



# Current Sensorless Microcontroller-Based Battery Management System with SOC and Active Cell Balancing

Muhammad Fikri Ardiansyah, Adha Imam Cahyadi and Oyas Wahyunggoro

*Departement of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada*

*\*Corresponding author email: fikriardan51@gmail.com*

---

## Abstract

Battery management system (BMS) has become an important research topic following the trend and development of the electric vehicle. Although research on Active Cell Balancing, SOC, and current estimation has been carried out, the previous work mostly focused on comparing and developing methods. In this research, we demonstrate the process of designing BMS hardware using a low-cost microcontroller and without using a current sensor. The SOC simulation results produce an RMSE of 0.0832% for the 100% -10% SOC-OCV curve, and the current estimation simulation produces an RMSE of 0.2576 A, while for testing using a 6-ohm pulse load, the RMSE error value is 0.3960 A. The Active Cell Balancing method was successfully performed in simulation with Simulink. Furthermore, our simulation and test results suggest that complex battery models and multiple SOC-OCV curves can be used for better current and OCV estimation results. Our experimental results are also useful to develop a guideline to design a microcontroller-based BMS.

*Keywords:* Active Cell Balancing, Current Estimation, Low-cost Microcontroller, State of Charge.

---

## 1. Introduction

Nowadays, batteries are widely used as a power supply for electronic devices such as cell phones and laptops. On a large-scale application, a battery can be used as a power supply for vehicles and even powering a building. Tesla, for example, is the most known company that utilizes batteries in their products, create bricks of battery called Powerwall that can power a building for a limited time (Tesla, 2020). Flexibility and portability are the main reason for using batteries as the main energy source in electronic devices. However, the large-scale consumption of batteries may harm the environment. Battery waste is a type of hazardous waste that can easily contaminate the environment when not handled properly (Skeete et al., 2020).

Improper use of batteries can cause damage and shorten battery life. A damaged battery can be the result of overcharge, over-discharge, overload, overheat, and short circuit. To overcome this problem a Battery Management System (BMS) is needed (Wei et al., 2020). State of Charge (SOC) is one of the most important variables and features that can describe the condition of battery capacity in a BMS (Chandran et al., 2021). Also, for multiple cell batteries, a cell balancing feature is needed to maintain the balance of capacity between battery cells. In the market, good quality BMS, has a fairly expensive price, especially for advanced applications such as electric vehicles. One way to reduce its costs is to eliminate current sensors. Putra et al. (2015) stated that the battery current can be estimated by deriving the battery model equation. In his research, the resulting current estimated error value is RMSE of 0.103 Amp for pulse load and 0.176 Amp for the varying-load experiment. Kupper et al. (2018) also estimated the current value through the battery model, with a more complex model involving multiple cell batteries.

Various researches have been done on BMS design, notably in the cell balancing method. Jalil (2017) applied the Active Cell Balancing method using a PWM controlled converter on a 4-cell lithium-polymer battery. The Active Cell Balancing method allows the transfer of energy between battery cells, so this method is considered more efficient than passive cell balancing which uses a shunt resistor (Zhang et al., 2020). Research on SOC had been done in (Fathoni, 2017), who comparing the three common methods, the coulomb counting method, the OCV method, and

the Kalman filter method. In this research, we aim to design a low-cost microcontroller-based Battery Management System with SOC and Active cell balancing features without using a current sensor.

## 2. Materials and Methods

### 2.1. Battery Modeling

Choosing a battery model is the first step in estimating battery current and estimating SOC. Although a battery model that is better at representative of the system has been developed, in this case, a simple battery model is more suitable because of the limitations of the microcontroller. The more complex the battery model used, the greater the computational load of the microcontroller. The simple model battery or the  $R_{int}$  model is shown in Figure 1.

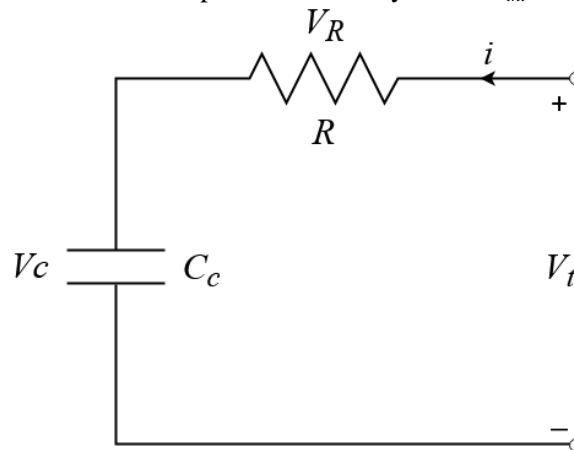


Figure 1. Simple Battery Model

It consists internal capacitance or battery capacity  $C_c$ , internal resistance  $R$ , capacitor voltage  $V_c$  or also known as Open Circuit Voltage (OCV), resistor voltage  $V_r$ , and the terminal voltage  $V_t$ . This battery model will be used for a 11.1V, 3s, 2200mah, 25C Lithium Polymer battery from gens ace.

### 2.2. Parameter Identification

There are two parameters that need to be determined in the  $R_{int}$  battery model. These parameters are the internal resistance  $R$  and the internal capacitance  $C_c$ . Internal resistance  $R$  can be obtained by discharging the battery on full charged condition with a periodic pulse-load profile. This periodic pulse load consists of 30s discharge with 1C current followed by a 30s rest time. The process will be continued until  $K$  periods where the battery capacity runs out (fully discharged).

$$R[j] = \frac{\Delta V[j]}{i} . \quad (1)$$

The determination of  $R$  expressed in (1) where  $j= 1, \dots, K$  and  $\Delta V[j]$  is the voltage drop between rest period to discharge period, in other words,  $\Delta V[j]$  is the voltage difference between OCV ( $V_c$ ) and the last terminal voltage within period  $j$  (Dewangga et al., 2018).

$$R = \frac{1}{K} \sum_j^K R[j] . \quad (2)$$

Although the internal resistance value changes within each period, in this research the value of  $R$  is considered constant to simplify calculations. This value can be obtained by calculating the mean of  $R[j]$  as shown in (2). As for the internal capacitance  $C_c$ , will be explained in the next section.

### 2.3. SOC Estimation

The method used in estimating SOC is the Open Circuit Voltage (OCV) method. This method consists of two-step, firstly is estimating OCV. The relationship between the terminal voltage  $V_t$  and OCV ( $V_c$ ) that derived from the battery model is used, which is represented by (1). This equation is obtained by applying the Kirchhoff laws to the battery model. However, since the SOC estimation algorithm will be performed using a microcontroller, it is necessary to discretize (1).

$$V_t(t) = V_c(t) + RC_c \frac{dV_c}{dt}. \quad (3)$$

Laplace transform operation is used to change the differential equation (1) to the complex function (2). Then, by applying bilinear transformation on (2), a discrete equation (3) is produced.

$$Vt(s) = V_c(s) + sRC_c V_c(s). \quad (4)$$

$$V_t(z) = V_c(z) + \frac{2RC_c(V_c(z) - z^{-1}V_c(z))}{T_s + T_s z^{-1}}. \quad (5)$$

The discrete equation (3) needs to convert to a discrete-time domain using the inverse Z-Transform to produce the final equation (4) with  $\beta = \frac{T_s}{T_s + 2RC_c}$ .

$$V_c[k] = \beta(V_t[k] + V_t[k - 1]) + (1 - 2\beta)V_c[k - 1]. \quad (6)$$

Secondly, an equation that can convert the OCV value to the SOC value is needed. This equation is a relation or function between SOC and OCV,  $SOC=f(OCV)$ , that can be obtained through polynomial fitting of the SOC-OCV curve (Zhang et al., 2018). This curve obtained by the same process when calculating the internal resistance R. Otherwise, a piecewise linear function can be used to represent the SOC-OCV curve (Dewangga et al., 2018).

## 2.4. Current Estimation

The battery current value may be estimated by solving (7). This equation achieved by applying the Ohm's Law (5) and the capacitor current equation (6).

$$i(t) = C_c \frac{dV_c(t)}{dt}, \quad (7)$$

$$i(t) = \frac{V_t(t) - V_c(t)}{R} \quad (8)$$

$$i(t) + RC_c \frac{di(t)}{dt} = C_c \frac{dV_t(t)}{dt} \quad (9)$$

Again, we must discretise (7) when using a digital hardware. Employing the Laplace and bilinear transform on (7) yields

$$I[k] = \left(\frac{1-\beta}{R}\right)(V_t[k] - V_t[k - 1]) + I[k - 1] \quad (10)$$

$$C_c = C_N \frac{dSOC}{dV_c} = C_N \frac{df_{soc-ocv}(V_c)}{dV_c} \quad (11)$$

Where  $\beta = \frac{T_s}{T_s + 2RC_c} \cdot C_c$  is the battery internal capacitance as expressed by (11), and  $C_N$  is the maximum capacitance of the battery. This also applies to the SOC estimation. For a summary, the current and SOC estimation process is shown in Algorithm 1.

---

### Algorithm 1. SOC and Current Estimation

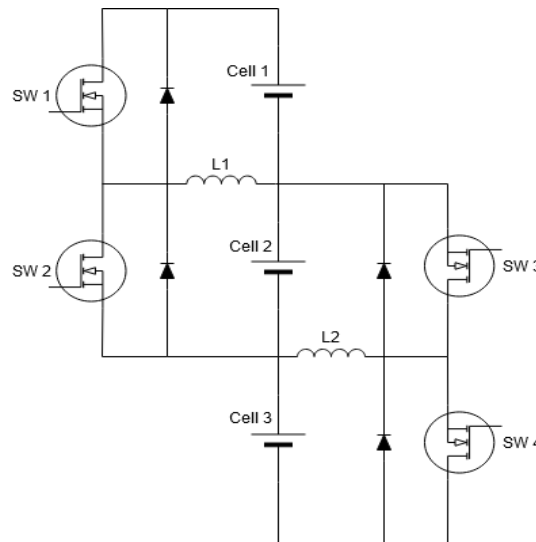
---

- 1:  $k = 0$
  - 2: initialize R,  $T_s, C_N$
  - 3: Read VT[0]
  - 4:  $VC[0] = VT[0]$
  - 5:  $SOC = f_{SOC}(VC)$
  - 6:  $I[0] = 0$
  - 7: Calculate  $C_c[0]$
  - 8: Calculate  $\beta[0]$
  - 9: **while**  $k > 0$  **do** :
  - 10:  $VT[k-1] = VT[k]$
  - 11:  $VC[k-1] = VC[k]$
  - 12: Read VT[k]
  - 13: Calculate VC[k] using (6)
  - 14:  $SOC = f_{SOC}(VC)$
  - 15: Calculate I[k] using (10)
  - 16: Update  $C_c$  using (11)
  - 17: Update  $\beta$
  - 18: **end while**
-

## 2.5. Active Cell Balancing

The active cell method is a battery balancing method based on the active transport of energy between cells. This method does not depend on the chemical characteristics of the battery and can be used for most types of batteries. There are many different types of active cell balancing method. Cell to cell, cell to battery, and battery to cell are the common types used in a BMS. The cell-to-cell method allows the transfer of energy between adjacent battery cells. This method is suitable for use in batteries with fewer cells as in this research.

Several types of circuits can be used to implement the cell to cell method, such as switched capacitors, converters, and PWM controlled converters. The PWM controlled converter circuit with the buck-boost converter was selected in this study. For three-cell battery implementation, the PWM controlled converter circuit is shown in Figure 2.



**Figure 2.** Active Cell Balancing Circuit with Buck-Coost Converter

The main idea of cell to cell method is that the battery cell with the highest capacity will transfer its energy to the adjacent and the lower one (Jalil, 2017). This method requires a decision transfer algorithm that is programmed through a microcontroller. This algorithm is used to determine the energy transfer sequence. When all of the cells SOC value is already known, the cells which should transfer its energy are decided. For example, when SOC cell 1 is greater than SOC cell 2, the transfer energy should flow from cell 1 to cell 2, accordingly the microcontroller will send a PWM to SW1 so the energy will flow from cell 1 to cell 2. The transfer decision algorithm is shown in Algorithm 2.

---

### Algorithm 2. Algorithm Transfer Decision

---

- 1: Read SOC1, SOC2, SOC3
  - 2: If SOC1>SOC2 do:
  - 3:     PWM to SW1
  - 4: If SOC2>SOC1 do:
  - 5:     PWM to SW2
  - 6: If SOC2>SOC3 do:
  - 7:     PWM to SW3
  - 8: If SOC3>SOC2 do:
  - 9:     PWM to SW4
  - 10: If SOC1=SOC2=SOC3 do :
  - 11:     Finish Balancing
- 

## 2.6. Procedure

There are three main components in the designed BMS. These components are Active Cell Balancing Circuit, Microcontroller, and Voltage Sensing Circuit. The voltage sensing is used to read the voltage of each battery cell using the LM324N IC. The voltage sensing circuit schematic can be found in Figure 4. After the battery voltage is obtained, the voltage data is then processed through a microcontroller to produce the overall SOC, SOC of each cell, and current  $i$ . the SOC of each cell data then reprocessed to perform the balancing process through a transfer decision

algorithm. Transfer decision algorithm generates a PWM signal that will drive the mofset so that the balancing process can be performed.

For the microcontroller, we use 32 bits Teensy 3.2 by PJRC. This microcontroller is an USB-based microcontroller development system, and has a small dimension. It can be programmed through Arduino IDE and also compatible with many Arduino library (Vaidyanathan et al., 2017; Mujiarto et al., 2019).

Figure 5 shows a block diagram of the research process. The research was started by identifying and determining parameters, the internal resistance (R) value, and the SOC-OCV curve. Furthermore, a simulation is carried out to test the current estimation equations and SOC using MATLAB software, and Active cell balancing simulation is done through MATLAB-Simulink. The external circuit board consists of two components, namely the voltage sensing circuit and the active cell balancing circuit. Finally, the process of implementing the SOC method and estimating the current into the microcontroller, and connecting the external circuit board with the microcontroller is carried out according to Figure 3.

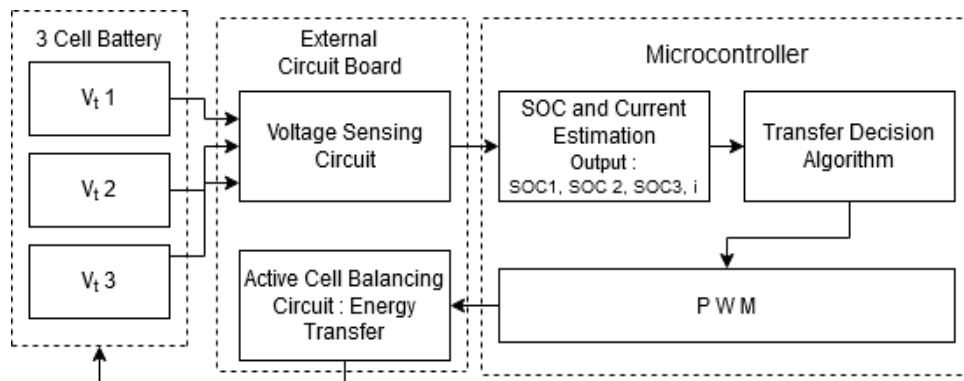


Figure 3. BMS Component

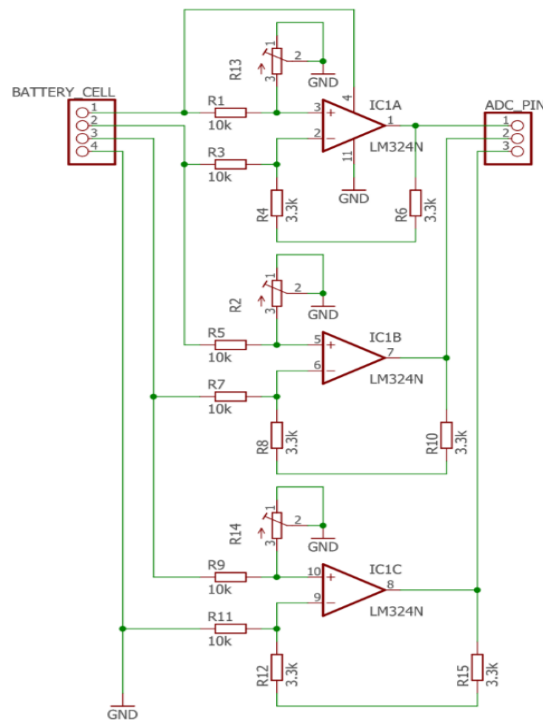


Figure 4. Voltage Sensing Schematic

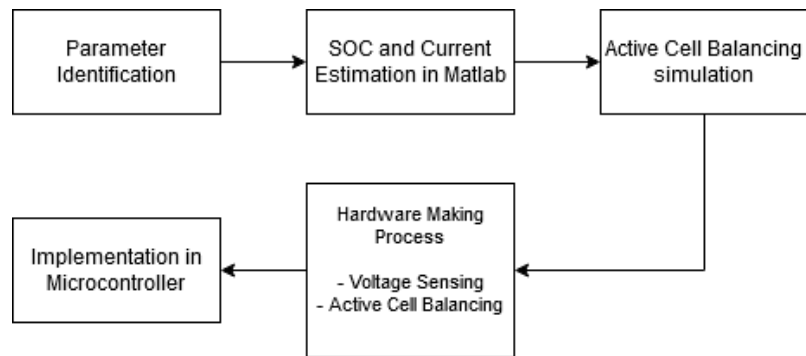


Figure 1. BMS making process

### 3. Results

#### 3.1. Determinate Internal Resistance R

The pulse test begins with a discharge process using 1C current, 2.2 A (according to the nominal battery capacity of 2.2 Ah). The discharge duration is 30 seconds and rest duration are 30 seconds, so it spends one minute in one cycle. This process is carried out for 2 hours for each battery cell. The result of this pulse test is the battery cell voltage ( $V_c$  and  $V_t$ ) as can be seen in Figure 7, and the battery current ( $i$ ) data. Through (1), there are a total of 120 Internal Resistance  $R$  values for each cell that achieved and graphically shown in Figure 6. From this figure, we can clearly observe that the internal resistance value is different for each battery cell, due to differences in the chemical characteristics of the battery cells. From these 120  $R$  values from each cell, we generated  $R_{savg}$  which is the average value of the three cell. Then, the internal resistance value  $R$  is the average value of  $R_{savg}$ . Achieved by (2) the value of the internal resistance  $R$  for a single cell battery is  $0.024669 \Omega$ .

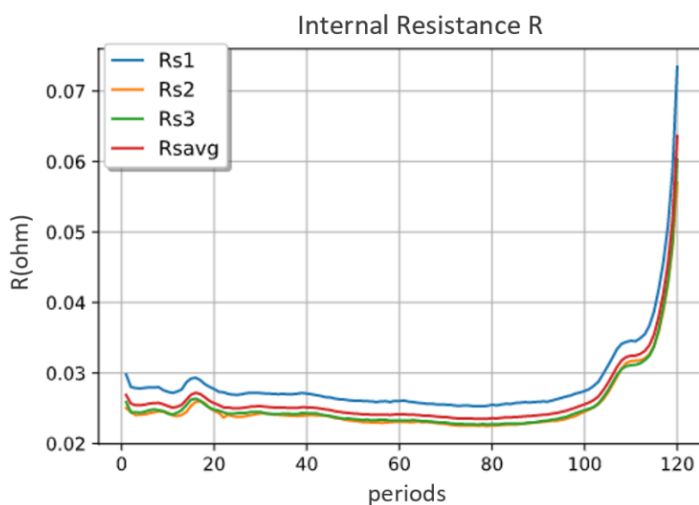


Figure 6. Internal Resistance Value of Cell Battery

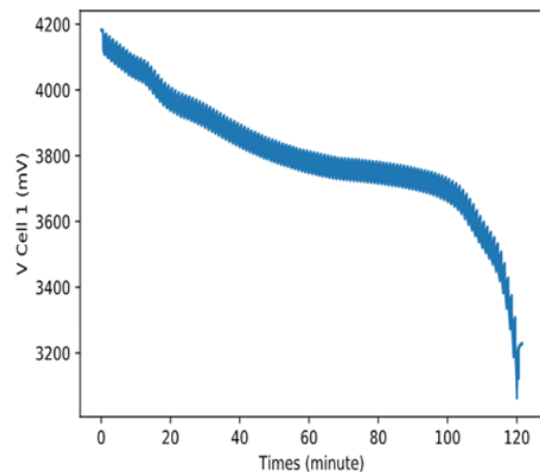


Figure 7. Pulse-Load experiment data

#### 3.2. SOC-OCV Curve

The SOC-OCV curve is a ratio curve between the value of the OCV and the battery capacity. From the data that has been collected, the making of the SOC-OCV curve begins by eliminating the terminal voltage so that only the OCV voltage remains. Due to this elimination of terminal voltages, it is necessary to make adjustments to the time axis, the origin of which has a range from 0 to 7,200 seconds (120 minutes) to 0 to 3,600 seconds (60 minutes). The SOC value is obtained by normalizing the axis or time value. In short, when the time is 0 seconds, the battery capacity is full or SOC 100%, and when the time is 3,600 seconds the battery is empty or SOC 0%.

There are differences in the three SOC-OCV curves. This is because each cell in the battery has different characteristics. The characteristics difference of battery cells occurs not only in the battery used in the test but basically, every battery or battery cell. So, for this BMS design, it is necessary to generalize the SOC-OCV curve that has been obtained. The easiest way to generalize the SOC-OCV curve is to use the average value of the SOC-OCV curve of the three cells so that the result is the average SOC-OCV curve (Average OCV) shown in Figure 8.

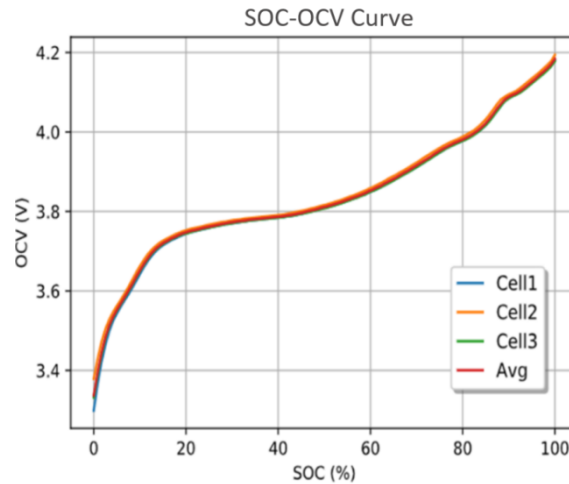


Figure 8. SOC-OCV Curve

### 3.3. SOC Estimation Result

In estimating SOC, the first thing to do is create a simulation using the battery data that has been obtained. This simulation aims to evaluate the performance of the OCV estimation equation (6). The first step in the OCV simulation is to create a function of  $SOC = f(OCV)$ , which will be used to update the value of  $C_c$  (11). This function is obtained using the polynomial curve fitting (polyfit) command/function in MATLAB. In the estimation of OCV by Bobby (Dewangga, 2015), the estimation results were compared when using normal SOC-OCV data and 80% - 20% SOC-OCV data. 80% - 20% SOC-OCV restrictions produce better estimated error values than when using normal SOC-OCV data. In this research, we compare three types of SOC-OCV curve, normal SOC-OCV data, SOC-OCV data with 10% restrictions (100% - 10% SOC), and SOC-OCV data with 20% restrictions (100% - 20% SOC).

Table 1. OCV Estimated Error Value

Curve	RMSE (mV)	Order
Normal SOC-OCV	0.1352	12
SOC-OCV 10% (100% - 10% SOC)	0.0560	10
SOC-OCV 20% (100% - 20% SOC)	0.0757	11

Polyfit function require a polynomial order. This polynomial order is obtained by trial error method, by testing the polynomial order of the polyfit function starting from the 6th polynomial order to the 15th polynomial order. The estimated error resulted from the three data shown in Table 1, and we can conclude that the 100% - 10% SOC-OCV has the better estimated error value. Figure 9 shows the OCV estimation results for the three-cell battery using the 100% - 10% SOC-OCV data from 1,200s to 1,800s. The comparison between estimated OCV and real OCV can be seen in Figure 10.

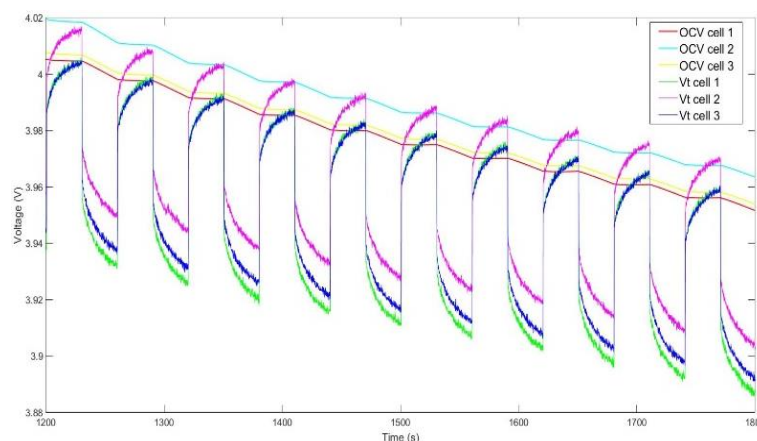
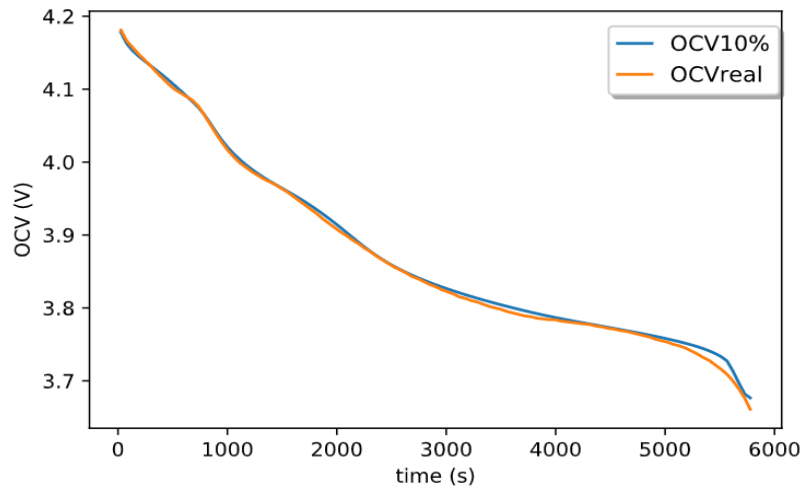


Figure 9. Simulation result of three battery cell voltage using 100% - 10% SOC data

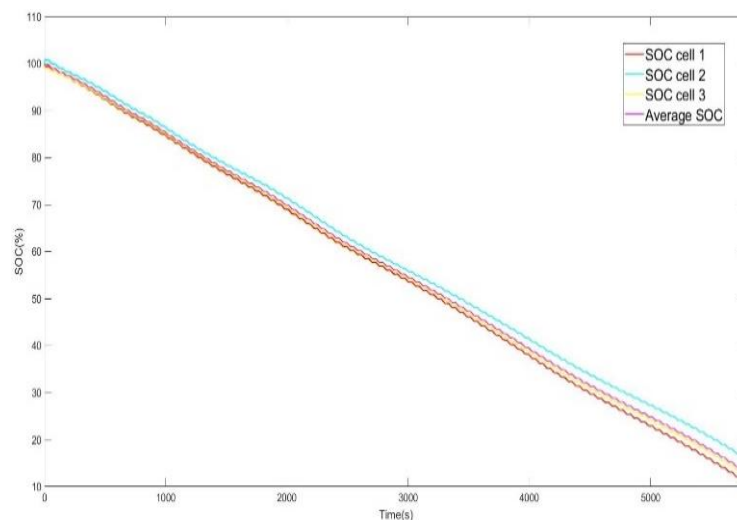


**Figure 10.** Comparison between estimated OCV and real OCV

After the OCV value is obtained, the SOC value is obtained by mapping the OCV value into the  $SOC = f(OCV)$  function. In a simulation using MATLAB, this mapping can be done through the polyfit and polyval commands. Polyfit is a function in MATLAB to create or form polynomial equations that can map two columns or arrays of data using the principle of polynomial regression. Meanwhile, polyval is a command to map a value (in this case OCV) through the function that has been obtained.  $OCV(k)$  is the value of OCV when state  $k$ . The SOC value generated through this equation is shown in Figure 11. The SOC Estimated Error Value can be seen in Table 2.

**Table 2.** SOC Estimated Error Value

Curve	RMSE (%)	Order
Normal SOC-OCV	0,2053	12
SOC-OCV 10% (100% - 10% SOC)	0.0832	10
SOC-OCV 20% (100% - 20% SOC)	0.2266	11



**Figure 11.** SOC result

The use of polyfit or regression methods with high order polynomials is likely to have an impact on the performance of the microcontroller. The use of polyfit will increase the computational load quite large, depending on the polynomial order used. Given that the microcontroller used is a low-cost microcontroller, this method is considered less efficient, so that another method is proposed, the Linear Piecewise Function. Linear piecewise function is a function that consists of a combination of linear sub-functions for each specific interval range in which the sub-function applies. Linear piecewise functions can be used to approach nonlinear curve functions by dividing them into several linear functions so that the computation required to calculate the nonlinear function becomes lighter because the calculations are based on linear functions.



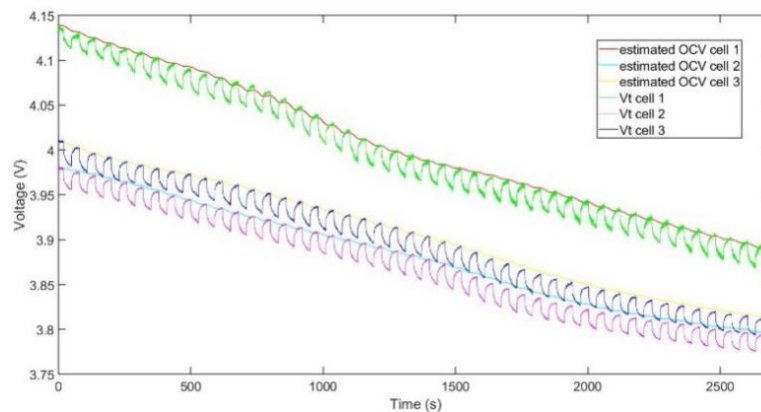
In implementing a piecewise linear function, and Look up Table (LUT) is needed which contains a sample of OCV<sub>n</sub> and SOC<sub>n</sub> data from the curve to be estimated, in this case the f curve (SOC-OCV). OCV<sub>m</sub> data is data obtained from the results of the OCV estimation process based on measured battery terminal voltage data. Furthermore, the SOC<sub>m</sub> value, the SOC data value that corresponds to OCV<sub>m</sub>, can be seen in equation (7).

$$f(OCV_m) \approx SOC_m = SOC_n + \frac{SOC_{n+1} - SOC_n}{OCV_{n+1} - OCV_n} \times (OCV_m - OCV_n) \quad (7)$$

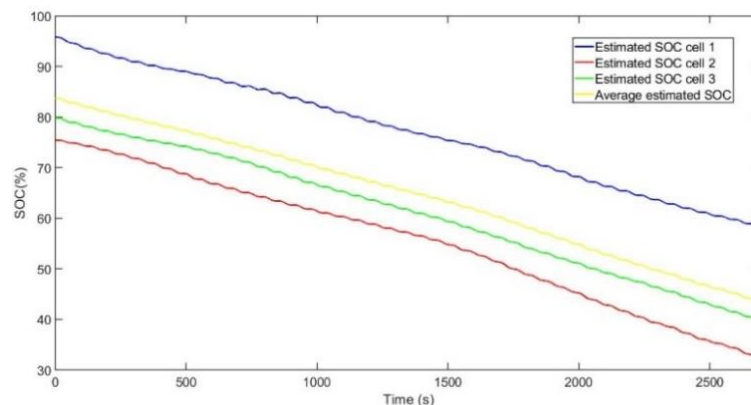
After the SOC estimation algorithm is implemented into the microcontroller program, the test is carried out using a resistive load of 6 Ω with a switching process that is run using a relay module for a duration of ± 2740 seconds so as to produce a discharge current in the form of pulses such as a pulse test testing process with a current load draw time of 30 seconds and a rest time of 30 seconds. The initial SOC condition of the battery is shown in Table 3. The voltage graph of the test results is presented in Figure 12, and the SOC result presented in Figure 13.

**Table 3.** SOC test Battery condition

Battery Cell	SOC (%)	
	Start	End
Cell 1	95,85	57.79
Cell 2	75.43	31.84
Cell 3	79.78	39.18



**Figure 12.** OCV estimation test result



**Figure 13.** SOC estimation test result

Although the estimation results of SOC using the OCV method are good, the OCV method relies on voltage measurements, so a high precision voltage sensing device is needed for a better result. Another drawback of this method is that, the SOC-OCV curve is affected by many factors. Different rates of current applied for the discharge cycle when making SOC-OCV curve, causes the OCV-SOC curve shifts up or down depending on the applied current value (Ipek et al., 2019). This indicates that the SOC estimation using the SOC-OCV curve will be more accurate when the battery current value is close or equal to the current value on the SOC-OCV curve. However, multiple SOC-OCV curve with different discharge current value can be used.

### 3.4. Current Estimation Result

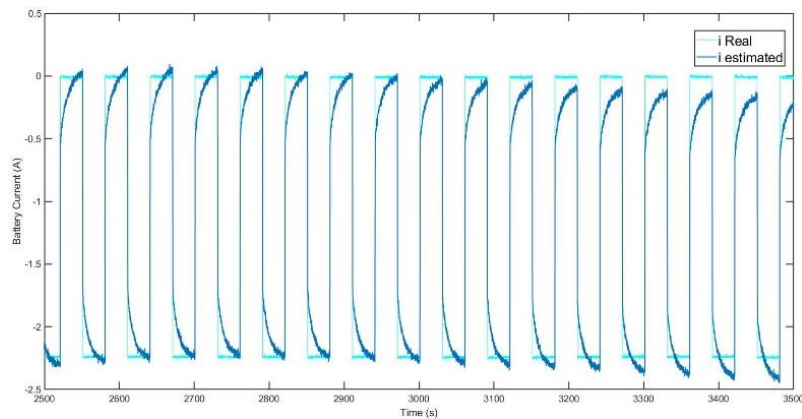
The first step in estimating battery current is to make the function  $SOC = f(OCV)$ . This function is used to update the value of  $C_c$  and the value of  $B$ . A simulation is performed using Matlab to evaluate the current estimation equation (10). The estimation error value is shown in Table 4 with a variation of the SOC-OCV curve. The 10% - 100% SOC-OCV curve has the smallest error value of 0.2576 A. Figure 14 show the simulation results of current estimation using the 100% - 10% SOC-OCV curve from 2,500s to 3,500s.

Furthermore, the current estimation is implemented into the microcontroller and tested by pulling the discharge current in the form of pulses with a current load withdrawal time for 30 seconds and a rest time for 30 seconds. The load used is a resistive load of  $6 \Omega$  with a switching process that is run using a relay module for a duration of  $\pm 2,740$  seconds. The current flowing in the battery is measured using the INA219 current sensor to compare its value with the estimated current.

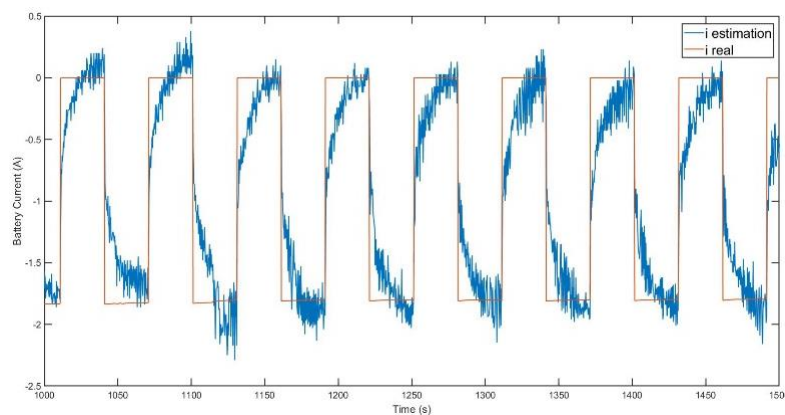
The graph of the test results presented in Figure 15 shows the average current estimated value of each cell compared to the measured current from 1,000s to 1,500s. In the transient region, when the current value change instantly, the difference between the estimated and the measured current of the sensor is high because the estimated current value takes time to converge to reach the true current value.

**Table 4.** Current simulation error

SOC-OCV curve	RMSE	Order
SOC20%	0.2785 A	12
SOC10%	0.2576 A	10
Normal SOC	0.7700 A	11



**Figure 14.** Current Estimation Simulation Result



**Figure 15.** Current Estimation Test Result

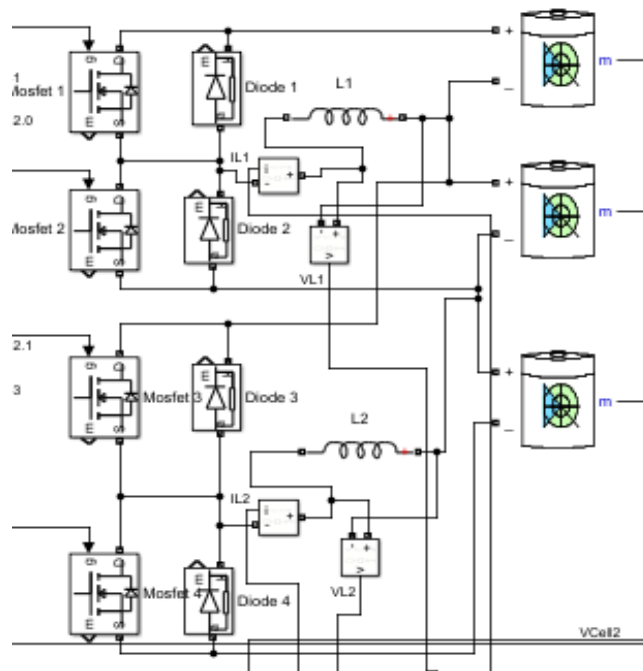
This happens because there are characteristics of the battery that are not perfectly modeled by the  $R_{int}$  battery model so that the current estimation based on the model becomes less accurate for real-time observation. However, the estimated current still can follow a pulse-shaped pattern with estimation error RSME 0.396 A.

### 3.5. Active Cell Balancing

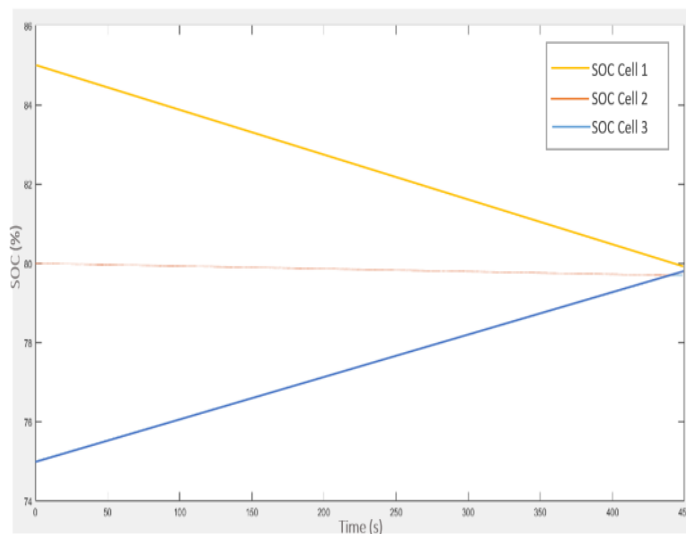
The design of the active cell balancing circuit begins with a simulation using MATLAB Simulink. The simulation is done by doing self-balancing of the three battery cells. The simulation circuit is shown in the Figure 16 with the SOC battery conditions according to the table. Based on the transfer decision algorithm in algorithm 2, there will be a flow of energy transfer from battery cell 3 to battery cell 2 and battery cell 2 to battery cell 1.

**Table 5.** SOC Battery condition in simulation

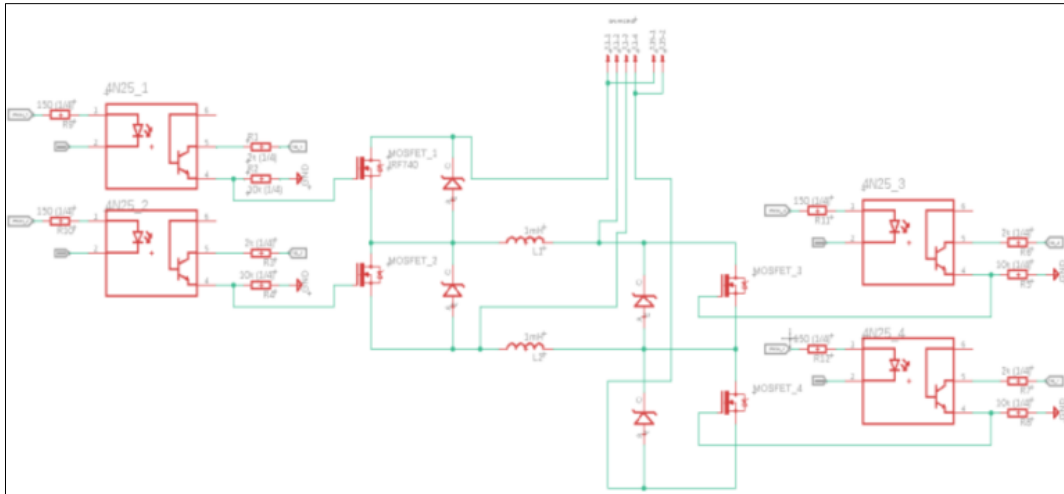
Battery Cell	SOC (%)	
	Start	End
Cell 1	75	79.69
Cell 2	80	79.7
Cell 3	85	80.6



**Figure 16.** Active Cell Balancing simulation circuit



**Figure 17.** Active cell balancing simulation result



**Figure 18.** Active cell balancing schematic circuit with mosfet driver

The simulation results shown in Figure 17, show that energy transfer has occurred, and the balancing process can take place. The graph of SOC cell 1 and cell 3 moves towards the 80% and the balancing process lasts for 450s. Figure 18 shows the circuit of active cell balancing hardware. This circuit is an implementation of Figure 2 and Figure 16. We used IRF740 and optocoupler 4N25 as mosfet drivers. Besides that, an inductor of 1mH is used.

#### 4. Conclusion

Battery management system (BMS) has become an emerging research topic following the trend and development of the electric vehicle. With the State of Charge (SOC) feature and the Cell balancing feature which is the main feature in a BMS, it is possible to maintain the battery life. The current estimation method makes it possible to eliminate the current sensor on BMS while reducing production costs. Although research on Active Cell Balancing, SOC, and current estimation has been carried out, the previous work mostly focused on comparing and developing methods. In this research, we demonstrate the process of designing BMS hardware using a low-cost microcontroller and without using a current sensor. The SOC value was obtained using the OCV method by utilizing the SOC-OCV curve. Current estimation and OCV estimation equations are achieved using a simple battery model ( $R_{int}$  model), whose performance is evaluated using a simulation through MATLAB. Meanwhile, Active Cell Balancing using PWM controlled converter (Buck-Boost Converter) was tested through simulation using SIMULINK. The SOC simulation results produce an RMSE estimation error of 0.0832% for the 100% - 10% SOC-OCV curve. The current estimation simulation produces an RMSE estimation error of 0.2576 A, while for testing using a 6  $\Omega$  pulse load, the RMSE error value is 0.396 A. SOC 1 = 85%, SOC 2 = 80%, and SOC 3 = 75% then after balancing it becomes SOC 1 = 79.69%, SOC2 = 79.7%, SOC3 = 80.6%. Furthermore, our simulation and test results suggest that complex battery models can be used to obtain better current estimation results. Although the estimation results of SOC using the OCV method are fairly good, the OCV need a high precision voltage sensing device, and need a better SOC-OCV curve for a better result.

#### References

- Dewangga, B. R. (2015). *Estimasi Arus pada Battery Management System Berbasis Sensorless Current Menggunakan Model Baterai Sederhana*. Yogyakarta: Andi Publisher.
- Chandran, V., K Patil, C., Karthick, A., Ganeshaperumal, D., Rahim, R., & Ghosh, A. (2021). State of charge estimation of lithium-ion battery for electric vehicles using machine learning algorithms. *World Electric Vehicle Journal*, 12(1), 38-49.
- Dewangga, B. R., Herdjunanto, S., & Cahyadi, A. (2018, August). Battery current estimation based on simple model with parameter update strategy using piecewise linear soc-ocv. In *2018 4th International Conference on Science and Technology (ICST)* (pp. 1-6). IEEE.
- Fathoni, G., Widayat, S. A., Topan, P. A., Jalil, A., Cahyadi, A. I., & Wahyunggoro, O. (2017, October). Comparison of State-of-Charge (SOC) estimation performance based on three popular methods: Coulomb counting, open circuit voltage, and Kalman filter. In *2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)* (pp. 70-74). IEEE.
- Ipek, E., Eren, M. K., & Yilmaz, M. (2019, August). State-of-charge estimation of li-ion battery cell using support vector

- regression and gradient boosting techniques. In *2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)* (pp. 604-609). IEEE.
- Jalil, A. (2017). *Active Cell Balancing Implementation to Lithium Polymer Battery Based on Pulse Width Modulation (PWM) Output Control*. Yogyakarta: Gava Media.
- Kupper, M., Brenneisen, J., Stark, O., Krebs, S., & Hohmann, S. (2018, June). Cascaded fractional kalman filtering for state and current estimation of large-scale lithium-ion battery packs. In *2018 Chinese Control And Decision Conference (CCDC)* (pp. 5071-5078). IEEE.
- Mujiarto, A. D., Komaro, M., Mohamed, M. A., Rahayu, D. S., Sanjaya, W. S. M., Mamat, M., & Sambas, A. (2019). Subiyanto, "Colored object detection using 5 Dof robot arm based adaptive Neuro-Fuzzy method," *Indonesian Journal of Electrical Engineering and Computer Science*, 13(1), 293-299.
- Putra, W. S., Dewangga, B. R., Cahyadi, A., & Wahyunggoro, O. (2015, November). Current estimation using Thevenin battery model. In *Proceedings of the Joint International Conference on Electric Vehicular Technology and Industrial, Mechanical, Electrical and Chemical Engineering (ICEVT & IMECE)* (pp. 5-9). IEEE.
- Skeete, J. P., Wells, P., Dong, X., Heidrich, O., & Harper, G. (2020). Beyond the EVent horizon: Battery waste, recycling, and sustainability in the United Kingdom electric vehicle transition. *Energy Research & Social Science*, 69, 101581.
- Tesla, Tesla Powerwall, Tesla, [Online]. Available: <https://www.tesla.com/powerwall>. [Accessed 23 November 2020].
- Vaidyanathan, S., Sambas, A., Mamat, M., & Sanjaya, W. M. (2017). A new three-dimensional chaotic system with a hidden attractor, circuit design and application in wireless mobile robot. *Archives of Control Sciences*, 27(4), 541-554.
- Wei, Z., Zhao, D., He, H., Cao, W., & Dong, G. (2020). A noise-tolerant model parameterization method for lithium-ion battery management system. *Applied Energy*, 268, 114932.
- Zhang, R., Xia, B., Li, B., Cao, L., Lai, Y., Zheng, W., & Wang, M. (2018). A study on the open circuit voltage and state of charge characterization of high capacity lithium-ion battery under different temperature. *Energies*, 11(9), 2408.
- Zhang, Z., Zhang, L., Hu, L., & Huang, C. (2020). Active cell balancing of lithium-ion battery pack based on average state of charge. *International Journal of Energy Research*, 44(4), 2535-2548.