



Politecnico di Torino

**Online state of health estimation of lithium-ion  
batteries based on Long Short Term Memory  
network for hybrid and electric vehicle**

Master degree

October 2022

Supervisor:  
Prof. Bonfitto Angelo  
Ing. Luciani Sara

Candidate:  
Yi Xiaolong

## Abstract

State of health (SOH) is a key parameter of lithium-ion battery. As the battery ages an accurate SOH estimation plays a important role in battery management system (BMS).

This thesis refers to a dataset from a laboratory test campaign conducted at LIM to train a LSTM network for estimating the SOH variation. The dataset is composed by 55 charging-discharging cycles and each of them has 43 different usage profiles. In order to age the battery, the entire test cycle lasted for several months. SOH decayed from initial 100% to 82%. In order to test the robustness of the network it uses three different profiles: charging profile, pulse profile and polarized random walk profile.

To catch the SOH degradation information, several features are computed from original voltage and current profile. They are state of charge (SOC), variation of state of charge (dSOC), state of energy (SOE), variation of state of energy (dSOE) and variation of voltage (dV).

Move sliding window approach is used for feature extraction. This approach can smooth the data and reduce the total amount of data while preserving the data information, thereby increasing the training speed.

LSTM is selected neural network for this thesis. Three different profiles were trained by LSTM and all achieved high accuracy.

The last section is focused on the implementation in Simulink. In order to better match the real working conditions, PRW profile is selected for Simulink modeling. Finally, the developed SOH estimation algorithm is combined with SOC estimator. The SOH and SOC estimator can interact with each other and use each other's data as input values. The model gets accurate results on PRW profile and charging profiles. The overall accuracy is almost equal to 1.99% for SOH estimation.

# Table of Contents

Chapter 1. Introduction .....	5
1.1 Growing demand for hybrid and electric vehicles.....	5
1.2 Aims and Objectives .....	6
1.3 Thesis outline .....	7
Chapter 2. Literature review .....	8
2.1 SOH introduction .....	8
2.2 Lithium-ion battery working principle and structure .....	8
2.2.1 Main advantages .....	9
2.2.2 Main shortcomings.....	10
2.2.3 Electrochemistry .....	11
2.3 Physical factors affecting on cell aging.....	11
2.4 SOH estimation methods overview .....	13
2.4.1 Incremental Capacity Analysis (ICA).....	14
2.4.2 Differential thermal voltammetry (DTV) analysis.....	16
2.4.3 Differential mechanical parameter (DMP) analysis .....	16
2.5 Introduction of neural networks.....	18
2.5.1 Basic components and structure of FNNs.....	18
2.5.2 Recurrent neural network.....	22
2.5.3 Long Short Term memory (LSTM) .....	26
2.5.4 A walk through step-by-step in LSTM .....	28
Chapter 3. SOH estimation methodology.....	32
3.1 Battery introduction .....	33
3.2 Test cycle introduction and training dataset selection.....	35
3.2.1 Charging profile .....	35
3.2.2 Discharging profile .....	36
3.2.3 Pulse profile .....	37
3.2.4 Polarized Random Walk profile.....	38
3.2.5 Random Walk profile.....	39
3.3 Feature introduction.....	41

3.3.1 Incremental calculation-based features.....	42
3.3.2 Time-based and envelope-based features .....	43
3.3.3 Snapshot-based features.....	43
3.4 Feature standardization .....	49
3.4.1 Introduction of zscore .....	50
3.5 SOH ground truth value estimation .....	54
Chapter 4. Network structure and training results.....	57
4.1 LSTM network structure and training parameters.....	57
4.2 Training Results .....	59
Chapter 5. Simulink modeling implementation .....	62
5.1 Simulink modeling overview.....	62
5.1.1 SOH estimation model.....	62
5.1.2 Combined model with SOC and SOH estimators.....	63
5.2 Simulink model of SOH estimator.....	63
5.3 SOC and SOH combined model. ....	66
5.4 Model test scenario and results.....	69
5.4.1 Simulation scenario description.....	69
5.4.2 Scenario test results.....	72
5.5 Possible improvements for SOH estimator.....	75
5.5.1 Butterworth filter introduction.....	75
5.5.2 data smoothing.....	77
5.5.3 Training result.....	80
5.5.4 Simulink model.....	82
5.5.5 Limitations of this approach .....	84
Chapter 6. Conclusion.....	85
Bibliography.....	87

# Chapter 1. Introduction

## 1.1 Growing demand for hybrid and electric vehicles

With the continuous maturity of hybrid and pure electric vehicle technology and the increasingly expensive international oil prices. Customers' demand for new energy vehicles is increasing day by day. As figure 1 shows, sales of hybrid and pure electric vehicles increased sixfold from 2015 to 2020.

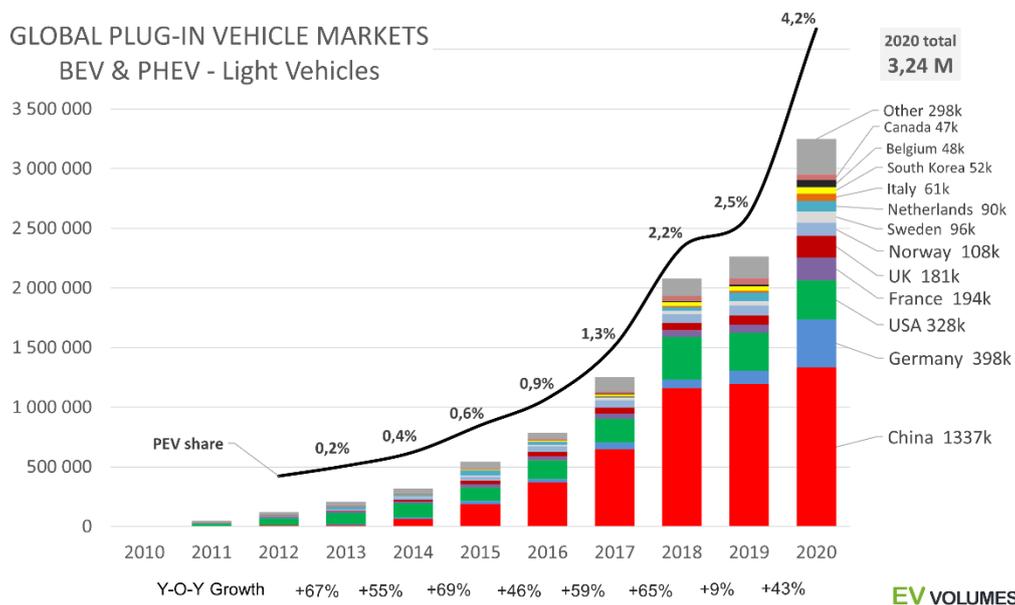


Fig.1 Global BEV & PHEV demanding

It is very important to be able to accurately estimate the SOH which is one of the most important parameters of the battery. First, it is directly related to the maximum available capacity of the battery. The calculation of many battery parameters depends on the SOH value, such as SOC. Secondly, due to huge battery usage, battery maintenance, scrapping problem must be taken seriously. The SOH states whether the battery has reached its expected life, which has great environmental significance. Third, with the continuous aging of the battery, the internal resistance continues to increase, and the internal structure is degraded and deformed, which will cause dangerous situations such as spontaneous combustion and explosion and pose a great threat to social security.

This places new demands on battery technology. Accurate estimation of the battery's state of charge and state of health has become an important research topic.

## 1.2 Aims and Objectives

The goal of this thesis is to design a real-time estimator of SOH based on artificial intelligence network. To train the network, a relevant battery charge-discharge test cycle should be designed first to age the battery, and the features related to the SOH changing should be extracted from the current and voltage data obtained in the cycle. After filtering, smoothing and other post-processing operations, the features will be used for network training. In order to verify the robustness of the method, three different kinds of cycles are tested, and all of them get good results.

Finally, in order to achieve the goal of real-time estimation, this work successfully established a Simulink model, and combined with the SOC estimator to form a combined network to estimate the two values of SOC and SOH at the same time.

The project process can be divided into the following steps:

1. Compare the different SOH calculation methods that exist and clarify the reasons for choosing an AI network.
2. Design the corresponding battery charge and discharge test cycle to obtain data such as current, voltage and temperature.
3. Perform data processing on the obtained data to obtain features for training the network.
4. Perform noise reduction, smoothing and other post processing on the features to make them match the requirements of network training.
5. Train the network and test it.
6. Build a Simulink network and combine the SOC estimator to reach the real-time SOH and SOC estimation.

## 1.3 Thesis outline

This thesis is structured as follows. The chapter 1 gives an overview of the motivation and states the objective of the thesis. The chapter 2 explains the concept of SOH and the physical and chemical factors that cause SOH decay. Then compares the existing traditional SOH calculation methods and explains the reasons for choosing artificial intelligence networks. Chapter 3 will describe in detail the design of the test loop, the extraction of training features, noise reduction, and smoothing. Chapter 4 will describe the artificial intelligence network structure and the corresponding training parameters and training results. The chapter 5 will build the Simulink model and describe the working principle of the model in detail then gives a possible improvement approach. Chapter 6 will summarize and discuss the results of the work and give suggestions for optimization. Chapter 7 is a link to all references used in this thesis.

## Chapter 2. Literature review

### 2.1 SOH introduction

Li-ion batteries are electro energy storage system. The performance of Li-ion batteries can be deteriorated by the degradation of their electrochemical medium which can cause power fade. This is called battery ageing.

State of Health (SOH) is the key parameter to measure the ageing phenomena. SOH describes the difference between a battery being used and a fresh battery considering the cell ageing. It is defined as the ratio of the maximum battery charge to its rated capacity. It is expressed as a percentage as below:

$$SOH\% = \left( \frac{Q_{max}}{C_r} \right) 100\% \quad (1)$$

Where  $Q_{max}/mAh$  is the maximum charge available of the battery.

$C_r$  is the rated capacity.

There are several factors such as battery chemistry and manufacturing affecting the ageing. The point that the battery fails to produce enough energy and power for its specific application, is commonly defined as the End of Life (EOL). Typically, batteries are considered at EOL when their SOH drop below 80% of the initial value. Here we can introduce ageing mechanisms and the main factors affecting on it.

### 2.2 Lithium-ion battery working principle and structure

Lithium-ion battery is a rechargeable battery that mainly relies on the movement of lithium ions between the positive and negative electrodes to work. Lithium-ion batteries use an intercalated lithium compound as an electrode material. At present, the main common cathode materials used for lithium-ion batteries are lithium cobalt oxide ( $LiCoO_2$ ), lithium manganate ( $LiMn_2O_4$ ), lithium nickelate ( $LiNiO_2$ ) and lithium iron phosphate ( $LiFePO_4$ ).

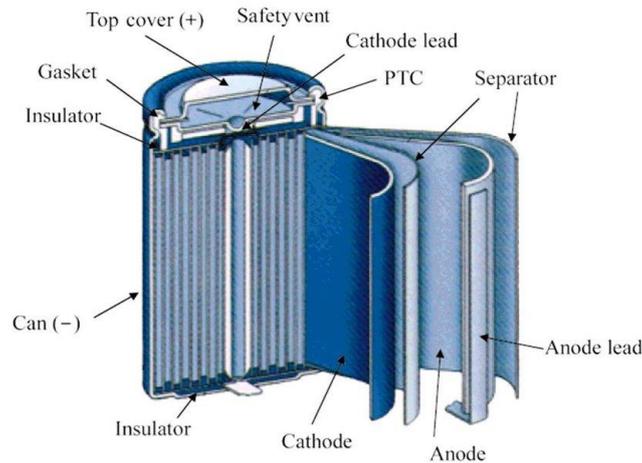


Fig.2 Lithium-ion cell internal structure

They are one of the most common types of rechargeable batteries in portable electronic devices, featuring high energy density, no memory effect, and only slow charge loss when not in use. In addition to consumer electronics, increasingly advanced lithium-ion batteries are also gaining popularity and can be used in the military, pure electric vehicles, and aerospace. For example, lithium iron phosphate batteries are becoming a common replacement for lead-acid batteries, which have historically been used in golf carts and utility vehicles, but this new and efficient battery has been able to break through the various shortcomings of lead-acid batteries, to achieve the goal of comprehensive replacement.

### 2.2.1 Main advantages

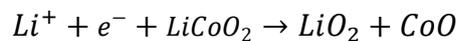
1. High energy density: It varies according to different electrode materials. Calculated by mass, it can reach 150-200Wh/kg (540-720kJ/kg); calculated by volume, it can reach 250-530Wh/L (0.9-1.9kJ/cm<sup>3</sup>).
2. High open circuit voltage: It varies with different electrode materials, up to 3.3 ~ 4.2V.
3. High output power: it varies with different electrode materials, up to 300 ~ 1500W/kg
4. No memory effect: The lithium iron phosphate lithium-ion battery has no memory

effect, the battery can be charged and discharged at any time without being discharged, and it is easy to use and maintain.

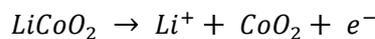
5. Low self-discharge: <5% ~ 10%/month. Due to the monitoring circuit of the intelligent lithium-ion battery, the operating current of this monitoring circuit is even higher than the self-discharge current.
6. Wide operating temperature range: it can work normally between -20°C ~ 60°C.
7. Fast charging and discharging speed

### 2.2.2 Main shortcomings

1. Over discharge: During over discharge, the excessively embedded lithium ions will be fixed in the lattice and cannot be released any more, resulting in accelerated shortening of life. Deep discharge (discharge when the voltage is less than 3.0V) is more likely to damage the battery. Therefore, it should be avoided to store when SOC is low. This process can be described by the following equation:



2. Overcharge: During overcharge, the electrode deintercalated too much lithium ions which can lead to damage for a long time, thereby irreversibly destroying the battery performance. Cars are often designed with a charging limit of about 70% based on this feature. Some products even recommend keeping it below 50% daily. The equation is:

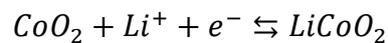


3. Aging: Unlike other rechargeable batteries, lithium-ion batteries will inevitably degrade during the use cycle (Cycle aging). Even if they are stored and not used, their capacity will decrease (Calendar aging).

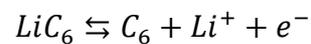
### 2.2.3 Electrochemistry

Like all chemical batteries, lithium-ion batteries are made up of three parts: a positive electrode, a negative electrode, and an electrolyte. Both process of lithium ions entering the positive and negative electrode material is called "intercalation", and the process of leaving is called "deintercalation". The following equations exemplify the chemistry.

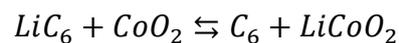
The positive electrode (cathode) half-reaction in the lithium-doped cobalt oxide substrate is:



The negative electrode (anode) half-reaction for the graphite is



The full reaction is:



### 2.3 Physical factors affecting on cell aging.

According to the recent research, there are three ageing modes [1]: the loss of lithium inventory (LLI); the loss of active material (LAM); the increasing of cell internal resistance.

LLI is mainly caused by the consumption of Li-ions by side reactions, such as on the surface of the negative electrode it can form some solid electrolyte interface (SEI) [2].

This will irreversibly consume Li-ions.

LAM is commonly caused by two factors. The first one is the structural changing of electrodes due to volume changes during life cycle. These include mechanical stress, particle cracking and reducing the density of lithium storage sites. The second factor is that the chemical decomposition and dissolution reactions [3]. The figure.3 shows the ageing factors and their effects.

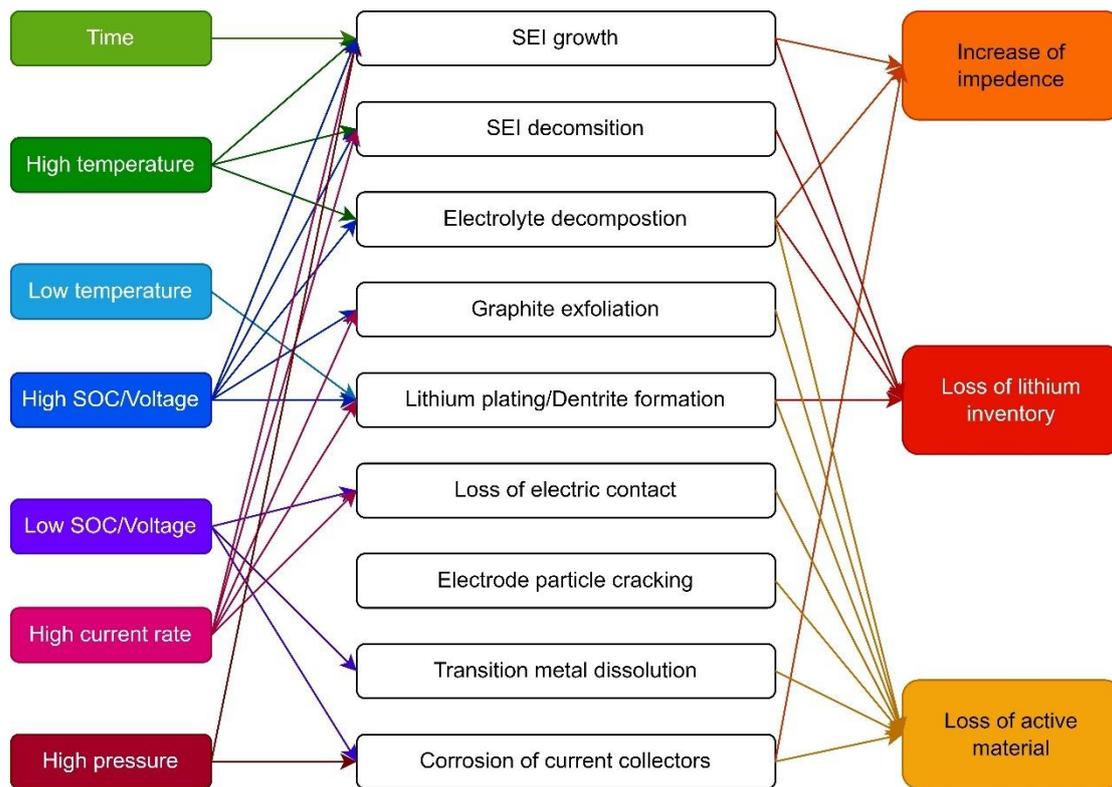


Fig.3 Ageing factors during usage

**High Temperatures:** It can accelerate side reactions, including SEI layer growth rates on the anode, this cause faster LLI and cell resistance increase [4].

**Low Temperatures:** Slow down the transport of Li ions in both electrodes and in the electrolyte. The electrolyte meets the graphite electrode, attempts of fast charging at low temperatures may thus create crowding of Li ions. This may form lithium dendrites and it will penetrate the separator thus cause a short circuit.

**Over-charge/discharge:** When a cell is overcharged, the cathode is over-delithiated and the anode is over-lithiated. The cathode material suffers from irreversible structural change when over-delithiated [5], followed by the dissolution of transition metal ions and active material decomposition [6]. Overcharging the cell can also generate significant heat, due to Joule effect [7].

**High currents:** High charge and discharge currents can cause localized overcharge and discharge to occur, leading to the same degradation reactions as generalized overcharge and over discharge. High current also generates more Joule heat which can cause ageing related to temperature problems.

**Mechanical stresses:** Cells are subjected to stress from different sources, such as manufacturing, electrode material expansion during operation, gas evolution in mechanically constrained cells and external loading during service. When stress exceeds the limit, the cell structure will be changed, and this can cause the failure.

## 2.4 SOH estimation methods overview

In this section we will discuss the main SOH analysis methods. In the first part we will discuss the traditional methods. Neural network approach will be discussed in the second part.

The model-based methodologies are highly dependent on mathematical physics models of the internal structural of the battery to estimate SOH. Various models such as the empirical models, electro circuit model can be applied for these approaches. Internal resistance and maximum capacity are often used as important parameters for it.

In this research [8], the author is using a physical-based model to predict SOH. Internal resistance-based model [9] is also be developed for SOH prediction. This research reported “Incremental capacity analysis (ICA) [10]” approach which is based on capacity and voltage. Other approaches such as dual adaptive H infinity filter [11] is also used for SOH estimation. The drawbacks of model-based approaches are very intuitive: they are highly sensitive to noise and parameter deviation. One the battery system becoming large and complicated, the creation and tuning of mathematical physics model would be a huge challenge.

To overcome the noise sensitivity, some filters like Kalman filter and particle filters are often used. Although it can solve the problem in some way, it increases the complexity and computation demand of the model.

Machine learning approaches can totally completely overcome this problem. It directly learns the features of the provided data. This approach completely skips the physical model creation and tuning thus it have a overwhelming advantage to deal with large

and complex system. According to different neural network, it could intrinsically have filter effect. Such like recurrent neural networks (RNNs) and long short-term memory (LSTM), they can memorize the data history thus automatically avoid the noise affection. Other sort of machine learning structural such as support vector machine (SVR) [12], multilayer perceptron algorithm [13], K-means clustering algorithm [14] are also used for SOH estimation.

Although machine learning approaches can mostly solve the conventional problem, they also have drawbacks. As we know these approaches are data-driven, it has a high requirement of training dataset. It requires the test profile can provide sufficient information so that to extract the features. The dataset should also cover whole using life of the battery so it can provide enough information among every SOH value. The acquisition of a such dataset takes a lot of time (several months even a year). The quality of the dataset is also important for model training. Less error is in dataset, better model quality it can get.

In the following sections, it will be discussed the ICA approach and LSTMs-based RNNs approach.

### 2.4.1 Incremental Capacity Analysis (ICA)

Incremental capacity analysis is a method that analyze the cell from electrochemical side. It can provide information about the internal cell state by voltage and current curves. The equation of ICA is as below:

$$ICA(U_{cell}) \left( \frac{Ah}{V} \right) = \frac{dQ(U_{cell})(Ah)}{dU_{cell}(V)} \quad (2)$$

Where Q is charged capacity and U<sub>cell</sub> is the cell voltage. According to the research of Dubarry et al [15]. each peak in the ICA curve has a unique shape, intensity and position, it represents a unique electrochemical process occurs in the cell. For this consideration, when the aging phenomena take into place, it will influence the shape if ICA curve thus

it can be analyzed. According to Elie Riviere et al. [16] the following ICA curve can represent this feature.

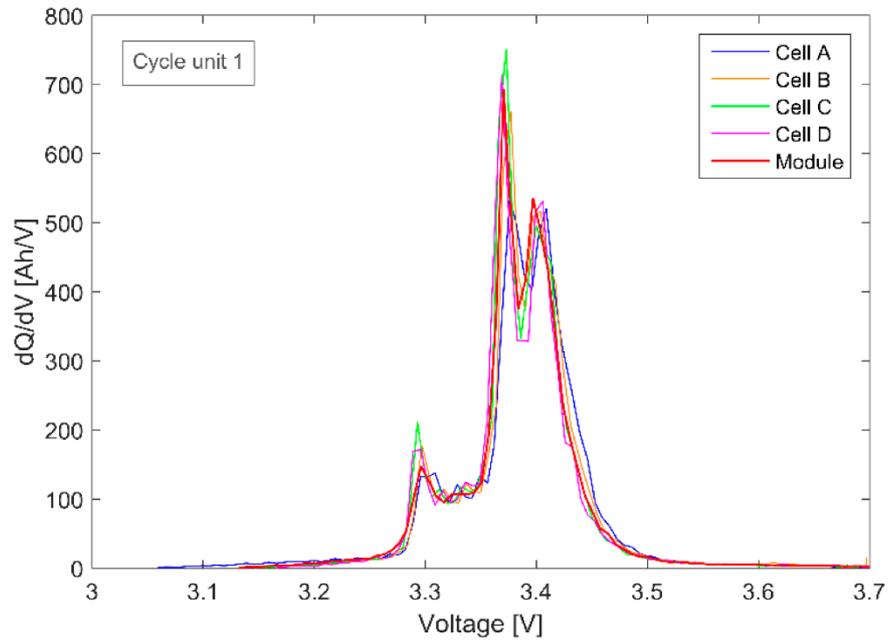


Fig.4 ICA curve for different batteries

There are three main peaks on IC curve, corresponding to the three voltage plateaus on the cell voltage versus stater of charge  $U_{cell} = f(\text{SoC})$  curve. In order to analyze ICA curve, the evaluation of the features of interest include peak height, area, position, these allows conclusions to be made regarding the capacity and changes in the internal resistance of a battery cell. ICA is suitable for non-destructive SOH determination as it only needs current, voltage data. The computing of the IC curve and the evaluation of the battery SOH also require a low computing effort, which is an advantage over model-based methods like equivalent circuit models that are obtained using electrochemical impendence spectroscopy (EIS). More information about alternative SOH estimation techniques can be found, for example, in an article by Berecibar et al. [17]

SOH estimator can use various peak as data. For example, the third peak. It can be easily seen that the area under ICA curve is the cell capacity between two voltage limits:

$$\int_{u_1}^{u_2} ICA(u) du = \int_{u_1}^{u_2} \frac{dQ(u)}{du} du = Q(u_2) - Q(u_1) \quad (3)$$

After calculate ICA area (variation of capacity), the SOH value can be calculated by following equation:

$$SOH(in\%) = \frac{ICA \text{ area (In \% of new cell area)} + \alpha}{\beta} \quad (4)$$

Where  $\alpha$  and  $\beta$  are two sensitive parameters need to defined according to different robust test. For example, the temperature and charging current have different  $\alpha$  and  $\beta$  value.

### 2.4.2 Differential thermal voltammetry (DTV) analysis

One of the important tools of SOH computing technology is DTV. It combines temperature measurement and ICA analysis. During discharge and charge, the DTV detects the cell surface temperature and plots temperature versus voltage (dT/dV). The main goal of DTV is to analyze the entropy-related information changes during battery operation. The study showed that as the battery ages, the position and amplitude of the peaks of the entropy curve change, like what the curve of a conventional ICA would represent.

The peak of the largest DTV change can be used to determine the aging of the battery. The analysis found that the position and height of the peaks changed significantly as the internal resistance of the cell increased. Generally, DTV can achieve higher C rate charge and discharge tests than traditional ICA. Another benefit of DTV is isothermal control of the battery, as it only affects the results if the temperature is significantly higher than the ambient temperature. This makes the experiments operationally easier and less expensive.

### 2.4.3 Differential mechanical parameter (DMP) analysis

The stress-strain parameters of the battery can also be used as the parameters for SOH analysis. According to the working principle of lithium-ion batteries, the intercalation/deintercalation process of lithium ions in electrode materials is related to the volume change and internal stress of the battery. Mechanical stress in the internal

structure of the cell is the result of electrode expansion against the constraints which is normal to the electrode plane. This stress can be measured by sensors mounted on the cell surface. Recent studies have found that this stress-SOH relationship is due to the growth of SEI, so  $d\varepsilon/dQ$ , and its second derivative, or  $d\varepsilon/dV$ , can be analyzed to obtain a curve like the ICA analysis.

Table 1 summarized the features and estimation methods used for those traditional SOH estimation methods.

Differential analysis	Current rate	Features	Battery chemistry	Ref
IC	C/10	Peak height	LFP	Jiang et al.
IC	C/2	Peak position	NMC	[18]
IC	1C	Peak area	NMC	Li et al. [19]
IC	C/2	Peak height	LFP	Tang et al. [20]
IC	C/20	Peak position	NMC	Weng et al.
DV	C/5	Regional cap.	LFP	[21]
DV	1C	Location interval	LFP	Zhang et al. [22]
				Berecibar [23]
				Wang et al. [24]

Table 1 Features and estimation methods for traditional SOH estimation

Table 2 shows the advantages and disadvantages for each method.

Methods	Advantages	Disadvantages
ICA analysis	<ol style="list-style-type: none"> <li>1. Easy to monitor, only needs two parameters</li> <li>2. Can be applied to different types of cells.</li> </ol>	<ol style="list-style-type: none"> <li>1. Limited to low current rate.</li> <li>2. Sensitive to measurement noise</li> <li>3. Influenced by the operation temperature</li> </ol>
DTV analysis	<ol style="list-style-type: none"> <li>1. Easy, only needs two parameters</li> </ol>	<ol style="list-style-type: none"> <li>1. Needs additional and calibrated temperature</li> </ol>

	2. Can be used for monitoring cells in parallel.	sensors 2. Sensitive to testing temperature variations
DMP analysis	1. Can be applied for cells with a high initial SOC. 2. Not limited to low and constant current rates. 3. Applicable to high current rates.	1. Needs additional equipment for the mechanical parameter. 2. Not applicable to cells constrained with hard covers 3. Difficult for online application

Table 2 Advantages and disadvantages for traditional methods

## 2.5 Introduction of neural networks

### 2.5.1 Basic components and structure of FNNs

As table 2 shows above, traditional methods have lots of limitations and they are hardly applicable for online real-time application.

Neural network is a new technique raised up in recent years, thanks to the quick development of hardware (CPU/GPU). The most basic function of a neural network is to learn features from the input data (such as text, pictures, signals etc...) and give its predictions.

#### *Perceptron:*

The basic unit of a neural network. perceptron can be considered as a mathematical mapping, it does some math calculation to the input, and produces one output, figure.5 shows a structure of Neuron.

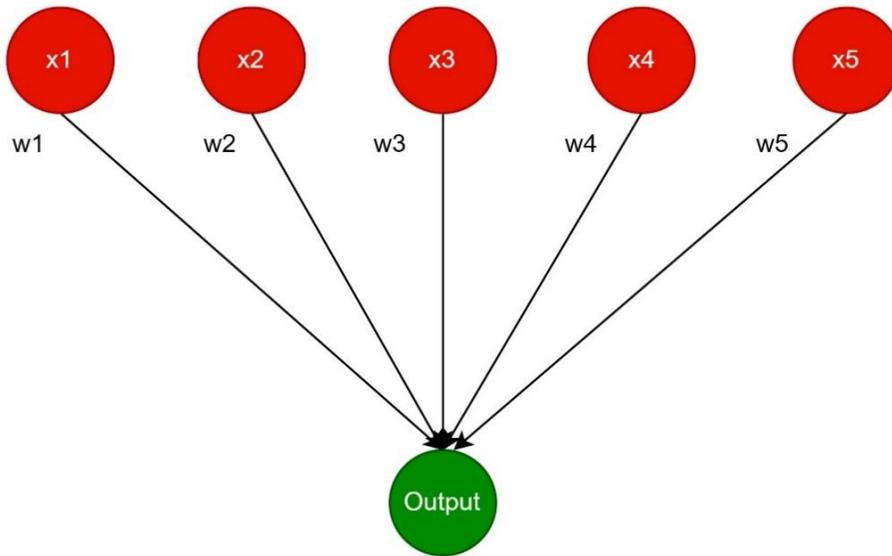


Fig.5 Neuron structure

The mathematical operation is the following expression, as eq5 shows:

$$f(x) = \sum_i w_i x_i + b = \langle w, x \rangle \quad (5)$$

Where  $w$  is weighting factor and  $b$  is bias.  $x_1$  to  $x_5$  are five features of the task. Perceptron adjusts themselves to minimize the loss function until the model get a very high accuracy. Learning process is to estimate the parameters  $w$  and  $b$ .

### Hidden layers

In neural networks, a hidden layer is positioned between the input and output layer. In hidden layers the functions apply weights to the inputs then directs them through an activation function as the output. The hidden layer gives nonlinearity to the inputs.

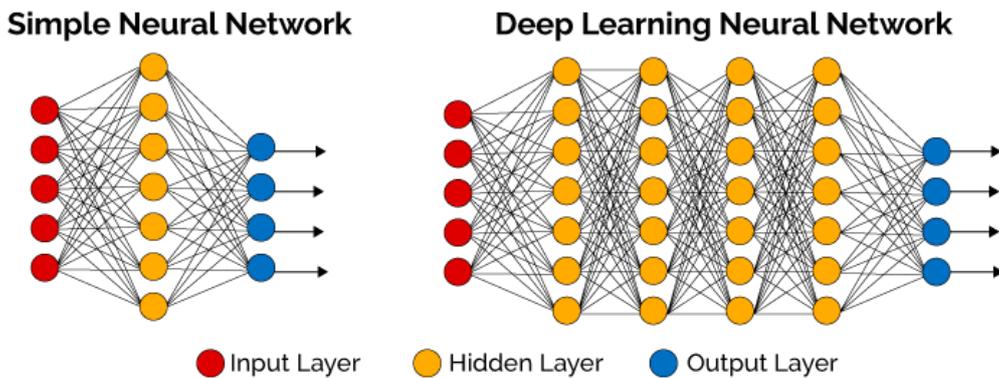


Fig.6 Structure of Neural network

### Activation functions

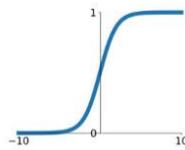
Activation function is usually located in the end of hidden layer, before the output layer. It turns the network from a simple linear matrix transformation to a complex nonlinear transformation thus increase the learning capability of the network.

Activation function is like a gate that checks if an incoming value is greater than threshold value. If the value is large enough, the activation function activates, otherwise it does nothing.

The table 3 shows some common activation functions:

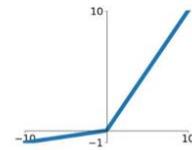
#### Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



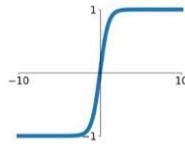
#### Leaky ReLU

$$\max(0.1x, x)$$



#### tanh

$$\tanh(x)$$

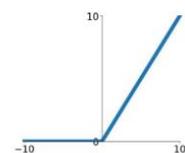


#### Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

#### ReLU

$$\max(0, x)$$



#### ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

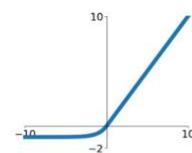


Table 3 Different activation functions

### Fully connected layers:

As mentioned above. A neural network architecture is composed by means of perceptron. Each individual perceptron consists of a function. In fully connected layers, the neuron applies a linear transformation to the input vector through a weight's matrix.

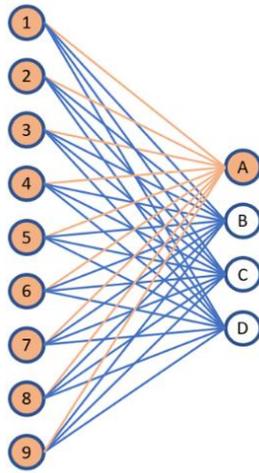


Fig.7 Fully connected layer

*Gradient Descent:*

Training process need to minimize the loss function by estimating weighting factor and bias. Gradient descent is designed for this aiming.

A gradient simply measures the change in all weights regarding the change in error. It can be considered as a slop function (derivative function). The higher the gradient, the steeper the slope and the faster a model can learn.

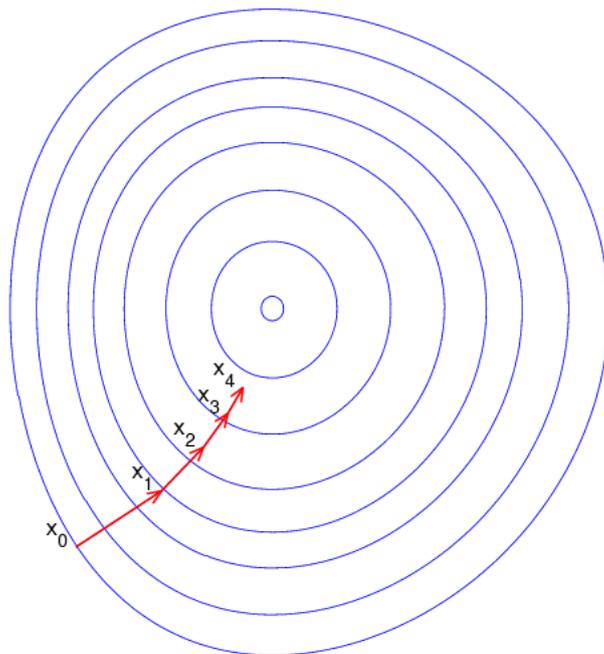


Fig.8 Gradient descent path

As figure.8 shows, the closed line represents contour line. The gradient descent starts from  $x_0$ . At  $x_0$ , it finds the steepest step to reach  $x_1$ , then the function finds the steepest step to descent from  $x_1$  to  $x_2$  and so on. This process can be expressed by eq6

$$b = a - \gamma \nabla f(a) \quad (6)$$

In eq6,  $b$  is the new value after descent,  $a$  represents the current value. The minus sign refers to the minimization part of gradient descent.  $\gamma$  is the learning rate, it determines the step length of each descent.  $\nabla f(a)$  is the direction of the steepest descent. It is crucial that to choose an appropriate learning rate  $\gamma$ . Small learning rate can cause a very slow descent speed, thus, increase training time. Large learning rate can make network not reach the local minimum because it bounces back and forward between the convex function. Figure.9 shows these conditions.

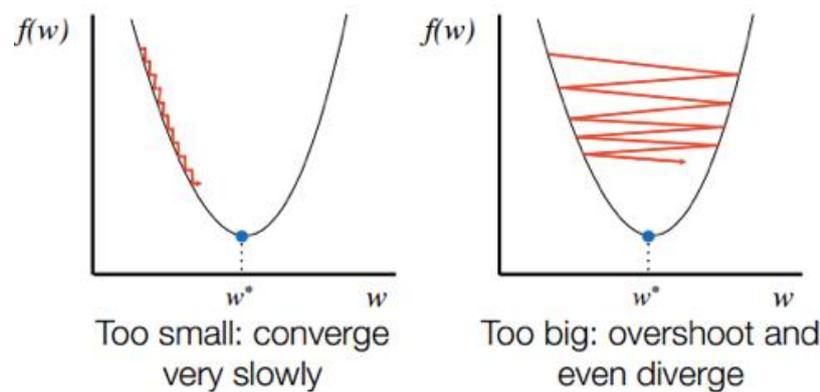


Fig.9 Influence of a not appropriate learning rate

### 2.5.2 Recurrent neural network

A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (nlp), speech recognition, and image captioning. They are incorporated into popular applications such as Siri, voice search, and Google Translate.

Traditional deep neural networks consider that inputs and outputs are independent from each other. RNN can take information from prior inputs to influence the current

input and output, thus the memory. Figure 10 shows RNN and normal FNN architecture.

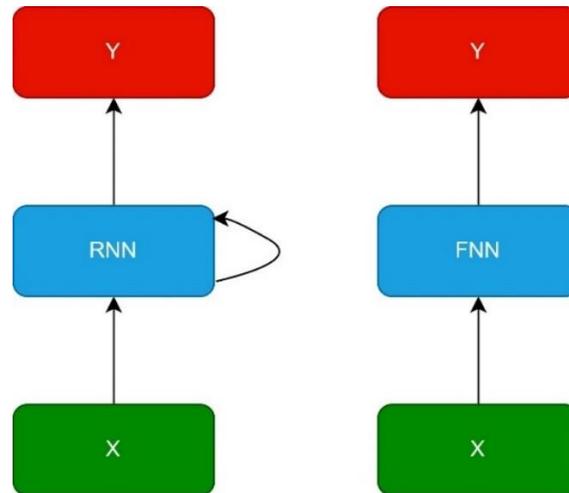


Fig.10 RNN and FNN architecture

We can process a sequence of vectors  $x$  by applying a recurrence formula at every time step:

$$h_t = f_W(h_{t-1}, x_t; \theta) \quad (7)$$

$h_t$  is new state;  $f_W$  is a certain function with parameters  $W$ ;  $h_{t-1}$  represent old state and  $x_t$  is the input vector at a certain time step.  $\theta$  is the set of parameters. Training of the RNNs is training  $\theta$ .

In vanilla RNN,  $f_W$  is usually a 'tanh' function. Thus  $h_t$  can be written as:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t; \theta) \quad (8)$$

$$y_t = W_{hy} h_t \quad (9)$$

Feedforward networks map one input to one output while RNN do not have this constraint. There are several different types of RNNs:

One-to-One:

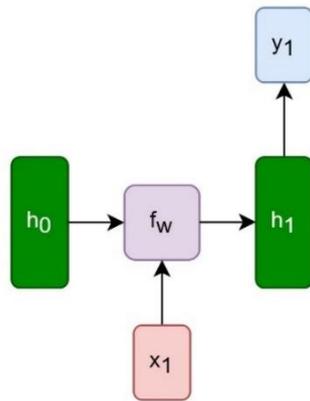


Fig. 11 One-to-One RNN structure

One-to-Many

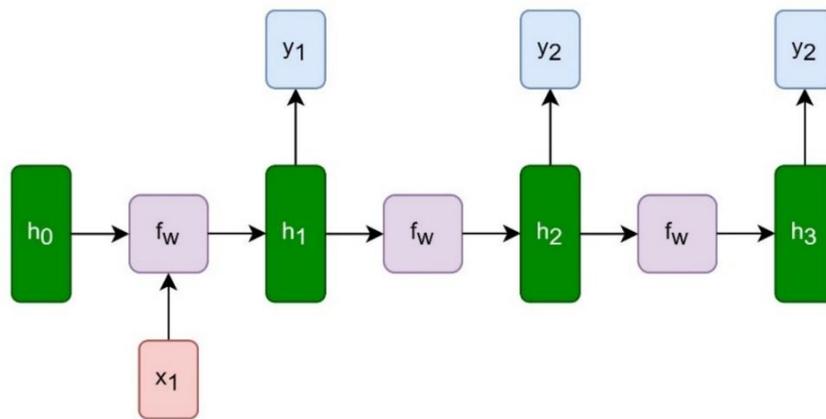


Fig. 12 One-to-Many RNN structure

Many-to-One:

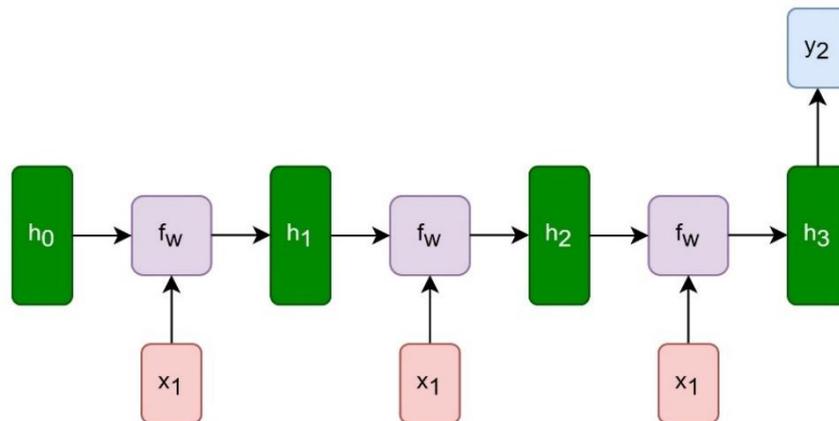


Fig. 13 Many-to-One structure

Many-to-Many:

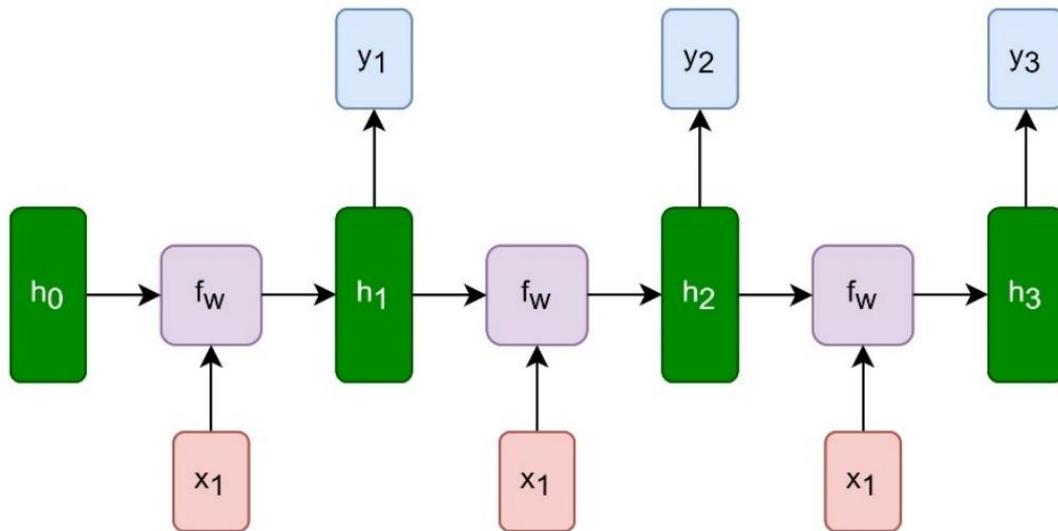


Fig. 14 Many-to-Many structure

Recurrent neural networks utilize ‘backpropagation through time (BPTT) algorithm’ to determine the gradients. The concept of BPTT is the same with respect to conventional backpropagation: the model correct itself by calculating errors from output layer to its input layer. But the difference between BPTT and conventional one is that BPTT sums error at each time step while conventional approach does not need to sum errors. This is because conventional approach does not need to share parameters in each layer (they do not have memory).

To explain it will, we can write a recurrence relation to show the iteration chain.

$$h_t = f(h_{t-1}, x_t; \theta) = f(f(h_{t-2}, x_{t-1}; \theta), x_t; \theta) = f(\dots f(h_0, x_1; \theta), \dots x_t; \theta) \quad (10)$$

To train the network means to train  $\theta$ . This means during the propagation it needs to compute all hidden states  $h_1, \dots, h_t$  to calculate the gradients of the network, so it is called BPTT.

When vanilla RNNs operate BPTT to train itself, there will be two problems: “Gradient exploding” and “Gradient vanishing”. These problems are usually caused by

inappropriate gradient size setting. When the gradient is too small, it continues becoming smaller thus the speed of gradient descent is getting slower and slower. At the end the algorithm will no longer learning. Gradient Exploding occur when the size of gradient is too large. In this case, the gradient will grow too large (in some case it will tend to NaN).

From eq10 we can derive a matrix equation to explain this problem:

$$h_t = f(\dots f(h_0, x_1; \theta), \dots x_t; \theta) = W^T(\dots W^T h_0 \dots) = (W_t)^T h_0 = Q^T \Lambda_t Q_0 \quad (11)$$

Repeated application of the same weighting parameter through all hidden layer leads to the eigenvalues  $\Lambda$  of the parameters to be raised to  $t$ , that is the length of the sequence. This means, as the iteration operating, eigenvalues  $\Lambda$  smaller than 1 is lead to vanishing eigenvalues to 0 while eigenvalues larger than 1 is led to exploding eigenvalues.

### 2.5.3 Long Short Term memory (LSTM)

As it has been mentioned in 4.2.3, RNNs have gradient vanishing and exploding challenge. To overcome this, long short-term memory has been developed. LSTM allows the network to remember longer sequences. They are using gate unit to avoid the problem.

LSTMs are most widely used structure of RNNs. Figure.15 gives a basic structure of LSTM.

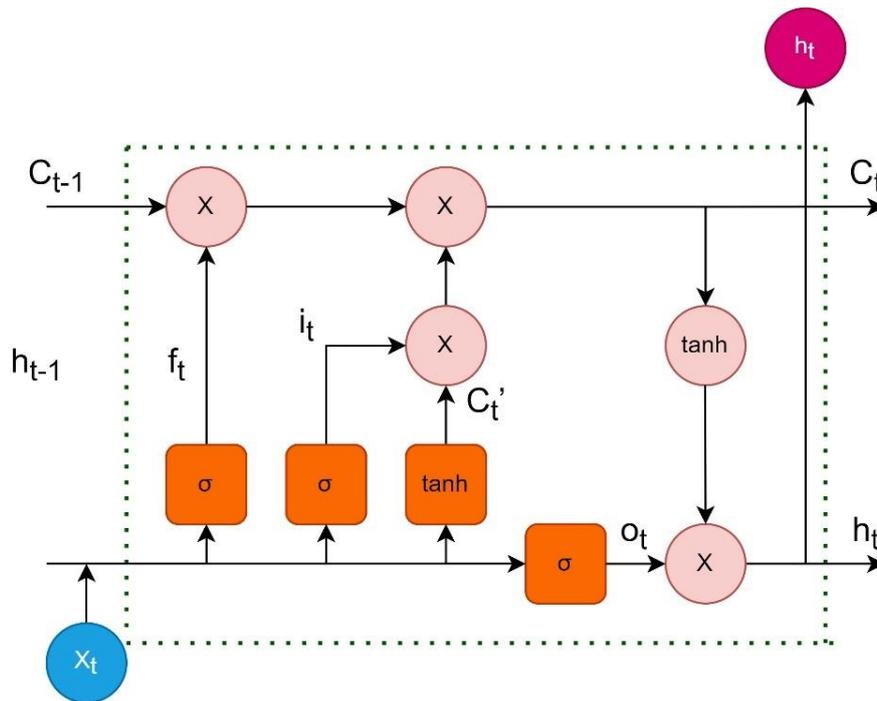


Fig.15 LSTM structure

The core of LSTMs is the state of cell, as shown in figure.16 red line.

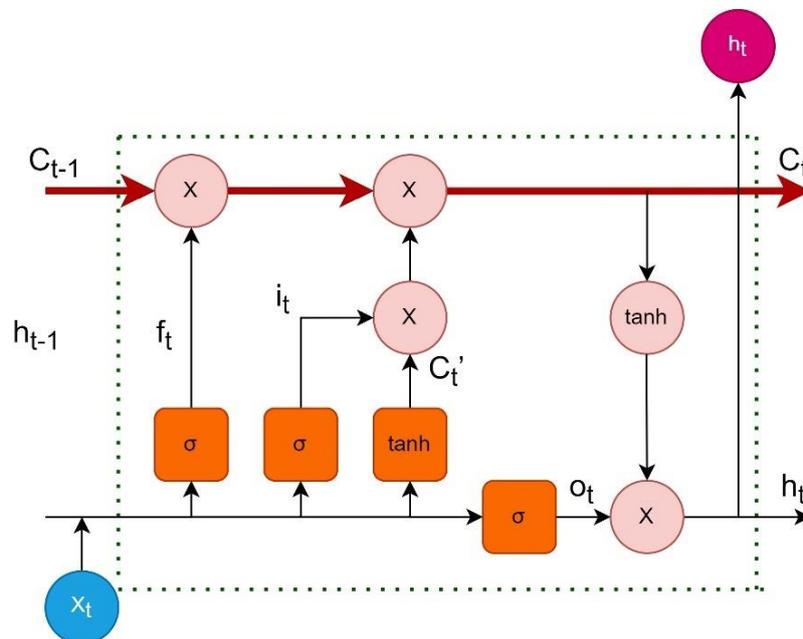


Fig.16

The gate unit is shown in figure.17. They are a way that optionally let information through. They are a sigmoid neural net layer and a pointwise multiplication operation.

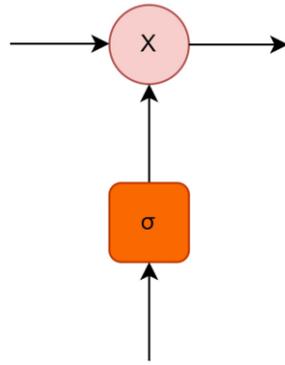


Fig.17 Gate unit

The sigmoid function can only output numbers between zero and one depending on if the signal is large enough to activate it. A value of zero simply means “nothing can pass” while a one value means “everything can pass”. The LSTMs has three of these gates to control the cell state.

#### 2.5.4 A walk through step-by-step in LSTM

As shown in figure.18 the first step of LSTM is to let sigmoid layer decide if the information can throw away from the gate. This sigmoid layer is also called “forget gate layer”. It outputs 1 or 0 to  $C_{t-1}$  depends on  $h_{t-1}$  and  $x_t$  value. 1 stands for “add  $h_{t-1}$  and  $x_t$  value into  $C_{t-1}$ ” while 0 stands for “completely get rid of this”. Eq12 can express this procedure.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \quad (12)$$

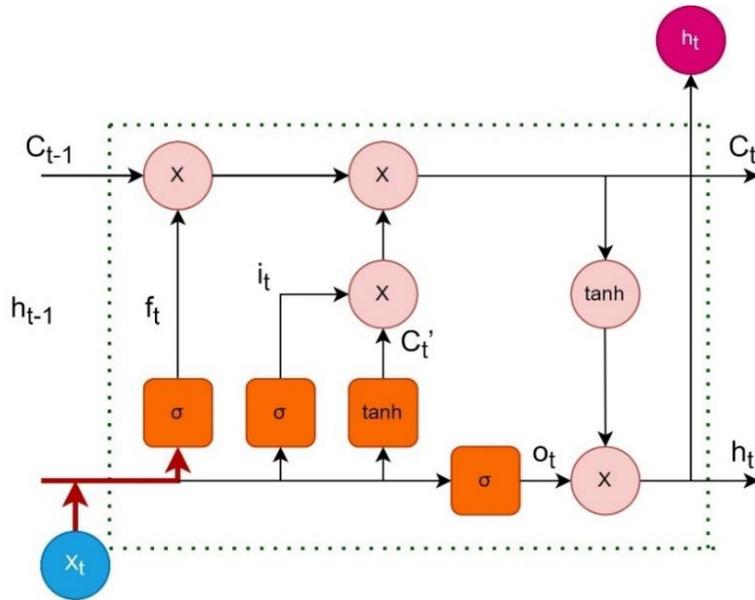


Fig.18

The second step is to decide if  $h_{t-1}$  and  $x_t$  are going to be memorized. As figure.19 shows, sigmoid function (known as “input gate layer”) is deciding the operation. Then a tanh function creates a new vector  $C_t'$ . Then  $C_t'$  will multiply it, which is the output of input gate layer. Since sigmoid can only output 1 or 0, this multiplication is deciding the final value is  $C_t'$  or 0 thus if it should be added into cell state. Eq13 and eq14 also express this procedure.

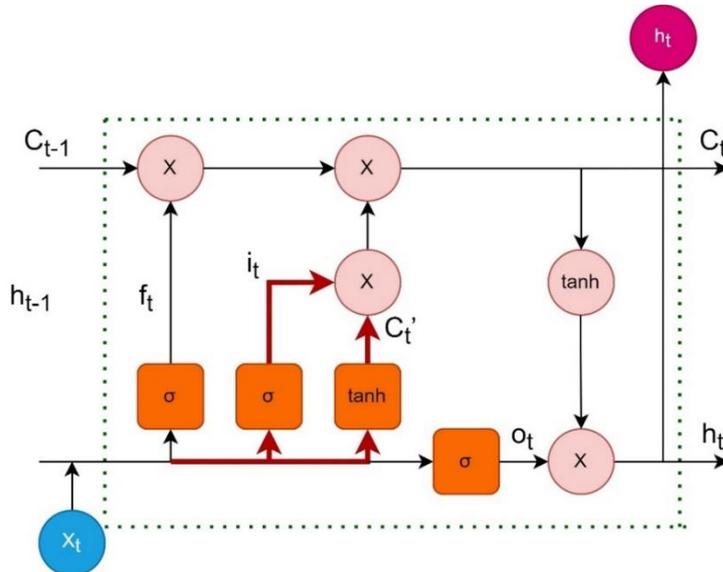


Fig.19

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \quad (13)$$

$$C'_t = \tanh(W_C * [h_{t-1}, x_t] + b_C) \quad (14)$$

The third step is to upgrade cell state  $C_{t-1}$  to  $C_t$  (shown as figure.20). This procedure is done by previous calculation. Two sigmoid layers decide if the new value should be added into cell state. Eq15 can represent it.

$$C_t = f_t * C_{t-1} + i_t * C'_t \quad (15)$$

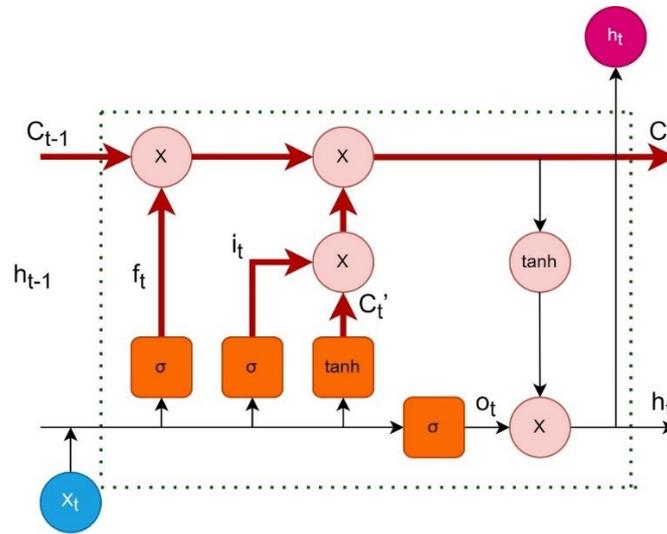


Fig.20

The final step is to decide what it is going to be output (figure.21 and eq16,17). Cell state is firstly applied by a tanh function so that push its value between -1 and 1. Then the third sigmoid layer applies to input to give  $O_t$ , according to the elements in vector  $O_t$  (1 or 0), network decide which element of the cell state should be output.

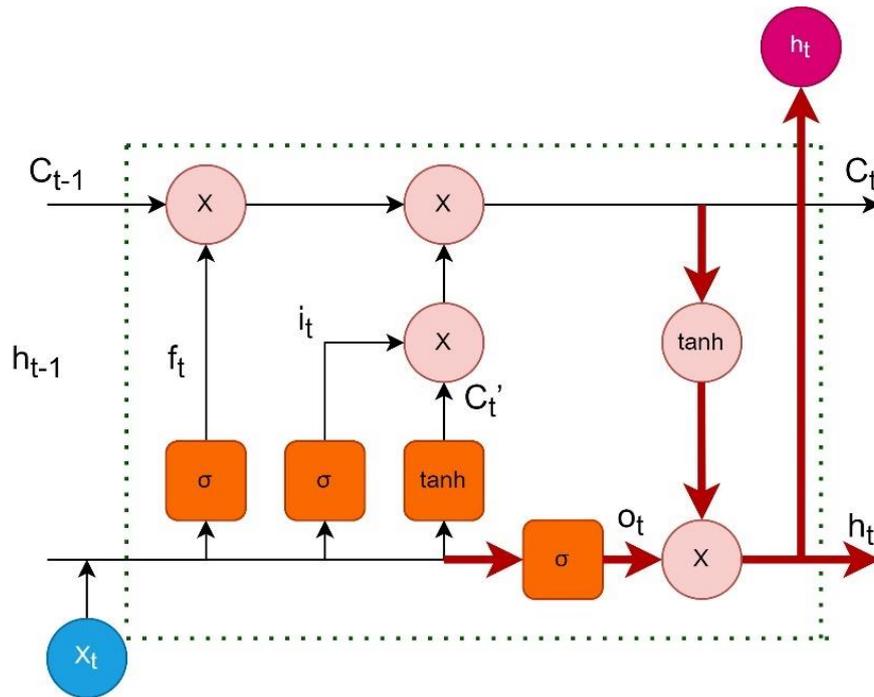


Fig.21

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \quad (16)$$

$$h_t = o_t * \tanh(C_t) \quad (17)$$

## Chapter 3. SOH estimation methodology

In this section the working principle and steps of the whole project will be described in detail, and a comparison and discussion of the results will be given at the end.

Here figure.22 shows a flow chart to explain it better.

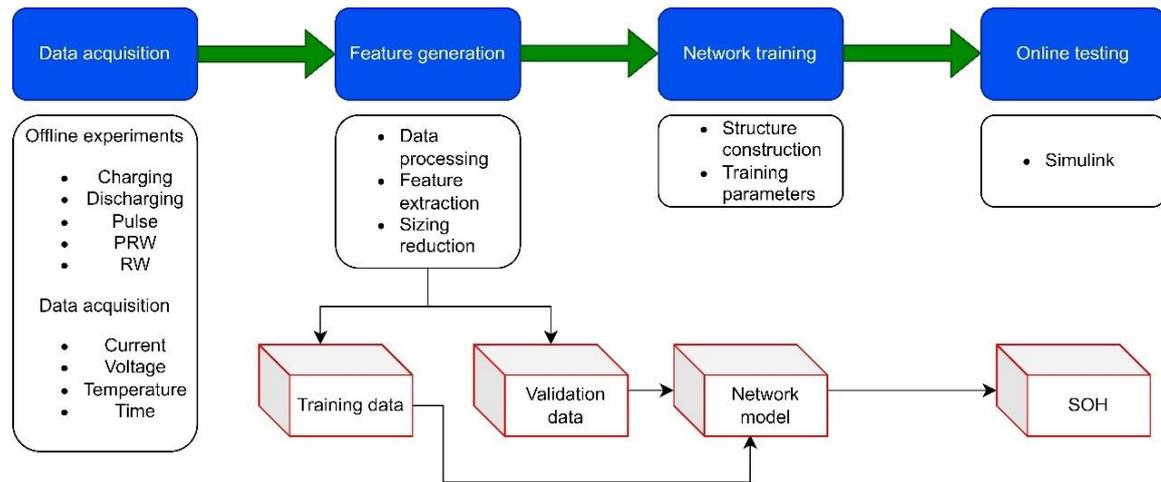


Fig.22 Model flow chart

Training process start with data acquisition phase. During this phase a test bench was operated a well-designed charging/discharging profile to age the cells. Current, voltage and temperature profile were collected as training data.

During feature generation phase. Some features were extracted from raw data. To do this, move sliding window approach has been applied. To test the robustness of the model, training data were collected from cell #3 while validation data were collected from cell #1.

Network training phase aimed to construct the neural network structure and train it with different group of parameters to get the lowest RMSE.

The last phase is the implementation on Simulink. Since in previous phase was already got a well-trained network. This phase is to replicate the whole model into Simulink and reach the real time application requirement.

### 3.1 Battery introduction

The battery cell used for data acquisition is Sony Murata US18650vtc6.



Fig.23 Sony Murata US18650vtc6 cell

This battery has the following parameters and characteristics.

Nominal Capacity at 0.2ItA	3120mAh 11.23Wh	Discharge 2.0V cut off at 23°C
Rated Capacity at 0.2ItA	3000mAh 10.8Wh	Discharge 2.0V cut off at 23°C
Capacity at 1ItA	2850mAh 10.12Wh	Discharge 2.5V cut off at 23°C
Capacity at 10ItA	2700mAh 9.18Wh	Discharge 2.5V cut off at 23°C
Nominal Voltage	3.6V	
Internal Impedance	13mΩ	Measured by AC1kHz
Cycle performance	53% Min. Of initial capacity at 300 cycles	10A discharge

Table 4 Parameters of cell VTC6

The standard charge condition for this battery is:

Charge method	Constant current/Constant voltage
Charge Up Voltage	4.2±0.05V
Charge Current	3.0A
Charge Time	2.5h
Ambient Temperature	23°C

Table 5 Standard charge condition of Sony Murata US18650vtc6

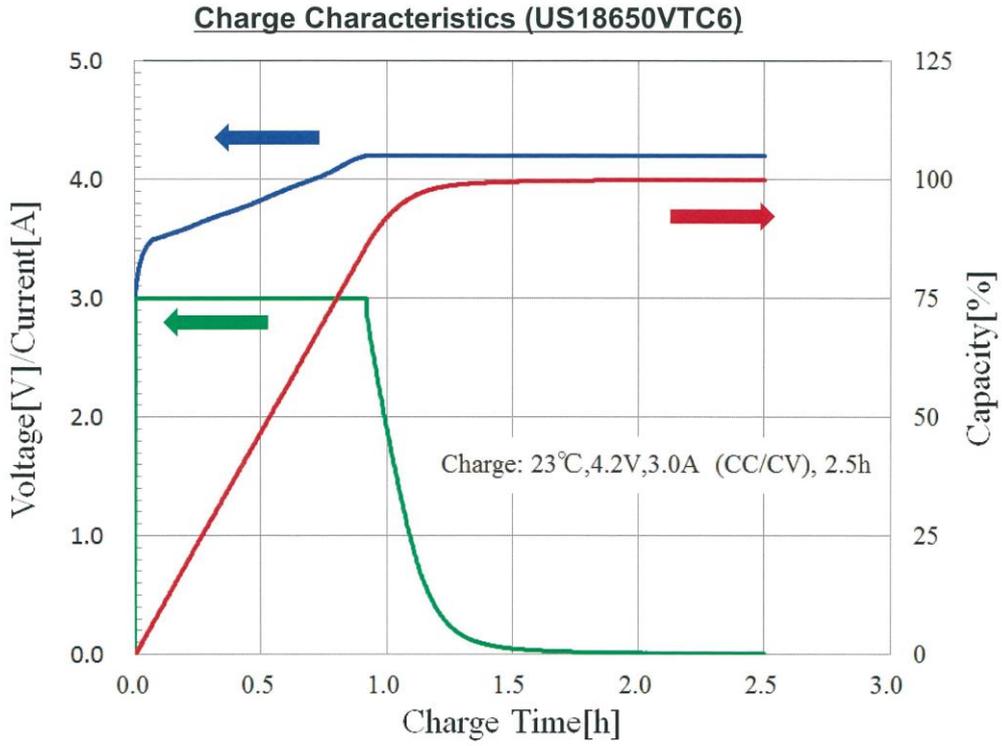


Fig.24 Charge characteristics of Sony Murata US18650vtc6

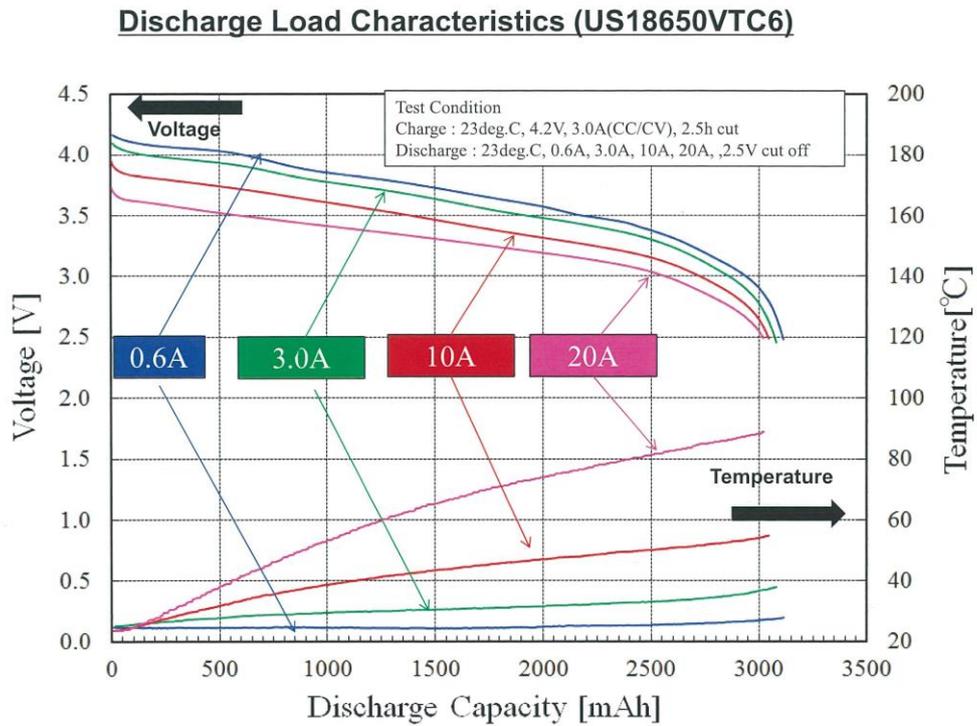


Fig.25 Discharge characteristics of Sony Murata US18650vtc6

The raw data is acquired from a test bench. The test bench has six VTC6 cells connected in series. Elithion cell boards aimed for cell voltages measurement while three LM35 Texas instrument temperature sensors measure the temperature profile. Considering the safety problem, an Elithion (Lithulmate) BMS and an emergency stop device are also connected on the test bench. An Arduino Mega board is connected via LAN to a PC to acquire the measured data.

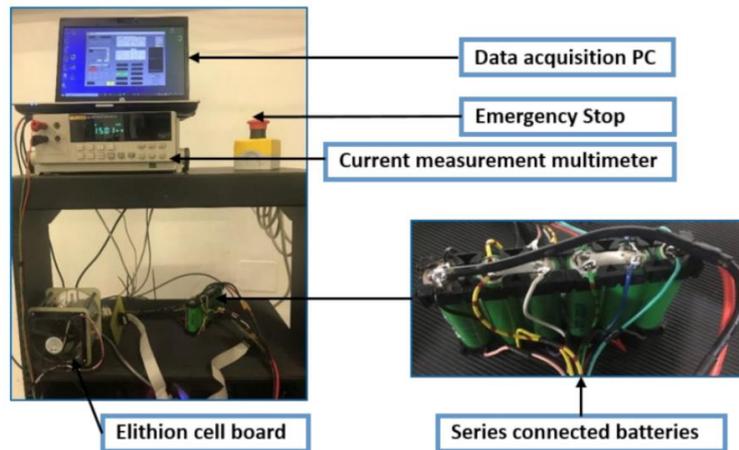


Fig.26 Experimental setup for data acquisition

The experiment is under a controlled ambient temperature condition (25 Celsius degree). Between each test phase the cells are cooled by natural convection with a certain air circulation in the room. The room temperature dose not influenced by cells. Three temperature sensors are installed on the surface of cell 1, 3, 4.

## 3.2 Test cycle introduction and training dataset selection

To age the cell, the main test cycle is composed by five different profiles: Charging profile; Discharging profile; Pulse profile; Polarized random walk profile. Random walk profile.

### 3.2.1 Charging profile

Cycle starts with 0% SOC and with  $C/2$  charging rate. In this experiment, all charging phases were fully charged, and all had the same  $C$  rate.

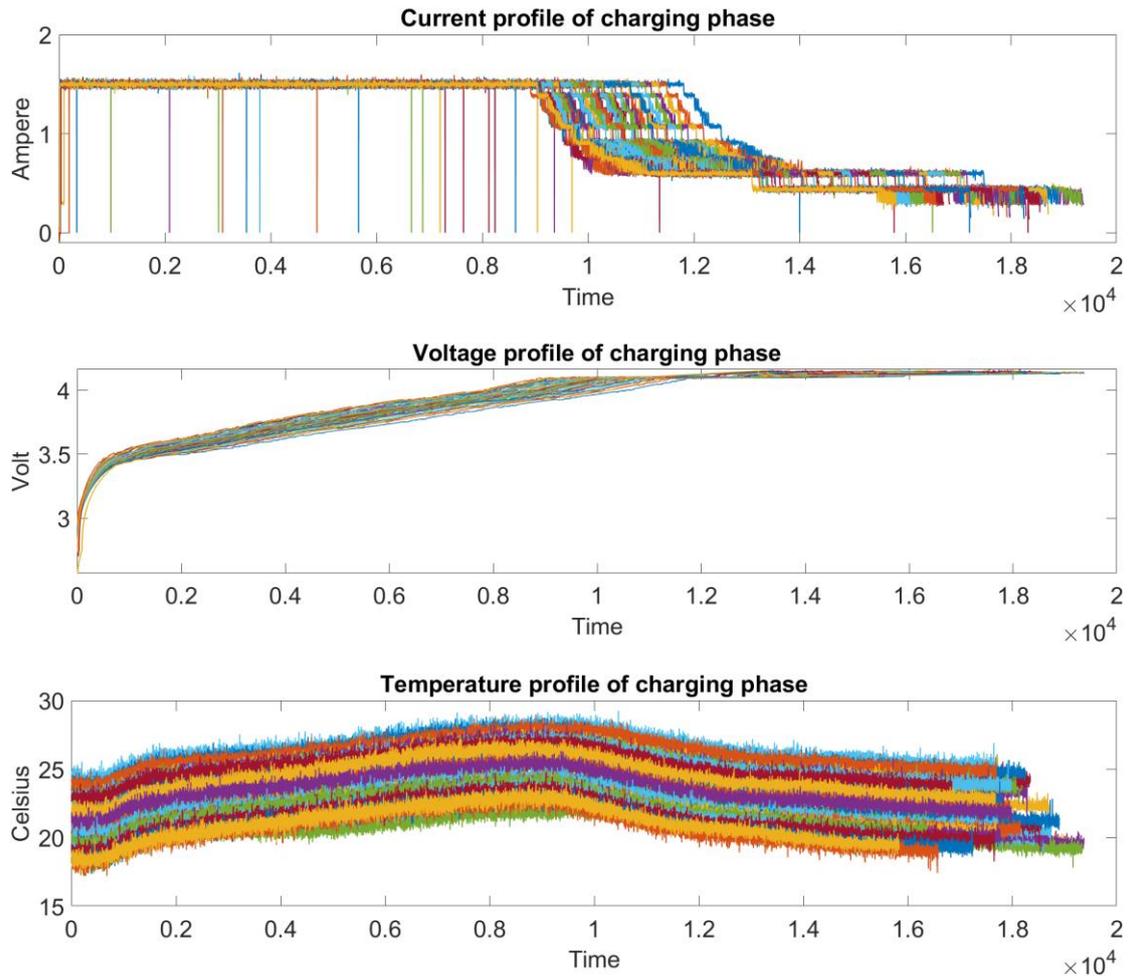
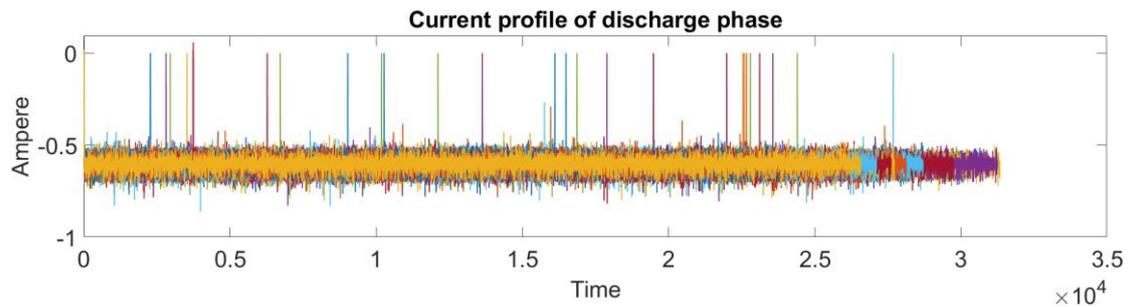


Fig.27 I,V,T profile of Charging phase

### 3.2.2 Discharging profile

Discharging phase start from 100% SOC aimed to fully discharge the battery. In some phase in order to simulate the real conditions, fully discharging can be done during two phases.



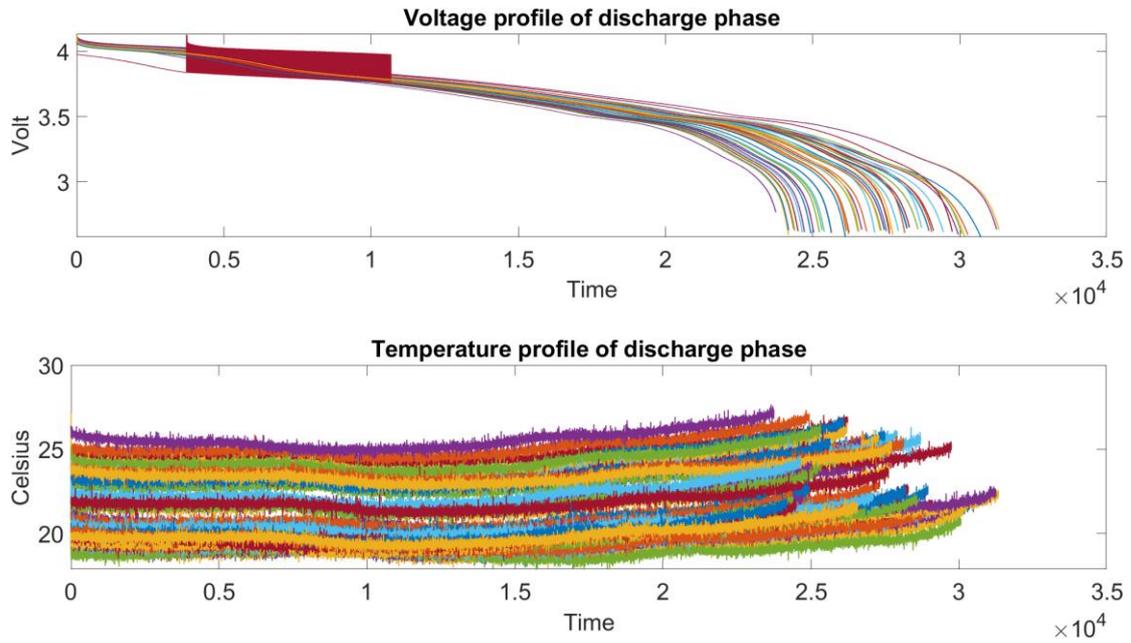
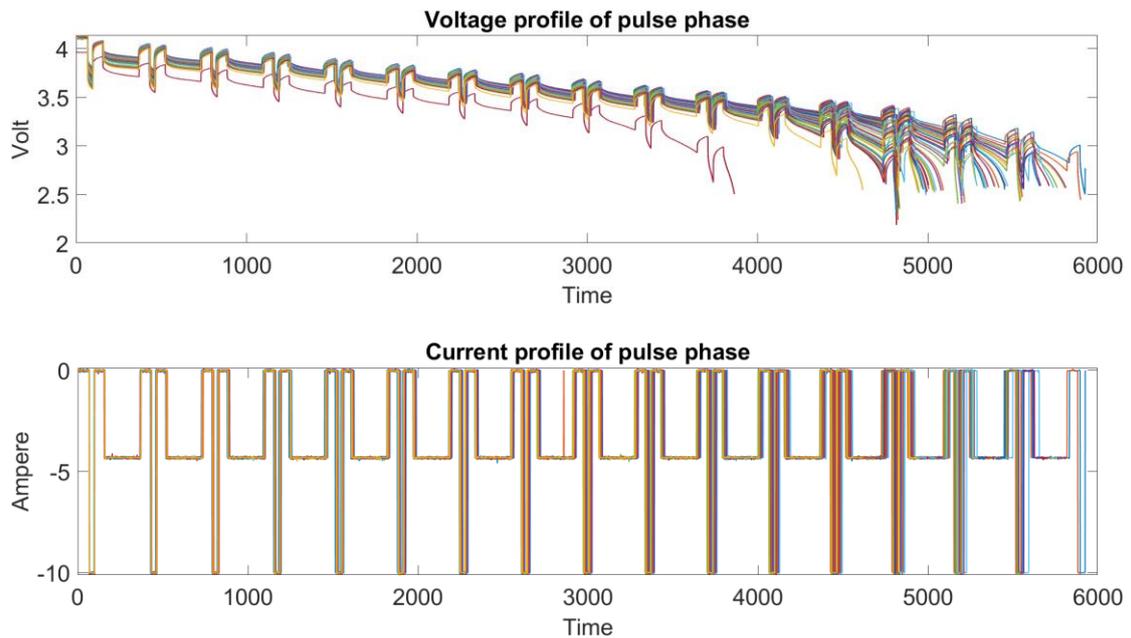


Fig.28 I,V,T profile of discharging phase

### 3.2.3 Pulse profile

This process is still a discharge, but with a very large dynamic range. This setting can best reflect the operation of the battery pack under the condition of regular dynamic power demand.



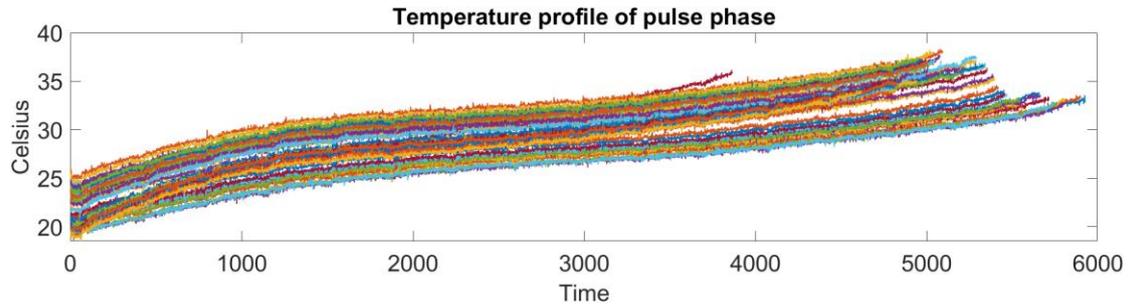


Fig.29 I,V,T profile of Pulse phase

### 3.2.4 Polarized Random Walk profile

This process is still discharge, but unlike pulse phase, in addition to the highly dynamically power requirements, a small amount of charging characteristics is added while ensuring overall phase is discharge. The appearance of charge and discharge is an irregular random signal. This phase can simulate the real use situation and has a high practical reference significance.

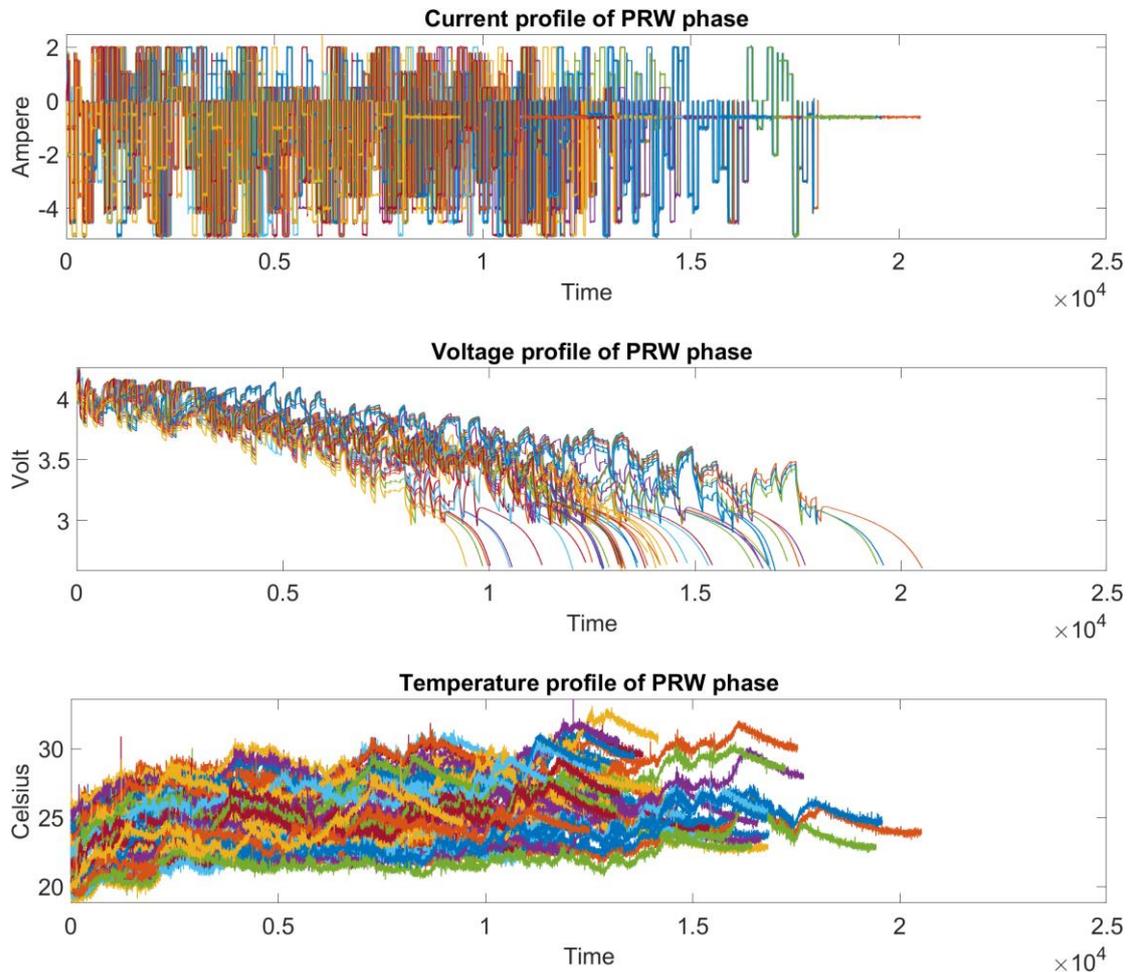


Fig.30 I,V,T of PRW phase

### 3.2.5 Random Walk profile

The random walk algorithm is used to generate random signals without any regularity, which is used for the aging battery. The sequence of the charging and discharging process and the magnitude of the current are completely random. The SOC remains unchanged throughout the random walk process

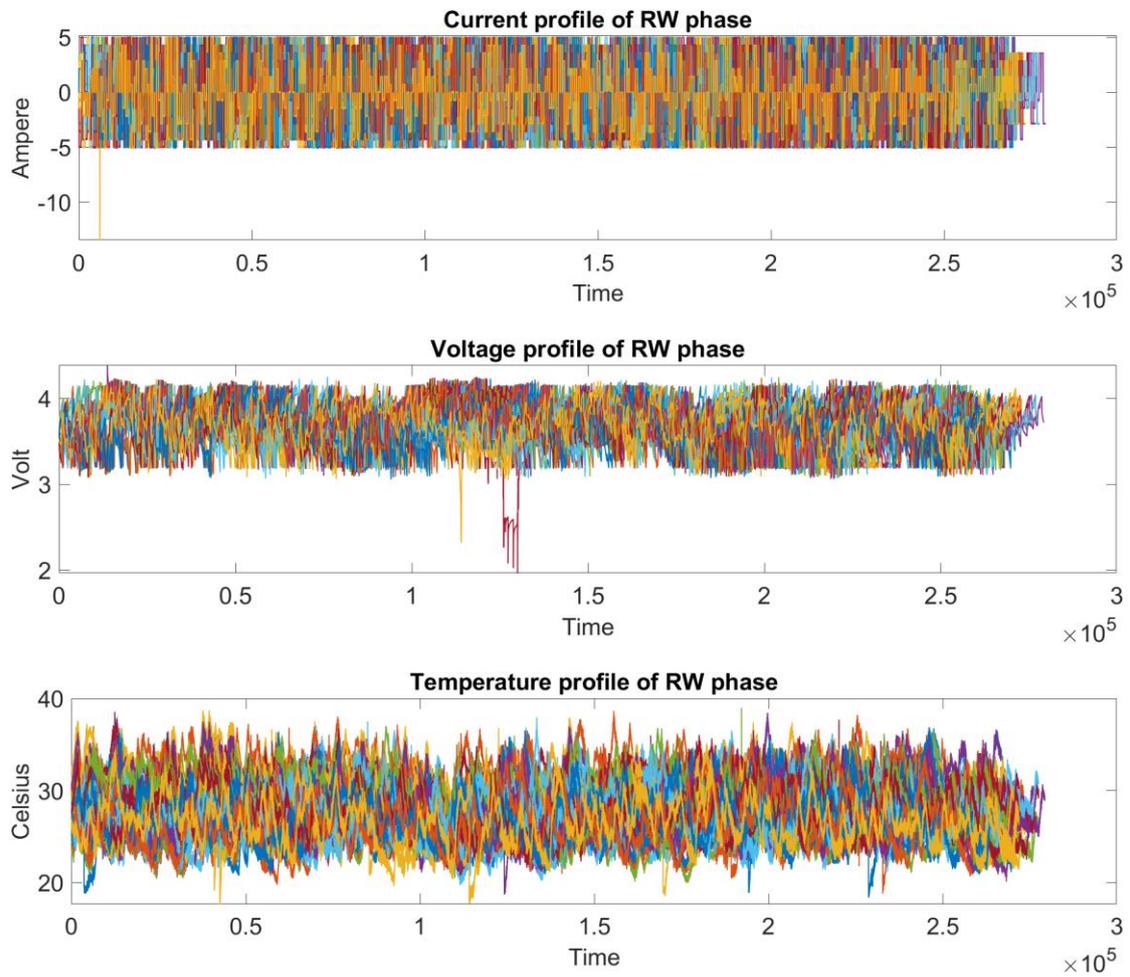


Fig.31 I,V,T profile of RW phase

All these 5 cycles are combined with a certain ordinary as follow:

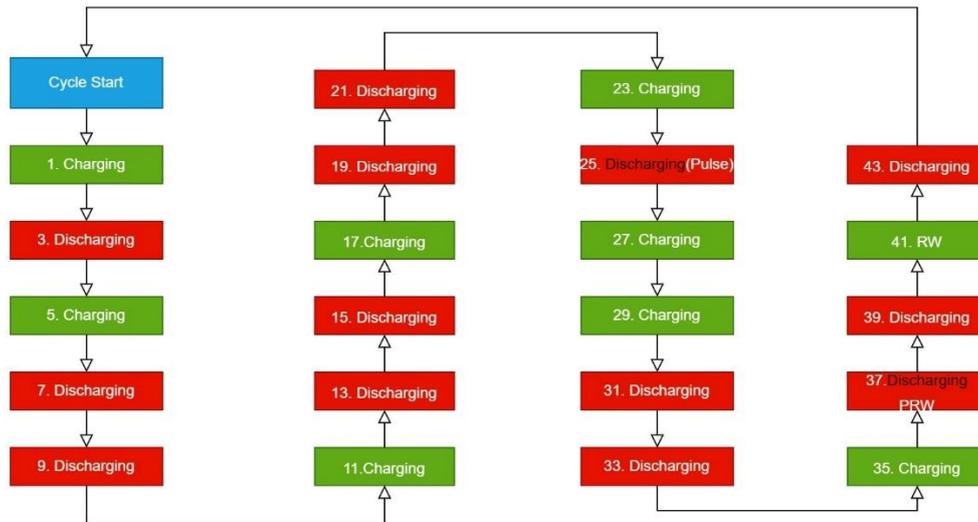


Fig.32 Test cycle

This cycle is composed by 43 phases. In figure there are only action phases, between each action phases a break phase has been inserted to cool down the cell.

Because SOH has a very low dynamic, this test cycle has been operated for 55 times.

The whole experiment lasted for several month.

After all these operations, the SOH reduced from 100% to almost 82%. All acquired data are saved in 2763 files which can be processed by MATLAB. Figure.33 shows a small part of these data.

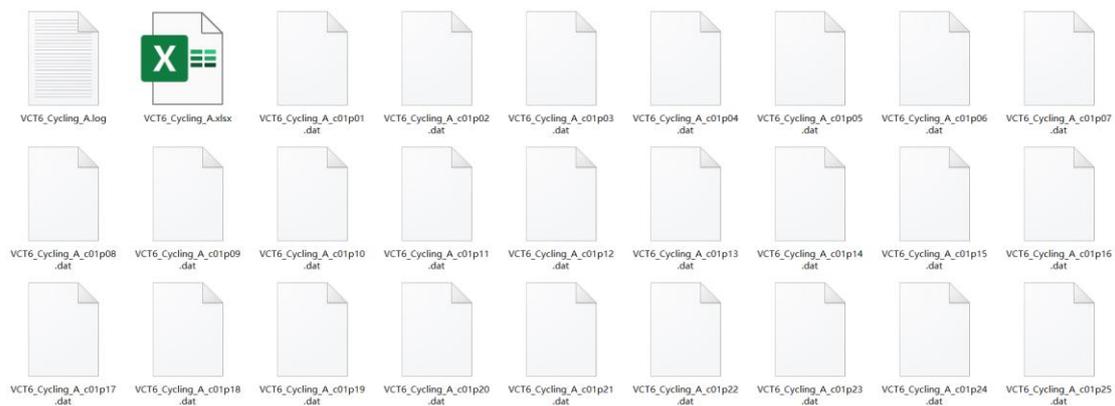


Fig.33 A part of all data

The training dataset of the network are “charging profile”, “pulse profile” and “polarized random walk profile”. There are two main reasons for select these three profiles.

First, to train a neural network the ground truth (label) value is necessary. As eq1 shows, to compute SOH it is necessary to know  $C_r$  for each test cycle since it varies from initial

value to a reduced value. Due to cycle design, not all phases are complete charging or discharging phase. These phases are complete charging/discharging profile, so it supports the calculation of SOH ground truth value. Therefore, the thesis will use a specific phase to represent the aging caused by all 43 phases. This is a compromise of data.

Second, these phases represent three different usage condition. Training on these three training sets can well verify the robustness of the network, thereby increasing the reliability of the experimental results.

It must be highlighting why the PRW phase was chosen instead of the RW phase as the training dataset. The purpose of RW phase is to age the battery, due to the signal is highly random, the features changing brought about by the reduction of SOH will be covered by random noise, so it is impossible to extract useful information from this cycle for network training. Moreover, in the real operating conditions, the charging of the electric vehicle during driving is brought by the energy recovery system. Although the charging provided by this system is random, the SOC is constantly decreasing, which is different from the RW phase (RW required SOC remains constant). Therefore, the PRW phase will be used as the training dataset instead of the RW phase.

### 3.3 Feature introduction

Feature generation is a pivotal step for the training process. Feature processing can highly influence the performance of network. It is necessary that to choose the high relevant and precision features as training dataset. Figure.34 summarized different feature variables for model training. They are divided into four perspectives, including incremental calculation, time, envelope area, and model parameters. The detailed procedure of these health features is expounded below [\[25\]](#).

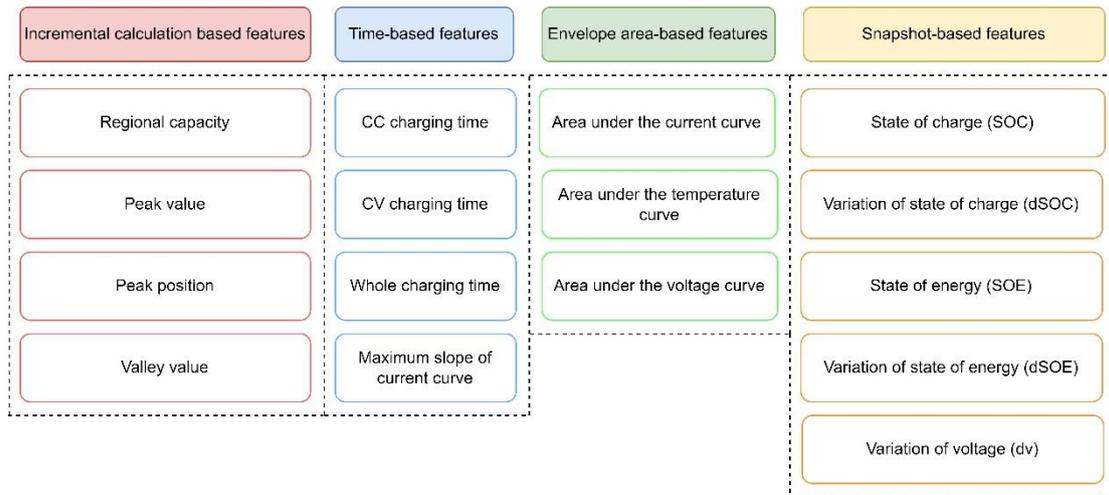


Fig.34 Feature summary

### 3.3.1 Incremental calculation-based features

As mentioned above, IC curve is an effective tool to measure battery capacity loss. As the battery aging, total capacity, peak value, peak value position and valley value will change regarding to the SOH reduction. These variations can be seen as indicators for SOH changing.

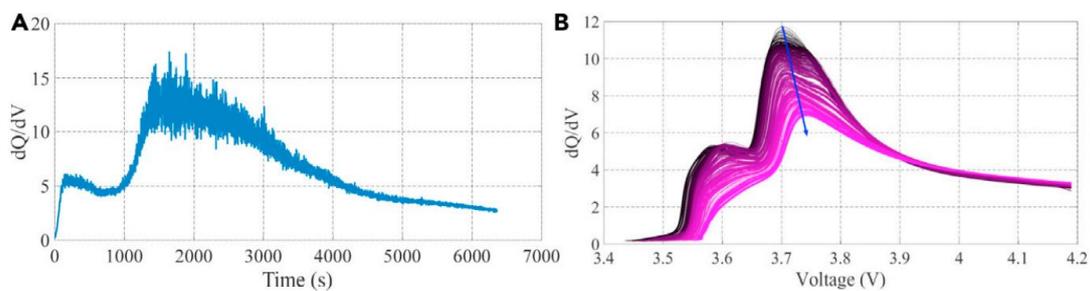


Fig.35 IC feature changing regarding to SOH reduction

But this approach needs to wait to a certain time point to let the peak value appears. This means that it cannot be used for real time application. Considering this drawback this kind of features are not selected for the thesis.

### 3.3.2 Time-based and envelope-based features

It has the same drawback as IC-based features. To get the features it needs to wait for CC and CV charging phase finishing. This limitation makes it cannot be used for real time application methods. Same situation for envelop-based features, it cannot be used.

### 3.3.3 Snapshot-based features

This features extraction method is based on a snapshot approach: move sliding window. This extraction approach can periodically generate the features which is a well match with real time application requirements.

Move sliding window can real-time trace battery states without waiting for specific features to appear (peak value, peak position etc.)

Here I would like to introduce the move sliding window working mechanism.

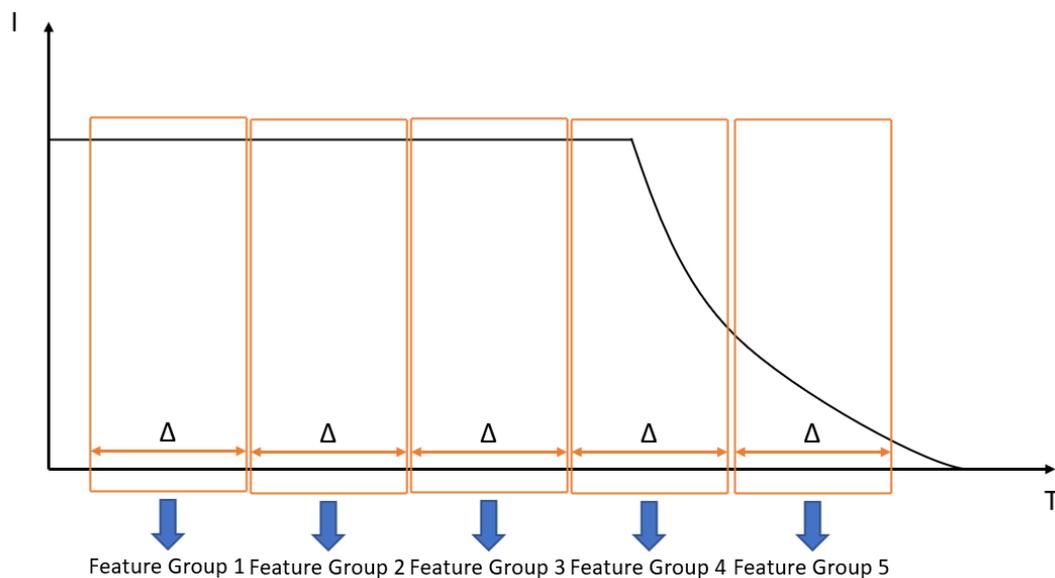


Fig.36 Move sliding window mechanism

The working principle is illustrated in figure.36 using the current-time graph as an example. To apply this approach a window width (delta) should be firstly defined. Considering the data length and training time this value is set to 120 sample points (60 seconds). The buffer size is a compromised chosen to minimize the noise and error of raw d When the window covers a part of data, calculate the corresponding feature

among this window (Feature group 1). Then the window slides to the next position and extracts the feature group 2. With this algorithm the following feature group can be also estimated.

Available features are SOC, dSOC, SOE, dSOE, dV. These features are the parametric variables of the cell that show the strong correlation with cell degradation. They can be summarized in table 6

Features	Unit
$\Delta V$	[V]
SOC	[%]
$\Delta SOC$	[%]
SOE	[Wh]
$\Delta SOE$	[Wh]

Table 6 Training features

These features can be computed from MATLAB script. It will introduce the calculation method and code script in detail.

### 3.3.3.1 SOC and dSOC estimation

SOC is an important parameter. Many variables and other parameters are dependent on it. The definition of SOC is the percentage of charge in a cell with respect to the maximum cell capacity. SOH is relevant to SOC and without knowing SOH, it is impossible to know the true value of SOC since SOH reduction influences the maximum capacity of the cell. The estimation of SOC is done by ‘Ampere-hour integral method’. Eq 18 show the calculation method.

$$SOC(t) = SOC(t_0) + \frac{\int_{t_0}^t i(t)dt}{C_{max}} \quad (18)$$

To calculate SOC, it is necessary to estimate Cmax which is the maximum capacity of the battery for a certain cycle. It should be updated at each cycle since SOH is continuously reducing. By updating Cmax, it can guarantee that SOC is always between the range of 0%-100% but the slop changes. this value is also computed by using Ampere-hour integral method. Eq 19 shows this process:

$$C_{max} = \int_0^{t_{end}} i(t) dt \quad (19)$$

The results are:

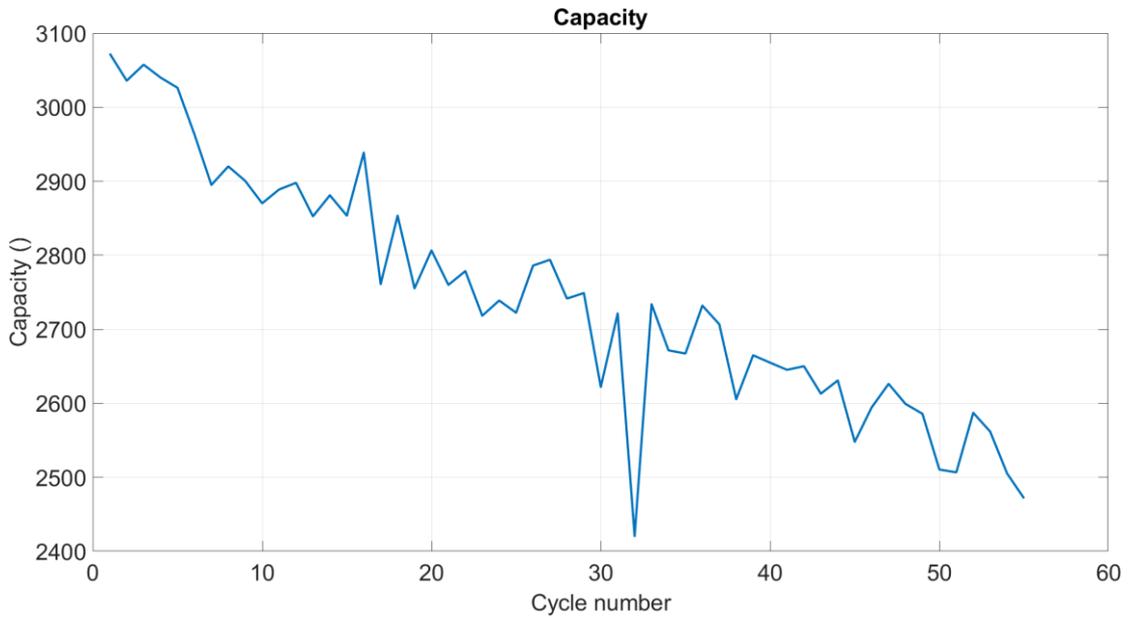
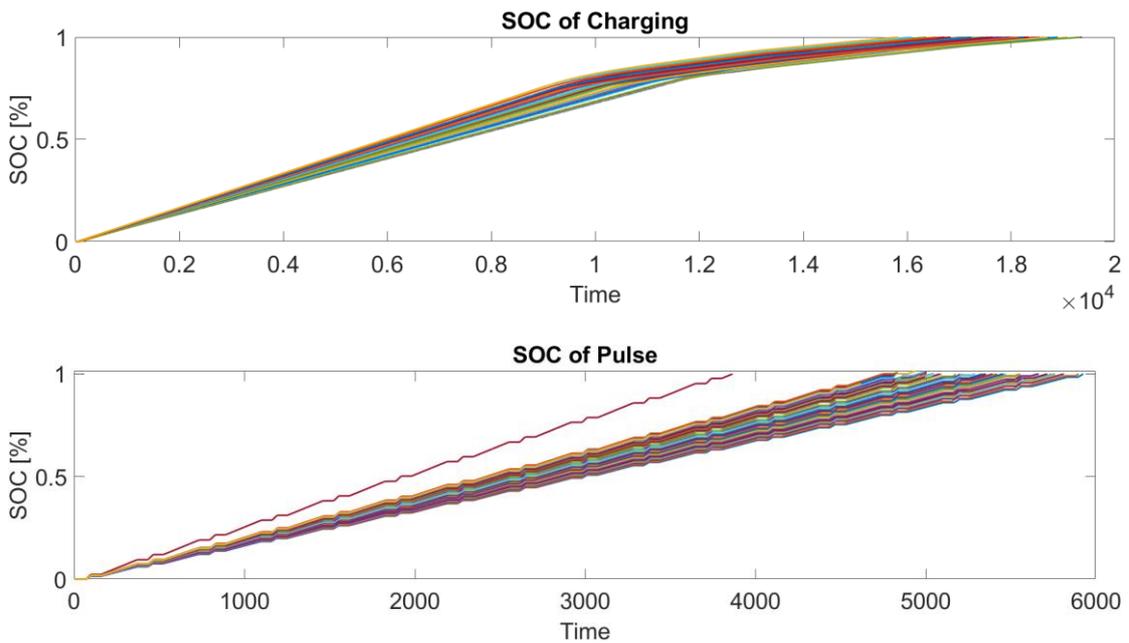


Fig.37 Maximum capacity for each cycle

And dSOC can be calculated as

$$dSOC = SOC(t) - SOC(t_0) \quad (20)$$

Figure.38 and figure.39 shows the SOC and dSOC profile.



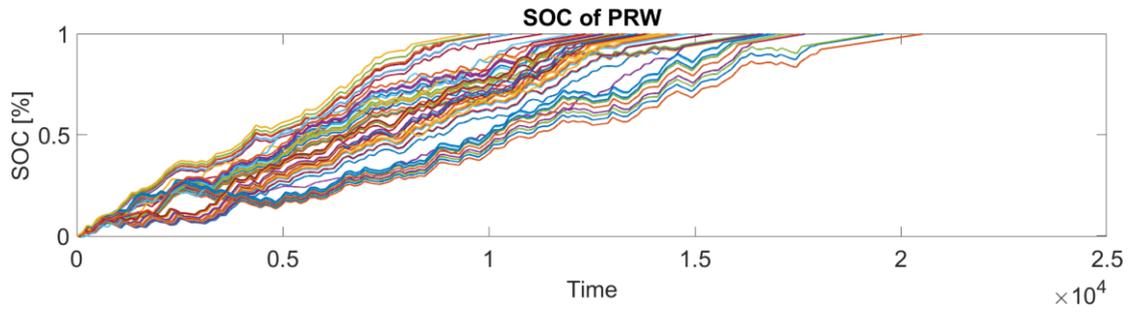


Fig.38 SOC profile for each phase

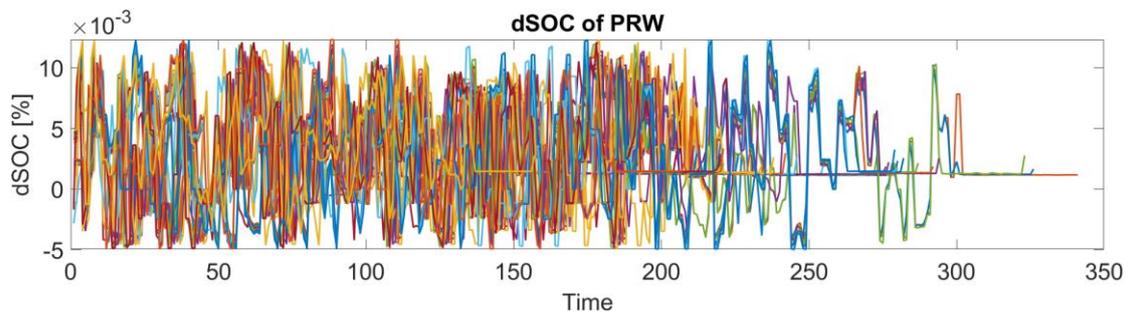
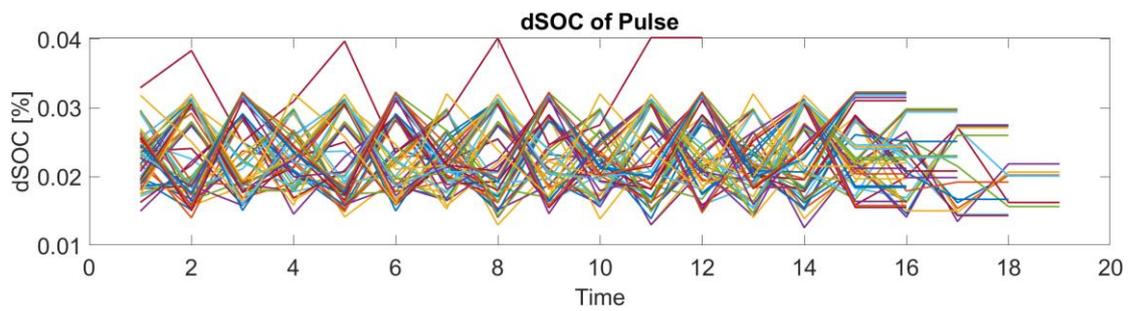
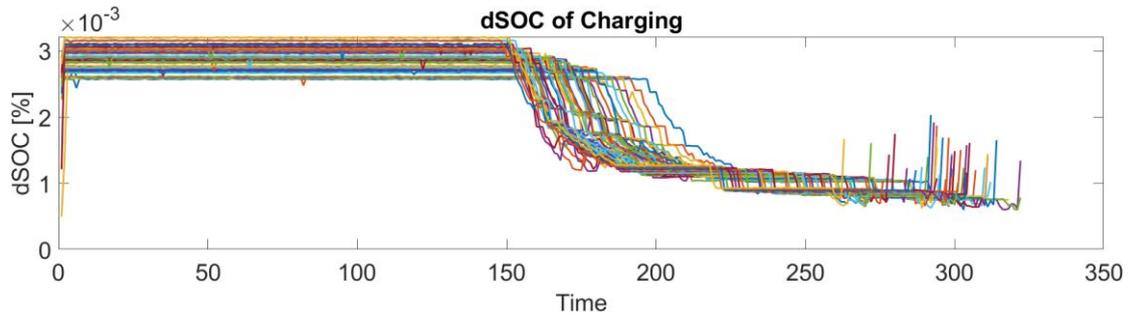


Fig.38 dSOC for each profile

### 3.3.3.2 SOE and dSOE estimation

As figure.38 shows above, at the upper and lower extreme, the dSOC curve has a large nonlinearity which can influence the network training.

To minimize this effect, it is necessary that introduce energy state to correlate. State of energy (SOE) is the integral of voltage and current over time, which can be defined by eq 21

$$SOE(t) = \int_{t_0}^t v(t)i(t)dt \quad (21)$$

And dSOE can be computed as:

$$dSOE(t) = SOE(t) - SOE(t_0) \quad (22)$$

The results are as follow:

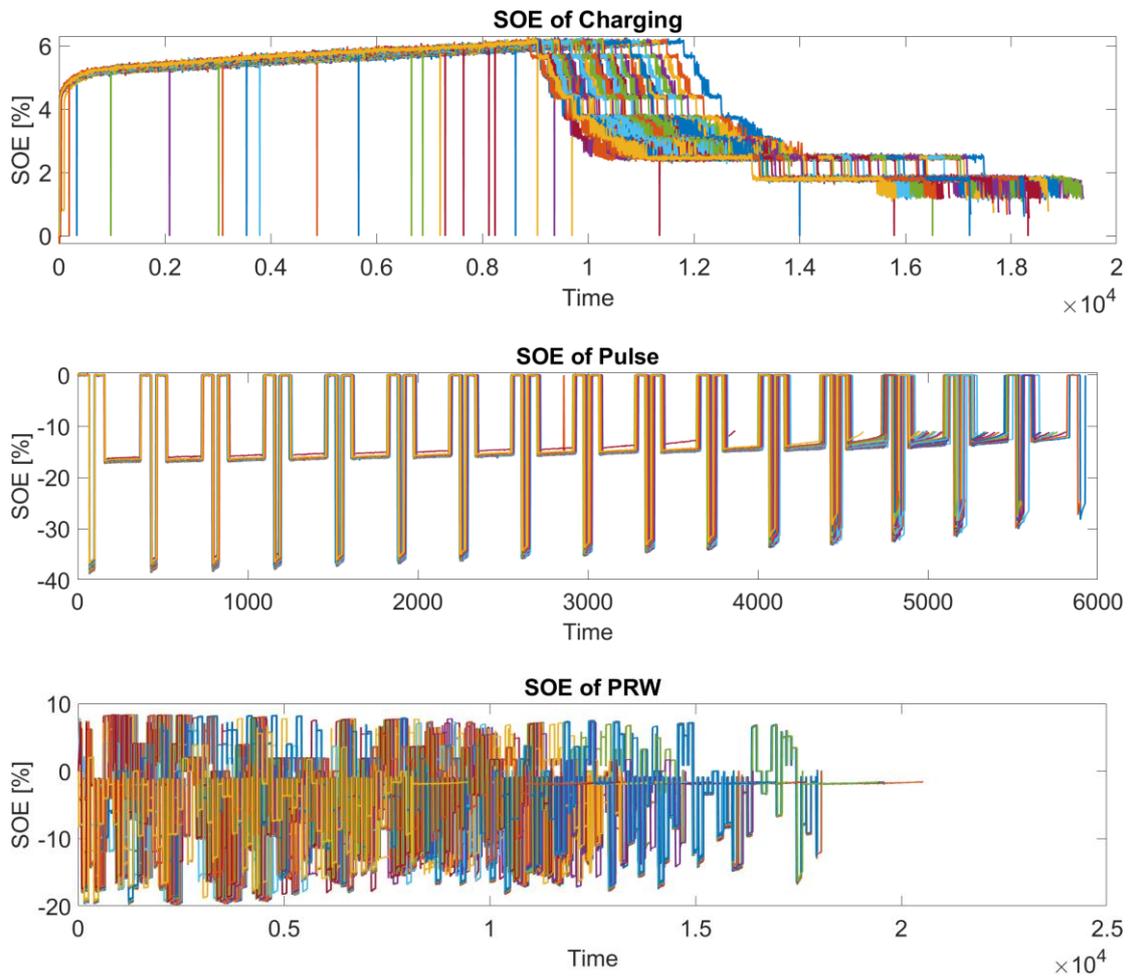


Fig.39 SOE profile for different phase

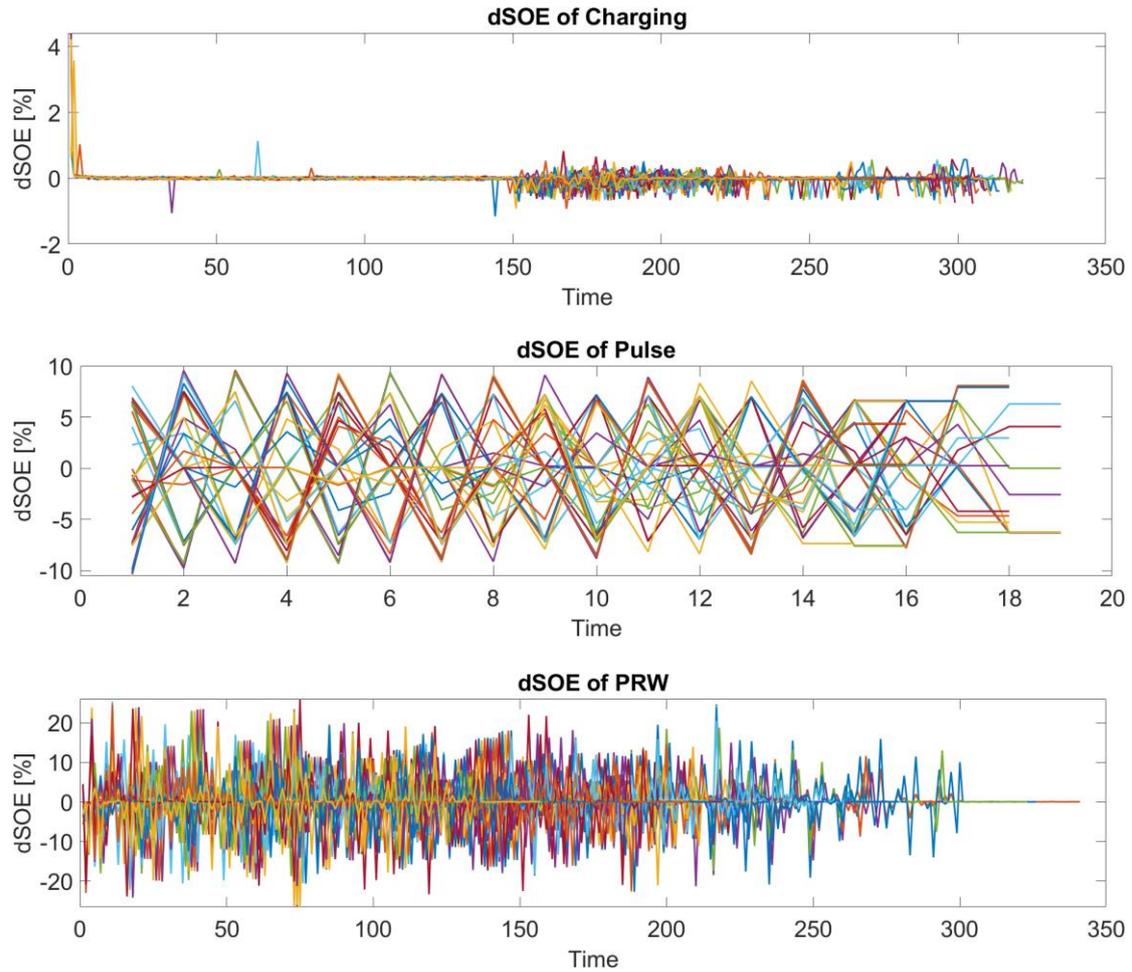


Fig.40 dSOE profile for different phase

### 3.3.3.3 $dV$ estimation

From electric circuit point of view, the terminal voltage is a summary of three components: open-circuit voltage ( $V_{oc}$ ) which is a equilibrium voltage and it's a function of SOC; Polarization voltage ( $V_p$ ) which is the representation of voltage dynamics and ohmic voltage ( $V_{ohm}$ ) which implies the reduction of energy state due to the capacity degradation. Eq 23 shows this relationship:

$$V(t) = V_{oc}(SOC(t)) + V_p(t) + V_{ohm}(t) \quad (23)$$

Theoretically, in lithium-ion battery, terminal voltage drops quicker following the reduction of SOH. This also cause a faster charge and discharge speed as cell ages.

The computation results are as follow:

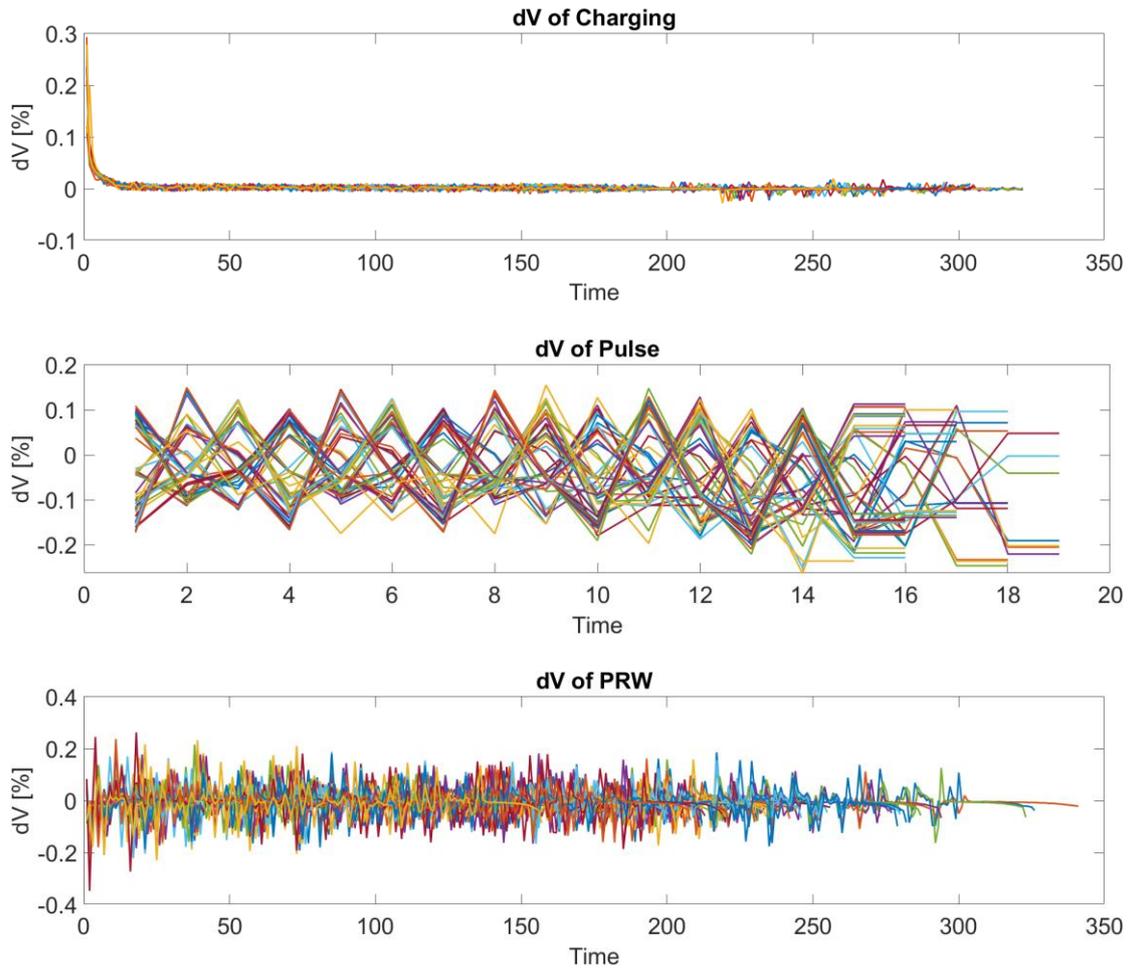


Fig.41 dV profile of different phase

### 3.4 Feature standardization

Standardization of training data is a very important post-processing step. As shown in table 6, between different features, their values are very different, and their units are also different. If these data are directly put into the network for training, the features with large numbers will have a greater impact on the weight matrix of the network, and vice versa. This will cause the network to not treat each feature equally, thus resulting the training failure.

To solve this problem, the data must be standardized. In this thesis, the data is standardized by using the 'z-score' function.

### 3.4.1 Introduction of z-score

For a variable  $x$  with mean  $\mu$  and deviation  $\sigma$ , the z-score of a value  $x$  is [19]

$$z = \frac{(x - \mu)}{\sigma} \quad (24)$$

For sample data with mean  $X$  and standard deviation  $S$ , the z-score of a data point  $x$  is

$$z = \frac{(x - X)}{\sigma} \quad (25)$$

z-scores measure the distance of a data from the mean in terms of the standard deviation. This is so called ‘standardization’ of a data. The standardization data set has mean 0 and standard deviation 1, and retains the shape properties of the original data set (same skewness and kurtosis),

Since the operation of z-score is based on mean value and deviation, in order to obtain complete and continuous data with the same degree of shape scaling standardization can only be used for all feature data in the training set. This operation can be used only in offline training process. Thanks to z-score also has a parameterization function, which enables it to be used in real-time computing systems.

In the process of offline training, the mean value and deviation of all features can be saved in advance, and the MATLAB function module can be built in Simulink according to the eq 25 to realize real-time standardized operation. This part of the content will be detailed in the next chapters.

Figure.42 to figure.43 shows features after standardization process. It can be clearly observed that the value of them has been scaled into the same magnitude. This ensures that the network learns equally for all features.

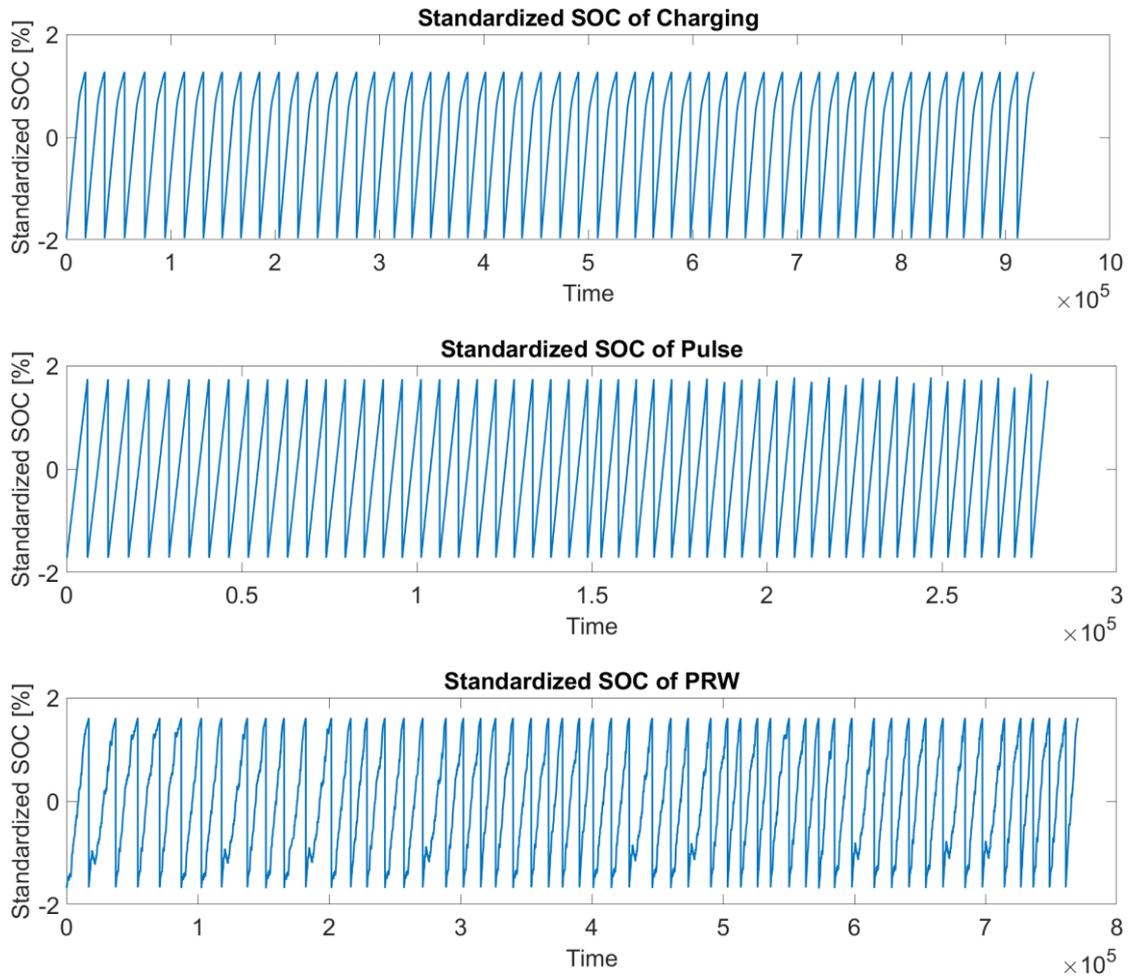
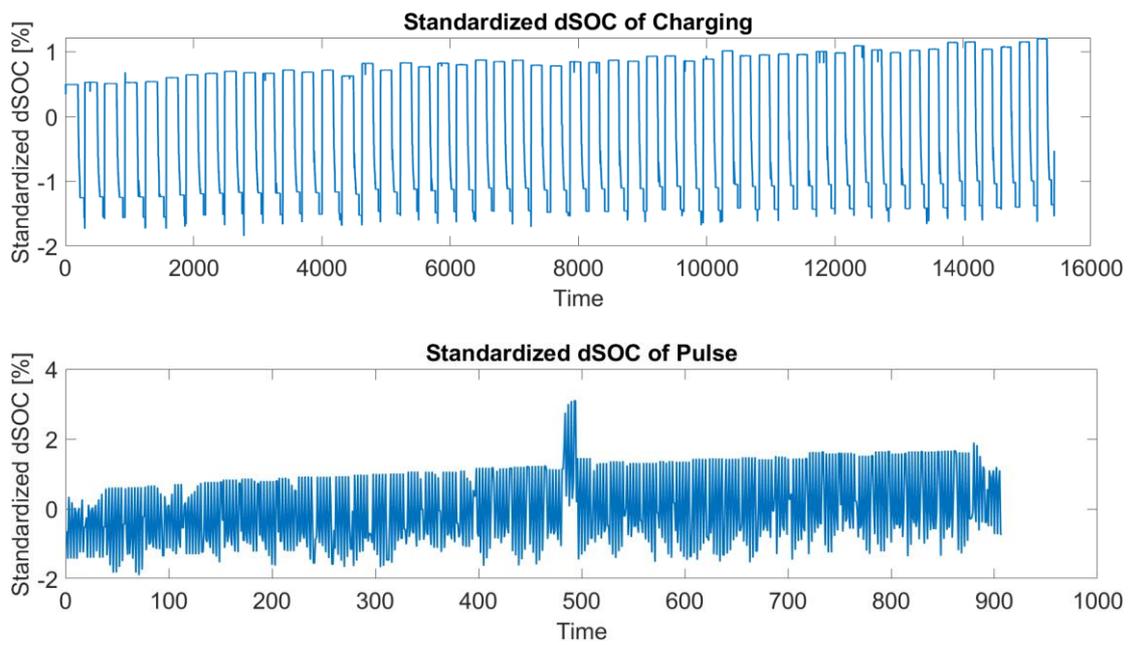


Fig.42 Standardized SOC feature



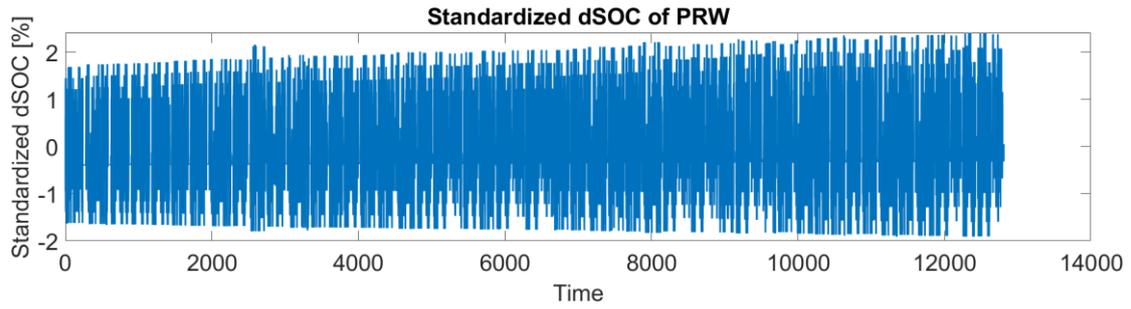


Fig.43 Standardized dSOC feature

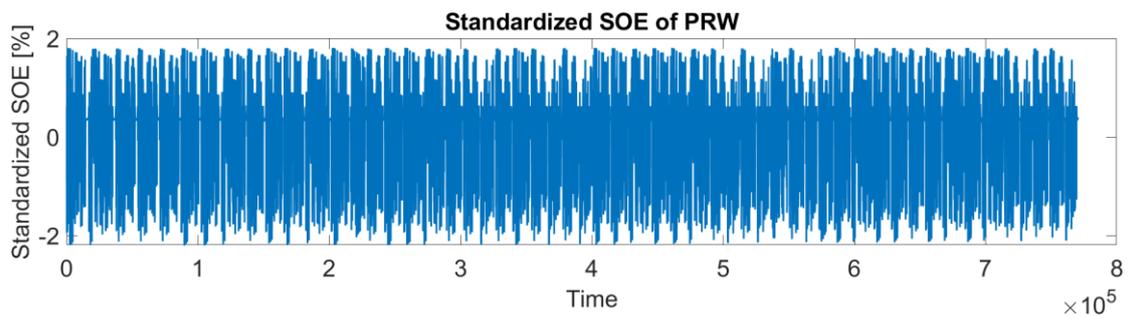
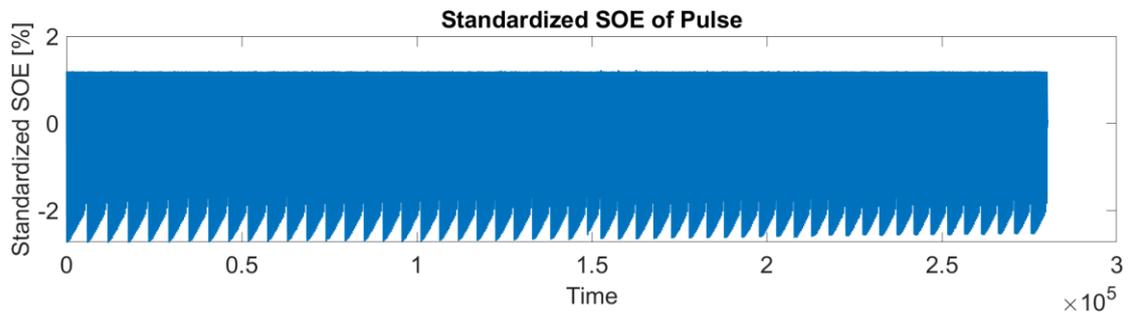
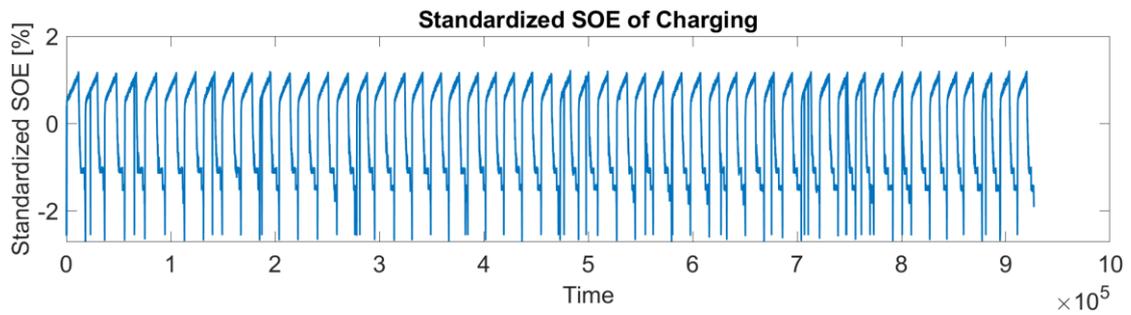


Fig.44 Standardized SOE feature

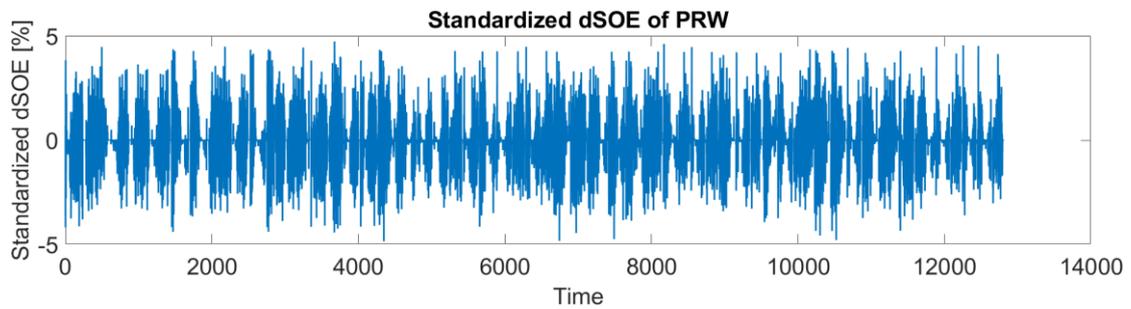
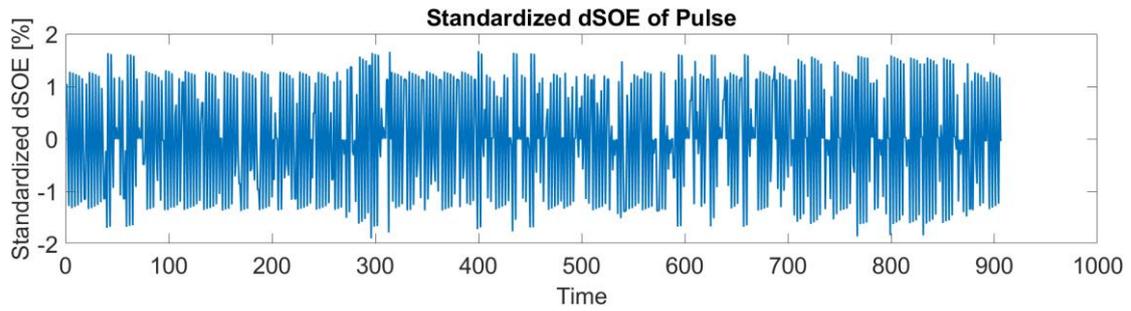
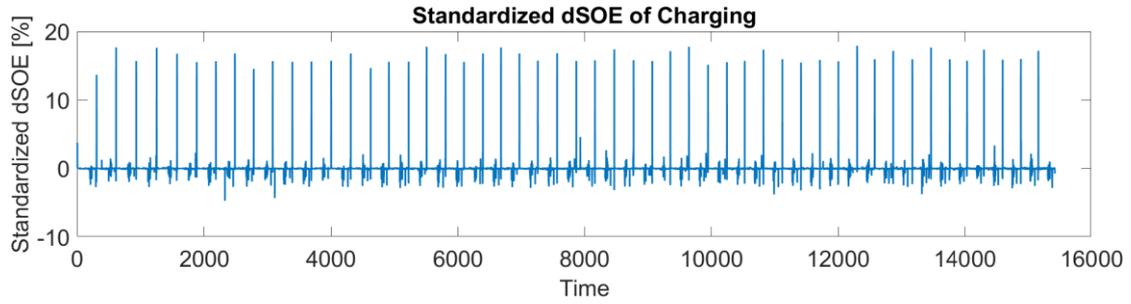
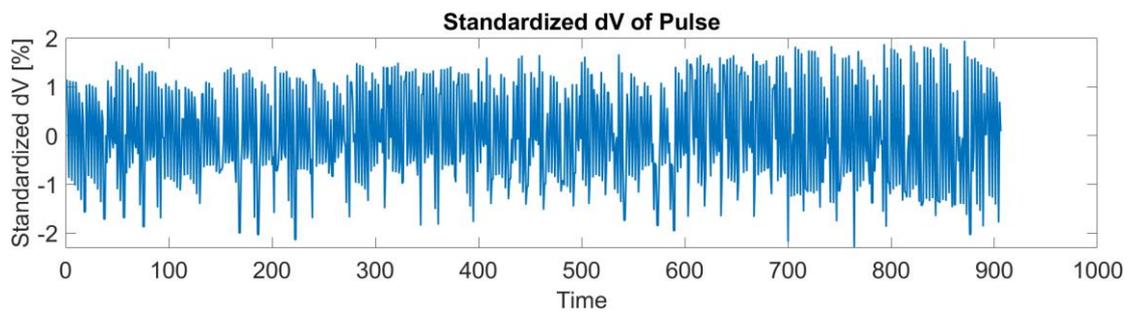
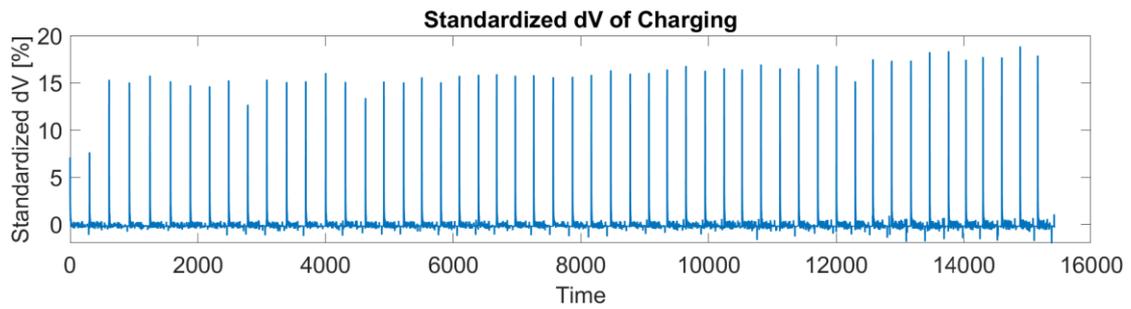


Fig.45 Standardized dSOE feature



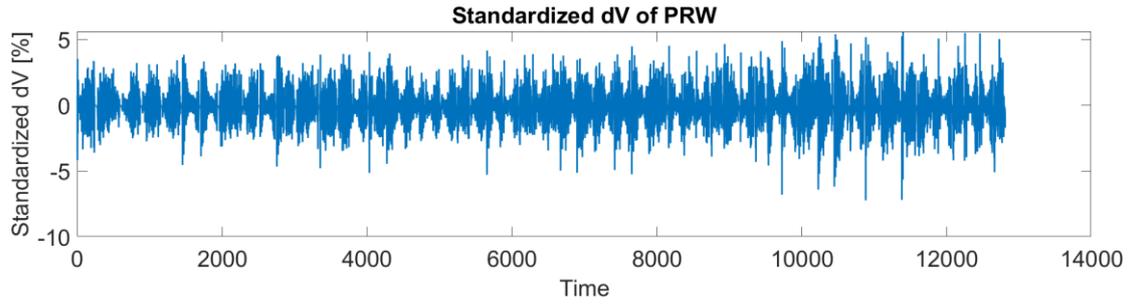


Fig.46 Standardized dV feature

### 3.5 SOH ground truth value estimation

To train the network, the SOH ground truth (label) is also necessary. This part will explain the estimation process of this value.

As it has been defined in section 1, the most critical step for SOH estimation is to calculate the maximum capacity for each cycle. This value has been computed in section 5.3, then following the eq 1 we can compute SOH ground truth value.

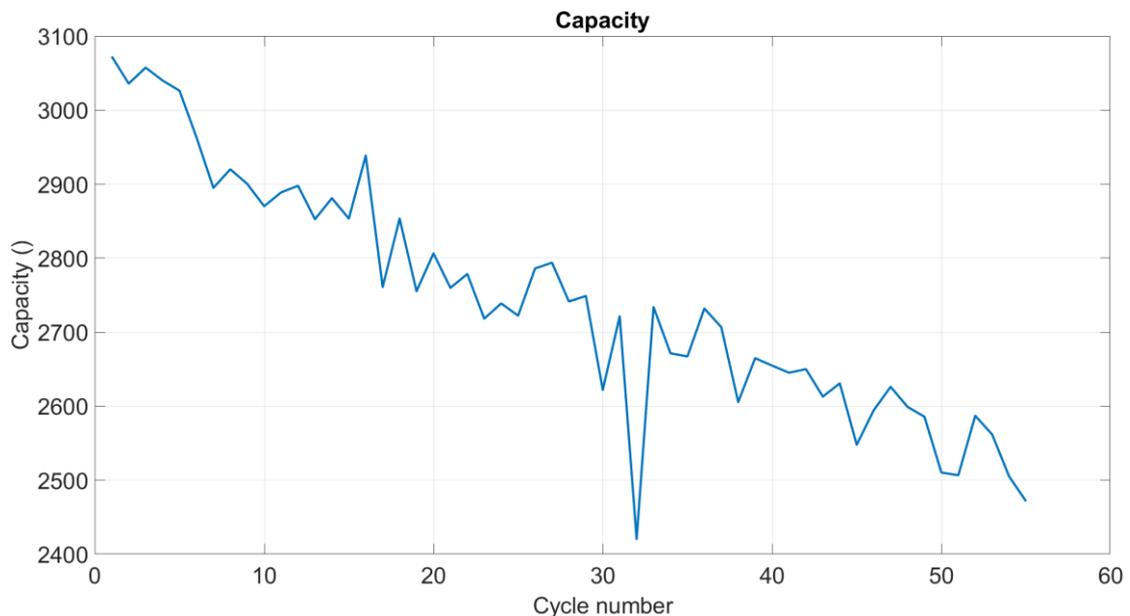


Fig.47 SOH ground truth value

However, due to the inevitable measurement error, it can be clearly seen that the original SOH data is not monotonically decreasing but has many fluctuations. As the reference value used for network training, the SOH must be accurate and eliminate all noise and errors as much as possible. Although the Butterworth filter used above can

filter the noise which is higher than a specific frequency, it cannot play the role of data continuity. Therefore, a high-order polynomial regression method is used to obtain a smoothed curve. The polynomial regression algorithm can be described by the following formula:

$$y(x, w) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{j=1}^M w_jx^j \quad (26)$$

Using MATLAB's curve fitting toolkit, it is easily to perform polynomial regression calculations to determine the value of the polynomial coefficient  $w_j$ . Table 7 shows these coefficient results:

Parameter	value
p1	0.006405
p2	-0.004941
p3	-0.04137
p4	0.02401
p5	0.089
p6	-0.03151
p7	-0.07823
p8	0.01324
p9	-0.0218
p10	0.9117

Table 7 Regression parameters

The regression curve is shown in figure.48 as follow:

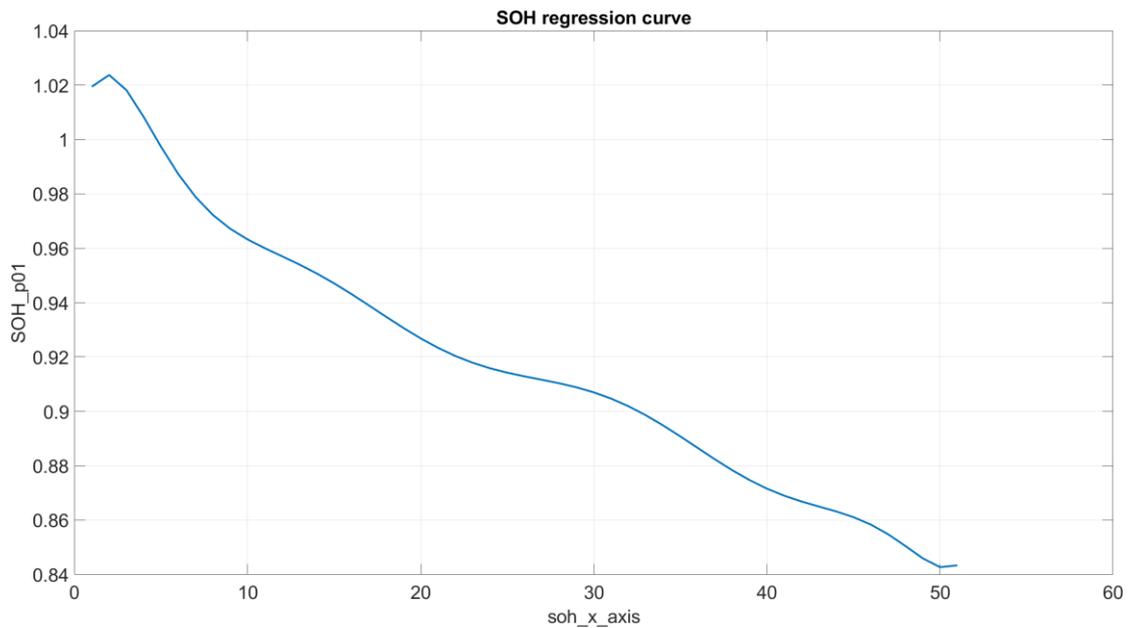


Fig.48 SOH regression curve

As pointed out in Section 5.2, in order to compromise the data structure, this thesis uses a specific phase to replace the aging effect caused by all phases. Considering that the dynamics of SOH is actually very low, in a phase, the change of SOH is almost null. Although this thesis uses a compromise strategy, in order to be more realistic, the invariance of SOH during a single phase will be preserved. Therefore, the SOH curve must also be stepped:

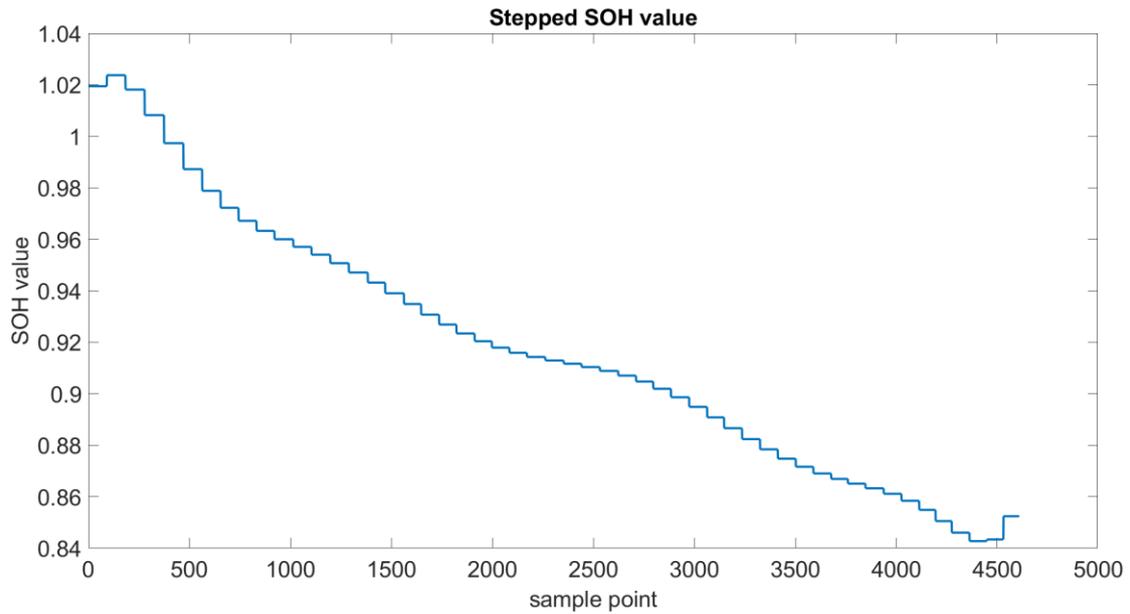


Fig.49 Stepped SOH ground truth

# Chapter 4. Network structure and training results

## 4.1 LSTM network structure and training parameters

As shown in figure.50, the network structure is as follows. First, the input features are reshaped into cells, which are directly input into the LSTM layer, and then go through the fully connected layer and the regression layer to get the SOH result.

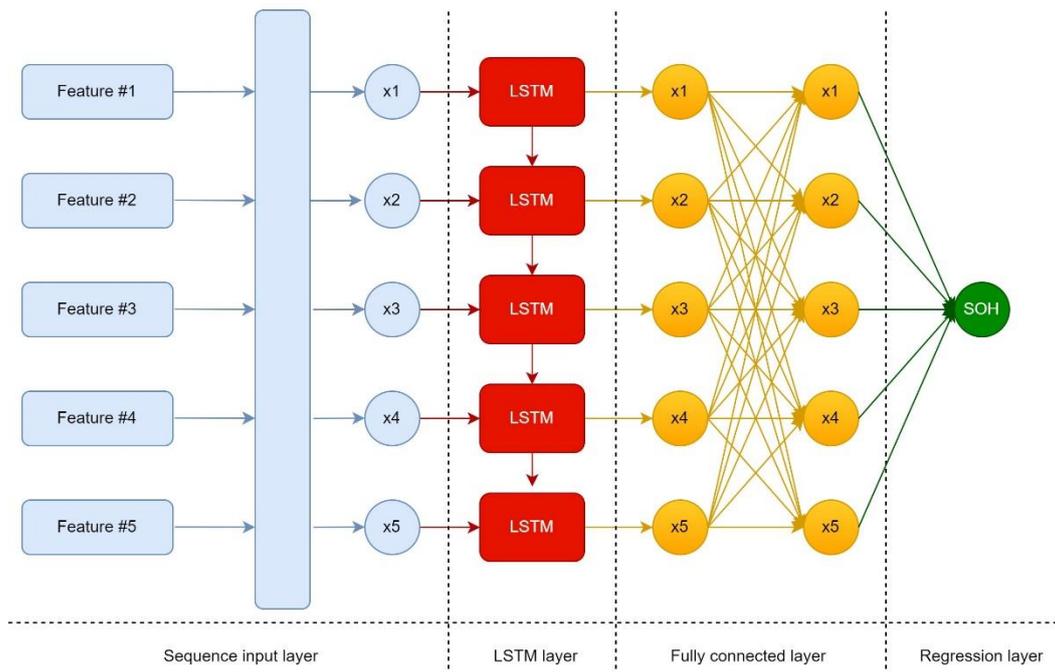


Fig.50 LSTM network structure

The definition of the entire network and training process is done using Matlab's deep learning network toolbox. To better explain the network structure, here is the corresponding code.

```
numFeatures = 5; %Number of features
numHiddenUnits = 10;
numResponses = 1; %Number of outputs, 1 SOH value.
maxEpochs = 5000;
miniBatchSize = 100;
```

```

LSTM = [ ...
    sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits,'OutputMode','last')
    fullyConnectedLayer(numResponses)
    regressionLayer];
options = trainingOptions('adam', ...
    'ExecutionEnvironment','cpu', ...
    'MaxEpochs',maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'GradientThreshold',1, ...
    'Verbose',false, ...
    'Plots','training-progress');
%%Training
net = trainNetwork(XTrain,YTrain,LSTM,options);
delete(findall(0))

```

The training parameters setting are as follow:

numFeatures	Number of features used for training.
numHiddenUnits	Number of LSTM layers.
numResponse	Number of estimation value (one in this case).
maxEpochs	Number of epochs for entire training.
miniBatchSize	Number of mini batch size for gradient descent process.

Table 8 Training parameter explanation

The training process uses five different features, so ‘numFeatures’ is set to 5. The number of LSTM layers can be set differently. The more layers, the longer the training time and the stronger the learning ability, but the more prone to overfitting. The maximum number of epochs is the total training amount. The minimum batch size defines the amount of data processed by the gradient descent algorithm each time. The

larger this parameter is, the larger the amount of data involved in the gradient descent each time, and the slower the gradient descent operation will be.

These parameters were randomly set within reasonable intervals, and the entire training process took several weeks to find the best training results.

Figure.51 shows the window of the training process.

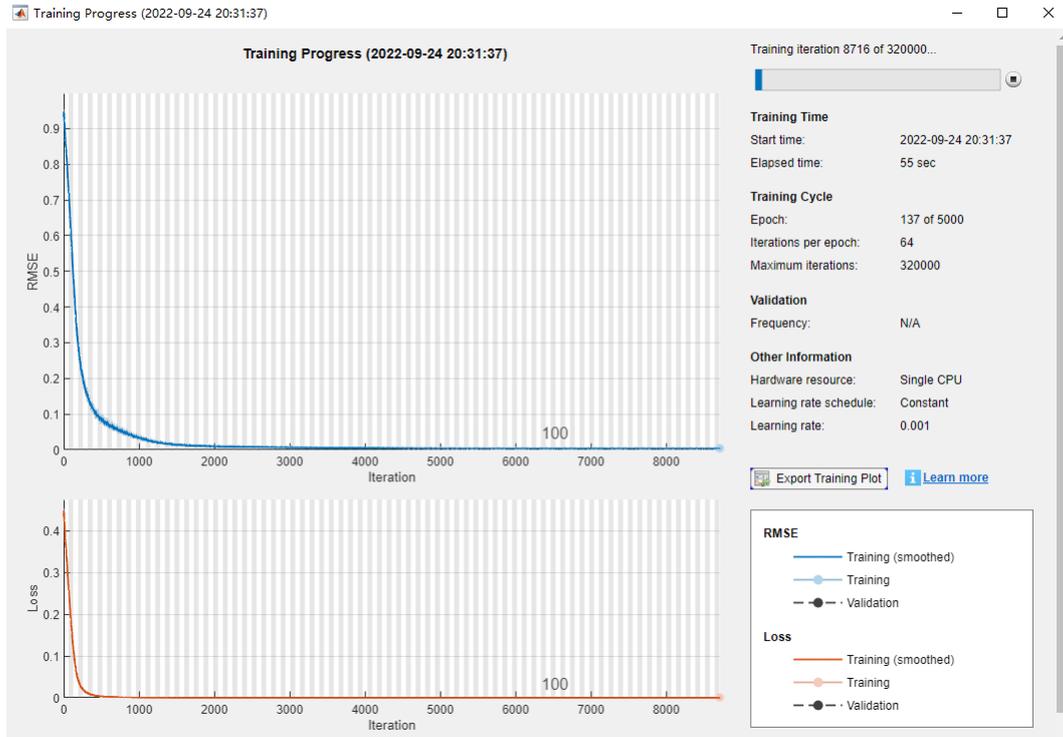


Fig.51 Training process

## 4.2 Training Results

Training results for charging, pulse and prw phase are as follow:

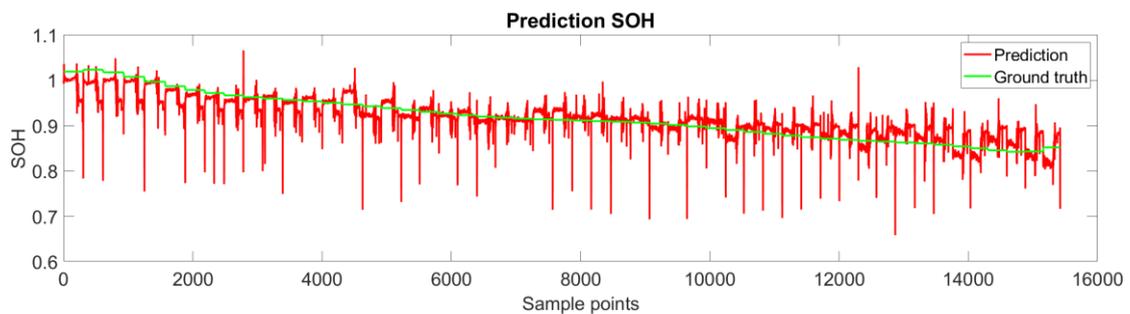


Fig.52 Charging phase result

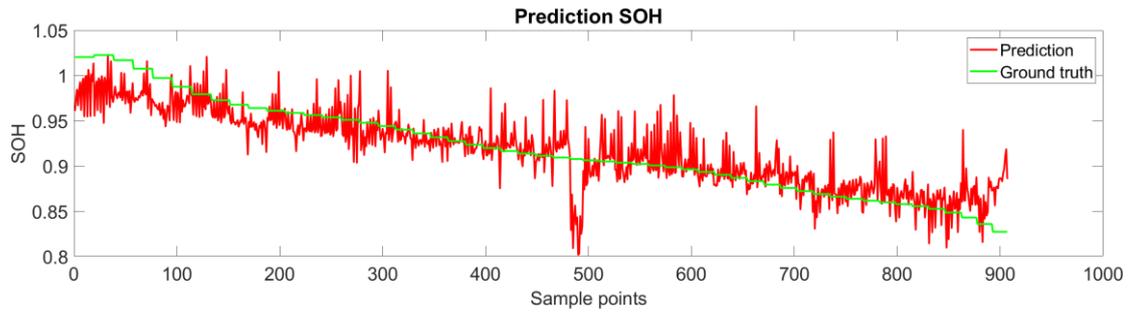


Fig.53 Pulse phase result

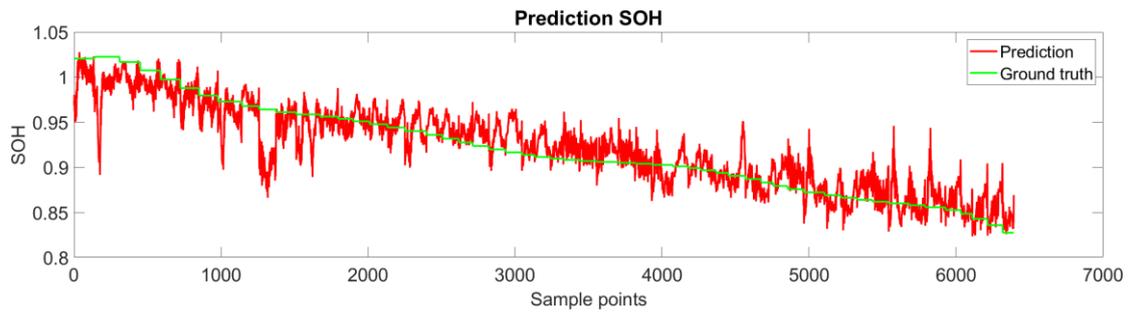


Fig.54 PRW phase result

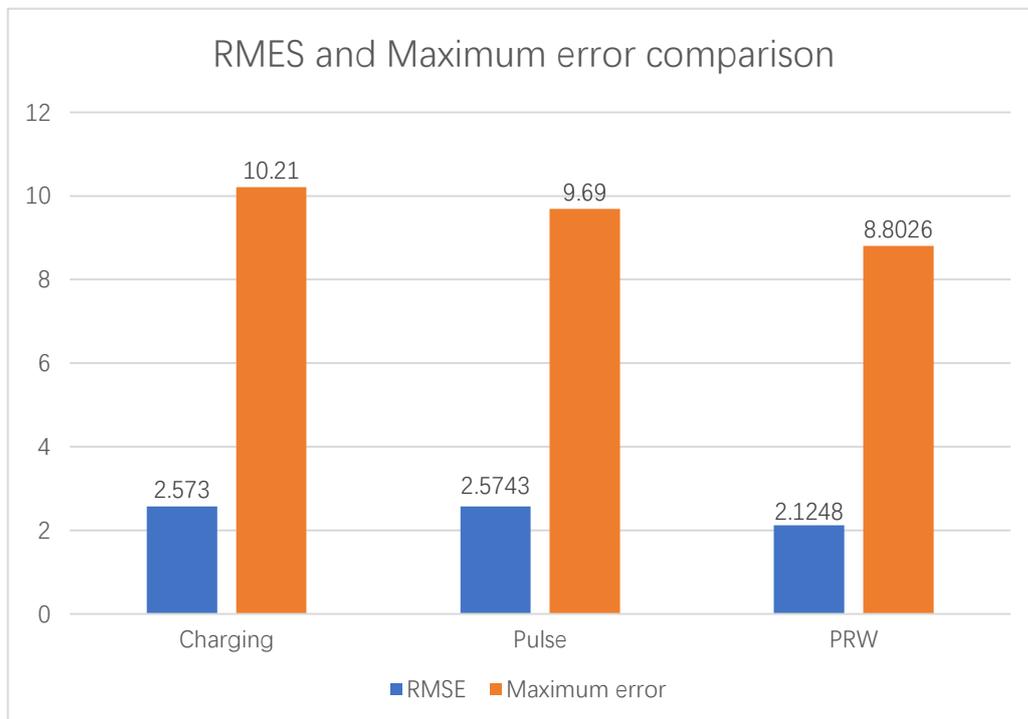


Table 9 RMSE and Maximum error comparison

It should be noted that for the charging cycle, since the charging strategy changes from CC (constant current) to CV (constant voltage) when the SOC is about 65%, the current-

voltage curve changes abruptly here, which has a huge impact on the training of the network. Therefore, the charging cycle should have the best training result, but it was affected by this factor, it performed poorly. Considering what was mentioned in 5.5, the SOH should theoretically keep invariance within a single cycle. Therefore, it can be speculated that if only the CC stage is used for training, the most precise results to the SOH ground truth will be obtained. Figure.55 shows the result only using CC strategy.

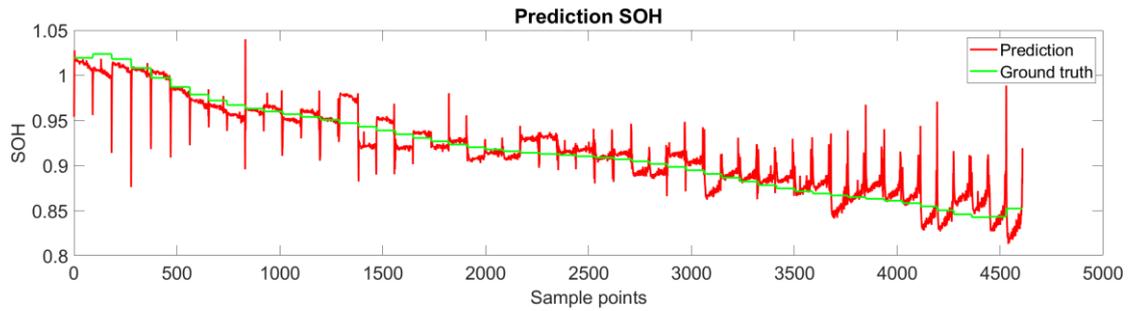


Fig.55 Charging phase only with CC strategy

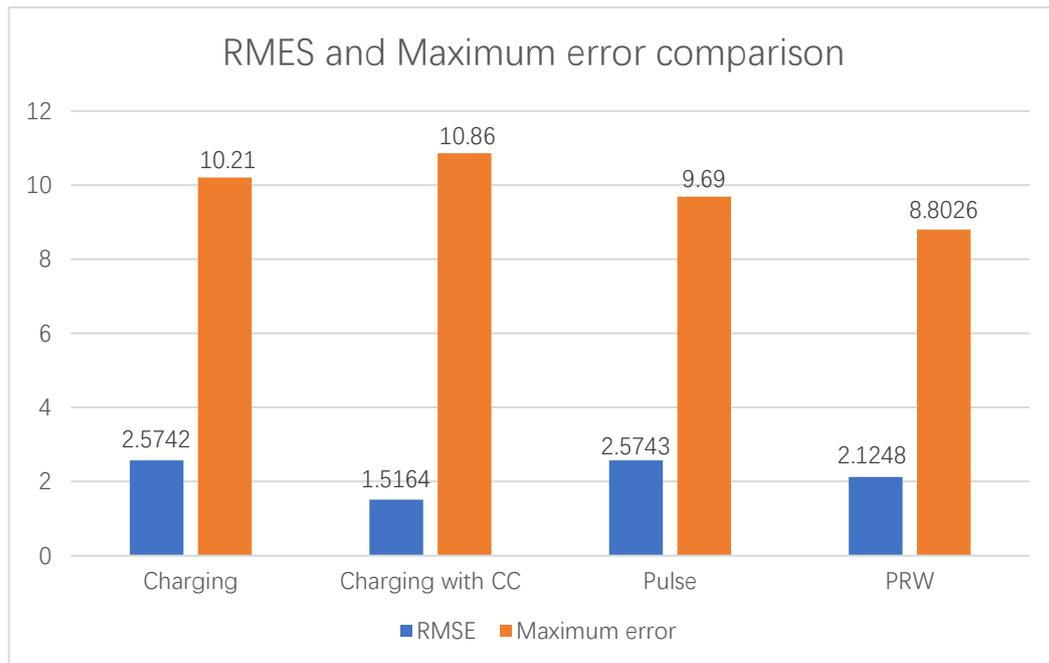


Table 9 All phase comparison

# Chapter 5. Simulink modeling implementation

## 5.1 Simulink modeling overview

### 5.1.1 SOH estimation model

As mentioned in the introduction, the final step of this thesis is to implement the modeling of the SOH prediction system in Simulink. In order to achieve this goal, it is necessary to reproduce the idea in Simulink and realize the real-time feature extraction function. For this reason, it is necessary to give the macro structure of the model here, and to elaborate on the module structure and specific functions in the following chapters

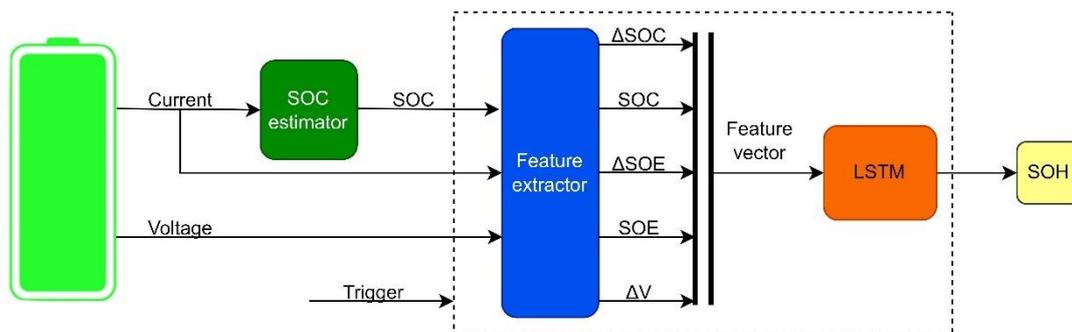


Fig.56 Simulink model flowchart for SOH only

As the figure.56 shows. The raw data is first collected from the battery, and the SOC calculator uses the current data to calculate the SOC value and feed it into the feature extractor. The feature extractor will extract five features in real time and feed them into the network for SOH estimation.

It should be noted that the feature extractor and SOH estimator do not work all the way but are controlled by a trigger mechanism. This is because a portion of the curve features would seriously interfere with the work of the SOH predictor. This part of the content will be detailed in the SOH and SOC combined network section.

### 5.1.2 Combined model with SOC and SOH estimators

Figure.57 shows the co-simulation model of SOH and SOC. In this model, the SOH and SOC computing modules interact with each other. The calculation results are mutual input signals.

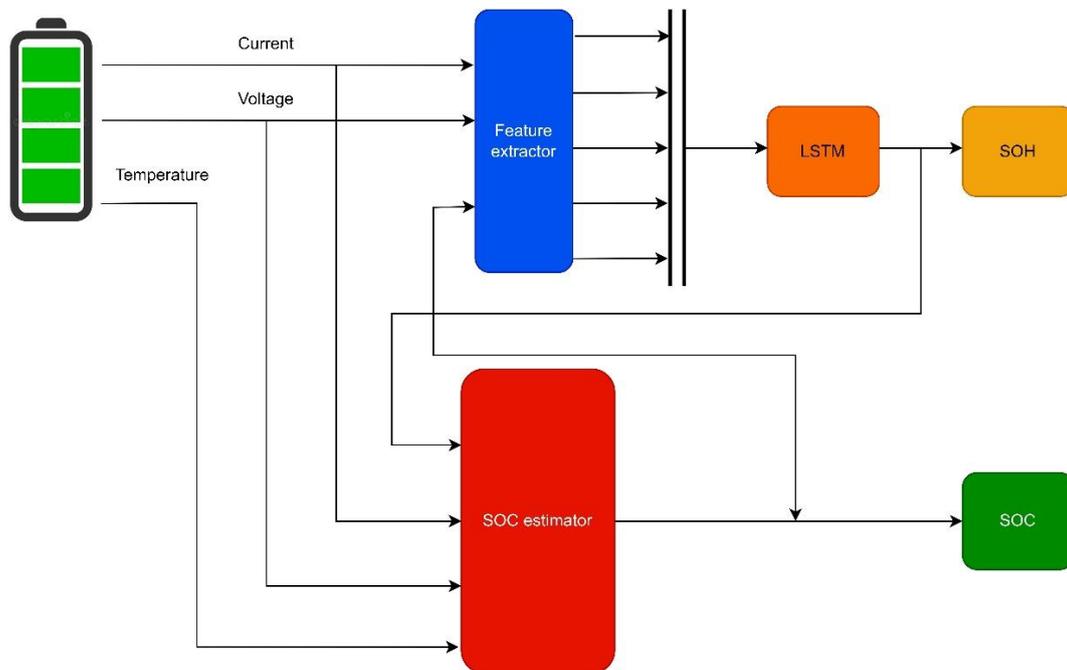


Fig.57 Combined network

Since there are multiple computing modules in the joint network, the computing efficiency will be greatly affected. Therefore, an additional trigger mechanism is designed in the module, and the corresponding module is triggered only when the calculation is required to improve the operation speed.

## 5.2 Simulink model of SOH estimator

After successfully training the network, this chapter will build the Simulink model. The model completes the whole process from raw data processing to SOH calculation.

The figure.58 shows the whole picture of the model. This article will explain the function of each module one by one.

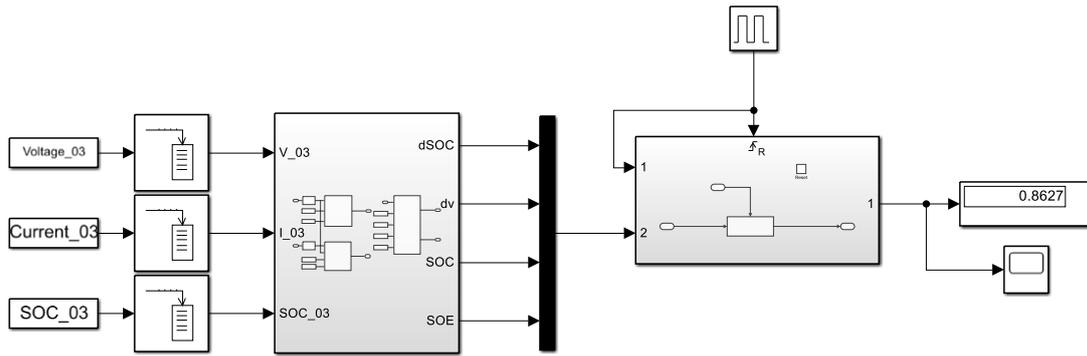


Fig.58 Overview of Simulink model

Start from left side blocks. Voltage, current and SOC are three blocks of raw data. Although in training process SOC is a calculated feature, considering the combined network in next step, in this model SOC is treated as a known feature. Three buffers are placed after raw data block. The purpose of these blocks is to implement the move sliding window approach. Due to the intrinsic characteristic of the buffer blocks, they will only output these data after the internal space is filled, which perfectly match the characteristic of the move sliding window, so the buffer size is set to 120 data points, which is consistent with the window width setting during training phase.

Next, the stored data will be sent into the feature extraction subsystem. Fix.xx shows the structure of the subsystem.

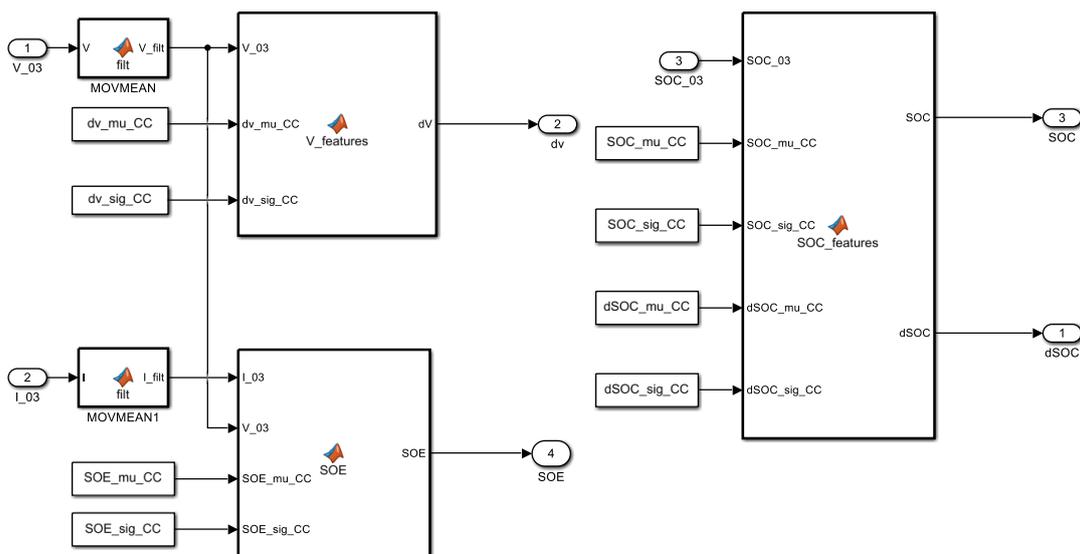


Fig.59 Internal structure of feature extractor subsystem

As in the training process, the current and voltage data are first fed into the ‘movmean’ function for noise reduction, and then fed into the corresponding MATLAB functions for feature extraction.

All extracted features will be formed into a feature vector through the ‘mux’ block. Note that the order of features in this vector must be the same as during training. According to the characteristics of the ‘mux’ block, the features should be connected to the ‘mux’ from top to bottom according to the arrangement order of the features in the feature vector during training.

The fused feature vector will be directly sent to the SOH module for calculation. For the SOH estimator works properly, a resettable and a triggered subsystems are used here.

Figure.61 shows the resettable subsystem. This subsystem is aimed for resetting the initial condition of the network before it estimate SOH, thus, to make it works properly.

It is triggered by a pulse generator. Figure.60 shows the pulse signal generated by it.

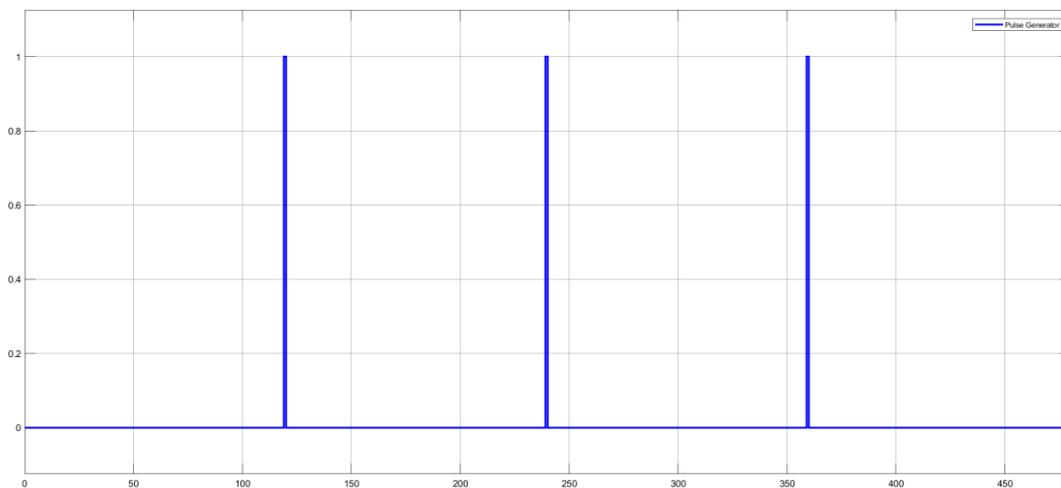


Fig.60 Pulse generator signal

Inside the resettable subsystem, the reset block is set with rising edge. This means, with such pulse signal, the subsystem is triggered to reset every 120 seconds.

Figure.62 shows the internal structure of resettable subsystem. Inside it, another triggered subsystem is placed. This subsystem uses the falling edge of the pulse signal as trigger.

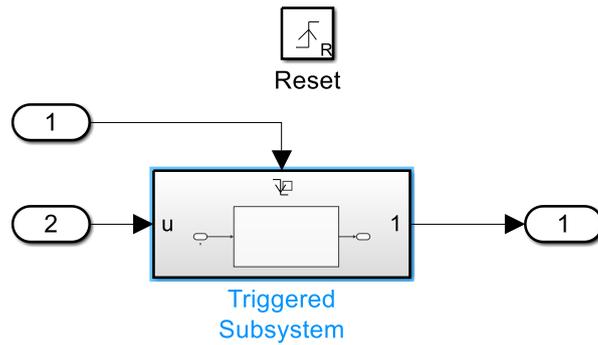


Fig.61 Internal structure of resettable subsystem

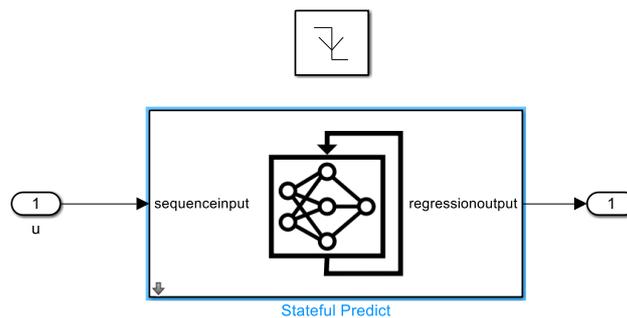


Fig.62 SOH estimator

To summarize the working principle of the two-layer subsystem: the calculation of SOH occurs at the 120th data point. A system reset is required before each SOH calculation. Therefore, according to the pulse signal, at the 119th data point, the rising edge appears, resettable subsystem makes the system reset itself, and then at the 120th data point, the falling edge appears, triggered subsystem calculates the SOH value. The pulse generator repeats this signal every 120 seconds, and the model following this signal periodically.

Different networks are selected inside 'stateful predict' block. By selecting charging, pulse and prw networks and feed the model with corresponding raw data, the model can estimate SOH for different phase.

### 5.3 SOC and SOH combined model.

The final goal of this thesis is to combine the SOC estimator for the computation of the two values.

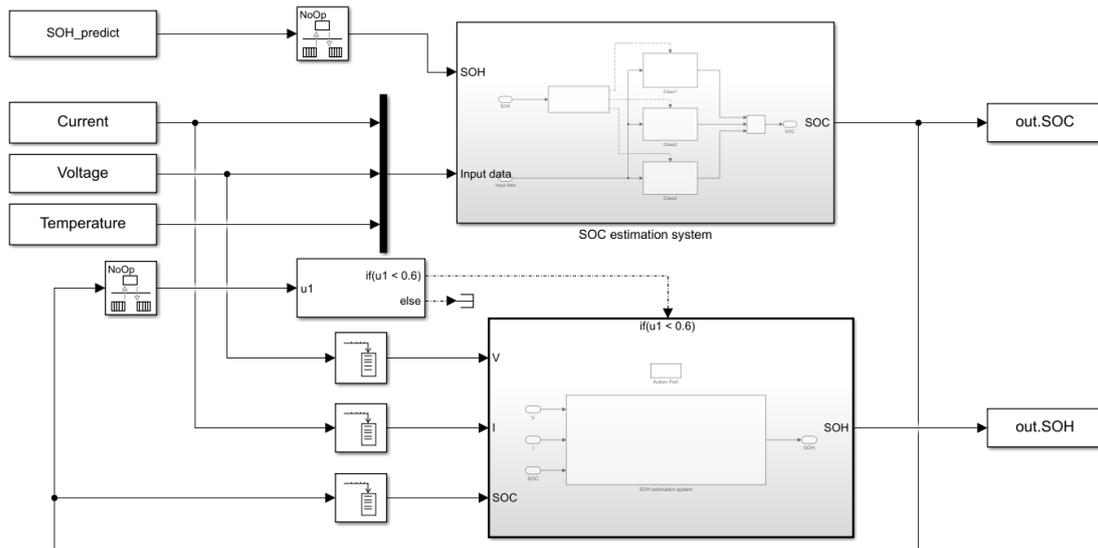


Fig.63 Combined model structure

The structure of the SOH part is the same as described in the previous section. The following will introduce the structure and function of the SOC part.

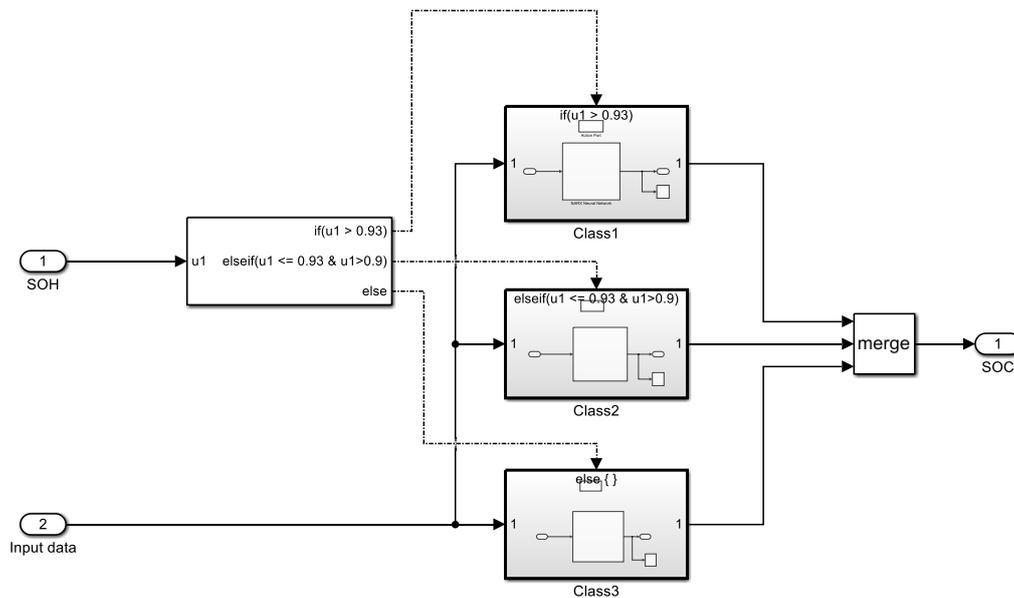


Fig.64 SOC estimator structure

SOC estimator is a combination of three different individual network. Each of them is trained in a certain range of SOH condition. Class1 is under  $SOH > 0.93$ ; Class2 is under  $0.93 > SOH > 0.9$  and class3 is under  $SOH < 0.9$ .

It has also designed 1 and 5class model as comparison. As figure.65 and figure.66 shows below

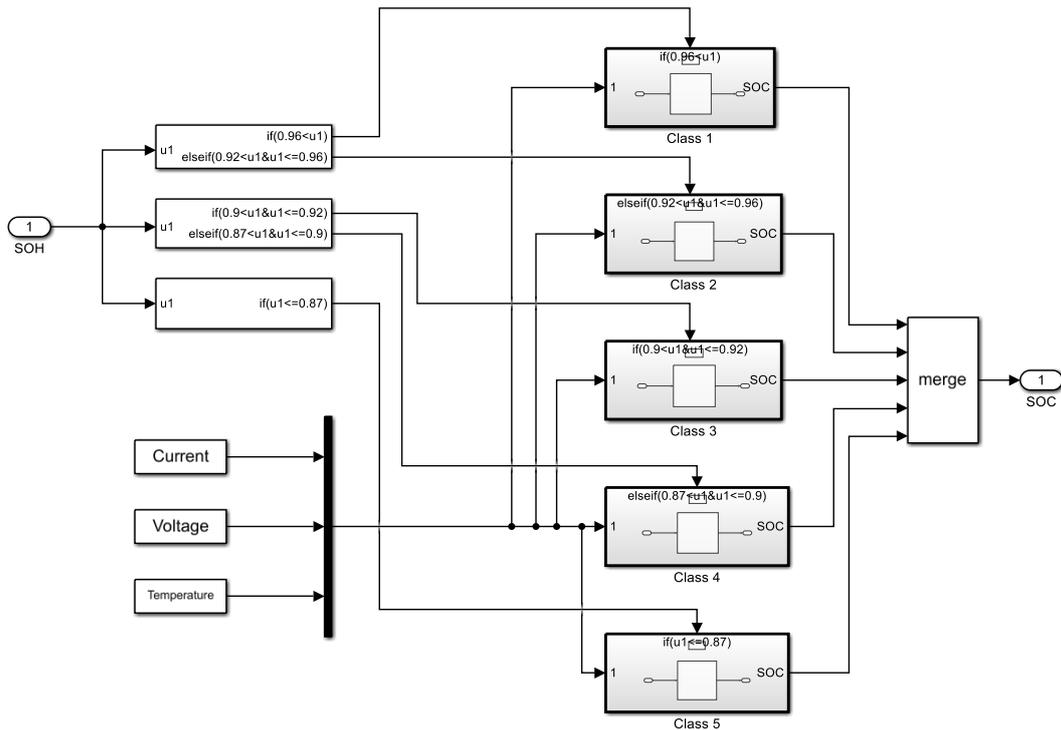


Fig.65 5 class model

In this structure SOH is not an input of SOC but a judgment signal. As shown in above, SOH value is sent into a 'if' block as judgment. SOC networks are placed in three 'if action subsystem'. A 'merge' block is used for SOC signal fusion.

Therefore, the estimated value of SOH is used as the judgment signal of the SOC estimator, and the value of the SOC estimator is used as the input feature of SOH. The two systems are combined and realize the joint simulation.

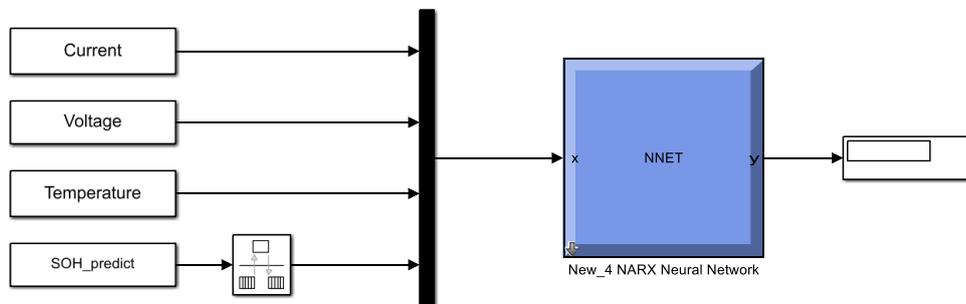


Fig.66 1 class model

1 The class model is different from other structures in that it uses SOH as an input feature rather than a judgment signal.

## 5.4 Model test scenario and results

### 5.4.1 Simulation scenario description

As mentioned above, the CC part of the charging phase will give the best SOH prediction results, and the PRW phase can best reflect the real working condition, so this thesis decided to reorganize the data set to realize the combination of charging and discharging. Figure.67 shows the new organization of dataset.

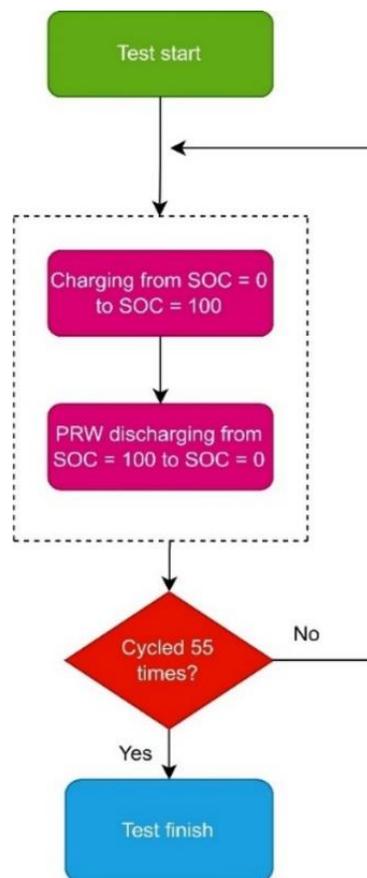


Fig.67 Combined model test cycle

Since the move sliding window used in the SOH part has a hysteresis of 120 data points, the SOC will assume  $SOH = 1$  for the first 120 data points and use this value for

calculations. When the first SOH value is calculated, it will be updated to the SOC estimator.

It must note that although the organization of the data is a continuous charge-discharge cycle, considering that there is no signal indicating the end of the charge-discharge process in the original data, this thesis uses an equivalent substitution method for training.

The process is, first 55 charge cycles are performed, and the model will get the SOC and SOH values for those cycles. Then 55 PRW discharges cycles are performed, and the SOH value used in those process is the value obtained in the corresponding charging process stored in advance. Although this method separates charge and discharge tests, its effect is completely equivalent to continuous charge and discharge. In the actual HIL test, since the signal of the end of the charging and discharging process can be obtained by using the CAN bus, the switch of the two networks can be easily defined.

The working logic of the scenario is shown in figure.68

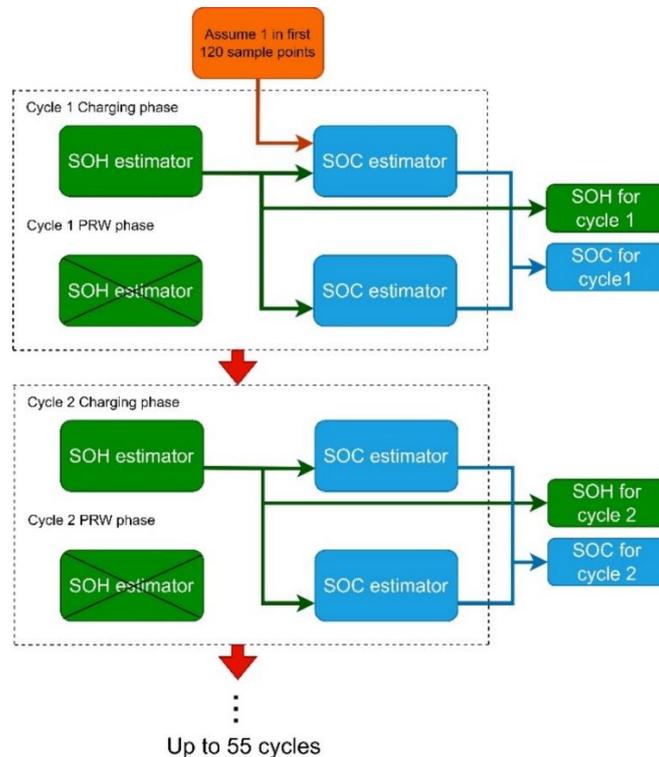


Fig.68 Combined model working logic

Another technique used here is so called "SOH interval extremes overlap". This technique is to solve the sensitive increasing around the SOH interval extremes.

Figure.69 shows this mechanism. Note that this is an exaggerated figure. The judgment numbers are just examples, not real one used in model.

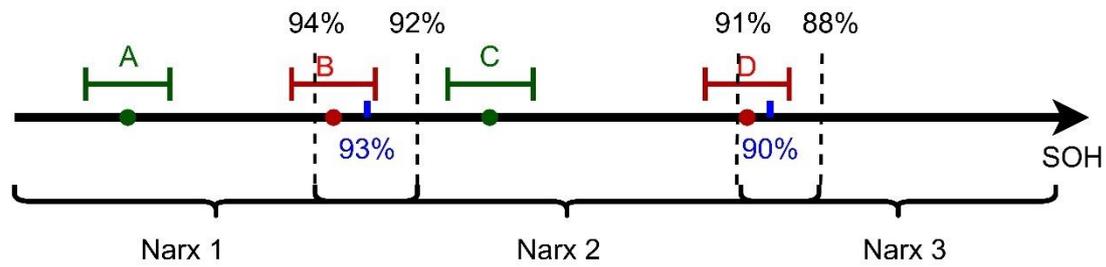


Fig.69 Explanation of training dataset overlapping.

93% and 87% SOH is switching

Narx1 is trained from 100% to 88% SOH.

Narx2 is trained from 90% to 83% SOH.

Narx3 is trained from 85% to 82% SOH.

The extremes of SOH interval are overlapped. This technique can solve the computation error due to network incorrect switching.

For example, A, B, C and D are four random SOH true value during the test. Each of them has a tolerance interval of error. It means that SOH could be any value during this interval. For A and C, the tolerances interval does not include 93% and 87% so there is no risk for incorrect switching of the network. However, with B and D, their tolerance intervals include switching point, and this can cause the switching of network, thus, it causes computation error if the extremes of training set do not overlap.

For example, B should be computed by Narx1, but the tolerance can cause the network switches from Narx1 to Narx2. Due to the overlapping, Narx2 has the capability to compute B's feature. This mechanism can solve the problem.

Figure.70 to figure.73 give the comparison for updated and original judgment condition.

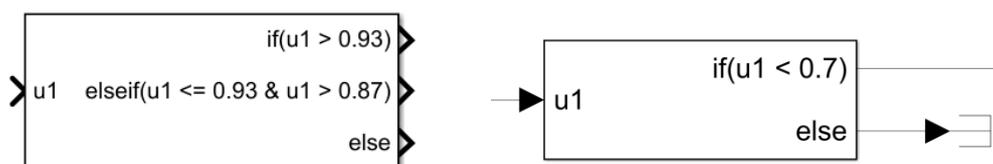


Fig.70 Original conditional judgment from 3-Class NARX combination option

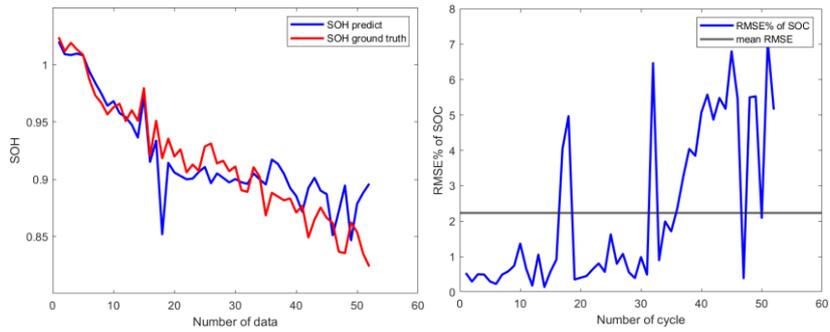


Fig.71 Original conditional judgment result SOH RMSE%=2.41% SOC mean RMSE%=2.23%

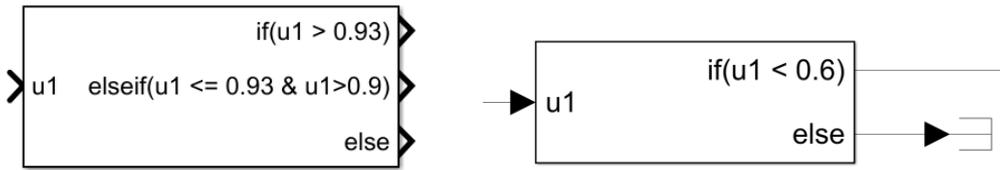


Fig.72 Updated conditional judgment from 3-Class NARX combination option

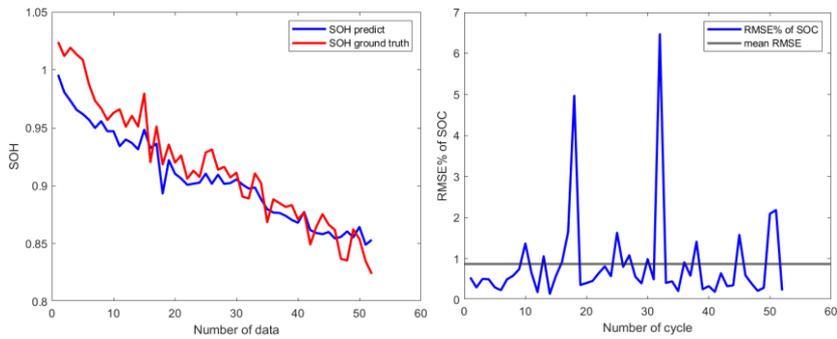
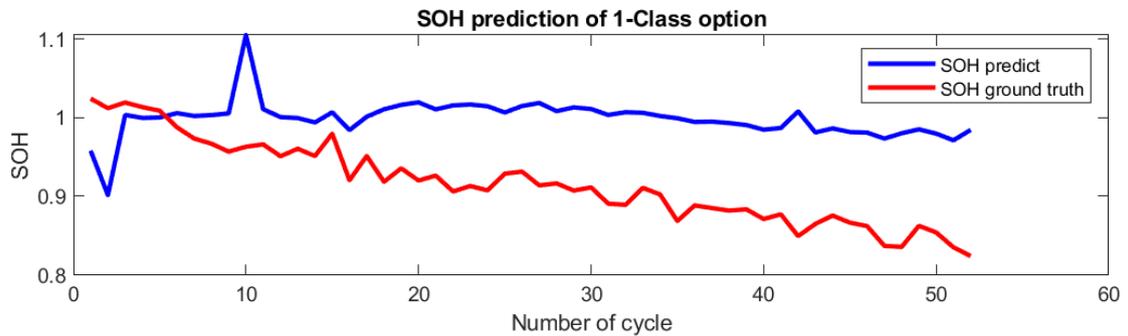


Fig.73 Updated conditional judgment result SOH RMSE%=1.99% SOC mean RMSE%=0.87%

#### 5.4.2 Scenario test results



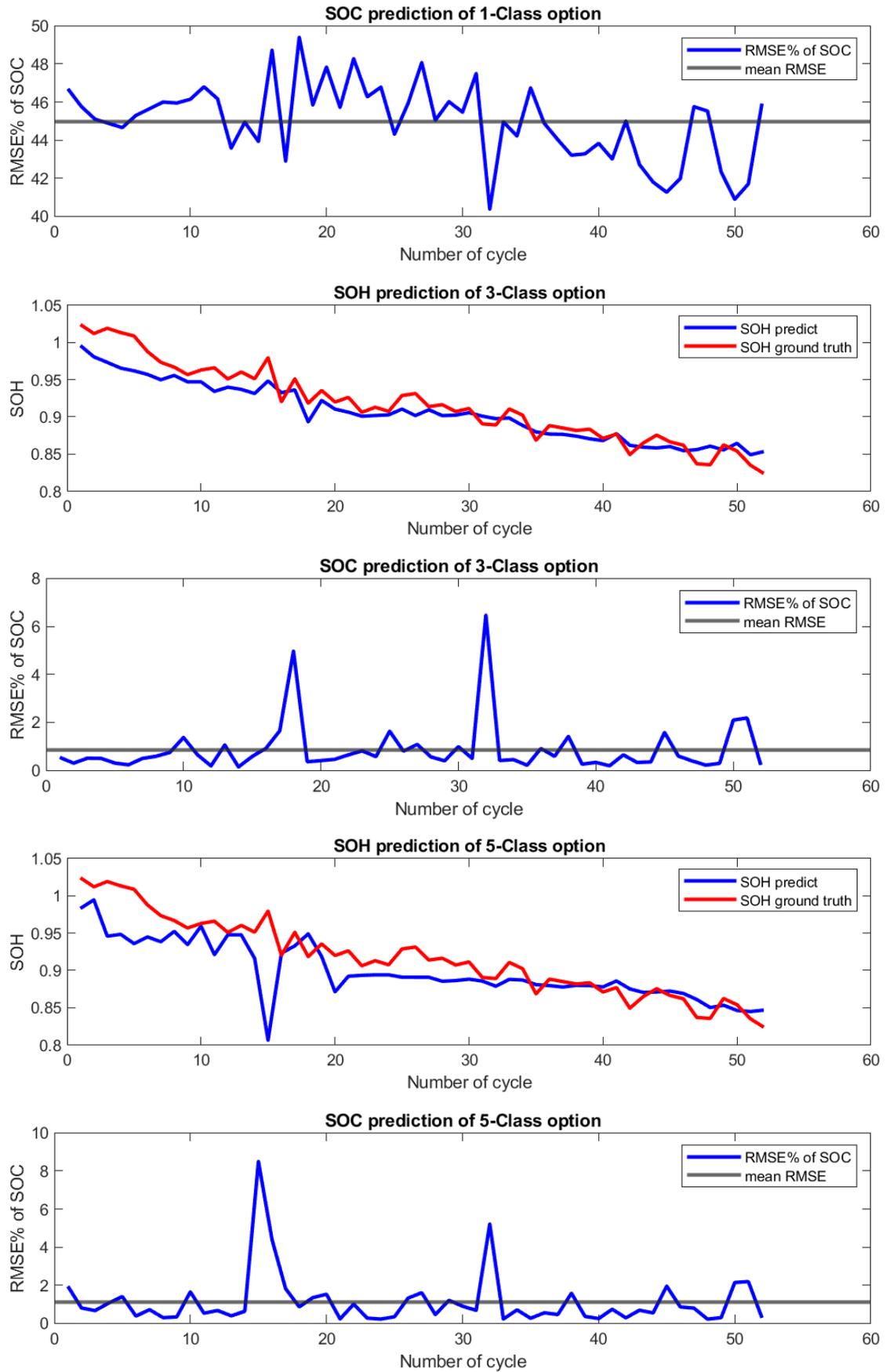


Fig.74 1,3,5 class SOH and SOC test results

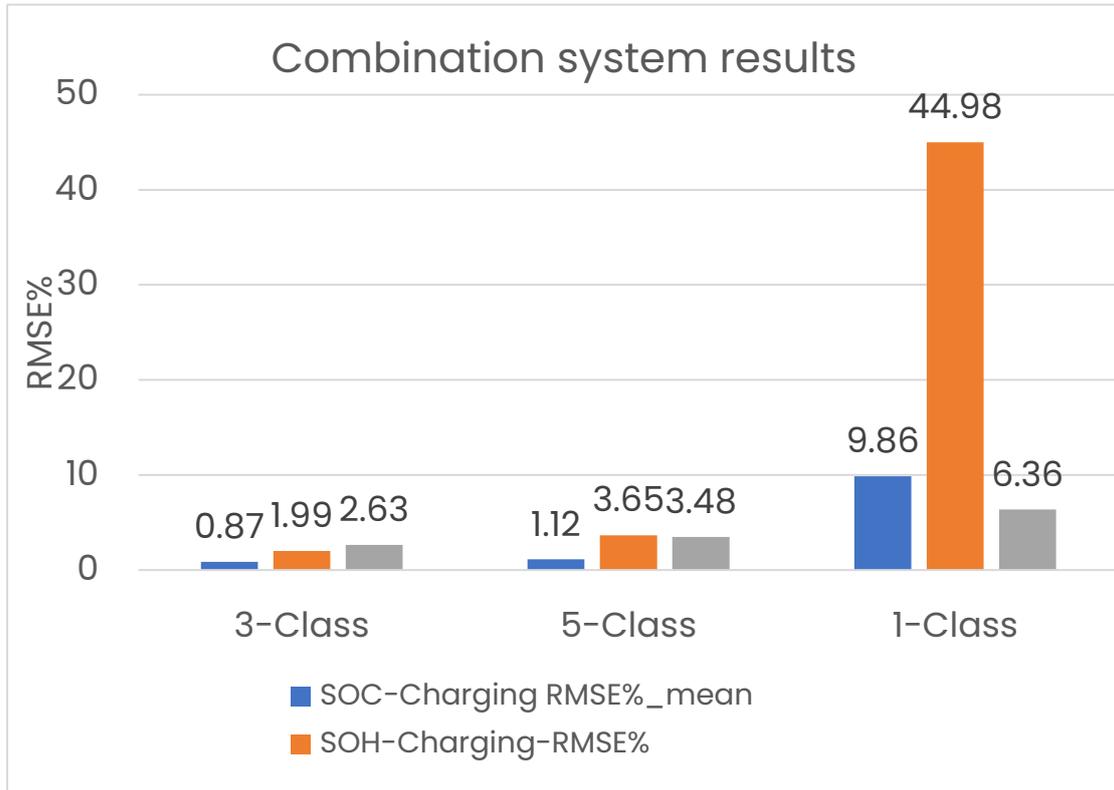


Table 10 Performance comparison for three models

From table 10 it is obvious that 1 class model has the worst performance. Due to SOH estimator and SOC estimator both have intrinsic errors, if SOH is as an input of SOC estimator the error of SOH will influence the performance of SOC estimator, and then SOH estimator itself will also be impacted. This model is too sensitive to the noise. Thus, it has the worst performance.

Models with 3 and 5 class are relatively not so sensitive to error because in those structures SOH is not an input signal but a judgment. Theoretically, 5 class model should have a better performance than 3 class model. But more class means smaller SOH interval for each network, which can cause wrong SOC network selection, and this can explain why 3 class model is better than 5 class.

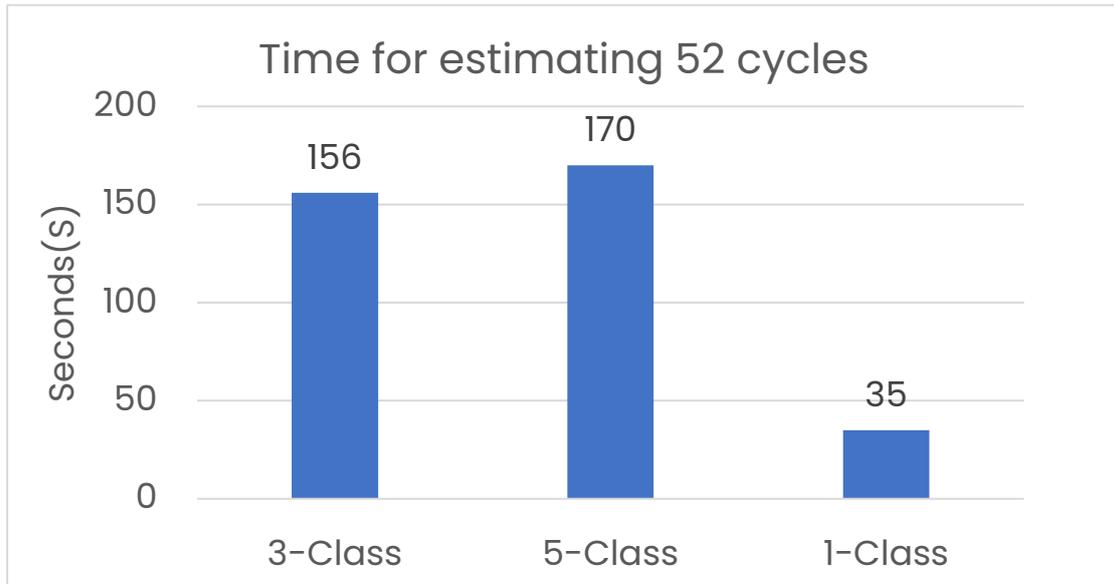


Table 11 Estimation time for three models

Table 11 also shows the estimation time consumption for three models. It is obvious that 5 class model have most computation task to do so it has the longest time consumption while 1 class model only need to compute two neural networks, the time consumption for it is the shortest one.

## 5.5 Possible improvements for SOH estimator

This section will introduce a filtering algorithm for result optimization. However, although this algorithm has achieved excellent results on a fixed training set, it lacks flexibility. In the end, this method is only used in the prediction of individual SOH and is not used in the combined model.

### 5.5.1 Butterworth filter introduction

Butterworth filter is the most famous filter among the filters designed by modern design methods. Because of its simple design and no obvious shortcomings in performance, it is easy to manufacture and achieve design performance. Among them, the Butterworth filter is characterized by the smoothest frequency response curve in the passband.

The transfer function of a Butterworth low-pass filter can be expressed as the square of the amplitude versus frequency as follows:

$$|H(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}} = \frac{1}{1 + \varepsilon^2 \left(\frac{\omega}{\omega_p}\right)^{2n}} \quad (27)$$

Among them,  $n$  is the filter order,  $\omega_c$  is the cut-off frequency, and  $\omega_p$  is the passband edge frequency.

General speaking Butterworth filter is a low-pass filter that filters out high-frequency noise and preserves low-frequency signals. I would like to illustrate this principle with the following simple example. As figure.75 shows, we generate a random noise signal as raw input data. In the meanwhile, we perform the fast Fourier transformation to check the frequency domain's distribution. Then we apply the Butterworth filter and set the cut off frequency is 100Hz. The bottom two images of figure.75 show the filter effect. From filtered FFT distribution all frequency higher than cut off frequency (100Hz) are filtered. And the distribution in time domain is much cleaner that the raw data.

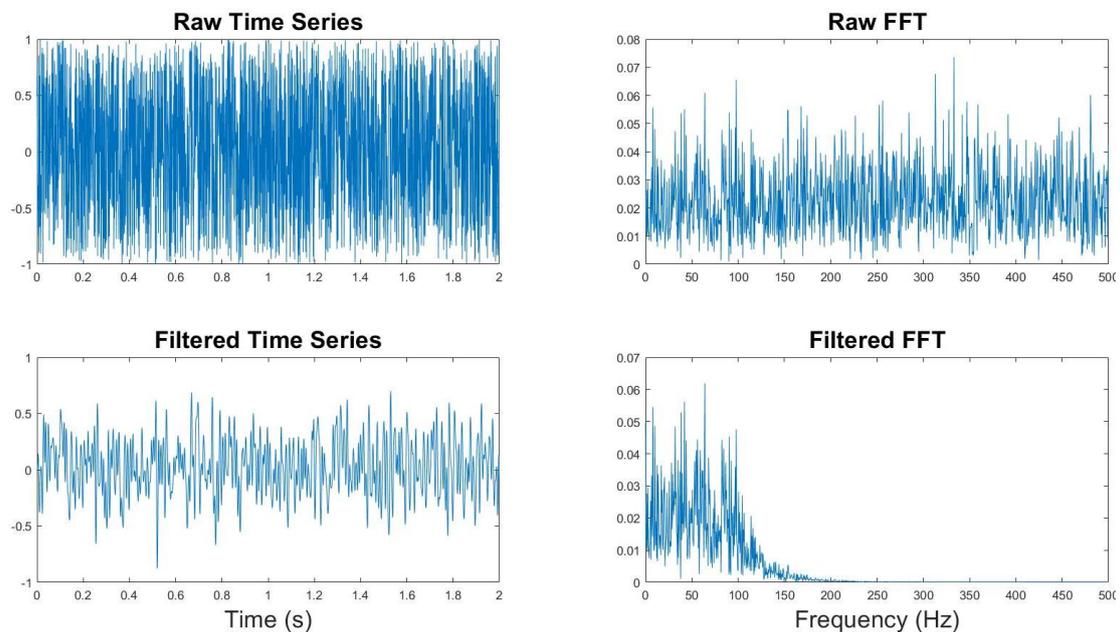


Fig.75 Butterworth filter

In MATLAB, to use the Butterworth filter, it must first define two transfer function coefficients of the filter: 'a' and 'b'. It can be defined by the following MATLAB function:

$$[b, a] = \text{butter}(n, \omega_n) \quad (28)$$

$$y = \text{filter}(b, a, x) \quad (29)$$

Where parameter  $n$  is the order of filter,  $\omega_n$  is cut off series,  $x$  is input data vector.

The results of data smoothing are showed in section 5.4.2

### 5.5.2 data smoothing

After data cleaning and smoothing, and connect all 55 cycles together, the results are as follow.

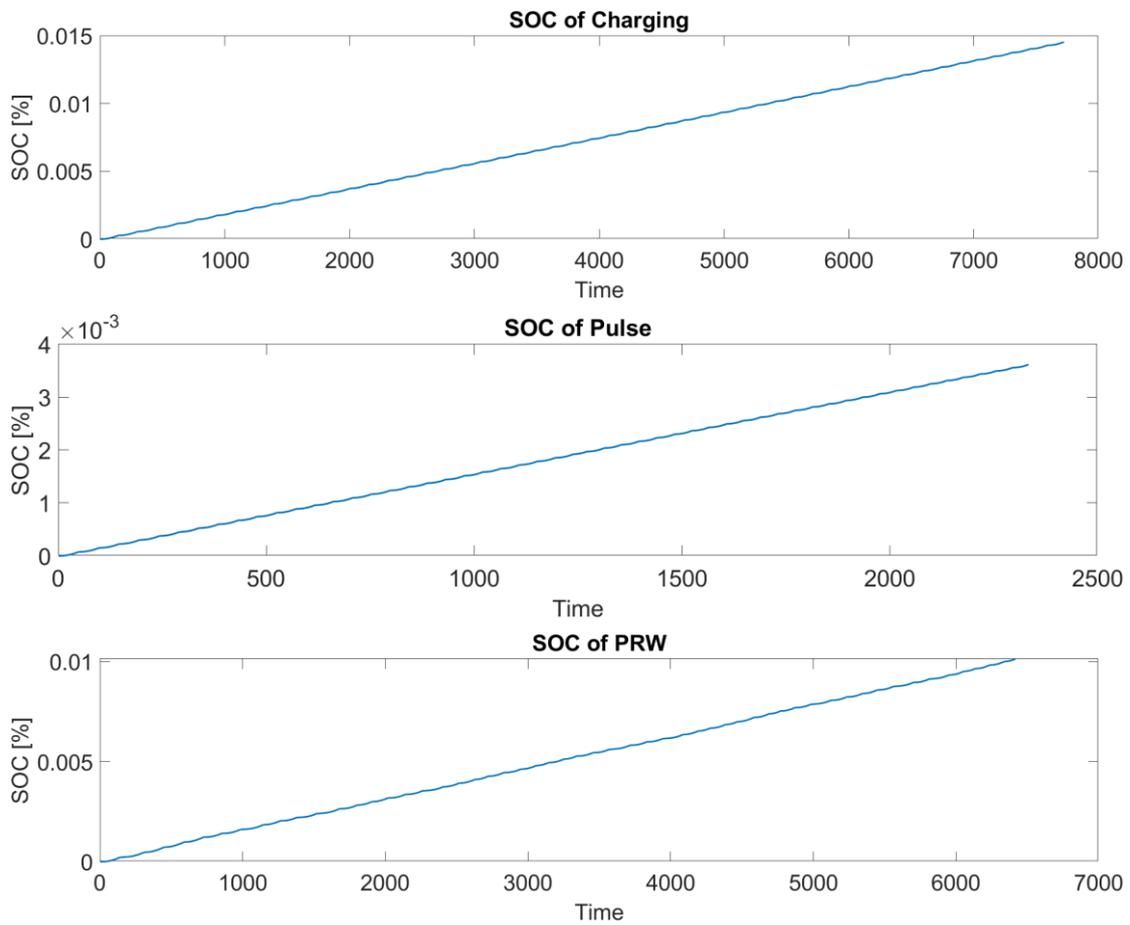


Fig.76 SOC profile after smoothing

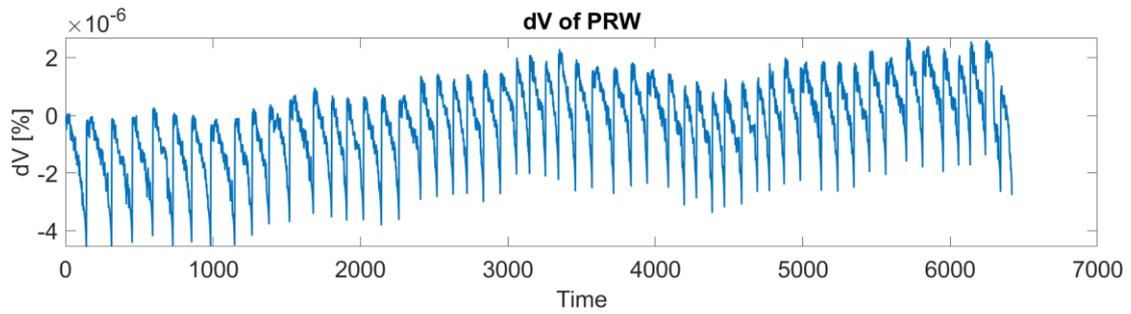
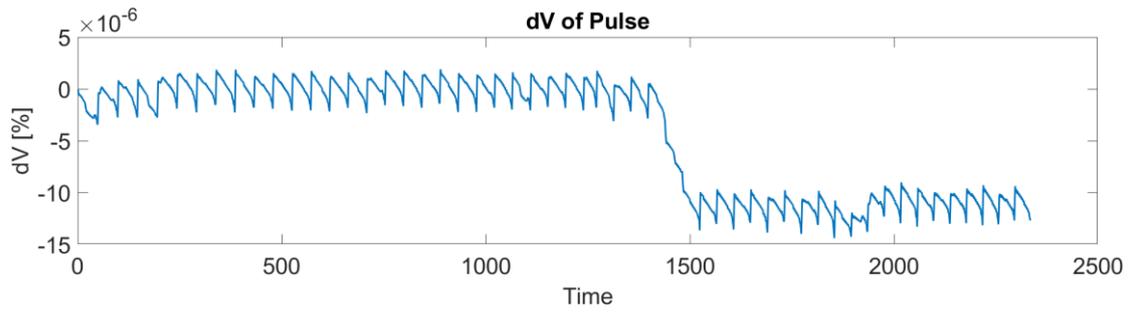
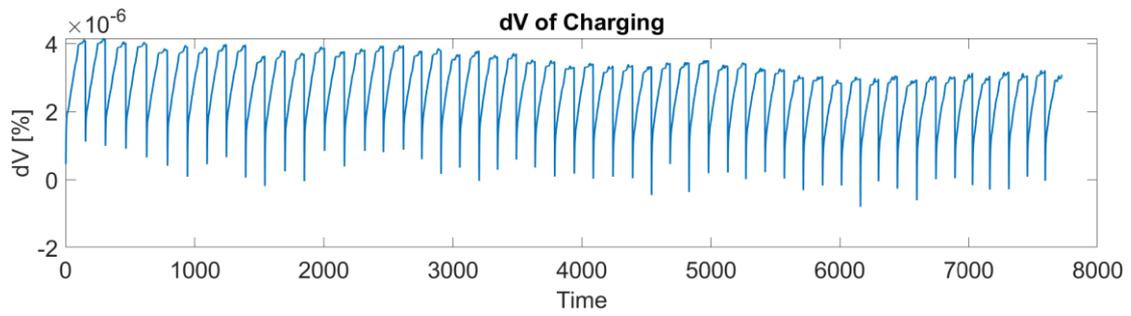
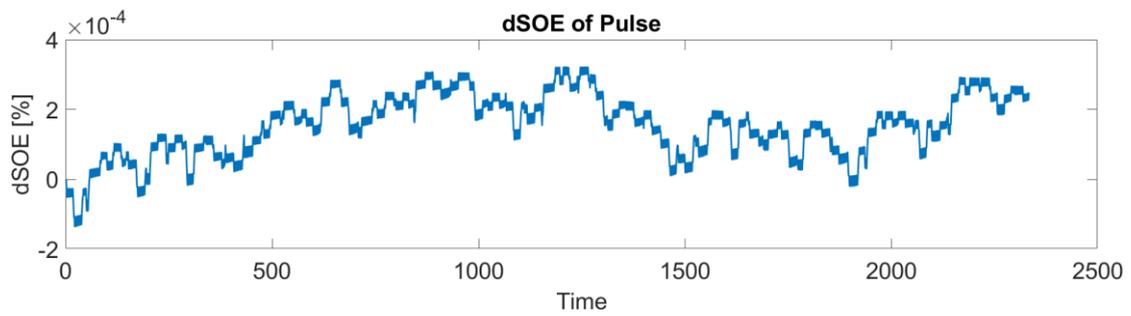
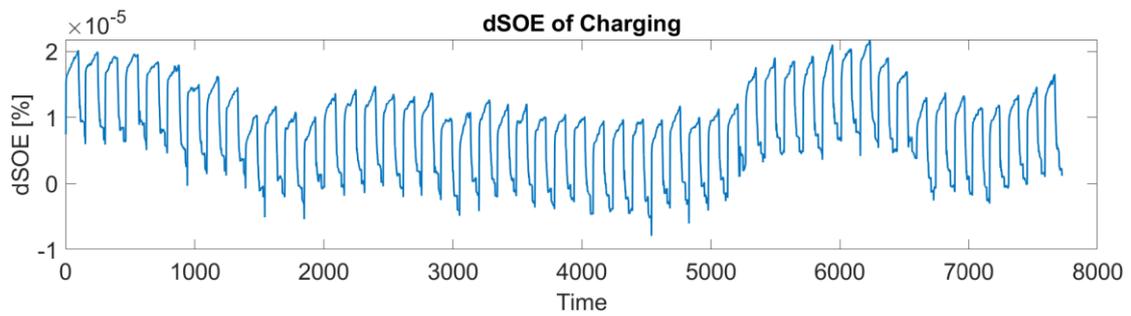


Fig.77 dV profile after smoothing



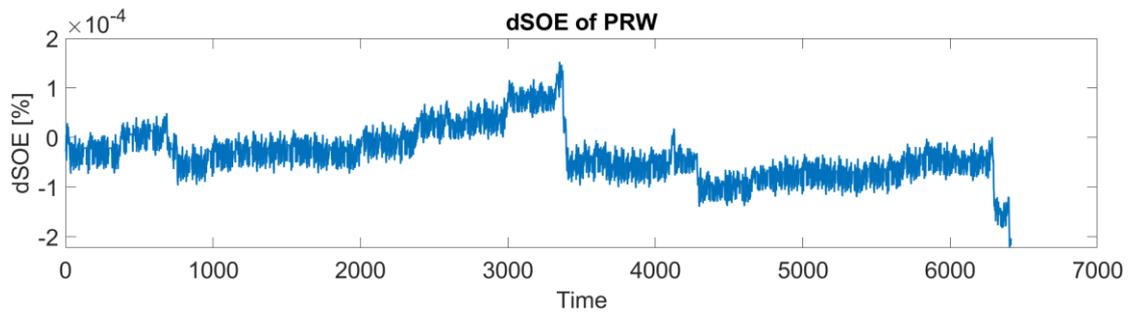


Fig.78 dSOE profile after smoothing

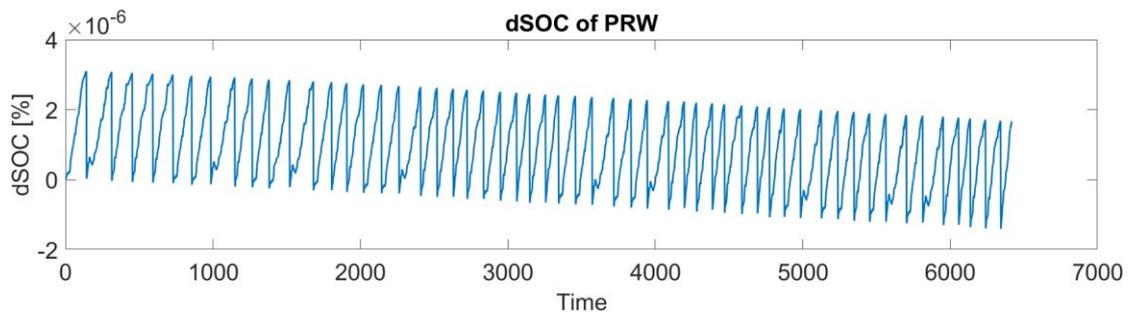
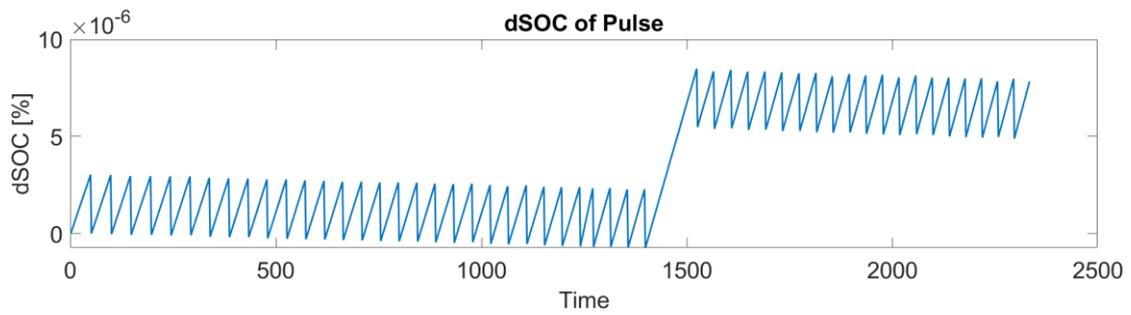
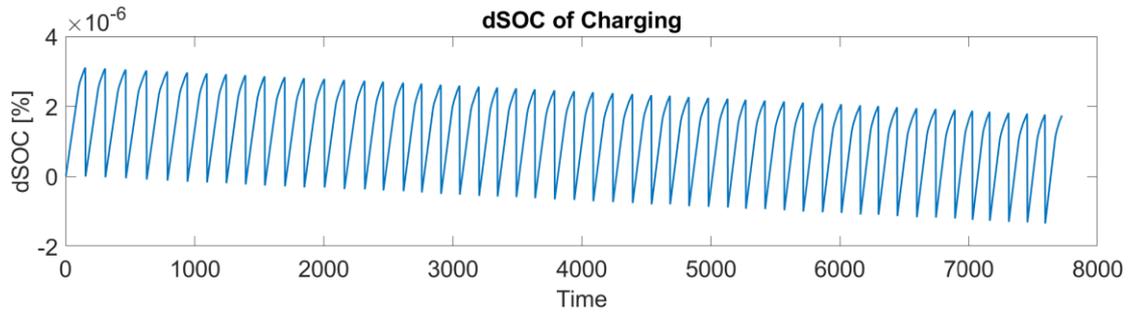
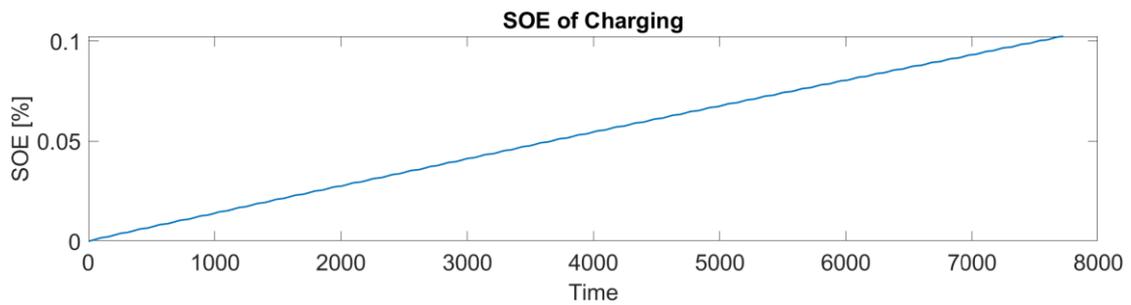


Fig.79 dSOC profile after smoothing



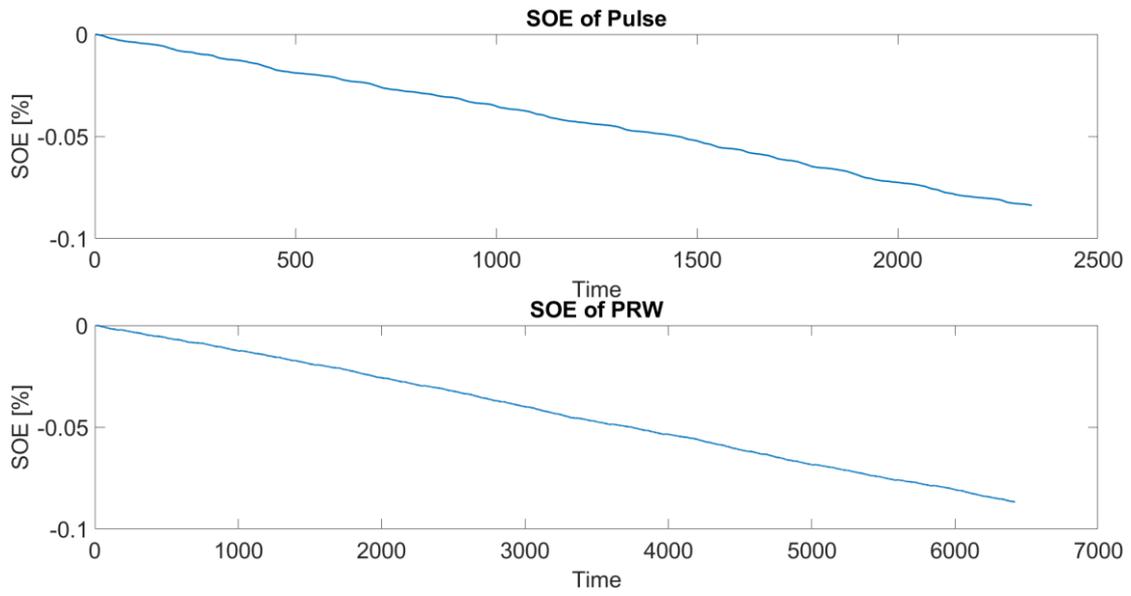


Fig.80 SOE profile after smoothing

### 5.5.3 Training result

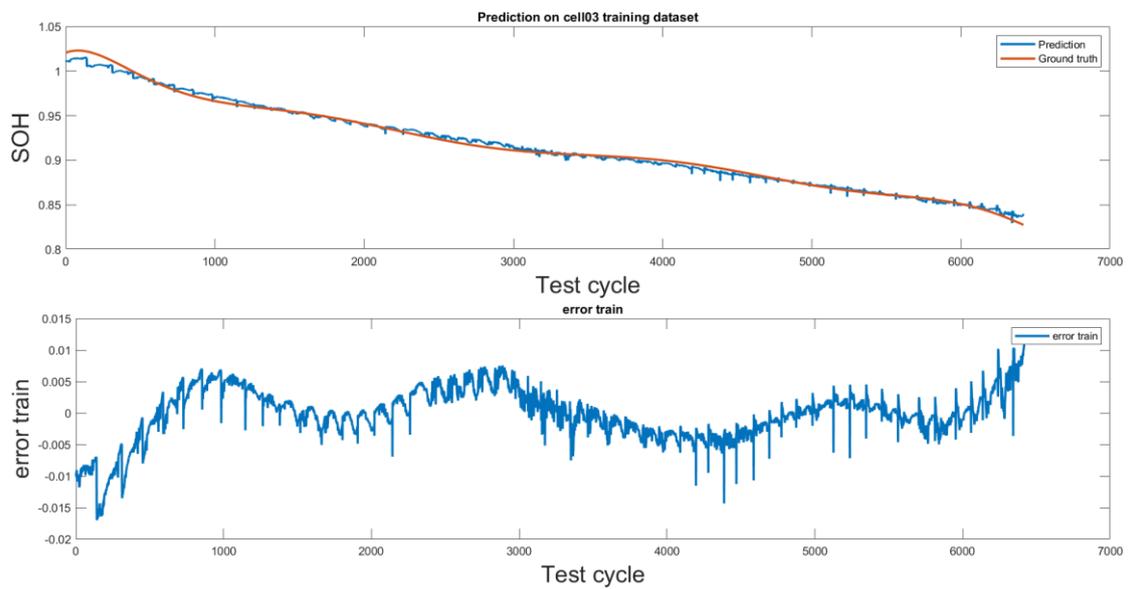


Fig.81 PRW phase training results 0.4035% 0.4535%MAX

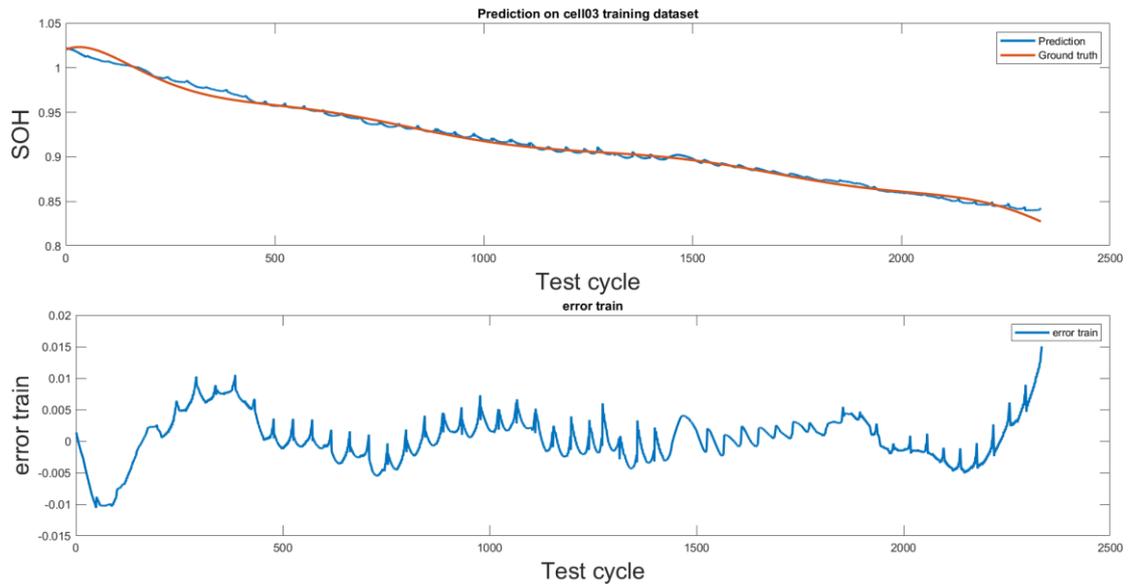


Fig.82 Pulse phase training results 0.3759, 0.3759max

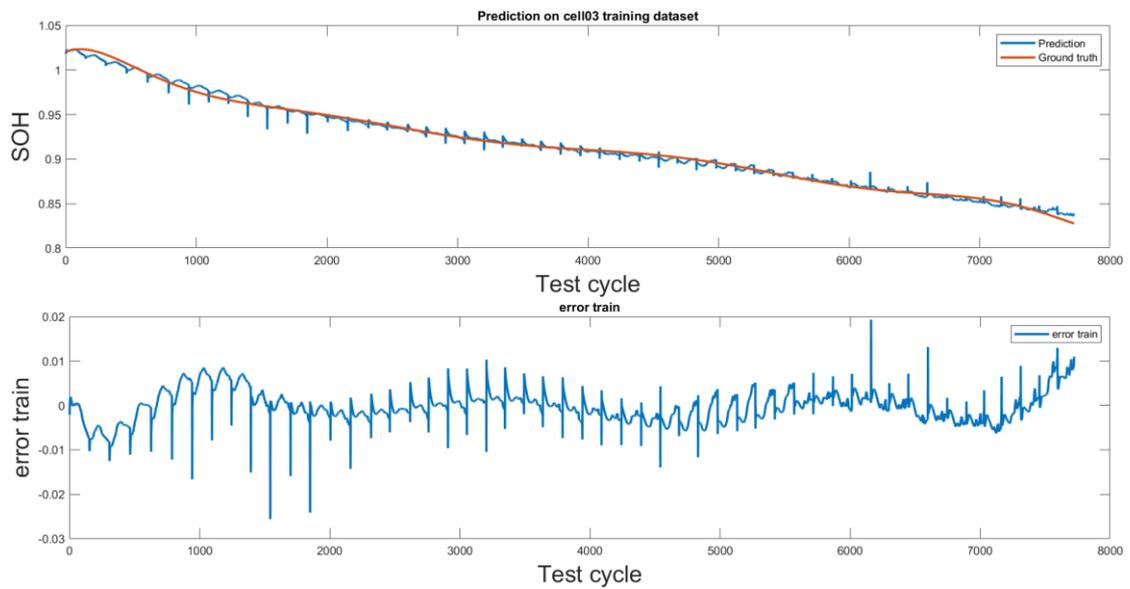


Fig.83 Charging phase training results 0.3418 0.3418

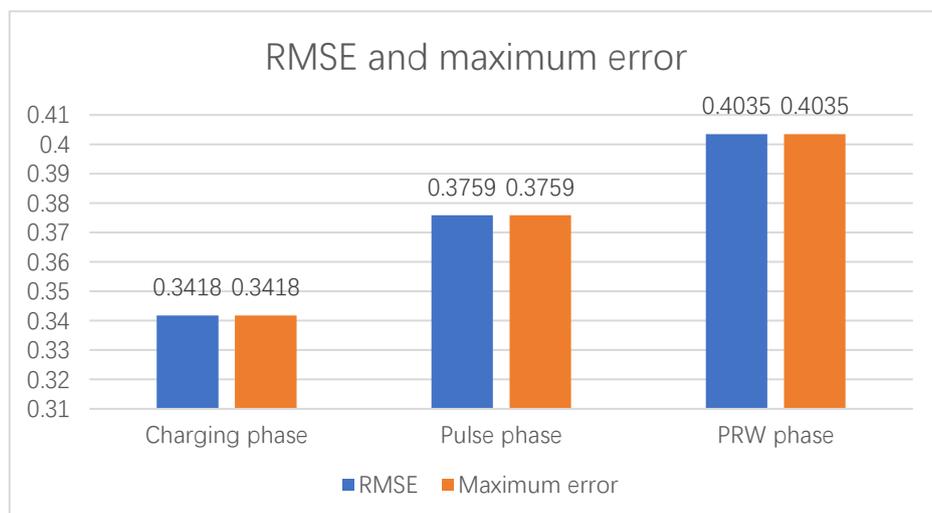


Table 12 RMSE and maximum error

#### 5.5.4 Simulink model

The macro structure of the model is the same as before, but the MATLAB function is used to realize the construction of the Butterworth filter. make it available for online real-time computing

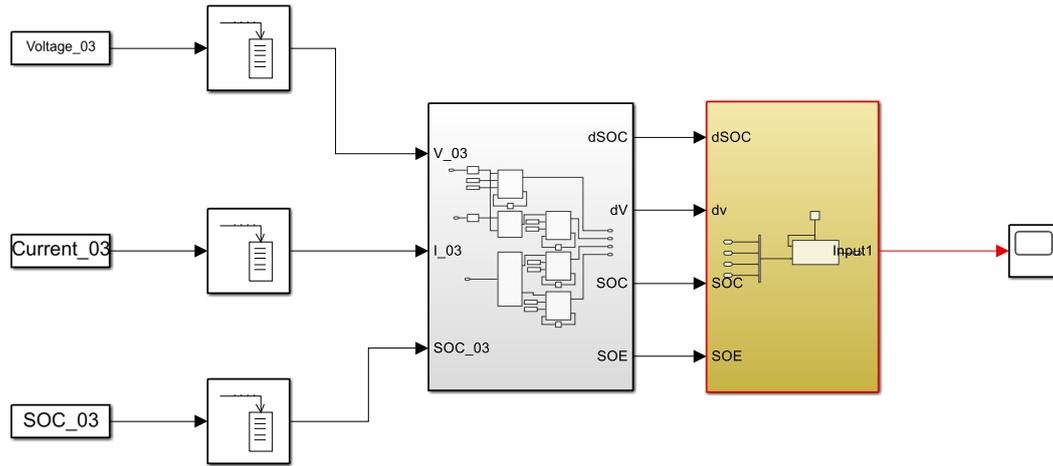


Fig.84 Macro structure of the model

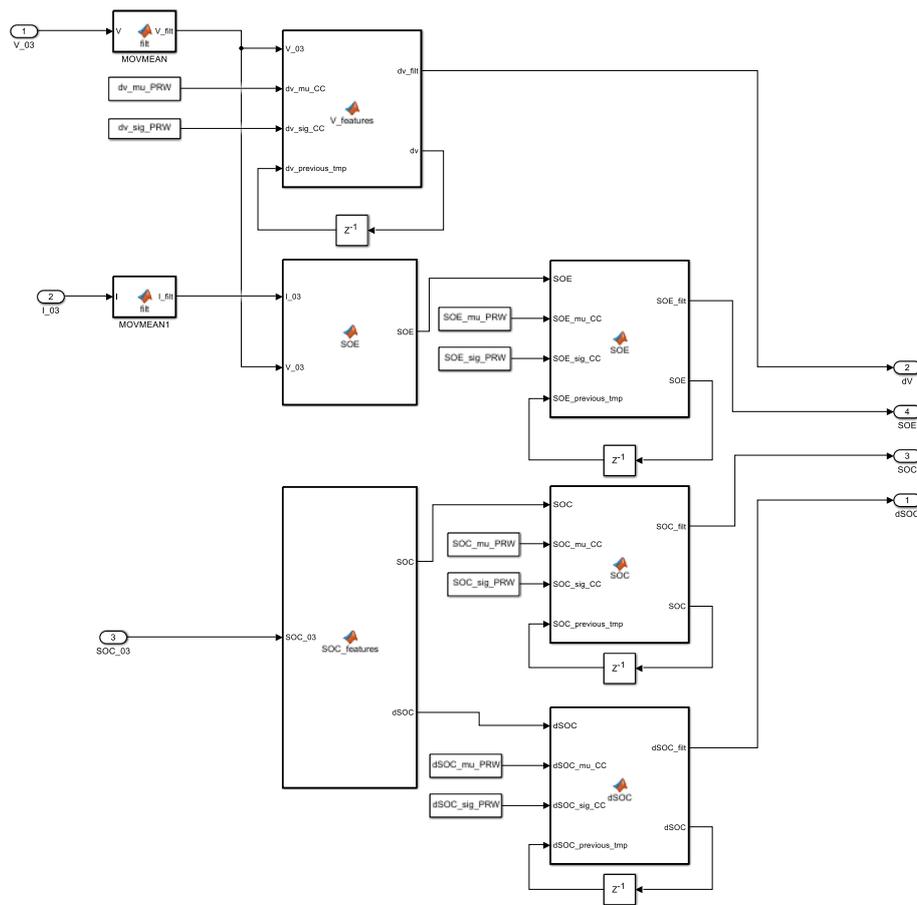


Fig.85 Feature extractors with Butterworth filter

The test result of Simulink model is bellow, which is the same result with MATLAB training process.

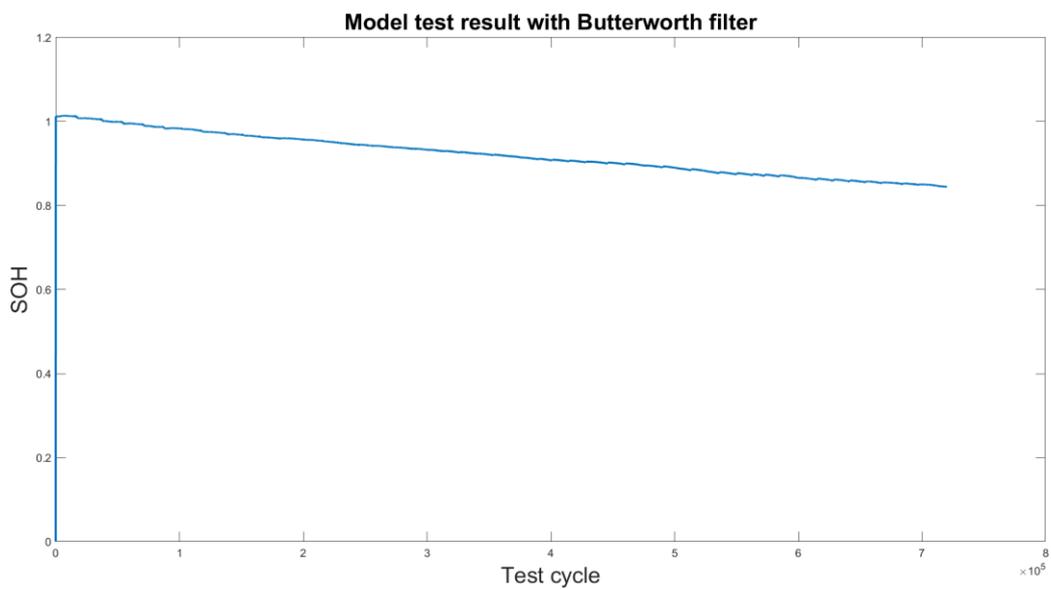


Fig.86 Simulink test result for PRW phase.

### 5.5.5 Limitations of this approach

In the training process, in order to obtain the best filtering effect, the Butterworth filter must wait for all features to be extracted before filtering the features.

Considering the Butterworth's transfer function, it has been found through testing that if real-time filtering is to be implemented in this process, the order of the input data must be in strict one-to-one correspondence with the data order during training. This requirement can be explained as follow:

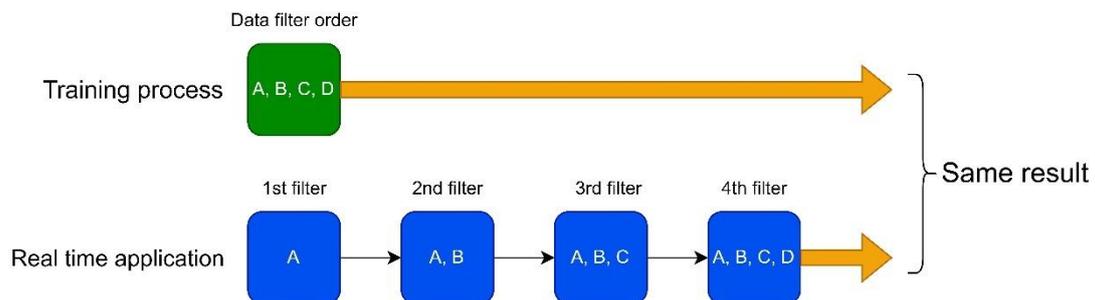


Fig.87 Data order of Butterworth filter for real time application

If the data arrangement in the training process is A, B, C, D. Then in real-time computing, in order to obtain the same result, the data order processed by Butterworth must be A, AB, ABC, ABCD. That is, when the next data is given, the previous data is also appended. This makes the data of the test set cannot be freely combined. That is, the test scenario described in Section 5.3 cannot be implemented. If the problem of this data combination can be further studied and solved, this approach can be then used for real-time calculation and greatly optimize the calculation results.

## Chapter 6. Conclusion

To sum up, the estimation of battery state of health based on artificial intelligence network can achieve accurate results. This method completely avoids the modeling of the internal circuit of the battery by building an uncomplicated neural network. Compared to traditional methods, such as ICA, neural network-based estimation models can achieve real-time prediction capabilities without waiting for the end of the entire test cycle. This method has great advantages when the battery structure is complex and difficult to model.

However, the implementation of this method relies on the acquisition of training datasets, a process that takes several months to achieve aging of the battery. Considering that the aging of the battery is divided into two types: calendar aging and cyclic aging, the months experienced in the entire test phase can only be evaluated by cyclic aging, and the calendar aging can only be obtained by waiting for a longer time. Therefore, this method has poor economy in the acquisition of training data.

It is also found that the measurement accuracy of the dataset is also an important factor affecting the training results. In the initial data processing process, it has been found that there are many unreasonable huge errors in the data set. For example, as shown in figure.37, SOH showed a huge drop on the 32nd cycle. For another example, as shown in figure.28, during the discharge cycle, the current oscillated in an abnormal square-wave-like manner, causing all other parameters measured throughout the cycle to deviate significantly from normal values. To deal with abnormal SOH value, it has been manually deleted the value of the 32nd cycle to ensure that it is not affected by it. For other errors, 'move sliding window' approach can filter them out.

For the process of feature extraction. How to select the features with the highest correlation with battery aging is a key area of research. The five features selected in this thesis have already expressed this correlation well, but it is necessary to continue to study the features with better correlation.

In the selection of neural network, LSTM is used in this thesis, mainly considering that LSTM has inherent advantages in processing time sequential data, which can eliminate the problems existing in traditional RNNs such as gradient vanishing and gradient explosion. According to table 9 LSTM shows a good learning capability with processed data.

In the construction of the Simulink model, the main difficulty is to use the MATLAB function blocks to compile the real-time training process. In this step, the data processing sequence of Simulink must be examined and tuned to match the network design requirements.

As for the use of Butterworth filter, although the conclusions in Section 5.4.3 have shown that this filter can greatly improve the training of the network, the main difficulty lies in its strict requirement of data consistency, which limits the test dataset's combination. This method could be used if it can guarantee that the same cycle as the test set would be used in the actual test, but such a requirement is hardly practical. Therefore, the entire statement in Section 5.4 only illustrates the availability of Butterworth filter for improving the training results but does not mean it can be used in real time application.

## Chapter 7. Bibliography

- [1] [Ageing mechanisms in lithium-ion batteries - ScienceDirect](#) J. Vetter et al.
- [2] [A review on lithium-ion battery ageing mechanisms and estimations for automotive applications - ScienceDirect](#) Anthony Barre et al.
- [3] [Degradation diagnostics for lithium ion cells - ScienceDirect](#) Christoph R.Birkel et al.
- [4] [High-temperature storage and cycling of C-LiFePO<sub>4</sub>/graphite Li-ion cells - ScienceDirect](#) K.Amine et al
- [5] [Abuse tolerance behavior of layered oxide-based Li-ion battery during overcharge and over-discharge - RSC Advances \(RSC Publishing\)](#) Kun Qian et al.
- [6] [Correlation between dissolution behavior and electrochemical cycling performance for LiNi<sub>1/3</sub>Co<sub>1/3</sub>Mn<sub>1/3</sub>O<sub>2</sub>-based cells - ScienceDirect](#) Honghe Zheng et al.
- [7] [Overcharge reaction of lithium-ion batteries - ScienceDirect](#) Takahisa Ohsaki et al.
- [8] [Adaptive Battery Management and Parameter Estimation Through Physics-Based Modeling and Experimental Verification | IEEE Journals & Magazine | IEEE Xplore](#)  
Christopher R.Lashway et al.
- [9] [State of Health Estimation of Lithium-Ion Batteries Using Capacity Fade and Internal Resistance Growth Models | IEEE Journals & Magazine | IEEE Xplore](#) Arijit Guha et al.
- [10] [State-of-Health Estimation of Lithium-Ion Batteries Using Incremental Capacity Analysis Based on Voltage–Capacity Model | IEEE Journals & Magazine | IEEE Xplore](#)  
Jiangtao He et al
- [11] [A New Method for State of Charge and Capacity Estimation of Lithium-Ion Battery Based on Dual Strong Tracking Adaptive H Infinity Filter \(hindawi.com\)](#) Zheng Liu et al.
- [12] [Real-Time State-of-Health Estimation of Lithium-Ion Batteries Based on the Equivalent Internal Resistance | IEEE Journals & Magazine | IEEE Xplore](#) Xiaojun Tan et al.
- [13] [Estimation of Li-ion Battery State of Health based on Multilayer Perceptron: as an EV Application - ScienceDirect](#) Jungsoo Kim et al
- [14] [Real-time state-of-health estimation for electric vehicle batteries: A data-driven approach - ScienceDirect](#) Gae-Won You et al

- [15] [Identify capacity fading mechanism in a commercial LiFePO<sub>4</sub> cell - ScienceDirect](#)  
Matthieu Dubarry et al.
- [16] [Batteries | Free Full-Text | Innovative Incremental Capacity Analysis Implementation for C/LiFePO<sub>4</sub> Cell State-of-Health Estimation in Electrical Vehicles \(mdpi.com\)](#) Elie Riviere et al.
- [17] [Critical review of state of health estimation methods of Li-ion batteries for real applications - ScienceDirect](#) M.Berecibar et al.
- [18] [State of health estimation of second-life LiFePO<sub>4</sub> batteries for energy storage applications - ScienceDirect](#) Jiang et al.
- [19] [A quick on-line state of health estimation method for Li-ion battery with incremental capacity curves processed by Gaussian filter - ScienceDirect](#) Li et al.
- [20] [A fast estimation algorithm for lithium-ion battery state of health - ScienceDirect](#)  
Tang et al.
- [21] [On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression - ScienceDirect](#) Weng et al.
- [22] [Efficient state of health estimation of Li-ion battery under several ageing types for aeronautic applications - ScienceDirect](#) Zhang et al.
- [23] [State of health estimation algorithm of LiFePO<sub>4</sub> battery packs based on differential voltage curves for battery management system application - ScienceDirect](#)  
Berecibar et al.
- [24] [On-board state of health estimation of LiFePO<sub>4</sub> battery pack through differential voltage analysis - ScienceDirect](#) Wang et al.
- [25] [State of health prediction of lithium-ion batteries based on machine learning: Advances and perspectives - ScienceDirect](#) Xing Shu et al.