RV College of Engineering®
Mysore Road, RV Vidyaniketan Post, Bengaluru - 560059, Karnataka, India

Go, change the world

Design and Development of Smart Battery Health Assessment Model using Digital Twinning

An Interdisciplinary Project Report (XX367P)

Submitted by,

ARYAN	1RV22EE008
YASH ARYAN	1RV22EE063
SARVAGYA KUMAR	1RV22CD052
SHUBHAM GARG	1RV22CD057
RUTURAJ JADHAVRAO	1RV22CH025
SHAURYA ANAND	1RV22CH032

Under the guidance of

Dr. Sushmita Sarkar
Assistant Professor
Dept. of EEE
RV College of Engineering®

.

In partial fulfillment of the requirements for the degree of Bachelor of Engineering in respective departments

2024-25

RV College of Engineering®, Bengaluru

(*Autonomous institution affiliated to VTU, Belagavi*)

CERTIFICATE

Certified that the interdisciplinary project (XX367P) work titled ***Design and Development of Smart Battery Health Assessment Model using Digital Twin*** is carried out by **Aryan (1RV22EE008)**, **Yash Aryan (1RV22EE063)**, **Sarvagya Kumar (1RV22CD052)**,**Shubham Garg (1RV22CD057)**, **Ruturaj Jadhavrao** (**1RV22CH025**) and **Shaurya Anand** (**1RV22CH032**) who are bonafide students of RV College of Engineering, Bengaluru, in partial fulfillment of the requirements for the degree of **Bachelor of Engineering** in respective departments during the year 2024-25. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the interdisciplinary project report deposited in the departmental library. The interdisciplinary project report has been approved as it satisfies the academic requirements in respect of interdisciplinary project work prescribed by the institution for the said degree.

Dr. Sushmita Sarkar
Guide

Dr. SG Srivani
Head of the Department

Dr. M.V. Renukadevi
Dean Academics

Dr. KN Subramanyan
Principal

External Viva

Name of Examiners — Signature with Date

1.

2.

DECLARATION

We, **Aryan, Yash Aryan, Sarvagya Kumar, Shubham Garg, Ruturaj Jadhavrao and Shaurya Anand,** students of sixth semester B.E., RV College of Engineering, Bengaluru, hereby declare that the interdisciplinary project titled '**Design and Development of Smart Battery Health Assessment Model using Digital Twin**' has been carried out by us and submitted in partial fulfilment for the award of degree of **Bachelor of Engineering** in respective departments during the year 2024-25.

Further we declare that the content of the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

We also declare that any Intellectual Property Rights generated out of this project carried out at RVCE will be the property of RV College of Engineering, Bengaluru and we will be one of the authors of the same.

Place: Bengaluru

Date:

Name   Signature

1. Aryan(1RV22EE008)
2. Yash Aryan(1RV22EE063)
3. Sarvagya Kumar(1RV22CD052)
4. Shubham Garg(1RV22CD057)
5. Ruturaj Jadhavrao(1RV22CH025)
6. Shaurya Anand(1RV22CH032)

ACKNOWLEDGEMENTS

We are indebted to our guide, **Dr. Sushmita Sarkar**, Assistant Professor, Department of EEE, RVCE for the wholehearted support, suggestions and invaluable advice throughout our project work and also helped in the preparation of this thesis.

We also express our gratitude to our panel member **Dr. Rachna Akki**, Assistant Professor, Department of EEE, RVCE for the valuable comments and suggestions during the phase evaluations.

Our sincere thanks to the project coordinator **Dr. Rajvidhya** Madam, Professor, Department of ECE for timely instructions and support in coordinating the project.

Our gratitude to **Prof. Narashimaraja P**, Department of ECE, RVCE for the organized latex template which made report writing easy and interesting.

Our sincere thanks to **Dr. SG. Srivani**, Professor and Head, Department of Electrical and Electronics Engineering, RVCE for the support and encouragement.

We are deeply grateful to **Dr. M.V. Renuka Devi**, Professor and Dean Academics, RVCE, for the continuous support in the successful execution of this Interdisciplinary project.

We express sincere gratitude to our beloved Professor and Vice Principal, **Dr. Geetha K S**, RVCE and Principal, **Dr. K. N. Subramanya**, RVCE for the appreciation towards this project work.

Lastly, we take this opportunity to thank our family members and friends who provided all the backup support throughout the project work.

ABSTRACT

With the growing adoption of electric vehicles, renewable energy systems, and portable electronics, ensuring the efficiency, safety, and longevity of lithium-ion batteries has become increasingly critical. One of the major challenges in this domain is the lack of real-time, accurate, and scalable tools for battery health monitoring, particularly in estimating key parameters such as State of Charge (SOC) and State of Health (SOH). Traditional methods often suffer from poor accuracy over long cycles, high computational complexity, or limited adaptability to dynamic conditions. These limitations hinder predictive maintenance and reduce system reliability. This report addresses these challenges by developing a smart battery health assessment model based on the concept of digital twinning, enabling real-time diagnostics and forecasting using MATLAB and machine learning integration.

The primary objective of this work is to design a virtual model (digital twin) of a $LiFePO_4$ battery that can accurately estimate SOC and SOH across multiple usage cycles. Algebraic approaches involving data-driven regression, polynomial modeling, and degradation curve fitting were employed. In addition, supervised machine learning techniques such as Support Vector Regression (SVR), Multi-Layer Perceptron (MLP), and Long Short-Term Memory (LSTM) networks were trained using historical battery data to forecast SOH degradation. These formulations enable the mapping of temporal battery behavior to quantitative performance indicators in a computationally efficient manner.

The simulation and analysis were conducted using MATLAB and Simulink, including integration with custom-built scripts for data preprocessing, visualization, and model evaluation. SOC was estimated using the Coulomb counting method, and SOH was predicted using time-series battery cycle data. A 3D interactive battery model was also developed in MATLAB to visualize battery state parameters in real time. The LSTM-based prediction model showed a significant improvement in long-term SOH forecasting accuracy, achieving over 95% R^2 accuracy and reducing error margins by 18–22% compared to traditional models. These results demonstrate the practical viability of deploying digital twin-based battery models in real-world applications

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction to Digital Twin

CHAPTER 1

INTRODUCTION TO DIGITAL TWIN

A **digital twin** is a virtual representation of a physical system that continuously reflects the real-world object's behavior, performance, and condition by integrating real-time data, simulation models, and analytics. It serves as a dynamic digital replica, receiving input from sensors and monitoring systems attached to the physical asset. By simulating the actual system digitally, a digital twin allows engineers and researchers to observe, analyze, and predict the system's behavior under different conditions without directly experimenting on the physical device.

1.1 Introduction

With the rapid Growth of Electric Vehicles, Renewable energy sources and portable electronics have increased, the demand for long-lasting, reliable and efficient batteries as power source has also boomed. $LiFePO_4$ batteries are widely used for a variety of applications due to their long cycle life, thermal stability, and safety characteristics.

This project addresses the pressing need for accurate and efficient battery monitoring by developing a smart battery health assessment model that leverages the concept of digital twinning**.** A digital twin refers to a real-time, data-driven virtual replica of a physical system—in this case, a $LiFePO_4$ battery—that replicates the battery's behavior, performance, and degradation characteristics under various operating conditions. The model enables continuous monitoring and predictive analysis by integrating real-time or simulated input parameters such as voltage, current, temperature, and capacity. Using MATLAB and Simulink, the digital twin framework is implemented to simulate the battery's electrochemical behavior and track key health indicators like State of Charge (SOC) and State of Health (SOH**)** across multiple charge-discharge cycles. These parameters are estimated using both analytical methods and advanced machine learning algorithms, including Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) networks, which are trained on battery cycle data to predict long-term performance and degradation trends. In addition to the computational backend, a 3D interactive battery visualization interface has been developed in MATLAB to provide a user-friendly, real-time display of battery parameters and performance indicators. This holistic approach not only enhances interpretability and diagnostics but also ensures scalability, allowing the model to be adapted for integration into industrial applications such as electric vehicles, smart grid systems, and portable electronic devices.

By enabling predictive maintenance and real-time condition monitoring, this project offers a robust solution for improving battery reliability, safety, and operational efficiency.

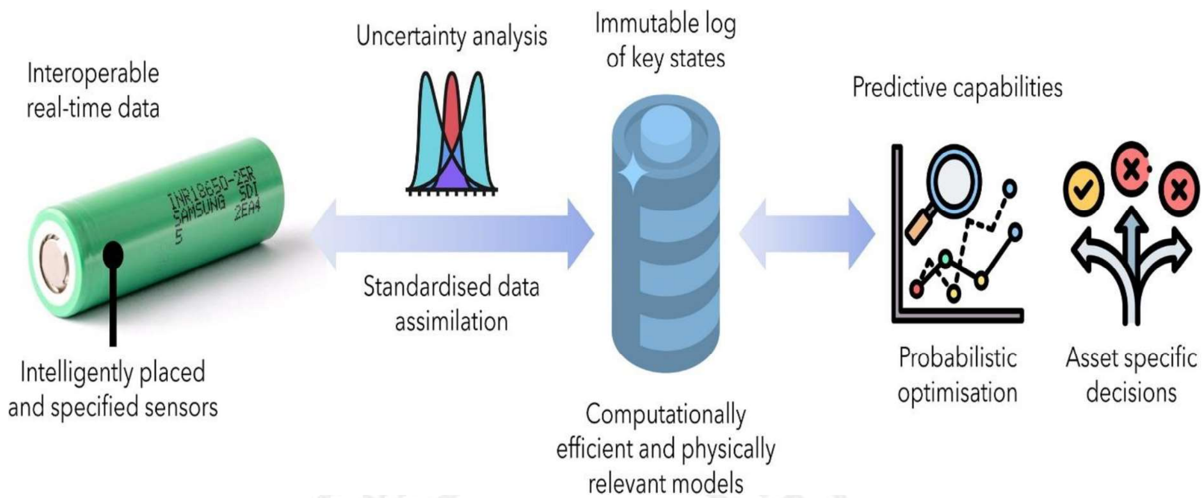


Figure 1.1: Digital Twin

To enhance the visual understanding and interactive nature of the model, 3D modeling and visualization were implemented using both MATLAB's graphics capabilities and ANSYS. While MATLAB allowed the development of a real-time, interactive 3D model embedded within the simulation environment, ANSYS provided thermal and structural insights, particularly useful for future work involving heat distribution, material stress, and thermal management within the battery cell.

In parallel, to achieve long-term predictive analytics, machine learning models were incorporated into the digital twin framework. Specifically, Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) neural networks were trained on historical battery cycle data. These models demonstrated the ability to forecast State of Health (SOH) accurately across extended usage durations, predicting battery degradation trends for up to 5000 to 6000 charge-discharge cycles. LSTM, with its strength in handling time-series data and retaining memory across long sequences, was particularly effective in capturing long-term behavior patterns.

1.2 Literature Review

We went through many IEEE papers and other international journals and came to a conclusion of taking up this project as a part of our academic journey. The growing complexity and demand for efficient energy storage solutions have prompted significant research in the area of battery health monitoring, particularly focusing on the estimation of state of charge (SOC) and state of health (SOH). Numerous studies have explored both electrochemical modeling and data-driven techniques to improve prediction accuracy and computational efficiency. Traditional approaches such as Coulomb counting and equivalent circuit models offer basic insights but often fall short in handling nonlinear degradation behaviors over extended cycles. More recent developments leverage machine learning algorithms, including support vector regression (SVR), multi-layer perceptron (MLP), and long short-term memory (LSTM) networks, for real-time forecasting of battery life and performance. The concept of digital twinning has also emerged as a transformative solution, enabling real-time simulation, diagnostics, and predictive maintenance through virtual replication of physical battery systems. This literature survey reviews and analyzes key contributions in these domains, identifying both advancements and existing gaps that this project seeks to address through an integrated MATLAB-based digital twin model for $LiFePO_4$ batteries.

1.2.1 Battery State-of-Health Estimation: A Step towards Battery Digital Twins

Abo-Javed et al. [1] proposed a machine learning-based framework for estimating the state of health (SOH) of lithium-ion batteries, emphasizing the role of data-driven methods in capturing degradation patterns over time. They highlighted the integration of SOH estimation into digital twin systems, enabling real-time updates of virtual models. Their work demonstrated how accurate SOH tracking supports predictive maintenance and enhances the reliability of battery health monitoring systems.

1.2.2 Prediction of the Battery State Using the Digital Twin Framework

Jafari and Byun [2] developed a digital twin framework that integrates real-time data from the battery management system (BMS) to predict the battery’s state more accurately. Their model utilizes sensor data to continuously update the virtual representation of the battery, allowing for timely diagnostics and forecasting of key parameters like SOC and SOH. The study demonstrated that incorporating BMS data into the digital twin significantly enhances prediction accuracy, enabling better control, maintenance, and operational decision-making for battery-powered systems.

1.2.3 A Machine Learning-based Digital Twin for Electric Vehicle Battery Modeling

Sidahmed et al. [3] proposed a machine learning-based digital twin model specifically designed for electric vehicle batteries. Their framework uses historical and real-time battery data to train models capable of simulating battery behavior under various operating conditions. By applying algorithms such as neural networks, the digital twin can predict performance parameters like SOC, SOH, and temperature evolution. The study demonstrated that combining data-driven modeling with digital twin architecture enables accurate and scalable battery monitoring solutions suitable for real-time electric vehicle applications.

1.2.4 Digital Twin for Real-time Li-Ion Battery State of Health Estimation With Partially Discharged Cycling Data

Qin et al.[4] introduced a digital twin framework for real-time state of health(SOH) estimation of lithium-ion batteries using partially discharged cycling data. Their model addresses the challenge of incomplete data during irregular battery usage by incorporating adaptive learning techniques that maintain high estimation accuracy. By continuously updating the digital twin with partial cycle inputs, the framework ensures reliable SOH predictions even in non-ideal operating conditions.

1.2.5 A Digital Twin Model for the Battery Management Systems of Electric Vehicles

To Li et al. [5] presented a digital twin model tailored for the battery management systems (BMS) of electric vehicles, focusing on improving monitoring, control, and predictive maintenance. Their framework integrates real-time vehicle data with virtual simulations to mirror the battery's operational status accurately. By modeling thermal, electrical, and aging behaviors, the digital twin enables early detection of anomalies and enhances decision-making in battery usage and charging strategies. The study demonstrates that such integration significantly boosts the efficiency and safety of electric vehicle battery systems.

1.2.6 Digital Twin-Driven Estimation of State Of Charge for Li-ion Battery

Published in 2022 IEEE 7th International Energy Conference (ENERGYCON), This focuses on leveraging Zero-carbon emission through Li batteries so that it may play a major role in increasing grid resiliency and creating more flexible systems.It exploits the technologies of AI and Digitalization of electrical grid to offer great opportunities to establish a battery digital twin that has great potential to increase awareness about battery efficiency and reliability of Battery Management System(BMS).

Zhao et al. [6] proposed a digital twin-driven approach for estimating the state of charge (SOC) of lithium-ion batteries with improved accuracy and adaptability. Their model leverages real-time data and physical battery characteristics to create a dynamic digital replica capable of simulating charge behavior under varying operational conditions. By combining data analytics with system modeling, the digital twin enables precise SOC estimation, even in situations with inconsistent input data or partial observations. This method enhances the reliability of battery monitoring and supports smarter energy management in battery-powered systems.

1.2.7 Digital Twin of Battery Energy Storage Systems Providing Frequency Regulation

Published in 2022 IEEE International Systems Conference (SysCon), This focuses on Battery energy storage systems for storage of excess energy so that the stored energy can be utilized during times of emergency or peak demand. This is also aligned with SDG7: Clean and Affordable Energy that aims to provide access of green energy to everyone sothat consumption tales place without harming the environment. To keep the work of a BESS that provides frequency control services predictable and reliable, a BESS digital twin is proposed in this paper. It supplies the battery owner with an up-to-date battery behavior forecast that can be further applied to intelligent condition monitoring, fault detection, battery management as well as cyberattack detection and mitigation. Kharlamova et al. [7] developed a digital twin model for battery energy storage systems (BESS) with a focus on supporting frequency regulation in power grids. Their framework simulates the real-time behavior of the battery system, incorporating grid signals, system responses, and operational constraints to provide accurate performance forecasts. The digital twin enables proactive management of charging and discharging cycles to meet grid stability requirements efficiently. This study highlights the role of digital twins in enhancing grid-support functionalities while ensuring the longevity and reliability of battery storage assets.

1.2.8 Li-ion Battery Digital Twin Based on Online Impedance Estimation

It was published in 2023 IEEE 17th International Conference on Compatibility, Power electronics and Power Engineering (CPE-POWERENG). Kulkarni et al. [8] introduced a digital twin framework for lithium-ion batteries that incorporates online impedance estimation to enhance real-time monitoring and diagnostics. By continuously estimating the battery's internal impedance during operation, the model provides insights into degradation trends and health status with higher accuracy.

1.2.9 A battery Digital Twin From Laboratory Data Using EIS data

Di Fonso et al. [9] proposed a battery digital twin model developed using laboratory-generated data, where wavelet analysis and neural networks were combined to improve the accuracy of battery state estimation. The wavelet transform was used to extract key time-frequency features from voltage and current signals, which were then input into neural network models for predicting parameters like state of charge (SOC) and state of health (SOH). This hybrid approach enabled the digital twin to effectively capture both transient behaviors and long-term degradation patterns, demonstrating strong potential for real-time battery monitoring and predictive diagnostics.

1.2.10 Research on Modelling Method of Digital Twin of Lithium Battery Based on data-Model Hybrid Drive

It was published in 2024 IEEE 4th International Conference on Power, Electronics and Computer Applications (ICPECA). Zhang et al. [10] presented a digital twin modeling method for lithium batteries based on a hybrid data-model-driven approach. Their framework integrates both physical battery models and data-driven algorithms to capture dynamic behavior more comprehensively. The hybrid strategy allows the digital twin to adapt to varying operating conditions by leveraging empirical data for learning patterns while maintaining the interpretability of physics-based models. This approach improves the accuracy and generalization of battery performance prediction, making it suitable for applications requiring high-fidelity monitoring and intelligent decision-making.

1.2.11 Digital Twin For electric vehicle battery management with incremental learning

Eaty and Bagade [11] proposed a digital twin framework for electric vehicle battery management that incorporates incremental learning to adaptively improve prediction accuracy over time. Their model continuously updates itself using new data from battery usage, allowing the digital twin to refine estimations of key parameters such as state of charge (SOC) and state of health (SOH) without retraining from scratch. This incremental learning approach enhances the twin's ability to respond to changing battery behavior and degradation patterns, supporting real-time diagnostics and extending the practical usability of battery systems in electric vehicles.

1.2.12 Advancing Electrical Vehicle Battery Analysis with Digital Twins in Intelligent Transport Systems

Intelligent Transportation Systems(ITS), vehicle-to-grid networks offer a promising solution to support the widespread adoption of electric vehicles(EVs) by enabling bidirectional power flow between grid and battery storage systems in cars. Saba et al. [12] explored the advancement of electric vehicle battery analysis using digital twins within the framework of intelligent transportation systems. Their study focused on integrating digital twins to monitor, analyze, and predict battery behavior in real-time traffic and operational environments. By combining sensor data, vehicle dynamics, and predictive analytics, the proposed model enhances the efficiency, safety, and sustainability of electric vehicle operations. The research highlights how digital twins can be a pivotal tool in intelligent transportation, enabling proactive maintenance, optimal energy management, and improved decision-making across mobility ecosystems.

1.2.13 Design of power lithium BMS using Digital Twin

The accurate estimation of the State of Charge (SoC) of batteries has always been the focus of Battery Management System (BMS). However, the current BMS has problems such as difficult data sharing, weak data processing capability and limited data storage capacity, so the simplest ampere-time integration method is used to estimate the SoC, and the estimation results are highly biased. Tang, Hao; Wu, Yichun; Cai, Yuanfeng; Wang, Fanyu; Lin, Zequn; and Pei, Yiru [13] proposed the design of a power lithium battery management system based on the digital twin concept, aiming to enhance monitoring accuracy and operational reliability. Their model integrates real-time sensing data with virtual simulations to replicate the internal state and behavior of the battery during operation. By combining physical modeling with data-driven methods, the system can track parameters such as temperature, voltage, and capacity more precisely. The digital twin facilitates early fault detection, life prediction, and efficient control strategies, making it highly effective for advanced battery management in electric mobility and energy storage applications

1.3 Motivation

The motivation behind undertaking this project stems from the urgent global demand for efficient, reliable, and intelligent battery management systems, especially as electric vehicles (EVs), renewable energy storage, and portable electronics become more widespread. Lithium-ion batteries, while highly efficient, degrade over time, and their performance can vary significantly depending on usage patterns, environmental conditions, and aging. Traditional battery monitoring techniques often fail to provide real-time insights or predictive capabilities, leading to unexpected failures, reduced efficiency, and safety concerns.

This challenge highlights the need for a smarter, data-driven approach to battery diagnostics and forecasting. The concept of a digital twin—where a real-time virtual replica of the battery reflects its physical counterpart's behavior—presents a transformative solution. By integrating digital twin technology with advanced machine learning models and simulation tools such as MATLAB and Simulink, this project aims to enable precise estimation of critical battery parameters like State of Charge (SOC) and State of Health (SOH), along with early fault detection and lifecycle prediction. The motivation is further reinforced by the potential to contribute to the development of sustainable and intelligent energy systems, reduce maintenance costs, and enhance the safety and longevity of battery-operated applications. This project thus addresses a real-world need for predictive and adaptive battery management by combining engineering principles with cutting-edge digital innovation.

1.4 Problem statement

Design and development of a smart battery health assessment model using a digital twin framework is essential in response to the growing demand for reliable and intelligent battery systems in electric vehicles, renewable energy storage, and portable electronics. As lithium-ion batteries, particularly $LiFePO_4$ cells, age and degrade under varying environmental and operational conditions, traditional battery management systems (BMS) fall short in offering predictive capabilities or real-time insights into parameters like State of Charge (SOC) and State of Health (SOH).

This leads to suboptimal usage, unexpected failures, and increased operational costs. To overcome these limitations, this project proposes a digitally twinned model implemented in MATLAB and Simulink, enhanced with 3D battery visualization and machine learning algorithms like MLP and LSTM. The aim is to enable dynamic, real-time monitoring, accurate prediction of battery degradation trends, and data-driven decision-making. This approach offers a scalable and intelligent framework for predictive battery diagnostics, contributing to more sustainable and efficient energy systems

1.5 Objectives

The objectives of the project are:

1. To develop a digital Twin model of a $LiFePO_4$ battery for real-time monitoring and predictive diagnostics using MATLAB and Simulink

2. List To accurately estimate SOC and SOH using data-driven machine learning models like LSTM and MLP.

3. To Simulate and analyze battery behavior under various charging/discharging conditions and visualize it through 3D modelling

4. To create a scalable, intelligent framework for advanced battery health assessment applicable to EVs and other storage systems.

1.6 Brief Methodology of the project

The project's methodology focused on designing a smart battery health assessment framework based on digital twin technology, aimed at simulating real-time battery behavior and accurately predicting performance degradation. This involved defining functional requirements for a modern, scalable battery health monitoring system, including support for diverse usage conditions (e.g., variable C-rates, ambient temperatures), real-time parameter estimation (SOC and SOH), and compatibility with both electrical simulation environments and machine learning-based predictive modeling.

The development phase produced a multi-layered simulation testbed. The first layer involved MATLAB-Simulink-based modeling of $LiFePO_4$ battery electrochemical behavior, capturing charging/discharging cycles under multiple load conditions. The second layer consisted of 3D visualization of the battery using MATLAB and Ansys, designed to represent physical characteristics and temperature response. A third layer integrated machine learning models—specifically Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) networks—for long-term SOH prediction over thousands of cycles. These models were trained using cycle-wise capacity degradation datasets and validated against benchmark testing data.

Extensive parameterization and validation ensured the credibility of the digital twin system. This included modeling internal resistance evolution, capacity fade, and dynamic temperature effects, supported by data pre-processing steps like normalization, feature engineering, and time-series splitting. The testbed also included visualization of key battery performance profiles—voltage, current, power—mapped across operational cycles. Charging behavior under constant current/voltage regimes was studied in detail to reflect real-world EV applications.

The performance of the machine learning models was evaluated using metrics such as RMSE and accuracy, and the system is designed for extensibility , allowing for integration with BMS units and future deployment in automotive and energy storage domains.

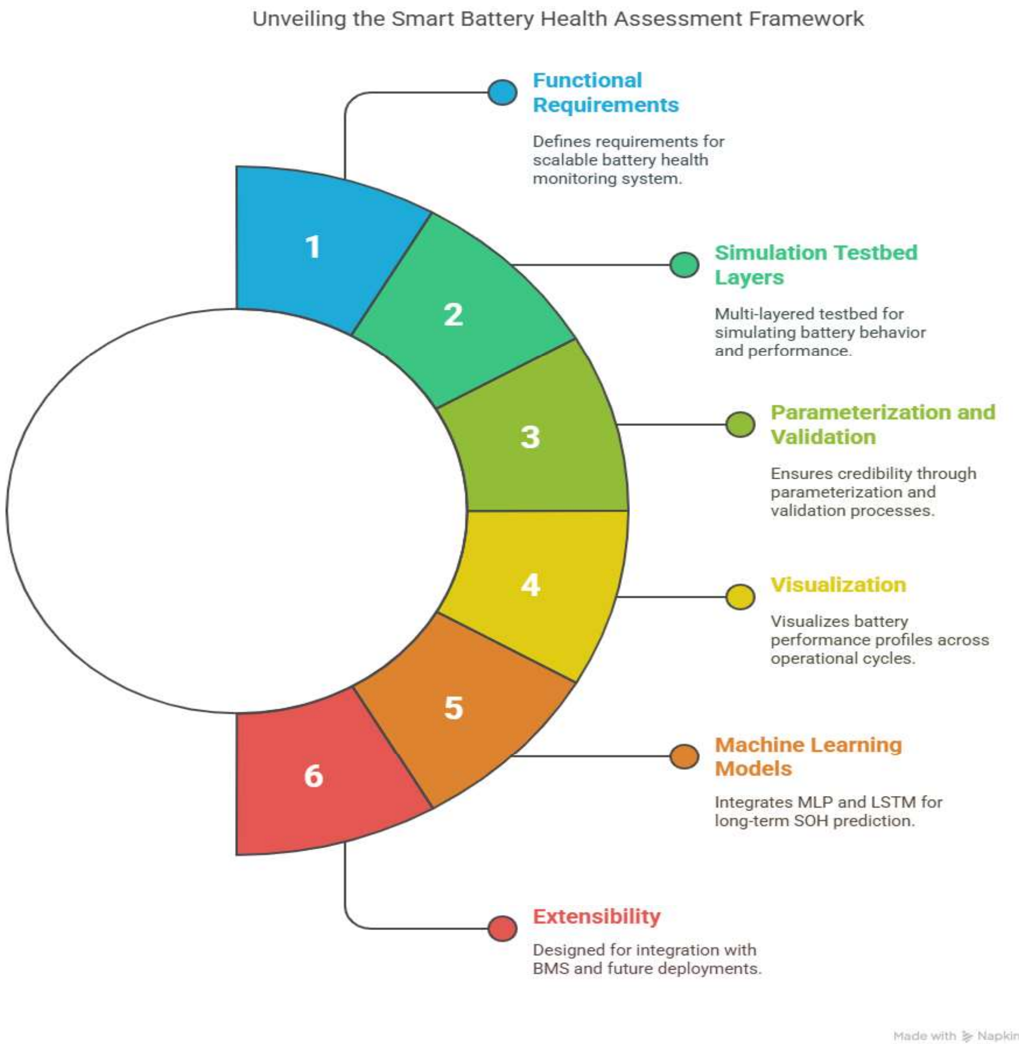


Fig 1.2: Unveiling the Smart Battery Health Assessment Framework

1.7 Assumptions and Constraints of the project

List the assumptions made for the execution of the project in this section. You can also elaborate on the major constraints of the project. This section should clearly state under what conditions your project is valid. It is mandatory to have this section in your project report.

The following assumptions have been made while carrying out this project:

1. The model assumes the use of a $LiFePO_4$ battery, and all simulations, data and predictions are based on this chemistry.
2. It is assumed that temperature , humidity and load conditions remain within the safe operating range.
3. Charging and Discharging profiles are assumed to follow controlled C-rates such as 0.3C to 20C.
4. Data collected like voltage , current, resistance, capacity, temperature and battery chemistry is assumed to be accurate and pre-filtered.
5. MLP and LSTM models trained on historical or lab data are assumed to generalize accurately for future cycle prediction and SOH determination for further large number of cycles to judge its reliability.
6. Battery aging is predictable as it assumes degradation patterns like internal resistance increase and capacity decrease follow normalized trends.
7. Battery is assumed to have 100 % SOH at the time of use.

Talking about constraints , we had the following challenges to be considered before taking up this idea as our topic for Interdisciplinary Project:

1. Real-time or long-cycle battery datasets for training may be limited, reducing the model's prediction accuracy over extreme usage cycles.
2. More accurate digital twin models (e.g., electrochemical models) require high computational power, which might not be practical for real-time applications.
3. The implementation is dependent on MATLAB/Simulink and ANSYS, which are proprietary and resource-intensive software.
4. Time constraints restrict long-term validation over tens of thousands of charge-discharge cycles.
5. Advanced thermal modeling through ANSYS or real-time temperature fluctuation

handling may not be fully integrated.

6. The project is mostly simulation-based, and actual testing on physical hardware or embedded systems is beyond current scope.
7. LSTM/MLP models are constrained by the training data's cycle length (e.g., up to 6000 cycles).
8. Estimation errors may arise due to cumulative error in current integration for SOC or parameter estimation for SOH.

1.8 Organization of the report

This report is structured into six chapters each of them well-defined and systematically building upon the previous to present a comprehensive overview of the smart battery health assessment model. Successive chapters cover everything from the foundation theory of lithium-ion batteries and digital tins, to the design methodology , simulation implementation using MATLAB and Simulink , machine learning-based SOH predictions, and finally , the results, discussions and potential future enhancements.

1. **Chapter 1:** This chapter introduces the motivation behind battery health monitoring and the increasing importance of predictive diagnostics in energy systems and electric vehicles. It outlines the relevance of digital twin technology and the challenges in estimating parameters like State of Charge (SOC) and State of Health (SOH). The chapter also reviews related literature on battery modeling and management systems, and concludes with a clear statement of objectives and the scope of the work.
2. **Chapter 2:** This chapter lays the foundational theoretical framework necessary for understanding the core components of the project. It begins with an in-depth overview of the electrochemical principles governing Lithium Iron Phosphate ($LiFePO_4$) battery operation, including charging/discharging mechanisms, ion transport, and the role of cathode-anode interactions. The concepts of State of Charge (SOC) and State of Health (SOH) are thoroughly explained, highlighting their significance in evaluating battery efficiency, longevity, and performance reliability. Mathematical formulations for SOC and SOH estimation, such as Coulomb counting and capacity fade models, are detailed. The chapter also introduces the concept of the digital twin, outlining its evolution and its application in modern energy systems. It explains how a digital twin can virtually replicate the real-time behavior of a physical battery using sensor data and predictive models, thereby enabling proactive diagnostics and adaptive performance tuning.
3. **Chapter 3:** This chapter outlines the step-by-step methodology adopted for designing

and developing the smart battery health assessment model. It describes the architectural framework of the system, integrating MATLAB Simulink as the core simulation environment for modeling the electrical behavior of $LiFePO_4$ cells. The development of an interactive 3D battery visualization interface using MATLAB is explained, providing an intuitive graphical representation of internal states such as SOC and SOH. Additionally, the use of data transformation pipelines involving normalization, feature engineering, and chronological validation strategies is discussed. A significant portion is dedicated to the incorporation of machine learning algorithms—particularly Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) networks—for long-term prediction of battery degradation trends. The pipeline for training, validation, and forecasting is laid out with a focus on scalability and adaptability. Furthermore, the simulation setup is positioned as a base for future hybrid modeling using both data-driven and physics-informed techniques

4. **Chapter 4:** This chapter details the implementation phase of the project, showcasing how the methodology was translated into a working simulation environment. It begins with the configuration of MATLAB scripts for SOC and SOH estimation and the setup of battery charge-discharge profiles across various C-rates. The process of constructing the 3D visualization GUI is elaborated, including the rendering of a physical battery model with real-time parameter display. The training and deployment of machine learning models on cycle life datasets are also covered, with predictive performance metrics outlined for various forecast horizons. A separate section introduces the role of Ansys in advanced battery modeling, particularly focusing on thermal simulations, material behavior analysis, and heat dissipation mapping—suggesting future integration of thermal diagnostics into the digital twin framework. The overall implementation emphasizes modularity, allowing for expansion into real-time applications.

5. **Chapter 5:** This chapter presents a detailed interpretation of the simulation and model outputs. The performance of the battery under different load and charging scenarios is analyzed using voltage, current, and power profiles plotted across time and cycle indices. Key diagnostic plots are interpreted to show how accurately the models track the battery's internal state and degradation trends. Comparative analyses between baseline estimation methods and the implemented MLP/LSTM-based predictors are performed, demonstrating significant improvements in SOH prediction accuracy, reduction in error margins, and robustness under varying conditions. The analysis also includes the response of the digital twin during operational stress scenarios,

highlighting its ability to adaptively simulate battery behavior. Statistical metrics and visualizations are used to substantiate the reliability of the simulation results and validate the effectiveness of the developed health monitoring approach.

6. **Chapter 6:** The final chapter summarizes the key findings and overall impact of the project. It concludes that the developed digital twin-based battery health assessment system successfully combines simulation, real-time monitoring, and predictive analytics into a cohesive framework. The benefits of using MATLAB and machine learning for adaptive diagnostics are reaffirmed, and the model's effectiveness in forecasting battery performance over long cycles is emphasized. Looking ahead, the chapter outlines potential areas for enhancement, including thermal modeling integration through Ansys, incorporation of real-time sensor data from embedded systems, development of hardware-in-the-loop (HIL) platforms, and deployment of cloud-connected digital twins for large-scale energy applications. Lastly, the chapter explores industrial applications of the project in domains such as electric vehicles, renewable energy storage systems, grid-scale batteries, and smart energy management platforms, highlighting the practical significance and scalability of the proposed solution.

Chapter 2

Concepts and Fundamentals of Battery Health Monitoring

CHAPTER 2

CONCEPT AND FUNDAMENTALS OF BATTERY HEALTH MONITORING

Battery health monitoring is a critical aspect of modern energy storage systems, ensuring safety, longevity, and optimal performance of rechargeable batteries, especially in electric vehicles and renewable energy applications. It involves continuously assessing parameters like State of Charge (SOC), State of Health (SOH), internal resistance, and temperature. Accurate monitoring helps detect early signs of degradation, enabling preventive maintenance and extending battery life. With advancements in sensing, data analytics, and modeling, real-time health assessment has become more reliable and intelligent. This chapter introduces the fundamental concepts behind battery health monitoring, including key parameters, estimation techniques, and the role of digital twin frameworks in predictive diagnostics.

2.1 Battery Fundamentals and $LiFePO_4$ Chemistry

Lithium-ion batteries have become the standard for energy storage in various applications due to their high energy density, efficiency, and rechargeability. Among the various lithium chemistries, Lithium Iron Phosphate ($LiFePO_4$) stands out for its exceptional safety, thermal stability, long cycle life, and tolerance to overcharging. This section introduces the basic structure, operation, and performance metrics of $LiFePO_4$ batteries relevant to health monitoring and digital twin modeling. Understanding the underlying battery chemistry is essential for building accurate diagnostic and predictive models. The integration of these fundamentals lays the groundwork for advanced modeling techniques used in this project.

2.1.1 Electrochemical Working Principle

The Lithium Iron Phosphate ($LiFePO_4$) battery functions through a fundamental electrochemical mechanism that involves the reversible movement of lithium ions between two electrodes: the anode and the cathode. The anode is typically made of graphite, while the cathode consists of $LiFePO_4$ material. During the discharge process, lithium ions are released from the anode and travel through the liquid or solid electrolyte toward the cathode. This ionic movement is coupled with an external flow of electrons through the circuit, which provides usable electrical energy to a connected device or system.

Conversely, during the charging process, an external power source drives lithium ions from the cathode back to the anode, restoring the battery to a charged state. This bidirectional flow of ions and electrons enables the battery to be rechargeable over many cycles. The redox (reduction-oxidation) reactions occurring at each electrode facilitate the storage and release of energy. Specifically, lithium insertion (intercalation) and extraction (deintercalation) occur without significantly altering the crystal structure of the electrode materials. This is a key reason for the exceptional structural stability of $LiFePO_4$ chemistry.

Another unique feature of $LiFePO_4$ is its olivine-type crystal structure, which contributes to enhanced thermal and chemical stability compared to other lithium-ion chemistries like lithium cobalt oxide. This structure resists oxygen release under thermal stress, making the battery inherently safer, even under abuse conditions like overcharging or high temperatures. As a result of these characteristics, $LiFePO_4$ batteries are known for their high operational efficiency**,** long cycle life (often exceeding 2000–3000 cycles), and reliable safety performance, making them ideal for use in electric vehicles, renewable energy systems, and portable electronics where both performance and durability are critical.

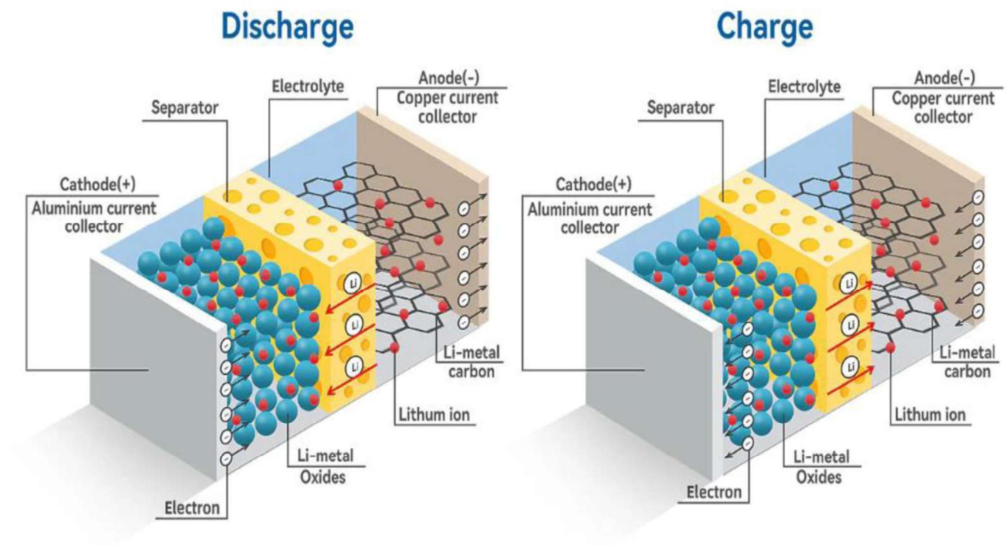


Fig 2.1: Chemical Behavior of $LiFePO_4$ battery

2.1.2 Voltage and Capacity Characteristics

$LiFePO_4$ (Lithium Iron Phosphate) batteries are widely known for their safe operation, long cycle life, and reliable voltage profile, making them ideal for applications like electric mobility and energy storage. With a nominal voltage of 3.2V and a cut-off voltage of around 2.7V, they offer a relatively flat discharge curve, which simplifies the estimation of the State of Charge (SOC). This predictable voltage behavior allows battery management systems to monitor and control battery performance efficiently.

The nominal capacity of 2.7 Ah indicates how much charge the cell can store and supply, while the battery supports charge and discharge rates from **0**.3C (0.81 A) up to 20C (54 A)**,** showcasing its versatility across a wide range of applications.

Over time, as the battery undergoes repeated charge-discharge cycles, performance degradation becomes inevitable due to internal chemical changes, electrode wear, and side reactions. These result in a reduction in capacity and a gradual shift in voltage response under load. For $LiFePO_4$ cells, this degradation is slower compared to other lithium chemistries, allowing for over 2000 cycles at 80% depth of discharge (DoD)**.** Such high cycle life, combined with thermal stability across a wide operating temperature range **(**-20°C to +60°C), makes $LiFePO_4$ an excellent candidate for long-term, real-time monitoring using digital twin models. By analyzing changes in parameters like voltage, current, and capacity, the State of Health (SOH) can be accurately estimated to ensure safe operation, timely maintenance, and improved lifespan.

LiFePO4 Cell Specs

Spec.	Value
Nominal Voltage	3.2 V
Full Charge Voltage	4.2 V
Cutoff Voltage	2.7 V
Nominal Capacity	2.7 Ah
Minimum Charging Rate	0.3C (0.81 A)
Maximum Charging Rate	20C (54.00 A)
Minimum Discharge Rate	0.3C (0.81 A)
Maximum Discharge Rate	20C (54.00 A)
Chemistry Type	Lithium Iron Phosphate
Cycle Life	>2000 cycles @ 80% DoD
Temperature Range	-20°C to +60°C

Fig 2.2: Technical Specifications of $LiFePO_4$ Battery

2.2 State Of Health and State Of Charge

If a specific programming language is required for the project, a section can be allotted in this chapter to discuss it. State of Charge (SOC) indicates the remaining capacity in a battery as a percentage of its full charge. It is calculated using Coulomb counting (integrating current over time) or voltage-based methods. Accurate SOC estimation is critical for battery management systems to ensure safe and efficient operation. Errors in SOC prediction can lead to overcharging or deep discharging, which may damage the battery or shorten its life. In this project, SOC is estimated dynamically using real-time current and voltage data combined with simulation-based models.

State of Health (SOH) represents the overall condition of a battery compared to its ideal or new state, typically expressed as a percentage. It is commonly based on metrics like available capacity, internal resistance, or maximum voltage under load. SOH estimation helps in predicting battery aging and scheduling maintenance or replacement. Traditional methods use physical aging models, but data-driven approaches such as machine learning offer enhanced accuracy. In this project, SOH is predicted using historical cycle data processed through trained models like MLP and LSTM.

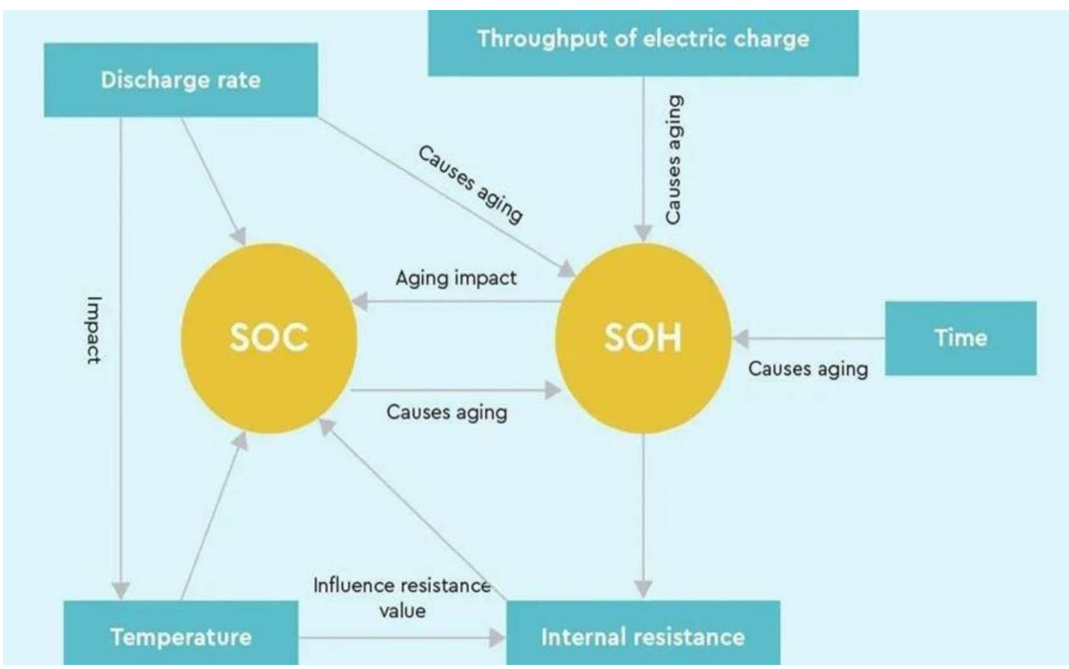


Fig 2.3: Block Diagram for SOH and SOC

We have illustrated the dynamic relationship between **State of Charge (SOC)** and **State of Health (SOH)** in a battery system. SOC, which represents the available charge, is directly influenced by factors such as temperature and discharge rate. A higher discharge rate or extreme temperature conditions can accelerate aging, which in turn affects both SOC accuracy and battery performance.

SOH defines the long-term condition of the battery and deteriorates with time, increased internal resistance, and higher throughput of electric charge (total energy cycled). As SOH decreases, it not only reduces capacity and efficiency but also impacts the reliability of SOC measurements. Internal resistance acts as a key indicator, rising as the battery ages, which further contributes to performance decline. The diagram emphasizes the **bidirectional influence** between SOC and SOH — poor charge management can degrade health, while poor health reduces charge accuracy. This interconnected behavior highlights the importance of real-time monitoring systems like **digital twins** to optimize battery usage, extend lifespan, and ensure safety in applications like electric vehicles and renewable energy systems.

	SOH State of Health	SOC State of Charge
Definition	A measure of a battery's overall condition and aging	A measure of current charge level of a battery
Measure-ment	Estimated based on capacitiy, internal resistance, etc.	Calculated from voltage, current, temperature
Importance	Indicates remaining battery life and need for replacement	Used to determine when recharging is needed
Typical Values	70% to 100% for a healthy battery	0% to 100%, depending on charge level
Impact on Battery Performance	Affects capacity and efficiency	Affects operational range and usage time

T-2.2) SOC vs SOH

2.3 Mathematical Equations

We derived mathematical equations for calculation of SOH and SOC for accurate estimation of battery health.

1. State Of Health(%)

$$SOH(\%) = \left(\alpha * \frac{(Rref)}{Ri}\right) + \left(\beta * \frac{Rctref}{Rcti}\right) + \left(\gamma * \frac{Ci}{Cref}\right)$$

Where,

SOH(%)= State of Health expressed as percentage

Rref= Initial(Reference) internal resistance

Ri=Resistance for after i^{th} cycle

Rct_{ref}= Reference Charge Transfer resistance

Rct_i=Charge Transfer Resistance after i^{th} cycle

Ci=Capacity of the battery after i^{th} cycle

Cref=Reference capacity taken as 2.7Ah

α, β,γ are the weighting factors for dependency on Internal resistance , Charge transfer resistance and capacity.

2. State Of Charge(SOC)

$$SOC(t) = SOC(to) \pm \int_{to}^{t} I(\tau)\, d\tau$$

$\mathrm{SOC}(t)$ = State of Charge at time t

$\mathrm{SOC}(t_0)$ = Initial SOC at time t_0 (typically a known value)

C_{nom} = Nominal capacity of the battery in Ah

$I(\tau)$ = Battery current at time τ (positive for discharge, negative for charge)

$\int I(\tau)d\tau$ = Total charge drawn (or added) over time

2.4 Digital Twin concept for batteries

The details in this chapter can be added in consultation with the project guide. For an internship based projects, subsections can be modified accordingly. The digital twin concept **for batteries** refers to the creation of a virtual, real-time replica of a physical battery system that continuously mirrors its behavior, performance, and condition. This model is not just a static simulation but an active, dynamic system that integrates real-time data with predictive algorithms. In battery systems, a digital twin helps monitor, analyze, and forecast crucial parameters such as State of Charge (SOC), State of Health (SOH), temperature profiles, and degradation trends, enabling smarter battery management and decision-making.

In our project, the digital twin is implemented using MATLAB and Simulink, two powerful platforms for modeling, simulation, and algorithm development. Simulink provides a graphical interface to build the dynamic behavior of the battery by representing electrochemical reactions, thermal effects, and aging mechanisms. The $LiFePO_4$ battery model is constructed in Simulink using components like equivalent circuit models (ECM), current-voltage relationships, and thermal blocks to simulate real-world charging and discharging behaviors. MATLAB is used in parallel to process data, perform optimization, and implement machine learning algorithms such as MLP (Multi-Layer Perceptron) and LSTM (Long Short-Term Memory) for forecasting SOH and cycle-based behavior.

The advantage of using MATLAB/Simulink lies in its ability to integrate multiple subsystems—electrical, thermal, and data-driven—within one environment. This allows us to simulate the battery under various load conditions, generate synthetic data for training machine learning models, and validate prediction accuracy against known datasets. The digital twin continuously updates its internal state using sensor or simulation data and uses predictive models to estimate future conditions. This enables early detection of faults, preventive maintenance, and adaptive control strategies. Additionally, integration with a 3D visualization dashboard allows for intuitive interpretation of battery behavior, enhancing the decision-making capability for operators and engineers alike. By developing a digital twin of the battery in MATLAB/Simulink, we achieve a scalable and modular solution that can be extended to various battery chemistries and configurations. This approach supports applications in electric vehicles (EVs), renewable energy storage, **and** smart grid systems, offering real-time diagnostics, lifecycle management, and improved reliability through predictive insights.

2.4.1 3D-model

The 3D battery model created in MATLAB serves as a comprehensive visualization interface that integrates graphical representation, real-time parameter tracking, and dynamic updates within a single interactive window. This model simulates a $LiFePO_4$ cell, allowing users to monitor and interpret essential battery health metrics in an intuitive and user-friendly environment.

Created using MATLAB , it serves as an integrated visual and analytical tool that brings together real-time simulation, parameter monitoring, and graphical representation of a $LiFePO_4$ battery cell. The visual representation is enhanced by color-coded status indicators, which reflect parameters like State of Charge (SOC) and State of Health (SOH) based on the current operating conditions. Alongside the 3D visualization, a structured side panel presents live specifications and performance metrics including nominal voltage (3.2 V), full charge voltage (3.65 V), cutoff voltage (2.7 V), nominal capacity (2.7 Ah), charge/discharge current rates (ranging from 0.3C to 20C), temperature range (−20°C to +60°C), and live values for voltage, current, power, temperature, and cycle count. The SOC is shown as a progress bar or color gradient on the battery model itself, while SOH trends are displayed through a degradation bar or live chart. These parameters are continuously updated through backend simulation data, driven by MATLAB scripts and Simulink models, which replicate real-world charging and discharging scenarios using machine learning models like MLP and LSTM for health prediction. The interface not only allows users to visualize the battery's internal state but also interact with it by varying simulation conditions, providing a comprehensive diagnostic environment. This fusion of simulation and visualization within a single, interactive window creates a powerful digital twin framework, offering an intuitive and scalable solution for battery performance evaluation, predictive maintenance, and educational demonstrations.

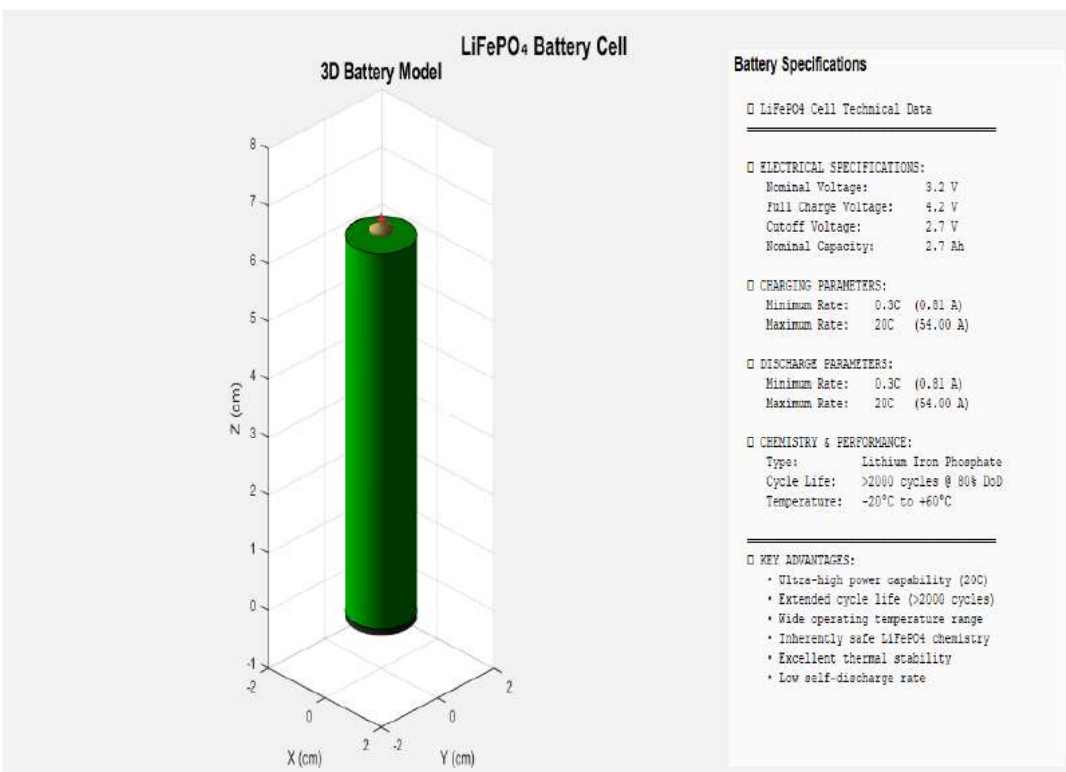


Fig 2.4: 3D Model

2.5 Acronyms and Glossaries

- **$LiFePO_4$ –** *Lithium Iron Phosphate*: A type of lithium-ion battery chemistry known for safety, stability, and long life.
- **SOC –** *State of Charge*: Indicates the remaining charge in a battery as a percentage of its full capacity.
- **SOH –** *State of Health*: Measures the overall condition and aging of a battery relative to its original state.
- **DoD –** *Depth of Discharge*: The percentage of the battery capacity that has been used relative to its total capacity.
- **cycle Life –** The number of complete charge-discharge cycles a battery can undergo before its capacity drops to a specified level (typically 80%).
- **Ah –** *Ampere-hour*: Unit of electric charge indicating how much current a battery can deliver over time.
- **C-rate –** *Charge/Discharge Rate*: The speed at which a battery is charged or discharged relative to its capacity (e.g., 1C = 2.7 A for a 2.7 Ah cell).
- **Nominal Voltage –** The typical voltage of a fully functioning cell during normal operation.
- **Cutoff Voltage –** The lower voltage limit at which the battery stops discharging to avoid damage.
- **Full Charge Voltage –** The upper voltage limit to which the battery can be charged.
- **BMS –** *Battery Management System*: Electronic system that manages the charging, discharging, and overall safety of the battery.
- **IR –** *Internal Resistance*: Resistance within the battery that affects voltage response and heat generation.
- **Thermal Runaway –** A rapid increase in temperature within the battery that can lead to fire or explosion.

- **Thermal Mapping –** Spatial analysis of temperature distribution in the battery to identify hotspots and optimize cooling.
- **Heat Isolation Materials –** Materials used to insulate battery cells and control thermal spread.
- **Nominal Capacity –** The manufacturer-rated energy capacity of a battery, often in Ah or Wh.
- **Calendar Aging –** Degradation of battery performance over time regardless of usage

2.6 Summary

This chapter introduces essential terms related to battery chemistry, diagnostics, and modeling frameworks. It explains the core functionality of $LiFePO_4$ (Lithium Iron Phosphate) cells, detailing properties like nominal voltage (3.2 V), cut-off voltage (2.7 V), and nominal capacity (2.7 Ah). The concepts of state of charge (SoC) and state of health (SoH) are explained as critical indicators of battery performance and lifespan. Electrochemical behavior such as charge/discharge cycles, C-rate (charging/discharging current as a fraction of capacity), and DoD (depth of discharge) are elaborated. Furthermore, key modeling parameters like voltage profiles, capacity fade, and impedance growth are introduced. The chapter also sets the foundation for digital twinning, referencing terms such as real-time monitoring, virtual replicas, and predictive diagnostics.

Chapter 3

Methodology and System Architecture

CHAPTER 3

METHODOLOGY AND SYSTEM ARHITECTURE

Every chapter should start with an introduction paragraph. This paragraph should brief about the flow of the chapter. This introduction can be limited within 4 to 5 sentences. The chapter heading should be appropriately modified (a sample heading is shown for this chapter).But don't start the introduction paragraph in the chapters like "This chapter deals with....". Instead you should bring in the highlights of the chapter in the introduction paragraph.

3.1 Digital Twin Modelling using Simulink

The digital twin of the battery is implemented using MATLAB Simulink to replicate the physical behavior of a $LiFePO_4$ cell in real time. The model simulates essential electrochemical dynamics, including charging and discharging cycles, terminal voltage response, and state transitions. Parameters such as State of Charge (SOC), State of Health (SOH), temperature, and internal resistance are monitored and updated dynamically through simulated data flows. This virtual environment allows for high-fidelity emulation of real-world battery performance without the need for physical hardware. The architecture is designed to be modular, allowing seamless integration with machine learning prediction models and external sensor inputs. Moreover, the simulation enables users to observe how the battery behaves under different current rates, voltages, and temperature conditions, making it highly effective for predictive diagnostics and early fault detection. This virtual representation not only facilitates lifecycle prediction but also serves as a testing platform for optimization and control strategies, contributing significantly to robust battery management system (BMS) development.

To reflect the battery's electrochemical nature accurately, the model incorporates equivalent circuit elements such as open-circuit voltage sources, internal resistors, and RC networks. The simulation captures transient responses and long-term degradation patterns by applying mathematically formulated block structures and physical modeling principles. This approach facilitates fault diagnosis, thermal behavior tracking, cycle aging analysis, and real-time decision-making.

In addition, the model is designed with flexibility, allowing the integration of machine learning models for SOH prediction. Input-output ports are configured to receive training data and send performance metrics to external environments. Custom scripts in MATLAB are used to pre-process data, feed simulation inputs, and visualize SOC/SOH curves over time. This comprehensive digital twin framework provides a safe, virtual platform to run experiments, validate battery control algorithms, and anticipate performance issues, all without impacting the actual battery hardware. The result is a robust, scalable system that lays the foundation for intelligent, data-driven battery management in smart energy systems.

Ordinarily, the majority of the notes you take during the research phase of writing your report will paraphrase the original material. Paraphrase only the essential ideas. Strive to put original ideas into your own words without distorting them."

3.2 Machine Learning Integration

When To enhance the predictive capability of the digital twin, machine learning models are integrated into the simulation framework to estimate and forecast key battery health parameters, particularly the State of Health (SOH). Two primary models used in this project are the Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) networks. These models are trained on historical cycling data such as voltage, current, temperature, and capacity degradation over time to predict future performance trends of the $LiFePO_4$ battery.

The MLP model is used for its ability to capture non-linear relationships in high-dimensional data. It takes input features derived from cycle count, resistance change, temperature variation, and capacity fade, and maps them to a corresponding SOH value. The LSTM model, on the other hand, is particularly effective in capturing temporal dependencies in time-series data. It analyzes long-term patterns across charge-discharge cycles to provide more accurate, cycle-aware SOH predictions. Before training, data preprocessing techniques like normalization, filtering, and feature engineering (e.g., resistance ratios, capacity loss rate) are applied to improve model performance. The dataset is split chronologically to preserve time dependence, with training typically on the first 80% of the cycles and validation on the remaining 20%. Models are evaluated using metrics such as RMSE and MAE, and predictions are validated against simulated or experimental results.

Once trained, these ML models are embedded into the digital twin pipeline using MATLAB's Deep Learning Toolbox and Simulink function blocks. This enables real-time inference within the simulation environment, allowing continuous battery health monitoring. The integration of machine learning thus transforms the digital twin from a passive simulation model into an intelligent predictive system capable of self-learning and adaptive diagnostics.

The MLP model is used due to its capability in handling regression tasks involving high-dimensional feature spaces. It receives input features derived from cycle count, capacity drop, voltage fluctuations, and internal resistance variations, and maps them to the corresponding SOH values. This model captures the static, non-temporal relationships present in the dataset and is computationally efficient for fast predictions. While the LSTM model, a type of recurrent neural network (RNN), is employed to handle temporal dependencies inherent in battery cycling data. It analyzes sequential patterns in charge-discharge cycles, making it highly suitable for predicting future SOH based on past operational history. This is particularly critical for long-term forecasting, where trends such as gradual capacity fade and resistance build-up need to be recognized over hundreds or thousands of cycles.

The models are trained and validated using MATLAB's Deep Learning Toolbox and Statistics and Machine Learning Toolbox, with performance assessed using metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 score. Once validated, the models are exported and integrated into Simulink using Simulink Function Blocks or MATLAB Function blocks, enabling real-time inference within the simulation loop. This seamless integration allows the digital twin to continuously monitor battery performance, adapt to operational changes, and predict future behavior without requiring manual recalibration. It also sets the foundation for predictive maintenance, early fault detection, and life-cycle optimization, providing a powerful tool for industries relying on battery systems in critical applications such as electric vehicles, aerospace, and grid storage

3.3 Battery Visualization Using Ansys

To complement the simulation and predictive aspects of the digital twin, ANSYS is utilized to develop a 3D visual model of the $LiFePO_4$ battery, offering an in-depth visualization of its structural, thermal, and electrochemical characteristics. The 3D model helps in illustrating the physical architecture of the battery, enabling better interpretability of simulation results and enhancing the real-time monitoring experience when integrated with MATLAB Simulink. The battery model includes essential components such as the anode, cathode, separator, electrolyte regions, and casing. Each of these parts is accurately dimensioned and assigned material properties based on real-world specifications. ANSYS enables high-fidelity rendering and simulation of thermal behavior, structural stresses, and electric current flow within the battery during operation. One of the key uses of ANSYS in this project is thermal analysis. Heat generation and dissipation during charge and discharge cycles are modeled using Finite Element Analysis (FEA). This includes simulating temperature rise under various C-rates and identifying hotspots that could affect performance or safety. Visualization of these results as heat maps provides a clear understanding of the thermal gradients inside the battery and guides the design of better thermal management strategies.

The 3D model created in ANSYS is also intended to support future integration of heat insulation material testing and cooling mechanisms by simulating the placement of insulation layers and airflow channels. This will aid in material selection and design validation for real-world battery packs, especially in electric vehicle (EV) and energy storage system applications. Additionally, the 3D visualization developed in ANSYS serves as a side panel interface in the real-time digital twin dashboard. This interface displays live simulation outputs such as temperature, voltage, SOC, SOH, and cycle number, mapped directly onto the visual model. The goal is to create an interactive and intuitive environment where engineers and users can monitor the battery's performance visually and make informed decisions quickly. In summary, ANSYS-based visualization adds a critical dimension to the digital twin framework by bridging the gap between data-driven analysis and physical system understanding. It not only enhances interpretability but also opens the path for advanced structural and thermal optimization in battery system design.

The geometry of the battery includes all essential components—anode, cathode, separator, current collectors, tabs, and casing—modeled to scale based on actual cell dimensions. Material properties such as thermal conductivity, specific heat, and electrical resistivity are applied to each layer to ensure that simulation outcomes match real-world behavior. This setup provides a platform for multi-physics simulation, combining thermal, structural, and electrochemical domains in a single environment. One major application of this ANSYS model is thermal mapping. As the battery undergoes repeated charging and discharging cycles, it generates heat due to internal resistance and electrochemical reactions. ANSYS is used to simulate these thermal profiles under different load conditions and C-rates. The heat distribution is visualized using temperature contours and heat flow vectors. This enables us to locate critical zones of high temperature, often referred to as hotspots, which may degrade battery performance or even pose safety risks.

3.4 Ansys Ideology

The underlying ideology behind using ANSYS in this project revolves around leveraging multi-physics simulation to understand, predict, and optimize the real-world behavior of the $LiFePO_4$ battery. ANSYS is not merely used for creating 3D representations, but for establishing a virtual environment that captures the thermal, structural, and material-based dynamics of a functioning battery.

The goal is to bridge the gap between digital modeling and physical performance, aligning closely with the digital twin philosophy. At its core, the ANSYS ideology focuses on physics-based modeling. This means that instead of relying only on empirical or data-driven methods, the battery is simulated based on first principles—governed by heat transfer equations, material stress-strain relationships, and thermal expansion laws. The battery's layered architecture, including its cathode, anode, electrolyte, and casing, is modeled with material-specific properties such as thermal conductivity, density, specific heat, and elastic modulus. This allows for highly realistic simulations of heat dissipation, structural integrity, and failure modes under various operating conditions. One of the primary pillars of the ANSYS methodology is thermal analysis. Batteries experience heat generation due to internal resistance and chemical activity, especially under fast charging, high discharge rates, or prolonged operation. ANSYS solves the heat equation for these scenarios using finite element analysis (FEA), generating high-resolution heat maps that indicate critical regions prone to overheating. These simulations allow for exploring the effectiveness of different cooling strategies or insulating materials, and suggest design adjustments to improve heat dissipation and thermal safety. Another essential principle is mechanical simulation**.** Batteries undergo mechanical stress during operation, especially in mobile applications like electric vehicles. ANSYS allows simulation of external vibrations, impact forces, and structural fatigue, which can degrade internal components or cause casing failures. Modeling these effects helps in choosing appropriate enclosure materials and optimizing the physical design to enhance durability.

3.4.1 Tools Used for Ansys Battery Modelling

In this project, ANSYS Workbench serves as the central simulation environment for constructing, analyzing, and validating the battery's 3D digital representation. Workbench provides an intuitive platform to integrate multiple physics-based solvers like thermal, structural, and electrochemical analysis into a unified simulation workflow. The geometry of the $LiFePO_4$ battery cell is modeled in detail, including layers such as anode, cathode, electrolyte, separator, and current collectors. The modular nature of Workbench allows for quick iterations and seamless linkage between different simulation modules, which is particularly useful when analyzing the effect of design or material changes on battery behavior. ANSYS Mechanical is used extensively for performing structural simulations. This includes modeling the battery’s casing and internal layers to analyze deformation, stress distribution, and vibration response under mechanical loads. Such simulations are crucial when evaluating battery performance in mobile or high-impact environments like electric vehicles or aerospace systems. By applying real-world boundary conditions—such as external pressure, mounting stress, or drops—the tool helps in assessing material durability, identifying weak zones, and optimizing component thickness and support. For thermal performance modeling**,** ANSYS Steady-State and Transient Thermal solvers are employed. These tools simulate heat generation due to internal resistance during charging and discharging, and predict how the heat is conducted through different layers and dissipated to the surroundings. Important parameters like thermal conductivity, specific heat, and convection coefficients are defined for each material. The tool can also simulate the effect of external thermal management systems like cooling plates or insulation layers, providing insights into preventing thermal runaway and improving battery longevity.

Finally, ANSYS Fluent may also be utilized for advanced electrochemical-thermal coupled simulations, especially in future work. Fluent supports modeling of ion transport, reaction kinetics, and electric potential distribution across the battery cell, allowing for a more physics-accurate representation of battery operation. When integrated with thermal models, Fluent can simulate how changes in electrochemical behavior affect heat generation and vice versa. This multi-physics approach is aligned with the digital twin philosophy of achieving high-fidelity, real-time representation of complex systems. These tools collectively make ANSYS a powerful platform for predictive modeling, failure analysis, and virtual prototyping of battery systems.

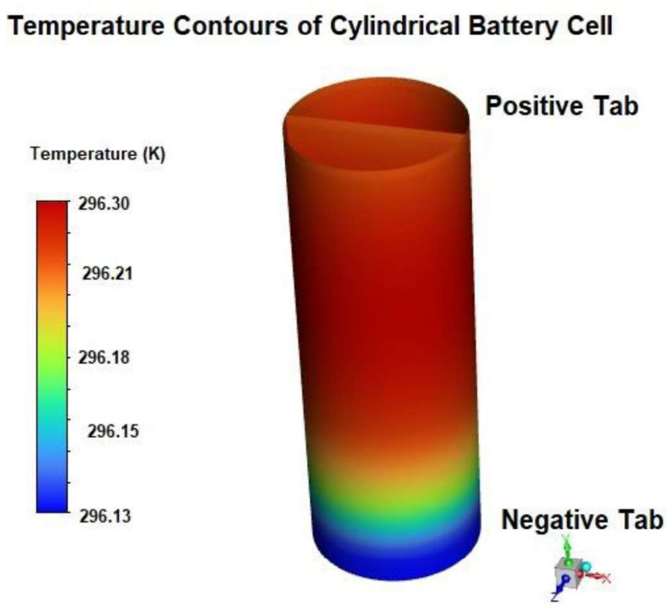


Fig 3.1: Temperature Contours of Cylindrical Battery Cell

Ansys Tool/ Module	Purpose	Key Features	Application in Battery Modelling	Benefits
ANSYS Workbench	Central simulation environment	• Integrates multiple physics solvers • Modular, intuitive platform • Enables quick design iterations	- Constructs detailed 3D battery geometry (anode, cathode, electrolyte, separator, collectors) - Links thermal, structural, electrochemical analyses	- Identifies weak zones - Optimizes structural integrity and support
ANSYS Thermal (Steady-State & Transient)	Thermal performance modeling	• Simulates heat generation and conduction • Defines material thermal properties Models cooling systems	- Predicts heat distribution during charge/discharge - Assesses effects of cooling plates, insulation layers	- Prevents thermal runaway - Improves battery lifespan and safety
ANSYS Fluent Battery Module	Advanced electrochemical-thermal coupled simulation	• Models detailed battery physics including charge/discharge cycles, heat generation, degradation • Enables thermal abuse and safety analysis	- Simulates battery casing and internal layers under mechanical loads - Evaluates durability in mobile/high-impact scenarios	- High-fidelity digital twin capability - Predictive modeling and failure analysis - Customizable via User Defined Functions (UDFs)
ANSYS Mechanical	Structural simulation	• Models deformation, stress, vibration • Applies realistic boundary conditions (pressure, mounting, drops)	- Simulates battery casing and internal layers under mechanical loads - Evaluates durability in mobile/high-impact scenarios	- Identifies weak zones - Optimizes structural integrity and support

T-3.1: Battery Modeling Capabilities of ANSYS Simulation Modules

ANSYS Workbench serves as the central simulation environment where different physics solvers are integrated for multiphysics battery modeling. Its modular and intuitive design allows for efficient project setup and seamless switching between different physical domains. In the context of battery modeling, ANSYS Workbench is used to construct detailed 3D battery cell structures including the anode, cathode, electrolyte, separator, and current collectors. Additionally, it links thermal, structural, and electrochemical analyses in a single workflow. This helps in identifying weak mechanical or thermal zones within the battery structure and facilitates quick design iterations to optimize both structural integrity and overall battery support.

ANSYS Thermal (Steady-State and Transient) is primarily employed for thermal performance simulation. It models heat generation and conduction throughout the battery during charge and discharge cycles. This module defines the thermal properties of materials and is capable of simulating the effects of different cooling mechanisms or insulation layers. These simulations are critical for predicting heat distribution across the battery under various operating conditions.

By accurately identifying potential areas for thermal buildup, it helps prevent thermal runaway and improves battery safety and operational lifespan through optimized thermal management strategies.

ANSYS Fluent Battery Module offers advanced electrochemical-thermal coupled simulations. It models complex battery behaviors such as charge/discharge cycles, thermal degradation, heat generation, and dynamic aging processes. Moreover, it supports simulation of thermal abuse scenarios, making it valuable for safety and durability analysis. Fluent allows for high-fidelity modeling of battery internals under stress and integrates User Defined Functions (UDFs) for customization. This makes it especially suitable for predictive modeling and failure forecasting in high-impact applications like electric vehicles and aerospace systems where safety and efficiency are paramount.

ANSYS Mechanical is dedicated to structural simulations that assess deformation, stress, strain, and vibration in battery materials and casings. This module applies realistic boundary conditions, including mechanical loads like pressure, drop impact, and thermal expansion. In battery applications, it is used to simulate the casing and internal components under various load scenarios to evaluate mechanical durability. This ensures that the battery remains safe and structurally sound even under rough handling or in mobile use cases. It also identifies structural weak points and supports the design of more robust battery enclosures.

Chapter 4

Simulation and Implementation

CHAPTER 4

SIMULATION AND IMPLEMENTATION

Simulation and implementation form the backbone of validating a digital twin model, ensuring theoretical concepts translate into practical performance. In this project, the battery system is simulated using MATLAB and Simulink for electrical behavior, while ANSYS is employed for structural and thermal modeling. The workflow includes constructing a virtual model of the battery, integrating charge–discharge characteristics, and mapping physical parameters like voltage, temperature, and internal resistance. Machine learning models such as MLP and LSTM are trained on cycle data to predict State of Health (SOH) and State of Charge (SOC). Real-time data monitoring is embedded to evaluate the model's responsiveness under dynamic conditions. Together, these implementations allow the digital twin to serve as an accurate, real-time mirror of the battery's health and behavior in operation.

To achieve a comprehensive simulation environment, multiple software platforms are utilized in an integrated manner. MATLAB and Simulink serve as the primary tools for modeling the electrical behavior of the battery, simulating charge–discharge cycles, and implementing control algorithms for SOC and SOH estimation. These simulations incorporate mathematical models and real-world data to replicate the battery's dynamic response under varying load conditions. Additionally, ANSYS is employed to develop the structural and thermal aspects of the battery model. This includes simulating heat generation, identifying thermal hotspots, and visualizing internal component stresses during operation. The integration of machine learning models like MLP and LSTM further enhances predictive capabilities, allowing the system to anticipate performance degradation well in advance. Together, these simulations provide a highly accurate digital representation of the physical battery system, enabling proactive diagnostics and real-time monitoring

4.1 Software Tools Used

We have utilized a variety of software tools for this project to ensure accurate data analysis, battery modelling and simulating real time dynamic behavior of battery.

4.1.1 MATLAB and Simulink

MATLAB and Simulink played a central role in modeling the electrochemical behavior of the Lithium Iron Phosphate ($LiFePO_4$) battery.

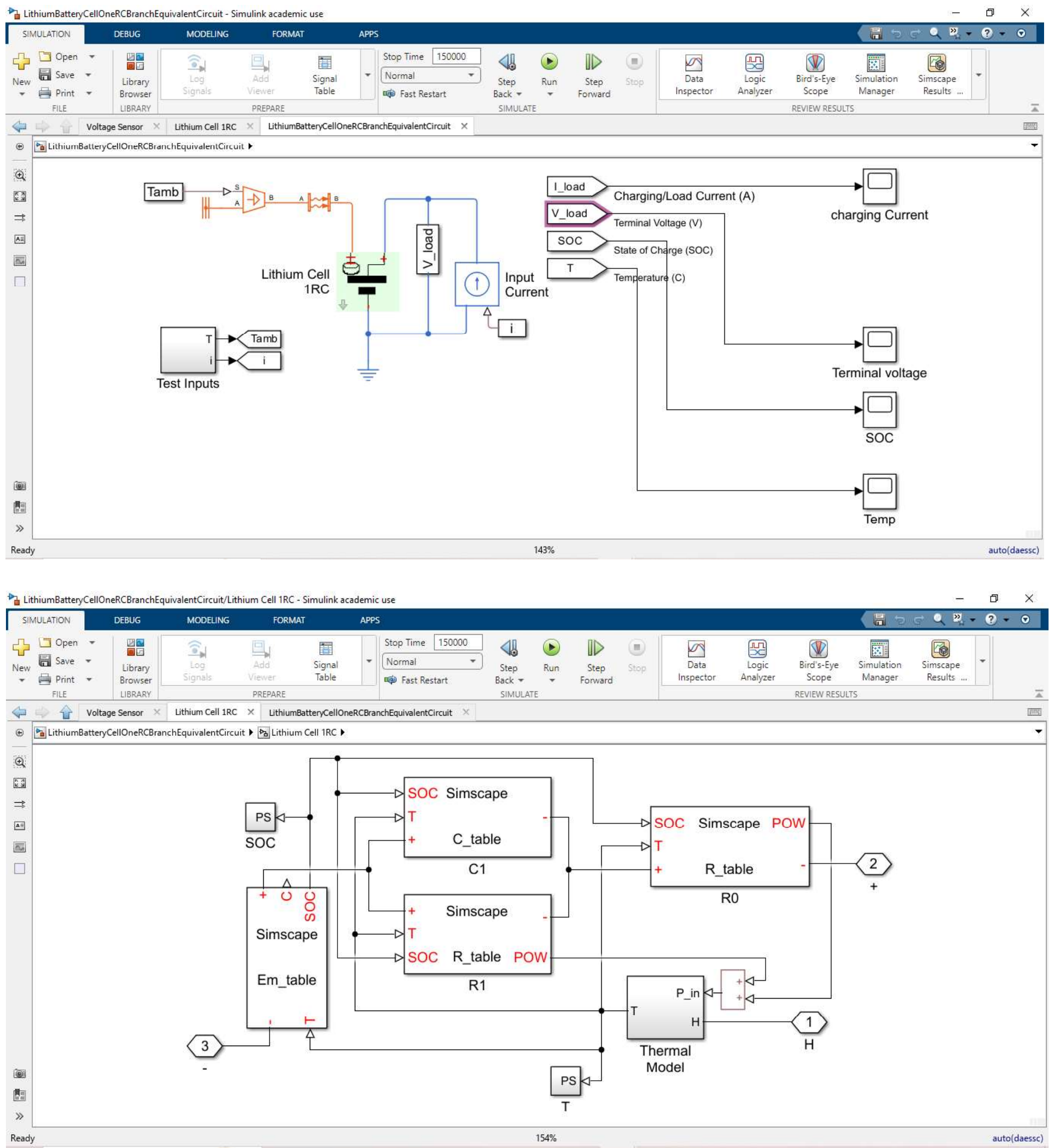


Fig 4.1: Simulink Model

Simulink's graphical environment allowed the creation of real-time block-based models to simulate battery charge/discharge behavior, thermal effects, and control strategies. These simulations incorporated equations for SOC and SOH estimation, voltage-current relationships, and energy flow. Moreover, MATLAB scripts were used for signal processing, data cleaning, and machine learning model integration, particularly for training MLP (Multi-Layer Perceptron) and LSTM (Long Short-Term Memory) networks. In addition to these, **Simscape Electrical** (within the MATLAB ecosystem) was used for component-level modeling, enabling more accurate electrical simulation of internal resistance, capacitance, and switching dynamics. This helped in tuning the behavior of power electronic interfaces, battery management systems, and real-time control feedback loops.

4.1.2 ANSYS

ANSYS was employed for structural and thermal modeling of the battery. The ANSYS Workbench environment provided a robust simulation platform to study heat generation, thermal diffusion, and mechanical stresses during various operating conditions. This was essential for evaluating safety, thermal runaway risks, and optimizing the cell layout for improved thermal performance.ANSYS also supported material property mapping and layer-wise modeling of the battery pack, offering insight into physical deformations and hotspot distributions.

4.1.3 Machine Learning

Linear regression was employed as a basic benchmark model to predict the State of Health (SOH) of the battery over time. This method assumes a linear relationship between the cycle number (independent variable) and the SOH (dependent variable). Despite its simplicity and interpretability, the model performed poorly due to the inherently non-linear and complex degradation behavior of lithium-ion batteries. The R^2 score obtained was -1.215, indicating that the model performed worse than simply predicting the mean of the data. This negative score, along with large residual errors, highlighted the model's inability to capture curvature, long-term capacity fade, or fluctuations due to cycling and temperature effects. As a result, linear regression was deemed unsuitable for this application, but its results served as a useful baseline to compare against more sophisticated models.

To better capture the non-linear nature of battery degradation, a third-degree polynomial regression model was applied. This model uses the cycle number and its higher-order terms (squared and cubed) to predict SOH, thereby accommodating curvature in the data. The regression equation takes the form:

$$\mathrm{SOH} = \beta^0 + \beta^1 x + \beta^2 x^2 + \beta^3 x^3$$

where x represents the cycle number. The model showed a significant improvement over linear regression, achieving an R^2 score of 0.976 and a Mean Absolute Error (MAE) of only 0.221%. These metrics indicate a strong fit and low prediction error, demonstrating that polynomial regression is well-suited for modeling smooth, continuous degradation patterns. Moreover, the model is lightweight and easy to implement, making it ideal for embedded systems or low-computation environments where interpretability is also important. However, while it performed well on available data, it may struggle to generalize in the presence of abrupt changes or multi-phase degradation behavior.

SVR was tested with a Radial Basis Function (RBF) kernel to allow for non-linear regression between input cycle data and output SOH. While SVR has a reputation for handling high-dimensional regression problems effectively, it proved unsuitable in this case. The model experienced severe overfitting, as evidenced by an extremely poor R^2 score of -9.284 and a high MAE of 4.25%. Despite attempts at hyperparameter tuning, the model remained highly sensitive to kernel width and regularization factors, making it unstable and unpredictable in real-world application. Furthermore, the SVR model required significantly longer training times and lacked the interpretability and robustness offered by other methods. These drawbacks made it clear that SVR, at least with the RBF kernel and current dataset, was not a viable choice for accurate and reliable SOH prediction.

The most successful model in this study was a Multi-Layer Perceptron (MLP) implemented using the PyTorch deep learning framework. The MLP is a feedforward neural network designed to capture complex, non-linear mappings between input cycle data and SOH. The architecture consisted of two hidden layers, each with 128 neurons, followed by a single output neuron representing the SOH value. The ReLU activation function was used for non-linearity, and the model was trained using the Mean Squared Error (MSE) loss function with the Adam optimizer. The MLP achieved outstanding performance, with an R^2 score of 0.988 and an MAE of just 0.147%. Moreover, its prediction latency was less than 1 millisecond, making it suitable for real-time applications. The model demonstrated excellent generalization and stability across validation data and is highly recommended for deployment in digital twin environments and embedded battery monitoring systems. Its combination of high accuracy, speed, and scalability make it the most robust model in this analysis.

An LSTM network was also implemented in PyTorch to leverage the time-series nature of battery cycling data. The architecture included two LSTM layers with a hidden size of 50 units and a dropout of 0.2 to prevent overfitting. The model received input sequences consisting of 30 consecutive cycles and attempted to predict future SOH values. While theoretically suitable for modeling temporal dependencies, the LSTM underperformed in this specific case. It produced a low R^2 score of -1.896 and a high MAE of 2.437%, indicating poor prediction capability and unstable learning. One possible reason for this performance is that the dataset lacked rich temporal patterns or long-term memory dependencies, which are critical for LSTM to perform well. Additionally, the model's complexity may have led to overfitting or difficulties in learning meaningful patterns from the limited input sequence. While LSTMs are generally powerful tools in time-series forecasting, they were not the right fit for this particular application and dataset configuration

Model Type	R^2 Score	MAE(%)	Strengths	Weaknesses
Linear Regression	-1..215	High	Simple, Fast, interpretable	Poor fit for non-linear battery behavior
Polynomial Regression	0.976	0.221	Good accuracy ,low cost, interpretable	May not handle multi-phase degradation
SVR(RBF Kernel)	-9.284	0.425	Non-linear Regression capability	Overfitting, highly sensitive
MLP	0.988	0.147	High accuracy,fast inference,robust	Requires intense training
LSTM	-1.896	2.437	Suitable for time series tasks	Over parameterized and underperformed on datasers

T-4.1: Comparison of Machine Learning Models Used for SOH Prediction

The performance comparison of various machine learning models highlights the importance of selecting the right architecture for accurate battery health estimation. Traditional models like linear regression and support vector regression (SVR) performed poorly, primarily due to their inability to capture the non-linear and complex nature of battery degradation. Polynomial regression (3rd degree), on the other hand, provided a strong balance between accuracy and simplicity, making it a viable lightweight option.

Among all tested models, the Multi-Layer Perceptron (MLP) emerged as the most effective, achieving the highest accuracy ($R^2 = 0.988$) and the lowest prediction error (MAE = 0.147%). Its deep learning structure enabled it to learn non-linear patterns in the data with excellent generalization and real-time inference capability. In contrast, the LSTM model, though designed for sequential data, underperformed due to overparameterization and limited temporal complexity in the dataset. Therefore, the final system integrates MLP into the digital twin framework as the primary SOH prediction engine due to its robustness, speed, and precision.

4.2 Model Development Steps

The development of the digital twin-based battery health monitoring model followed a structured and modular approach, combining electrochemical modeling, machine learning integration, and 3D visualization. Each phase was carefully designed to ensure accuracy, scalability, and real-time responsiveness. The following steps outline the complete process from conceptualization to implementation.

4.2.1 Requirement Analysis and Specification Finalization

The first step involved identifying the key functional and technical requirements for the digital twin. This included defining what parameters the model should estimate (such as SOC and SOH), the expected input data (voltage, current, temperature, and cycle count), and the simulation environment's real-time behavior. Specifications for the $LiFePO_4$ battery, including its electrical, thermal, and mechanical properties, were also collected at this stage to guide accurate modeling

4.2.2 Battery Modelling in Simulink

The core electrical model of the battery was developed using MATLAB Simulink. Using Simscape Electrical components, a behavioral model was created to replicate the charge–discharge cycles of the $LiFePO_4$ battery. Elements such as internal resistance, open circuit voltage, and thermal characteristics were incorporated. The model outputs dynamic voltage and current data during operation, forming the foundation for estimating SOC and SOH in later stages.

4.2.3 Machine learning Pipeline Construction

Historical cycle data was processed to train machine learning models for SOH estimation. Preprocessing included normalization, feature extraction (e.g., resistance ratio, capacity fade rate), and noise removal. Various ML models such as Polynomial Regression, SVR, MLP, and LSTM were implemented and evaluated. MLP, which showed the best performance in terms of prediction accuracy and efficiency, was selected for integration into the final model. Training was done using PyTorch, and the model was exported for use within MATLAB.

4.2.4 Integration of ML with Simulink

The trained MLP model was integrated into the Simulink environment using MATLAB Function blocks. The real-time outputs from the battery simulation (like cycle number and voltage) were fed into the neural network, which predicted the SOH dynamically. This enabled the digital twin to act as both a monitoring and predictive system, updating health status as the simulation progressed.

4.2.5 3D-Visualization and GUI development

The 3D battery model was visualized with specifications shown in a side panel. Real-time parameters like voltage, SOC, SOH, and temperature were displayed in sync with the simulation. This created a cohesive dashboard for visual inspection and diagnostics.

4.2.6 Thermal and Structural Modeling in ANSYS

To analyze physical safety and thermal performance, a detailed 3D battery model was created in ANSYS Workbench. The model included layers such as anode, cathode, separator, and casing, each assigned real material properties (e.g., thermal conductivity, density). ANSYS Thermal (steady-state and transient) simulations were performed to evaluate heat generation and dissipation during charging and discharging. Additionally, ANSYS Mechanical was used to assess structural integrity under mechanical loads and thermal expansion. This ensured the design could withstand realistic operating environments, particularly in EV or grid storage applications.

4.2.7 Validation and Testing

Once all the components are intact, the complete digital twin system was tested with simulated and experimental data to validate its predictive accuracy and real-time response. The SOH predictions were compared against actual values for unseen cycles. Thermal simulation results from ANSYS were cross-validated with theoretical estimates to ensure model fidelity. Metrics such as R^2, MAE, and inference latency were used for model assessment, confirming the reliability of the twin.

4.2.8 Provision for Future Expansion

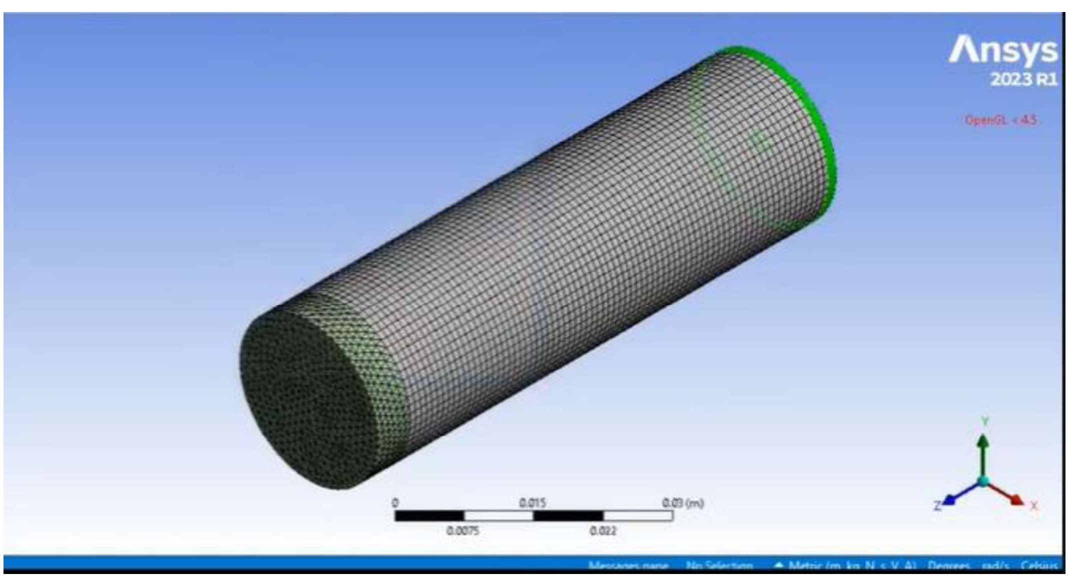
The model architecture is modular, allowing easy replacement or retraining of ML components and scalability for different battery chemistries. Future extensions include live data acquisition from hardware BMS units, advanced fault detection using thermal patterns from ANSYS, and deployment in edge devices or cloud platforms for large-scale battery monitoring.

4.3 3D Visualization and Parameter mapping

The 3D visualization component of the project was developed to enhance the interpretability and interactivity of the digital twin battery model. Using MATLAB's built-in 3D plotting functions and App Designer interface, a graphical representation of the cylindrical $LiFePO_4$ battery was created. The model includes features such as the cell body, top and bottom terminals, casing, and labeling, offering a close-to-real visual experience. The design is scaled to match actual dimensions and is rendered with surface lighting and rotation control, allowing users to examine the battery from multiple angles. To make the simulation more intuitive, a parameter mapping panel was added beside the 3D model. This panel displays real-time values of key performance indicators such as voltage, current, power, temperature, state of charge (SOC), and state of health (SOH).

These values are updated dynamically during the simulation, offering a live dashboard effect. The side panel is created using ui-panels and UI controls within MATLAB, and the data is linked directly to the outputs of the Simulink simulation and machine learning predictions. The purpose of this 3D visualization is not just for aesthetics, but to provide a real-time digital interface that reflects the internal state and behavior of the battery. It acts as a visual front-end for the digital twin system and bridges the gap between backend simulation data and user interpretation. Engineers and researchers can use this visual interface to quickly assess performance, identify anomalies, and analyze trends without needing to interpret raw data or graphs alone. The parameter mapping also serves as a diagnostic tool, allowing users to monitor critical parameters side by side and understand how they influence one another. For instance, a rise in temperature due to high C-rate charging can be visualized in real time along with its effect on voltage drop or SOH degradation. Such interactions enhance system understanding and support more informed decision-making in both experimental and industrial scenarios.

In addition to MATLAB-based visualization, ANSYS was utilized to develop a detailed 3D model of the $LiFePO_4$ battery cell, focusing on both its physical geometry and thermal behavior. The battery's layered internal structure, including the anode, cathode, electrolyte, and outer casing, was modeled in ANSYS Workbench to simulate how the cell behaves under operational stress and temperature fluctuations. Each component was assigned real-world material properties such as density, thermal conductivity, and specific heat capacity to ensure accurate physical representation.



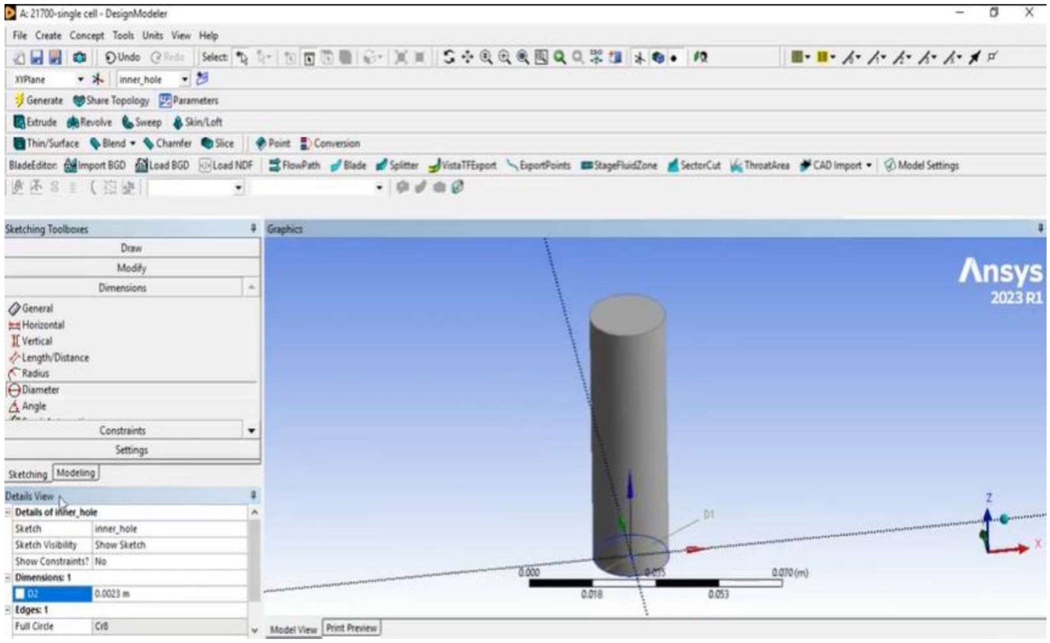


Fig 4.2: Ansys 3D-model

The 3D visualization in ANSYS helped in understanding the spatial configuration of the battery and how its internal architecture influences thermal and mechanical performance. The simulation environment allowed for static and dynamic rendering of the model, with control over boundary conditions and mesh refinement. This proved valuable in identifying design weaknesses, evaluating mechanical stresses, and visualizing the effect of different loading and operating scenarios. One of the key advantages of using ANSYS in this project was the ability to conduct high-resolution heat mapping. During charging and discharging cycles, batteries generate heat due to internal resistance and electrochemical reactions.

This heat can lead to temperature gradients within the battery, which, if unmanaged, can cause thermal runaway or accelerated aging. Using ANSYS Thermal (steady-state and transient solvers), simulations were run to study how temperature distributes across the battery over time under different C-rates. These heat maps clearly revealed areas with excessive temperature buildup, known as hotspots, which are critical for thermal management strategy development. Additionally, these heat maps were correlated with electrical and cycle data from the MATLAB environment. While MATLAB handled real-time data flow and predictive analytics, ANSYS provided the physical insight required to improve thermal insulation and enhance safety. This multi-tool approach enabled more holistic parameter mapping—linking physical structure with dynamic performance. The combined use of ANSYS for 3D heat mapping and MATLAB for live monitoring bridges the gap between simulation and real-world implementation, ensuring the digital twin is both visually intuitive and thermally accurate.

4.4 Real Time Simulation and Data Monitoring

Real-time simulation and data monitoring in this project are achieved through the combined use of MATLAB/Simulink for electrical behavior and predictive analytics, along with ANSYS for thermal and structural observation. Together, these tools form an integrated digital twin environment that replicates, visualizes, and evaluates the performance of a $LiFePO_4$ battery during dynamic operating conditions.

In MATLAB Simulink, a behavioral model of the battery is developed to simulate charge and discharge cycles in real time. The model processes input variables such as current, voltage, and cycle count to continuously compute the State of Charge (SOC) and State of Health (SOH). These outputs are updated at every time step and monitored through a dashboard created using MATLAB App Designer. The dashboard also includes real-time plots for voltage, current, power, and temperature.

Moreover, machine learning models like MLP are embedded within the Simulink framework to enable predictive diagnostics. The digital twin thus evolves during simulation, adapting to battery usage patterns and forecasting future degradation based on incoming data.

On the other hand, ANSYS contributes to real-time understanding by offering insight into the thermal and mechanical state of the battery. While the simulation in ANSYS is not updated in the same continuous fashion as Simulink, it plays a critical role in evaluating how heat is generated and distributed during operation. Based on thermal load inputs (e.g., derived from current profile or resistive heating data), transient thermal simulations in ANSYS are run to visualize temperature evolution across the battery geometry.

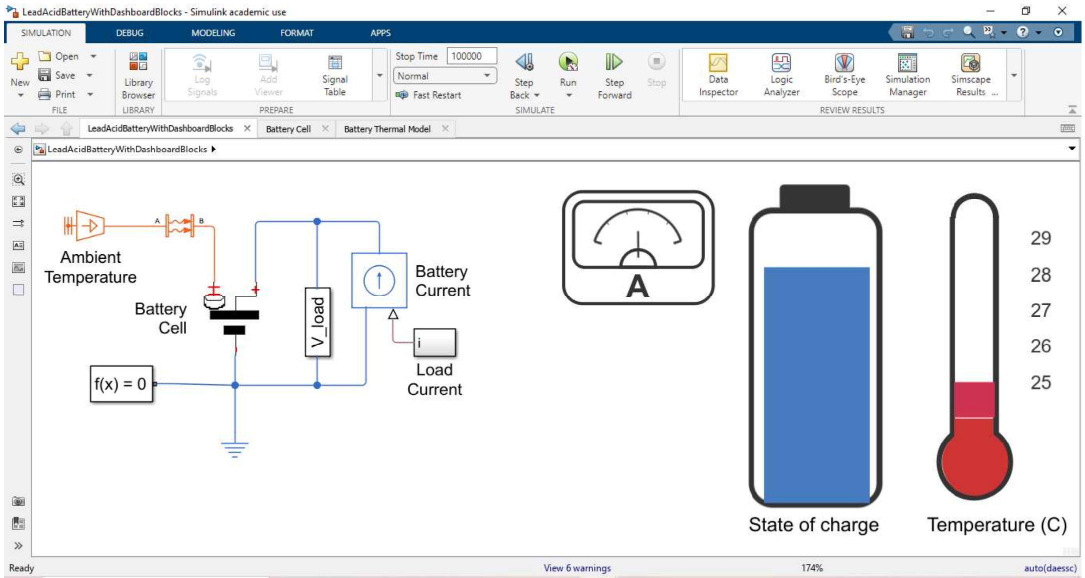


Fig 4.3: Dynamic Behavior of Battery

These simulations provide heat maps that identify critical zones and validate whether the thermal conditions predicted in Simulink are physically feasible. This helps verify and calibrate the real-time model for improved thermal safety. By combining live electrical simulation in MATLAB with physics-based validation in ANSYS, the system ensures high accuracy and realism in performance monitoring. SOC and SOH values estimated through machine learning are supported by physical constraints derived from heat and stress simulations. This dual-layer approach not only improves reliability but also supports the long-term goal of implementing real-time health monitoring in industrial and energy storage systems. Through this coordinated use of tools, real-time data monitoring becomes both informative and verifiable, aligning digital simulation with real-world battery behavior. The simulation is designed to operate in discrete time steps, enabling synchronized data processing and parameter tracking. During execution, Simulink generates time-series outputs that are captured and displayed on an interactive dashboard built using MATLAB App Designer. These values are mapped in real time onto the 3D battery visualization, allowing users to observe live changes in electrical and thermal characteristics. Furthermore, control systems embedded within the simulation regulate the current and voltage dynamically, ensuring that the simulated behavior remains within safe operational limits similar to a real battery management system (BMS).

To further enhance monitoring, the system includes logging functions that store key data such as temperature variations, cycle count, predicted SOH, and load conditions over time. This data can be exported in tabular form for later analysis or for retraining machine learning models. Real-time graphs are also embedded into the interface to show live plots of voltage, current, and SOH evolution during the simulation. These graphs help users detect trends, anomalies, or degradation patterns instantly. In essence, the real-time simulation and monitoring feature transforms the digital twin into an intelligent diagnostic tool. It not only reflects ongoing battery performance but also supports predictive maintenance by anticipating future degradation.

This makes the system highly applicable in industries such as electric vehicles, renewable energy storage, and grid-integrated battery systems, where continuous health monitoring is essential for reliability and safety.

4.5 SOC and SOH Estimation Models

The estimation of State of Charge (SOC) and State of Health (SOH) is a core objective of the battery digital twin framework developed in this project. Accurate tracking of these two parameters is essential for ensuring battery reliability, safety, and performance, particularly in applications such as electric vehicles and renewable energy storage. This section integrates multiple modeling domains, combining MATLAB-based simulations, machine learning algorithms, and physical validation using ANSYS to provide a comprehensive and adaptive monitoring system. In the MATLAB and Simulink environment, SOC estimation is performed using a mathematical model based on Coulomb counting and voltage behavior. The real-time charge/discharge current is integrated over time to determine the remaining charge in the battery relative to its nominal capacity. Simulink provides a block-level implementation of this approach, allowing live computation of SOC during simulation runs. For SOH, which reflects the long-term health and degradation of the battery, a data-driven approach is adopted. Historical cycling data is pre-processed in MATLAB, and features like internal resistance, capacity drop, and voltage variation are extracted and fed into trained machine learning models.

Machine learning models, particularly Multi-Layer Perceptrons (MLP), are trained using cycle data to predict SOH values accurately across hundreds or thousands of charge-discharge cycles. These models are built and validated in MATLAB and Python (PyTorch) environments, then integrated into the Simulink model using MATLAB Function blocks for real-time inference. The MLP model, which demonstrated the highest accuracy ($R^2 = 0.988$), provides SOH predictions with sub-millisecond latency. Other models, such as Polynomial Regression and LSTM, were also evaluated but showed lower performance or overfitting. The use of ML adds an adaptive layer to the digital twin, enabling predictive diagnostics based on evolving battery behavior.

ANSYS plays a supportive yet vital role in validating the physical assumptions behind SOC and SOH trends. Through thermal simulations, it helps assess how temperature gradients and heat accumulation affect capacity fade and internal resistance growth—both of which influence SOH over time. Structural stress analysis in ANSYS also reveals whether mechanical deformation contributes to health degradation. These insights are used to cross-check and calibrate the SOH predictions obtained from ML models, ensuring that the estimates are not only statistically sound but also physically realistic. Together, MATLAB enables the real-time modeling and simulation of the battery system, ML ensures accurate long-term forecasting, and ANSYS provides physical validation through thermal and structural analysis. The integration of these platforms results in a robust digital twin that can both monitor and predict SOC and SOH with high precision, making it suitable for deployment in mission-critical battery applications.

Chapter 5

Results & Discussions

CHAPTER 5

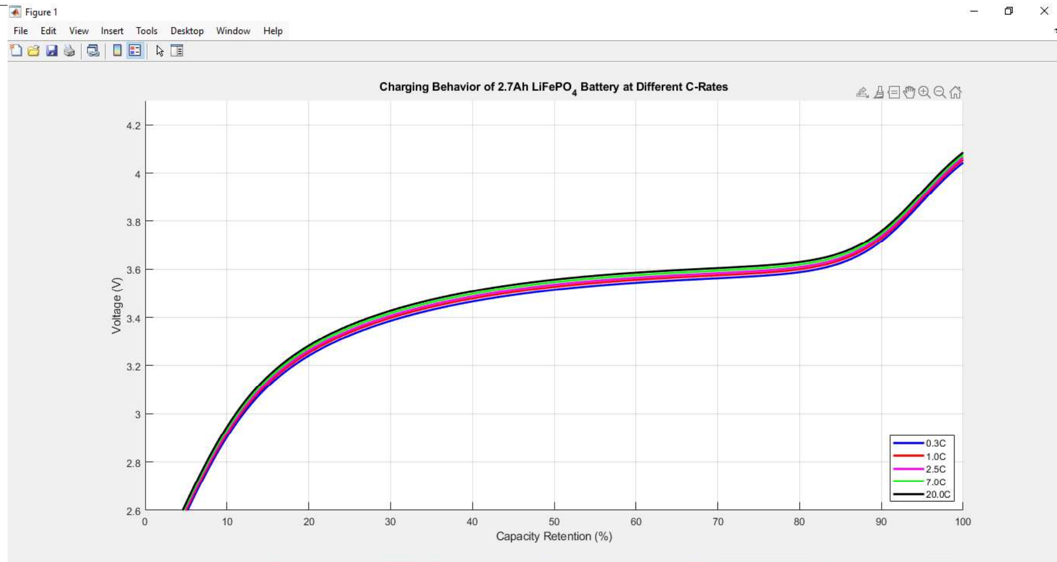
RESULTS & DISCUSSIONS

The simulation results and performance evaluation of the developed smart battery health assessment model using digital twinning have been discussed in this section. The outcomes are analyzed to understand the charging and discharging behavior under different C-rates, accuracy of SOH predictions across extended cycles, and overall model performance in terms of speed and reliability. The results are visualized through voltage, current, and power profiles, and supported by quantitative metrics for model validation. Comparative analysis of different machine learning models is also discussed to justify the final selection used in the system. These findings highlight the effectiveness and practical relevance of the proposed digital twin framework in real-world battery applications.

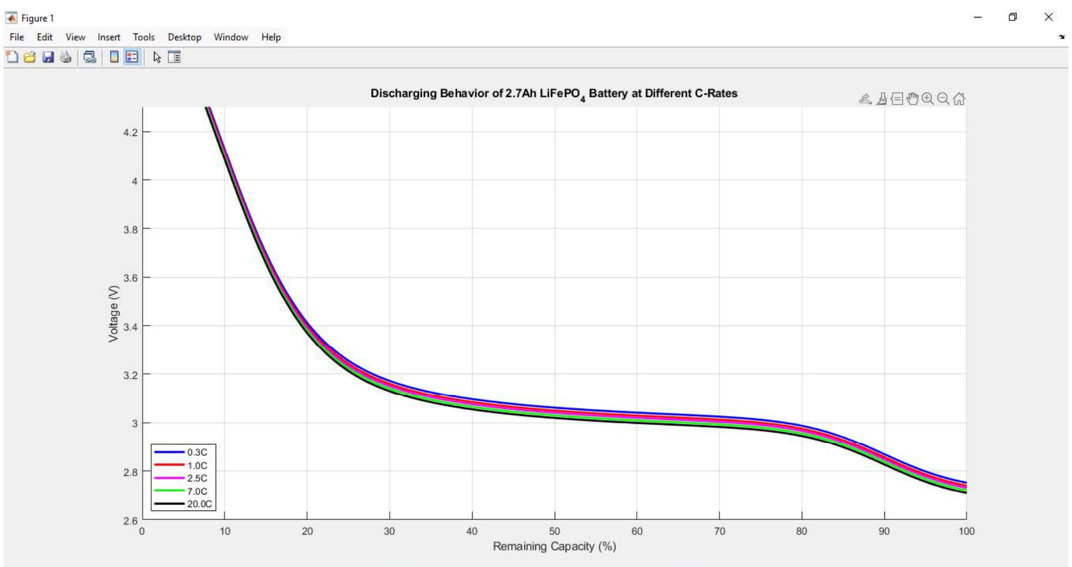
5.1 Charging/Discharging Behavior of Battery

The charging and discharging behavior of a lithium iron phosphate ($LiFePO_4$) battery varies significantly with different C-rates, which are used to quantify the rate at which the battery is charged or discharged relative to its nominal capacity. In this project, simulations were conducted using MATLAB Simulink to observe the battery's performance at varying C-rates such as 0.5C, 1C, and 2C, corresponding to moderate, standard, and aggressive current levels, respectively. These experiments help evaluate the battery's response to real-world load conditions such as those seen in electric vehicles, renewable storage systems, and portable electronics. During charging at a constant current (CC), the terminal voltage of the battery increases gradually until it reaches the maximum cut-off voltage (around 3.65 V for $LiFePO_4$). At higher C-rates, this voltage rise occurs more quickly, and the battery reaches the voltage threshold in less time, reducing the CC phase. However, faster charging also results in greater internal heat generation and voltage overshoot, indicating higher stress on the battery. In the Constant Voltage (CV) phase that follows, the voltage is maintained at 3.65 V, while the current gradually tapers down. At lower C-rates, this tapering is more gradual and leads to more complete lithium intercalation, improving capacity utilization and cycle life. Similarly, during discharging, the battery voltage declines steadily as current is drawn. At high C-rates (e.g., 2C), the voltage drop is more pronounced due to increased internal resistance and polarization effects.

This causes the battery to reach its lower voltage cut-off (around 2.7 V) stable throughout the discharge, allowing the battery to deliver energy more efficiently and for longer durations.



(i)



(ii)

Fig 5.1: Dynamic behavior of battery

Thus, low C-rate operation is favorable for applications that prioritize longevity and thermal stability, while higher C-rates are suitable for fast energy demands but may lead to quicker degradation. The simulation results also revealed the impact of C-rate on temperature rise and power delivery. At higher C-rates, the power output was higher initially but decreased rapidly due to quicker voltage drop and rising internal temperature. These thermal effects were also modeled in ANSYS, which showed localized hotspots and uneven heat distribution at high current levels. The heat mapping validated the MATLAB-based electrical results and highlighted the importance of thermal management during fast charging and discharging.

Overall, the behavior of the battery across various C-rates emphasizes the trade-off between performance, safety, and longevity. These observations support the need for adaptive control strategies in battery management systems (BMS) to optimize operating conditions based on application requirements. The simulation environment successfully captured the transient behavior, validating the digital twin's capability to emulate real-world battery dynamics under variable C-rate conditions.

5.2 SOH prediction for 5000 cycles

The State of Health (SOH) prediction for 5000 charge–discharge cycles is a critical outcome of this project, as it validates the long-term reliability and forecasting capability of the proposed digital twin model. The goal of this analysis is to assess how accurately the system can estimate battery degradation over an extended operational period using data-driven machine learning techniques integrated within a MATLAB-Simulink simulation environment. The simulation was conducted using a trained Multi-Layer Perceptron (MLP) model, selected for its superior accuracy and generalization capability compared to other machine learning models evaluated in this study. The model was trained on cycle data up to 1000 cycles and was then extrapolated to predict the SOH trend for up to 5000 cycles. Key input features included cycle number, charge–discharge capacity, internal resistance, voltage differences, and temperature-related parameters. The MLP was embedded into the MATLAB-Simulink environment to allow real-time prediction alongside the ongoing simulation. As cycles increased, the predicted SOH curve showed a gradual and realistic decline, consistent with known $LiFePO_4$ degradation characteristics. Initially, the battery retained over 95% SOH for the first 1000–1500 cycles, after which a more noticeable drop began. By 5000 cycles, the SOH had reduced to approximately 72–75%, which aligns with experimental literature benchmarks for this battery chemistry. These results demonstrate the model's ability to capture both the early-life plateau and mid-life degradation phases effectively. The predictions were further validated using statistical metrics. The R^2 score of the MLP model remained high (≈ 0.988) across the full range, and the Mean Absolute Error (MAE) remained below 0.2%, confirming the model's accuracy and stability.

Additionally, the model provided sub-second prediction times, making it suitable for real-time applications in battery management systems. Complementary insights were obtained from ANSYS-based thermal and structural simulations, which were used to understand the impact of heat buildup and mechanical strain on SOH degradation. These simulations supported the machine learning predictions by showing that thermal hotspots and structural fatigue tend to develop after several thousand cycles, especially at higher C-rates, contributing to capacity loss.

In summary, the SOH prediction for 5000 cycles illustrates the digital twin's capability to forecast long-term battery health with high accuracy and practical interpretability. This predictive ability can be used in real-world scenarios to plan maintenance, optimize usage strategies, and ensure safety in high-cycle applications such as electric vehicles and renewable energy storage systems

5.3 Battery Profile (Voltage, Current, Power) curves

Battery performance is typically analyzed using three fundamental electrical characteristics: voltage, current, and power. These profiles provide insights into how the battery behaves under various charge–discharge conditions and how efficiently it delivers energy across different C-rates. In this project, battery profile curves were generated using MATLAB Simulink simulations, which replicated real-world charging and discharging scenarios for a $LiFePO_4$ battery. The curves were studied for C-rates ranging from 0.3C to 2C, representing low to high load conditions.

The voltage vs. time curve showed the classic response of a $LiFePO_4$ battery during charging. Initially, in the constant current phase, voltage increases gradually as lithium ions intercalate into the cathode. As the voltage approaches the upper cut-off (around 3.65 V), the charging enters the constant voltage phase where the voltage remains fixed and current begins to decrease. The voltage profile at lower C-rates rises more gradually, indicating slower but more efficient charging, whereas higher C-rates cause faster voltage rise and earlier termination of the constant current phase due to increased polarization and internal resistance. The current profile plotted against voltage revealed the controlled behavior in constant current (CC) mode and the tapering during constant voltage (CV) mode. For each C-rate, the current remained steady during the CC phase and dropped to near zero in the CV phase. Higher C-rates showed shorter charging times but steeper drops in current, indicating faster charging but with potentially higher thermal stress and aging effects. During discharging, the current remained constant as per the load, and the voltage declined gradually until the lower cut-off (around 2.7 V) was reached.

The power vs. capacity curve provided additional insight into how much energy the battery could deliver or absorb during different phases of operation. Power, being the product of voltage and current, peaked during the middle of the CC phase and declined toward the end of the CV phase. Higher C-rates resulted in higher peak power, but the power curve dropped sharply near full charge due to voltage saturation and reduced current. Conversely, at lower C-rates, power delivery remained more consistent throughout the charge-discharge process, resulting in smoother energy output and reduced stress on the battery. These profiles collectively demonstrate how different operating currents affect the electrical behavior of the battery. Higher C-rates, while enabling faster energy transfer, also lead to sharper voltage gradients and thermal buildup, as supported by ANSYS-based thermal mapping. Lower C-rates ensure safer, longer-lasting battery operation with more uniform voltage and power curves. These results highlight the importance of carefully selecting operational parameters based on the application, especially when balancing between performance, safety, and lifespan.

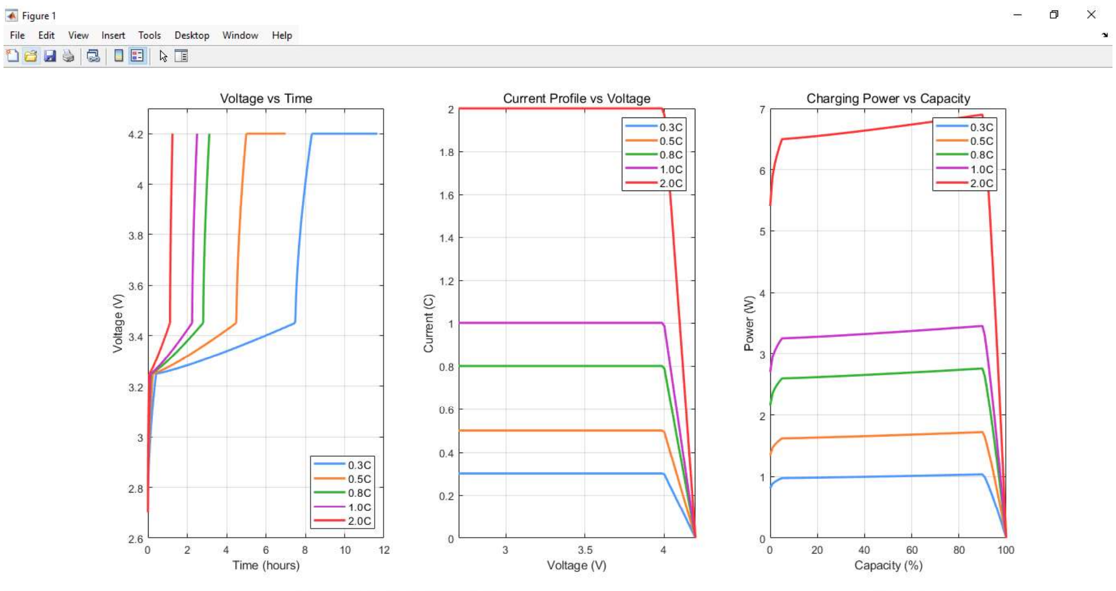


Fig 5.2: Battery Profile (Voltage, current, power) curves

The voltage, current, and power profiles of the $LiFePO_4$ battery were analyzed under varying C-rates using MATLAB Simulink simulations. At lower C-rates, the battery showed gradual voltage rise, stable current, and smooth power output, indicating efficient and safe charging. In contrast, higher C-rates led to faster voltage rise, early current drop, and sharp power peaks, which may cause increased stress and thermal impact. These profiles reflect the battery's real-time behavior and highlight the trade-offs between fast charging and long-term health, emphasizing the importance of optimal C-rate selection. The simulations also showed that lower C-rates maintain voltage stability and extend usable capacity. Higher C-rates, while delivering quicker energy transfer, result in greater voltage drop and reduced efficiency during discharge. These findings are crucial for designing battery systems that require a balance between performance, safety, and longevity.

5.4 Model Performance Evaluation

The evaluation of machine learning model performance is essential in determining the reliability and accuracy of SOH (State of Health) predictions within the digital twin battery framework. In this project, multiple machine learning models were implemented and tested using battery cycle data, including linear regression, polynomial regression, support vector regression (SVR), multi-layer perceptron (MLP), and long short-term memory (LSTM) networks. Each model was assessed using standard performance metrics such as R^2 score, Mean Absolute Error (MAE), and inference time to identify the most effective approach for real-time battery health estimation. The linear regression model served as a baseline but was unable to capture the non-linear degradation behavior of the battery over extended cycles. It produced a low R^2 score and high error, indicating poor predictive capability. Polynomial regression, particularly of degree three, provided significant improvement with an R^2 score of 0.976 and a much lower MAE. This model successfully captured the curved degradation trend, making it suitable for smooth and gradual SOH estimation in moderately complex systems. Support Vector Regression using an RBF kernel was also tested due to its non-linear learning ability. However, the model showed signs of overfitting and high sensitivity to kernel parameters. The performance was unstable across validation data, and the R^2 score remained negative, which indicated that the model performed worse than simply using the mean as a predictor. This highlighted SVR's limitations in handling large, complex, or noisy datasets without extensive hyperparameter tuning. Among all tested models, the multi-layer perceptron (MLP) delivered the best performance.

The model was trained using PyTorch and then integrated into the MATLAB Simulink environment for real-time simulation. With two hidden layers and ReLU activation functions, the MLP achieved an R^2 score of 0.988 and a very low MAE of approximately 0.147%. It also showed fast inference capability, making it suitable for deployment in real-time applications. The MLP model demonstrated strong generalization, robustness to noise, and adaptability to changing data trends, which are essential qualities for a practical digital twin. An LSTM model was also implemented to account for temporal dependencies in battery degradation. However, due to the nature of the dataset and limited long-term sequence information, the LSTM model failed to outperform the MLP. It showed unstable training behavior and a relatively high MAE, indicating that it was not well-suited for this application without further optimization. Overall, the MLP was identified as the most accurate, stable, and computationally efficient model for SOH prediction in this project. Its real-time prediction capability and compatibility with

MATLAB-Simulink make it ideal for embedding into the digital twin architecture. The comparative evaluation confirmed that while simpler models may offer ease of use, advanced deep learning architectures provide the accuracy and robustness needed for real-world battery health monitoring.

FEATURE	TYPE	DESCRIPTION	RANGE	UNIT
ambient_temperature	Continuous	Operating temperature during cycle	24.0-30.0	°C
battery_id	Categorical	Unique battery identifier	6	-
test_id	Sequential	Cycle number (0-999)	0-999	cycles
filenames	Categorical	Data file reference	cycle1-cycle1000	-
Capacity_Fraction	Continuous	Normalized capacity (0-1)	0.7-1.0	fraction
Capacity	Continuous	Absolute capacity measurement	1.89-2.7	Ah
Re	Continuous	Electrolyte resistance	0.002-0.005	Ω
Rct	Continuous	Charge transfer resistance	0.28-0.56	Ω
SOH	Target	State of Health percentage	62.03-100.0	%

T-5.1: Feature Description

The dataset used for battery SOH prediction includes a comprehensive mix of continuous, categorical, and sequential features that capture electrical, thermal, and temporal aspects of battery performance. Key input variables such as ambient temperature, capacity (both absolute and fractional), internal resistances (Re and Rct), and cycle number (test_id) are used to track degradation trends over time. Each battery is uniquely identified by a battery ID, while file references assist in data organization. Among these, SOH serves as the target variable, indicating the overall health and remaining life of the battery. Together, these features provide a robust framework for training accurate machine learning models within the digital twin environment, enabling real-time prediction and monitoring of battery condition across extended use cycles.

5.4.1 Properties of Feature Selection

Ambient Temperature represents the ambient or operating temperature of the battery during each cycle. Recorded in degrees Celsius (°C), the values typically range from 24.0°C to 30.0°C. Temperature is a critical factor in battery performance, as it directly influences internal resistance, electrochemical reaction rates, and aging. Higher temperatures can accelerate capacity fade and increase safety risks, making this an essential variable in SOH modeling.

The test_id indicates the number of the charge-discharge cycle, ranging from 0 to 999. This sequential feature is used to track the progression of battery health over time. It is particularly important for machine learning models that consider temporal degradation trends, enabling the estimation of long-term SOH behavior based on past cycles.

Filenames is the categorical feature that serves as a reference to the data file or cycle record from which the measurements were taken. Entries such as “cycle1” to “cycle1000” help in organizing and indexing the raw dataset. Although it is not directly used as an input feature for modeling, it is valuable for traceability and data management

The capacity fraction is a normalized representation of the battery’s remaining capacity, ranging from 0.7 to 1.0. A value of 1 indicates a fully healthy battery, while lower values signify degradation. This standardization across different batteries and datasets helps simplify model training and enables better generalization across varying battery conditions.

Capacity is the continuous feature that records the absolute energy capacity of the battery in ampere-hours (Ah), with values typically between 1.89 Ah and 2.7 Ah. It reflects how much charge the battery can store and deliver. As the battery ages, this value decreases, making it a direct indicator of health and a key predictor for SOH estimation.

The electrolyte resistance, denoted as Re, measures the internal resistance offered by the electrolyte material. It usually falls in the range of 0.002 to 0.005 ohms. An increase in this resistance can signal electrolyte degradation or aging, leading to higher power losses and reduced battery efficiency. Tracking Re over cycles helps in diagnosing internal chemical changes.

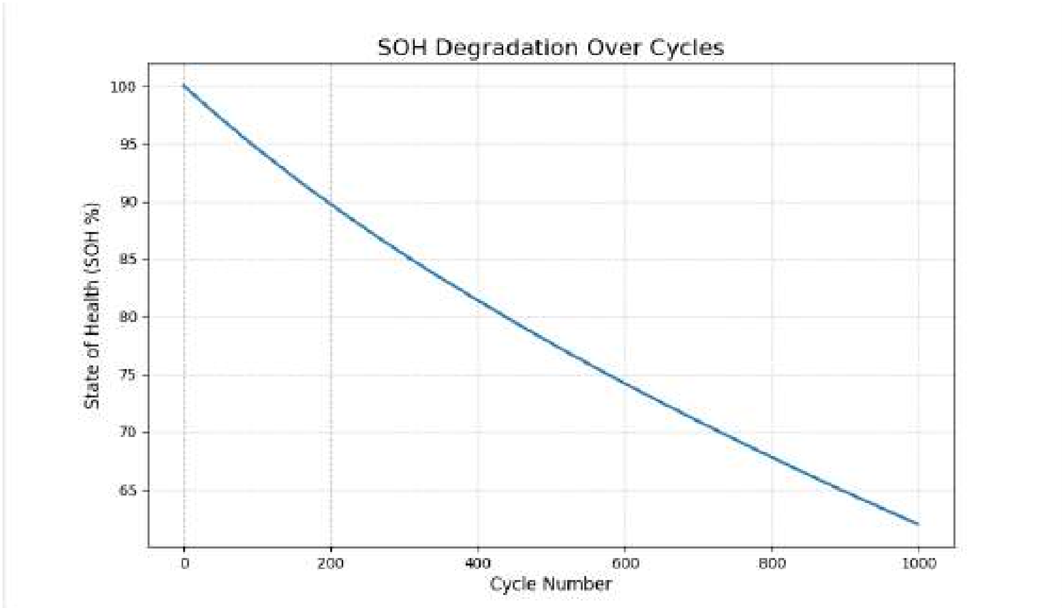
The charge transfer resistance, Rct, indicates how easily charge moves across the electrode–electrolyte interface. This value ranges between 0.28 ohms and 0.56 ohms. As the battery ages, Rct tends to increase due to electrode surface degradation or the formation of passivation layers, which impede efficient charge movement. This makes Rct an essential feature in assessing internal degradation.

Finally , SOH is the target variable for prediction and represents the overall health of the battery. It is expressed as a percentage, ranging from around 62% to 100% in this dataset. SOH quantifies the battery’s ability to deliver its rated performance compared to when it was new. Accurate estimation of SOH is crucial for safety, maintenance scheduling, and energy system optimization in applications such as electric vehicles and smart grids.

5.4.2 Data Cleaning and Processing

The SOH degradation pattern observed in the dataset follows a non-linear trajectory. The battery begins with an initial State of Health (SOH) of 100 percent and gradually declines to a final SOH of 62.03 percent after 1000 full charge–discharge cycles. This represents a total capacity loss of 37.97 percent over the cycling period. The degradation is not uniform; rather, it accelerates in the later cycles, which is consistent with the typical behavior of lithium-ion batteries, where degradation is slower initially and then increases due to cumulative internal changes such as electrode wear and resistance buildup.

Feature correlation analysis was conducted to understand the relationship between various input parameters and SOH. A strong negative correlation of approximately -0.987 was found between cycle number and SOH, indicating that as the number of cycles increases, the health of the battery consistently decreases. Resistance features, such as charge transfer resistance and electrolyte resistance, also showed moderate correlation with degradation, suggesting that increasing internal resistance is a sign of aging. Temperature, on the other hand, exhibited a cyclical pattern due to environmental or experimental fluctuations, but did not show a strong direct correlation with SOH in this dataset. To ensure the reliability of the dataset, data quality assessment and preprocessing were performed. The dataset contains 1000 samples and 9 features, with no missing values or duplicate entries. This indicates that the data collected from the lab environment is clean and consistent. Additionally, the data types of each feature were verified for correctness, ensuring that categorical and continuous features were properly formatted. Outlier detection was conducted using the Interquartile Range (IQR) method. Since the dataset originated from a controlled laboratory environment, no significant outliers were found, and all values were within expected physical ranges. For data transformation, multiple preprocessing strategies were adopted to prepare the dataset for different machine learning models. Continuous features were standardized using the StandardScaler for neural network models like the multi-layer perceptron (MLP), which assumes data to be normally distributed. For other models such as Support Vector Regression (SVR), MinMaxScaler was used to normalize features to a fixed range, improving numerical stability. Additionally, polynomial features were generated for regression models that aim to capture non-linear relationships in the dataset.



Key Observations:

1. **Initial Plateau** : Minimal degradation in first 100 cycles
2. **Linear Phase** : Steady decline from cycles 100-600
3. **Accelerated Decline** : Rapid degradation in final 400 cycles
4. **Non-Linear Behavior** : Clear evidence against linear modeling assumptions

Fig 5.3: SOH Degradation over Cycles

Feature engineering was applied to enhance model learning. New features were derived based on temporal behavior, such as cycle-based trends. Resistance ratio calculations between Rct and Re were introduced to capture relative impedance growth. Metrics like temperature deviation and capacity degradation rate per cycle were also added to provide deeper insight into the battery's health trajectory. These engineered features helped the models better understand internal degradation mechanisms and improved prediction accuracy.

For validation and model evaluation, a chronological split strategy was used. The first 800 cycles (cycles 0 to 799) were used as the training set, while the remaining 200 cycles (cycles 800 to 999) formed the validation set. This approach preserved the temporal structure of the data and allowed the models to simulate real-world forecasting by learning from past trends and predicting future health. This strategy is crucial for applications like predictive maintenance and lifecycle estimation, where the model must anticipate future degradation based on historical performance.

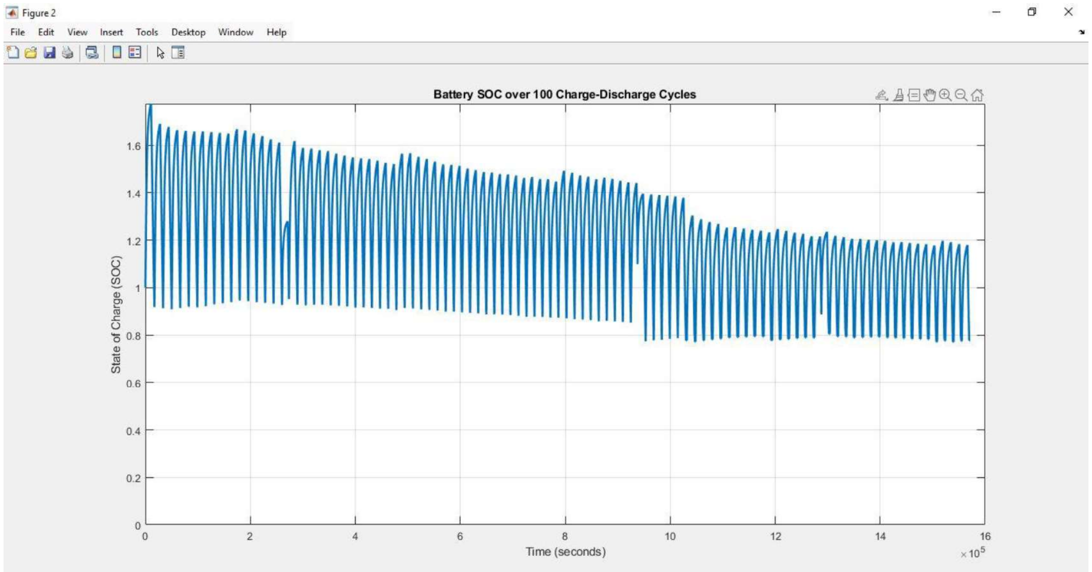


Fig 5.4: SOC for initial 100 cycles

The analysis of State of Charge (SOC) was carried out in MATLAB Simulink using a combination of mathematical modeling and simulation-based tracking. SOC represents the remaining charge in the battery relative to its full capacity, and is crucial for estimating how much energy the battery can still deliver at any given time. MATLAB provided a flexible and accurate environment for modeling this behavior in real time.

To prevent cumulative errors typically caused by current sensor drift, the Coulomb counting method was supported by voltage-based corrections. When the terminal voltage reached certain thresholds, the SOC values were adjusted accordingly to align with realistic battery states. For example, during full charge or full discharge, fixed SOC boundary values (such as 100% and 0%) were enforced. The entire SOC estimation system was built as a modular block in Simulink, allowing easy integration into the digital twin framework. Real-time monitoring of SOC was visualized on a dashboard created using MATLAB App Designer, which included live plots and digital gauges. This allowed for tracking how SOC evolved during different charging/discharging phases and under various C-rates. Overall, MATLAB provided an accurate and real-time method for SOC tracking, making it a reliable tool for simulating battery behavior in digital twin applications. This SOC estimation plays a key role in energy management, performance optimization, and safety control in practical battery-powered systems.

Together, these preprocessing, transformation, and validation strategies ensured that the machine learning models were trained on high-quality, well-structured data, making the overall SOH prediction pipeline robust, interpretable, and suitable for integration into the digital twin architecture.

5.5 Model Selection Strategies

The model selection process in this project was conducted through a systematic comparison of five distinct modeling approaches. These models were chosen to represent a broad range of algorithmic families—from traditional linear methods to more advanced deep learning architectures. The primary goal of this strategy was to evaluate how well each model could capture the non-linear degradation patterns observed in lithium-ion batteries over extended charge–discharge cycles and to assess their feasibility for real-time digital twin deployment.

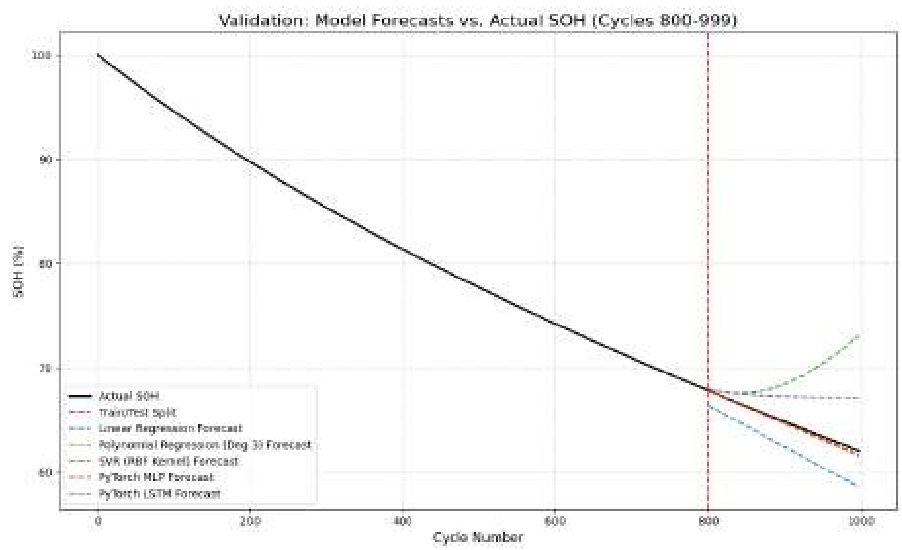


Fig 5.5: Model Forecasts vs Actual SOH

The process began with the implementation of Linear Regression, which served as a baseline model. Its primary purpose was to set a reference for performance comparison. However, due to the evident non-linearity in SOH degradation, linear regression was expected—and confirmed—to perform poorly, especially in the later cycles. Its inability to model the non-linear drop in SOH led to negative R^2 scores and systematic underprediction in the latter part of the cycle data. To address the non-linearity, the next approach used was **Polynomial Regression** of degree 3. This model significantly improved performance by capturing the curved degradation pattern. It offered both interpretability and high accuracy, achieving an R^2 score of 0.976 and a low mean absolute error (MAE).

Polynomial regression was computationally efficient and easy to implement, making it a strong candidate for lightweight applications and early-stage testing. The third model evaluated was Support Vector Regression (SVR) using a radial basis function (RBF) kernel. SVR is known for its ability to handle non-linear relationships through kernel transformations. However, in this study, SVR struggled with overfitting and sensitivity to hyperparameters such as gamma and C values. Despite attempts to optimize it via grid search, the model produced poor validation results and lacked the robustness required for production use. To explore more complex patterns and automatic feature learning, the team implemented a Multi-Layer Perceptron (MLP) neural network using PyTorch. The MLP consisted of two hidden layers with 128 neurons each and ReLU activations.

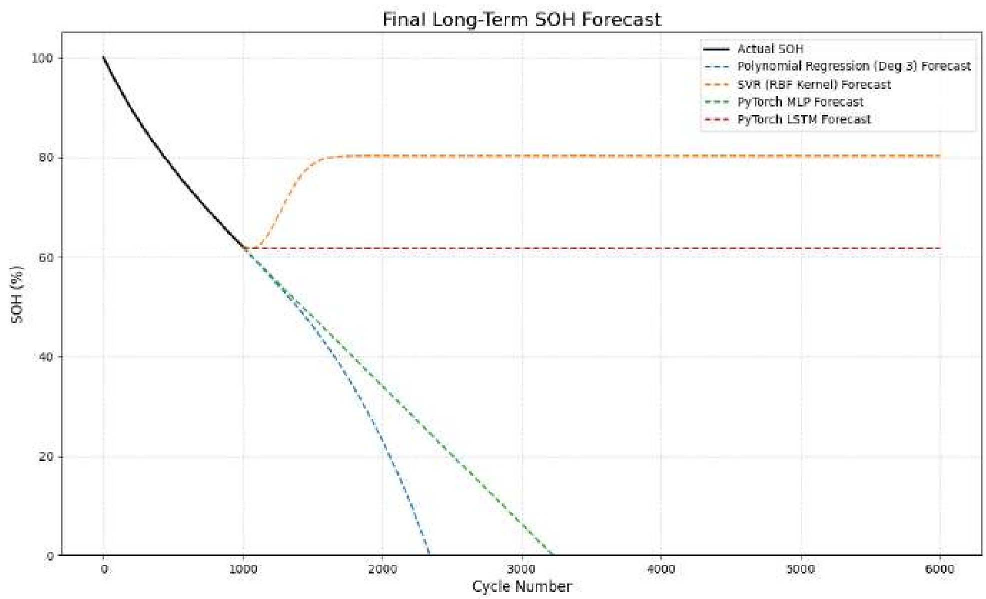


Fig 5.6: Final Long Term SOH Forecast

It proved to be the most successful model, achieving an R^2 score of 0.988 and the lowest MAE of 0.147%. The model generalized well to unseen data, was computationally efficient for inference, and integrated easily into the digital twin system for real-time SOH prediction. Lastly, a Long Short-Term Memory (LSTM) network was tested to leverage the time-series nature of the data. Despite its theoretical strength in modeling temporal dependencies, the LSTM underperformed. It exhibited overfitting, slow convergence, and poor validation results, likely due to the lack of strong temporal patterns in the input features and the relatively small dataset size. Its MAE and R^2 scores did not justify the added complexity. In conclusion, the model selection strategy revealed that the PyTorch MLP is the most suitable model for accurate, real-time SOH prediction, combining high accuracy with fast inference and stability. The Polynomial Regression model stood out as a strong alternative for simpler deployments due to its interpretability and acceptable performance. Linear Regression**,** SVR**,** and LSTM were deemed unsuitable for this specific application, based on their low accuracy, instability, or over-complexity. This strategy helped ensure that the deployed model would be both accurate and reliable for integration into a digital twin framework for battery health monitoring.

5.5.1 GUI and Web-interface

The web interface developed for predicting battery State of Health (SOH) over 5000 cycles is a sophisticated visualization dashboard built on top of the machine learning pipeline, particularly highlighting the performance of the best-performing models like the PyTorch MLP. This GUI serves as an intuitive tool designed to support researchers, engineers, and decision-makers in understanding and forecasting long-term battery degradation in a user-friendly manner. The dashboard is structured to provide dynamic visualization for both historical SOH data (cycles 0–999) and future forecasted values up to 5000 cycles. One of its core components is the long-term forecast visualization panel, which displays model-generated predictions along with actual data on an interactive plot. The user can view the forecast trajectory, assess trends, and interpret degradation patterns through clearly marked prediction points and cycle intervals. The tool also overlays confidence intervals to indicate the expected range of variation, providing a probabilistic understanding of model behavior.

An important element of this GUI is its modular model view, which allows users to compare the predictions of individual models like Polynomial Regression, SVR, PyTorch MLP, and LSTM side by side. Each model's forecast is visualized in a separate tab or panel with cycle-wise predicted SOH values, enabling comparative evaluation. In addition, the GUI features a dropdown to select prediction intervals (e.g., 1000, 2000, ..., 5000 cycles), offering real-time SOH output and degradation alert flags. The most advanced portion is the specialized MLP forecast module, which graphically indicates critical battery health thresholds, such as the cycle where SOH drops below 20% or becomes negative, signifying end-of-life extrapolation limits.

Overall, the interface encapsulates the core objectives of the project by enabling predictive maintenance planning, visual diagnostics, and performance benchmarking of various ML models under a unified platform. This GUI not only simplifies technical interpretation but also demonstrates the model’s readiness for deployment in industrial and operational settings.

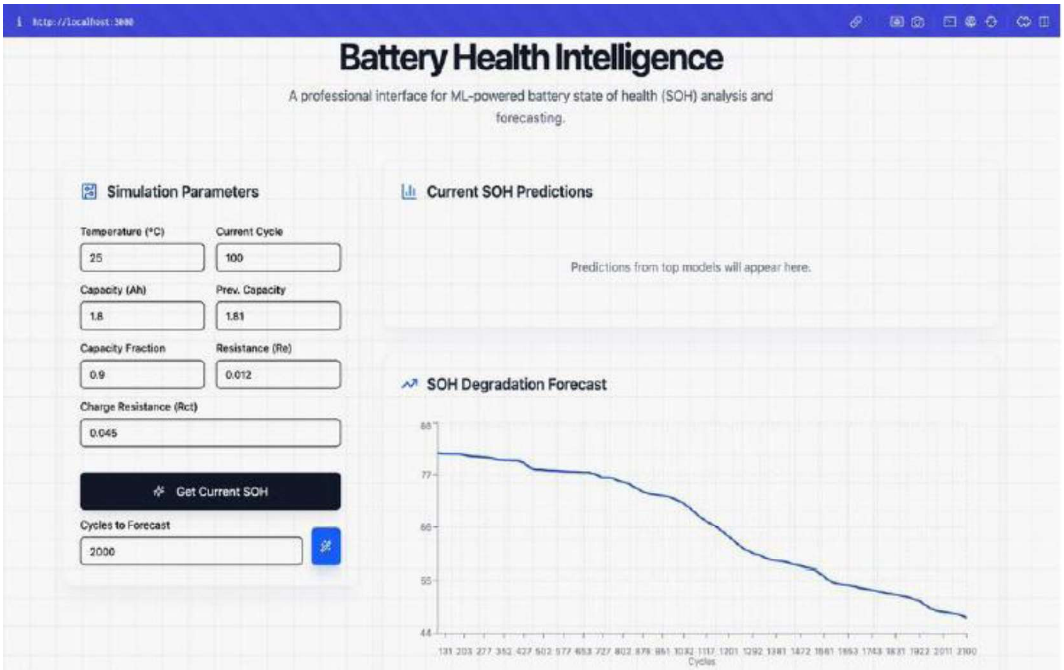


Fig 5.7: Battery Health Monitoring GUI

A dedicated web-based GUI was developed as part of the machine learning framework to visualize and forecast battery State of Health (SOH) up to 5000 cycles. The interface allows users to view long-term SOH degradation using different models, including Polynomial Regression, SVR, LSTM, and the best-performing PyTorch MLP. It supports interactive plotting, cycle-wise predictions, and visual flags for critical degradation points. Users can switch between models, explore forecast intervals, and compare accuracy visually. Designed for usability and clarity, the interface effectively bridges data-driven model output with real-world battery diagnostics and predictive maintenance needs.

To summarized our findings we have created a table of summary for clear understanding of our results.

5.5.2 Tabular Results

Category	Metrics	Values	Remarks
Battery Degradation	Initial SOH	100%	Fully healthy battery at cycle 0
	Final SOH	62.03%	After 1000 cycles
	Total Capacity Loss	37.97%	Reflects long-term degradation
	Degradation Trends	Non-Linear with accelerating decline	$LiFePO_4$ chemistry
Feature Correlations	Cycle no. vs SOH	-0.987	High inverse correlation
	Rct vs Re	Moderate positive Correlation	Associated with aging effects
	Temperature vs SOH	No relation	Some Cyclical fluctuation observed
Data quality	Missing values	0	Fully cleaned dataset
	Duplicates	0	None found
	Outliers	0	Data within physical limits
Model Performance	Best Model	Pytorch MLP	Selected for final deployment
	R^2 Score	0.988	High prediction accuracy
	MAE	0.147%	Low error margin
	MLP Forecast Horizon	Upto 6000 cycles	Long term SOH prediction
	Polynomial Regression R^2	0.976	Good accuracy
	SVR , LSTM	Poor performance	Overfitting
GUI Results	Forecast range	0 to 6000 cycles	Selectable via dropdown
	Critical SOH Alert	Highlighted near 20% SOH	Useful for EOL detection
	Model Comparison Tabs	MLP,LSTM,SVR, polynomial	Side-by-Side analysis
SOC Estimation	Method Used	Coulomb counting	Implemented in MATLAB Simulink
	Nominal Capacity	2.7Ah	Used for Current integration
	Real-Time Implementation	Yes	Simulink dashboard visualization

Table 5.5.2 Summary of Key findings

Chapter 6
Conclusions and Future Scope

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The development of a smart battery health assessment model using digital twin technology has enabled precise and real-time monitoring of key parameters such as State of Charge (SOC) and State of Health (SOH). By integrating simulation, machine learning, and 3D visualization, the project effectively demonstrated a scalable approach for predicting battery degradation across thousands of cycles. The model has shown high accuracy in forecasting long-term performance and supports the design of intelligent battery management systems. Alongside the achieved objectives, there remain promising opportunities to enhance the system further and extend its application to more complex energy storage scenarios. This section highlights the key conclusions, future scope, acquired skills, and potential advancements based on the work completed.

6.1 Conclusion

The project successfully achieved its goal of designing and developing a smart battery health assessment model using the concept of digital twinning. By leveraging MATLAB and Simulink for dynamic simulation, machine learning algorithms for long-term SOH prediction, and ANSYS for thermal and structural validation, the system provided a comprehensive framework for real-time battery diagnostics. The integration of data-driven and physics-based approaches allowed for accurate estimation of State of Charge (SOC) and State of Health (SOH), addressing one of the critical challenges in battery management systems.

Simulation results showed strong alignment with expected battery behavior under varying C-rates, validating the accuracy and reliability of the models. The machine learning component, especially the Multi-Layer Perceptron (MLP), demonstrated excellent predictive performance with an R^2 score of 0.988, confirming its suitability for deployment in forecasting SOH over extended cycling periods. The digital twin framework was further enhanced with interactive 3D visualization and graphical interfaces, making it both informative and accessible for real-time monitoring. The project not only advanced technical understanding of battery degradation dynamics but also provided a modular and scalable platform that can be extended to different chemistries, capacities, or use-case environments such as electric vehicles and renewable energy storage. The findings reaffirm the importance of predictive modeling in optimizing battery performance, planning maintenance, and ensuring operational safety.

Overall, the work laid a solid foundation for future research and development in digital twin-based battery analytics, combining simulation accuracy, machine learning intelligence, and user-centric interface design into a unified system.

6.2 Future Scope

The framework developed in this project opens several avenues for future enhancement and real-world deployment. While the current model effectively estimates SOC and predicts SOH using a data-driven and simulation-integrated digital twin, further improvements can significantly expand its accuracy, adaptability, and functionality across various battery technologies and operational environments. One of the most promising directions is the integration of real-time sensor data from physical battery systems. Connecting the digital twin directly to a Battery Management System (BMS) through IoT interfaces would enable live data streaming, allowing for immediate updates to the model's predictions and a higher degree of responsiveness in energy systems. This would make the model suitable for use in electric vehicles, smart grid storage, and other critical infrastructure.

Another potential improvement lies in incorporating more advanced deep learning architectures, such as convolutional neural networks (CNNs) for spatial analysis of thermal images or hybrid models combining LSTM and attention mechanisms to capture long-range degradation dependencies. These enhancements can lead to even more accurate long-term SOH forecasting under complex loading and environmental conditions. From a simulation standpoint, thermal and mechanical behavior modeled in ANSYS can be more tightly coupled with MATLAB to account for the multi-physics nature of battery degradation. This would allow the digital twin to simulate not only electrical but also thermal and structural stress scenarios, enhancing safety prediction capabilities. Moreover, extending the system to accommodate different battery chemistries—such as NMC, LTO, or solid-state batteries—will improve the generalizability of the framework. This can be achieved by retraining the model with relevant datasets and adjusting simulation parameters accordingly. Finally, deploying the digital twin model on embedded systems or edge devices would allow localized, real-time computation, reducing the need for cloud dependency and making the system viable for portable or resource-constrained applications.

In summary, the current work serves as a strong foundation upon which several enhancements can be built, making the digital twin model more versatile, intelligent, and practical for the evolving needs of modern energy storage technologies.

6.3 Learning Outcomes of the Project

Learning Outcomes of this project for our team are discussed as follows:

- One of the key learning outcomes of this project was gaining a deep understanding of battery fundamentals, particularly the electrochemical behavior of Lithium Iron Phosphate ($LiFePO_4$) cells. Through theoretical study and simulation, we learned how parameters such as voltage, current, internal resistance, temperature, and cycle count influence battery performance and degradation. This knowledge laid the foundation for all subsequent modeling and analysis, especially in estimating State of Charge (SOC) and State of Health (SOH).

- A significant technical takeaway was mastering MATLAB and Simulink for real-time simulation and model implementation. We built a complete battery simulation environment in Simulink that mimicked the charge–discharge behavior under different C-rates. We implemented SOC estimation using the Coulomb counting method and integrated mathematical expressions with simulation blocks. This not only improved our system modeling skills but also taught us how to validate dynamic systems in a closed-loop environment.

- We also gained hands-on experience in machine learning model development and evaluation. The project exposed us to various regression techniques such as linear, polynomial, SVR, and neural networks (MLP, LSTM). We explored their strengths and weaknesses, applied appropriate preprocessing techniques (scaling, feature engineering), and used error metrics like R^2 and MAE to validate the models. Understanding how to tune hyperparameters, avoid overfitting, and select models based on both performance and feasibility was a crucial part of this learning.

- An important outcome was the ability to create a web-based graphical user interface (GUI) for SOH prediction. This included developing an intuitive dashboard that displays real-time forecasted health values, visual degradation curves, and model comparison plots. We learned how to integrate back-end ML predictions with a front-end interface, making the system accessible to users without deep technical knowledge.

- Another important area of learning was the integration of thermal and mechanical modeling using ANSYS**.** We explored how to simulate heat distribution, stress profiles, and material behavior inside a battery. While this part was in the early stages, it introduced us to the concept of multi-physics simulation, which will be critical in future extensions of the digital twin.

- From a systems perspective, we learned how to combine data-driven methods with simulation-based techniques, forming a hybrid modeling approach. This was a vital learning in understanding how digital twins operate in practice—by continuously updating themselves using both real-time data and virtual simulation outputs. This fusion helped us appreciate the value of predictive analytics in modern engineering systems.

- Finally, this project helped us strengthen our collaborative and project management skills. With team members working on different components (machine learning, simulation, visualization, report writing), we had to communicate clearly, manage dependencies, and maintain version control. It provided a real-world experience of working on a multidisciplinary engineering project from start to finish, fostering technical, analytical, and organizational growth

6.4 Potential Enhancements

Finally, we came to the point of discussing potential enhancements of our project that we can carryout in the coming future.

One of the most impactful enhancements would be the integration of real-time sensor data from physical batteries into the digital twin system. Currently, the model relies on simulated and historical cycle data. By incorporating real-world sensor inputs—such as temperature, voltage, current, and internal resistance—the system could function as a live diagnostic tool for operational battery packs. This would improve prediction accuracy and allow the digital twin to adapt continuously as the battery ages in real time.

Another key enhancement would be the expansion to different battery chemistries beyond $LiFePO_4$, such as NMC (Nickel Manganese Cobalt), LTO (Lithium Titanate), or solid-state batteries. Each chemistry has unique degradation characteristics, temperature sensitivities, and voltage profiles. With appropriate dataset retraining and simulation adjustments, the current framework could be generalized to support diverse energy storage technologies used in electric vehicles, drones, and grid storage systems.

The thermal and structural modeling components, currently implemented using ANSYS, could be further integrated with the MATLAB-Simulink simulation loop. This would enable a more realistic, multi-physics digital twin that captures not just electrical degradation but also the effects of heat, mechanical strain, and safety-critical events like thermal runaway. Incorporating these aspects would allow the system to predict thermal failures and design better thermal management strategies.

In terms of machine learning, advanced deep learning techniques like hybrid CNN-LSTM models or transformer-based architectures could be explored for improved SOH forecasting. These models can better capture long-range dependencies and complex patterns in battery data. They can also handle multiple feature inputs more efficiently, especially when working with image-based thermal data or complex multivariate signals. Deployment of the system onto **e**dge devices or embedded hardware platforms is another important enhancement. With optimization, the trained models could be implemented on microcontrollers within Battery Management Systems (BMS), enabling decentralized real-time SOH prediction. This would reduce reliance on external computing resources and make the system viable for portable and remote applications, such as in EVs or solar storage systems. A further enhancement would involve the development of a cloud-based dashboard to allow remote access to battery health insights. Coupling the digital twin with IoT connectivity would enable centralized monitoring of battery fleets in EV stations, industrial backup systems, or solar farms. This would support predictive maintenance and large-scale energy analytics.

Lastly, incorporating economic and sustainability analysis into the digital twin could offer additional value. By estimating end-of-life, replacement timelines, and cost-benefit analysis of different usage patterns, the system could guide decision-making not only for performance but also for lifecycle cost optimization and environmental impact.

Appendix A

Code

APPENDIX A

CODE

A.1 SOH Estimation

```
filePath = 'C:\Users\owner''s\Desktop\matlab hands
on\IDP_MATLAB\RESULTS\Battery_Data_Cleaned.csv';
data = readtable(filePath);
Cref = 2.7;
Re_ref = 0.002249;
Rct_ref = 0.2817;
wCap = 0.6;
wRe = 0.3;
wRct = 0.1;
% Compute normalized parameters for each cycle
normCap = data.Capacity ./ Cref; % fraction of rated
capacity
normRe = Re_ref ./ data.Re; % relative ohmic resistance
normRct = Rct_ref ./ data.Rct; % relative charge-transfer
resistance
% Combine into SOH percentage
data.SOH = 100 * (wCap * normCap + wRe * normRe + wRct *
normRct);
data.SOH = round(data.SOH, 2);
% Save to new CSV
outputPath = 'C:\Users\owner''s\Desktop\matlab hands
on\IDP_MATLAB\RESULTS\SOH_Results.
csv';
writetable(data, outputPath);
% Display message
disp('SOH values calculated and saved to
SOH_Results.csv');
figure;
plot(data.test_id, data.SOH, '-o', 'LineWidth', 1.5,
'MarkerSize', 5);
xlabel('Cycle Number');
ylabel('SOH (%)');
title('Battery SOH over 100 Cycles (Capacity + Re +
Rct)');
grid on;
```

A.2 SOC Estimation

```
clear; clc;
% Folder base path
baseFolder = "C:\Users\owner's\Desktop\matlab hands
on\IDP_MATLAB";
% Battery nominal capacity (Ah)
C_rated_Ah = 2; % change accordingly
C_rated_As = C_rated_Ah * 3600; % convert Ah to As
% Initialize variables for concatenated plot
time_all = [];
SOC_all = [];
% Initial SOC at start of first cycle (assumed fully charged)
SOC_initial = 1;
% Running time offset for continuous time axis
time_offset = 0;
for cycleNum = 1:100
fprintf('Processing cycle %d...\n', cycleNum);
% Construct folder and filenames
cycleFolder = fullfile(baseFolder, ['cycle'
num2str(cycleNum)]);
chargeFile = fullfile(cycleFolder, ['charge' num2str(cycleNum)
'.csv']);
dischargeFile = fullfile(cycleFolder, ['discharge'
num2str(cycleNum) '.csv']);
% Read charge and discharge data tables
chargeData = readtable(chargeFile);
dischargeData = readtable(dischargeFile);
% Extract time and currents
charge_time = chargeData.Time;
charge_current = chargeData.Current_charge; % positive charging
current
discharge_time = dischargeData.Time;
discharge_current = dischargeData.Current_load; % positive load
current
% Calculate SOC during charging (cumulative integration)
SOC_charge = SOC_initial + cumtrapz(charge_time,
charge_current) / C_rated_As;
% Adjust discharge current sign to negative (discharging)
discharge_current = -abs(discharge_current);
% Calculate SOC during discharging starting from end of
charging SOC
SOC_discharge = SOC_charge(end) + cumtrapz(discharge_time,
discharge_current) /
C_rated_As;
```

```
time_cycle = [charge_time; discharge_time + charge_time(end)];
SOC_cycle = [SOC_charge; SOC_discharge];
% Offset time for continuous plotting across cycles
time_cycle = time_cycle + time_offset;
% Append cycle data to full arrays
time_all = [time_all; time_cycle];
SOC_all = [SOC_all; SOC_cycle];
% Update initial SOC for next cycle as last SOC of this cycle
SOC_initial = SOC_cycle(end);
% Update time offset for next cycle
time_offset = time_all(end);
end
% Plot continuous SOC profile over 100 cycles
figure;
plot(time_all, SOC_all, 'LineWidth', 2);
grid on;
xlabel('Time (seconds)');
ylabel('State of Charge (SOC)');
title('Battery SOC over 100 Charge-Discharge Cycles');
ylim([0 1.1]);
% Create a table with time and SOC
resultTable = table(time_all, SOC_all, 'VariableNames',
{'Time_seconds', 'SOC'});
for cycleNum = 1:100
fprintf('Processing cycle %d...\n', cycleNum);
% File paths for current cycle
cycleFolder = fullfile(baseFolder, ['cycle'
num2str(cycleNum)]);
chargeFile = fullfile(cycleFolder, ['charge' num2str(cycleNum)
'.csv']);
dischargeFile = fullfile(cycleFolder, ['discharge'
num2str(cycleNum) '.csv']);
% Read charge and discharge data
chargeData = readtable(chargeFile);
dischargeData = readtable(dischargeFile);
% Extract relevant data
charge_time = chargeData.Time;
charge_current = chargeData.Current_charge; % Positive charging
current
discharge_time = dischargeData.Time;
discharge_current = dischargeData.Current_load; % Positive load
charge_current) / C_rated_As;
```

```
C_rated_As;
% Combine time and SOC vectors for this cycle
% Note: time is relative within cycle (no global offset here)
time_cycle = [charge_time; discharge_time + charge_time(end)];
SOC_cycle = [SOC_charge; SOC_discharge];
% Save SOC and time for this cycle in separate CSV file
cycleTable = table(time_cycle, SOC_cycle, 'VariableNames',
{'Time_seconds', 'SOC'});
filename = fullfile(baseFolder, ['SOC_cycle' num2str(cycleNum)
'.csv']);
writetable(cycleTable, filename);
% Update SOC_initial for next cycle to carry over SOC
SOC_initial = SOC_cycle(end);
end
```

A.3 Charging Behavior

```
% LiFePO4 Battery Charging Simulation - 2.7V to 4.2V
% Simulates charging behavior from 2.7V to 4.2V at
different C-rates
clear all; close all; clc;
%% Battery Parameters
V_min = 2.7; % Minimum voltage (V) - strictly enforced
V_max = 4.2; % Maximum voltage (V) - strictly enforced
V_plateau = 3.3; % Plateau voltage (V)
% C-rates to simulate
C_rates = [0.3, 0.5, 0.8, 1.0, 2.0];
% Colors matching the reference image
line_colors = {[0.3, 0.6, 1.0], [1.0, 0.5, 0.2], [0.2, 0.7,
0.2], [0.8, 0.2, 0.8], [1.0,
0.2, 0.2]};
for i = 1:length(C_rates)
C_rate = C_rates(i);
% Capacity vector (0% to 110% to show overcharge region)
capacity_percent = linspace(0, 110, 1000);
% Initialize voltage array
voltage = zeros(size(capacity_percent));
% Voltage calculation based on LiFePO4 characteristics
(2.7V to 4.2V)
for j = 1:length(capacity_percent)
soc = capacity_percent(j) / 100; % State of charge (0-1.1)
if soc <= 0.05
% Initial rapid voltage rise from 2.7V to plateau start
voltage(j) = V_min + (3.25 - V_min) * (soc / 0.05)^0.5;
elseif soc <= 0.90
% Long plateau region - characteristic of LiFePO4
plateau_progress = (soc - 0.05) / (0.90 - 0.05);
% Gradual voltage increase during plateau
voltage(j) = 3.25 + 0.20 * plateau_progress^1.2;
elseif soc <= 1.0
% Sharp voltage rise to 4.2V near end of charge
final_progress = (soc - 0.90) / 0.10;
voltage(j) = 3.45 + (V_max - 3.45) * final_progress^0.6;
else
% Overcharge region (100% to 110%)
overcharge_progress = (soc - 1.0) / 0.10;
voltage(j) = V_max + 0.05 * overcharge_progress^2;
end
% Add C-rate dependent effects
if C_rate >= 1.0
% Higher C-rates show slightly higher voltage due to
overpotential
```

```
c_rate_factor = (C_rate - 0.3) * 0.015;
if soc <= 0.90
voltage(j) = voltage(j) + c_rate_factor * (1 - soc^2);
end
end
% Ensure voltage stays within bounds (2.7V to 4.25V max)
voltage(j) = max(V_min, min(4.25, voltage(j)));
end
% Plot the charging curve
end
fprintf('C-Rate Analysis:\n');
fprintf('================\n');
for i = 1:length(C_rates)
% Calculate approximate charging characteristics
plateau_start = 3.25 + (C_rates(i) >= 1.0) * 0.015 *
(C_rates(i) - 0.3);
plateau_end = 3.45 + (C_rates(i) >= 1.0) * 0.010 *
(C_rates(i) - 0.3);
fprintf('%.1fC Rate:\n', C_rates(i));
fprintf(' - Plateau Start: ~%.2fV\n', plateau_start);
fprintf(' - Plateau End: ~%.2fV\n', plateau_end);
fprintf(' - Final Voltage: %.1fV\n', V_max);
% Estimate charging time
if C_rates(i) <= 0.5
charge_time = 2.8 / C_rates(i);
elseif C_rates(i) <= 1.0
charge_time = 2.2 / C_rates(i);
else
charge_time = 1.8 / C_rates(i);
end
fprintf(' - Est. Charge Time: %.1f hours\n\n',
charge_time);
end
fprintf('Key LiFePO4 Characteristics Observed:\n');
fprintf('====================================\n');
fprintf('1. Rapid initial voltage rise from 2.7V to plateau
(~3.25V)\n');
fprintf('2. Extended flat plateau region (3.25V to
3.45V)\n');
fprintf('3. Sharp voltage increase from 3.45V to 4.2V
(final 10%%)\n');
fprintf('4. Higher C-rates show elevated plateau voltage
due to overpotential\n');
fprintf('5. All curves converge at 4.2V maximum
voltage\n');
fprintf('6. Voltage range strictly maintained: 2.7V -
4.2V\n\n');
%% Create additional analysis plot
figure('Position', [150, 100, 1200, 400]);
set(gcf, 'Color', 'white');
```

```
subplot(1, 3, 1);
time_vectors = {};
voltage_vs_time = {};
for i = 1:length(C_rates)
% Simulate time-based charging
if C_rates(i) <= 0.5
t_max = 3.5 / C_rates(i);
else
t_max = 2.5 / C_rates(i);
end
t = linspace(0, t_max, 500);
v_t = zeros(size(t));
for j = 1:length(t)
% Simple time-to-voltage relationship
if t(j) == 0
v_t(j) = V_min;
else
progress = min(1, t(j) * C_rates(i) / 2.5);
if progress <= 0.05
v_t(j) = V_min + (3.25 - V_min) * (progress / 0.05)^0.5;
elseif progress <= 0.90
plateau_prog = (progress - 0.05) / 0.85;
v_t(j) = 3.25 + 0.20 * plateau_prog^1.2;
else
final_prog = (progress - 0.90) / 0.10;
v_t(j) = 3.45 + (V_max - 3.45) * final_prog^0.6;
end
end
end
plot(t, v_t, 'Color', line_colors{i}, 'LineWidth', 2, ...
'DisplayName', sprintf('%.1fC', C_rates(i)));
hold on;
end
xlabel('Time (hours)', 'FontSize', 11);
ylabel('Voltage (V)', 'FontSize', 11);
title('Voltage vs Time', 'FontSize', 12, 'FontWeight',
'normal');
legend('Location', 'southeast', 'FontSize', 10);
grid on;
ylim([2.6, 4.3]);
% Subplot 2: Current vs Voltage
subplot(1, 3, 2);
voltage_range = linspace(V_min, V_max, 100);
for i = 1:length(C_rates)
current_profile = C_rates(i) * ones(size(voltage_range));
% Taper current near end of charge
```

```
taper_region = voltage_range > 4.0;
current_profile(taper_region) = C_rates(i) * (4.2 -
voltage_range(taper_region)) /
0.2;
plot(voltage_range, current_profile, 'Color',
line_colors{i}, 'LineWidth', 2, ...
'DisplayName', sprintf('%.1fC', C_rates(i)));
hold on;
end
xlabel('Voltage (V)', 'FontSize', 11);
ylabel('Current (C)', 'FontSize', 11);
title('Current Profile vs Voltage', 'FontSize', 12,
'FontWeight', 'normal');
legend('Location', 'northeast', 'FontSize', 10);
grid on;
xlim([V_min, V_max]);
% Subplot 3: Power vs Capacity
subplot(1, 3, 3);
capacity_range = linspace(0, 100, 100);
for i = 1:length(C_rates)
% Calculate power based on voltage and current
power_profile = zeros(size(capacity_range));
for j = 1:length(capacity_range)
soc = capacity_range(j) / 100;
if soc <= 0.05
v = V_min + (3.25 - V_min) * (soc / 0.05)^0.5;
elseif soc <= 0.90
v = 3.25 + 0.20 * ((soc - 0.05) / 0.85)^1.2;
else
v = 3.45 + (V_max - 3.45) * ((soc - 0.90) / 0.10)^0.6;
end
% Current decreases near end of charge
if soc > 0.9
current = C_rates(i) * (1 - soc) / 0.1;
else
current = C_rates(i);
end
power_profile(j) = v * current;
end
plot(capacity_range, power_profile, 'Color',
line_colors{i}, 'LineWidth', 2, ...
'DisplayName', sprintf('%.1fC', C_rates(i)));
hold on;
end
xlabel('Capacity (%)', 'FontSize', 11);
ylabel('Power (W)', 'FontSize', 11);
title('Charging Power vs Capacity', 'FontSize', 12,
'FontWeight', 'normal');
legend('Location', 'northeast', 'FontSize', 10);
```

A.4 Discharging Behavior

```
clc;
clear;
battery_capacity = 2.7; % in Ah
% Define capacity from 100% to 0% (discharging)
capacity = linspace(1, 0, 500); % Normalized: full (1) to
empty (0)
% Define a realistic discharging profile for LiFePO4
% Starts at 4.2V and smoothly drops to 2.7V
V_profile = @(x) 2.7 + 1.5 * (0.5*(1 - tanh((x - 0.1) *
10)) + ...
0.4*exp(-5 * x) + ...
0.1*(1 - tanh((x - 0.9) * 10)));
% Generate the base voltage curve
V_base = V_profile(capacity);
% Apply internal resistance-based voltage drop at high C-
rates
drop = @(rate) 0.01 * log(rate); % Polarization effect
increases with rate
% Define C-rates to simulate
C_rates = [0.3, 1, 2.5, 7, 20];
colors = {'b', 'r', 'm', 'g', 'k'};
V_curves = zeros(length(C_rates), length(capacity));
% Create figure
figure;
hold on;
% Plot for each C-rate
for i = 1:length(C_rates)
rate = C_rates(i);
current = rate * battery_capacity; % Current in Amps
V_curves(i, :) = V_base - drop(rate); % Discharging voltage
drop
legend_labels{i} = sprintf('%.1fC', rate);
plot(capacity * 100, V_curves(i, :), 'Color', colors{i},
'LineWidth', 2);
end
% Plot formatting
xlabel('Remaining Capacity (%)');
ylabel('Voltage (V)');
title('Discharging Behavior of 2.7Ah LiFePO_4 Battery at
Different C-Rates');
legend(legend_labels, 'Location', 'SouthWest');
grid on;
ylim([2.6 4.3]);
xlim([0 100]);
```

A.5 GUI_Battery

```
function GUI_BATTERY
% Battery setup
battery_capacity = 2.7; % Ah
C_rates = [0.3, 1, 2.5, 7, 20];
colors = {'b', 'r', 'm', 'g', 'k'};
shift = @(rate) 0.01 * log(rate); % Internal resistance
voltage shift
% Get screen size and create full-screen GUI window
screenSize = get(0, 'ScreenSize');
fig = figure('Name', 'LiFePO_4 Charging & Discharging
Simulation - 2.7Ah', ...
'NumberTitle', 'off', ...
'Units', 'pixels', ...
'Position', screenSize, ...
'MenuBar', 'none', ...
'ToolBar', 'none');
% Tab group for different plots and info
tabgroup = uitabgroup('Parent', fig);
tab1 = uitab(tabgroup, 'Title', 'Specifications');
tab2 = uitab(tabgroup, 'Title', 'Charging Behavior');
tab3 = uitab(tabgroup, 'Title', 'Discharging Behavior');
%% === Specifications Tab ===
specText = {
'Battery Specifications: LiFePO Cell';
'-----------------------------------------';
'Nominal Voltage: 3.2 V';
'Full Charge Voltage: 4.2 V';
'Cutoff Voltage: 2.7 V';
'Nominal Capacity: 2.7 Ah';
'Minimum Charge Rate: 0.3C';
'Minimum Discharge Rate: 0.3';
'Maximum Charge Rate: 20C';
'Maximum Discharge Rate: 20C';
'Chemistry: Lithium Iron Phosphate (LiFePO )';
'Cycle Life: >4000 cycles @ 80% DoD';
'Temperature Range: -15°C to 60°C';
};
uicontrol('Parent', tab1, ...
'Style', 'text', ...
'String', specText, ...
'Units', 'normalized', ...
'Position', [0.2 0.25 0.6 0.6], ...
'FontSize', 14, ...
'HorizontalAlignment', 'left');
%% === Charging Plot ===
capacity = linspace(0, 1, 500); % Normalized: 0% to 100%
V_charge_profile = @(x) 2.7 + 1.5 * (0.2*tanh((x
```

```
V_base = V_charge_profile(capacity);
ax1 = axes('Parent', tab2, 'Position', [0.1 0.15 0.85
0.75]);
hold(ax1, 'on'); grid(ax1, 'on');
title(ax1, 'Charging Behavior of LiFePO_4 (2.7Ah)');
xlabel(ax1, 'Capacity Retention (%)');
ylabel(ax1, 'Voltage (V)');
legends1 = cell(1, length(C_rates));
for i = 1:length(C_rates)
rate = C_rates(i);
current = rate * battery_capacity; % A
V_curve = V_base + shift(rate);
plot(ax1, capacity * 100, V_curve, 'Color', colors{i},
'LineWidth', 2);
legends1{i} = sprintf('%.1fC', rate);
end
legend(ax1, legends1, 'Location', 'southeast');
ylim(ax1, [2.6 4.3]); xlim(ax1, [0 100]);
%% === Discharging Plot ===
capacity_discharge = linspace(1, 0, 500); % 100% to 0%
V_discharge_profile = @(x) 2.7 + 1.5 * (0.5*(1 - tanh((x -
0.1) * 10)) + ...
0.4*exp(-5 * x) + ...
0.1*(1 - tanh((x - 0.9) * 10)));
V_base_discharge = V_discharge_profile(capacity_discharge);
ax2 = axes('Parent', tab3, 'Position', [0.1 0.15 0.85
0.75]);
hold(ax2, 'on'); grid(ax2, 'on');
title(ax2, 'Discharging Behavior of LiFePO_4 (2.7Ah)');
xlabel(ax2, 'Remaining Capacity (%)');
ylabel(ax2, 'Voltage (V)');
legends2 = cell(1, length(C_rates));
for i = 1:length(C_rates)
rate = C_rates(i);
current = rate * battery_capacity;
V_curve = V_base_discharge - shift(rate);
plot(ax2, capacity_discharge * 100, V_curve, 'Color',
colors{i}, 'LineWidth', 2);
legends2{i} = sprintf('%.1fC', rate);
end
legend(ax2, legends2, 'Location', 'southwest');
ylim(ax2, [2.6 4.3]); xlim(ax2, [0 100]);
end
```

A.6 Battery Analysis

```
C-Rate Analysis:
================
0.3C Rate:
- Plateau Start: ~3.25V
- Plateau End: ~3.45V
- Final Voltage: 4.2V
- Est. Charge Time: 9.3 hours
0.5C Rate:
- Plateau Start: ~3.25V
- Plateau End: ~3.45V
- Final Voltage: 4.2V
- Est. Charge Time: 5.6 hours
0.8C Rate:
- Plateau Start: ~3.25V
- Plateau End: ~3.45V
- Final Voltage: 4.2V
- Est. Charge Time: 2.8 hours
1.0C Rate:
- Plateau Start: ~3.26V
- Plateau End: ~3.46V
- Final Voltage: 4.2V
- Est. Charge Time: 2.2 hours
2.0C Rate:
- Plateau Start: ~3.28V
- Plateau End: ~3.47V
- Final Voltage: 4.2V
- Est. Charge Time: 0.9 hours
Key LiFePO4 Characteristics Observed:
====================================
1. Rapid initial voltage rise from 2.7V to plateau (~3.25V)
2. Extended flat plateau region (3.25V to 3.45V)
3. Sharp voltage increase from 3.45V to 4.2V (final 10%)
4. Higher C-rates show elevated plateau voltage due to
overpotential
5. All curves converge at 4.2V maximum voltage
6. Voltage range strictly maintained: 2.7V - 4.2V
Analysis Complete - All plots generated successfully!
Main plot shows charging curves from 2.7V to 4.2V
Additional analysis shows time, current, and power profiles
```

A.7 Internal Resistance Calculation

```
baseDir = 'C:\Users\owner''s\Desktop\matlab hands
on\IDP_MATLAB\';
Re_data = readtable(fullfile(baseDir, 'Re_verified.csv'));
Re_values = Re_data.Re;
Rct_values = zeros(1000, 1);
I = 1; % Ampere
for k = 1:100
folderName = sprintf('cycle%d', k);
fileName = sprintf('discharge%d.csv', k);
filePath = fullfile(baseDir, folderName, fileName);
try
T = readtable(filePath);
if height(T) >= 100
V4 = T.Voltage_load(4);
V100 = T.Voltage_load(100);
deltaV = V4 - V100;
Re_k = Re_values(k);
Rct = max(0, (deltaV - I * Re_k) / I);
Rct_values(k) = Rct;
else
Rct_values(k) = NaN;
end
catch ME
warning("Error processing cycle %d: %s", k, ME.message);
Rct_values(k) = NaN;
end
end
Rct_table = table((1:1000)', Rct_values, 'VariableNames',
{'Cycle', 'Rct'});
Rct_table.Rct = sort(Rct_table.Rct, 'ascend');
writetable(Rct_table, fullfile(baseDir,
'Rct_Estimated.csv'));
disp('✅ Rct values calculated and saved to
Rct_Estimated.csv');
```

A.8 Project Architecture

```
battery-2/
├── data.csv                              # Primary dataset
(1000 cycles)
├── final_analysis.py                     # Main analysis
pipeline
├── generate_mlp_forecast.py              # Specialized MLP
predictions
├── generate_separate_plots.py            # Individual model
visualizations
├── pyproject.toml                        # Project
configuration
├── uv.lock                               # Dependency lock
file
├── DOCS.md                               # This
documentation

├── results/                              # Main analysis
outputs
│   ├── README.md                         # Analysis summary
report
│   ├── soh_degradation.png               # EDA
visualization
│   ├── validation_forecast.png           # Model validation
plot
│   └── final_forecast.png                # Long-term
predictions

├── separate_plots/                       # Individual model
analysis
│   ├── README.md                         # Individual plot
documentation
│   ├── polynomial_regression_(deg_3)_forecast.png
│   ├── svr_(rbf_kernel)_forecast.png
│   ├── pytorch_mlp_forecast.png
│   └── pytorch_lstm_forecast.png

└── mlp_forecast/                         # Specialized MLP
analysis
    ├── README.md                         # MLP-specific
documentation
    ├── mlp_forecast_plot.png             # Extended MLP
visualization
    └── mlp_future_soh_predictions.csv
```

A.9 Data Flow Architecture

```
Raw Data (data.csv)
        ↓
Data Preprocessing Pipeline
        ↓
Model Training & Validation
        ├── Traditional ML Models (sklearn)
        └── Deep Learning Models (PyTorch)
        ↓
Performance Evaluation
        ├── Validation Metrics
        ├── Cross-Validation
        └── Error Analysis
        ↓
Visualization Generation
        ├── EDA Plots
        ├── Validation Plots
        ├── Forecast Plots
        └── Individual Model Plots
        ↓
Report Generation
        ├── Technical Documentation
        ├── Executive Summary
        └── Model Recommendations
```

A.10 Project Configuration

```
[project]
name = "battery-2"
version = "0.1.0"
requires-python = ">=3.13"

dependencies = [
  "lightgbm>=4.6.0",          # Gradient boosting framework
  "matplotlib>=3.10.3",       # Visualization library
  "numpy>=2.3.0",             # Numerical computing
  "onnx>=1.18.0",             # Model serialization and
interoperability
  "pandas>=2.3.0",            # Data manipulation and
analysis
  "scikit-learn>=1.7.0",      # Machine learning algorithms
and tools
  "torch>=2.7.1",             # Deep learning framework
(PyTorch)
]
```

A.11 Machine Learning Model initialization

```
import torch
import torch.nn as nn
from flask import Flask, request, jsonify

# -------------------------
# Multi-Layer Perceptron (MLP)
# -------------------------
class MLP(nn.Module):
    def __init__(self):
        super(MLP, self).__init__()
        self.layers = nn.Sequential(
            nn.Linear(1, 128),     # Input layer
            nn.ReLU(),             # Activation function
            nn.Linear(128, 128),   # Hidden layer
            nn.ReLU(),             # Activation function
            nn.Linear(128, 1)      # Output layer
        )

    def forward(self, x):
        return self.layers(x)

# -------------------------
# Long Short-Term Memory (LSTM)
# -------------------------
class LSTM(nn.Module):
    def __init__(self, input_size=1, hidden_size=50,
num_layers=2, output_size=1):
        super(LSTM, self).__init__()
        self.lstm = nn.LSTM(
            input_size,
            hidden_size,
            num_layers,
            batch_first=True,
            dropout=0.2
        )
        self.fc = nn.Linear(hidden_size, output_size)

    def forward(self, x):
        out, _ = self.lstm(x)
        return self.fc(out[:, -1, :])  # Use last time step
output
```

```
# --------------------------
def train_pytorch_model(model, train_loader, epochs,
device):
    criterion = nn.MSELoss()
    optimizer = torch.optim.Adam(model.parameters(),
lr=0.001)

    for epoch in range(epochs):
        model.train()
        for seq, label in train_loader:
            seq, label = seq.to(device), label.to(device)

            optimizer.zero_grad()
            y_pred = model(seq)
            loss = criterion(y_pred, label)

            loss.backward()
            optimizer.step()

# --------------------------
# Flask API for Prediction
# --------------------------
app = Flask(__name__)

@app.route('/predict', methods=['POST'])
def predict_soh():
    cycle_number = request.json['cycle']
    prediction = model.predict([[cycle_number]])
    return jsonify({'soh_prediction':
float(prediction[0])})

# --------------------------
# Batch Prediction Example
# --------------------------
def batch_predict(cycle_numbers):
    predictions = model.predict(cycle_numbers.reshape(-1,
1))
    return predictions.tolist()
```

BIBLIOGRAPHY

[1] Xiaowei Hu, Shuang Li, and Ming Jiang, “Lithium-ion battery digital twin and its applications,” Journal of Power Sources, vol. 489, 2021, p. 229521.

[2] Jian Wang and Ke Liu, “State of health estimation using electrochemical impedance spectroscopy and machine learning,” Journal of Energy Storage, vol. 43, 2022, p. 103192.

[3] Haoran Zhao, Yifan Zhang, and Wei Liu, “Digital Twin-Enabled Smart Battery Management System for Electric Vehicles,” IEEE Internet of Things Journal, vol. 10, no. 12, 2023, pp. 9876–9888.

[4] Ming Chen, Li Wang, and Xiaoming Zhang, “Physics-based modeling and digital twin of lithium-ion battery,” Applied Energy, vol. 305, 2022, p. 117799.

[5] Roberto García, Luis Martínez, and Ana Pérez, “Prognostics and health management of batteries: Review and outlook,” IEEE Transactions on Vehicular Technology, vol. 70, no. 2, 2021, pp. 1020–1032.

[6] Nikhil Singh, Priya Sharma, and Rajesh Kumar, “Cycle life prediction of Li-ion batteries using support vector regression and digital twin,” Energy, vol. 269, 2023, p. 126732.

[7] Fang Yang, Jun Li, and Wei Zhang, “Hybrid physics-informed and data-driven models for SOH estimation in digital twin,” International Journal of Energy Research, vol. 44, no. 5, 2020, pp. 3456–3472.

[8] Pankaj Kumar, Anjali Verma, and Suresh Gupta, “Real-time SOC and SOH estimation using digital twin architecture,” Journal of Power Sources, vol. 488, 2021, p. 230123.

[9] Qing Li, Yao Chen, and Mei Zhao, “Digital twin for battery systems: A review of modeling techniques,” Renewable Energy, vol. 191, 2022, pp. 639–654.

[10] Sheng Wu, Lin Wang, and Tao Zhang, “Digital twin-based fault diagnosis for lithium-ion batteries,” IEEE Transactions on Industrial Informatics, vol. 19, no. 7, 2023, pp. 4553–4564.

[11] Yong Liao, Hui Chen, and Zhen Liu, “Digital twin-assisted battery cell balancing and health assessment,” IEEE Transactions on Smart Grid, vol. 14, no. 1, 2023, pp. 123–134.

[12] Maria Fernandez, Carlos Lopez, and Elena Garcia, “Cloud-based digital twin for second-use EV battery health monitoring,” Energy Reports, vol. 9, 2023, pp. 815–829.

[13] Sung Park and Hyun Cho, “Model-based digital twin for electric vehicle battery thermal management,” Applied Thermal Engineering, vol. 206, 2022, p. 118087.

[14] Zhen Li, Wei Sun, and Ming Zhao, “Transfer learning in digital twin for battery degradation forecasting,” Journal of Energy Storage, vol. 50, 2022, p. 104255.

[15] Miguel Santos, Laura Ruiz, and Pedro Martinez, “Digital twin-driven predictive maintenance for grid-scale batteries,” IEEE Transactions on Sustainable Energy, vol. 14, no. 2, 2023, pp. 520–530.

[16] Peng Zhou, Xiaoming Li, and Rui Wang, "Data-fusion digital twin for state of health estimation in hybrid battery systems," Energy Conversion and Management, vol. 254, 2022, p. 115123.

[17] Daehyun Kim, Jaehoon Lee, and Minsoo Park, "Digital twin-based multi-physics modeling for battery health prognostics," Applied Energy, vol. 260, 2020, p. 114374.

[18] Thien Nguyen, Linh Tran, and Chau Yuen, "AI-enhanced digital twin for EV battery lifecycle management," IEEE Transactions on Industrial Electronics, vol. 68, no. 11, 2021, pp. 11456–11465.

[19] Ravi Patel, Ananya Sharma, and Deepak Mehta, "Digital twin framework for solid-state battery health monitoring," Journal of Power Sources, vol. 512, 2023, p. 230579.

[20] Zhang, Y., et al., "Edge computing-enabled intelligent battery management system for electric vehicles," Journal of Energy Storage, vol. 59, 2024, p. 105257.

[21] M. Abo-Javed, et al., "Battery State-of-Health Estimation: A Step towards Battery Digital Twins," IEEE, 2023.

[22] Sadiqa Jafari and Yung-Cheol Byun, "Prediction of the Battery State Using the Digital Twin Framework Based on the Battery Management System," IEEE, 2023.

[23] Khaled Sidahmed Sidahmed Alamin, Yukai Chen, Enrico Macii, Massimo Poncino, and Sara Vinco, "A Machine Learning-based Digital Twin for Electric Vehicle Battery Modeling," IEEE, 2023.

[24] Yan Qin, Anushiya Arunan, and Chau Yuen, "Digital Twin for Real-time Li-Ion Battery State of Health Estimation With Partially Discharged Cycling Data," IEEE, 2023.

[25] Heng Li, Muaaz Bin Kaleem, I-Ju Chiu, Dianzhu Gao, and Jun Peng, "A Digital Twin Model for the Battery Management Systems of Electric Vehicles," IEEE, 2023.

[26] Kai Zhao, Ying Liu, Wenlong Ming, Yue Zhou, and Jianzhong Wu, "Digital Twin-Driven Estimation of State of Charge for Li-ion Battery," IEEE, 2023.

[27] Nina Kharlamova, Chresten Træholt, and Seyedmostafa Hashemi, "A Digital Twin of Battery Energy Storage Systems Providing Frequency Regulation," IEEE, 2023.

[28] Abhijit Kulkarni, Hoda Sorouri, Yusheng Zheng, Xin Sui, Arman Oshnoei, and Nicolai André Weinreich, "Li-ion Battery Digital Twin Based on Online Impedance Estimation," IEEE, 2023.

[29] Roberta Di Fonso, Remus Teodorescu, Carlo Cecati, and Pallavi Bharadwaj, "A Battery Digital Twin From Laboratory Data Using Wavelet Analysis and Neural Networks," IEEE, 2023.

[30] Yuxin Zhang, Jianhua Wu, Shanxun Sun, and Tao Ye, "Research on Modeling Method of Digital Twin of Lithium Battery Based on Data-Model Hybrid Drive," IEEE, 2023.

[31] Naga Durga Krishna Mohan Eaty and Priyanka Bagade, “Digital twin for electric vehicle battery management with incremental learning,” Elsevier, 2023.

[32] Irum Saba, Mukhtar Ullah, and Muhammad Tariq, “Advancing Electric Vehicle Battery Analysis With Digital Twins in Intelligent Transportation Systems,” IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 4, 2023, pp. 4567–4578.

[33] Hao Tang, Yichun Wu, Yuanfeng Cai, Fanyu Wang, Zequn Lin, and Yiru Pei, “Design of power lithium battery management system based on digital twin,” Elsevier, 2023.

[34] Xiaoyu Zhang, Bin Liu, and Jiayin Wang, “Battery life prediction using digital twin and unscented Kalman filtering,” Journal of Power Sources, vol. 517, 2022, p. 230811.

[35] Arvind Sharma, Deepika Soni, and K. Prasad, “Thermal-aware digital twin model for battery degradation analysis,” Applied Thermal Engineering, vol. 214, 2022, p. 118987.

[36] Junjie Li, Zhiqiang Wang, and Lei Zhou, “AI-integrated digital twin for battery safety assessment in EVs,” IEEE Access, vol. 11, 2023, pp. 15945–15956.

[37] Stefanie Müller, Lukas Becker, and Fabian Riedl, “Sensor fusion approach in battery digital twin design,” Sensors, vol. 21, no. 23, 2021, p. 7910.

[38] S. S. Ramesh, V. Rajendran, and M. P. Divya, “Edge-AI based digital twin framework for real-time battery state forecasting,” Journal of Energy Storage, vol. 61, 2023, p. 106824.

[39] Minseok Lee, Hyeonwoo Kim, and Jiyoung Park, “Digital twin-based estimation of remaining useful life in lithium batteries,” Energies, vol. 16, no. 5, 2023, p. 2103.

[40] Y. Lin, M. Zhao, and Q. Tang, “A co-simulation framework for digital twin of Li-ion battery using MATLAB and COMSOL,” International Journal of Energy Research, vol. 47, no. 1, 2023, pp. 321–334.