Technical Final Report - Amy Gibbon and Joe Garvey

It is recommended that this report is read in a Jupyter Notebook, and it is written to be read in such a format.

Introduction

LiFETIME are a startup company looking to enable the second-life use of Lithium-ion batteries. Currently, batteries, especially those used in EVs, are deemed to reach “end-of-life” (EOL) when their maximum battery capacity falls to 70-80% of their rated value. LiFETIME aim to provide a low-cost, open-source solution for cell health testing for second-life battery applications. This device will measure how much future potential a battery still has left, allowing the operator to decide whether it can continue to be used, and in what environments or use cases it can be used. While LiFETIME focus on the hardware of their design, the task given for the GM2 project is an investigation into developing a model suitable for practical analysis of real-world cells. The brief was broad, with there being a plethora of methods of cell health analysis, so the first objective was to review existing literature on lithium-ion cell modelling, and better familiarise ourselves with the physics of battery degradation. Following a combination of discussion with LiFTIME and literature review it was concluded that there are two potential routes we could pursue, either Electrochemical Impedance Spectroscopy (EIS) data, or Incremental Capacity Analysis (ICA), both of which will be outlined in this report. This progressed into a brief analysis of equivalent circuit modelling (ECM), as a method of analysing EIS data. The team has collected sufficient primary data to allow us test the aforementioned analysis techniques. Prior to our project, the data had not yet been used for any application other than for model validation. Because of time limitations, our project has focused on delivering key research, analysis of data and preliminary investigation into ICA, EIS and ECM, rather than attempting to deliver a model.

↳ *2 cells hidden*

Background - *Amy*

Lithium-ion battery degradation

Lithium-ion battery degradation is incredibly complex. Below is a brief cheat sheet to key routes of degradation, to give context to the problem we are trying to assess.

Key degradation modes of Lithium-ion batteries:

- **LLI:** Loss of lithium inventory

- **LAM:** Loss of active materials
- **ORI:** Ohmic resistance increase
- **FRD:** Faradic Rate Degradation

SEI layer formation

- Permanently binds some lithium
- Reduces capacity as less Li+ available to move

Lithium plating

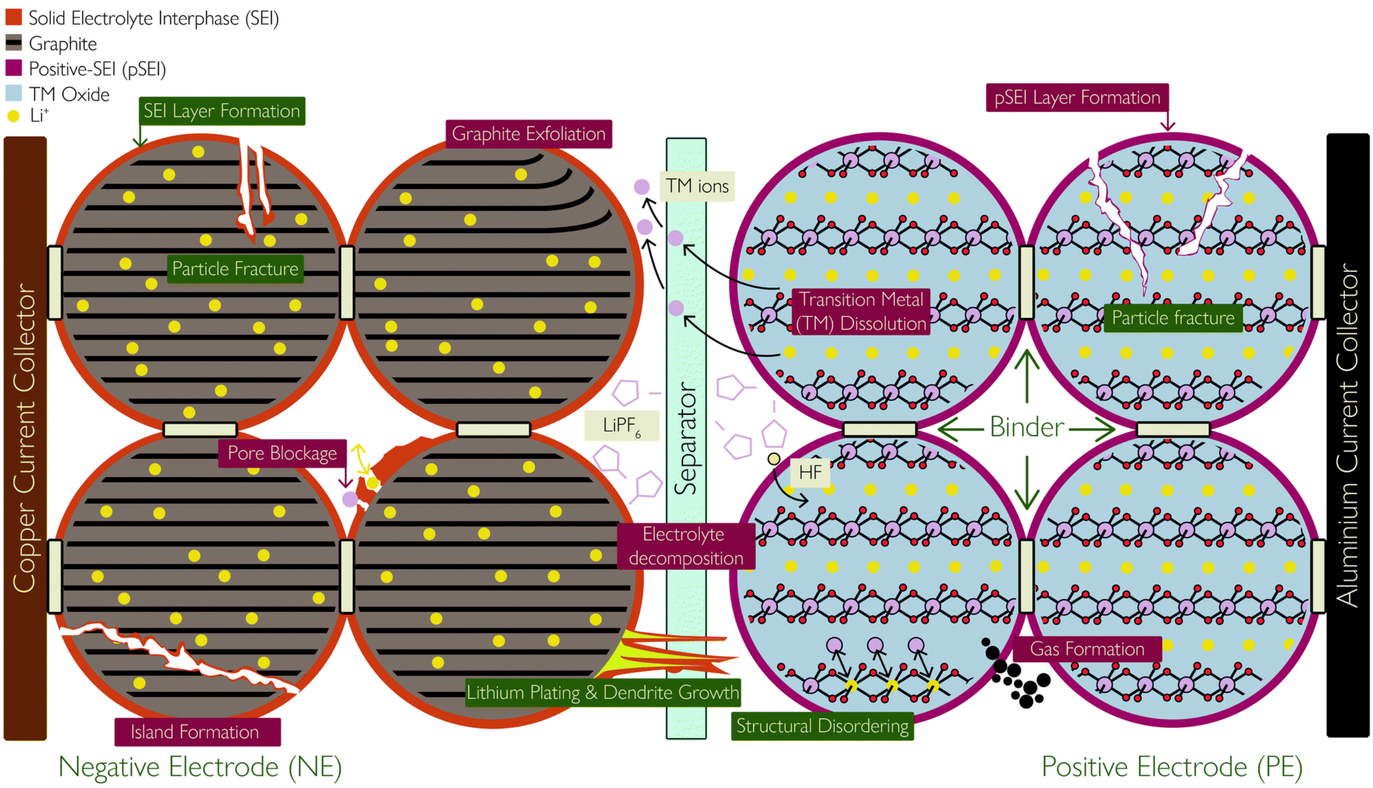
- Occurs when charging at cold temperatures or fast charging
- Graphite can’t accept the Li+ well or quickly enough so Li builds up as a plate
- This can lead to dendritic formation which can even stretch across the electrolyte and through the separator to cause short circuiting

Particle fracture

- Caused by thermal cycling
- Opens up more surface area for SEI to occur leading to more capacity loss

Structural disordering

- Happens in the presence of moisture
- Electrolyte reacts to form acidic HF
- This acid then reacts with the lithium oxide at the cathode, causing loss of electrolyte and reducing Li availability



Reference: Edge, Jacqueline & O'Kane, Simon & Prosser, Ryan & Kirkaldy, Niall & Patel, Anisha & Hales, Alastair & Ghosh, Abir & Ai, Weilong & Chen, Jingyi & Jiang, Jason & Li, Shen & Pang, Mei-Chin & Bravo Diaz, Laura & Tomaszewska, Anna & Marzook, Mohamed & Radhakrishnan, Karthik

& Wang, Huizhi & Patel, Yatish & Wu, Billy & Offer, Gregory. (2021). Lithium Ion Battery Degradation: What you need to know. Physical Chemistry Chemical Physics. 23. 10.1039/D1CP00359C.

Electrochemical Impedance Spectroscopy

Electrochemical Impedance Spectroscopy (EIS) is a method of spectroscopy that will be considered in this report. It detects the electrochemical reaction happening in the battery. An electric signal is applied with various frequencies to the battery system, giving an output that is a spectrum with a real and imaginary part, capturing the change in resistance of the battery. It is important to note that different frequencies correlate with different elements of the battery, and that the EIS spectrum is a Nyquist plot as a function of frequency, highlighting the different corresponding physical components. It is non-invasive to obtain spectrum, causing no damage to battery, and it can be done quickly in just 10-15 minutes. EIS spectrum changes with cycle number, making it a good indicator of degradation and SOH (State of Health). EIS data is usually interpreted using an equivalent circuit model (ECM), with a different ECM being required for each battery system (Middlemiss et al., 2020).The circuit model described by Westerhoff

Project outline: Weeks 3 and 4

Following our research and improved understanding of the problem, as well as discussion with LiFETIME, we have identified the following set of tasks as being the most important, and what we plan to pursue.

Tasks:

1. Clean, process and visualise the cycling data. Investigate how capacity changes with cycle number.
2. Clean, process and visualise the EIS data. Investigate how the EIS changes with cycle number.
3. Choose an ECM, using literature and our EIS data. Either replicate an ECM from literature, or create a unique one to match our EIS data.
4. Get a good picture of how each ECM component links to a physical aspect of the cell through literature review. Document this well.
5. Have a go at fitting the parameters of the ECM for a single SOC, cycle number and battery type
6. Investigate whether the ECM is valid for the other battery types, find the error quantitively
7. Investigate how the parameters change with SOC and cycle number
8. Develop an algorithm that takes an EIS spectrum, fits the parameters of the ECM found earlier, and returns the error between the expected parameters and the actual parameters. Use the research on the physical link between ECM components and cell components to output an indication of what areas of the cell have unusual readings and therefore may be

degrading, and to what extent. This may be tricky without any data for this, so we might need to find some online or create some.
9. Investigate whether it is possible to estimate current capacity of a cell from a single EIS reading, by finding the ECM and virtually cycling the model, producing a curve, and calculating capacity from that. This could be really powerful if possible, but would require strong understanding of how the model parameters change as the cell charges/discharges.

The tasks are in order of priority. From conversation with LiFETIME, completing tasks 1 and 2 would be good result. Reaching task 5 would be excellent, and the further analysis of tasks 6 and 7 would be a bonus. We agree that it is unlikely tasks 7 and 8 will be completed given our limited timeframe, however they will be discussed given recommendations for future work.

Imports and installs

[ ] ↳ *4 cells hidden*

Capacity analysis - *Joe*

[ ] ↳ *4 cells hidden*

EIS Analysis - *Amy*

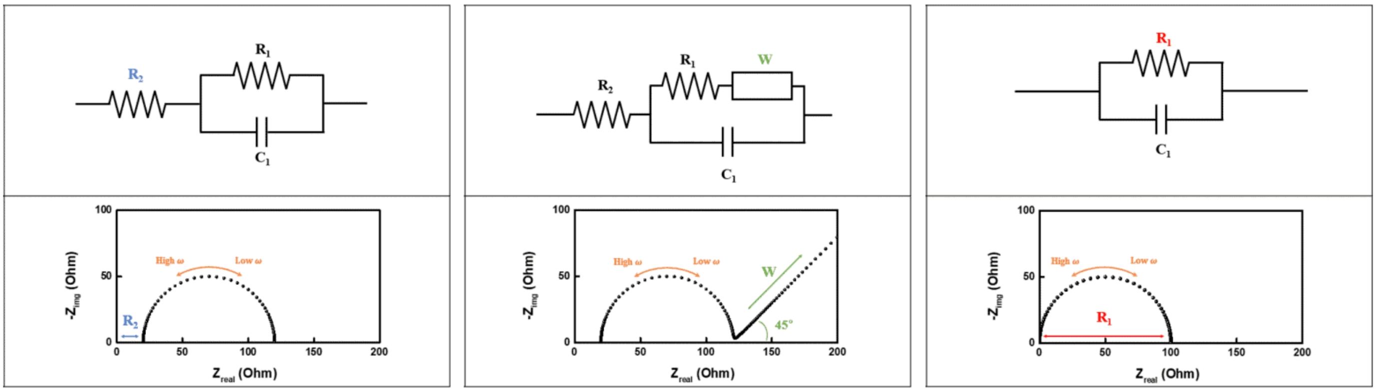
This section will outline analysis of the EIS data provided by LiFETIME. As previously outlined in the background section, EIS is a method of spectroscopy that detects the electrochemical reaction happening in the battery, by running a current or voltage through the battery at a range of frequencies. The result is a Nyquist plot of real impedance against imaginary impedance. The data provided by LiFTIME is potentiostatic, which means that the impedance data was found by applying a constant potential to the working electrode for a certain amount of time. First, the data is processed and all spectra are plotted. Then for the NX001 battery, the data is interpolated to create a full data frame with synthetic EIS data for every cycle, which is used to demonstrate degradation occuring in between tests that took place several months apart.

[ ] ↳ *15 cells hidden*

ECM Analysis - *Amy*

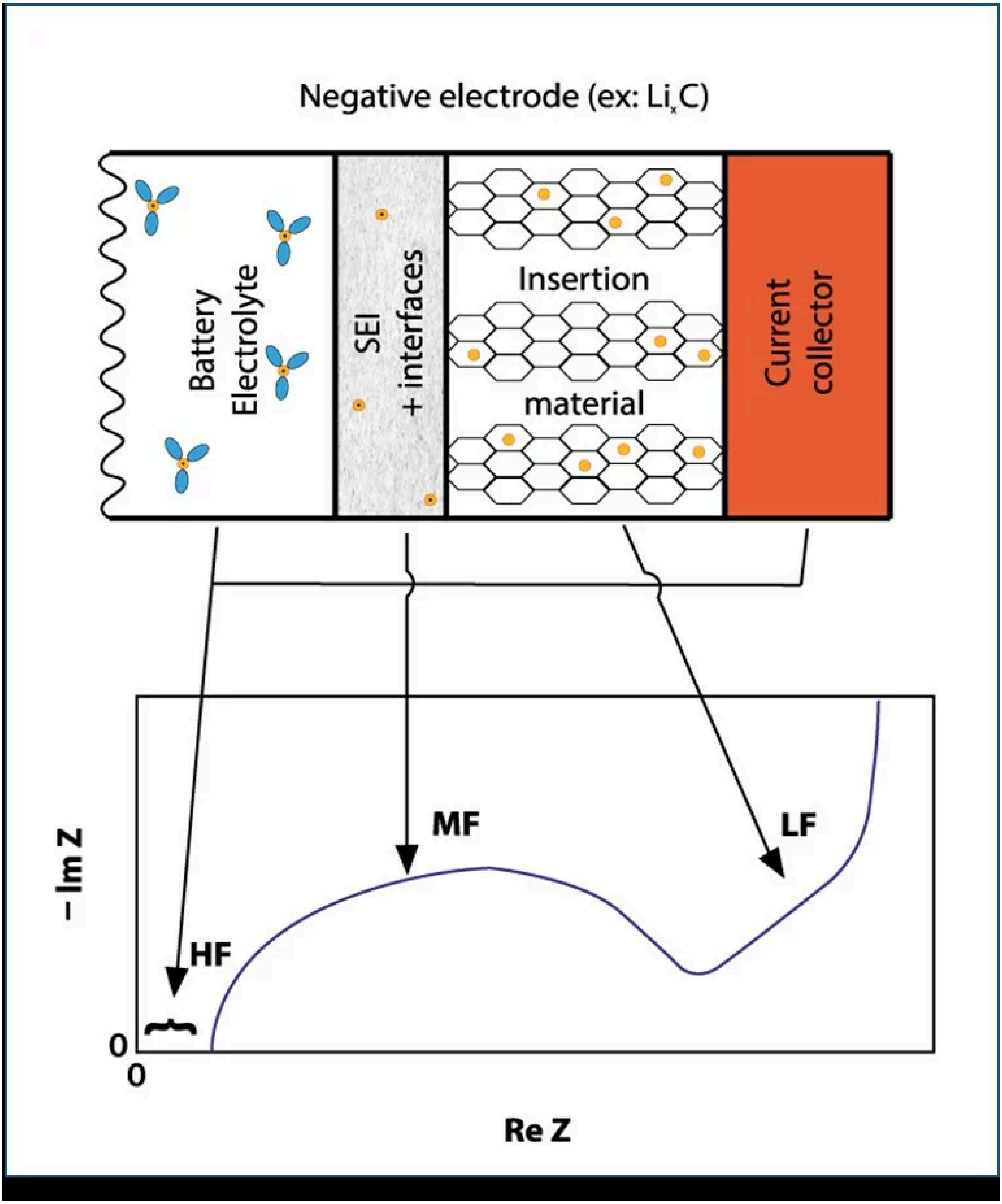
This section covers preliminary work into fitting an equivalent circuit model (ECM) to Electrochemical Impedance Spectroscopy (EIS) data. For the sake of time and efficienct, pre-existing Python packages where explored. Three packages seem to be suitable for this application: impedance.py (Murbach et al., 2020), scipy.optimize.curve_fit (Virtanen et al., 2020) and pyEIS (Knudsen, 2019). The fitting will be tested on a single cycle for a single cell. Cell NX001 and cycle number 103, 33% SOC was chosen, as it offers strong features.

The figure below, taken from a study on modeling and applications of Electrochemical Impedance Spectroscopy (EIS) for lithium-ion batteries, gives a good introduction to how simple circuits influence the EIS spectrum. On the far left, a resistor and capacitor in parallel contribute a semi-circle. In the first image, there is a resistor R2 and an RC component (p(R1,C1)), and the resistor R2 shifts the semi-circle along the real axis. The Warburg element, W, contributes the 45 degree assymptote, as seen in the central image. The constant phase element (CPE), not shown in these images, is a capacitive element, that has a Nyquist plot similar to that of a capacitor (a straight line), and when in parallel with a resistor it also produces a semi-circle shape. It is useful because it introduces a frequency-independent negative phase between current and voltage, so this is another component that will feature in the following circuits, generally replacing a capacitor.

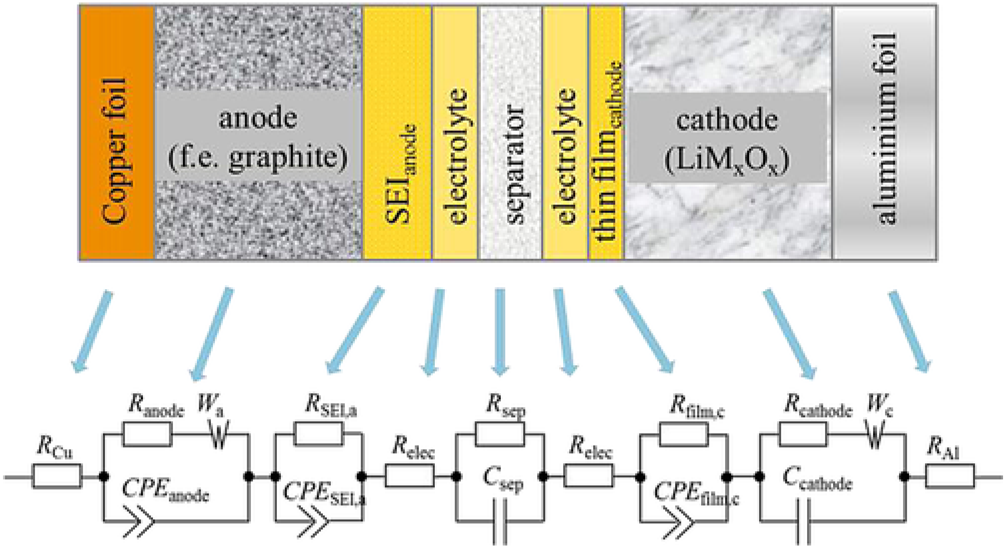


Choi W, Shin HC, Kim JM, Choi JY, Yoon WS. Modeling and Applications of Electrochemical Impedance Spectroscopy (EIS) for Lithium-ion Batteries. J Electrochem Sci Technol. 2020;11(1):1-13. Published online 2020 Jan 20. doi: https://doi.org/10.33961/jecst.2019.00528.

[Joe] Properties of the EIS spectrum can be correlated with properties of specific regions of the cell. (Biologic.net "Why use EIS for battery research?" (June 2023)) suggests that high frequencies correspond to the ohmic resistance of the electrolyte, the middle frequencies to SEI capacitance and the electron transfer rate and low frequencies give the user information about the diffusion processes of species within the insertion material. However, this information should be verified against another source, given that biologic do not cite a source for this and they seemingly manufacture cell cyclers (conflict of interest).



Westerhoff et Al also suggest that the ECM can be used to extrapolate information about each layer within the cell. In their results, a sufficiently complex ECM can be used to represent and model subsequent layers in the cell. These models consist of resistors, capacitors, inductors and constant phase elements. Differing levels of complexity yield different degrees of information about the cell.

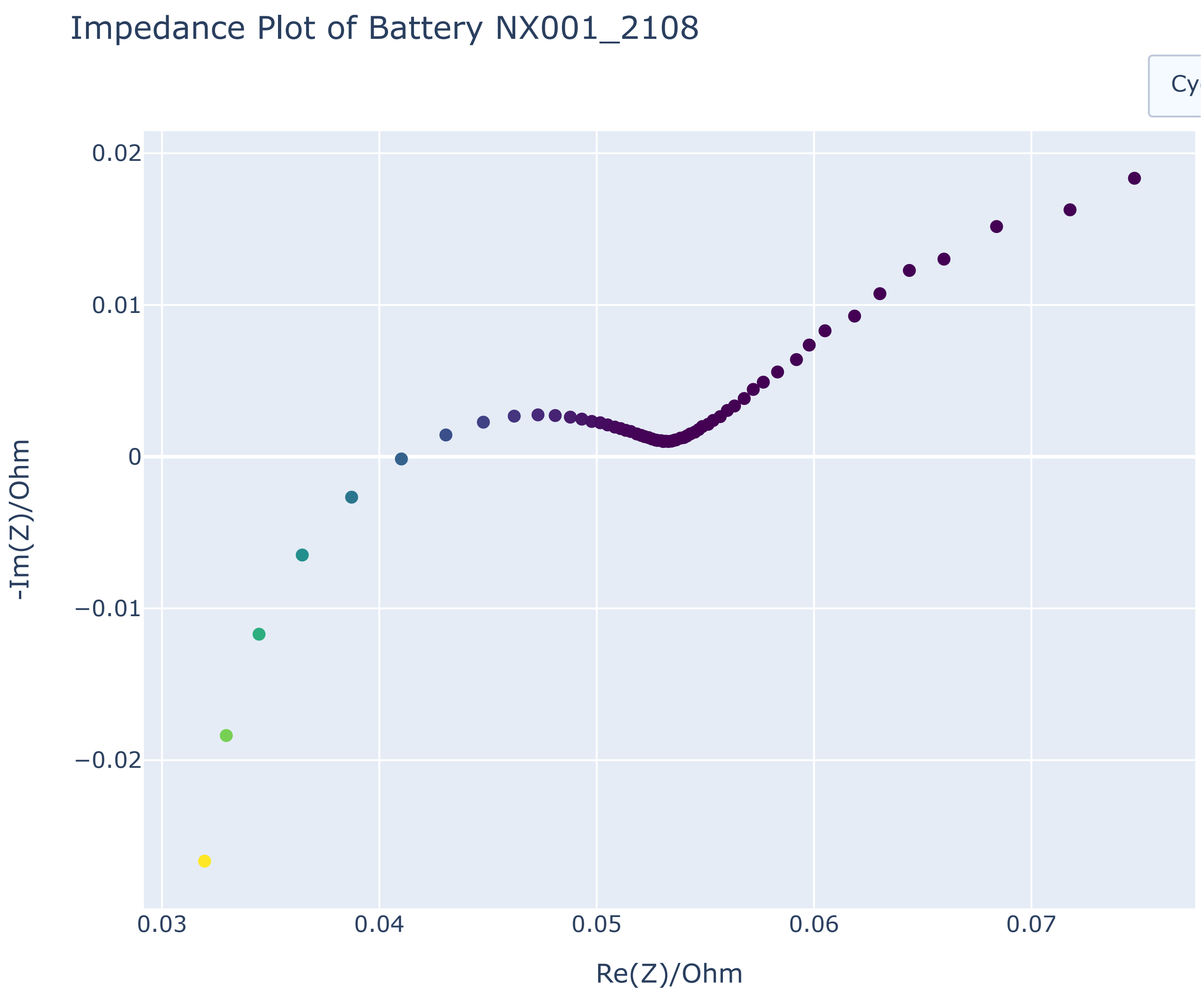


Reference:

Westerhoff, U., Kurbach, K., Lienesch, F. & Kurrat, M. Analysis of Lithium-Ion Battery Models Based on Electrochemical Impedance Spectroscopy. Energy Technol. 4, 1620–1630 (2016).

› Extract the specific data required and plot

Show code



Above is a plot of the chosen cycle: NX001, 21/08/2023, cycle 103, 33% SOC.

Define a function to fit the circuit model using impedance.py

Show code

Choosing a model

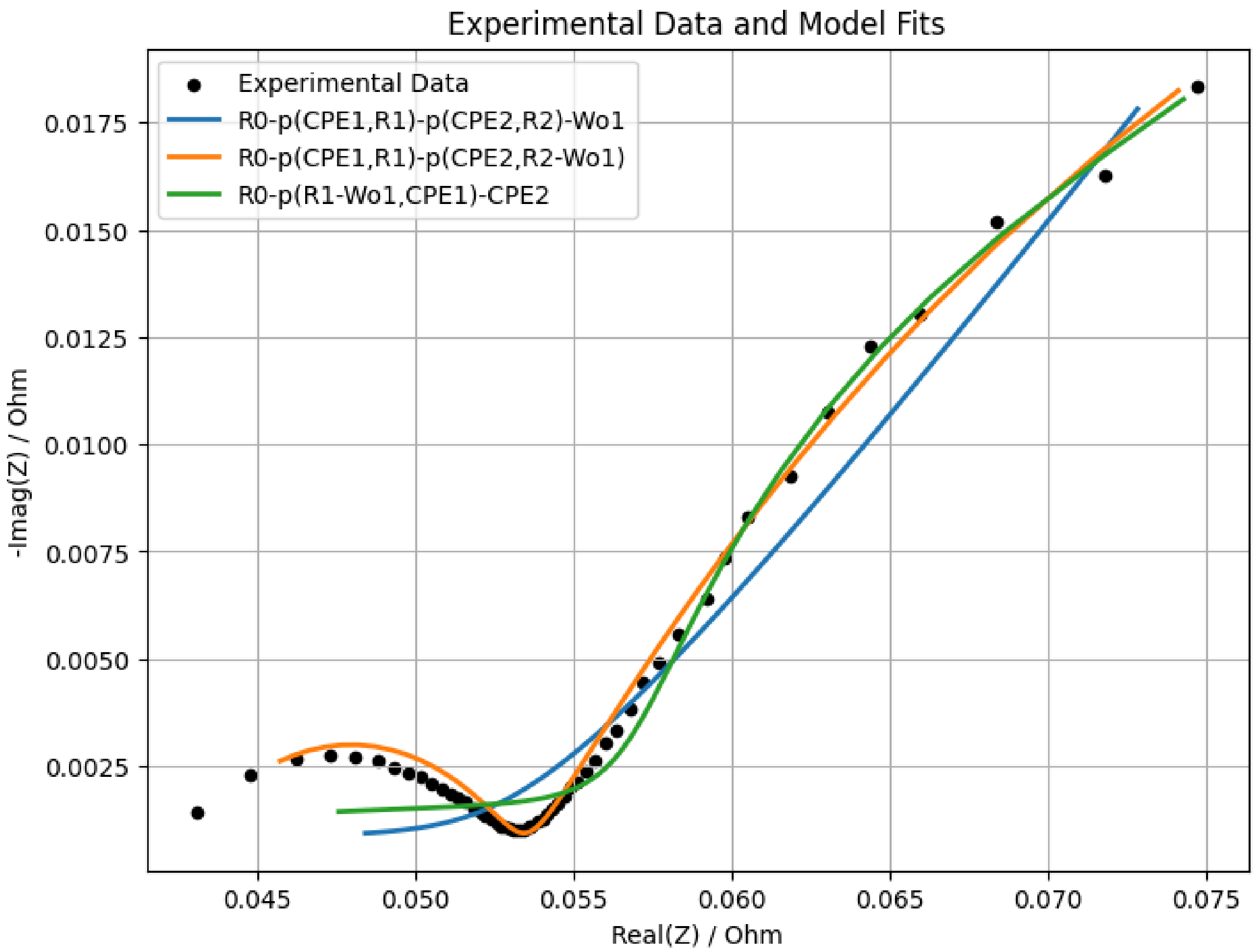
Unfortunately, with limited time and little electrical background, existing research was leant heavily upon. One paper used data from lithium-ion batteries, tested a selection of circuit models, and illustrated similar appearing spectrum (Zhu et al., 2019). However, this paper used machine learning methods to fit their circuit models, which wasn't feasible for our dataset size. A wide array of circuits were tested, although most were clearly not valid and were discarded. The circuit model described by Westerhoff mentioned in the introduction to this section was also tested, but its result was a poor fit, so has not been included.

Warning: This cell contains a very circuit that is very slow to fit

Show code

Run remaining circuits and plot

Show code




```
Parameters for circuit R0-p(CPE1,R1)-p(CPE2,R2)-Wo1: [7.49916080e-07 3.174680(
 3.17396326e+01 1.36182191e-02 1.93565542e-01 3.40996328e-01
 3.52417383e+03]
MSE for circuit R0-p(CPE1,R1)-p(CPE2,R2)-Wo1: 2.57470065480461e-06
Parameters for circuit R0-p(CPE1,R1)-p(CPE2,R2-Wo1): [4.23429014e-02 4.140961
 1.61843957e+02 5.97300994e-01 1.03733005e-01 1.32517647e-10
 1.95509700e+02]
MSE for circuit R0-p(CPE1,R1)-p(CPE2,R2-Wo1): 3.2715179061232345e-07
Parameters for circuit R0-p(R1-Wo1,CPE1)-CPE2: [1.24613037e-26 1.68049194e-02
 4.00558372e+02 1.00000000e+00 1.74534879e+01 1.92393202e-02]
MSE for circuit R0-p(R1-Wo1,CPE1)-CPE2: 1.2559715164343796e-06
```

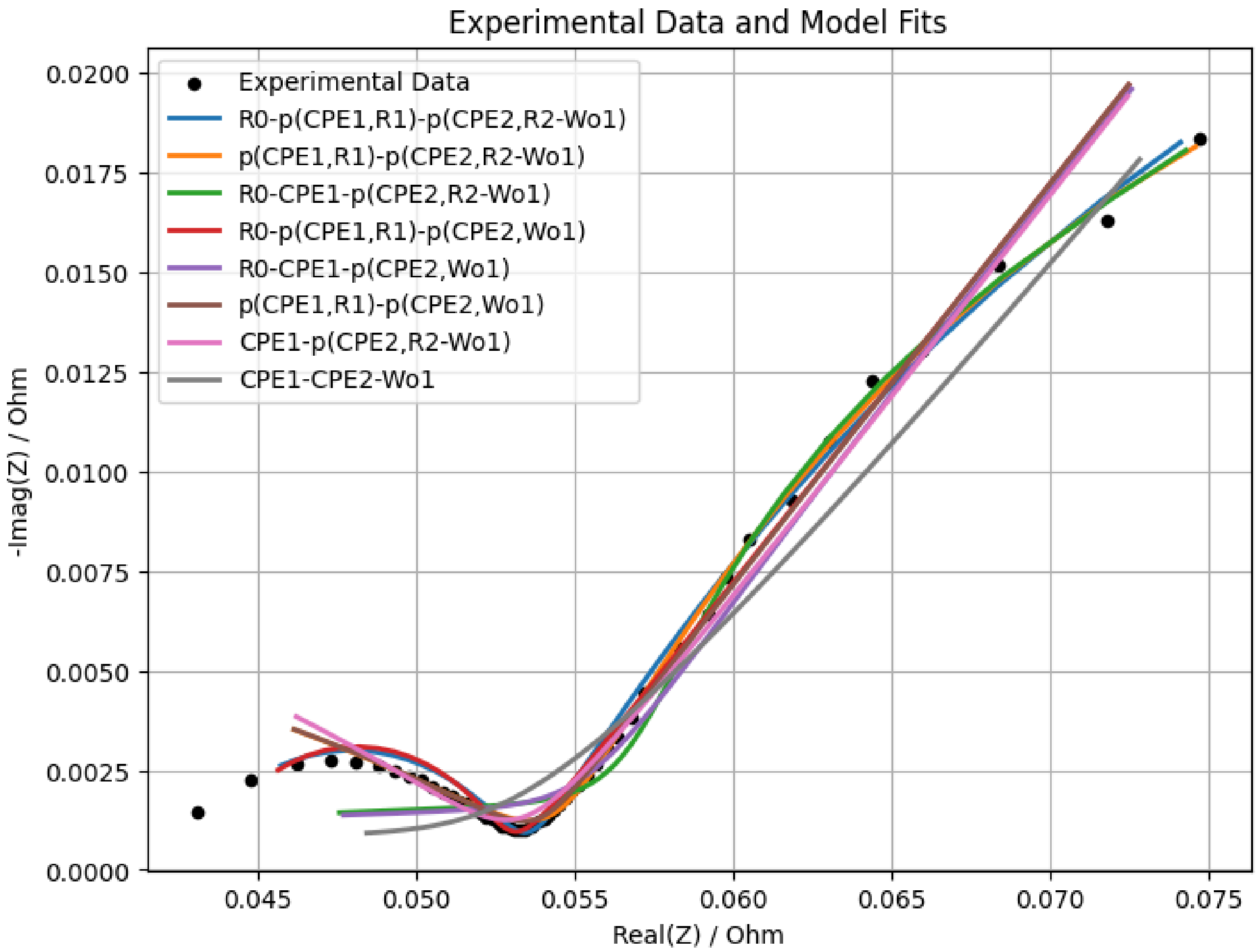
From this analysis, it was found that impedance.py is streamlined and easy to use, but its very slow for complex circuit models. While the slow circuit didn't turn out to be a good fit, it is a good illustration of how testing new circuits can be very time consuming with this software, and it's not clear which circuits will take longer or shorter to run. There is also little control over how it runs, so it doesn't seem to be the best option to use. One useful learning, was that the R0-p(CPE1,R1)-p(CPE2,R2-Wo1) circuit gave the best fit, with an MSE of just 3.27x10^-7. The parameters found were:

- **R0**: 4.23×10^{-2} Ω
- **CPE1**: Q_1 = 4.14×10^{-1}, n_1 = 6.26×10^{-1}
- **R1**: 1.11×10^{-2} Ω
- **CPE2**: Q_2 = 1.62×10^{2} Ω, n_2 = 5.97×10^{-1}
- **R2**: 1.04×10^{-1} Ω
- **Wo1**: A = 1.33×10^{-10} Ω, B = 1.96×10^{2} Ω

These are all fairly reasonable parameters, and the smallest resistances R0, R1 and R2 suggest they may not be needed. Next, we can try all permuations of removing R0, R1 and R2 or a combination of them, to see which plots are still accurate.

Try removing R0, R1, and R2

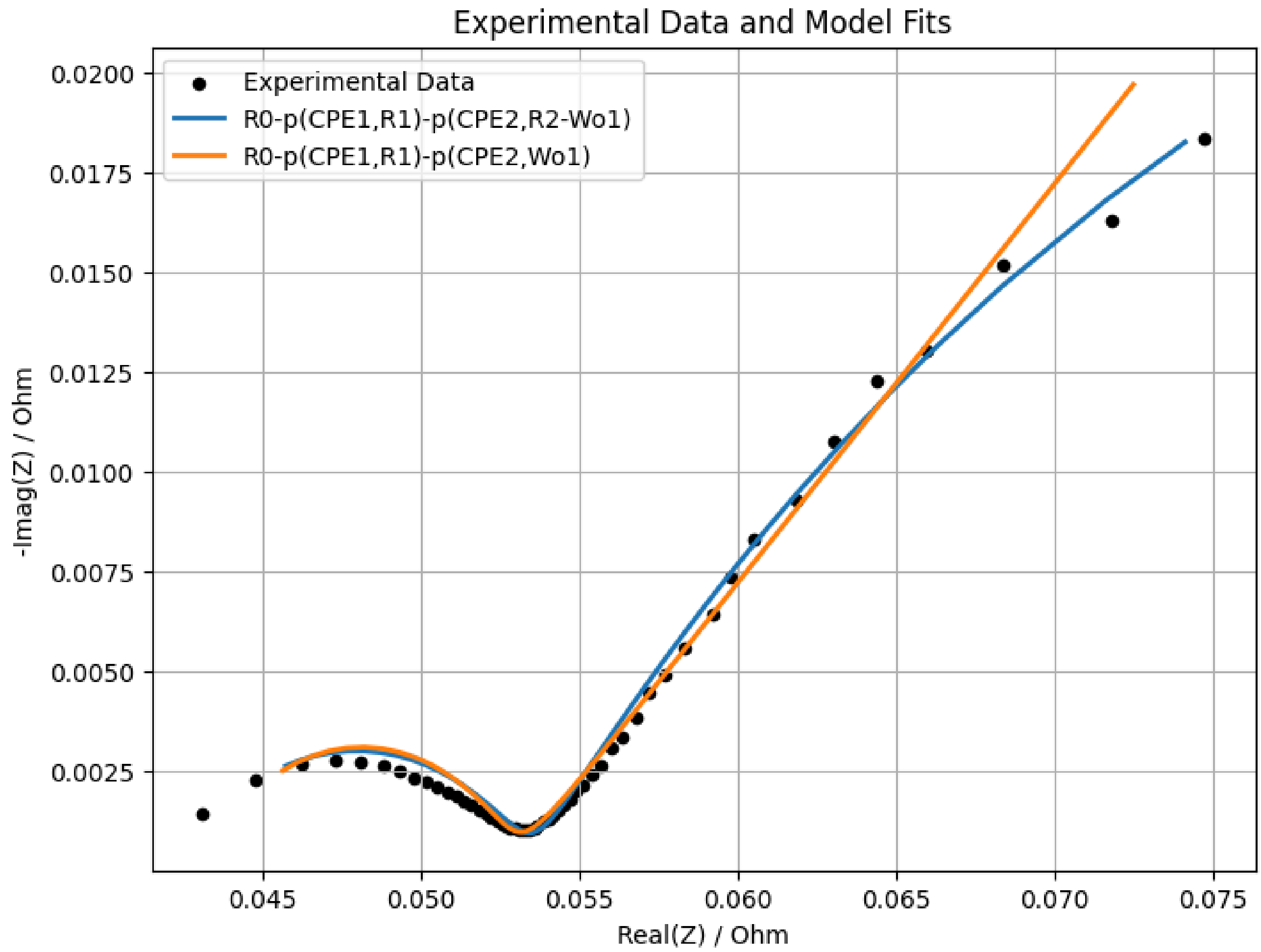
Show code



Evidently, some combination of R0, R1 and R2 is required, as many of the graphs no longer fit. This is interesting, as it highlights which components contribute to which aspects of the plot. Only one remains a good option even with the simplification, R0-p(CPE1,R1)-p(CPE2,Wo1). Note, the original circuit without any resistors removed is shown on the plot in blue.

Compare the two results

Show code




```
Parameters for circuit R0-p(CPE1,R1)-p(CPE2,R2-Wo1): [4.23429014e-02 4.140961
 1.61843957e+02 5.97300994e-01 1.03733005e-01 1.32517647e-10
 1.95509700e+02]
MSE for circuit R0-p(CPE1,R1)-p(CPE2,R2-Wo1): 3.2715179061232345e-07
Parameters for circuit R0-p(CPE1,R1)-p(CPE2,Wo1): [4.31592471e-02 2.36886911e-
 7.27052732e-01 1.00000000e+00 2.07027835e-01 9.45524134e+02]
MSE for circuit R0-p(CPE1,R1)-p(CPE2,Wo1): 5.06461259269949e-07
```

While the simplified circuit captures the features well for high frequencies, it performs less well for low frequencies. This shows that even though the resistance values are small, every component is needed. R0-p(CPE1,R1)-p(CPE2,R2-Wo1) will be used for further analysis.

While this package worked well, it does give a bit less control as you are working with a preprescribed package. Next, the previous plots were attempted to be replicated using scipy.optimize.

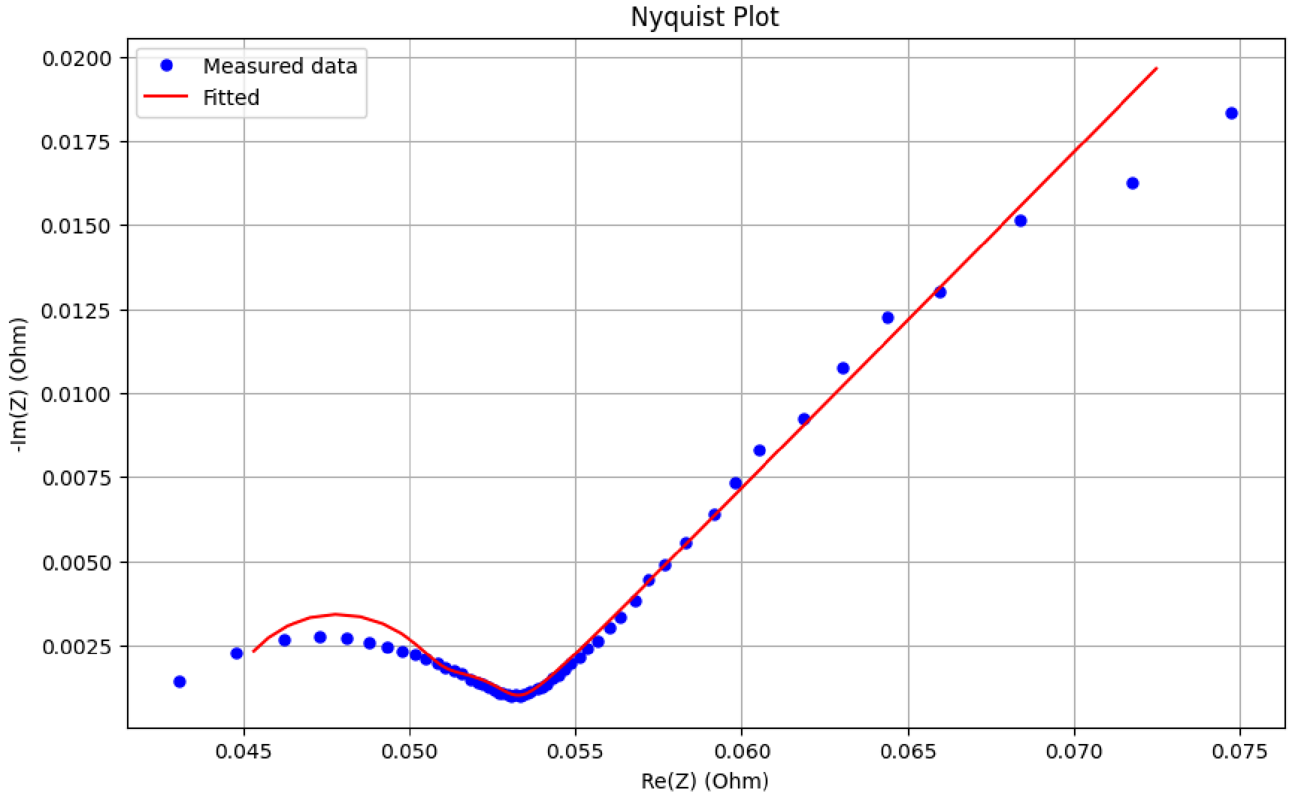
Define the components and the circuit model

Show code

Fit model and plot

Show code

```
Optimized parameters: [0.04439674 0.02429772 1.00118993 0.00650554 0.99996657
 0.00199657 0.00671809]
Mean Squared Error (MSE): 4.400509920632424e-07
```



Clearly, there is something going wrong with the fitting, as the curve is fitting well in the densely populated region but much less well on the left and right of this region where there is less data.

This is likely due to the MSE being evenly weighted, so it can reach the tolerance by fitting well in the area where there are more data points, and less well where there are fewer data points and hence less penalty is small. To help fix the issue, a weighting array `weights` was tested, that assigns a weight of 0.1 to the data points in the mid-section (between 0.0473 and 0.0583 Re(Z)/Ohm). The aim of this was to reduces the influence of these points on the MSE calculation, addressing the over-weighting issue. The code for this can be found in ECM_analysis.ipynb.

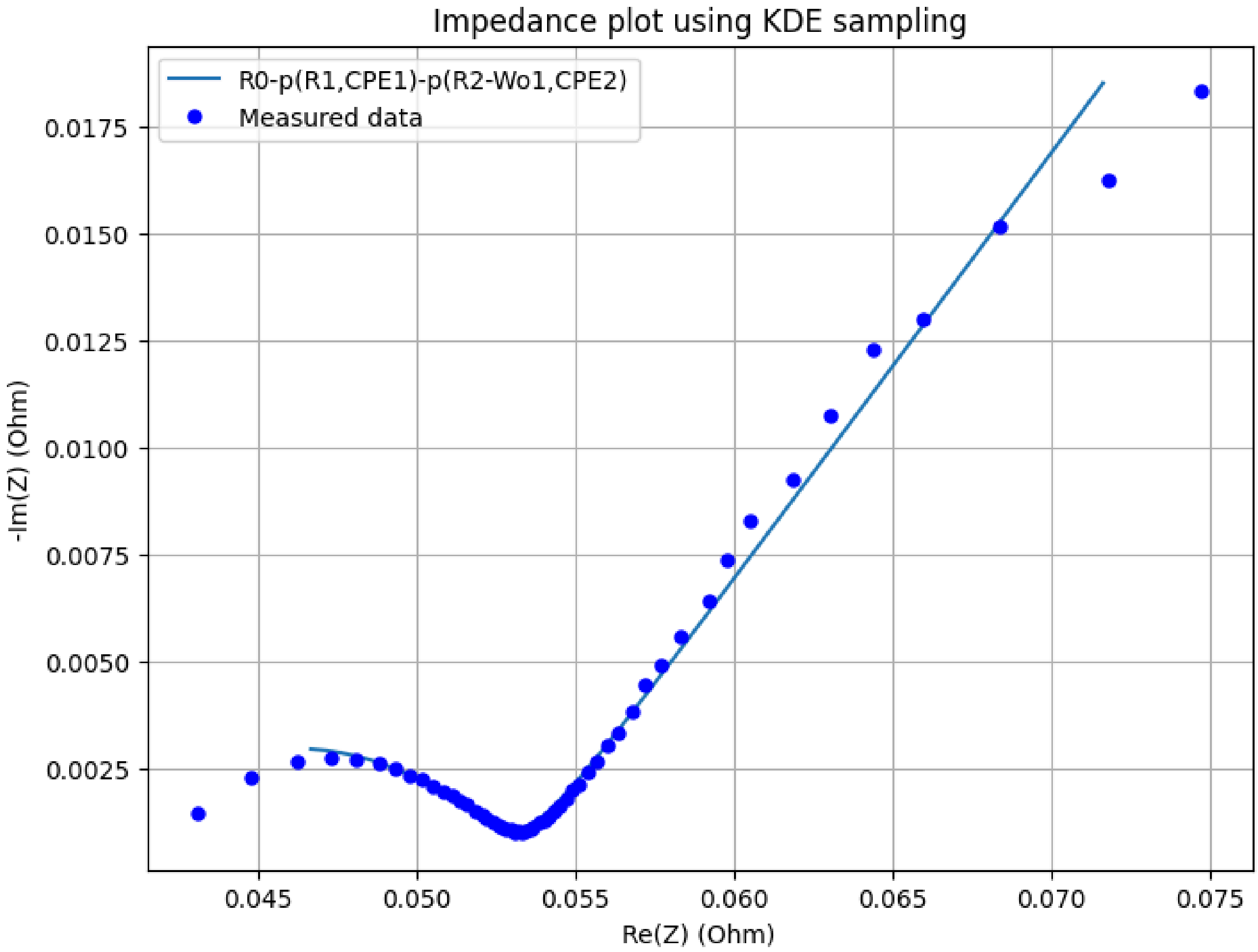
However, even with adjusted parameters, this did not solve the issue, producing a near-identical graph to the above. Instead, a more complex approach using SciKit's kernel density estimator (KDE) was tested (Pedregosa et al., 2011). The idea of this is to produce a probability density function that describes where the points lie, so that the sampling can be performed in a manner that is proportional to the density of data. This way, the data is evenly weighted in the optimisation algorithm.

In the code cell below, the `sampling_ratio` can be moved between 0+ and 1, with 1 being every data point is included and 0 meaning no data points are selected.

Plot using KDE

Show code

```
Optimized parameters: [3.88545480e-02 5.00069700e-01 5.00322135e-01 1.4115356
 5.00000195e-01 4.99998661e-01 1.79148217e-04 6.34954671e-03]
Mean Squared Error (MSE): 8.462275516625079e-07
```



This resulting fit is much better. The MSE is 8.46 × 10^-7, and the parameters for this circuit are:

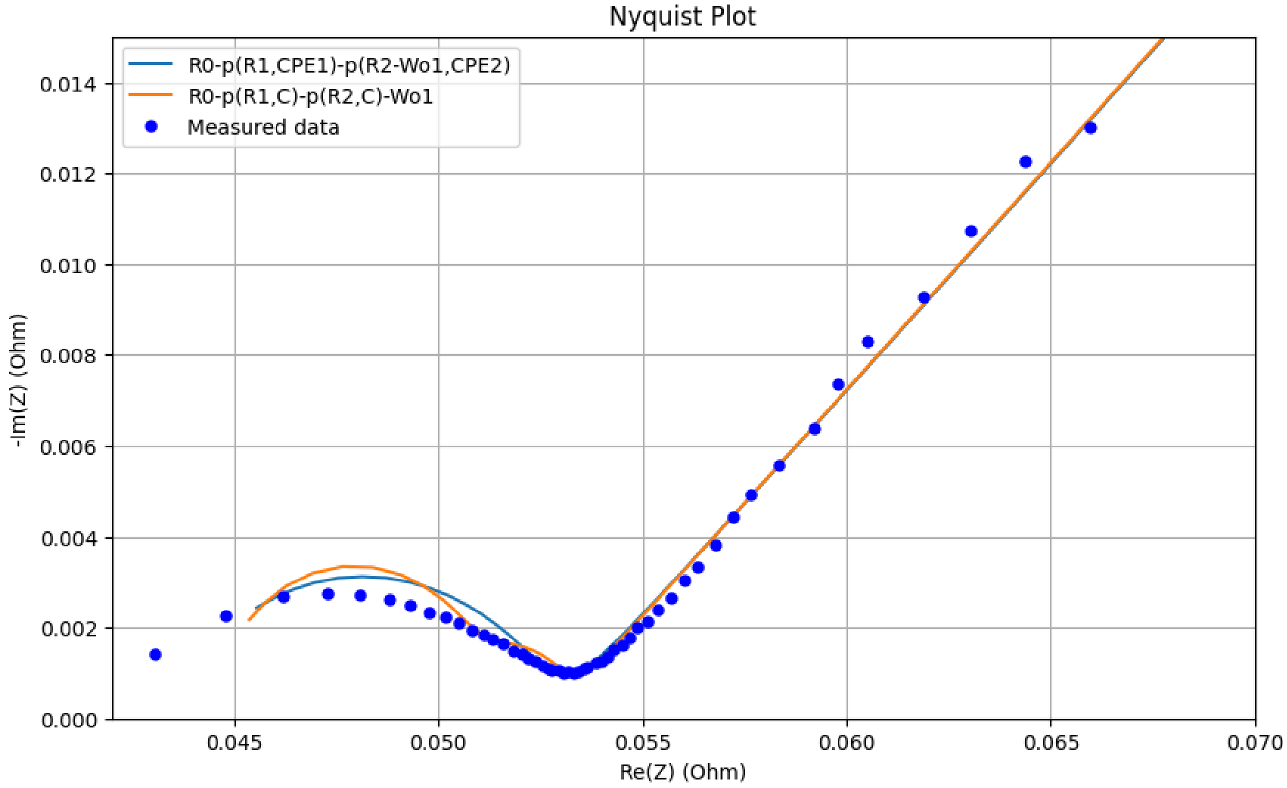
- **R0**: 3.88 × 10^-2 Ω
- **CPE1**: Q_1 = 5.00 × 10^-1, n_1 = 5.00 × 10^-1
- **R1**: 1.41 × 10^-2 Ω
- **CPE2**: Q_2 = 5.00000101e-1 Ω, n_2 = 5.00 × 10^-1
- **R2**: 1.87 × 10^-4 Ω
- **Wo1**: A = 6.33 × 10^-3 Ω, B = 1.33 × 10^-10 Ω

One effect that has appeared is that there is no generated fitting for the first few data points, as seen on the left hand side of the plot. There were suspicions that this was due to the sampling, but when when the first 5 data points are ensured to be included, the fit is much less tight, as can be seen below. Note the number of data points to use can be changed in the final line of the box (n), along with whether KDE is used and the sampling ratio, as before.

Force the use of the first 5 data points, compare to other circuit models

Show code

```
Optimized parameters for R0-p(R1,CPE1)-p(R2-Wo1,CPE2): [4.33535130e-02 1.3044
 2.20011469e-01 7.31102243e-01 9.21923832e-03 6.41386254e-03]
Optimized parameters for R0-p(R1,C)-p(R2,C)-Wo1: [0.04455238 0.00633373 0.027
```

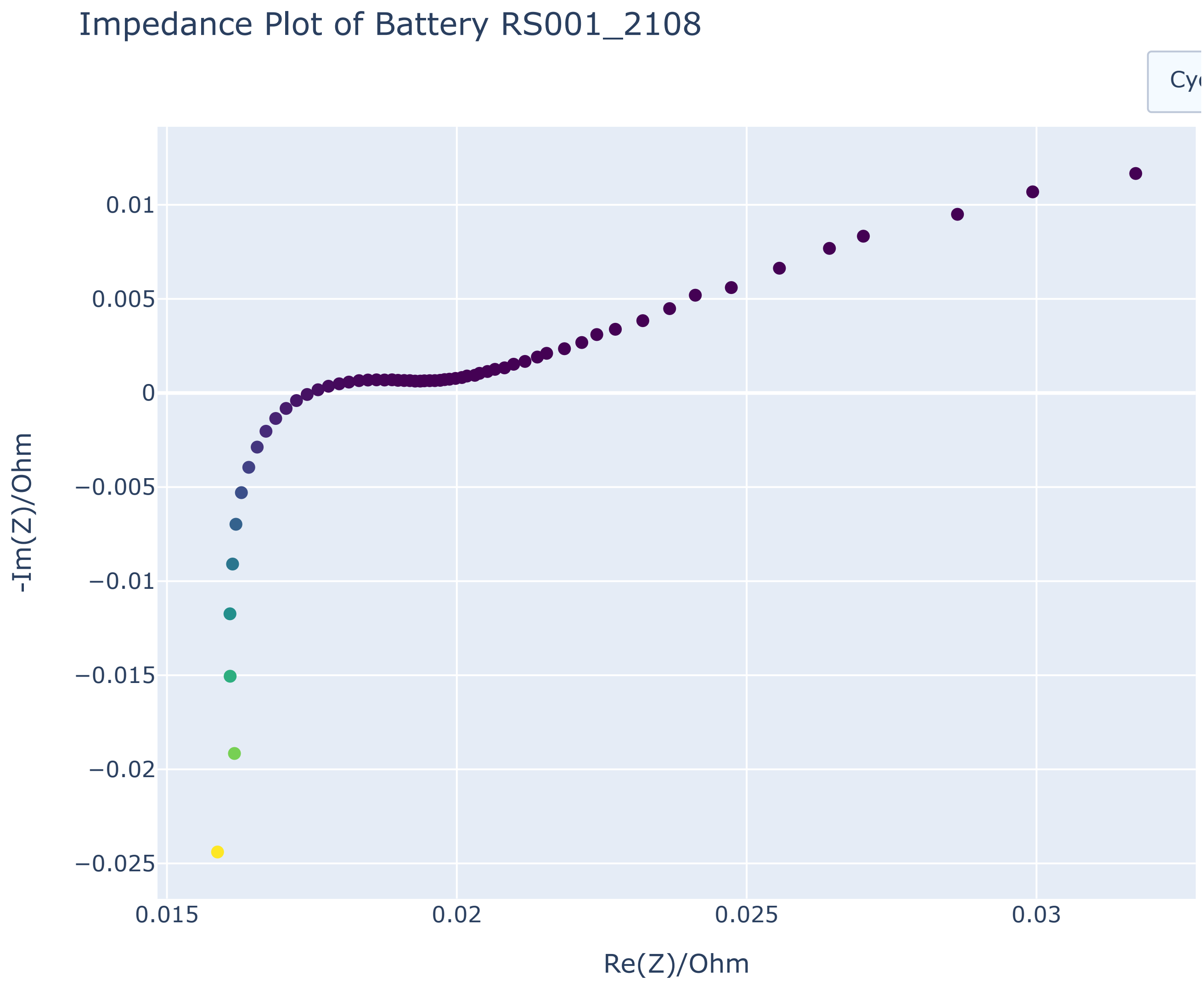


While the line does project further than before, the graphs fit less well as a result. This is not expected, so this should be investigated further.

Finally, this circuit model will be applied to two different batteries, the SG and the RS.

Extract the specific data required and plot for RS001

Show code

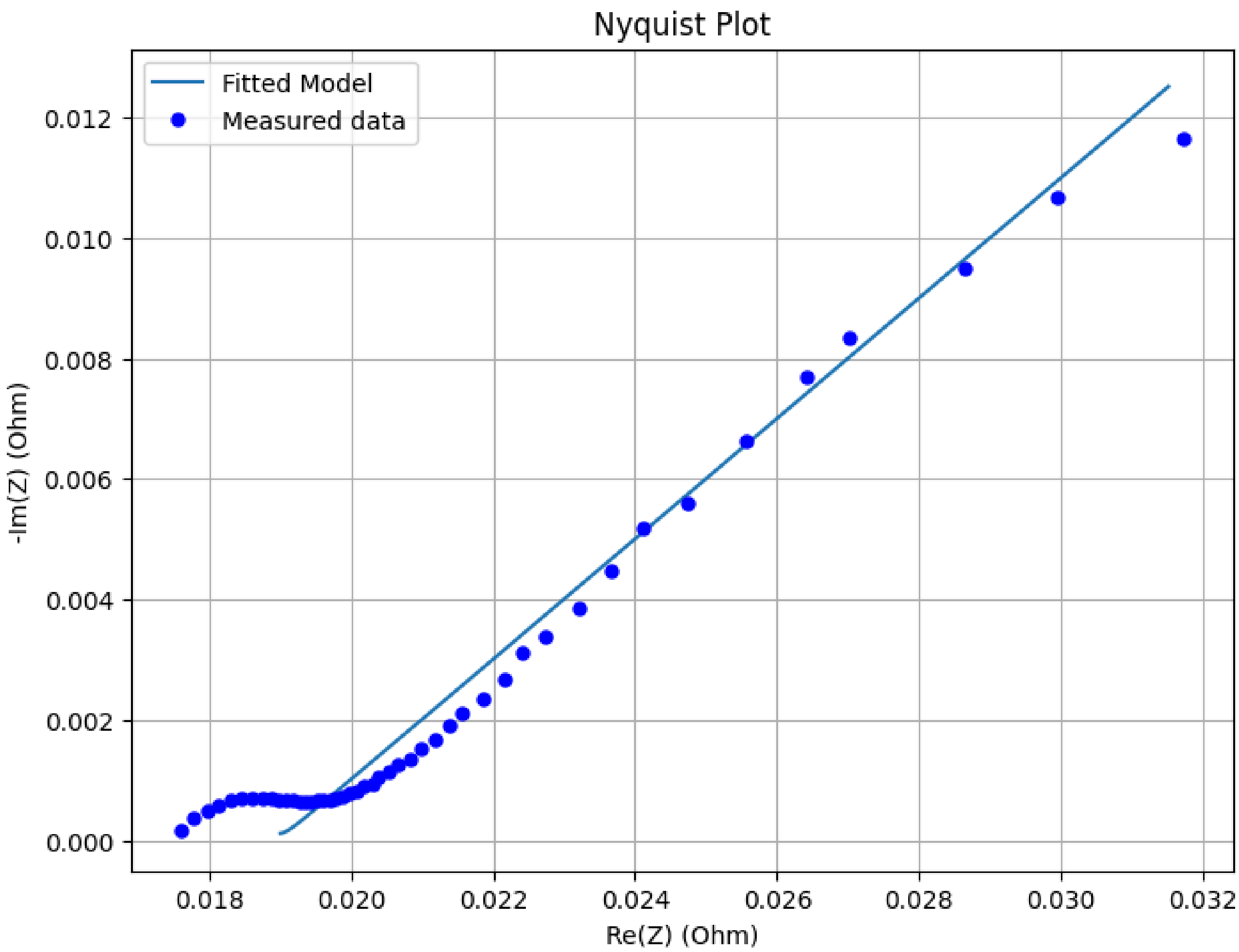


Above is a plot of the chosen cycle: RS001, 21/08/2023, cycle 103, 33% SOC.

Attempt to fit previous circuit model

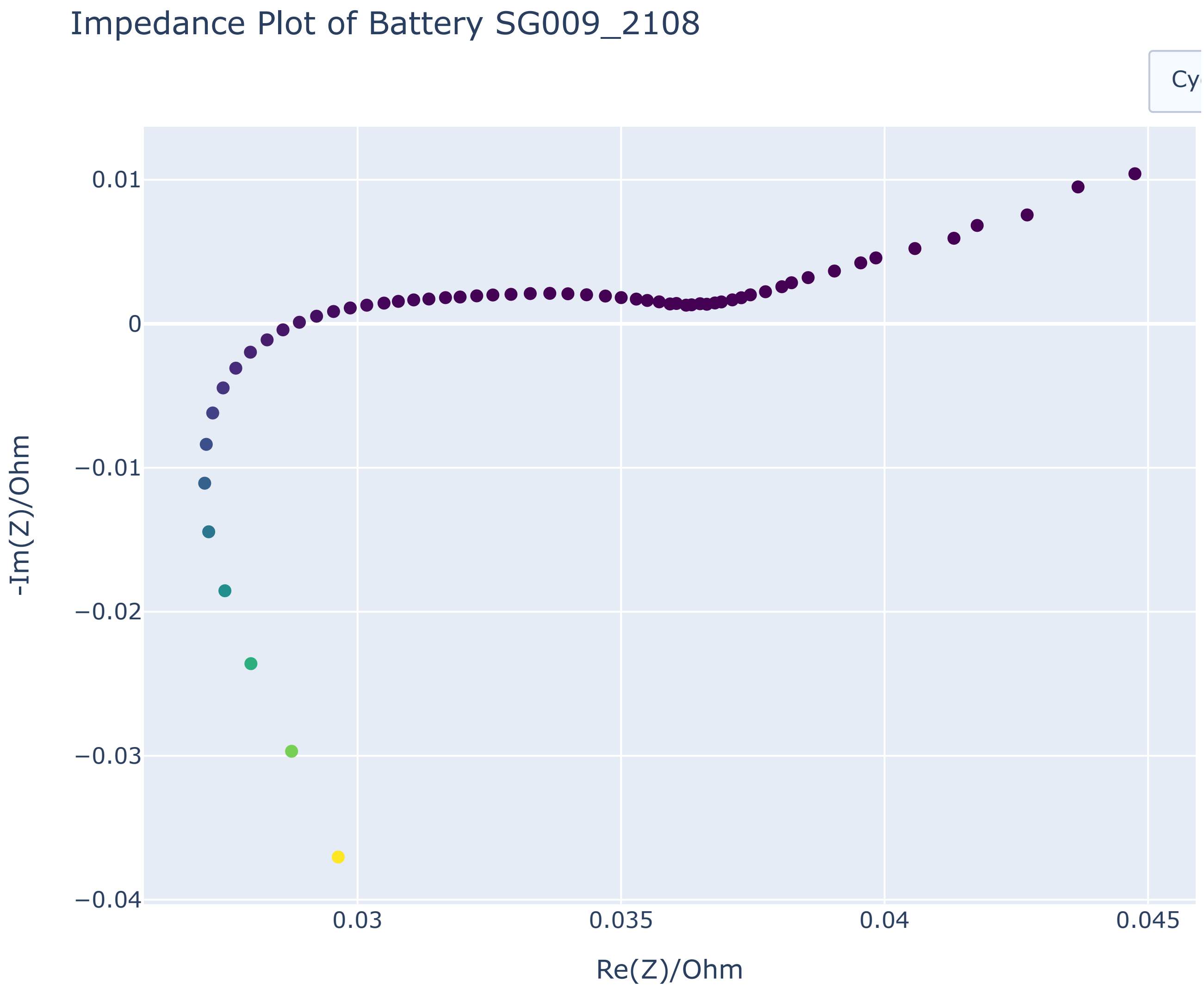
Show code

```
Optimized parameters: [1.72181529e-02 5.00010362e-01 5.00022746e-01 1.7772977
 5.00000645e-01 5.00001057e-01 1.00000000e-06 4.29083101e-03]
Mean Squared Error (MSE): 3.0273255681617646e-07
```



Extract the specific data required and plot for SG009

Show code

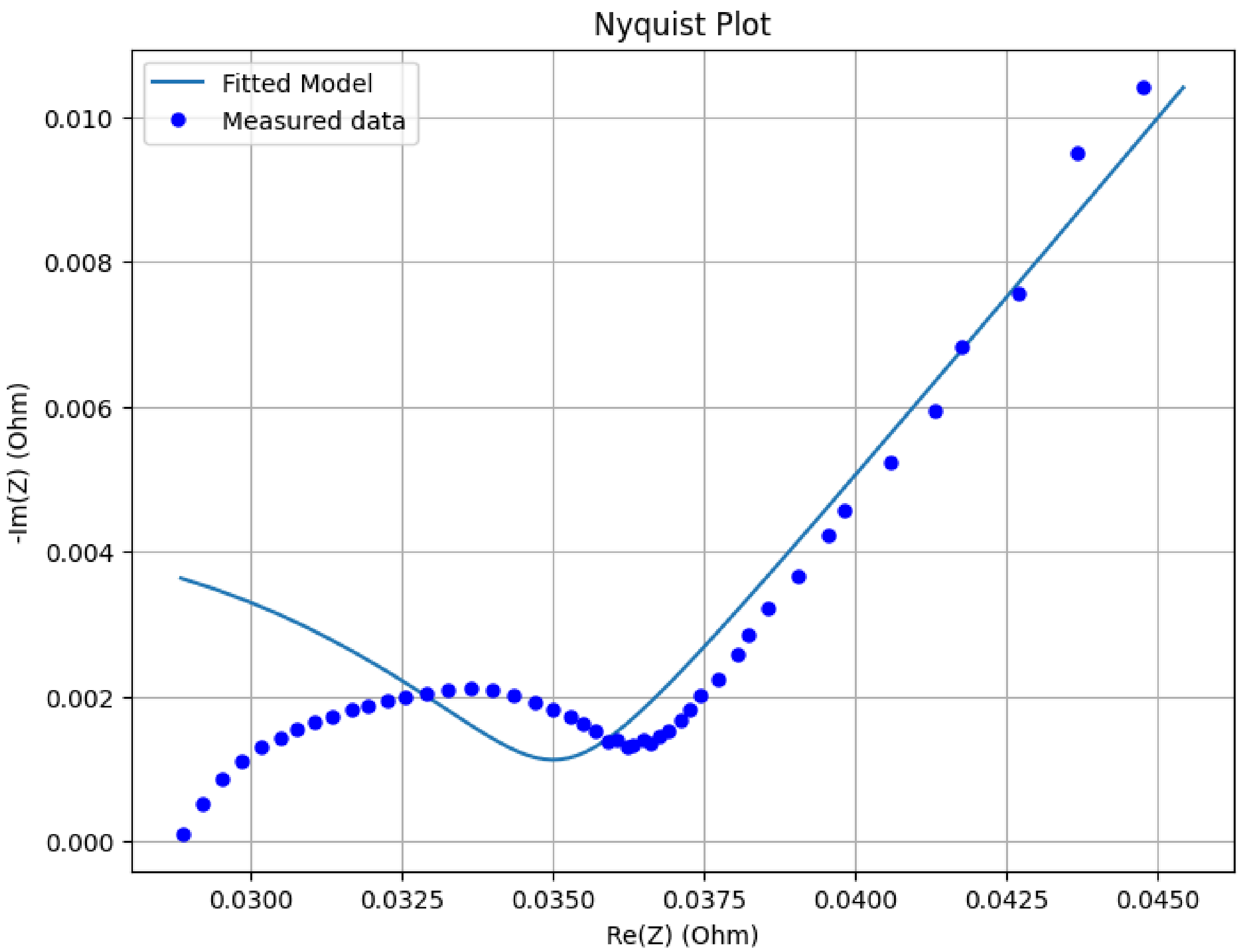


Above is a plot of the chosen cycle: SG009, 21/08/2023, cycle 106, 33% SOC.

Attempt to fit previous circuit model

Show code

```
Optimized parameters: [1.57016122e-02 5.00087183e-01 5.00225227e-01 1.9383199
 5.00000703e-01 4.99997134e-01 1.00000000e-06 3.55455184e-03]
Mean Squared Error (MSE): 1.284934766774553e-06
```



It is clear for both cells that the function cannot fit to the new shapes, at least not without tailoring the initial guesses, which is not ideal.

Challenges

Clearly, one key challenge is that an ECM that fits well for one battery chemistry does not fit well with other batteries. Another challenge is the fact that EIS spectra are different for different states of charge, with the 33% test shapes differing from the 66% test shapes. The level of charge may need to be set.

In summary, there lots of different software packages that can be used to do this fitting, with software being very specific to EIS, while others, such as scipy, are designed for general optimisation. First Impedance.py was used, a python package designed specifically for this purpose, which gave a great fit, but was extremely slow. Next we tried SciPy, which was much quicker but worse fit, as the data in the middle of the plot is much more densely populated and is incorrectly weighting the result. This plot was improved upon by something called Kernel Density

Estimation. The idea of this was to sample data points proportionally to how close together they are, and this resulted in an improved fit. Overall, some close fitting was achieved with a particular circuit model (R0-p(R1,CPE1)-p(R2-Wo1,CPE2)), but this did not translate to close fitting on other battery types, at least for the same initial guesses. Providing alternative initial guesses may improve the results, but tailoring them too closely to the battery type defeats the purpose of a universal model. Unfortunately, there was not time to test pyEIS, so it is recommended that if analysis in this area is continued then this packaged should be explored too.

Recommendations for future work

We believe that EIS is a more practical method for gathering data on cell health, as it is much quicker and less costly. It is possible to fit ECM models to EIS data, but further work is needed to assess model generality, computation time and prediction error. In particular, we recommend continuation with the following tasks:

1. Investigate further whether the ECM found is valid for the other battery types, find the error quantitively
2. Investigate how the parameters change with SOC and cycle number
3. Develop an algorithm that takes an EIS spectrum, fits the parameters of the ECM found earlier, and returns the error between the expected parameters and the actual parameters. Use the research on the physical link between ECM components and cell components to output an indication of what areas of the cell have unusual readings and therefore may be degrading, and to what extent. This may be tricky without any data for this, so we might need to find some online or create some.
4. Investigate whether it is possible to estimate current capacity of a cell from a single EIS reading, by finding the ECM and virtually cycling the model, producing a curve, and calculating capacity from that. This could be really powerful if possible, but would require strong understanding of how the model parameters change as the cell charges/discharges.

Future research could also consider new ways of gathering data. This could include crowdsourced, real world data or mutually beneficial collaborations with manufacturers. This may prove difficult due to many devices being limited to measuring whole battery characteristics, obscuring the influence of individual cell failures.

As reflected by Microsoft in their BatteryML work, it is unknown just how relevant laboratory cycling data is to real-world loading conditions. Models often standardise the cells loading conditions, resulting in an unusual contradiction where a model is trained on controlled laboratory data, only to be deployed on uncontrolled and variable loading scenarios. This is precisely why we recommend that further research investigates alternate means of gathering cell data.

A model that we did not consider was based on work from Peter M. Attia et al (2022). They show that the "knee" behaviour of cell degradation is easier predicted than SoH. A possible future approach could consider a classifier model that judges whether a cell will pass its "knee" point in the near future. This may prove easier than a true capacity estimate, and is useful for 2nd life classification as a battery which has passed this point is unlikely to be useful for much longer. For example, a cell may reach 60% SoH but be usable, whilst another that has suffered a particularly harsh life may reach the knee point at 95% SoH.

Further detail on these recommendations and considerations can be found in the research summary on the wiki.

Conclusion

The EIS analysis was particularly successful, with the provision of simple code for cleaning the data. The generation of a synthetic dataset was an interesting exercise, and introduced some new learnings on the degradation of batteries when not in use. The ECM fitting results were promising for the initial battery type, but the written algorithms failed to fit to other battery types. The investigation of pyEIS, a python package for ECM fitting, is a recommended next step, alongside further investigation into the relation between ECM components and the physical