
THE UNIVERSITY OF
WAIKATO
*Te Whare Wānanga o Waikato*

# Distinguishability of Rechargeable Battery Equivalent Circuit Models

by

Eden Poihipi

A thesis submitted in fulfillment for the
degree of Master of Engineering

in the
Division of Health, Engineering, Computing and Science, School of Engineering

February 2023

# Declaration of Authorship

I, Eden Poihipi, declare that this thesis titled, 'Distinguishability of Rechargeable Battery Equivalent Circuit Models' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, th tis thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Eden Poihipi

Date: 21 February 2023

# Acknowledgements

I would like to acknowledge the Battery Management Group at the University of Waikato which provided immense support for this thesis over the last 2 years, in particular Vance Farrow and Christopher Dunn.

I would like to thank Professor Jonathan Scott, for being a remarkable supervisor.

Thank you to my friends and Whānau for their ongoing aroha and support.

Kia whakatōmuri te haere whakamua.

# Abstract

The contention made by Berthier et al. - that competing fractional-element equivalent-circuit models of battery cells are indistinguishable in the presence of noise - is revisited in this work. A set of six equivalent circuit models (ECMs) were chosen as a subset of Westerhoff's generalised ECM from 2016. These six ECMs had increasing complexity, and gave a simulated output similar to that of an idealised Lithium battery impedance spectrum. By extending the frequency range of Berthier, using methods suggested by both Mauracher & Karden, and Hasan & Scott, sets of simulated electrochemical impedance spectra (EIS) are generated. A Nelder-Mead based optimisation method, which employs multiple restarts, is used to fit each ECM to the sets simulated data. The data was produced with 0%, 1%, 3% and 5% noise. The optimisations showed that all six ECMs were distinguishable at 0% noise, but as the noise increased - distinguishability is lost. Typically a 4-element ECM is shown to model the simulated data with precision which is comparable to the error in the data. When optimising the ECMs against measured data (measured with an Egilent 66332A, E5270 and a Keithley 2460A) from INR and NCA batteries, it is seen that the 3-element ECMs are an adequate model. However there is some variation between the two battery chemistries. Battery ageing and the measurement system used also contribute to variation in which ECM models the data best. Further work on this project should include repeating this fitting method on different battery chemistries, and collating and fitting ECMs to a battery which is measured at different points as it ages.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

The inner workings of a battery are widely speculated, the mystery inviting electrochemists and engineers alike. Electrochemists seek to characterise the internal processes of a battery by using ECMs to dedicate a component to a battery segment. Electrical engineers wish to model and predict state of charge (SoC) and state of health (SoH) of a battery, which could be achieved by knowing the exact contents of a battery model. This is the motivation for this project.

Although rechargeable batteries have been used for more than 100 years, it is still unreliable to predict when a battery will fail in-situ [1]. Implantable medical devices must have their batteries removed from a person at set intervals, despite its electric vehicle counterpart indicating a similar battery could still have years of use left [2]. This disparity occurs because the remaining life of a battery cannot be reliably predicted. Battery ageing is a well researched field, however external indicators of battery ageing are challenging to detect.

The literature reveals the most common way to model a battery is by using an Equivalent Circuit Model (ECM). It is widely disputed which ECM best represents battery behaviour, some containing simple R-C networks, while others use fractional derivative capacitors (constant phase elements). The breadth of models is discussed in full length in Chapter 2.

The top graph of Figure 1.1 shows the impedance of a Lithium-Titanate cell over a range of frequencies. The figure highlights data that was previously unseen from Electrochemical Impedance Spectroscopy data (EIS) due to measurement limitations. It is hypothesised that this low frequency data could be the key to relating Equivalent Circuit

Model (ECM) components to segments of the data, an example of what this might look like is seen in the bottom graph of Figure 1.1.

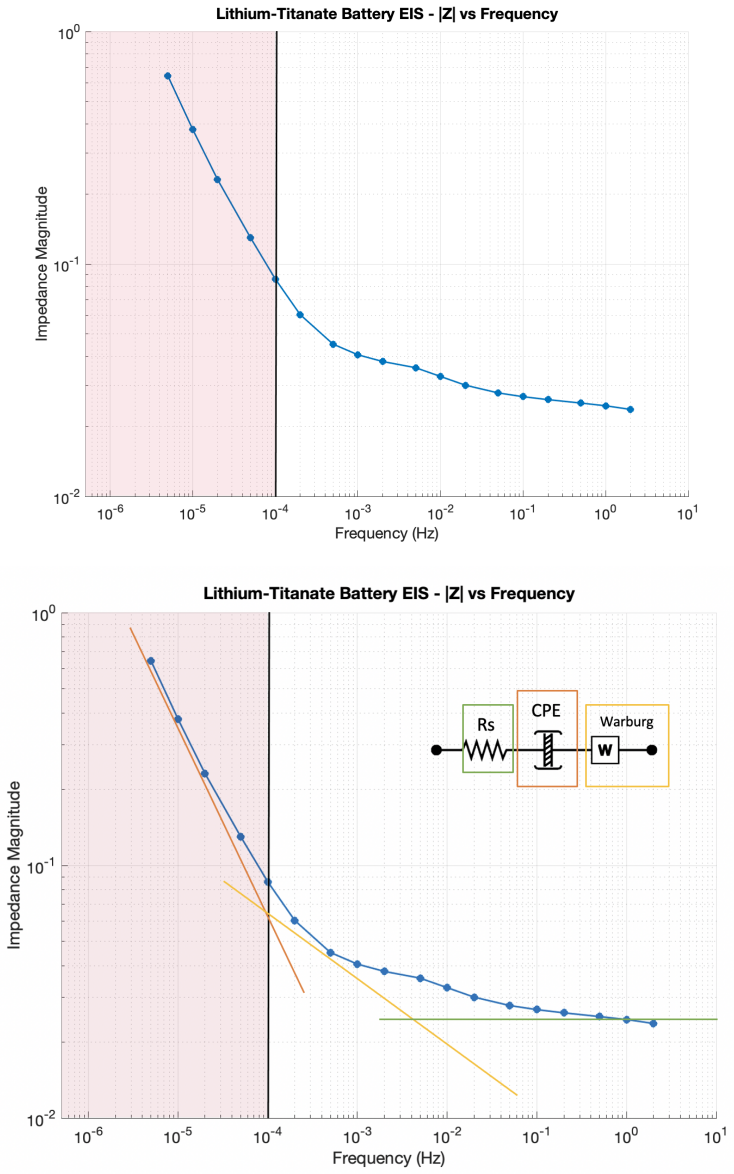


FIGURE 1.1: The top graph shows EIS measurement of a Lithium-Titanate cell, and the bottom graph shows how an ECM could be represented by the EIS data. The highlighted (red) region on both graphs data that, previously, could not be obtained reliably. The bottom graph overlays an equivalent circuit model to demonstrate the relationship between the data and each circuit component. The green line represents the series resistance, the orange line represents a constant phase element and the yellow line represent a Warburg element in this particular case.

## 1.2 Project goals

The aims of the project are three-fold:

(1) Determine a set of Equivalent Circuit Models

(2) Produce a program to optimise each Equivalent Circuit Model against measured battery data, in order to determine each components parameters.

(3) Determine if the Equivalent Circuit Models are distinguishable from one another when optimised

## 1.3 Published Work

The core work of this thesis was written into a journal article at the end of 2021. The paper was co-authored with Professor Jonathan Scott and Chris Dunn. It was accepted by Elsevier in March 2022, and published in the Journal for Electroanalytic Chemistry [3]. The bulk of this thesis, particularly Chapters 3 & 4, are based off of this published work. The paper has been added at the bottom of this thesis for convenience (Appendix A). This was a tremendous achievement, and contributed to the pool of papers published by the battery management team at the University of Waikato.

## 1.4 Thesis structure

Chapter 1 introduces how impedance data from a battery can be simulated using Equivalent Circuit Models (ECMs), discusses why this is important to do and how others have done it in the past.

Chapter 2 is a literature review critiquing the approaches of electro-chemists and electrical engineers alike. Both parties share a common goal of finding accurate battery circuits to provide information for practical applications. This literature review ponders the strengths and weaknesses of each party, and proposes an extension to the methods presented.

Chapter 3 describes how the ECMs were chosen, the optimisation method and the full fitting process used to fit theoretical circuits to physical battery data.

Chapter 4 shows examples of the methodology, discusses what determines if ECMs are distinguishable from one another, and draws a conclusion on whether ECMs can be distinguishable from one another with noise added to the simulated data.

Chapter 5 is a discussion on the validity of the methods produced in this thesis. This chapter analyses the results from the methodology and testing, and draws conclusions on fitting ECMs to data.

## 1.5 Background

Battery research can be likened to the ocean, both vast and used frequently, yet they both contain secrets still to be discovered. This thesis will draw back the curtains a little more, in hopes to reveal insights into equivalent-circuit models (ECMs) for modelling rechargeable batteries.

### 1.5.1 Electrochemical Impedance Spectroscopy

The first rechargeable battery was the lead-acid battery, invented by Gaston Planté, which was used to power early electric vehicles and telegraph systems [4]. Throughout the twentieth century the nickel-cadmium battery, alkaline battery, and lithium-ion battery were developed bringing battery technology up to its current point in 2022 [5, 6]. Researchers are still working on developing more efficient battery technologies - the need for expansion is vital in the future of transportation, computing and renewable energy solutions. s A key characteristic of rechargeable batteries is their impedance. Impedance is a complex quantity which measures a components resistance to AC current, or voltage, and is usually expressed as:

$$Z = R + jX \tag{1.1}$$

Where Z is impedance (ohms), $j$ represents the imaginary unit, R is the resistance of the component and X is the reactance of the component.

Impedance is typically measured by applying a sinusoidal AC current to a cell [7], then measuring the phase shift from the cells response. There are many impedance measurement tool manufacturers which measure complex impedance [8, 9]. In a linear system the potential difference of the cell can be represented by:

$$V_t = V_s sin(\omega t) \tag{1.2}$$

where $V_t$ is the potential at time, t, $V_s$ is the amplitude and $\omega$ is the angular frequency. Angular frequency can be calculated by:

$$\omega = 2\pi f \tag{1.3}$$

where $\omega$ is angular frequency, and f is frequency (Hz).

Thus the battery's response current, $I_t$, can be shown by:

$$I_t = I_s sin(\omega t + \phi) \tag{1.4}$$

where $I_s$ is the amplitude of the input signal, $\omega$ is the angular frequency, t is time, and $\phi$ is the phase shift between the signal and the battery response. Using Ohms law it is clear to see that Impedance is shown by:

$$Z = \frac{E_t}{I_t} = Z_0 \frac{sin(\omega t)}{sin(\omega t + \phi)} \tag{1.5}$$

then applying Euler's relationship [10] we reach the final expression for Impedance, Z($\omega$):

$$Z(\omega) = Z_0(cos\phi + jsin\phi) \tag{1.6}$$

### 1.5.2 Measuring impedance & Simulating Constant Phase Elements

While the tools exist to measure impedance, there is a limit on the lower frequency limit with most of these devices. Hasan and Scott, who are a part of the battery management team at the University of Waikato, solved this problem by devising a new method of measuring EIS data from rechargeable batteries. Initially they used a Solartron 1260A on a pair of NiMH batteries, finding a discrepancy between the results [11]. This discrepancy was not repeatable which led them to believe that the measurements at a lower bound at 1mHz were not reliable. This led to the development of their low frequency measurement system. The procedure for measuring impedance is listed in their paper [11], and is summarised below:

- A test current is supplied to the battery
- The State of Charge of the battery is preset for each test frequency
- The amplitude of the test current is fixed according to the following equations:

$$Q_{in} = \frac{I_0}{\pi f} \tag{1.7}$$

and,

$$Q_{out} = -\frac{I_0}{\pi f} \tag{1.8}$$

- A sinusoidal current waveform is created at each test frequency
- Measure the response over multiple cycles and store the results in a file
- A Hann window is applied to account for any drift, then the magnitude and phase data is extracted using a Discrete Fourier Transform.
- The complex impedance is calculated using Ohm's law, dividing the voltage by the stimulus current at each test frequency

This algorithm eliminates offset drift (making measurements repeatable) and distortions in impedance measurements. This is the method used by the battery management group, and so this thesis. Figure 1.2 shows the setup of the measurement system. The equipment used are the Agilent 66332A, E5270, and the Keithley 2460A).

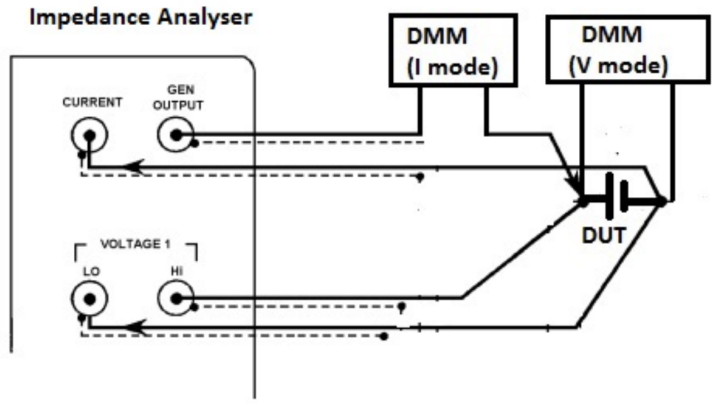


FIGURE 1.2: The connections of the Impedance Measurement System used by the Battery Management Group and the University of Waikato. The current and voltage of the DUT (Device Under Test) are measured by the two DMM's (Digital Multimeters), attached to the Impedance Analyser.

The battery management group have also been working on simulating fractional capacitors, or constant phase elements (CPEs). SPICE is a commonly used circuit simulator available for use, however there is no support for simulating CPEs. Thus a circuit was created to mimic the behaviour of a CPE in SPICE. Before this work, a common workaround was to use a network of resistors and capacitors. However, a less popular method is shown in (Figure 1.3) which aligns with the work of Morrison [12].

Morrison [12] showed that conceptually, in both the high and low end frequencies, the network is infinite which is impractical for simulation hence the termination branches. The SPICE model that was settled upon takes 6 inputs: the order of the CPE, upper and lower frequency bounds, the branch density, a base frequency and the magnitude of the impedance at the base frequency. The base frequency is used to calculated the

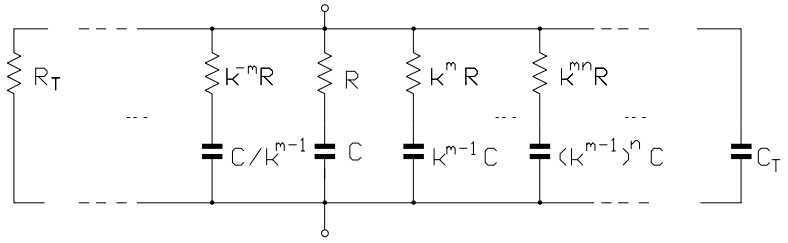


FIGURE 1.3: A network of parallel R & C elements, as described by Morrison [12], with a terminating R on the low characteristic frequency end, and a terminating C on the high characteristic frequency end [13]. The "home" branch has a resistance and capacitance of $R_0$ and $C_0$respectively, and k is the scaling factor on the R branches.

'home branch,' then the branch density is taken into account to determine the number of branches for the network. The branches are repeated until they reach termination at each end (maximum and minimum frequency). The paper describing this method is currently in pre-print to be published by IEEE [14]. The SPICE model acts as verification for simulated data in other programs (Matlab, python etc) such as the optimiser written for this thesis.

## 1.6 Introduction

There is an abundance of literature which presents varying battery models, the focus of this thesis will be the works of Berthier et al [15].

Berthier et al [15] was an electrochemist who sought to discover whether you could distinguish ECMs which contained constant phase elements (CPEs). They determined a method to distinguish circuits numerically, using statistical methods. Berthier et al [15] chose 4 arbitrary ECMs then synthesised a dataset modelled after measured electro-chemical impedance spectroscopy (EIS) data. The data was reproduced with mathematically produced 'noise' added in the increments: 0%, 1%, 2.5% and 5% noise. They then used the fitting method for nonlinear least squares outlined by Fletcher et al [16], and an adapted form of the Nelder & Mead method [17], to fit each ECM to the synthesised data. It was found that the noise-free, synthesized EIS data resulted in mathematically distinguishable ECMs. However, when they attempted to optimise ECMs against sythensized EIS data with greater than 1% noise, the distinguishability was lost.

This thesis examines the work of Berthier et al, offering an extension of their work to date as well as updating the methods and circuitry used in their paper.

Firstly, data will be presented in bode plot format rather than Argand diagrams (or Nyquist plots). This is to capture important frequency information [18].

Secondly, an outline of the exact fitting method, and process, will be discussed. This includes reasoning for the choice of ECMs which was not explicitly given by Berthier and colleagues.

Lastly, an extended frequency range will be examined in this thesis. Berthier and colleagues had a limited frequency range in their experiment, ranging from 100 rad/s (16 Hz) to 1 Mrad/s (160 kHz). While this range may seem adequate, it would be advantageous to choose a frequency range that best represents real-life battery applications. Batteries in appliances are cycled (charged and discharged) infrequently. Car and cellphone batteries can be cycled no less than once a week in some instances. These extended cycling periods equate to frequencies from approximately 1 µHz to 1 mHz (with the equivalent C-rates being 0.006 to 4). This infers that using, and measuring, extra low frequency (ELF) data is vital to accurately represent rechargeable battery usage [11, 19, 20].

# Chapter 2

# Background of ECMS (Literature Review)

In 1947, a seminal paper on battery modelling using fractional elements was published. Randle [21] conducted an experiment to model Faradaic conduction in a cell. He describes an electrical equivalent-circuit model of a battery using a fractional capacitor of order 0.5 (Warburg element) which can be extracted from EIS data - specifically impedance data. This paper acted as a basis for fractional capacitors, or constant phase elements (CPE's), being included in equivalent-circuit models of batteries. Randle's ideas and findings bled into Brug's [22] paper which came about nearly 40 years later, as an extension of using CPEs in a equivalent-circuit model.

Brug et al [22] used heavy math to demonstrate the impedance response of EIS data. They mention that CPEs are relevant to modelling impedance data and should not be discounted when considering equivalent circuit models. However, Brug et al [22] expressed in their paper how they did not agree with some of Randle's [21] findings. The key information that Brug et al and Randle disagreed on related to which part of the battery a CPE could represent. While Randle believed a CPE could describe behaviours of the outer layer of a battery, Brug et al proposed the idea that the outer layer was represented by something else entirely. Instead Brug et al described how a CPE has a high influence on frequency dispersion even at small phase angles, making it a significant part of battery equivalent-circuit models. This formative paper built upon Randle's [21] initial findings to offer an alternative use for CPE's. Brug et al's [22]

By 1994, Westerlund et al has published 2 papers on the matter of CPE's [23, 24]. In the first paper [23], Westerlund et al focuses on the mathematics behind capacitors, offering a new mathematical relationship for the calculation of capacitors. This is important as CPEs are a form of capacitor. A few years later Westerlund et al published a second

paper [24]. Westerlund et al describe the mathematics behind fractional capacitors, offering a theory that all capacitors are fractional, with modern plastic dielectric capacitors having an order from 0.9978 to 0.99995 rather than 1 as most people believed. Westerlund et al [24] also proved using Laplace transforms that each CPE has an alpha value, which can differentiate two fractional-order elements (in series for example). This paper distills years of experience from Randle and Brug et al in a few pages, providing a deep and insightful understanding of the mathematics of CPEs, and how they may be used in equivalent-circuit models. Westerlund et al also published a paper in 2016 [25], at a similar time to Westerhoff et al [25], on equivalent-circuit models which use CPEs.

In 2016, Westerhoff et al [25] proposed a chemically derived ECM. Westerhoff and colleagues model contained CPEs which demonstrated how electrochemists have recognised fractional-derivative functions as key components in battery equivalent circuit models, since Randles paper in 1947 [21]. Each of the 16 components, or group of components, represented a segment of a battery (i.e. the anode, cathode etc). Westerhoff and colleagues suggested ways to simplify the ECM noting that some components in their model could be discarded depending on the application. The models Westerhoff and colleagues used were based on experimental impedance spectra data [26, 27], with the most reduced model presented containing as few as 8 components. This is significant because it established that not all of the components were relevant for real-life battery applications, hinting that a simpler model could be as representative of a Lithium battery as their initial 16 component model. This conclusion is similar to the discourse of Berthier [15], whose paper was published 15 years prior.

Berthier [15] asserted that although there are simpler ECMs that are still representative of impedance output, there is still speculation on which model is the most representative of a rechargeable battery. From this speculation, Berthier [15] conducted a theoretical study from an electrochemists point of view, to determine if 4 simplified ECMs could be distinguished from one another mathematically. This was done by synthesizing impedance data as a control set, then judging how well each circuit model fit this data. Berthier [15] concluded that the circuits chosen in his paper were not mathematically distinguishable from one another i.e. the chosen fractional ECMs all fit Berthier's synthesized impedance data almost equally as well.

## 2.1 Non-Fractional Equivalent Circuit Models

There is a tsunami of literature which examine non-fractional equivalent circuit models, typically based on Thévenin RC equivalent circuits [28, 29, 30, 31, 32]. First order [33, 34, 35] and second order [36, 37, 38] RC circuits, with one/two capacitors respectively,

are typically used to model Lithium-Ion Batteries to determine SoC, SoH and simulate impedance data. The trade-off between complexity and computational speed is a key consideration when running any form of simulations. RC ECMs are still popular for this as they are simpler, less intensive (both mathematically and computationally) despite it being shown that they are not appropriate [39]. This informs the decision to disregard non-fractional ECMs, giving the concession that the computations for this project could be more intensive. RC ECMs are generally simulated using high-level programming languages such as MATLAB & Simulink [29, 35]. The use of these programs, with built in functions, will be crucial for speeding up computationally intensive simulations.

## 2.2 Extra Low Frequency Measurements

This path leads us to Hasan & Scott [11], who, in 2019, showed that the Electrochemical Impedance Spectroscopy (EIS) data at Extra-Low Frequencies (ELF) points towards an ECM that contains CPE's. They elude to this fact by showing the EIS response below a milihertz which is significantly lower than many papers which measure EIS data (manually, or with 'off-the-shelf' equipment) [40, 41, 42, 43, 44]. The idea of measuring at ELF is supported by Mauracher & Karden [20] who suggested it was essential to measure at frequencies in the microhertz range to see more EIS data, which may reveal insights into the correct ECM. These two papers emphasise the idea of measuring at ELF in order to see a larger amount of EIS data which may be the key to updating the work of Berthier et al.

## 2.3 Bringing it all together

Berthier's [15] paper is a culmination of decades of research starting with Randle planting the seed of using CPEs to represent segments of a battery. Randle paved the way for Brug and Westerlund to extend this knowledge through mathematics, leading to the understanding the CPEs can be easily modelled and calculated as elements in equivalent-circuit models. People like Berthier and Westerhoff furthered this train of thought by testing various configurations of CPE based equivalent-circuits to try and best estimate a battery model.

This literature review shows a clear disconnect between electrochemists and electrical engineers - the former having fractional ECM with limited optimisation tools, while the latter focuses on simpler models with rigorous optimisation tools and methods. The culmination of these two approaches, as well as using an extended frequency range, could be the key to updating, and providing solutions for, Berthier et al's previous work.

# Chapter 3

# Methodology

This section is heavily based on the paper which was published to Elsevier in the Journal of Electroanalytical Chemistry - April 2022 - which has been included in Appendix A.

## 3.1 Choosing Equivalent Circuit Models

Equivalent-circuit models (ECMs) are a widely discussed topic in literature, with many methods for deciding which ECMs are most representative of physical batteries. Westerhoff et al [25] described three methods of battery ECM characterisation in their paper. The first was 'white-box' modelling, which is a theoretical method which employs fundamental physics and chemistry to deduce a potential battery ECM [45]. The second is 'black-box' modelling, a practical method which uses blind fitting on sets of measured battery data to deduce a potential battery ECM [27]. The third, and final method, is the 'gray-box' modelling which is a happy medium between the 'white' and 'black' methods, using both theoretical driven circuits and measured data sets to derive ECMs.

For this thesis, the chosen ECMs will be based on the modelling and characterisation of batteries using 'gray-box' modelling from Westerhoff et al [25]. 'White-box' modelling is purely theoretical thus cannot always be applied to physical batteries. The range of devices, and conditions, in which batteries operate suggests purely theoretical modelling could produce ECM models that are less representative in a physical space. Alternatively, 'black-box' modelling is resource intensive as it is an empirical method. Gathering large, accurate data sets from batteries requires large amounts of time and computationally heavy software for 'black-box' modelling to be effective [27]. Furthermore, this modelling type is ideal for linear relationships which is not appropriate for this thesis.

As discussed in Chapter 2, Westerhoff et al presented a CPE based general battery model (Figure 3.1) in their paper [25] which will be used as a spring board to determine which ECMs will be used for optimisation. They elude to this general circuit (Figure 3.1) having excess components for practicality. This means there's a correlation between the number of components in the ECM and its accuracy. They also concluded that the number of components could be determined by the level of accuracy required for the application. Thus it is possible to simplify the circuit, by eliminating components in Figure 3.1, without effecting the overall impedance output of the circuit.

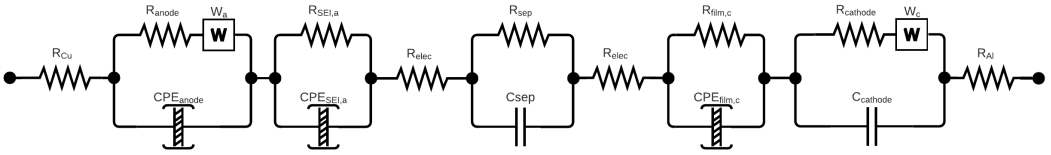


FIGURE 3.1: Potential complete equivalent circuit of a battery according to Westerhoff [25]. Constant phase elements (CPEs) are used to characterize electrochemical behavior at the anode, solid electrolyte interface (SEI), separator (sep), and cathodic thin film (film,c). Note also resistances (R) associated with the copper and aluminium foils (Cu and Al), anode, SEI, electrolyte (elec), separator, thin film and cathode, and capacitances (C) at the separator and cathode. Warburg elements are denoted by $W_a$ and $W_c$ at the anode and cathode, respectively.

To broaden the scope of Berthier's [15] method, six ECMs will be considered - an additional two - which can be seen in Figure 3.3. The first consideration when choosing the simplest form ECM was the impedance output of a measured battery. Figure 3.2 shows the impedance magnitude and phase responses from rechargeable batteries, including various chemistries and capacities. Therefore, the simplest circuit will be the ECM with the least number of components which yields the same basic shapes as Figure 3.2.

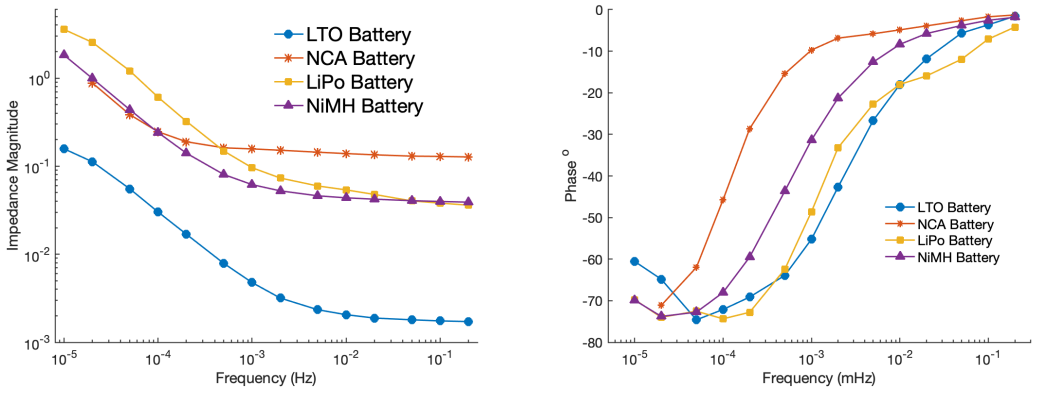


FIGURE 3.2: Impedance magnitude (left) and phase (right) of a selection of rechargeable batteries. LTO is a lithium titanate cell of 40 Ah capacity, NCA is a 4.8 Ah lithium nickel cobalt aluminium cell, LiPo is a 250 mAh lithium polymer cell, and NiMH is a 2.5 Ah nickel-metal hydride cell.

The simplest ECM (Figure 3.3, upper left circuit) consists of two components: a resistor in series with a constant phase element (CPE). While this is akin to the Thévenin equivalent (R-C) circuit, the double-layer capacitance of a solid electrode (found in most batteries) is not purely capacitive. The CPE, in place of the capacitor, accounts for this fact to give a more accurate representation. The impedance of a CPE is calculated using Equation 3.1

$$Z_{CPE} = \frac{1}{C_F s^{\alpha}} \tag{3.1}$$

where $C_F$ is the CPE value (its pseudo- or fractional capacitance, a constant with the dimensions $Fs^{\alpha-1}$), $\alpha$ is the CPE fractional order exponent, $s = j2\pi f$ (where $2\pi f = \omega$, the angular frequency), and $f$ is the frequency in hertz [24]. The total impedance of the R-CPE model can therefore be found by

$$\begin{aligned} Z_{R-CPE} &= R_s + Z_{CPE} \\ &= R_s + \frac{1}{C_F s^{\alpha}} \end{aligned} \tag{3.2}$$

where $R_s$ is the series resistance and $Z_{CPE}$ is the impedance of the CPE (Equation 3.1) in ohms.

By adding a Warburg element (W) to R-CPE we obtain our second ECM. A Warburg element is a CPE with an alpha of 0.5, which appeared first in Randles electrode reaction experiment [21]. By fixing the alpha value to 0.5 [46], a Warburg describes Faradaic impedance in the presence of redox processes in semi-infinite linear diffusion systems [18]. The impedance of a Warburg element is given by

$$Z_{\text{Warburg}} = \frac{1}{C_W s^{\frac{1}{2}}} \tag{3.3}$$

and so the impedance of the R-CPE-W model is given by

$$\begin{aligned} Z_{\text{R-CPE-W}} &= R_s + Z_{CPE} + Z_{\text{Warburg}} \\ &= R_s + \frac{1}{C_F s^{\alpha}} + \frac{1}{C_W s^{\frac{1}{2}}} \end{aligned} \tag{3.4}$$

where $R_s$ is the series resistance, $Z_{CPE}$ is the impedance of the CPE (Equation 3.1) and $Z_{\text{Warburg}}$ is the impedance of the Warburg element (Equation 3.3).

In practical applications, the conditions of a Warburg element are often not completely met, and so it can be replaced by a CPE to give an extra degree of freedom. (Figure 3.3, middle left circuit, and so forth.) The total impedance of the R-CPE-CPE model is

given by

$$\begin{aligned} Z_{\text{R-CPE-CPE}} &= R_s + Z_{\text{CPE1}} + Z_{\text{CPE2}} \\ &= R_s + \frac{1}{C_F s^{\alpha}} + \frac{1}{C_W s^{\alpha_2}} \end{aligned} \tag{3.5}$$

where $R_s$ is the series resistance, and $Z_{\text{CPE1}}$ and $Z_{\text{CPE2}}$ are the impedances of the two CPEs, and we keep the $W$ subscript to show the origin of the element.

Small phase deviations at high and low frequency extents suggest a parallel element could be present in the ECM. To accommodate these deviations, additional components can be added in parallel to the ECM. The R-CPE-CPE-Rp circuit will have impedance

$$Z_{\text{R-CPE-CPE-Rp}} = R_s + [Z_{\text{CPE1}} + Z_{\text{CPE2}}] \parallel R_p \tag{3.6}$$

where $R_p$ is the parallel resistance. This equation can be expanded by noting that the impedance of parallel elements can be written

$$X \parallel Y = \frac{1}{\frac{1}{X} + \frac{1}{Y}} = \frac{XY}{X+Y} \tag{3.7}$$

leading to

$$Z_{\text{R-CPE-CPE-Rp}} = \frac{[R_s + Z_{\text{CPE1}} + Z_{\text{CPE2}}]\, R_p}{R_s + Z_{\text{CPE1}} + Z_{\text{CPE2}} + R_p} \tag{3.8}$$

and then

$$Z_{\text{R-CPE-CPE-Rp}} = \frac{\left[R_s + \frac{1}{C_F s^{\alpha}} + \frac{1}{C_W s^{\alpha_2}}\right] R_p}{R_s + \frac{1}{C_F s^{\alpha}} + \frac{1}{C_W s^{\alpha_2} + R_p}}. \tag{3.9}$$

The form of (3.6) is typically preferred over (3.9) because it is more intuitive and easier to comprehend. As the expansions rapidly become arduous, parallel notation will be used from here onwards.

The final two ECMs to be investigated have impedances $Z_5$ and $Z_6$

$$\begin{aligned} Z_5 &= Z_{\text{R-CPE-CPE-CPEp}} \\ &= R_s + [Z_{\text{CPE1}} + Z_{\text{CPE2}}] \parallel Z_{\text{CPEp}} \\ &= R_s + \left[\frac{1}{C_F s^{\alpha}} + \frac{1}{C_W s^{\alpha_2}}\right] \parallel \frac{1}{C_P s^{\alpha_p}} \end{aligned} \tag{3.10}$$

where the $\parallel$ operator is taken to have precedence over addition, $\alpha_p$ is the order of the parallel CPE, and $C_P$ the proportionality constant of the parallel CPE; and

$$\begin{aligned} Z_6 &= Z_{\text{R-CPE-CPE-Rp-CPEp}} \\ &= [[R_s + Z_{\text{CPE1}}] \parallel Z_{\text{CPEp}} + Z_{\text{CPE2}}] \parallel R_p \\ &= \left[\left[R_s + \frac{1}{C_F s^{\alpha}}\right] \parallel \frac{1}{C_P s^{\alpha_p}} + \frac{1}{C_W s^{\alpha_2}}\right] \parallel R_p \end{aligned} \tag{3.11}$$

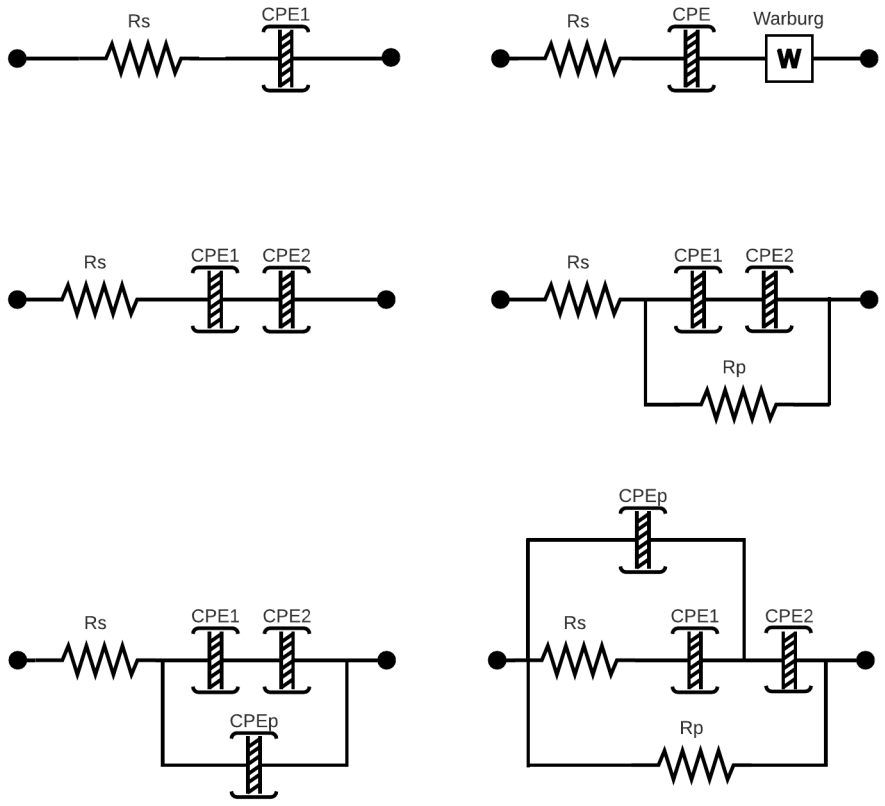


FIGURE 3.3: The equivalent-circuit models (ECMs) considered in this thesis. For convenience and clarity, they are identified by their sequence of elements. The top left circuit can be identified as "R-CPE", the top right as "R-CPE-W", the middle left as "R-CPE-CPE", middle right as "R-CPE-CPE-Rp", then "R-CPE-CPE-CPEp" and finally "R-CPE-CPE-Rp-CPEp".

## 3.2 Optimisation of ECMs

### 3.2.1 Nelder-Mead Simplex Approximation

Due to the complex nature of the equations for optimisation, an appropriate fitting algorithm must be used. The Nelder-Mead Simplex (NMS) approximation [17] was chosen as it is a numerical method - i.e. it doesn't use derivatives for its computations. The NMS approximation minimises a non-linear multidimensional function [47] using an iterative direct search method. This method uses simplexes with N +1 vertices, where N is the number of variables to be found in your function. The general steps are to:

1. Generate an initial simplex from the given vertices (arbitrary sets of points)
2. Check if the termination conditions have been met

3. If the termination conditions have not been met, transform the simplex
4. Repeat steps 2 and 3 until the termination conditions have been met, then move to step 5
5. Return the best vertex (set of points) of the simplex and the value of the function

Following on from that, an example of the full NMS method is outlined. Both are based on the following references: [17, 47, 48, 49] with a visual representation in Figure 3.4. For a 2D function (N = 2) the NMS method requires 3 (N +1) arbitrary points to form the starting simplex. The method then tries to minimise the function using the actions listed blow:

1. The function is evaluated at each point and ordered from lowest function value to highest. For a 2D function, f, let us call the 3 vertices **L**, **N** and **H** so we can calculate f(**L**), f(**N**) and f(**H**). Let us assume that f(**L**)<f(**N**)<f(**H**), correlating to points 'L,' 'N' and 'H' respectively in Figure 3.4.
2. The centroid (mean) is calculated between the 2 best points.The centroid would be calculated between **L** and **N** in this example, noted as **Ce**.
3. The algorithm then tries one or more of these methods to minimise the function (Note: it is not necessary for all of these methods to be used in every iteration).
    - The worst point is reflected through the centroid calculated previously to see if the function has been minimised, then checked for convergence. For this example, point **H** would be reflected through the centroid of **L** and **N** to create a new point **R**.
    - If f(**R**)<f(**L**), then **R** is extended further in that direction to give a new point **E**. If f(**E**)<f(**R**), replace the worst point (**H**) with the reflected/extended point (**E**). Otherwise, replace the worst point (**H**) with the reflected point (**R**). Check for convergence.
    - If extension does not minimise the function, the reflected point, **R** is contracted both inwards and outwards. **Ci** occurs at the midpoint of the worst point(**H**) and the centroid (**Ce**). **Co** occurs at the midpoint between reflected point (**R**) and the centroid (**Ce**). If (f(**Ci**) OR f(**Co**))<f(**N**), replace **H** with the contracted point which gives the smallest value of f, and check for convergence.
    - If contraction does not minimise the function further, then move the two worst points (**N** and **H**) closer to the best point (**L**), usually to the centroid (**Ce**).

4. The convergence of the function is checked at the end of each iteration. The two main conditions for convergence are the number of iterations and the standard deviation of the simplex which are determined by the user. Whichever condition is met first will cause the program to terminate. For this thesis, the chosen number of iterations was set to 10,000, with a standard deviation tolerance

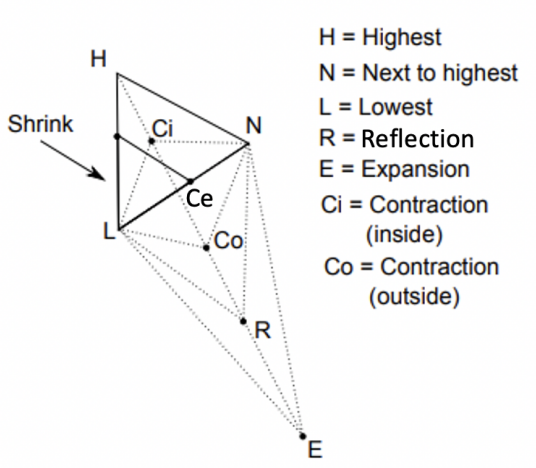


FIGURE 3.4: A visual representation of the Nelder-Mead Simplex method for a 2D function. This shows how each step of the NMS method influences the size of the simplex (the triangle for this example), until it reaches the smallest area possible and thus the minimum values of the function.

## 3.3 Fitting Algorithm & Sequence

This is the method used to update the work of Berthier and colleagues [15]. Following their general method, data is simulated from one of the ECMs. Then, each ECM is optimised (or fit) to this data to see which provides the 'best' fit. The 'best' fit is the circuit which produces the smallest Root-Mean-Square Error (RMSE) between the simulated and the optimised data. The detailed steps of the fitting sequence are listed below:

1. Choose an ECM, then calculate its magnitude and phase data using the appropriate Equation from 3.1 to 3.11. These are referred to as the **simulated ECM** and **simulated data** respectively. Generally the most complex model (R-CPE-CPE-Rp-CPEp) is chosen as the simulation model as it has the most degrees of freedom, since it has the most components.

2. Select an ECM to fit to the **simulated data**, this is called the **fitting ECM**. Then, input an initial 'guess' at the parameters for the **fitting ECM**. The initial guesses are generally informed - alpha values are bound between 0 and 1 and series

resistance values are typically less than 1 Ω based on measured battery data. The amount of noise (as a percentage) can also be input.

3. Use the initial guess parameters to generate an fmp file (Appendix B) for the **fitting ECM**— using the appropriate Equation from 3.1 to 3.11. This is called the **fitting data**.

4. Input the initial guesses, **simulated data** and the **fitting data** into the Nelder-Mead Algorithm for optimisation. The Matlab code for Nelder-Mead algorithm uses the methodology from reference [50] which was altered to allow for complex function inputs and a custom RMSE calculation (Appendix E, Appendix D).

5. The optimisation is carried out, calculating the RMSE between the **simulated data** and **fitting data** (Appendix E). Convergence is reached when the number of iterations is exceeded, or the tolerance for the RMSE is met. During the fitting process, the initial parameter guesses are replaced with randomly generated values to simulate the multi-restart of the optimisation. If the multi-restart values result in a lower RMSE, these parameters are favoured.

6. Use the optimised parameters to generate an fmp file, and graph the **simulated data** and **fitting data** for comparison.

This sequence is carried out for each ECM that is to be fitted to data. The sequence is carried out in what we call a 'progressive' fit, meaning R-CPE (the simplest model) is fitted to the data first. The RMSE is noted, then the output of the R-CPE fit are used as the inputs for the next circuit (R-CPE-W), and so on and so forth up until the most complex R-CPE-CPE-Rp-CPEp model.

# Chapter 4

# Distinguishability of ECMs

This chapter is also based on the paper which was published to Elsevier in the Journal of Electroanalytical Chemistry - April 2022 - which has been included in Appendix A.

## 4.1 Generate Ideal Simulated Data

In this section, Berthier et al's [15] work is revised showing the fitting algorithm used to determine if ECMs are distinguishable with noise added to the data.

To begin, an ECM is chosen to generate the simulated data with zero noise added (ideal case). The most complex model has the highest number of parameters, and so the highest degrees of freedom (8) making it the perfect candidate. The R-CPE-CPE-Rp-CPEp model was given a set of model parameter values that will be used for calculation. These parameters were estimated, but inferred from real measured data. Table 4.1 contains the values used in Equation 3.11 to generate the magnitude and phase of the simulated data. This data was generated over a frequency range of 0.1 µHz to 100 Hz in a 1-2-5 step pattern per decade. The output is shown in Figure 4.1.

TABLE 4.1: Synthetic Model Parameter Values

| Parameter | $R_s$ | $\alpha$ | $C_F$ | $C_W$ | $\alpha 2$ |
|---|---|---|---|---|---|
| Value | 0.05 | 0.75 | 10,000 | 500 | 0.40 |
| Parameter | $R_p$ | $\alpha_p$ | $C_p$ | | |
| Value | 500 | 0.15 | 0.8 | | |

The asymptotes correlating to each of the five elements are also plotted. This enables the reader to relate different plot points to specific parts. For instance, the series resistance, Rs, has the highest influence on the horizontal section of the magnitude plot between

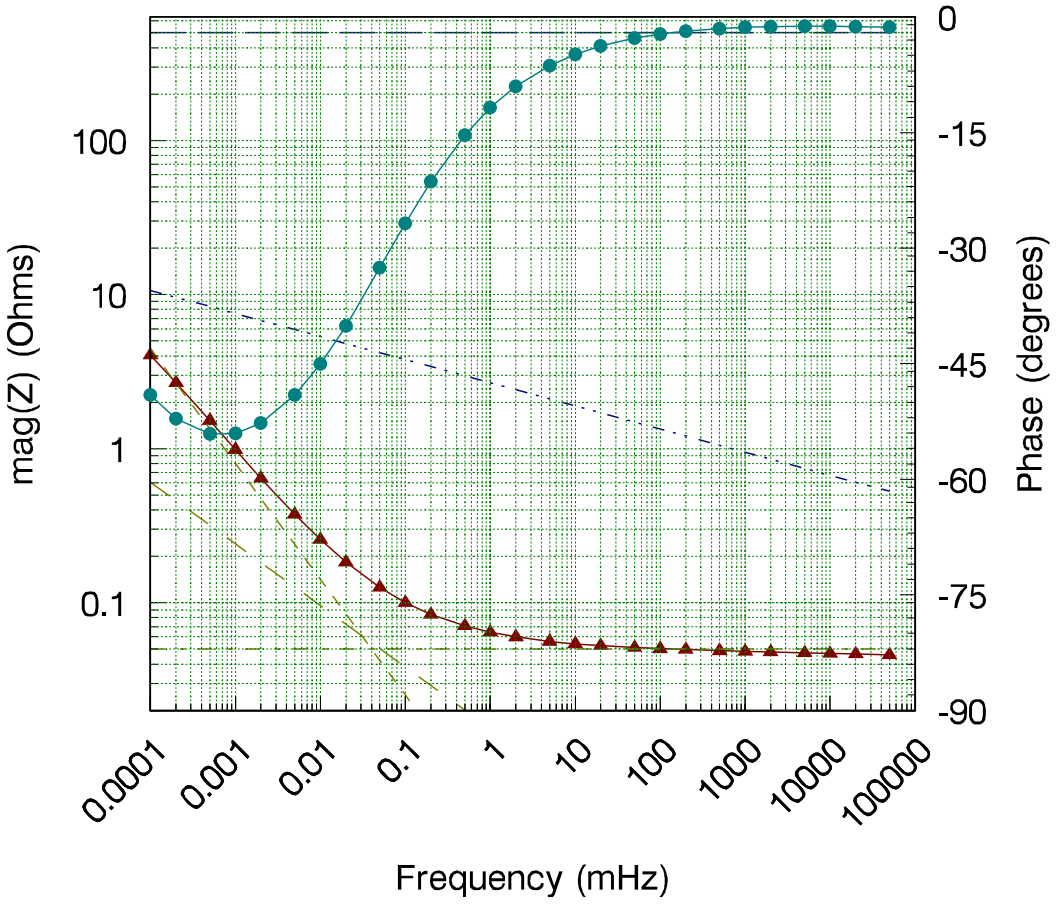


FIGURE 4.1: Plot of magnitude (red triangles) and phase (cyan dots) generated from the eight-parameter model. The additional straight lines represent the impedance of the five elements of the model, three CPEs plus series and parallel pure resistance.

1mHz and 1Hz, whereas the first CPE has the highest influence the straight-line region between 1Hz and 20Hz. A trained eye can detect additional cues in the impedance data which points towards the most appropriate ECM. The curling of the phase at $<1\,\mu$Hz suggests a parallel element could be present (Rp or CPEp), and the rounded corner at $500\,\mu$Hz implies a fractional series element is present (CPE).

## 4.2 Ideal Simulated Data

Each ECM (Figure 3.3) was fitted to the noise free simulated data and the RMSE from each fit is shown in Figure 4.2. The outcome is as expected; only the correct model produces the anticipated RMSE of effectively zero (Table 4.2 shows it is actually 0.000001, which is the input tolerance for the fitting algorithm causing it to converge). However, it is also true that the parallel elements have a far smaller effect on the fit than the three series elements. Also, observe that the fit is substantially improved when the

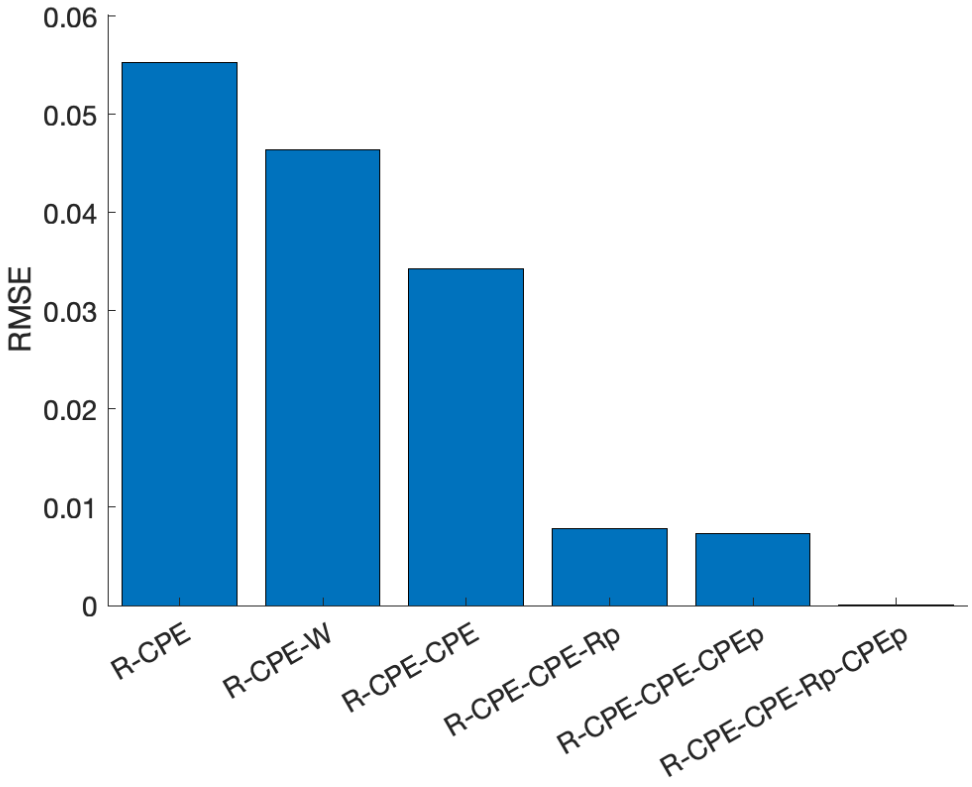


FIGURE 4.2: RMSE obtained fitting each equivalent-circuit models to ideal (noise free) simulated data.

Warburg is allowed to become a CPE, which may not be expected given the location of the Warburg's asymptote in Figure 3.3.

## 4.3 Simulated Data with Noise

Noise was then added to the simulated data using Matlab's in-built random number generator. The noise is added as a percentage, with the increments being 1%, 3% and 5%. The noise was added, or subtracted, to each phase and magnitude value of the noise free simulated data. The noise was added using 30 different seeds, producing 30 fmp files (for each 1%, 3% and 5% of noise). Each set of 30 was optimised giving 30 RMSE values, which was then averaged to give the mean RMSE. This is to ensure a range of possible noise is taken into account. The standard deviation of the mean RMSE is calculated to give error bars for each RMSE. The outcomes of this method of adding noise at 1%, 3% and 5% are shown in Figure 4.3. For comparison, the noisy results are presented next to the noise-free data from Figure 4.2. As anticipated, adding noise raises the RMSE in each case.

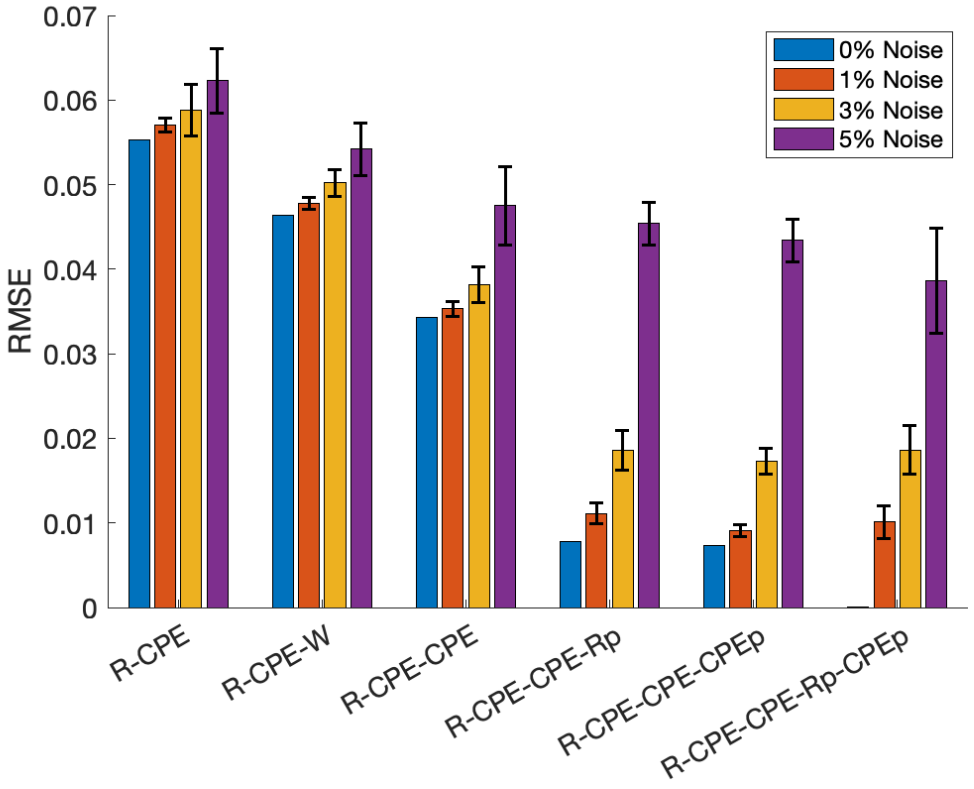


FIGURE 4.3: Graph of the RMSE obtained fitting each equivalent-circuit models to 0%, 1%, 3% and 5% noise simulated data. The height of the bar represents the average RMSE and error bars on each bar represent one standard deviation of each fit.

Observe that the RMSE does vary for a given noise contribution, however the same general shape is seen. The RMSE is at its highest for the R-CPE model, then reduces until it plateaus from the R-CPE-CPE-Rp model onwards. For this thesis it is assumed, as it was for Berthier et al [15], that two ECMs are indistinguishable if their RMSE error bars overlap. For example, from the 3% noise data in Figure 4.3 it can be seen that the R-CPE-CPE-Rp, R-CPE-CPE-CPEp and R-CPE-CPE-Rp-CPEp all have overlapping error bars. Thus they are statistically indistinguishable from one another. The graph displays how the increased complexity of the model (due to the increased number of components) is not justified by the quality of the data.

The final RMSE value indicates how well each model fits the data. For ideal data (noise free), the RMSE should tend towards zero if the ECM is a good fit. For data with noise, the RMSE should tend towards the noise level i.e. if the data had 5% noise the RMSE would tend towards 0.05 if the ECM is a good fit. It is clear to see that the first three ECMs are inadequate compared to the last three. The last three ECMs typically have an RMSE that is similar to the noise added, however all of their error bars overlap, so what does this mean for which ECM is the 'best' fit?

For <5% noise, the most restrictive of the final three models (R-CPE-CPE-Rp) should be chosen as the best representative ECM. This conclusion is drawn for both 1% and 3% additional noise. In other words, it is hard to tell the difference between the final three once noise is as low as 1%. Given the RMSE with incorrect models but operating with noise-free data, it can be anticipated that the last three models will become indistinct when noise reaches about half a percentage point.

For the data with 5% noise, it can be seen that only the first two ECMs are distinguishable from one another. The last four fail this test, and so it can be said that distinguishability between ECMs is favoured in data with low noise.

TABLE 4.2: Table of RMSE and Standard Deviation (STD) values for each ECM with the corresponding added noise

| ECM | Noise Added | RMSE | STD |
|---|---|---|---|
| R-CPE | 0% | 0.0553 | 0 |
| | 1% | 0.0570 | 0.000819 |
| | 3% | 0.0588 | 0.003 |
| | 5% | 0.0623 | 0.0038 |
| R-CPE-W | 0% | 0.0464 | 0 |
| | 1% | 0.0478 | 0.000727 |
| | 3% | 0.0502 | 0.0016 |
| | 5% | 0.0542 | 0.0031 |
| R-CPE-CPE | 0% | 0.0343 | 0 |
| | 1% | 0.0353 | 0.000837 |
| | 3% | 0.0382 | 0.0021 |
| | 5% | 0.0475 | 0.0046 |
| R-CPE-CPE-Rp | 0% | 0.0078 | 0 |
| | 1% | 0.111 | 0.0012 |
| | 3% | 0.0186 | 0.0023 |
| | 5% | 0.0454 | 0.0025 |
| R-CPE-CPE-CPEp | 0% | 0.0073 | 0 |
| | 1% | 0.0091 | 0.000745 |
| | 3% | 0.173 | 0.0016 |
| | 5% | 0.0434 | 0.0025 |
| R-CPE-CPE-Rp-CPEp | 0% | 0.000001 | 0 |
| | 1% | 0.101 | 0.0019 |
| | 3% | 0.0816 | 0.0029 |
| | 5% | 0.0386 | 0.0062 |

## 4.4 Measured Data

The impedance magnitude and phase data, obtained from actual battery measurements, were subjected to the modelling procedure outlined above. A modified version of the approach originally explained by Scott and Hasan [11] was used to take measurements.

A two-quadrant power supply (model 66332A dynamic measurement source) was used to deliver a multiple-sinewave current with multiple frequencies ranging from 0.5 Hz to 2Hz. The software for the Raspberry Pi 4 was created in-house, written in C, and connected via a Prologix-compatible open-source GPIB interface. The software can be set up to distribute current or charge displacement in a variety of ways across all frequencies.

Impedance, $Z$, of an electrochemical system around some steady or quasi-steady state can be determined by:

1. applying a small-signal multitonal sinusoidal current, where $I(t) = |I|e^{j(\omega t+\phi_I)}$;
2. measuring the voltage response $V(t) = |V|e^{j(\omega t+\phi_V)}$;
3. calculating $Z(\omega) = \frac{|V|}{|I|}e^{j(\phi_V-\phi_I)}$) [51].

$Z(\omega)$ is made up of real and imaginary parts:

$$Z(\omega) = Z_0\cos(\phi) + jZ_0\sin(\phi)$$

where $Z'_{real} = Z_0\cos(\phi)$, the resistance of the system, and $Z''_{imag} = Z_0\sin(\phi)$, capacitance and/or inductance, representing energy storage [52]. In this thesis, data is displayed in Bode plots which present magnitude $|Z(\omega)| = |V|/|I|$ and phase $\arg(Z(\omega)) = \phi_V - \phi_I$.

Current and voltage were logged by the 66332A approximately every 0.1 s. Magnitudes and phases of both parameters at the frequencies of interest, required to permit the above calculations, were determined using a discrete Fourier transform (DFT) after the method first described by Scott and Parker [53].

### 4.4.1 NMC 18650 Cell vs NCR/NCA 21700 Cell

Measured data were obtained for a near-new Lithium Nickel Cobalt Aluminium oxide (NCA) 21700 cell and a used Lithium Nickel Manganese Cobalt Oxide (NMC) 18650 cell (Figure 4.5). The results of attempting to fit all six ECMs to these batteries appear in Figure 4.4. The associated model parameters appear in Tables 4.4 and 4.3. Observing Figure 4.4, the R-CPE-CPE model is clearly the best option for the NMC cell while the R-CPE-CPE-Rp model is the best option for the NCA cell. This measured data shows the same trend of the last three (NCA), or four (NMC), ECMs being indistinguishable from one another - affirming that the RMSE for the more sophisticated ECMs is very slightly lower, but not by a sufficient amount to justify the complexity.

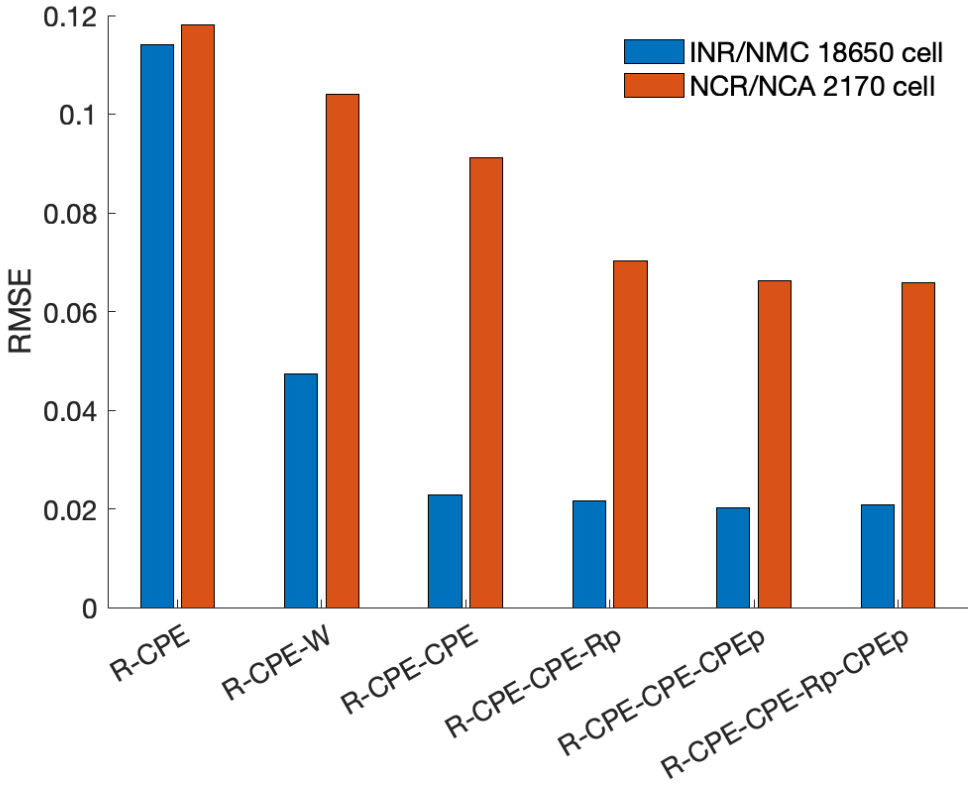


FIGURE 4.4: RMS error obtained fitting each ECM to the two data sets presented in Figure 4.5. The left-hand blue bars correspond to the NMC cell.

TABLE 4.3: ECM Parameters for NCA 2170 cell

| Parameter | $R_s$ | $\alpha$ | $C_F$ | $C_W$ | $\alpha 2$ | $R_p$ |
|---|---|---|---|---|---|---|
| Value | 0.0503 | 0.99 | 22,230 | 230 | 0.272 | 44 |

TABLE 4.4: ECM Parameters for NMC 18650 cell

| Parameter | $R_s$ | $\alpha$ | $C_F$ | $C_W$ | $\alpha 2$ |
|---|---|---|---|---|---|
| Value | 0.0330 | 0.99 | 14,180 | 187 | 0.27 |

### 4.4.2 INR Lithium Battery

Measured data were obtained for an 160mAh INR Lithium Battery and can be seen in Figure 4.6. The RMSE from the fit can be seen in Figure 4.7. Even with a large amount of noise present in the measured data, the ECMs are distinguishable up to the R-CPE-CPE-Rp model. Following the trend seen above, the last two models give an improvement on the RMSE but not by enough to justify their complexity.

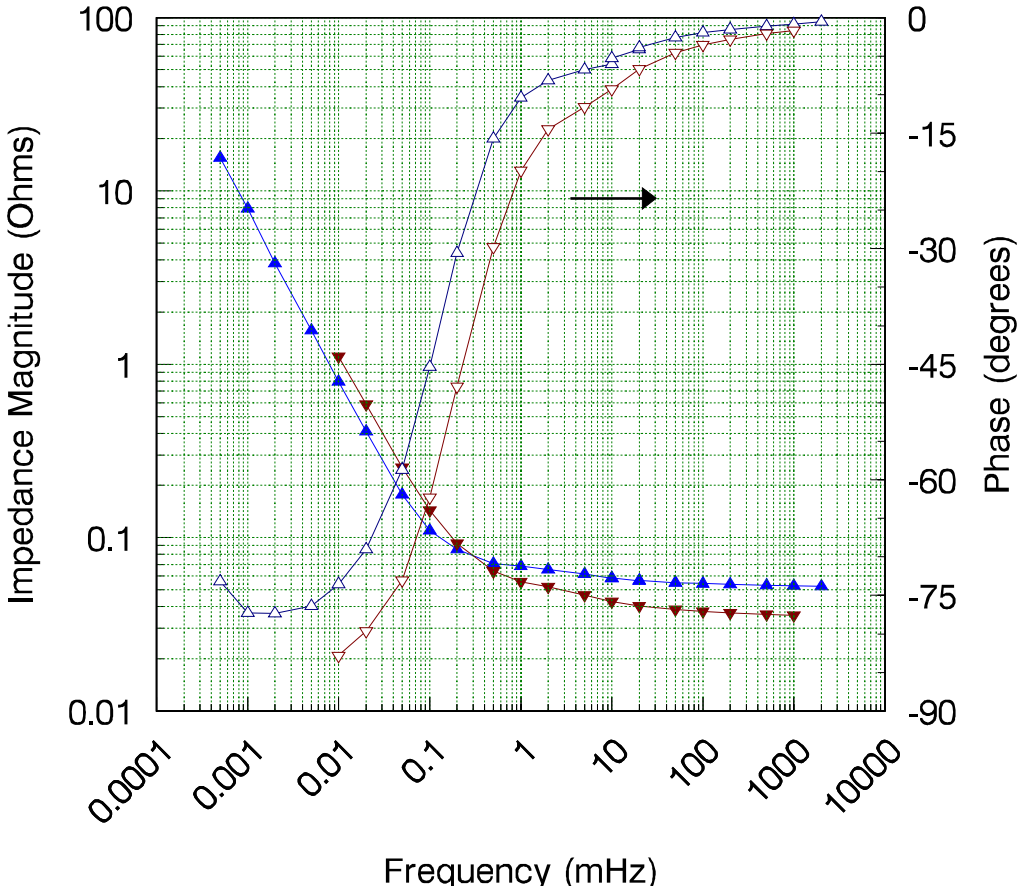


FIGURE 4.5: Plot of measured impedance magnitude (solid blue up triangles) and phase (open blue up triangles) of a lithium nickel cobalt aluminium oxide (NCR/NCA) 2170 cell and magnitude (solid red down triangles) and phase (open red down triangles) measured on a lithium nickel manganese cobalt oxide (INR/NMC) 18650 cell.

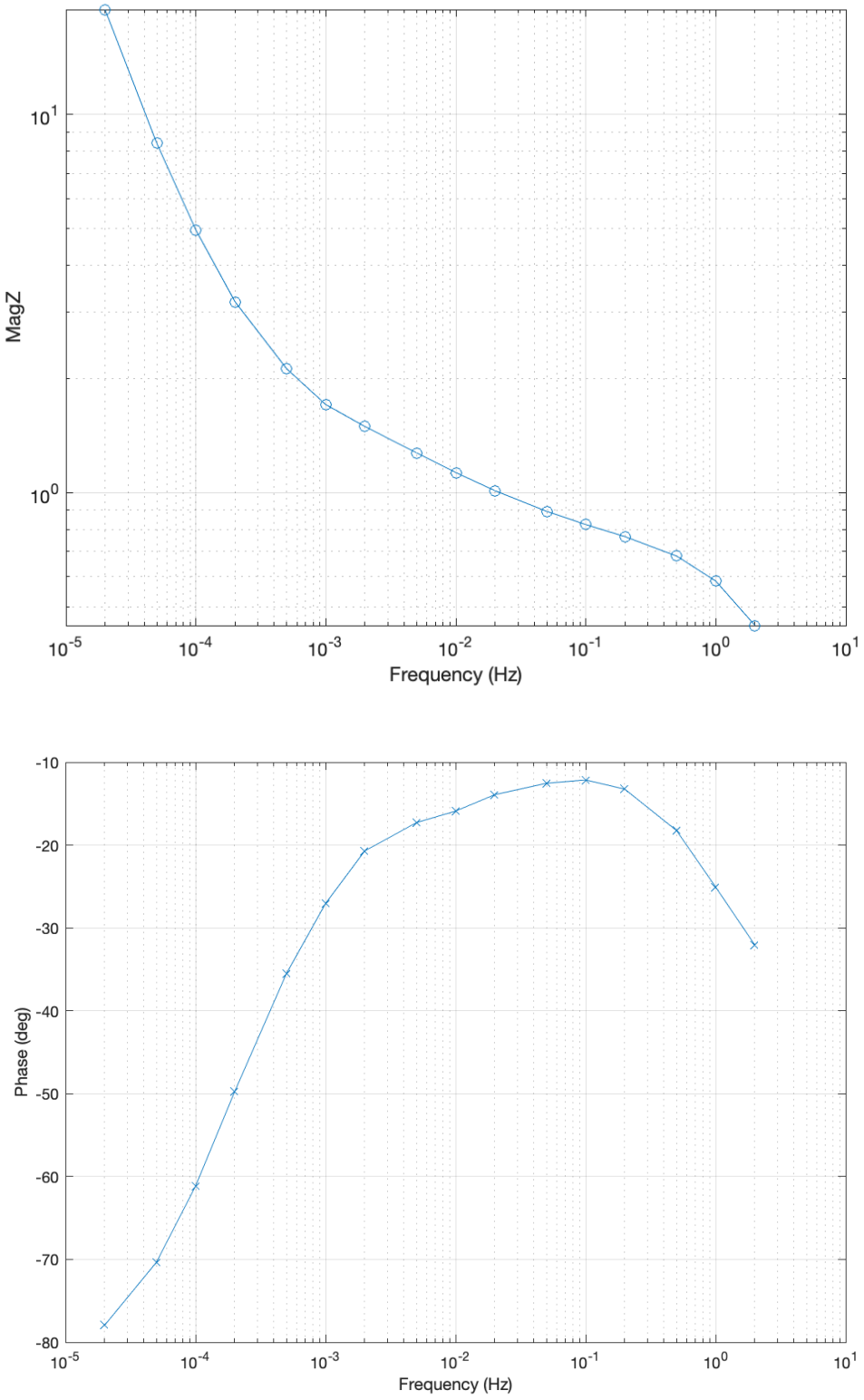


FIGURE 4.6: Plot of measured impedance magnitude (top graph) and phase (bottom graph) of an 160mAh INR Lithium Battery

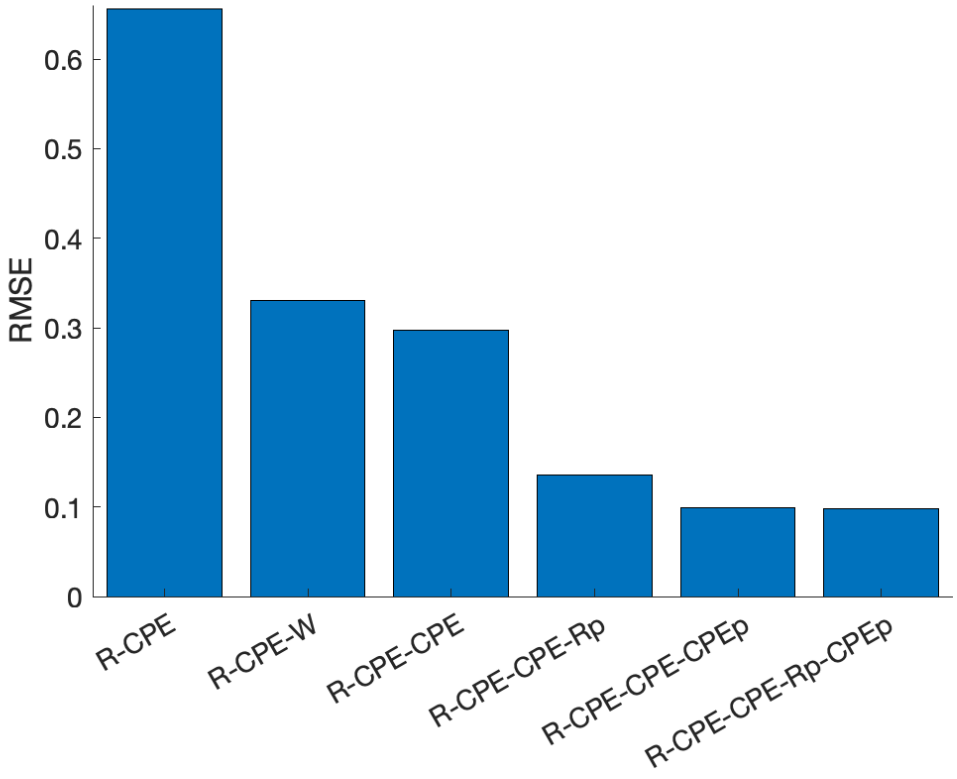


FIGURE 4.7: Plot of the RMSE from fitting each ECM to a 160mAh INR Lithium Battery

# Chapter 5

# Discussion

## 5.1 Time Domain Measurements & Comparison

Vance Farrow, a PhD student in the battery management team at the University of Waikato, is working on battery SoH. Farrow is in the process of publishing a paper showing a passive, fractional ECM in the time and frequency domains and their implications for battery SoH [54]. Farrow uses the Riemann-Liouville integral [55] to produce time domain battery voltage predictions from parameters found from EIS optimisation methods used in this thesis.

Farrow [54] measured the voltage, and EIS data, of a 2.5 Ah 18650 Lithium-Cobalt Oxide (INR) cell. The stimulus current used for measurement can be seen in Figure 5.1. The current was chosen to simulate a simplified 12 day cycle of an electric car battery.

The measured voltage was recorded, and the EIS data was filed for analysis. Using Farrows time domain prediction software is a key to demonstrating the validity of the work in this thesis. The workflow for the validity is outlined below:

- Measure the battery data to gain the voltage response and EIS data
- Optimise parameters for the R-CPE-CPE model against the measured EIS data
- Use the optimised parameters in Farrows [54] time domain prediction code to give the predicted voltage response for the optimised circuit
- Compare the measured voltage response to the optimised set to see if the optimised parameters produce a voltage response that is the same shape, and is similar to, the measured data.

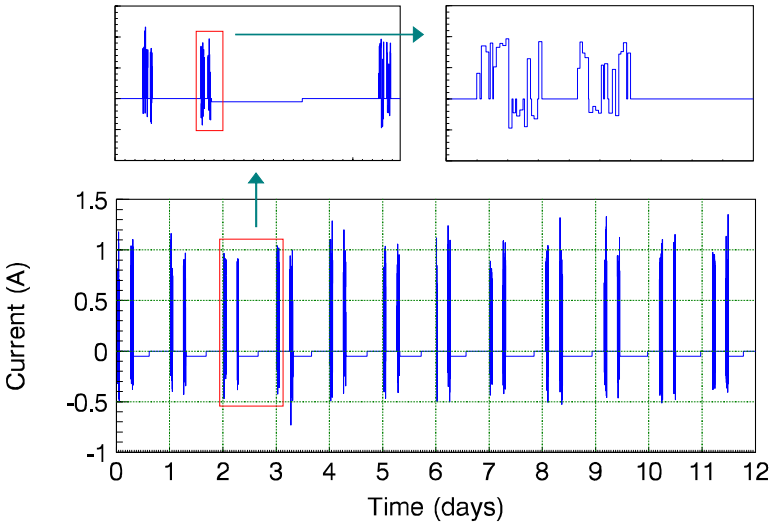


FIGURE 5.1: The stimulus current is shown here, simulating a simplified version of vehicle current over a 12 day cycle. The top two graphs are zoomed in sections of the bottom for clarity.

This process was carried out and the results are shown in Figure 5.2, with the error between the measured and optimised data plotted in Figure 5.3. From these two figures, it can be seen that the optimised prediction fits exceedingly well to the measured data in the time domain. There are discrepancies at each steady state peak, shown by the gap between the orange and blue traces in Figure 5.2. However, upon further inspection in Figure 5.3 it can be seen that the error only fluctuates between 20 to 40 mV. That equates to approximately a 1% error between the measured and optimised data in the worst case scenario. This error value is lower than typical measurement error in this area meaning it does not seem to be a significant error. This shows how the optimisation carried out in this thesis aligns in the frequency and time domains for battery measurement data - this gives credibility to the work done in this thesis.

## 5.2 Further work

- Trying this method on different battery chemistries to get a better idea of the expectation of the ECM that best represents each type of battery. This could provide insights to general trends that different battery chemistries exhibit i.e. Is one batter chemistry more likely to have higher alpha values than others?

- Repeating the model fit as a battery ages to see if, and how, the ECM parameters change. This was not included in this thesis since it would take years of measurements to compile this data. Tesla guarantees their batteries for 8 years [56], and other batteries can be guaranteed for longer. The battery management group is

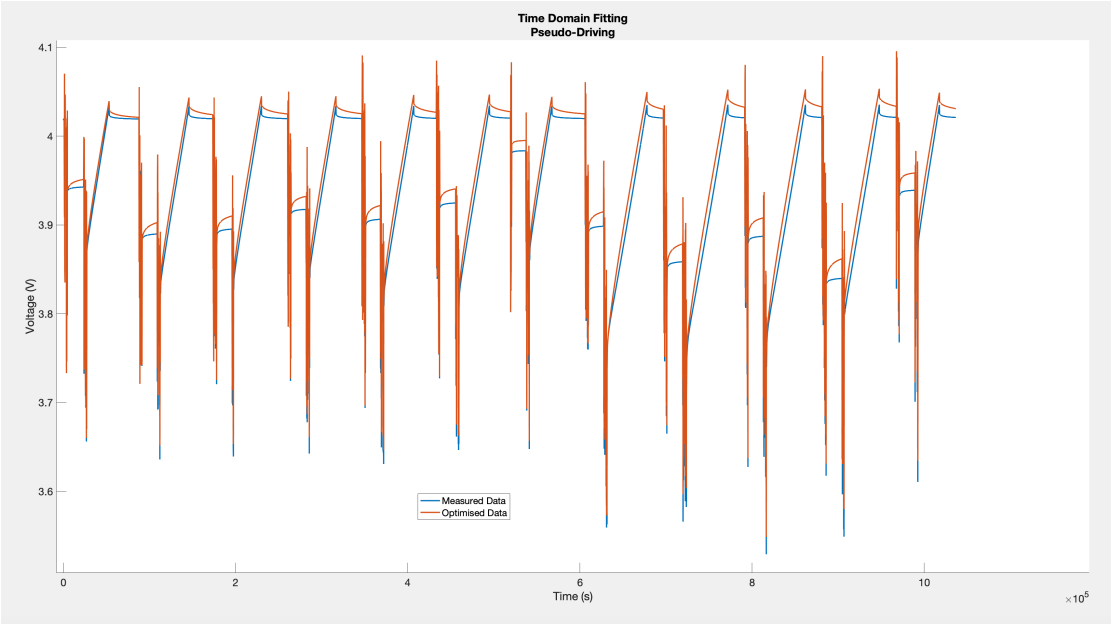


FIGURE 5.2: The blue waveform is the measured data from the 18650 INR, and the orange waveform is the time domain prediction from Farrow's software using the R-CPE-CPE optimised parameters

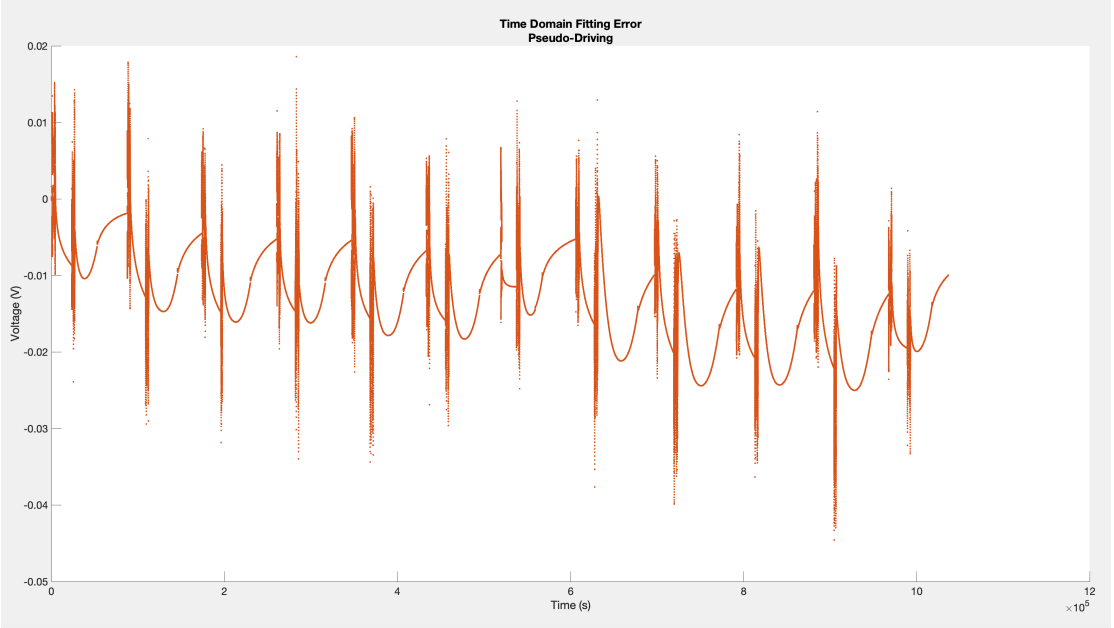


FIGURE 5.3: This shows the error as an absolute voltage value between the measured and optimised data

currently starting the 'Solar Cycler' project which cycles the batteries in solar garden flood lights (which contain 18650 Lithium batteries) daily. These solar lights will be cycled until battery failure. As the solar light battery cycles, the EIS data will be recorded to see how the EIS data changes as the batter degrades. Once the battery is at the end of its life, this work can be done.

# Chapter 6

# Conclusion

By using fractional equivalent circuit models that were motivated by recent, extra-low frequency EIS measurements, we have reviewed Berthier's work. It has been shown that using extra-low frequency data, given measurements with the amount of noise & error now available from the Dunn-Scott method [57], does allow equivalent circuit models to be distinguished from one another. Six equivalent circuit models were evaluated, each more complex than the one before it.

When optimising parameters to simulated data, the Nelder-Mead approach proved to be effective, but only in the presence of a multi-restart function. This reduced the likelihood of the RMSE getting stuck in a saddle point, and added to the robustness of the overall fit.

When fitting to simulated data with no noise, all six circuits were distinguishable from one another. At 3% noise the three most complex ECMs were indistinguishable and at 5% noise the four most complex ECMs were indistinguishable. This demonstrates that distinguishability favours low, or ideally zero, noise.

The battery type can be a key determinant on which ECM can be justified. For instance, the 18650 INR justified the R-CPE-CPE model compared to the 2170 NCA battery which justified the R-CPE-CPE-Rp model. Even noisy data, i.e. the 160mAh INR Battery, can still justify up to the R-CPE-CPE-Rp model. This highlights how differing battery capacities and chemistries can justify different ECMs, keeping in mind this can change over time (with battery ageing).

A good measurement system (Agilent 66332A, E5270, Keithley 2460A are used here) also contributes to the efficacy of the optimisation. The optimisation can only be as good as the data, it is unreasonable to expect to optimise an ECM to a smaller degree

than the error in the data. This must also be taken into account when determining which ECMs are justified.

A gap between the work of electrochemists and electrical engineers is revealed by the extensive literature on gray-box, equivalent-circuit models; the former have a keen understanding of the circuit elements, while the latter have the tools to fit and use them. Obtaining insights into which ECMs fit EIS data best will help electrochemists to associate internal battery components with branches in the ECM. This will also enable electrical engineers to advance research to detect degradation and ageing in a battery and predict failure well in advance of the event.

# Chapter 7

# Bibliography


[1] Rui Xiong, Linlin Li, and Jinpeng Tian. Towards a smarter battery management system: A critical review on battery state of health monitoring methods. *J. Power Sources*, 405:18–29, 2018.

[2] Noshin Omar, Mohamed Daowd, Omar Hegazy, Grietus Mulder, Jean-Marc Timmermans, Thierry Coosemans, Peter Van den Bossche, and Joeri Van Mierlo. Standardization work for bev and hev applications: Critical appraisal of recent traction battery documents. *Energies*, 5(1):138–156, 2012.

[3] Eden Poihipi, Jonathan Scott, and Christopher Dunn. Distinguishability of battery equivalent-circuit models containing cpes: Updating the work of berthier, diard, and michel. *Journal of Electroanalytical Chemistry*, 911:116201, 2022.

[4] P. Kurzweil. Gaston planté and his invention of the lead–acid battery—the genesis of the first practical rechargeable battery. *Journal of Power Sources*, 195:4424–4434, 07 2010.

[5] Takashi Eguro. *Ni-Cadmium Batteries*, pages 1358–1363. Springer New York, New York, NY, 2014.

[6] Mogalahalli V Reddy, Alain Mauger, Christian M Julien, Andrea Paolella, and Karim Zaghib. Brief history of early lithium-battery development. *Materials (Basel)*, 13(8), Apr 2020.

[7] Andrzej Lasia. *Electrochemical Impedance Spectroscopy and its Applications*. Springer New York, New York, NY, 2014.

[8] Zurich Instruments. Mfia 500 khz / 5 mhz impedance analyzer.

[9] Hioki Products & Services. Chemical impedance analyzer im3590.

[10] Branko Grünbaum. *Convex Polytopes, Euler's Relation.* Number 146-160. Springer New York, NY, 2003.

[11] Jonathan Scott and Rahat Hasan. New Results for Battery Impedance at Very Low Frequencies. *IEEE Access*, 7:106925–106930, 2019.

[12] R. Morrison. RC Constant-Argument Driving-Point Admittances. *IRE Trans. Circuit Theory*, 6(3):310–317, 1959.

[13] Sinduja Seshadri and Jonathan Scott. Correction to "Compact Nonlinear Model of an Implantable Electrode Array for Spinal Cord Stimulation" [Jun 14 382-390]. *IEEE Trans. Biomed. Circuits Syst.*, 12(4):963–964, 2018.

[14] Marcus Wilson, Logan Cowie, Vance Farrow, Michael Cree, and Jonathan Scott. Simulating fractional capacitors with the spice circuit simulator. *IEEE Trans. Ind. Electron. PRE-PRINT*, 2023.

[15] F. Berthier, J.-P. Diard, and R. Michel. Distinguishability of equivalent circuits containing CPEs: Part I. Theoretical part. *J. Electroanal. Chem.*, 510(1):1–11, 2001.

[16] R. Fletcher and C. Xu. Hybrid Methods for Nonlinear Least Squares. *IMA J. Numer. Anal.*, 7(3):371–389, 1987.

[17] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *Comput. J.*, 7(4):308–313, January 1965.

[18] Andrzej Lasia. Impedance of the Faradaic Reactions in the Presence of Mass Transfer. In *Electrochemical Impedance Spectroscopy and its Applications*, pages 85–125. Springer New York, New York, NY, 2014.

[19] Rahat Hasan and Jonathan Scott. Extending Randles's Battery Model to Predict Impedance, Charge–Voltage, and Runtime Characteristics. *IEEE Access*, 8:85321–85328, 2020.

[20] P. Mauracher and E. Karden. Dynamic modelling of lead/acid batteries using impedance spectroscopy for parameter identification. *J. Power Sources*, 67(1-2):69–84, 1997.

[21] J. E. B. Randles. Kinetics of rapid electrode reactions. *Discuss. Faraday Soc.*, 1:11, 1947.

[22] G.J. Brug, A.L.G. van den Eeden, M. Sluyters-Rehbach, and J.H. Sluyters. The analysis of electrode impedances complicated by the presence of a constant phase element. *J. Electroanal. Chem. Interf. Electrochem.*, 176(1-2):275–295, 1984.

[23] Svante Westerlund. Dead matter has memory! *Phys. Scr.*, 43(2):174–179, 1991.

[24] S. Westerlund and L. Ekstam. Capacitor theory. *IEEE Trans. Dielect. Electr. Insul.*, 1(5):826–839, 1994.

[25] Uwe Westerhoff, Kerstin Kurbach, Frank Lienesch, and Michael Kurrat. Analysis of Lithium-Ion Battery Models Based on Electrochemical Impedance Spectroscopy. *Energy Technol.*, 4(12):1620–1630, 2016.

[26] Larasmoyo Nugroho and Rini Akmeliawati. Comparison of black-grey-white box approach in system identification of a flight vehicle. *Journal of Physics Conference Series, IOP Science*, 1130(1), 2018.

[27] J. Brucker, W. G. Bessler, and R. Gasper. Grey-box modelling of lithium-ion batteries using neural ordinary differential equations. *Energy Informatics*, 4(15), 2021.

[28] Monowar Hossain, S. Saha, M. E. Haque, M.T Arif, and AMT Oo. A parameter extraction method for the thevenin equivalent circuit model of li-ion batteries. *2019 IEEE Industry Applications Society Annual Meeting*, pages 1–7, 2019.

[29] Lijun Zhang, Hui Peng, Zhansheng Ning, Zhongqiang Mu, and Changyan Sun. Comparative research on rc equivalent circuit models for lithium-ion batteries of electric vehicles. *Applied Sciences*, 7(10), 2017.

[30] Hyun Woo You, Jun Ik Bae, So Jeong Cho, Jong Moon Lee, and Se-Hun Kim. Analysis of equivalent circuit models in lithium-ion batteries. *AIP Advances*, 8(12):125101, 2018.

[31] S. Nejad, D.T. Gladwin, and D.A. Stone. A systematic review of lumped-parameter equivalent circuit models for real-time estimation of lithium-ion battery states. *Journal of Power Sources*, 316:183–196, 2016.

[32] Mohamed Assaad Hamida, Ragab A. El-Sehiemy, Ahmed R. Ginidi, Ehab Elattar, and Abdullah M. Shaheen. Parameter identification and state of charge estimation of li-ion batteries used in electric vehicles using artificial hummingbird optimizer. *Journal of Energy Storage*, 51:104535, 2022.

[33] Yi-Hsien Chiang, Wu-Yang Sean, and Jia-Cheng Ke. Online estimation of internal resistance and open-circuit voltage of lithium-ion batteries in electric vehicles. *Lancet*, 196:3921–3932, 04 2011.

[34] Xuning Feng, Caihao Weng, Minggao Ouyang, and Jing Sun. Online internal short circuit detection for a large format lithium ion battery. *Applied Energy*, 161:168–180, 01 2016.

[35] M. Mathew, Q.H. Kong, J. McGrory, and M. Fowler. Simulation of lithium ion battery replacement in a battery pack for application in electric vehicles. *Journal of Power Sources*, 349:94–104, 2017.

[36] Yong Tian, Dong Li, Jindong Tian, and Bizhong Xia. State of charge estimation of lithium-ion batteries using an optimal adaptive gain nonlinear observer. *Electrochimica Acta*, 225:225–234, 01 2017.

[37] Jiahao Li, Joaquin Klee Barillas, Clemens Guenther, and Michael A. Danzer. A comparative study of state of charge estimation algorithms for lifepo4 batteries used in electric vehicles. *Journal of Power Sources*, 230:244–250, 2013.

[38] Bizhong Xia, Zhen Sun, Ruifeng Zhang, and Zizhou Lao. A cubature particle filter algorithm to estimate the state of the charge of lithium-ion batteries based on a second-order equivalent circuit model. *Energies*, 10(4), 2017.

[39] Rahat Hasan and Jonathan Scott. Application of Swingler's method for analysis of multicomponent exponentials with special attention to non-equispaced data. In *2016 IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 12–15, Melaka, Malaysia, March 2016. IEEE.

[40] Mikal Cugnet, Jocelyn Sabatiera, Stephane Laruelle, Sylvie Grugeon, Bernard Sahut, Alain Oustaloup, and Jean-Marie Tarascon. On lead acid battery resistance and cranking-capability estimation. *IEEE Transactions on Industrial Electronics*, 57(3):909–917, 3 March 2010.

[41] Jocelyn Sabatier, Mathieu Merveillaut, Junior Mbala Francisco, Franck Guillemard, and Denis Porcelatto. Fractional models for lithium-ion batteries. *2013 European Control Conference (ECC)*, pages 3458–3463, July 2013.

[42] Bhaskar Saha, Kai Goebel, Scott Poll, , and Jon Christophersen. Prognostics methods for battery health monitoring using a bayesian framework. *IEEE Transactions on Instrumentation and Measurement*, 58(2):291–296, February 2009.

[43] Arijit Guha and Amit Patra. Online estimation of the electrochemical impedance spectrum and remaining useful life of lithium-ion batteries. *IEEE Transactions on Instrumentation and Measurement*, 67(8):1836–1849, 2018.

[44] Hicham Chaoui, Navid Golbon, Imad Hmouz, Ridha Souissi, and Sofine Tahar. Lyapunov-based adaptive state of charge and state of health estimation for lithium-ion batteries. *IEEE Transactions on Industrial Electronics*, 62(3):1610–1618, 2015.

[45] B. Sohlberg and E. W. Jacobsen. Grey box modelling – branches and experiences. *IFAC Proceedings Volumes*, 41(2):11415–11420, 2008.

[46] Tom T. Hartley, Jean-Claude Trigeassou, Carl F. Lorenzo, and Nezha Maamri. Energy Storage and Loss in Fractional-Order Systems. *J. Comput. Nonlinear Dyn.*, 10(6):061006, 2015.

[47] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998.

[48] K. I. M. McKinnon. Convergence of the nelder–mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.

[49] Carl T Kelley. *Iterative method for optimization*, volume Section 8.1. SIAM, 1999.

[50] Bret Dahme. Nelder-mead simplex solver with robost input options. Matlab FIle Exchange, June 2020.

[51] Changfu Zou, Lei Zhang, Xiaosong Hu, Zhenpo Wang, Torsten Wik, and Michael Pecht. A review of fractional-order techniques applied to lithium-ion batteries, lead-acid batteries, and supercapacitors. *J. Power Sources*, 390:286–296, 2018.

[52] Woosung Choi, Heon-Cheol Shin, Ji Man Kim, Jae-Young Choi, and Won-Sub Yoon. Modeling and Applications of Electrochemical Impedance Spectroscopy (EIS) for Lithium-ion Batteries. *J. Electrochem. Sci. Technol*, 11(1):1–13, 2020.

[53] Jonathan Scott and Anthony Parker. Distortion analysis using SPICE. *J. Audio Eng. Soc.*, 43(12):1029–1040, December 1995.

[54] Vance Farrow, Michael Cree, Jonathan Scott, and Marcus Wilson. Passive, fractional, battery equivalent-circuit model in time and frequency domains. Prepublishing but will be published by mid 2023. As a part of the Battery Management Group, I have been a part of seeing this paper come together., 2023.

[55] Abdon Atangana. *Fractional Operators and Their Applications*, pages 79–112. 01 2018.

[56] Tesla. Vehicle warranty, tesla support.

[57] Christopher Dunn and Jonathan Scott. Achieving reliable and repeatable electrochemical impedance spectroscopy of rechargeable batteries at extra-low frequencies. *IEEE Trans. Instrum. Meas. PRE-PRINT*, 2022.

# Appendix A

# Distinguishability of Battery Equivalent-Circuit Models Containing CPEs: Updating the Work of Berthier, Diard, and Michel

Journal of Electroanalytical Chemistry 911 (2022) 116201

Contents lists available at ScienceDirect

Journal of Electroanalytical Chemistry

journal homepage: www.elsevier.com/locate/jelechem

# Distinguishability of Battery Equivalent-Circuit Models Containing CPEs: Updating the Work of Berthier, Diard, & Michel

Eden Poihipi, Jonathan Scott, Christopher Dunn

*University of Waikato, Private Bag 3105, Hamilton 3240, New Zealand*

ARTICLE INFO

*Keywords:*
Constant Phase Elements
Equivalent Circuit Model
Battery Model
Frequency Response
EIS

ABSTRACT

This manuscript revisits the assertion of Berthier et al. that competing fractional-element equivalent-circuit models of battery cells are indistinguishable in the presence of noise. Starting with Westerhoff's general equivalent-circuit model (ECM) of 2016 and an idealized impedance spectrum of a lithium battery, a contemporary set of possible ECMs are chosen. Their distinguishability is investigated. Using the extended frequency range recommended by Mauracher & Karden in 1997 or Hasan & Scott in 2019, an approach is presented that permits selection of the appropriate model, even in the presence of noise. For the given frequency range, models with up to five elements or eight real parameters are studied. Fitting to measured data with straightforward numerical methods and choosing the most primitive appropriate sub-circuit is shown to reproduce the data. Typically a three-element, five-parameter ECM is shown to model real, measured data with precision comparable to individual sample error.

## 1. Introduction

Electrochemists seek the equivalent-circuit model (ECM) of a battery to characterize its internal processes, while electrical engineers seek the same model in order to predict state-of-charge (SoC) and state-of-health (SoH). There is today a vast amount of literature in which models are presented and applied.

Berthier et al. [1] presented a paper in 2001 that set out to determine from an electrochemical perspective whether ECMs containing constant phase elements (CPEs) were numerically distinguishable from one another. These authors used CPEs in models based on electrochemical impedance spectroscopy (EIS) data, and explored their hypothesis by numerically fitting four arbitrary ECMs to synthesized EIS data using the optimization method of Fletcher et al. [2]. They concluded that the ECMs are distinguishable provided that the synthesized EIS data are noise-free. Each circuit was further optimized against noisy synthesized EIS data using 0%, 1%, 2.5% and 5% noise. The authors concluded that circuits became indistinguishable from one another once the noise level exceeded 1%.

Berthier's group used angular frequencies from 100radians/s (16Hz) to 1Mrad/s (160kHz), and displayed results in the form of Argand diagrams (sometimes called Nyquist plots). These frequency bounds and mode of presentation impose significant limitations. Moreover, optimization was applied without delving into details of the process, and equally for each ECM irrespective of its complexity.

Batteries in appliances such as cellphones, computers, and vehicles are commonly cycled as infrequently as once a week or longer, and certainly no more often than every few minutes. These periods equate roughly to frequencies ranging from 1 $\mu$Hz to 1mHz, or equivalently C-rates from 0.006 to 4. In the case of rechargeable batteries, this implies the need to measure at extra low frequencies (ELFs) to encompass typical usage patterns [3–5].

This manuscript aims to update the work of Berthier and colleagues, with use of an expanded frequency range and presentation of data in Bode plot form. This offers advantages where frequency information is important [6]. In addition, a guided fitting process will be used, and a progressive set of ECMs will be drawn from recent literature [7,8].

## 2. Battery Equivalent-Circuit Models

Westerhoff and colleagues [7] suggested that there are three ways to characterize a battery, referred to as 'black,' 'gray' and 'white' box models. White box is a theoretical modelling method which returns to fundamental physics and a detailed understanding of the physical chemistry to determine behavior potentially leading to an ECM [9]. For example, [10] starts with white-box methods, than transitions to grey-box models through the fitting of an ECM. Black box modelling, on the other hand, employs a blind empirical modelling method using

https://doi.org/10.1016/j.jelechem.2022.116201
Received 16 December 2021; Accepted 2 March 2022
Available online 5 March 2022

sets of measured data to extract a battery model [11]. This method works best for linear relationships and is not often used on its own [9] since model extraction requires large, accurate, data sets and computationally intensive software [11]. Gray box modelling is the combination of the White and Black methods, using both theoretical understanding of the system and measured data for a more reliable model [10,11].

Westerhoff and colleagues [7] described 'gray' box modeling, in which quantitative ECMs were generated on the basis of experimental data. Westerhoff et al. presented a general battery model (Fig. 1). Like Berthier 15 years before him, Westerhoff includes CPEs, sometimes identified as "fractional capacitors", in his model. Since Randles in 1947 [12], electrochemists have acknowledged the presence of these fractional-derivative functions in their models. Conversely, the engineering literature contains many ECM models that are based at best on second-(integral)-order models [13], despite it having been shown that they are not appropriate[14]. The separate electrical and electrochemical components in a battery are represented by individual circuits, each connected in series with the others. The manuscript suggested that this general model can be simplified leaving fewer sub-circuits while continuing to represent the characteristics of a battery, depending upon the significance of each component to the whole impedance. The authors then compared the output impedance spectra of these sub-circuits to data obtained experimentally. They concluded that the chief determinant of the number of components required for each sub-circuit was the accuracy required for the application in question. Hence, it is possible to reduce the number of components in Fig. 1 without losing the ability to represent the impedance output of a rechargeable battery accurately.

In this manuscript six ECMs will be considered. They will be used in a progressive model fitting approach. The distinguishability and performance of each of the ECM sub-circuits is explored by numerically fitting each one to simulated battery data. The six ECMs appear in Fig. 2.

The first ECM reduces the Westerhoff circuit to its simplest form. This ECM contains two elements: a single CPE and a series resistor. (Fig. 2, upper left circuit.) The impedance of an ideally polarizable liquid electrode (e.g. mercury, amalgam, or indium-gallium) can be modeled by an R-C circuit, but the double-layer capacitance of a solid electrode is not purely capacitative, displays frequency dispersion, and cannot be modeled adequately by an ECM based on capacitors [15]. Instead, CPEs are used to emulate this non-ideally polarizable behavior. The equation for the impedance of a CPE is

$$Z_{CPE} = \frac{1}{C_F s^{\alpha}} \tag{1}$$

where $C_F$ is the CPE value (its pseudo- or fractional capacitance, a constant with the dimensions $Fs^{\alpha-1}$), $\alpha$ is the CPE fractional order exponent, $s = j2\pi f$(where $2\pi f = \omega$, the angular frequency), and $f$ is the frequency in hertz [16]. The total impedance of the R-CPE model can therefore be found by

$$\begin{aligned} Z_{R-CPE} &= R_s + Z_{CPE} \\ &= R_s + \tfrac{1}{C_F s^{\alpha}} \end{aligned} \tag{2}$$

where $R_s$is the series resistance and $Z_{CPE}$is the impedance of the CPE (Eq. 1) in ohms.

It is worth mentioning at this point how the particular six sub-circuits have been selected. Why these six, given the large choice afforded by starting with the circuit of Fig. 1?

The choice is guided by measured impedance data. Fig. 3 shows the magnitude and phase of impedance measured on a small selection of batteries covering different chemistries and capacities. The traces all share a basic shape associated with the base R-CPE circuit. As will become clear below, the other circuits that we have chosen to use in this study correspond to characteristics observed in at least a few measured examples.

The R-CPE model can be extended by the addition of a Warburg element (W) to become the second ECM, R-CPE-W. (Fig. 2, upper right circuit.) Warburg elements are CPEs with an alpha bound to a value of one-half. They famously appeared in the electrode reaction experiments of Randles reported in 1947 [12], and describe Faradaic impedance in the presence of redox processes in semi-infinite linear diffusion systems [6]. Their generalization to arbitrary fractional order yields the CPE described above [17]. The impedance of a Warburg element is given by

$$Z_{\text{Warburg}} = \frac{1}{C_W s^{\frac{1}{2}}} \tag{3}$$

and so the impedance of the R-CPE-W model is given by

$$\begin{aligned} Z_{\text{R-CPE-W}} &= R_s + Z_{CPE} + Z_{\text{Warburg}} \\ &= R_s + \tfrac{1}{C_F s^{\alpha}} + \tfrac{1}{C_W s^{\frac{1}{2}}} \end{aligned} \tag{4}$$

where $R_s$is the series resistance, $Z_{CPE}$is the impedance of the CPE (Eq. 1) and and $Z_{\text{Warburg}}$ is the impedance of the Warburg element (Eq. 3).

In practice, assumptions made in the theory that anticipates a Warburg element are often not completely met. The Warburg can be replaced by a CPE to give an extra degree of freedom. (Fig. 2, middle left circuit, and so forth.) The total impedance of this model is given by

$$\begin{aligned} Z_{\text{R-CPE-CPE}} &= R_s + Z_{\text{CPE1}} + Z_{\text{CPE2}} \\ &= R_s + \tfrac{1}{C_F s^{\alpha}} + \tfrac{1}{C_W s^{\alpha_2}} \end{aligned} \tag{5}$$

where $R_s$is the series resistance, and $Z_{\text{CPE1}}$and $Z_{\text{CPE2}}$are the impedances of the two CPEs, and we keep the $W$ subscript to show the origin of the element.

Readers familiar with circuit theory will appreciate that small phase deviations at high and low frequency extents suggest a parallel element in the mix. Additional components can be added in parallel with other elements in an ECM to accommodate deviations in battery phase response. The R-CPE-CPE-Rp subcircuit will have impedance

$$Z_{\text{R-CPE-CPE-Rp}} = R_s + [Z_{\text{CPE1}} + Z_{\text{CPE2}}] \| R_p \tag{6}$$

where $R_p$ is the parallel resistance. This equation can be expanded by noting that the impedance of parallel elements can be written

$$X \| Y = \frac{1}{\frac{1}{X} + \frac{1}{Y}} = \frac{XY}{X+Y} \tag{7}$$

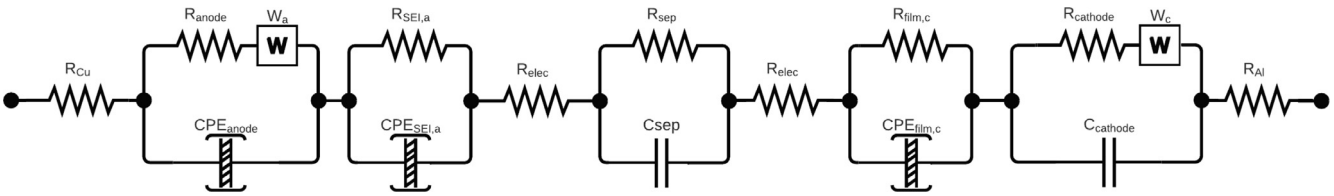


**Fig. 1.** Potential complete equivalent circuit of a battery according to Westerhoff [7]. Constant phase elements (CPEs) are used to characterize electrochemical behavior at the anode, solid electrolyte interface (SEI), separator (sep), and cathodic thin film (film,c). Note also resistances (R) associated with the copper and aluminium foils (Cu and Al), anode, SEI, electrolyte (elec), separator, thin film and cathode, and capacitances (C) at the separator and cathode. Warburg elements are denoted by $W_a$ and $W_c$ at the anode and cathode, respectively.

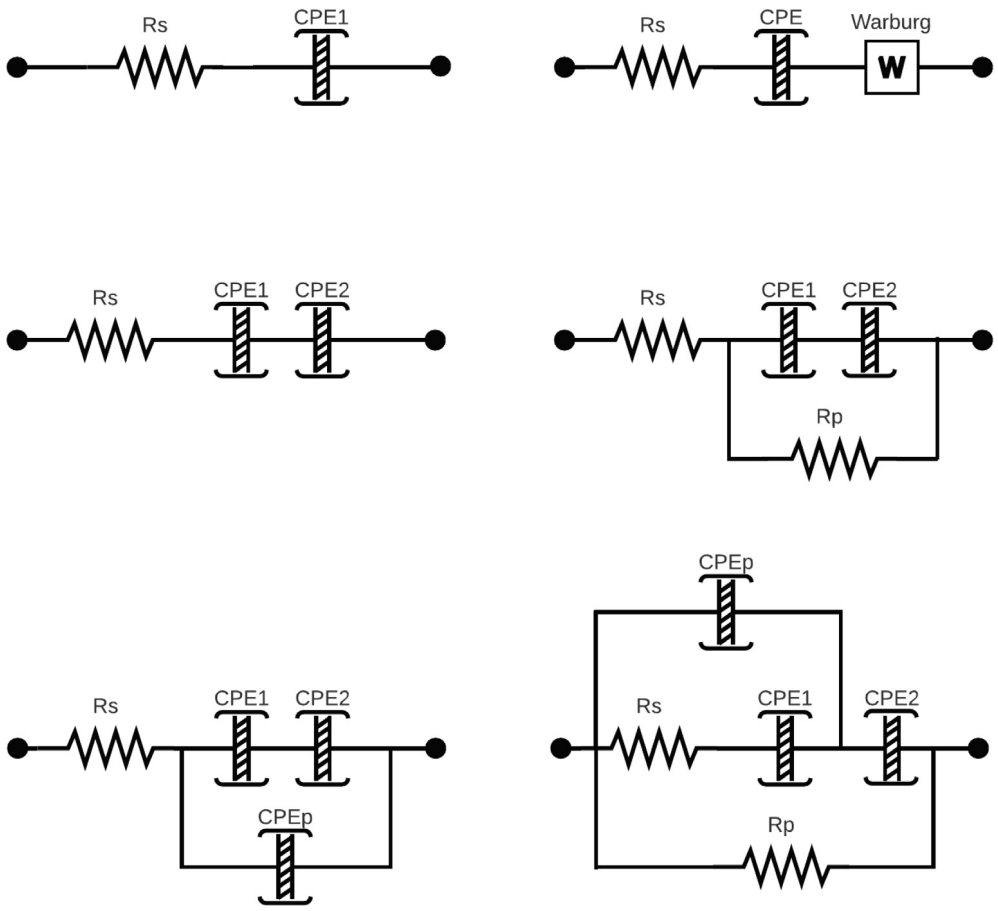


**Fig. 2.** The equivalent-circuit models (ECMs) considered in this manuscript. For ease of communication we identify them by the sequence of elements. The top left circuit can be identified as "R-CPE", the top right as "R-CPE-W", the middle left as "R-CPE-CPE", middle right as "R-CPE-CPE-Rp", then "R-CPE-CPE-CPEp" and finally "R-CPE-CPE-Rp-CPEp".

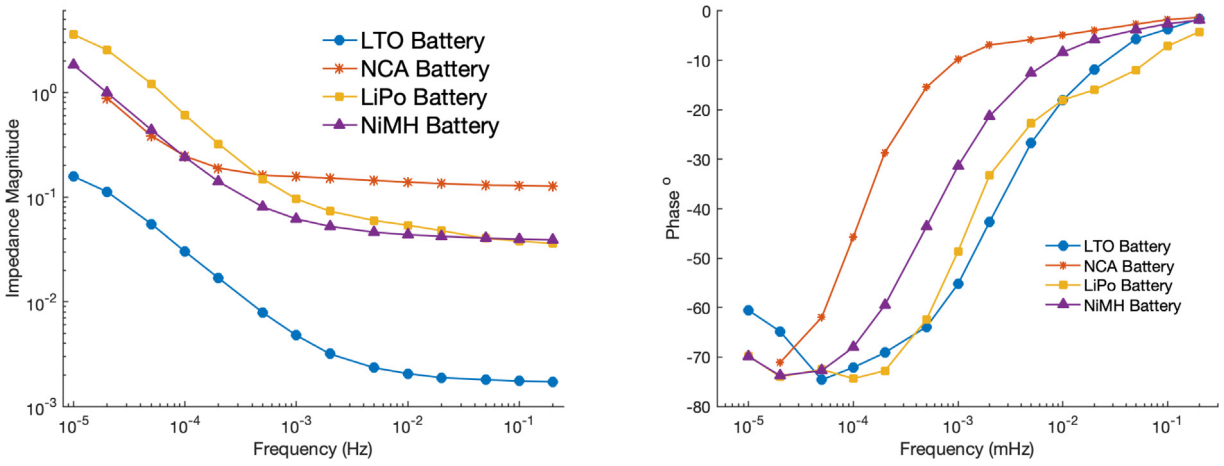


**Fig. 3.** Impedance magnitude of a selection of rechargeable batteries. LTO is a lithium titanate cell of 40Ah capacity, NCA is a 4.8Ah lithium nickel cobalt aluminium cell, LiPo is a 250mAh lithium polymer cell, and NiMH is a 2.5Ah nickel-metal hydride cell.

leading to

$$Z_{\mathrm{R-CPE-CPE-Rp}} = \frac{[R_s + Z_{\mathrm{CPE1}} + Z_{\mathrm{CPE2}}]R_p}{R_s + Z_{\mathrm{CPE1}} + Z_{\mathrm{CPE2}} + R_p} \tag{8}$$

and then

$$Z_{\mathrm{R-CPE-CPE-Rp}} = \frac{\left[R_s + \frac{1}{C_F s^a} + \frac{1}{C_W s^{a_2}}\right]R_p}{R_s + \frac{1}{C_F s^a} + \frac{1}{C_W s^{a_2} + R_p}}. \tag{9}$$

The form of (6) is usually preferred by engineers over (9) because it is more intuitive and easier to assimilate. As the expansions rapidly become more onerous we will restrict ourselves to the parallel notation from here onwards.

The final two ECMs to be investigated have impedances $Z_5$ and $Z_6$

$$\begin{aligned} Z_5 &= Z_{\mathrm{R-CPE-CPE-CPEp}} \\ &= R_s + [Z_{\mathrm{CPE1}} + Z_{\mathrm{CPE2}}] \| Z_{\mathrm{CPEp}} \\ &= R_s + \left[\frac{1}{C_F s^a} + \frac{1}{C_W s^{a_2}}\right] \| \frac{1}{C_P s^{a_P}} \end{aligned} \tag{10}$$

where the || operator is taken to have precedence over addition, $\alpha_p$ is the order of the parallel CPE, and $C_P$ the proportionality constant of the parallel CPE; and

$$\begin{aligned} Z_6 &= Z_{\mathrm{R-CPE-CPE-Rp-CPEp}} \\ &= \left[[R_s + Z_{\mathrm{CPE1}}] \| Z_{\mathrm{CPEp}} + Z_{\mathrm{CPE2}}\right] \| R_p \\ &= \left[\left[R_s + \tfrac{1}{C_F s^{\alpha}}\right] \| \tfrac{1}{C_P s^{\alpha_p}} + \tfrac{1}{C_W s^{\alpha 2}}\right] \| R_p \end{aligned} \tag{11}$$

## 3. Fitting Algorithm

Following the lead of Berthier and colleagues [1], data will be generated from one of the ECMs with varying amounts of numerical noise, and each model will be fitted to that data.

### 3.1. Numerical Optimization Method

The numerical optimizer was created in Matlab, using both the inbuilt toolboxes and previously published examples of real-parameter black box optimization algorithms [18–20]. The optimization method utilizes the downhill simplex algorithm of Nelder & Mead [21,22], a direct-search method which aims to minimize non-linear, multidimensional functions. For a function with *n*dimensions, the optimizer takes a set of $n + 1$vertices to create a simplex. The vertices then move iteratively in order to reduce the size of the simplex until a termination value is reached. The main limitation of the Nelder-Mead approximation is potential for convergence of the simplex to local minimum, called a 'saddle', rather than the global minimum.

The method is analogous to multiple skiers on a mountain. As the $n + 1$ skiers traverse downward on various paths, they all end up at the lowest point in their environment. However, each skier may end up at different locations and elevations when they reach the 'bottom'. One way to address this problem is to have multiple random starting points and then evaluate which starting point minimized the function the furthest. In some cases we applied this approach using a second program written in C. The C program could apply thousands of random multiple starts with variable range of starting points in a period of about a minute.

The root-mean-square error (RMSE) used as the optimization goal is calculated from the impedances defined in Eqs. (2)–(11) above. Impedance values are complex, and the RMSE is defined as the square root of the average of the squared distance in the complex impedance (Argand) plane. Various alternatives, weighting magnitude and phase differently, etc., were tried, but proved to offer no advantages.

### 3.2. Fitting Sequence

Choosing the appropriate model to fit given data commences with the simplest ECM, the R-CPE model. The model has only three parameters, converges quickly, and generally finds the global minimum without restarts. The final root-mean-square error (RMSE) reflects how well the model fits the data. If the model perfectly fits ideal data, the RMSE will approach zero. If the model perfectly fits data with a certain amount of random white noise, the RMSE will settle at a value close to that noise level, so that 1% added noise will result in a final RMSE of about 0.01. If the model does not fit the data, the extent of the mismatch also appears in the RMSE.

The second step of the model selection process moves to the next-simplest model, in this case the R-CPE-W model, with four parameters. The fit starts at the model parameter values returned in the R-CPE case. If the previous fit has settled at a point where the RMSE was largely defined by noise in the data, then this attempt will not improve upon the fit, and it may be concluded that the quality of the data does not justify this more complicated model, even if that model might be more correct. However, if the added element results in a lower RMSE, it must be concluded that the extended model does represent the data better.

This process is repeated for the ECMs with increasing degrees of freedom, the R-CPE-CPE model, with five parameters, in the third step, and so on. During the progressive fitting, it is possible for parameter values to be returned that are impossible. For example, $\alpha$ cannot exceed 1.0, and $C_p$ cannot be negative. Such an event signals that the model and fit cannot be appropriate for the data.

## 4. Simulated Data

A set of model parameter values were chosen for the ECM with eight degrees of freedom, the R-CPE-CPE-Rp-CPEp model. Values appear in Table 1. The magnitude and phase computed using Eq. 11 are plotted in Fig. 4. Also plotted in the figure are the asymptotes corresponding to each of the five elements. This permits the reader to associate regions in the plot with different elements; for example, the horizontal section of the magnitude plot between 1mHz and 1Hz is dominated by the series resistance, $R_s$, and the straight-line region

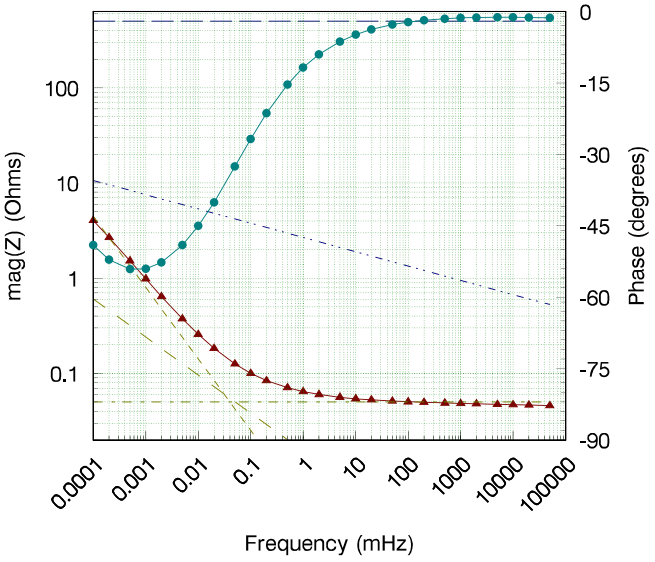


**Fig. 4.** Plot of magnitude (red triangles) and phase (cyan dots) generated from the eight-parameter model. The additional straight lines represent the impedance of the five elements of the model, three CPEs plus series and parallel pure resistance.

**Table 1**
Synthetic Model Parameter Values

| Parameter | $R_s$ | $\alpha$ | $C_F$ | $C_W$ | $\alpha 2$ |
|---|---|---|---|---|---|
| Value | 0.05 | 0.75 | 10,000 | 500 | 0.40 |
| Parameter | $R_p$ | $\alpha_p$ | $C_p$ | | |
| Value | 500 | 0.15 | 0.8 | | |

between 1 $\mu$Hz and 20 $\mu$Hz is dominated by the first CPE. A practised eye can pick up other hints about what ECM might be appropriate. For example, the curling up of the phase below 1 $\mu$Hz hints at a parallel element, and the rounded, gradual nature of the corner at 500 $\mu$Hz hints at a second fractional series element.

## 5. Model Discrimination & Selection

Fig. 5 depicts the RMSE obtained fitting each of the ECMs to ideal noiseless data computed according to Section 4. The result is not surprising; only the correct model yields the expected RMSE of zero. Nevertheless, it is also possible to say that the parallel elements are having much less impact on the fit than the three series elements. Also note that the extra degree of freedom acquired when the Warburg is allowed to become a CPE significantly improves the fit, something that might not be expected from the position of its asymptote in Fig. 4.

Adding noise at 1% and 3% leads to the results in Fig. 6. This plot contain a great deal of information. The noiseless data from Fig. 5 is reproduced adjacent to the noisy results for comparison. As expected, adding noise increases the RMSE in all cases. Error bars have been added to the noisy RMSE results. In each case 30 simulations were run with different added noise, and the mean and standard deviation (SD) of the RMSE calculated across these runs. The error bars show the $\pm 1.0$SD extents.

Observe that the RMSE results do vary somewhat for given noise contribution, but not greatly. Where different models return RMSE values that lie within the error margins, the conclusion must be that the models are indistinguishable. One way of interpreting this is to say that the quality of the data does not justify the added complexity of the model with a higher number of degrees of freedom. In the example, the R-CPE-CPE-Rp model has six degrees of freedom, the R-CPE-CPE-CPEp model seven, and the R-CPE-CEP-Rp-CPEp model eight. It is clear that the first three models (with three, four, and five degrees of freedom) all fall short of the final three. Amongst the final three, the R-CPE-CPE-Rp model, as the most constrained, is to be preferred. This conclusion is reached both in the case of 1% noise, and 3% added noise. In other words, once noise reaches as little as 1% it becomes impossible to distinguish between the final three models. Given the RMSE with incorrect models but operating with noise-free data, it can be anticipated that the last three models will become indistinct when noise reaches about half a percentage point.

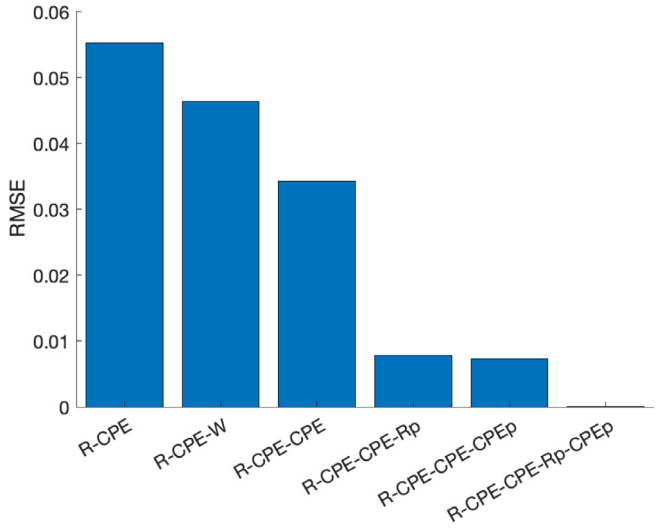


**Fig. 5.** RMS error obtained fitting the various equivalent-circuit models to ideal (noiseless) synthesized data.

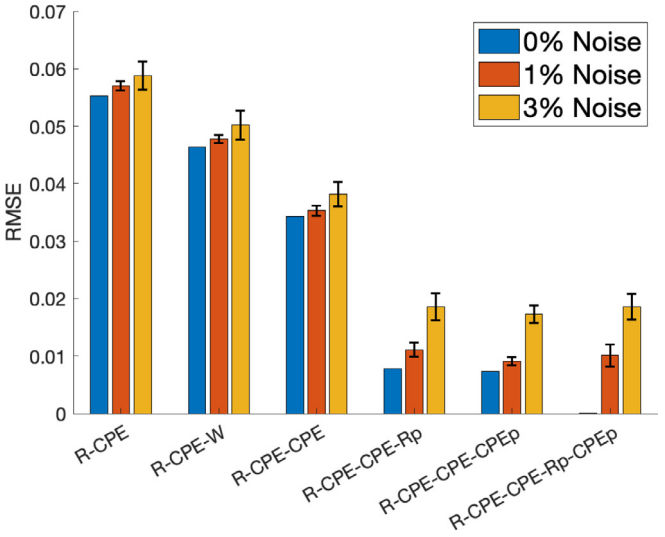


**Fig. 6.** RMS error obtained fitting the various equivalent-circuit models to ideal (noiseless), 1 percent noise, and 3 percent noise synthesized data. Error bars show 1 standard deviation in expected RMS error.

## 6. Measured Data

The modelling process described above was applied to impedance magnitude and phase data measured on real batteries. Measured data were obtained for a near-new lithium nickel cobalt aluminium oxide (referred to variously as NCA or NCR) 21700 cell and a used but healthy lithium nickel manganese cobalt oxide (referred to variously as NMC or INR) 18650 cell. Measurements are made with a developed form of the method initially described by Scott and Hasan [4]. For the present experiment, a multiple-sinewave current consisting of multiple frequencies ranging between 0.5 $\mu$Hz and 2Hz was delivered using a two-quadrant power supply (model 66332A dynamic measurement source) controlled by software developed in-house, written in C, and running on a Raspberry Pi 4 connected via a Prologix-compatible open-source GPIB interface. The 66332A delivers up to 5A of current, and has voltage and current resolutions of 5mV and 1.32mA, respectively. The software can be configured to distribute either current or charge displacement across all frequencies in various fashions.

Impedance, $Z$, of an electrochemical system around some steady or quasi-steady state can be determined by:

1. applying a small-signal multitonal sinusoidal current, where $I(t) = |I|e^{j(\omega t+\phi_I)}$;
2. measuring the voltage response $V(t) = |V|e^{j(\omega t+\phi_V)}$;
3. calculating $Z(\omega) = \frac{|V|}{|I|}e^{j(\phi_V-\phi_I)}$)[23].

$Z(\omega)$ is made up of real and imaginary parts:

$Z(\omega) = Z_0\cos(\phi) + jZ_0\sin(\phi)$where $Z\prime_{real} = Z_0\cos(\phi)$, the resistance of the system, and $Z\prime\prime_{imag} = Z_0\sin(\phi)$, capacitance and/or inductance, representing energy storage [8]. In this manuscript, we display Bode plots which present magnitude $|Z(\omega)| = |V|/|I|$ and phase $\arg(Z(\omega)) = \phi_V - \phi_I$.

Current and voltage were logged by the 66332A approximately every 0.1s. Magnitudes and phases of both parameters at the frequencies of interest, required to permit the above calculations, were determined using a discrete Fourier transform (DFT) after the method first described by Scott and Parker [24]. The resulting magnitude and phase plots for the batteries used here are shown in Fig. 7.

The results of attempting to fit the various ECMs to these batteries appear in Fig. 8. The associated model parameters appear in Tables 2 and 3.

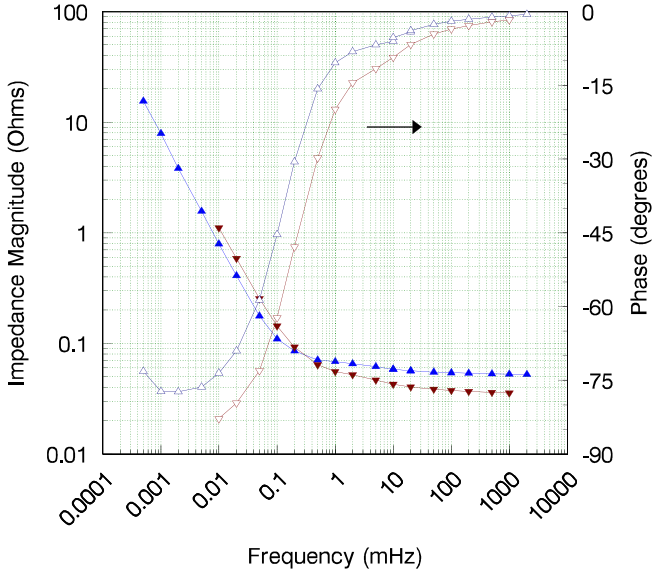


**Fig. 7.** Plot of measured impedance magnitude (solid blue up triangles) and phase (open blue up triangles) of a lithium nickel cobalt aluminium oxide (NCR/NCA) 2170 cell and magnitude (solid red down triangles) and phase (open red down triangles) measured on a lithium nickel manganese cobalt oxide (INR/NMC) 18650 cell.

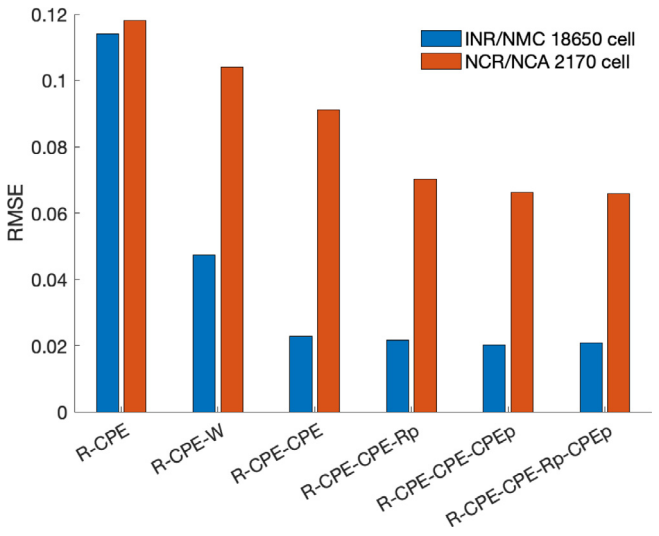


**Fig. 8.** RMS error obtained fitting each ECM to the two data sets presented in Fig. 7. The left-hand blue bars correspond to the NMC cell.

**Table 2**
ECM Parameters for NMC 18650 cell

| Parameter | $R_s$ | $\alpha$ | $C_F$ | $C_W$ | $\alpha 2$ |
|---|---|---|---|---|---|
| Value | 0.0330 | 0.99 | 14,180 | 187 | 0.27 |

**Table 3**
ECM Parameters for NCA 2170 cell

| Parameter | $R_s$ | $\alpha$ | $C_F$ | $C_W$ | $\alpha 2$ | $R_p$ |
|---|---|---|---|---|---|---|
| Value | 0.0503 | 0.99 | 22,230 | 230 | 0.272 | 44 |

Observing the RMSE for the various models in Fig. 8, it is clear for the NMC cell, for which less data is available, that the R-CPE-CPE model is the appropriate choice. More than five degrees of freedom cannot be justified, and the error in the data is relatively small in measurement terms. In the case of the NCA cell, the noise is much larger, but the data span a greater frequency range. The RMSE settles to a value almost triple that of the other cell. Nevertheless, the data clearly justify a six-parameter model, whose parameters appear in Table 3. The more complex models have very slightly less RMSE, but not by a margin that we consider to warrant the complexity.

## 7. Conclusion

We have revisited the work of Berthier, employing fractional equivalent-circuit models inspired by recent, extra-low frequency EIS measurements. Our conclusion is that it is possible to distinguish between a selection of ECMs. With real, noisy data there will be models between which it will not be possible to make a reliable selection. We say of these that their extra degrees of freedom are not justified in the face of the quantity and quality of data available. In other words, adding degrees of freedom is not justified if the RMSE does not decrease by an extent greater than the expected uncertainty. Up to that point, the model with more degrees of freedom and lower RMSE is preferred.

The rich literature on gray-box, equivalent-circuit models exposes a disconnect between the work of electrochemists and electrical engineers, the former have great insight into the circuit elements, and the latter the tools to fit and employ them.

It has become clear that a guided fitting approach and multiple restarts in the optimization can help in finding a global minimum, especially in the case of models with more than a few degrees of freedom. When selecting between several ECMs, the most simple is fitted first, and the results of the first fit used as the starting point for fitting the next-most complicated model. Multiple restarts improve the likelihood that the optimization will not be caught in a saddle.

A wide range of frequencies improves the sensitivity of the RMSE to parameters in a given ECM. The frequency range should be chosen appropriately in view of the intended application of the model.

The ability to obtain values for parameters of elements in the ECM of a battery should allow electrochemists to associate internal components of a battery with branches in the ECM. The same ability should allow electrical engineers to observe degradation & ageing in a battery and predict failure well ahead of the event.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] F. Berthier, J.-P. Diard, R. Michel, Distinguishability of equivalent circuits containing CPEs: Part I. Theoretical part, J. Electroanal. Chem. 510 (1) (2001) 1–11. doi:10.1016/S0022-0728(01)00554-X..
[2] R. Fletcher, C. Xu, Hybrid Methods for Nonlinear Least Squares, IMA J. Numer. Anal. 7 (3) (1987) 371–389, https://doi.org/10.1093/imanum/7.3.371. url: https://academic.oup.com/imajna/article-lookup/doi/10.1093/imanum/7.3.371.
[3] P. Mauracher, E. Karden, Dynamic modelling of lead/acid batteries using impedance spectroscopy for parameter identification, J. Power Sources 67 (1–2) (1997) 69–84, https://doi.org/10.1016/S0378-7753(97)02498-1.
[4] J. Scott, R. Hasan, New Results for Battery Impedance at Very Low Frequencies, IEEE Access 7 (2019) 106925–106930, https://doi.org/10.1109/ACCESS.2019.2932094.
[5] R. Hasan, J. Scott, Extending Randles's Battery Model to Predict Impedance, Charge-Voltage, and Runtime Characteristics, IEEE, Access 8 (2020) 85321–85328, https://doi.org/10.1109/ACCESS.2020.2992771.
[6] A. Lasia, Impedance of the Faradaic Reactions in the Presence of Mass Transfer, in: Electrochemical Impedance Spectroscopy and its Applications, Springer New York, New York, NY, 2014, pp. 85–125. doi:10.1007/978-1-4614-8933-7_4..

[7] U. Westerhoff, K. Kurbach, F. Lienesch, M. Kurrat, Analysis of Lithium-Ion Battery Models Based on Electrochemical Impedance Spectroscopy, Energy Technol. 4 (12) (2016) 1620–1630, https://doi.org/10.1002/ente.201600154.
[8] W. Choi, H.-C. Shin, J.M. Kim, J.-Y. Choi, W.-S. Yoon, Modeling and Applications of Electrochemical Impedance Spectroscopy (EIS) for Lithium-ion Batteries, J. Electrochem. Sci. Technol 11 (1) (2020) 1–13, https://doi.org/10.33961/jecst.2019.00528.
[9] B. Sohlberg, E.W. Jacobsen, Grey box modelling – branches and experiences, IFAC Proceedings Volumes 41 (2) (2008) 11415–11420.
[10] L. Nugroho, R. Akmeliawati, Comparison of black-grey-white box approach in system identification of a flight vehicle, Journal of Physics Conference Series, IOP Science 1130 (1)..
[11] J. Brucker, W.G. Bessler, R. Gasper, Grey-box modelling of lithium-ion batteries using neural ordinary differential equations, Energy Informatics 4 (15)..
[12] J.E.B. Randles, Kinetics of rapid electrode reactions, Discuss. Faraday Soc. 1 (1947) 11, https://doi.org/10.1039/df9470100011.
[13] W. Cao, S.-L. Wang, C. Fernandez, C.-Y. Zou, C.-M. Yu, X.-X. Li, A novel adaptive state of charge estimation method of full life cycling lithium-ion batteries based on the multiple parameter optimization, Energy Science and Engineering 7 (5) (2019) 1544–1556.
[14] R. Hasan, J. Scott, Application of Swingler's method for analysis of multicomponent exponentials with special attention to non-equispaced data, in: 2016 IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA), IEEE, Melaka, Malaysia, 2016, pp. 12–15. doi:10.1109/CSPA.2016.7515794. url:http://ieeexplore.ieee.org/document/7515794/.
[15] A. Lasia, Dispersion of Impedances at Solid Electrodes, in: Electrochemical Impedance Spectroscopy and its Applications, Springer, New York, New York, NY, 2014, pp. 177–201, https://doi.org/10.1007/978-1-4614-8933-7_8.
[16] S. Westerlund, L. Ekstam, Capacitor theory, IEEE Trans. Dielect. Electr. Insul. 1 (5) (1994) 826–839, https://doi.org/10.1109/94.326654.
[17] T.T. Hartley, J.-C. Trigeassou, C.F. Lorenzo, N. Maamri, Energy Storage and Loss in Fractional-Order Systems, J. Comput. Nonlinear Dyn. 10 (6) (2015), https://doi.org/10.1115/1.4029511 061006.
[18] N. Hansen, Benchmarking the Nelder-Mead downhill simplex algorithm with many local restarts, in: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, Association for Computing Machinery, Montreal, Québec, Canada, 2009, pp. 2403–2408. url:https://doi.org/10.1145/1570256.1570335.
[19] U.M. Sundar, Numerical Optimization Using MATLAB (2015). url:https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/solutions/automotive/files/in-expo-2015/numerical-optimization-using-matlab.pdf..
[20] A. Parkhomenko, (2017). url:https://www.andrii-parkhomenko.net/files/Numerical%20optimization%20in%20Matlab.pdf..
[21] J.A. Nelder, R. Mead, A Simplex Method for Function Minimization, Comput. J. 7 (4) (1965) 308–313, https://doi.org/10.1093/comjnl/7.4.308.
[22] W.H. Press, Numerical recipes in C : the art of scientific computing, 2nd Edition,., Cambridge University Press, Cambridge [England], 1992.
[23] C. Zou, L. Zhang, X. Hu, Z. Wang, T. Wik, M. Pecht, A review of fractional-order techniques applied to lithium-ion batteries, lead-acid batteries, and supercapacitors, J. Power Sources 390 (2018) 286–296, https://doi.org/10.1016/j.jpowsour.2018.04.033.
[24] J. Scott, A. Parker, Distortion analysis using SPICE, J. Audio Eng. Soc. 43 (12) (1995) 1029–1040.

# Appendix B

# FMP Files

| Frequency (Hz) | Magnitude | Phase (degrees) |
|---|---|---|
| 1.00E-05 | 1.11E+00 | -82.81 |
| 2.00E-05 | 5.89E-01 | -79.62 |
| 5.00E-05 | 2.53E-01 | -73.05 |
| 1.00E-04 | 1.44E-01 | -62.29 |
| 2.00E-04 | 9.25E-02 | -47.9 |
| 5.00E-04 | 6.40E-02 | -29.8 |
| 1.00E-03 | 5.54E-02 | -19.87 |
| 2.00E-03 | 5.20E-02 | -14.46 |
| 5.00E-03 | 4.65E-02 | -11.61 |
| 1.00E-02 | 4.27E-02 | -9.29 |
| 2.00E-02 | 4.04E-02 | -6.66 |
| 5.00E-02 | 3.86E-02 | -4.56 |
| 1.00E-01 | 3.76E-02 | -3.56 |
| 2.00E-01 | 3.68E-02 | -2.84 |
| 5.00E-01 | 3.60E-02 | -2.09 |
| 1.00E+00 | 3.55E-02 | -1.67 |

FIGURE B.1: An example of a .fmp, or fmp, file. This is a simple file containing 3 entries per row, using a space as a delimiter. The columns show the frequency (f), magnitude (m) and phase (p) for a given circuit/battery.

A .fmp, or fmp, file contains the frequency (f), magnitude (m) and phase (p) for a given circuit/battery. For this thesis, these are used exclusively as they are simple making them computationally trivial. An example can be seen below in Figure B.1. Note: the headers have been added for clarity, they are omitted from actual fmp files).

# Appendix C

# Overall Fitting Code

```
%% Inputs
% .fmp data formatted in an nx3 array called "fmpFit"


% Print fitting options, and get input from the user. The following inputs
% are valid:

% R-CPE = 1
% R-CPE-W = 2
% R-CPE-CPE = 3
% R-CPE-CPE//Rp = 4
% R-CPE-CPE//CPEp = 5
% R-CPE-CPE//Rp//CPEp = 6

fprintf("Please enter a number to select circuit to fit to");
c=input("Type the number of circuit you want to fit: ");

% Import the fmp file as the dataset to fit each ECMS to
f = fmpFit(:,1);
m = fmpFit(:,2);
p = fmpFit(:,3);

clf
close all

% A fit is done according to the number input

%% R-CPE Fit
if c == 1

    % Get input from the user with an initial guess for each
    % circuit parameter
    initR = input("Input initial parameter guesses for R: ");
    initA = input("Input initial parameter guesses for Alpha: ");
    initCF = input("Input initial parameter guesses for CF: ");

    % Store the guesses in an array
    initGuesses = [initR, initA, initCF];

    fprintf("Fitting R-CPE\n\n")

    % Use the R_CPE_Fit function to fit the ECM to the imported dataset
    [resCF, resA, resR, RMSE, mGen, pGen] = R_CPE_Fit(f, m, p, initGuesses);
    fprintf("Parameter Values Estimated:\n Rs = %g\n Alpha = %g\n CF = %g\n\n",resR,resA,resCF);
    fprintf("RMSE = %g\n", RMSE);

%% R-CPE-W Fit
elseif c==2
    initR = input("Input initial parameter guess for R: ");
    initA = input("Input initial parameter guess for CPE Alpha: ");
    initCF = input("Input initial parameter guess for CPE CF: ");
    initW = input("Input initial parameter guess for W CF: ");

    initGuesses = [initR, initA, initCF, initW];

%    initGuesses = [0.0495,0.7,8000,1000];

    fprintf("Fitting R-CPE-W\n\n")
    [resCF, resA, resW,resR, RMSE, mGen, pGen] = R_CPE_W_Fit(f, m, p, initGuesses);
    fprintf("Parameter Values Estimated:\n Rs = %g\n Alpha = %g\n CF = %g\n Warburg Cf = %g\n\n",resR,resA,resCF, resW);
    fprintf("RMSE = %g\n", RMSE);

%% R-CPE-CPE Fit
elseif c==3
    initR = input("Input initial parameter guess for R: ");
    initA = input("Input initial parameter guess for CPE Alpha: ");
    initCF = input("Input initial parameter guess for CPE CF: ");
    initA2 = input("Input initial parameter guess for CPE2 Alpha: ");
    initCF2 = input("Input initial parameter guess for CPE2 CF: ");

    initGuesses = [initR, initA, initCF, initA2, initCF2];

%    initGuesses = [0.0495,0.7,8000,0.4,1000];

    fprintf("Fitting R-CPE-CPE\n\n")
    [resCF, resA, resCF2, resA2,resR, RMSE, mGen, pGen] = R_CPE_CPE_Fit(f, m, p, initGuesses);
    fprintf("Parameter Values Estimated:\n Rs = %g\n Alpha1 = %g\n CF1 = %g\n Alpha2 Cf = %g\n CF2 Cf = %g\n\n",resR,resA,resCF, resA2, resCF2);
    fprintf("RMSE = %g\n", RMSE);
```

FIGURE C.1: Part 1 of the .main file for the fitting algorithm. This file gets input from the user then uses this information to optimise an ECM against simulated data.

```
%% R-CPE-CPE//Rp Fit
elseif c==4
    initR = input("Input initial parameter guess for R: ");
    initRp = input("Input initial parameter guess for Rp: ");
    initA = input("Input initial parameter guess for CPE Alpha: ");
    initCF = input("Input initial parameter guess for CPE CF: ");
    initA2 = input("Input initial parameter guess for CPE2 Alpha: ");
    initCF2 = input("Input initial parameter guess for CPE2 CF: ");

    initGuesses = [initR, initRp, initA, initCF, initA2, initCF2];

%    initGuesses = [0.0495,50,0.7,8000,0.4,1000];

    fprintf("Fitting R-CPE-CPE//RP\n\n")
    [resCF, resA, resCF2, resA2,resR,resRp, RMSE, mGen, pGen] = R_CPE_CPE_RP_Fit(f, m, p, initGuesses);
    fprintf("Parameter Values Estimated:\n Rs = %g\n Rp = %g\n Alpha1 = %g\n CF1 = %g\n Alpha2 = %g\n CF2 = %g\n\n",resR,resRp,resA,resCF, resA2, resCF2);
    fprintf("RMSE = %g\n", RMSE);
%% R-CPE-CPE//CPEp
elseif c==5

    initR = input("Input initial parameter guess for R: ");
    initA = input("Input initial parameter guess for CPE Alpha: ");
    initCF = input("Input initial parameter guess for CPE CF: ");
    initA2 = input("Input initial parameter guess for CPE2 Alpha: ");
    initCF2 = input("Input initial parameter guess for CPE2 CF: ");
    initAp = input("Input initial parameter guess for CPEp Alpha: ");
    initCFp = input("Input initial parameter guess for CPEp CF: ");

    initGuesses = [initR, initA, initCF, initA2, initCF2, initAp, initCFp];

%      initGuesses = [0.0495,0.7,8000,0.4,1000, 0.5, 0.15];

    fprintf("Fitting R-CPE-CPE//CPEp\n\n")
    [resCF, resA, resCF2, resA2,resCF3,resA3,resR, RMSE, mGen, pGen] = R_CPE_CPE_CPEp_Fit(f, m, p, initGuesses);
    fprintf("Parameter Values Estimated:\n Rs = %g\n Alpha1 = %g\n CF1 = %g\n Alpha2 = %g\n CF2 = %g\n AlphaP = %g\n CFP = %g\n\n",resR,resA,resCF,resA2,resCF2,resA3,resCF3);
    fprintf("RMSE = %g\n", RMSE);
%% R-CPE-CPE//Rp//CPEp
elseif c==6

    initR = input("Input initial parameter guess for R: ");
    initRp = input("Input initial parameter guess for Rp: ");
    initA = input("Input initial parameter guess for CPE Alpha: ");
    initCF = input("Input initial parameter guess for CPE CF: ");
    initA2 = input("Input initial parameter guess for CPE2 Alpha: ");
    initCF2 = input("Input initial parameter guess for CPE2 CF: ");
    initAp = input("Input initial parameter guess for CPEp Alpha: ");
    initCFp = input("Input initial parameter guess for CPEp CF: ");

    initGuesses = [initR, initRp, initA, initCF, initA2, initCF2, initAp, initCFp];

%    initGuesses = [0.05 400 0.85 10000 0.3 500 0.15 0.5];

    fprintf("Fitting R-CPE-CPE//Rp//CPEp\n\n")
    [resCF, resA, resCF2, resA2,resCF3,resA3,resR,resRp, RMSE, mGen, pGen] = R_CPE_CPE_Rp_CPEp_Fit(f, m, p, initGuesses);
    fprintf("Parameter Values Estimated:\n Rs = %g\n Rp = %g\n Alpha1 = %g\n CF1 = %g\n Alpha2 = %g\n CF2 = %g\n AlphaP = %g\n CFP = %g\n\n",resR,resRp,resA,resCF,resA2,resCF2,resA3,resCF3);
    fprintf("RMSE = %g\n", RMSE);
end
```

FIGURE C.2: Part 2 of the .main file for the fitting algorithm. This file gets input from the user then uses this information to optimise an ECM against simulated data.

```
139
140     %% Plotting
141
142     % Plot the given magnitude values vs frequency
143 -   figure('Name', 'MagZ vs F')
144 -   loglog(f, m, '-o', ...
145         'Color', [0.00,0.45,0.74],...                % blue
146         'LineWidth',1.5,...
147         'LineStyle', '-',...
148         'MarkerSize',10,...
149         'MarkerEdgeColor',[0.00,0.45,0.74])
150 -   hold on
151 -   loglog(f, mGen, '-o', ...
152         'Color', [0.85,0.33,0.10],...                % orange
153         'LineWidth',1.5,...
154         'LineStyle', '-',...
155         'MarkerSize',10,...
156         'MarkerEdgeColor',[0.85,0.33,0.10])
157
158 -   xlabel('Frequency (Hz)')
159 -   ylabel('Impedance Magnitude')
160 -   legend("Target","Fitted")
161
162     % Plot the given phase values vs frequency
163 -   figure('Name', 'Phase vs F')
164 -   semilogx(f, p, '-o', ...
165         'Color', [0.00,0.45,0.74],...                % Blue
166         'LineWidth',1.5,...
167         'LineStyle', '-',...
168         'MarkerSize',10,...
169         'MarkerEdgeColor',[0.00,0.45,0.74])
170 -   hold on
171 -   semilogx(f, pGen, '-o', ...
172         'Color', [0.85,0.33,0.10],...                % Orange
173         'LineWidth',1.5,...
174         'LineStyle', '-',...
175         'MarkerSize',10,...
176         'MarkerEdgeColor',[0.85,0.33,0.10])
177 -   xlabel('Frequency (Hz)')
178 -   ylabel('Phase ( ^o )')
179 -   legend("Target","Fitted")
```

FIGURE C.3: Part 3 of the .main file for the fitting algorithm. This file gets input from the user then uses this information to optimise an ECM against simulated data.

# Appendix D

# Nelder-Mead Function (Code)

This is the Nelder-Mead Code named "Simplex," which is a modified version of reference [50]. Note: The Graphing function from this set of functions is not used as it does not display adequate data.

```
function [argument, minima, iterationCount] = Simplex(performance_function, n_variable, varargin)
%% Get mandatory and optional Inputs using inputParser;

[setting] = InputParser( performance_function, n_variable, varargin );

%% Initial Conditions & Constants

n = n_variable+1; % number of vertex
p = 1; %reflection coefficient
chi = 2; %expansion coefficient
gamma = 0.5; %contraction coefficient
sig = 0.5; %shrink coefficient

% generate initial conditions for the solver
vertex = InitiateSimplex(n,setting);

%%
% starting point to get into the while loop
ii = 1;
convergence = setting.tolerance * 10;
while(convergence >= setting.tolerance && ii <= setting.iteration)
    %% Sort based on performance
    ii = ii+1;
    vertex = Sort(vertex);
    centroid = Centroid(vertex,n,setting);

    if setting.plot
        convergenceTrace(ii)=vertex(1).value;
    end

    %% Reflect
    % Reflect the worst vertex through the centroid of the remaining vertex
    % If the reflected vertex value is between the best and next-to-worst
    % vertex, replace the worst vertex with the reflected vertex

    Reflect.coord = (1+p).*centroid.coord - p.*vertex(n).coord;
    Reflect.value = PerformanceFunctionCaller(Reflect,setting);

    if(vertex(1).value <= Reflect.value && Reflect.value < vertex(n-1).value)
        vertex(n)=Reflect;
    elseif(Reflect.value < vertex(1).value)
        %% Expand
        % If the reflected vertex is better than then extend the reflected
        % vertex further past the centroid. Replace the worst solution with the
        % the better or the reflected or expanded.

        Expand.coord = (1-chi).*centroid.coord+chi.*Reflect.coord;
        Expand.value = PerformanceFunctionCaller(Expand,setting);

        if(Expand.value < Reflect.value)
            vertex(n) = Expand;
        else
            vertex(n) = Reflect;
        end

    elseif(vertex(n-1).value <= Reflect.value)
        %% Contract
        % The reflected vertex is worse than the next-to-worst vertex.
        % Contract the worst solution to a location between itself and the
        % reflection.

        ContractOut.coord = (1-gamma).*centroid.coord + gamma.*Reflect.coord; %Contract Outside
        ContractOut.value = PerformanceFunctionCaller(ContractOut,setting);

        ContractIn.coord = (1-gamma).*centroid.coord + gamma.*vertex(n).coord;  %Contract Inside
        ContractIn.value= PerformanceFunctionCaller(ContractIn,setting);

        if(vertex(n-1).value <= Reflect.value && Reflect.value < vertex(n).value && ContractOut.value <= Reflect.value)
            vertex(n) = ContractOut;

        elseif(vertex(n).value <= Reflect.value && ContractIn.value < vertex(n).value) %Contract Inside
            vertex(n) = ContractIn;
        end

    else
        %% Shrink
        % Shrink the simplex towards the best vertex.  This improves
        % convergence if the solution is non-smooth

        for jj=2:n
            vertex(jj).coord = (1-sig).*vertex(1).coord + sig.*vertex(jj).coord;
            vertex(jj).value = PerformanceFunctionCaller(vertex(jj),setting);
        end
    end
```

```
    end

    convergence = Convergence(vertex);
end

% collect outputs after sorting to ensure index 1 is the desired solution
vertex = Sort(vertex);
argument = vertex(1).coord;
minima = vertex(1).value;
iterationCount = ii;

%% Plot
if setting.plot
    figure();
    % shift up to make log plot work avoiding zeroes
    semilogy(convergenceTrace - min(convergenceTrace) + setting.tolerance );
    xlabel('iterations');
    ylabel( ['performance index - min performance ', num2str(min(convergenceTrace))] );
    title(['Performance Vs ', num2str(ii), ' Iterations']);
    grid on;
end

end
function [setting] = InputParser( performance_function, n_variable, nameValuePair )
%% Wrapper function for handeling user inputs and some error checking

% fill the parcer with the list of variables and their requirements
p = FillInputParser(n_variable);


% where the parcing actualy happens
p.parse( performance_function, n_variable, nameValuePair{:} );

% error checking and formatting that doesn't fit within the parcer
setting = ProcessInputParser( p );

end
```

```
function p = FillInputParser(n_variable)
%% Fill the inputParser with all information to get the required,
% optional, and optional name-value pairs.  Also determines default vaulues
% for anything not set, and determines basic error checking of any inputs.

p = inputParser;

%% Mandatory inputs
argName = 'performance_function';
validation = @(x) validateattributes( x, {'function_handle'}, {'scalar'} );
p.addRequired( argName, validation );

argName = 'n_variable';
validation = @(x) validateattributes( x, {'numeric'}, {'scalar','positive',} );
p.addRequired( argName, validation );

%% Optional inputs

% no optional inputs

%% name-value pairs
argName = 'tolerance';
default = 1*10^-8; % default tolerance for convergence
validation = @(x) validateattributes( x, {'numeric'}, {'scalar','positive'} );
p.addParameter( argName, default, validation );

argName = 'iteration';
default = 10000; % default maximum number of iterations
validation = @(x) validateattributes( x, {'numeric'}, {'scalar','positive'} );
p.addParameter( argName, default, validation );

argName = 'lowBoundary';
default = ones(n_variable,1) * -inf; % unbounded
validation = @(x) validateattributes( x, {'numeric'}, {'vector','numel',n_variable} );
p.addParameter( argName, default, validation );

argName = 'highBoundary';
default = ones(n_variable,1) * inf; % unbounded
validation = @(x) validateattributes( x, {'numeric'}, {'vector','numel',n_variable} );
p.addParameter( argName, default, validation );

argName = 'plot';
default = false; % do not plot
validation = @(x) validateattributes( x, {'logical','numeric'}, {'scalar'} );
p.addParameter( argName, default, validation );

argName = 'initial';
default = nan(n_variable+1,n_variable); % random input conditions will be filled in later
validation = @(x) validateattributes( x, {'numeric'}, {'size',[n_variable+1,n_variable]} );
p.addParameter( argName, default, validation );

argName = 'seed';
default = 17; % default seed for the random number generator
validation = @(x) validateattributes( x, {'numeric'}, {'scalar','positive'} );
p.addParameter( argName, default, validation );

argName = 'additionalData';
default = []; % default is empty array if no additional data is passed in
% no validation
p.addParameter( argName, default );

end

function setting = ProcessInputParser( p )
%% Fill setting struct with inputs taking the priority of: name-value pairs,
% setting struct which was passed in, and then the default

setting = p.Results;

% any additional formatting requirements

% iteration limit must be an int
setting.iteration = ceil(setting.iteration);

end
```

```
function performance = PerformanceFunctionCaller(vertex,setting) %Write your function in matrix form
%% Call the performance_function after applying boundry conditions
% The performance function must be arranged such that a lower number
% represents a better fit.  Positive, negative, zero, and non-finite
% numbers are acceptable.

% If the performance function is not continuous or not smooth, the simplex
% method may nto produce desirable results.

% apply limits
if sum(vertex.coord >= setting.highBoundary(:)') || sum(vertex.coord <= setting.lowBoundary(:)')
    performance = inf;
    return;
end

% call the function using 1 or two inputs depending on if additionalData is
% emtpy
if isempty(setting.additionalData)
    % call performance function with 1 input
    performance = setting.performance_function( vertex.coord);
else
    % call performance function with 2 input
    performance = setting.performance_function( vertex.coord, setting.additionalData);
end

% make sure the function output is correctly formatted
if (length(performance)~=1) || ~isnumeric(performance)
     error('simpleSimples:performance_function','The performance function must always output a single numeric value: ''%s''\n',char(setting.performance_function));
end

end
```

```
function vertex = InitiateSimplex(n,setting)
%% Create initial conditions which fall between the boundary conditions
% becuase of the potential for complex boundary conditions, it's possible
% to get randomly generated boundary conditions which are all inf.  if this
% happens an error is thrown advising to supply a valid sead.  If some
% values are inf, random initial conditions are filled in untill no initial
% conditions are inf.  will attemp up to the maximum iteration count to
% find all non-inf solutions.  If still unsuccessfull, it will throw a
% warning and continue.

% kick off the random number generator
rng(setting.seed);

% loop till all initial values are non-inf
value = inf(n,1);
vertex(1:n) = struct();
jj = 0;
while sum( value~=inf ) ~= n && jj < setting.iteration
    jj = jj + 1;

    for ii = 1:n
        % start with a random array
        vertex(ii).coord = [rand(n-1,1) .* (setting.highBoundary(:) - setting.lowBoundary(:)) + setting.lowBoundary(:)]';

        % push over any initial conditions given
        for jj = 1:(n-1)
            if ~isnan(setting.initial(ii,jj))
                vertex(ii).coord(jj) = setting.initial( ii, jj);
            end
        end

        % fill in the values for all inital guesses
        vertex(ii).value = PerformanceFunctionCaller(vertex(ii),setting);
        value(ii) = vertex(ii).value;

        % if value is good, push coord over to the initial settings
        for jj = 1:(n-1)
            if vertex(ii).value ~= inf
                setting.initial(ii, jj) = vertex(ii).coord(jj);
            else
                % the seed falls outside of boundary conditions
                setting.initial(ii, jj) = nan;
            end
        end
    end
```

```
    % if all initial conditions are inf, you could have bad boundry
    % conditions.  supply a seed to ensure this is not the case
    if sum( value~=inf ) == 0
        error('simpleSimples:InitialSimplex','The initial conditions provided no valid results.  Consider provideing initial conditions');
    end

    if jj == setting.iteration
        warning('simpleSimples:InitialSimplex','The initial conditions provided solutions with invalid results, and attempts to remove all invalid results were unsuccessfull.
    end

end
end
```

```
function vertex = Sort(vertex)
%% sort vertices into ascending order based on .value
    [~,index] = sort([vertex.value]);
    vertex = vertex(index);

end
```

```
function centroid = Centroid(vertices,n,setting)
%% Compute the vector average of all points except the hightest error; i.e.,
% the center of the "face" of the simplex across from the high point.
% We will subsequently explore along the ray from the high point through
% that center.

    Sum=zeros(1,(n-1));
    for i=1:n-1 % array is already sorted, last element has highest error
        Sum=Sum+vertices(i).coord;
    end
    centroid.coord=Sum./(n-1);
    centroid.value=PerformanceFunctionCaller(centroid,setting);

end
```

```
function[convergence] = Convergence(vertices)
% Convergence is based on the RMSE between the simulated and measure data.
% See 'RMSE' function for details.
% If any vertices have non-finite values then convergence will not be reached.

    convergence=RMSE([vertices.value]);

    % handle INF vallues in the vallue array
    if isnan(convergence)
        convergence = inf;
    end

end
```

# Appendix E

# RMSE Function (Code)

This function calculates the RMSE between the simulated and fitting data, which determines the convergence of the optimisation.

```
% Calculates the Root-Mean-Square-Error between
% the measured and simulated data
function [rmseCOMPLEX] = rmseCalc(simulatedM, simulatedP, measuredM, measuredP, freq) %#o

% Zmeasured is the impedance of the measured data
% Zsim is the impedance of the simulated data from the Simplex function

% These 2 lines combine the magnitude and phase to generate
% impedance values.
Zmeasured = measuredM.*cos(pi.*measuredP./180) + 1i.*(measuredM.*sin(pi.*measuredP/180));
Zsim = simulatedM.*cos(pi.*simulatedP./180) + 1i.*(simulatedM.*sin(pi.*simulatedP/180));

E = abs((Zsim - Zmeasured)./Zmeasured); % Calculates the error
SQE = E.^2;                             % Squares the error
MSE = sum(SQE(:));
MSE = MSE/length(freq);                 % Takes the mean of the sum of errors
rmseCOMPLEX = sqrt(MSE);                % Square root to get the RMSE
end
```

FIGURE E.1: The RMSE calculation function in Matlab Code.