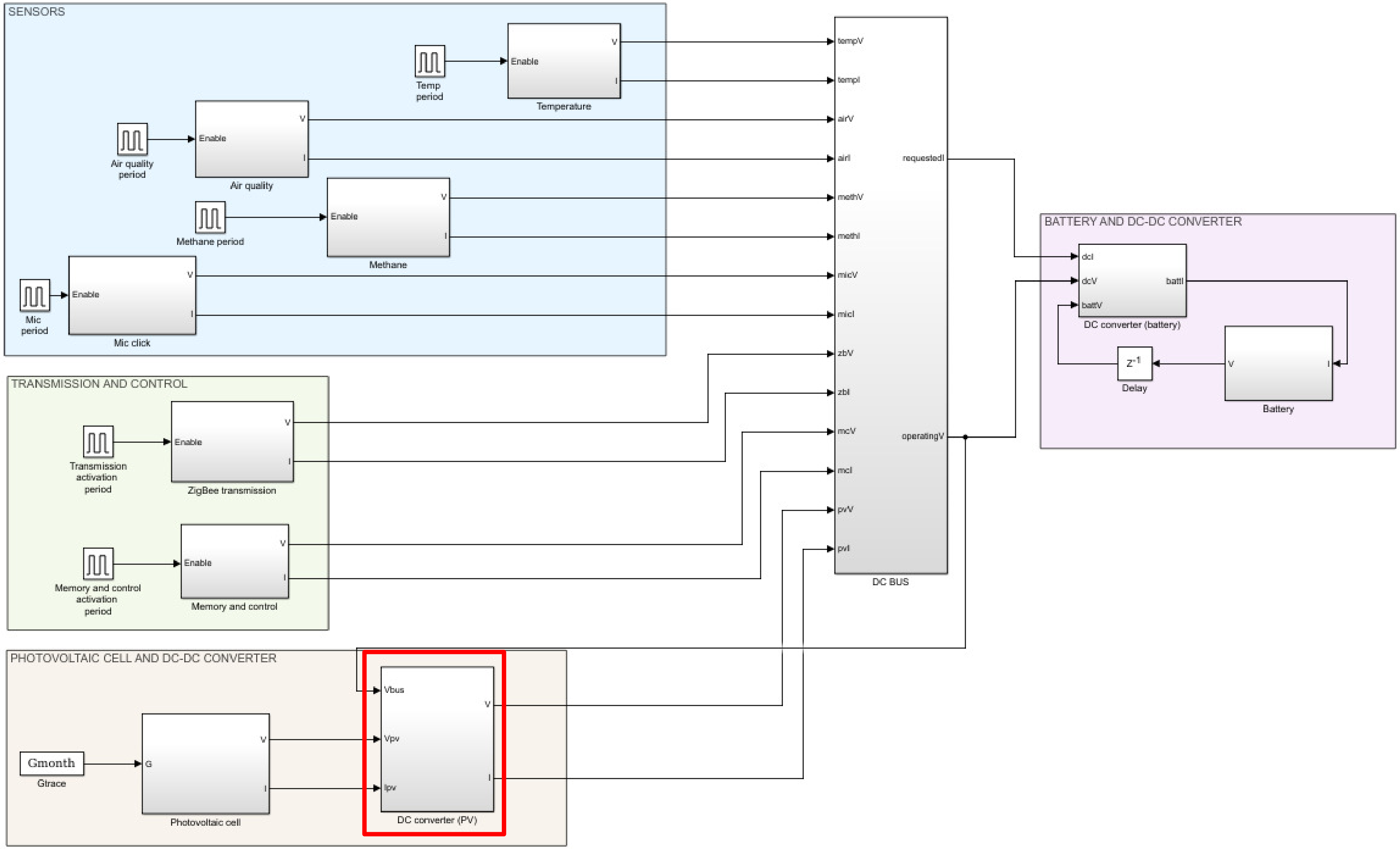
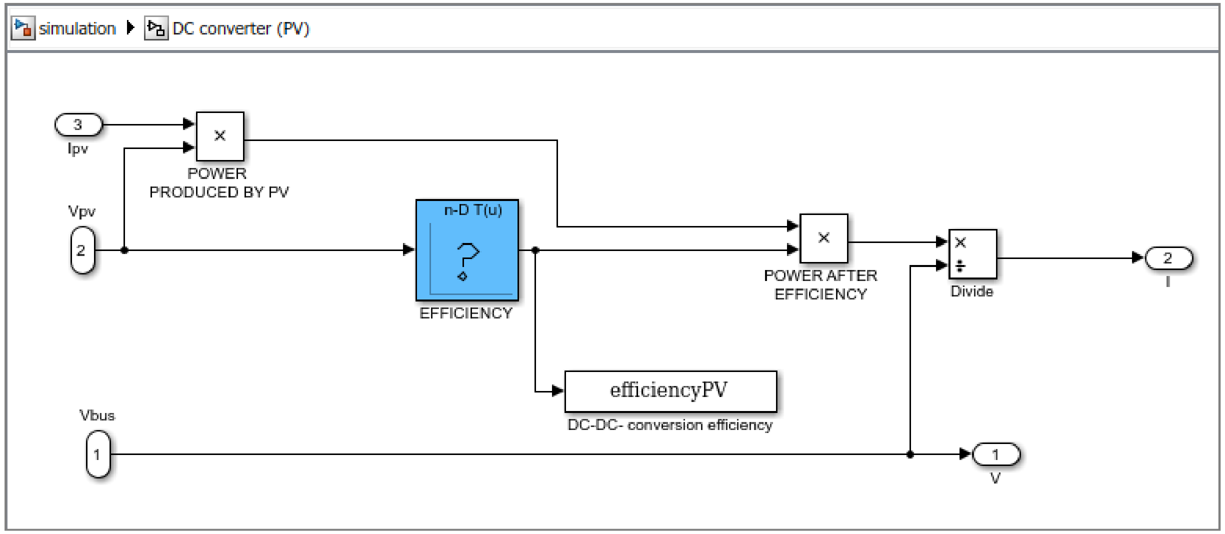
Lab 3 – Day 2
Model of DC-DC converters and battery

Simulink implementation



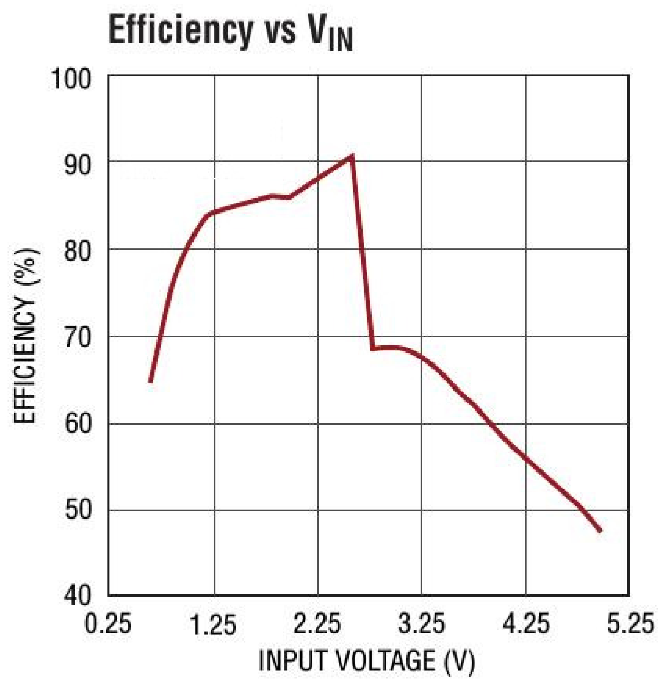
DC-DC converter of PV module

- DC-DC converter of the photovoltaic module
  - Contains a lookup table of efficiency given input PV voltage
    - Efficiency in [0, 1]
  - Block implementation:
    - $(V_{PV} \cdot I_{PV}) \cdot \text{efficiency} = (V_{OUT} \cdot I_{OUT})$
    - $V_{OUT} = 3.3V$ (DC bus voltage)
    - $I_{OUT} = (V_{PV} \cdot I_{PV} \cdot \text{efficiency})/V_{OUT}$



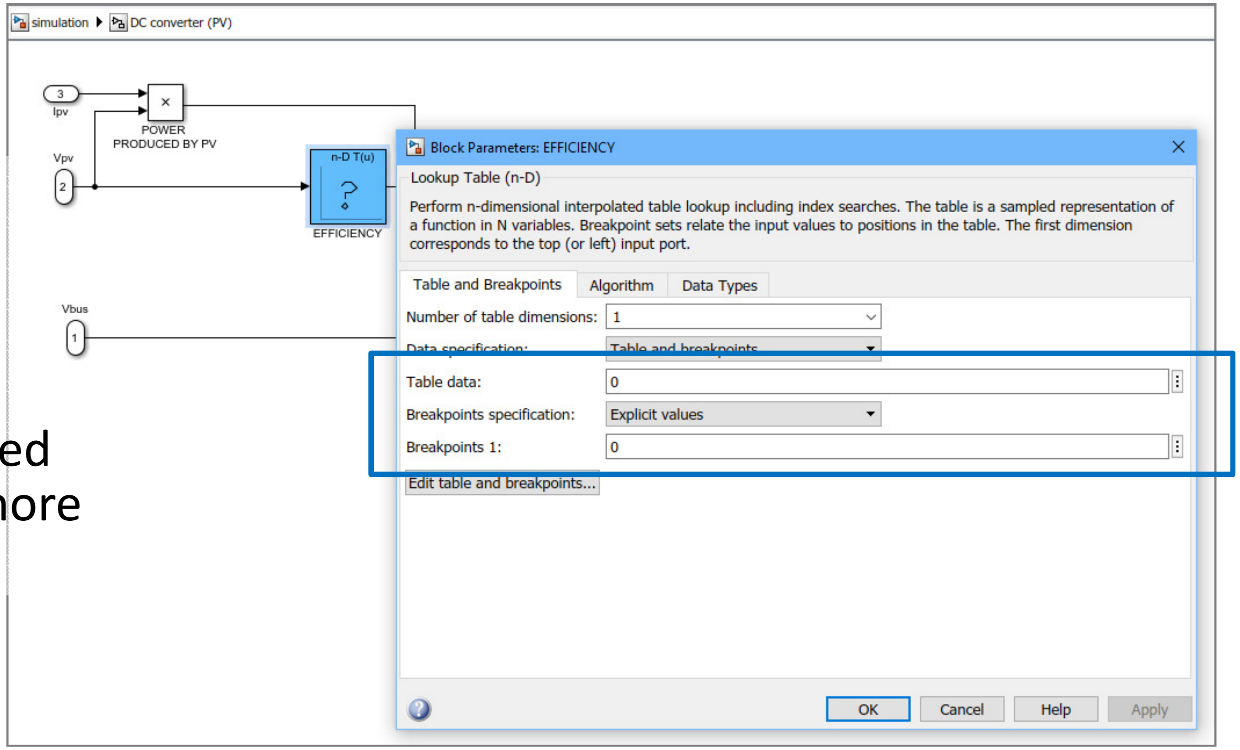
DC-DC converter of PV module

- Must populate the lookup table of efficiency given input voltage of the photovoltaic module
  - Open the **PV_DCDCconverter.pdf** file
    - Page 5: curve of efficiency w.r.t. input voltage
  - Use the samples of the curve to populate the lookup table

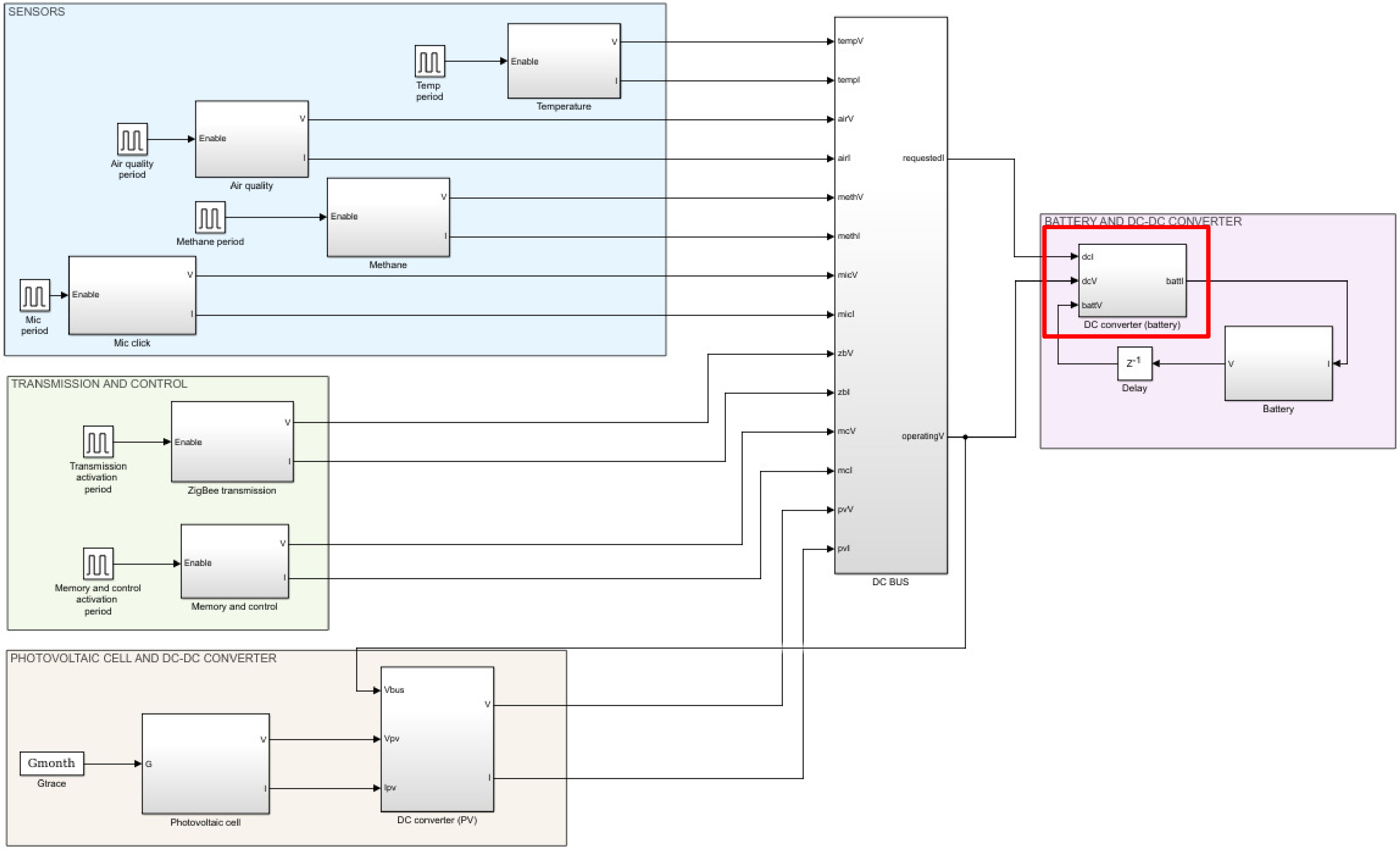


DC-DC converter of PV module

- How to:
  1. Digitize the curve
  2. Import file in Matlab
  3. Set the lookup table parameters
     - Voltage samples = breakpoints
     - Efficiency samples = table data
     - More digitized samples = more accurate

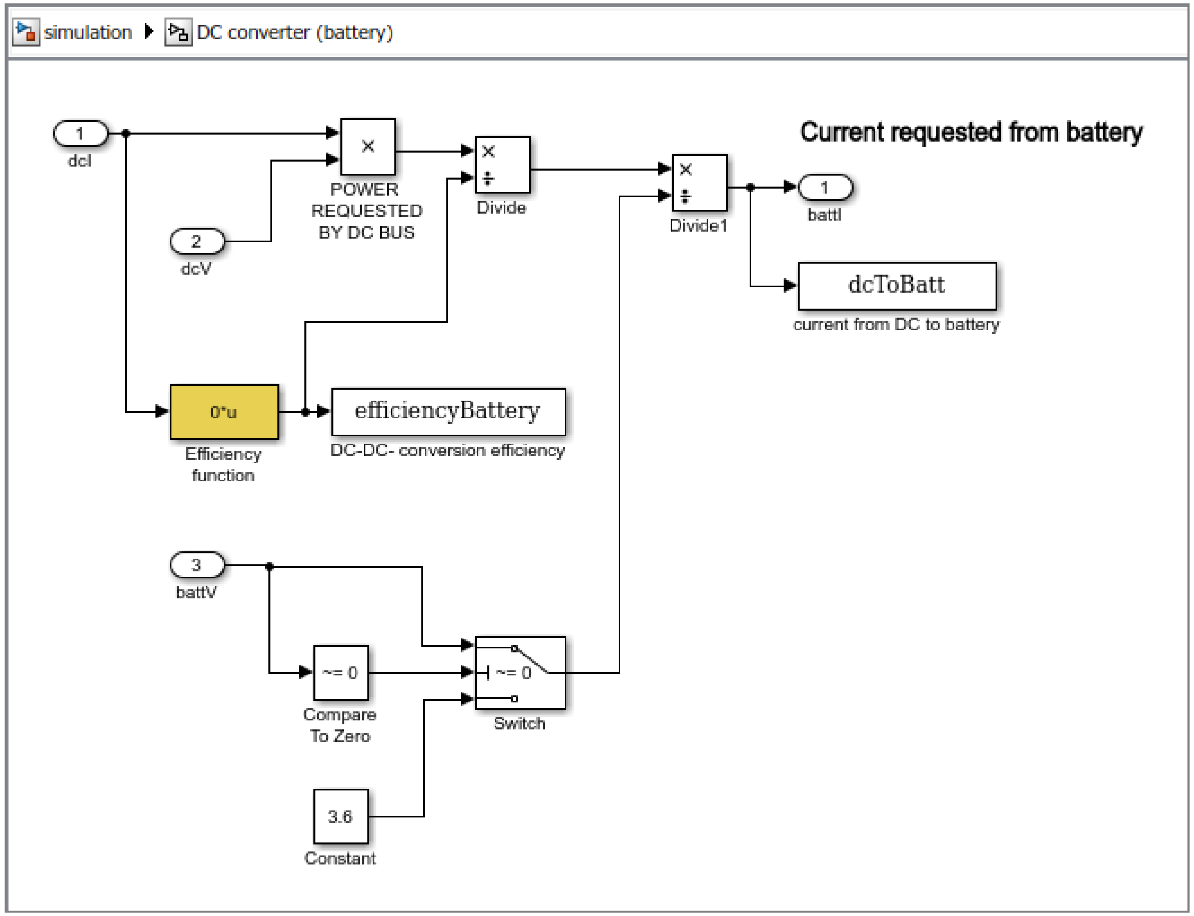


Simulink implementation



DC-DC converter of battery

- DC-DC converter of the battery

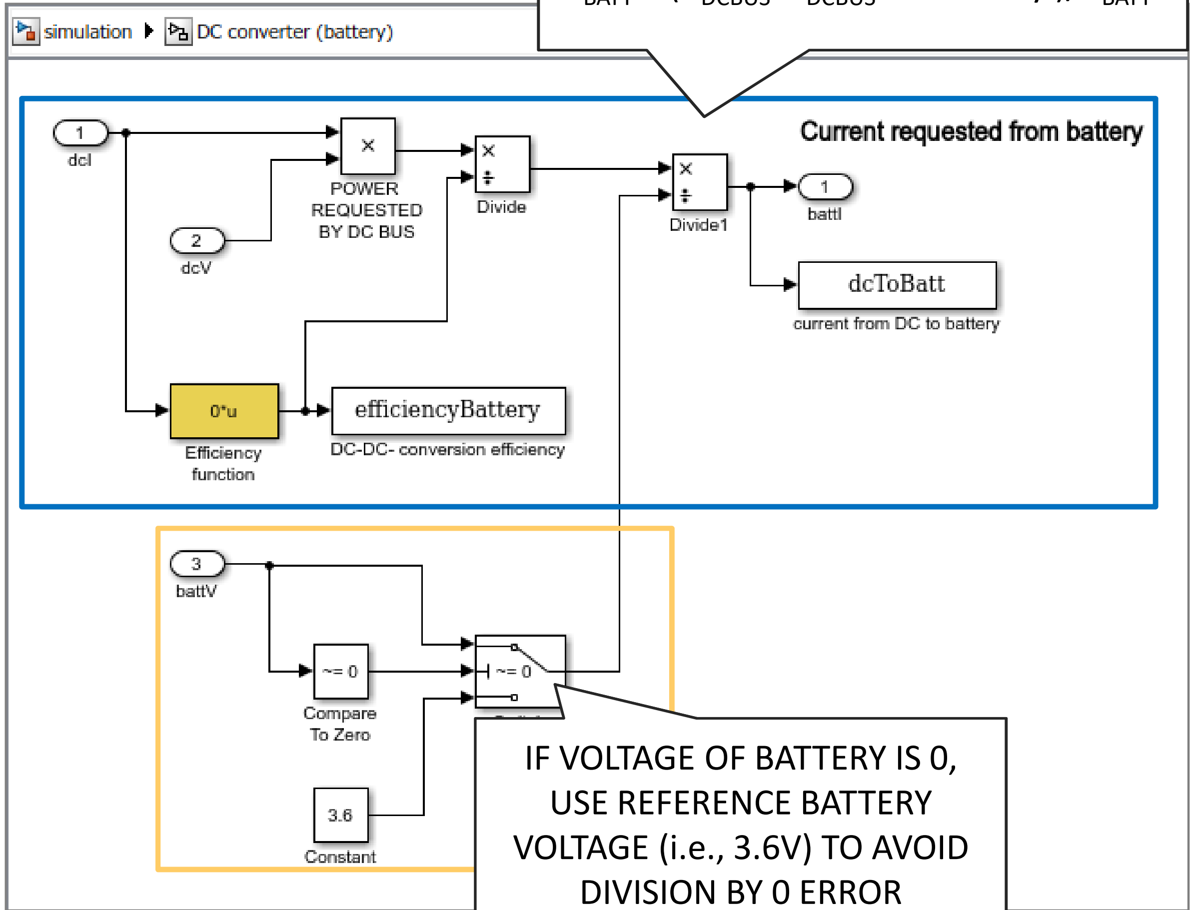


DC-DC converter of battery

- DC-DC converter of the battery

$(V_{DCBUS} \cdot I_{DCBUS}) \cdot \text{efficiency} = (V_{BATT} \cdot I_{BATT})$

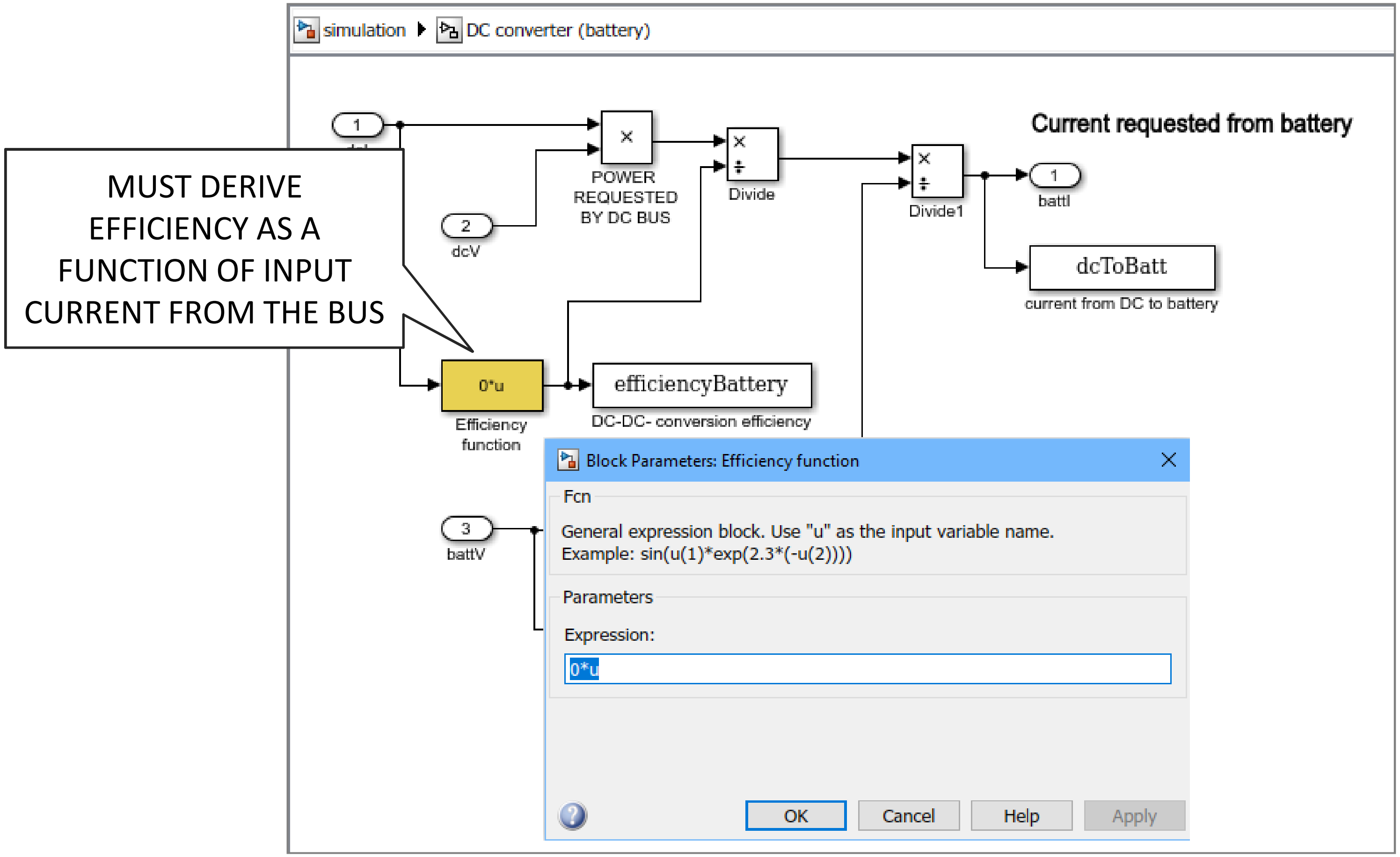
$I_{BATT} = (V_{DCBUS} \cdot I_{DCBUS} \cdot \text{efficiency})/V_{BATT}$



IF VOLTAGE OF BATTERY IS 0, USE REFERENCE BATTERY VOLTAGE (i.e., 3.6V) TO AVOID DIVISION BY 0 ERROR

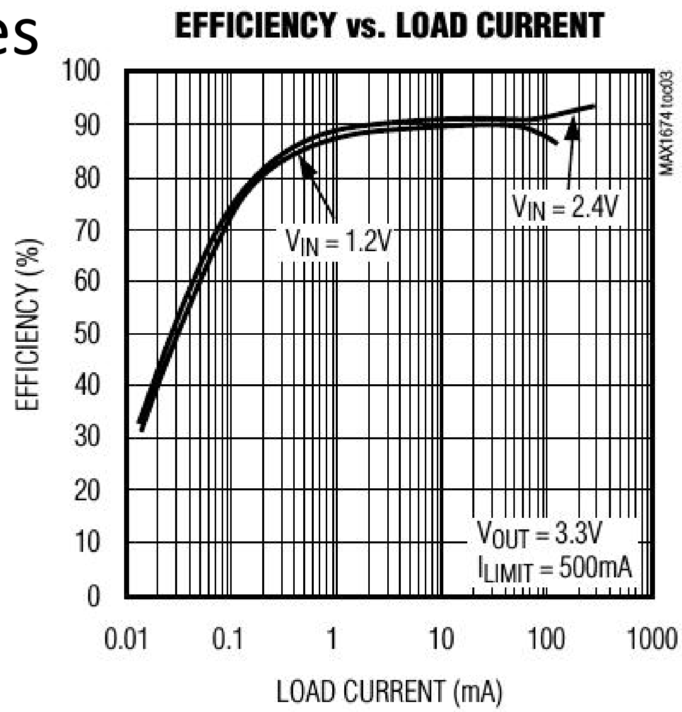
DC-DC converter of battery

- DC-DC converter of the battery



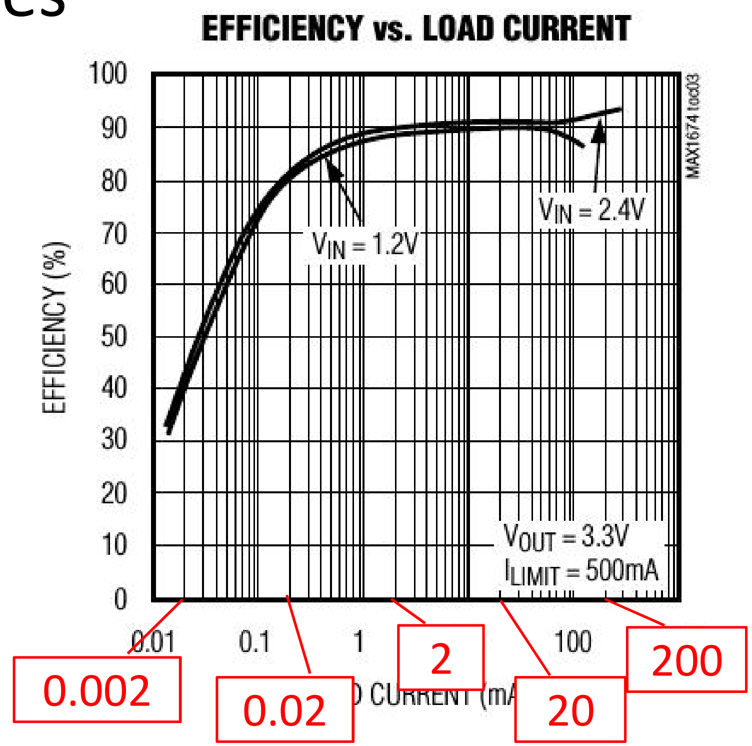
DC-DC converter of battery

- Open the **BATT_DCDCconv.pdf** file
  - Page 4: curve of efficiency given input current with voltage to the bus = 3.3V
- How to:
  1. Digitize one of the two curves
     - We avoid dependency from battery voltage
  2. Import the data in Matlab

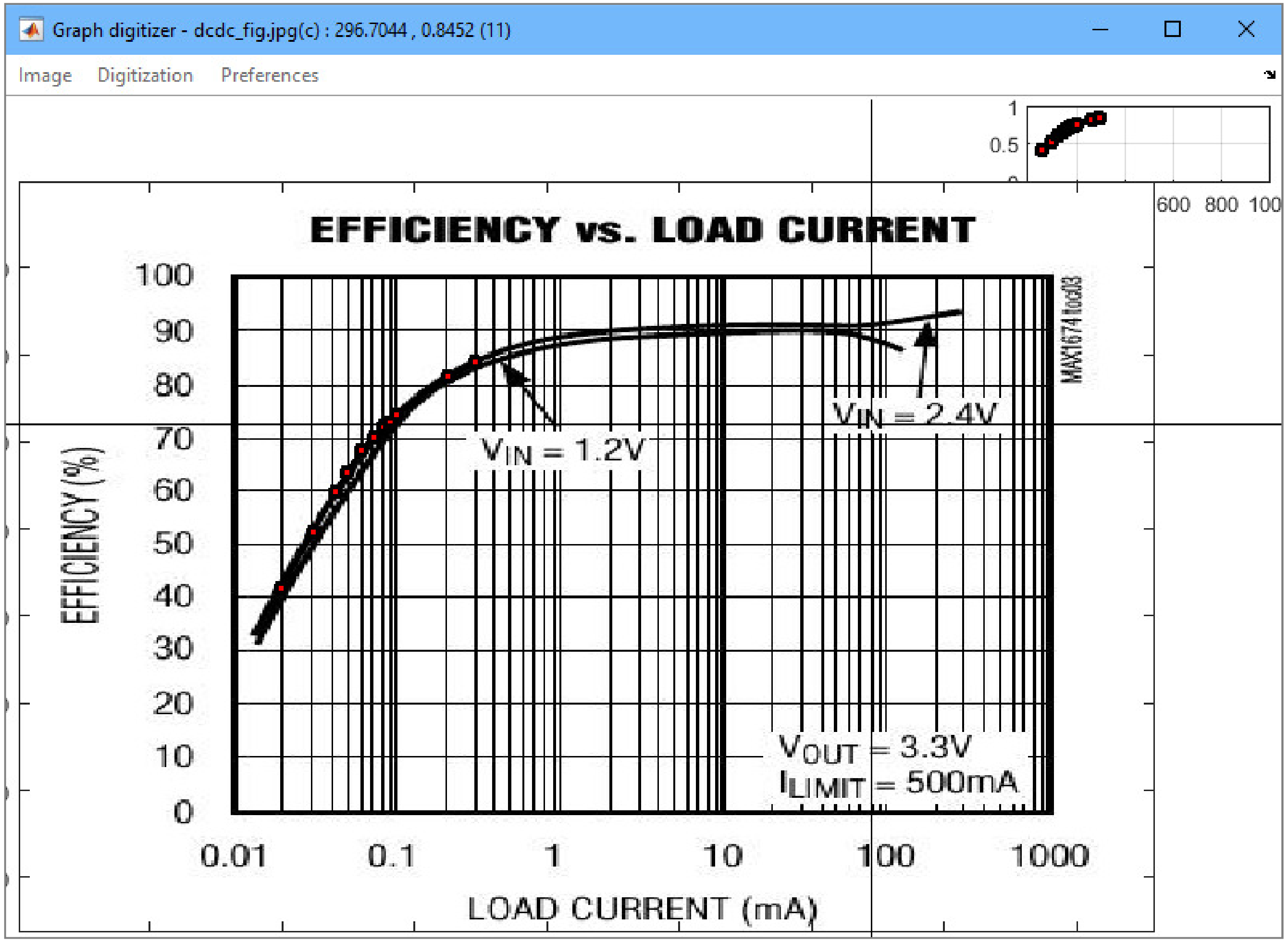


DC-DC converter of battery

- Open the **BATT_DCDCconv.pdf** file
  - Page 4: curve of efficiency given input current with voltage to the bus = 3.3V
- How to:
  1. Digitize one of the two curves
     - We avoid dependency from battery voltage
     - **ISSUE: x axis is in logarithmic scale**
       - **Sample points corresponding to the vertical lines**
       - **Can replace the x coordinate of samples with known values**
  2. Import the data in Matlab

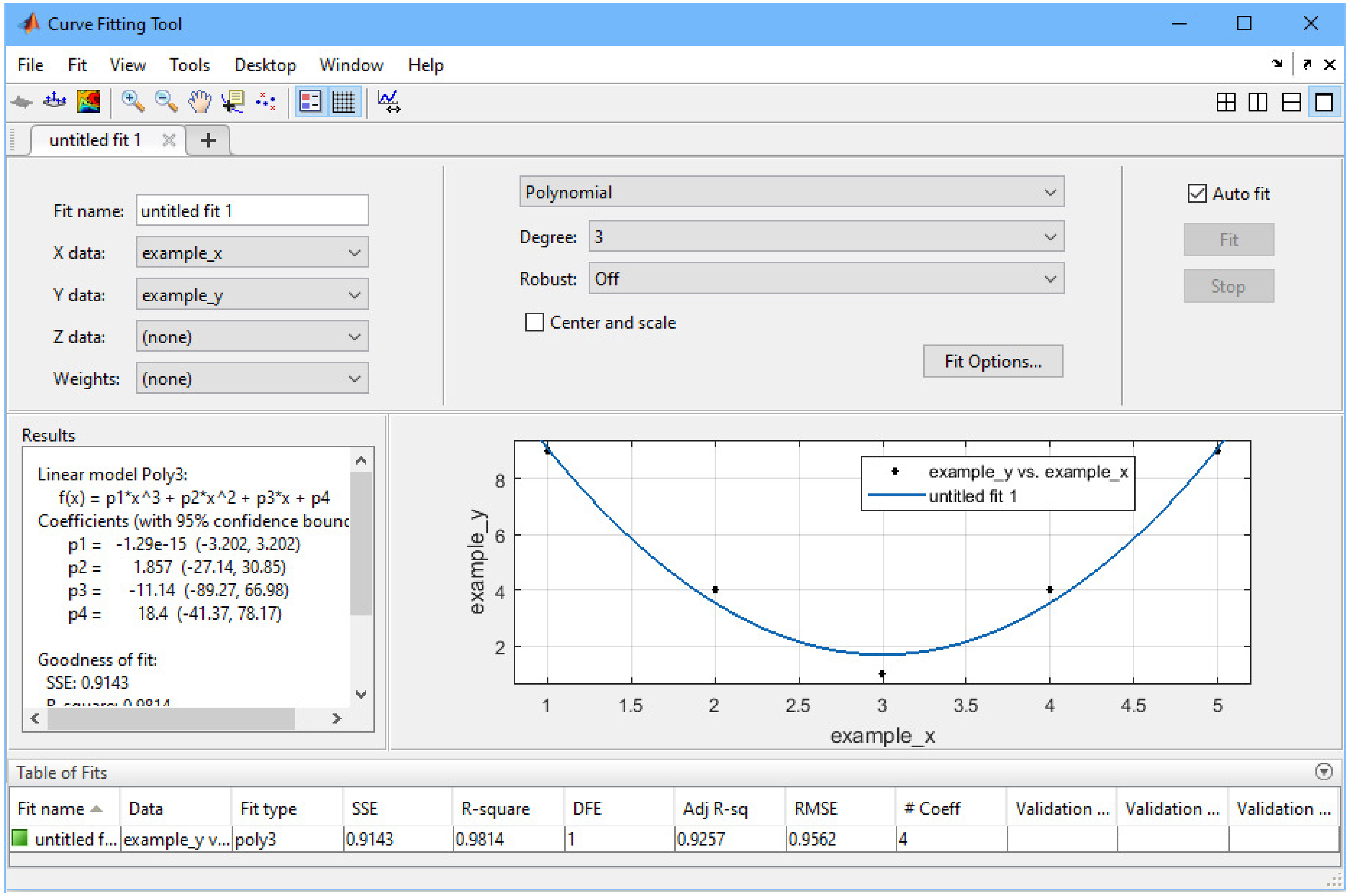


DC-DC converter of battery



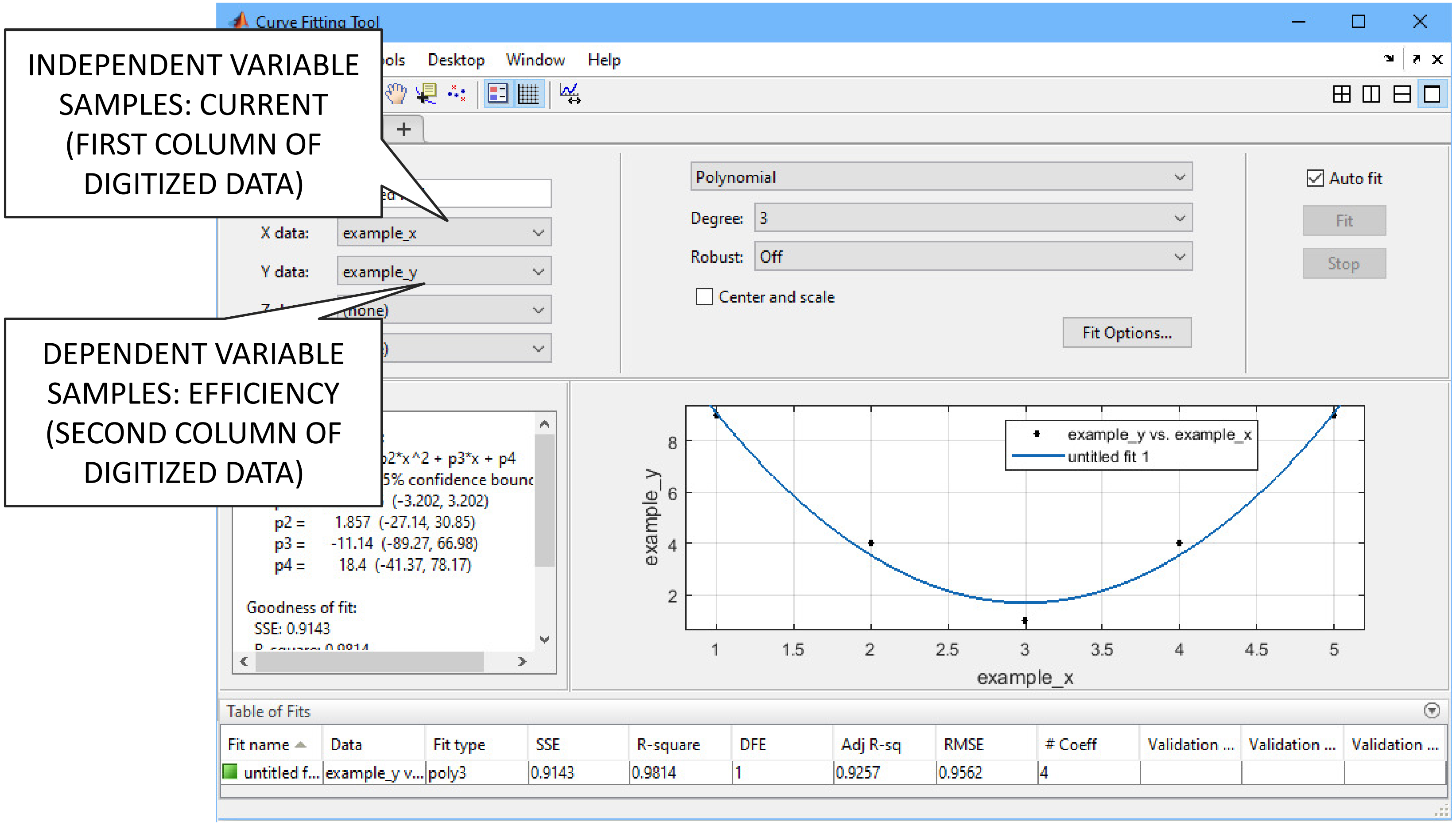
DC-DC converter of battery

3. Fit the samples to a polynomial with CurveFit tool
   – Apps > Curve Fitting



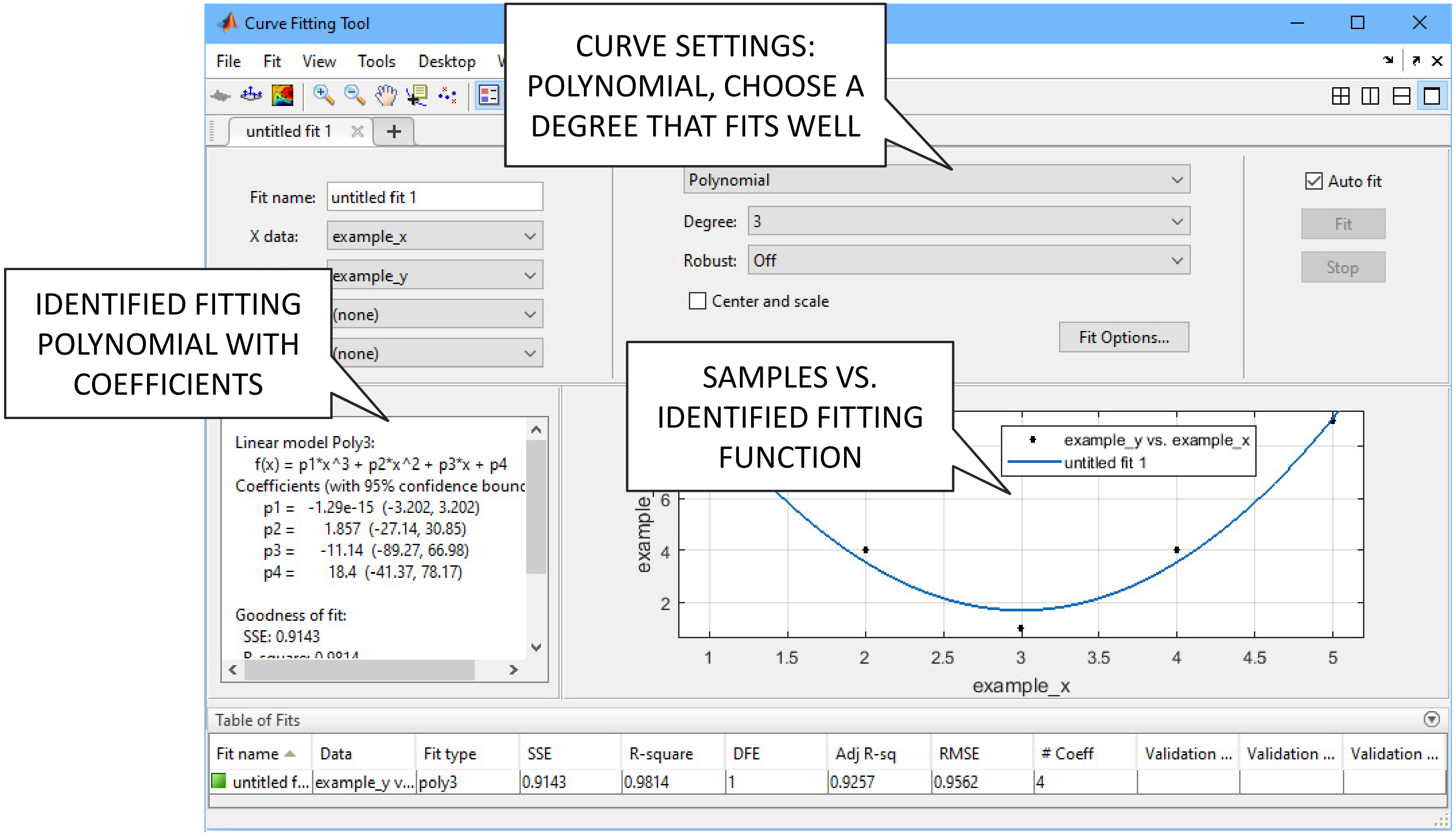
DC-DC converter of battery

3. Fit the samples to a polynomial with CurveFit tool
   – Apps > Curve Fitting



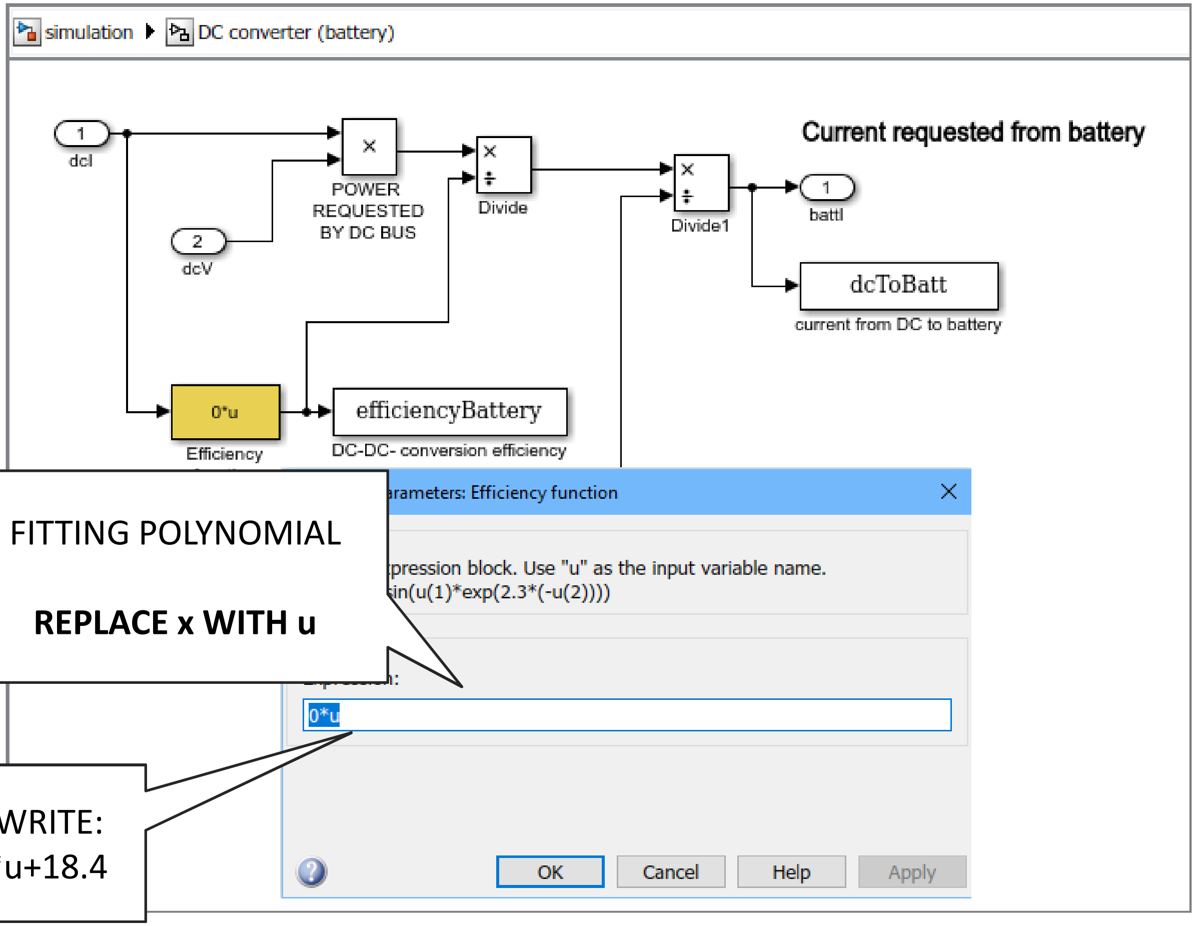
DC-DC converter of battery

3. Fit the samples to a polynomial with CurveFit tool
   – Apps > Curve Fitting (link)



DC-DC converter of battery

4. Write the polynomial in the efficiency function block

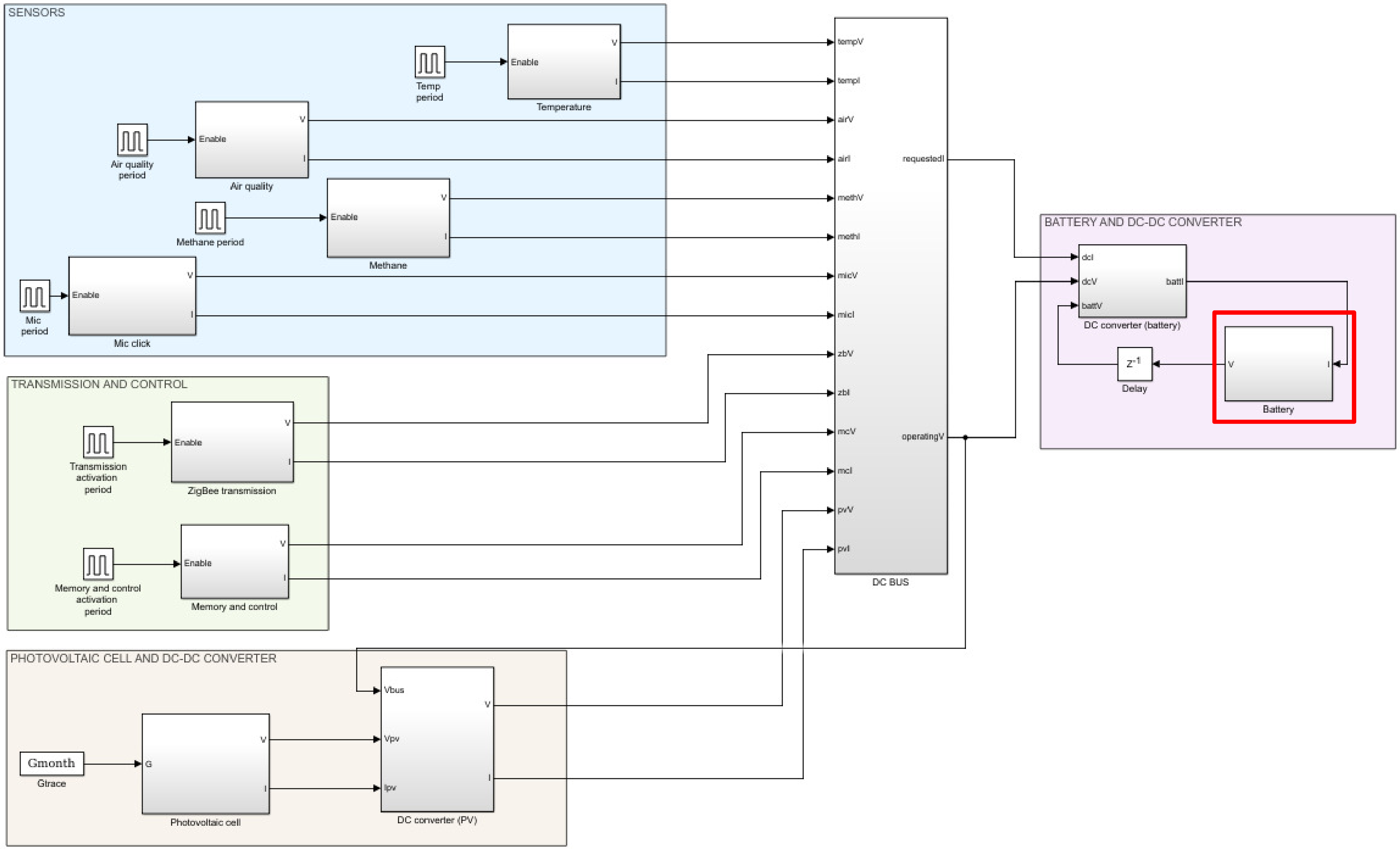


FITTING POLYNOMIAL

REPLACE x WITH u

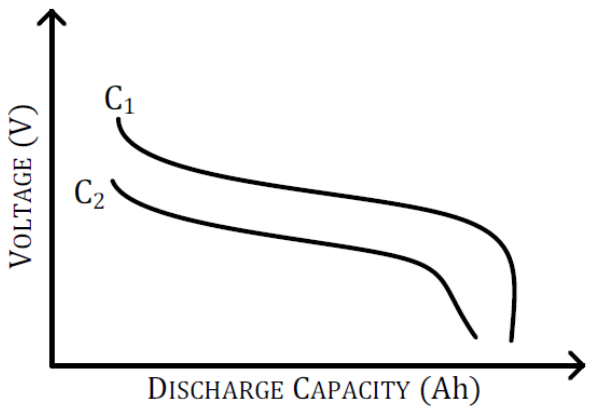
FOR FORMER EXAMPLE, I WOULD WRITE:
-1.29e-15*u^3 + 1.857*u^2-11.14*u+18.4

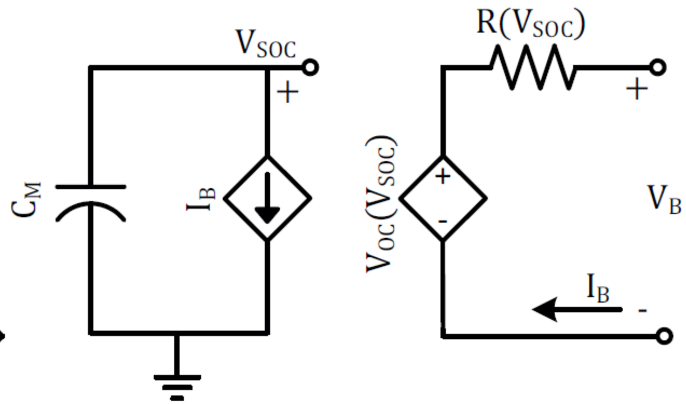
Simulink implementation



Battery model

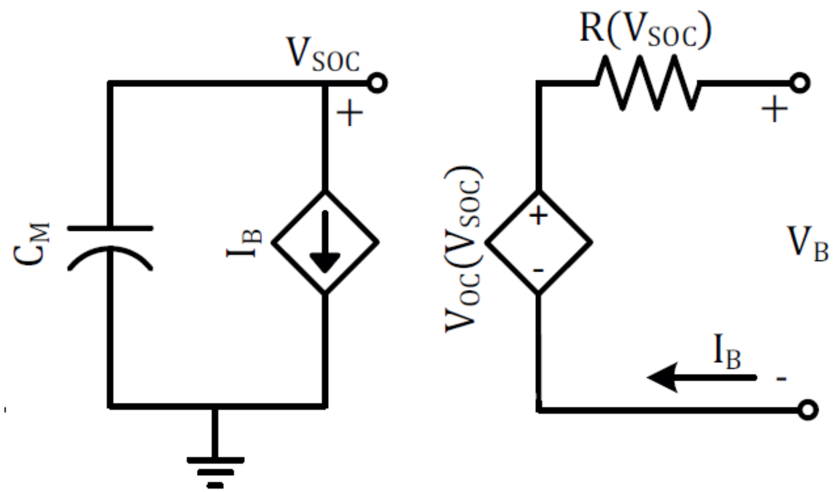
- Model: a circuit model
  - Configuration and settings extracted solely from data provided in the datasheet
  - Model made of two branches:
    - Left: models battery lifetime and the SOC
    - Right: models battery dynamics





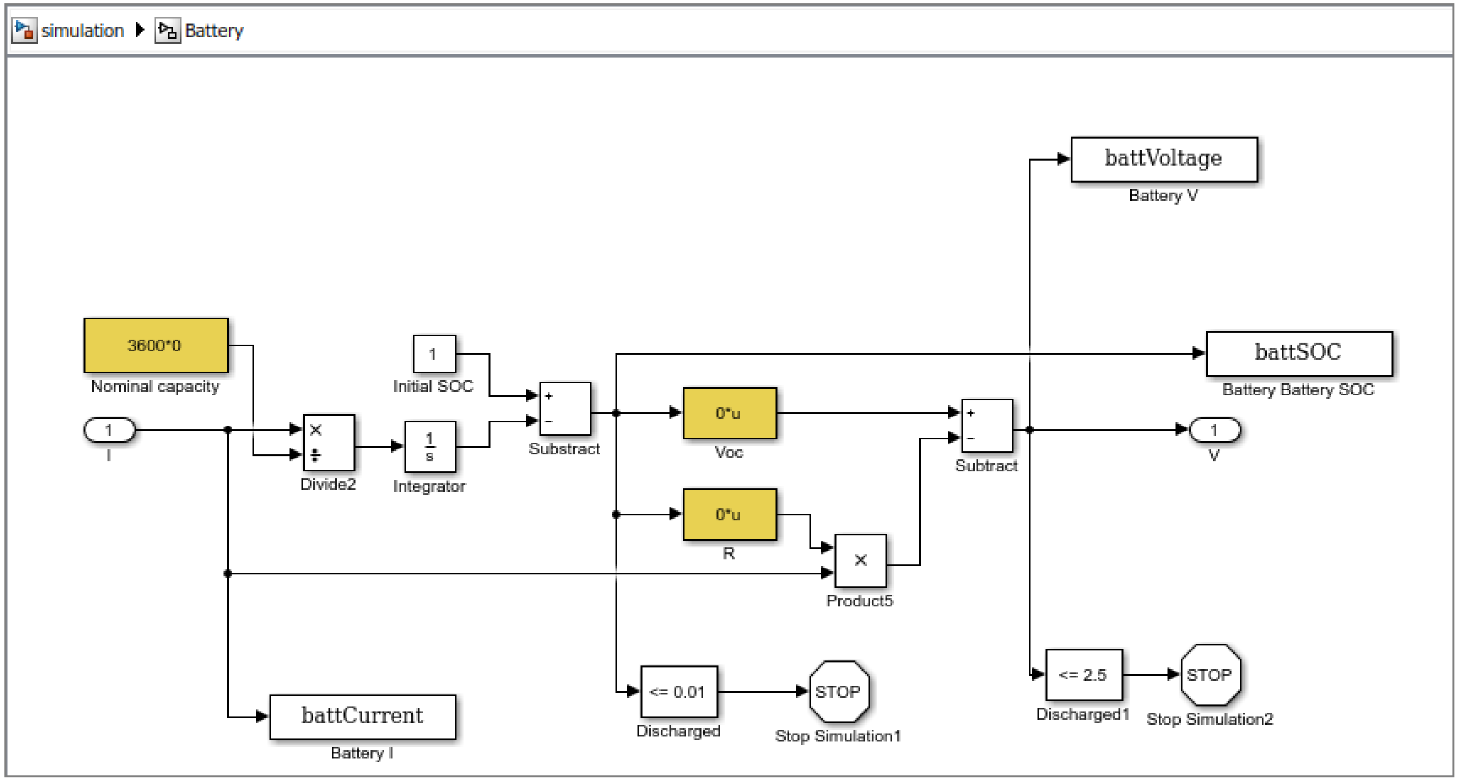
Battery model

- Left:
  - C_M available usable capacity
  - I_B discharge current
  - V_{SOC} state of charge
- Right:
  - V_{OC} models the dependency between battery voltage vs. the SOC
  - I_B discharge current
  - R series resistance modeling voltage drop due to ohmic losses (i.e., the internal resistance of the battery)
  - V_B is battery voltage



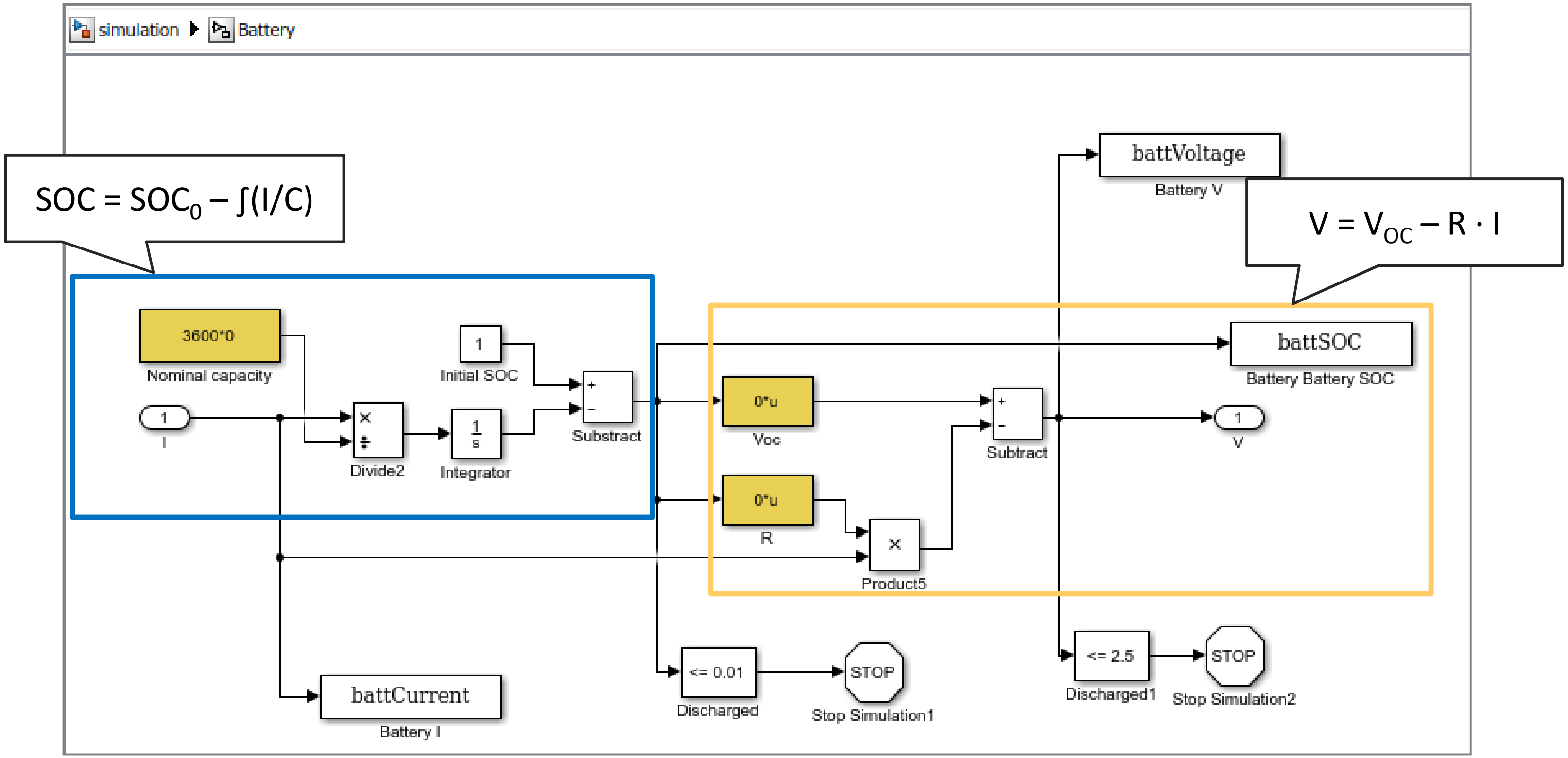
Battery model

- Not implemented as a circuit but as equations solving the circuit
  - Simulation faster than implementing the circuit



Battery model

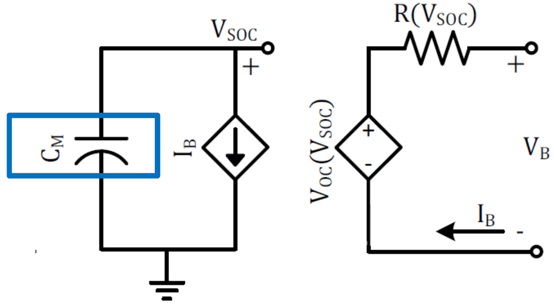
- Not implemented as a circuit but as equations solving the circuit
  - Simulation faster than implementing the circuit



Battery model

- We must reconstruct the values for the single elements of the circuit from the datasheet

- Capacity
  - 3200mAh (3.2 Ah)
  - Timestep is 1s
    - Multiply per 3600 (seconds in 1 hour)

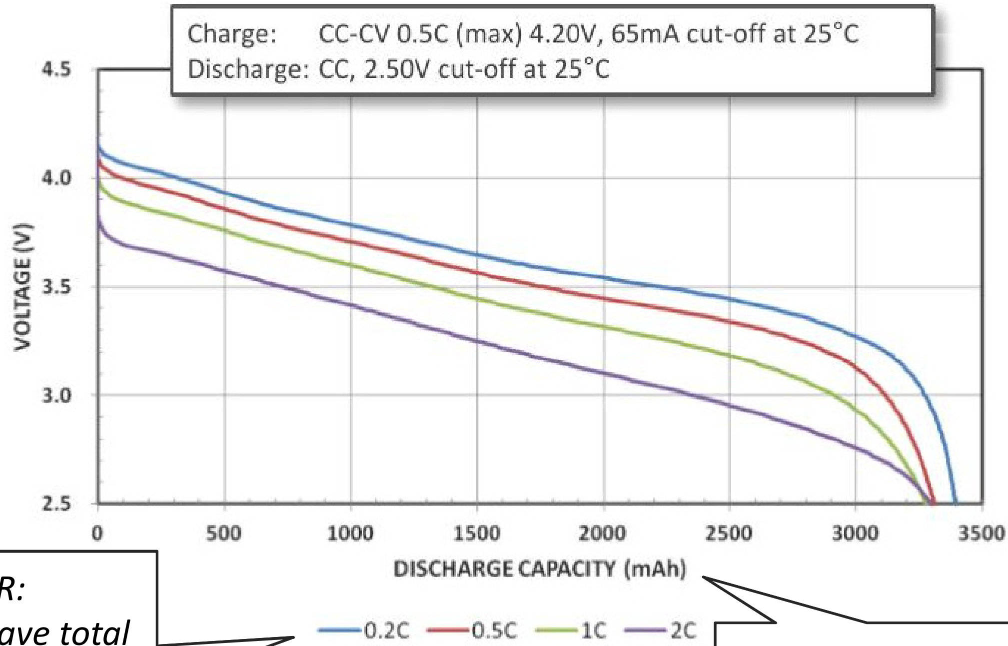


Specifications

Rated capacity[1]	Min. 3200mAh
Capacity[2]	Min. 3250mAh Typ. 3350mAh
Nominal voltage	3.6V
Charging	CC-CV, Std. 1625mA, 4.20V, 4.0 hrs
Weight (max.)	48.5 g
Temperature	Charge*: 0 to +45°C Discharge: -20 to +60°C Storage: -20 to +50°C
Energy density[3]	Volumetric: 676 Wh/l Gravimetric: 243 Wh/kg

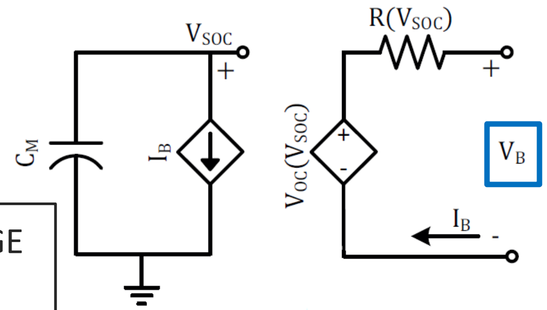
Battery model

- V_B battery voltage
  - Function of the load current and of the SOC



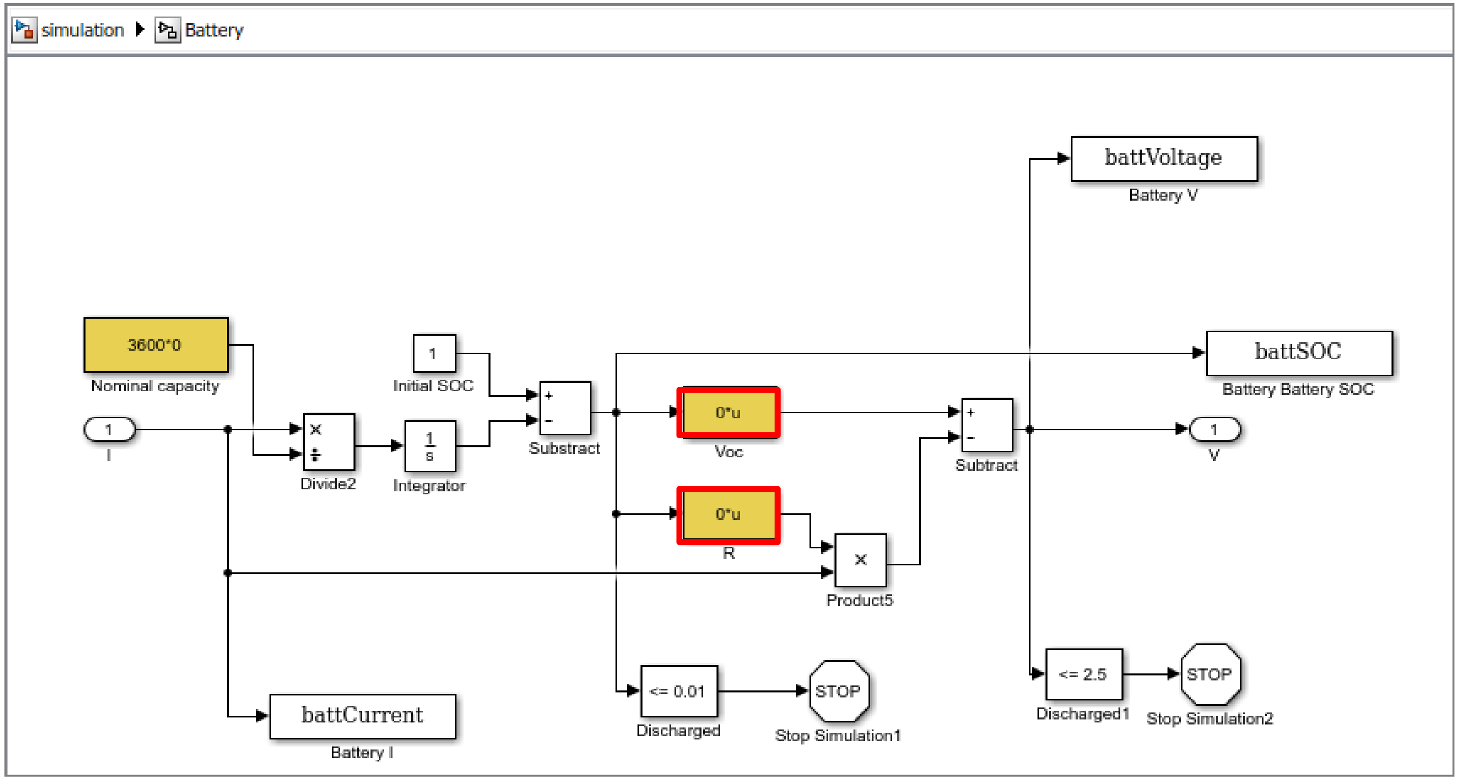
REMINDER:
1C = current to have total discharge in 1 hour

≈ STATE OF CHARGE OF THE BATTERY



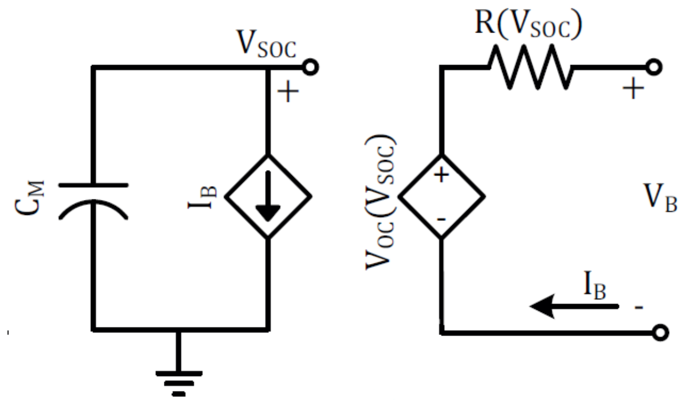
Simulink implementation

- Simulink models the battery as V_{OC} and R as a function of SOC
  - How can we relate V_{OC} and R to V, and thus SOC?



Battery model

- Derive V_{OC} and R as a function of V
  - Solving the equations associated with the right hand side branch of the circuit
  - $V_{OC} = V_{curve1} + R \cdot I_{curve1}$
  - $R = \frac{V_{curve2} - V_{curve1}}{I_{curve1} - I_{curve2}}$



  - Notation:
    - $curve1$ and $curve2$ refer to the operation with two different discharge currents, i.e., to two different discharge curves obtained with different discharge current values

Battery model

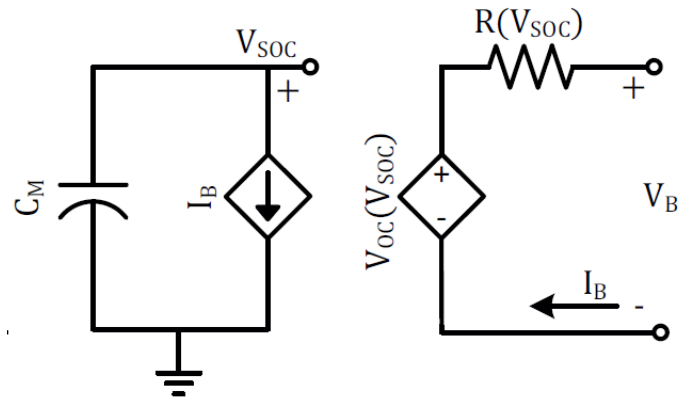
- In case you were wondering how we got here…

  - Equation solving the system:
    - $V_{OC} = R\,I + V$
  - Applied for both currents:
    - $\begin{cases} V_{OC} = RI_{curve1} + V_{curve1} \\ V_{OC} = RI_{curve2} + V_{curve2} \end{cases}$

  - $V_{OC} = RI_{curve1} + V_{curve1} = RI_{curve2} + V_{curve2}$
  - $R(I_{curve1} - I_{curve2}) = V_{curve2} - V_{curve1}$
  - $R = \frac{V_{curve2} - V_{curve1}}{I_{curve1} - I_{curve2}}$

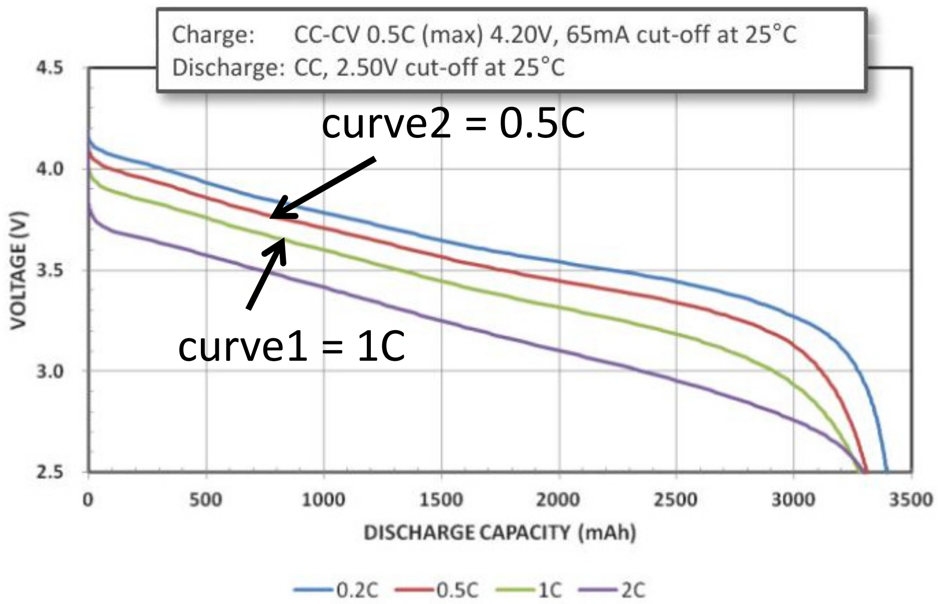
Battery model

- Derive V_{OC} and R as a function of V
  - Solving the equations associated with the right hand side branch of the circuit
  - $V_{OC} = V_{curve1} + R \cdot I_{curve1}$
  - $R = \frac{V_{curve2} - V_{curve1}}{I_{curve1} - I_{curve2}}$
  - How do we populate the equations?



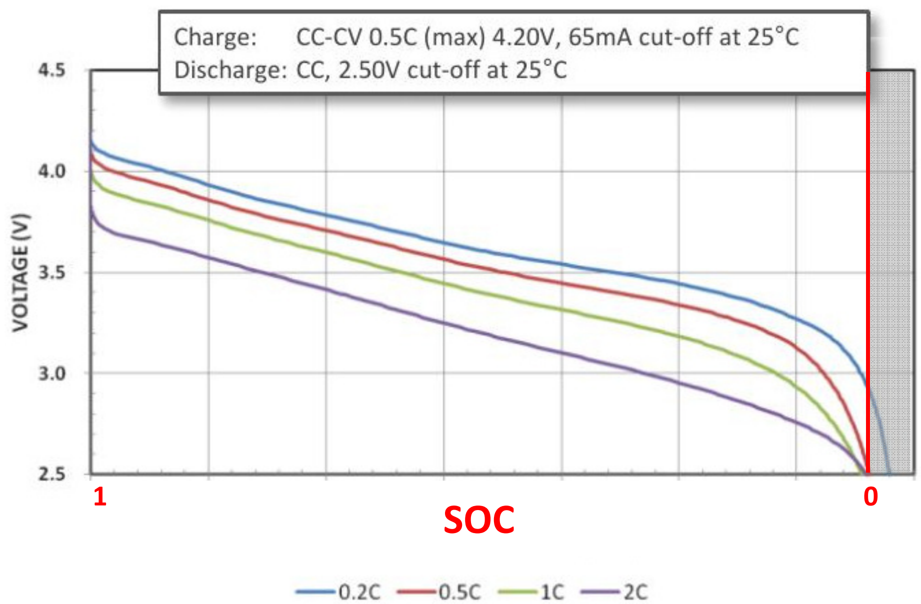
Battery model

- Derive V_{OC} and R as a function of V
  - $V_{OC} = V_{curve1} + R \cdot\ I_{curve1}$
  - $R = \frac{V_{curve2} - V_{curve1}}{I_{curve1} - I_{curve2}}$
  - Need two curves of V given the discharge capacity
    - See example →
    - Values of V given SOC and discharge current



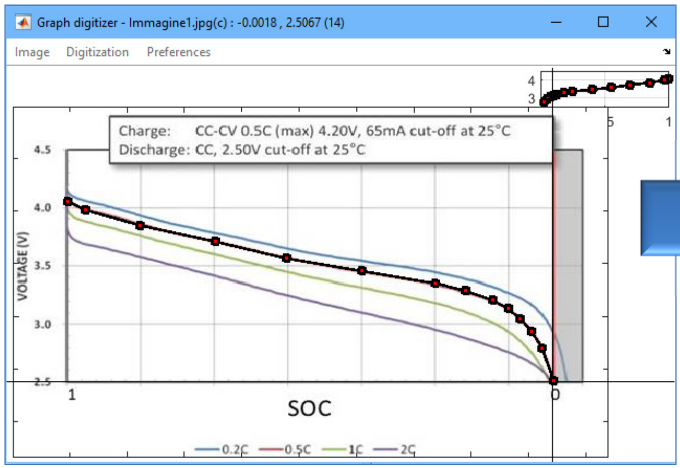
Battery model

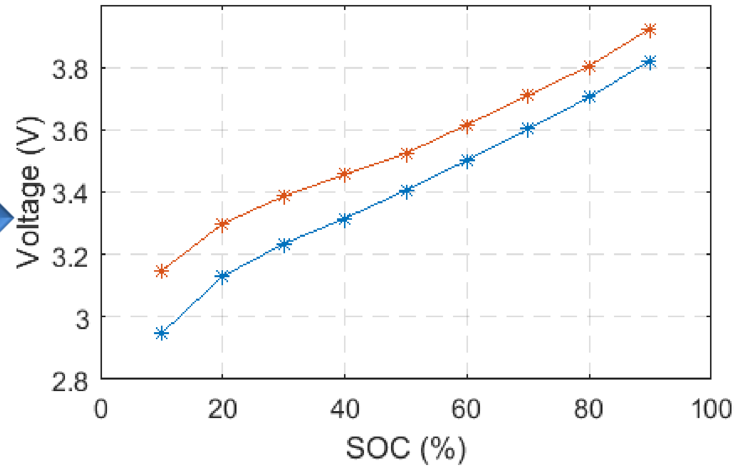
- V battery voltage: how to
  1. Download the datasheet figure (course website)
  2. Determine SOC axis
     - Fix the origin where the lower curve ends



Battery model

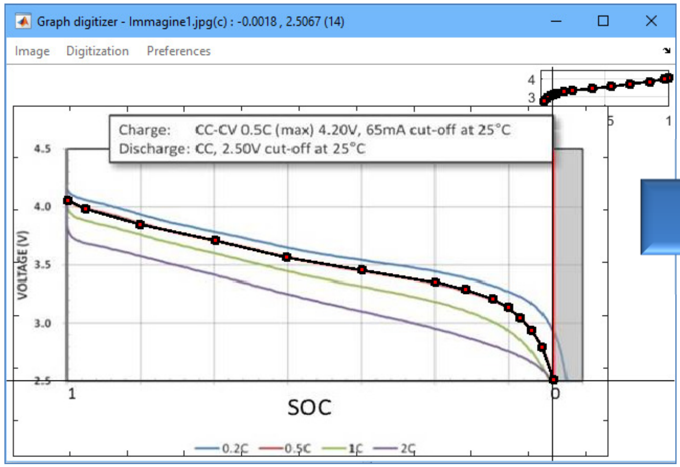
- V battery voltage: how to
  3. Use one of the image digitizer tools to extract data from the figure
     - Extract values of two of the voltage vs. SOC curves
  4. Interpolate the two data sets
     - Given the same value of SOC, we have the corresponding value of V for both the curves

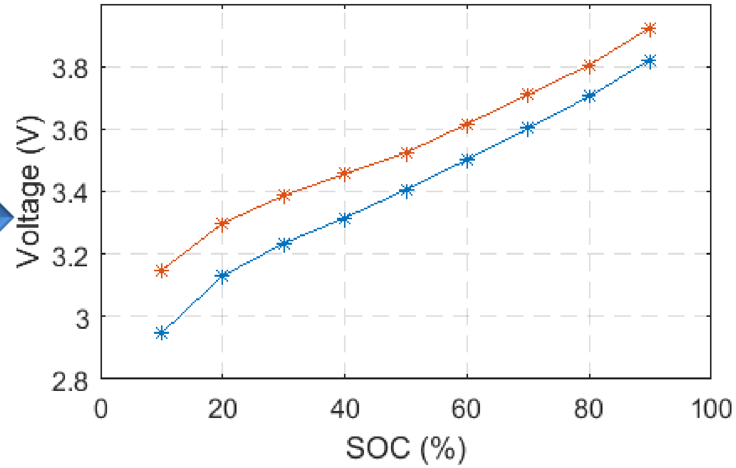




Battery model

- Interpolation: Matlab function `interp1` (link)
  - `newY = interp1(x,y,newX)`
    - Given the known samples: x values and corresponding y values
    - And the new x values newX
    - Calculate the corresponding y values newY
  - In our case:
    - x and y are the coordinates of the digitized samples
    - newX are samples of SOC (e.g., from 0 to 1 with step 0.1)

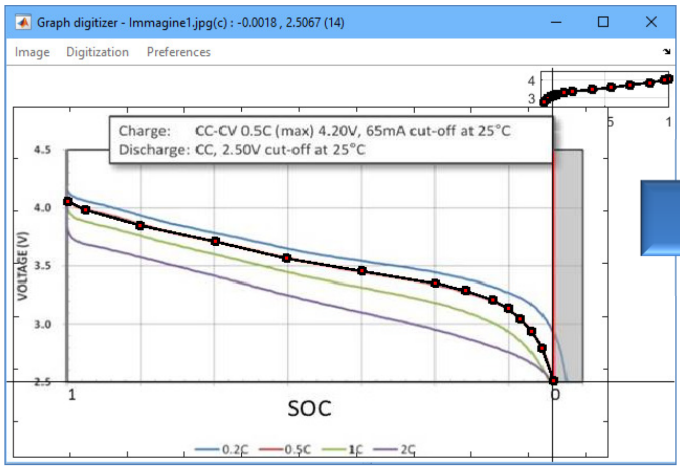


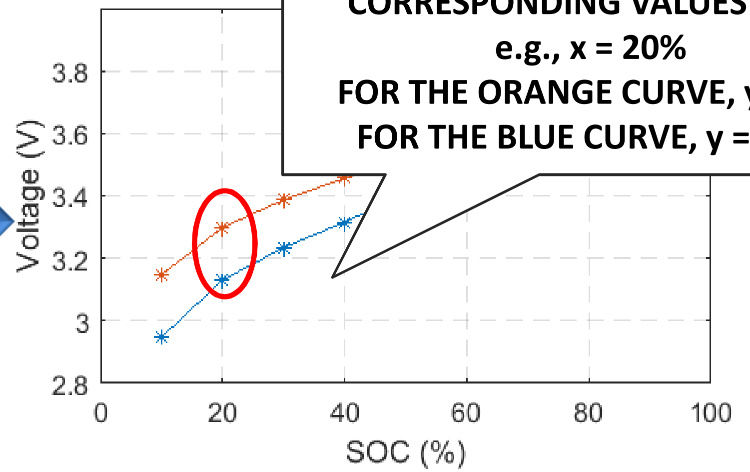


Battery model

- Why do we need interpolation?
  - The samples of the two curves will most probably correspond to different values of SOC (i.e., x)
  - Thus we can not compare the two curves!
    - We have y values corresponding to different x coordinates!

AFTER INTERPOLATION, GIVEN THE SAME x VALUE WE HAVE THE CORRESPONDING VALUES OF y:
e.g., x = 20%
FOR THE ORANGE CURVE, y = 3.3
FOR THE BLUE CURVE, y = 3.12





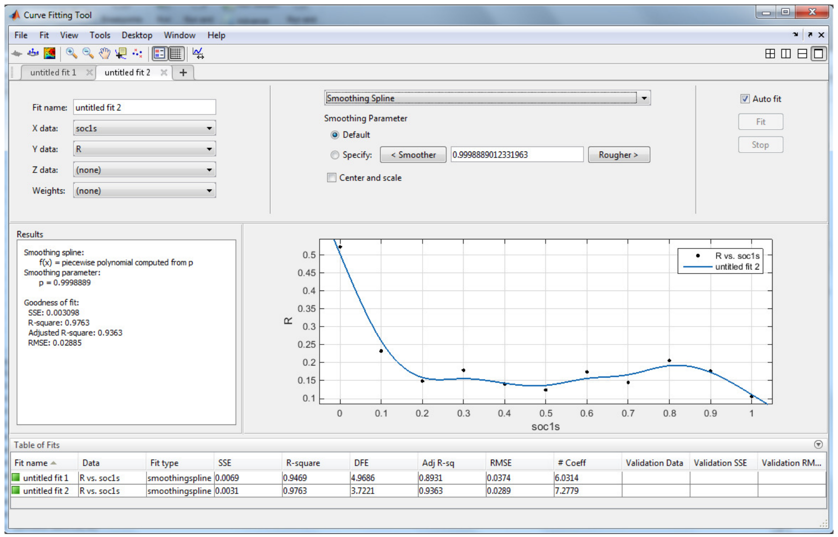
Battery model

- Derive V_{OC} and R
  - Solving the equations associated with the right hand side branch of the circuit
  - $V_{OC} = V_{curve1} + R \cdot I_{curve1}$
  - $R = \frac{V_{curve2} - V_{curve1}}{I_{curve1} - I_{curve2}}$

- Obtain values for the parameters of our battery model in *some points* of the SOC range
  - $\underbrace{SOC \rightarrow V}_{\text{WE KNOW FROM THE SAMPLES}} \underbrace{\rightarrow V_{OC}\,, R}_{\text{WE DERIVE FROM THE EQUATIONS}}$

Battery model

- So we fit the resulting data into functions by using the CurveFit app (link)
  - Express V_{OC} and R as functions of V_{SOC}, i.e., of the state of charge
  - Allow to derive V_{OC} and R for any value of SOC as some function/polynomial



Battery model

- This gives us all information to populate the battery model!
  - We derived the values or the equations modeling the various elements of the circuit

Simulink implementation

PUT IN THE FUNCTION BLOCKS THE FUNCTION/POLYNOMIAL

REMEMBER THAT THE VARIABLE REPRESENTING INPUT IS u (NOT x)