عنوان :

گزارش شبیه سازی باتری به روش کالمن فیلتر

نویسنده :

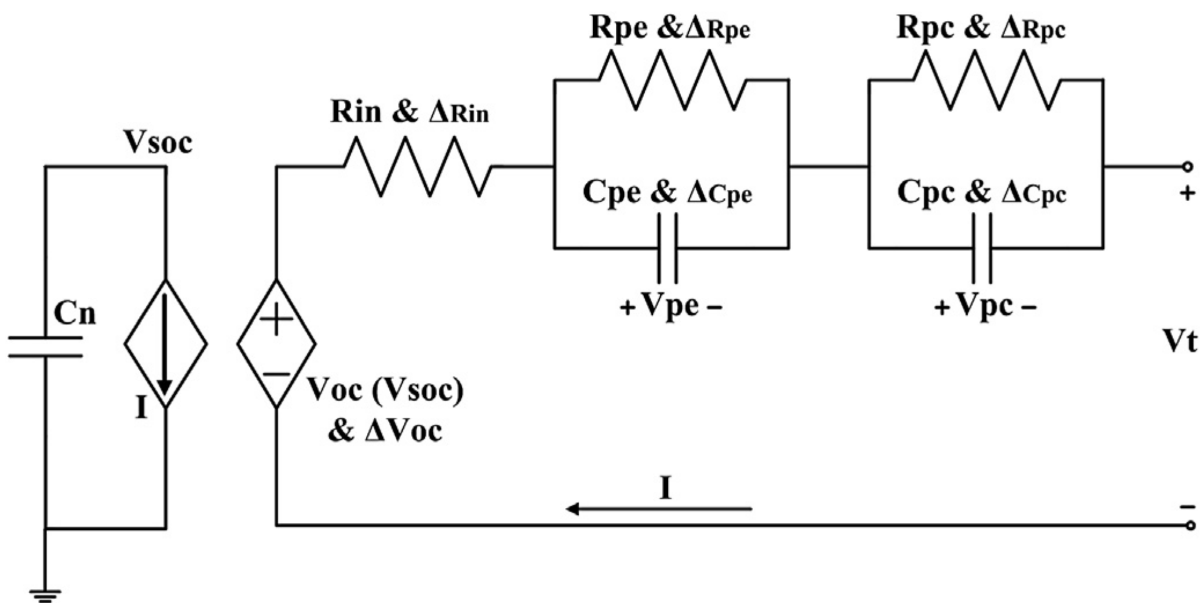
نیما آقایان مشهدی

مرداد ۱۴۰۰

فهرست

گزارش

در این پژوهش مقاله [۱] به روش UKF شبیه سازی شد. تمامی کدها با توجه به فرمول‌های زیر نشان شده‌اند. مدل باتری و مشخصات آن در قسمت زیر قرار داده شده است.



$$V_t = V_{oc}(Soc) - V_{pe} - V_{pc} - IR_{in} + \Delta_{uncertain} \quad (2)$$

The time derivatives of SOC and polarisation voltages give

$$\dot{S}oc = -(I/C_n) + \Delta f_2 \quad (3)$$

$$\dot{V}_{pe} = -V_{pe}/(R_{pe}C_{pe}) + I/C_{pe} + \Delta f_3 \quad (4)$$

$$\dot{V}_{pc} = -V_{pc}/(R_{pc}C_{pc}) + I/C_{pc} + \Delta f_4 \quad (5)$$

$$V_{oc}(Soc) = \kappa Soc + v \quad (6)$$

$$\dot{V}_t = -\kappa(I/C_n) + V_{pe}/(R_{pe}C_{pe}) - I/C_{pe} + V_{pc}/(R_{pc}C_{pc}) - I/C_{pc} + \Delta f_1 \tag{7}$$

$$\begin{aligned} \dot{V}_t &= -a_1 V_t + a_1 V_{oc}(Soc) - a_3 V_{pe} - a_4 V_{pc} - b_1 I + \Delta f_1 \\ \dot{S}oc &= a_2 V_t - a_2 V_{oc}(Soc) + a_2 V_{pe} + a_2 V_{pc} + \Delta f_2 \\ \dot{V}_{pe} &= -a_4 V_{pe} + b_2 I + \Delta f_3 \\ \dot{V}_{pc} &= -a_3 V_{pc} + b_3 I + \Delta f_4 \end{aligned} \tag{8}$$

where $a_1 = 1/(R_{pe}C_{pe}) + 1/(R_{pc}C_{pc})$, $a_2 = 1/(R_{in}C_n)$, $a_3 = 1/(R_{pc}C_{pc})$, $a_4 = 1/(R_{pe}C_{pe})$, $b_1 = \kappa/C_n + R_{in}/(R_{pe}C_{pe}) + 1/C_{pe} + R_{in}/(R_{pc}C_{pc}) + 1/C_{pc}$, $b_2 = 1/C_{pe}$ and $b_3 = 1/C_{pc}$.

کد قسمت مدل باتری

```
function [vt,socdot,vpedot,vpcdot,y,voc]  = fcn(s,vpe,vpc,I)
%% Battery Parameters
Rpe=4.96e-3;
Rpc=2.86e-3;
Cpe=4.93e3;
Cpc=14.33e3;
Rin=102.5e-3;
Cn=5*3600;
k=0.55;
%%
a1 = 1/(Rpe*Cpe)+1/(Rpc*Cpc);
a2 = 1/(Rin*Cn);
a3 = 1/(Rpc*Cpc);
a4 = 1/(Rpe*Cpe);
b1 = k/Cn + 1/Cpe + 1/Cpc + Rin/(Rpe*Cpe)+ Rin/(Rpc*Cpc);
b2 = 1/Cpe;
b3 = 1/Cpc;
%%
voc=k*s+3.625;
vt=voc-vpe-vpc-Rin*I;
socdot=a2*vt - a2*(voc) + a2*vpc + a2*vpe;%-0.0326*uncer;
vpedot=-a4*vpe + b2*I;%+0.0122*uncer;
vpcdot=-a3*vpc+b3*I;%+0.0204*uncer;
y=vt;
```

قسمت الگوریتم کالمن UKF بر اساس این متن اجرا شده است.

Initialize with:

$$\hat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0] \tag{7.35}$$

$$\mathbf{P}_0 = \mathbb{E}[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T] \tag{7.36}$$

$$\hat{\mathbf{x}}_0^a = \mathbb{E}[\mathbf{x}^a] = [\hat{\mathbf{x}}_0^T \ \mathbf{0} \ \mathbf{0}]^T \tag{7.37}$$

$$\mathbf{P}_0^a = \mathbb{E}[(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)^T] = \begin{bmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{\mathbf{v}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}^{\mathbf{n}} \end{bmatrix} \tag{7.38}$$

For $k \in \{1, \ldots, \infty\}$,

Calculate sigma points:

$$\boldsymbol{\mathcal{X}}_{k-1}^a = \left[\hat{\mathbf{x}}_{k-1}^a \quad \hat{\mathbf{x}}_{k-1}^a + \gamma\sqrt{\mathbf{P}_{k-1}^a} \quad \hat{\mathbf{x}}_{k-1}^a - \gamma\sqrt{\mathbf{P}_{k-1}^a}\right] \tag{7.39}$$

Time update:

$$\boldsymbol{\mathcal{X}}_{k|k-1}^x = \mathbf{F}[\boldsymbol{\mathcal{X}}_{k-1}^x, \mathbf{u}_{k-1}, \boldsymbol{\mathcal{X}}_{k-1}^v] \tag{7.40}$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}_{i,k|k-1}^x \tag{7.41}$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1}^x - \hat{\mathbf{x}}_k^-][\mathcal{X}_{i,k|k-1}^x - \hat{\mathbf{x}}_k^-]^T \tag{7.42}$$

$$\boldsymbol{\mathcal{Y}}_{k|k-1} = \mathbf{H}[\boldsymbol{\mathcal{X}}_{k|k-1}^x, \boldsymbol{\mathcal{X}}_{k-1}^n] \tag{7.43}$$

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,k|k-1} \tag{7.44}$$

Measurement update equations:

$$\mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-][\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T \tag{7.45}$$

$$\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-][\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T \tag{7.46}$$

$$\mathcal{K}_k = \mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}^{-1} \tag{7.47}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathcal{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \tag{7.48}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathcal{K}_k \mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} \mathcal{K}_k^T \tag{7.49}$$

where, $\mathbf{x}^a = [\mathbf{x}^T \ \mathbf{v}^T \ \mathbf{n}^T]^T$, $\boldsymbol{\mathcal{X}}^a = [(\boldsymbol{\mathcal{X}}^x)^T \ (\boldsymbol{\mathcal{X}}^v)^T \ (\boldsymbol{\mathcal{X}}^n)^T]^T$, $\gamma = \sqrt{(L+\lambda)}$, λ=composite scaling parameter, L=dimension of augmented state, $\mathbf{R}^{\mathbf{v}}$=process noise cov., $\mathbf{R}^{\mathbf{n}}$=measurement noise cov., W_i=weights as calculated in Eqn. 7.34.

Table 7.3.1: Unscented Kalman Filter (UKF) equations

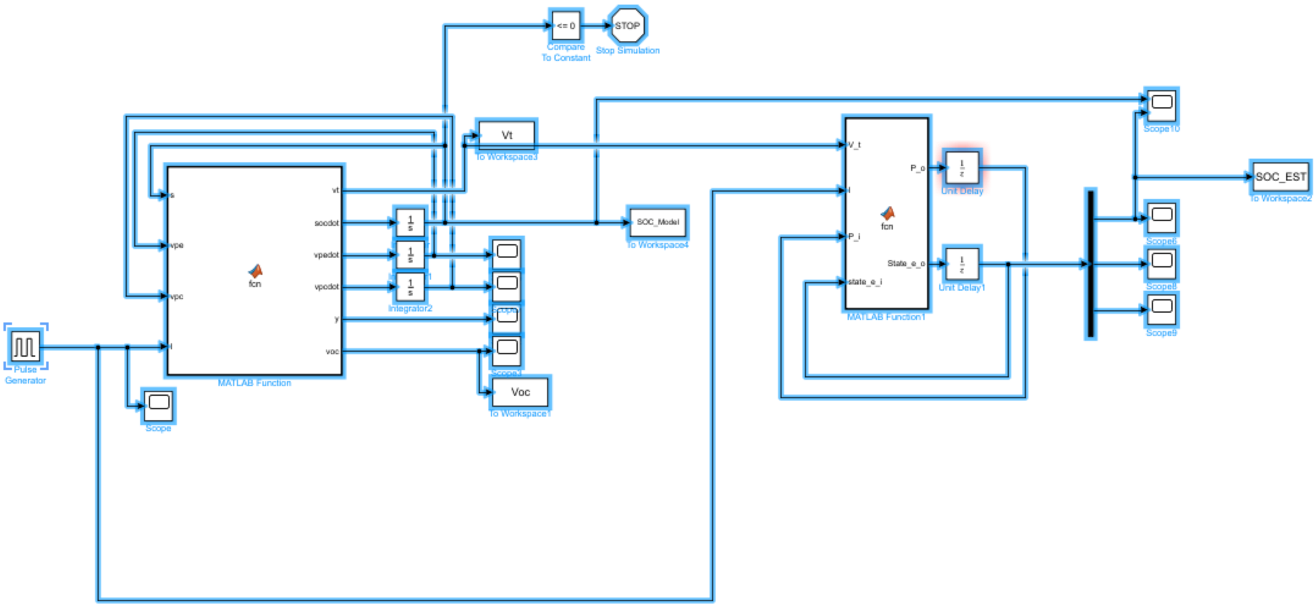
هر قسمت از کد با توجه به این فرمول ها نشان گذاری شده است.

کد قسمت تخمین‌گر

```
function [P_o,State_e_o] = fcn(V_t,I,P_i,state_e_i)
%%
%{
sigma_xx(:,k)=[
    sigma_x(1,k)=soc(k-1);
    sigma_x(2,k)=Vpe(k-1);
    sigma_x(3,k)=Vpc(k-1);
]

sigma_xxx(:,i)-->
sigma_xxx(1,k)= soc(k-1);
sigma_xxx(2,k)= Vpe(k-1);
sigma_xxx(3,k)= Vpc(k-1);
sigma_z(1,k)=vt;
%}
%% Battery Parameters
R_pe=4.96e-3;
R_pc=2.86e-3;
C_pe=4.93e3;
C_pc=14.33e3;
R_in=102.5e-3;
C_n=5*3600;
%%
n=3;
n_m=2*n+1;
kappa=0;
alpha=0.6;
beta=300;
%% 7.34
lambda=alpha^2*(n+kappa)-n;
wm = ones(n_m,1)*1/(2*(n+lambda));
wc = wm;
wm(1)= lambda/(lambda+n);
wc(1)= lambda/(lambda+n)+1-alpha^2+beta;
sigma_x=zeros(3,n_m);
sigma_xx=zeros(3,n_m);
sigma_xxx=zeros(3,n_m);
sigma_z=zeros(1,n_m);
P_sqrt=chol(P_i,'lower');
%%
% Q = diag([0.02 0.1 0.2]);R=0.2;
%% 7.39 Calculate sigma point
for i=1:1:n_m
    if i==1
        sigma_x(:,i)= state_e_i;
    elseif i>=2 && i<=n+1
        sigma_x(:,i)= state_e_i+sqrt(n+lambda)*(P_sqrt(:,i-1));
    elseif i>n+1
        sigma_x(:,i)= state_e_i-sqrt(n+lambda)*(P_sqrt(:,i-1-n));
```

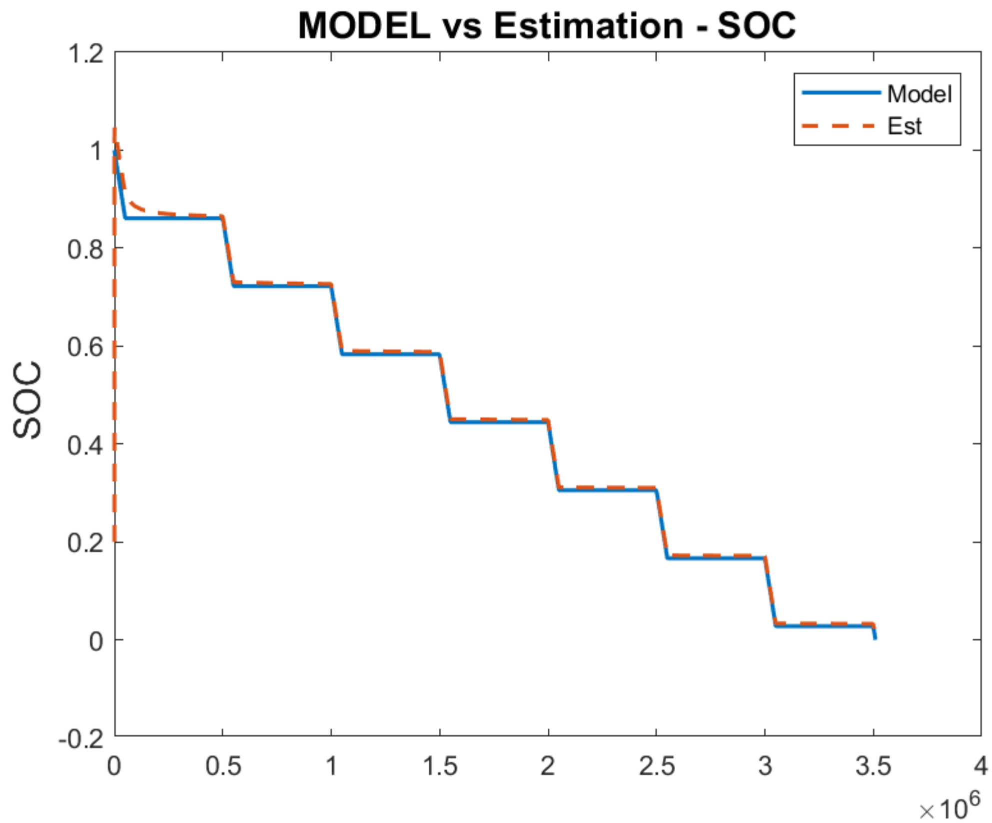
```
    end
end
%% Time Update 7.41
for k=1:1:n_m
    sigma_xx(:,k)=[sigma_x(1,k)-0.01*(I/C_n);
                   sigma_x(2,k)+0.01*((-1/(R_pe*C_pe))*sigma_x(2,k)+I/C_pe);
                   sigma_x(3,k)+0.01*((-1/(R_pc*C_pc))*sigma_x(3,k)+I/C_pe)
                   ];
end
%% 7.41
x_n=sigma_xx*wm;
P_n=zeros(3,3);
%% 7.42
for k=1:n_m
    P_n=P_n+wc(k)*(sigma_xx(:,k)-x_n)*(sigma_xx(:,k)-x_n)'  ;
end

P_sqrt=chol(P_n,'lower');
%% 7.39 Calculate sigma point
for i=1:1:n_m
    if i==1
        sigma_xxx(:,i)= x_n;
    elseif i>=2 && i<=n+1
        sigma_xxx(:,i)= x_n+sqrt(n+lambda)*(P_sqrt(:,i-1));
    elseif i>n+1
        sigma_xxx(:,i)= x_n-sqrt(n+lambda)*(P_sqrt(:,i-1-n));
    end
end

%% Time Update
for k=1:1:n_m
% V_oc=3.608-1.209*sigma_xxx(1,k)^4 + 3.055*sigma_xxx(1,k)^3 -
2.215*sigma_xxx(1,k)^2 +0.9301*sigma_xxx(1,k);
V_oc=0.55*sigma_xxx(1,k)+3.625;
sigma_z(:,k)=V_oc-sigma_xxx(2,k)-sigma_xxx(3,k)-I*R_in;
end
z_n=sigma_z*wm;
%% Measurement update equations
Pyy=10^-5;
Pxy=zeros(n,1);
%% 7.45 , 7.46
for k=1:1:n_m
    Pyy=Pyy+wc(k)*(sigma_z(:,k)-z_n)*(sigma_z(:,k)-z_n)'  ;
    Pxy=Pxy+wc(k)*(sigma_xxx(:,k)-x_n)*(sigma_z(:,k)-z_n)';
end
%% 7.47-49
K=Pxy/Pyy;
State_e_o=x_n+K*(V_t-z_n);
P_o=P_n-K*Pyy*K';
```

مدل در متلب



<= 0
Compare To Constant
STOP
Stop Simulation
Vt
To Workspace3
SOC_Model
To Workspace4
Voc
To Workspace1
SOC_EST
To Workspace2
MATLAB Function
MATLAB Function1
Integrator2
Pulse Generator
Scope
Scope10
Unit Delay
Unit Delay1
Scope9

نتایج شبیه‌سازی



نتایج نشان می‌دهد که الگوریتم **UKF** به خوبی مدل را تخمین زده است.

مراجع

[۱] X. Chen, W. Shen, Z. Cao, and A. Kapoor, "A novel approach for state of charge estimation based on adaptive switching gain sliding mode observer in electric vehicles," *Journal of Power Sources*, vol. 246, pp. 667-678, 2014.