EKF Implementierung zur Batterie SOC-Schätzung

Pramayuda H. Saleh

1 Batterie Modell

Aus dem Paper[1] wurde ein elektriches Schaltungsmodell von der Batterie wie in Abb.1 genommen.

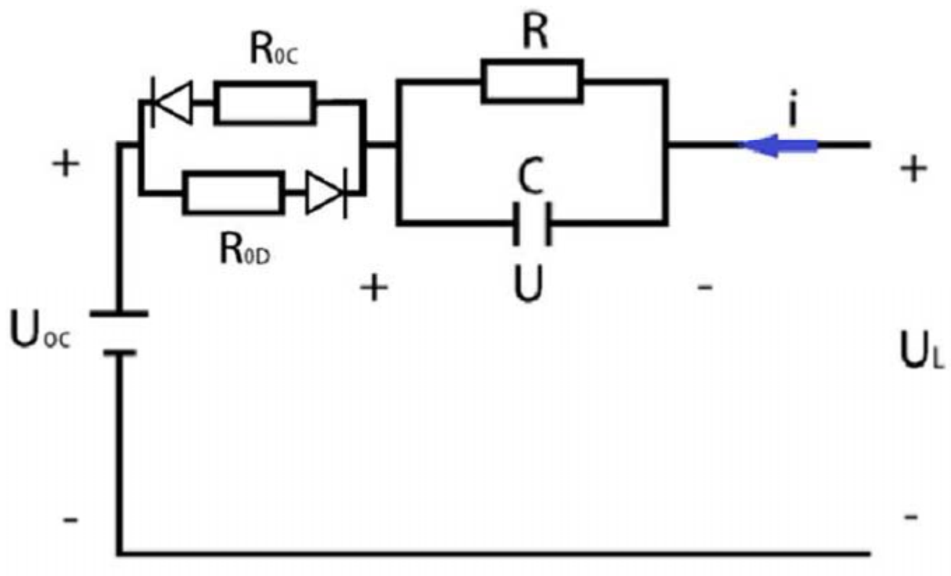

Abbildung 1: Batterie Modell

2 Zustandsraummodell

Die vorgeschlagene Zustände in der Literatur sind $x(t) = \begin{pmatrix} U_L(t) & U(t) & \frac{1}{R} & \frac{1}{C} \end{pmatrix}^T$ mit Ein- und Ausgang $(u(t), y(t)) = \begin{pmatrix} i(t), & U_L(t) \end{pmatrix}$. Leider ist dieser Ansatz nicht genau einsetzbar, da die linearisierte A Matrix 2 Terme U_{OC} und R_0 enthält, was nicht berechnet wurde. Deswegen werden andere Zustände für das System verwendet:

$$x(t) = \begin{pmatrix} \frac{1}{C} & \frac{1}{R} & \frac{1}{R_0} & U_{OC}(t) & U(t) \end{pmatrix}^T$$

mit Ein- und Ausgang:

$$(u(t), y(t)) = \begin{pmatrix} U_L(t), & i(t) \end{pmatrix}$$

Hier sind $x_j(t)$ mit $j \in (1, 4)$ Parameter, die von dem erweiterten Kalmanfilter (EKF) geschätzt werden.
Dazu braucht man Gleichungen zur Bestimmung der nichtlinearen Funktionen $\dot{x}(t) = f(x(t), u(t))$ und $y(t) = g(x(t), u(t))$:

$$\dot{U}(t) = \frac{1}{C}\left(i(t) - i_R(t)\right) = \frac{1}{C}\left(i(t) - \frac{U(t)}{R}\right) \tag{2.0.1}$$

$$U_L(t) = U_{OC}(t) - U(t) - R_0 i(t) \tag{2.0.2}$$

$$i(t) = \frac{1}{R_0} \left(U_{OC}(t) - U(t) - U_L(t) \right) \tag{2.0.3}$$

Die Funktion $g(x(t), u(t))$ ist genau die rechte Seite der Gleichung 2.0.3. Die Funktionsmatrix $f(x(t), u(t))$ ist dann:

$$f = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{C} \left(\frac{U_{OC} - U_L}{R_0} - U \left(\frac{1}{R} + \frac{1}{R_0} \right) \right) \end{pmatrix}$$

3 EKF Implementierung

Aus dem nichtlinearen Modell werden die Matrizen $A[k]$ und $C[k+1]$ aus der diskretisierten Funktion $f(x_k, u_k)$ hergeleitet.

$$f(x_k, u_k) = \begin{pmatrix} x_{1k} \\ x_{2k} \\ x_{3k} \\ x_{4k} \\ x_{5k} + \Delta x_{1k} \left[x_{3k} \left(x_{4k} - u_k \right) - x_{5k} \left(x_{2k} + x_{3k} \right) \right] \end{pmatrix}$$

$$A[k] = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \Delta \left[x_3(x_4 - u) - x_5(x_2 + x_3) \right] & -\Delta x_1 x_5 & \Delta x_1 (x_4 - x_5 - u) & \Delta x_1 x_3 & 1 - \Delta x_1 (x_2 + x_3) \end{pmatrix}_{|x = x_{k|k}, u}$$

$$C[k+1] = \begin{pmatrix} 0 & 0 & x_4 - x_5 - u & x_3 & -x_3 \end{pmatrix}_{|x = \hat{x}_{k+1|k}, u = uk}$$

4 Ergebnis

Als initiale Zustände werden $x_{k_0+1|k_0} = \begin{pmatrix} 0.0011\frac{1}{F} & 23.8095\frac{1}{\Omega} & -1.9\frac{1}{K\Omega} & 12V & 0V \end{pmatrix}^T$. Die Auswahl von initialen Zuständen sind nur jetzt Platzhalter mit der Höffnung, dass die Zustände unabhängig von Initialwerten richtig konvergieren, und der Annahme, dass die Kapazität C am Anfang nicht geladen ist.

Diagonalelementen von P_0 sind das Quadrat des maximalen Fehler. $W[k]$ ist eine Einheitsmatrix mit Diagonalelementen $0,1$. $Z[k]$ wurde ungefähr aus dem angenommene Rauschen von gemessenem Ausgang geschätzt.

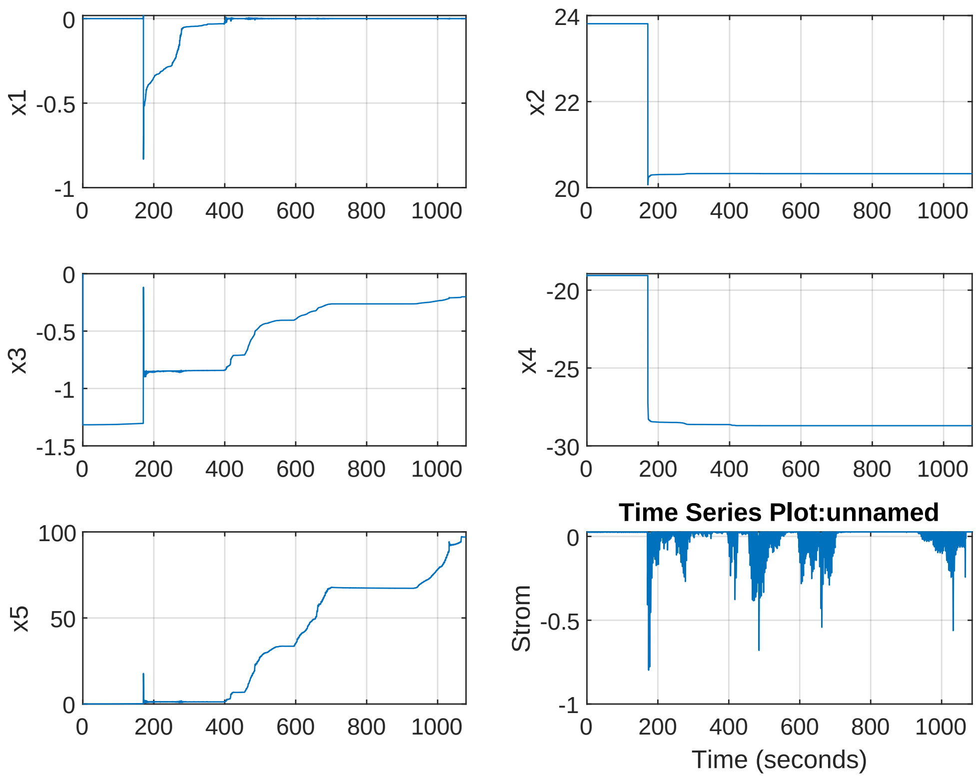

Abbildung 2: Schätzung von Zuständen

Die Resultat in Abb.2 ist nicht wie erwartet. Kapazität C oder die Umkehrung des 1. Zustand ist negativ, was positiv sein sollte. Die andere Zustände $\frac{1}{R_0}$ und U_{OC} sind auch negativ. Letztes mal ist eine sehr größe Spannung über die Kapazität zu sehen ist.
Ein möglicher Fehler könnte bei der Messung auftreten, da die gemessene Spannung U_L im Bereich $]25,1;25,6[$ liegen, obwohl die verwendete Batterie eine maximale Klemmenspannung von $12V$ hat. Es ist auch möglich, dass das verwendete Modell nicht passt.

5 Matlab Funktion

```
function x_k  = extended_kf(u_k,y_k,x0, P0, W_k, Z_k, Delta)
    % Extended Kalman Filter for estimating Capacity and U_L

    % Initialisierung äPrdiktion bei k = 0
    persistent P_k_p x_k_p
    if isempty(P_k_p)
        P_k_p = P0;
    end
    if isempty(x_k_p)
        x_k_p = x0;
    end

    % y_hat_k, C, K, x_k, P_k
    y_hat_k = x_k_p(3)*(x_k_p(4)-x_k_p(5)-u_k);
    C_k = [0, 0, x_k_p(4)-x_k_p(5)-u_k, x_k_p(3), -x_k_p(3)];
    K_k = P_k_p*C_k'/(Z_k + C_k*P_k_p*C_k');
    x_k = x_k_p + K_k*(y_k - y_hat_k);
    P_k = P_k_p - K_k*C_k*P_k_p;

    % Calculate A, prediction of x and P
    A_k = [[eye(4), zeros(4,1)];
       [Delta*(x_k(3)*(x_k(4)-u_k)-x_k(5)*...
        (x_k(2)+x_k(3))), -Delta*x_k(1)*x_k(5), Delta*x_k(1)*...
        (x_k(4)-u_k-x_k(5)),
           Delta*x_k(1)*x_k(3),1-Delta*x_k(1)*(x_k(2)+x_k(3))]];
    x_kp1_p = [x_k(1);x_k(2);x_k(3);x_k(4);x_k(5)+Delta*x_k(1)*...
        (x_k(3)*(x_k(4)-u_k)-x_k(5)*(x_k(2)+x_k(3)))];

    P_kp1_p = A_k*P_k*A_k' + W_k;

    % Update persistent variable
    x_k_p = x_kp1_p;
    P_k_p = P_kp1_p;
end
```

Literatur

[1] R. Yamin und A. Rachid. “Modeling and Simulation of a Lead-Acid Battery Packs in MATLAB/Simulink: Parameters Identification Using Extended Kalman Filter Algorithm”. In: 2014 UKSim-AMSS 16th International Conference. IEEE, März 2014. ISBN: 978-1-4799-4922-9.