

Ontwikkeling en implementatie van een BMS op een bestaande lithium-ijzerfosfaat batterij bank

Masterproef 2016 – 2017
Karel Van Peteghem

Inhoudstabel

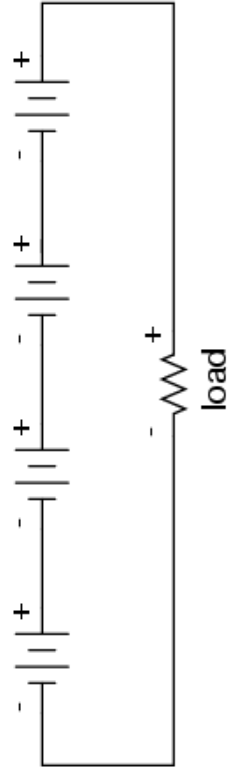
1. Opdrachtomschrijving
2. Aanpak semester 1
3. Elektrisch model
4. Programma
5. Resultaten metingen
6. Aanpak semester 2
7. Planning

Opdrachtsomschrijving

Ontwikkeling en implementatie van een BMS

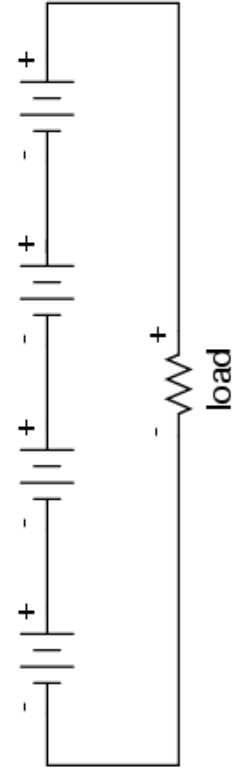
Onbalans

- Probleem voor cellen in serie
 - Cellen zijn door fabrikant opgedeeld maar zijn nooit identiek
 - Kleine verschillen in zelfontlading, weerstand en temperatuur karakteristieken kunnen leiden tot onbalans
- Op verschillende manieren ontstaat onbalans



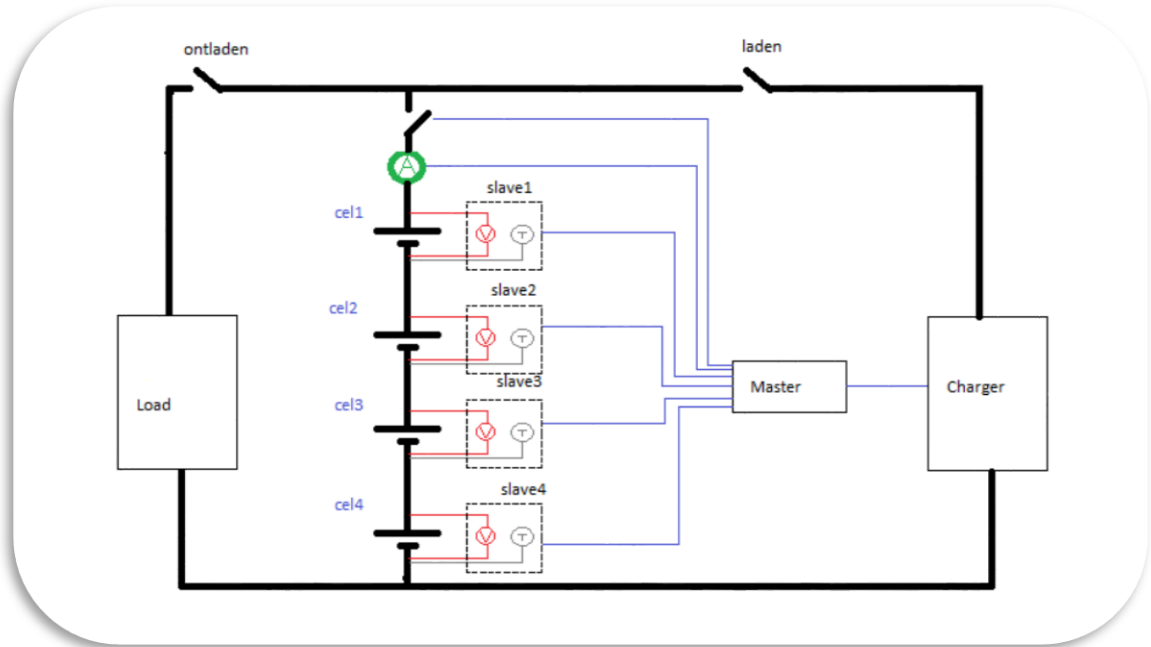
Onbalans

- Waarom is dit een probleem?
- Bepaalde cellen laden sneller op
- Andere cellen ontladen sneller
- De boven- en onderlimiet wordt sneller bereikt door deze onbalans
- Onbalans verlaagt hierdoor de performantie



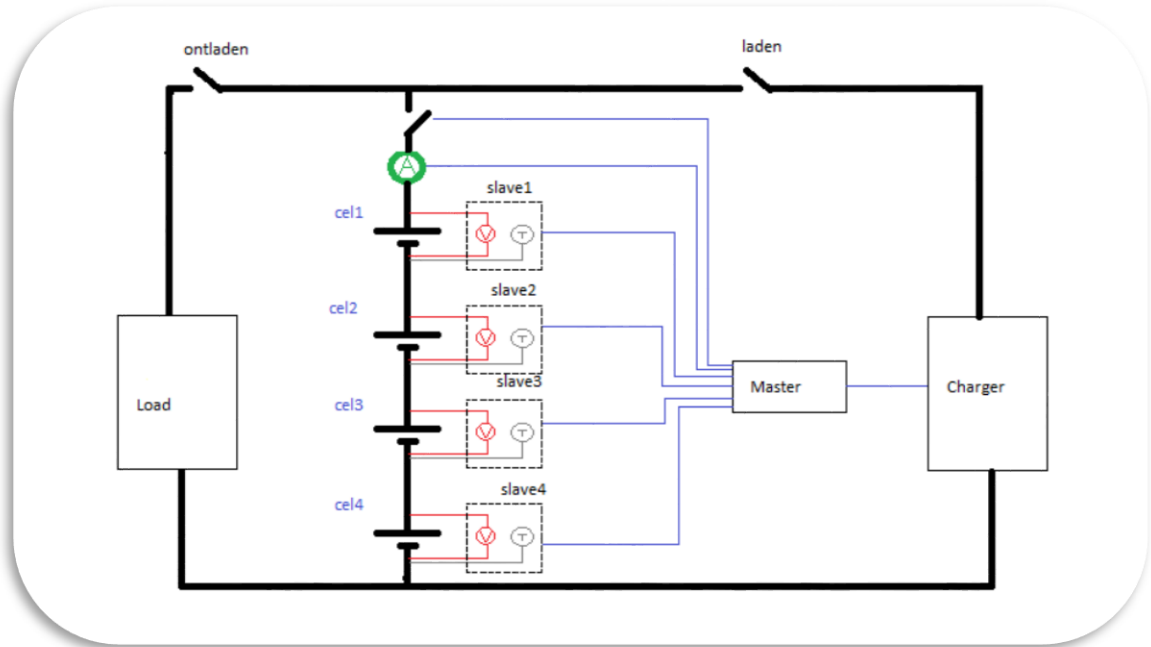
Het doel

- De performantie van het de batterijbank te verhogen en levensduur verhogen
- Hoe?
- De SOC van de cellen gelijkhouden door ze te ballanceren



Balanceren

- Het balanceren gebeurt aan de hand van de master die geprogrammeerd wordt met een BMS
- Actief of Passief

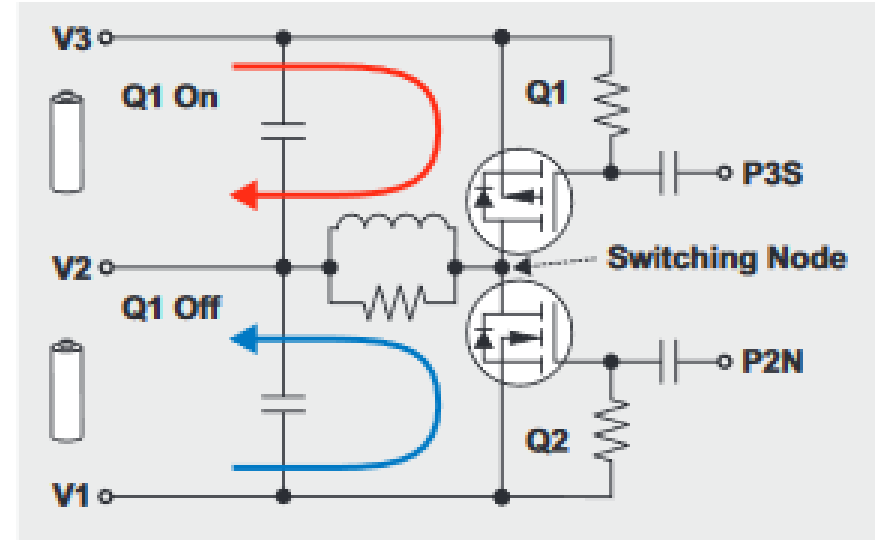


Passief balanceren

- Passieve balancerings is aanwezig
- Dit wordt meestal toegepast in de goedkopere BMS
- De cel krijgt een bypass weerstand die de cellen met een te hoge SOC ontladtd.
- Dit passief balanceren gebeurt het best wanneer gebaseerd op SOC.

Actief balanceren

- Vereist meer componenten en regeling
- Energie-efficiënter omdat energie wordt overgezet van cel tot cel



Aanpak semester 1

Aanpak semester 1

- Balanceren gebeurt aan de hand van SOC
- Modelleren van de cel
- Chemisch of elektrisch benaderen

Aanpak semester 1

- De meest gebruikelijke manieren om SOC te schatten:
 - Ah-tellen
 - OCV aanpak
 - Terminal voltage aanpak
 - Kalman filtering
 - Neurale netwerken
- Deze hebben voor en nadelen.

Aanpak semester 1

- Ah-counting is de basis van SOC schatten

$$S(t) = S(0) - \int_0^t \frac{\eta I(t)}{C_n} dt$$

- Indien de initiële SOC gekend is kan men dus de SOC op elk moment bepaald worden
- Probleem: houdt geen rekening met andere invloeden.

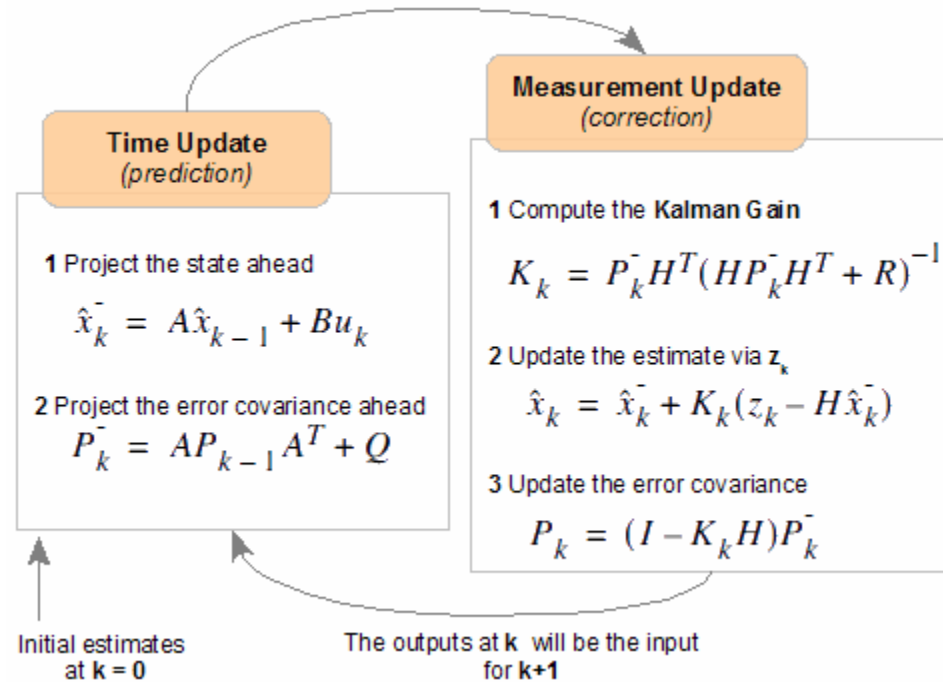
Aanpak semester 1

- Kalman filtering

-Wordt veel gebruikt bij BMS

-steunt op kansrekenen

-telkens wordt er een schatting gemaakt en deze wordt gecorrigeerd door een meting



Aanpak semester 1

- Hieruit ontstond de Kalman-Ah techniek
- Deze methode is populair bij SOC schatting

→ De “time update/prediction” zal gebeuren aan de hand van een elektrisch model

→ De “measurement update/correction” zal gebeuren aan de hand van de Ah-methode

Elektrisch model

Elektrisch model

- Een BMS is hoogstafhankelijk van een goed elektrisch model.
- Het model moet statisch en dynamisch gedrag goed kunnen inschatten.

Elektrisch model

- In de literatuur zijn verschillende modellen te vinden variërend van zeer eenvoudig tot zeer ingewikkeld
- Uit het doctoraatsonderzoek van prof. Noshin Omar is een elektrisch model voortgevloeid

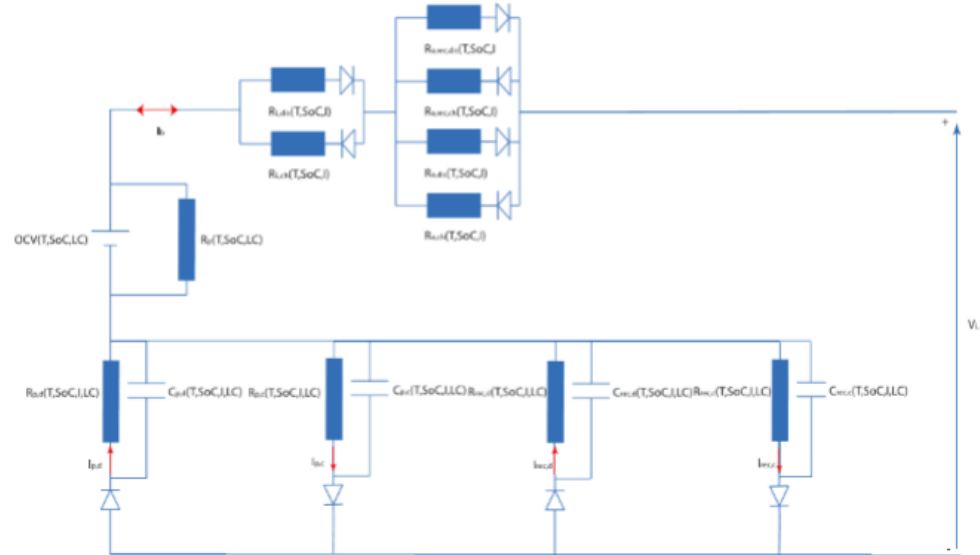
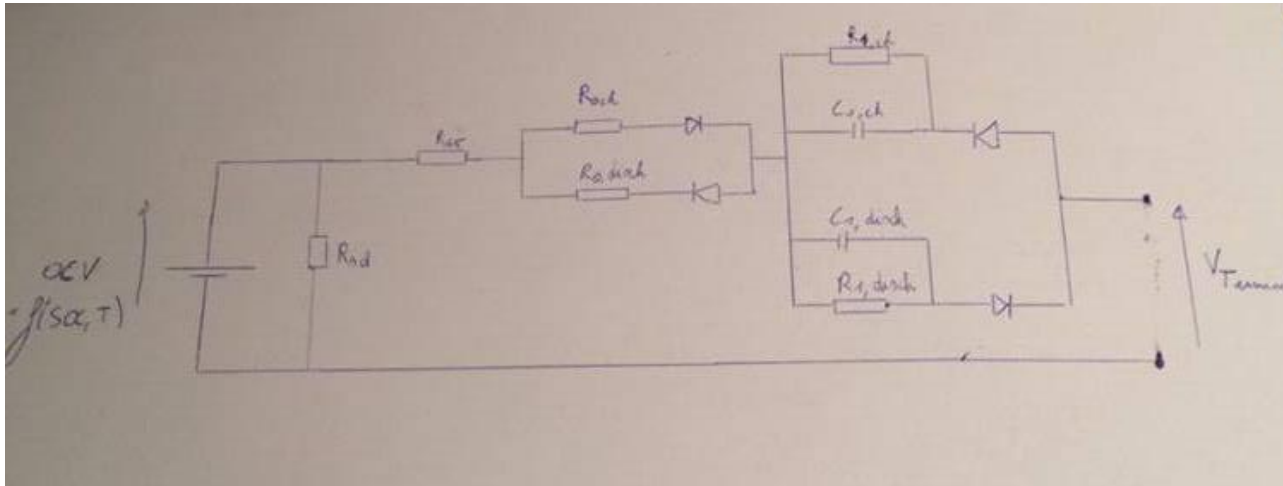


Figure 10. Extended Noshin's electrical battery model.

Elektrisch model



- R_{sd} = zelfontlading
- R_{0v} = veroudering
- $R_{0, ch}$ or $disch$ = uitgangsweerstand bij op- of ontladen
- $R_1, C_1 \rightarrow$ recovery effect en hysteresis

Elektrisch model

- Invloeden op het model?
 - Spanning, Temperatuur , Stroom**
 - inwendige weerstand
 - hysterese
 - zelfontlading
 - veroudering
 - sampletijd
 - eigenverbruik van elektronica
 - laad- en ontlaadrendement

Elektrisch model

- Verschillen tussen deze modellen:
- oplaad of ontlaadaafhankelijkheid van de veroudering
- Het gedrag als de belasting tegen 0Ω bedraagt (grote stromen dus)

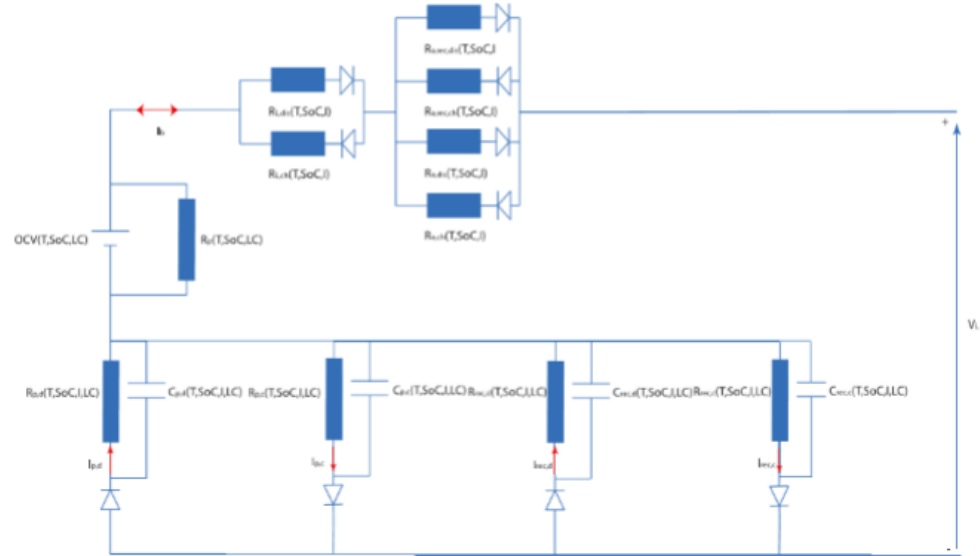
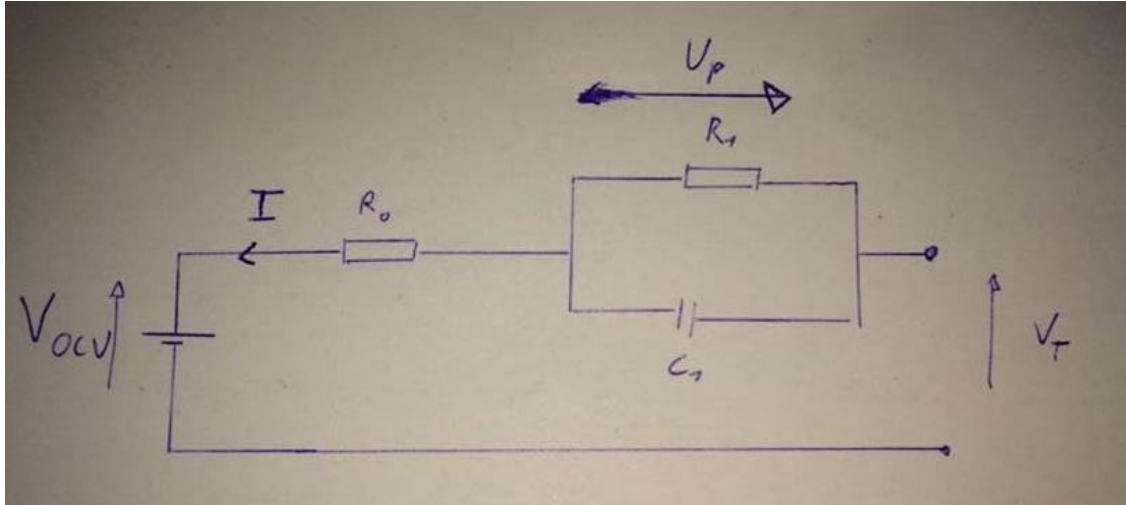


Figure 10. Extended Noshin's electrical battery model.

Elektrisch model



- Basisschema dat gebruikt wordt voor rekenwerk.
- De waarden voor de basisschema worden bepaald door parameterbepaling

Elektrisch model

$$\begin{cases} V_T = V_{ocv} + V_p + I \cdot R_o \\ \frac{dV_p}{dt} = \frac{I}{C_1} - \frac{V_p}{R_1 \cdot C_1} \end{cases}$$

- De volgende differentiaalvergelijkingen kunnen worden opgesteld

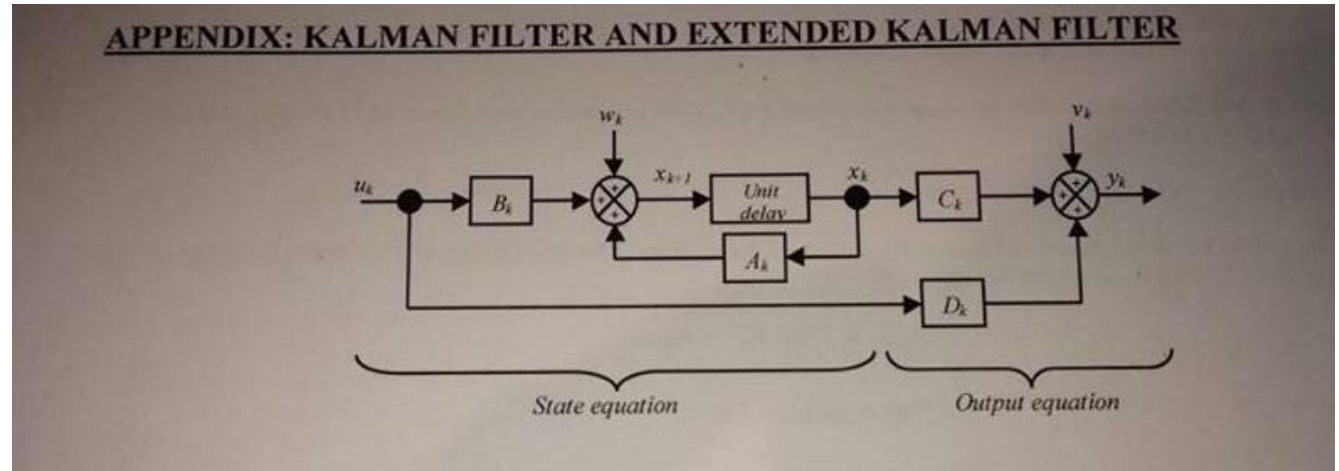
Elektrisch model

$$\begin{cases} V_{p,k+1} = V_{p,k} e^{(-\Delta t/\tau)} + \left[1 - e^{(-\Delta t/\tau)}\right] R_p \cdot I_k \\ V_{t,k+1} = V_{acv,k+1} + V_{p,k+1} + I_{k+1} \cdot R_o \end{cases}$$

- De differentiaalvergelijkingen kunnen worden omgezet naar discrete tijd

Elektrisch model

- Dit is het algemene blokschema van de kalmanfilter
- Met bijhorende vergelijkingen



$$\begin{cases} x_k = A_k x_{k-1} + B_k u_{k-1} + w_k \\ y_k = C_k x_k + D_k u_k + v_k \end{cases}$$

Elektrisch model

- Dit is het algemene blokschema van de kalmanfilter
- Met bijhorende vergelijkingen
- Uit de vergelijkingen van de discrete tijd kunnen de matrixen van de kalmanfilter bepaald worden

$$\begin{cases} x_k = A_k x_{k-1} + B_k u_{k-1} + w_k \\ y_k = C_k x_k + D_k u_k + v_k \end{cases}$$

$$\begin{cases} x(t) = \begin{bmatrix} U_p & SOC \end{bmatrix}^T \\ y(t) = U_t(t) \end{cases}$$

Elektrisch model

- Deze matrixen kunnen gevonden worden en zullen gebruikt worden voor de kalmanfiltering

$$\begin{cases} x_k = A_k x_{k-1} + B_k u_{k-1} + w_k \\ y_k = C_k x_k + D_k u_k + v_k \end{cases}$$

$$\begin{cases} A_k = \begin{bmatrix} e^{-\frac{T}{R_p C_p}} & 0 \\ 0 & 1 \end{bmatrix} & C_k = [1 \ \partial U_{ocv,k} / \partial SOC_k] \\ B_k = \begin{bmatrix} R_p \left(1 - e^{-\frac{T}{R_p C_p}} \right) \\ \frac{T}{3600 \times C_b} \end{bmatrix} & D_k = [R_0] \end{cases}$$

Elektrisch model

- Matrix-berekening nodig voor kalmanfilter.
- Dit principe werkt aan de hand van de extended kalman filter.
- Om de waarden van covariantiematrixen R_k en Q_k niet te moeten schatten wordt gebruik gemaakt van AEKF waarbij R_k en Q_k bij elke recursie wordt herberekend.

- Initialisatie

$$\begin{cases} \hat{x}_0 = E(x_0) \\ P_0 = \text{var}(x_0) \end{cases}$$

- Time Update

$$\hat{x}_{k+1}^- = A_k \hat{x}_k^+ + B_k u_k + w_k$$

$$P_k^- = A_k P_k A_k^T + Q_k$$

- Measurement update

$$H_k = P_k^- C_k^T / (C_k P_k^- C_k^T + R_k)$$

$$\hat{x}_k^+ = \hat{x}_k^- + H_k (y_k - C_k \hat{x}_k^- - D_k u_k)$$

$$P_k = (I - H_k C_k) P_k^-$$

Programma / BMS

Programma

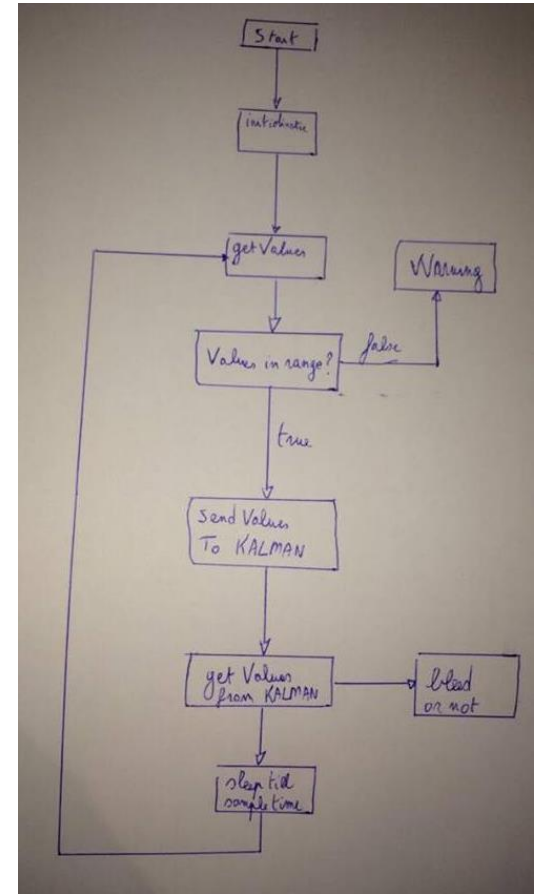
- Het programma bestaat momenteel 3 delen
- Het hoofdprogramma
- Parameterbepaling
- Kalman filter

Programma

- Het hoofdprogramma

Flowchart = opbouw van het hoofdprogramma

-bestaat uit een loop die warnings genereert als er fouten zijn.



Programma

- Het hoofdprogramma

-Data uit deze loop kan dan gebruikt worden voor een online uitlezing en bij de database



```
#-----Put all data in one row-----

listOfData.append(VoltageSlave1)
listOfData.append(VoltageSlave2)
listOfData.append(VoltageSlave3)
listOfData.append(VoltageMaster)
listOfData.append(Current)
listOfData.append(T)
listOfData.append(SOC0)
listOfData.append(I1)
listOfData.append(I2)
listOfData.append(I3)
listOfData.append(tm.strftime("%H:%M:%S", tm.localtime()))

previousData = [] #this row can be used for recursion
nowData = []

#-----

global totalruntime
totalruntime = 0

#-----

while(listOfData[0] < cut_off_voltage_high and listOfData[1] < cut_off_voltage_h

    print("programm running")

    if (totalruntime == 0):
        ''' initialize '''
        nowData = listOfData

        #do something to listOfData

    else
```


Programma

- Parameterbepaling

Dit programma zal via interpolatie uit gegevens de exacte waarden berekenen die gebruikt zullen worden voor het matrix rekenen

```
#-----  
  
#let op: SOC2 en SOC1 hebben een andere betekenis dan die van verder in het prog  
  
def getR0(SOC1, SOC2, Ib, I_b, T):  
    '''values without underscore are the k1'th (Ib) and values with underscore a  
  
    meanIb = (Ib + I_b)/2.0 # 2.0 delen voor floats  
    meanSOC = (SOC1 + SOC2)/2.0  
  
    SOCValues = []  
    IValues = []  
    TValues = []  
  
    sheet = R0dishtable  
    #-----if you want to change length of data, change these Values  
  
    rangeT = 4  
    #-----  
  
    rangeSOC = 1  
    value = 0  
    delta = sheet.cell_value(3,0)-sheet.cell_value(2,0)  
    while (value < 1):  
        value+=delta  
        rangeSOC+=1  
  
    for x in range(rangeSOC):  
        SOCValues.append(sheet.cell_value(x+2,0))  
  
    for x in range((sheet.nrows-2)/rangeSOC):  
        IValues.append(sheet.cell_value((x+2)+(rangeSOC-1)*x,1))  
  
    for x in range(rangeT):  
        TValues.append(sheet.cell_value(0,4*x+1))
```

Programma

- Parameterbepaling

Bijvoorbeeld

$T=16^{\circ}\text{C}$ in range (10 – 20)

$I = 70\text{A}$ in range (50 – 100)

$\text{SOC} = 83\%$ in range (80 – 85)

Er zijn nu $2^3=8$ waarden waar $4+2+1=7$ keer een interpolatieberekening wordt op uitgevoerd om tot 1 eindwaarde te komen

A	B	C	D	E	F	G	H	I	J
Temperat	0 °C			Temperat	10 °C			Temperat	20
SOC	I	Value		SOC	I	Value		SOC	I
0	5	1,1		0	5	1		0	5
0,05	5	1,3		0,05	5	1		0,05	5
0,1	5	1,1		0,1	5	1		0,1	5
0,15	5	1,4		0,15	5	1		0,15	5
0,2	5	1,1		0,2	5	1		0,2	5
0,25	5	1,1		0,25	5	1		0,25	5
0,3	5	1,5		0,3	5	1		0,3	5
0,35	5	1,1		0,35	5	1		0,35	5
0,4	5	1,1		0,4	5	1		0,4	5
0,45	5	1,1		0,45	5	1		0,45	5
0,5	5	1,6		0,5	5	1		0,5	5
0,55	5	1,1		0,55	5	1		0,55	5

Programma

- Kalman filter

In dit programma gebeuren de matrixberekeningen.

Dit programma is nog niet af.

```
-----  
#-----  
SOC1 = 0.5 #set value  
Upk = 1.96 #get this value out of main programm  
  
sigmaSOC = 0.02 #estimate of sqrt(variance) of SOC  
sigmaUpk = 0.02 #estimate of sqrt(variance) of Upk  
  
X0 = np.matrix([[Upk],[SOC1]])  
P0 = np.matrix([sigmaSOC**2,0],[0,sigmaUpk])  
#-----  
  
'''  
There is data at the k'th sample point. --> k  
There is data at the k-1'th sample point. --> k_1  
  
There is data at the k'th sample point after measurement update --> km  
There is data at the k-1'th sample point. --> k_1m  
'''  
  
Xk  
Xk_1  
Xkm  
Xk_1m  
  
uk  
uk_1 = #get current in this variable  
ukm  
uk_1m  
#-----TIME UPDATE -----  
  
Xk = Ak * Xk_1m + Bk * uk_1 # (1)  
  
Pk = Ak * Pkv * (Ak.transpose()) + Qk # (2)  
  
#-----MEASUREMENT UPDATE -----
```

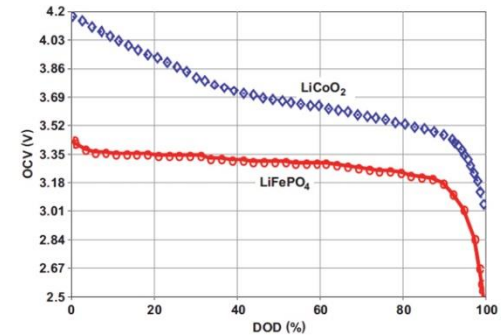
Resultaten metingen

Resultaten en metingen

- Er zijn momenteel 2 grote metingen gebeurd.(?)
- Meting1: Ontladen voor telkens 20 minuten en 3uur relaxeren met een ontladstroom van 15A tot de cutoff spanning
- Meting2: Opladen voor telkens 20 minuten en 3uur relaxeren met een ontladstroom van 15A tot de cutoff spanning

Resultaten en metingen

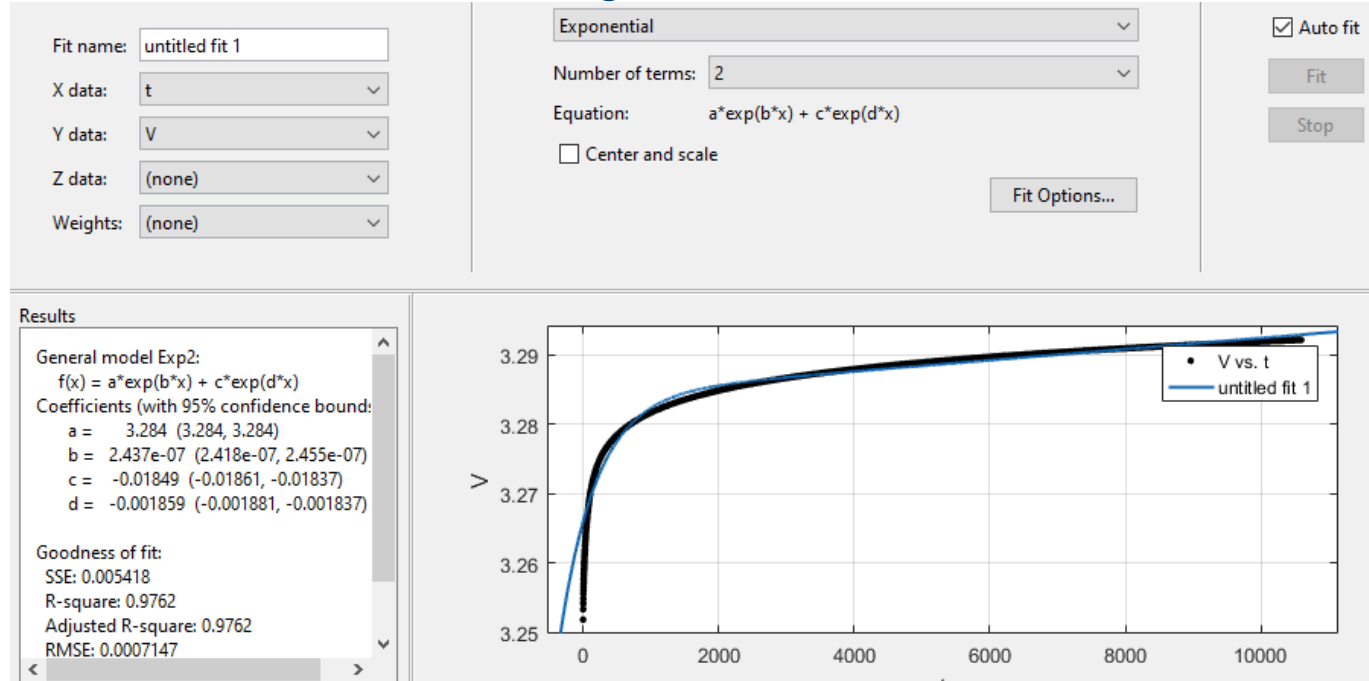
- Aan de hand van meting 1 en meting 2 kan de $OCV = f(SOC, T=20^{\circ}\text{C})$ curve worden opgesteld
- Deze wordt in literatuur vaak benaderd door een polynomiaal. Dit zorgt ook voor rekengemak in programmatie.



$$y = P_1 \times SOC^7 + P_2 \times SOC^6 + P_3 \times SOC^5 + P_4 \times SOC^4 + P_5 \times SOC^3 + P_6 \times SOC^2 + P_7 \times SOC + P_8$$

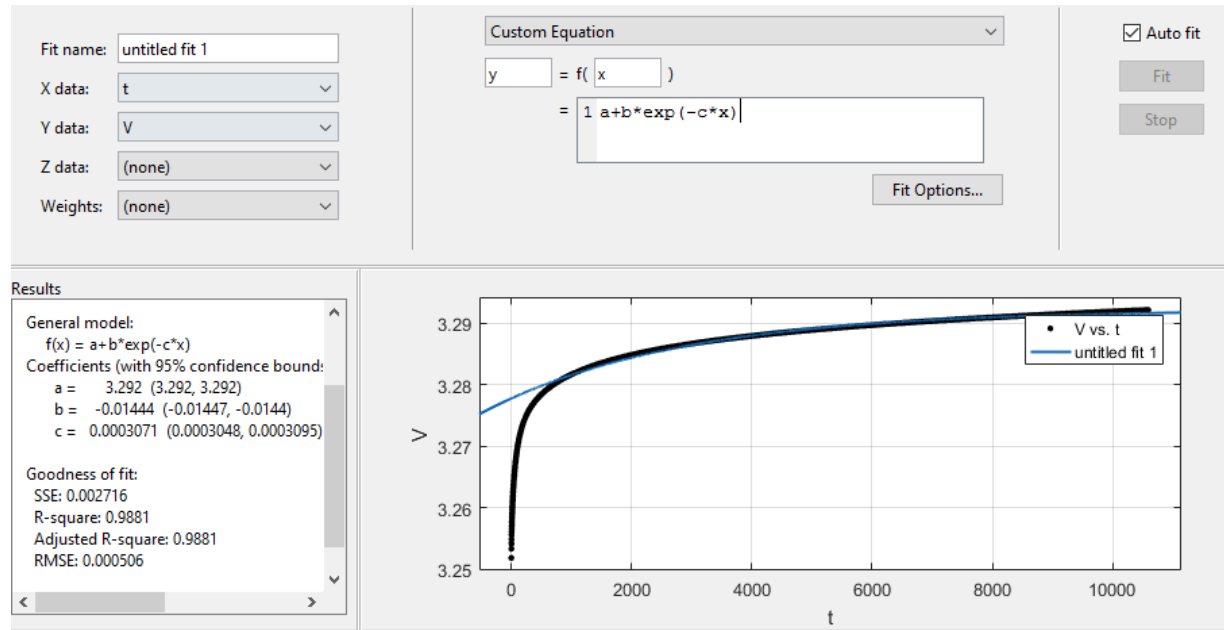
Resultaten en metingen

- 2 RC takken = zeer goede fit $R^2 = 0,995$



Resultaten en metingen

- 1 RC tak → $R^2 = 0,988$



Resultaten en metingen

- De fit met 2 RC takken is zeer goed.
- De fit met 1 RC tak is ook nog goed

Welke wordt gekozen?

-In Noshin's model en veel literatuur wordt er gekozen voor 1 RC tak

-In sommige gevallen kiest men voor 2 RC takken

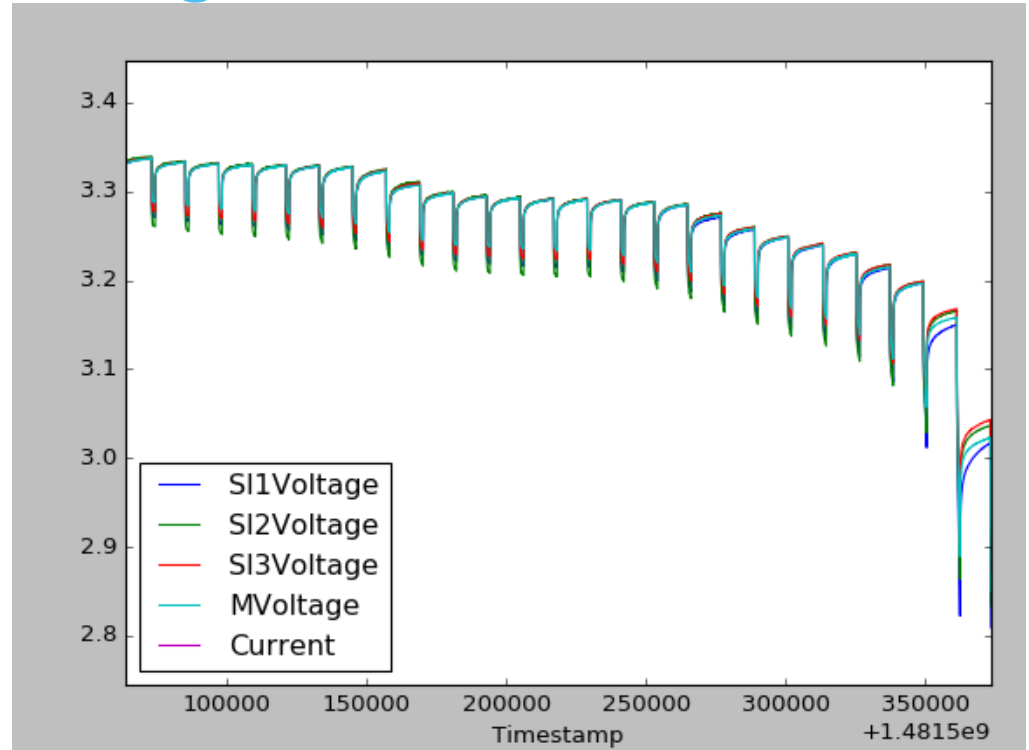
In literatuur is meer informatie te vinden over 1 RC tak systemen

Uitbreiden naar een 2RC tak systeem is wel wat werk maar zeker niet onmogelijk

Resultaten en metingen

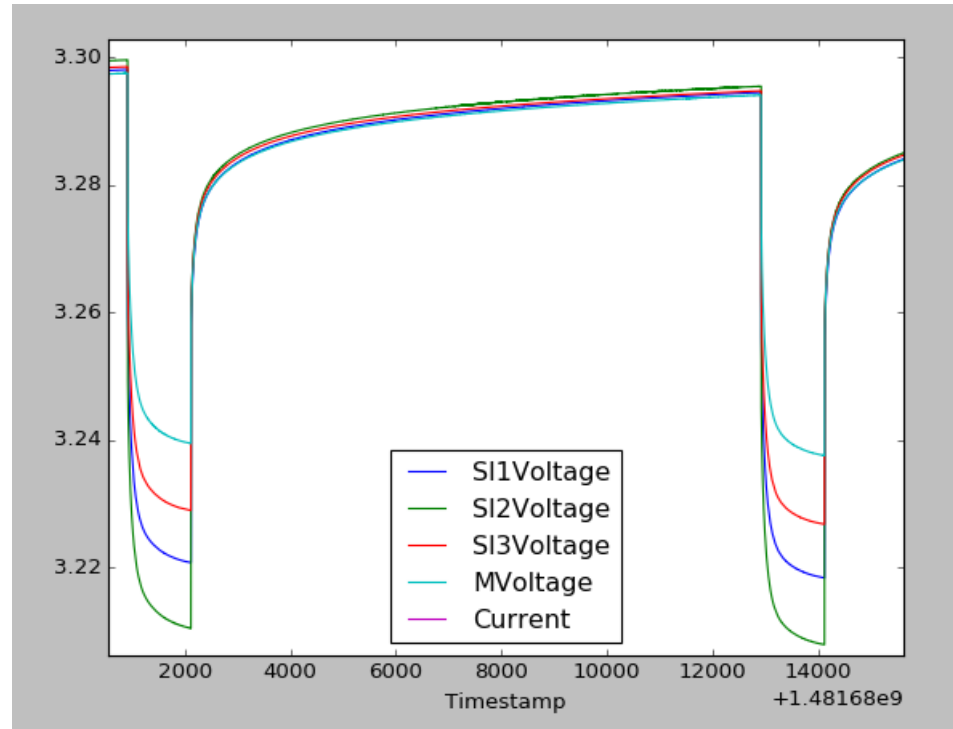
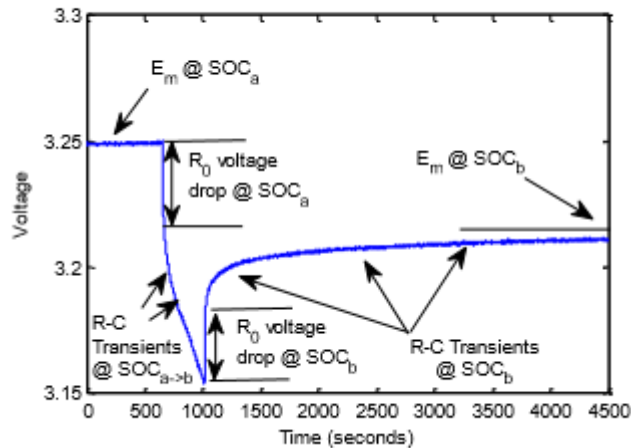
- METING 1

Ontladen-rusten-ontladen-rusten-...



Resultaten en metingen

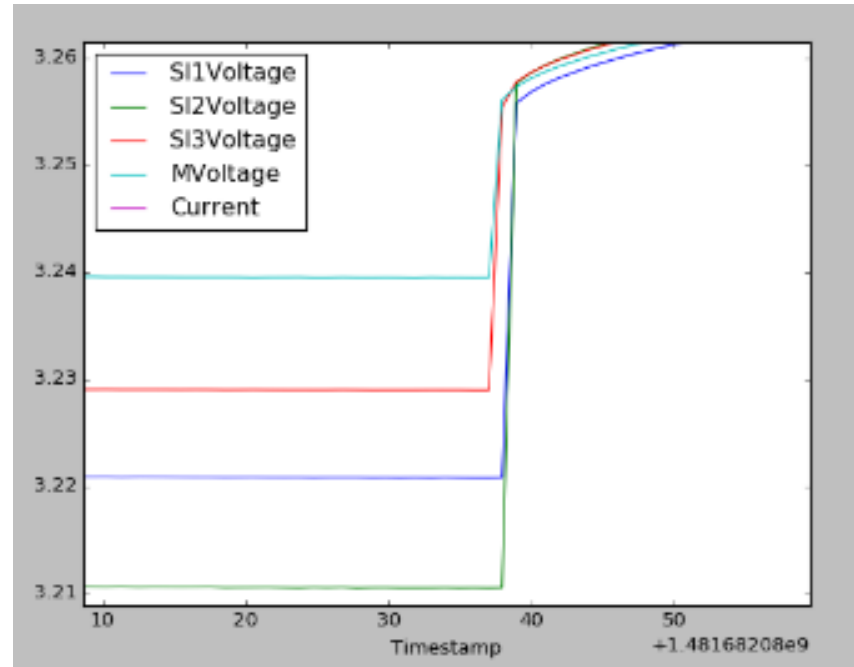
- Test op een deel van de meting



Resultaten en metingen

De inwendige weerstand R_0 is hetzelfde in beide gevallen (1RC keten of 2)

In dit geval $1\text{m}\Omega < R_0 < 3\text{m}\Omega$



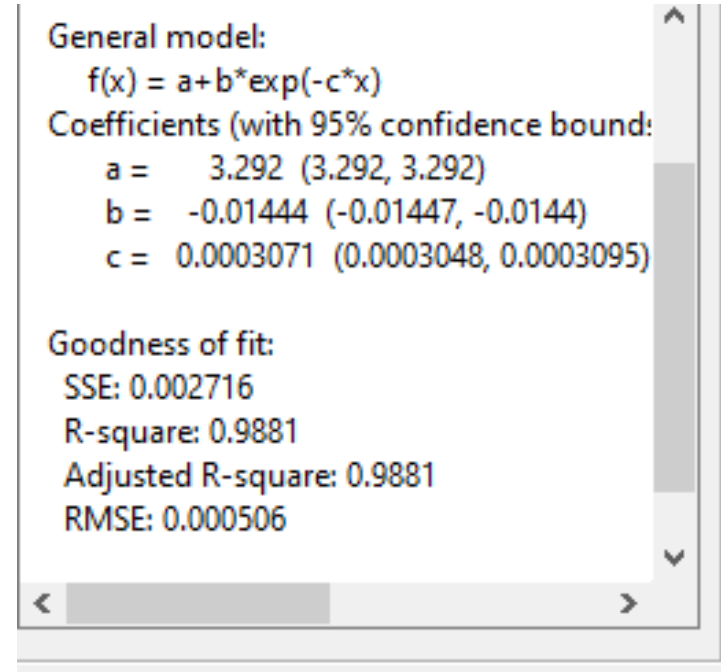
Resultaten en metingen

- Test op een deel van de meting

- $RC = 0,000307$

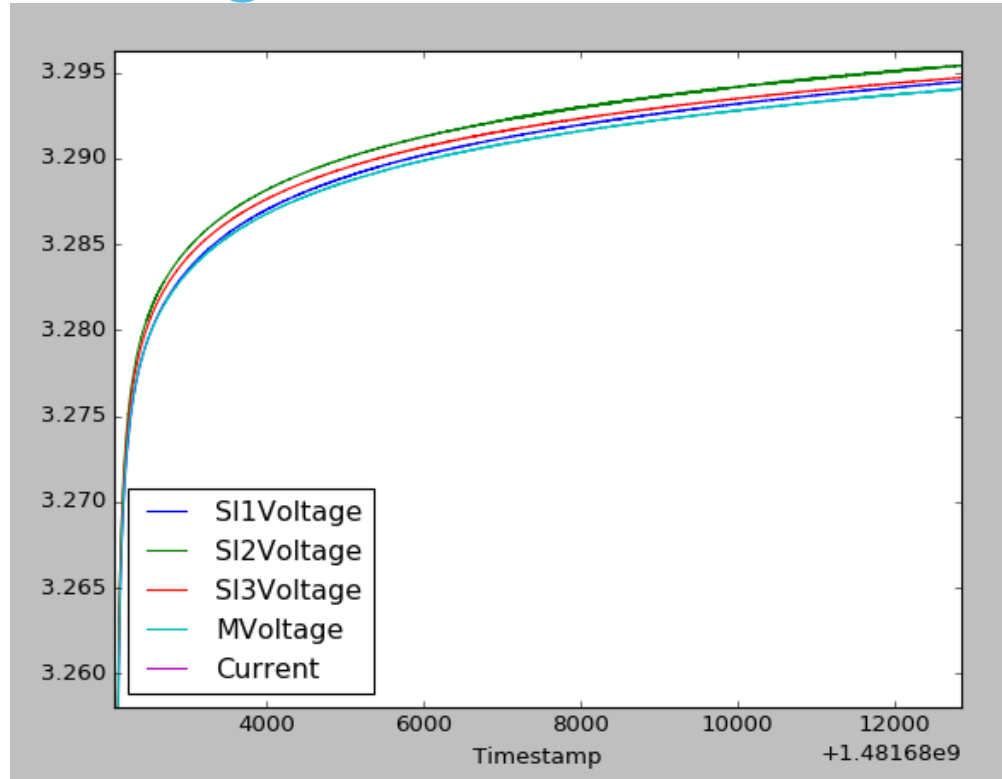
- $\frac{\frac{5}{RC}}{3600} = 4,5h$

na 4,5uur zou de relaxatie
quasi niet meer veranderen
volgens de fit
→ komt goed overeen



Resultaten en metingen

- → voor een voorbeeld
- $R_0 = 2\text{m}\Omega$
- $R_1 = 2,6\text{m}\Omega$
- $C_1 = 0,11\text{F}$
- Dit gebeurt telkens voor deelcycli en hieruit kan telkens 3 parameters bepaald worden



Aanpak semester 2

Aanpak semester 2

- De metingen

-Hetzelfde stroomien moet uitgevoerd worden bij verschillende belastingen en temperaturen om meer meetdata te verkrijgen.

-Deze meetdata moet dan worden geanalyseerd of er effectief een noemenswaardige invloed is op het model

Te beginnen bij een ontlaad en oplaadstroom van (hoe?) $50A = C/2$
Dit is de standaard discharge current van de cellen.

Aanpak semester 2

- Het programma
 - Zo snel mogelijk afwerken van Kalman filtering
 - Testen SOC estimatie aan de hand van datasets

Aanpak semester 2

- Het bleeden

-Onderzoeken vanaf welke Δ SOC tussen de cellen het interessant wordt om te bleeden

-Bij elk minuscule klein ladingsverschil hoeft niet gebleed te worden.

→ Vanaf wanneer wordt bleeden performant?

Planning

Tijdsbenutting
Voorstel planning semester

Planning semester 2

taak	periode
Afwerken programma	Februari-Maart
Metingen uitvoeren	Februari - April
Testen programma	Maart - April
Onderzoek naar bleeden	April
Boek schrijven	April – mei

Bedankt voor uw aandacht