

Zusammenfassung: 29. Juni 2021

Pramayuda Hidayah Saleh

1 Schätzung von Batterieparameter mit Systemidentification-Toolbox

Um der Fit-Wert des geschätzten Modells zu verbessern wird die Parameterschätzung für das Batteriemodell mithilfe des Systemidentification-Toolbox und die *nlgreyest()*-Funktion gemacht. Das nichtlineare Modell enthält 3 Parameter für die $SoC - U_{OC}$ -Kurve (a_1, a_2, a_3) und 4 für die Batterieparameter ($\frac{1}{C}, \frac{1}{R}, \frac{1}{R_{0,d}}, \frac{1}{R_{0,c}}$). Die erste 3 Parameter werden fix gehalten und die Anderen werden mit Initialwert x_0 (wie bei EKF) geschätzt.

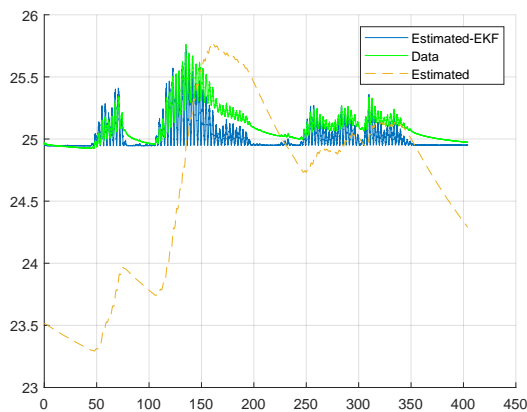


Abbildung 1: y-Achse: Spannung, x-Achse: Zeit, Schätzung mit Initialparameter x_0

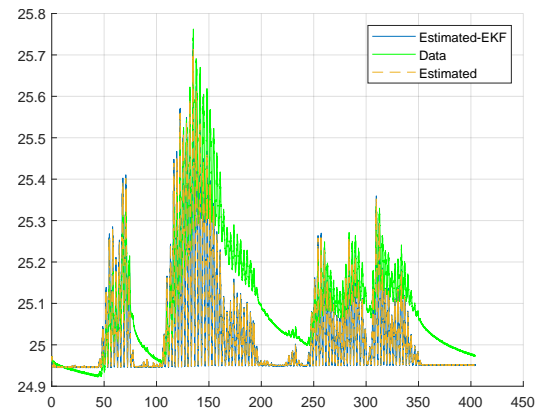


Abbildung 2: y-Achse: Spannung, x-Achse: Zeit, Schätzung mit Initialparameter aus dem EKF x_{conv}

Das Toolbox braucht mehr als 1 Stunde und das Ergebnis ist schlechter als das vom EKF (niedriger $fit_{compare,NRMSE}$). Deswegen werden die konvergierte, vom EKF geschätzten Parameterwerten als Initialwert für das *idnlgrey* Modell verwendet, das aber ähnlichen simulierten Ausgang wie die Schätzung mit nur EKF ergibt. Beide Ergebnisse können bei Abbildung 1 und 2 gesehen werden. Der Vergleich zwischen den Initialparameter und geschätzten Parameter sind unten geschrieben

Parameter	Aus EKF	Aus <i>nlgreyest</i> -Funktion
$\frac{1}{C}$	$4,79 \frac{1}{F}$	$3,38 \frac{1}{F}$
$\frac{1}{R}$	$39,68 \frac{1}{\Omega}$	$39,18 \frac{1}{\Omega}$
$\frac{1}{R_{0,d}}$	$1,009 \frac{1}{\Omega}$	$1,028 \frac{1}{\Omega}$
$\frac{1}{R_{0,c}}$	$1,009 \frac{1}{\Omega}$	$1,009 \frac{1}{\Omega}$

Tabelle 1: Parametervergleich zw. EKF und *nlgreyest*-Funktion

Nun werden ($\frac{1}{C}, \frac{1}{R}, \frac{1}{R_{0,d}}, \frac{1}{R_{0,c}}$) fix gehalten und (a_1, a_2, a_3) werden geschätzt (mit Initialwerte

aus der EKF Schätzung). Der simulierte Ausgang des neuen Modells hat einen besseren $\text{fit}_{\text{compare},\text{NRMSE}}$. Die Schätzung mit dem Toolbox ist jetzt schneller allerdings dauert es noch mehr als 15 Minuten und der Ausgang läuft wie in Abbildung 3

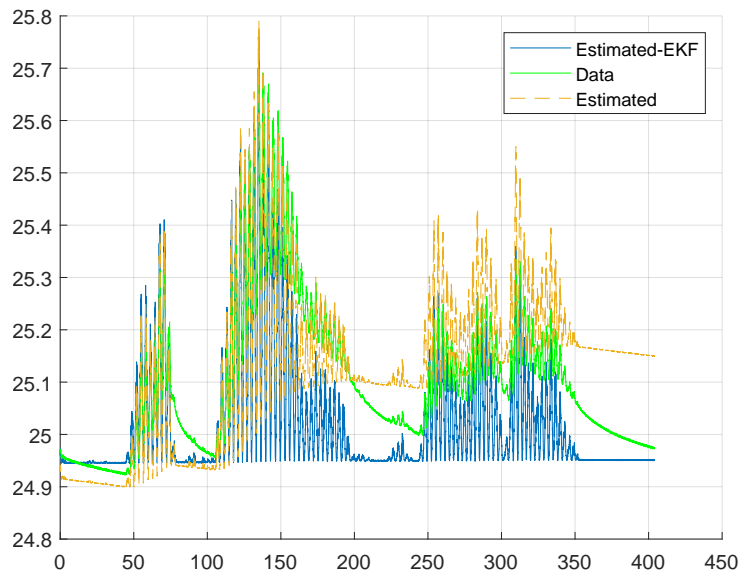


Abbildung 3: y-Achse: Spannung, x-Achse: Zeit, Schätzung mit festen Batterieparameter aus dem EKF x_{conv}

Alle Werte von $\text{fit}_{\text{compare},\text{NRMSE}}$ werden unten in der Tabelle dargestellt.

Algorithm	$\text{fit}_{\text{compare},\text{NRMSE}}$ in %
EKF	20.02
TRR ¹ mit Init-wert x_0	-479.76
TRR mit Init-wert x_{conv}	17.23
TRR mit festen Batterieparameter x_{conv}	33.20
TRR mit geschätzten U_{OC} -Parameter und Init-wert x_{conv}	32.96
TRR zum Schätzen aller Parameter mit Init-wert x_{conv}	31.87

Tabelle 2: $\text{fit}_{\text{compare},\text{NRMSE}}$ -Werte (TRR = Trust Region based on Interior-Reflective Newton Method)

Die Schätzung mit EKF und danach *nlgreyest*-Funktion zur Schätzung von Parameter für $\text{SoC} - U_{OC}$ -Kurve funktioniert am besten, allerdings macht die Kurve jetzt physikalisch kein Sinn.

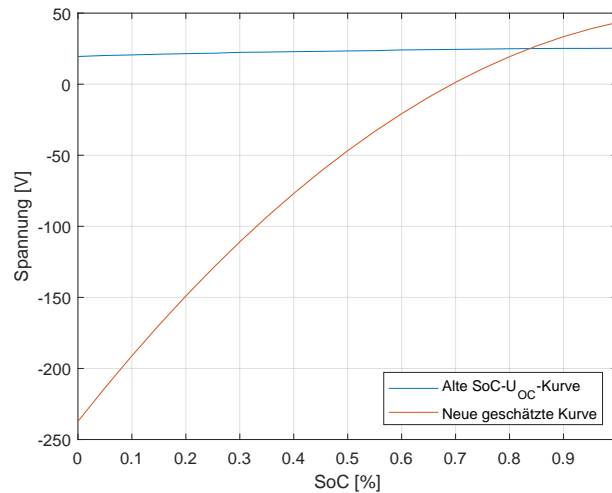


Abbildung 4: $SoC - U_{OC}$ -Kurve

2 Änderung von Parameter bei $SoC - U_{OC}$ -Kurve

Der simulierte Ausgang stimmt nicht mit dem gemessenen Ausgang überein. Eine Vorschlag zur Verbesserung ist die Änderung von Parameter a_1 , a_2 und a_3 bei der $SoC - U_{OC}$ -Kurve. Bei Abbildung ?? scheint es, dass die Spannung U_{OC} entweder zu klein ist (wenige Wirkung auf die Ausgang U_L hat) oder zu schnell abklingt. Wenn die Formell für U_{OC}

$$U_{OC}(SoC) = a_1 SoC^2 + a_2 SoC + a_3, a_1 < 0, a_2 > 0$$

angeschaut wird, können die Probleme gelöst, indem:

1. a_3 erhöht wird, oder
2. a_1 verkleinert wird, damit die Steigung für höhere SoC kleiner wird.

Mit der geänderte Parameter ist das Ergebnis allerdings nicht stark geändert, da der EKF die Batterieparameter so schätzt, dass die Änderung bei der SoC-Kurve fast ausgeglichen wird.

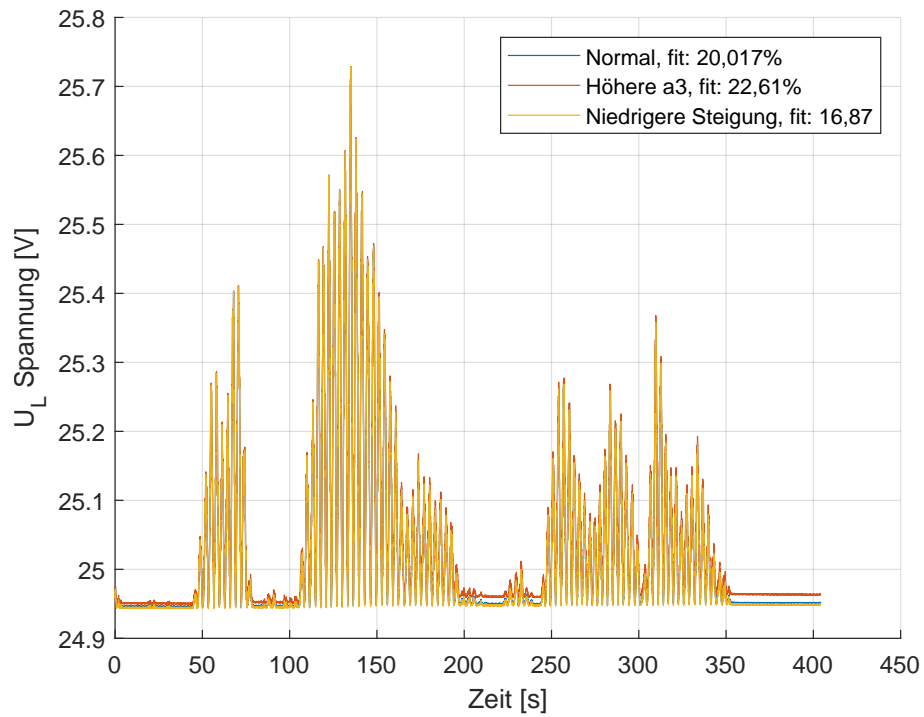


Abbildung 5: Simulierter Ausgang \hat{y} mit geänderter SoC Kurve

Die geschätzten Parameter sind folgende:

Parameter	Normale SoC	Höhere a_3	Niedrigere Steigung
$\frac{1}{C}$	$4,79 \frac{1}{F}$	$2,73 \frac{1}{F}$	$4,31 \frac{1}{F}$
$\frac{1}{R}$	$39,68 \frac{1}{\Omega}$	$39,68 \frac{1}{\Omega}$	$39,68 \frac{1}{\Omega}$
$\frac{1}{R_0}$	$1,009 \frac{1}{\Omega}$	$1,018 \frac{1}{\Omega}$	$1,006 \frac{1}{\Omega}$

Tabelle 3: Parametervergleich bei geänderten $SoC - U_{OC}$ -Parameter

3 Code für nichtlineares Batteriemodell

```
1 function [dx, y] = lead_battery_nl(t, x, u, a1, a2, a3, Cm1, Rm1,
    ...
2 R0dm1, R0cm1, varargin)
3 %% Setup
4 % Cm1 = 1/C; Rm1 = 1/R; R0m1 = 1/R0
5 Q_e = 2.9*3600;
6 if u>=1
7     R0m1 = R0cm1;
8 else
9     R0m1 = R0dm1;
10 end
11
12 %% Output
13 y = a1*x(1)^2+a2*x(1)+a3 + x(2) + u/R0m1;
14
15 %% Derivative
16 dx(1) = u/Q_e;
17 dx(2) = Cm1*(u-x(2)*Rm1);
18
19
20 end
```