논문 21-46-02-16 The Journal of Korean Institute of Communications and Information Sciences '21-02 Vol.46 No.02
https://doi.org/10.7840/kics.2021.46.2.314

# BiLSTM을 이용한 SoC와 SoP 공동 추정

요 니•, 안 젤 라*, 전 일 수*, 임 완 수°

# Co-Estimation of SoC and SoP Using BiLSTM

Kumbayoni Lalu Muh•, Angela C. Caliwag*, Il-Soo Jeon*, Wansu Lim°

요 약

본 논문은 리튬이온배터리의 안정성, 신뢰성과 효율성을 높이기위해 BiLSTM 기법을 이용한 전력상태 (SoP, state of power)-잔존용량 (SoC, state of charge) 공동 추정 기법을 제안하였다. SoP와 SoC는 리튬이온배터리의 성능에 직접적인 영향을 미치므로 SoP와 SoC 개별 추정이 아닌 두 파라미터의 상관 관계를 분석하고, 분석한 결과를 BiLSTM을 이용하여 학습한 후 공동 추정을 실시하였다. 제안한 알고리즘은 Urban Dynamometer Driving Schedule 데이터를 이용하여 검증하였으며, 실험 결과 공동 추정의 최소 평균제곱근오차(RMSE)는 14.66으로 기존 기법의 오차 20.46보다 약 28% 감소하였다.

**키워드 :** 전력상태, 잔존용량, 공동추정, BiLSTM, 리튬이론배터리
**Key Words :** State of power(SoP), State of charge(SoC), Co-estimation, BiLSTM, Lithium-ion battery

ABSTRACT

This paper proposes a Lithium-ion battery state-of-charge and state-of-power co-estimation algorithm. In states co-estimation algorithm, battery state-of-charge is considered in state-of-power estimation and vice versa. Unlike conventional methods, the proposed method takes into account the effect of the current battery state-of-power in one-step ahead state-of-charge estimation. Since battery states are not directly measurable, a bidirectional long-short term memory model is used to co-estimate the states using the measurable battery parameters (such as voltage and current). The model is trained and tested using Urban Dynamometer Driving Schedule. The results show that the co-estimating battery state-of-charge and state-of-power has higher accuracy (approximately 28.35%) than independent estimation.

## Ⅰ. Introduction

SoP is one of the important states of battery to ensure the safe operation of the battery. Since there is no sensor available to measure the SoP, it needs to be estimated[1]. Researches proposed estimation algorithms to estimate the SoP. Different general-purpose algorithms mostly used filtering algorithm such as Kalman filter[2], Particle filter[3,4], Extended Kalman filter[5], etc. Those algorithms can precisely estimate the states of the battery. Despite that, the computational cost is too high due to the

※ 본 연구는 금오공과대학교 학술연구비로 지원되었음(2018104075).
• First Author : Kumoh National Institute of Technology, Department of Electronic Engineering, 학생회원
° Corresponding Author : Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering, wansu.lim@kumoh.ac.kr, 정회원
* Kumoh National Institute of Technology, Department of Electronic Engineering, 학생회원
논문번호 : 202007-150-C-RN, Received July 13, 2020; Revised September 29, 2020; Accepted October 29, 2020

matrix multiplication. Other methods such as ampere-hour coulomb and open-circuit voltage methods are conducted due to the excellent performance in real-time, low computational cost, easy implementation, high robustness, and easy implementation. Nevertheless, drawbacks such as low accuracy and regular calibration requirement exist if we use these methods in the dynamic system. TheECM-based method has high accuracy and robustness. Nonetheless, it requires a highly accurate battery model otherwise it will not fully capture the battery characteristic if the battery model is not accurate[6]. It can be improved by considering the battery parameters such as voltage diffusion, voltage polarization, hysteresis, etc. Considering those parameters every time we need to estimate the state is time-consuming and computationally expensive. Machine learning method is well known for its strong adaptability and high accuracy[7]. On the other hand, it also has drawbacks such as high dependency on the order of time-series training data and high computational complexity. The dependency on the order of time-series training data and time-consuming parameter estimation can be solved by training the data forward and backward using BiLSTM.

In this paper, the BiLSTM is utilized to estimate battery SoC and to estimate the battery SoP using the estimated SoC. Using this method, an estimation of SoP can be obtained with lower computational cost compared to the ECM-based SoP estimation method.

### 1.1 Overview of BiLSTM

BiLSTM is one of the recurrent neural network (RNN) type, which is the enhancement of the LSTM[8-10]. In Fig. 1, BiLSTM architecture is shown[11]. It consists of the input sequence, BiLSTM layer, hidden layer, and output layer. The input sequence consists of a sequence of voltage and current. These data are loaded to the BiLSTM layer.

In the BiLSTM layer, the learning process is divided by 2: backward, forward. The weight is concatenated in the hidden layer. The output consists of optimum SoC at some time interval. BiLSTM

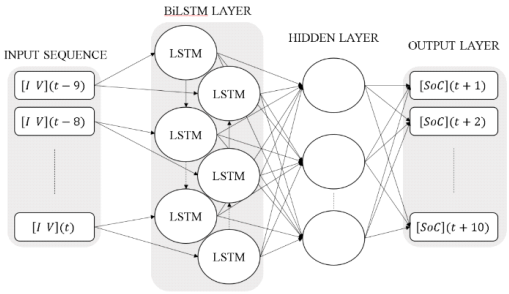


Fig. 1. The architecture of BiLSTM on estimating the SoC case

connects hidden layers forward and backward to the same output. It also has the same capability with LSTM wherein it was used to model nonlinear systems in the BMS model in this study. BiLSTM is incorporated in this study for two reasons: (1) To account for the dynamic and non-linearity of the data and (2) to apply the multi-step forecasting in order to forecast the SoP for the next time interval. As like LSTM, BiLSTM requires a huge amount of data in order to avoid overfitting. This drawback of BiLSTM can be resolved by limiting the iteration of the training, or by applying some regularization method such as dropout[12]. The LSTM has a good performance for forecasting the SoC using measurable values since it can capture long-distance dependencies better. However, LSTM can only capture the forward behavior of the quantity of the battery and sometimes the nonlinearity of the battery quantity behavior cannot correctly be represented only by having the historical information.

## Ⅱ. Proposed SoC and SoP Co-estimation Methodology

The co-estimation of battery SoC and SoP proposed in this study is shown in Fig. 2.

In step 1: the battery SoC is estimated using BiLSTM, with current and voltage as the explanatory variables.

In step 2: the battery OCV is identified using the normal linear interpolation and extrapolation.

In Step 3 and Step 4: the maximum discharge current within the voltage and SoC constraints is

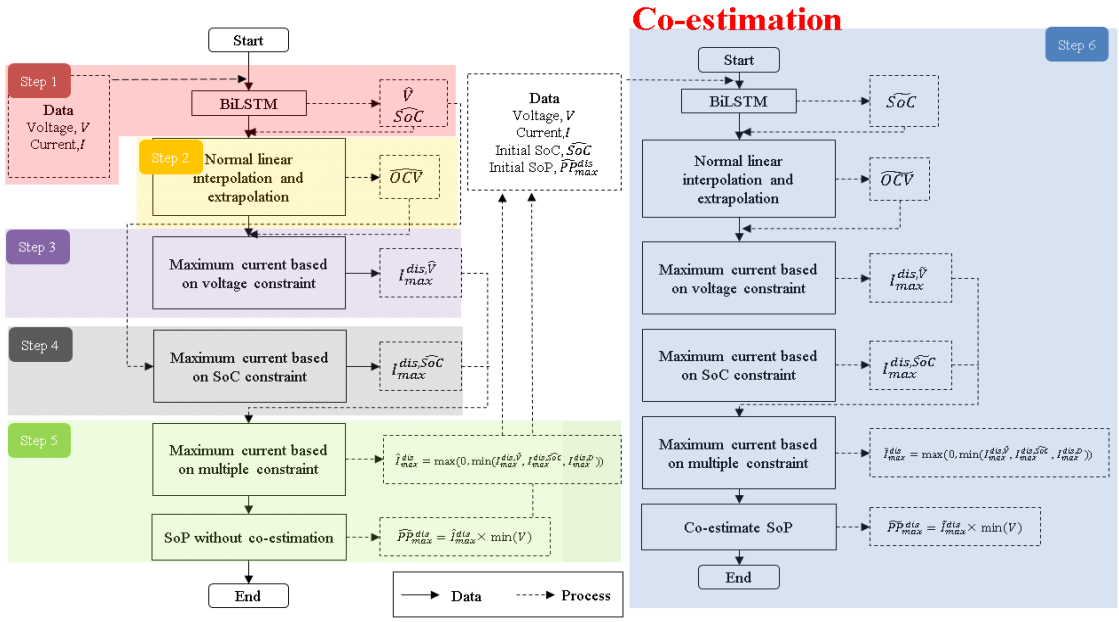


Fig. 2. Flow chart of battery SoC and SoP co-estimation using BiLSTM

predicted by applying the OCV and the estimated SoC respectively.

In step 5: the SoP is calculated by getting the minimum of maximum discharge current based on multiple constraints and multiplied it with minimum design voltage.

In step 6: the initial SoC and initial SoP are used as the explanatory variables to update the estimate SoC. In this step, it is assumed that a previously obtained SoP is available.

The process from step 2 until step 5 is operated with updated SoC. The co-estimation utilizes the relationship between the SoC and SoP. This estimation is more accurate than estimating the states individually using different algorithms. The advantages of using the co-estimation are the list as follows:

1. The impact of the initial estimates of SoC and SoP is taken into account. The SoC estimation is updated with the result of previous SoP estimation, thus improving the accuracy of the SoC estimation.

2. The impact of SoC on the SoP is taken into account. The OCV has a direct relationship with the SoC. In addition to that, the initial SoP also influences the estimated SoC. That is, the value of the OCV is updated with the value of the updated SoC. With the updated SoC and OCV, the accuracy of the SoP estimation can be significantly improved.

SoP describes the maximum peak of power that can be distributed without violating some design constraints. These design constraints are the factors considered in restricting and maintaining the discharging power at a certain range. The constraints are classified into two: 1) constraints that need to be estimated (such as voltage constraint, and SoC constraint), and 2) constraints that are already set by the battery manufacturer called design constraints (such as power constraint and current constraint).

### Step 1: SoC and Voltage Response Estimation

Optimum SoC information used as the benchmark, and the BiLSTM is expressed as:

$$D=\begin{bmatrix} \Psi(t-9)...\Psi(t) & SoC(t)...SoC(t+9) \\ \Psi(t-8)...\Psi(t) & SoC(t)...SoC(t+10) \\ \Psi(n-9)...\Psi(n) & SoC(n)...SoC(n+9) \end{bmatrix} \quad (1)$$

$SoC(t)$ is the optimum $SoC$ at time $t$ and $\psi(t)$ is the input that consists of vector, also at time $t$. The input vector is defined as $\psi(t)=V(t),I(t)$ where $V(t),I(t)$ are the voltage and current at time $t$, respectively. In our UDDS data, we have both

charging and discharging current. To simplify the study, we considered only the battery discharge. The negative values, which are the values below zero, represents charging current while the values above zero are the discharging current. In order to get the optimum forecasting result, the battery cell current in the UDDS drive cycle data was normalized with the minimum value set to zero and maximum value to set to one. That is,

$$z(t) = x(t) - \min(x) / \max(x) - \min(x) \quad (2)$$

where $x = (x(t)...(n))$ and $z(t)$ is the normalized data at time $t$. The battery cell current is forecasted by using:

$$\begin{bmatrix} I(t-10) \\ V(t-10) \end{bmatrix}, \begin{bmatrix} I(t-9) \\ V(t-9) \end{bmatrix} \cdots \begin{bmatrix} I(t-1) \\ V(t-1) \end{bmatrix} \quad (3)$$

to obtain

$$\widehat{SoC}(t+1),\ \widehat{SoC}(t+2) \ldots\ldots \widehat{SoC}(t+5) \quad (4)$$

**Step 2: Normal Linear Interpolation and Extrapolation**

In Step 2, the open-circuit voltage is computed. It consists of (1) the process of normal linear interpolation and extrapolation using estimated SoC and 2) the output of the process which is the predicted OCV. The principle of the function is straightforward. To compute the OCV at any SoC, we perform table lookup processes using estimated OCV.

$$OCV(SoC, T) = OCV_0(SoC) + T \times OCV \quad (5)$$

The special cases where the SoC is out of range also needs to be handled. The lookup table has to provide a reasonable answer for the SoC that is not between minimum and maximum SoC. In that case, it must perform what so-called normal linear extrapolation[13].

We can compute the OCV with the equation as follows [13]:

$$OCV = (SoC - SoC_k)\frac{OCV_{k+1} - OCV_k}{SoC_{k+1} - SoC_k} + OCV_k \quad (6)$$

where $SoC_k$ is the closest lower value of $SoC$, and $SoC_{k+1}$ is the closest higher value of $SoC$. If we need to extrapolate the data outside the table, the equation is essentially similar to before except we take the slope instead of taking the two surrounding points. Since there are no surrounding points, we take the first two data points in the table to find the slope. Except that, everything in Eq. (6) is the same. That is,

$$OCV = (SoC - SoC_0)\frac{OCV_1 - OCV_0}{SoC_1 - SoC_0} + OCV_0 \quad (7)$$

To extrapolate the other end of the table, we take the closest two points at the end of the table:

$$OCV = (SoC - SoC_N)\frac{OCV_N - OCV_{N-1}}{SoC_N - SoC_{N-1}} + OCV_N \quad (8)$$

**Step 3: Maximum Current based on Voltage Constraint**

For the voltage constraint, assume that we have a very simple cell model. In this model, we set the voltage as the difference between the open-circuit voltage evaluated at the present SoC and the product of cell current and the series resistance of the cell.

$$V(t) = OCV(SoC(t)) - I(t)R \quad (9)$$

We can rearrange this expression to compute the current. The battery cell current is equal to the OCV minus terminal voltage all divide by resistance.

$$I(t) = \frac{(OCV(SoC(t)) - V(t))}{R} \quad (10)$$

The maximum discharge current limited by voltage is calculated as [14]:

$$I_{\max,n}^{dis,V} = \frac{OCV(SoC(t)) - \min(V))}{R_{dis,} \triangle T} \quad (11)$$

**Step 4: Maximum Current based on SoC Constraint**

For the SoC constraint, we need to consider the case of constant current over the time interval.

$$SoC(t+\triangle T) = SoC(t) - (\triangle T/Q) I_n \quad (12)$$

The SoC after time interval is equal to the difference between the current SoC and the product of the current and the time interval divide by the capacity. This can be replaced with the SoC that is estimated in step 1.If the design limits of the future SoC is always with in the minimum value of SoC and a maximum value of SoC, the current is computed by reversing Eq. (12). Thus, enforcing the design limits[14].

$$I_{\max}^{dis,V} = (SoC(t) - \min(SoC)) \triangle T/Q \quad (13)$$

**Step 5: Maximum Current based on Multiple Constraint**

To compute the peak power, we assume that we are concerned only with keeping the terminal voltage of the cell between $\max(V)$ and $\min(V)$.The power constraint is the maximum absolute power permitted by the load. This can be calculated by using [14]:

$$PP_{\max}^{dis} = n_s n_p \min(V) \frac{\max(SoC) - \min(SoC)}{\triangle T/Q} \quad (14)$$

Peak power is set as unlimited since it does not consider enforcing design limits on SoC, future SoC, and the maximum absolute current that designated by electronic.

We can combine all of those limits enforced at the same time. Discharging the current at a level that's higher than either one of these computations is not allowed, because if we do so, one of the design limits will be broken. Thus, leading to cell damage. The maximum discharge current within the limits is expressed as:

$$I_{\max}^{dis} = \max(0, \min(I_{\max}^{dis,SoC}, I_{\max}^{dis,V}, I_{\max}^{dis,D})) \quad (15)$$

Based on this expression, we obtained the current that is closest to zero among all of the limiting current: based on design limit, SoC constraints, and voltage constraints. To compute the maximum charge current, a similar equation is used. The charging current is always negative. When we want to find the value that is closest to zero, the maximum value is used and assigned sense instead of the minimum value in order to avoid the violation of any limits. The pack power can be computed by multiplying the pack current with the minimum design voltage of every cell in the battery pack. It compute power by multiplying the limiting factor that's based on maximum load power design limit $PP_{\max}^{dis}$ and multiply the number of cells in series to get the overall battery pack power level. Each of the future voltage may be different because of the different initial SoC and so forth.

$$PP_{\max}^{dis} = \min(PP_{\max} \sum_{n=1}^{n_s} I_{\max}^{dis}(\hat{V}(t+\triangle T))) \quad (16)$$

The SoC estimation and SoP estimation are based on the battery model that has been collected from the HPPC. The SoP estimation is based on the estimated SoC. Over the whole SoC range, the model parameter of HPPC is identified by the BiLSTM algorithm.

**Step 6: Co-Estimation**

When the maximum discharging current is reduced and the maximum charging current is maximized, SoC is close to its lower design limit. On the other hand, when the maximum discharging current is maximized and the maximum charging current is reduced, SoC is close to its higher limit. By mapping the sequences between input and output, it is able to represent nonlinear dynamic systems. To apply the BiLSTM on the SoC estimation, the dataset used to train the network is

given by:

$$D=\begin{bmatrix} \gamma(t-9)...\gamma(t) & SoC(t)...SoC(t+9) \\ \gamma(t-8)...\gamma(t) & SoC(t)...SoC(t+10) \\ \gamma(n-9)...\gamma(n) & SoC(n)...SoC(n+9) \end{bmatrix} \quad (17)$$

$SoC(t)$ is the optimum SoC at time $t$ and $\gamma(t)$ is the input that consists of vector at time $t$. The input vector is defined as $\gamma(t)=[V(t), I(t), \widehat{SoC}(t), PP^{dis}_{\max}(t)]$, where $V(t), I(t), \widehat{SoC}(t), PP^{dis}_{\max}(t)$ are the voltage, current, initial estimated SoC and initial estimated SoP at time , respectively. In the co-estimation process, it is assumed that a previously obtained SOP is already available. Otherwise, initial value of SOP has to be estimated using the conventional SoC. The new SoC obtained is used to compute the new OCV by using Normal linear interpolation and extrapolation Eqs. (6), (7), and (8), in step 2. Co-estimated OCV is used to compute the maximum current based on voltage constraint as in Eq. (11). Parallelly, maximum discharge current based on SoC constraint is estimated using Eq. (13). The co-estimated SoP can be obtained by using Eq. (16).

## Ⅲ. Simulation Results

### 3.1 Voltage responses estimation results: Step 1

A BiLSTM is able to represent nonlinear dynamic systems by mapping the sequences between input and output. To apply the BiLSTM on voltage response estimation, the dataset used to train the network is given by:

$$D=\begin{bmatrix} \Psi(t-9)...\Psi(t) & V(t)...V(t+9) \\ \Psi(t-8)...\Psi(t) & V(t)...V(t+10) \\ \Psi(n-9)...\Psi(n) & V(n)...V(n+9) \end{bmatrix} \quad (18)$$

where $V(t)$ is the measured voltage value at time $t$ $t$and $\psi(t)$ is the input vector at time $t$. The input vector is defined as $\psi(t)=[V(t), I(t)]$where $V(t), I(t)$ are the voltage and current at time $t$, respectively. To obtain the estimated voltage $V$ at time $t$ until time $t+9$, a linear transformation is performed using a final fully connected layer on the hidden state tensor $h(t)$. This is done by follows:

$$V(t)...(t+9)=M_{out}h(t)+b \quad (19)$$

where $M_{out}$ and $b$ are the fully connected layer's weight matrix and biases, respectively.

Forecasting can be performed using single values or multiple values. The forecasting using single value is called univariate forecasting while forecasting using multiple values is called multivariate forecasting[15]. To verify the accuracy of BiLSTM in forecasting the battery voltage response, we applied BPNN and LSTM. To make an equal comparison, battery voltage and battery current are used as explanatory variables. In the BPNN architecture, instead of 100 nodes, 312 nodes are used since it is the most optimal number of nodes used in [16].

Table 1 shows the RMSE result of forecasting voltage response using each of the methods in the UDDS dataset. As we can see, BiLSTM outperforms BPNN with a difference of 0.455404 RMSE, while BiLSTM outperforms LSTM with a difference of 0.6378%. Fig. 3 shows all of the voltage actual measurements and estimated voltage with BPNN, LSTM, and BiLSTM. Blackline represents the actual measurements; yellow line with dashes represent BPNN; blueline with dashes represent BiLSTM; and magenta line with dashes represent LSTM. It can be seen that the whole of the estimated values is following the actual data.

For further observations, we zoomed the data and present it in the subfigure. In the zoomed figure we can see that BPNN has the furthest prediction compared to BiLSTM and LSTM. However, between LSTM and the actual, and BiLSTM and the

Table 1. Voltage response fore casting RMSE using existing machine learning approach

| Machine Learning | RMSE |
|---|---|
| BPNN[16] | 0.504010 |
| LSTM[17] | 0.048916 |
| BiLSTM | 0.048606 |

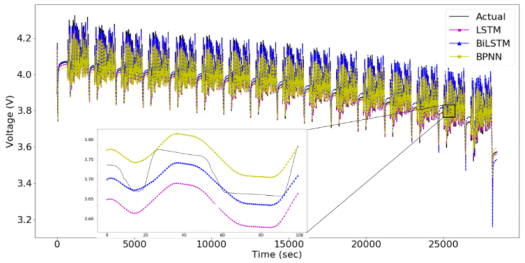


Fig. 3. Actual battery voltage response compared with Forecasted battery voltage response using another machine learning technique

actual. We can see close predictions between both of the predictions. This is listed in Table 1. That is, the RMSE of the prediction using LSTM is not that different from the RMSE of prediction with BiLSTM.

The voltage response is forecasted by using:

$$\begin{bmatrix} V(t-10) \\ I(t-10) \end{bmatrix}, \begin{bmatrix} V(t-9) \\ I(t-9) \end{bmatrix} \cdots\cdots \begin{bmatrix} V(t-1) \\ I(t-1) \end{bmatrix} \quad (20)$$

Obtaining:

$$\hat{V}(t+1), \hat{V}(t+2) \cdots\cdots . \hat{V}(t+5) \quad (21)$$

To provide points of comparison, we took the RMSE between the forecasted battery voltage response and the actual battery voltage response. The RMSE of battery voltage response estimation for UDDS is 0.048606 RMSE.

### 3.2 Initial SoC estimation results: Step 1

The RMSE between the estimated SoC using BiLSTM and the optimum SoC is 0.008054. This is 21.74% less than estimating SoC using LSTM which only obtained the RMSE of 0.010292. The SoC of the cell increased by about five percent during each cycle. We assume that the vehicle was driving downhill most of the way so it was charge adding cycle. Between the cycles, the SoC decreased to about ten percent with intervening discharge pulses. Overall, the entire normal operating range of this cell is between ten percent and ninety percent SoC.

Fig. 4 shows the comparison between the estimate of SoC using HPPC, LSTM, BiLSTM, and BPNN

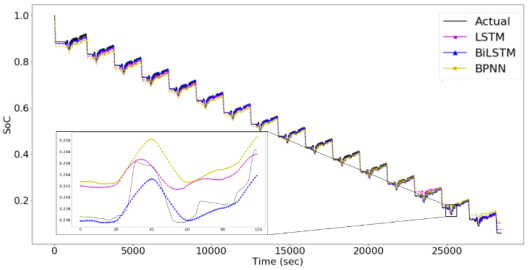


Fig. 4. Battery cell current comparison between the HPPC SoC and the BiLSTM estimated SoC

method. There are a total of 16 cycles. In each cycle, the SoC is increased around 5% until it finally drops by around 10% on the cycle transition until it fully discharged. 70% of the first data are used for the training. We took the data from 25200 to 25500 to be analyzed. As we can see here the estimate BiLSTM consistently followed the HPPC estimate data which we considered as the ground truth. LSTM estimated around 150 to 240 is below the benchmark while the BPNN is over the benchmark. BiLSTM itself seems to be overestimated with a small gap. Based on the figure, we can conclude that BiLSTM outperforms the other machine learning estimator.

### 3.3 Maximum Current based on Multiple Constraint results: Step 5

Fig. 5 is shown to validate that BiLSTM achieved good accuracy in estimating the SoP. Another machine learning method such as BPNN[16] and LSTM[17] are applied. Notice that 70% of first data are similar since these data use also as the training. The peak power follows the peak power based on voltage in the beginning. However, after around six and a half hours, the SoP starts to follow the peak

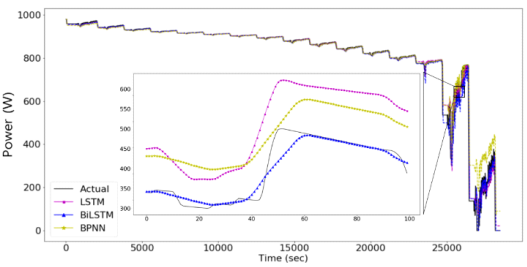


Fig. 5. Comparison between SoP estimated by BiLSTM, HPPC, LSTM and BPNN

Table 2. RMSE of Initial SoP using an existing machine learning approach

| Machine Learning | RMSE |
|---|---|
| BPNN[16] | 43.424 |
| LSTM[17] | 21.314 |
| BiLSTM | 20.455 |

power based on SoC. If we look closer to the zoomed figure, the BiLSTM has a closer line to the benchmark compared to other machine learning. The BPNN and LSTM prediction are over the HPPC estimated until time 160 in the figure. We somehow expect that the estimated result is under the benchmark since we don't want the estimated SoP higher than the actual SoP which can damage the cell of the battery. The RMSE comparison is listed in Table 2.

BiLSTM obtained RMSE of 20.455, the lowest among the machine learning mentioned in Table 2 which are BPNN and LSTM. BPNN obtained RMSE of 43.424, which reflects that BPNN accuracy is not better than LSTM and BiLSTM. Based on voltage response, SoC and SoP estimation results are listed in Tables 1 and 2. The BiLSTM overcome the LSTM by 1-20% less error. Due to these results, co-estimation is conducted using only BiLSTM.

### 3.4 Co-Estimation Results: Step 6

As we can see in Fig. 6, the co-estimation result is closer to the HPPC which we used as the ground truth in this case. This co-estimated SoC is used to compute the OCV using Eqs. (6), (7), and (8). The new OCV is referred as the co-estimated OCV. Using the co-estimated OCV and co-estimated SoC, maximum discharge current based on voltage and SoC constraints are estimated. The minimum from the maximum discharge current based on voltage, SoC, and design constraint is taken.

Fig. 7 shows the co-estimation results compared to estimation results without the co-estimation and HPPC. Co-estimated refers to the SoP values after the co-estimation process. Estimated refer to the SoP values without the co-estimation process. We can see that both, the co-estimated and the estimated algorithm results follow the HPPC curve.

In Fig. 8 we zoomed the data to be analyzed. The data is taken from around time 25400 seconds in the data. The co-estimated data overcome the estimated data without co-estimation. We can see that around the small pulse in the data 50th to 150th, co-estimated SoP is close to the HPPC estimate. Nevertheless, the estimate outside the range mentioned earlier is having good accuracy also. It means that estimating using HPPC and BiLSTM co-estimation has met the agreement. The large gap between HPPC and the others is due to the poor SoC estimation using HPPC. Poor SoC estimation using SoC is shown in Fig. 6. The HPPC method does not consider the dynamic behavior of the battery and assumes that the battery is always the

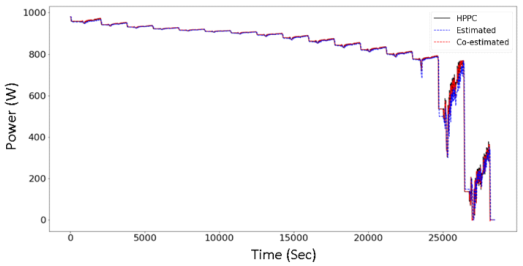


Fig. 7. Co-estimated SoP using BiLSTM

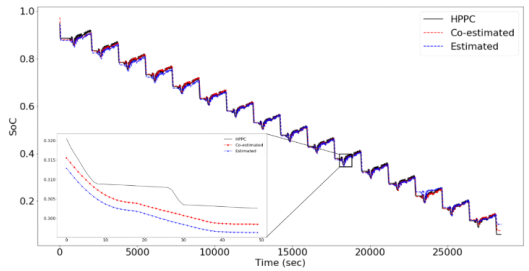


Fig. 6. Maximum Current based on multiple constraint

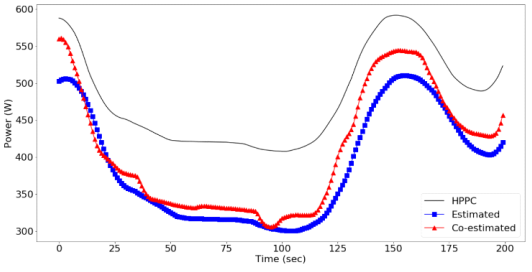


Fig. 8. Co-estimated SoP using BiLSTM zoomed figure

Table 3. SoC and SoP Co-estimation final results

| Data training | SoC without co-estimation | SoP without co-estimation | Co-estimate SoC | Co-estimate SoP |
|---|---|---|---|---|
| 30% | 0.030 | 26.01 | 0.0102 | 20.48 |
| 60% | 0.012 | 22.35 | 0.0029 | 15.04 |
| 70% | 0.008 | 20.46 | 0.0024 | 14.66 |

same. However, this is not the case. On the other hand, the other methods were able to learn and adapt to the changes in the behavior and characteristics of the battery.

Table 3 lists the overall results of the estimation. Using HPPC SoP as a benchmark, the proposed framework of the co-estimation algorithm can significantly increase the accuracy by 28.35 % compared to the previous framework that excludes the initial SoC and SoP as the explanatory variable. The RMSE of SoC is 0.0024 by using 70% of the first data as training. SoP using the co-estimation algorithm has 14.66 RMSE. Table 6 shows the initial states and final state estimation result using 30%, 60%, and 70% data as training. It is clear that the training using more data results in better accuracy of the estimate.

## Ⅳ. Conclusion

The result of this paper addresses the computational cost problem, and the dynamicity of the states of battery. SoC and SoP are both estimated. We have studied the SoP estimations based on multiple constraints, in two phases. Initial SoC and SoP joint estimation, and SoC and SoP co-estimation. First, in the initial SoC and SoP estimation, we estimate the SoP using the SoC that is estimated using BiLSTM to capture the conceptual information from the measurable values. Using the initial estimated SoC, we compute the OCV using linear interpolation for the terminal voltage within the SoC range, and linear extrapolation for the terminal voltage outside the SoC range. The OCV is used to calculate the maximum current based on voltage constraint. The SoC is also used to calculate the maximum current by eq (20). Second, in the SoC and SoP co-estimation, the initial estimated SoP and the initial estimated SoC used as additional information alongside the current and voltage to estimate the final SoC. This final SoC uses to compute the OCV again to calculate the maximum current based on voltage constraint. And the SoC use to calculate the maximum current based on the SoC constraint. The final SoP obtained by getting the minimum of maximum current based on voltage, SoC, and design limit constraint multiplied by minimum design voltage.

## References

[1] P. Shen, M. Ouyang, L. Lu, J. Li, and X. Feng, "The co-estimation of state of charge, state of health, and state of function for lithium-ion batteries in electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 92-103, 2018.

[2] R. Xiong, H. He, F. Sun, X. Liu, and Z. Liu, "Model-based state of charge and peak power capability joint estimation of lithium-ion battery in plug-in hybrid electric vehicles," *J. Power Sources*, vol. 229, pp. 159-169, 2013.

[3] A. El Mejdoubi, H. Chaoui, H. Gualous, P. Van Den Bossche, N. Omar, and J. Van Mierlo, "Lithium-ion batteries health prognosis considering aging conditions," *IEEE Trans. Power Electron.*, vol. 34, no. 7, pp. 6834-6844, 2019.

[4] D. B. De Alencar, C. De Mattos Affonso, R. C. L. De Oliveira, J. L. M. Rodríguez, J. C. Leite, and J. C. R. Filho, "Different models for forecasting wind power generation: Case study," *Energies*, vol. 10, no. 12, 2017.

[5] C. Montella, "The kalman filter and related algorithms: A literature review," *Res. Gate*, pp. 1-17, May 2014.

[6] J. Lu, Z. Chen, Y. Yang, and Ming L. V., "Online estimation of state of power for lithium-ion batteries in electric vehicles using genetic algorithm," *IEEE Access*, vol. 6, pp.

20868-20880, 2018.

[7] C. Huang, Z. Wang, Z. Zhao, L. Wang, C. S. Lai, and D. Wang, “Robustness evaluation of extended and unscented kalman filter for battery state of charge estimation,” *IEEE Access*, vol. 6, pp. 27617-27628, 2018.

[8] W. J. Jiang, N. Zhang, P. C. Li, and N. Chen, “A temperature-based peak power capability estimation method for lithium-ion batteries,” *Procedia Eng.*, vol. 187, pp. 249-256, 2017.

[9] X. Liu, C. Zheng, J. Wu, J. Meng, D. I. Stroe, and J. Chen, “An improved state of charge and state of power estimation method based on genetic particle filter for lithium-ion batteries,” *Energies*, vol. 13, no. 2, 2020.

[10] I. Arasaratnam, “Novel battery SOC/SOP/SOH estimation algorithms in a unified framework,” in *Proc. Arasaratnam*, 2014.

[11] C. Vidal, P. Malysz, P. Kollmeyer, and A. Emadi, “Machine learning applied to electrified vehicle battery state of charge and state of health estimation: State-of-the-Art,” *IEEE Access*, vol. 8, pp. 52796-52814, 2020.

[12] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” *ICLR 2015*, pp. 1-8, Feb. 2015.

[13] C. Unterrieder, M. Lunglmayr, S. Marsili, and M. Huemer, “Battery state-of-charge estimation using polynomial enhanced prediction,” *Electron. Lett.*, vol. 48, no. 21, p. 1363, 2012.

[14] S. Xiang, G. Hu, R. Huang, F. Guo, and P. Zhou, “Lithium-ion battery online rapid state-of-power estimation under multiple constraints,” *Energies*, vol. 11, no. 2, p. 283, 2018.

[15] G. Papacharalampous, H. Tyralis, and D. Koutsoyiannis, “One-step ahead forecasting of geophysical processes within a purely statistical framework,” *Geosci. Lett.*, vol. 5, no. 1, pp. 1-19, 2018.

[16] M. A. Hannan, M. S. H. Lipu, A. Hussain, M. H. Saad, and A. Ayob, “Neural network approach for estimating state of charge of lithium-ion battery using backtracking search algorithm,” *IEEE Access*, vol. 6, pp. 10069-10079, 2018.

[17] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, “Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6730-6739, 2018.

**요 니** (Kumbayoni Lalu Muh)
2018년 3월~현재 : 금오공과대학교 전자공학부 석사
<관심분야> 인공지능, 빅데이터 분석

**안 젤 라** (Angela C. Caliwag)
2019년 8월 : 금오공과대학교 전자공학과 석사
2019년 10월~현재 : 금오공과대학교 연구원
<관심분야> 회로 설계, 인공지능

**전 일 수** (Il-Soo Jeon)
1995년 3월 : 경북대학교 전자공학과 박사
2004년 3월~현재 : 금오공과대학교 전자공학부 교수
<관심분야> 보안, 네트워크

**임 완 수** (Wansu Lim)
2010년 8월 : GIST 정보통신공학과 박사
2014년 9월~현재 : 금오공과대학교 전자공학부 교수
<관심분야> 지능형 제어, 임베디드 시스템
[ORCID:0000-0003-2533-3496]