TU WIEN Informatics

Betriebsdatenbasierte Abschätzung und Vorhersage des Batteriezustands in maritimen Anwendungen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Software Engineering und Internet Computing

eingereicht von

Komić Nejra, BSc
Matrikelnummer 11719704

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Associate Prof. Dr. techn. Dipl.-Ing. Clemens Heitzinger
Zweitbetreuung: Associate Prof. Dr. techn. Dipl.-Ing. Christoph Hametner

Wien, 5. Juni 2025

_____   _____
Komić Nejra   Clemens Heitzinger

TU WIEN Informatics

Operational Data-Driven SoH Estimation and Prediction for Maritime Battery Systems

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Software Engineering and Internet Computing

by

Komić Nejra, BSc
Registration Number 11719704

to the Faculty of Informatics

at the TU Wien

Advisor: Associate Prof. Dr. techn. Dipl.-Ing. Clemens Heitzinger
Second advisor: Associate Prof. Dr. techn. Dipl.-Ing. Christoph Hametner

Vienna, June 5, 2025

_____   _____
            Komić Nejra   Clemens Heitzinger

Erklärung zur Verfassung der Arbeit

Komić Nejra, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel" habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 5. Juni 2025

_____

Komić Nejra

Danksagung

Ich möchte all jenen danken, die mich während der Entstehung dieser Masterarbeit begleitet, unterstützt und motiviert haben.

Mein besonderer Dank gilt **Prof. Christoph Hametner**, der mich durch alle Phasen der Arbeit fachlich betreut und in zahllosen Besprechungen wertvolle Empfehlungen gegeben hat. Ebenso danke ich **Prof. Clemens Heitzinger** für seine konstruktiven Ideen und seine Unterstützung in der Schlussphase der Arbeit.

Ein herzliches Dankeschön an meine Kollegen bei der **AVL**, ohne eure Hilfe wäre diese Arbeit nicht möglich gewesen. **Darko Stern**, der jederzeit erreichbar war und mir in organisatorischen wie inhaltlichen Fragen den Rücken stärkte, **Hans-Michael Kögeler**, der mir ausführliche Hintergrundinformationen zur Verfügung stellte und **Georg Engel**, der mich in der Implementierungsphase unterstützte und die Daten in auswertbarer Form bereitstellte.

Meinen Eltern danke ich von Herzen für ihre bedingungslose Unterstützung, die mein Studium erst ermöglicht hat.

Ein großes Dankeschön geht auch an meine kleinen Brüder; mein größter Erfolg ist es, euch beiden ein gutes Vorbild zu sein.

Abschließend danke ich meinem Verlobten für seinen starken emotionalen Rückhalt während der gesamten Entstehung dieser Masterarbeit.

Acknowledgements

Kurzfassung

Ziel dieser Arbeit ist es, die Batteriekapazität aus Betriebsdaten zu schätzen, Degradationstrends zu erkennen und die Ergebnisse mit manuell durchgeführten Kapazitätstests zu vergleichen.

Im Fokus steht eine vergleichende Bewertung von drei Methoden zur Schätzung des State of Health (SoH) einer Batterie unter Verwendung betrieblicher Messdaten. Die verwendeten Methoden umfassen die Coulomb Counting Methode, einen kaskadierten State of Charge (SoC) Beobachter sowie eine probabilistische Schätzmethode, die den Metropolis-Hastings-Algorithmus verwendet.

Die erste Methode, das Coulomb Counting, schätzt die Kapazität durch Integration des Stroms über die Zeit zwischen Ruhephasen. Der SoC-Observer baut auf dieser Methode auf, indem er den SoC dynamisch anpasst – basierend auf dem Fehler zwischen der vorhergesagten und der gemessenen Spannung. Diese Abweichungen liefern über längere Zeiträume wertvolle Hinweise auf die Batterie-Degradation. Der Ansatz mit dem Metropolis Hastings Algorithmus betrachtet die Kapazitätsschätzung als ein Problem der bayesschen Inferenz und liefert ein Markov Chain sowie eine Posterior Verteilung, wodurch sowohl der Kapazitätswert als auch die Unsicherheit der Schätzung ermittelt werden.

Obwohl die SoH-Schätzung ausschließlich anhand von Spannungs- und Strommessungen aus dem Betrieb eine Herausforderung darstellt, zeigen die Ergebnisse, dass alle drei Methoden in der Lage sind, Degradationstrends zuverlässig zu erkennen. Besonders an den Referenztagen, an denen manuelle Kapazitätstests durchgeführt wurden, stimmten die Schätzwerte der probabilistischen Methode und des SoC-Observers sehr gut mit den Testergebnissen überein.

Diese Ansätze bieten eine solide Grundlage für die Echtzeit- und automatisierte Überwachung des Batteriezustands (SoH) in maritimen Anwendungen und erhöhen die Betriebssicherheit durch die frühzeitige Erkennung von Degradationstendenzen.

Abstract

The aim of this thesis is to estimate battery capacity from operational data, detect degradation trends and evaluate the results against manually conducted capacity tests.

It presents a comparative study of three methods for estimating the State of Health (SoH) of a ship's battery from real-world operational data. The approaches include the Coulomb Counting Method, a cascaded State of Charge (SoC) Observer and a probabilistic estimation method which uses the Metropolis-Hastings algorithm.

The first Coulomb Counting method estimates the capacity by integrating current over time between rest periods. The SoC Observer method builds upon Coulomb Counting by adjusting the SoC dynamically based on the error between the predicted and the measured voltage, hence reducing the error accumulation. While Coulomb Counting only estimates SoC, the SoC Observer estimates SoH indirectly by observing consistent errors between measured and predicted voltage, which suggests a mismatch in assumed capacity and points to the battery degradation. The Metropolis-Hastings approach formulates capacity estimation as a Bayesian inference problem and outputs a Markov Chain and a posterior distribution, providing both the capacity estimate and the uncertainty in the estimate.

Although estimating the SoH based solely on the operational voltage and current measurements is challenging, the results demonstrate that all three approaches successfully detected degradation trends over time. Furthermore, for the reference test days— periods when manual capacity tests were performed — the capacity estimates from both the probabilistic method and the SoC Observer showed strong agreement.

The results demonstrate that reliable SoH tracking is feasible with operational data and offer a foundation for real-time, automated SoH monitoring in maritime applications, enhancing operational reliability through early detection of degradation trends.

Contents

Introduction

1.1 Motivation

The shift toward clean and sustainable energy independence is one of the European Union's top priorities with the goal of becoming the world's first climate-neutral continent by 2050 [3]. Central to this aim is the rapid expansion of battery technology. When it comes to battery performance and testing, there are some major challenges, such as capacity fade over time, temperature sensitivity, measurement noise, etc., which is why more research into this field is crucial to develop new approaches.

One of the most important aspects of a battery is its State of Health (SoH). Traditional methods for calculating the SoH, such as capacity tests, are time consuming, costly, and require the system to be taken out of operation. The results of manual capacity tests, where battery has to be fully discharged and charged, can also be influenced by various factors, such as load variability and temperature changes. These tests also contribute further to battery degradation [23]. Repeated capacity tests can accelerate battery aging and reduce battery lifespan. Therefore, accurate and non-intrusive monitoring and estimation of SoH during normal operation is crucial.

The challenge in estimating SoH is that there is no single reliable source of battery health. Furthermore, SoH is tightly coupled with State of Charge (SoC) because SoC is calculated relative to the battery's current capacity, which depends on its SoH. As the battery degrades and SoH decreases, failing to account for this change leads to inaccurate SoC estimation.

Another challenge with real-world application and operational data is that, unlike in laboratory tests, complete discharge data is often unavailable, which can limit the accuracy of SoH estimation.

1.2 Objectives

The main objective of this thesis is to identify a reliable and non-invasive method for SoH estimation based on fleet data. It includes an analysis of operational data from battery cells installed on a ship in service for approximately five years, as well as an evaluation of three methods regarding their effectiveness in detecting battery degradation.

The methods, each with its advantages and disadvantages, include:

- **The Coulomb Counting method** estimates battery capacity by integrating current over time between rest periods.

- **The Cascaded State of Charge (SoC) Observer** is a model-based method that improves the accuracy of the Coulomb Counting estimation by introducing a voltage error correction. It also tracks voltage error deviations over time to gain insights into battery capacity.

- **The probabilistic Metropolis-Hastings method** formulates capacity estimation as a Bayesian inference problem, providing both the capacity value and its associated uncertainty.

These three approaches aim to provide information about battery degradation trends and aging processes, with the goal of reducing the need for frequent manual capacity tests.

1.3   Structure of the Thesis

This thesis is structured into four main chapters: Introduction, Fundamentals and Data, State of Health Estimation Methods and Results and Discussion.

Introduction chapter introduces the *Motivation* and the *Objectives* of this thesis.

The chapter **Fundamentals and Data** gives an overview of the fundamental concepts necessary to understand the content of this thesis, such as battery design and battery management systems. SoC, SoH definitions as well as *State of the Art* of the existing approaches for estimating SoC and SoH are explored in this section. Furthermore, Basics of *Probability Theory, Bayesian Inference and Parameter Estimation* are also explained. *Description of the Data* gives the reader an overview of the operational data, in particular, how the data was collected, in which operating conditions the ship was active, etc.

The third chapter explores the **State of Health Estimation Methods** and explains the relevant approaches and algorithms used for the estimation.

Results and Discussion presents the results of each method, as well as an overall comparison of the results from all three estimation methods. A dedicated section on **Contributions** highlights the original aspects of this work and outlines which methods and results add value to the industry.

Fundamentals and Data

This chapter provides the theoretical and practical background for the methods used in this thesis. Section 2.1 introduces some basic information about the batteries. Then the basic principles behind State of Charge 2.2 and State of Health 2.3 estimation are explored, as well as the State of the Art in these fields.

Section 2.4 introduces the theoretical background needed to apply the Bayesian inference and probabilistic parameter estimation. Finally, the last Section 2.5 provides detailed description of the operational data used in this work, including how it was collected and the role in validating the estimation methods.

2.1 Battery Fundamentals

To enable accurate SoH estimation and the application of appropriate methods, it is first necessary to understand how batteries are designed and how a typical battery management system operates.

2.1.1 Battery design

For several years now, most high-voltage storage systems have had a modular design. Generally, the structure can be divided into four geometric levels: cell, module, pack and array.

The cell is the smallest unit of a battery and is constructed as an electrochemical storage device consisting of an anode, cathode, electrolyte, and cell casing [4]. Besides many different cell chemistries that can be considered, one of the most important factors is the cell type. The three most commonly used cell types – cylindrical, prismatic and pouch cells – are illustrated in the first column in the Figure 2.1.

The module groups several pairs of identical cells and has a mechanical support structure to fix and possibly secure the cells, as well as electrical connections within the module. Depending on the design, especially for high performance batteries, cooling devices (cooling plates, connectors), electronic components, and the connection technology to the battery system can also be integrated [4].

Cells and multiple modules can be arranged in parallel or series – in larger applications a mix of parallel and series connections is also possible. The type of connection, either parallel or series, affects the voltage and capacity of the module or battery system. When cells are connected in series, the voltages of each cell are summed, resulting in a higher overall voltage

while maintaining the same capacity. In contrast, connecting cells in parallel combines their capacities, keeping the voltage constant [19].

A battery pack combines multiple modules that are assembled together in a larger housing. It includes many systems, such as systems for monitoring and control of the battery operation – the Battery Management System.

Figure 2.1 shows the complete assembly process from individual cells to battery modules and packs. The second column illustrates the battery modules assembled from each cell type, while the third column shows how these modules are integrated into full battery packs.
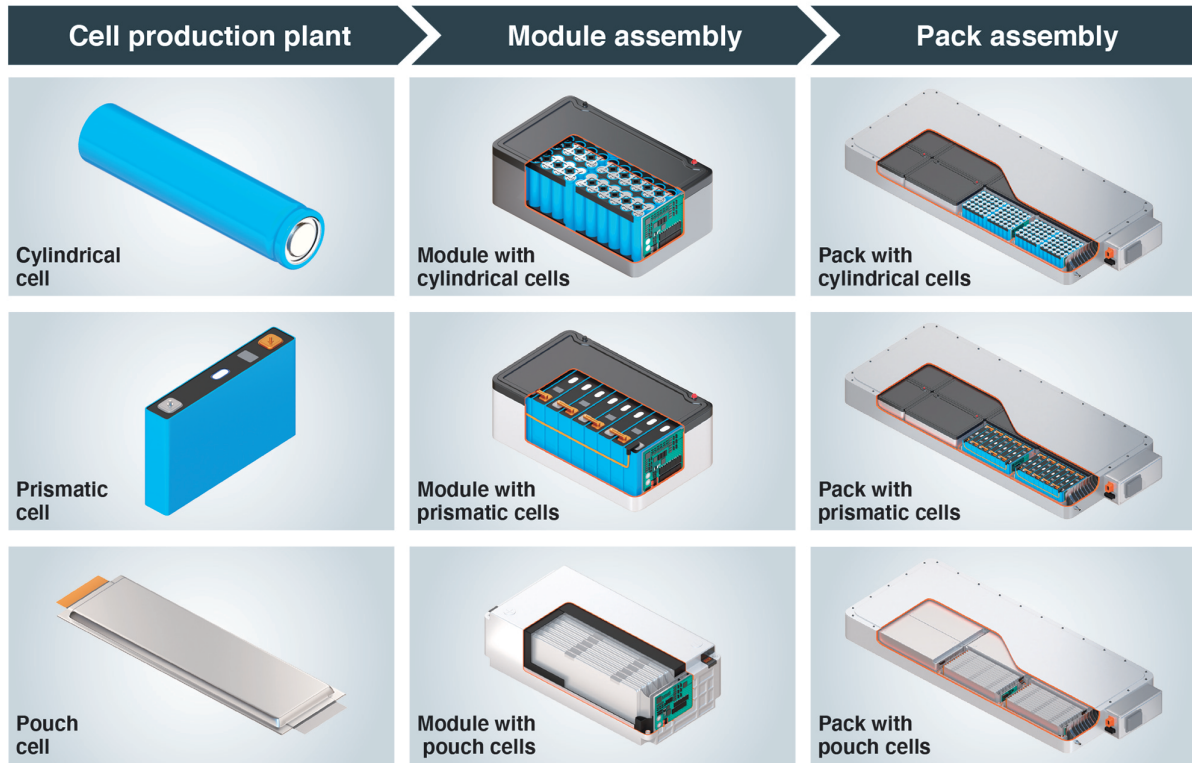

Figure 2.1: Assembly of different types of cells in battery packs [11]

Array consists of one or more packs used to power the application and managed by the battery management system (BMS). It typically includes the cooling system, safety mechanisms, and control mechanisms for interaction with the overall system.

In large applications such as electric ships, battery arrays are typically designed to meet the energy requirements, to comply with safety standards and to manage thermal conditions. As the highest level in the multi-level structure, these arrays often include redundancy and isolation mechanisms to ensure long-term operational safety and reliability.

2.1.2   Battery Management System

Battery Management System (BMS) is **the central control unit of the battery system**, which constantly tracks the parameters such as temperature, current, voltage through a network of sensors. The main goal of a BMS is ensuring the reliable and safe operation of the battery [20]. It continuously monitors key variables such as current, temperature, resistance, as well as the SoC and the SoH which are considered to be the most important ones [28].

2.2 State of Charge (SoC) Estimation Principles

The State of Charge is a measure of **the remaining available charge in the battery**, typically expressed as a percentage, where 100% indicates a fully charged and 0% indicates a fully discharged battery. SoC at time t can be defined as

$$\text{SoC}(t) = \frac{Q_{\text{stored}}(t)}{C_{curr}(t)}, \tag{2.1}$$

where

- $Q_{\text{stored}}(t)$ is **the amount of charge currently stored in the battery** at time t, typically measured in Ampere-hours (Ah). It changes continuously with charge and discharge current.

- $C_{curr}(t)$ is the **current available capacity** of the battery at time t. It is the maximum amount of charge the battery can store or deliver at time t, given its present State of Health, temperature and other conditions. At the battery's begin-of-life (BOL), $C_{curr}(0)$ equals the nominal capacity C_{nom}. With aging and degradation, $C_{curr}(t)$ decreases and thus differs from the C_{nom}.

Reliable SoC allows for reliable prediction of the remaining operational range of a ship. The difficulty in accurately estimating and monitoring SoC and SoH lies in the effect of other non-linear internal chemical reaction and physical processes of the battery. Furthermore, battery's actual capacity $C_{curr}(t)$ changes over time. Many BMS approximate $C_{curr}(t)$ with the fixed nominal capacity which introduces a drift in the computed SoC as the battery ages.

Traditional SoC Estimation Methods can be broadly categorized into four main groups [25]: *Coulomb Counting, Open-Circuit Voltage, Model Based* and *Machine Learning.*

2.2.1 Coulomb Counting Method (CCM)

The Coulomb Counting measures the charge flowing into and out of the battery. It estimates SoC by **integrating the current over time**.

The fundamental equation for Coulomb Counting is

$$\text{SoC}(t) = \text{SoC}(t_0) + \frac{1}{C} \int_{t_0}^{t} I(\tau)\, d\tau, \tag{2.2}$$

where

- $\text{SoC}(t)$ is the State of Charge at time t,

- $\text{SoC}(t_0)$ is the initial State of Charge at the starting time t_0,

- C is the battery capacity in Ampere-hours (Ah),

- $I(\tau)$ is the current at time τ,

- $\int_{t_0}^{t} I(\tau)\, d\tau$ is the integral of the current over time, representing the total charge in Ampere-hours.

The Coulomb Counting method is advantageous for the applications where simplicity and low computational requirements are essential, but it has significant limitations, including error accumulation and the need for an initial SoC value. Therefore, when it comes to more complex applications, it is common to combine Coulomb Counting together with some other methods, such as OCV to accomplish a more accurate and reliable SoC estimation [1].

2.2.2 OCV Method

An open circuit is a condition where the battery is at rest, meaning **no current flows in or out of the battery**. In this state, the voltage measured across the battery terminals is called the Open-Circuit Voltage (OCV). In battery packs, the OCV is measured on cell level – usually monitored with voltage sensors by the BMS [37].

When the battery is at rest – in an open-circuit condition – the OCV provides a reliable indication of the battery's State of Charge [27]:

$$\text{SoC} = f(OCV). \tag{2.3}$$

The function $f(\text{OCV})$ is a numerical lookup table that returns the SoC corresponding to a given OCV value. It is derived from an OCV curve fitted using controlled laboratory measurements. OCV also varies with temperature [37].

The battery's OCV and its SoC's relationship is generally monotonically increasing [22], meaning as the SoC increases, so does the OCV. Therefore, the quality of the OCV-based model directly impacts the accuracy of the resulting SoC estimation [35]. However, practical challenges arise because this relationship can vary significantly among different battery chemistries, manufacturing differences, aging, temperature effects and hysteresis.

2.2.3 The Model-Based Method (MBM)

This method combines measured battery values, such as voltage, current and temperature to a battery model. It makes use of different battery models like Equivalent Circuit Model (ECM), Electrical Equivalent Circuit Model (EECM), and Electrochemical Impedance Model (ECIM).

The Equivalent Circuit Model (ECM) represents a battery using an electrical circuit composed of elements like resistors and capacitors. This model captures the battery's dynamic behavior by simulating its voltage response to current inputs. ECMs are widely used due to their simplicity, low computational requirements, and suitability for real-time applications like SoC estimation in BMS [12].

The Electrical Equivalent Circuit Model (EECM) is a specific type of ECM that focuses on accurately representing the electrical behavior of batteries. Such approaches are particularly interesting for online SoC estimation due to their high accuracy and low complexity. However, in general, ECM-based approaches involve unknown variables and nonlinear equations [25].

The Electrochemical Impedance Model (ECIM) utilizes Electrochemical Impedance Spectroscopy (EIS) to analyze the frequency response of a battery. This method provides detailed insights into the battery's internal processes, such as charge transfer resistance and double-layer capacitance. ECIMs are valuable for assessing the State of Health (SoH) and diagnosing degradation mechanisms, although they require specialized equipment and are more complex to implement [24].

Models are typically also used in combination with an observer, such as a **Luenberger observer or an Extended Kalman Filter**. In the observer framework, the chosen battery model provides the *prediction* step, while the discrepancy between the predicted and the measured

voltage is fed back to correct the internal state estimate. Observers therefore inherit the physical insight of ECM/EECM/ECIM models, while compensating for model uncertainty, sensor noise, and aging-induced parameter drift in real-time [25].

2.2.4 Machine Learning Methods

Machine learning methods such as Artificial Neural Networks, Fuzzy logic, and Support Vector Mechanisms are commonly used to estimate SoH. Although machine learning methods can consider real-world dynamic conditions and are suitable for all battery types, their accuracy relies heavily on **extensive training datasets** and they require a lot of computation power for handling large data sets, making them resource-intensive [25].

2.3 State of Health (SoH): Definition and Relevance

State of Health (SoH) is the crucial parameter for battery monitoring and estimating SoH is one of the primary challenges in the battery application.

SoH is defined as the **battery's ability to store and deliver energy relative to its condition when new** [9]. The SoH can be expressed as

$$\text{SoH} = \frac{C_{\text{curr}}(t)}{C_{\text{nom}}},\tag{2.4}$$

where

- $C_{curr}(t)$ is the **current available capacity** of the battery at time t, reflecting degradation compared to its original state.

- C_{nom} is the **nominal capacity** defined by the battery manufacturer at the begin-of-life (when the battery was new).

In this thesis, we aim to estimate $C_{curr}(t)$. Once we estimate the current capacity $C_{curr}(t)$, it can be be divided by the C_{nom} to express the battery's SoH. SoH is usually expressed in percent (%) and the capacity in Ampere-hours (Ah).

Another parameter which is usually monitored alongside SoH is **internal resistance** of the battery. When SoH of the battery decreases, the internal resistance generally increases and is therefore monitored alongside capacity and could help estimate the SoH [34]. According to the definition in terms of resistance, the SoH can be expressed as

$$\text{SoH}_R(t) = \frac{R_{\text{EOL}} - R_{\text{curr}}}{R_{\text{EOL}} - R_{\text{new}}},\tag{2.5}$$

where

- R_{EOL} is the maximum internal resistance value specified by the manufacturer, beyond which the battery is considered to have reached the end of its usable life,

- R_{new} is the internal resistance of a brand new cell,

- R_{curr} is the internal resistance under load.

Typically, a battery reaching about 80% of its nominal capacity is deemed end-of-life and should be serviced or replaced, as defined by standards such as IEEE 1188 and many EV warranties.

Capacity usually declines gradually through repetitive charging and discharging until it reaches the **knee-point** after which it goes through rapid and irreversible deterioration to reach its end-of-life [26]. Knee point is basically a transition from a slow capacity degradation rate to a fast one. The paper [32] identifies that capacity degradation often follows a non-linear trajectory, with an initial slow decline that becomes more pronounced after the knee point. An example of such capacity degradation and knee point is visible in Figure 2.2.
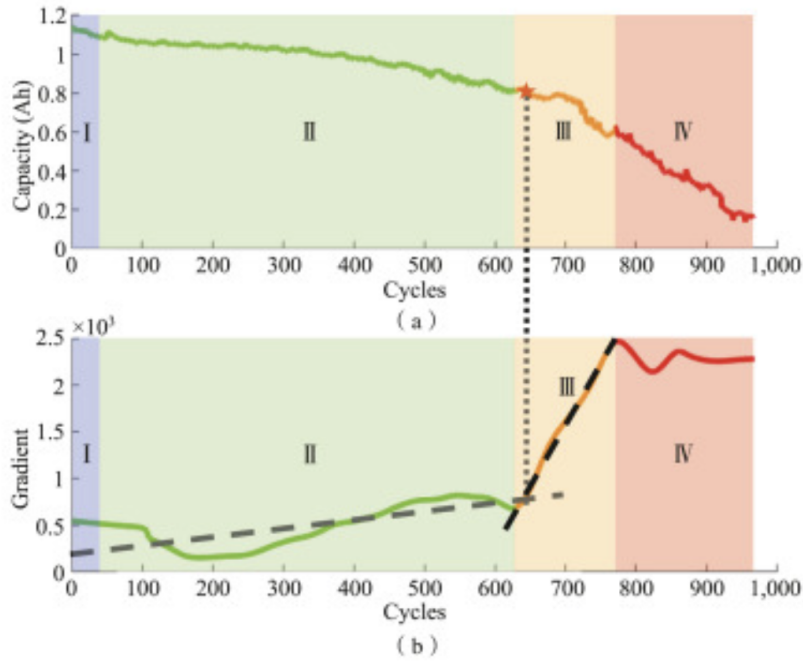

Figure 2.2: Capacity degradation partition [32]

2.3.1   SoH Estimation: State of the Art

Existing studies on battery aging come from many various fields and can be classified into experimental, model-based and data-driven methods.

Experimental methods typically include direct calculation methods and indirect analysis approaches.

- Direct methods involve quantifying battery aging by performing full charge-discharge cycles, therefore establishing a complete reference cycle [7].

- Indirect analysis methods estimate battery aging through observation of symptoms or indicators of degradation rather than directly measuring capacity or impedance.

Spectroscopy and electrochemical techniques include detailed analytical measurements of battery characteristics. Electrochemical Impedance Spectroscopy (EIS) is one of the electrochemical models that measures the impedance of the battery over a range of frequencies [6]. Another widely used analytical method is Incremental Capacity Analysis (ICA), which compares incremental capacity with voltage curve to detect changes in battery state [30]. A clear disadvantage of such methods is that they often require strict hardware conditions or rely extensively
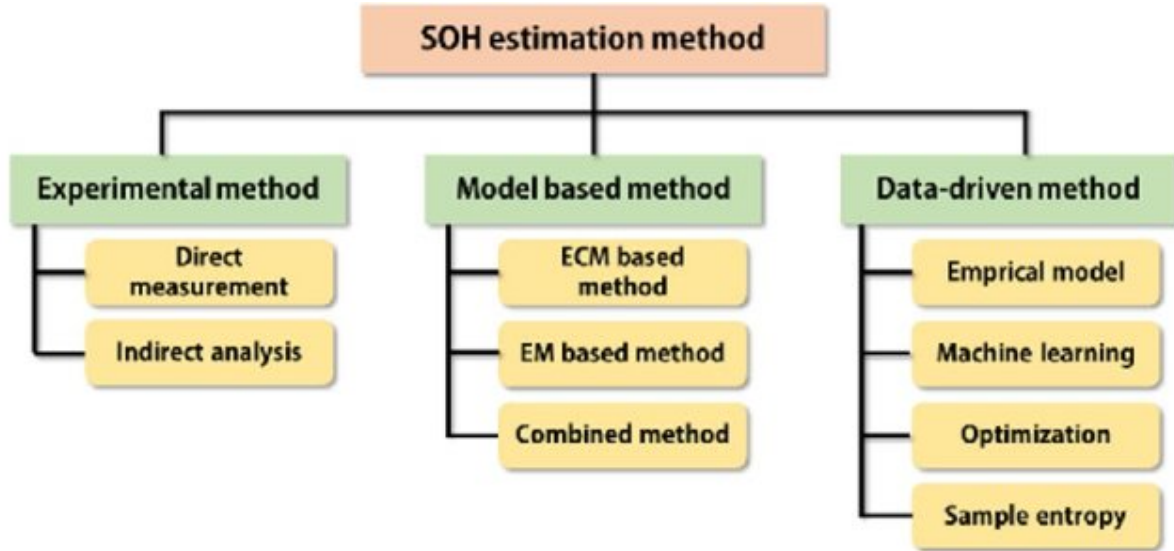
Figure 2.3: Classification of battery SoH estimation methods [21]

on accurately derived model parameters [5], thus limiting their practical implementation for online montoring.

The second category of SoH estimation methods that has gained significant popularity in the recent years, as illustrated in Figure 2.3 are **Model-based Methods** [36].

Equivalent Circuit Model (ECM) is widely used online SoH estimation model in electric vehicles and in various engineering applications. ECM represents the battery using simplified electrical circuits composed of elements such as resistors, capacitors, voltage sources etc. [31]. The model establishes the relationship between the external measurable characteristics (such as voltage and current) and the internal states of the battery to estimate SoH.

Data-Driven Methods estimate SoH based the correlation between different aging factors and battery degradation [16]. They utilize machine learning and deep learning algorithms to estimate battery capacity [29].

In general, each approach has inherent strengths and limitations and is often constrained by underlying assumptions. Furthermore, many data-driven approaches decouple the estimation of SoC and SoH and treat them as independent variables without adequately addressing their inherent interdependencies, potentially affecting the accuracy and reliability of the estimations [13].

2.4 Probability Theory, Bayesian Inference and Parameter Estimation

When working with real-world data, there are many uncertainties, such as noisy measurements, operating conditions, etc. **Probability theory** provides a framework for the quantification of such uncertainties.

Events are subsets of the sample space $\mathcal{X}$ that represent outcomes or sets of outcomes of an experiment. **Probability distribution** is a function

$$P : \mathcal{X} \to [0, 1], \tag{2.6}$$

that assigns probabilities to these events, satisfying the axioms of probability.

A **discrete random variable** X is one that has countable outcomes, such as the result of rolling a die or the number of heads in a series of coin tosses. The probability distribution of a discrete random variable is described by a **probability mass function (PMF)** $f(x)$, which assigns a probability to each possible value x of X:

$$f(x) = P(X = x), \tag{2.7}$$

which must satisfy the following two conditions:

- **Non-negativity:** $0 \leq f(x) \leq 1$ for all x.

- **Normalization:** $\sum_{x \in \mathcal{X}} f(x) = 1$, meaning the total probability over all possible values of X is 1 [8].

In **conditional probability**, we can account for information that we already have about our universe or the system to update our expectation. Therefore, the conditional probability $P(A \mid B)$, meaning the probability of A given that B with $P(B) > 0$ occurs is usually written as

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}. \tag{2.8}$$

Bayes' Theorem:
Let A and B be two events such that $P(B) > 0$. Then, the conditional probability of A given B is

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}. \tag{2.9}$$

Proof: From the definition of conditional probability 2.8, we have

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \quad \text{and} \quad P(B \mid A) = \frac{P(A \cap B)}{P(A)}.$$

By solving both expressions for $P(A \cap B)$ and equating the results, we obtain

$$P(A \mid B)P(B) = P(B \mid A)P(A),$$

which leads to Bayes' Theorem.

One primary goal of **statistical inference** is to estimate unknown parameters. We differentiate between frequentist and Bayesian statistical inference. In **frequentist inference**, parameters are treated as fixed but unknown quantities.

In contrast, in **Bayesian inference**, parameters are treated as random variables described by probability distributions. As new information becomes available, these distributions are updated using Bayes' Theorem 2.9 by combining:

- $P(A)$: The **prior probability** – belief in A before seeing data.

- $P(B \mid A)$: The **likelihood** – how likely the data is given A.

- $P(B)$: The **evidence** or **marginal likelihood** – overall probability of the observed data.

- $P(A \mid B)$: The **posterior probability** – updated belief in A after observing B.

Intuitively, we can consider a simple example: suppose you test positive for a rare disease. What is the probability that you actually have the disease?

We can now apply Bayes' Theorem to compute this probability:

$$P(\text{Disease} \mid +) = \frac{P(+ \mid \text{Disease})P(\text{Disease})}{P(+)}, \tag{2.10}$$

This expression uses the prior probability of disease, the likelihood of a positive test result, and the total probability of testing positive.

Bayesian parameter estimation computes the posterior probability distribution of a parameter q given observed data d, according to Bayes' Theorem 2.9

$$\pi(q \mid d) = \frac{\pi(d \mid q)\,\pi_0(q)}{\pi_D(d)}. \tag{2.11}$$

Here, $\pi_0(q)$ is the prior distribution of the parameter, $\pi(d \mid q)$ is the likelihood, and the denominator is the marginal likelihood, which normalizes the posterior $\pi(q \mid d)$.

2.4.1 The Logistic Function

The logistic differential equation model, originally proposed by Verhulst in 1838 [**?**], is commonly used to describe sigmoidal growth. It is defined by the following initial value problem:

$$f'_{\text{gen}}(t) = \frac{r_1}{r_2} f_{\text{gen}}(t) \left(1 - \frac{f_{\text{gen}}(t)}{r_2}\right), \quad f_{\text{gen}}(0) = r_3,$$

with the explicit solution:

$$f_{\text{gen}}(t) = \frac{r_2 r_3}{r_3 + (r_2 - r_3)e^{-r_1 t}}.$$

A simplified form of the logistic function is often used in practice:

$$f_{\text{sim}}(t) = \frac{q_1}{1 + e^{-q_2(t - q_3)}}, \tag{2.12}$$

with parameters

$$q_1 \in \mathbb{R}^+, \quad q_2 \in \mathbb{R}^+, \quad q_3 \in \mathbb{R}.$$

Although both functions describe sigmoidal growth, the parameter mapping is slightly different with:

- q_1 representing the **carrying capacity**, that is: $\lim_{t \to \infty} f_{\text{sim}}(t) = q_1$,

- q_2 controlling the **growth rate**, where larger values lead to a steeper transition,

- q_3 denoting the **inflection point**, the time t at which the function reaches half the carrying capacity q_1.

To fit such models to real-world data, we require a probabilistic framework, which is introduced in the following sections.

2.4.2 The Likelihood Function

In the Bayes' Theorem 2.11, the likelihood is defined as the probability of observing the data d, given some parameter q.

In this research, we assume independent and identically distributed errors that are normally distributed, i.e., $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. This assumption is justified by the fact that the measurement noise in voltage and current sensors is typically unbiased and exhibits approximately constant variance across the measurement range.

The likelihood function for the parameter vector q based on the observed data $\{d_1, \ldots, d_N\}$ is given by

$$\pi(d \mid q) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(d_i - f(t_i, q))^2}{2\sigma^2}\right). \tag{2.13}$$

This can be written as

$$\pi(d \mid q) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{S(q)}{2\sigma^2}\right),$$

where

$$S(q) := \sum_{i=1}^{N} (d_i - f(t_i, q))^2$$

is the sum of squared residuals between the model output and the observed data [10].

2.4.3 The Markov Chain

Definition (Markov Chain):

A sequence of random variables $\{X_n\}_{n\in\mathbb{N}}$ is called a *Markov chain* if it satisfies the *Markov property*, that is, for all $n \in \mathbb{N}$ and all states $x_1, \ldots, x_n, x_{n+1} \in \mathcal{X}$, we have

$$P(X_{n+1} = x \mid X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = P(X_{n+1} = x \mid X_n = x_n).$$

Definition (Transition Matrix)
Let $\{X_n\}_{n\in\mathbb{N}}$ be a Markov chain with a finite or countable state space S. The *transition matrix* P is defined as

$$P_{ij} = \mathbb{P}(X_{n+1} = j \mid X_n = i) \quad \text{for all } i, j \in S.$$

That is, P_{ij} gives the probability of transitioning from state i to state j in one time step.

A Markov chain stationary distribution (also known as equilibrium distribution) is defined by the condition

$$\pi = \pi P,$$

meaning that the distribution remains unchanged after applying the *transition matrix P* [10].

To ensure that a Markov chain has a unique stationary distribution, we need to exclude Markov chains where some states can never be reached, no matter how many steps we take. Such Markov

chains are called reducible. Meaning we only consider **irreducible** Markov chains, where every state can be reached from every other state.

Furthermore, we need to ensure that the Markov chain is **aperiodic**. A chain is aperiodic if all states are aperiodic, meaning there is no fixed cycle length and no states are visited at regular time intervals.

A third criterion is that the probabilities of transitioning from one state to another remain constant over time. Markov chains with this property are called **homogeneous**.

These conditions are required for a Markov chain to admit a unique stationary distribution and to converge to it over time, as stated in the following theorem:

Theorem 1 (Unique Stationary Distribution [10, Theorem 14.15]). *Every finite, homogeneous, irreducible, and aperiodic Markov chain has a unique stationary distribution π. Furthermore, the Markov chain converges in distribution, i.e.,*

$$X_n \xrightarrow{D} X$$

and

$$\mathbb{P}(X = i) = \pi_i \quad \text{for all } i,$$

meaning that the distribution of X_n converges to the stationary distribution π for every initial distribution p_0.

Here, *convergence in distribution* (denoted $X_n \xrightarrow{D} X$) means that the probability distribution of X_n approaches that of the random variable X, which is distributed according to the stationary distribution π.

2.4.4 The Metropolis-Hastings Algorithm

Figure 2.4 visualizes how the Metropolis-Hastings algorithm works for a one-dimensional posterior probability distribution with a Gaussian (normal) shape. It shows how samples/candidates are drawn from a prior distribution and how proposed values are either accepted or rejected based on the likelihood. These accepted candidates are then approximating the posterior distribution.
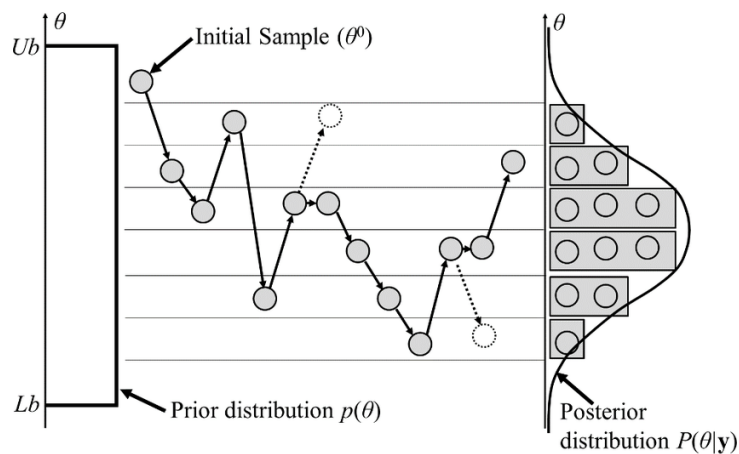

Figure 2.4: Illustration of the Metropolis-Hastings algorithm [15]

Before implementing the algorithm, we need to define the acceptance probability. In the Bayes' theorem (2.11), the integral

$$\int \pi(d|q)\pi_0(q)\,dq \tag{2.14}$$

is used for the normalization. One of the main advantages of the Metropolis-Hastings algorithm is that it avoids calculating this normalization constant directly. This constant, known as the marginal likelihood or evidence, is often intractable in high-dimensional or analytically complex models, as it requires integrating over the entire sample space. The M-H algorithm avoids computing this integral directly by sampling from the posterior distribution instead.

The general formula for the acceptance probability in the M-H algorithm, that defines if the new proposed sample q' should be rejected or accepted is defined as

$$A(q' \mid q_n) = \min\left(1, \frac{\pi(d \mid q')\,\pi_0(q')\,J(q_n \mid q')}{\pi(d \mid q_n)\,\pi_0(q_n)\,J(q' \mid q_n)}\right). \tag{2.15}$$

In this work, the proposal distribution J used to generate candidate capacity values is a **univariate normal distribution**

$$J(q' \mid q_n) := \mathcal{N}(q_n, \sigma^2), \tag{2.16}$$

where q_n is the current capacity estimate and σ is the proposal standard deviation. This choice of proposal is symmetric, meaning

$$J(q' \mid q_n) = J(q_n \mid q'). \tag{2.17}$$

This symmetry implies that the proposal terms cancel in the acceptance ratio, resulting in a simplified acceptance probability

$$A(q' \mid q_n) = \min\left(1, \frac{\pi(d \mid q')\pi_0(q')}{\pi(d \mid q_n)\pi_0(q_n)}\right). \tag{2.18}$$

This step converts the abstract Bayesian inverse problem into a practical Markov-chain procedure that can be executed with only likelihood and prior calculations. We therefore operate on the unnormalized posterior distribution

$$p(\text{posterior}) = p(\text{likelihood})\,p(\text{prior}). \tag{2.19}$$

Because the raw probabilities are extremely small, all calculations are carried out in log-space to prevent numerical underflow

$$\log \pi(q \mid d) = \log \pi(d \mid q) + \log \pi_0(q), \tag{2.20}$$

We then exponentiate the difference of log-posteriors to obtain the ratio of the posterior probabilities

$$\frac{\pi(q' \mid d)}{\pi(q \mid d)} = \exp\left(\log \pi(q' \mid d) - \log \pi(q \mid d)\right). \tag{2.21}$$

Finally, this expression is applied on the acceptance probability formula 2.18

$$\alpha = \min\left(1, \exp\left(\log \pi(q' \mid d) - \log \pi(q \mid d)\right)\right). \tag{2.22}$$

The method and the application of the Bayesian parameter estimation is further explored in Section *Methods* 3.3, where we implement a Metropolis-Hastings Algorithm to estimate the SoH of the battery.

2.5 Description of the Operational Data

This chapter offers an insight into the battery system in operation and a statistical analysis of the key variables obtained from the data.

It begins with a description of the battery system as the source of the acquired data. The following section describes the data structure and highlights the most relevant variables. In Section **Operating Conditions** we present both typical and atypical operational patterns and in the final Section **Data Processing**, the outlier handling.

2.5.1 Dataset

The data analyzed in this Master thesis was collected by Corvus Energy. Corvus offers industry-leading energy storage systems (ESS) [2].

They contain a large portfolio of ESS solutions, however the specific vessel from which the data was collected is called the **„Orca Energy Storage System"**, shown in Figure 2.5. A typical Orca Energy Storage System can have one battery pack or multiple battery packs in parallel, depending on the power and energy requirements of the installation.

Figure 2.5: Orca Energy Air Cooled Battery Pack [2]

Each battery pack contains 14–18 modules connected in series and a module comprises 12 series elements (SEs), where each SE is two NMC cells connected in parallel. The used cells are pouch NMC cells; two in parallel give a nominal capacity of 128 Ah for the series element analysed in this thesis. In the following thesis, when referencing a cell, it actually implies a cell pair with a combined nominal capacity given by the manufacturer of 128 Ah.
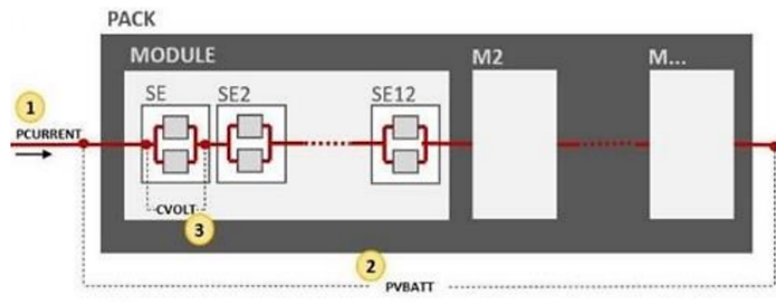

Figure 2.6: Schematic overview of a battery pack [2]

Figure 2.6 shows the pack layout and the measurement points for pack current (PCURRENT), pack battery voltage (PVBATT), and cell voltage (CVOLT).

Description of the Operational Data

The combined dataset collected from the cell contains data from the beginning of April 2020 until the middle of November 2024.

In total, there is approximately five years of operational data, i.e. about 1,705 days. For each day, data, is recorded at one-second interval, resulting in 86,400 entries per day.

For each cell, measurements such as voltage, temperature and SoC are available and these variables along with the units are presented in Table 2.1.

Variable	Description	Unit
PCURRENT	Pack Current	A – Ampere
PVBATT	Pack Battery Voltage	V – Volt
CVOLT	Cell Voltage (over Series Element)	V – Volt
CTEMP	Cell Temperature (for Series Element)	°C – Degrees Celsius
CSOC	Cell State of Charge (for Series Element)	% – Percentage
EventTime	Unix timestamp in milliseconds	

Table 2.1: Description of the operational data variables [2]

2.5.2 Operating Conditions

The data was collected from a vessel operating in Northern Europe, where the battery is exposed to a cold and harsh environment.
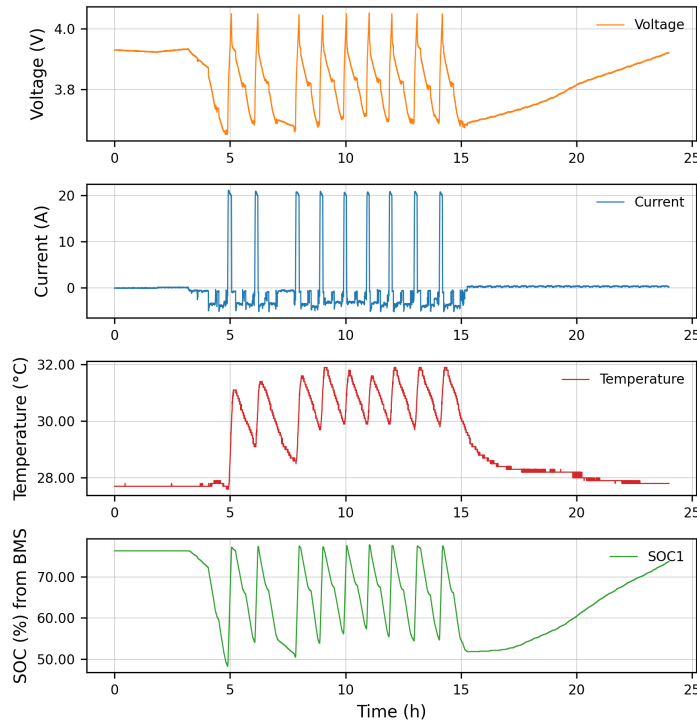

Figure 2.7: Typical day of ship's operation

The pattern illustrated in Figure 2.7 shows a ship's typical day of operation. It consists of charge-discharge cycles during the day and slow overnight charging. The ship typically operates

during the day along a standard defined route, completing approx. 9 trips a day, between 4:00 and 15:00. During the night, the ship is docked and the battery is slowly charged. This consistent operational profile is interesting because it allows for comparisons of battery measurements across different time periods under similar conditions.

This standard operation is generally observed on weekdays. On the weekends, the vessel is idle or under testing. Since SoH estimation requires the battery to be in use, days without battery operation are excluded from the analysis. More details on data processing and outlier handling is in Section 2.5.3.

During the standard operation, the State of Charge (SoC) typically ranges between 40% and 80%. This narrow operating window presents a challenge, as SoC and SoH estimation is more accurate when a wider charge-discharge cycle is available. However, on a few special days – referred to as annual capacity test days, the battery was discharged and charged significantly more than usual. An example of such day is visible in Figure 2.8.
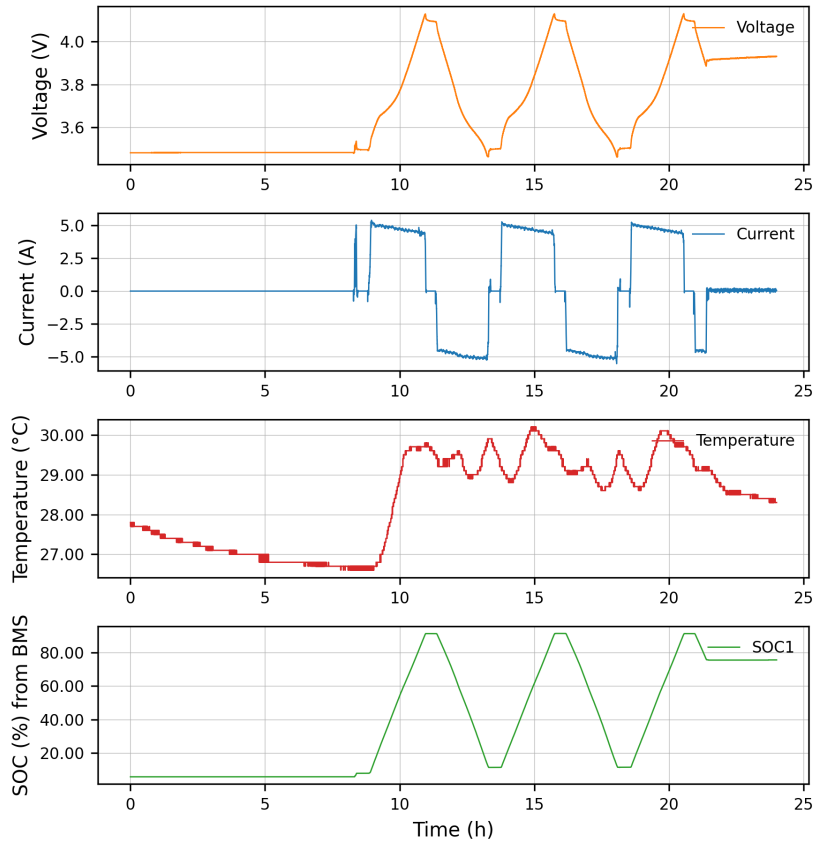

Figure 2.8: Capacity test day

While the specific methodology of these tests, as well as the measured true capacity remains unknown, the behavior of the voltage, current, and temperature channels during these inspections provides valuable insight into the battery health, since the battery was used in wider SoC range compared to the standard operation.

In Figure 2.8, we can observe a controlled charge and discharge behavior from the voltage curve, where the **voltage** reaches peak of approx. 4.1 V, coinciding with an SoC approaching 100%. The **current** profile shows distinct step-like behavior, where battery undergoes alternating charging and discharging cycles. The **temperature** plot provides additional insight into the thermal dynamics of the battery and we can observe that as the battery charges the temperature steadily increases, peaking around 30°C.
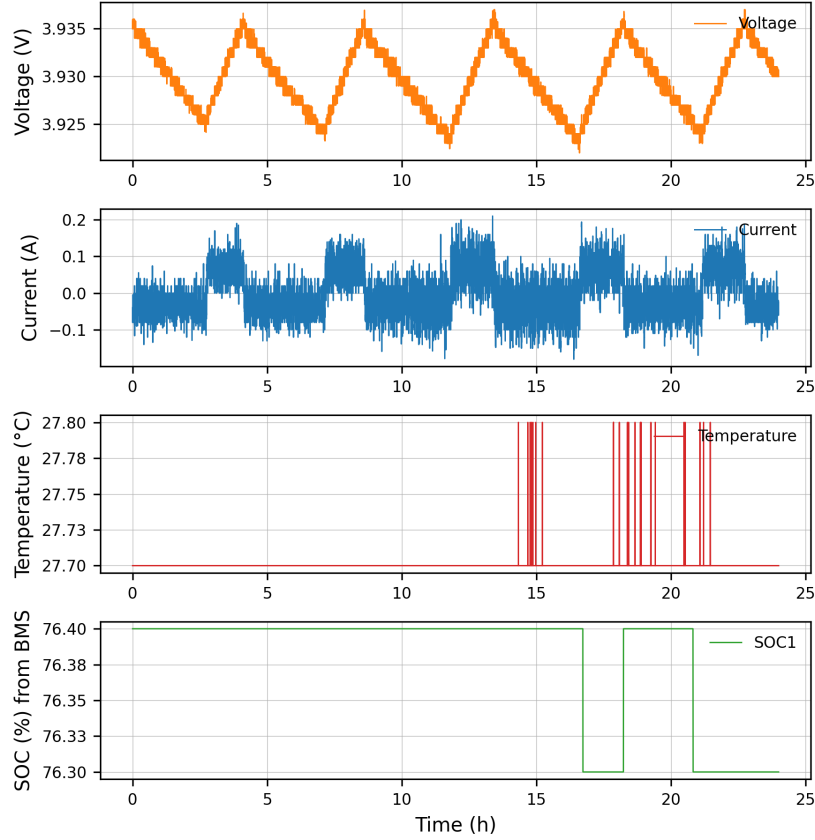
Figure 2.9: Outlier day when ship is idle

Interestingly, while the temperature follows a rising trend during charging, it does not drop as rapidly during discharging, but instead it remains elevated. It seems that the battery retains heat even when it is discharging. This thermal behavior is typical of batteries, where the heat generated during charging is not immediately dissipated. In such cases, the elevated temperature, combined with frequent charging, could indicate potential thermal stress on the battery [18].

In total there are three such special test days over a five year period. In the following thesis, we will use these **capacity test days as reference points to evaluate the SoH estimation methods**, as they provide insights into battery health not observable during regular operation.

2.5.3 Data Preprocessing

The raw data was collected and preprocessed in MDF format with the most relevant variables already extracted and given as in Table 2.1. Table 2.1 already contains the filtered data and includes key parameters derived from the raw dataset.

Only days with normal vessel operation are useful for SoH estimation, so days on which the ship was idle must be marked as outliers and excluded from the estimation. Maritime Battery Management System (BMS) data is especially prone to such artifacts because the sensors work in a harsh environment.

Two common approaches for outlier handling found in the literature are:

- **Statistic-based Method:** compare each value with the expected 1-D distribution, such as modified Kalman filtering. This approach is effective but limited to single-channel data. This limitation arises because statistical methods like Kalman filtering typically

assume univariate signals and rely on probabilistic models that are not easily extendable to multivariate or correlated inputs without significant modification [17].

- **Distance-based Method:** measure how far a point lies from its neighbors in multi-dimensional space by implementing a *k*-nearest-neighbor algorithm [14]. This method handles multiple channels but is computationally heavier because it requires calculating pairwise distances between points, which requires more computational power as the number of data points and dimensions increases.

For the present dataset, a simple application-driven rule is sufficient, where a day is flagged as an outlier if the battery current stays within ± 20 A for the entire 24 h period.

Figure 2.9 shows one such idle day: the current profile is flat and the SoC channel barely moves, confirming that no meaningful capacity information can be detected. Only days that indicate clear current-activity are relevant to the SoH-estimation.

State of Health Estimation Methods

State of Charge (SoC) and State of Health (SoH) are critical parameters that cannot be directly measured. Instead, they must be estimated based on measurable quantities such as current, voltage, temperature, and battery age.

In the following chapter, three different methods for estimating SoH are proposed. The first method is based on the **Coulomb Counting**, where the total charge passed during the day is used to calculate the battery's capacity.

The second approach incorporates a model-based **SoC Observer** that adjusts capacity whenever a systematic voltage error is detected, effectively implementing a combined SoC and SoH estimation.

Finally, the **Metropolis-Hastings** method, frames capacity as an unknown parameter in a Bayesian model where the algorithm samples from the posterior distribution by comparing measured and simulated voltages and returns the most likely capacity together with its uncertainty.

3.1   SoH Estimation Based on Coulomb Counting Methods

The first method combines Coulomb Counting with the Open Circuit Voltage (OCV) method. The idea is to calculate the initial SoC using the OCV curve and then to use the Coulomb Counting formula to continuously track battery capacity changes during operation. Crucial here is to measure OCV while the battery pack is at rest to determine the initial SoC.

The first step is to identify the beginning and the end of daily operation (charging and discharging events). This initial and final point, where battery is at rest and current is zero are called **Rest Periods**. These two points are critical, as they provide stable conditions for using OCV method to calculate the SoC.

The OCV-SoC curves, illustrated in Figure 3.1, are extracted from lab tests on pristine cells at different temperatures. The x-axis represents the OCV of the battery, ranging approximately from 3.0 V to 4.2 V and the y-axis represents the SoC of the battery. The plot includes OCV-SoC curves for six different temperatures (-10°C, 0°C, 15°C, 25°C, 35°C, 45°C), which are differentiated by color.

In particular, the red curve, illustrating voltage behavior at 25 °C is used in this research for mapping voltage from the rest points to the SoC, as it represents the most common operating temperature.
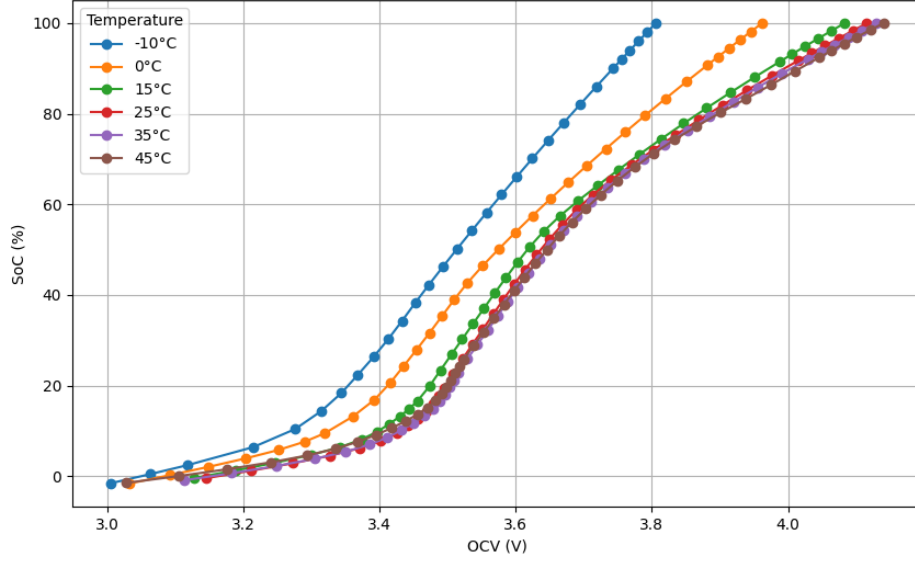
Figure 3.1: OCV-SOC Curves from lab tests at different temperatures

3.1.1   Identifying Rest Periods

Estimating the SoC using OCV is only accurate if the battery is at rest. That is why the critical step is to identify rest points which can be mapped to the SoC. To understand how rest periods are detected, we can observe a typical daily operation illustrated in Figure 3.2, with two plots: one representing the voltage (orange) and the other the current (blue) measurements over the 24-hour period.
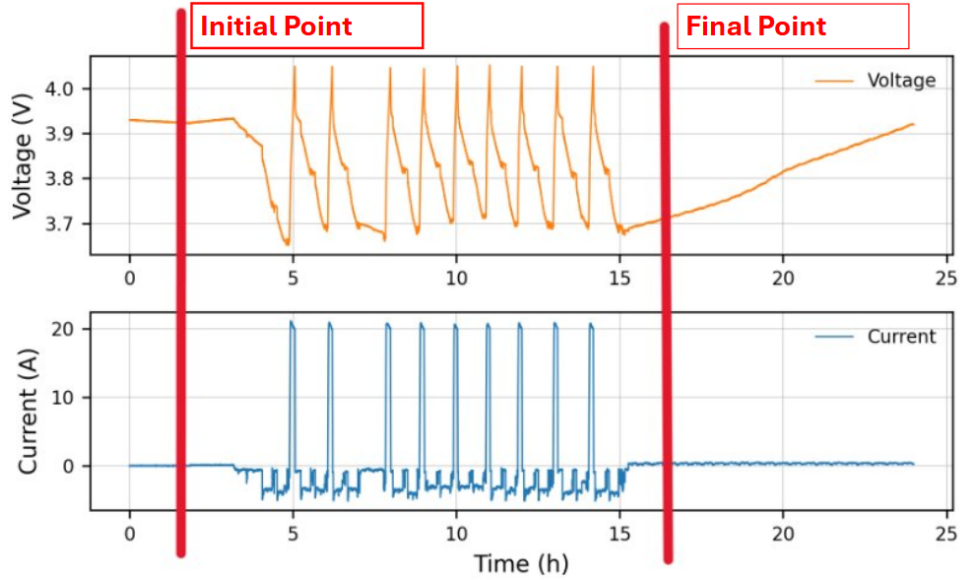

Figure 3.2: Initial and final point for Coulomb Counting

In Figure 3.2, at the beginning of the day, the current is close to zero, indicating the battery is at rest. This is a key period where the initial SoC can be determined using the OCV method. Similarly, another rest period can be observed towards the end of the day, starting at approximately 15 o'clock – the current remains near zero, and the voltage steadily increases. This suggests the battery is undergoing a slow charge.

Such rest periods are crucial for accurately determining the SoC. In this specific case, two points illustrated in Figure 3.2; the initial point at 3:00 and the final point at 15:30, are selected. In

the period between these periods, the difference in voltage is significant, allowing for an accurate capacity calculation.

These points are selected for this particular day and to generalize the method, we have developed the Algorithm 1 that automatically detects the rest periods and allows for different initial and final points, based on the specific voltage and current profiles.

The algorithm first filters out the rows where the current is between the given threshold to make sure that the day is not an outlier as described in Section 2.5. If there is significant activity during the day above the defined threshold – then the rows in which the current is very small and under or equal to 10 Amperes are filtered out and saved in *zero_current_data*. From these rows, we pick two time points with minimum and maximum voltage, which are then set as the initial and final point, respectively.

Algorithm 1: Find initial and final rest points in a day, with the highest voltage difference.

---

1 Function `FindVoltageExtremePointsForOneDay(`*data (current and voltage measurements for one day), current_threshold = 50 A*`)`**:**

2 | **if** $\forall i \in data,\ -current\_low\_threshold \leq current_i \leq current\_low\_threshold$ **then**

3 | | mark day as an outlier

4 | **end**

5 | **else**

6 | | zero_current_data $\leftarrow$ rows in data where $|\text{current\_measurement}| \leq 10\,\text{A}$

7 | | initial_point $\leftarrow$ time of minimum voltage in zero_current_data

8 | | final_point $\leftarrow$ time of maximum voltage in zero_current_data

9 | | **if** *initial_point > final_point* **then**

10 | | | Swap initial_point and final_point

11 | | **end**

12 | | **return** initial_point, final_point

13 | **end**

3.1.2   Calculating Capacity with Coulomb Counting

After identifying the rest points, we map the voltage measurements at the initial and final points to their corresponding SoC values from the OCV curve. By subtracting these SoC values, we obtain the SoC change. Next, we integrate the current over time to determine the total charge passed through the battery during the selected period.

Using the total charge passed and the known SoC change between the two points, we can calculate the battery's capacity.

$$\text{Battery Capacity (Ah)} = \frac{\text{Total Charge Passed (Ah)}}{\text{SoC change (\%)}} \times 100. \tag{3.1}$$

For each day, we map the rest periods and estimate battery capacity using Coulomb Counting. Figure 3.3 highlights the daily estimations over the entire dataset, a four-year period from 2020 to 2024, with the blue dots representing the estimated capacity on a single day.

The x-axis illustrates time from April 2020 through November 2024. The y-axis denotes the battery capacity in ampere hours (Ah). We can observe that most of the estimated capacities are in the approximately 90 Ah to 120 Ah range.

The red X's indicate the test days on which the capacity tests were conducted. The capacity on those days was calculated individually, taking into account the unique charging and discharging patterns during these special days.
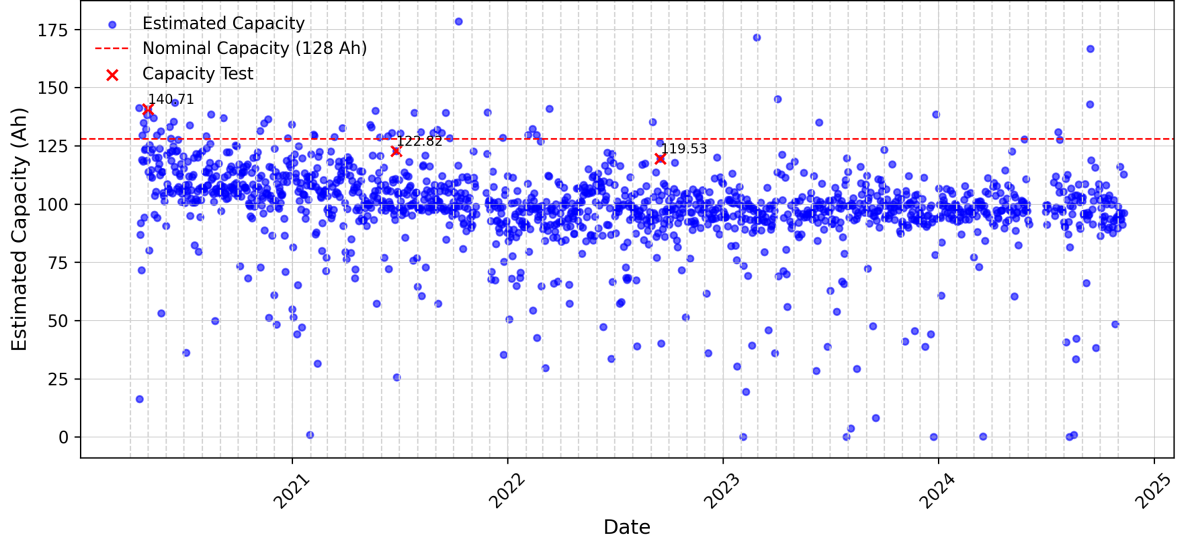
Figure 3.3: Battery Capacity Estimation with Coulomb counting

As can be seen in Figure 3.3, the proposed method gives reasonable results, but the challenge is that due to different operating conditions, the estimated capacity for each day varies significantly. Since it is not expected that the unknown true capacity changes significantly every day, it would make sense to filter out the days on which accurate estimation is not possible and to focus only on those days on which we can reliably estimate the capacity. This is further described in the following section.

3.1.3 Voltage Difference

As described in the previous chapter, we estimate the SoH by calculating the capacity using the Coulomb Counting formula between two points each day. This method assumes that the change in SoC between these two points is significant. Since the voltage at rest is mapped to SoC via the SoC-OCV curve, the voltage difference between the initial and final point correlates with SoC difference. If the voltage difference is close to zero, the SoC difference is minimal, making the denominator in the capacity formula (3.1) near-zero leading to unstable or invalid SoH estimates.

The relationship between voltage difference and battery capacity can be seen in Figure 3.4. Each figure represents one month of operation. The x-axis shows the voltage difference between the initial and final point and y-axis the estimated capacity for a single day of operation.
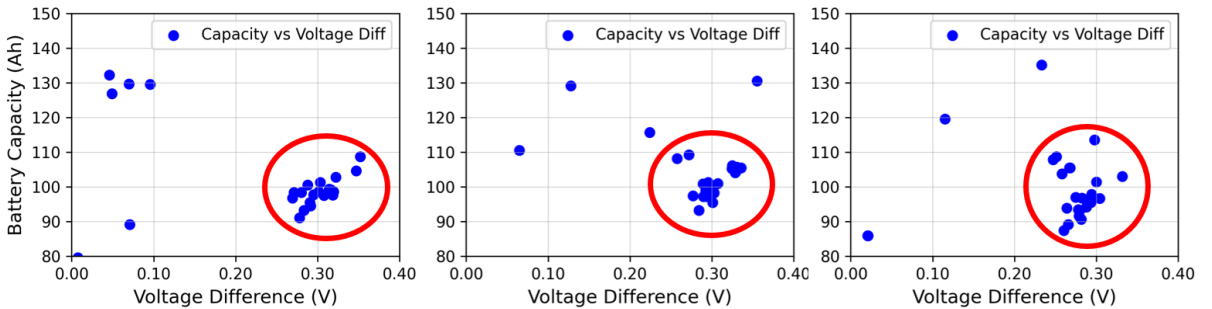

Figure 3.4: Relationship between the capacity and voltage difference

We observe that the capacity estimates derived from Coulomb counting are most reliable when applied over a sufficiently wide State of Charge (SoC) window, which in practice corresponds to

a noticeable voltage difference between the initial and final points.

Figure 3.4 shows over multiple months that when the voltage difference is small — especially below 0.1 V — the estimated capacities vary wildly and often deviate significantly from the expected value. On the other hand, when the voltage difference is larger — typically above 0.2 V — the SoC window is wide enough to make the calculation more stable. In these cases, the estimated capacities tend to cluster more tightly and are more consistent with the expected capacity range.

Since Coulomb counting for SoH estimation works best when there is a substantial change in voltage between the initial and final measurement point, we have taken this information into account and present improved results in Section 4.2.

3.2 Capacity Estimation Using a Combined SoC and SoH Observer

A combined method for SoC and SoH estimation accounts for the intrinsic relationship between SoC and SoH, allowing a more unified and efficient estimation process.

The Observer presented in this section improves the previous approach based on the OCV and Coulomb Counting **by incorporating a dynamic, online framework for SoH estimation.**

A purely Coulomb Counting scheme drifts because of a bias in the current sensor and capacity degradation, while an OCV–SoC look up gives a reference only during rest periods. The coupled Observer presented here improves the previous approach by tackling two distinct error sources:

- short-term SoC drift caused by sensor bias or integration noise,

- long-term capacity mismatch that appears as a persistent bias between predicted and measured terminal voltage.

Figure 3.5 shows the SoC Observer framework. The measured current $i(t)$ is first integrated by the Coulomb Counting block to obtain a provisional State of Charge $\widehat{\mathrm{SoC}}(t)$. This estimate, together with $i(t)$, is fed to the battery model, which maps $\widehat{\mathrm{SoC}}$ to a predicted terminal voltage $\hat{u}(t)$.

The observer forms the *voltage error*

$$e(t) = u(t) - \hat{u}(t), \tag{3.2}$$

where $u(t)$ is the measured terminal voltage. The error signal drives two feedback paths:

- **Fast loop (green in Fig. 3.5):** The error produces a ΔSoC correction that is added to the integrator output, removing short-term drift caused by a sensor bias and noise.

- **Slow loop (blue):** A 24 h average of $e(t)$ represents a *systematic error*. If this long-term mean is non-zero, the present capacity estimate $\widehat{C}$ in the Coulomb Counting denominator is inaccurate. The capacity-update block therefore adjusts $\widehat{C}$ and writes it back into the integrator.

By updating $\widehat{\mathrm{SoC}}$ on fast time-scales and $\widehat{C}$ on slow ones, the observer simultaneously suppresses sensor-induced SoC drift and tracks capacity degradation, providing SoH information without the need for periodic full charge–discharge tests.
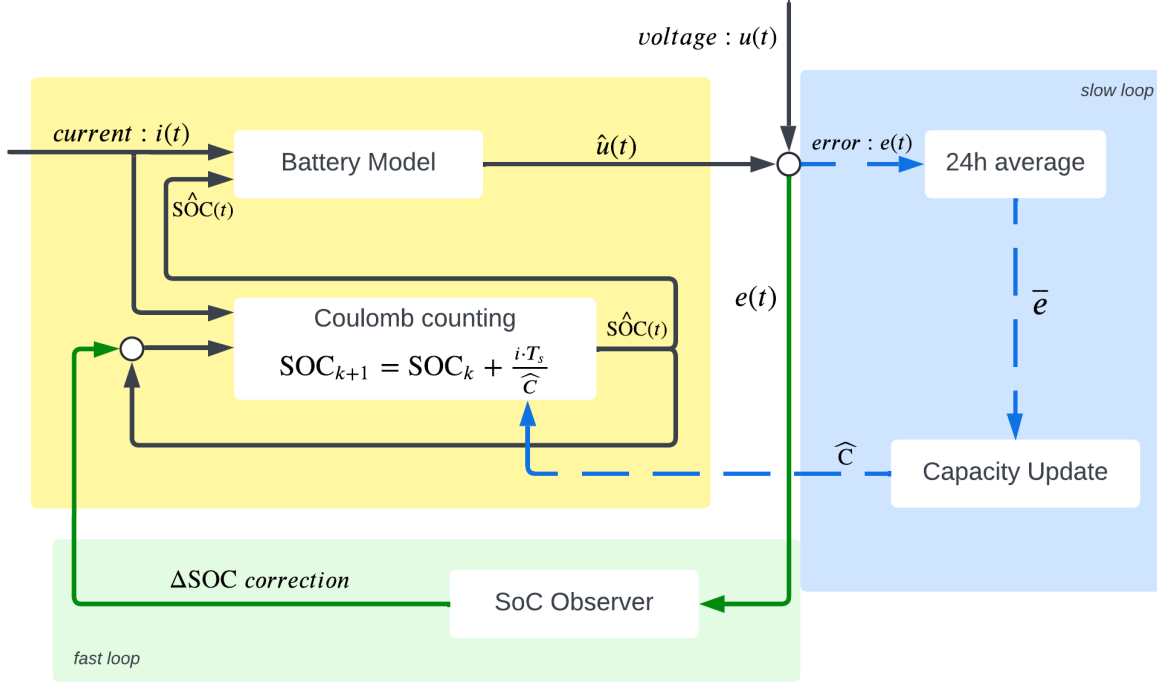
Figure 3.5: SoC Observer

3.2.1   Battery Voltage Model

At the core of the observer lies the voltage model, which predicts terminal voltage based on the current State of Charge and current. This model primarily uses two components:

- **OCV-SoC Relationship:** The OCV-SoC curve (Figure 3.1) maps the battery's open-circuit voltage (OCV) to a corresponding SoC value.

- **Internal Resistance:** The model incorporates a simple internal resistance element.

The battery voltage model 3.3 receives two inputs:

- **Current (i):** The applied current at the current time step, coming from the BMS channel.

- $\widehat{\text{SoC}}$: The SoC estimated via the Coulomb counting, with SoC Correction.

$$u = \text{OCV}(\widehat{\text{SoC}}) + i \cdot R_{\text{int}} \tag{3.3}$$

i is the current and R_{int} is the internal resistance.

Figure 3.6 shows two runs of the same 24 h data set, differing only in the capacity value $\widehat{C}$ used by the Coulomb Counting integrator and the battery model.

In the left-hand column, the integrator assumes **the nominal capacity** $C_{\text{nom}} = 128$ Ah. Because the cell has aged, this value is *too high*. The accumulated SoC (blue curve, bottom-left) therefore drifts upward relative to the BMS SoC (green), and the predicted voltage (blue, top-left) **shows a constant offset from the measured voltage** (orange).

The right-hand column is illustrating **an updated capacity** where the integrator uses the capacity estimate obtained by the observer, $\widehat{C} = 110$ Ah. With the correct capacity the Coulomb
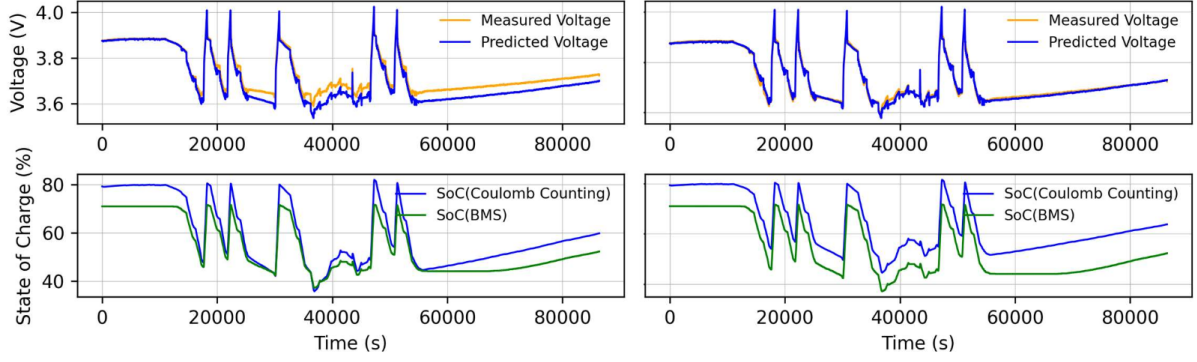
Figure 3.6: Battery Voltage Model with different capacity value $\widehat{C}$ used by the Coulomb Counting

Counting SoC aligns with the BMS SoC, eliminating the systematic voltage error; **the predicted and measured voltages now coincide.**

Thus an incorrect $\widehat{C}$ manifests as a persistent voltage bias, while an updated $\widehat{C}$ removes that bias and validates the model.

The implemented battery model can be categorized as the **Equivalent Circuit Model (ECM)**. However, the Observer framework is modular and this model can be improved and replaced by other more complex ECMs or electrochemical models that may offer different advantages.

The next step is to implement the SoC Observer, where the predicted voltage from the battery model is used to calculate a SoC correction that assists in creating a SoH estimation framework. The Figure 3.6 does not include SoC Correction and Observer yet; this is introduced in the next chapter.

3.2.2 Implementation of the SoC Observer

The full implementation of this concept is visible in Algorithm 2. The Observer iterates over the given data (day, month, year) and in each step, the raw SoC is first estimated via Coulomb Counting *CoulombCountStep* 3. Then, a correction ΔSoC is applied to account for short-term voltage error, resulting in the corrected SoC estimate $\widehat{\text{SoC}}$. This corrected $\widehat{\text{SoC}}$ is then passed into the battery model (3.3) to predict the terminal voltage.

In line 9, the voltage error is calculated as defined in (3.2). The observer then uses this voltage error to compute the SoC correction ΔSoC. By adding this correction to the SoC estimate obtained from Coulomb counting, we obtain the new estimate $\widehat{\text{SoC}}$, which is appended to the SoC list and used as the input to *CoulombCountStep* in line 7 of the next iteration. This process continues until all rows in the input file have been processed.

Algorithm 2: SoC Observer for SoH Estimation

---

1 Function `SoCObserver`(*I_array (A), V_meas (V), capacity (Ah), soc_init (%)*):
2 | soc_list ← [soc_init]
3 | voltage_predictions ← [V_meas[0]]
4 | soc_corrections ← [0]   // No correction for first step
5 | **for** $i \leftarrow 1$ **to** *length(I_array)* -1 **do**
6 | | current ← I_array[$i-1$]
7 | | new_soc ← `CoulombCountStep`(soc_list[-1], current, capacity)
8 | | V_pred ← battery_model.calculate_voltage(new_soc, current)
9 | | voltage_error ← V_meas[i] − V_pred
10 | | soc_correction ← observer.get_soc_correction(voltage_error)
11 | | new_soc ← new_soc + soc_correction
12 | | Append(new_soc, soc_list)
13 | **end**

Algorithm 3: Compute updated SoC from current and capacity

---

1 Function `CoulombCountStep`(*current_soc (%), current (A), capacity (Ah)*):
2 | new_soc_increment ← $\left(\frac{\text{current} \cdot \left(\frac{1}{3600} \right)}{\text{capacity}} \right) \cdot 100$
3 | updated_soc ← current_soc + new_soc_increment
4 | **return** updated_soc

Figure 3.6 presents one full day of operation in which the observer starts with a deliberately *wrong* initial state: the integrator is set to SoC = 50 % while the true SoC (and therefore the open-circuit voltage) is closer to 80%. The Coulomb Counting block 3 is in this case referred to as an integrator because it accumulates the current over time to estimate the total charge change, which directly corresponds to the SoC. The factor 1/3600 converts current from amperes into ampere-hours, and multiplying by 100 expresses the resulting SoC as a percent.

The goal of Figure 3.6 is to show that the observer can eliminate the short-term voltage error by correcting SoC.

We introduce the SoC correction (ΔSOC) in each step of our Observer that corrects the SoC to match the measured voltage. The correction step is a proportional gain that adjusts the estimated SoC based on the difference between the measured and predicted voltage. If the predicted voltage is lower than the measured voltage, SoC is increased slightly to better align the model output with the observation, therefore:

- In the first step: $\Delta SOC = 0$.

- For each row:
$$\Delta SOC = correction\ step \times voltage\ error. \tag{3.4}$$

The top graph in Figure 3.6 shows the predicted voltage (blue) rising rapidly to meet the measured voltage (orange) after the initial mismatch. This convergence is driven by the proportional update ΔSoC as given by 3.4. Here we intentionally use a relatively large correction step so the effect is visible within a single day; in normal operation smaller gains yield a smoother trajectory. This update represents the fast (green) loop in the Observer framework Figure 3.5.

In addition to the SoC correction, the observer should dynamically adjust capacity C to reflect the battery's actual State of Health (SoH). If the degradation is not reflected in the model, the SoC estimation will be biased, leading to persistent voltage prediction errors.
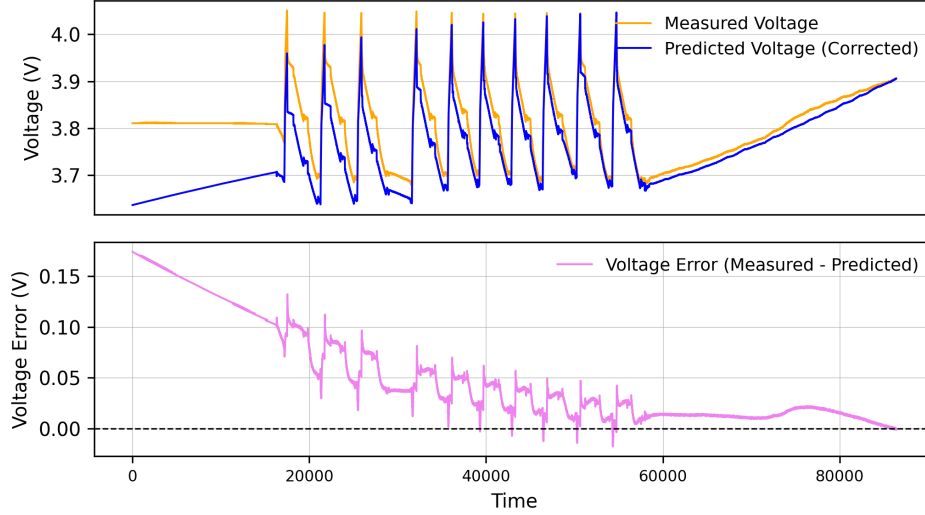
Figure 3.7: Voltage Convergence over time with SoC Observer

Therefore, we need an approach to adjust capacity during the simulation or real-world operation based on the information from the SoC Observer. Traditionally, Kalman filters are used to estimate SoC and SoH in real time by treating capacity as a time-varying state [25] [33]. However, in this research, we are using the cascaded SoC Observer with capacity adjustment based on the Minimum Root Mean Squared Error (RMSE) between the predicted and measured voltage. With this approach, we aim to address the slow loop where $\widehat{C}$ is being corrected.

3.2.3 RMSE Evaluation Across Capacity Range

We can observe that in the Coulomb counting formula 2.2, capacity (C) in the denominator has an effect on the overall SoC estimation and hence the correctness of the estimated $\widehat{\text{SoC}}$ depends on the assumed battery capacity. Over time, as the battery degrades due to aging, cycling, and temperature effects, the capacity estimate in the Coulomb counting should be adapted as well.

If the capacity C used in the Coulomb counting formula does not reflect the actual, aged capacity, SoC estimation becomes progressively inaccurate. When SoC is wrong due to an incorrect capacity assumption, $\hat{u}$ will deviate from the true measured voltage u. Voltage Error, defined as (3.2) in that case becomes systematic rather than random.

To identify the capacity that best matches each day's data, we solve an optimization problem where the cost function is the root-mean-square error (RMSE) between the predicted voltage $\hat{u}(t)$ and the measured voltage $u(t)$, as defined in (3.2).

1. **Initialise SoC.**
   On the *first* day, SoC_0 is set from the morning OCV value. On every subsequent day the simulation starts from the SoC that ended the previous day, ensuring a continuous trajectory.

2. **Grid-search candidate capacities.**
   A grid search is performed over candidate capacities C_i to estimate the true battery capacity.

   On the *first day*, a global search is performed using a uniform grid from 90 Ah to 150 Ah, with a spacing of 1 Ah. This results in 61 candidate capacities

   $$C_i \in \{90, 91, 92, \ldots, 150\} \text{ Ah}.$$

On **subsequent days**, a local search is performed around the previous day's optimal capacity C^*_{prev}, with a spacing of 1 Ah, resulting in 3 candidate capacities

$$C_i \in \{C^*_{\text{prev}} - 1, \ C^*_{\text{prev}}, \ C^*_{\text{prev}} + 1\} \text{ Ah.}$$

For each trial capacity C_i, we then:

2.1. Integrate the current to obtain $\widehat{\text{SoC}}_{C_i}(t)$.

2.2. Predict the voltage $\hat{u}_{C_i}(t)$ with the battery model (3.3).

3. **Compute cost function and choose $\widehat{C}$.**

$$\text{RMSE}(C_i) := \sqrt{\frac{1}{N} \sum_{k=1}^{N} (u[k] - \hat{u}_{C_i}[k])^2} \tag{3.5}$$

The capacity value C_i that minimizes this cost, i.e.

$$\widehat{C}_d := \arg\min_{C_i} \text{RMSE}(C_i),$$

is taken as the capacity estimate for that day. Figure 3.8 illustrates the approach for two grid resolutions.

4. **Compute daily SoH.**
Use Equation (2.4) with $\widehat{C}_d$.

Figure 3.8: RMSE Cost Function of Voltage Errors with different capacity range

This grid-search strategy supplies a robust daily capacity estimate without the need for gradient information or full charge–discharge calibration.

In Section 4.3, this approach is applied to the entire dataset.

3.3   Probability-Based SoH Estimation

So far, we have analyzed the battery data and estimated a single best value for the capacity, which hides how sensitive that estimate is to sensor noise, model mismatch, or data sparsity. That is why estimating SoH with probability-based method can be beneficial, because it can give us a confidence interval to assess how reliable the battery model actually is.

Furthermore, RMSE does not use priors, whereas Bayesian methods incorporate prior knowledge about the data into the result. A Bayesian approach returns a full posterior giving a built-in

confidence interval that can stabilize the estimate by folding in prior knowledge such as the nominal capacity or previous day's estimation.

The following section utilizes **Markov-chain Monte Carlo (MCMC)**, a numerical method to approximate Bayesian posterior distributions, for SoH estimation. This approach is implemented in scope of this thesis in a Metropolis-Hastings Algorithm.

We now apply Bayesian inference to the problem of SoH estimation, where the unknown parameter q is the battery capacity C, and the observed data d corresponds to voltage error V_{error}.

- $\pi(q \mid d)$: **Posterior distribution** is the updated battery capacity $q_{\text{posterior}}$ after observing the data V_{error}.

- $\pi_0(q)$: **Prior distribution** is the last estimated capacity or initial capacity q_{prior}.

- $\pi(d \mid q)$: **Likelihood** is the probability of observing data d (e.g., voltage error) given a specific capacity q. It reflects how well the current capacity explains the observed data.

- $\pi_D(d)$: **Normalization factor**.

Essentially, we are estimating the posterior probability distribution of the battery capacity q given observed data d (e.g., voltage error V_{error}).

Applying Bayesian method to a parameter estimation problem has an advantage of providing information on how spread out the probability distribution of the parameter is. If the distribution is peaked and localized around a certain value, the parameter can be calculated precisely. After calculating the probability distributions, we can easily find the **credible intervals**, which estimate the parameter by specifying the range of possible values.

To illustrate how this approach works, we have constructed a generic example with a simple logistic function, similar to the example in the book [10]. To implement the Metropolis-Hastings Algorithm, we first introduce **The Logistic Function**, **The Likelihood Function** and **The Markov Chain**. These core concepts are necessary to understand the implementation of the algorithm.

The Logistic Function for Synthetic Example

In this thesis, we adopt a simplified form of the logistic function for synthetic data generation and parameter estimation 2.12. The result of the synthetic measurements generated by this function is illustrated in 3.9.

Later in this thesis, we will apply the same logistic framework to operational data. In that context, the red curve (representing the true function in synthetic data) is replaced by the **measured voltage**, serving as the ground truth. The estimated voltage will be plotted for multiple candidate capacities, each yielding a separate curve. This illustrates how deviations in the estimated capacity impact the predicted voltage trajectory over time.

The Likelihood for Capacity Estimation

In the context of this thesis, the likelihood function quantifies how well a specific capacity value explains the observed battery data. Specifically, it represents the probability of observing the **voltage error** (3.2) for a given capacity estimate.

In (2.11), the likelihood is combined with the prior $\pi_0(q)$ to update the posterior $\pi(q \mid d)$. It determines how strongly new data (voltage measurements and their error V_{error}) influence the posterior. If the likelihood is high, the measured data fits well to the proposed capacity q,
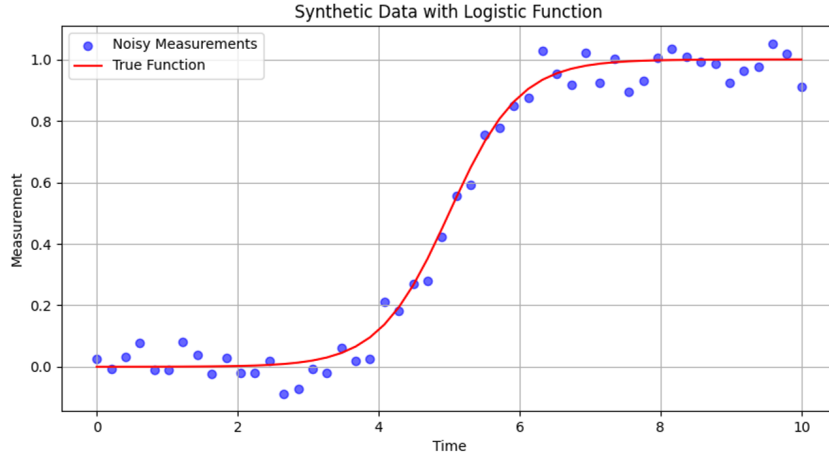
Figure 3.9: Logistic Function and Synthetic Measurements

indicating a plausible estimate. Low likelihood suggests poor agreement between the data and the proposed capacity.

The Markov Chain

In the following Metropolis-Hastings algorithm, we generate a Markov chain over capacity values by repeatedly performing the following steps (algorithm steps are defined more detailed in the Metropolis-Hastings Algorithm Implementation Section (3.3.1)):

1. We start with initial guess for capacity.

2. Propose a new value based on the current one.

3. Accept or reject it based on the likelihood.

The resulting sequence of accepted values forms a Markov chain. Ideally, this chain initially explores the parameter space and subsequently converges toward the most probable State of Health (SoH) value. Once the chain reaches its stationary distribution, the samples accurately represent the Bayesian posterior distribution.

To demonstrate this mechanism, we implement a generic textbook example from the book [10], which is visualized in Figure 3.10. Two variants are shown:

- **Typical MCMC (left):** a standard Random Walk MCMC with fixed Gaussian proposals

- **Converging MCMC (right):** a numerically stabilized variant that uses log-likelihood and tuned proposal distributions.

Figure 3.10 is a *single-parameter toy example*: the Metropolis–Hastings chain is allowed to vary only q_2 while the other model parameters q_1 and q_3 are kept fixed at their true values. This one-dimensional setup makes the behavior of the algorithm easy to see: the accepted samples (black trace) hop around the red dashed line, which marks the true value of q_2 and the lower figure shows the resulting posterior histogram together with its mean, median, and MAP estimates.

The mechanism is exactly what we need for capacity tracking. Having introduced the core concepts with a generic example, we now apply the same Metropolis-Hastings algorithm to the battery capacity estimation task, where the unknown variable is the *battery capacity C* instead of q_2.
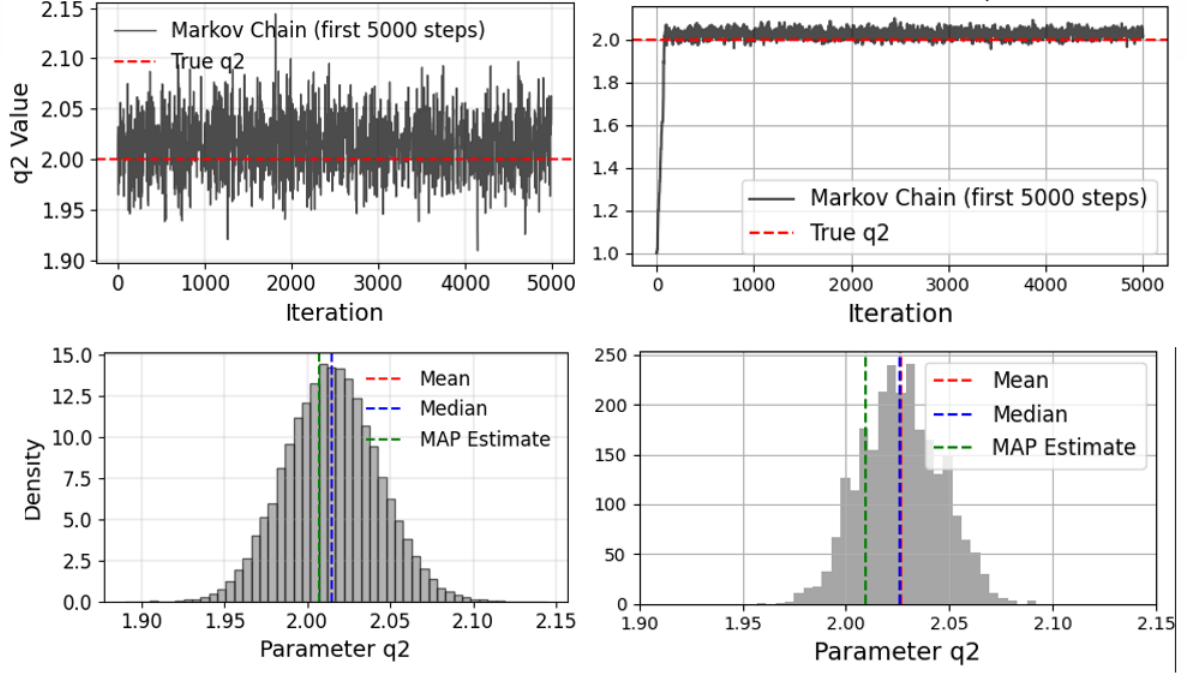
Figure 3.10: Textbook MCMC and Converging MCMC Example

3.3.1 Metropolis-Hastings Algorithm: Implementation

Pseudocode 4 presents a custom implementation of the Metropolis-Hastings algorithm, tailored to the problem of capacity estimation using daily voltage and current data. While it follows standard MCMC structure, it is adapted for the use of log-posterior evaluation to improve numerical stability. We iterate over the entire dataset and execute Algorithm 4 for each day, hence each algorithm call returns a result for a single day.

The core computation is handled by the function `LogPosterior`, which returns the sum of `LogLikelihood` and `LogPrior`. We insert the variable C representing the battery capacity into the Formula 2.21

$$\frac{p(C' \mid \mathbf{d})}{p(C \mid \mathbf{d})} = \exp\left(\log p(C' \mid \mathbf{d}) - \log p(C \mid \mathbf{d})\right) = \exp\left(\text{log\_post\_proposed} - \text{log\_post\_current}\right),$$
(3.6)

which is applied in lines 11 and 12 of the algorithm for both the proposed and current capacity values.

In line 14, the acceptance probability determines whether the proposed capacity value is accepted or rejected and is computed as given in the Formula

$$\alpha = \min\left(1, \exp(\text{log\_post\_proposed} - \text{log\_post\_current})\right).$$
(3.7)

The following pseudocode summarizes the full algorithm for one day's data:

Algorithm 4: Metropolis-Hastings algorithm for capacity estimation.

---

1 Function MetropolisHastings(*capacity_init (Ah), iterations, proposal_std (Ah), V_meas (V), I_array (A), soc_init (%), capacity_nominal (Ah)*):

2 | samples ← [capacity_init]

3 | accepted ← 0

4 | **for** $i \leftarrow 1$ **to** *iterations* **do**

5 | | current_estimate ← samples[-1]

6 | | proposed ← sample from $\mathcal{N}$(current_estimate, proposal_std)   // Propose new capacity

7 | | **if** *proposed $\leq$ 0.5 · capacity_nominal* **or** *proposed > 1.2 · capacity_nominal* **then**

8 | | | Append current to samples   // Reject out of bounds proposals

9 | | | **continue**

10 | | **end**

11 | | log_post_current ← logPosterior(current_estimate, V_meas, I_array, capacity_nominal, soc_init)

12 | | log_post_proposed ← logPosterior(proposed, V_meas, I_array, capacity_nominal, soc_init)

13 | | log_diff ← log_post_proposed − log_post_current

14 | | accept_prob ← $\min(1, \exp(\text{log\_diff}))$

15 | | **if** *uniform(0, 1) < accept_prob* **then**

16 | | | Append proposed to samples   // Accept proposed capacity

17 | | | accepted ← accepted + 1

18 | | **end**

19 | | **else**

20 | | | Append current to samples   // Reject and re-use current capacity

21 | | **end**

22 | **end**

23 | **return** samples, accepted / iterations

On the *first* day we set `capacity_init` to the nominal value and `SoC_init` from the OCV rest point. For every subsequent day, the chain starts from the previous day's capacity estimate, allowing the posterior to evolve smoothly with ageing.

The other input arguments of Algorithm 4 are:

- `iterations`: total samples to draw; longer chains give a finer posterior but increase run time.

- `proposal_std`: standard deviation of the normal proposal kernel (2.16); here fixed at 0.5, which sets the average step size in capacity space.

- `V_meas, I_array`: measured voltage and current channels. Current drives Coulomb counting (3); predicted voltage V_{pred} is compared with V_{meas} in the likelihood.

- `SoC_init`: initial SoC mapped from the OCV rest point on day 1 and thereafter taken from the previous day's final SoC.

- `capacity_nominal`: $C_{\text{nom}} = 128$ Ah.

Now that the Metropolis-Hastings algorithm and all necessary model components (likelihood, prior, SoC correction, voltage prediction) have been defined, we are ready to apply this method to synthetic battery data. This synthetic test case allows us to validate the approach under controlled conditions where the true capacity is known with the results shown in Section 4.4.

Results and Discussion

This chapter presents the results of the SoH Estimation with different approaches. First we will talk about reference values and comparison metrics with which the results are evaluated. Then we will present the results of all three methods separately. Subsection 4.5 *Comparison of Estimation Methods* shows a direct comparison of the results obtained from all three methods. Finally, the different assumptions and ideas behind each of the method, as well as the advantages and disadvantages of each method are discussed.

4.1 Evaluation Metrics

When comparing different methods, it is important to understand how each method is evaluated and how the results can be compared. The evaluation focuses on how well the estimated capacities reflect true battery degradation over time. Therefore, the ability of the method to **track trends in battery degradation** is very crucial.

Another important factor is whether the method is sensitive enough that the smaller changes in the voltage are detected, but at the same time stable enough to avoid random fluctuations on a day to day basis.

In routine operation the battery cycles through a narrow SoC range, providing limited information and making precise SoH estimation challenging.

However, we have information about three special days on which **the capacity tests** were performed and the battery operated in a wider range. For those days, we only have the raw voltage-and-current logs and not the measured capacity values. Because the the vessel was cycled over a much wider SoC range during the tests than during standard operation, there is more information in the data. We therefore treat the model's capacity estimate on those test days as our most accurate reference and mark them with red crosses in the subsequent result graphs.

The nominal reference capacity (128 Ah) is also used as a baseline for comparing capacity over time.

In summary, the methods are evaluated based on their ability to track degradation trends, **the stability over time, sensitivity to operational data** and noise, and how closely the estimated results align with reference values from capacity test days.

4.2   SoH Estimation Based on Coulomb Counting Methods
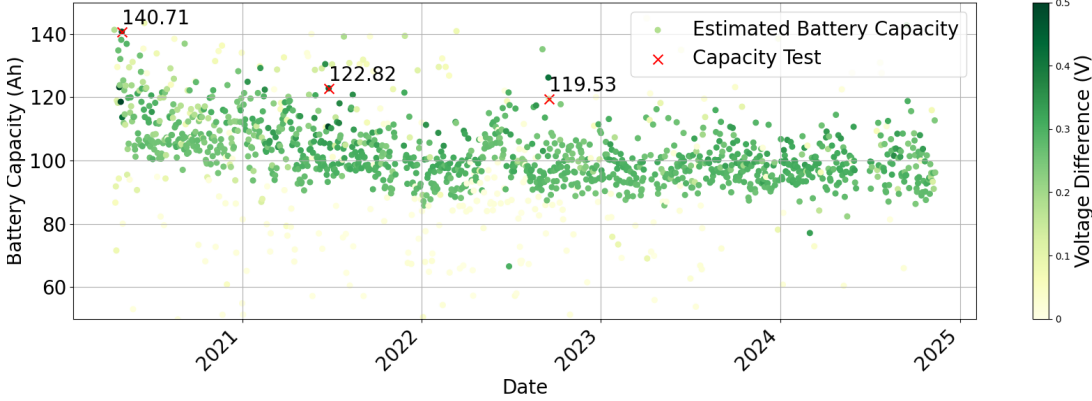

Figure 4.1: Battery Capacity Over Time with Voltage Difference

In Section 3.1, we have introduced the Coulomb Counting method used for day-by-day estimation of capacity. We have established a correlation between the voltage difference between the two rest points in a day and accuracy of capacity estimate. The previously shown result in Figure 3.3 has been improved in Figure 4.1 by incorporating information about voltage difference. For each capacity estimate for a day, represented as a dot, the color intensity that corresponds to the voltage difference between two rest points has been added. Darker green color indicates higher voltage difference, and lighter green/yellow points indicate a lower voltage difference, as visible in the color bar on the right.

The plot includes three red 'X' markers, which highlight specific days on which capacity tests are performed. The estimations for capacity test days are separately calculated and considered to be more accurate due to the wider range of the battery operation. We observe that **capacity values for test days** (140.71 Ah, 122.82 Ah, 119.53 Ah) **are significantly higher than the trend.**

The behavior visible in Figure 4.1 mirrors the typical battery aging trajectory: an initial rapid capacity decline followed by a slower tapering as internal processes stabilize [32], which is **consistent with the expected degradation trend** shown in Figure 2.2, where capacity decreases gradually over time until the knee point is reached.

4.3   Capacity Estimation Using a Combined SoC and SoH Observer

As described detailed in the *Methods* Section (3.2), a simulation with the SoC Observer with an SoC Correction is executed for every day with some range of capacities.

Figure 4.2 shows such evaluation over a period of one month. Every day, the SoC Observer runs with 15 different capacity values and the optimal capacity is selected based on the minimal voltage error.

The results for September 2022 in Figure (4.2) show the estimated capacity for each day, as well as the capacity estimate on the test day marked with a red 'X' marker. We can observe that the capacity estimation of the test day does not deviate significantly from the other estimations of the days before and after.

Figure 4.3 illustrates this evaluation with the SoC Observer and RMSE Optimization over the entire dataset. In this case, we started the evaluation for the initial day with 30 capacities and
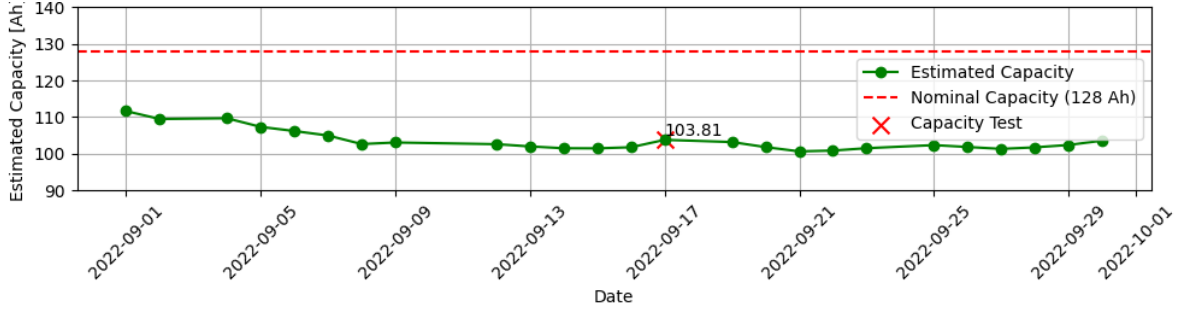
Figure 4.2: SoH Estimation with SoC Observer and RMSE optimization for September 2022

every subsequent day, the SoC evaluates three values closest to the optimal capacity of the previous day. The capacity with the lowest voltage error is selected as the optimal capacity for that day and plotted.
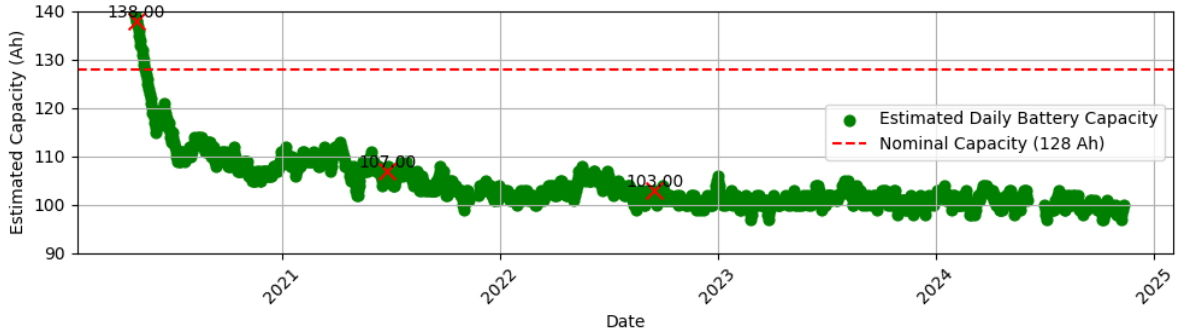

Figure 4.3: SoH Estimation with SoC Observer and RMSE optimization for the entire dataset

Similar to the results with the previous method, a **degradation trend can be observed** with this approach as well.

4.4 Probability-Based SoH Estimation

We have implemented a Metropolis-Hastings Algorithm to generate samples from a probability distribution and utilized a Bayesian parameter estimation approach to obtain the results presented in this section. Since we have implemented a generic example to show how the algorithm works, in the following paragraphs, we will apply the algorithm on the **Synthetic Data** and finally on the **Operational Data**.

4.4.1 Simulation with the Synthetic Data

To be as close as possible to the operational data, the synthetic data for this simulation was created from the real data from the current channel of one regular operation day illustrated in Figure 2.7. For the purpose of simulation, we used around 30,000 samples (half of the day) of the data.

We **assumed a true known capacity**, which was used together with the current channel to compute SoC via Coulomb counting. Consequently, we used the battery model introduced in Section 3.2.1 to calculate the voltage and added Gaussian noise with a standard deviation of 0.01 V.

We then simulated the results using the Metropolis-Hastings Algorithm 4 for two different values of true capacity. The results are shown in Figure 4.4, which includes plots of predicted versus

measured voltage and the corresponding Markov Chains. The true capacity in the left plots (purple border) is 110Ah, and in the right plots (blue border) it is 100Ah.

Figure 4.4 shows the *Voltage Prediction vs. Measured (with noise)* comparison, illustrating both the measured (synthetic) voltage and the predicted voltage from the battery model. In both cases, the Markov Chain converges close to the true capacity, and the predicted voltage accurately tracks the measured (synthetic) voltage.
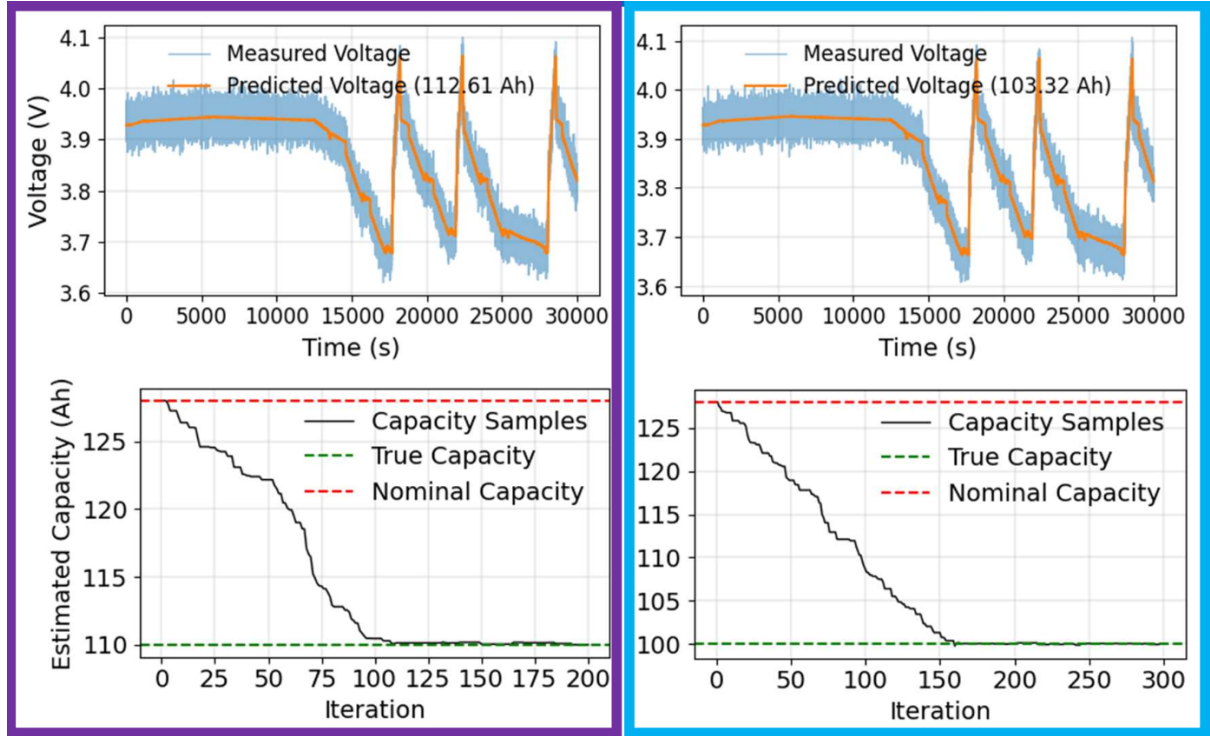

Figure 4.4: Probability-based estimation with synthetic data different true capacities

For the second experiment with synthetic data, some variation was added to test the algorithm on differently shaped data:

- increased *proposal_std* to 2.0 (2.16),

- reduced sample size to 2000,

- 1000 algorithm iterations.

Even with this variation shown in Figure 4.5, shows a Markov chain with fast convergence to the true capacity, similar to the previous results. Since we added more iterations, after around the 100th iteration, the Markov Chain is close to the true value. Similarly to the previous results, the predicted voltage also accurately follows the measured voltage.

Furthermore, we can observe that the posterior distribution is narrow and peaked, which indicates that the confidence in the capacity is high. It suggests that the estimate is stable and since the Markov Chain merges to the true capacity, it indicates that the posterior accurately reflects the simulated data.

Now that we have simulated an example with synthetic data, the next section shows the results when applying the Metropolis-Hastings Algorithm on the operational data.
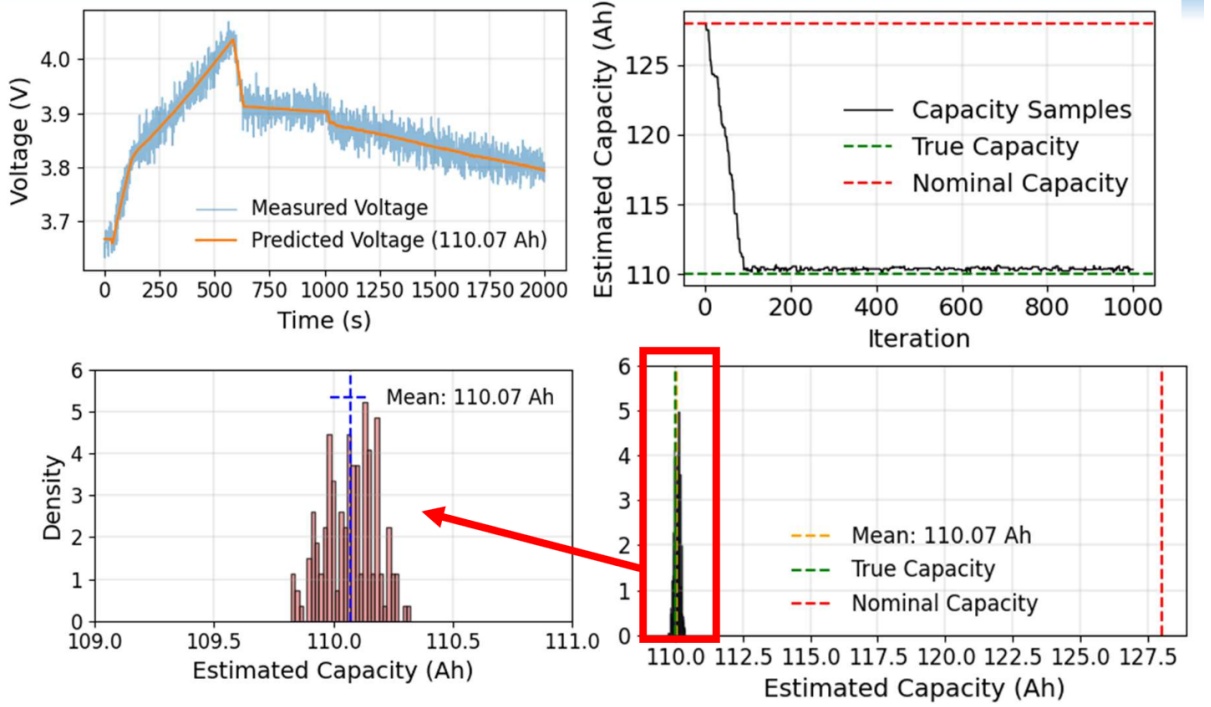
Figure 4.5: Probability-based estimation with synthetic data and reduced sample size

4.4.2 Simulation with the Operational Data

The next step is to apply this method on the operational data. The same day from which current channels data was used to generate synthetic data is visible in Figure 4.6; this time with operational data. Although we do not know the true capacity, as we had with the synthetic data, we can observe a similar trend in the **Voltage Comparision** and **Markov chain** (4.6). In the left figure, the estimated voltage is overlapping with the measured voltage and the right figure shows a Markov chain converging to the estimated capacity.

Figure 4.6: Probability-based SoH Estimation with Operational Data and 3000 Samples

As the previous experiment showed that Metropolis-Hastings is capable of estimating the capacity on the operational data, we want to complete our framework by adding the SoC Observer. The SoC Observer is beneficial for the SoC correction based on the voltage error that reduces the Coulomb Counting drift that may arise due to the sensor noise. The results of M-H with SoC Observer are illustrated in the following section in the three different periods: one day, one month and the entire dataset.

For the M-H algorithm execution of **one day** we have chosen the capacity test day in May 2020. Figure 4.7 shows again on the left side voltage plot and on the right side Markov Chain with the proposed capacity samples.
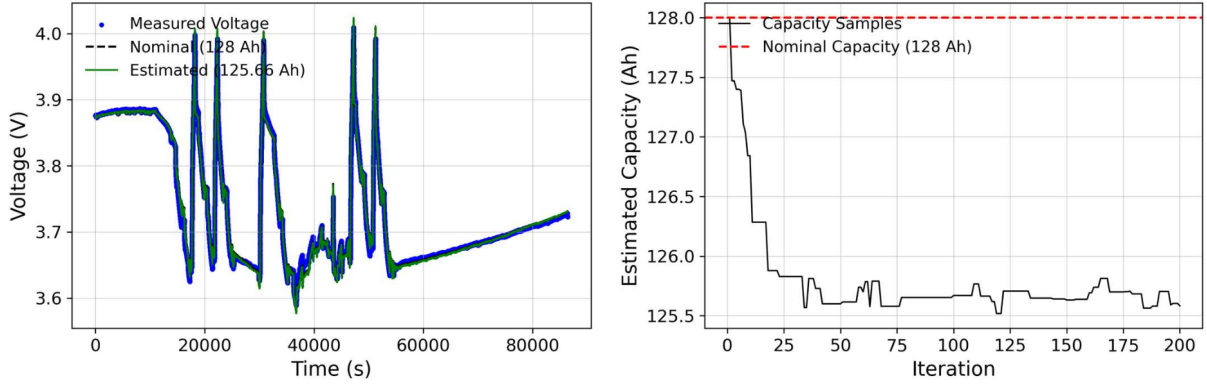

Figure 4.7: One Day: Bayesian SoH Estimation with Operational Data

The results of the Bayesian method for the September 2022 are visible in Figure 4.8, where the starting capacity is 110 Ah. We can observe that the estimated capacity by Metropolis-Hastings algorithm and the SoC observer in the previous method is almost identical for the capacity test day on 17th September 2022.

Figure 4.8: Bayesian SoH Estimation with Operational Data over one month

A Metropolis-Hastings execution over the entire dataset is shown in Figure 4.9. Initial capacity is 128 Ah and standard deviation for the proposals 0.5. Important to mention is that for the first day the number of algorithm iterations is 50 and every subsequent day, the algorithm iterations are set to 10. This result is very similar to the result from the previous method (4.3).
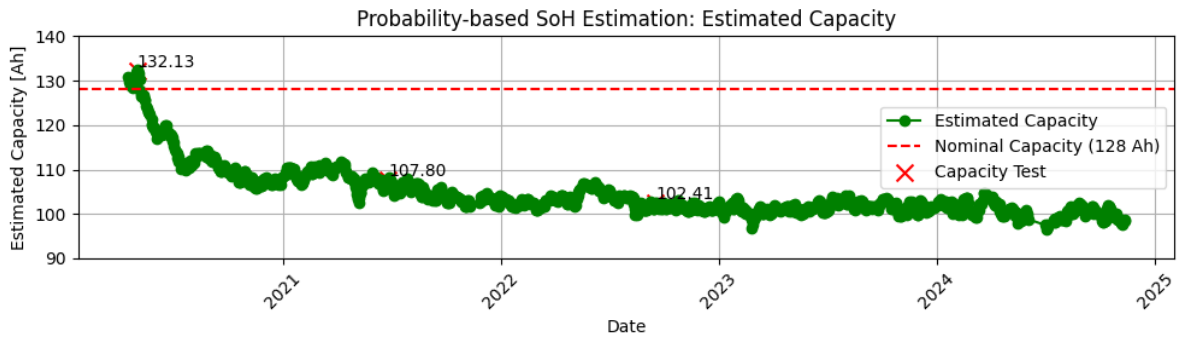

Figure 4.9: Bayesian SoH Estimation with Operational Data

4.5 Comparison of Estimation Methods

Figure 4.10 shows a visual comparison of all three approaches in a single illustration.
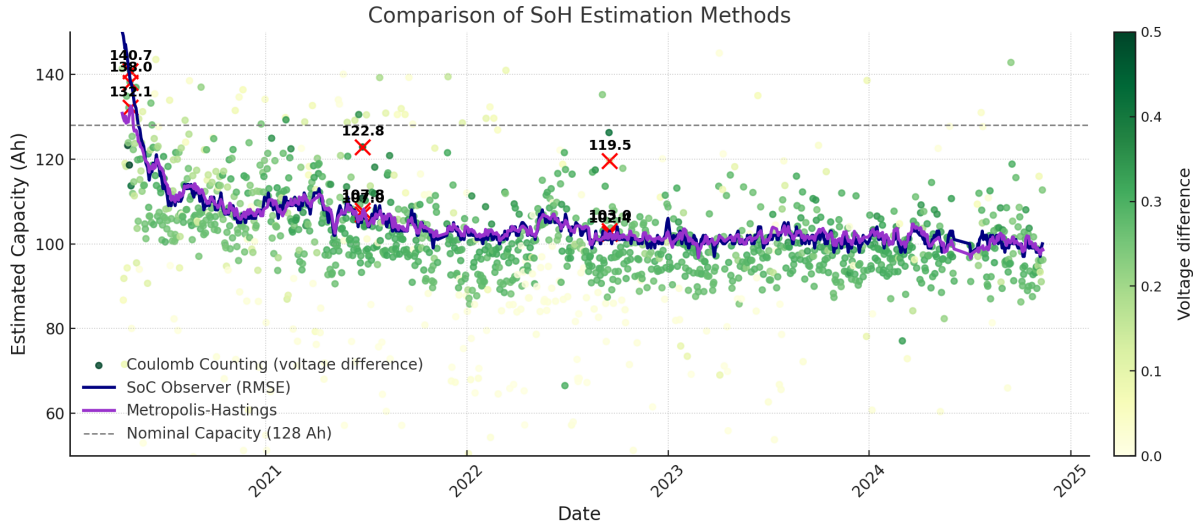

Figure 4.10: Comparison of Estimation Methods

In the first Coulomb counting method, we estimated capacity for each day independently. Due to the assumptions necessary to apply this method and different ship's operational profile each day, the capacity estimates vary significantly from one day to the other. Even when taking the voltage difference into account, there are still many different feasible capacity estimations for a single day. SoC Observer and Metropolis-Hastings methods produced better results in this regard, as the estimations from these approaches were not limited to a single day but dynamic and took the previous estimation into account for the subsequent day.

A clear battery degradation trend is noticeable in the results of all approaches and when considering the trendline from the approach one, all three approaches seem to agree about the degradation trend and the approximate capacity value. Especially RMSE and M-H methods converge to the same result.

When it comes to the capacity tests,**SoC Observer and Metropolis-Hastings approaches agree about the battery capacity value on the capacity test days**, especially for the tests in June 2021 and September 2022. The results of the first capacity test on May 2020 deviate maybe due to the beginning of the measurement period and the value not being refined/stabilized yet. The test day capacity estimations from the Coulomb Counting approach, however show significantly higher results for the capacity estimate compared to the other two approaches and compared to the overall capacity trend.

In general, the SoC and probability-based approach are very similar and tend to agree about the capacity estimate, whereas the simple Coulomb Counting approach shows more variability in the estimated capacity values.

4.6 Discussion

Each presented method has its advantages and disadvantages as well as some underlying assumptions.

The **Coulomb Counting** method is the simplest and most commonly used. It estimates the capacity by integrating current over time between rest periods. The first assumption is that

we can detect two points in a day when the battery is not in use and current is near zero. The other assumption is that between these two points, there is some voltage difference which allows us to integrate the current over time. The well-known challenge of the Coulomb Counting is the sensor drift and accumulative SoC integration.

The **SoC Observer** approach improves upon Coulomb Counting by correcting the integrated SoC based on the voltage error. The State of Health (SoH) is subsequently estimated by minimizing a systematic correction (i.e., minimizing the remaining voltage error) over a given data record, such as one day of operational data. The main advantage is that this can be done during normal operation, since we do not need the specific rest periods, like in the first method. There is a SoC correction which compares the error between the measured and predicted voltage and improves the accuracy of SoC and capacity estimates. However, the main assumption here is that the battery model, used for the voltage prediction, is accurate and reflects the dynamics of the battery system.

The **Probabilistic Metropolis-Hastings** method frames the capacity estimation as a Bayesian inference problem. This method is very similar to the second method, with the difference that it returns the estimated capacity's posterior distribution rather than a single point value. We take into account both prior knowledge and the likelihood of the data. The challenge with this method is the need for tuning parameters like standard deviation and the number of iterations.

Practical Relevance and Limitations

The proposed SoH estimation methods, offer an effective way to estimate battery health from operational data alone. These results are relevant for real-world application, as a reliable SoH estimation is crucial for the safe operation of maritime battery systems and the capacity tests are time-consuming, require downtime and contribute to battery degradation.

In this work, we have presented three methods in which just by using the voltage and current channels from the BMS, we were able to estimate the SoH during typical operation of a vessel. The SoC Observer is especially relevant as a dynamic approach that is able to continuously track SoC and estimates SoH during the real-world operation.

Since the methods only rely on the operational data, there are understandably some limitations. The methods rely on the accurate current and voltage measurements, therefore a sensor drift can affect the result of both Coulomb counting and SoC Observer. Furthermore, the accuracy of the Observer and Coulomb counting depends relies on the correctness of the OCV-SoC model and the underlying battery model used for voltage prediction.

4.7 Contributions

I have analyzed the operational battery data from a ship and implemented a capacity estimation framework that can be used as an online capacity estimation mechanism. The SoC Observer improves the standard Coulomb Counting by incorporating SoC correction and minimizes the error by estimating the correct SoH. I have implemented the Metropolis-Hastings algorithm for the capacity estimation problem and applied it both for synthetic and for the ship's operational data.

I have implemented a clean framework for capacity estimation that can be integrated into real-world applications and the battery management system.

CHAPTER 5

Conclusion

This thesis presented three approaches for estimating State of Health (SoH) of NMC batteries using only operational data. Starting with a basic Coulomb Counting method, a visible degradation trend was observed. While this method provided valuable initial insights into the SoH decline, it has inherent limitations due to assumptions regarding rest points and the need for sufficient voltage difference between the rest points.

The SoC Observer approach builds upon Coulomb Counting by dynamically correcting the integrated SoC based on voltage errors within a cascaded observer framework. Over time, systematic errors within the framework indicate the capacity decline. Although this method enhances accuracy, its effectiveness depends on the quality of the underlying battery model.

The third method, a probabilistic approach based on the Metropolis-Hastings algorithm, extends the concept by generating a posterior distribution over capacity estimates. This not only provides point estimates but also quantifies uncertainty, offering deeper insights into the confidence of the SoH assessment.

To validate these methods, capacity test days were used as reference points. The results indicate that there is SoH information in the voltage error and that all three approaches consistently indicate a degradation trend. The probabilistic method with SoC Observer, in particular, showed strong confidence in capacity estimations, as visible by sharp posterior distribution. Additionally, tests on synthetic data with known capacity confirmed the robustness of the approach, with the Markov Chain illustrating clear convergence.

Despite its strengths, challenges remain when working solely with operational data as the methods rely on the accurate sensor measurements. Nevertheless, the proposed methods offer significant practical advantages for continuous SoH monitoring without the need for manual capacity tests.

Future work could focus on enhancing the SoC Observer by integrating a more advanced battery model, incorporating RC elements and temperature effects.

Overview of Generative AI Tools Used

ChatGPT (Version GPT-4o, OpenAI, May 2025) was used occasionally as a linguistic aid: to find synonyms and improve phrasing. However, the content and conceptual design of this work predominantly reflect my own contribution.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] H. Bergveld. Battery management systems: design by modelling. *International Journal of Chemical Reactor Engineering*, 1:1–10, 2001. *INT J CHEM REACT ENG.*

[2] Corvus Energy. Corvus energy website. `https://corvusenergy.com/`. Accessed: 2025-03-25.

[3] DigiBatt. About digibatt. `https://digibattproject.eu/about`. Accessed: 2025-04-01.

[4] Andreas Dörnhöfer. *Betriebsfestigkeitsanalyse elektrifizierter Fahrzeuge: Multilevel-Ansätze zur Absicherung von HV-Batterien und elektrischen Steckkontakten.* Springer Vieweg, Wiesbaden, January 2019.

[5] Alexander Farmann, Wladislaw Waag, Andrea Marongiu, and Dirk Uwe Sauer. Critical review of on-board capacity estimation techniques for lithium-ion batteries in electric and hybrid electric vehicles. *Journal of Power Sources*, 281:114–130, 2015.

[6] Miran Gaberšček. Impedance spectroscopy of battery cells: Theory versus experiment. *Current Opinion in Electrochemistry*, 32:100917, 2022.

[7] Matteo Galeotti, Lucio Cinà, Corrado Giammanco, Stefano Cordiner, and Aldo Di Carlo. Performance analysis and soh (state of health) evaluation of lithium polymer batteries through electrochemical impedance spectroscopy. *Energy*, 89:678–686, 2015.

[8] Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes.* Oxford University Press, 2001.

[9] H. He, R. Xiong, J. Fan, and S. Li. Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *eTransportation*, 1:100003, 2019.

[10] C. Heitzinger. *Algorithms with JULIA: Optimization, Machine Learning, and Differential Equations Using the JULIA Language.* Springer International Publishing, 2022.

[11] HIOKI E.E. Corporation. Focusing on weld quality to boost the performance of xEV batteries; helping improve quality with accurate and high-speed super-low-resistance measurement. `https://www.hioki.com/global/industries-solutions/manufacturing/rm3545.html`. Accessed: 2025-04-10.

[12] Xiaosong Hu, Shengbo Li, and Huei Peng. A comparative study of equivalent circuit models for li-ion batteries. *Journal of Power Sources*, 198:359–367, 2012.

[13] Haichi Huang, Chong Bian, Mengdan Wu, Dong An, and Shunkun Yang. A novel integrated soc–soh estimation framework for whole-life-cycle lithium-ion batteries. *Energy*, 288:129801, 2024.

[14] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB)*, pages 392–403, New York City, USA, 1998. Morgan Kaufmann.

[15] Jaewook Lee, Woosuk Sung, and Joo-Ho Choi. Metamodel for efficient estimation of capacity-fade uncertainty in li-ion batteries for electric vehicles. *Energies*, 8(6):5538–5554, 2015.

[16] Cheng Lin, Aihua Tang, and Wenwei Wang. A review of soh estimation methods in lithium-ion batteries for electric vehicle applications. *Energy Procedia*, 75:1920–1925, 2015. Clean, Efficient and Affordable Energy for a Sustainable Future: The 7th International Conference on Applied Energy (ICAE2015).

[17] Hancong Liu, Sirish Shah, and Wei Jiang. On-line outlier detection and data cleaning. *Computers Chemical Engineering*, 28(9):1635–1647, 2004.

[18] Yongjie Liu, Zhiwu Huang, Liang He, Jianping Pan, Heng Li, and Jun Peng. Temperature-aware charging strategy for lithium-ion batteries with adaptive current sequences in cold environments. *Applied Energy*, 352:121945, 2023.

[19] Celina Mikolajczak, Pe Kahn, Kevin White, and Richard Long. *Lithium-Ion Batteries: Hazard and Use Assessment*. Springer, New York, NY, 2011.

[20] Solmaz Nazaralizadeh, Paramarshi Banerjee, Anurag K. Srivastava, and Parviz Famouri. Battery energy storage systems: A review of energy management systems and health metrics. *Energies*, 17(5):1250, 2024.

[21] Seongyun Park, Jeongho Ahn, Taewoo Kang, Sungbeak Park, Youngmi Kim, Inho Cho, and Jonghoon Kim. Review of state-of-the-art battery state estimation technologies for battery management systems of stationary energy storage systems. *Journal of Power Electronics*, 20(3):880–892, August 2020.

[22] Prarthana Pillai, Sneha Sundaresan, Pradeep Kumar, Krishna R. Pattipati, and Balakumar Balasingam. Open-circuit voltage models for battery management systems: A review. *Energies*, 15(18):6803, 2022.

[23] Sunil K. Pradhan and Basab Chakraborty. Battery management strategies: An essential review for battery state of health monitoring techniques. *Journal of Energy Storage*, 51:104427, 2022.

[24] Jeff Shepard. What does electrochemical impedance spectroscopy have to do with Li-ion health? *Battery Power Tips*, 2023. Accessed: 2025-06-10.

[25] Prashant Shrivastava, Tey Kok Soon, Mohd Yamani Idna Bin Idris, and Saad Mekhilef. Overview of model-based online state-of-charge estimation using kalman filter family for lithium-ion batteries. *Renewable and Sustainable Energy Reviews*, 113:109233, 2019.

[26] Suyeon Sohn, Ha-Eun Byun, and Jay H. Lee. Cnn-based online diagnosis of knee-point in Li-ion battery capacity fade curve. *IFAC-PapersOnLine*, 55(7):181–185, 2022. 13th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS).

[27] Jinpeng Tian, Rui Xiong, Weixiang Shen, and Fengchun Sun. Electrode ageing estimation and open circuit voltage reconstruction for lithium ion batteries. *Energy Storage Materials*, 37:283–295, 2021.

[28] Ufine Battery. Battery state of charge and battery state of health. `https://www.ufinebattery.com/blog/battery-state-of-charge-and-battery-state-of-health/`, 2025. Accessed: 2025-04-10.

[29] Erik Vanem, Qin Liang, Carla Ferreira, Christian Agrell, Nikita Karandikar, Shuai Wang, Maximilian Bruch, Clara Salucci, Christian Grindheim, Anna Kejvalova, Øystein Alnes, Kristian Thorbjørnsen, Azzeddine Bakdi, and Rambabu Kandepu. Data-driven approaches to diagnostics and state of health monitoring of maritime battery systems. *Annual Conference of the PHM Society*, 15(1), October 2023.

[30] Ester Vasta, T. Scimone, Giovanni Nobile, Otto Eberhardt, Daniele Dugo, Massimiliano De Benedetti, Luigi Lanuzza, G. Scarcella, Luca Patané, Paolo Arena, and Mario Cacciato. Models for battery health assessment: A comparative evaluation. *Energies*, 16(2):632, January 2023.

[31] Shunli Wang, Yongcun Fan, Daniel-Ioan Stroe, Carlos Fernandez, Chunmei Yu, Wen Cao, and Zonghai Chen. Chapter 2 - electrical equivalent circuit modeling. In Shunli Wang, Yongcun Fan, Daniel-Ioan Stroe, Carlos Fernandez, Chunmei Yu, Wen Cao, and Zonghai Chen, editors, *Battery System Modeling*, pages 47–94. Elsevier, 2021.

[32] Teng Wang, Yuhao Zhu, Wenyuan Zhao, Yichang Gong, Zhen Zhang, Wei Gao, and Yunlong Shang. Capacity degradation analysis and knee point prediction for lithium-ion batteries. *Green Energy and Intelligent Transportation*, 3(5):100171, 2024.

[33] Fangfang Yang, Yinjiao Xing, Dong Wang, and Kwok-Leung Tsui. A comparative study of three model-based algorithms for estimating state-of-charge of lithium-ion batteries under a new combined dynamic loading profile. *Applied Energy*, 164:387–399, 2016.

[34] Ruxin Yu, Gang Liu, Linbo Xu, Yanqiang Ma, Haobin Wang, and Chen Hu. Status of health estimation methods. *Encyclopedia*, 2025. Accessed: 2025-04-24.

[35] Ruifeng Zhang, Bizhong Xia, Baohua Li, Libo Cao, Yongzhi Lai, Weiwei Zheng, Huawen Wang, Wei Wang, and Mingwang Wang. A study on the open circuit voltage and state of charge characterization of high capacity lithium-ion battery under different temperature. *Energies*, 11(9):2408, September 2018.

[36] Wei Zhang, Fenfen Ma, Sibei Guo, Xin Chen, Ziqi Zeng, Qiang Wu, Shuping Li, Shijie Cheng, and Jia Xie. A model cathode for mechanistic study of organosulfide electrochemistry in li-organosulfide batteries. *Journal of Energy Chemistry*, 66:440–447, 2022.

[37] Fangdan Zheng, Yinjiao Xing, Jiuchun Jiang, Bingxiang Sun, Jonghoon Kim, and Michael Pecht. Influence of different open circuit voltage tests on state of charge online estimation for lithium-ion batteries. *Applied Energy*, 183:513–525, 2016.