McGill University

MECH501: Probability, Statistics and Machine Learning in Mechanical Engineering

State of Charge Estimation of Lithium-Ion Batteries using EKF and CNN

Louise Quincy 261032701

Contents

List of Figures iii

1 Introduction 1
- 1.1 Problem statement . . . 1
- 1.2 Dataset . . . 1
- 1.3 Methods overview . . . 2
  - 1.3.1 Extended Kalman Filter . . . 2
  - 1.3.2 Convolutional Neural Network . . . 2

2 Methodology 3
- 2.1 EKF-Based SOC Estimation . . . 3
  - 2.1.1 Battery Model . . . 4
  - 2.1.2 Parameter Identification and Lookup Table Generation . . . 4
  - 2.1.3 EKF Theory . . . 4
  - 2.1.4 EKF Matrices . . . 5
  - 2.1.5 Process Noise Matrix . . . 5
  - 2.1.6 EKF Implementation Steps . . . 7
- 2.2 CNN-Based SOC Estimation . . . 7
  - 2.2.1 Theoretical Overview . . . 7
  - 2.2.2 Input Representation and Preprocessing . . . 7
  - 2.2.3 Model Architecture . . . 7
  - 2.2.4 Training Methodology . . . 8
  - 2.2.5 Uncertainty Estimation with MC Dropout . . . 8
  - 2.2.6 Model Performance and Convergence . . . 8

3 Results Discussion 10
- 3.1 Performance Evaluation . . . 10
  - 3.1.1 CNN SOC Prediction and Error . . . 10
  - 3.1.2 EKF SOC Prediction and Error . . . 11
  - 3.1.3 Overall Comparison . . . 11
- 3.2 Real-Time Feasibility . . . 12
- 3.3 Summary . . . 12

4 Conclusion 13

References 14

Appendices 15

A Appendix 15

B Lookup Table Used for EKF Parameter Retrieval **16**

C Battery Datasheet: Melasta SLPB8995132HV **17**

List of Figures

2.1 EKF-based SoC estimation architecture: current input and model prediction are corrected using measured voltage to estimate the battery's internal state and SoC. Rzepka et al. (2021) 3
2.2 Equivalent circuit model with two RC elements and hysteresis. Plett (2015) . 4
2.3 Implementation steps for the EKF algorithm. 6
2.4 CNN training loss over 200 epochs using MSE as the loss function. 9

3.1 CNN-based SoC estimation with predictive uncertainty. 10
3.2 Zoomed-in CNN SOC prediction with uncertainty. 11
3.3 EKF-based SoC estimation with ± 2 confidence bounds. 11
3.4 Zoomed-in comparison of CNN and EKF SOC estimators. 12

Chapter 1

Introduction

1.1 Problem statement

The estimation of a battery's State of Charge (SOC) is critical in electric vehicles and energy systems, as it informs energy management decisions such as whether to recharge, engage in power-saving modes, or continue regular operation. Unlike a fuel tank, the SOC cannot be directly measured and must be estimated using measurable signals such as voltage, current, and temperature.

Formula Student is an international engineering competition where teams design, build, and race small-scale formula-style electric vehicles. McGill Formula Electric (MFE) participates in this competition and uses a 600V battery pack made from Melasta SLPB8995 Li-Polymer pouch cells. One of the most important events in the competition is the Endurance event, which stresses the vehicle over an extended period (typically 22 km or 30 minutes). Points are awarded based on completion and energy efficiency SAE International (2025). Accurate SOC estimation directly impacts the team's ability to finish the event and score maximum points.

Traditionally, classical model-based approaches like the Extended Kalman Filter (EKF) are used for SOC estimation. Recently, data-driven methods such as Convolutional Neural Networks (CNNs) have shown promise due to their ability to model complex non-linear relationships and edge cases from raw data. This project compares these two methods based on their performance in terms of root mean square error (RMSE) and real-time feasibility.

1.2 Dataset

The dataset used in this project consists of simulated data from a 2RC Equivalent Circuit Model (ECM) with hysteresis. The model parameters were derived from charge/discharge test data of Melasta cells used by MFE. The parameters are stored in lookup tables (LUTs) as functions of open-circuit voltage (OCV), SOC, and temperature.

The dataset includes time-series data of current, voltage, and true SOC, generated under dynamic conditions to emulate real-world race scenarios.

1.3 Methods overview

1.3.1 Extended Kalman Filter

EKF is a recursive filter that estimates the internal states of a nonlinear system by linearizing around the current estimate. It uses a state-space formulation:

$$z[k] = f(z[k-1], u[k]) + \delta[k] \tag{1.1}$$

$$y[k] = g(z[k], u[k]) + \epsilon[k] \tag{1.2}$$

For SOC estimation, the system is modeled using a 2RC ECM with hysteresis. The system state vector includes charge q, two RC voltages, hysteresis voltage, and instantaneous ohmic voltage drop. The model is nonlinear due to the hysteresis and the OCV-SOC relationship. This implementation follows the methodology in Rzepka et al. (2021) from Energies, which provides a structured approach for tuning the process noise matrix Q using parameter uncertainties and sensor noise. The battery model includes a variable Coulombic efficiency and is implemented with LUTs.

1.3.2 Convolutional Neural Network

CNNs are deep learning models particularly effective at capturing spatial patterns, but they can also be used for time-series prediction. In this project, the CNN takes input a window of past voltage and current measurements as an input and outputs the estimated SOC.

The architecture consists of convolutional layers followed by dense layers. The model is trained using mean squared error (MSE) loss. Training was performed over 200 epochs, and convergence was observed in the loss curve. Data augmentation and normalization were used to improve generalization

Chapter 2

Methodology

2.1 EKF-Based SOC Estimation

The EKF is a recursive estimator that uses a mathematical model of the battery to predict internal states and correct them using actual sensor measurements. Figure 2.1 illustrates the overall EKF-based SoC estimation architecture.

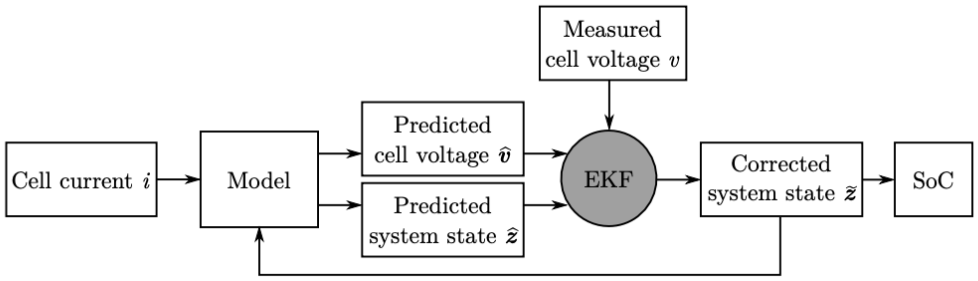

Figure 2.1: EKF-based SoC estimation architecture: current input and model prediction are corrected using measured voltage to estimate the battery's internal state and SoC. Rzepka et al. (2021)

At each time step:

1. The battery model predicts the system state $\hat{\boldsymbol{z}}$ and output voltage $\hat{v}$ based on the previous state and current input.
2. The EKF compares the predicted voltage $\hat{v}$ with the measured terminal voltage v.
3. The Kalman gain is computed and used to update the predicted state to a corrected state $\tilde{\boldsymbol{z}}$.
4. The corrected state is then fed back into the model and used to compute the SoC from the charge state $q[k]$.

This feedback-based correction allows the EKF to account for model uncertainty, noise, and real-world discrepancies while maintaining real-time feasibility. This work implements an Extended Kalman Filter (EKF) for battery State of Charge (SoC) estimation based on the structured methodology presented by Rzepka et al. (2021). The battery is modeled using the equivalent circuit model (ECM) theory introduced by Plett (2015), which captures dynamic behavior through resistive, capacitive, and hysteresis elements.

2.1.1 Battery Model

The ECM used in this project is illustrated in Figure 2.2. It consists of an open-circuit voltage source $v_{oc}(q[k])$, two resistor-capacitor (RC) pairs (R_1, C_1) and (R_2, C_2), a series resistance R_0, and a hysteresis voltage component $v_h[k]$. The terminal voltage $v[k]$ of the battery is given by:

$$v[k] = v_{oc}(q[k]) - v_{R0}[k] - v_{R1}[k] - v_{R2}[k] + v_h[k] \tag{2.1}$$

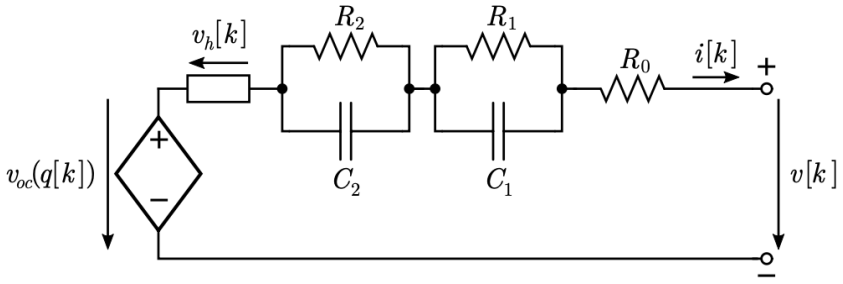

Figure 2.2: Equivalent circuit model with two RC elements and hysteresis. Plett (2015)

The states evolve as follows:

$$\eta[k-1] = \begin{cases} 1 & \text{if } i[k-1] \geq 0 \\ \tilde{\eta} & \text{if } i[k-1] < 0 \end{cases} \tag{2.2}$$

$$q[k] = q[k-1] - \eta[k-1] \cdot i[k-1] \cdot \Delta t \tag{2.3}$$

$$v_{Rj}[k] = e^{-\Delta t/\tau_j} v_{Rj}[k-1] + R_j(1 - e^{-\Delta t/\tau_j}) i[k-1], \quad j = 1, 2 \tag{2.4}$$

$$v_h[k] = e^{-|\eta[k-1]i[k-1]\gamma/Q_{cell}|\Delta t} v_h[k-1] - M(1 - e^{-|\eta[k-1]i[k-1]\gamma/Q_{cell}|\Delta t})(i[k-1]) \tag{2.5}$$

$$v_{R0}[k] = R_0 i[k] \tag{2.6}$$

2.1.2 Parameter Identification and Lookup Table Generation

To parameterize the ECM, experimental charge/discharge data was collected from Melasta Li-polymer SLPB8995 pouch cells. For each test, the model voltage $\hat{v}[k]$ was fitted to the measured voltage $v[k]$ by minimizing the residual sum of squares:

$$\min_{R_0, R_1, \tau_1, R_2, \tau_2, \gamma} \sum_k (\hat{v}[k] - v[k])^2 \tag{2.7}$$

The parameters were then tabulated as functions of OCV, SOC, and temperature using a lookup table (LUT). Interpolation between LUT values was done using linear interpolation.

2.1.3 EKF Theory

The nonlinear nature of the ECM, particularly due to hysteresis and $v_{oc}(q)$, necessitates the use of an EKF. The EKF operates in two main steps: prediction and correction.

EKF Algorithm

Given state vector:

$$z[k] = \begin{bmatrix} q[k] & v_{R1}[k] & v_{R2}[k] & v_h[k] & v_{R0}[k] \end{bmatrix}^T \tag{2.8}$$

and input:

$$u[k] = \begin{bmatrix} i[k-1] & (i[k-1]) & i[k] \end{bmatrix}^T \tag{2.9}$$

Prediction step:

$$\hat{z}[k] = f(\tilde{z}[k-1], u[k]) \tag{2.10}$$
$$\hat{P}[k] = A[k]\tilde{P}[k-1]A[k]^T + Q[k] \tag{2.11}$$
$$\hat{y}[k] = g(\hat{z}[k], u[k]) \tag{2.12}$$

Kalman gain:

$$K[k] = \hat{P}[k]C[k]^T(C[k]\hat{P}[k]C[k]^T + R[k])^{-1} \tag{2.13}$$

Correction step:

$$\tilde{z}[k] = \hat{z}[k] + K[k](y[k] - \hat{y}[k]) \tag{2.14}$$
$$\tilde{P}[k] = (I - K[k]C[k])\hat{P}[k] \tag{2.15}$$

2.1.4 EKF Matrices

The Jacobians of f and g yield the matrices:

$$A[k] = \text{diag}(1, e_1, e_2, e_h, 0) \tag{2.16}$$

$$B[k] = \left.\frac{\partial f}{\partial u}\right|_{z[k-1],u[k]} = \begin{pmatrix} -\eta[k-1]\cdot\Delta t & 0 & 0 \\ R_1(1-e_1) & 0 & 0 \\ R_2(1-e_2) & 0 & 0 \\ -\frac{\eta\gamma\Delta t}{Q}((i[k-1])\cdot v_h[k-1] + M)\cdot e_h & -M(1-e_h) & 0 \\ 0 & 0 & R_0 \end{pmatrix} \tag{2.17}$$

$$C[k] = \begin{bmatrix} \partial v_{oc}/\partial q & -1 & -1 & 1 & -1 \end{bmatrix} \tag{2.18}$$

2.1.5 Process Noise Matrix

The total process noise $Q[k]$ consists of two parts: parameter uncertainty and input current noise:

$$Q[k] = J[k]Q_p[k]J[k]^T + B[k]S[k]B[k]^T \tag{2.19}$$

Q_p is the diagonal covariance of ECM parameters:

$$Q_p = \text{diag}(\sigma^2_{R_0}, \sigma^2_{R_1}, \sigma^2_{\tau_1}, \dots) \tag{2.20}$$

$S[k]$ is the covariance matrix of current noise, derived from sensor datasheet specifications.

5. Step-by-Step Guide to Implementation

This chapter gives an overview of the necessary steps to get a working EKF implementation for SoC estimation. It summarizes the tasks that have to be performed before and during operation of the battery system. Figure 5 shows this as a step-by-step guide.

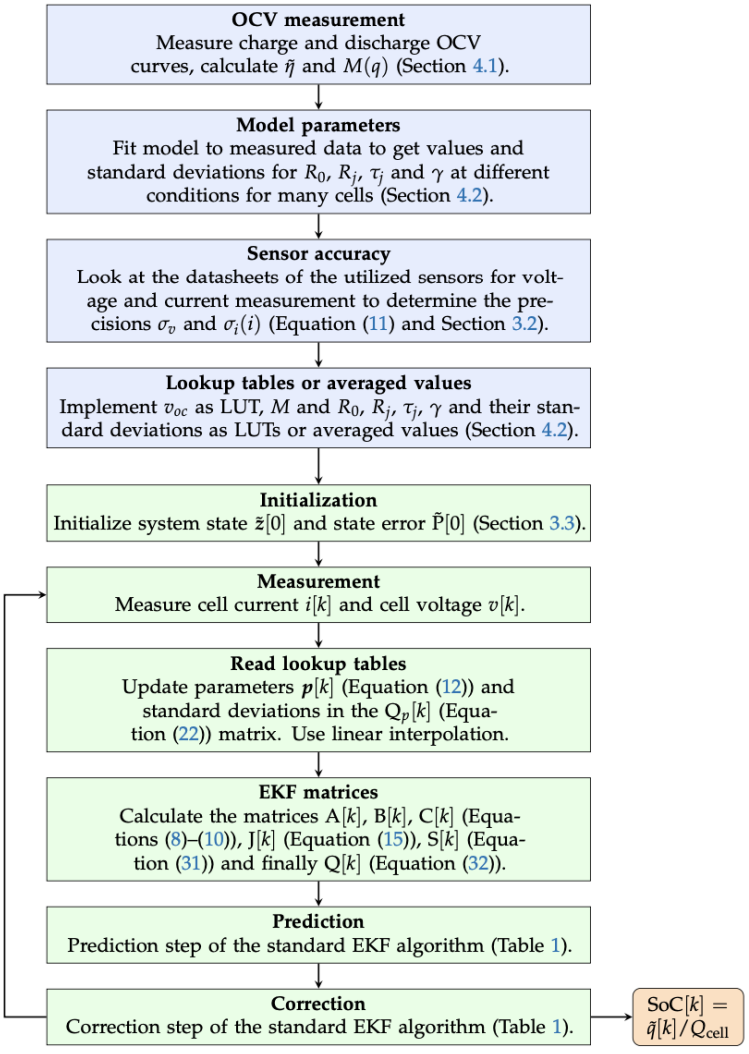

Figure 5. Steps before (blue) and during operation (green) of the proposed EKF for SoC estimation of a lithium-ion battery cell.

Figure 2.3: Implementation steps for the EKF algorithm.

2.1.6 EKF Implementation Steps

The implementation follows the flowchart in Figure 2.3 presented by Rzepka et al. Rzepka et al. (2021):

1. OCV curves and hysteresis $M(q)$ were extracted from long-rest measurements.
2. ECM parameters were fitted to charging/discharging pulse data.
3. LUTs for v_{oc}, M, R_j, τ_j, and other parameters were constructed.
4. Initial states $\tilde{z}[0]$ and $\tilde{P}[0]$ were determined using rest condition assumptions.
5. At each timestep, sensor measurements and LUTs were used to update matrices $A[k]$, $B[k]$, $C[k]$, $J[k]$, $S[k]$, and $Q[k]$.
6. Prediction and correction steps were applied sequentially.

2.2 CNN-Based SOC Estimation

2.2.1 Theoretical Overview

Convolutional Neural Networks (CNNs), traditionally used for feature extraction in images, can also be effectively adapted for time-series regression tasks. In the context of State of Charge (SOC) estimation, CNNs are leveraged to extract temporal dependencies and nonlinear patterns from voltage and current measurements, enabling end-to-end learning from raw input to SOC predictions.

The CNN model takes as input past voltage and current data, using convolutional filters to capture temporal patterns. These features are then passed through dense layers to produce a single output representing the predicted SOC at the current timestep.

2.2.2 Input Representation and Preprocessing

The input to the CNN is a time-series window of length $T = 25$ timesteps, with each timestep consisting of two features: voltage and current. Thus, each sample $x \in \mathbb{R}^{25\times 2}$. Prior to training, the inputs and outputs (true SOC) are normalized using a MinMaxScaler to scale values between 0 and 1.

2.2.3 Model Architecture

The architecture of the CNN is as follows:

- **Input Layer**: 25 timesteps of voltage and current (25×2)
- **1D Convolutional Layer**: 32 filters, kernel size = 3, activation = ReLU
- **1D Convolutional Layer**: 64 filters, kernel size = 3, activation = ReLU
- **Dropout Layer**: $p = 0.2$ to prevent overfitting
- **Flatten Layer**
- **Fully Connected (Dense) Layer**: 64 neurons, activation = ReLU
- **Output Layer**: 1 neuron, linear activation (regression output)

The model is implemented in PyTorch and trained using the Adam optimizer with a learning rate of 10^{-3}.

2.2.4 Training Methodology

The CNN was trained on simulated data generated from the same 2RC ECM with hysteresis used in the EKF implementation. Ground truth SOC was computed using the Coulomb counting method during simulation. The training procedure included:

- **Epochs**: 200
- **Loss function**: Mean Squared Error (MSE)
- **Batch size**: 32
- **Early stopping**: based on validation loss

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.21}$$

2.2.5 Uncertainty Estimation with MC Dropout

To estimate model uncertainty, Monte Carlo (MC) dropout was applied during inference. By running $T = 100$ stochastic forward passes with dropout enabled, a predictive distribution was obtained:

$$\mu = \frac{1}{T}\sum_{t=1}^{T}\hat{y}^{(t)} \tag{2.22}$$

$$\sigma^2 = \frac{1}{T}\sum_{t=1}^{T}(\hat{y}^{(t)} - \mu)^2 \tag{2.23}$$

This allows computing a confidence interval around the SOC prediction according to Gal and Ghahramani (2016)

2.2.6 Model Performance and Convergence

Figure 2.4 shows the convergence of the model during training. The loss significantly decreases within the first 50 epochs and continues to converge smoothly.

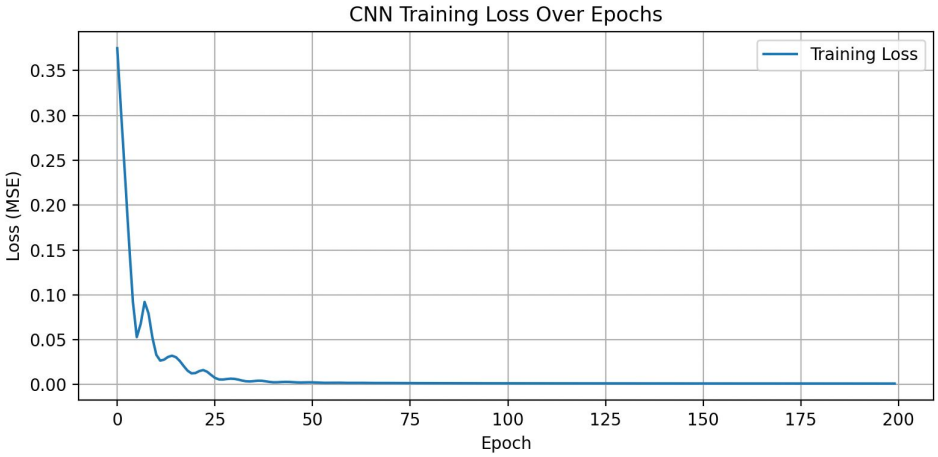

Figure 2.4: CNN training loss over 200 epochs using MSE as the loss function.

Chapter 3

Results Discussion

This chapter presents the evaluation and comparison of the Extended Kalman Filter (EKF) and Convolutional Neural Network (CNN) for State of Charge (SoC) estimation based on simulated battery cycling data. The comparison is based on Root Mean Square Error (RMSE) and real-time feasibility considerations.

3.1 Performance Evaluation

3.1.1 CNN SOC Prediction and Error

Figure 3.1 shows the CNN-predicted SoC and its associated uncertainty compared with the ground truth. The model tracks SoC with high fidelity across all operating regions, and the ± 2 confidence intervals remain well-bounded.

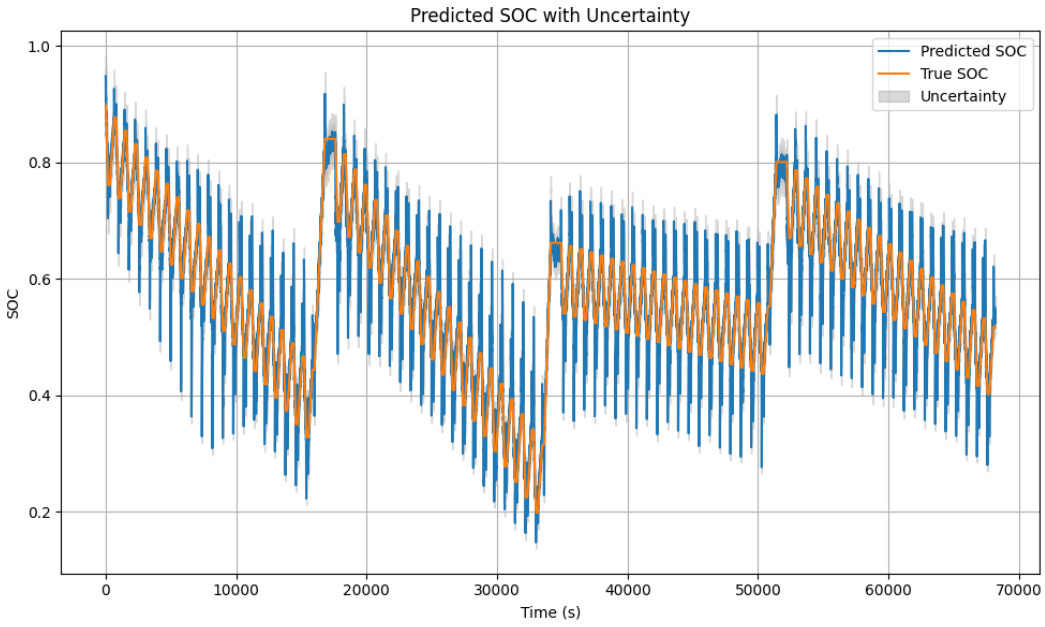

Figure 3.1: CNN-based SoC estimation with predictive uncertainty.

$$\text{RMSE}_{CNN} = 0.038 \quad \text{or } 3.8\% \tag{3.1}$$

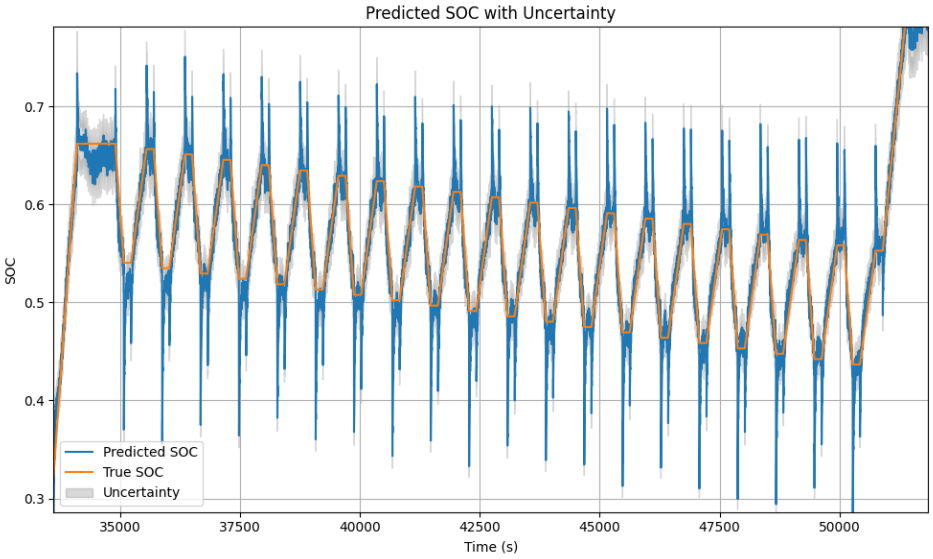

Figure 3.2: Zoomed-in CNN SOC prediction with uncertainty.

3.1.2 EKF SOC Prediction and Error

Figure 3.3 shows the EKF performance.

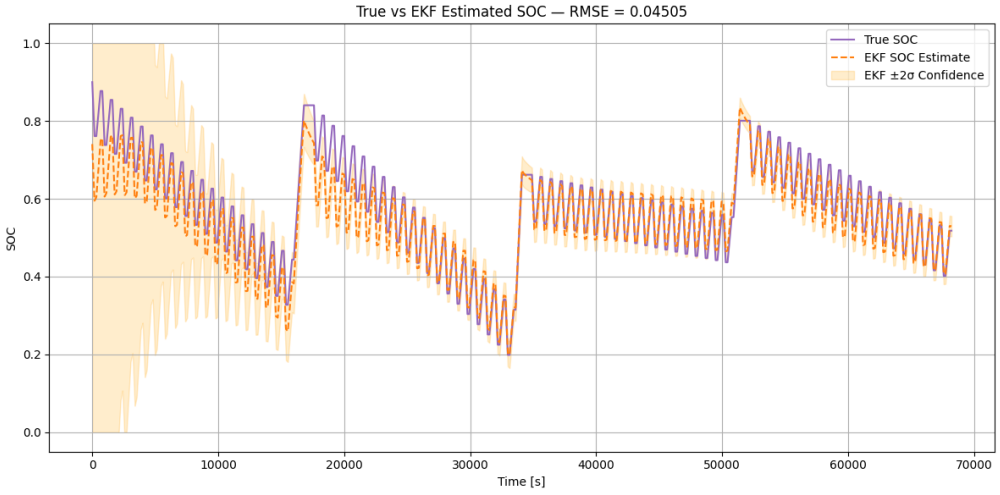

Figure 3.3: EKF-based SoC estimation with ± 2 confidence bounds.

The error margin is slightly larger than CNN's, especially at the start due to model mismatch and initial state uncertainty. The RMSE for EKF was calculated as:

$$\text{RMSE}_{EKF} = 0.04505 \quad \text{or } 4.5\% \tag{3.2}$$

3.1.3 Overall Comparison

A direct comparison of the CNN and EKF predictions versus ground truth is shown in Figure **??**. CNN achieves an RMSE of 3.8%, outperforming EKF by approximately 0.7%. However, the EKF was implemented using a manually tuned process noise covariance matrix Q_p. With better

optimization of Q_p (for example, using Expectation Maximization or Bayesian approaches), the performance of EKF could likely improve significantly.

Zoomed-in views confirm the higher variance of CNN's predictions during transients but tighter overall tracking compared to EKF in steady-state.

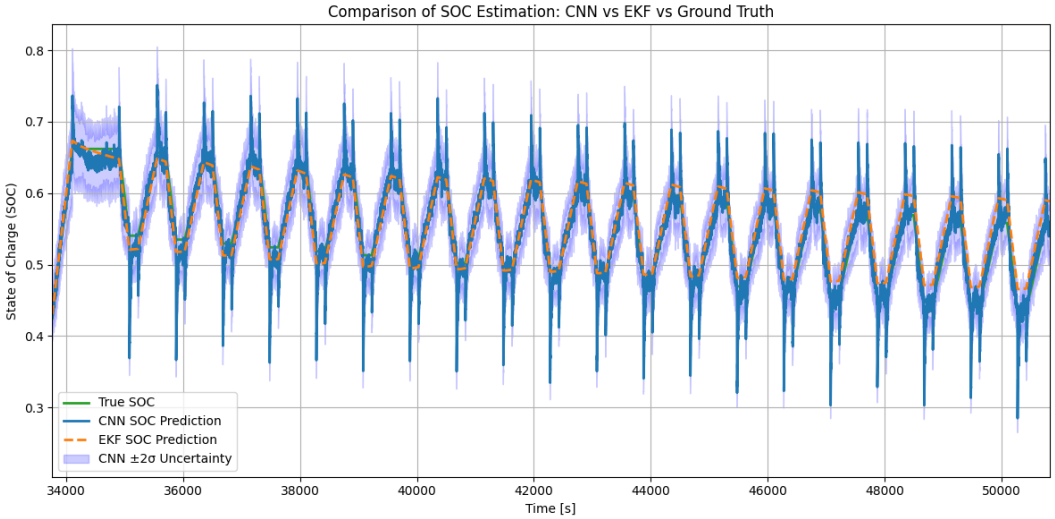

Figure 3.4: Zoomed-in comparison of CNN and EKF SOC estimators.

3.2 Real-Time Feasibility

From an implementation perspective, the EKF is far more computationally lightweight. It involves recursive matrix multiplications and simple nonlinear functions, making it ideal for embedded microcontroller deployment in real-time Battery Management Systems (BMS). The CNN model, while accurate, requires a GPU for training and is heavier in terms of real-time inference, especially when uncertainty quantification (MC dropout) is applied.

Future work may include CNN model quantization, ONNX deployment, or pruning to reduce latency for embedded targets.

3.3 Summary

The CNN outperformed the EKF in terms of pure accuracy on the simulated dataset, achieving an RMSE of 3.8% compared to 4.5% for EKF. However, EKF's performance could potentially be improved with better-tuned noise parameters. The EKF is significantly more suitable for real-time applications in constrained environments, while CNNs are promising for offline or hardware-accelerated deployments.

Chapter 4

Conclusion

This project presented and compared two approaches for battery State of Charge (SoC) estimation: a model-based Extended Kalman Filter (EKF) and a data-driven Convolutional Neural Network (CNN). Both methods were implemented using a 2RC equivalent circuit model with hysteresis, and trained or tested on high-fidelity simulated data generated from this model.

The CNN achieved higher accuracy with a lower RMSE (3.8%) compared to the EKF (4.5%). However, the EKF's performance could be improved further by optimally tuning its process noise covariance matrix Q_p. Moreover, the EKF remains more suitable for real-time, resource-constrained applications due to its low computational cost.

It is important to note that the methods were only evaluated using simulation data. The next step is to validate these approaches on real-world driving data from the Formula Electric race car using Hardware-in-the-Loop (HIL) testing. This will provide further insights into their robustness and generalizability under real battery conditions.

References

Gal, Y. and Ghahramani, Z. (2016), Dropout as a bayesian approximation: Representing model uncertainty in deep learning, *in* 'International Conference on Machine Learning', pp. 1050–1059.

Plett, G. L. (2015), *Battery Management Systems, Volume I: Battery Modeling*, Artech House, Norwood, MA, USA.

Rzepka, S., Kupper, C. and Lienkamp, M. (2021), 'Implementing an extended kalman filter for soc estimation of a li-ion battery with hysteresis: A step-by-step guide', *Energies* **14**(12), 3733.

SAE International (2025), 'Formula SAE Rules 2025'. Section EV 4.6 – Endurance Event. **URL:** *https://www.sae.org*

Appendix A

Appendix

Appendix B

Lookup Table Used for EKF Parameter Retrieval

The lookup table used in the EKF implementation provides battery model parameters as functions of SOC, temperature, and OCV. These were obtained by fitting experimental charge-discharge data from the Melasta SLPB8995132HV cell to a 2RC hysteresis model. Table B.1 shows a preview of the first 10 entries.

Table B.1: First 10 entries of the EKF lookup table used for parameter retrieval.

Temp	C-rates	SOC	OCV	R0	R1	C1	R2	C2
20	1	1.0	4.339	0.003539	0.007617	6761.29	1e-6	1260.24
20	1	0.9	4.123	0.003539	0.007687	6445.91	1e-6	1166.63
20	1	0.8	4.017	0.003539	0.007789	6054.43	1e-6	4133.27
20	1	0.7	3.914	0.003568	0.007672	5749.40	1e-6	1085.37
20	1	0.6	3.820	0.003612	0.006022	5996.43	1e-6	217.20
20	1	0.5	3.785	0.003712	0.004437	7637.85	1e-6	4232.02
20	1	0.4	3.671	0.003799	0.004142	8078.66	1e-6	242.03
20	1	0.3	3.554	0.003900	0.004126	9689.06	1e-6	2671.65
20	1	0.2	3.403	0.004008	0.004572	8411.92	1e-6	427.16
20	1	0.1	3.204	–	–	–	–	–

Appendix C

Battery Datasheet: Melasta SLPB8995132HV

The following datasheet describes the specifications of the Li-Polymer battery cell used in this project (SLPB8995132HV, 3.8V, 14Ah). Key specs include:

- Nominal Voltage: 3.8V
- Capacity: 14,000 mAh
- Max Continuous Discharge: 140A
- Peak Discharge (3s): 210A
- Charging Voltage: 4.35V
- Cut-off Voltage: 3.0V
- Operating Temp: Charge (0–45°C), Discharge (-20–60°C)

[pages=-, scale=0.92]Product Specification for Li-Polymer SLPB8995132HV 3.8V 14000mAh 10C-A0-2023.10.06 1 (1).pdf