Abstract

Estimating state-of-charge (SoC) is essential for proper and secure battery management, especially when developing the Battery Management System (BMS). This project addresses this issue for lithium-ion batteries, the battery technology that is currently the most promising. To use the Kalman Filter Theory and create an algorithm for calculating SoC, an electric model of the cell is found and confirmed. The algorithm's primary goal is to reject measurement noise and parametric errors and to be adaptable to various cells made by the same manufacturer and using the same technology. Design criteria that shorten the convergence time or make the estimation resistant to noisy data are offered for this use.

1 Introduction

Nowadays, lithium-ion batteries are the preferred energy source for a wide range of electric applications, from tiny devices to Electric Vehicles (EVs). Lithium-ion batteries are the subject of major investment from manufacturers and academics due to their high power and energy capacities as well as the potential for development. Given the relatively recent technology we are using, there are still a number of battery application-related issues that need to be resolved.Among all research endeavours, SoC estimate is crucial because it enables the management of the battery in an accurate and secure manner. Although there isn't a single SoC definition, SoC is typically understood to be the ratio between a cell's available capacity and its maximum achievable capacity. Therefore, rather than being a real physical quantity, the SoC of a battery is an abstract energy idea. It must be guessed because there is no way to quantify it. Over time, several procedures and techniques have been created in order to provide estimates that are ever-more accurate. The most used methods are Coulomb counting, Open-Circuit-Voltage (OCV) estimates, and impedance spectroscopy.However, each of them has some drawbacks that make the methodology unfeasible for practical purposes.

In this project we tried to apply Kalman Filter algorithm for SoC estimation of battery. To apply Kalman filter, we use Rint model of battery i.e., we consider battery as combination of voltage source and an internal resistance. We implemented the algorithm in Arduino board and also use voltage and current sensor for battery readings. Kalman filter is mostly applicable for linear systems, but Li-ion batteries are not works linearly. So, we consider non-linear ESC battery model. To apply Kalman Filter to non-linear battery model we used modified Kalman Filter known as Extended Kalman Filter (EKF). We implemented the algorithm on Arduino board and used LCD display to display the SoC.

2 Aim of the project

- To determine SoC of a battery using Arduino.
- Implementation of Kalman Filter for SoC estimation
- Implementation of Extended Kalman Filter on Arduino for SoC estimation considering non-linear model of battery

3 Components used

Arduino board, Voltage sensor, Current sensor, LCD display, one Li-ion battery.

4 Batter model

To implement SoC algorithms on Arduino, we need to consider battery models for SoC calculation. First, for implementing general Kalman filter we use a linear model of battery cell.

4.1 Rint model

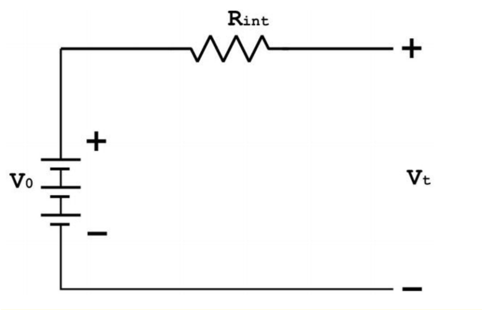


Figure 1: Resistive Thevenin battery model.[5]

For this model, we take battery as a combination of a voltage source and a internal resistance R_{int}.

$$V_t = V_0 - i(t)R_{int}$$

$$z(t) = z(t_{0)} - \int_{t_0}^{t} \frac{i(t)}{C_n}\, dt$$

Here, $z(t_0)$ is the initial state of charge, C_n is the nominal capacity of the battery.
We use this model linear model for general Kalman filter implementation.
For, the implementation of Extended Kalman filter, we use the following non-linear model.

4.2 ESC model

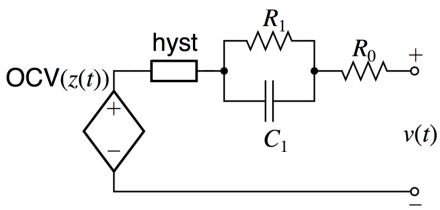


Figure 2: ESC battery model

ESC stands for enhanced self-correcting battery model.In this model we consider hysteresis effect of battery and RC connection is also considered in this model[3]. The ESC battery model models OCV (open circuit voltage) as a function of SoC, linear polarization, diffusion voltages, SoC-varying hysteresis and instantaneous hysteresis. When battery is in rest output voltage converges towards OCV + hysteresis and when battery is in use the output voltage converges towards OCV + hysteresis - $i(t)\sum R$.

Output equation for ESC cell model :

$$y_k = OCV(z_k) + M_0 s_k + M h_k - R_1 i_{R1,k} - R_0 i_k$$

5 Kalman Filter

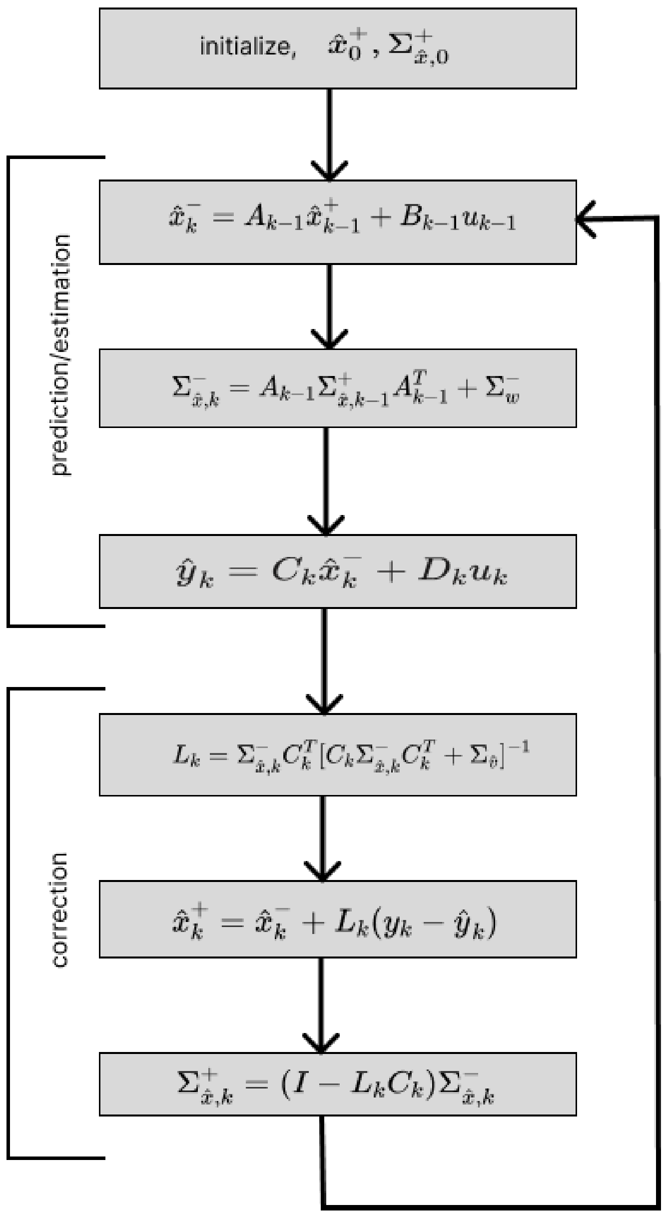
The Kalman filter is a mathematical algorithm that is used to estimate the state of a system based on a series of observations that may contain noise or errors.
The basic idea behind the Kalman filter is to use a probabilistic model of the system and the observations to iteratively update the estimate of the system state. The algorithm works by computing a prediction of the system state based on the previous estimate and the known dynamics of the system, and then combining this prediction with a new observation to produce an updated estimate. The update process takes into account both the uncertainty of the prediction and the uncertainty of the observation, using a set of equations that minimize the mean squared error of the estimate.
One of the key features of the Kalman filter is its ability to handle both linear and nonlinear systems, as well as systems with multiple sources of noise or uncertainty. The algorithm can be adapted to different types of systems by adjusting the mathematical model and the parameters used in the update equations. For example, in the case of a nonlinear system, the Kalman filter can be extended to use a more complex model, such as an extended Kalman filter or a particle filter.
For our project we derive six equation using kalman filter algorithm to estimate SoC of the battery. Later, we modified the Kalman filter to use it for non-linear system.

5.1 Kalman filter and SoC

Kalman filter works recursively measuring the parameter and then predicting the next step based on the error between current measurement and previous prediction. Kalman Filter assumes a cell model of the form

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1}$$
$$y_k = C_k x_k + D_k u_k$$

The following chart shows the kalman filter algorithm used for measuring SoC in our project[1].



Here,

$$
\begin{aligned}
\hat{x}_k- &= \text{estimated state of charge in k state} \\
\hat{x}_{k-1}^- &= \text{estimated state of charge in} \\
&\quad \text{previous (k-1) state} \\
\Sigma_{\hat{x}} &= \text{error covariance matrix of } \hat{x} \\
u_k &= \text{input, in our case current} \\
y_k &= \text{output, in our case voltage} \\
\hat{y_k} &= \text{measured voltage} \\
L_k &= \text{Kalman gain at k state}
\end{aligned}
$$

$A_k, B_k, C_k.D_k$ are constants that we get from linear battery model equations.

To use kalman filter on battery model, we consider the linear rint battery model. Also, we need OCV vs SoC curve of the battery model which we obtained experimentally. Then we use curve fitting method and got relation between OCV and SoC. We can write SoC(z_k) as

$$
\begin{aligned}
z_{k+1} &= z_k - \frac{1}{3600Q} i_k \\
v_k &= 2.85 + 0.89z_k - R_0 i_k \\
v_k - 2.85 &= 0.89z_k - R_0 i_k \\
y_k &= 0.7z_k - R_0 i_k
\end{aligned}
$$

comparing the above equations with state space equations and defining z_k as x_k, i_k as u_k, considering $R_0 = 0.01\ \Omega$ we got A = 1, B = $-1X10^{-4}$, C = 0.7 and D = -0.01.

In the above process we used Rint model of the cell to implement Kalman filter. But, most of the Li-ion batteries shows non-linear behavior. So, next we will use ESC cell model with extended kalman filter.

5.2 Extended Kalman Filter and SoC

To get more accuracy on SoC estimation, we will use non-linear model of cell with modified version of kalman filter - extended kalman filter (EKF)[1][2]. To implement EKF, we consider generalized system dynamics as

$$
\begin{aligned}
x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}) \\
y_k &= f(x_k, u_k, v_k)
\end{aligned}
$$

EKF makes two simplifying assumptions when adapting general sequential inference equations to a nonlinear system:

- When computing estimates of the output of a nonlinear function, EKF assumes $\mathbb{E}[f(x)] = f(\mathbb{E}[x])$.
- When computing covariance estimates, EKF uses Taylor-series expansion to linearize the system equations around the present operating point.

The following are the steps to estimate and update state equations using EKF[4] :

1. The state prediction step is approximated as
$$
\begin{aligned}
\hat{x}_k^- &= \mathbb{E}[f(x_{k-1}, u_{k-1}, w_{k-1}|Y_{k-1})] \\
&\approx f(\hat{x}_{k-1}^+, u_{k-1}, \tilde{w}_{k-1})
\end{aligned}
$$
2. The prediction-error covariance:
$$
\begin{aligned}
\Sigma_{\tilde{x},k}^- &= \mathbb{E}[(\tilde{x}_k^-)(\tilde{x}_k^-)^T] \\
&\approx \hat{A}_{k-1}\Sigma_{\tilde{x},k-1}\hat{A}_{k-1}^T + \hat{B}_{k-1}\Sigma_{\tilde{w}}\hat{B}_{k-1}^T
\end{aligned}
$$
3. System output is estimated to be
$$
\begin{aligned}
\hat{y}_k &= \mathbb{E}[h(x_k, u_k, v_k|Y_{k-1})] \\
&\approx h(\hat{x}_k^-, u_k, \tilde{v}_k)
\end{aligned}
$$
4. Estimator gain matrix (Kalman Gain) :
$$L_k = \Sigma_{\tilde{x},k}^-\hat{C}_k^T[\hat{C}_k\Sigma_{\tilde{x},k}^-\hat{C}_k^T + \hat{D}_k\Sigma_{\tilde{v}}\hat{D}_k^T]^{-1}$$
5. State update equation :
$$\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k)$$
6. Error covariance measurement update
$$\Sigma_{\tilde{x},k}^+ = \Sigma_{\tilde{x},k}^- - L_k\Sigma_{\tilde{y},k}L_k^T$$

State space equation for ESC cell model

$$\underbrace{\begin{bmatrix} i_{R_1,k+1} \\ h_{k+1} \\ z_{k+1} \end{bmatrix}}_{\boldsymbol{x_{k+1}}} = \underbrace{\begin{bmatrix} A_{\mathrm{RC}'} & 0 & 0 \\ 0 & A_{H,k} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} i_{R_1,k} \\ h_k \\ z_k \end{bmatrix}}_{\boldsymbol{x,k}} + \underbrace{\begin{bmatrix} B_{\mathrm{RC}'} & 0 \\ 0 & (A_{H,k}-1) \\ -\frac{\Delta t}{Q} & 0 \end{bmatrix}}_{\boldsymbol{B}} \underbrace{\begin{bmatrix} i_k + w_k \\ \mathrm{sgn}(i_k + w_k) \end{bmatrix}}_{\boldsymbol{u_k} \text{ with process noise}}$$

Here,

$$\begin{aligned}
y_k &= \text{measured output} \\
\hat{A}_{k-1} &= \frac{\partial f(x_{k-1}, u_{k-1}, w_{k-1})}{\partial x_{k-1}} \\
\hat{B}_{k-1} &= \frac{\partial f(x_{k-1}, u_{k-1}, w_{k-1})}{\partial w_{k-1}} \\
\hat{C}_k &= \frac{\partial h(x_k, u_k, v_k)}{\partial x_k} \\
\hat{D}_k &= \frac{\partial h(x_k, u_k, v_k)}{\partial v_k} \\
\Sigma_x &= \text{error covariance of matrix of } x \\
\Sigma_y &= \text{error covariance of matrix of } y \\
L &= \text{Kalman Gain matrix}
\end{aligned}$$

more about the equations and their derivations can be found in the references[2].
We recursively calculate these parameter and accordingly estimate SoC of the cell. For the implementation of the EKF algorithm, we use the following state space equations for ESC model given above on the page. There, A_{RC} = time constant of RC circuit, $A_{H,k}$ is hysteresis constant[1].

6 Simulation

We use one voltage sensor to measure the voltage across the battery and current sensor (ACS 712) to measure the current coming out of the battery. We used resistive load to discharge the battery and measure the SoC of the battery, and display it on a LCD screen. For LCD screen display, we use I2C module. In the LCD we display the SoC of the battery and the measurement error.
Schematic Circuit diagram of our simulation process is given in the next page. Also, for the implementation, we needed the OCV vs SoC curve, which we obtained by discharging the cell at a constant current of C/20. First we charged the battery to its full capacity and then discharged it at room temperature. We took data up to 70% charge of the battery and then used curve fitting method to find the relationship between OCV and SoC, which comes out to be

$$OCV = 0.889 z_K + 2.8525$$

We used ICR-18650 Li-ion cell which has total capacity of 1200mAh, nominal voltage 3.7V and life cycle of 300 cycle. The KF and EKF algorithm was implemented of Arduino uno micro-controller.

6.1 Simulation results :

Using KF algorithm, we got an error of approx 10% (max) and using EKF we got an error of less than 5%. The error mentioned above was calculate comparing OCV vs SoC curve and our estimated SoC using Arduino implemented algorithm.

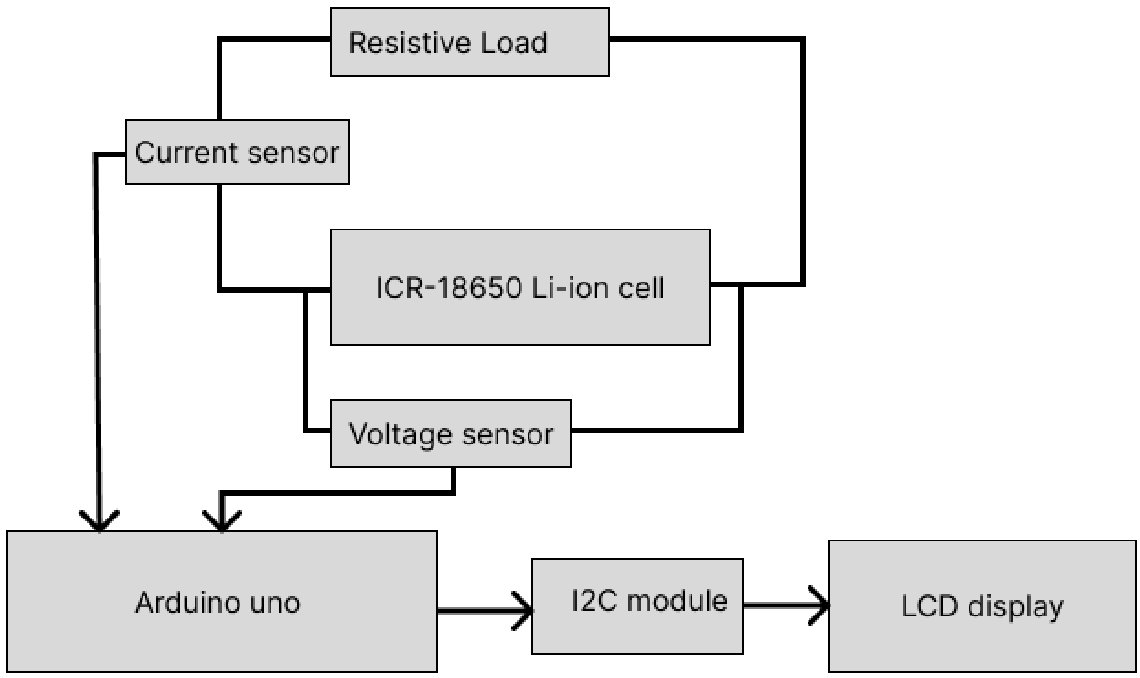


Figure 3: Schematic diagram of the circuit

7 Conclusion

We observe that the out put SoC that we got using kalman filter algorithm is quite correct, though since we our equipments used for the measurements are not error free, so some error was introduced in the output.
To get more accurate result we are trying to implement Extended Kalman filter considering the non-linear cell model.We were able to implement the kalman filter algorithm on Arduino. During experimental simulation, we measure voltage directly from the cell as it was less than 5V. But, for more than 5V battery, it will be more accurate to use voltage sensor. The project showed that kalman filter algorithm can be used to measure the SoC of battery depending on the use of it.
Also, further more improvements can be done to minimize the error. We consider constant values for hysteresis, which is not completely accurate. If we use variable hysteresis and look up table for OCV vs SoC relation, then we can more accurate results.

References

[1] Bjørn Kvaale. “Range Anxiety Alleviation for Electric Vehicles using a Dual Extended Kalman Filter with a Nonlinear Battery Model”. MA thesis. NTNU, 2017.

[2] Benedikt Rzepka, Simon Bischof, and Thomas Blank. “Implementing an extended Kalman filter for SOC estimation of a Li-ion battery with hysteresis: A step-by-step guide”. In: *Energies* 14.13 (2021), p. 3733.

[3] Pierfrancesco Spagnol, Stefano Rossi, and Sergio M Savaresi. “Kalman filter SoC estimation for Li-ion batteries”. In: *2011 IEEE International*

Conference on Control Applications (CCA). IEEE. 2011, pp. 587–592.

[4] Carlo Taborelli and Simona Onori. “State of charge estimation using extended Kalman filters for battery management system”. In: *2014 IEEE International Electric Vehicle Conference (IEVC)*. IEEE. 2014, pp. 1–8.

[5] Wenlu Zhou et al. “Review on the battery model and SOC estimation method”. In: *Processes* 9.9 (2021), p. 1685.