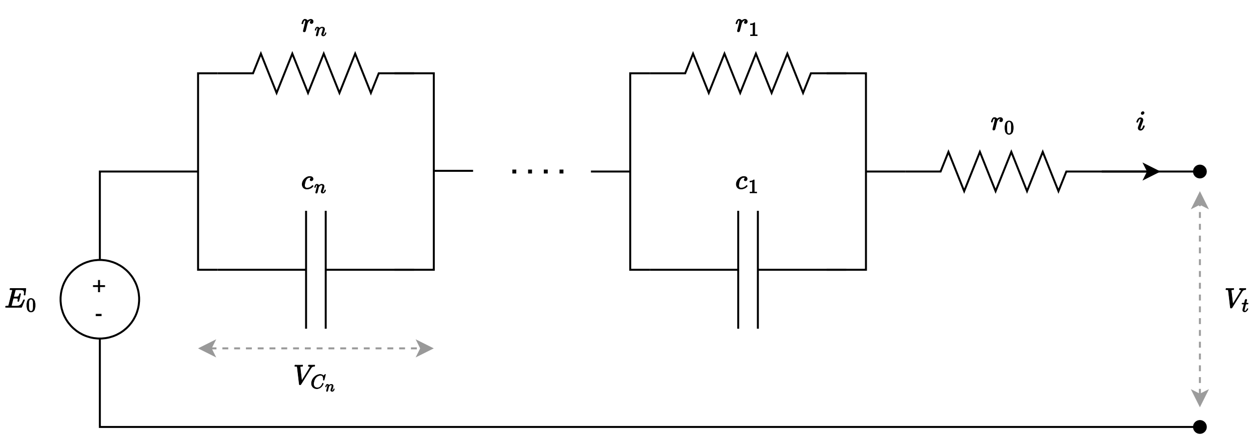
batteryModelingNotes

Physics (modeling)

ECM - equivalent circuit models

Schematic



$E_0 \rightarrow open\ circuit\ voltage\ (V)$
$V_t \rightarrow terminal\ voltage\ (V) \leftarrow output\ of\ the\ model$
$i \rightarrow current\ draw\ (A) \leftarrow input\ to\ the\ model$
$r_n, c_n \rightarrow resistance\ and\ capacitances\ (\Omega, F)$

Model equations

Electrical model

Positive i is discharge.

$$V_t = E_0 - \sum_1^n V_{c_n} - ir_0$$

$$\frac{dV_{c_n}}{dt} = \frac{i}{c_n} - \frac{V_{c_n}}{r_n c_n}$$

$$\frac{d(soc)}{dt} = \frac{-i}{Q_{Ah} \times 3600}$$

Thermal model

$$P_{heatGen} = i^2 r_0 + \sum_1^n \frac{V_{c_n}^2}{r_n}$$

$$P_{heatOut} =$$

ECM vs. electrochemical models (EChM)

1. EChM are usually coupled nonlinear partial differential equations, which take significantly higher time to run, but they characterize battery impedance more accurately
2. They typically have higher number of parameters to be identified and hence are prone to overfitting
3. Fractional order models (FOM) seem to be a middle ground between ECM and EChM

Parameter estimation

The following parameters need to be estimated

1. SOC - OCV curve
    1. charge curve
    2. discharge curve
    3. average curve?
2. Coulombic charge efficiency η_c
3. Capacity
    1. total (C/30)
    2. discharge (rated - usually 1C)
    3. nominal (tbd)
4. r_0, r_n, c_n tables wrt SOC, temperature, charge/discharge (and maybe C-rate)

r_0, r_n, c_n estimation

A comprehensive method is not documented. Only a discharge part without temperature dependence is exemplified in the code `xyz.m`

Scaling the ECM model

Overview

Generally, when the objective is to model a battery pack, the cell params are scaled as per cell-architecture of the pack such that the pack is represented as a giant cell. That approach works well on the system level when the intent of the pack model is to provide *voltage* to other powertrain models and consume a *current*. The overall objective of the vehicle model in that case is power balance and overall energy consumption prediction.

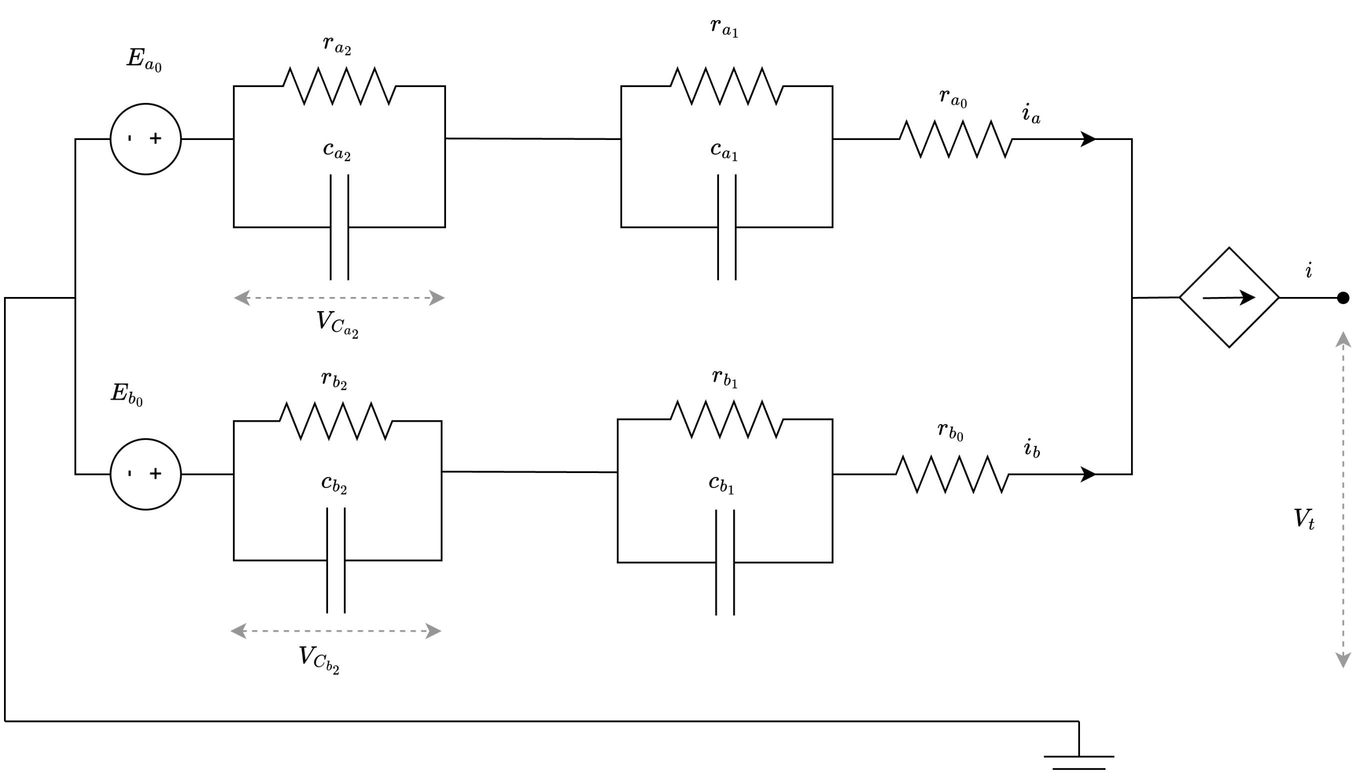
This approach does not work well if the battery model (plant) is used for BMS model development or simulation. The BMS models at a *minimum* needs the max and min cell states (voltage, SOC, temperature), and preferably needs a spread of these states depending on the computational capacity.

The modeling approach in that case is to put together a bunch of *unit cell ECM models*. A unit cell model takes current as input and produces voltage as output. If we arrange a bunch of cells ECM models in series they all consume the same current (total current drawn from the battery) - so no problems there. But as soon as we put ECM models in parallel, we need to find the current splits for the ECM model branches.

We need to solve a bunch of simultaneous equations which gives both the voltages and split-currents. It has been indicated to me that a parallel set of ECM model's current split can be calculated by a simple parallel resistance (r_0) divider circuit. This approach does not work. The initial drop of a ECM model has to do with r_0 and the RC pairs do not play any role in deciding the current, but beyond that the RC pairs come into picture and the current split evolves over time (even when the total current drawn i is constant).

Comparing two models solving approaches

Model architecture



Equations

As stated above, there are two parts to solving a network ECM

1. finding branch currents
2. finding capacitor voltages (derivatives)

Branch currents

Branch currents depend of parameters, state variables and inputs (i_{br}) from **all** branches.

Simultaneous solution

$$i_a = (E_{a_0} - E_{b_0} + ir_{b_0} - V_{C_{a_1}} - V_{C_{a_2}} + V_{C_{b_1}} + V_{C_{b_2}})/(r_{a_0} + r_{b_0})$$
$$i_b = (E_{b_0} - E_{a_0} + ir_{a_0} - V_{C_{b_1}} - V_{C_{b_2}} + V_{C_{a_1}} + V_{C_{a_2}})/(r_{a_0} + r_{b_0})$$

Resistor divider solution (NOT correct)

$$i_a = i \times \frac{r_{b_0}}{r_{a_0} + r_{b_0}}$$
$$i_b = i \times \frac{r_{a_0}}{r_{a_0} + r_{b_0}}$$

It can be seen that the resistor divider solution is part of the simultaneous equation solution, but doesn't capture the time dynamics.

Capacitor voltage

Capacitor voltages depend on parameters, state variables and input (i_{br}) from the **respective** branch. We'll have 4 equations similar to this for the 4 RC pairs.

$$\dot{V}_{C_{a_1}} = \frac{i_a}{C_{a_1}} - \frac{V_{C_{a_1}}}{r_{a_1} C_{a_1}}$$

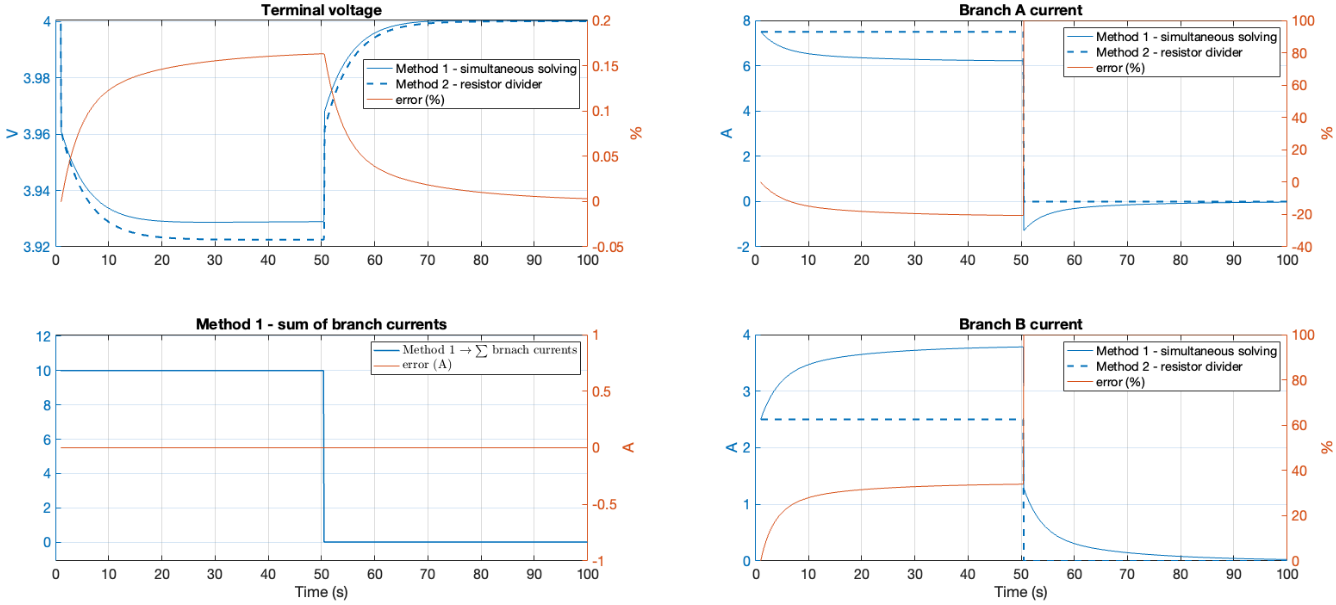
Terminal voltage

$$V_t = E_{a_0} - i_a r_{a_0} - V_{C_{a_1}} - V_{C_{a_2}}$$

The same value will be obtained for KVL on the other branch (*b*)

Result comparison

ecmScalingMethods.m
The two parallel branches for the below plot has r_{b_0} different to r_{a_0}. Everything else is same.
Blue lines are values (voltage, current), orange lines are errors.
From the 1st subplot it can be seen that the initial drop is same, but the RC pairs play in deciding how the branch current split evolves over time. This behavior is present if any parameter varies (r_0, r_x, c_x, E_0).



Quick doubts

1. Finding out OCV-SOC curve $\rightarrow$ the C/30 charge/discharge cycles are wrt time. It establishes 0% and 100% SOC in terms of pre-defined v_{min} and v_{max}. How do you define SOC-OCV in the intermediate points?
   - We find capacity Q and convert the time x-axis to SOC (%) as $SOC = (1 - \sum_1^k \frac{i}{Q}) \times 100\ \%$
2.

Asides

1. Fractional order models

BMS

SOC estimation (real time)

1. Direct methods
    1. CC (Coulomb couting)
    2. ECC (Enhanced CC)
2. Model-based methods
    1. Kalman filtering
3. Data-driven methods

Define SOC

SOC is the ratio of the present charge content of the cell to the maximum possible charge content at a pre-defined temperature and C-rate

$$SOC = \frac{Q_{remaining}}{Q_{max}} \times 100\ \%$$

Direct methods

Coulomb counting (CC)

In this method, the SOC of a cell (battery) is estimated by counting the amount of charge (coulombs) entering or leaving the battery

$$SOC(t) = SOC(t_0) + \frac{1}{Q_{max_{As}}} \int_0^T idt \times 100\ \%$$

Problems

Cumulative error becomes larger and larger

- strong dependence of initial value of SOC
- requires high precision current measurement which is subject to temperature drift usually
- does not consider health of the battery (unless $Q_{max_{As}}$ is calibrated)

Enhanced coulomb counting (ECC)

Problems to fix from CC

1. initial soc problem
2. integration error problem

ECC follows the crux of counting charge, but adds corrections in terms of:

- reseting battery SOC from SOC-OCV table as it rests beyond its largest time constant (this solves the initial SOC value problem with CC)
    - it may be a table or a analytic curve (usually a logarithm fit is used to avoid the need of higher order polynomial fit)
- adding correction terms like discharge efficiency and Peukert equation coefficient and generic current-based polynomial to the term that is integrated (this solves the lack of precision with current integration)

- the additional coefficients ($\eta_{coulumb}$, k, n, a_n) are tuned from repeated power draw $\leftrightarrow$ rest cycles with SOC-OCV resets baked in between the cycles

$$SOC(t) = SOC(t_0) + \frac{\eta_{coulomb}}{Q_{max_{As}}} \int_0^T [i^k + \sum_0^n (a_n i^n)] \, dt \times 100 \ \%$$

- this method works for a particular power draw cycle

Note: Direct measurement methods encompass all types of estimations, like internal resistance, capacity, etc.. SOC specifically can also be estimated from OCV and EIS non-real-time applications

Model-based methods

I read somewhere $\rightarrow$ The main idea is to use a model to predict OCV from measurements like current and voltage, and then use the OCV to lookup SOC from the OCV-SOC relationship.

Non-EKF

- A state-space battery model is created and SOC is used as one of the *unobservable* states
- Measurements of current, temperature and terminal voltage error (predicted vs. measured) are used to observe the internal state of SOC
- The filter (observer) gain is tuned to balance how much we trust the predicted voltage vs. the measure voltage (basically how much weightage do we give to the voltage error)
- This method relies heavily of the accuracy of the battery model, specifically in corner cases like high/low SOCs, high C-rates, high/low temperatures
- Hence usually both model-based (HiFi) and ECC-based (LoFi) algorithms are run simultaneously and an algorithm decides when to use what, or if we need to fuse the SOCs predicted by both
    - typically each estimation algorithm relies on sensors and these sensors have a validity bit (separate validity algorithm)
    - depending on validity of critical signals to an algorithm, BMS can decide to switch from HiFi to LoFi estimation

EKF

- Most widely used SOC estimator
- In EKF we linearize the non-linear system and use KF on the linear system
- It overcomes both the issue of lack of sensor accuracy and the initial value problem
- An extension to EKF is adaptive-EKF
    - in the real world the measurement and observation noise change in real time
    - AEKF estimates and iteratively updates the noise covariance
- Some use Kalman filtering without linearization like cubature KF (CKF) and unscented KF (UKF)

References

1. Fractional-order modeling and parameter identification for lithium-ion batteries
2.