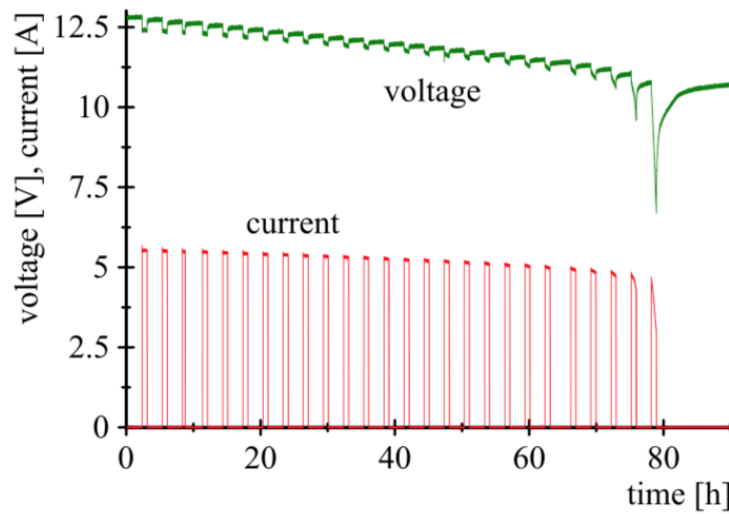


Figure 7 - Course of Voltage and Current While Discharging followed by Charging

Figure (7) represents:

- The first interval (1) represents the discharging phase, as the current is withdrawn from the battery to the load
- The second interval is the dormant phase which is the drained battery phase
- The third interval is the battery charging initiation process

However, the most important phase for a battery management system is the discharging phase. Hence, below is a representation of the discharging phase only with an intermittent current withdrawal

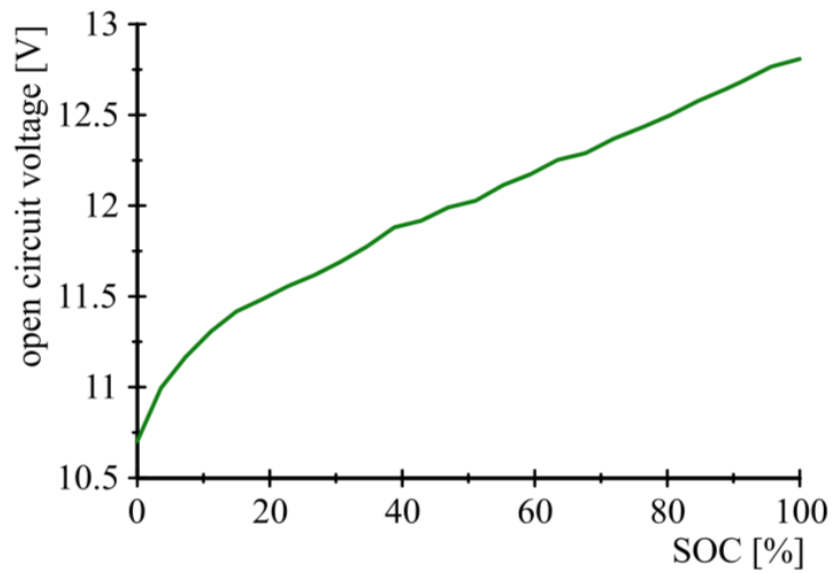


*Figure 8 - Battery Discharging Voltage and Current*

In the above figure:

- Discharging time = 45 minutes
- Withdrawn current = 5A approximately
- Intermittent phases = 2 hours
- 

Finally, a good knowledge of the battery Open circuit voltage is essential for any battery management system as it is the key for finding the remaining charge in the battery, which is represented below:



*Figure 9 - Battery Open Circuit Voltage vs State of Charge*

Figure (9) is the final outcome of the battery model. As by acquiring this graph, an easy correlation between the battery Open circuit voltage (which is easy to measure) and the battery state of charge is obtained which is the key correlation for any battery model.

### 3. Model Verification and Analysis

#### 3.1. Battery Model Verification

Before starting any analysis in order to find an algorithm for the BATMAN Algorithm, the battery model developed should be verified first. Verified in a way that the battery model behavior must be similar to the behavior of a battery in real life. Because if there was any inaccuracy in the battery model, it means that all the analysis that will be carried on in the future using the same model was based on errors, in which the authenticity of the model will be questionable. In order to avoid that, a model verification was required to ensure that the further tests are based on a solid model with realistic and actual results.

As a result, we had gathered actual Current and Voltage data from previously monitored batteries, then compare them with the ones simulated by our model.

Tools used for the Verification Experiment:

- a. Data from previously Monitored Batteries
- b. MATLAB Coder
- c. SIMULINK platform

Procedure:

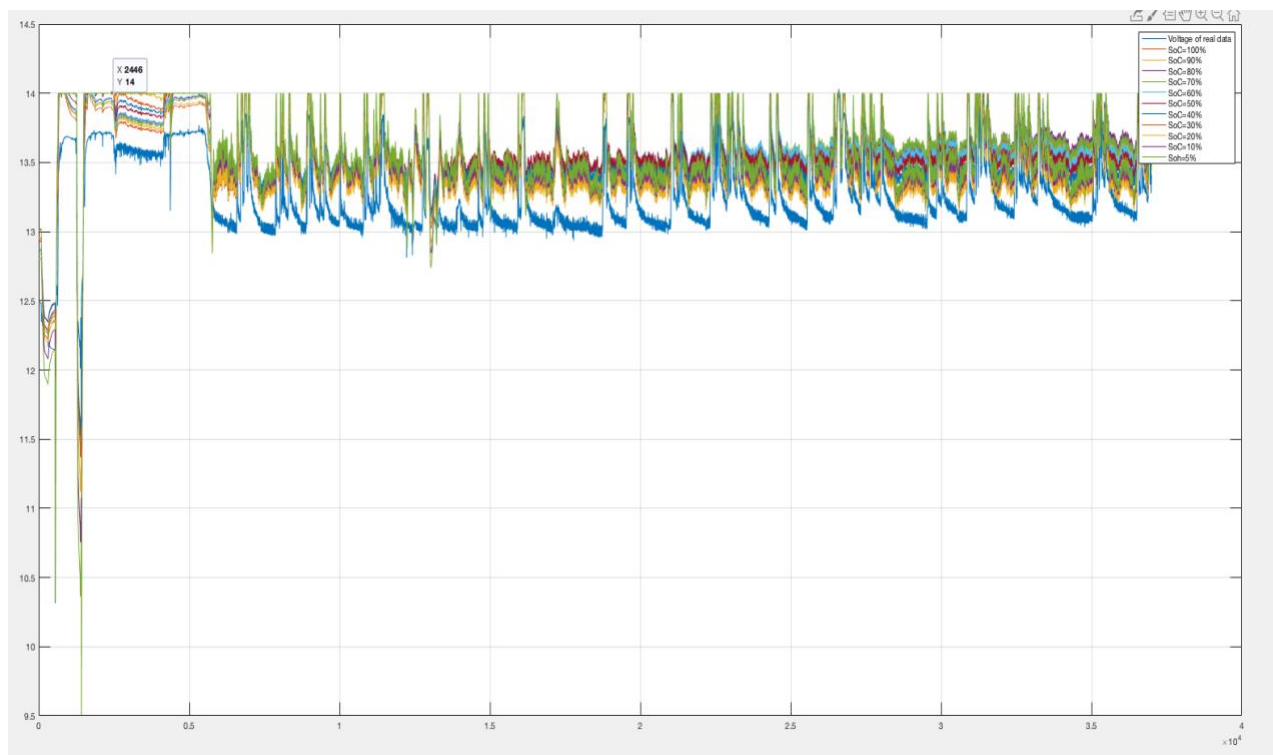
- Developing a MATLAB Script
- Developing a SIMULINK Model in consistency with the MATLAB script
- Generate data using the model and plot the data obtained from real batteries
- Analyze the data and compare real data to simulated ones

### 3.2. Real Voltage Vs. Simulated Voltage

Using our Real battery voltage, below is a list of comparison between the real voltage and the simulated ones on MATLAB-SIMULINK

(Please note for the next figures, on the X-Axis is time in seconds and the Y-Axis is the Terminal Voltage in Volts)

#### a. Battery 1 using (DATA1)

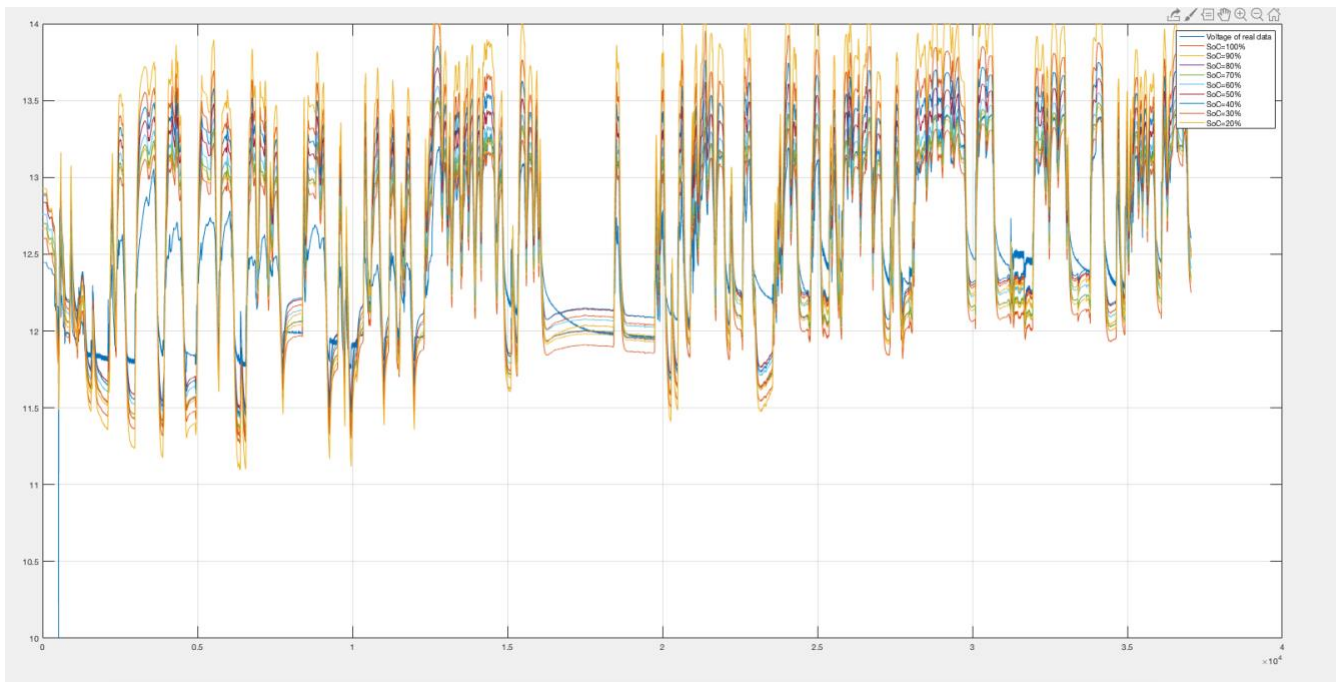


*Figure 10-Battery 1 Verification*

The previous figure shows the Voltage of the real Batteries Vs the simulated one at different (SOH) as the battery model changes depending on the current SOH.

The battery model appears to be consistent with the real battery voltage. But up to this point, verification is not enough as more models are needed.

b. Battery 2 using (DATA2)



*Figure 11-Battery 2 Verification*

In figure 6, it also appears for the battery model simulations to be consistent with the real data, even with different SOH over the course of the experiment.

Nevertheless, it is noticeable in the beginning of the graph that the dynamics in the cranking phase is also consistent with the real data, though some edges are stiff but overall it is a very good result.

It is also noticeable the correlation between the State of Health and the Voltage. In a way that it is a direct relation. when the SOH decreases, the maximum voltage decreases accordingly.



c. Battery 3 using (DATA3)

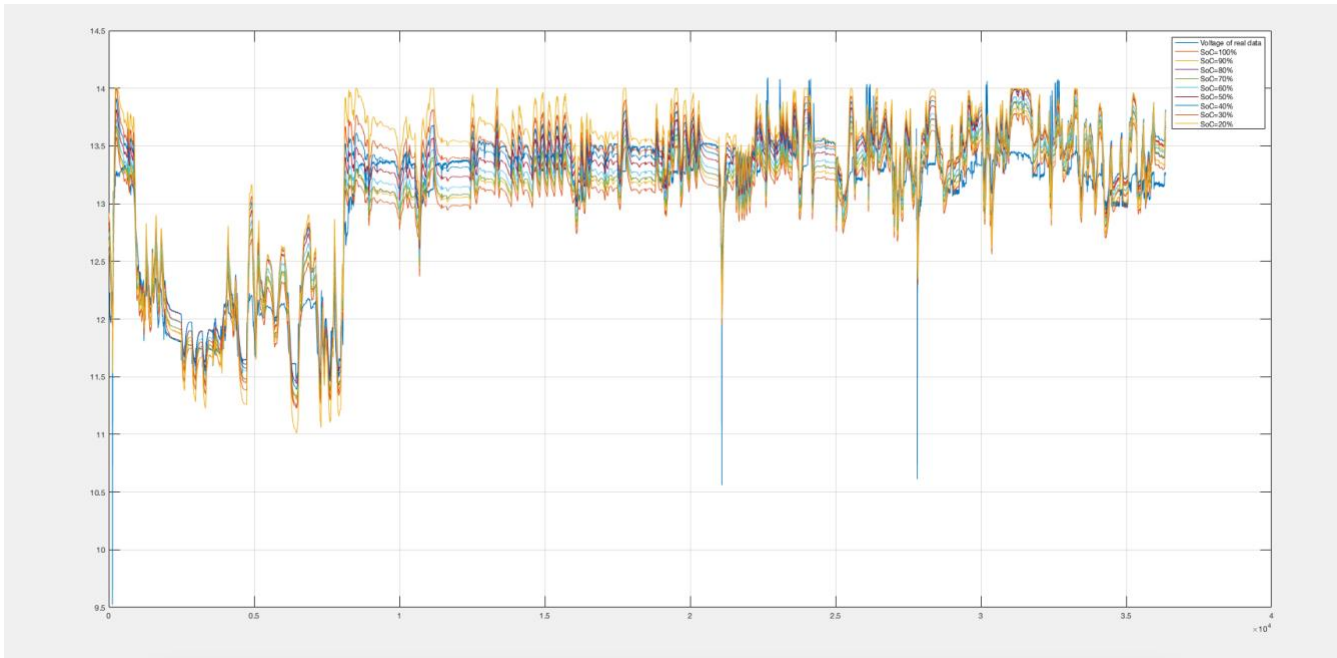


Figure 12-Battery 3 Verification

By testing a 3<sup>rd</sup> battery, it appears for the model to be following the behavior of a real battery perfectly, but to be precise a final test was performed

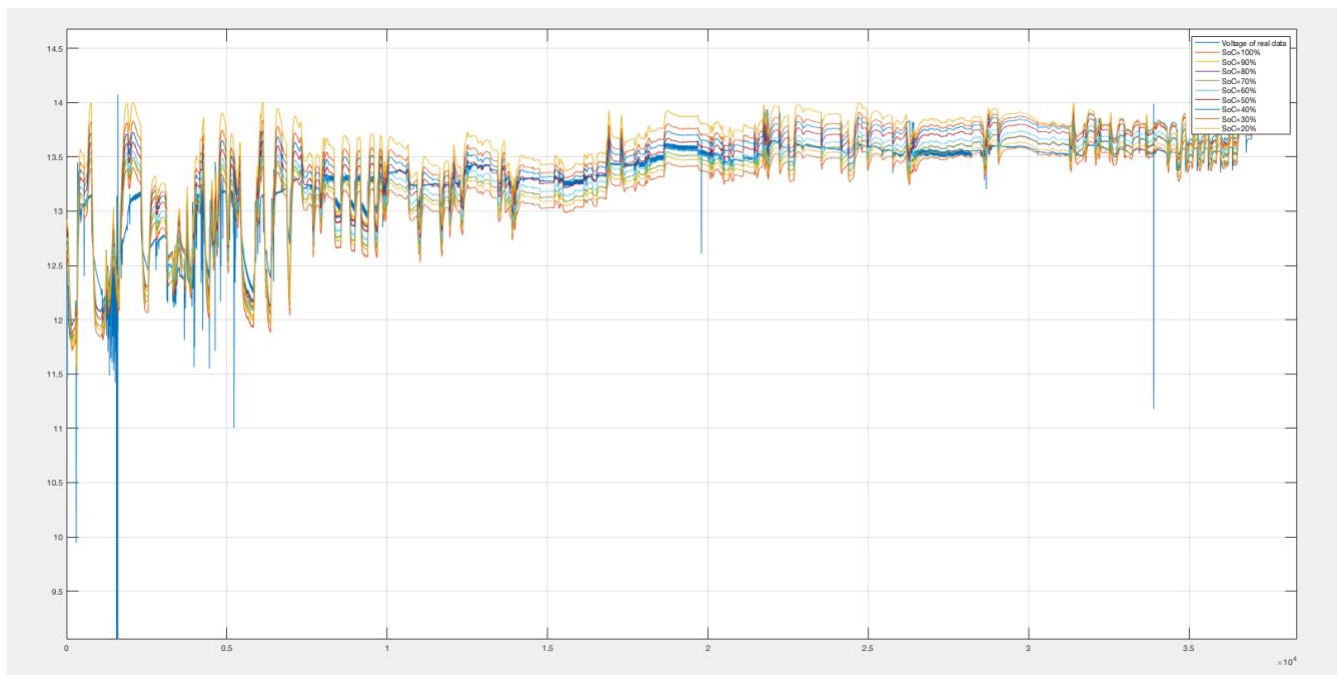


Figure 13-Battery 3 Verification

## Model Verification Conclusion

In conclusion, the battery model developed by our colleagues is consistent, as this model was tried with 4 different batteries that their data was obtained in different conditions. In addition to the fact that these batteries have different battery capacities in (Ah), which means the batteries have different specifications. Yet, our model was consistent to those changes and had persistent output voltage. In this way, we can proceed.

### 3.3. Output Voltage Analysis

To proceed with the analysis, only voltage output from battery 1 was used to develop the algorithm, however, after finishing the model it was tested with all types of data. Also, the analysis was performed by all different types of voltages and it gave the same results.

#### 3.3.1. Observations

First observation was the peak to peak voltage value, which is:

For different (SOH):

$$\text{PeaktoPeak Voltage} = V_{\max} - V_{\min} \text{ (for successive peaks)}$$

The Peak to peak voltage seems to be almost the same until a certain point, the PeaktoPeak voltage starts to increase. This break point is the 40% marker in the state of health level

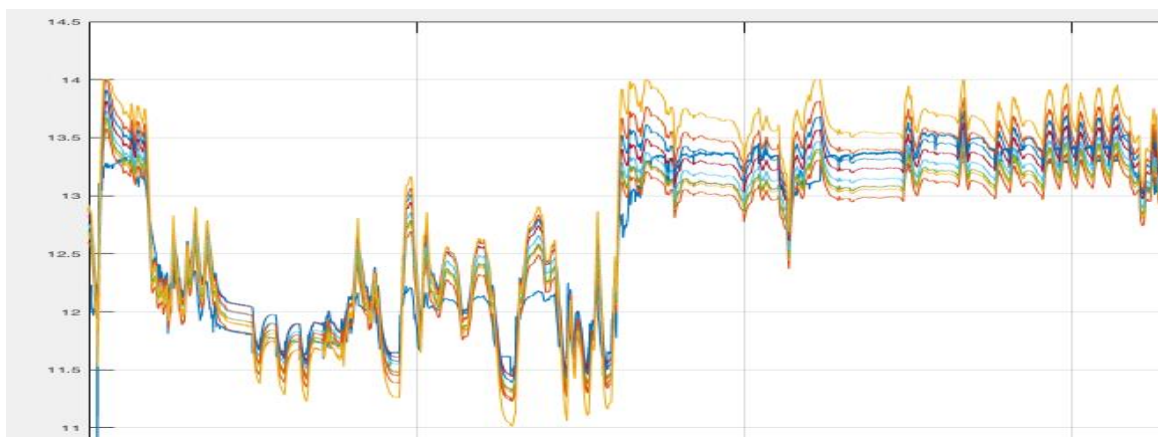
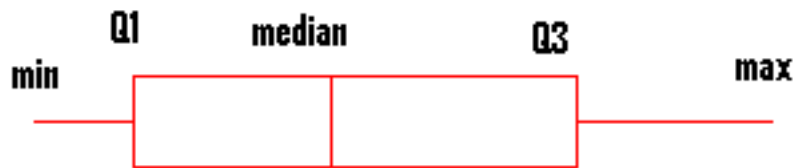


Figure 14-PeaktoPeak Voltage

### 3.3.2. Box Plots

The previous plot is a plot of voltage over time, so it is not very expressive in terms of peak to peak voltages, thus, a box plot was needed in order to have a deeper view.

A box plot in descriptive analysis is a method of representing groups through a box shaped plot which gives the maximum and minimum values, the median and the standard deviation of points outside of the dense area of the boxplot.

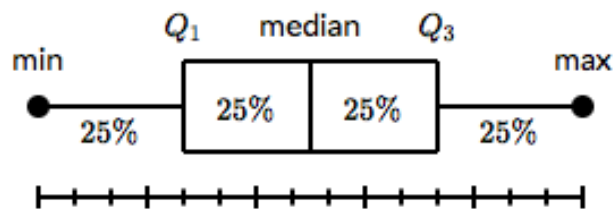


*Figure 15-Boxplots Representation*

Where:

- Max: is the maximum value in the Data Set
- Min: is the minimum value in the Data Set
- Median: is the middle value of the Data Set
- Q1: is the 1<sup>st</sup> quartile which is the median of the Data Set below the median
- Q3: is the 1<sup>st</sup> quartile which is the median of the Data Set Above the median

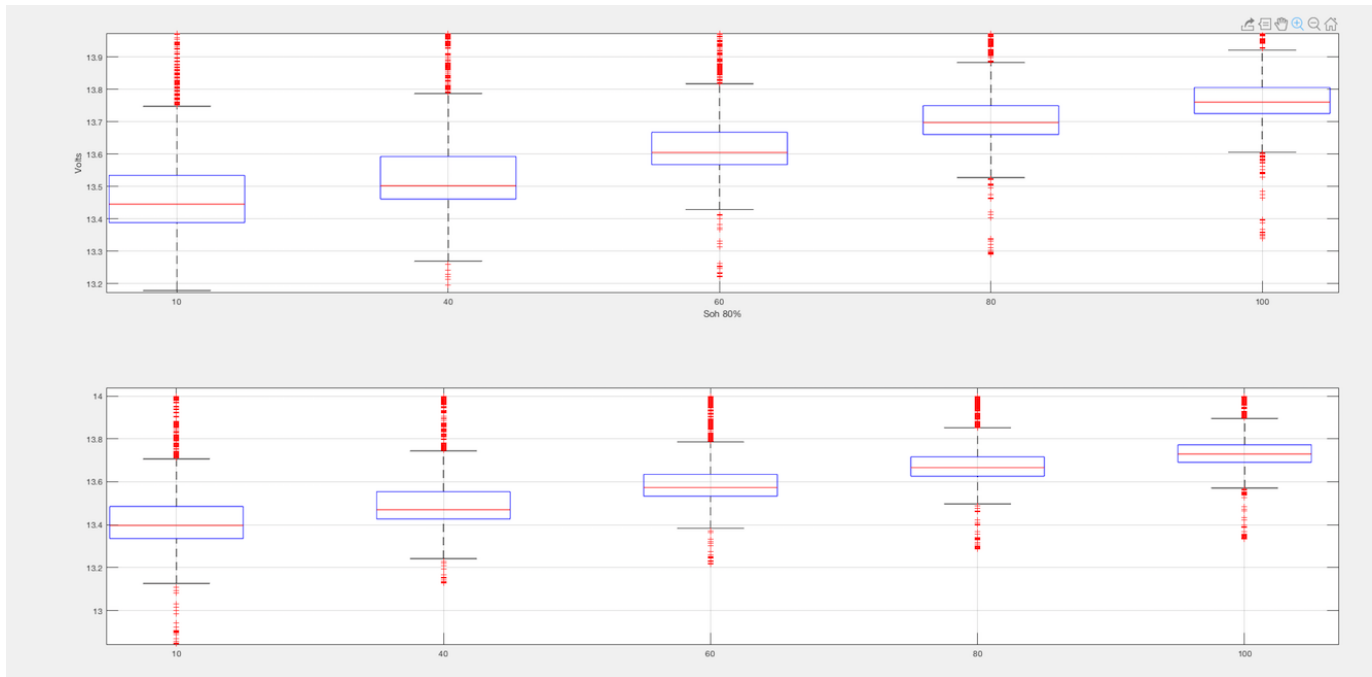
Each of the above parameters divide the Data Set into 4 areas almost 25% each of the full data



*Figure 16-BoxPlots Percentages*

The box plots can be horizontal or vertical, it does not matter the layout of the plot.

- By box-plotting the output voltage data, here are the results



*Figure 17-Voltage Data Boxplot*

These results show a difference in the voltage peak to peak difference, unfortunately, there was not any visible pattern, unless we start to address the slope of the peak to peak voltage. Because the slope is an indication of any increase or decrease in the peak to peak voltage.

This matter will be addressed in the next chapter

## 4. Algorithm key parameters and constraints

### 4.1. Slope Analysis

After the detection of the peak to peak voltage after the boarder of the 40% SOH, it seemed convenient to study the behavior of the sloe of the voltage, aka. rate of change of voltage over time. Because the peak to peak change means a change in the slope of the voltage.

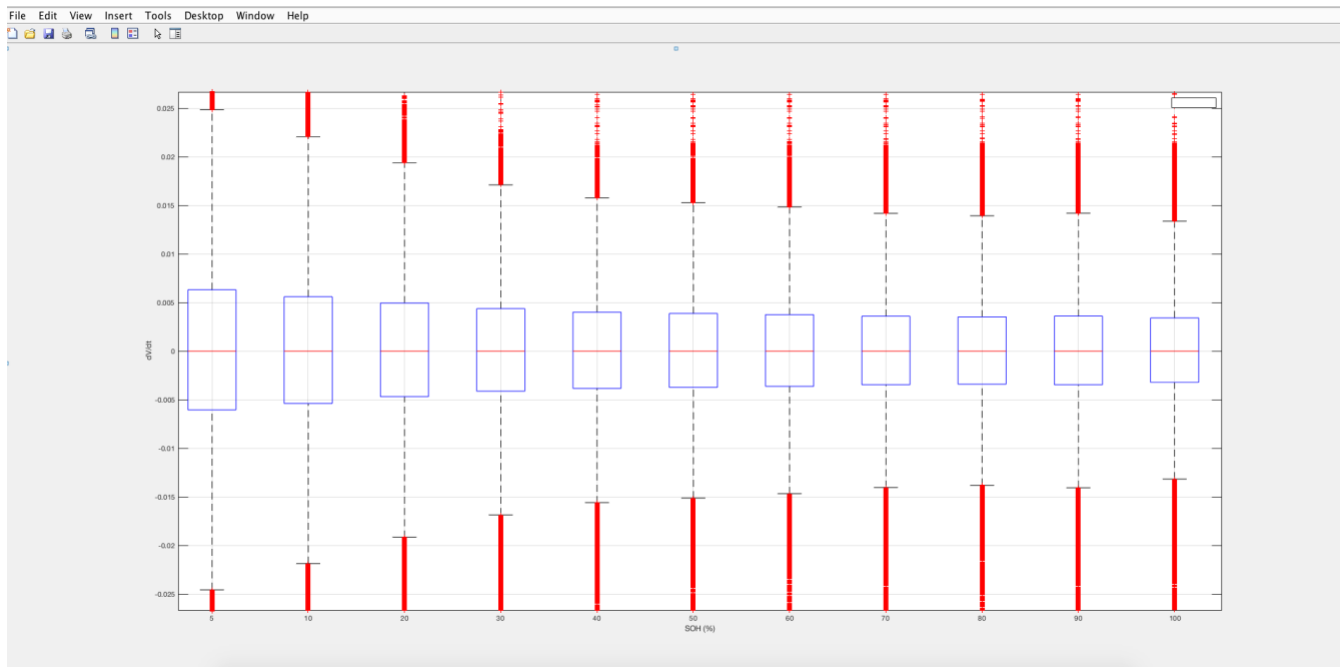


Figure 18- Slope Boxplot

In order to have a closer look on the previous data, a pattern was detected, in which the peak to peak voltage starts to increase after 40% of battery health. There is a clear breakpoint between all the previous State of Health before and after the 40%.

This observation means that, we could possibly find the state of health of the battery if it decreases below 40%, which in fact is a progress with this little data available.

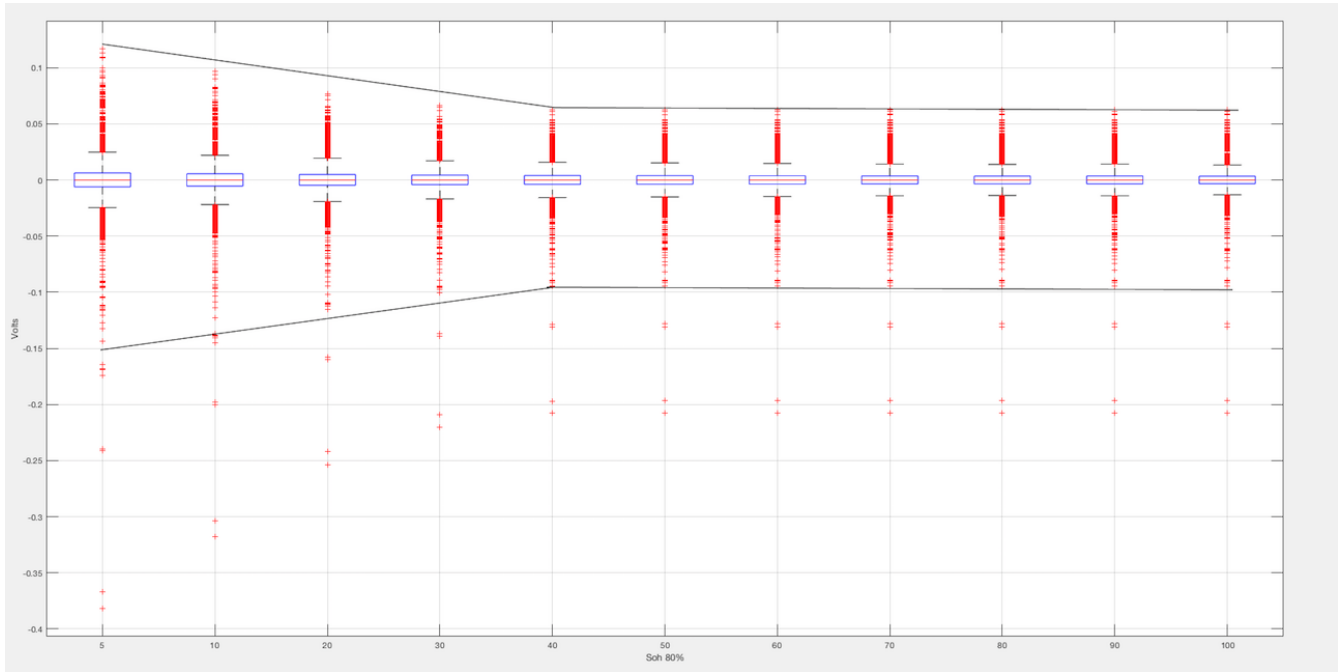


Figure 19-Slope Boxplot pattern

In the previous figure, the pattern to be addressed is clearly visible, and it certifies the relation between the State of health and the voltage rate of change peak to peak difference.

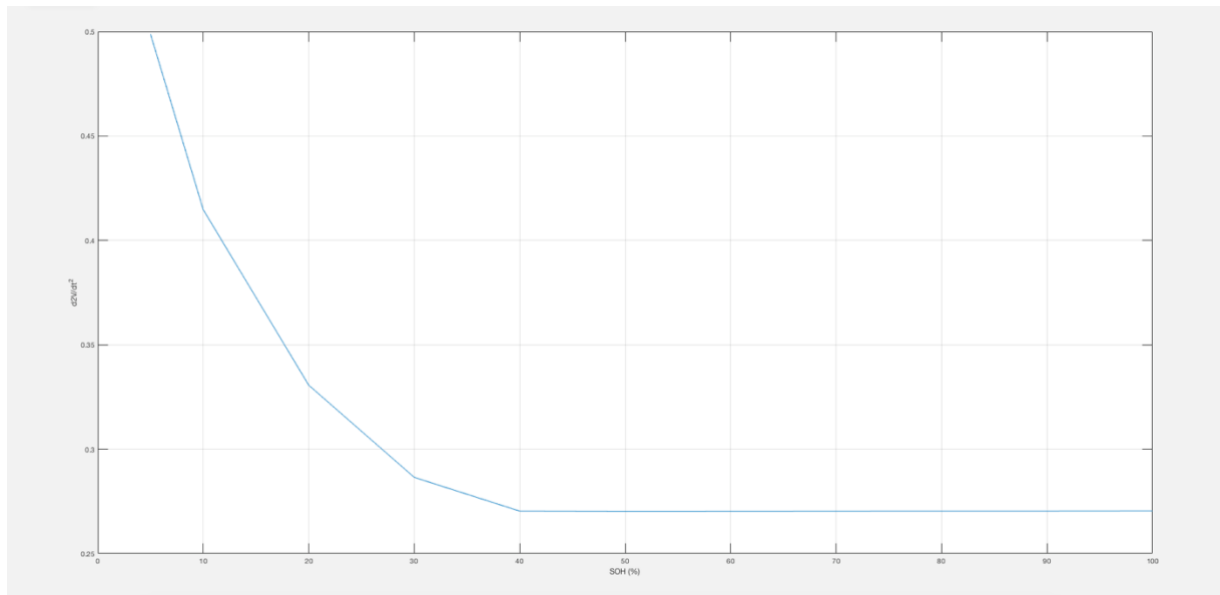
#### 4.2. Parameter 1 $\longrightarrow (\delta)$

Hence, the first parameter that concerned us was the previous (Maximum Peak to Peak voltage) and we gave it a notation of a lowercase delta ( $\delta$ ). Later, this parameter and other parameters will be generated from different models and all fed to a Support Vector Machine for pattern identification in which will be discussed in further chapters.

### 4.3. Parameter 2 $\longrightarrow (\Delta)$

The first conclusion drew us to investigate more the Peaks and the change of it. The second interesting parameter was the rate of change of the previous parameter ( $\delta$ ).

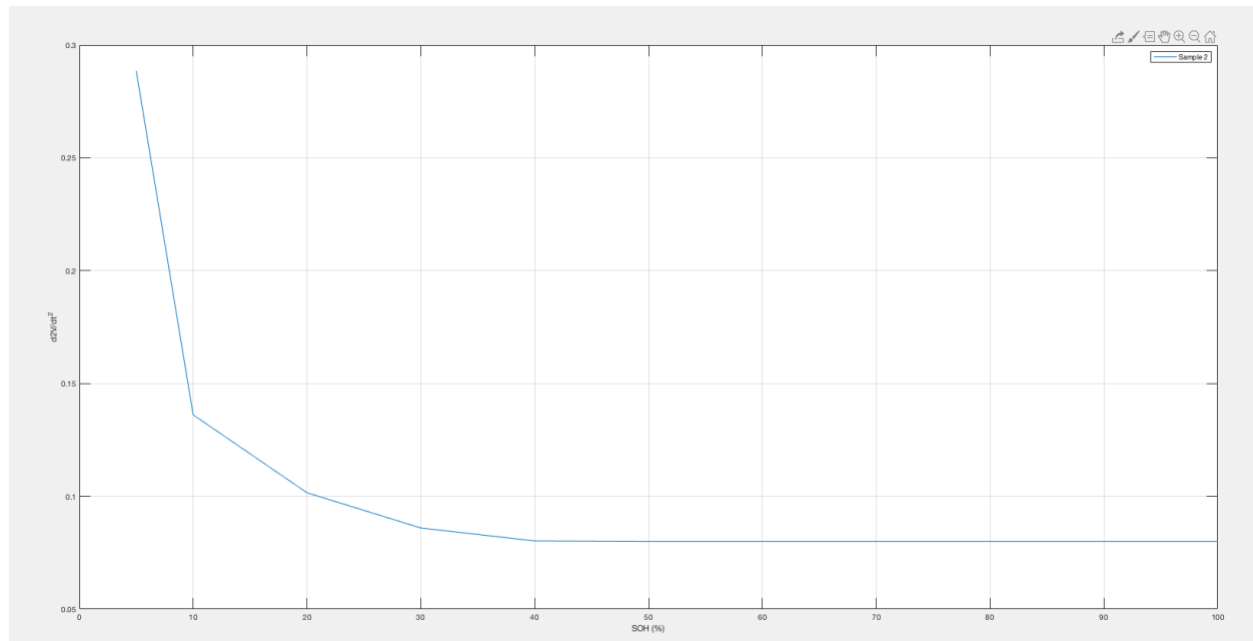
The idea came up because it was necessary to find another parameter that gives the same indication by using a single plot over time. In addition to the fact that it will strengthen the identification process for the state of health Estimation Process.



*Figure 20-Parameter 2 Plot Data*

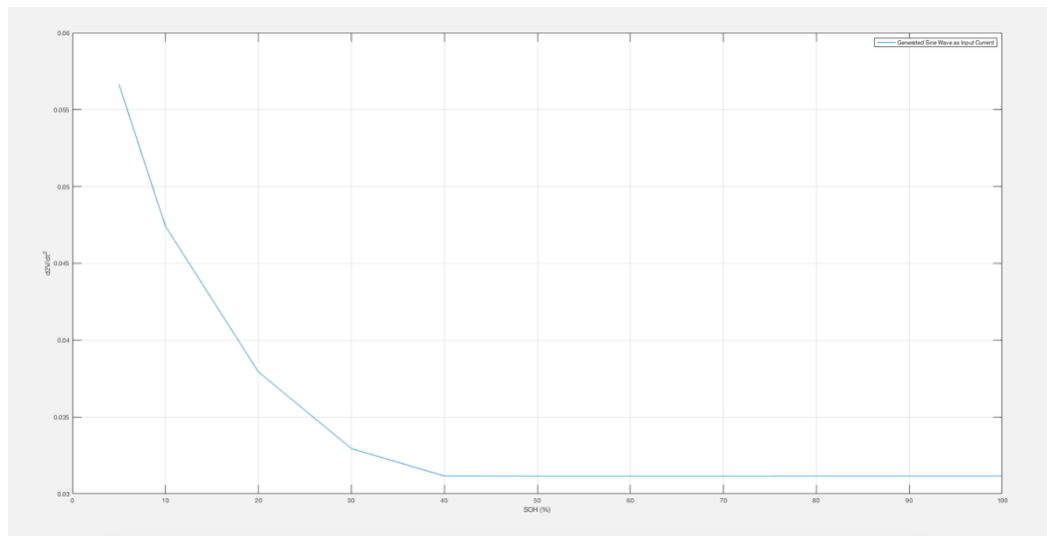
The previous figure shows the rate of change of the peak to peak maximum and minimum difference of the slope of the whole data set. It ensures that the previous parameter is solid, and it gives us the chance to use this parameter as a stand-alone factor.

It is clearly evident that below 40% of (SOH) there is an increase after a constant value over the higher (SOH). Different current inputs were tested, not only real current profiles taken from real batteries, but also imaginary slow and fast input currents



*Figure 21-Parameter 2 Plot Data 2*

Figure (16) illustrates the same parameter but obtained from a different battery current input. Same results are obtained.



*Figure 22-Parameter 2 Plot Sinewave*

Unlike the previous data, figure (17) illustrates the same parameter but with imaginary input data, just to ensure that this parameter has solid results over random input currents and that it doesn't only apply to real data, and the results were as promising as expected. Which leads to one conclusion which is that these parameters are independent from the current withdrawn from the battery.



- Constraints:

As a first constraint that was taken into consideration is the initial State of Charge of the battery.

After applying few tests that will be shown below regarding parameter 2, it was clear that the estimation process can be only held if and only if the battery has a relatively high initial state of charge. As if the battery is low on charge, the battery behavior is unpredictable, so any estimations done will be inaccurate.

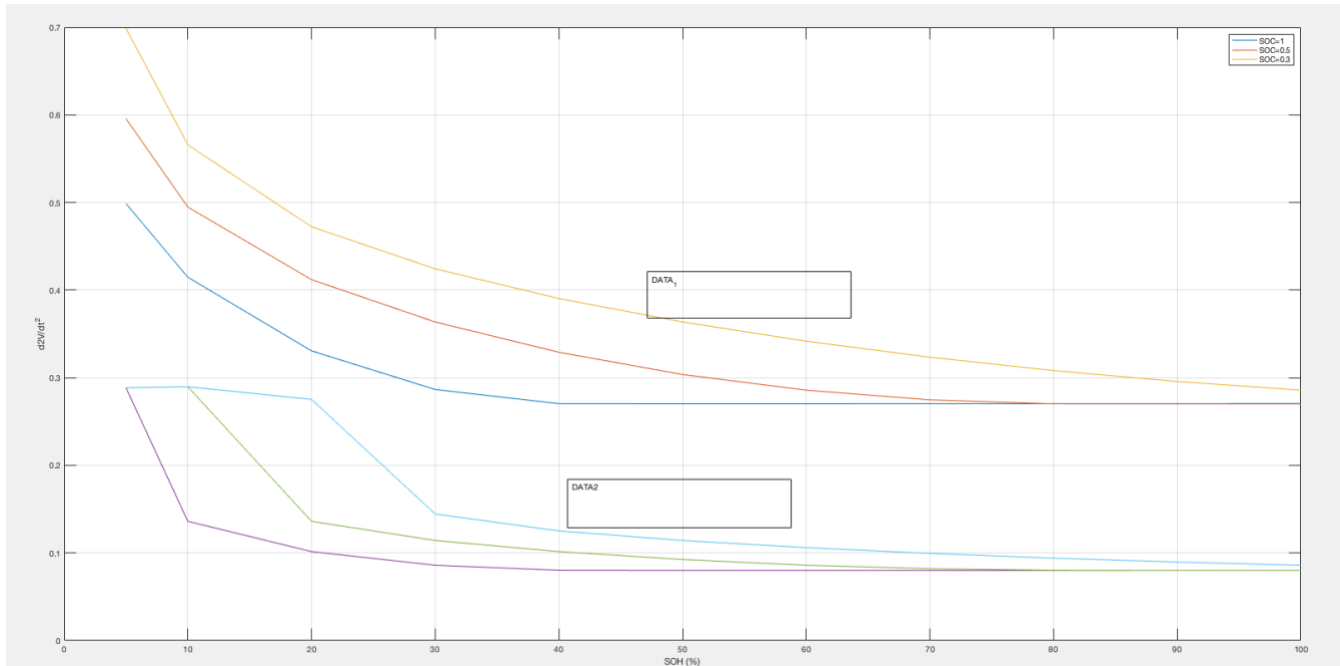


Figure 23 - Parameter 2 with different initial SOC

Figure 18 illustrates the following:

- First and second plot were generated with different input data, both show the relation between parameter 2 on the vertical and different battery state of health on the horizontal.

Plot with DATA1:

- Yellow plot is the data generated with battery initial state of charge which is 30% of the battery State of health (Very low initial SOC). This plot illustrates the fact in which that if the battery's initial SOC is very low, Parameter 2 behavior does not follow the estimation trend, as Parameter 2 is increasing all over the plot from the start to the end, which defies the fact that above 40% SOH the battery is always healthy with parameter 2 constant.
- Red plot is the data generated with battery initial state of charge which is 50% of the battery State of health (Moderate initial SOC). This plot illustrates the fact in which that if the battery's initial SOC moderate, Parameter 2 behavior does not follow the estimation trend, as Parameter 2 is increasing all over the plot from the start to the end, which defies the fact that above 40% SOH

the battery is always healthy with parameter 2 constant.

- C. Blue plot is the data generated with battery initial state of charge which is equal to that of the battery State of health (High initial SOC). This plot illustrates the fact in which that if the battery's initial SOC is high, Parameter 2 behavior follows the estimation trend, as Parameter 2 is increasing all over the plot from the start to the end, which supports the fact that above 40% SOH the battery is always healthy with parameter 2 constant.

Plot with DATA2:

- A. Baby blue plot is the data generated with battery initial state of charge which is 30% of the battery State of health (Very low initial SOC). This plot illustrates the fact in which that if the battery's initial SOC is very low, Parameter 2 behavior does not follow the estimation trend, as Parameter 2 is increasing all over the plot from the start to the end, which defies the fact that above 40% SOH the battery is always healthy with parameter 2 constant.
  - B. Green plot is the data generated with battery initial state of charge which is 50% of the battery State of health (Moderate initial SOC). This plot illustrates the fact in which that if the battery's initial SOC moderate, Parameter 2 behavior does not follow the estimation trend, as Parameter 2 is increasing all over the plot from the start to the end, which defies the fact that above 40% SOH the battery is always healthy with parameter 2 constant.
  - C. Purple plot is the data generated with battery initial state of charge which is equal to that of the battery State of health (High initial SOC). This plot illustrates the fact in which that if the battery's initial SOC is high, Parameter 2 behavior follows the estimation trend, as Parameter 2 is increasing all over the plot from the start to the end, which supports the fact that above 40% SOH the battery is always healthy with parameter 2 constant.
- Conclusion:  
Parameter 2 estimation only applies if and only if the battery's initial state of charge is relatively high.

#### 4.4. Parameter 3 $\longrightarrow (\sigma^2)$

Performing further tests showed that studying the Variance of the Maximum/Minimum Voltage peaks over a window of time seemed interesting and promising.

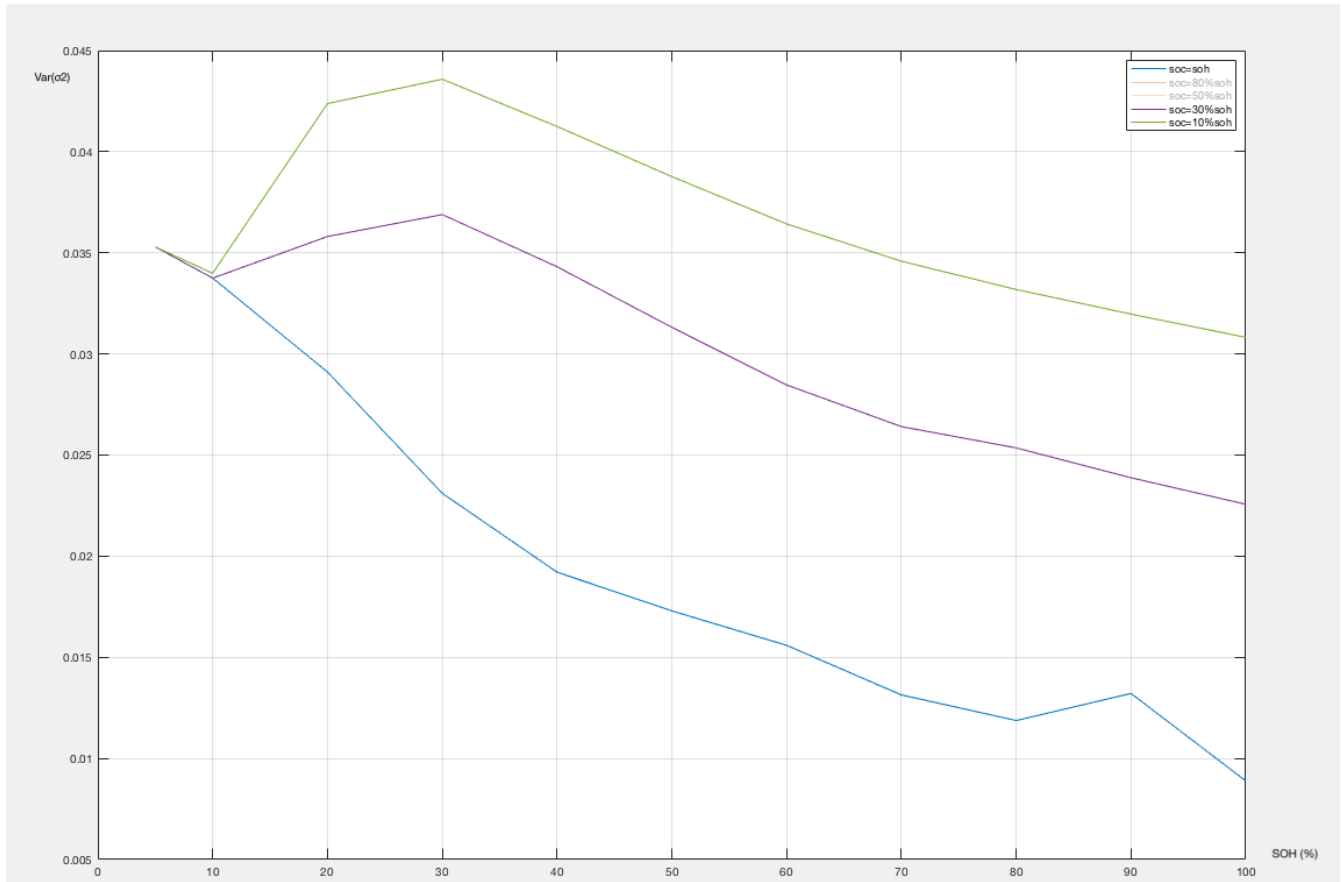


Figure 24 – Variance vs SOH Data

In the above plot:

Input data. DATA1 from Battery 1

- A. The blue plot shows the relation between the variance with respect to the SOH, this parameter shows an increasing trend with the decrease of the battery state of health throughout the whole experiment. This trend was interesting to us as a factor that could be fed to the estimating algorithm, in the sense that:

$$V_{peaktopeak} \propto 1/\text{Variance}$$

Hence, knowing the Variance over a window of time will help us understand the SOH of the battery, as the following cases applies:

- i. Variance is constant over time, which is a representation that the battery SOH is maintained at the same level, not increasing or decreasing but maybe there is a slight change which is undetectable
- ii. Variance is decreasing over a window of time, which means that the battery life is degrading rapidly, and actions should be taken
- iii. Third case which the variance will be decreasing over time is NOT applicable. Because it means that the battery health is automatically regaining over time which is practically impossible without any intervention.

- Constraints

However, the same constraint of parameter 2 still applies for parameter 3, which is the initial State of Charge. Which will be explained in the second and the third plot.

- B. Green plot shows the variance over SOH with initial charge of 10% (Very low initial SOC), but below 30% SOH the curve trend is flipping to decreasing from increasing, which has no physical sense, because as mentioned before, when the battery's initial SOC is very low, the battery behavior is unpredicted. Hence, parameter 3 with very low initial SOC does not follow the detection behavior.
- C. Purple plot shows the variance over SOH with initial charge of 30% (Low initial SOC), but below 30% SOH the curve trend is flipping to decreasing from increasing, which has no physical sense, because as mentioned before, when the battery's initial SOC is low, the battery behavior is unpredicted. Hence, parameter 3 with low initial SOC does not follow the detection behavior.

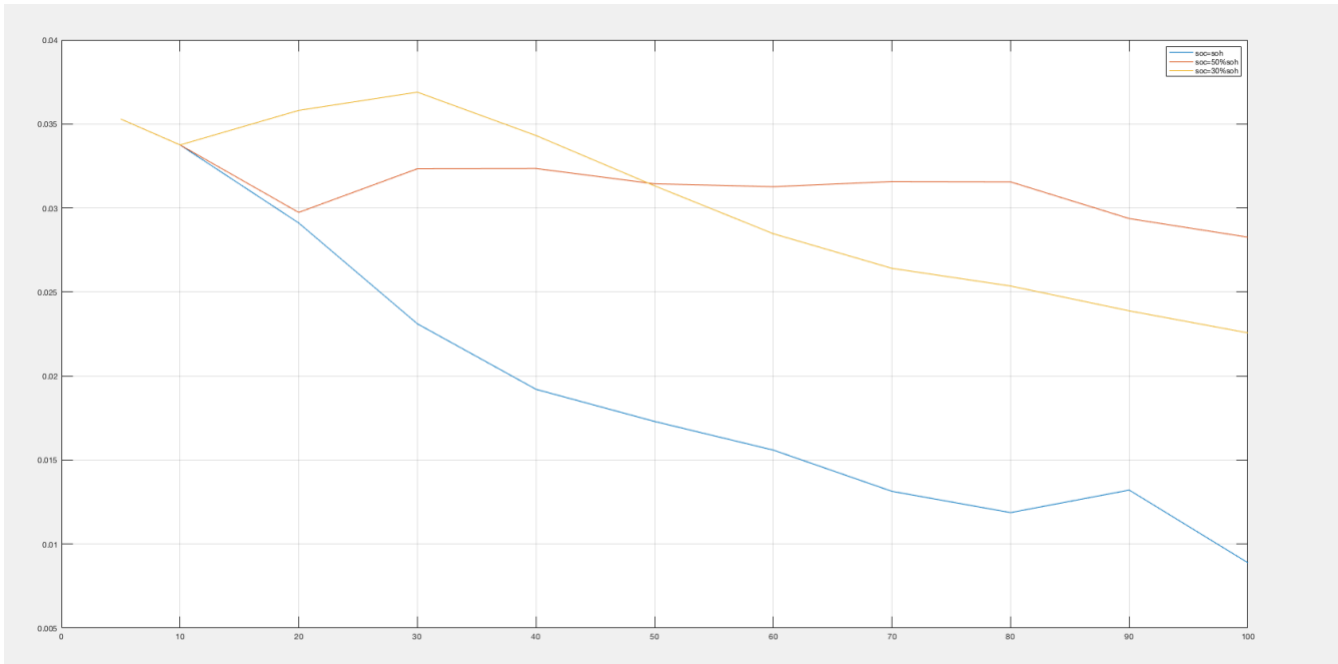


Figure 25 - Figure 19 – Variance vs SOH Data2

In the above plot:

Input data. DATA2 from Battery 2

- A. Yellow plot shows the variance over SOH with initial charge of 30% (Low initial SOC), but below 30% SOH the curve trend is flipping to decreasing from increasing, which has no physical sense. Hence, parameter 3 with very low initial SOC does not follow the detection behavior.
- B. Red plot shows the variance over SOH with initial charge of 50% (Moderate initial SOC), but below 30% SOH the curve trend is flipping to decreasing from increasing, which has no physical sense. Hence, parameter 3 with moderate initial SOC does not follow the detection behavior.
- C. The blue plot shows the relation between the variance with respect to the SOH, this parameter shows an increasing trend with the decrease of the battery state of health throughout the whole experiment, which will be the parameter used for the experiment

- Conclusion:  
Parameter 3 estimation only applies if and only if the battery's initial state of charge is relatively high after testing with two different data sets for verification.

## 5. Classification & Support Vector Machine (SVM)

### 5.1. Support Vector Machine Overview

Next step was implementing the SOH detection algorithm. A classifier method was chosen as our goal is to determine whether the SOH of the battery is higher or lower than 40%. One of the best common classifiers is the Support Vector Machine (SVM).

#### Definition:

A Support Vector Machine is a discriminative classifier mainly functioning by defining a plane that separates the data, which is called a Hyperplane (in multidimensional space). This hyperplane divides the data by creating a separating plane that distinguish the class of the input data accordingly. [\[6\]](#)

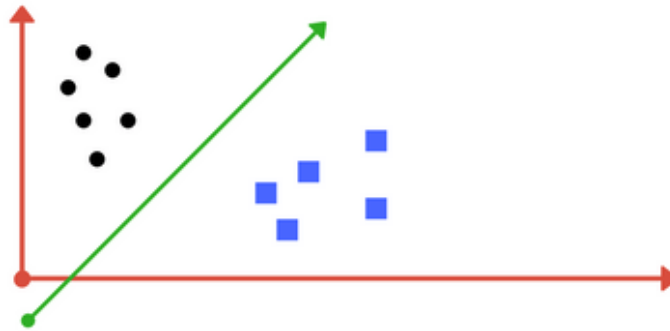


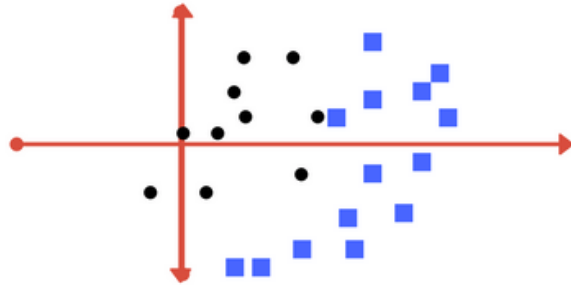
Figure 26- SVM Hyperplane

For instance, the green straight line cuts the dataset into two distinguished borders, classifying new data in order to decide which class it belongs to. Hence, any new input data is analyzed and the SVM algorithm gets to decide which class it belongs to, if it is to the left of the green straight line, so it belongs to the black data set, and vice versa, this process is called ***Separation of classes***.

- SVM parameters
  1. Regularization Parameter
  2. Kernel Parameter
  3. Gamma Parameter
  4. Margin

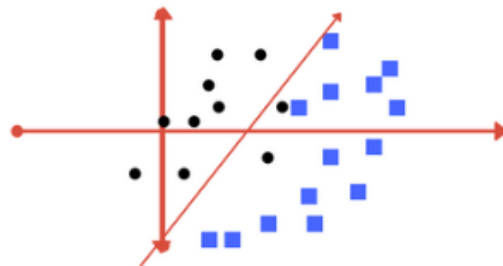
## 1. Regularization Parameter

Complications start to immerge when we have inseparable data sets, or in other words mixed data. As shown in figure (21) the data is completely separable, thus, there is no complications when creating a hyperplane.



*Figure 27 - SVM Regularization Parameter 1*

In figure (22), the data seem inseparable as if we make a hyperplane, some of the data that belongs to a different data set will be in another section, the figure below illustrates this problem



*Figure 28 - SVM Regularization Parameter 2*

The regularization parameter decides how complex the hyperplane is, thus, avoiding any misclassification. Hence, this parameter is a trade-off between the complexity of the hyperplane and the classification accuracy. Keeping in mind that the more complex the hyperplane is, the more compilation time it is needed to actually classify the data, and the bigger the size of the program. This is a problem because this defines the hardware requirements for the processor and memory of the microcontroller. So, the less complex the better in terms of cost and mass production.



However, below is an illustration of a more complex hyperplane avoiding any class mixing, thus any misclassification.

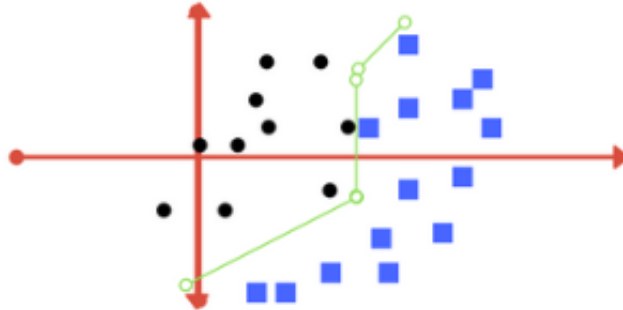


Figure 29 - SVM Regularization Parameter 3

In conclusion to this issue, the hyperplane gives a completely perfect hyperplane cutting the data set into two distinctive sections, while the hyperplane in figure 23 gives a poor classification with definitive error occurrences.

## 2. Kernel Parameter

Basically, the Kernel Parameter is the parameter responsible for creating the hyperplane, which turns the hyperplane creation from data classification to linear algebra. There are many ways of determining the Kernel, for instance, there is a linear and polynomial kernel.

- Linear Kernel

Which for new inputs is defined as the dot product between the new input ( $a$ ) and those supporting vectors ( $a_i$ ) and is calculated as the following:

$$f(a) = B(0) + \sum (C_i * (a \cdot a_i))$$

Where,

$B_0$  and  $C_i$  are the coefficients that will be created from the dataset by the trained algorithm

- Polynomial Kernel

The Polynomial Kernel is more complex. Where its complexity increases with the order of the polynomial, for instance, a third order Kernel will produce a more complex hyperplane than the quadratic Kernel, but the quadratic kernel is less heavy and takes less time to produce an answer.

The Polynomial Kernel is defined as:

$$K(a, ai) = (1 + \sum (a * ai)^d)$$

Where, K is the next step prediction, d decides the order of the kernel whether if it will be quadratic or 3rd order... etcetera.

### 3. Gamma Parameter

The gamma parameter defines the closeness of a certain data to the separation line; thus, low gamma means the data is far away from the separating surface so it cannot be used in the calculation of the separating surface. While high gamma means that the data is close to the separation surface, thus it can be used in the calculation of the separating surface. The following figures shows the difference between low and high gamma.

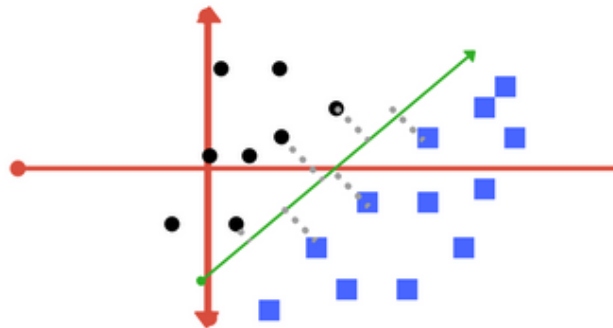


Figure 30 - High Gamma

Figure 25 - shows the high gamma points, where are very close to the separating surface.

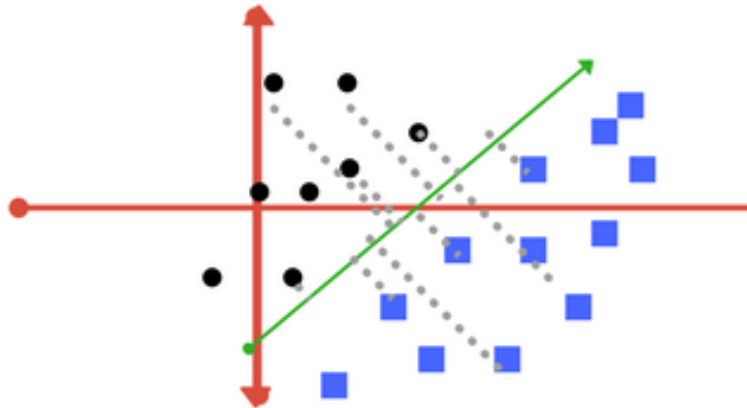


Figure 31 - Low gamma

Figure 26 shows the low gamma points, which can be seen to be very far away from the separating surface

#### 4- Margin Parameter

The margin is the parameter that defines how equidistant the separating surface is from each data set. As the more equidistant the separating factor is, the better it is for the classification process.

The figure above shows a good margin, in which the separating surface is almost equidistant from both classes. While the figure below shows a bad margin as the separating surface is closer to the blue data set more than the black data set.

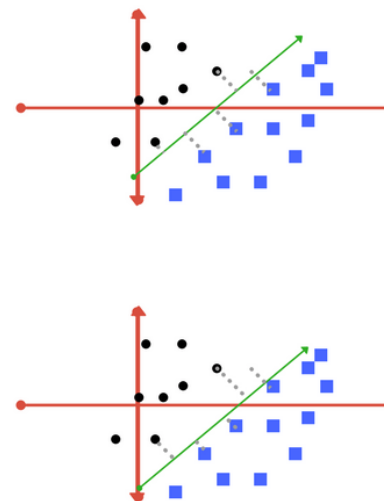


Figure 32 – Margins

Procedure:

In order to produce a machine learning algorithm using support vector machine, we have to provide the following

- a. Input Data Set
- b. Response Data Set
- c. Training Data Set
- d. Validation Data Set

## 5.2. Input data set

The input data set is all the data generated from all simulation, in addition to the real data set that was collected from batteries under observation. In our case, our data set was created using 3 different real batteries data and 3 imaginary current input profiles which may match some operating conditions

Method:

$$\begin{pmatrix} \sigma^2 & \delta & \Delta \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}$$

In which:

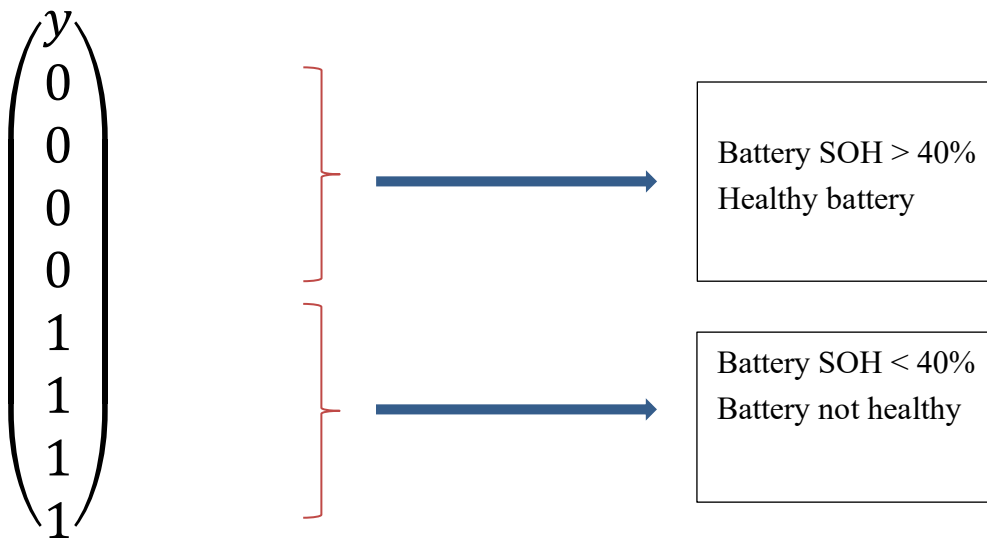
- Parameter 3 fills column 1 vertically
- Parameter 2 fills column 2 vertically
- Parameter 1 fills column 3 vertically

### 5.3. Response Data Set

The response data set is the answer of the algorithm detection, in which, our goal is to determine whether the battery state of health is above or below 40%, hence, our response data set is going to be a matrix of a binary output in which:

- i. If the response yields 0, it means that the battery is healthy, and the state of health is above 40% and there are no issues.
- ii. If the response yields 1, it means that the battery is not healthy and the state of health is below 40% and actions should be taken as the battery is not functioning with its optimum capabilities and not providing the required functionalities.

The response is given a notation (y) where,  $Y = (0,1)$



## 5.4. Training Data Set

The training data set is composed of both the input data and the response data, because this matrix is going to be fed to the support vector machine application, in order to process the data and produce an extractable model which takes any new input value and gives a response.

$$\begin{array}{c}
 \text{Input Data} \qquad \qquad \qquad \text{Response} \\
 \left( \begin{array}{ccc|c}
 \sigma^2 & \delta & \Delta & y \\
 \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot
 \end{array} \right)
 \end{array}$$

In the previous matrix, each row represents a single input. Each row is considered to give a certain response which we already know, hence, this matrix is provided to the learning algorithm in order for the application to study the data set and generate the SVM parameters which we stated in section 1 of this chapter. After generating the model using the data, in real time applications, each of the three parameters will be calculated and put together in a single row to be fed as input to the detection algorithm. Which will subsequently produce a certain response (0 or 1). The more training data you have, the more accurate the model will be, but also, the time needed for the classification process will be longer.

## 5.5. Validation Data Set

The Validation data set is a set of data extracted from the training data, which will be used after training the model to verify that the model is producing the desired output. Furthermore, the validation data set can determine the accuracy of the model in percentage.

The validation data set is chosen according to the amount of data at hand. Which is a trade-off between the time and accuracy, in such a way that:

- If the data set is small, you will be obliged to choose a small validation data set, because the rest of the data will be used to train the model, and with more training data, the accuracy will be accordingly high
- If the data set is big in size, a larger validation set could be considered for more accuracy. But taking into consideration the model size and the time taken for the model training.

## 6. SVM implementation and Testing

### 6.1. SVM details

First step to generate a real time model for testing, is to create the SVM classifier model. This section highlights the support vector machine details that we used in order to create the model.

#### a. SVM order

A ***Quadratic*** support vector machine was used in our case. The decision was made according to different facts which are: Model accuracy, training time and model complexity and size.

#### b. Data set

Our data is composed as illustrated above in the previous section

New Session

**Data set**

Workspace Variable

Test\_final 720x4 double

☒ Use columns as variables  
☐ Use rows as variables

**Response**

column\_4 double 0 .. 1

**Predictors**

	Name	Type	Range
<input checked="" type="checkbox"/>	column_1	double	0.00778223 .. 4.19583
<input checked="" type="checkbox"/>	column_2	double	0.0781761 .. 0.858589
<input checked="" type="checkbox"/>	column_3	double	-0.000209443 .. 0.212372
<input type="checkbox"/>	column_4	double	0 .. 1

Add All Remove All

[How to prepare data](#)

Table 3 - Real Data Set



Our data set matrix 'Test\_Final' consists of 4 columns. The first three columns contain the three-prediction parameter in sequence, then the fourth column is the response 'Y'.

c. Validation data set

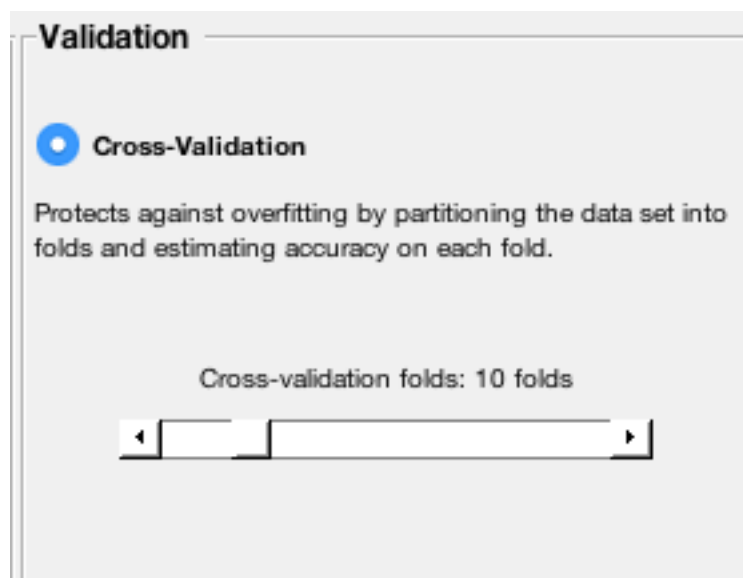
The validation method used is called Cross-Validation, in which it protects against overfitting by splitting the data set into folds, then, estimating the accuracy on each fold.

But, the decision taken to decide how many folds will be used for estimation and how many will be used for validation depends on the amount of data we have, unfortunately in our case, we could not obtain more data other than that we have and those generated from the model.

Accordingly,

Cross-Validation folds = 10 folds

Estimation folds = Data Set Folds – 10(Validation Folds)



*Figure 33 - Validation Folds*

d. SVM training app data

▼ Current Model	
<b>Model 1: Trained</b>	
<b>Results</b>	
Accuracy	95.7%
Prediction speed	~6300 obs/sec
Training time	9.8041 sec
<b>Model Type</b>	
Preset: Quadratic SVM	

*Figure 34 - Training App Details*

Above is the application report regarding our classifier, it shows all the important results and information we need to understand about our model.

- Test Accuracy : 95.7%  
Which is an excellent result, as it is almost a 100% estimation accuracy
- Prediction Speed: almost 6300 obs/sec  
Which is a relatively high speed of operating
- Training time: 9.8041 seconds  
Which is very fast training time, but this is due to the fact that the amount of data at hand is relatively small, thus, the trainer did not need much time to create the model. Also, the input data are almost separable as there is a clear border between the performance of the battery before and after 40%

## 6.2. Model in the loop Testing

After completing the classifier model, a model was to be created in order to test the outcome of the model with different data.

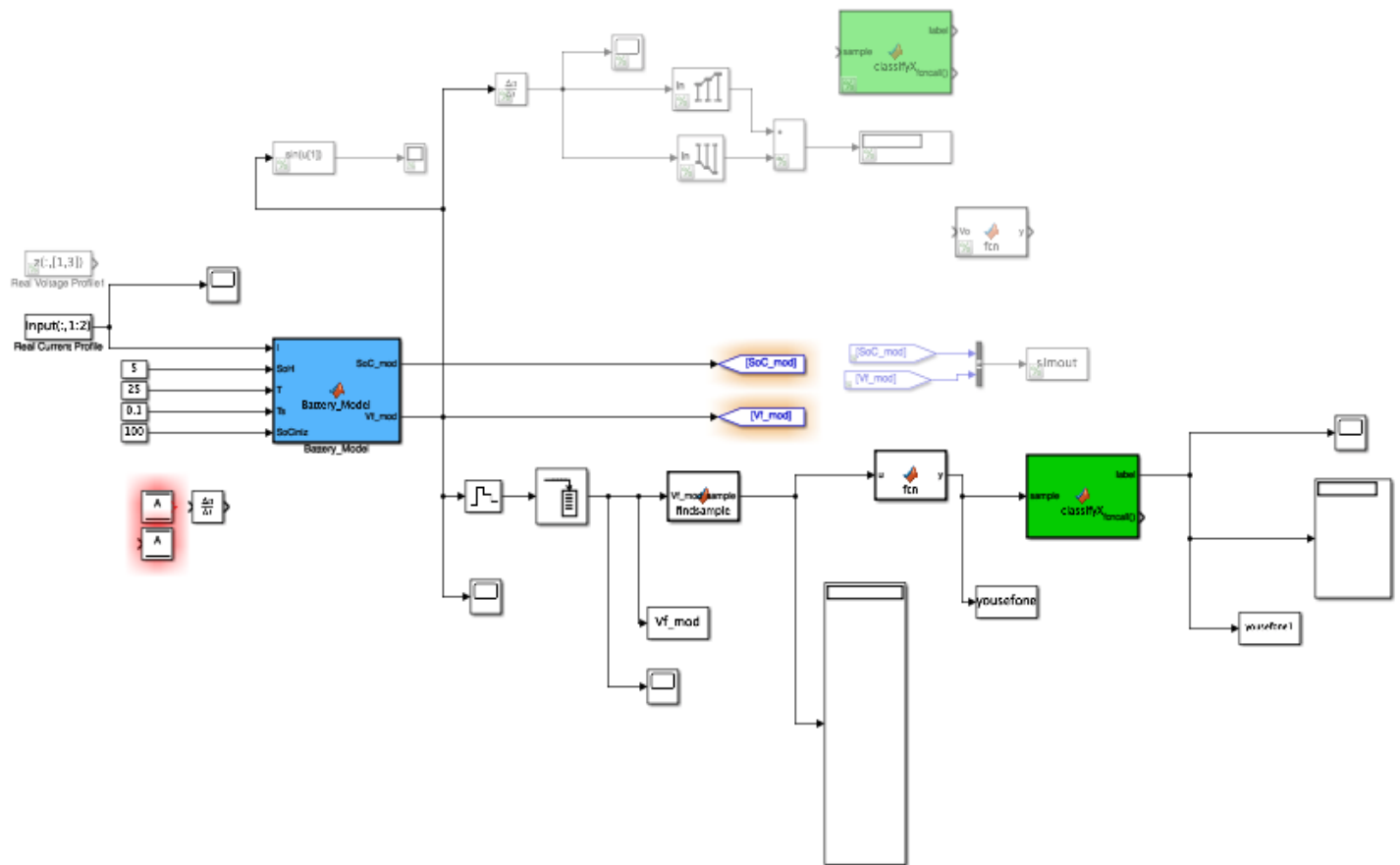


Figure 35 - MIL Model

Test Tool:

- MATLAB\_R2019a
- SIMULINK TOOL

Test Procedure:

- The input data set was taken from a previously monitored battery in order to ensure that the results obtained apply to real data extracted from batteries and not by model generation
- The output voltage is extracted from the battery model and fed into the MATLAB script for processing and calculating the parameters. ( all calculations and MATLAB coding are available at the end of the document in the Appendix section )
- Finally, the calculated parameters are fed directly into the SVM model and within a certain time delay the output is visible

Test Results:

	1	2
1	1	
2	1	
3	1	
4	1	
5	1	
6	1	
7	1	
8	1	
9	1	
10	1	
11	1	
12	1	
13	1	
14	1	
15	1	
16	1	
17	1	
18	1	
19	1	
20	1	
21	1	
22	1	
23	1	
24	1	
25	1	
26	1	
27	1	
28	1	
29	1	
30	1	
31	1	
32	1	
33	1	
34	1	
35	1	

	1	2
30	1	
31	1	
32	1	
33	1	
34	1	
35	1	
36	1	
37	0	
38	0	
39	0	
40	0	
41	0	
42	0	
43	0	
44	0	
45	0	
46	0	
47	0	
48	0	
49	0	
50	0	
51	0	
52	0	
53	0	
54	0	
55	0	
56	0	
57	0	
58	0	
59	0	
60	0	
61	0	
62	0	
63	0	
64	0	
65	0	

	1	2
69	0	
70	0	
71	0	
72	0	
73	0	
74	0	
75	0	
76	0	
77	0	
78	0	
79	0	
80	0	
81	0	
82	0	
83	0	
84	0	
85	0	
86	0	
87	0	
88	0	
89	0	
90	0	
91	0	
92	0	
93	0	
94	0	
95	0	
96	0	
97	0	
98	0	
99	0	
100	0	

Table 4 – Non-Real Time test results

The previous test was done starting from a battery at 100% State of health, this battery is supposedly degrading until it reaches 1% of State of health (as 0% State of health is a factionary state).

Where (yfit) is the predictor answer that yields (1) for low (SOH) and (0) for high (SOH)

- The test should yield:

$$0 = \text{SOH}(100\% - 40\%)$$

$$1 = \text{SOH}(40\% - 1\%)$$

- Actual results

$$0 = \text{SOH}(100\% - 37\%)$$

$$1 = \text{SOH}(40\% - 36\%)$$

Test error percentage:

Since the test should give us an indication at 40%, rather it gave us an indication at 37% which means that there is a 3% of error margin in the model.

Accordingly:

Model Accuracy : 97%

This is an excellent result for a phase one project that will be further tested, developed and improved. Moreover, after releasing the BAT-MAN device into the market, we will have the ability to collect more data from real battery performances. Giving the change for the model to train more effectively and produce more accurate and faster results.

## 7. BAT-MAN in real time implementation and Testing

### 7.1. Real Time Model

A real time model is a model that executes at the same rate as the actual clock, in which it provides data at the actual timings that the application will run through.

It is important to implement a real time model in order to test the algorithm in real time with real data, calculating the error percentages and time delays. Moreover, a real time model is needed for code generation for the application which is the final interface of BAT-MAN.

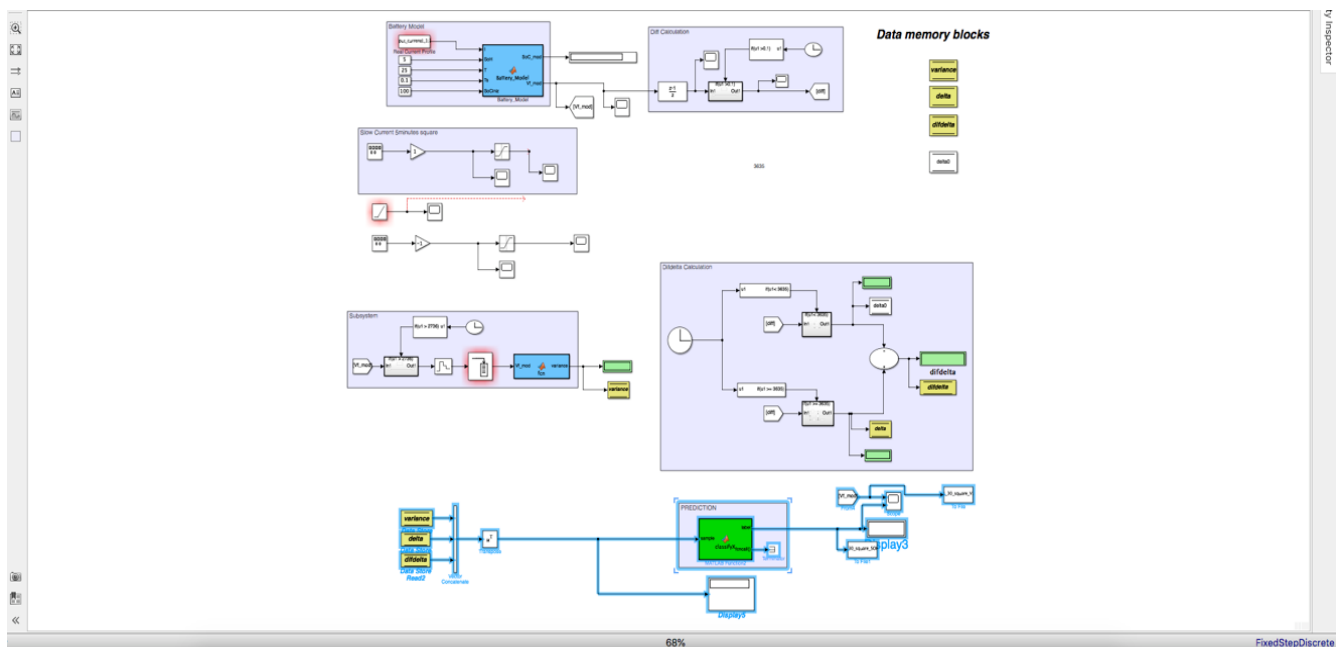


Figure 36 - Real Time Model

### Model Information

- Model creation and code generation are to be implemented using MATLAB SIMULINK
- Input data is real data extracted from previously monitored batteries to give real results

## 7.2. Test Initialization

### Test Procedure

- a. Input data is fed to the battery model to be processed
- b. Prediction parameters are calculated and stored in memory
- c. Prediction parameter are fed into the Support Vector Machine application for prediction

### Test and model output time

Tests were applied by analyzing data by fixing a window of time, starting from different (SOH) and slowly decaying the (SOH) over the window of time. The algorithm was able to detect the changes but with a delay for the algorithm to detect the change in (SOH).

***Testing time*** is approximately 72000 Seconds (20hours)

### Testing Data

Tests were made using real data and simulated data, in order to verify the correct operation of the algorithm detection regardless of the input data source.

### Test conditions

- The ultimate condition for optimal results is that the battery should be sufficiently charged. In which the battery State of Charge (SOC) should be larger than (50%) as mentioned in the previous analysis sections.
- Battery initial temperature: 25 degrees Celsius. However, the algorithm and battery model does not show any significant reaction to the change in temperatures.
- Battery initial state of charge is initially different for each test, in order to detect if the algorithm reacts correctly to the battery initial state of health.
- Tests were applied with initial State of Health of (80%, 50%, and 30%) of real input data and (50%, and 30%) of simulated input data.



### 7.3. Testing using Real input data

Please note that all of the following figures in this section and the next section represent the following:

- Upper plot (voltage) in Volts
  - Lower plot (Binary 1,0) represent the answer of the prediction which yields a binary 0 when the battery state of health is greater than 40% and yields a binary 1 when the SOH gets below 40%.
  - Both the upper and lower figures are simulated Vs Time in seconds on the X-Axis)
  - Initial State of health is referred to as SOHi
- 
- Test 1
  - Initial SOH = 80% using real data

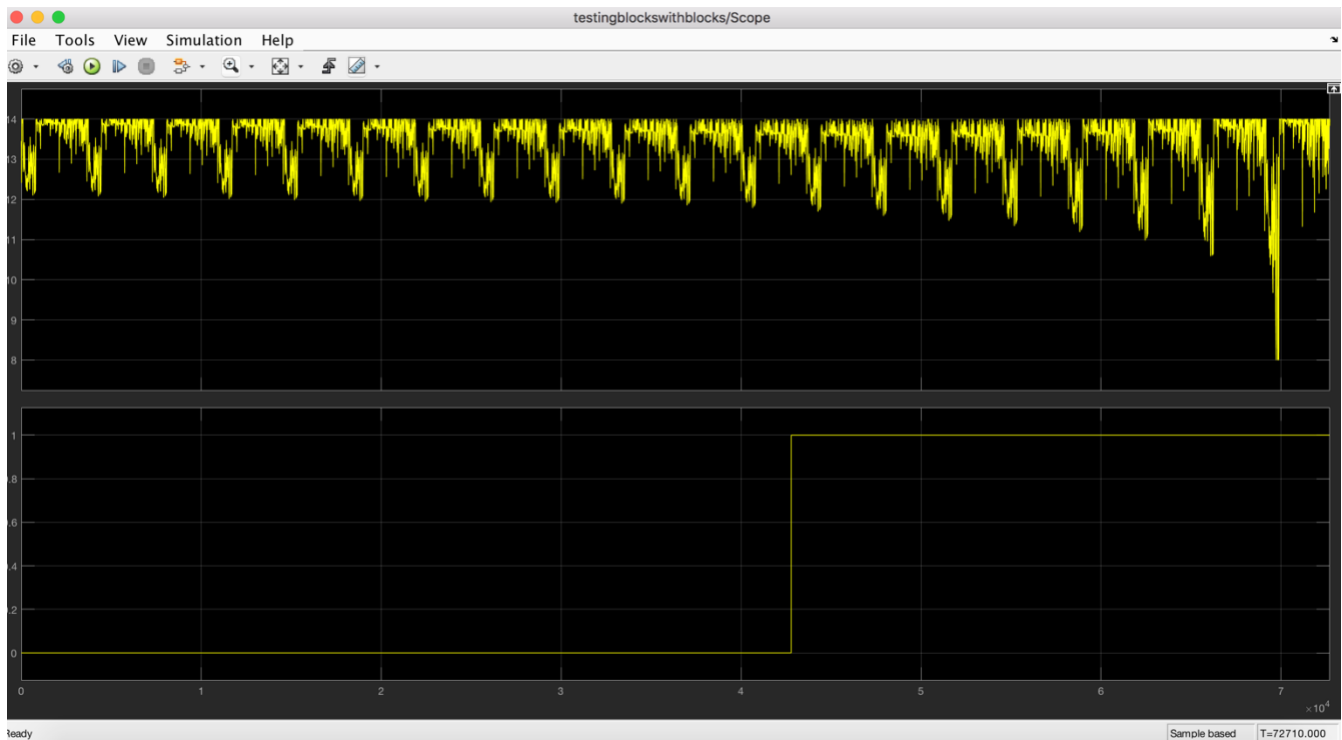


Figure 37 – Real Data Test 80% SOHi

## Observations

- 1- The algorithm successfully detected the SOH decrease
- 2- There is a time delay between the actual SOH decrease below the 40% marker and the algorithm detection timing
- 3- This time was calculated to be Approximately 7000 Seconds (2 hours).

## Conclusion

- 1- The SOH detection algorithm is working as designed and gives the actual results
- 2- The time delay of the 7000 seconds may seem to be long, but in fact, this time delay is relatively short when compared to the actual time that the battery state of health takes to degrade below the 40% level. Batteries normally degrade in health in months and sometimes years, hence, this time delay will not affect the critical situation in which the battery may drastically decrease in health. Moreover, this time delay will be studied further with the aim of making it as short as possible. This will be easily done when BAT-MAN project is launched into the market and we start gathering more batteries information through our cloud.

- Test 2
- Initial SOH = 50% using real data

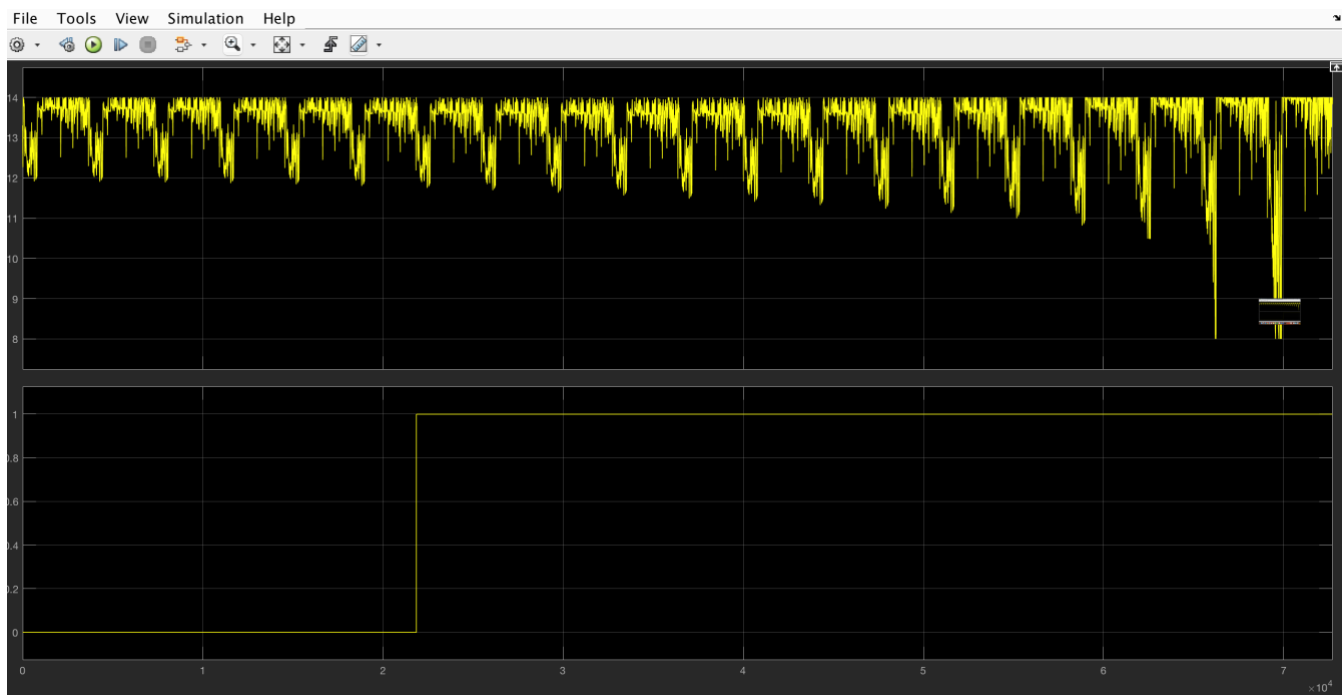
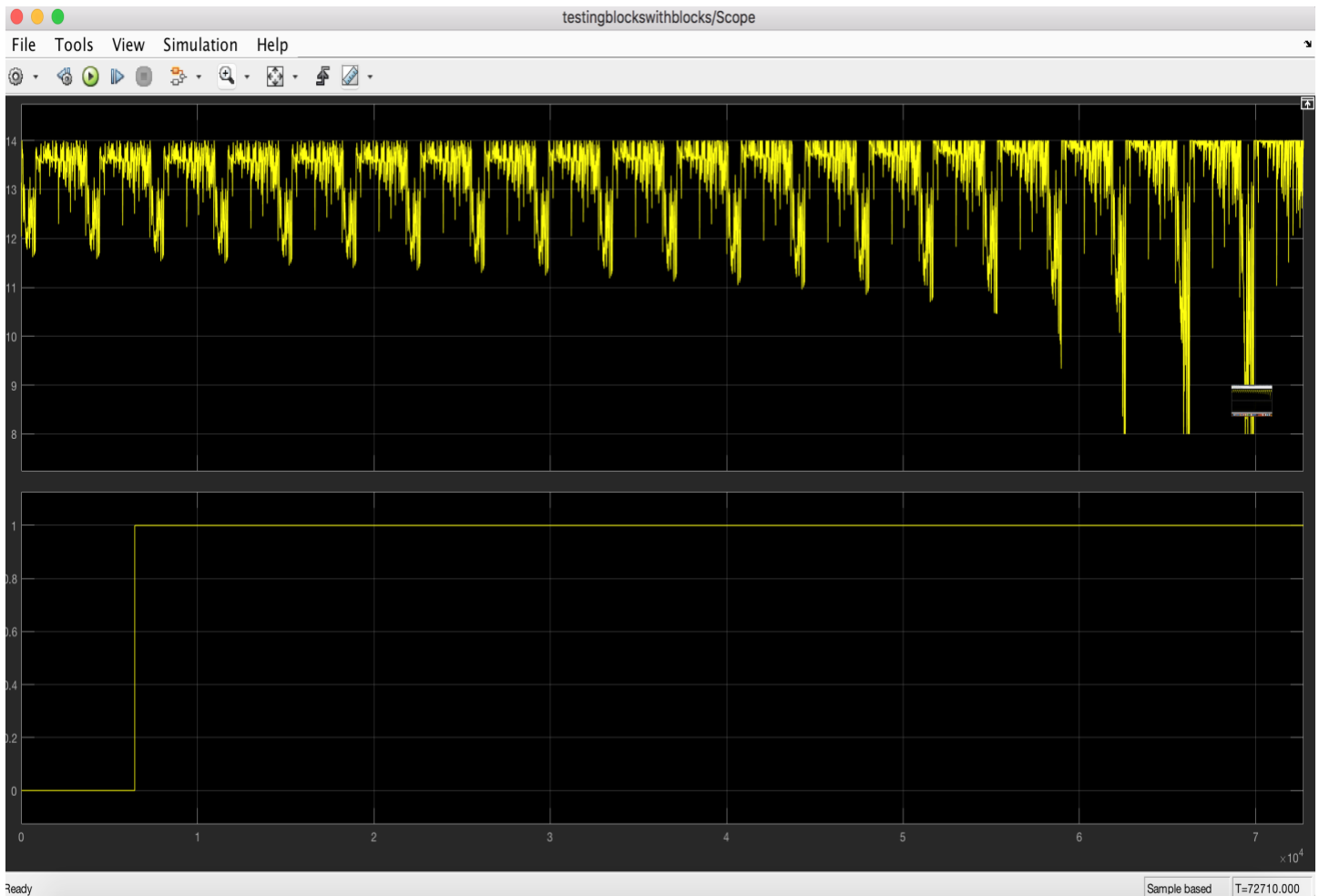


Figure 38 - Real Data Test 50% SOHi

- Same results were obtained from the prior test, but this test shows that the output yields a binary 1 at an earlier stage. This is because of the initial state of health which is 50%, so the battery is was on the verge of degrading below 40%.
  - The same time delay of 7000 seconds is still present.
- 
- Test 3
  - Initial SOH = 30% using real data



*Figure 39 - Real Data Test 30% SOHi*

- The same results are still present, but here, the battery state of health was already below 40%, which means that this time taken for the algorithm to yield a binary 1 is the total time delay for the algorithm detection process. As in fact, the model output without a time delay should yield 1 instantly.

## 7.4. Testing using Simulated input data

The following tests were applied using simulated input data (input currents). More specifically, the input current profile was changing slowly in order to test the model when working with different applications and not only automotive batteries.

- Test 4
- Initial SOH = 50% using simulated data

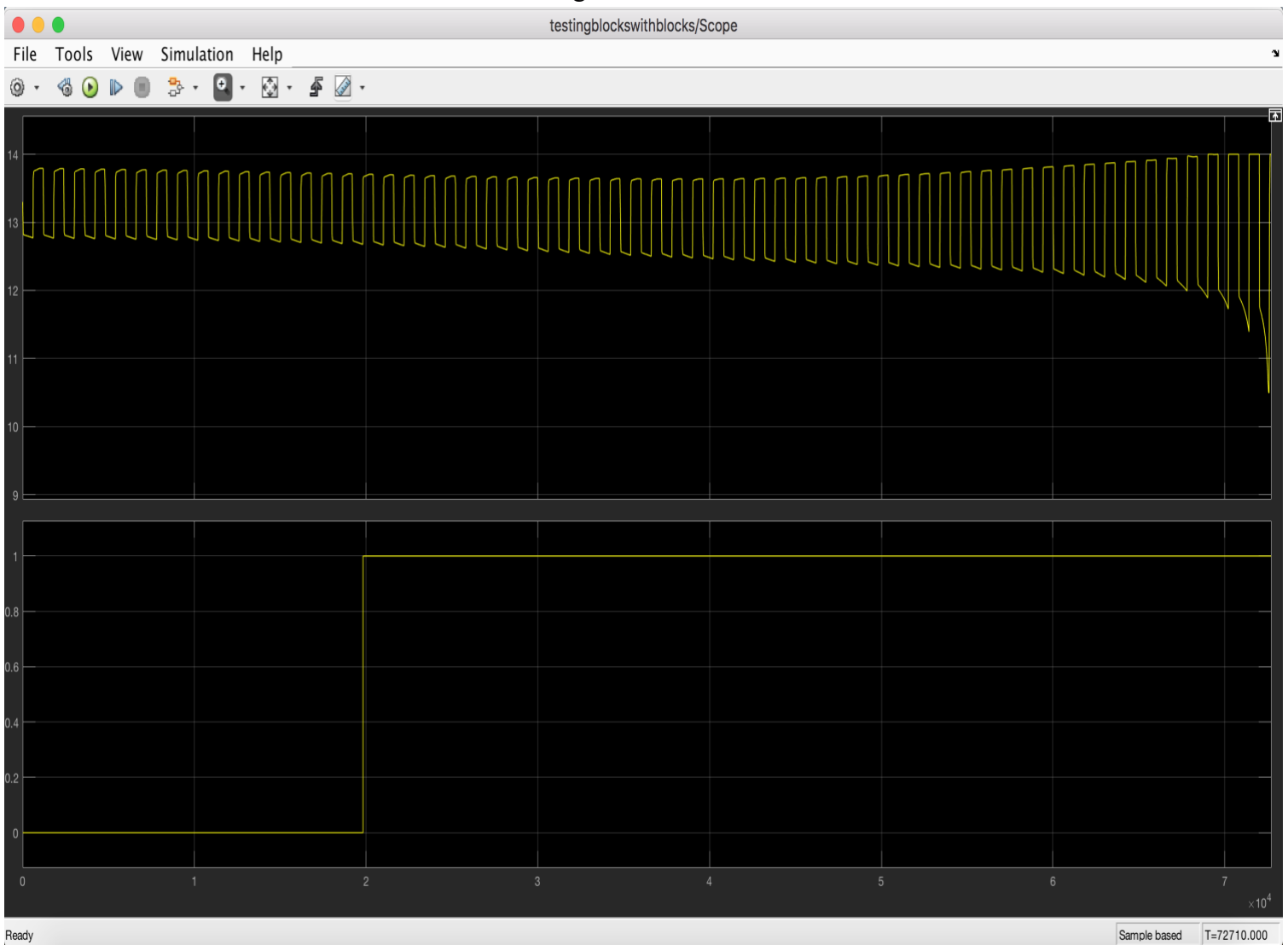
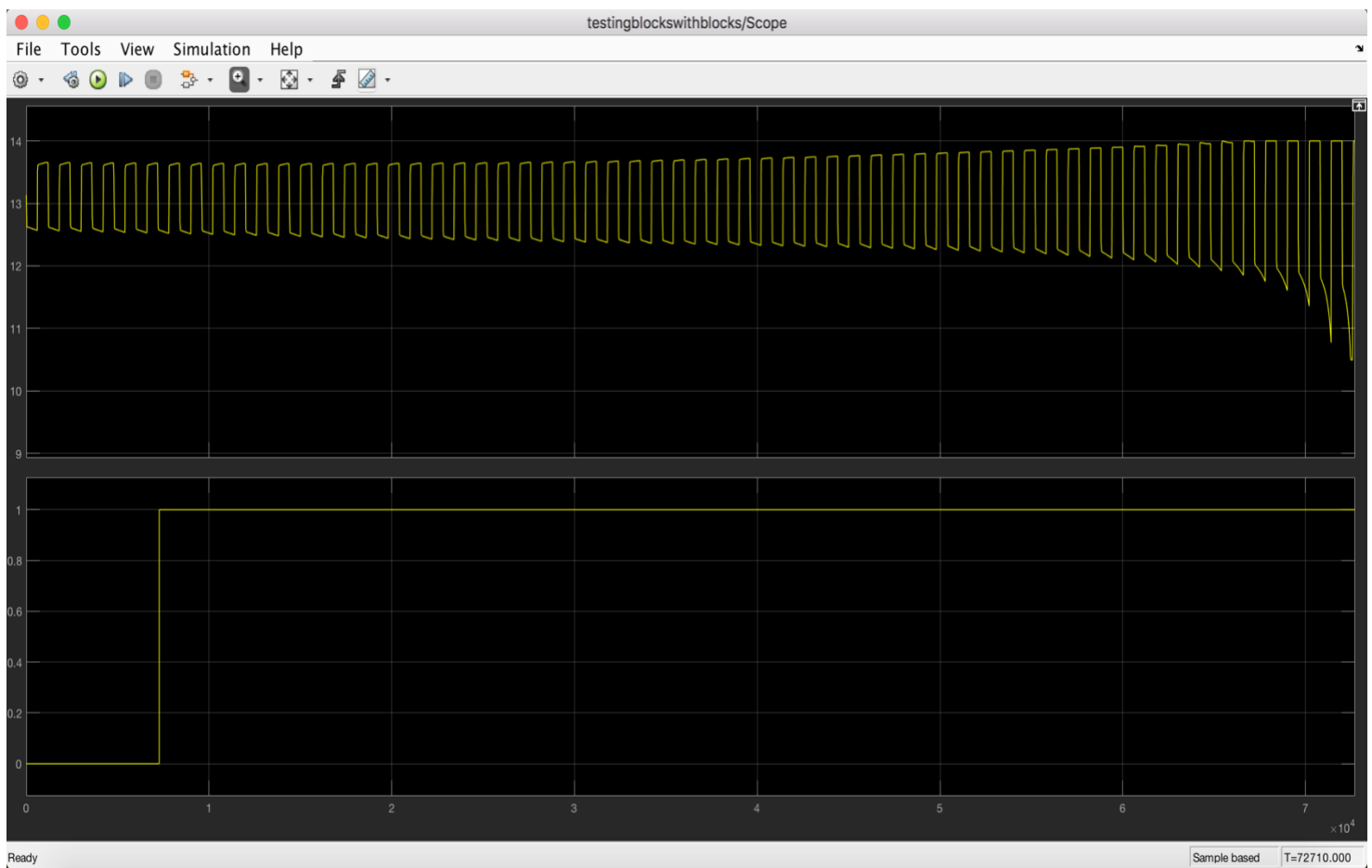


Figure 40 - Simulated Data Test 50% SOHi

- Simulated data give the same exact results of the real data, which proves that the model is solid and accurate and can work with various input current data set. Hence, it could be applied to any device because it is independent of the current withdrawn from the battery.
- Same time delay still applies with the exact same amount of time
  - Test 5
  - Initial SOH = 30% using real data



*Figure 41 - Simulated Data Test 30% SOH*

The final test gives results similar of that of test 3, with the same amount of time delay.

### Test conclusion

Our BAT-MAN model provides accurate estimation to the battery State of Health change when degrading below 40% with certain conditions. This information as mentioned is a key factor to sustain the battery healthy and to detect any faults or error

### Fault detections

A fault is present if one of the following cases apply

- a. The output yields a sudden binary 1, which can happen due to any discontinuities.
- b. The output yields a binary 0 after yielding a binary 1. This is practically impossible because physically it means that the battery health regained again after degrading.
- c. If any binary 1 pulse is present, which is a sign that the algorithm detection failed to classify the data in the correct partition for a short period of time.

## 8. Conclusion

In conclusion, the whole idea of developing a new device that has a new functionality that was never implemented by any other similar device was an extraordinary challenge. At the start of this project it seemed almost impossible to develop an algorithm for detection using only the Voltage information of a running battery by a simple procedure of connecting a device to the terminals of a running battery for a certain amount of time. But after trying different approaches and algorithms, and by deep testing I was able to create the algorithm that detects the decrease of the State of Health of the battery when it falls below 40% which is the border between a healthy and an unhealthy battery.

As a matter of fact, all the above work has successfully achieved the following:

- A fully functioning battery model that can be used not only for State of health detection, but also can be used for various applications such as State of charge estimation, enhanced battery performance, and understanding more the battery behavior.
- A model that is based on a solid algorithm with a 97% accuracy that is able to give indication on whether the battery State of Charge is above or below the boarder of 40% and gives notice to the user to take the required actions.
- A complete Real time model that will be used for code generation in order to be uploaded for the microcontroller and the above layers.

Development Phase:

The aim is to continuously improve BATMAN in terms of accuracy, precision and speed. This is aimed to be done after producing BATMAN to the market and starting to collect data from users that will help us understand the correct way to improve our product.

# APPENDIX

## 1. MATLAB scripts

### 1.1 MATLAB script for battery model verification and Algorithm key parameters calculations:

```
close all
clear all
clc

%% Slope of Output Voltage

Params                                % Run the parameters m.file

load real_data5.txt                  % Load real data from text file

z= real_data5;                        %Defining variable for stop time
figure(2)
title('Voltage of real data')
plot(z(:,3))                          %Plotting real Voltage
figure(7)
title('Voltage of real data')
plot(z(:,3))
Vr=z(:,3);
hold on

%% Defining Initial values for Simulink Model
i=0;
Rsoc=0.75;
T=25;

% for Rsoc= [1 0.8 0.5 0.3 0.1];
for Soh=[100 90 80 70 60 50 40 30 20]; %Loop Objective: Plot Voltage, current, Peak values boxplots and SOC for different SOH

SoC=Soh*Rsoc;
i=i+1;
SOH(Soh)=Soh;
sim BatteryModel;                    %Running Simulink Model

V_nc = Vo(6000:end);                  % Voltage neglecting Cranking (V_nc)

[pks_MAX,locs_MAX] = findpeaks(V_nc);
[pks_MIN1,locs_MIN] = findpeaks(-V_nc);
```



```

pks_MIN=-pks_MIN1 ;
%
LMaxima_full_idx = locs_MAX;           %Finding Local Max index
% LMinima_full_idx = locs_MIN;         %Finding Local Min index
%% Peak difference

Peak_difference(Soh,:) = (pks_MAX(1:2314) - pks_MIN(1:2314));

%% calculating variance

variance(Soh)= var(pks_MAX)'; % Variance for real data

%% BOX PLOTTING

if(i==1)                               %Grouping for BoxPlot
    Maxima = pks_MAX;
    g_max=repmat(({Soh}),length(pks_MAX(:,1)),1);
    Minima = pks_MIN;
    g_min=repmat(({Soh}),length(pks_MIN(:,1)),1);
else
    Maxima = [Maxima; pks_MAX];
    Minima = [Minima; pks_MIN];
    g_max=[g_max;repmat(({Soh}),length(pks_MAX(:,1)),1)];
    g_min=[g_min;repmat(({Soh}),length(pks_MIN(:,1)),1)];
end

figure(1)
subplot(2,1,1)
boxplot(Maxima,g_max)                  %Box Plotting Maximum Peaks
grid on
xlabel('Soh 80%')
ylabel('Volts')

subplot(2,1,2)
boxplot(Minima,g_min)
grid on

figure (5)
subplot(3,1,1)                        %Simulated output Voltage Plot
hold on
plot(real_data1(:,3))
title('Voltage of real data')
hold on
plot(Vo),grid on
hold on
title('Voltage of Simulated data')
legend('Soh=100%','Soh=80%','Soh=60%','Soh=40%','Soh=20%','Soh=5%','Soh=1%','Soh=0.5%')
subplot(3,1,3),grid on

```

```

hold on
plot(SOC),grid on                                %State of Charge (SOC) Plot
title('SOC')
legend('SoC=100%','SoC=80%','SoC=60%','SoC=40%','SoC=20%','Soh=5%','Soh=1%','Soh=0.5%')
hold on

figure (7)
                                %Simulated output Voltage Plot
plot(Vo),grid on
hold on
% legend('Voltage of real data','SoH=100%','SoH=80%','SoH=60%','SoH=40%','SoH=20%','SoH=5%')
legend('Voltage of real data','SoC=100%','SoC=90%','SoC=80%','SoC=70%','SoC=60%','SoC=50%','SoC=40%','SoC=30%','SoC=20%','SoC=10%',
'Soh=5%')

%% Slope Plot

y=diff(Vout.data);                                %Slope Plot
figure(4)
title('Slope')
plot(y)
hold on
legend('Soh=100%','Soh=80%','Soh=60%','Soh=40%','Soh=20%','Soh=5%','Soh=1%','Soh=0.5%')

end                                                % END OF MAIN LOOP

```

## 1.2 Variance Vs State of Health

```

%% Variance vs SOH plot

SOH=SOH(SOH~=0);
SOH=SOH';

if (Rsoc==1);

variance100=variance(variance~=0);
variance100=variance100';
figure (6)
plot(SOH,variance100)
legend('soc=soh','soc=80%soh','soc=50%soh','soc=30%soh','soc=10%soh')
hold on
grid on
clear SOH
end
if (Rsoc==0.8);

variance80=variance(variance~=0);

```

```

variance80=variance80';
figure (6)
plot(SOH,variance80)
legend('soc=soh','soc=80%soh','soc=50%soh','soc=30%soh','soc=10%soh')
hold on
grid on
clear SOH
end

```

```

if (Rsoc==0.5);

```

```

variance50=variance(variance~=0);
variance50=variance50';
figure (6)
plot(SOH,variance50)
legend('soc=soh','soc=80%soh','soc=50%soh','soc=30%soh','soc=10%soh')
hold on
grid on
clear SOH
end

```

```

if (Rsoc==0.3);

```

```

variance30=variance(variance~=0);
variance30=variance30';
figure (6)
plot(SOH,variance30)
legend('soc=soh','soc=80%soh','soc=50%soh','soc=30%soh','soc=10%soh')
hold on
grid on
clear SOH
end

```

```

if (Rsoc==0.1);
variance10=variance(variance~=0);
variance10=variance10';
figure (6)
plot(SOH,variance10)
legend('soc=soh','soc=80%soh','soc=50%soh','soc=30%soh','soc=10%soh')
grid on
end

```

```

end      %END OF Rsoc LOOP

```

### 1.3 Input current creation code

```
function u = InputWave()

global N

duty = 50;
period = 100;

sample_high = (period/100) * duty;
sample_low = period - sample_high;

high = 1;
samples = 0;
for i = 1:N
    samples = samples + 1;
    switch high
        case 1
            if(samples > sample_high)
                samples = 1;
                high=0;
            end
        case 0
            if(samples > sample_low)
                samples = 1;
                high=1;
            end
        end
    u(i) = high*(-10);
end
```

### 1.4 Support Vector Machine Function and Application

```
%% SVM TRAINER
function [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
% Input:
%   trainingData: a matrix with the same number of columns and data type
%   as imported into the app.
%
% Output:
%   trainedClassifier: a struct containing the trained classifier. The
%   struct contains various fields with information about the trained
%   classifier.
%   trainedClassifier.predictFcn: a function to make predictions on new
%   data.
%
%   validationAccuracy: a double containing the accuracy in percent. In
%   the app, the History list displays this overall accuracy score for
%   each model.
```

```

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
% Convert input to table
inputTable = array2table(trainingData, 'VariableNames', {'column_1', 'column_2', 'column_3', 'column_4'});

predictorNames = {'column_1', 'column_2', 'column_3'};
predictors = inputTable(:, predictorNames);
response = inputTable.column_4;
isCategoricalPredictor = [false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
classificationSVM = fitsvm(...
    predictors, ...
    response, ...
    'KernelFunction', 'gaussian', ...
    'PolynomialOrder', [], ...
    'KernelScale', 0.43, ...
    'BoxConstraint', 1, ...
    'Standardize', true, ...
    'ClassNames', [0; 1]);

% Create the result struct with predict function
predictorExtractionFcn = @(x) array2table(x, 'VariableNames', predictorNames);
svmPredictFcn = @(x) predict(classificationSVM, x);
trainedClassifier.predictFcn = @(x) svmPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.ClassificationSVM = classificationSVM;
trainedClassifier>About = 'This struct is a trained model exported from Classification Learner R2019a.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new predictor column matrix, X, use: \n yfit = c.predictFcn(X) \nreplacing "c" with the name of the variable that is this struct, e.g. "trainedModel". \n \nX must contain exactly 3 columns because this model was trained using 3 predictors. \nX must contain only predictor columns in exactly the same order and format as your training \ndata. Do not include the response column or any columns you did not import into the app. \n \nFor more information, see <a href="matlab:helpview(fullfile(docroot, "stats", "stats.map"), "appclassification_exportmodeltoworkspace")">How to predict using an exported model</a>');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
% Convert input to table
inputTable = array2table(trainingData, 'VariableNames', {'column_1', 'column_2', 'column_3', 'column_4'});

predictorNames = {'column_1', 'column_2', 'column_3'};
predictors = inputTable(:, predictorNames);
response = inputTable.column_4;
isCategoricalPredictor = [false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationSVM, 'KFold', 5);

% Compute validation predictions

```

```
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);
```

```
% Compute validation accuracy
```

```
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```

## 1.5 Non real time testing

```
close all
```

```
clear all
```

```
clc
```

```
tic
```

```
%% Slope of Output Voltage
```

```
load('real_data4.txt')
```

```
input=real_data4(:,[1 2]);
```

```
figure(2)
```

```
plot(real_data4(:,3));
```

```
grid on
```

```
Params % Run the parameters m.file
```

```
%% Defining Initial values for Simulink Model
```

```
i=0;
```

```
T=25;
```

```
Rsoc=1;
```

```
% sim BatteryModelOnly_Testing;
```

```
% for Soh= 100:-1:99; %Loop Objective:Plot Voltage, current, Peak values boxplots and SOC for different SOH
```

```
for Soh = 100:-1:1
```

```
% for Soh=100
```

```
SoC=Soh*Rsoc;
```

```
i=i+1;
```

```
sim BatteryModelOnly_Testing; %Running Simulink Model
```

```
% figure (8)
```

```
% plot(Vo)
```

```
% hold on
```

```
% grid on
```

```
V_nc = Vo(6000:end); %Voltage neglecting Cranking (V_nc)
```

```
[pks_MAX,locs_MAX] = findpeaks(V_nc);
```

```
[pks_MIN1,locs_MIN] = findpeaks(-V_nc);
```

```
pks_MIN=-pks_MIN1 ; %M
```

```
LMaxima_full_idx = locs_MAX; %Finding Local Max index
```

```
LMinima_full_idx = locs_MIN; %Finding Local Min index
```

```
%% Slope Calculation
```

```
slope=diff(V_nc);
```

```
% figure
```

```
% plot (slope)
```

```
%% DELTA (Slope peak to peak Values)
```

```
% SOH=SOH(SOH~=0);
```

```
delta2(Soh)= max(slope) - min(slope) ; %
```

```

delta1=delta2';
delta=delta1(delta1~=0);

difdelta1= -diff(delta);
% difdelta2= [ 0; delta];
% delta_data1(end)=[];
variance1(Soh)= var(pks_MAX); % Variance for real data
var1= variance1(variance1~=0);
% sample= [var1 delta_data1 difdelta2];

end % END OF MAIN LOOP
difdelta2= [ difdelta1(1); difdelta1];
sample= [var1 delta difdelta2];
%
load ('Test_final.mat');
trainingData = Test_final;

% % Prediction
QuadraticSVM(trainingData)
tic
yfit = ans.predictFcn(sample)
toc
open yfit
load laughter
sound(y,Fs)

```

## 1.6 Test Results

```
close all
clear all
clc
%% Slope of Output Voltage

Params                                % Run the parameters m.file
load real_data1.txt                  % Load real data from text file
z= real_data1;
for Soh=[100,80,60,40,20]

SoC=100;
%factor to adapt the SoC
SoC=Soh*SoC/100;

sim BatteryModelOnly_F;              %Running Simulink Model
InputData=smooth;

figure(1)
title('SoC estimated')

plot(real_data1(:,1),real_data1(:,3))
hold on
plot(InputData.Vf_mod.Time,InputData.Vf_mod.Data)
grid on
xlabel('Time')
Ysabel('Volts')
legend('Voltage of real data', 'Voltage of Model')
end
```



## 1.7 Initialization for Real Time Testing

```
close all
clear all
clc
load ('real_data4.txt')
input=real_data4(:,[1 2]);
% input = input(9000:end,:);
f= repmat(input,20,1);
% x=repelem(input,1,1);

% i=1;
% p(length(input),1)=[3635.4];
% for i=1:1:36355

%   p(i)=[(p(i) + 0.1)];

% end 27356
% f=f;
xStart = input(end,1);
dx = 0.1;
N = length(input);
x = (xStart+0.1) + (0:(20*(N-1)))*dx;
x=x(1:690745);
x=x';
f(N+1:end,1)=x;
input_current=f;
```

## 2. SIMULINK Models

### 2.1 Battery Model

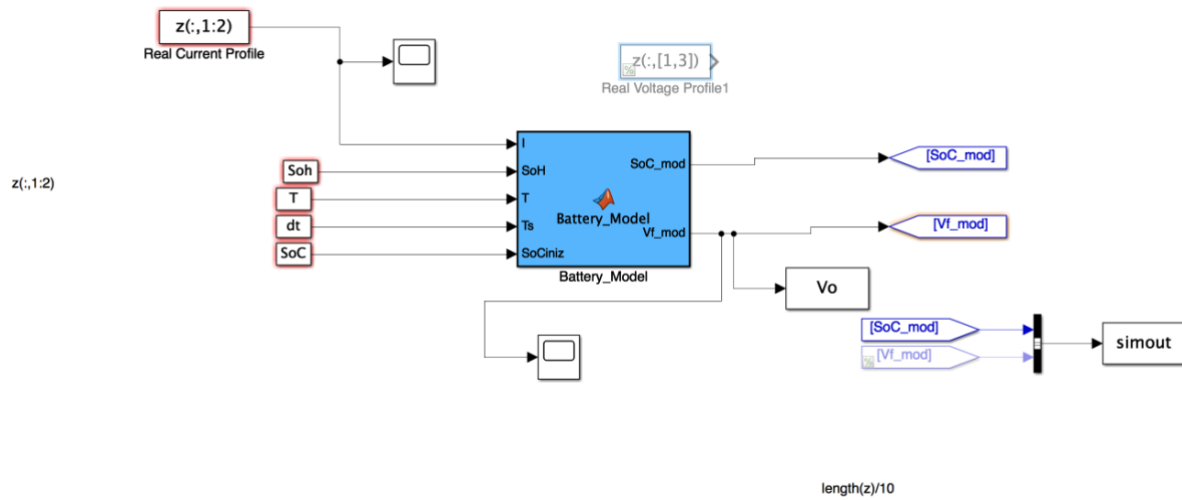


Figure 42 - Battery SIMULINK model

### 2.2 Generated current Profile

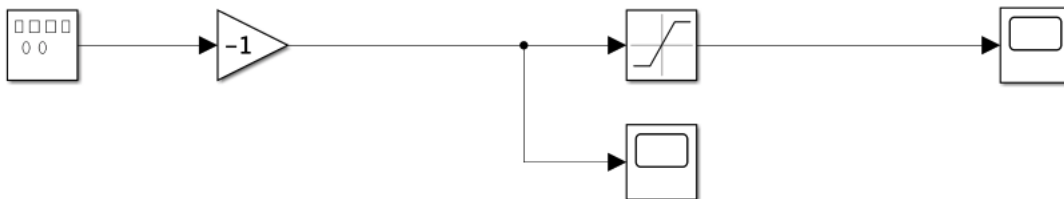
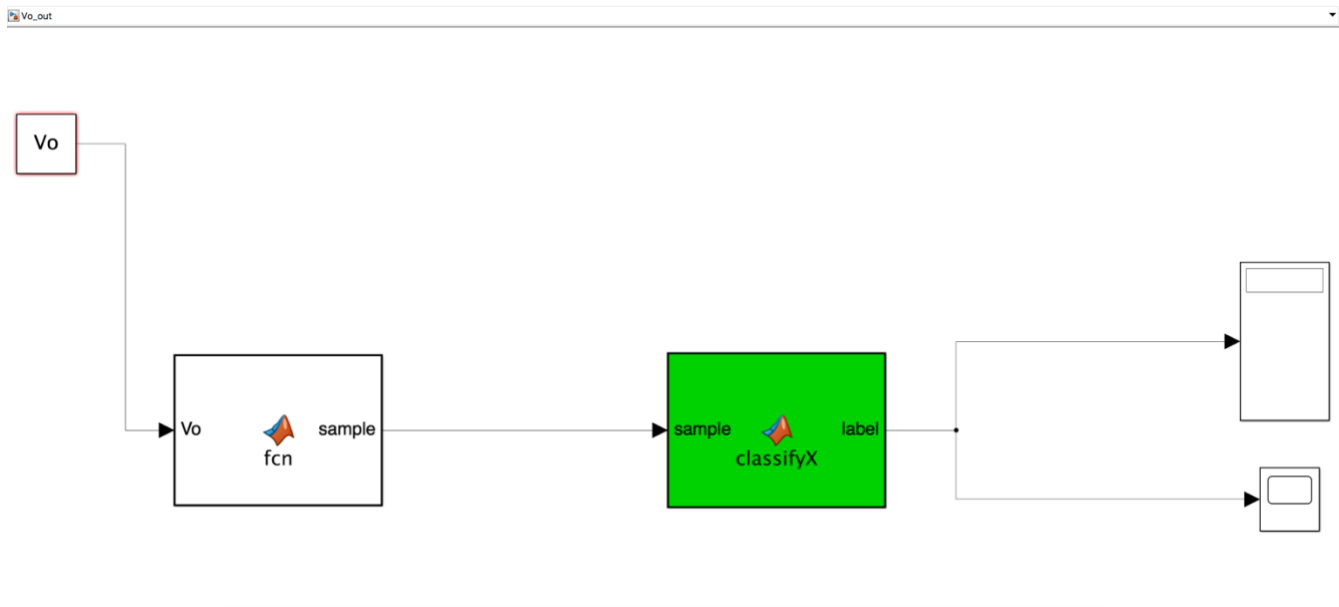


Figure 43 - Input current SIMULINK model

## 2.3 Support Vector Machine Model



*Figure 44 - Support Vector Machine SIMULINK Model*

## 2.4 Non-real time testing Model

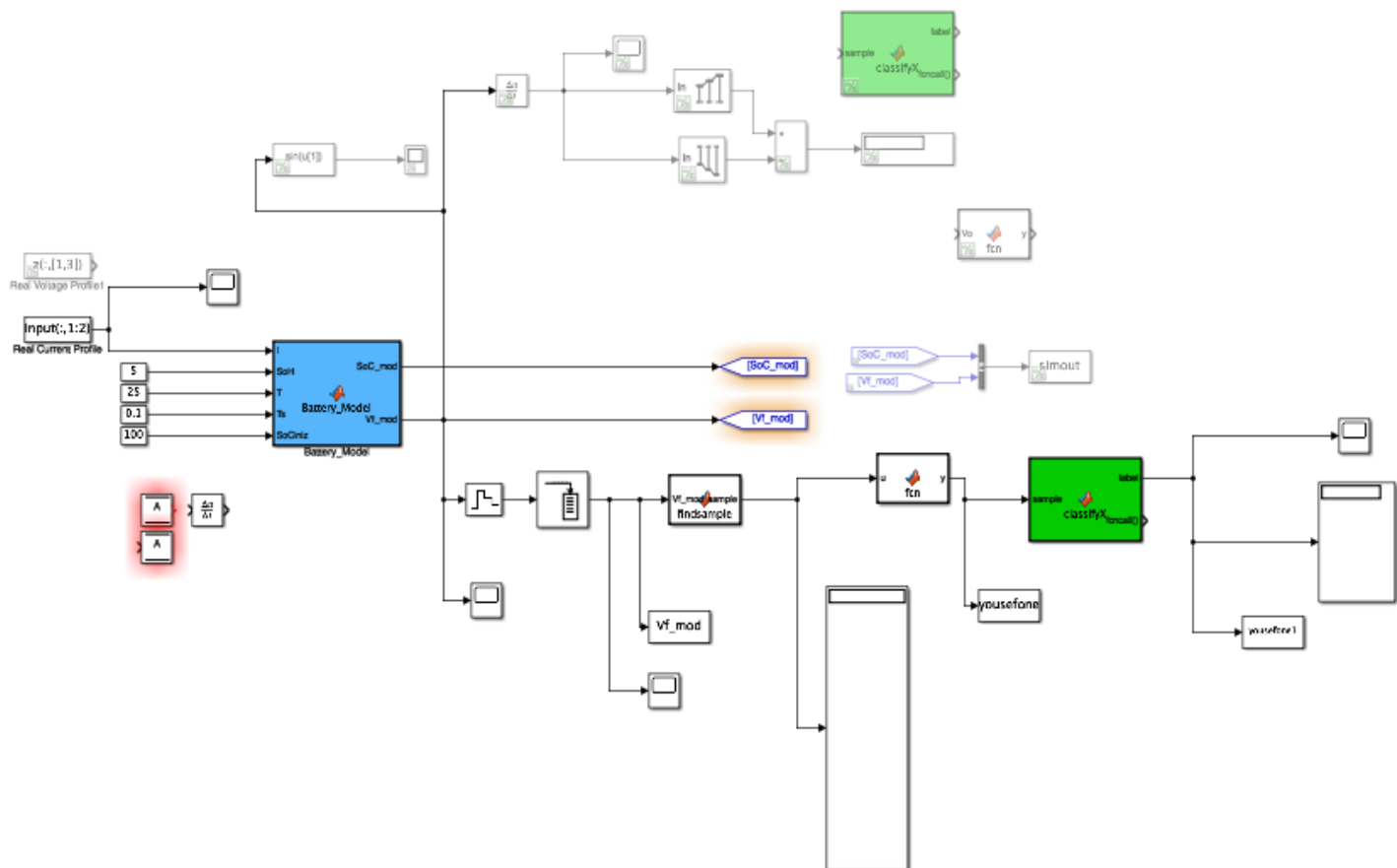


Figure 45 - Non real time testing SIMULINK model

## 2.5 Real Time testing Model

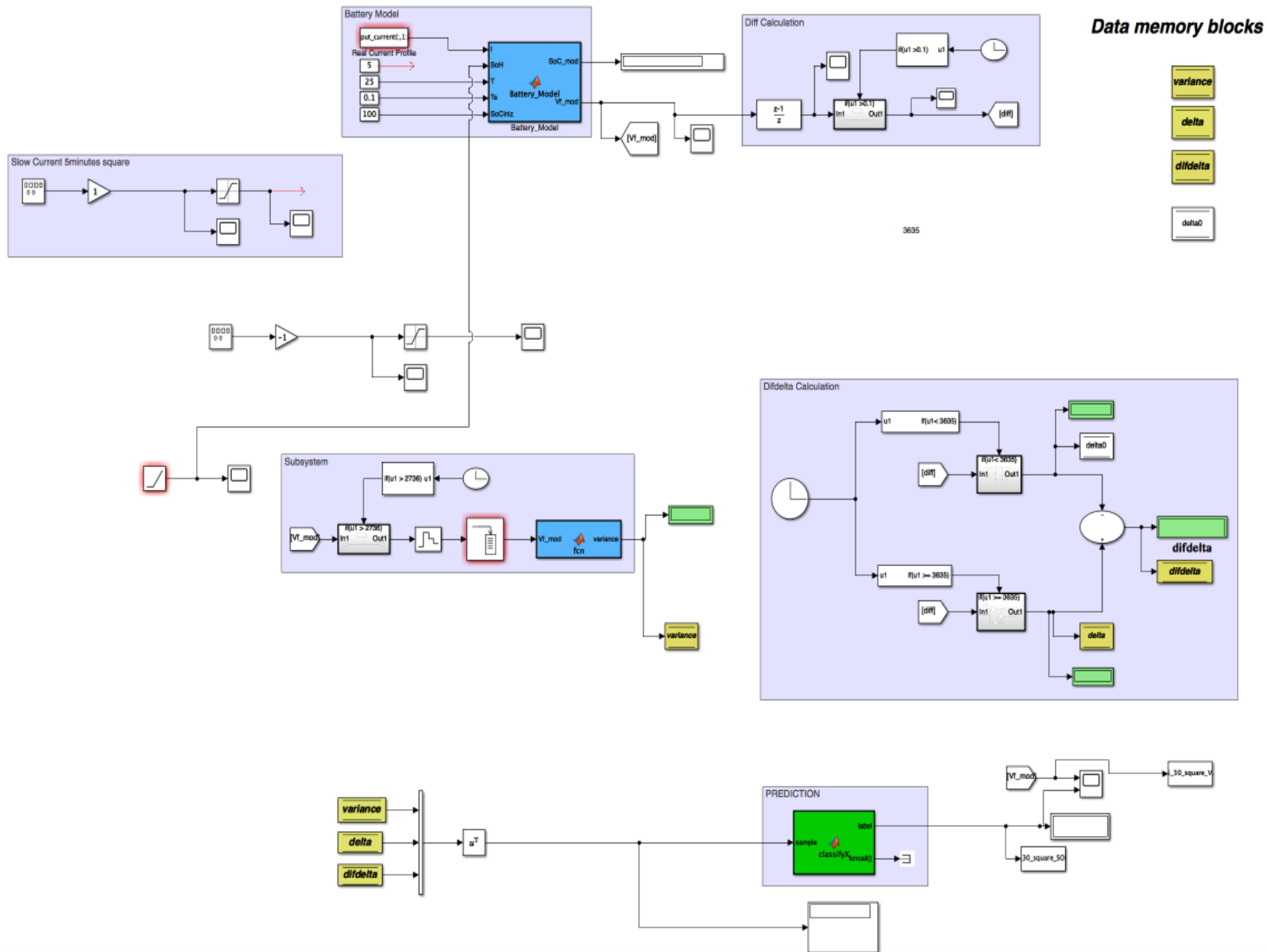


Figure 46 - Real time testing SIMULINK model



# Bibliography

1. Closer look to Battery State of Health  
<https://www.analog.com/media/en/technical-documentation/technical-articles/A-Closer-Look-at-State-Of-Charge-and-State-Health-Estimation-Techniques-....pdf>
2. Battery State of Charge Determination <https://www.mpoweruk.com/soc.htm>
3. Brain Technologies, Preliminary analysis of BATMAN project
4. DEVICE PATENT (United States of America Brevetto n. 8437908 B2, 2013)
5. Generalized Lead-Acid based Battery Model used for a Battery Management System  
URL : <https://www.athensjournals.gr/technology/2016-3-3-4-Dost.pdf>
6. MATHWORK Support Vector Machine Theory, URL  
  
<https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html>
7. MODEL BASED SOFTWARE DESIGN course notes, Politecnico di Torino  
01OUZQW, 01OUZOV

