HW 4

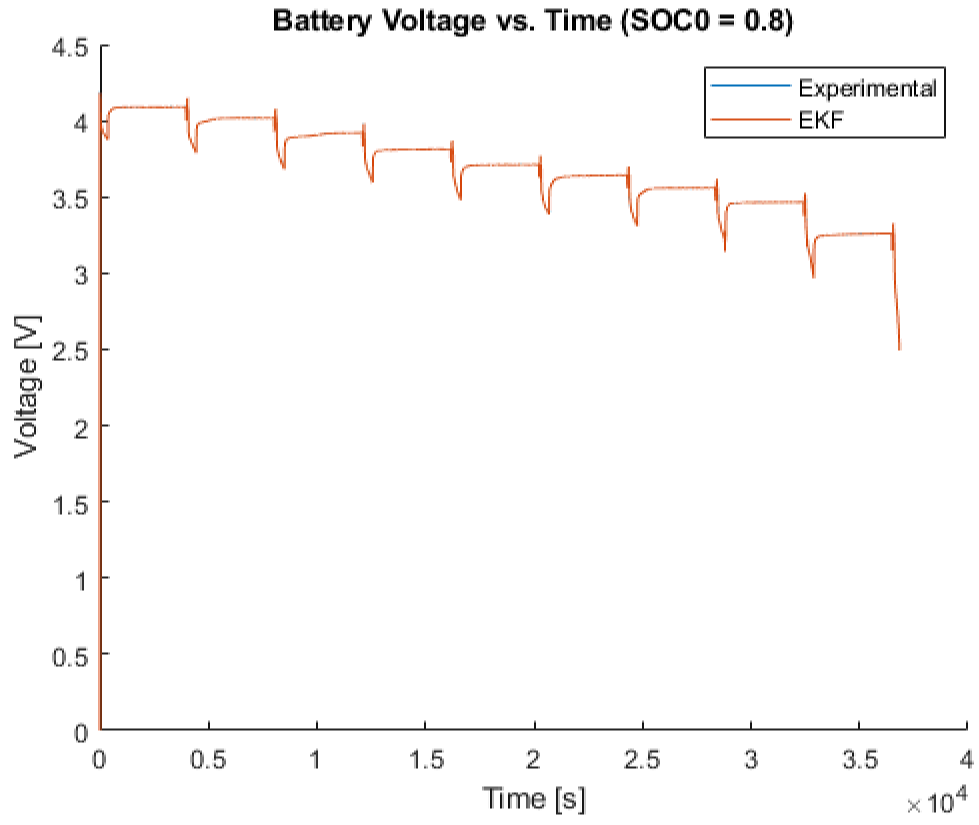
```
% Authors: Jessica, Tom, Miles
% ENERGY295
% 11/23/2021
```

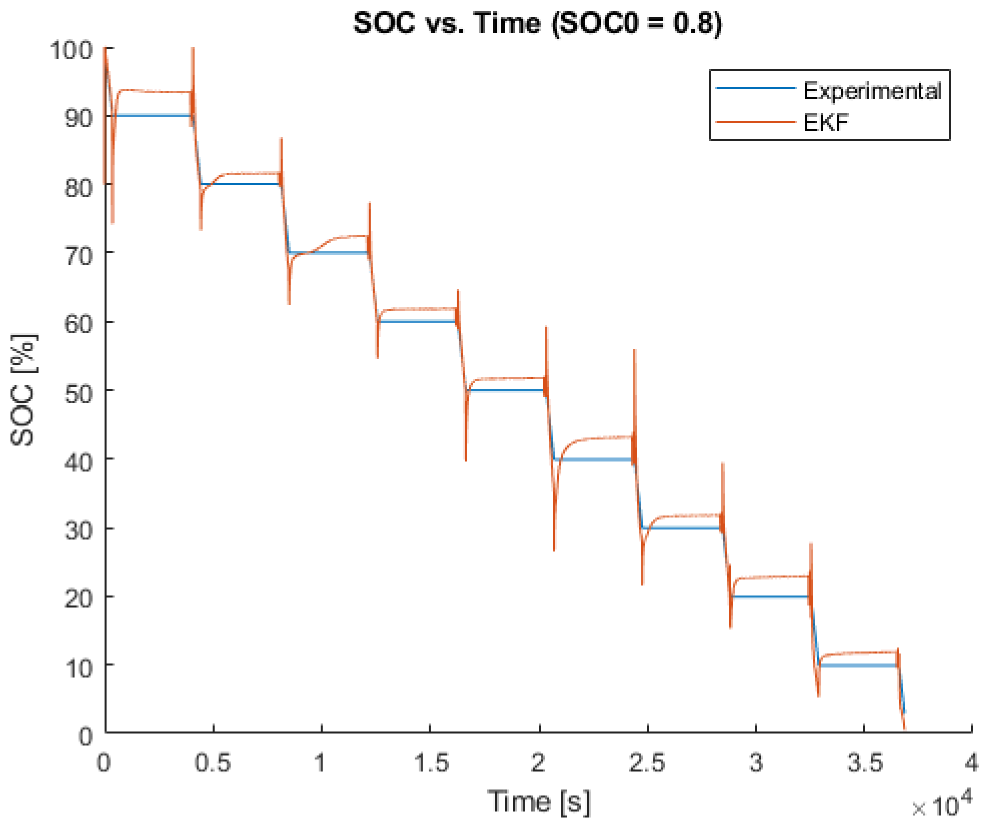
Problem 1 and 2

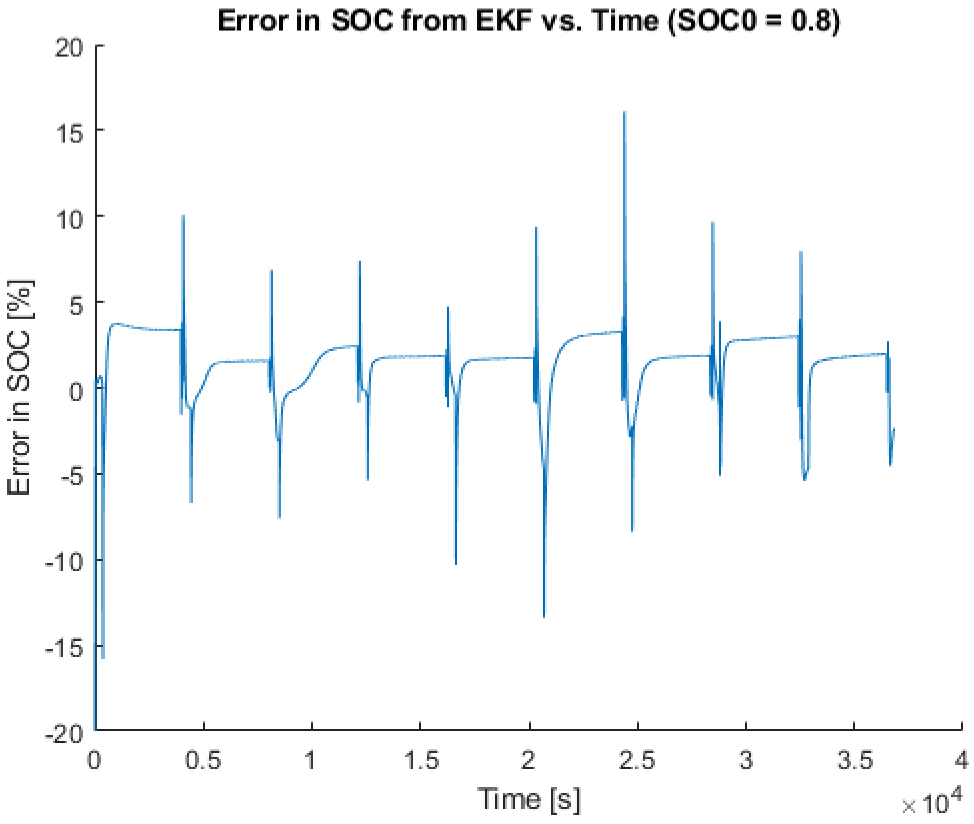
```
clear all; close all; format compact; clc

load('HW4_batt_params.mat');
load('HW4_expt_data.mat');

N = length(V_expt);
W = 10;
t = [0:dt:(N-1)]';

% Initialize
P_t0 = diag([0.1,0.001,0.001]);
P_t = P_t0;
R = 8.432e-4;
Q = diag([1000*R, 0.1*R, 0.01*R]);
disp('Q,R, and P values used...')
disp('Q = ');
disp(Q);
disp('R = ');
disp(R);
disp('P = ');
disp(P_t0);

deltaSOC = 0.01;
SOC_CC = SOC_init - (cumtrapz(t, I_expt)/3600)/capacity;
SOC0s = [0.8,0.4,0.25];
for i = 1:1:length(SOC0s)
    SOC0 = SOC0s(i);
    soc_ekf = zeros(N,1);
    soc_aekf = zeros(N,1);
    for adaptive = [false, true]
        x_t = [SOC0;0;0];
        [x_t,Vb,L] =
 EKF(dt,V_expt,I_expt,x_t,P_t,Q,R,R0_chg,R0_dischg,R0_dischg_4A,...

 R1_chg,R1_dischg,C1_chg,C1_dischg,R2_chg,R2_dischg,C2_chg,C2_dischg,...

 soc_chg,soc_dischg,capacity,Voc_vs_SOC,deltaSOC,adaptive,W);
        if adaptive
            EKF_type = 'A'; %for adaptive
            soc_aekf = x_t(1,:);
        else
            EKF_type = ''; %for just ekf
            soc_ekf = x_t(1,:);
        end

        figure(); set(gcf,'color','w'); hold on;
        plot(t,V_expt);
        plot(t,Vb);
        xlabel('Time [s]'); ylabel('Voltage [V]');
        title(['Battery Voltage vs. Time (SOC0 = ' num2str(SOC0) ')']);
        legend('Experimental',[EKF_type 'EKF']);
        percent_rmse_Vb = calc_percent_rmse(V_expt,Vb);
        disp(['RMS Error in Vb (' EKF_type 'EKF with SOC0 = ' num2str(SOC0) ')
 = ' num2str(percent_rmse_Vb) '%']);

        figure(); set(gcf,'color','w'); hold on;
        plot(t,SOC_CC*100);
        plot(t,x_t(1,:)*100);
        xlabel('Time [s]'); ylabel('SOC [%]');
        title(['SOC vs. Time (SOC0 = ' num2str(SOC0) ')']);
        legend('Experimental',[EKF_type 'EKF']);

        figure(); set(gcf,'color','w'); hold on;
        plot(t,(x_t(1,:) - SOC_CC')*100);
        xlabel('Time [s]'); ylabel('Error in SOC [%]');
        title(['Error in SOC from ' EKF_type 'EKF vs. Time (SOC0 = '
 num2str(SOC0) ')']);
        percent_rmse_SOC = calc_percent_rmse(SOC_CC,x_t(1,:));
        disp(['RMS Error in SOC (' EKF_type 'EKF with SOC0 = '
 num2str(SOC0) ') = ' num2str(percent_rmse_SOC) '%']);

        figure();set(gcf,'color','w');
        subplot(3,1,1);
        plot(t,L(1,:));
        xlabel('Time [s]');ylabel([EKF_type 'EKF Gain on SOC [%/V]']);
        title([EKF_type 'EKF Gain vs. Time (SOC0 = ' num2str(SOC0) ')']);
        subplot(3,1,2);
        plot(t,L(2,:));
        xlabel('Time [s]');ylabel([EKF_type 'EKF Gain on V1 [V/V]']);
        title([EKF_type 'EKF Gain vs. Time (SOC0 = ' num2str(SOC0) ')']);
        subplot(3,1,3);
        plot(t,L(3,:));
        xlabel('Time [s]');ylabel([EKF_type 'EKF Gain on V2 [V/V]']);
        title([EKF_type 'EKF Gain vs. Time (SOC0 = ' num2str(SOC0) ')']);
    end
    figure(); set(gcf,'color','w'); hold on;
    plot(t,SOC_CC*100);
    plot(t,soc_ekf*100);
    plot(t,soc_aekf*100);
    xlabel('Time [s]'); ylabel('SOC [%]');
    title(['SOC vs. Time (SOC0 = ' num2str(SOC0) ')']);
    legend('Experimental','EKF','AEKF');
end
```

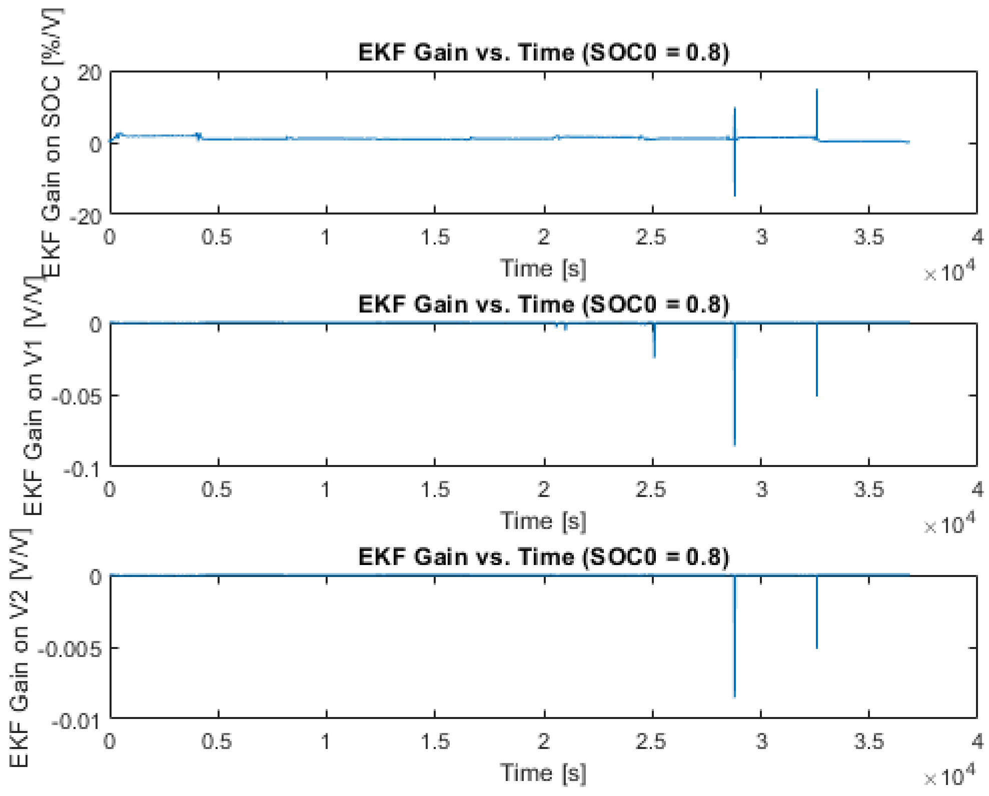
```
function result = derivative(x,y,x_point,delta_x)
    y1 = interp1(x, y, x_point-delta_x, 'linear','extrap');
    y2 = interp1(x, y, x_point+delta_x, 'linear','extrap');
    result = (y2-y1)/(2*delta_x);
end

function y = saturate(x,lb,ub)
    if x < lb
        y = lb;
    elseif x > ub
        y = ub;
    else
        y = x;
    end
end

function prmse = calc_percent_rmse(x_actual,x_corrupted)
    N = length(x_actual);
    prmse = sqrt((1/N)*sum((reshape(x_corrupted,[],1) - reshape(x_actual,
[],1)).^2))*(100*N/sum(x_actual));
end

function [x_t,Vb,L] =
 EKF(dt,V_expt,I_expt,x_t,P_t,Q,R,R0_chg,R0_dischg,R0_dischg_4A,...

 R1_chg,R1_dischg,C1_chg,C1_dischg,R2_chg,R2_dischg,C2_chg,C2_dischg,...
        soc_chg,soc_dischg,capacity,Voc_vs_SOC,deltaSOC,adaptive,W)
    N = length(V_expt);
    for i = 2:N
         % Predict
        if I_expt(i-1) < 0
            R0 = interp1(soc_chg, R0_chg, x_t(1,i-1), 'linear','extrap');
            R1 = interp1(soc_chg, R1_chg, x_t(1,i-1), 'linear','extrap');
            R2 = interp1(soc_chg, R2_chg, x_t(1,i-1), 'linear','extrap');
            C1 = interp1(soc_chg, C1_chg, x_t(1,i-1), 'linear','extrap');
            C2 = interp1(soc_chg, C2_chg, x_t(1,i-1), 'linear','extrap');
            dR0 = derivative(soc_chg, R0_chg, x_t(1,i-1),deltaSOC);
            dR1 = derivative(soc_chg, R1_chg, x_t(1,i-1),deltaSOC);
            dR2 = derivative(soc_chg, R2_chg, x_t(1,i-1),deltaSOC);
            dC1 = derivative(soc_chg, C1_chg, x_t(1,i-1),deltaSOC);
            dC2 = derivative(soc_chg, C2_chg, x_t(1,i-1),deltaSOC);
        else
            if I_expt(i-1)>=4
                R0 = interp1(soc_chg, R0_dischg_4A,
 x_t(1,i-1), 'linear','extrap');
                dR0 = derivative(soc_dischg, R0_dischg_4A,
 x_t(1,i-1),deltaSOC);
            else
                R0 = interp1(soc_chg, R0_dischg,
 x_t(1,i-1), 'linear','extrap');
                dR0 = derivative(soc_dischg, R0_dischg, x_t(1,i-1),deltaSOC);
            end
            R1 = interp1(soc_chg, R1_dischg, x_t(1,i-1), 'linear','extrap');
            R2 = interp1(soc_chg, R2_dischg, x_t(1,i-1), 'linear','extrap');
            C1 = interp1(soc_chg, C1_dischg, x_t(1,i-1), 'linear','extrap');
            C2 = interp1(soc_chg, C2_dischg, x_t(1,i-1), 'linear','extrap');
            dR1 = derivative(soc_dischg, R1_dischg, x_t(1,i-1),deltaSOC);
            dR2 = derivative(soc_dischg, R2_dischg, x_t(1,i-1),deltaSOC);
            dC1 = derivative(soc_dischg, C1_dischg, x_t(1,i-1),deltaSOC);
            dC2 = derivative(soc_dischg, C2_dischg, x_t(1,i-1),deltaSOC);
        end


        A = [0, 0, 0; ...
             ( x_t(2,i-1)*(R1*dC1 + C1*dR1)/(R1*C1)^2 - I_expt(i-1)*dC1/
C1^2 ), ( -1/(R1*C1) ), 0;
             ( x_t(3,i-1)*(R2*dC2 + C2*dR2)/(R2*C2)^2 - I_expt(i-1)*dC2/
C2^2 ), 0,              ( -1/(R2*C2) )];
        B = [-1/(3600*capacity); 1/C1; 1/C2];

        x_tp(:,i) = x_t(:,i-1) + dt*[-I_expt(i-1)/(3600*capacity);...
                                    -x_t(2,i-1)/(R1*C1) + I_expt(i-1)/C1;...
                                    -x_t(3,i-1)/(R2*C2) + I_expt(i-1)/C2];
        x_tp(1,i) = saturate(x_tp(1,i),0,1);
        P_tp = A*P_t*A' + Q;

        % Correct
        if I_expt(i-1) < 0
            dR0 = derivative(soc_chg, R0_chg, x_t(1,i-1),deltaSOC);
        else
            if I_expt(i-1)>=4
                dR0 = derivative(soc_dischg, R0_dischg_4A,
 x_t(1,i-1),deltaSOC);
            else
                dR0 = derivative(soc_dischg, R0_dischg, x_t(1,i-1),deltaSOC);
            end
        end
        dV_dSOC = derivative(Voc_vs_SOC(:,1), Voc_vs_SOC(:,2), ...
                                    x_tp(1,i),deltaSOC);
        C = [(dV_dSOC - I_expt(i)*dR0), -1, -1];

        L(:,i) = P_tp * C' * inv(C*P_tp*C' + R);
        Voc = interp1(Voc_vs_SOC(:,1), Voc_vs_SOC(:,2),
 x_tp(1,i),'linear','extrap');
        V = Voc - x_tp(2,i) - x_tp(3,i) - I_expt(i)*R0;
        x_t(:,i) = x_tp(:,i) + L(:,i)*(V_expt(i) - V);
        x_t(1,i) = saturate(x_t(1,i),0,1);
        P_t = (eye(length(A)) - L(:,i)*C)*P_tp;

        Voc = interp1(Voc_vs_SOC(:,1), Voc_vs_SOC(:,2),
 x_t(1,i), 'linear','extrap');
        Vb(i) = Voc - x_t(2,i) - x_t(3,i) - I_expt(i)*R0;

        if adaptive
            % Adapt
            d(i) = V_expt(i)-V;
            if length(Vb) < W
                D = 1/W*sum(d*d');
                delta_x = x_t-x_tp;
                L_adapt = delta_x/d;
                Q = L_adapt*D*L_adapt';
            else
                D = 1/W*sum( d(i-W+1:i) * d(i-W+1:i)' );
                delta_x = x_t(:,i-W+1:i) - x_tp(:,i-W+1:i);
                L_adapt = delta_x/d(i-W+1:i);
                Q = L_adapt*D*L_adapt';
            end
        end
    end
end
```

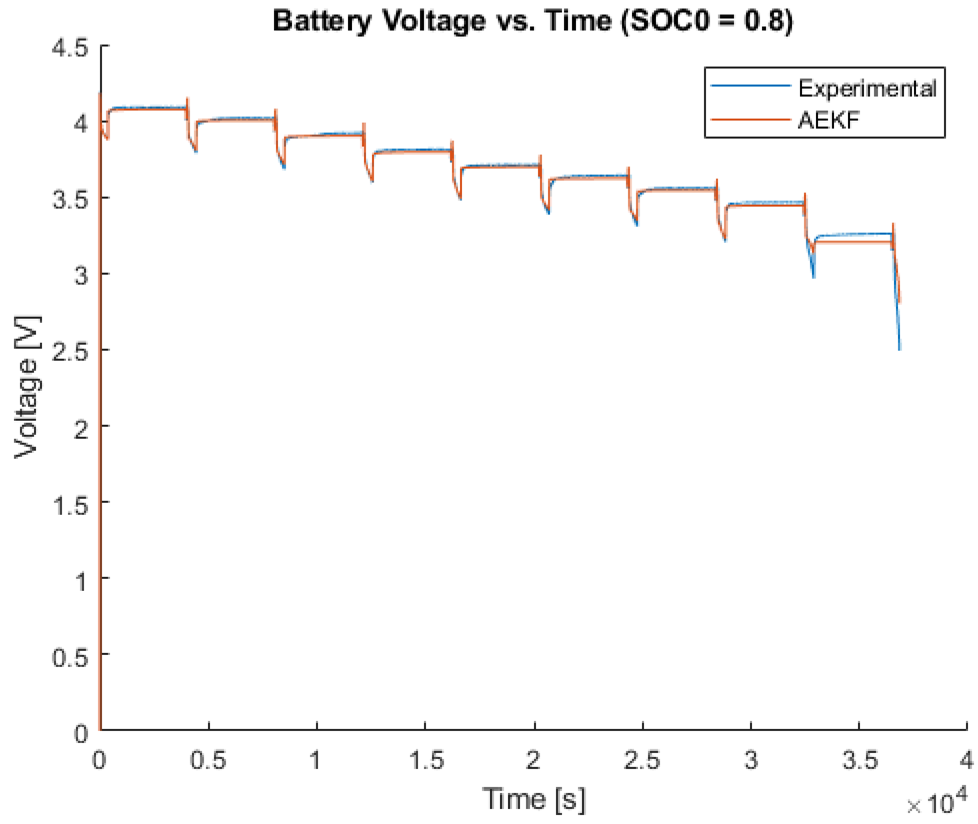
```
Q,R, and P values used...
Q =
    0.8432         0         0
         0    0.0001         0
         0         0    0.0000
R =
   8.4320e-04
P =
    0.1000         0         0
         0    0.0010         0
         0         0    0.0010
RMS Error in Vb (EKF with SOC0 = 0.8) = 0.59128%
RMS Error in SOC (EKF with SOC0 = 0.8) = 4.6634%
RMS Error in Vb (AEKF with SOC0 = 0.8) = 1.0284%
RMS Error in SOC (AEKF with SOC0 = 0.8) = 0.52541%
RMS Error in Vb (EKF with SOC0 = 0.4) = 0.59118%
RMS Error in SOC (EKF with SOC0 = 0.4) = 4.7001%
RMS Error in Vb (AEKF with SOC0 = 0.4) = 1.0279%
RMS Error in SOC (AEKF with SOC0 = 0.4) = 0.794%
RMS Error in Vb (EKF with SOC0 = 0.25) = 0.59137%
RMS Error in SOC (EKF with SOC0 = 0.25) = 4.7241%
RMS Error in Vb (AEKF with SOC0 = 0.25) = 1.0281%
RMS Error in SOC (AEKF with SOC0 = 0.25) = 0.92426%
```

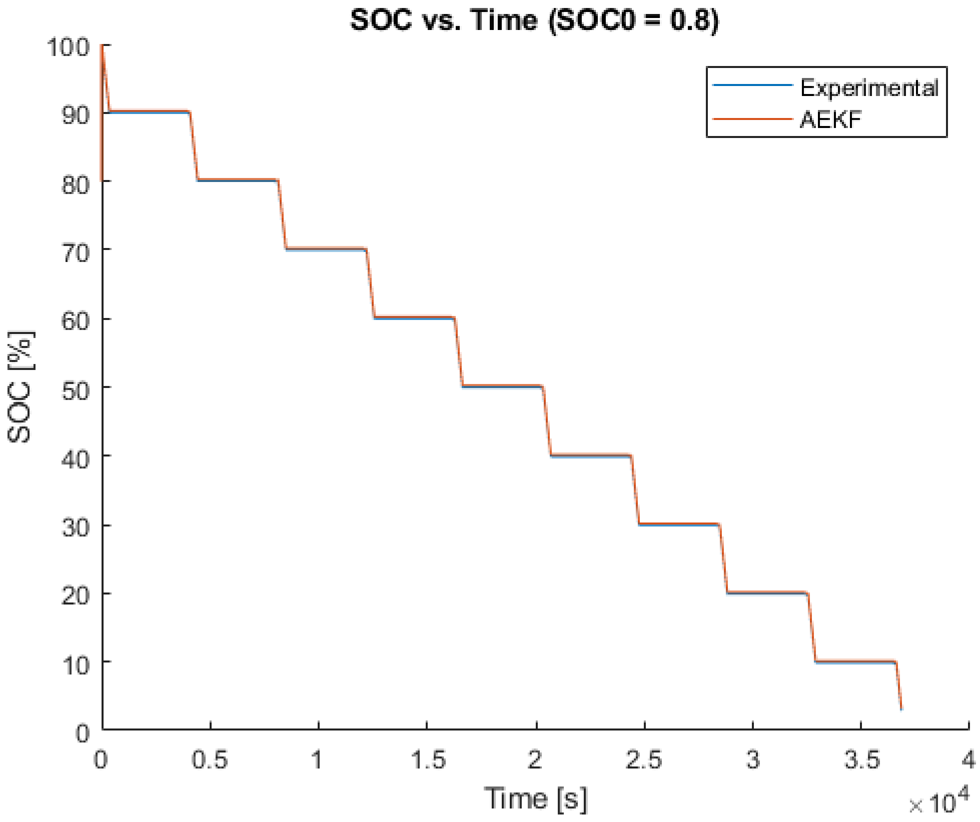

Battery Voltage vs. Time (SOC0 = 0.8)
Voltage [V]
Time [s]
×10^4
Experimental
EKF
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

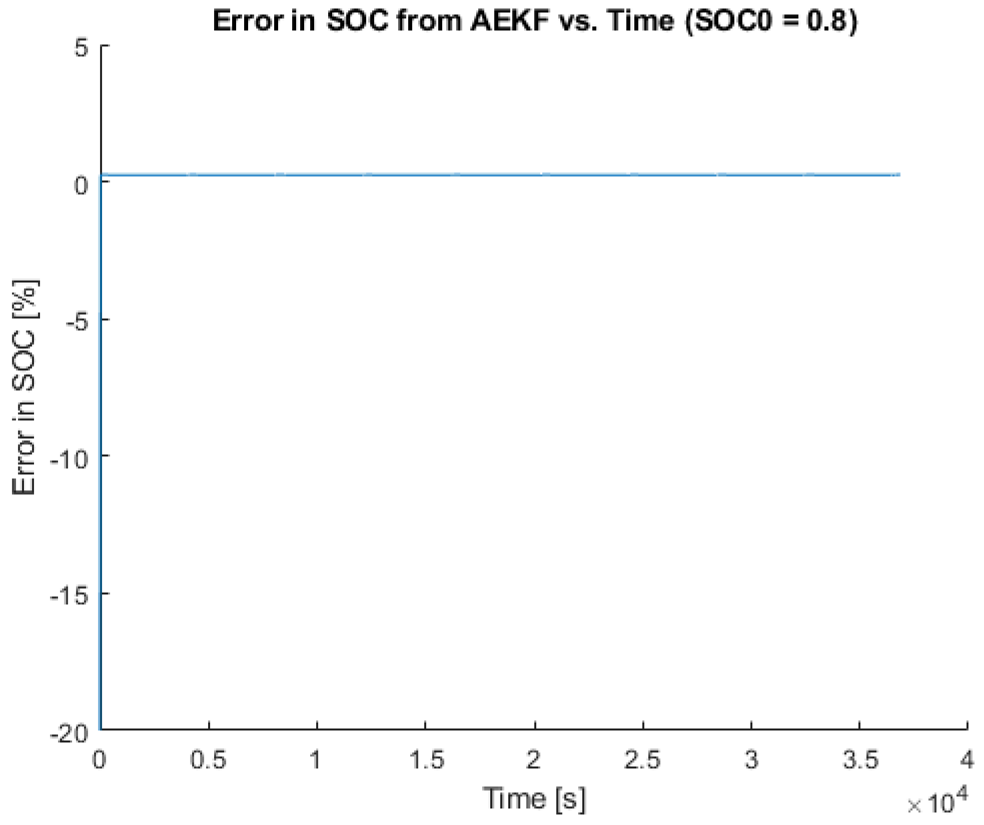

SOC vs. Time (SOC0 = 0.8)
SOC [%]
Time [s]
×10^4
Experimental
EKF
0
10
20
30
40
50
60
70
80
90
100

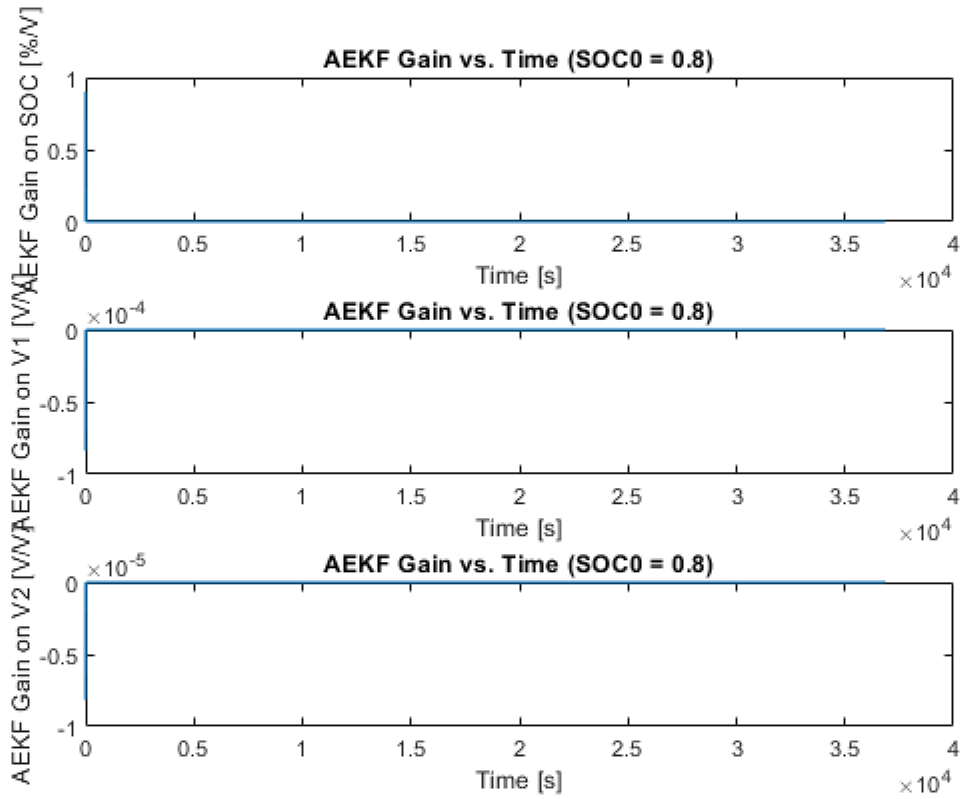

Error in SOC from EKF vs. Time (SOC0 = 0.8)
Error in SOC [%]
20
15
10
5
0
-5
-10
-15
-20
0
0.5
1
1.5
2
2.5
3
3.5
4
Time [s]
×10^4

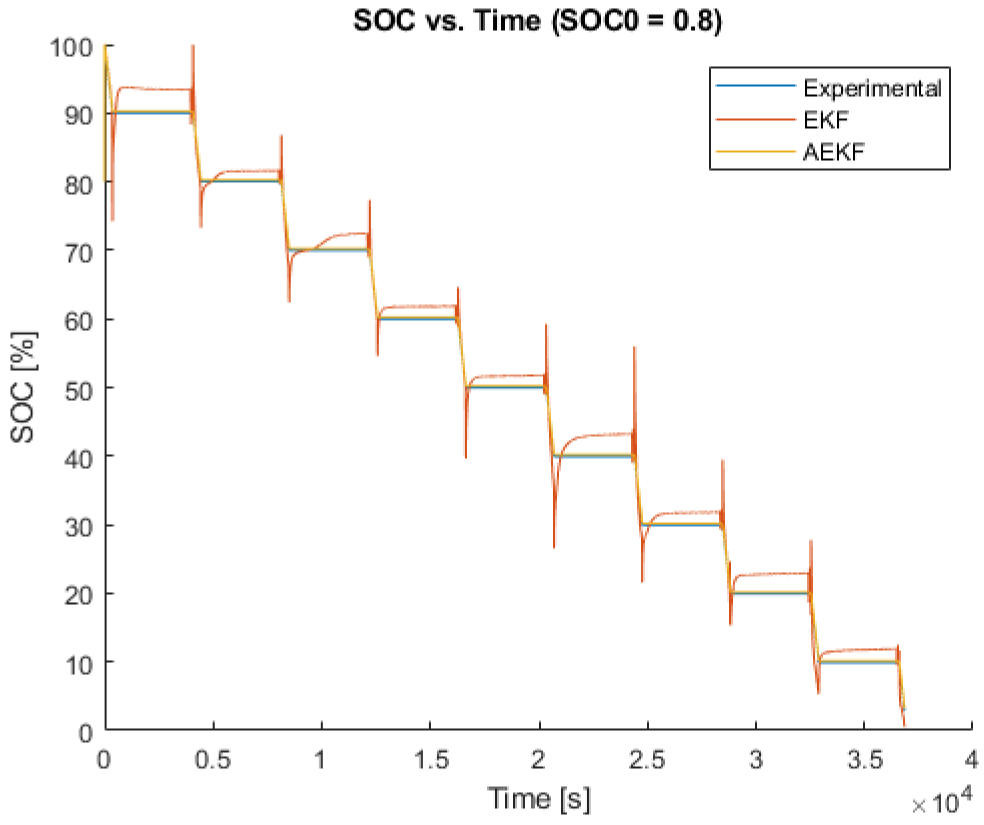

EKF Gain vs. Time (SOC0 = 0.8)
EKF Gain on SOC [%/V]
20
0
-20
Time [s]
×10^4
EKF Gain vs. Time (SOC0 = 0.8)
EKF Gain on V1 [V/V]
0
-0.05
-0.1
Time [s]
×10^4
EKF Gain vs. Time (SOC0 = 0.8)
EKF Gain on V2 [V/V]
0
-0.005
-0.01
0
0.5
1
1.5
2
2.5
3
3.5
4
Time [s]
×10^4

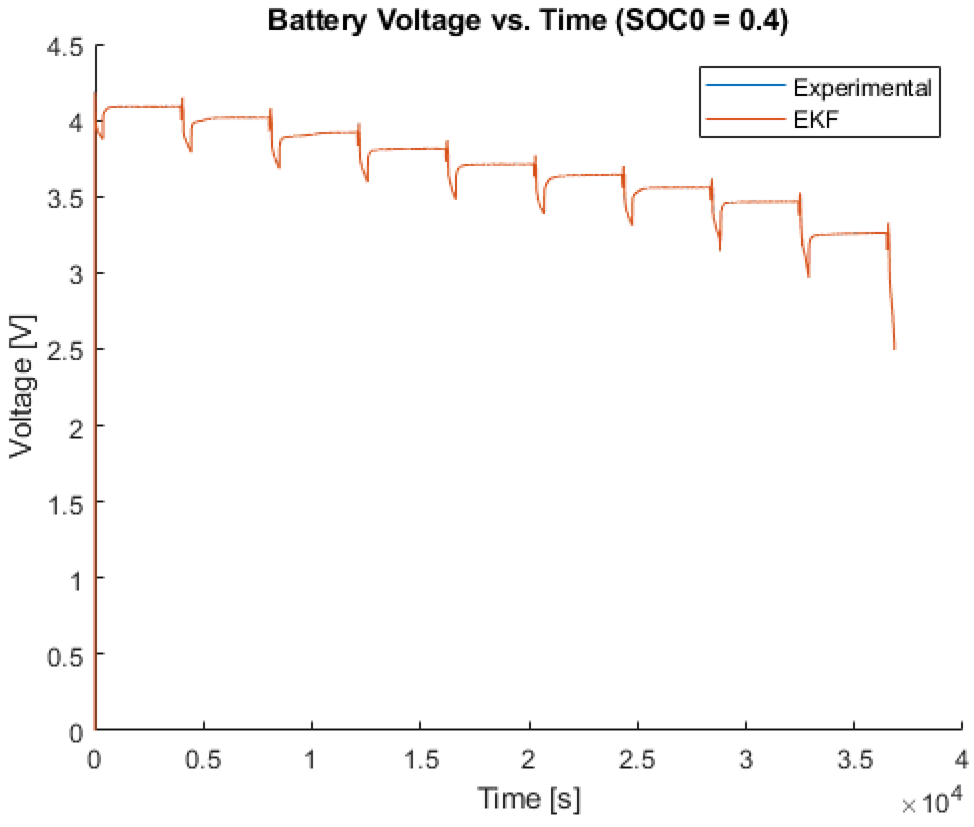

Battery Voltage vs. Time (SOC0 = 0.8)
Experimental
AEKF
Voltage [V]
Time [s]
×10^4
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

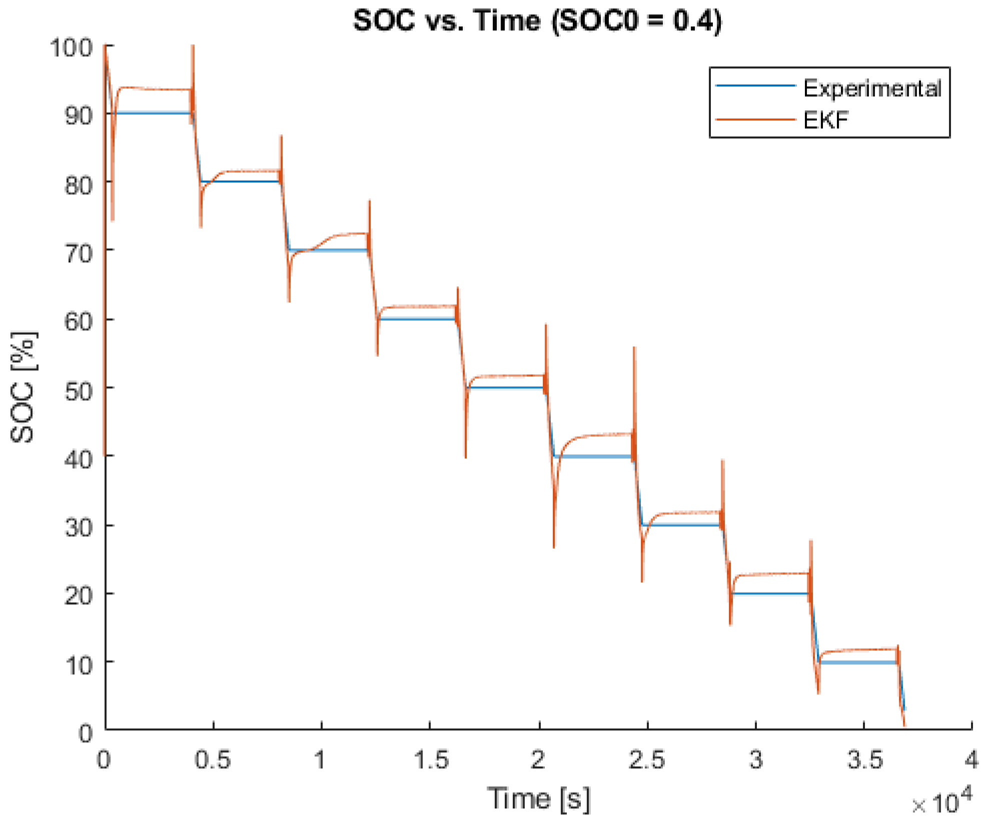

SOC vs. Time (SOC0 = 0.8)
Experimental
AEKF
SOC [%]
Time [s]
×10^4
0
10
20
30
40
50
60
70
80
90
100

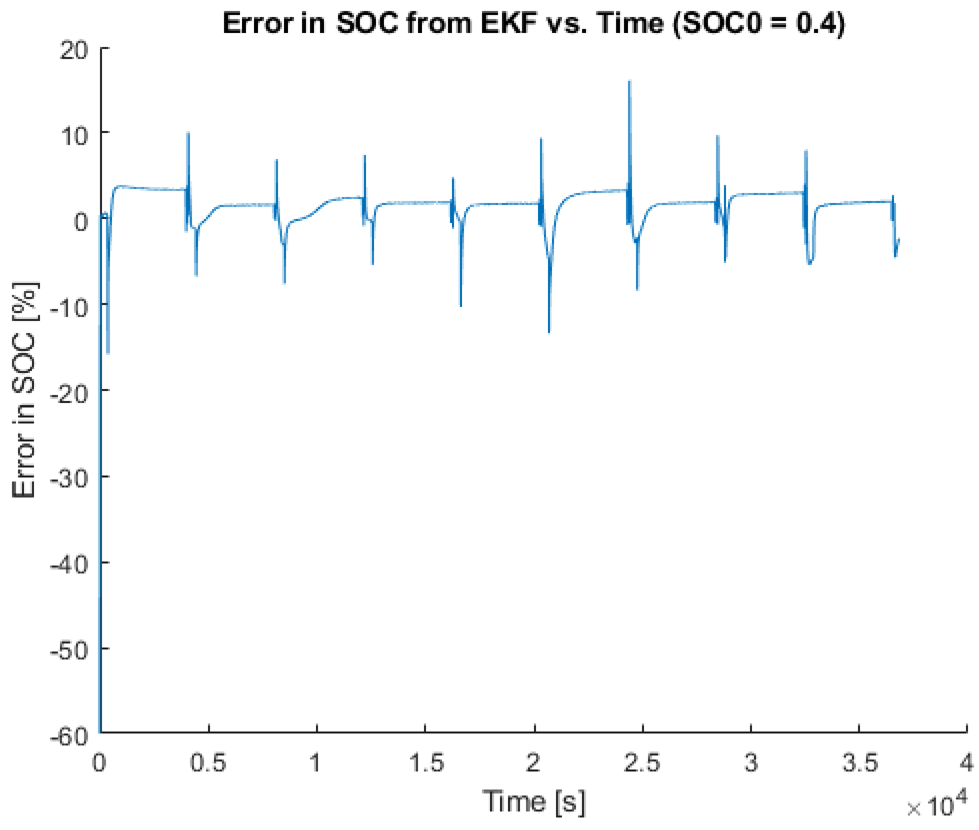

Error in SOC from AEKF vs. Time (SOC0 = 0.8)
Error in SOC [%]
Time [s]
×10^4
5
0
-5
-10
-15
-20
0
0.5
1
1.5
2
2.5
3
3.5
4

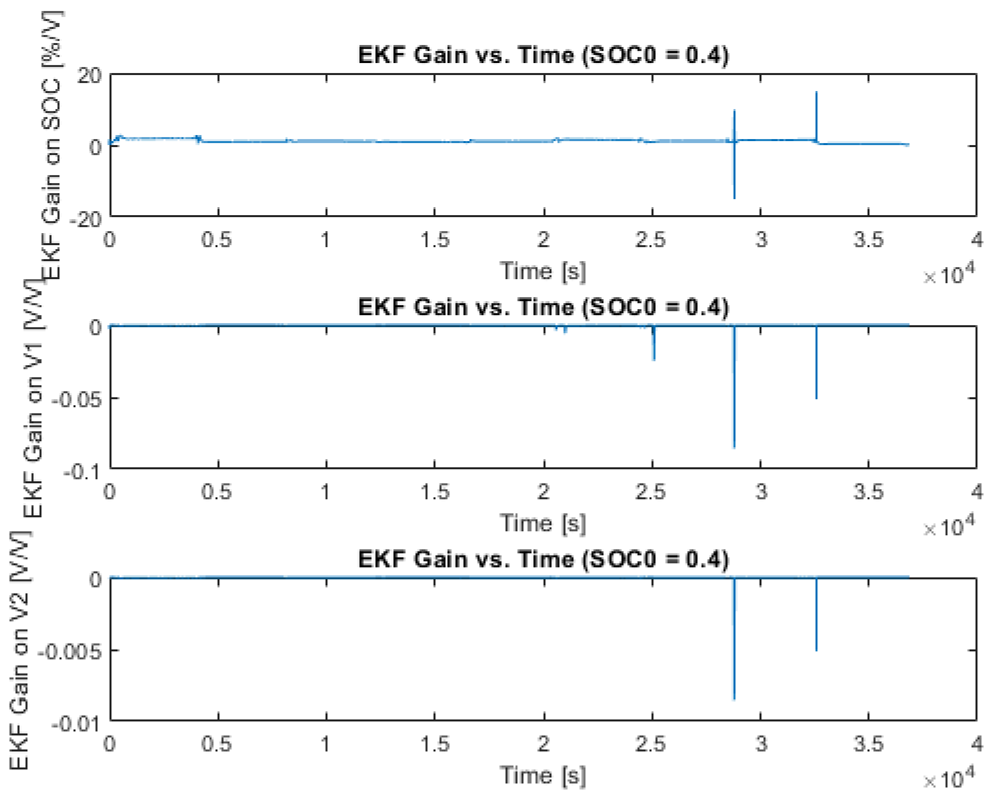

AEKF Gain vs. Time (SOC0 = 0.8)
AEKF Gain on SOC [%/V]
AEKF Gain on V1 [V/V]
AEKF Gain on V2 [V/V]
Time [s]
×10^4
×10^-4
×10^-5
1
0.5
0
-0.5
-1

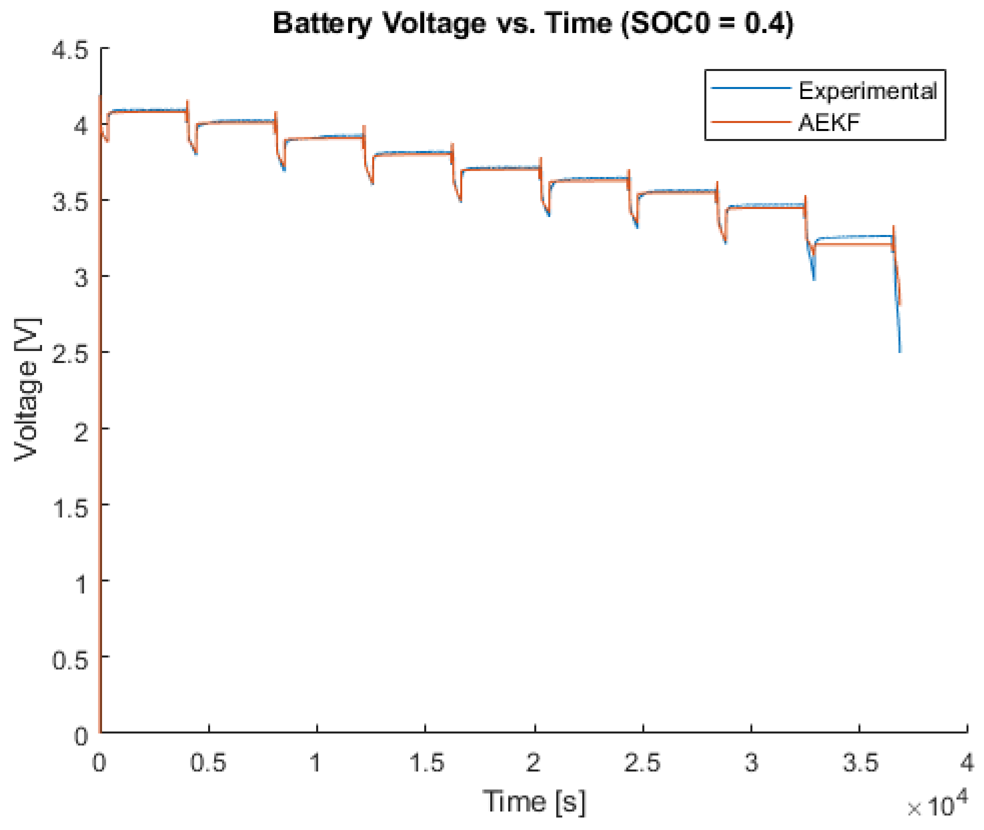

SOC vs. Time (SOC0 = 0.8)
SOC [%]
Time [s]
×10^4
Experimental
EKF
AEKF

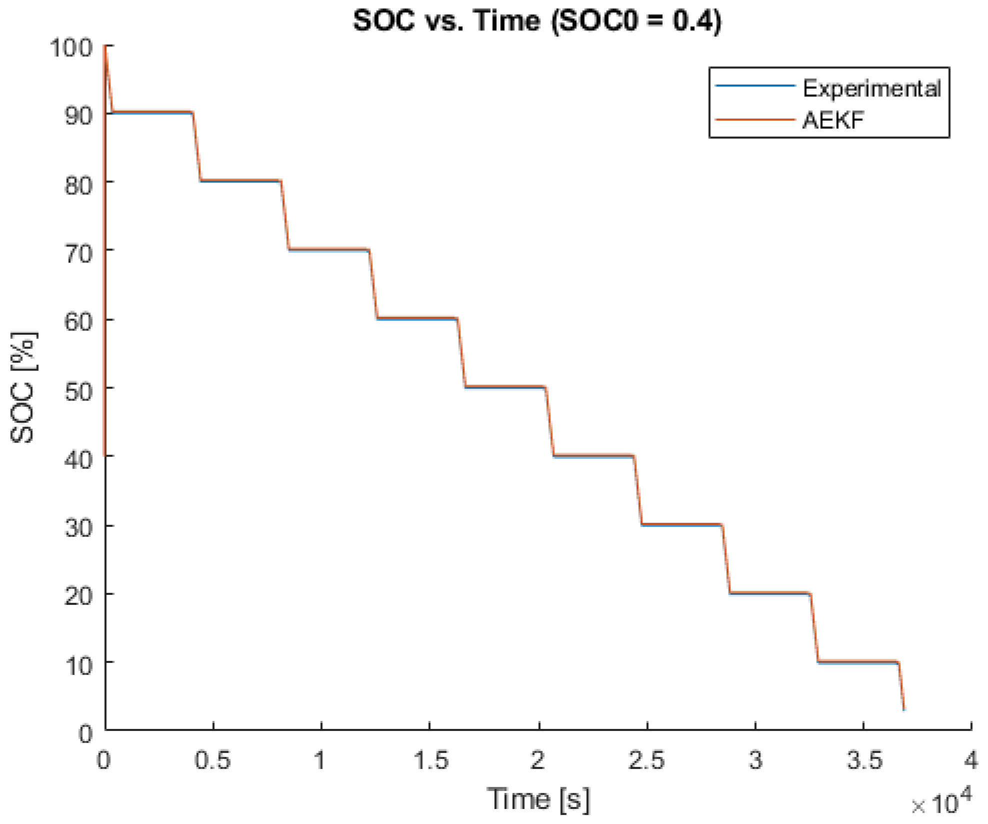

Battery Voltage vs. Time (SOC0 = 0.4)
Voltage [V]
Time [s]
×10^4
Experimental
EKF

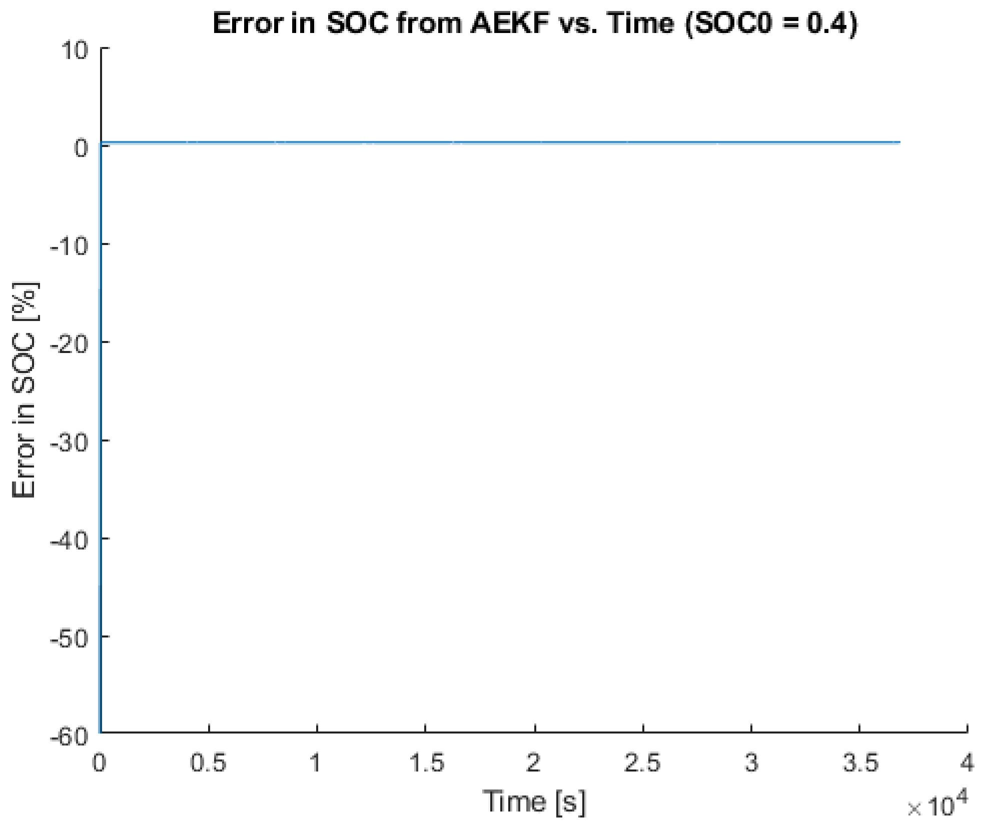

SOC vs. Time (SOC0 = 0.4)
Experimental
EKF
SOC [%]
Time [s]
×10^4

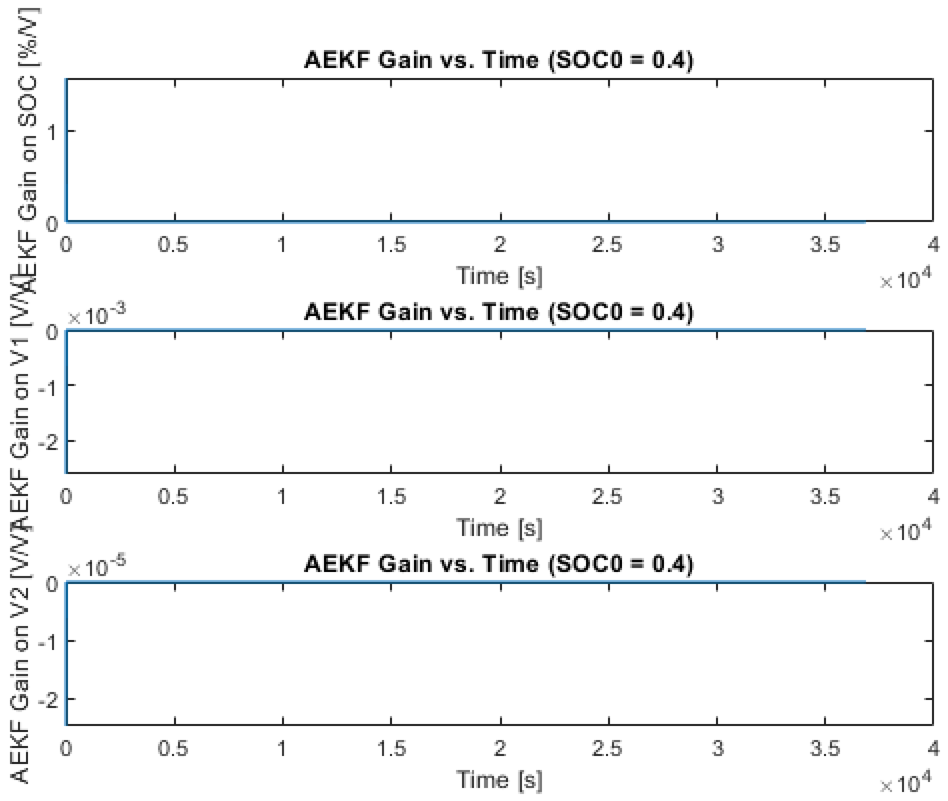

Error in SOC from EKF vs. Time (SOC0 = 0.4)
Error in SOC [%]
Time [s]
×10^4

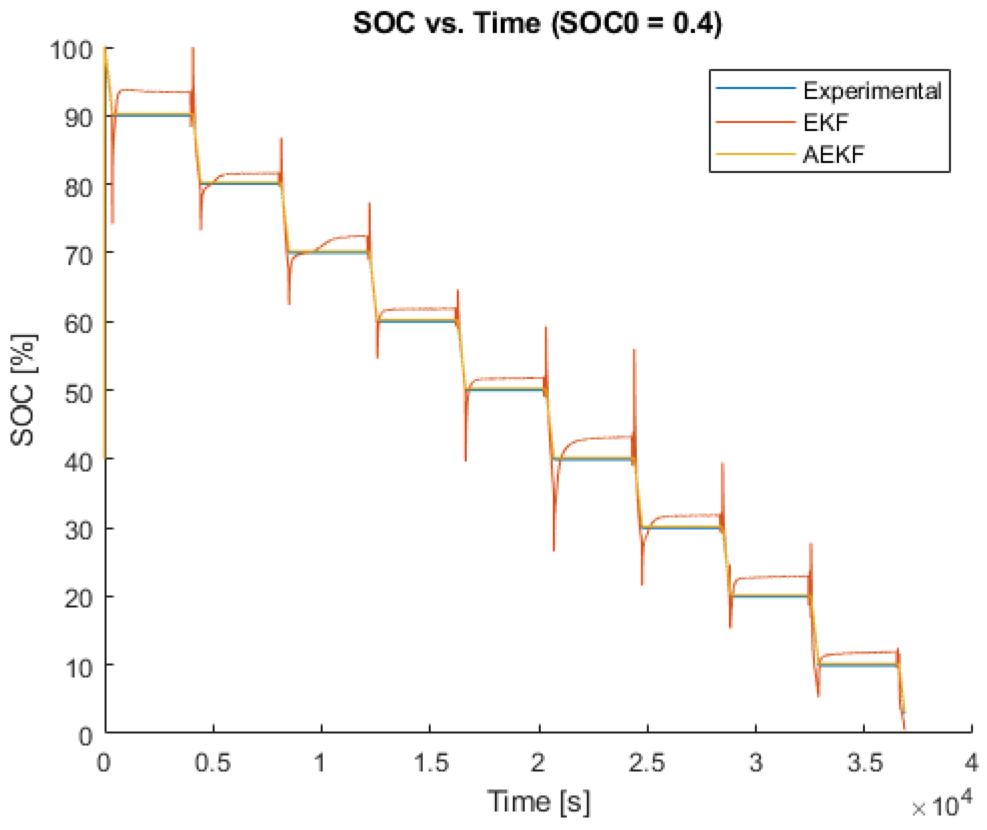

EKF Gain vs. Time (SOC0 = 0.4)
EKF Gain on SOC [%/V]
EKF Gain on V1 [V/V]
EKF Gain on V2 [V/V]
Time [s]
×10^4

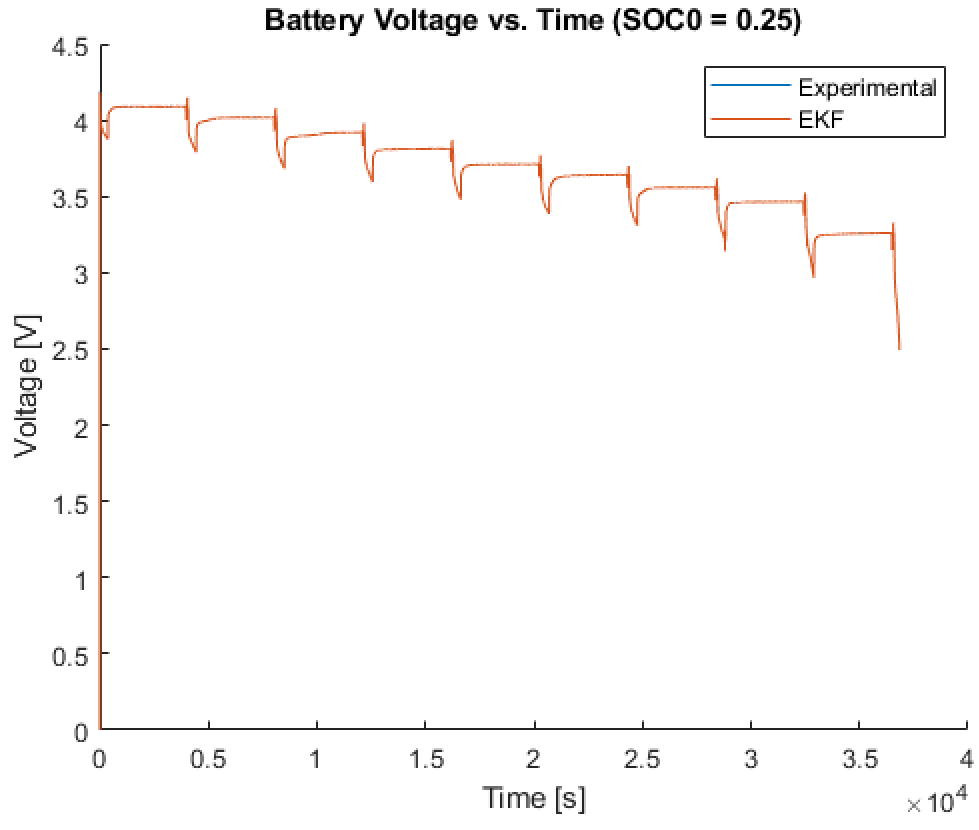

Battery Voltage vs. Time (SOC0 = 0.4)
Voltage [V]
Time [s]
×10^4
Experimental
AEKF

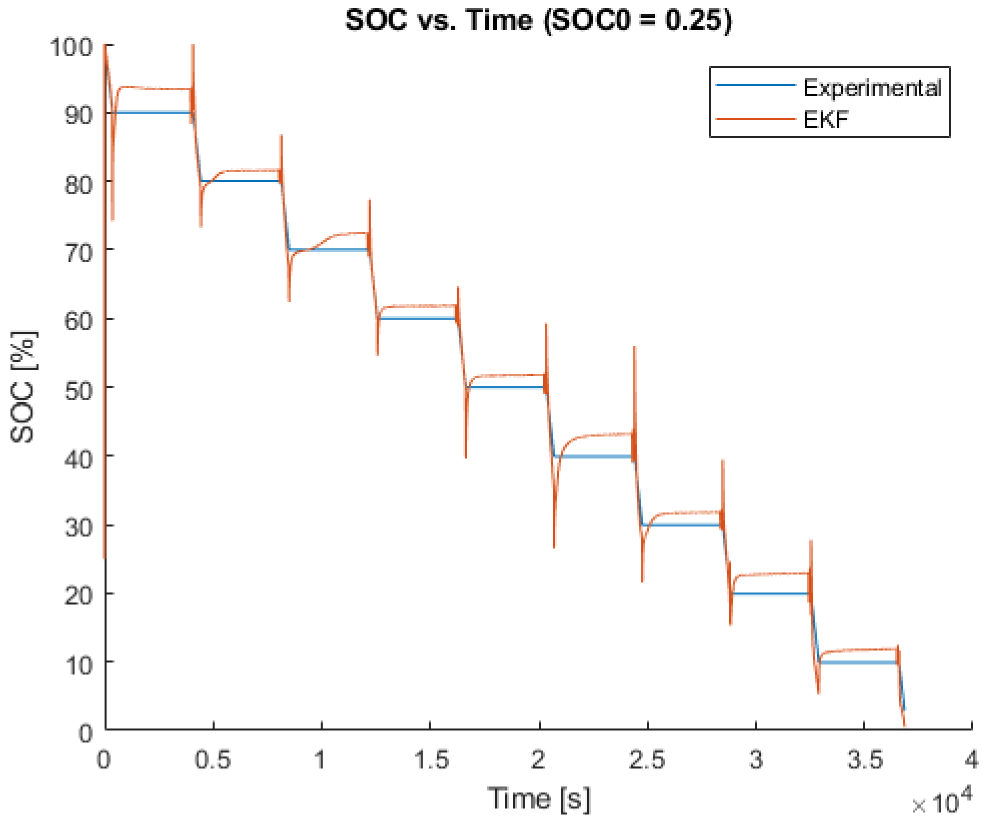

SOC vs. Time (SOC0 = 0.4)
Experimental
AEKF
SOC [%]
Time [s]
×10^4

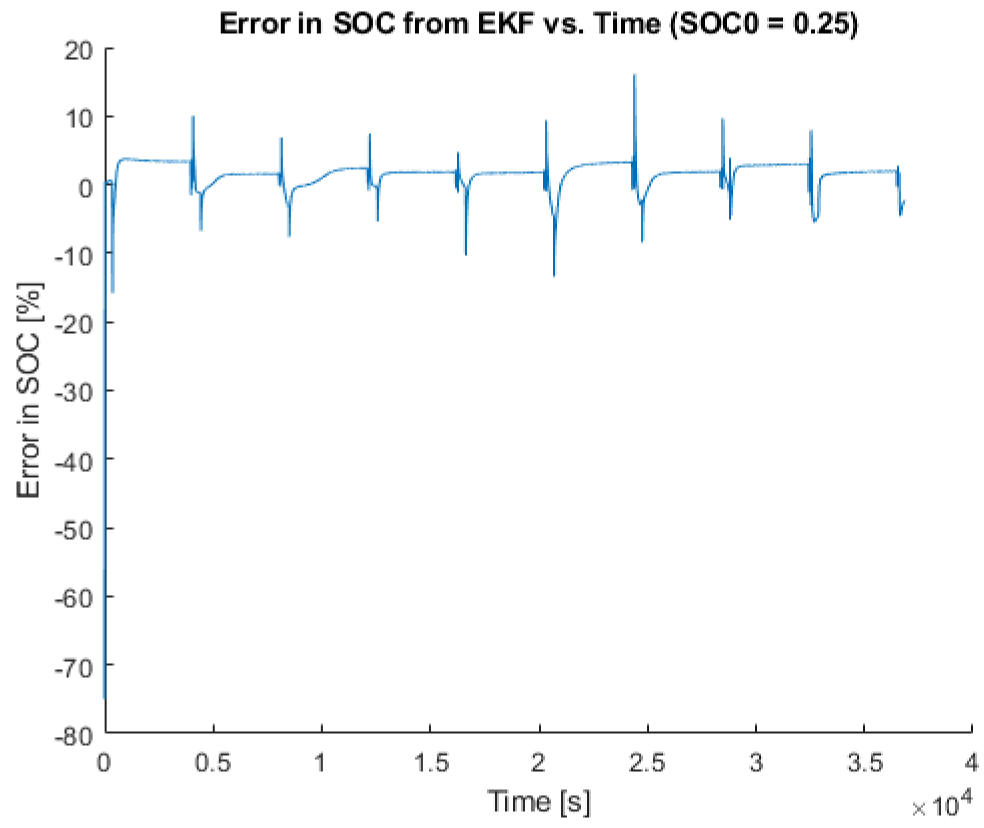

Error in SOC from AEKF vs. Time (SOC0 = 0.4)
Error in SOC [%]
Time [s]
×10^4


AEKF Gain vs. Time (SOC0 = 0.4)
AEKF Gain on SOC [%/V]
AEKF Gain on V1 [V/V]
×10^-3
AEKF Gain on V2 [V/V]
×10^-5
Time [s]
×10^4

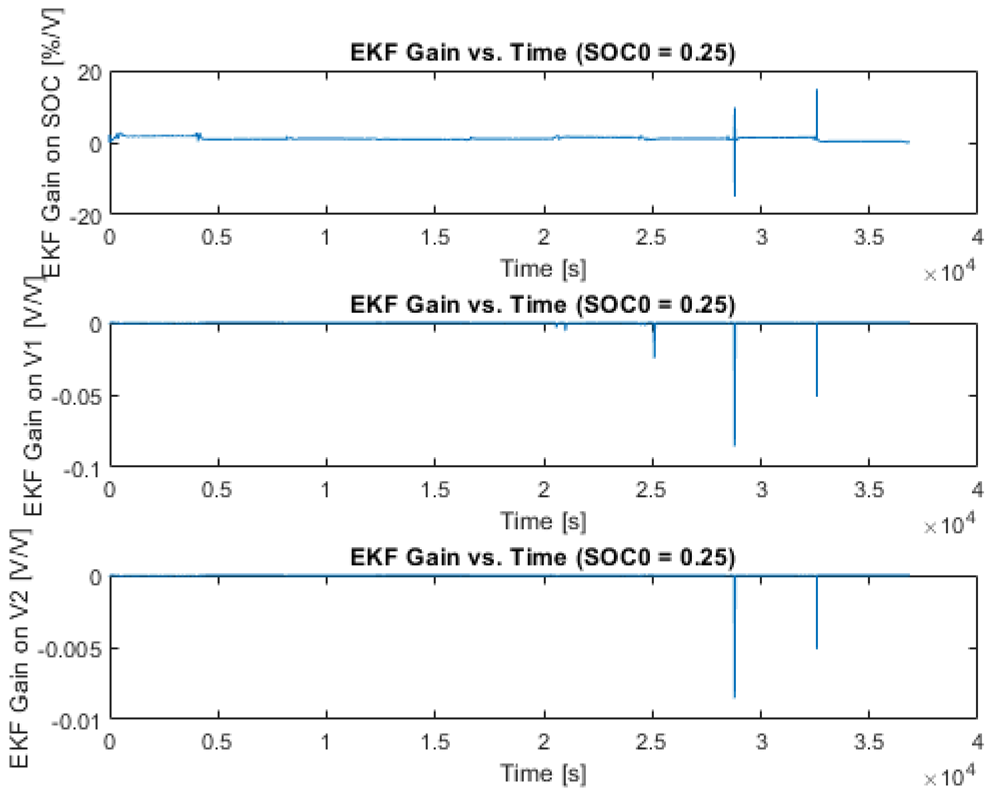

SOC vs. Time (SOC0 = 0.4)
SOC [%]
Time [s]
×10^4
Experimental
EKF
AEKF


Battery Voltage vs. Time (SOC0 = 0.25)
Voltage [V]
Time [s]
×10^4
Experimental
EKF
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

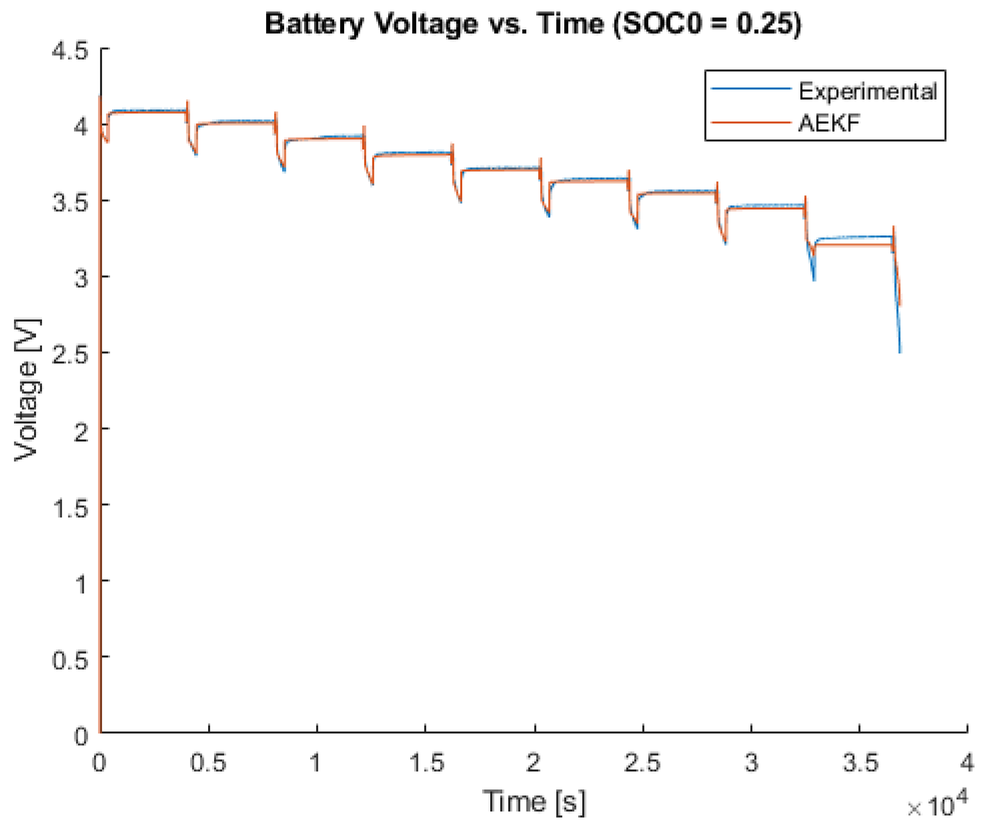

SOC vs. Time (SOC0 = 0.25)
SOC [%]
Time [s]
×10^4
Experimental
EKF
0
10
20
30
40
50
60
70
80
90
100



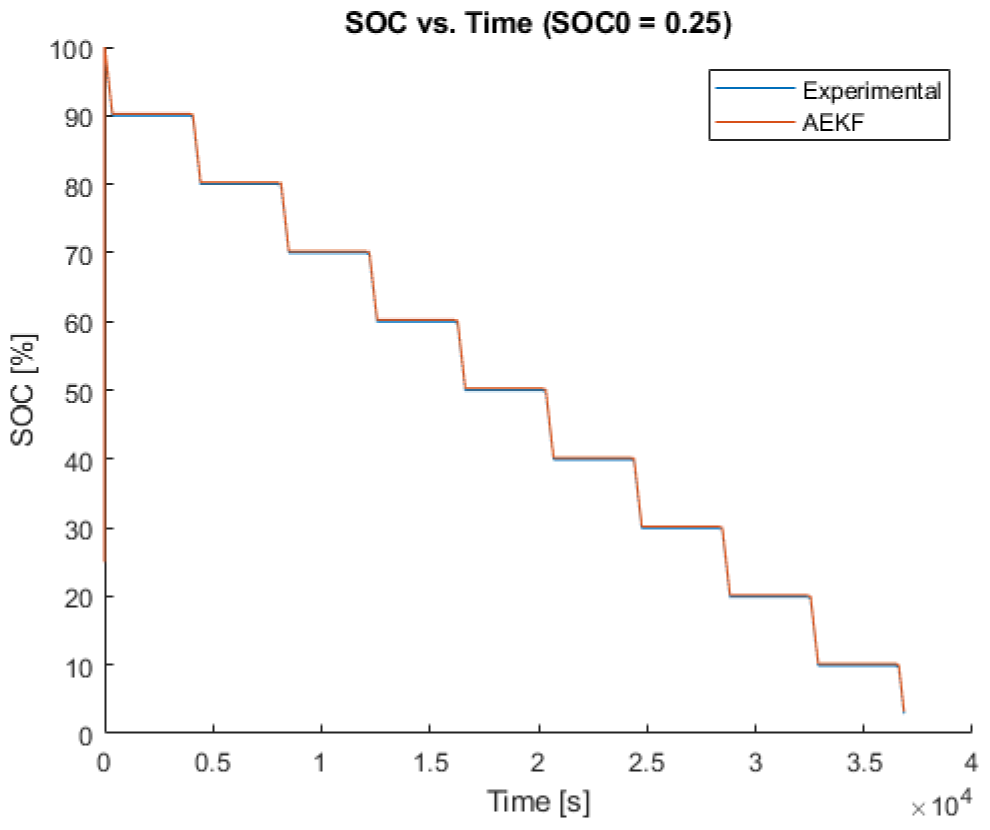
Error in SOC from EKF vs. Time (SOC0 = 0.25)
Error in SOC [%]
Time [s]
×10^4



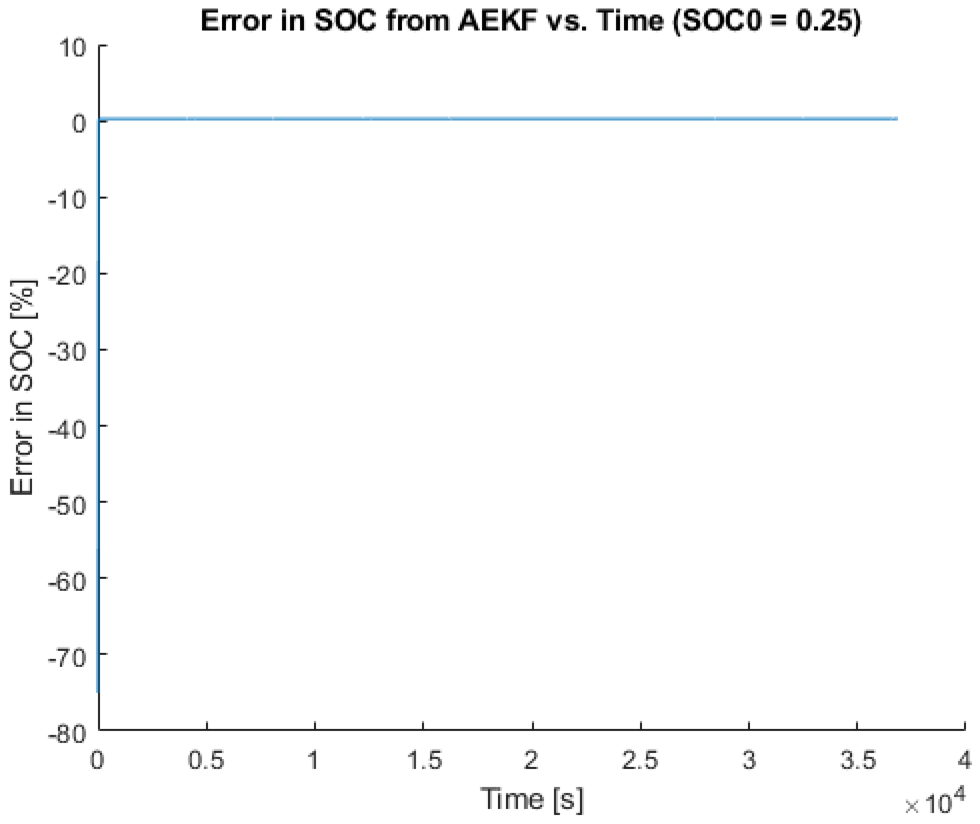
EKF Gain vs. Time (SOC0 = 0.25)
EKF Gain on SOC [%/V]
EKF Gain on V1 [V/V]
EKF Gain on V2 [V/V]
Time [s]
×10^4

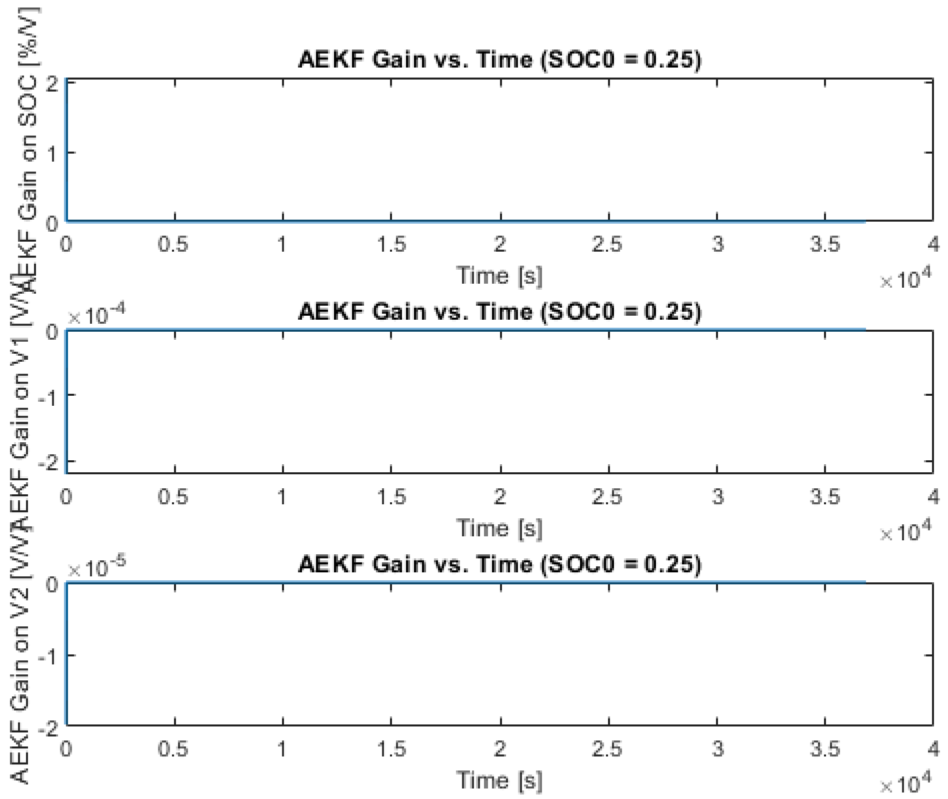


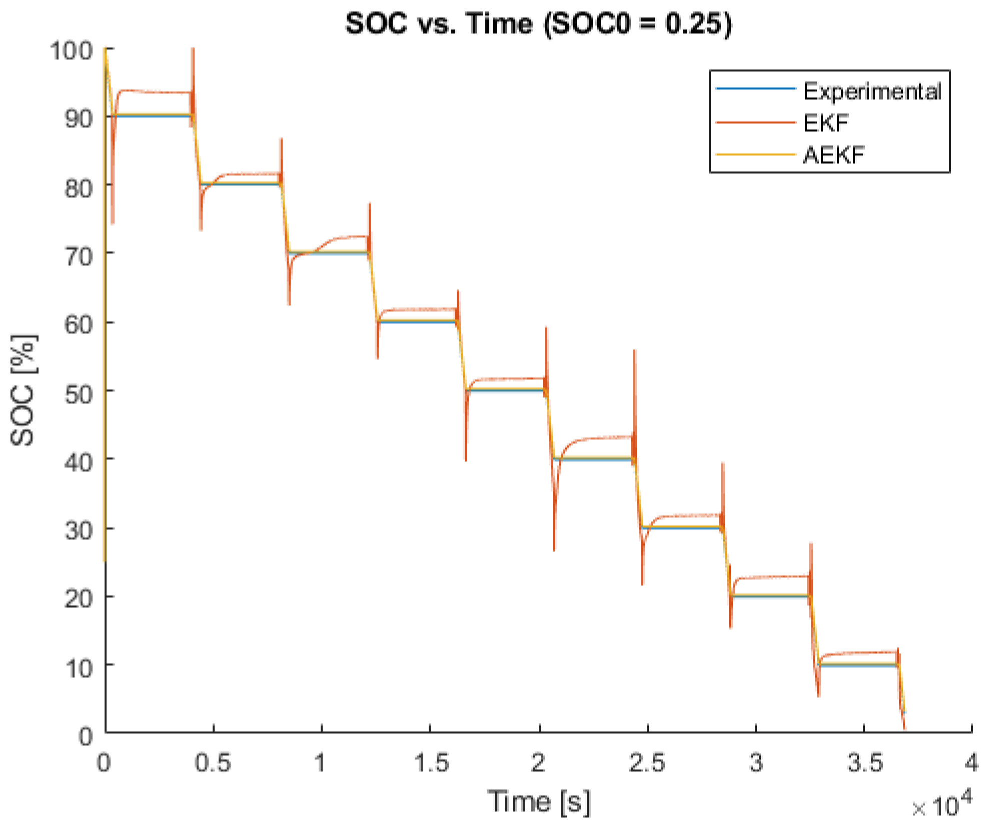
Battery Voltage vs. Time (SOC0 = 0.25)
Experimental
AEKF
Voltage [V]
Time [s]
$\times 10^4$



SOC vs. Time (SOC0 = 0.25)
Experimental
AEKF
SOC [%]
Time [s]
$\times 10^4$


Error in SOC from AEKF vs. Time (SOC0 = 0.25)
Error in SOC [%]
Time [s]
×10^4


AEKF Gain vs. Time (SOC0 = 0.25)
AEKF Gain on SOC [%/V]
AEKF Gain on V1 [V/V]
AEKF Gain on V2 [V/V]
Time [s]
×10^4
×10^-4
×10^-5



Published with MATLAB® R2021b