Battery Modelling

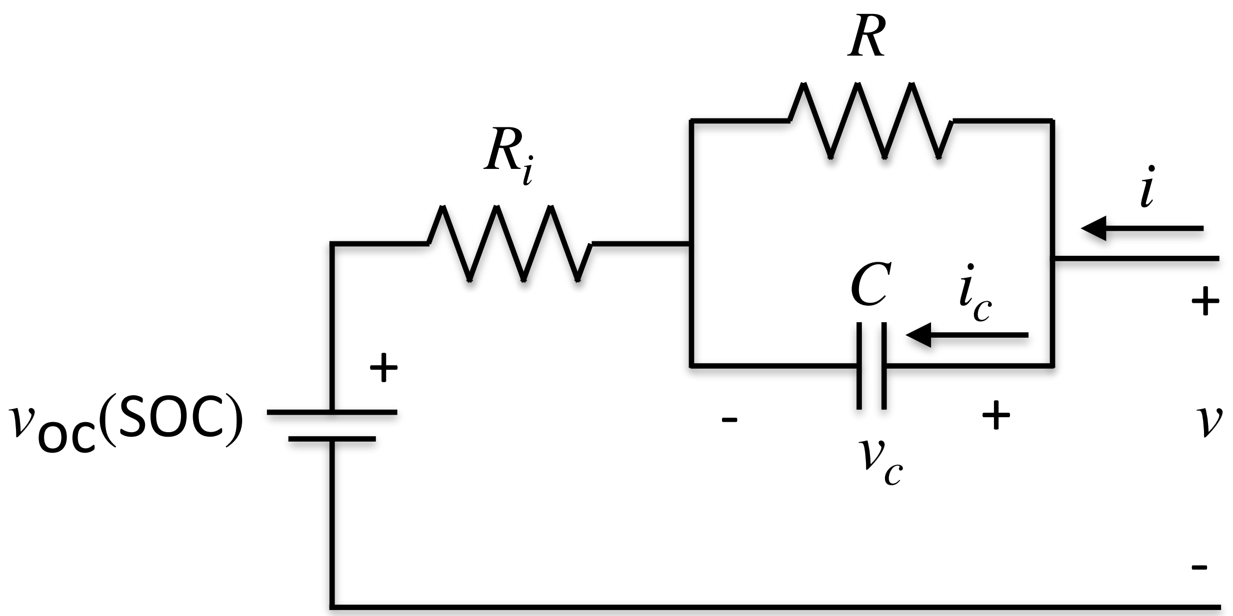
TSFS19 Battery Systems - Lectures 3 and 4

Mattias Krysander

LINKÖPINGS UNIVERSITET

Battery Modelling – Equivalent Circuit Models

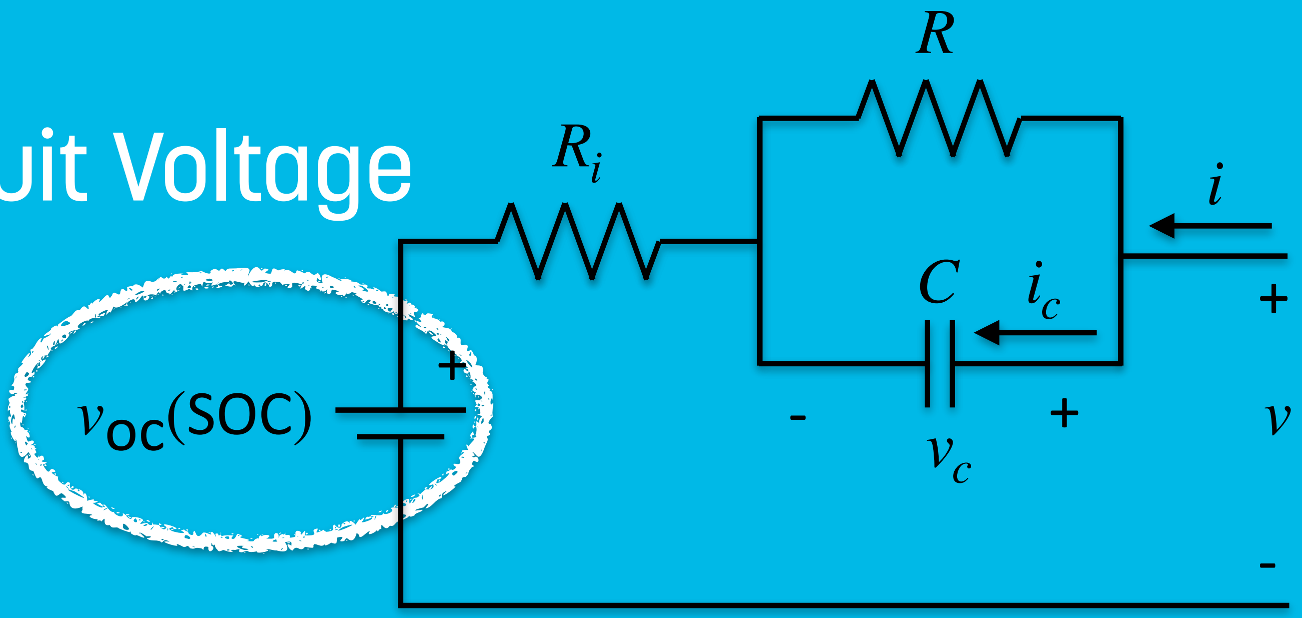
- Equivalent circuit model (ECM) describing the relationship between the voltage of the battery v and current i ($i > 0$ during charging)
- The model consists of 3 parts:
  - The charge capacity Q of the cell.
  - Open circuit voltage as a function of SOC, $v_{\text{oc}}(\text{SOC})$
  - Battery impedance describing the polarization losses.
- The parts can be modeled separately using different measurements.



Battery Modelling Lectures – Content

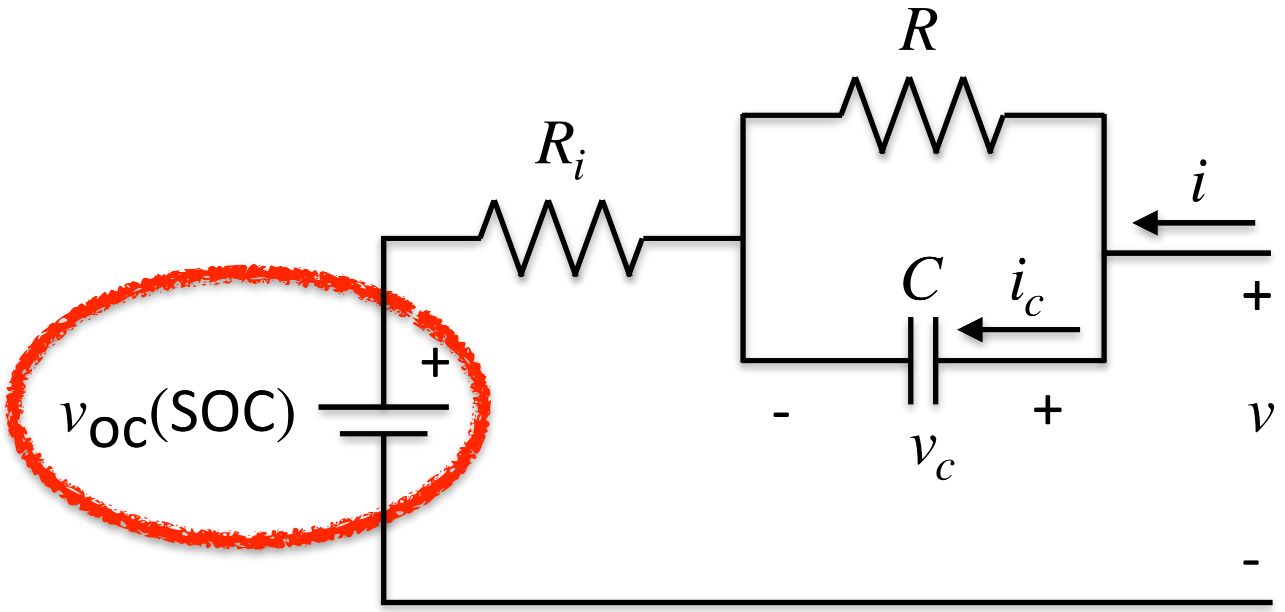
- Show **typical experiments** used for parameterization.
- **Identify battery capacity** from measurements.
- Identify the **open circuit voltage as a function of SOC** from measurement data.
- Convert a time-continuous model into a time-discrete model
- **Identify battery impedance** from
  - a current step using voltage response characteristics.
  - dynamic operation using linear regression and non-linear minimization methods.
- **Assumptions**:
  - Impedance is constant, i.e., does not significantly depend on SOC.
  - The OCV is only dependent on SOC.
  - Temperature is fixed to 25°C, so temperature dependence need not be considered.
- This and the next lecture prepare for the first computer lab where an ECM will be developed from measurement data.

Capacity and Open Circuit Voltage



Open Circuit Voltage

- Requirements for identifying OCV:
- Measurements from the entire SOC range, from fully charged to discharged battery.
- If the battery has rested ($i = 0$) long enough, $v_c = 0$ and $v_{OC}(\mathrm{SOC}) = v$ which can be measured.
- To save time in measuring the OCV curve, it is usually sufficient to use a small current.

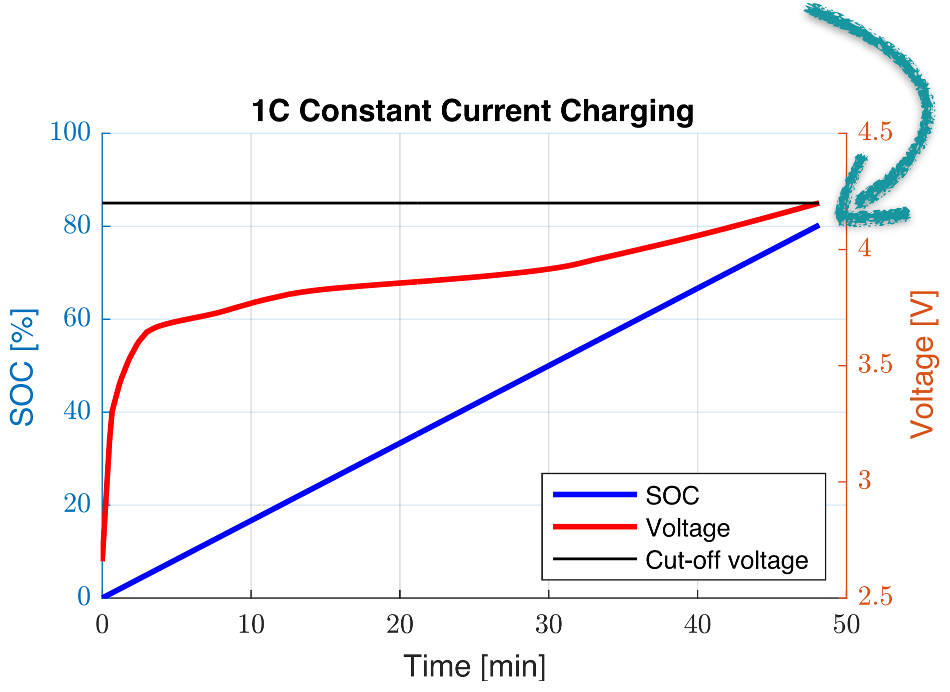


Capacity Estimation

- Definitions:
- The **capacity** is the amount of charge going from 0% SOC to 100% SOC.
- **0% SOC**: OCV = V_{min} (the minimum allowed terminal voltage)
- **100% SOC**: OCV = V_{max} (the maximum allowed terminal voltage)

- To get to 100%/0% SOC CCCV charge/discharge needs to be performed.
- A low current can be used to get close to 100%/0% SOC, the smaller the better.

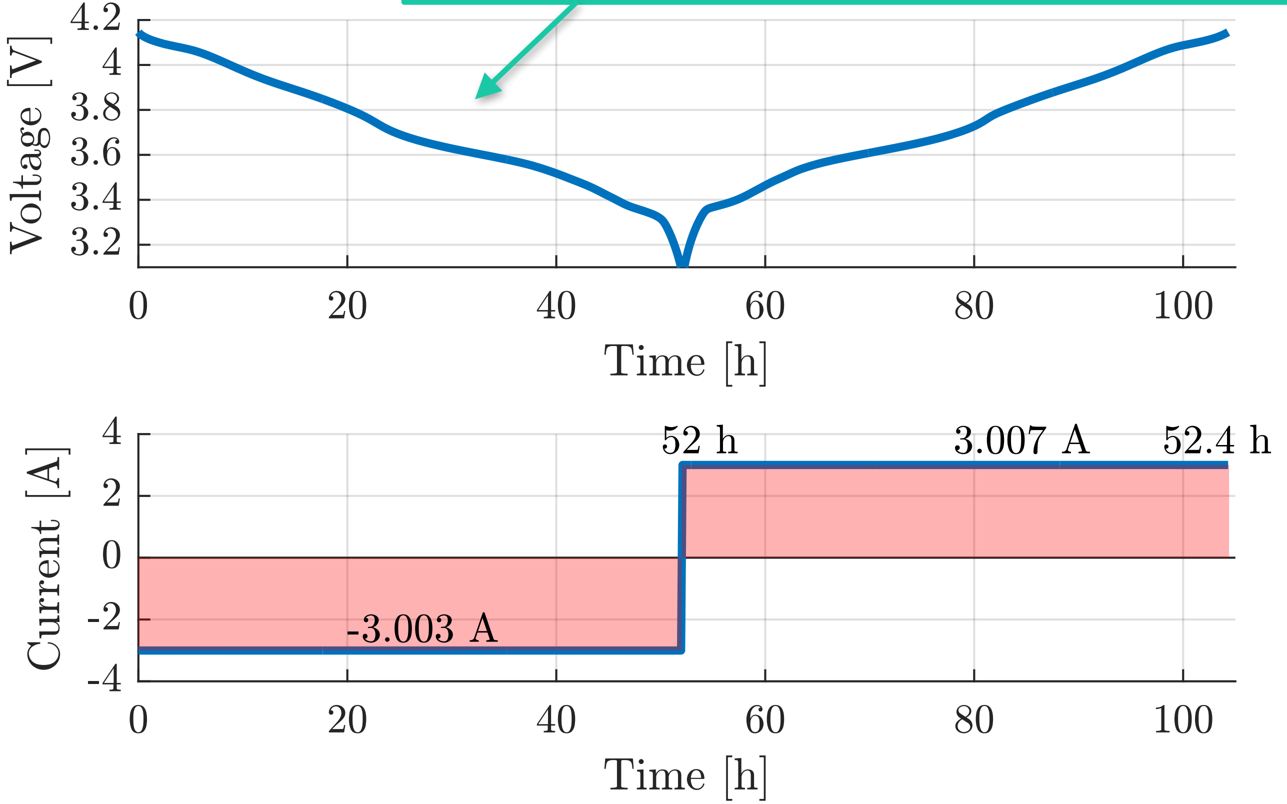
SOC is not 100 % after CC-charging



Cycling Test/Capacity Estimation

- Low current, here 3 A (C/50)
- Cycling
  - Discharge from maximum to minimum allowable voltage: 4.15 V->3.1V
  - Charging: 3.1V->4.15V
- The (discharge) capacity can be calculated:
  - $Q = -\int_{\text{full}}^{\text{empty}} i dt = -I\Delta t =$
  - $= 3.003 \cdot 52 = 156.2 \text{ Ah}$

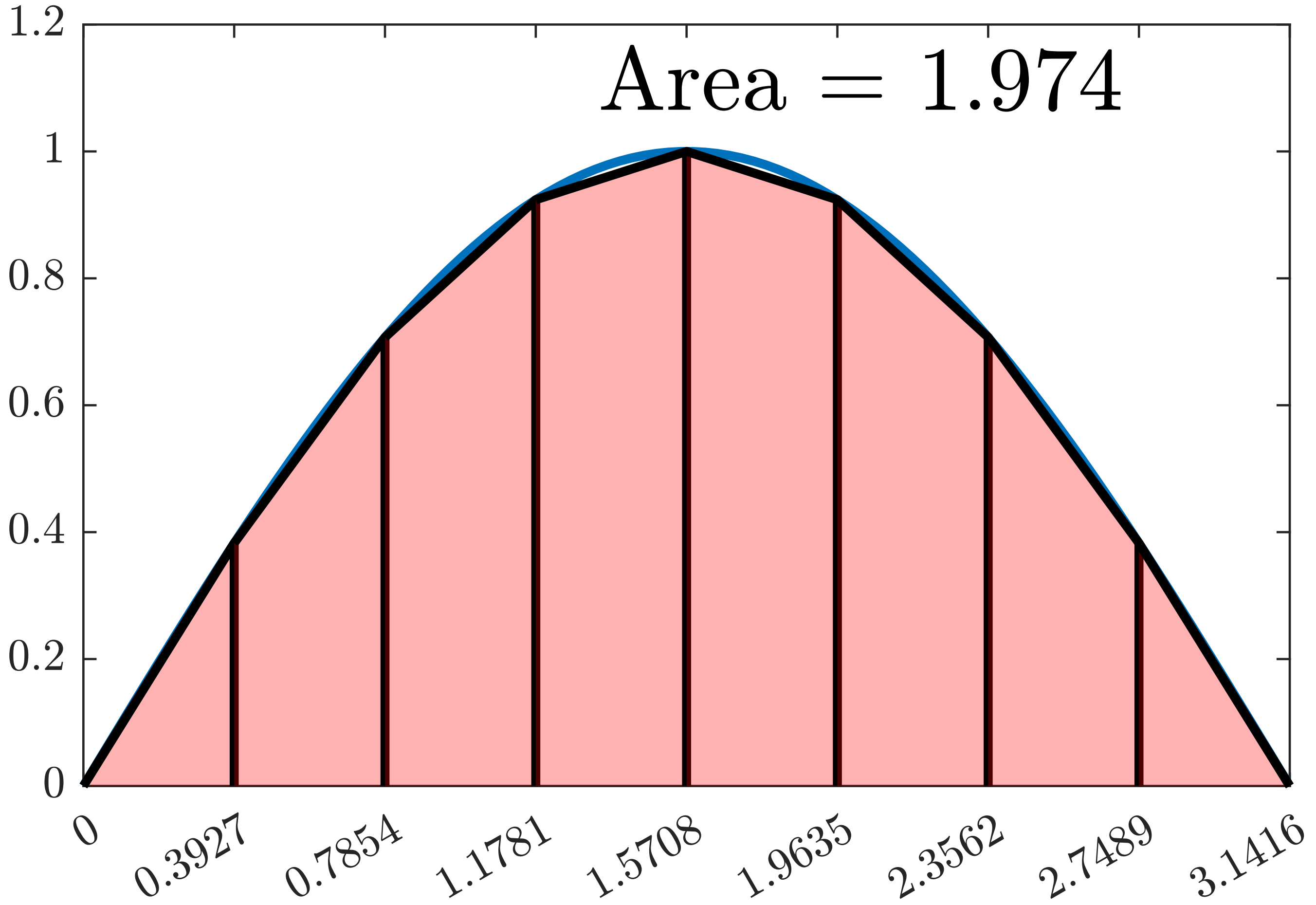
It looks like an OCV curve but plotted against time, not the battery's state of charge



Numerical Integration: The trapezoidal rule

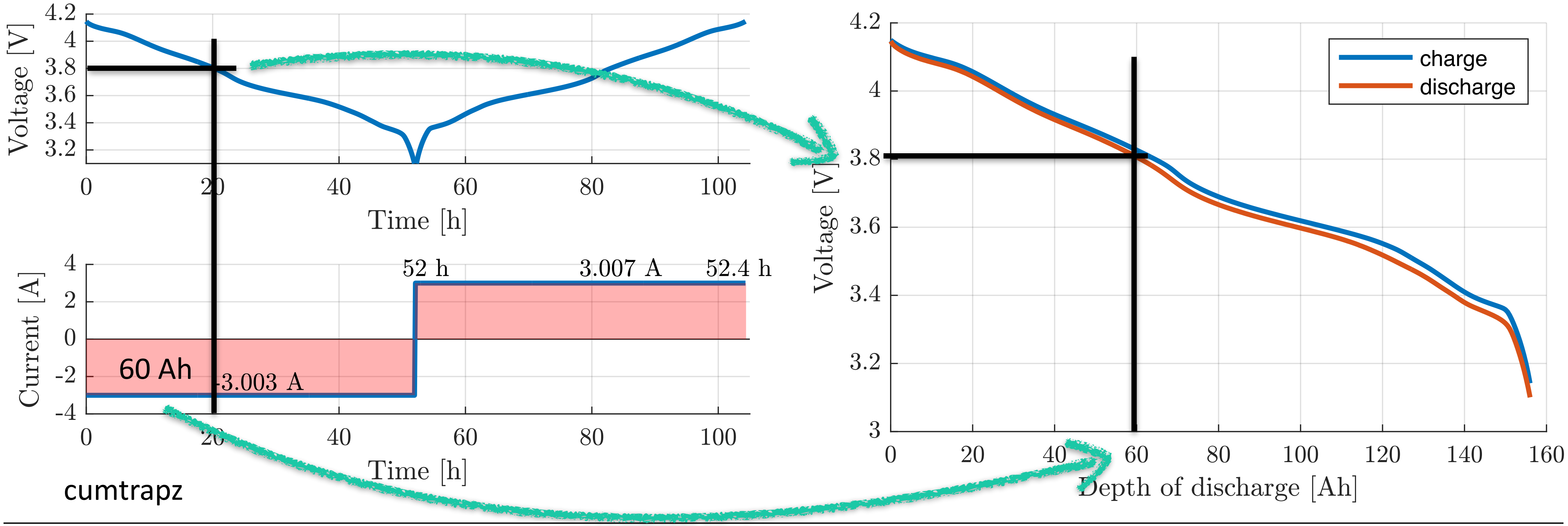
- Compute $\int_0^{\pi} \sin(x)\,dx =$
  $= [-\cos(x)]_0^{\pi} = 1 - (-1) = 2$
- Numerical approximation:

```
x = [0:pi/8:pi];
y = sin(x);
Area = trapz(x,y);
```



Voltage as a function of DoD

- Illustration of voltage calculation as a function of DoD.



Relationship between SOC and DOD

SOC is defined, as you may recall, by

$$z = \frac{\text{remaining charge}}{\text{total capacity}}$$

This can be expressed with the battery's total capacity Q_{tot} and DoD as

$$z = \frac{Q_{\text{tot}} - \text{DoD}}{Q_{\text{tot}}}$$

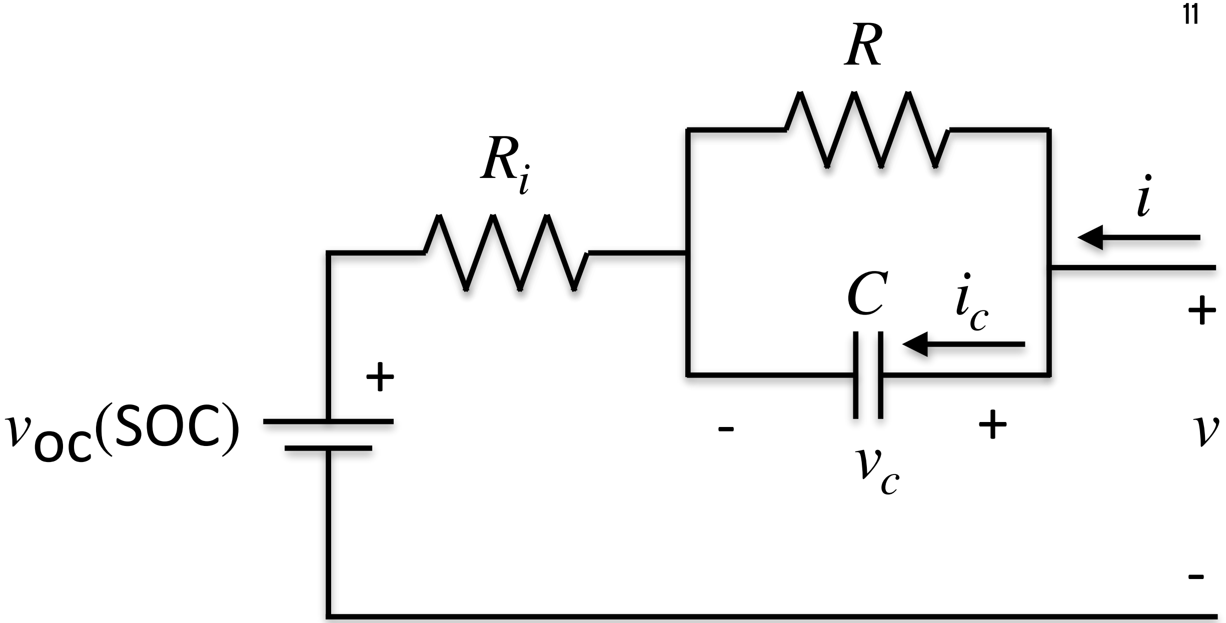
Open Circuit Voltage (OCV)

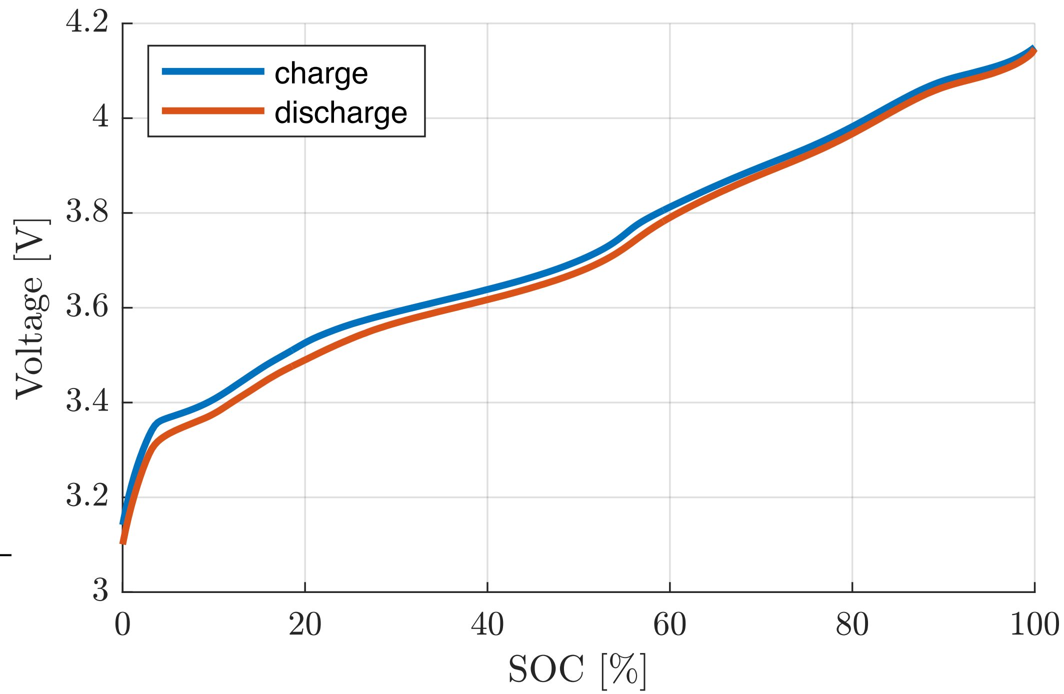
For constant current $I_{\text{dis}} < 0$ and $I_{\text{chg}} > 0$ during discharge and charge, the following applies:

$$v_{\text{dis}}(z) = v_{\text{oc}}(z) + I_{\text{dis}}(R_i(z) + R(z))$$
$$v_{\text{chg}}(z) = v_{\text{oc}}(z) + I_{\text{chg}}(R_i(z) + R(z))$$

If $I_{\text{dis}} = -I_{\text{chg}}$ then OCV can be approximated by

$$v_{\text{oc}}(z) \approx \frac{v_{\text{chg}}(z) + v_{\text{dis}}(z)}{2}$$





OCV-function for High SOC-values

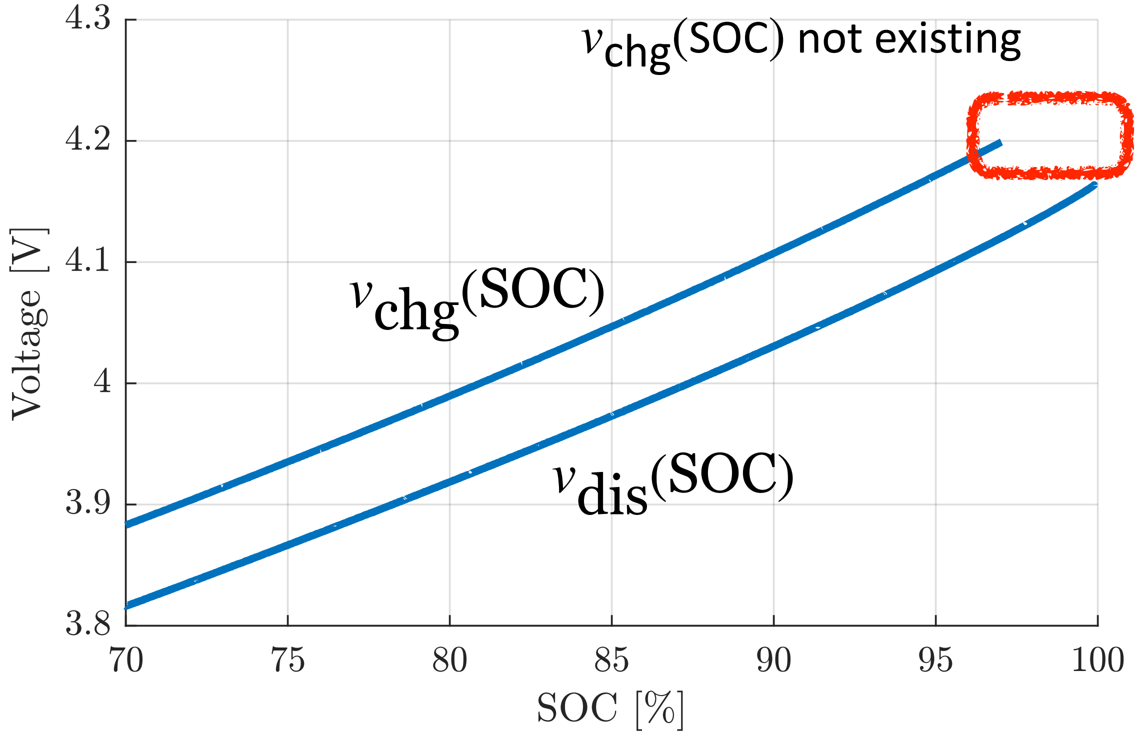
- $v_{\text{oc}}(\text{SOC}) \approx \dfrac{v_{\text{chg}}(\text{SOC}) + v_{\text{dis}}(\text{SOC})}{2}$ cannot be used for high SOCs, see figure.
- Assume the resistance is not changing much for high SOCs: $\tilde{R} = R_i(\text{SOC}) + R(\text{SOC})$
- From

$$v_{\text{dis}}(\text{SOC}) = v_{\text{oc}}(\text{SOC}) + I_{\text{dis}}\tilde{R}$$

$$v_{\text{chg}}(\text{SOC}) = v_{\text{oc}}(\text{SOC}) + I_{\text{chg}}\tilde{R}$$

we get $$\tilde{R} = \frac{v_{\text{chg}}(\text{SOC}) - v_{\text{dis}}(\text{SOC})}{I_{\text{chg}} - I_{\text{dis}}}$$

- Compute $\tilde{R}$ for a high SOC when both voltages exist.
- Estimate OCV as $v_{\text{oc}}(\text{SOC}) = v_{\text{dis}}(\text{SOC}) - I_{\text{dis}}\tilde{R}$
- Check also that $v_{\text{oc}}(100) = v_{\text{cut-off}}$



OCV-function Implementation

- In Matlab the OCV-function can be implemented as interpolation in a table as

```
% SOC = [0:0.01:1]; example SOC grid
% OCV(i) is the open circuit voltage at SOC(i)
OCVf = @(z) interp1(SOC,OCV,z)
SOCf = @(v) interp1(OCV,SOC,v)
```

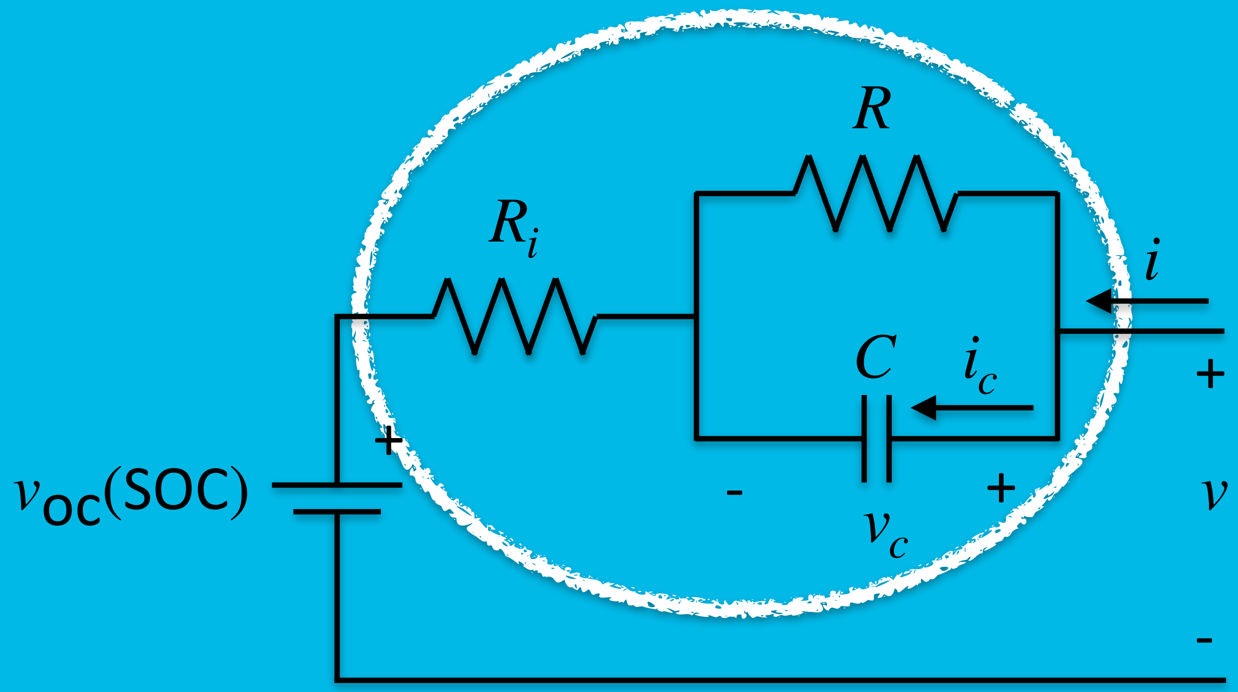
- Given a SOC-value z the OCV voltage ocv is computed as

```
ocv = OCVf(z)
```

- Given a open circuit voltage v the SOC-value soc is computed as

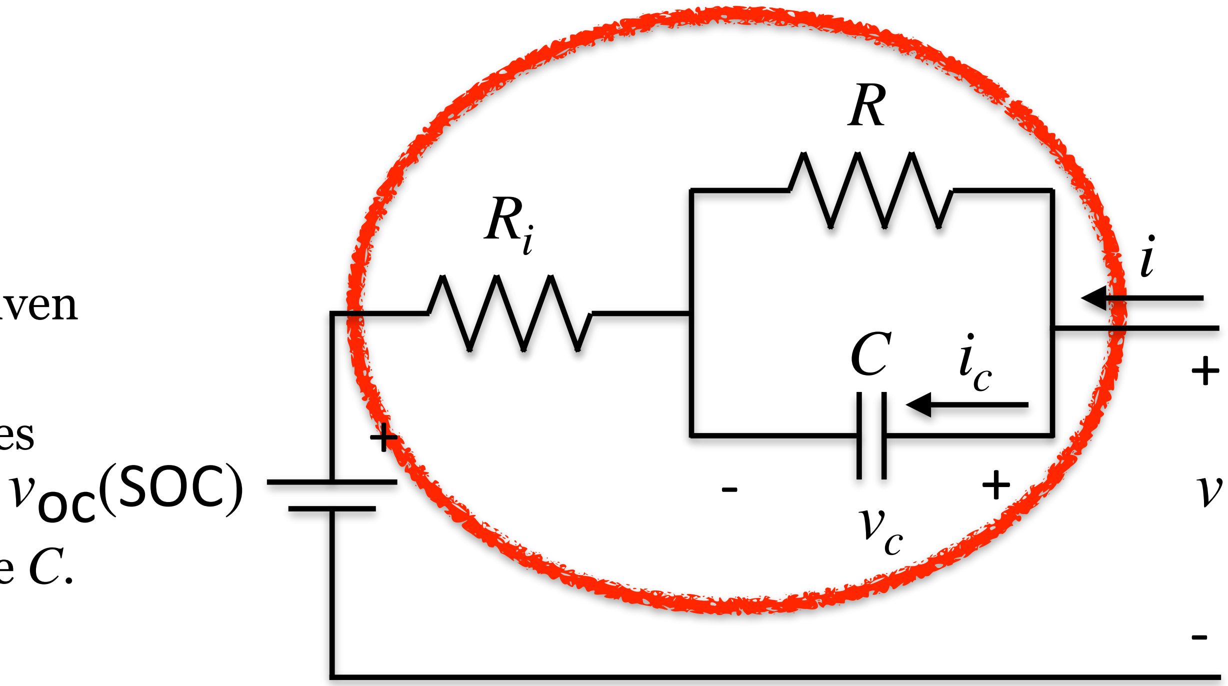
```
soc = SOCf(v)
```

Impedance Modeling



Impedance Estimation

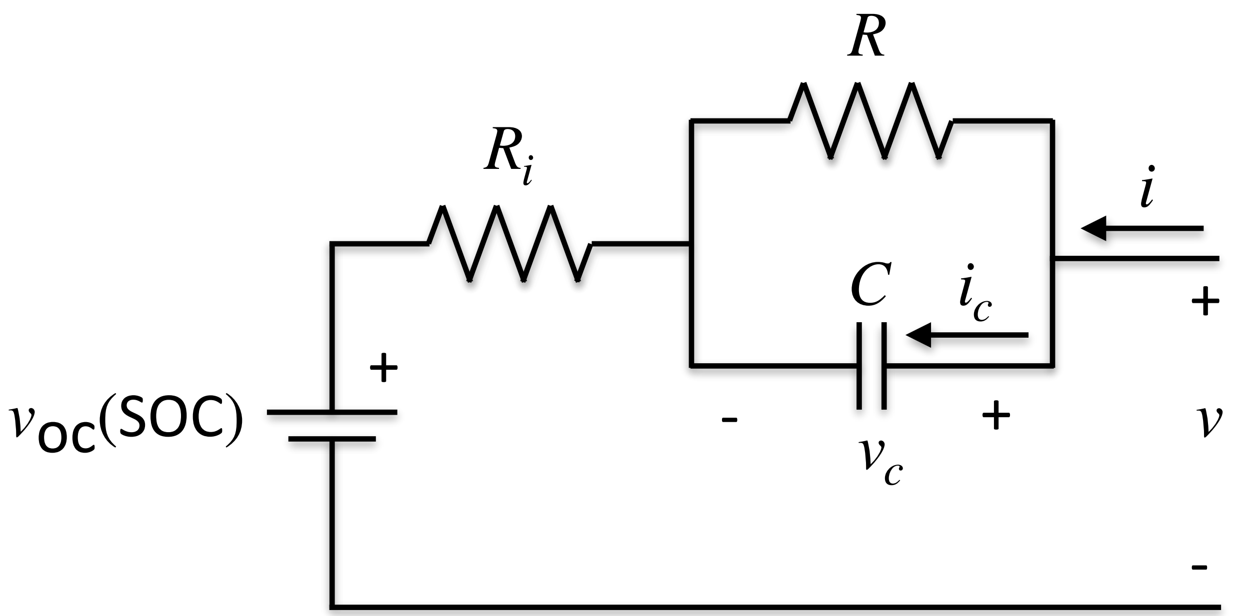
- The impedance in this model depends on the parameters R_i, R, and C.
- The parameters can depend on SOC and temperature.
- Next, a method for estimating parameters for a given SOC and temperature is described.
- Identify the impedance using data where SOC does not change significantly.
- The current must vary to estimate the capacitance C.
- Here, a step in current is chosen, but sinusoidal waves at different frequencies or white noise with a broad frequency content are also commonly used.



Fundamental Idea for Parameter Identification

1. Set up the battery model so that the voltage $v(t)$ can be estimated from the current
   $\hat{v}(t) = h(i(t)\,|\,\mathrm{SOC}, R, R_i, C)$
2. Calculate the parameters so that the model's estimated voltage $\hat{v}$ matches (as closely as possible) the measured voltage v

$$\min_{R,R_i,C} \|v - \hat{v}\|$$



RMSE - Root mean square error is one good option

Differential Equations in State-Space Form

Differential equations in the state-space form in continuous-time are written as

$$\frac{dx}{dt} = f(x(t), u(t)) \qquad \text{State equation}$$

$$y(t) = g(x(t), u(t)) \qquad \text{Measurement equation}$$

Where

- $x(t)$ is the state vector (the variables that are differentiated)
- $u(t)$ the input vector
- $y(t)$ the output vector
- f and g are known often non-linear functions.

If the input signal $u(t)$ for $t \geq 0$ and the initial state $x(0)$ is given the output $y(t)$ for $t \geq 0$ can be computed.

We want to write the battery model in state-space form with $i(t)$ as input and $v(t)$ as output.

Derivation of State-Space Form

States: z, v_c
Input: i
Output: v
Constants: R_i, R, C, Q
Function: $v_{OC}(z)$

Circuit laws give:

KVL: $v(t) = v_{OC}(z) + R_i\, i(t) + v_c(t)$ (1) (measurement equation)

Capacitance: $i_c(t) = C\dfrac{dv_c}{dt}(t)$ (2) (state equation for v_c)

KVL: $v_c = R(i - i_c) = Ri - Ri_c$ (3) (used for elimination of i_c in (2))

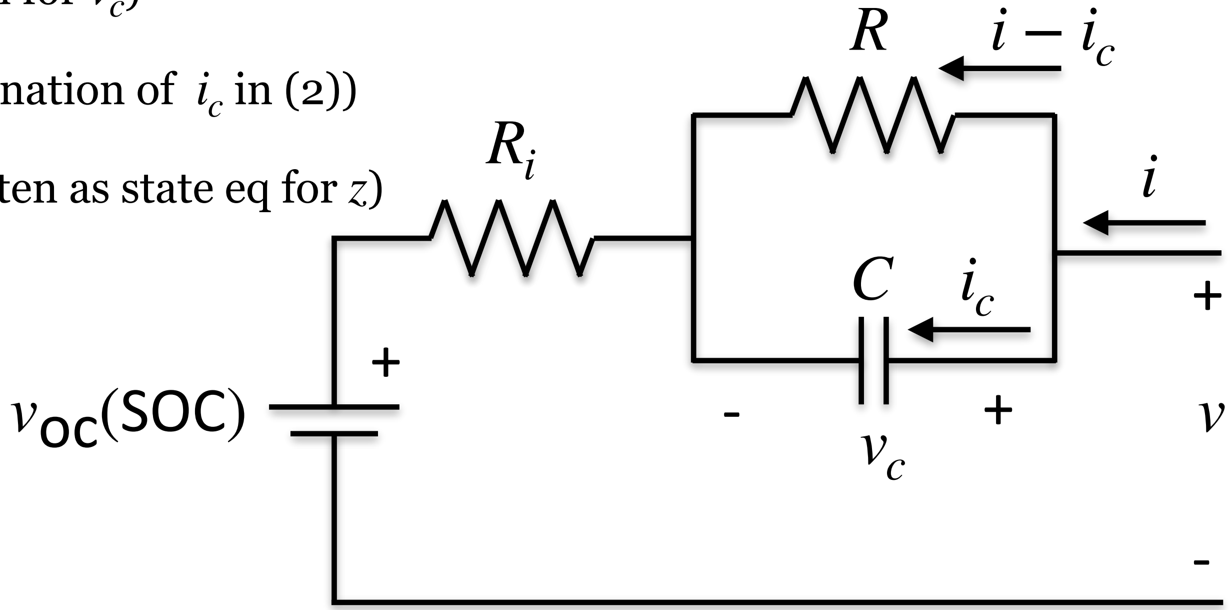
SOC: $z = \dfrac{1}{Q}\displaystyle\int i\,dt$ (4) (Can be rewritten as state eq for z)

State-space form:

$$\frac{dv_c}{dt} = -\frac{v_c}{RC} + \frac{i}{C} \qquad (5)$$

$$\frac{dz}{dt} = \frac{1}{Q} i \qquad (6)$$

$$v(t) = v_c + v_{OC}(z) + R_i\, i \qquad (7)$$



Model Parameters from Current Step Response

Problem Statement

Given the state-space form

$$\frac{dv_c}{dt}(t) = -\frac{v_c(t)}{RC} + \frac{i(t)}{C} \tag{5}$$

$$v(t) = v_c(t) + v_{\text{OC}}(t) + R_i\, i(t) \tag{7}$$

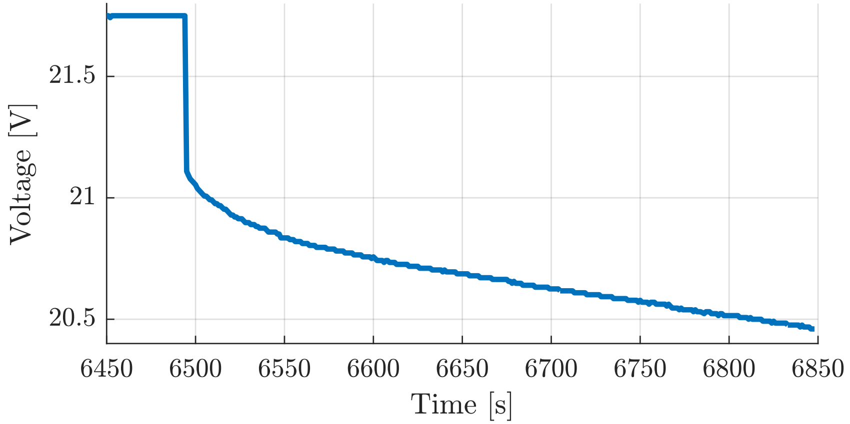
an input $i(t)$ and an initial value of the state $v_c(0)$ the output $v(t)$ can be computed.

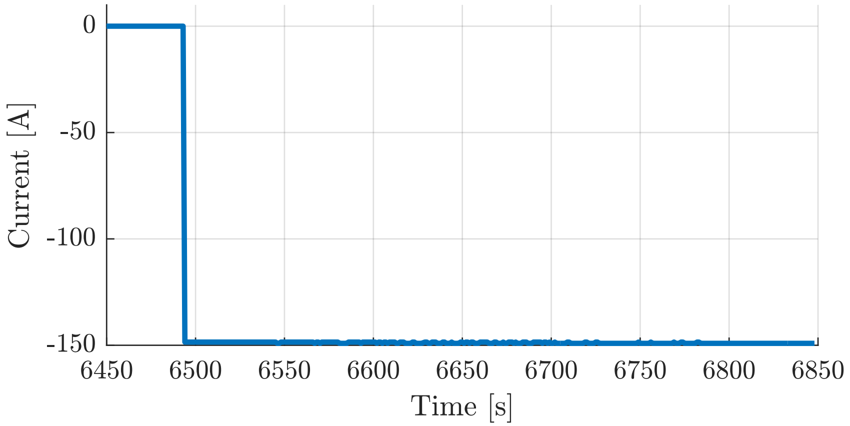
Assume a current step:

$$i(t) = \begin{cases} 0 & t < 0 \\ I & t \geq 0 \end{cases}$$

and initial condition $v_c(0) = 0\,\text{V}$, i.e.,

$$v(t) = v_c(t) + v_{\text{OC}} + R_i\, i(t) = v_{\text{OC}} \quad \text{för} \quad t < 0$$





Analytical Solution of Differential Equations

An inhomogeneous first-order differential equation is $\dot{y} + ay = f(t)$ and the solution is $y = y_h + y_p$ where $y_h = ke^{-ax}$ and y_p are determined using initial conditions and $f(t)$.

$$\frac{dv_c}{dt}(t) + \frac{v_c(t)}{RC} = \frac{I}{C} \qquad (8)$$

The homogeneous solution is $v_{c,h}(t) = ke^{-\frac{t}{RC}}$ and the particular solution is $v_{c,p}(t) = IR$.

$v_c(t) = v_{c,h}(t) + v_{c,p}(t) \quad => \quad v_c(t) = ke^{-\frac{t}{RC}} + IR$

The constant k is calculated from the initial condition $v_c(0) = 0$ which gives

$v_c(0) = ke^{-\frac{0}{RC}} + i(0)R \;<=>\; 0 = k \cdot 1 + IR \;<=>\; k = -IR$

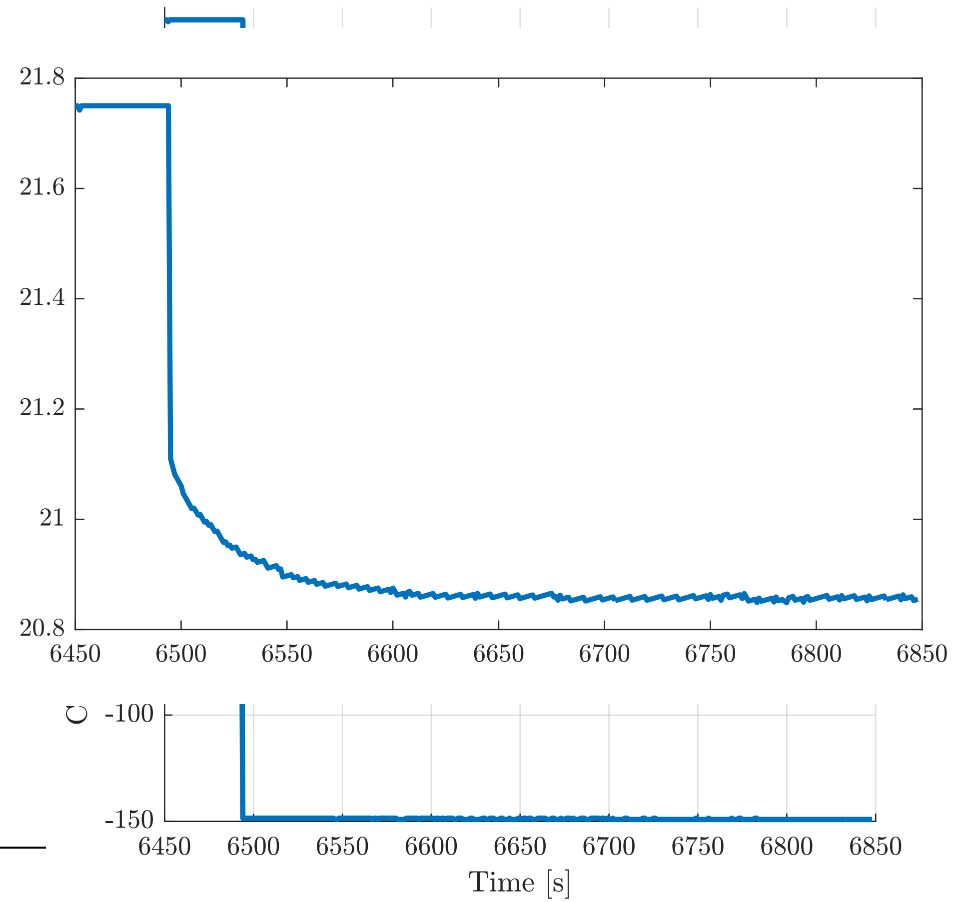
This gives the solution

$$v_c(t) = IR(1 - e^{-\frac{t}{RC}}) \qquad (9)$$

and if it is substituted into (7), it yields

$$\boxed{v(t) = \begin{cases} v_{\text{OC}} + RI(1 - e^{-\frac{t}{RC}}) + R_i I & t \geq 0 \\ v_{\text{OC}} & t < 0 \end{cases} \qquad (10)}$$

$v(t) = v_{\text{OC}} + \underbrace{RI + R_iI}_{\text{constant}}$ as $t \to \infty$ => v_{OC} decreases due to discharge. Assume a linear decrease and compensate for this.



Computation of Model Parameters

v_{OC}: $v(0-) = v_{OC} = 21.75\ \text{V}$

R_i: $v(0) = v_{OC} + IR_i = 21.11\ \text{V}$

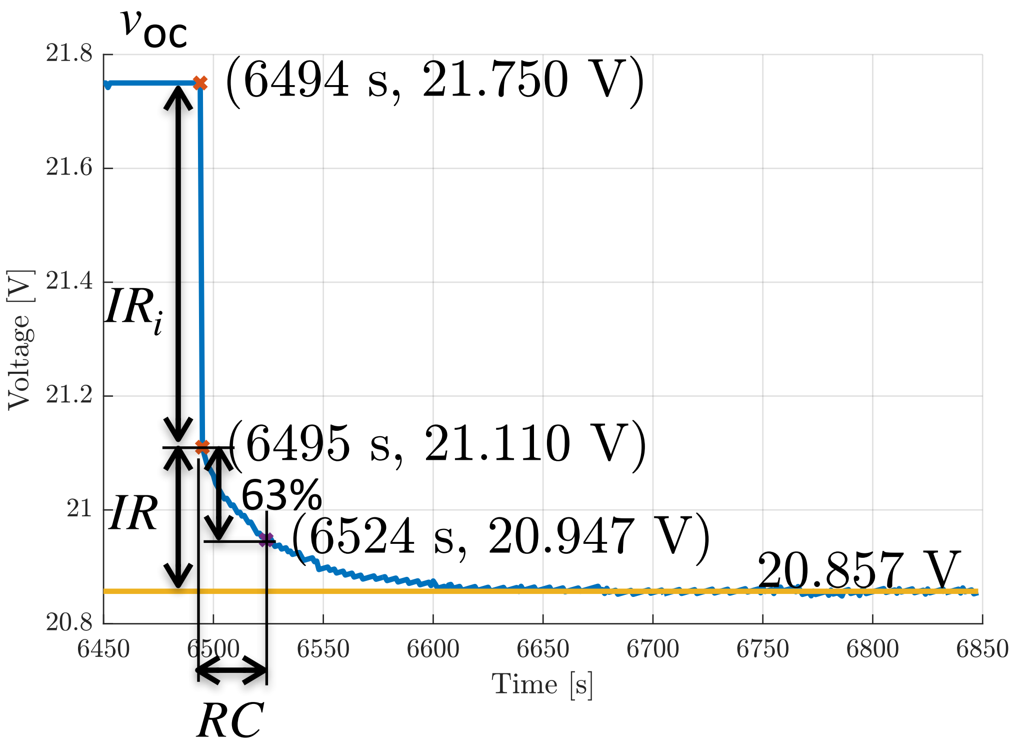
$$R_i = \frac{v(t) - v_{OC}}{I} = \frac{21.11 - 21.75}{-149} = 4.3\ \text{m}\Omega$$

R: $v(t) = v_{OC} + I(R_i + R(1 - e^{-\frac{t}{RC}})) \rightarrow v_{OC} + I(R_i + R)$ då $t \rightarrow \infty$

$$R = \frac{v(t) - v_{OC}}{I} - R_i = \frac{20.857 - 21.75}{-149} - 4.3 \cdot 10^{-3} = 1.7\ \text{m}\Omega$$

C: The time constant τ for the system is the point in time $t = \tau$ when $1 - e^{-1} = 63\ \%$ of the total change of the step response is reached. This gives:

$$\frac{\tau}{RC} = 1 \iff C = \frac{\tau}{R} = \frac{30}{1.7}\ \text{kF} = 17.6\ \text{kF}$$



$$v(t) = v_{OC} + I(R_i + R(1 - e^{-\frac{t}{RC}}))\ \text{for}\ t \geq 0$$

Simulation of a time-discretized model

Assume that all signals are sampled at $y(t), y(t+\Delta t), y(t+2\Delta t), \ldots$ and denote them $y_k, y_{k+1}, \ldots$.

$$\frac{dv_c}{dt}(t) = -\frac{v_c(t)}{RC} + \frac{i(t)}{C},$$

$$\frac{dv_c}{dt}(t) \approx \frac{v_c(t+\Delta t) - v_c(t)}{\Delta t} = \frac{v_{c,k+1} - v_{c,k}}{\Delta t}$$

(Forward Euler method)

$$v(t) = v_c(t) + v_{\text{oc}} + R_i\, i(t)$$

Time-discrete model

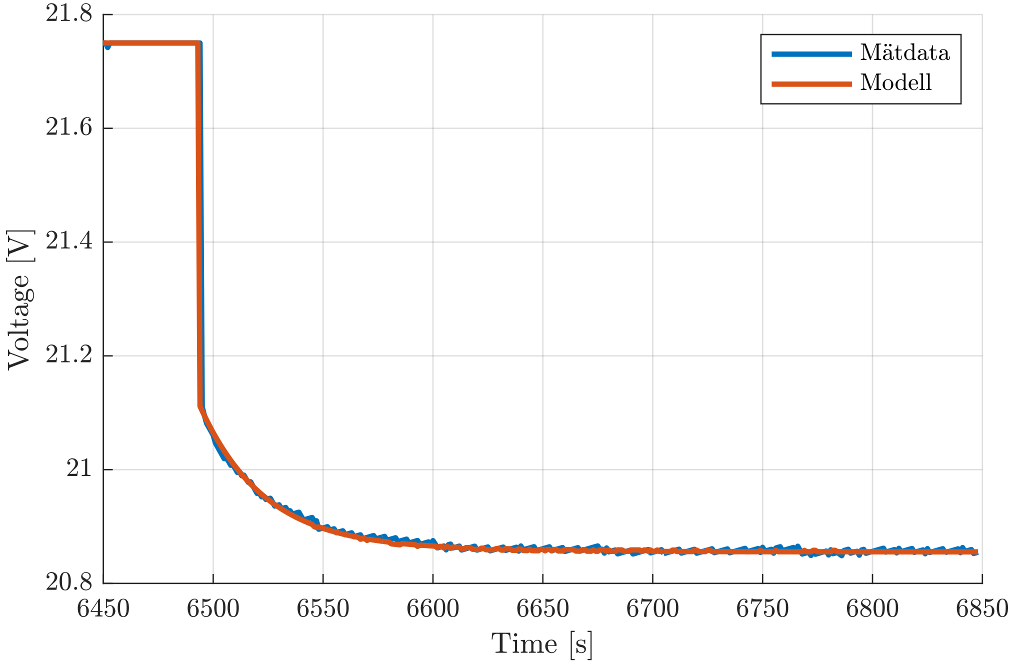
$$v_{c,k+1} = v_{c,k}\left(1 - \frac{\Delta t}{RC}\right) + \frac{\Delta t}{C} i_k$$

$$v_k = v_{c,k} + v_{\text{oc},k} + R_i\, i_k$$

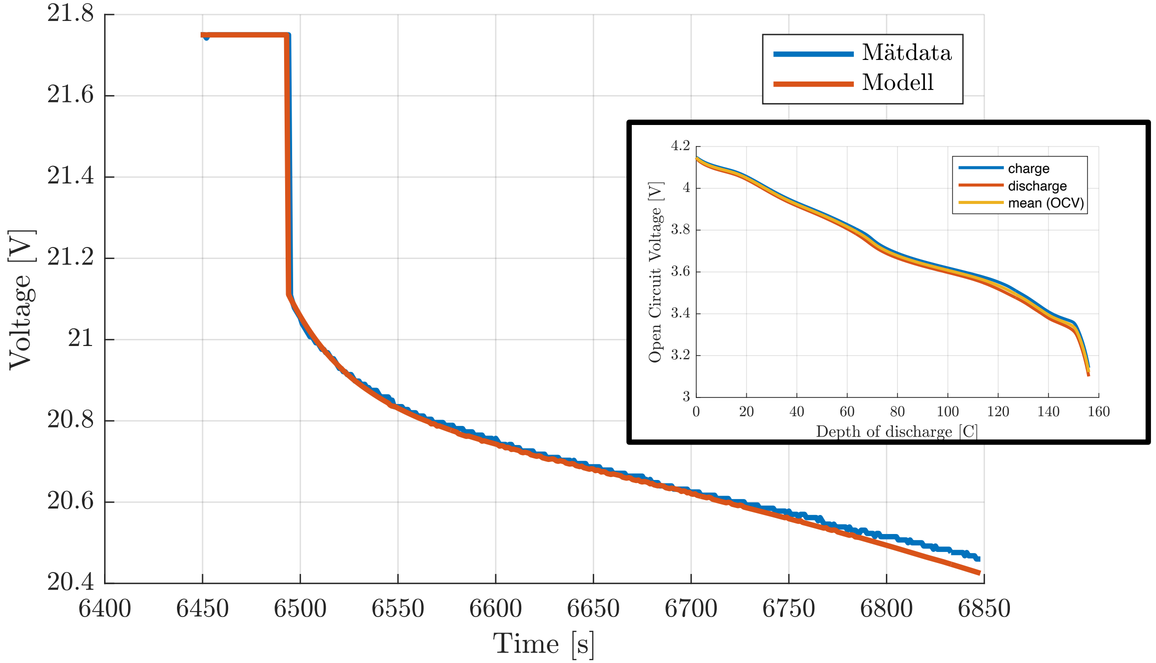
```
n = numel(i);   % Input i = [i_1, ..., i_n]'
vc = zeros(n,1);% Initial condition vc(1) = 0
v  = zeros(n,1);
for k = 1:n-1
    v(k) =  vc(k) + voc(k) + Ri*i(k);
    vc(k+1) = vc(k)*(1-dt/(R*C)) + dt/C*i(k);
end
v(n) =  vc(n) + voc(n) + Ri*i(n);
```

How good was the model?

The step response with compensation of v_{OC}



The step response without compensation of v_{OC} but where an estimated OCV curve is used to calculate $v_{\mathrm{OC}}(\mathrm{DOD})$

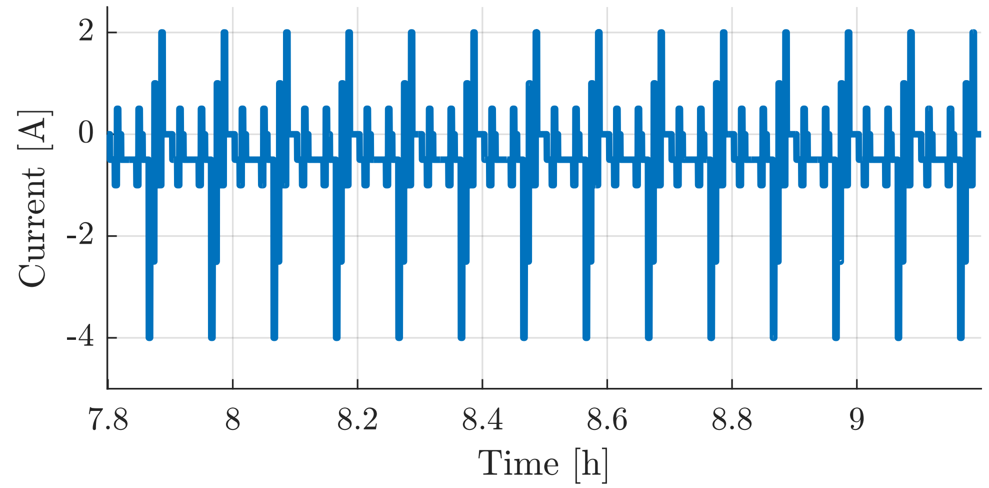


Model Parameters from Dynamic Operation

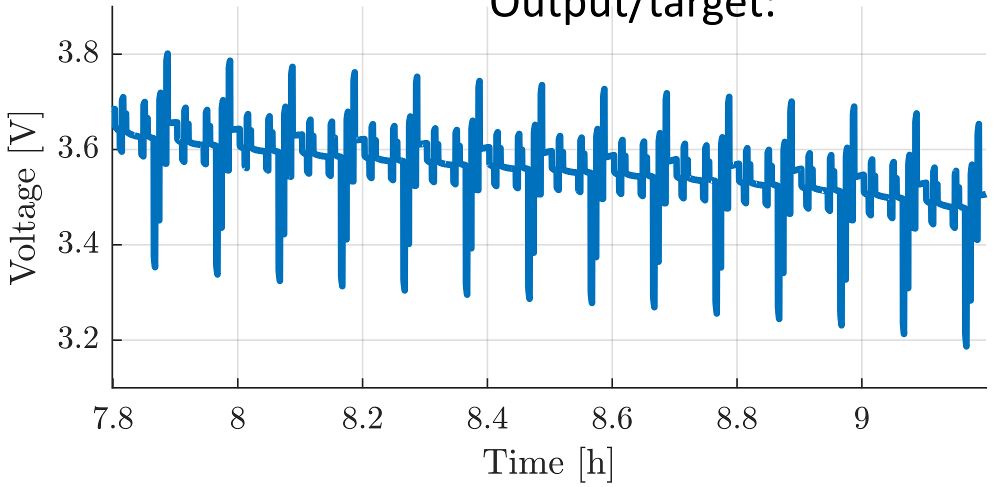
Linear Regression for Dynamic Operation

- Consider the data to the left.
- Estimation over a range of different SOC-levels.
- Same problem formulation:
    - Given: $\hat{v}(t) = h(i(t) \,|\, \mathrm{SOC}, R, R_i, C)$
    - Find: $\min_{R,R_i,C} \|v - \hat{v}(R, R_i, C)\|$
- Methods
    - Linear regression
    - Non-linear minimization methods.

Input:



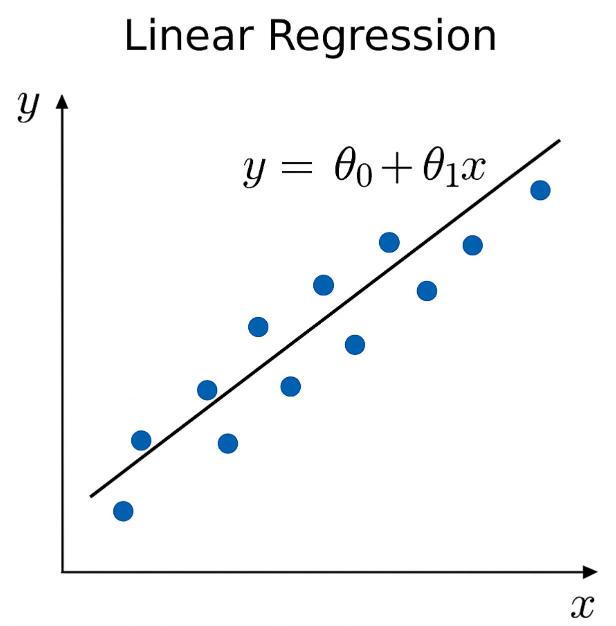
Output/target:



Linear Regression

Linear Regression

- Model equation $Y = X\theta + \varepsilon$ where
  - $Y \in \mathbb{R}^{n\times 1}$: vector of observed outputs
  - $X \in \mathbb{R}^{n\times p}$: matrix of observed inputs
  - $\theta \in \mathbb{R}^{p\times 1}$: parameter vector to be identified
  - $\varepsilon \in \mathbb{R}^{n\times 1}$: error/noise
- Objective:
  - $\hat{\theta} = \min_{\theta} \| Y - X\theta \|^2$
- Closed-form solution
  - $\hat{\theta} = \left(X^\top X\right)^{-1} X^\top Y$ (In Matlab: Theta = X\Y;)
- Model prediction:
  - $\hat{Y} = X\hat{\theta}$



Some Properties

Cons:

- Limited to/assume linear models

Pros:

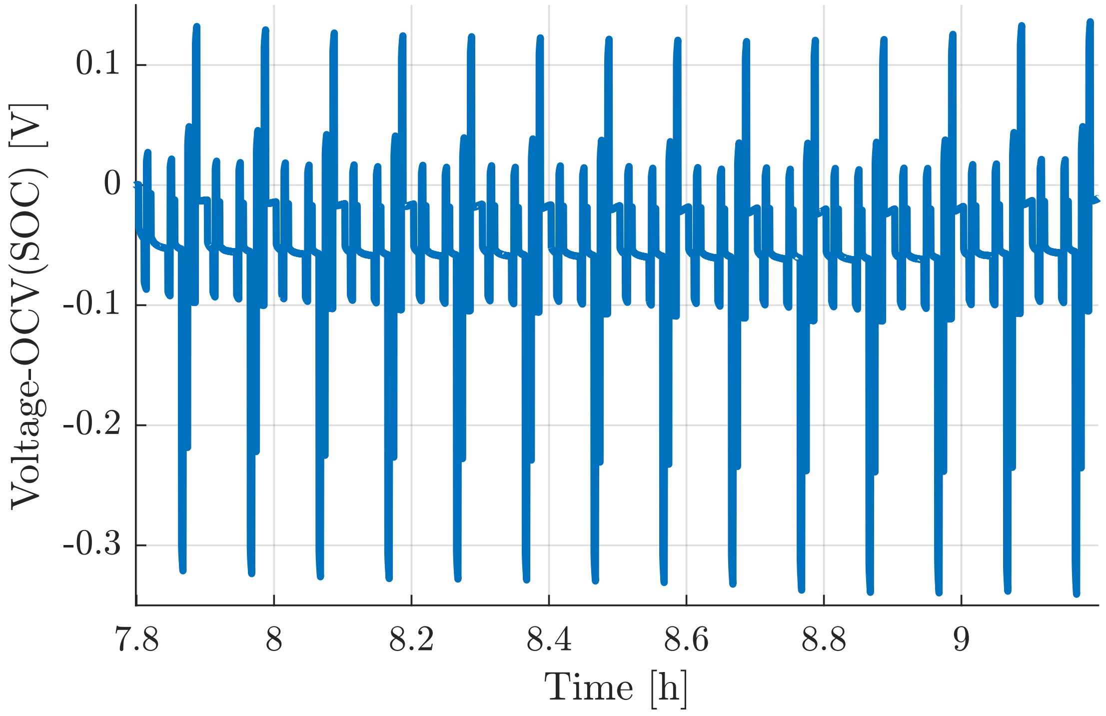
- Unique solution (if X har full column rank)
- Fast to compute
- Good baseline model

Remove Trend Caused by Changing SOC

- The OCV-function is not linear, but since it has been identified the impedance can be computed using linear regression.
- Alt 1: Use estimated OCV-function and Coulomb-counting

```
n = numel(i); % Insignal v = [v_1, ..., v_n]'
z = SOCf(v(1)); % Initial SOC given by voltage
for k = 1:n-1
    z(k+1) = z(k) + dt * i(k)/Q;
end
v_tilde = v - OCVf(z);
```

- Alt 2: Use a general detrend-function.
- The remaining part is a linear system.



Parameter Estimation using Linear Regression

- $Y = X\theta$ where

$$Y = \begin{bmatrix} \frac{\tilde{v}_2 - \tilde{v}_1}{\Delta t} \\ \vdots \\ \frac{\tilde{v}_n - \tilde{v}_{n-1}}{\Delta t} \end{bmatrix}, X = \begin{bmatrix} \frac{i_2 - i_1}{\Delta t}. & -\tilde{v}_1 & i_1 \\ \vdots & \vdots & \vdots \\ \frac{i_n - i_{n-1}}{\Delta t} & -\tilde{v}_{n-1} & i_{n-1} \end{bmatrix}, \text{and } \theta = \begin{bmatrix} R_i \\ \frac{1}{RC} \\ \frac{R_i}{RC} + \frac{1}{C} \end{bmatrix}$$

In Matlab solved by: `Theta = X\Y;`

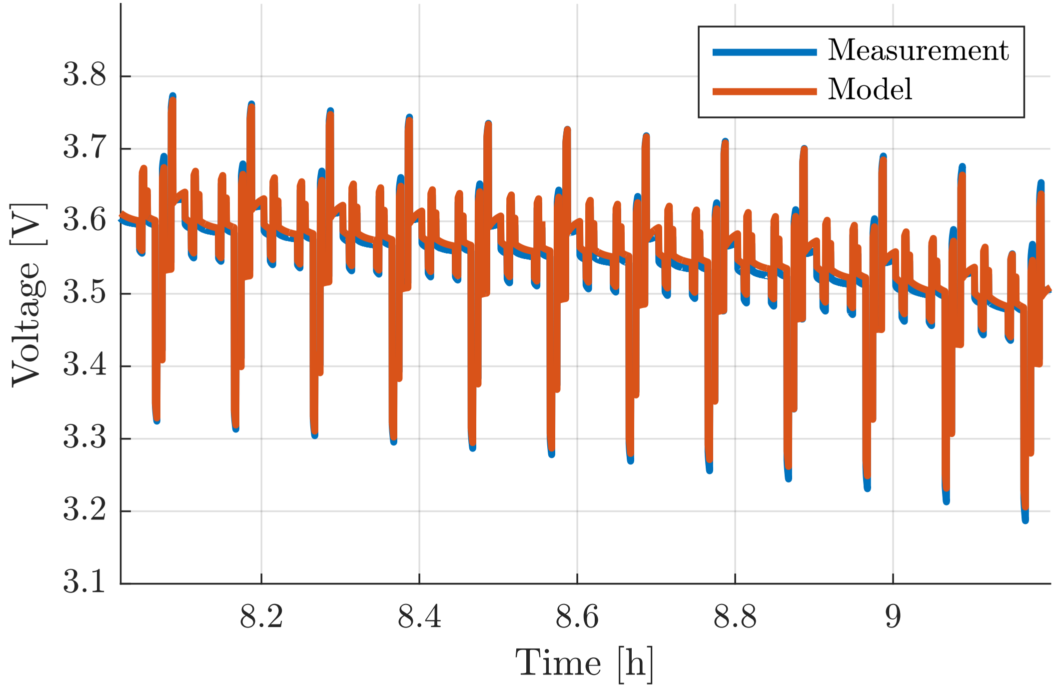
Parameters obtained by solving the non-linear equation system:

$$R_i = \theta_1, C = \frac{1}{\theta_3 - \theta_1\theta_2}, \text{and } R = \frac{\theta_3}{\theta_2} - \theta_1$$

Simulation of Fitted Model

- An example code for simulating the fitted model.

```
n = numel(i);    % Insignal i = [i_1, ..., i_n]'
vc = zeros(n,1);% Initial condition vc(1) = 0
z = SOCf(v(1)); % Initial SOC given by voltage
v  = zeros(n,1);
for k = 1:n-1
    v(k) = vc(k) + OCVf(z(k)) + Ri*i(k);
    z(k+1) = z(k) + dt * i(k)/Q;
     vc(k+1) = vc(k)*(1-dt/(R*C)) + dt/C*i(k);
end
v(n) =  vc(n) + OCVf(z(n)) + Ri*i(n);
```



Non-linear Minimization Methods

Fminunc, x = fminunc(f,x0). (Unconstrained solver)

Objective: Find the local minimum of a function $f(\mathbf{x})$ with a start in $\mathbf{x}_0$

- The default solver is the BFGS quasi-Newton method.

Update Rule: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k = \mathbf{x}_k - \alpha_k \mathbf{H}_k^{-1} \nabla f(\mathbf{x}_k)$

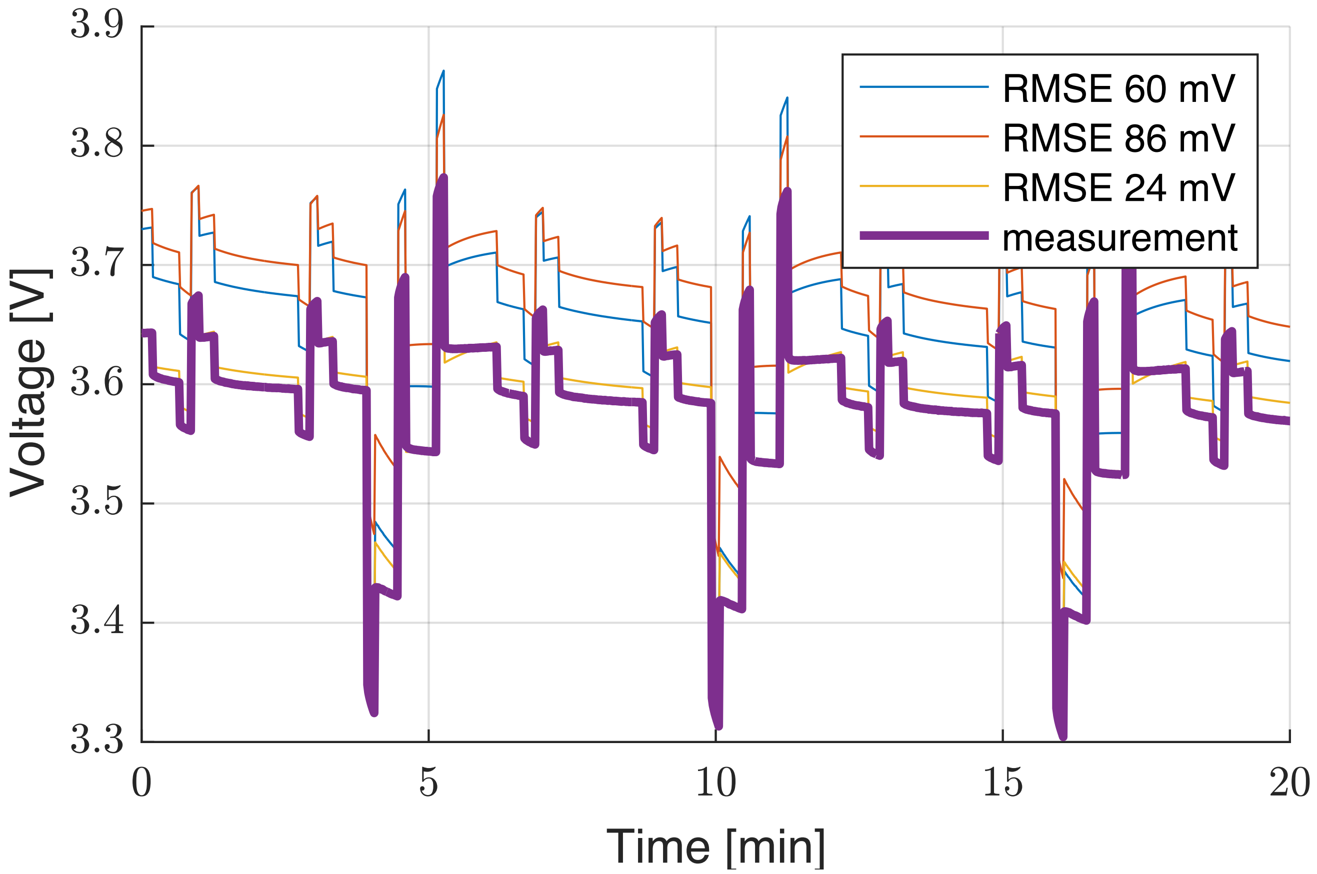
- $\mathbf{H}_k$: Approximation of Hessian matrix.
- $\nabla f(\mathbf{x}_k)$: Gradient of the function.
- α_k: Step size (determined by line search)

Summary Workflow

- Compute search direction from gradient and Hessian approximation $\mathbf{H}_k \mathbf{p}_k = -\nabla f(\mathbf{x}_k)$
- Stepsize α_k optimization
- Step update $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
- Hessian approximation update $\mathbf{H}_{k+1}$

Battery Parameter Estimation by Function Minimisation

- $f(\mathbf{x})$: A scalar measure of model fit.
  - Rot Mean Square Estimate (RMSE)
- $\mathbf{x}$: A vector of the parameters to be estimated.
- Example of simulation with different parameter values.

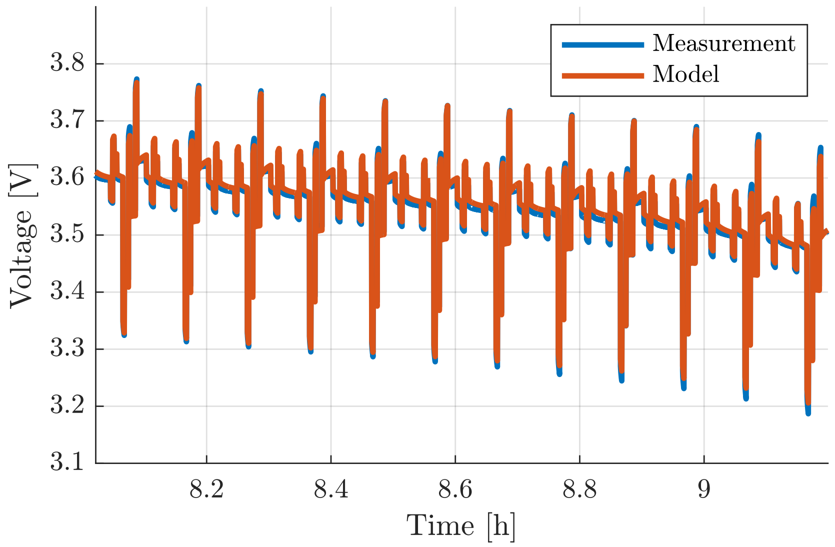


Define Optimization Function

ECM1RC.m

```
function [r,vhat] = ECM1RC(x,data,OCVf)
% Init
v = data.v;
i = data.i;
dt = diff(data.t);
% Set parameters
Ri = x(1);
C  = x(2);
R  = x(3);
% Simulate model
n = numel(i);
vc = zeros(n,1);% Initial condition vc(1) = 0
z = SOCf(v(1)); % Initial SOC
vhat  = zeros(n,1);
for k = 1:n-1
    vhat(k) = vc(k) + OCVf(z(k)) + Ri*i(k);
    z(k+1) = z(k) + dt(k) * i(k)/Q;
    vc(k+1) = vc(k)*(1-dt(k)/(R*C)) + dt(k)/C*i(k);
end
vhat(n) =  vc(n) + OCVf(z(n)) + Ri*i(n);
% Evaluate mean square error
r = mean((v-vhat).^2);
```

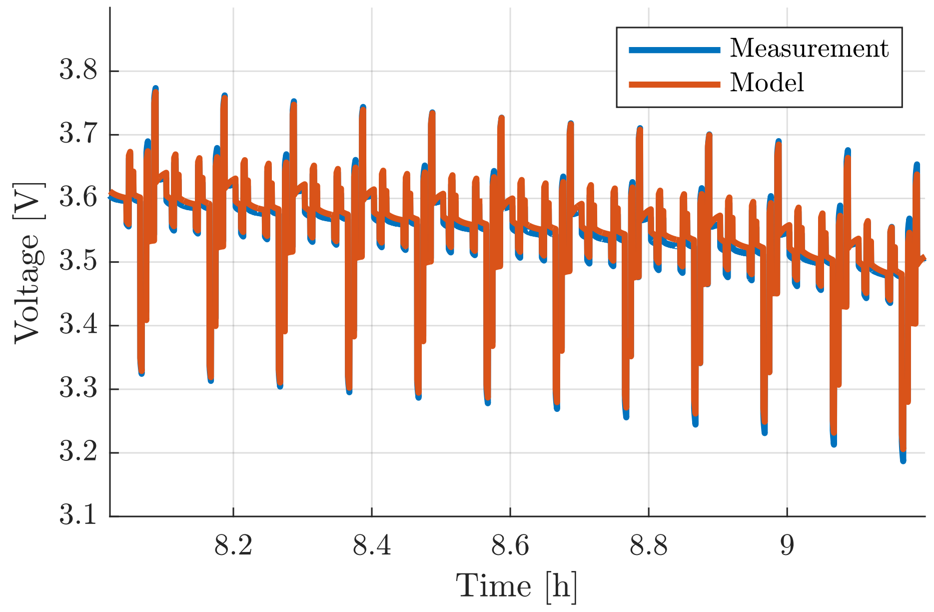
```
% Define a function with only
% parameter values x as input
fun = @(x) ECM1RC(x,data,OCVf);

% Find minimizing values
[x,r] = fminunc(fun,x0)
```



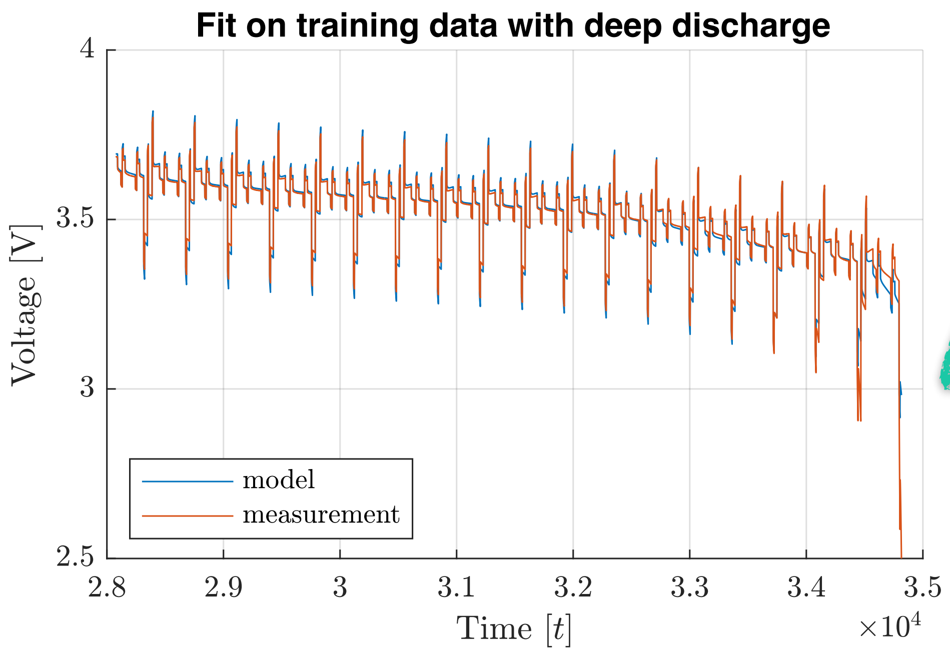
Setting up the Minimization Problem

- Selection of training data

High SOC and OCV-function dependence near fully discharged



Non-deep discharge



Deep discharge (difficult)

Extending The Parameter Set

With non-linear techniques, it is easy to include, e.g.,

- the initial SOC as an additional parameter to optimize.
- the capacity Q

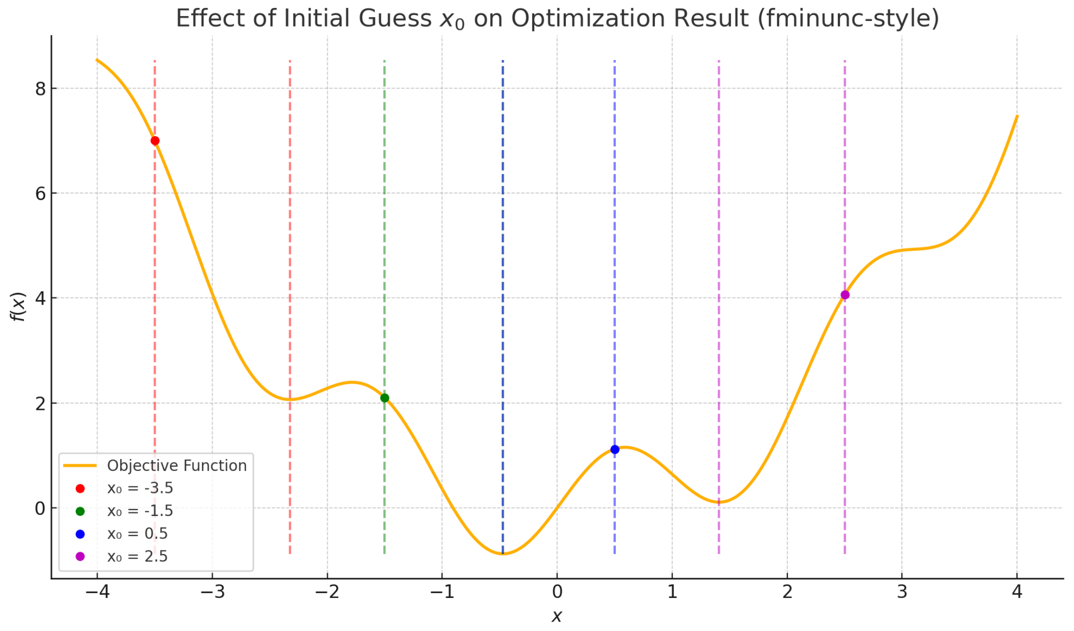
However, introducing more parameters makes the minimization problem more complex with possible local minima resulting in wrong solutions.

```
% Set parameters
Ri = x(1);
C  = x(2);
R  = x(3);
z  = x(4); % Initial SOC
Q  = x(5);

% Simulate model
n = numel(i);
vc = zeros(n,1);% Initial condition vc(1) = 0
vhat  = zeros(n,1);
for k = 1:n-1
    z(k+1) = z(k) + dt(k) * i(k)/Q;
    vhat(k) = vc(k) + OCVf(z(k)) + Ri*i(k);
    vc(k+1) = vc(k)*(1-dt(k)/(R*C)) + dt(k)/C*i(k);
end
vhat(n) =  vc(n) + OCVf(z(n)) + Ri*i(n);
```

Why is Initial Guess x_0 Important?

```
[x,r] = fminunc(fun,x0)
```

- These methods are local, not global — so they only "see" the landscape near x_0
- A poor initial guess can cause the optimizer to converge to a **bad local minimum**, or even fail to converge.
- A good x_0 (i.e., one near the true minimum) can drastically reduce the number of iterations and function evaluations.



Normalization

- The parameters are of different orders of magnitude, e.g., $R \approx 10\ \mathrm{m\Omega}$, $C \approx 1\ \mathrm{kF}$.
- Normalization is to rescale the parameters to the same size.
- Normalized parameters are easier to optimize with possible improvements in both the speed of convergency and accuracy.
- Implementation:
  - Original problem: $x^* = \arg\min_x f(x)$
  - Assume x_0 is a good guess for $x^* = z \circ x_0$ (elementwise product), then $z_i \approx 1$
  - Solve the normalized problem: $z^* = \arg\min_z f(z \circ x_0)$
  - Recover the original parameters $x^* = z^* \circ x_0$
- Normalization is similar to feature scaling in machine learning.

Lab 1 - Equivalent Circuit Modeling

Lab information

- Register lab groups at Lisam before Thursday at 16:00.
- Nominal time: 2 + 2 h scheduled sessions
- Lab 1 and 2 will be in Python
- Scheduled in normal classrooms, **work on your laptop**
- Material for lab 1:
    - Python_installation.md is a quick guide to get started
    - Instructions with questions to answer
    - Code skeleton where the tasks are solved by completing the code
    - Datasets including slow charge cycles and dynamic tests

Learning Outcomes

- **Compute battery capacity** from measurements.
- Compute **open circuit voltage as a function of SOC** from measurement data.
- Convert a time-continuous model into a time-discrete model
- **Compute battery impedance** from
  - A current step using voltage response characteristics.
  - Dynamic operation using
    - linear regression
    - nonlinear minimization techniques
- Describe **experiments** for parameter identification of the different parts of the ECM.

TSFS19 Battery Systems - Lectures 3 and 4

Mattias Krysander

www.liu.se

LINKÖPINGS UNIVERSITET