

A neural network lpv framework for ecm battery models

Diogo Lopes Fernandes

October 23, 2025

Abstract

1 Introduction

The global transition towards electrification and sustainable energy systems has established lithium-ion batteries (LIBs) as the predominant energy storage technology for high-demand applications. Owing to their high energy density, long cycle life, and low self-discharge rates, LIBs are integral to the operation of modern electric vehicles (EVs), hybrid electric vehicles (HEVs), e-bikes, and autonomous mobile robots [1]. The viability, safety, and performance of these systems are directly dependent on the precise and reliable operation of the energy storage unit.

To ensure this reliability, a sophisticated Battery Management System (BMS) is essential for monitoring and controlling the battery pack [2]. The primary functions of a BMS include ensuring operation within the safe operating area (SOA), managing cell balancing, and—most critically—providing accurate real-time estimations of the battery’s internal states, such as the State of Charge (SoC) and the State of Health (SoH) [3]. The fidelity of these estimations is entirely reliant on the accuracy of the underlying battery model used by the BMS algorithms [4], especially under the dynamic load profiles of automotive or robotic applications.

In the literature, battery modeling strategies are broadly categorized into three main families. The first is physics-based (or electrochemical) models, which offer the highest fidelity by describing the internal electrochemical processes. The foundational model in this category is the Doyle-Fuller-Newman (DFN) model, also known as the Pseudo-two-Dimensional (P2D) model [5]. While comprehensive, the DFN model involves complex systems of partial differential equations (PDEs), making it computationally prohibitive for real-time BMS applications [6]. Consequently, significant research focuses on developing Reduced-Order Models (ROMs), such as the Single Particle Model (SPM) [7, 5], or designing complex state observers for the DFN model itself [8], all attempting to balance physical accuracy with computational feasibility.

A second major category is empirical or data-driven models. These "black-box" approaches leverage machine learning (ML) algorithms, such as Recurrent Neural Networks (RNN), NARX networks, or Support Vector Machines (SVM), to map battery inputs (e.g., current, temperature) directly to outputs (e.g., voltage, SoC) [9, 10, 11]. While these models can achieve high accuracy, they often require large and diverse training datasets and face significant challenges with extrapolation, generalization, and a lack of physical interpretability (the "black-box" problem) [12, 9].

The third approach, which strikes an optimal balance for BMS applications, is the Equivalent Circuit Model (ECM). ECMs have become the industry standard due to their intuitive structure, low computational overhead, and sufficient accuracy for state estimation [2]. These models represent the battery's electrical behavior using a combination of electrical components, including a voltage source (representing the Open-Circuit Voltage or OCV), an internal ohmic resistance (R_{int}), and one or more Resistor-Capacitor (RC) pairs to capture the dynamic polarization phenomena [13].

The primary challenge in deploying high-fidelity ECMs, however, is their parameterization. The values of the circuit components (resistances and capacitances) are not static; they exhibit strong non-linear dependencies on the battery's operating conditions, including the State of Charge, temperature, and even the battery's State of Health (SoH) or aging level [4]. Ignoring these variations leads to significant errors in state estimation. Therefore, accurate battery modeling necessitates not only a suitable ECM structure but also robust online parameter identification techniques capable of tracking these multi-timescale parameter variations in real-time [14, 15, 16].

2 Equivalent Circuit Models

2.1 First-Order Equivalent Circuit Model (1RC)

The first-order Equivalent Circuit Model (ECM), often referred to as the first-order Thevenin model or simply the 1RC model, is widely considered the industry standard for Battery Management System (BMS) applications [2]. Its predominance is due to the fact that it offers an ideal balance between model fidelity and computational complexity. Although it is significantly more accurate than the simple R_{int} (internal resistance) model by capturing the battery's transient dynamic response, it remains computationally lightweight enough to be executed in real-time on BMS microcontrollers, unlike complex electrochemical models [1].

The 1RC model structure represents the battery's electrical behavior through a circuit composed of four main components, where each models a distinct physical phenomenon:

- **Voltage Source (V_{OCV}):** The Open-Circuit Voltage (OCV) is an ideal voltage source representing the battery's thermodynamic equilibrium po-

tential. The V_{OCV} value is not constant; it has a strong non-linear relationship with the battery's State of Charge (SoC) [4].

- **Ohmic Resistor (R_0):** This resistor, also called the internal resistance, models the instantaneous Joule effect losses. Physically, it represents the combined resistance of the electrolyte, separators, and electrical contacts.
- **The RC Pair (R_1 and C_1):** This is the core of the model's dynamic part. The pair, composed of a resistor (R_1) and a capacitor (C_1) in parallel, models the battery's transient phenomena, primarily the charge-transfer polarization (R_1) and the double-layer capacitance (C_1) at the electrode/electrolyte interface.

2.2 Model Equations

The 1RC model is mathematically described by a system of state equations. The two state variables (internal states) of the model are the State of Charge ($SoC(t)$) and the voltage across the polarization capacitor ($V_1(t)$). The model output is the terminal voltage ($V_t(t)$). Adopting the convention where the current $I(t)$ is positive for discharge, the equations are as follows:

2.2.1 State Equation: SoC Estimation

The State of Charge is calculated using the Coulomb Counting method, which integrates the current over time. The differential equation for SoC is given by:

$$\frac{dSoC(t)}{dt} = -\frac{\eta I(t)}{Q} \quad (1)$$

Where η is the Coulombic efficiency (usually close to 1), $I(t)$ is the battery current, and Q is the total nominal capacity (in Ampere-seconds). This equation is the basis for most SoC estimators [?].

2.2.2 State Equation: Polarization Dynamics

The second state equation describes the variation of the voltage $V_1(t)$ across the RC pair. Applying Kirchhoff's Current Law at the RC pair's node, where the current $I(t)$ splits between R_1 and C_1 , and rearranging into state-space form, we have:

$$\frac{dV_1(t)}{dt} = -\frac{V_1(t)}{R_1 C_1} + \frac{I(t)}{C_1} \quad (2)$$

Where the term $\frac{V_1(t)}{R_1 C_1}$ represents the natural decay of the polarization voltage (the time constant is $\tau_1 = R_1 C_1$) and the term $\frac{I(t)}{C_1}$ represents the voltage buildup on the capacitor due to the current flowing into the RC pair.

2.2.3 Output Equation: Terminal Voltage

Finally, the terminal voltage $V_t(t)$ (the voltage measured by the BMS) is calculated by summing the voltage drops in the circuit, based on Kirchhoff's Voltage Law:

$$V_t(t) = V_{OCV}(SoC) - I(t)R_0 - V_1(t) \quad (3)$$

Where $V_{OCV}(SoC)$ is the equilibrium voltage (dependent on SoC), $I(t)R_0$ is the instantaneous ohmic voltage drop, and $V_1(t)$ is the transient (polarization) voltage drop calculated by Equation 2.

2.3 Model Importance and Challenges

The importance of this state-space formulation is that it provides the perfect foundation for modern estimation algorithms, such as the Extended Kalman Filter (EKF), which uses this model to predict the battery's behavior and correct its estimates of SoC and V_1 [?, 14].

However, the main challenge in implementing a high-fidelity 1RC ECM—and the central focus of this work—is that its parameters (V_{OCV} , R_0 , R_1 , and C_1) are not fixed values. They are, in fact, non-linear functions that vary drastically with the State of Charge (SoC), temperature, and the battery's State of Health (SoH) [4]. Therefore, for the model to be accurate under all operating conditions, a robust method is required to identify and update these variable parameters in real-time [15, 14].

3 Second-Order Equivalent Circuit Model (2RC)

To achieve higher fidelity in capturing the battery's dynamic behavior, the 1RC model can be extended. The Second-Order Equivalent Circuit Model (2RC), also known as the Dual Polarization (DP) model, is a common enhancement [2, 13]. This model is preferred when a single RC pair is insufficient to accurately represent the complex polarization phenomena, which often occur at different timescales.

The 2RC model retains the foundational components of the 1RC model—the OCV voltage source (V_{OCV}) and the ohmic resistor (R_0)—but adds a second RC pair (R_2 , C_2) in series with the first. This structure allows the model to distinguish between two different transient processes [4]:

- **First RC Pair (R_1, C_1):** Typically represents the "fast" dynamics, such as the charge-transfer polarization and double-layer capacitance at the primary interface.
- **Second RC Pair (R_2, C_2):** Represents a "slower" dynamic process, most commonly associated with mass transport or diffusion polarization within the electrolyte or active material.

3.1 Model Equations

By adding the second RC pair, the model's state-space representation is expanded to include three state variables: the State of Charge ($SoC(t)$), the polarization voltage of the first pair ($V_1(t)$), and the polarization voltage of the second pair ($V_2(t)$).

3.1.1 State Equation: SoC Estimation

The estimation of the State of Charge remains unchanged and is calculated using the Coulomb Counting method, as previously defined in Equation 1.

3.1.2 State Equations: Polarization Dynamics

We now have two independent state equations to describe the voltage across each RC pair. Both are analogous to the 1RC model's polarization equation:

$$\frac{dV_1(t)}{dt} = -\frac{V_1(t)}{R_1C_1} + \frac{I(t)}{C_1} \quad (4)$$

$$\frac{dV_2(t)}{dt} = -\frac{V_2(t)}{R_2C_2} + \frac{I(t)}{C_2} \quad (5)$$

Where R_1C_1 and R_2C_2 represent the distinct time constants for the fast and slow polarization phenomena, respectively.

3.1.3 Output Equation: Terminal Voltage

The terminal voltage $V_t(t)$ is now the sum of the OCV, the instantaneous ohmic drop, and the voltage drops across *both* polarization pairs:

$$V_t(t) = V_{OCV}(SoC) - I(t)R_0 - V_1(t) - V_2(t) \quad (6)$$

3.2 Model Trade-offs

The inclusion of the second RC pair allows the 2RC model to provide a more accurate fit to experimental data, especially under highly dynamic load profiles where multiple polarization effects are significant [13]. This improved accuracy, however, comes at the cost of increased computational complexity. The BMS must now track three internal states instead of two, and, more importantly, it must identify and update two additional variable parameters (R_2 and C_2), which also depend non-linearly on SoC, temperature, and SoH.

4 Partnership for a New Generation of Vehicles (PNGV) Model

Another common structure found in the literature is the model developed by the Partnership for a New Generation of Vehicles (PNGV) initiative [2]. At first

glance, the PNGV model appears structurally similar to the 1RC model, as it contains one ohmic resistor (R_0) and a single RC pair (R_1, C_1) to model the transient polarization dynamics.

The fundamental difference lies in how the State of Charge (SoC) and Open-Circuit Voltage (V_{OCV}) are represented. In the 1RC and 2RC models previously discussed, the SoC is a state calculated by Coulomb counting (Eq. 1), and the V_{OCV} is determined via a highly non-linear lookup function of that state ($V_{OCV}(SoC)$). The PNGV model linearizes this relationship by treating the V_{OCV} *itself* as a state variable. This is accomplished by modeling the battery's total charge storage as a single, large "bulk capacitor" (C_b) [2].

4.1 Model Equations

The PNGV model is also a second-order system, but its states are the polarization voltage ($V_1(t)$) and the open-circuit voltage ($V_{OCV}(t)$).

4.1.1 State Equation: Polarization Dynamics

The equation for the transient RC pair is identical to that of the 1RC model (Equation 2):

$$\frac{dV_1(t)}{dt} = -\frac{V_1(t)}{R_1 C_1} + \frac{I(t)}{C_1} \quad (7)$$

4.1.2 State Equation: Open-Circuit Voltage

The SoC (and thus the OCV) is modeled by integrating the current into the bulk capacitor C_b . This replaces the separate Coulomb counting equation. Assuming a discharge current $I(t)$ is positive, the OCV state decreases:

$$\frac{dV_{OCV}(t)}{dt} = -\frac{I(t)}{C_b} \quad (8)$$

Here, C_b is a very large capacitance representing the change in OCV with respect to the accumulated charge (measured in Ah/V or Farads).

4.1.3 Output Equation: Terminal Voltage

The output equation is identical in form to the 1RC model, but V_{OCV} is now a state variable from Equation 8 instead of a static function:

$$V_t(t) = V_{OCV}(t) - I(t)R_0 - V_1(t) \quad (9)$$

4.2 Model Trade-offs

The primary advantage of the PNGV model is that it simplifies the state estimation algorithm by avoiding the non-linear $V_{OCV}(SoC)$ lookup table. This can

make the model formulation more linear, which is advantageous for standard Kalman filters.

However, this simplification comes at a significant cost. The $V_{OCV}(SoC)$ relationship is known to be highly non-linear, especially for LFP (Lithium Iron Phosphate) chemistries. Approximating this curve with a single, constant capacitance C_b is only accurate for a very small SoC range. To maintain accuracy, the C_b parameter itself must be varied as a non-linear function of the voltage V_{OCV} (or SoC) [2]. Consequently, the non-linearity is not eliminated but merely shifted from a state function ($V_{OCV}(SoC)$) to a variable parameter ($C_b(SoC)$). This model, therefore, still faces the same central challenge as all other ECMs: all its parameters (R_0 , R_1 , C_1 , and C_b) are variable and require robust identification.

5 System Identification Methods

5.1 Transfer function model

An alternative approach to parameter identification involves moving from the time-domain state-space representation (Equations 1-6) to the Laplace s-domain. This method treats the battery as a linear system and approximates its behavior using a transfer function (TF). The ECM's impedance $Z(s)$ can be derived by applying the Laplace transform to the output equation, assuming the input is the current $I(s)$ and the output is the resulting voltage drop, or overpotential, $V_{drop}(s) = V_t(s) - V_{OCV}(s)$.

For the 1RC model, the impedance $Z(s)$ is:

$$Z_{1RC}(s) = \frac{V_{drop}(s)}{I(s)} = R_0 + \frac{R_1}{sR_1C_1 + 1} \quad (10)$$

Similarly, for the 2RC model, the impedance is the sum of the components:

$$Z_{2RC}(s) = \frac{V_{drop}(s)}{I(s)} = R_0 + \frac{R_1}{sR_1C_1 + 1} + \frac{R_2}{sR_2C_2 + 1} \quad (11)$$

5.2 Hypotheses and Crucial Simplifications

While powerful, this approach is only valid if several critical simplifying assumptions are made. The very nature of a transfer function requires the system to be **Linear Time-Invariant (LTI)**.

1. **LTI Assumption (Time-Invariance):** The primary simplification is that all model parameters— R_0 , R_1 , C_1 , R_2 , C_2 —are assumed to be **constant** during the identification experiment. This hypothesis is a significant simplification, as it is the central premise of this work that these parameters are, in fact, highly variable.

2. **LTI Assumption (Linearity):** The system's response must be linear. The main non-linearity in an ECM is the $V_{OCV}(SoC)$ relationship. To linearize the system, this component must be removed **before** identification. This is

achieved by subtracting the pre-measured OCV from the terminal voltage. The input data for identification is the current $I(t)$, and the output data is the overpotential $V_{drop}(t) = V_t(t) - V_{OCV}(SoC(t))$.

3. Local Validity: Because of these simplifications, any transfer function identified is only valid for a very small operating window (e.g., at a constant temperature and within a narrow SoC range, such as 5–10%). To model the battery’s full behavior, one would need to estimate a large set of LTI models, one for each operating point (SoC, Temp), effectively creating a lookup table of models.

5.3 Identification using System Identification Tools

With these simplifications, the parameter identification task becomes a ”black-box” or ”grey-box” system identification problem. This process is well-supported by software packages like the MATLAB System Identification Toolbox.

Using dynamic input-output data (e.g., from an HPPC test or a drive cycle), the function ‘tfest’ can be employed. This function estimates the coefficients of a transfer function $G(s)$ of a specified order (number of poles and zeros) that best fits the measured input-output data.

For example, to identify a 2RC model (Equation 11), which has two poles, one would instruct ‘tfest’ to find a second-order transfer function. The function would process the input data $I(t)$ and the output data $V_{drop}(t)$ to find the optimal coefficients. This ”black-box” time-domain modeling approach is a fast and effective method for capturing the system’s dynamics without iterative optimization [17, 14].

Once the transfer function is estimated, the physical ECM parameters (R_0 , R_1 , C_1 , etc.) can be extracted from the resulting coefficients through algebraic manipulation (e.g., partial fraction decomposition or by matching the coefficients of the polynomial).

6 Grey-Box Identification with Differentiable Solvers

The transfer function method, while fast, forces a critical simplification: the system must be treated as Linear Time-Invariant (LTI). This fundamentally conflicts with the known physics of a battery, where parameters are non-linear functions of state [4]. A more robust approach is ”grey-box” identification. In this paradigm, we retain the known non-linear Ordinary Differential Equation (ODE) structure of the ECM (e.g., the 2RC state-space model) and perform an optimization to find the parameters $\theta = \{R_0, R_1, C_1, R_2, C_2, \dots\}$ that minimize the error between the simulated voltage V_{model} and the experimental data V_{exp} .

This creates a complex, non-linear optimization problem. Historically, solving such problems was difficult, often requiring gradient-free methods (which are slow) or manual derivation of gradients (which is error-prone).

6.1 Differentiable Programming with JAX and Diffrax

A modern solution is found in the Python ecosystem through "differentiable programming". This approach leverages the **JAX** library for automatic differentiation (`jax.grad`) and just-in-time (`jit`) compilation [18]. The ECM, which is a system of ODEs, can be implemented using a differentiable ODE solver, such as **Diffrax**.

The entire simulation process—from integrating the ODEs (Eq. 4, 5) with **Diffrax** to calculating the final voltage error (Eq. 6)—becomes a single, end-to-end differentiable function. We can then use `jax.grad` to automatically and efficiently compute the exact gradient of the loss function (e.g., RMSE) with respect to every parameter θ we wish to identify. These gradients are then fed into a high-performance optimizer (like ADAM or L-BFGS, available in libraries like **Optax**) to solve the minimization problem.

6.2 Robustness via Multiple Shooting

A standard simulation ("single shooting") integrates the ODE from t_0 to t_{final} . If the time-series data is long (as in a drive cycle), this approach can be numerically unstable, suffering from vanishing or exploding gradients. A more robust strategy is **multiple shooting**.

In this method, the full time-series is partitioned into M smaller, independent intervals. The optimization problem is then reformulated as:

1. Find the global parameters θ (the ECM parameters R_0, R_1, \dots).
2. Find the local initial states x_i for *each* interval $i \in [1, M]$.
3. Minimize the sum of simulation errors within all intervals.
4. Add a "continuity constraint" to the loss function, penalizing any mismatch between the simulated end-state of interval i and the (optimized) initial state of interval $i + 1$.

This strategy stabilizes the optimization, as gradients only need to propagate through short time windows. Furthermore, the simulation of all M intervals is perfectly parallelizable, which can be efficiently implemented using `jax.vmap`.

6.3 Advantages of the Grey-Box Approach

This JAX/Diffrax implementation provides significant advantages. It allows for the handling of fully non-linear models, which is impossible with transfer functions. Instead of identifying a table of static parameters for discrete SoC/Temp points, we can identify a single, **phenomenological model** where the parameters themselves are non-linear functions (e.g., $R_1 = f(\text{SoC}, T, \theta_{R1})$), and the optimizer finds the underlying coefficients θ_{R1} . This results in a continuous model that generalizes across the entire operating range, bridges the gap between simple ECMs and complex physical models, and is a key concept in Scientific Machine Learning [19].

7 Proposed Hybrid LPV-NN Model Architecture

The central challenge in ECM accuracy, as established in the literature, is that the model parameters are not constant. They are highly non-linear functions of the battery’s internal states, primarily the State of Charge (SoC) and temperature, as well as its State of Health (SoH) [4, 15]. The ”grey-box” identification method described previously is effective at finding a single, static set of parameters, $\theta_{p,nominal}$, valid at a specific operating point.

To create a model with high fidelity across the entire operating range, we propose a hybrid, phenomenological model structured as a Linear Parameter-Varying (LPV) system. In this architecture, the underlying ECM structure (e.g., 2RC) and its state-space equations are preserved. However, the parameter vector $\theta_p = [R_0, R_1, C_1, R_2, C_2]^T$ is no longer static. Instead, it is defined as a dynamic function of a ”scheduling vector,” \mathbf{x} . As a first step, we define this vector by the most dominant non-linear dependency:

$$\mathbf{x} = [SoC(t)] \quad (12)$$

(Note: This vector can be extended to $\mathbf{x} = [SoC(t), T(t), SoH(t)]$ in a more comprehensive model).

7.1 Parameter Variation Modeling

We propose modeling the instantaneous parameter value $\theta_p(\mathbf{x})$ as a non-linear deviation from the nominal parameters $\theta_{p,nominal}$ obtained via the grey-box identification. This relationship is defined as:

$$\theta_p(\mathbf{x}) = \theta_{p,nominal} \odot (1 + \delta_{NN}(\mathbf{x})) \quad (13)$$

Where \odot denotes element-wise multiplication, and $\delta_{NN}(\mathbf{x})$ is a vector of scaling factors output by a shallow Neural Network (NN). This NN is specifically tasked with learning the non-linear dependency $\delta = f(SoC)$. This formulation is advantageous as it anchors the model to a physically-identified baseline ($\theta_{p,nominal}$) and uses the NN only to learn the *variation*.

7.2 Proposed Neural Network Architectures

We propose exploring two primary architectures for the mapping function $\delta_{NN}(\mathbf{x})$:

7.2.1 Multi-Layer Perceptron (MLP)

A standard feed-forward neural network with one or two hidden layers. This network would take SoC as its input and output the vector $\delta = [\delta_{R0}, \delta_{R1}, \dots]^T$.

- **ReLU Activation:** Using the Rectified Linear Unit (ReLU) function in the hidden layers provides computational efficiency and helps mitigate vanishing gradient issues during training.

- **Tanh Activation:** Using the hyperbolic tangent (\tanh) function for the *output* layer is highly recommended. Since \tanh bounds the output between $[-1, 1]$, it naturally constrains the deviation δ , preventing the optimizer from finding physically unrealistic parameters (e.g., it prevents $1 + \delta$ from becoming negative and resulting in a negative resistance).

7.2.2 Radial Basis Function Network (RBFN)

An RBFN is a powerful universal approximator, particularly well-suited for low-dimensional interpolation problems like this one [20]. The network’s output is a weighted sum of radial basis functions. The key advantage is that the basis functions (e.g., ‘Gaussian’ or ‘Inverse Quadratic’) create “local” responses, allowing the model to fit the parameter variations in specific SoC regions without adversely affecting others. This architecture is often more data-efficient than MLPs when the input dimension is small (i.e., only SoC).

7.3 Advantages of the Hybrid Approach

This hybrid “ECM-NN” approach merges the strengths of physics-informed and data-driven models.

1. **Phenomenological Model:** It is not a “black-box” (like a pure NARX model [21, 11]) because it retains the physically interpretable ECM state-space structure. The NN is only used to **schedule** the parameters for this physical structure.
2. **Non-linear Capability:** It explicitly captures the strong non-linear dependencies [4] that simpler online methods (like standard RLS [14]) or LTI models (‘tfest’) ignore.
3. **Training Integration:** This entire hybrid model—the ECM’s ODEs (solved by `Difffrax`) and the NN parameter functions—can be implemented as a single, end-to-end differentiable program in `JAX`. The NN weights simply become part of the overall parameter vector θ to be optimized by the grey-box solver, allowing the model to learn the parameter variations across the entire operating range from a single, dynamic dataset.

7.4 Advantages and Limitations of the Hybrid Approach

This hybrid “ECM-NN” approach merges the strengths of physics-informed and data-driven models.

1. **Phenomenological Model:** It is not a “black-box” (like a pure NARX model [21, 11]) because it retains the physically interpretable ECM state-space structure. The NN is only used to **schedule** the parameters for this physical structure.

2. **Non-linear Capability:** It explicitly captures the strong non-linear dependencies [4] that simpler online methods (like standard RLS [14]) or LTI models (‘tfest’) ignore.
3. **Training Integration:** This entire hybrid model—the ECM’s ODEs (solved by `DiffRax`) and the NN parameter functions—can be implemented as a single, end-to-end differentiable program in `JAX`. The NN weights simply become part of the overall parameter vector θ to be optimized by the grey-box solver.
4. **Interpolation vs. Extrapolation Trade-off:** The use of NNs (MLPs or RBFNs) as function approximators allows the model to capture the complex non-linear parameter variations with high fidelity *within the operating range of the training data*. This strong interpolation capability, however, comes with a well-known limitation of data-driven models: the risk of poor generalization and extrapolation. The model’s accuracy is not guaranteed for operating regions (e.g., SoC or temperature ranges) that were not sufficiently represented in the training dataset [12, 9].

8 Results

9 Conclusion

References

- [1] N. N. Mawuntu, B.-Q. Mu, O. Doukhi, and D.-J. Lee, “Modeling of the battery pack and battery management system towards an integrated electric vehicle application,” *Energies*, vol. 16, no. 20, p. 7165, 2023.
- [2] M. Tekin and M. Karamangil, “Comparative analysis of equivalent circuit battery models for electric vehicle battery management systems,” *Journal of Energy Storage*, vol. 86, p. 111327, 2024.
- [3] V. Damodaran, T. Paramadayan, D. Natarajan, R. Kumar C, P. R. Kanna, D. Taler, T. Sobota, J. Taler, M. Szymkiewicz, and M. J. Ahamed, “Development of a fast running equivalent circuit model with thermal predictions for battery management applications,” *Batteries*, vol. 10, no. 6, p. 215, 2024.
- [4] M.-K. Tran, M. Mathew, S. Janhunen, S. Panchal, K. Raahemifar, R. Fraser, and M. Fowler, “A comprehensive equivalent circuit model for lithium-ion batteries, incorporating the effects of state of health, state of charge, and temperature on model parameters,” *Journal of Energy Storage*, vol. 43, p. 103252, 2021.
- [5] J. Piruzjam, G. Liu, L. Rubacek, M. Frey, and T. Carraro, “On the analytical solution of single particle models and semi-analytical solution of p2d

- model for lithium-ion batteries,” *Electrochimica Acta*, vol. 492, p. 144259, 2024.
- [6] Z. Khalik, M. C. F. Donkers, and H. J. Bergveld, “Model simplifications and their impact on computational complexity for an electrochemistry-based battery modeling toolbox,” *Journal of Power Sources*, vol. 488, p. 229427, 2021.
 - [7] J. Li, K. Adewuyi, N. Lotfi, R. G. Landers, and J. Park, “A single particle model with chemical/mechanical degradation physics for lithium ion battery state of health (soh) estimation,” *Applied Energy*, vol. 212, pp. 1178–1190, 2018.
 - [8] R. Drummond and S. R. Duncan, “Observer design for the doyle-fuller-newman li-ion battery model without electrolyte dynamics,” *Journal of Energy Storage*, vol. 23, pp. 250–257, 2019.
 - [9] T. Kawahara, K. Sato, and Y. Sato, “Battery voltage prediction technology using machine learning model with high extrapolation accuracy,” *International Journal of Energy Research*, vol. 2023, p. 5513446, 2023.
 - [10] Q. Wang, M. Ye, M. Wei, G. Lian, and C. Wu, “Co-estimation of state of charge and capacity for lithium-ion battery based on recurrent neural network and support vector machine,” *Energy Reports*, vol. 7, pp. 7323–7332, 2021.
 - [11] Z. Xia and Y. Wu, “A hybrid network of narx and ds-attention applied for the state estimation of lithium-ion batteries,” *International Journal of Electrochemical Science*, vol. 19, p. 100632, 2024.
 - [12] A. Valizadeh and M. H. Amirhosseini, “Machine learning in lithium-ion battery: Applications, challenges, and future trends,” *SN Computer Science*, vol. 5, no. 717, 2024.
 - [13] J. Khalfi, N. Boumaaz, A. Soulmani, and E. M. Laadissi, “An electric circuit model for a lithium-ion battery cell based on automotive drive cycles measurements,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 4, pp. 2798–2810, 2021.
 - [14] H. Y. Pai, Y. H. Liu, and S. P. Ye, “Online estimation of lithium-ion battery equivalent circuit model parameters and state of charge using time-domain assisted decoupled recursive least squares technique,” *Journal of Energy Storage*, vol. 62, p. 106901, 2023.
 - [15] Z. Yang and X. Wang, “An improved parameter identification method considering multi-timescale characteristics of lithium-ion batteries,” *Journal of Energy Storage*, vol. 59, p. 106462, 2023.

- [16] H. P. G. J. Beelen, H. J. Bergveld, and M. C. F. Donkers, “On experiment design for parameter estimation of equivalent-circuit battery models,” in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 356–361, 2018.
- [17] J. Khalfi, N. Boumaaz, A. Soulmani, and E. M. Laadissi, “Box–jenkins black-box modeling of a lithium-ion battery cell based on automotive drive cycle data,” *World Electric Vehicle Journal*, vol. 12, no. 3, p. 102, 2021.
- [18] P. Kidger and T. Lyons, “Diffrax: a JAX library for numerical differential equation solving,” *arXiv preprint arXiv:2106.13642*, 2021.
- [19] C. Rackauckas, Y. Ma, J. Martensen, C. O’Connor, J. Götting, C. Schaub, J. D’Ambrosio, S. Agrawal, U. An, A. Al-Hajri, *et al.*, “Universal differential equations for scientific machine learning,” *arXiv preprint arXiv:2001.04385*, 2020.
- [20] D. S. Broomhead and D. Lowe, “Radial basis functions, multi-variable functional interpolation and adaptive networks,” *Royal Signals and Radar Establishment Memorandum*, vol. 4148, 1988.
- [21] P. Takyi-Aninakwa, S. Wang, H. Zhang, Y. Xiao, and C. Fernandez, “A narx network optimized with an adaptive weighted square-root cubature kalman filter for the dynamic state of charge estimation of lithium-ion batteries,” *Journal of Energy Storage*, vol. 68, p. 107728, 2023.