

DEPARTMENT OF MECHANICAL ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROPAR  
RUPNAGAR-140001, INDIA



## CP301: Developmental Engineering Project Report

Topic: Creating an IoT Setup for Battery Digital Twin

*Under the guidance of -*

**Dr. Manish Agrawal**

**Assistant Professor, Department of Mechanical Engineering**

**IIT Ropar**

**&**

**Dr. Jagpreet Singh**

**Assistant Professor, Department of Computer Science Engineering**

**IIT Ropar**

Submitted by:

**Jai Kumar Prajapati (2022MEB1316)**

**Kultaj Singh (2022MEB1321)**

**Sahil Nitin Mhapsekar (2022MEB1324)**

## Acknowledgement

---

We extend our heartfelt gratitude to our supervisors, **Dr. Manish Agrawal** and **Dr. Jagpreet Singh**, for providing us with the opportunity to work on this interdisciplinary project. Their profound expertise, insightful feedback, and unwavering support have been instrumental in overcoming technical challenges and driving the project forward. As we progress, we are committed to building upon the foundation laid under their mentorship.

We would also like to acknowledge the invaluable contributions of our peers and colleagues. Their constructive feedback, collaborative spirit, and thought-provoking discussions have significantly enriched our project. Their shared insights have not only refined our approach but also fostered an environment of creativity and innovation.

Furthermore, we express our sincere appreciation to the researchers and scholars whose pioneering work has served as a source of inspiration and guidance for our project. Their contributions to the field have provided a strong foundation upon which we have built our research.

This project has been an immensely rewarding journey, and we are excited to continue exploring advancements in battery digital twins and IoT technologies, contributing to the ever-evolving landscape of energy storage and management.

# Contents

---

Acknowledgement .....	2
Introduction .....	4
Literature Review.....	5
Electrochemical Impedance Spectroscopy .....	5
Impedance Metrology and Calibration .....	5
EIS Measurement Techniques .....	5
Interpretation Methods.....	6
Validation Approaches.....	6
SoC and SoH Calculation .....	7
ECM Parameters from Typical Nyquist Plot.....	7
Kalman Filter .....	8
Adaptive Extended H-Infinity Filter (AEHF).....	9
Cloud Implementation of SoC and SoH Estimation .....	11
Methodology .....	13
Measurement Techniques .....	13
Interpretation Methods.....	13
Validation Approaches.....	13
Future Directions .....	13
Components.....	14
Battery .....	14
Charging/Discharging Module .....	15
Power Adapter.....	15
Power Supply for Microcontroller .....	15
STM32G473xE Microcontroller.....	16
TI C2000 Microcontroller .....	17
Implementation of EIS on STM32 .....	18
Methodology .....	19
Procedure.....	19
Observations and Results .....	20
Major Problems Faced .....	21
Progress we could make .....	22
Conclusion.....	23
Future Work.....	24
References .....	24

## Introduction

---

The development of an IoT setup for a battery digital twin represents a critical step toward addressing the growing need for advanced battery management systems. Lithium-ion batteries (LIBs) are central to modern applications, from electric vehicles to renewable energy storage, but their performance and longevity depend heavily on accurate real-time monitoring of internal states such as State of Charge (SoC) and State of Health (SoH). Traditional monitoring methods, such as voltage and current measurements, provide limited insights into the complex electrochemical processes within batteries, often failing to detect early signs of degradation. To overcome this limitation, we have focused on creating an IoT-enabled system that integrates Electrochemical Impedance Spectroscopy (EIS) with a battery digital twin—a virtual replica that simulates and predicts battery behaviour in real time.

EIS measures a battery's impedance response across a range of frequencies, offering detailed insights into internal resistance, charge transfer kinetics, and diffusion processes. By incorporating EIS into the digital twin framework, we aim to enable real-time monitoring, diagnostics, and optimization of battery performance. This approach enhances state estimation accuracy and facilitates predictive maintenance, ultimately improving safety and extending battery life.

We begin with a review of EIS literature, focusing on measurement techniques, data interpretation, and validation. With this information, we elect to perform Potentiostatic EIS (PEIS) for high-resolution impedance spectra and Dynamic EIS (DEIS) to capture dynamic behaviour during charge/discharge cycles. For data interpretation, we choose Equivalent Circuit Modelling (ECM) and Distribution of Relaxation Times (DRT), supported by the Kramers-Kronig Relations for data validation. The IoT setup design, optimized for high-speed signal processing and impedance measurement, is complete, and component acquisition is now in progress.

In the coming weeks, we will assemble and calibrate the equipment, then validate EIS data accuracy. Once operational, the system will integrate with a cloud-based digital twin platform, using machine learning for real-time analysis, state prediction, and degradation detection, ensuring adaptability and accuracy. In summary, this project develops an IoT setup for a battery digital twin, leveraging EIS for real-time monitoring and optimization. By combining advanced measurement, robust analysis, and cloud analytics, we aim to improve battery performance, safety, and longevity in applications like electric vehicles and renewable energy storage.

## Literature Review

### Electrochemical Impedance Spectroscopy

Internal impedance is a critical parameter in lithium-ion batteries (LIBs), influencing voltage, rate capability, efficiency, and capacity. As LIBs age, impedance increases due to degradation of electrode materials, electrolyte, and electrical contacts, as noted by Meddings et al. (2020). Electrochemical Impedance Spectroscopy (EIS) has emerged as a non-destructive diagnostic tool for LIB characterization, enabling quality assurance, state estimation (SOC, SOH, SOF), and second-life applications. However, challenges persist in ensuring measurement quality, reproducibility, and reliable interpretation of impedance data, necessitating robust metrological practices and advanced analytical techniques.

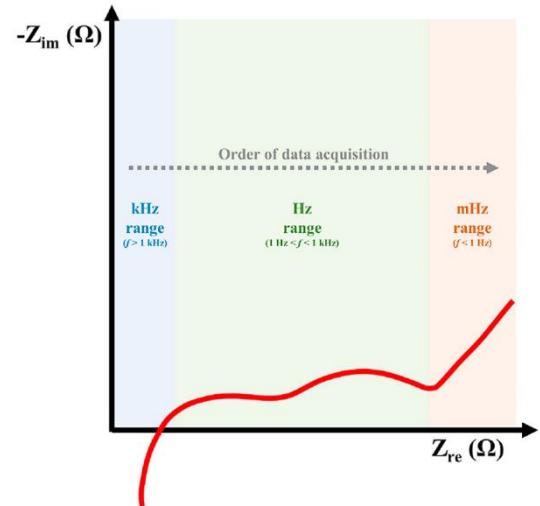


Fig. 1. Graphical representation of a typical Li-ion cell EIS measurement presented in a complex impedance plot (i.e., Nyquist plot).  $f$  is the frequency and, the order of data acquisition refers to the order in which frequencies are typically interrogated.

### Impedance Metrology and Calibration

Accurate impedance measurement requires traceability to the International System of Units (SI), as emphasized by Meddings et al. (2020). Proper measurement setups, including Open/Short/Load (OSL) compensation, are essential for minimizing errors, particularly in low-impedance systems like LIBs. Factors such as magnetic coupling and test fixture design significantly influence measurement uncertainty, highlighting the need for careful experimental design and calibration protocols.

### EIS Measurement Techniques

EIS techniques vary in complexity and application:

- **Conventional EIS:** Involves applying a sinusoidal perturbation (current or voltage) and measuring the response. Galvanostatic EIS (GEIS) is often preferred for commercial LIBs due to its superior signal-to-noise ratio and compatibility with low-impedance systems (Meddings et al., 2020).
- **Dynamic EIS (DEIS):** Measures impedance during charge/discharge cycles, providing insights into dynamic behavior but requiring careful handling of non-stationarity.
- **Non-Linear Techniques:** Non-linear frequency response analysis (NFRA) and non-linear EIS (NLEIS) offer additional information on electrochemical processes but demand sophisticated interpretation.
- **Multi-Sine and Time-Domain Methods:** These approaches reduce measurement time, particularly for low-frequency measurements, but may compromise signal-to-noise ratio.
- **Single-Frequency and DC Resistance Measurements:** Useful for rapid quality assurance and state estimation, though they provide limited information compared to full-spectrum EIS.

## Interpretation Methods

Interpreting EIS data involves translating impedance spectra into meaningful insights about battery processes:

- **Equivalent Circuit Modelling (ECM):** The most common approach, ECM uses electrical circuit elements (resistors, capacitors, inductors, CPEs, Warburg elements) to represent physical processes. However, over-parameterization and the lack of direct correlation between circuit elements and physical processes remain challenges.
- **Distribution of Relaxation Times (DRT):** DRT transforms impedance spectra into the time domain, identifying overlapping processes and guiding ECM development. It enhances the physical relevance of models but requires careful selection of regularization parameters.
- **Physics-Based Modelling:** Advanced models, such as the Barsoukov Transmission Line Model (TLM), integrate fundamental physical equations (e.g., Butler-Volmer, diffusion) to provide more accurate interpretations of impedance spectra. These models are computationally intensive but offer deeper insights into battery behavior.

## Validation Approaches

Validating EIS data is critical for ensuring reliability and accuracy:

- **Kramers-Kronig Relations:** These mathematical relationships validate impedance data by ensuring linearity, causality, and stationarity. Automated tools, such as the linear Kramers-Kronig transform, simplify this process.

$$Z'(\omega) = Z'(\infty) + \frac{2}{\pi} \int_0^{\infty} \frac{x Z''(x) - \omega Z''(\omega)}{x^2 - \omega^2} dx$$

and

$$Z''(\omega) = -\frac{2\omega}{\pi} \int_0^{\infty} \frac{Z'(x) - Z''(\omega)}{x^2 - \omega^2} dx$$

The equations shown above can be used to calculate the real part of the impedance from its imaginary part and vice versa but they can also be formulated in terms of the magnitude of the impedance and its phase. In both cases the calculated values can be compared to the measured ones and the degree of convergence describes the quality of the measurement.

- **Postmortem Analysis:** Involves disassembling aged cells and analysing electrode materials using techniques like SEM, XPS, XRD, and symmetric cell reconstruction. This approach validates EIS interpretations but requires meticulous sample handling to avoid contamination and artifacts.

Our selected methods - PEIS, DEIS, ECM, DRT, Kramers-Kronig Relations, and Machine Learning Integration - are tailored to the specific requirements of real-time continuous EIS on a 3000mAh battery undergoing repetitive charge/discharge cycles. These choices ensure accurate, efficient, and insightful characterization of the battery's impedance behaviour under dynamic operating conditions.

We rejected alternatives such as **Non-Linear Techniques (NFRA, NLEIS)** and **Physics-Based Modelling (TLM)** due to their complexity and computational intensity, which are not aligned with our goal of real-time monitoring. Similarly, **Postmortem Analysis** was excluded as it is not feasible for continuous, non-destructive monitoring.

By combining these methods, we aim to achieve a comprehensive understanding of the battery's performance and degradation mechanisms, paving the way for improved battery management and longevity. This approach not only addresses the technical challenges of real-time EIS but also aligns with the broader goal of advancing battery diagnostics and prognostics for commercial applications.

## SoC and SoH Calculation

### ECM Parameters from Typical Nyquist Plot

The **Equivalent Circuit Model (ECM)** is widely used for battery modelling due to its simplicity and effectiveness in capturing the dynamic behavior of batteries. The ECM parameters can be extracted from the **Nyquist plot**, which is obtained from **Electrochemical Impedance Spectroscopy (EIS)**. The Nyquist plot typically consists of a semicircle and a straight line, representing the charge transfer resistance and diffusion processes, respectively.

The **extended Thevenin model** is a common ECM used for battery modelling, which includes:

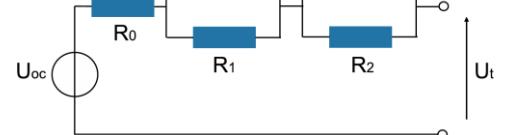
- **Ohmic resistance ( $R_0$ )**: Represents the internal resistance of the battery.
- **Polarization resistances ( $R_1, R_2$ )**: Represent the charge transfer and diffusion resistances.
- **Polarization capacitances ( $C_1, C_2$ )**: Represent the double-layer capacitance and diffusion capacitance.

The ECM equations are:

$$\begin{aligned}\dot{U}_1 &= -\frac{U_1}{R_1 C_1} + \frac{I}{C_1} \\ \dot{U}_2 &= -\frac{U_2}{R_2 C_2} + \frac{I}{C_2} \\ U_t &= U_{oc} - U_1 - U_2 - IR_0\end{aligned}$$

where:

- $U_t$  is the terminal voltage,
- $U_{oc}$  is the open-circuit voltage (OCV),
- $I$  is the current,
- $U_1$  and  $U_2$  are the voltage drops across the RC circuits.



**Schematic of the extended Thevenin model**

## Kalman Filter

The **Kalman Filter (KF)** is a recursive algorithm used for state estimation in dynamic systems. It is particularly useful for **State of Charge (SoC)** estimation due to its ability to handle noisy measurements and model uncertainties.

The Kalman Filter consists of two main steps:

The state estimate and estimate variance matrix be  $x_{t-1}$  and  $P_{t-1}$  at time  $t-1$  given by

$$x_{t-1} = \begin{bmatrix} SoC(\%) \\ Current(I) \end{bmatrix}$$

$$P_{t-1} = \begin{bmatrix} \sigma_{estimate}^2 & 0 \\ 0 & \sigma_{estimate}^2 \end{bmatrix}$$

Next in the prediction step, the change in SoC (%) is accounted for using the coulomb counting method, which is the charge taken out from the battery. This is represented as a transformation matrix  $F_t$  given by

$$F_t = \begin{bmatrix} 1 & -\frac{\Delta t}{10800} \times 100 \\ 0 & 1 \end{bmatrix}$$

The new state estimate  $x_t$  and estimate variance  $P_t$  after the prediction step is given by

$$x_t = F_t \begin{bmatrix} SoC(\%) \\ I \end{bmatrix}$$

$$P_t = F_t \begin{bmatrix} \sigma_{estimate}^2 & 0 \\ 0 & \sigma_{estimate}^2 \end{bmatrix} F_t^T$$

Next, in the measurement step, using the SoC-OCV mapping, the change in SoC is computed by multiplying the observation matrix with the state estimate. Here observation matrix would be an identity matrix as we are not performing any manipulation on the measurements.

$$\bar{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$y_t = \bar{H} \begin{bmatrix} SoC(OCV) \\ I \end{bmatrix}$$

where  $H_{bar}$  is the observation matrix, and  $y_t$  is the state measurement. Also, we consider noise in measurements, and this is an input of how much weight measurement is given than prediction while computing the final estimate.

$$\bar{R} = \begin{bmatrix} \sigma_{meas}^2 & 0 \\ 0 & \sigma_{meas}^2 \end{bmatrix}$$

A low value in observation noise  $R_{bar}$  indicates that we assign more weight to measurements than predictions. Finally, compute the next best estimate by computing Kalman Gain, multiply it with residual, and add it to the previous estimate. The residual vector  $z_t$  is given as the difference between the state measurement and state prediction.

$$z_t = y_t - \bar{H}x_t$$

$$S_t = \langle \bar{H}|P_t|\bar{H}^T \rangle + \bar{R}$$

The Kalman gain is computed using the residual variance, observation, and prediction variance matrices are given by

$$K_t = P_t \bar{H}^T S_t^{-1}$$

Using Kalman gain, the final state estimates  $x_t$  and state variance matrices are updated as given by

$$x_t = x_t + K_t z_t$$

$$P_t = (1 - K_t)P_t$$

The above steps are recursively applied to get the next best estimates from the previous estimate.

## Adaptive Extended H-Infinity Filter (AEHF)

The **Adaptive Extended H-Infinity Filter (AEHF)** is a robust state estimation algorithm that minimizes the worst-case estimation error. It is particularly useful for SoC estimation under model uncertainties and noise.

Considering the discrete-time model of a general non-linear system in a state-space form, we have

$$x_{k+1} = Ax_k + Bu_k + w_k$$

$$y_k = Cx_k + Du_k + v_k$$

Where  $u_k$  and  $y_k$  represent the system inputs and outputs,  $x_k$  denotes the system states,  $w_k$  and  $v_k$  are the system process noise and measurement noise, the matrices  $A_k$ ,  $B_k$ ,  $C_k$ , and  $D_k$  describe the system dynamics.

The SoC calculation using AEHF algorithm consists of three steps:

### 1. Estimation Step:

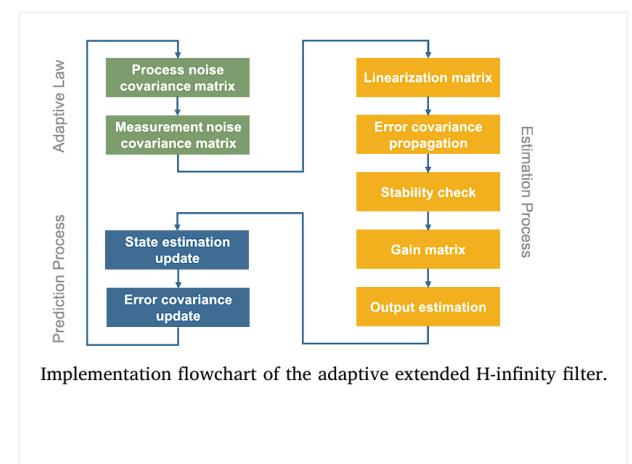
$$\hat{x}_k^- = A_{k-1}\hat{x}_{k-1}^+B_{k-1}$$

$$P_k^- = A_{k-1}P_{k-1}A_{k-1}^\top + Q_{k-1}$$

Where  $Q_{k-1}$  is the weighting matrix. Afterward, the observer stability criteria are checked by

$$(P_k^-)^{-1} - \theta_k^{-2}L_k^\top L_k > 0$$

Where  $\theta_k$  is the tuning parameter.



Furthermore, the gain matrices are updated as follows:

$$R_{e,k} = \begin{bmatrix} R_{k-1} & 0 \\ 0 & -\theta_k^2 I \end{bmatrix} + \begin{bmatrix} C_k \\ L_k \end{bmatrix} P_k^- [C_k^\top \quad L_k^\top]$$

$$H_k = P_k^- [C_k^\top \quad L_k^\top] R_{e,k}^{-1}$$

$$H_{s,k} = P_k^- C_k^\top (R_{k-1} + C_k P_k^- C_k^\top)^{-1}$$

Where  $R_{e,k}$  and  $H_{s,k}$  are transition matrices and  $H_k$  is the H-infinity matrix. The estimation of the system output  $y^k$  is provided by

$$\hat{y}_k = C \hat{x}_k^- + D u_k + v_k.$$

## 2. Prediction Step:

In the prediction process, the state and error covariance matrix for the next time step is calculated by

$$\hat{x}_{k+1} = A \hat{x}_k^- + B u_k + H_k e_k$$

$$P_k = \left( I - H_k \begin{bmatrix} C_k \\ L_k \end{bmatrix} \right) P_k^-$$

Where  $e_k$  represents the output estimation error as follows:

$$e_k = y_k - \hat{y}_k$$

## 3. Adaptive Step:

The tuning parameter  $\theta_k$  and the weighting matrices  $Q_k$  and  $R_k$  play important roles in H infinity filters and are adjusted in each time step as follows:

$$\begin{aligned} \theta_k &= \alpha \sqrt{\max(\text{eig}(P_k^-))} \\ R_k &= \frac{1}{N} \sum_{j=k-N+1}^k (e_j e_j^\top - C_j P_j C_j^\top) \\ Q_k &= \frac{1}{N} \sum_{j=k-N+1}^k (H_{s,k} e_k e_k^\top H_{s,j}^\top - A_{j-1} P_{j-1} A_{j-1}^\top) \end{aligned}$$

## SOC Estimation with AEHF

The SOC of the battery is defined as the ratio of the present charge to the nominal battery capacity  $C_N$ , given by

$$\text{SOC}_t = \text{SOC}_0 - \frac{1}{C_N} \int_{t_0}^t \eta I dt$$

Where  $\text{SOC}_t$  and  $\text{SOC}_0$  are the SOC of the battery at initial time  $t_0$  and current time  $t$ , respectively,  $\eta$  is the Coulomb efficiency, and  $I$  denote the current flowing through the voltage source.

With the transformation into a discrete-time system, the input matrix  $u_k$ , output matrix  $y_k$  and the state matrix  $x_k$  as shown, are defined as follows:

$$u_k = I_k$$

$$y_k = U_{t,k}$$

$$x_k = [SoC_k \quad U_{1,k} \quad U_{2,k}]^\top$$

The parameter matrices  $A_k$ ,  $B_k$ ,  $C_k$  and  $D_k$  are defined by

Where  $U_{oc,k}(SOC)$  represents the non-linearity between SOC and  $U_{oc}$ , which can be measured with experiments. The measurement and identification of the parameters in the parameter matrices will be introduced in the next sections.

$$A_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \exp\left(\frac{-\Delta t}{R_1 C_1}\right) & 0 \\ 0 & 0 & \exp\left(\frac{-\Delta t}{R_1 C_1}\right) \end{bmatrix}$$

$$B_k = \begin{bmatrix} \frac{\eta \Delta t}{C_N} \\ R_1 \left(1 - \exp\left(-\frac{-\Delta t}{R_1 C_1}\right)\right) \\ R_2 \left(1 - \exp\left(-\frac{-\Delta t}{R_2 C_2}\right)\right) \end{bmatrix}$$

$$C_k = \left[ \frac{dU_{oc,k}(SOC)}{dSOC} \Big|_{\hat{SOC}_k} - 1 \quad -1 \right]$$

$$D_k = [-R_0]$$

## Cloud Implementation of SoC and SoH Estimation

The **cloud-based implementation** of SoC and SoH estimation leverages the computational power and data storage capabilities of cloud platforms. The key components of the cloud implementation include:

**Data Acquisition:** The BMS-Slave collects voltage, current, and temperature data from the battery cells and transmits it to the cloud via IoT protocols (e.g., MQTT, TCP/IP).

**Data Preprocessing:** The raw data is pre-processed to generate time-frequency scalograms using **Continuous Wavelet Transform (CWT)**:

CWT can be represented using

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$$

where  $\psi_{a,b}(t)$  is called the child wavelet which is shifted in time by  $b$  and scaled in frequency by  $a$ . The wavelet function  $\psi(t)$  is called the mother wavelet. Two conditions that the wavelet function has to satisfy are  $\int_{-\infty}^{\infty} \psi(t) dt = 0$  and  $\int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1$

The child wavelets are convoluted over the measured signal to get the wavelet coefficients. This is given by

$$w(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi^*\left(\frac{t-b}{a}\right) dt$$

where  $w(a, b)$  is the wavelet coefficient, and  $\psi^*$  is the complex conjugate of the wavelet function.

Scalograms are the result of applying the CWT to the measured signals (current and voltage). Scalograms are two-dimensional plots of the CWT coefficients over time and frequency. The time-frequency images generated by CWT from the measured signals (both current and voltage) are inputs to the deep learning models for predicting the SoH of the battery.

**Model Training and Estimation:** The pre-processed data is used to train the AEHF and PSO-based models for SoC and SoH estimation. The AEHF is used for real-time SoC estimation, while the PSO algorithm is used for periodic SoH estimation:

- The SOHC of the battery is usually defined as follows:

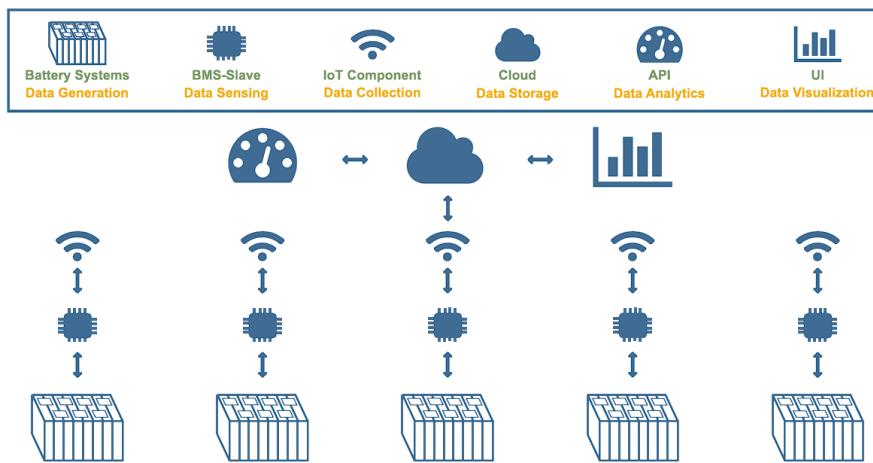
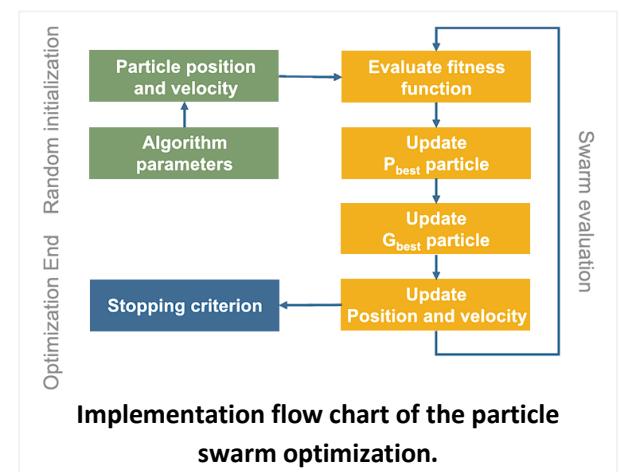
$$SOH_{C,t} = \frac{C_{N,t}}{C_{N,0}}$$

where  $C_{N,0}$  and  $C_{N,t}$  are the nominal capacity and the remaining capacity at current time  $t$ . The  $SOH_R$  of the battery is determined from the ohmic resistance increasing, namely

$$SOH_{R,t} = \frac{R_{0,t}}{R_{0,0}}$$

where  $C_{N,t}$  and  $R_{0,t}$  are the estimated capacity and ohmic resistance at time  $t$ .

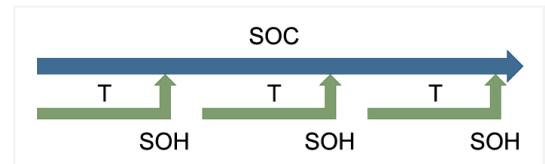
**Data Visualization:** The estimated SoC and SoH values are visualized in a user interface (UI) for real-time monitoring and diagnostics.



**Continuous Learning:** The cloud platform supports **continual learning** by periodically updating the model parameters using new data. The reservoir sampling technique is used to create a memory buffer for fine-tuning the model:

- Memory Buffer = Reservoir Sampling ( $D_N, M$ )

where  $D_N$  is the training data, and  $M$  is the size of the memory buffer.



The cloud implementation ensures accurate and real-time monitoring of battery states, enabling predictive maintenance and optimization of battery performance.

## Methodology

---

### Measurement Techniques

We have chosen **Potentiostatic EIS (PEIS)** and **Dynamic EIS (DEIS)** for our experiment. PEIS provides high-resolution impedance spectra across a wide frequency range, ensuring precise control of voltage perturbations and maintaining linearity in the battery's response. DEIS complements PEIS by capturing impedance behaviour during active charge/discharge cycles, enabling real-time monitoring of dynamic conditions.

### Interpretation Methods

For interpreting the impedance data, we will use **Equivalent Circuit Modelling (ECM)** and **Distribution of Relaxation Times (DRT)**. ECM provides a straightforward way to quantify key battery parameters, such as charge transfer resistance and diffusion coefficients, while DRT enhances the accuracy of ECMS by identifying overlapping processes in the impedance spectra.

### Validation Approaches

To ensure the reliability of our impedance data, we will use **Kramers-Kronig Relations** to validate the linearity, causality, and stationarity of the measurements. This mathematical validation is essential for ensuring the accuracy of our data before further analysis.

### Future Directions

To enhance the efficiency and accuracy of our data analysis, we will integrate **Machine Learning (ML)** algorithms into our workflow. ML will be used to automate the fitting of ECMS and DRTs, predict battery states (e.g., SOC, SOH), and identify patterns in the data that may indicate early signs of failure or degradation.

---

# Components

---

## Battery

### General Specifications

- Model Name: INR18650-30Q
- Battery Type: Lithium-ion (LiNiCoMnO<sub>2</sub>)
- Nominal Voltage: 3.6V
- Nominal Capacity: 2,950mAh (minimum)
- Standard Charge Method: CCCV (Constant Current-Constant Voltage)
- Charge Current: 1.50A (standard), 4A (rated)
- Charge Cut-off Voltage: 4.20 ± 0.05V
- Charge Cut-off Current: 150mA (standard), 100mA (rated)
- Standard Charge Time: 180 min (standard), 70 min (fast charge)
- End of Discharge Voltage: 2.5V

### Discharge & Cycle Life

- Max Continuous Discharge Current: 15A (at 25°C)
- Cycle Life: 60% capacity retention after 250 cycles at 15A discharge
- Discharge Cut-off Voltage: 2.5V

### Impedance & EIS-Related Data

- Initial Internal Impedance: ≤ 26mΩ (measured at AC 1kHz after standard charge)
- Temperature Dependence of Discharge Capacity:
  - -20°C: 60% capacity
  - -10°C: 75% capacity
  - 0°C: 80% capacity
  - 23°C: 100% capacity
  - 60°C: 95% capacity
- Charge Rate Capabilities:
  - 0°C: 80% of max capacity
  - 5°C: 90%
  - 23°C: 100%
  - 45°C & 50°C: 95%
- Discharge Rate Capabilities:
  - 0.6A (0.2C): 100%
  - 5A: 97%
  - 10A: 100%
  - 15A: 97%
  - 20A: 95%

### Physical & Environmental Parameters

- Cell Weight: Max 48.0g
- Cell Dimensions:
  - Height: Max 65.0mm
  - Diameter: Max 18.4mm
- Operating Temperature Range:
  - Charge: 0°C to 50°C (recommended max 45°C)
  - Discharge: -20°C to 75°C (recommended max 60°C)

- Storage Temperature Range:
  - 1.5 years: -30°C to 25°C
  - 3 months: -30°C to 45°C
  - 1 month: -30°C to 60°C

#### Safety & Handling

- Overcharge Protection: No fire/explosion under 20V, 20A charging
- Short-Circuit Protection: No fire/explosion under 80mΩ short-circuit at 20°C & 55°C
- Forced Discharge Protection: No fire/explosion with reverse charge at 3A for 90 min
- Thermal Stability: No fire/explosion up to 130°C
- Drop Test: No leakage/explosion per IEC62133
- Vibration Test: No leakage/explosion per UN38.3

Source: [https://e2e.ti.com/cfs-file/\\_key/communityserver-discussions-components-files/196/INR18650\\_2D00\\_30Q\\_5F00\\_datasheet.PDF](https://e2e.ti.com/cfs-file/_key/communityserver-discussions-components-files/196/INR18650_2D00_30Q_5F00_datasheet.PDF)

## Charging/Discharging Module

- **Input Voltage:** DC11-18V
- **Circuit Power:**
  - Max Charge: 60W
  - Max Discharge: 5W
- **Charge Current Range:** 0.1-6.0A
- **Discharge Current Range:** 0.1-2.0A
- **LIPo/LiFe/Liton Cells:** 1-6S
- **NIMH/NiCd Cells:** 1-15S
- **Pb battery Voltage:** 2-20V

Source: [https://www.skyrc.com/iMAX\\_B6mini\\_Charger](https://www.skyrc.com/iMAX_B6mini_Charger)

## Power Adapter

General Specifications:

- **Voltage:** 12V DC
- **Power:** 60-Watt
- **Type:** DC Power Supply AC Adapter SMPS (Switched-Mode Power Supply)
- **Function:** AC to DC Converter
- **Output:** 12 Volt 5 Amp (12V 5A)

## Power Supply for Microcontroller

General Specifications:

- **Input Voltage:** 6.5V to 12V DC (via DC jack) or 6.5V to 23V DC (via screw terminal)
- **Output Voltage:** 3.3V and 5V DC
- **Output Current:** Up to 700mA (combined output for both 3.3V and 5V)

## STM32G473xE Microcontroller

Essential Specifications:

### Core & Performance

- **Core:** ARM Cortex-M4 with **Floating Point Unit (FPU)**
- **Clock Speed:** Up to **170 MHz**
- **Math Accelerator:**
  - **CORDIC (Coordinate Rotation Digital Computer):** Accelerates **trigonometric functions**, useful for impedance calculations.
  - **FMAC (Filter Math Accelerator):** Optimized for signal filtering and processing.

### Analog & Signal Processing

- **Analog-to-Digital Converters (ADCs):**
  - **5 x 12-bit ADCs** with up to **42 channels**.
  - **Resolution up to 16-bit** with **hardware oversampling**.
  - **Sampling Time:** 0.25 µs.
  - **Input Range:** 0 to 3.6V.
- **Digital-to-Analog Converters (DACs):**
  - **7 x 12-bit DACs** (3 buffered external channels, 4 unbuffered internal channels).
  - **Sampling Rate:** 1 MSPS (buffered), 15 MSPS (unbuffered).
- **Operational Amplifiers (OPAMPs):**
  - **6x programmable OPAMPs**, configurable as PGA (programmable gain amplifier), essential for **signal conditioning** in EIS.
- **Comparators:**
  - **7 x ultra-fast rail-to-rail comparators**, useful for signal threshold detection.
- **Voltage Reference Buffer (VREFBUF):**
  - Supports **2.048V, 2.5V, 2.9V** reference voltages for stable ADC/DAC operations.

### Timers & Signal Generation

- **High-Resolution PWM Timers:**
  - **3 x 16-bit, 8-channel motor control timers**, up to **8 PWM outputs**.
  - Dead-time generation and emergency stop features.
  - Used for **excitation signal generation** in EIS.
- **Basic Timers:**
  - TIM6 and TIM7 available for **precision timing control** in impedance measurement.

### Communication & Data Handling

- **SPI Interfaces:**
  - **4 x SPI** interfaces (supports 4 to 16-bit data frames).
  - Required for high-speed **data transfer from ADCs/DACs**.
- **DMA Controller:**
  - **16-channel DMA** for high-speed **data transfer without CPU intervention**.
  - Useful for **continuous impedance measurement and logging**.

### Power & Operating Conditions

- **Supply Voltage: 1.71V to 3.6V**.
- **Low Power Modes:** Sleep, Stop, Standby, Shutdown.
- **VBAT Mode:** Supports RTC and backup registers for **data retention during power loss**.

Source: <https://www.st.com/resource/en/datasheet/stm32g473cb.pdf>

## TI C2000 Microcontroller

**Essential specifications of the TI C2000 (TMS320F28002x) microcontroller:**

### Core & Performance

- **Core:** C28x DSP core with **Floating Point Unit (FPU)**
- **Clock Speed:** 100 MHz
- **Processing Power:** 100 MIPS (Million Instructions Per Second)
- **Math Accelerators:**
  - **TMU (Trigonometric Math Unit):** Optimized for complex mathematical operations, useful for EIS data processing.
  - **Fast Integer Division Unit (Fast DIV):** Accelerates division operations, reducing computation time for impedance analysis.

### Analog & Signal Processing

- **Analog-to-Digital Converters (ADCs):**
  - 2 x 12-bit ADCs with up to **3.45 MSPS (Mega Samples Per Second)**.
  - **8 channels per ADC**, allowing parallel sampling for fast EIS measurements.
  - **Post-processing hardware** for real-time data filtering and thresholding.
- **Comparators:**
  - 4 x **Windowed Comparators (CMPSS)** with integrated 12-bit DACs for precise voltage threshold detection.
- **Enhanced Pulse Width Modulation (PWM):**
  - **7 PWM modules, 14 outputs (8 high-resolution)**.
  - **150 ps resolution** for fine control of excitation signals in EIS.

### Timers & Signal Generation

- **High-Resolution ePWM (Enhanced PWM):**
  - Supports advanced **switching techniques** for frequency sweeps in impedance measurements.
- **Capture Modules (eCAP, eQEP, HRCAP):**
  - Useful for **timing analysis and synchronization** of signals in EIS experiments.
- **High-Speed Clocking:**
  - **2 x 10 MHz oscillators** for accurate timekeeping in signal generation.

### Communication & Data Handling

- **SPI Interfaces:**
  - 2 x SPI (supports up to 50 MHz for high-speed data transfer from ADCs).
- **Direct Memory Access (DMA):**
  - 6-channel DMA enables real-time **data acquisition and transfer without CPU intervention**.

### Power & Operating Conditions

- **Supply Voltage: 1.2V (core), 3.3V (I/O)**
- **Low Power Modes:** Sleep, Standby, and Deep Sleep to optimize power consumption.
- **Temperature Range: -40°C to 125°C**, making it suitable for **temperature-sensitive impedance measurements**.

Sources: <https://www.ti.com/lit/gpn/TPS75005>, <https://www.ti.com/lit/ml/swap126/swap126.pdf>

**Miscellaneous Equipment:** Bread board, Jumper wires

## Implementation of EIS on STM32

---

We had to apply Electrochemical Impedance Spectroscopy (EIS) in our project on the STM32G474RE microcontroller. EIS is a powerful technique used to analyse the internal characteristics of electrochemical systems such as lithium-ion batteries by applying a small AC signal and measuring the impedance response over a range of frequencies. For this, we first focused on learning the STM32 development environment and hardware.

The first exercise that we carried out was LED blinking. This informed us about the GPIO pin layout as well as the coding and uploading process in STM32CubeIDE, hence enabling us to test the operational condition of the microcontroller. We made use of the embedded HAL libraries to toggle the status of a GPIO pin connected to the on-board LED with an intervening short delay between the ON and OFF stages in the while(1) loop. This exercise acquainted us with our first working interface with the microcontroller.

Having achieved blinking of the LED, we proceeded to implement more complex peripherals needed in our EIS configuration — like DAC to provide the excitation signal, ADC to capture the response signal, and UART for serial communication with the PC.

To systematically approach the implementation, I divided the project into two major phases:

- i) **Sine wave generation** — creating a sinusoidal signal of controllable frequency and amplitude using the onboard DAC.
- ii) **EIS measurement and analysis** — applying the signal across a Li-ion battery, capturing the resulting response via ADC, and analysing impedance across different frequencies.

During the course of the project, we focused primarily on the first phase, i.e., sine wave generation. Within this, we were able to successfully implement the following two tasks independently:

1. **Frequency Control:** We generated a sine wave of desired frequency using a lookup table with DMA and a timer-triggered DAC output. This allowed smooth waveform generation at various fixed frequencies.
2. **Amplitude Control:** We generated a sine wave of a specific voltage amplitude by scaling the DAC values appropriately, though frequency control could not get incorporated simultaneously in this version.

However, we were not able to integrate both frequency and amplitude control into a single implementation. This integration would have allowed for full parameter tuning of the excitation signal, which is essential for complete EIS functionality.

The **second phase** of the project — involving real-time impedance measurement, frequency sweep, current/voltage data acquisition, and Nyquist plotting — could not be completed due to time constraints and ongoing debugging of the waveform generation.

## Methodology

To start the experimental implementation of the Electrochemical Impedance Spectroscopy (EIS) setup, we first used the SkyRC iMAX B6 Mini Charging and Discharging Module to charge and discharge the lithium-ion battery (INR18650-30Q) in a controlled setting. This process allowed the battery to be prepared for testing at different states of charge.

Subsequent to the above, we created the signal generation circuit and data acquisition circuit that employs the STM32G474RE microcontroller. The setup was supported by various devices like jumper wires, resistor, LED—one utilized in verification of GPIO earlier—and USB data cable.

The main objective of the hardware setup was to create a sinusoidal signal through the DAC peripheral of the STM32G474RE. It would be utilized as the excitation voltage for EIS measurement. We modified some variants of firmware for sine generation in STM32CubeIDE — some emphasized the control of frequency through timers and DMA, whereas others emphasized the control of amplitude through scaling of the DAC buffer values.

All these project files were stored individually for documentation and analysis (and we have shared on Google Drive: [\[Link\]](#)), and corresponding output data was received via UART and plotted on the serial monitor. UART output was also logged via a Python program and stored in Excel file format for further processing.

## Procedure

Steps to make a project in STM32:

1. Go to File > New > STM32 Project, search for STM32G474RE (or select NUCLEO-G474RE), click Next, enter the project name, and click Finish.
2. Open the .ioc configuration view, click on the required pins or use the left-hand menu to enable peripherals like GPIO, ADC, DAC, UART, etc., and adjust the clock settings if needed.
3. Go to Project > Generate Code, ensure the toolchain is set to STM32CubeIDE, and click Generate to create the initialization code.
4. Open Src > main.c, scroll to the `while(1)` loop, and write the application logic using HAL functions to interact with the peripherals.
5. Click on the hammer icon in the toolbar to build the project, then check the console to ensure the build completes without any errors.
6. Connect the STM32G474RE board to the PC using a USB cable; the board should be automatically detected via ST-LINK in STM32CubeIDE.
7. Click the green Run button, select STM32 Cortex-M C/C++ Application in the popup, confirm the debug configuration, and click Run to flash the code onto the board.
8. Configure USART in the .ioc file, add HAL UART transmit code in `main.c`, open serial terminal PuTTY, select the correct COM port, and view the UART output.

## 9. Data Receiving and Plotting:

- a. Run a Python script in the terminal that uses the serial, time, and openpyxl libraries to connect to the STM32 via the correct COM port (e.g., COM7), read real-time UART data, and write it directly into an Excel file (.xlsx) with structured headers like time, applied voltage, and received voltage.
- b. Use another Python script to simulate voltage changes in real time by reading the Excel file (or a live data stream) and dynamically plotting the values using libraries like matplotlib, updating the graph as new data arrives to visualize system behaviour. (But we rejected this idea because there was very much time lag between data received and getting it plotted.)

Sources: [YouTube Videos Link](#), [Website](#).

Files: All the files can be accessed from [CP301](#).

## Observations and Results

---

The Observations and results can be accessed from the folder Generated Plots in the above link naming CP301.

In the experimental phase of our project, we focused on implementing the first part of the Electrochemical Impedance Spectroscopy (EIS) setup — generating a sine wave using the STM32G474RE microcontroller. This sine wave serves as the excitation signal to be applied across the battery and a known series resistor. We implemented two parts independently:

### 1. Frequency-Controlled Sine Wave Generation:

Using DMA, TIM6, and DAC, we successfully generated sine waves of varying frequencies by adjusting the sampling rate and buffer update timings. This helped us simulate the application of AC signals at different excitation frequencies, which is crucial for performing EIS over a frequency spectrum.

### 2. Amplitude-Controlled Sine Wave Generation:

In a separate implementation, we scaled the values in the DAC buffer to generate sine waves with controlled voltage amplitudes, including low-level signals such as 10 mV peak-to-peak — typical for non-invasive EIS measurements.

Although both functionalities worked individually, we were unable to integrate frequency and amplitude control into a unified system, due to complexities in buffer synchronization and real-time parameter tuning. This remains a key challenge to be addressed in future development.

To validate our sine wave generation setup, we applied a 10mV peak-to-peak sine wave across a series combination of a **3.3V lithium-ion battery** and a **7.5 Ω resistor**. The output signals — the total applied voltage and the voltage across the resistor — were logged via UART and recorded using a Python script into an Excel file. We then used MATLAB and Python to analyse the results.

The following key outcomes were obtained from the reading "serial\_data\_20250424\_012840":

- **Estimated Frequency of the Signal:**  $\approx 20.2$  Hz
- **Amplitude of Applied Voltage:**  $\approx 4.80$  mV
- **Amplitude of Voltage Across  $7.5 \Omega$  Resistor:**  $\approx 2.462$  mV
- **Calculated Battery Impedance:**  $\approx 7.122 \Omega$
- **Phase Shift (Current Lags Voltage):**  $\approx 101.82^\circ$

These results validate that our STM32-based system can generate a valid low-amplitude excitation signal and capture the voltage response needed to compute the impedance of the battery at a given frequency. The observed **phase lag** and impedance magnitude are consistent with the behaviour of an electrochemical system under AC excitation, indicating successful proof-of-concept for the first stage of EIS implementation.

## Major Problems Faced

---

1. During high-frequency signal generation using the STM32, we encountered physical limitations due to the program execution time. The loop responsible for generating and transmitting the waveform took approximately 4 milliseconds per iteration. This delay was caused by the time taken for the control logic execution as well as the UART transmission, which alone consumed around 2.6 milliseconds. As a result, the maximum achievable waveform frequency was severely limited, making it difficult to operate at higher frequencies.
2. While we managed to control either the voltage or the frequency individually, synchronised control of both parameters was not achievable due to lack of relevant technical resources. We were unable to find sufficient documentation or references that demonstrated how to vary both voltage and frequency in a controlled and coordinated manner within the same STM32 program. This lack of information significantly hindered our ability to implement a fully dynamic waveform generation system.

# Progress we could make

---

## 1. Literature Review and Methodology Development

The literature review and methodology development phase are complete, providing a solid foundation for the experimental setup and data analysis.

---

## 2. IoT Setup Design and Component Acquisition

The IoT setup design is finalized, and component acquisition is in progress. The assembly and calibration of the equipment will commence once all components are received.

---

## 3. Algorithm Development for SoC and SoH Estimation

The algorithms are theoretically developed and ready for implementation. Testing and validation will begin once the IoT setup is operational.

---

## 4. Firmware Development and Signal Generation

Separate implementations for frequency-controlled and amplitude-controlled sine wave generation have been developed and validated on STM32. These were used to create fixed-frequency excitation signals suitable for impedance measurements. UART-based data logging is operational and provides reliable voltage readings for further analysis.

---

## 5. Experimental Execution and Data Logging

Initial EIS experiments using a 10mV sine wave across a 3.3 V battery and a  $7.5\ \Omega$  resistor were conducted. Voltage response was logged via UART and stored in Excel format. The data was analysed to estimate impedance and phase response at  $\sim 20$  Hz, validating the effectiveness of the hardware setup.

---

### Current Progress vs. Total Objectives

- **Progress So Far:**
    - Completed the theoretical and design phases, including literature review, methodology development, IoT setup design, and algorithm development.
    - Sourced and identified all necessary components for the experimental setup.
    - Performed initial EIS measurements and validated impedance estimation.
  - **Remaining Tasks:**
    - Conduct EIS measurements to validate data accuracy.
-

## Conclusion

---

This project represents a substantial advancement toward the development of a comprehensive IoT-enabled digital twin framework for lithium-ion battery systems, designed to facilitate real-time condition monitoring and predictive diagnostics. The foundation was established through an in-depth literature review of Electrochemical Impedance Spectroscopy (EIS), State of Charge (SoC) and State of Health (SoH) estimation techniques, and digital twin implementation strategies. From this foundation, a comprehensive methodology was devised—centred on Potentiostatic and Dynamic EIS measurements, interpreted using Equivalent Circuit Modelling (ECM) and Distribution of Relaxation Times (DRT), and validated via Kramers-Kronig relations.

The hardware implementation involved designing a signal generation and acquisition system using the STM32G474RE microcontroller. We successfully generated sine waves with controllable frequency and amplitude using DAC, DMA, and timer configurations. Independent implementations for both parameters demonstrated the feasibility of waveform control for EIS excitation signals. UART-based communication and Python-based data logging provided a reliable pipeline for capturing experimental readings.

Initial experiments were conducted by applying a 10mV sine wave across a 3.3 V lithium-ion battery and a known series resistor. The measured data was logged and analysed, enabling the calculation of complex impedance and phase shift at  $\sim$ 20 Hz. These results validated the STM32-based setup as a viable platform for low-amplitude, fixed-frequency impedance measurements.

Despite the progress made, the project faced many challenges primarily due to limitations in our implementation. Inefficiencies in the code and data handling introduced delays that restricted high-frequency waveform generation. Furthermore, we were unable to achieve synchronised control of both amplitude and frequency, largely due to constraints in our approach and the absence of prior experience with such integration. These issues, while constraining, have provided valuable insight into the system's complexity and identified clear directions for future refinement.

This project demonstrates the feasibility of integrating EIS with IoT and machine learning for advanced battery diagnostics and lays the groundwork for scalable, cloud-integrated solutions in smart energy management and electric mobility.

## Future Work

---

The next phase will focus on integrating amplitude and frequency control into a unified signal generation system, a functionality not fully achieved in the current stage. This integration is essential for enabling dynamic excitation in Electrochemical Impedance Spectroscopy (EIS). Efforts will also be directed toward optimising firmware performance to reduce latency and support higher frequency operations, addressing the implementation inefficiencies identified during testing.

Once the remaining components are assembled and calibrated, full-spectrum EIS measurements will be conducted across various battery states. These measurements will feed into the cloud-based digital twin platform, where the previously developed AEHF and Kalman Filter algorithms will be deployed for real-time State of Charge (SoC) and State of Health (SoH) estimation.

In parallel, the system will be extended to incorporate continual learning mechanisms on the cloud, allowing for incremental model updates based on live data. The setup will be validated under simulated electric vehicle battery conditions to assess performance under dynamic loads. Additionally, focus will be placed on streamlining data acquisition, communication, and processing to ensure accurate, low-latency operation suitable for real-world deployment.

This future work aims to elevate the current prototype into a reliable, scalable, and intelligent solution for advanced battery monitoring and management.

For Future Work we have uploaded everything on Google Drive which can be accessed following the link: [\[CP301\]](#).

## References

---

1. <https://www.cell.com/action/showPdf?pii=S2542-4351%2823%2900195-2>
2. <https://www.sciencedirect.com/science/article/abs/pii/S0360544223020935>
3. <https://www.sciencedirect.com/science/article/pii/S2352152X20308495>
4. <https://www.sciencedirect.com/science/article/abs/pii/S0957417423009466>
5. [Application of electrochemical impedance spectroscopy to commercial Li-ion cells: A review](#)