Method of voltage setting for power battery simulator using successive nearest-neighbor interpolation

Jiliang Yi, Huabing Tan, Jin Zhang and Zhongqi Li

Abstract
The power battery simulator is important equipment in new energy vehicle test platform and other industrial fields, and the battery model is the key to emulate the battery characteristics accurately. A novel method of given voltage for the power battery simulator is proposed in the paper. According to the state of charge (SOC) of the power battery, three sub-tables with different resolutions are established corresponding to the initial, stationary, and final stage of SOC. To ensure the accuracy of a given voltage under small data capacity, a successive nearest-neighbor interpolation (SNNI) algorithm is proposed for processing battery model data and the iterative calculation is carried out for the data obtained by looking up the table until it approaches the required value. Simulations and experiments are done on various capacity data tables, which are obtained by employed different sampling ratios to reduce the data capacity, and the influence of iteration number selection on the accuracy of the algorithm is discussed. The results show that, after 4–5 iterations, the given voltage accuracy is controlled within 0.03 V by the SNNI algorithm with only 1.6% capacity of the standard model table, which is greatly improved compared with the traditional method, and the feasibility is verified by executing the proposed algorithm in a DSP system.

Keywords
look-up table, nearest-neighbor interpolation, power battery, state of charge, given voltage

Introduction
With the increasing pressure of energy crisis and environmental pollution, new energy vehicles, especially pure electric vehicles, have become one of the important solutions. Some governments announced that they will completely ban the sale of traditional fuel vehicles within the next 20 years. China has also launched an exit plan for traditional fuel vehicles. The new energy automobile industry has been identified as one of the key development directions of strategic emerging industries.

To achieve the breakthrough of the key technology of electric vehicles, the electric vehicle manufacturers need advanced experimental conditions, and the test platform is the preferred way, which combines the advantages of software simulation and real vehicle test. The software simulation can shorten the development time and reduce the demand for hardware resources, which is suitable for application in the stage of system modeling and algorithm verification. Although the real vehicle test mode is closest to the real environment, it is not flexible and costly. The test platform can not only simulate the actual working conditions of the real vehicle to evaluate the performance parameters of the whole vehicle, but also test the key components. It is of great significance to reduce the test cost and shorten the development cycle.

Using a battery simulator instead of an ordinary DC power supply can overcome some shortcomings such as high cost and inconvenience. More importantly, it can simulate the characteristics of power battery which are not possessed by the ordinary DC power supply. So the power battery simulator is indispensable equipment for electric vehicle test platforms.

Modeling is the basic step of all kinds of vehicle test platform, which can be divided into static model and dynamic model. Compared with the static model, the dynamic model can reflect more details of the simulated object, but it is also relatively more complex. The battery model, which directly
affects the performance of the battery simulator, is the key factor to realize the high fidelity volt-ampere characteristic of the battery.\textsuperscript{10} The main task of building the battery model is to set the reference voltage of the battery simulator according to the SOC and the load current.

The existing methods of establishing the battery model include (1) Using a standard battery model. Its accuracy is high, but the model contains exponential function, which is difficult to realize in the chip, so it is not suitable for the dynamic system.\textsuperscript{11} (2) Fitting the volt-ampere characteristic curve of the battery by section. Although the calculation of the method is reduced, the error increases significantly. The higher-order equation is used to improve the accuracy, but it is necessary to fit multiple curves under the condition of a large range of working current. Therefore, it results in large computation and long algorithm time and difficult to keep up with the real-time dynamic response of the system.\textsuperscript{12,13} (3) Look-up table: It just obtains the battery characteristic data from the standard table by calculating the standard mathematical model in advance or the experimental data. The method can calculate quickly, but there is a contradiction between accuracy and data capacity when directly using the look-up table data.\textsuperscript{14,15}

The look-up table method is easy to implement and widely used in practice. However, in order to reduce the storage cost, low-resolution data are usually used, so an interpolation algorithm is often used. Interpolation algorithm is generally used in the field of image processing, such as single-image super-resolution\textsuperscript{16,17} and image reconstruction technique.\textsuperscript{18} The traditional methods such as the nearest-neighbor algorithm\textsuperscript{19} and the bilinear interpolation algorithm\textsuperscript{20,21} are either not accurate enough or too much computation. Therefore, it is necessary to develop an interpolation algorithm with good performance for reducing the data sample and calculation under the premise of high-accuracy.

In this paper, a new method named successive nearest-neighbor interpolation (SNNI) is proposed to give the output voltage of the power battery simulation system. SNNI algorithm is employed to refine the resolution of the battery model table, so as to improve the precision of given voltage. The key processing of SNNI is iterative operation, which is widely used in many fields, such as optimal bidding strategy for power market,\textsuperscript{22} configuration of autonomous CHP system,\textsuperscript{23} model parameter estimation of the PEMFCs,\textsuperscript{24} and robust control.\textsuperscript{25,26} In an iterative way, a large extraction ratio can be adopted for the standard power battery model table to reduce the capacity of the model table; thus, the storage capacity requirement of the battery simulation system is reduced. The effectiveness of this method is proved by simulations and experiments, and the simulation results show that the proposed algorithm can control the absolute error within 0.03 V with only 1.6% capacity of the standard model table.

### Background knowledge

#### Power battery simulator

The typical power battery simulator includes a bidirectional PWM rectifier and a bidirectional DC/DC converter. The controller of the bidirectional PWM rectifier in Figure 1 controls the rectifier unit to obtain a stable DC voltage, while the bidirectional DC/DC controller is used to control...
the output port voltage that conforms to the volt-ampere characteristics of the power battery.

As shown in Figure 1, the output voltage of the bidirectional DC/DC converter is controlled to accurately track the given voltage \( U^* \) of the battery model. Therefore, the performance of the battery simulator is determined by the voltage tracking ability of the controller and the tracking performance of the battery simulator is determined by the given voltage \( U^* \) of the battery model. Therefore, the directional DC/DC converter is controlled to accurately track the given voltage \( U^* \) of the battery model tables, and the SNNI interpolation algorithm.


Figure 3. Workflow diagram.

Table 1. Rules for zoning.

| Zone | Rules |
|------|-------|
| 1    | \( \text{scc} < \text{SOC}(m) + d\text{SOC}(m)/2 \) \& \( i < (i(m) + d(i(m))/2 \) |
| 2    | \( \text{scc} > \text{SOC}(m) + d\text{SOC}(m)/2 \) \& \( i < (i(m) + d(i(m))/2 \) |
| 3    | \( \text{scc} < \text{SOC}(m) + d\text{SOC}(m)/2 \) \& \( i > (i(m) + d(i(m))/2 \) |
| 4    | \( \text{scc} > \text{SOC}(m) + d\text{SOC}(m)/2 \) \& \( i > (i(m) + d(i(m))/2 \) |

Common interpolation method

The battery model tables are built based on the estimated SOC and the sampled current \( i \). They are the indexes employed to locate the corresponding output voltage values in the model tables, which are obtained from the standard battery models or the test data. The system can only save limited data because of the limited memory, so the reduced model tables are set up to meet the memory saving requirements, and the traditional interpolation algorithm is improved to meet the accuracy requirements.

SNNI algorithm

From the above introduction, it can be seen that the calculation burden of the nearest-interpolation algorithm is light, but it can only output the voltage value in the model table, which will cause the output voltage of the battery simulator to be discontinuous and inaccuracy. The bilinear interpolation algorithm can give a smooth output voltage, but the calculation burden is heavy. In this paper, based on the nearest-interpolation algorithm, multiple iterative operations are used to make the output voltage value close to the output voltage value of the original model data, which is very close to the actual battery output voltage value. Because the method approximates the actual value by iteration, it is named as successive nearest-neighbor interpolation (SNNI) algorithm.

Workflow

Figure 3 is the flow chart of the voltage given method for power battery simulator in this paper. As can be seen from Figure 3, it includes three parts: SOC estimation, power battery model tables, and the SNNI interpolation algorithm. Among them, SOC estimation is realized by using amperetime method according to the initial value \( \text{SOC}^* \) and the load current \( i \).

Considering that the output voltage changes rapidly when the power battery SOC in the stage of beginning and end, the model table is divided into three sub-tables. They are as follows: when \( \text{soc} \geq 85\% \), it is called model sub-table 1; and when \( 85\% > \text{soc} \geq 20\% \), it is called model sub-table 2; and when \( \text{soc} < 20\% \), it is called model sub-table 3. Since the coordinate position of the \( U_1 \) is the shortest distance from the \( U \) point, it is assigned to the point as the output value.

The bilinear interpolation is a linear interpolation in two directions by using the four adjacent voltages in the model table. Taking Figure 2 as an example, the four adjacent \( U \) points in the model table are \( U_1 \) (\( i_1, \text{SOC}_1 \)), \( U_2 \) (\( i_1 + d(i)/1, \text{SOC}_1 \)), \( U_3 \) (\( i_1, \text{SOC}_1 + d\text{SOC}_1 \)), and \( U_4 \) (\( i_1 + d(i)/1, \text{SOC}_1 + d\text{SOC}_1 \)), respectively. To obtain the voltage value of the point \( U (i, \text{SOC}) \), two linear interpolations are carried out in the \( i \) direction and the SOC direction. The specific calculation process and related formulas are illustrated in Ref. 15.
Table 2. \( U_x(m) \) update method.

| Zone 1 | Zone 2 | Zone 3 | Zone 4 |
|--------|--------|--------|--------|
| Un \((m+1)\) | Unchanged | Formula(2) | Formula(3) | Formula(5) |
| U2 \((m+1)\) | Formula(1) | Unchanged | Formula(5) | Formula(3) |
| U3 \((m+1)\) | Formula(4) | Formula(5) | Unchanged | Formula(1) |
| U4 \((m+1)\) | Formula(5) | Formula(4) | Formula(2) | Unchanged |

Figure 3 shows that the four adjacent voltage values in the model sub-table can be determined by using the estimated SOC and the sampled current \( i \) after selecting the corresponding sub-table. The four voltage values are iteratively calculated by the SNNI algorithm, and the given voltage value \( U^* \) is finally output.

**Principle of SNNI algorithm**

As mentioned above, the SNNI algorithm looks up the power battery model sub-table according to the estimated SOC and the sampled load current \( i \), and then obtains four adjacent voltage values \( U_1 \)–\( U_4 \). The four points are regarded as an initial rectangle, and they are iteratively calculated as follows until a given voltage value is obtained.

1. Based on the rules of Table 1, the rectangle is divided into four small ones, defined as the zones 1–zone 4; let \( m = 1 \), \( m \) is the number of iterations.
2. The nearest one of the \( U_n(m)(n=1,2,3,4) \) in the same zone with \( U \) is the invariant point.
3. The four vertex \( U_1(m+1) \), \( U_2(m+1) \), \( U_3(m+1) \), \( U_4(m+1) \) of the update zone determined by points \( U \) are updated with the formula in Table 2, which are defined in (6)–(9).
4. Let \( m=m+1 \), if \( m<M \) (\( M \) is the maximum number of iterations), return to the step (1) for locating further refined zone, otherwise the algorithm is terminated.

The formulas used to update \( U_n \) in Table 2 are

\[
\begin{align*}
U(m+1) &= \frac{U(m)+U+1(m)}{2} \\
(1) \\
U(m+1) &= \frac{U(m)+U+1(m)}{2} \\
(2) \\
U(m+1) &= \frac{U(m)+U+1(m)}{2} \\
(3) \\
U(m+1) &= \frac{U(m)+U+1(m)}{2} \\
(4) \\
U(m+1) &= \frac{\sum_{n=1}^{4} U_n(m)}{4} \\
(5)
\end{align*}
\]

Formula (1)–(5) describe the voltage value updating methods of the left point, the right point, the upper point, the lower point, and the diagonal point relative to the invariant point.

The coordinates SOC and \( i \) also need to be updated after each iteration and the updating formulas of zone 1–zone 4 are list in (6)–(9).

\[
\begin{align*}
\text{SOC}(m+1) &= \text{SOC}(m) \\
i(m+1) &= i(m) \\
(6) \\
\text{SOC}(m+1) &= \text{SOC}(m) + \frac{\Delta \text{SOC}(m)}{2} \\
i(m+1) &= i(m) + \frac{\Delta i(m)}{2} \\
(7) \\
\text{SOC}(m+1) &= \text{SOC}(m) + \frac{\Delta \text{SOC}(m)}{2} \\
i(m+1) &= i(m) + \frac{\Delta i(m)}{2} \\
(8) \\
\Delta \text{SOC}(m+1) &= \frac{\Delta \text{SOC}(m)}{2} \\
\Delta i(m+1) &= \frac{\Delta i(m)}{2} \\
(9) \\
\end{align*}
\]

The resolution updating formula of SOC and \( i \) can be expressed as

\[
\begin{align*}
\Delta \text{SOC}(m+1) &= \frac{\Delta \text{SOC}(m)}{2} \\
\Delta i(m+1) &= \frac{\Delta i(m)}{2} \\
(10)
\end{align*}
\]

When \( m = M \), formula (5) is used to calculate the given voltage \( U^* \).

**Simulations**

**Determining the number of iteration**

As described in the Common interpolation method section, the proposed algorithm needs to be completed through multiple iterations. So the first step is to determine the number of iterations. Figure 4 shows the error between the given voltage \( U^* \) and the reference value after each iteration under different SOC and different currents \( i \). From Figure 4, we can see that the absolute error is less than 0.005 V when the number of iterations is greater than 5. When the number of iterations is 10, the absolute error is within 0.0005 V. In this paper, the number of iterations is set to 10.
Determining model table data capacity

In order to get higher resolution, large samples are often used in the look-up table method, but this will put forward high requirements for the system memory capacity. Therefore, under the condition of ensuring the accuracy of the output voltage, the data capacities of the battery model table need to be reduced as far as possible. In order to verify the given voltage accuracy of proposed method under the premise of reducing the data capacity, the model sub-tables are established by using the five extraction ratios as shown in Table 3, and the accuracy of the given voltage is evaluated under different conditions.

In the range of SOC and i (i.e., 5%≤SOC≤100%, -100 A≤i≤100 A), using the standard power battery model, the sub-tables of 1000 samples are randomly generated. The extraction modes in Table 3 are used to calculate the given voltage, and the statistical results of the absolute error are shown in Table 4.

As can be seen from Table 4, the absolute errors of D1 and D3 extraction mode are over 1V, and the standard deviations are greater than 0.3 V. In this case, the results are too divergent to meet the accuracy requirements, and the reason is the proportion of extraction is too large on the row direction in sub-table 1 and sub-table 3. The absolute errors of D2, D4, and D5 extraction mode are under 0.2 V, and the standard deviations are less than 0.02 V. These statistics show that the algorithm has good performance in all four zones by using D2, D4, and D5 extraction mode. Because the D2 extraction mode has the smallest capacity and can meet the accuracy requirement, it is selected as the data capacity for the subsequent experiments.

Comparison with nearest-neighbor interpolation

A standard power battery model in MATLAB/SIMULINK is used to generate benchmark table, and the bilinear interpolation algorithm is used to find the reference table to obtain the standard voltage value. Based on the D2 extraction mode, the nearest-interpolation algorithm (M1) and the SNNI method (M2), which is proposed in this paper, are used to calculate the given voltage of the battery simulator. The absolute errors of the given voltages obtained by the two methods are analyzed. The simulations were carried out with fixed SOC (5%≤SOC≤100%, random value) and different current (−100 A≤i≤100 A, 100 samples are randomly generated). Parts of the results are shown in Figure 5.

It can be clearly seen from Figure 5(a) that under the condition of SOC 13.7% and using benchmark table, the absolute errors are less than 0.0004 V using M2, while the results of M1 are about 1.2 V. When the D2 extraction mode table is applied and other conditions are unchanged, the absolute errors of M2 are less than 0.03 V while the results of M1 are about 1.7 V. Therefore, the M2 has higher accuracy under the same condition. At the same time, the results show that the SOC value of 13.7% is in the range of great change which makes the difference of two adjacent coordinate’s voltage become larger. Therefore, the absolute errors of M1 become greater. However, the errors caused by the decrease of resolution are greatly reduced by successive approximation of M2.

It can be seen from Figure 5(b) that under the condition of SOC 39.5% and using benchmark table, the absolute errors of M2 are less than 0.0004 V while the results of M1 are about 0.2 V. When the D2 extraction mode table is applied and other conditions are unchanged, the absolute errors of M2 are less than 0.02 V while the results of M1 are about 0.6 V. The reason that the absolute errors are less than that of SOC with 13.7% is because the SOC with 39.5% is in the smooth range of power battery characteristic curve.

A large number of experiments have been carried out to verify the effectiveness of M2 under the condition of 5%–100% SOC, and the results show that the absolute errors are all less than 0.03 even using D2 extraction mode table. Compared with M1, M2 has obvious advantages in the accuracy of the given voltage.

Experiments

To verify the availability of the proposed method in the actual system, an experimental setup is built to realize the power battery simulator, as shown in Figure 6. The main circuit of

---

**Table 3.** Sub-tables extraction ratio.

| Extraction mode | Row/column | Sub-table1 | Sub-table2 | Sub-table3 | Percentage of original capacity, % |
|-----------------|------------|------------|------------|------------|----------------------------------|
| D1              | Row        | 5:1        | 5:1        | 5:1        | 0.5                              |
|                 | Column     | 40:1       | 40:1       | 40:1       |                                  |
| D2              | Row        | 1:1        | 5:1        | 1:1        | 1.6                              |
|                 | Column     | 30:1       | 30:1       | 30:1       |                                  |
| D3              | Row        | 2:1        | 2:1        | 2:1        | 5                                 |
|                 | Column     | 10:1       | 10:1       | 10:1       |                                  |
| D4              | Row        | 1:1        | 1:1        | 1:1        | 20                                |
|                 | Column     | 5:1        | 5:1        | 5:1        |                                  |
| D5              | Row        | 1:1        | 2:1        | 1:1        | 32                                |
|                 | Column     | 2:1        | 2:1        | 2:1        |                                  |

**Table 4.** Accuracy comparison of different extraction modes.

| Statistics       | D1       | D2       | D3       | D4       | D5       |
|------------------|----------|----------|----------|----------|----------|
| Absolute error   |          |          |          |          |          |
| Maximum          | 10.6074  | 0.1288   | 1.8146   | 0.0115   | 0.0123   |
| Minimum          | 0.0002   | 5.39e-6  | 3.87e-6  | 6.33e-7  | 5.05e-7  |
| Average value    | 0.8655   | 0.0110   | 0.0984   | 9.44e-4  | 0.0016   |
| Standard deviation | 2.4619  | 0.0197   | 0.3001   | 0.0024   | 0.0024   |
the setup is a bidirectional DC/DC converter with fast dynamic response and high precision. The controller adopts the TMS320F28335 DSP processor, which is mainly used to sample the voltage and current of the main circuit and output a pair of complementary adjustable PWM wave with dead zone. The drive circuit operates according to the DSP PWM command signal to drive the DC/DC circuit. The output of the DC/DC converter can accurately track the given voltage of the battery model. The modules relationships of the setup are shown in Figure 7.

The CCS6.0 software run in PC is used for debugging when the experiments are carried out. The DC power supply is used as the input of the DC/DC and the auxiliary power supply is employed for the DSP board, the drive board, and the sampling board. The output of DC/DC, which is tracking the given voltage by the proposed method, is

Figure 5. Absolute errors of M1 and M2.

Figure 6. Prototype of power battery simulator.

Figure 7. Schematic diagram of the experimental setup.

Figure 8. Experimental scenario.

Figure 9. Output voltage of the setup.
observed by oscilloscope. The experimental scenario is shown in Figure 8.

The proposed algorithm is programmed by CCS and downloaded to DSP chip to run. Due to the small memory occupation of the D2 extraction mode sub-tables and the small amount of computation, the proposed algorithm can run in real time.

A large number of experiments have been done with the setup to verify the feasibility of the proposed method. This can be illustrated by the results of an experiment, as shown in Figure 9. In this case, the SOC=56.95% and \( i=30.82\) A, it can be seen that the absolute error is about 0.11 V. The error is larger than that of simulation, because in addition to the error caused by the algorithm, the result is also affected by the noise interference.

The absolute errors of tracking voltage on the setup in 100 consecutive experiments are shown in Figure 10. It can be seen that the absolute errors are less than 0.62. Therefore, the proposed method can accurately output a given voltage in a real system and can realize real-time dynamic response. Because of the advantages of small computation, smooth output voltage, and high precision, the proposed algorithm is convenient for practical application.

Conclusions

In this paper, a new method using SNNI is proposed for output voltage of power battery simulator. Under the condition that the data capacity of the model table is reduced by 98.4%, the given voltage accuracy of the proposed algorithm is controlled within 0.03 V, which is greatly improved compared with the traditional method. Therefore, the storage capacity requirement of the system can be greatly reduced on the premise of ensuring the accuracy, and system cost is reduced. In general, the proposed algorithm only needs 4–5 iterations to achieve the accuracy requirements, and the algorithm time is reduced because SNNI uses simple operation rules. By embedding the proposed algorithm into TMS320F28335 DSP chip, it can be easily integrated into the power battery simulation system, so the proposed method can be easily applied to low-cost systems. For further research, on the one hand, through the optimization of boundary points, the table look-up method based on SNNI will be more perfect, so as to enhance the stability in practical application.

On the other hand, for different applications, the development of flexible power battery simulator is worthy of further study.

Acknowledgments

This work was partially supported by the natural science foundation of Hunan Province of China (2019J60055) and the scientific research projects of the Department of education of Hunan Province of China (18A272).

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Jiliang Yi https://orcid.org/0000-0002-1908-7930
Huabing Tan https://orcid.org/0000-0001-9496-7578
Jin Zhang https://orcid.org/0000-0003-4520-0682

References

1. Genikomsakis KN, Angulo Gutierrez Thomas ID, Thomas D, et al. Simulation and design of fast charging infrastructure for a university-based e-carsharing system. IEEE Trans Intell Transp Syst 2018; 19(9): 2923–2932.
2. Yan X, Wang L, Chai Z, et al. Electric vehicle battery simulation system for mobile field test of off-board charger. Energies 2019; 12(15): 1–17.
3. Cheng Y, Van Mierlo J, Lataire P, et al. Test Bench of Hybrid Electric Vehicle with the Super Capacitor based Energy Storage. Vigo, Spain: IEEE International Symposium on Industrial Electronics, 2007, pp. 147–152.
4. Lee WC and Drury D. Development of a hardware-in-the-loop simulation system for testing cell balancing circuits. IEEE Trans Power Electron 2013; 28(12): 5949–5959.
5. Chen S, Chen Y, Zhang S, et al. A novel integrated simulation and testing platform for self-driving cars with hardware in the loop. IEEE Trans Intell Vehicles 2019; 4(3): 425–436.
6. Wang H, He L, Qi X, et al. Simulation system of the performance of power battery for electrical vehicle based on Internet of things. In: Proceedings of 2012 International Conference on Measurement. Harbin, China: Information and Control, 2012, pp. 681–684.
7. Hidalgo-León R, Urquizo J, Litardo J, et al. Simulation of battery discharge emulator using power electronics device with cascaded P-I control. Buenos Aires, Argentina: IEEE International Conference on Industrial Technology (ICIT), 2020, pp. 959–964.
8. Amrhein M and Krein PT. Dynamic simulation for analysis of hybrid electric vehicle system and subsystem interactions, including power electronics. IEEE Trans Vehicular Technol 2005; 54(3): 825–836.
9. Cao Y, Kroese RC and Krein PT. Multi-timescale parametric electrical battery model for use in dynamic electric vehicle simulations. IEEE Trans Electrific Power Syst 2016; 2(4): 432–442.
10. Cheng CS, Chung HSH, Lau RWH, et al. Time-domain modeling of constant phase elements for simulation of...
11. Kroeze RC and Krein PT. Electrical battery model for use in dynamic electric vehicle simulations. Rhodes, Greece: IEEE Power Electronics Specialists Conference, 2008, pp. 1336–1342.

12. Hamed Y, Shafie A, Mustaffa ZB, et al. An application of K-Nearest Neighbor interpolation on calibrating corrosion measurements collected by two non-destructive techniques. In: IEEE 3rd International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Malaysia, Kuala Lumpur, 2015, pp. 1–5.

13. He C, Li W and Chik D. Waveform compensation of ECG data using segment fitting functions for individual identification. In: 13th International Conference on Computational Intelligence and Security (CIS), Hong Kong, 2017, pp. 475–479.

14. Jardin P and Baudoin G. Filter lookup table method for power amplifier linearization. IEEE Trans Vehicular Technol; 56(3): 1076–1087.

15. Molina A., Rajamani K. and Azadet K. Concurrent dual-band digital predistortion using 2-D lookup tables with bilinear interpolation and extrapolation: direct least squares coefficient adaptation. IEEE Trans Microwave Theor Tech 2017; 65(4): 1381–1393.

16. Suresha D and Prakash HN. Single picture super resolution of natural images using N-Neighbor Adaptive Bilinear Interpolation and absolute asymmetry based wavelet hard thresholding. In: 2nd International Conference on Applied and Theoretical Computing and Communication Technology. Bangalore: iCATcCT, 2016, pp. 387–393.

17. Zhang Y, Wang P, Bao F, et al. A single-image super-resolution method based on progressive-iterative approximation. IEEE Trans Multimedia 2020; 22(6): 1407–1422.

18. Wang Y, Wan W, Wang R, et al. An improved interpolation algorithm using nearest neighbor from VTK. In: International Conference on Audio, Language and Image Processing. Shanghai, China, 2010, pp. 1062–1065.

19. Rukundo O and Maharaj BT. Optimization of image interpolation based on nearest neighbour algorithm. In: International Conference on Computer Vision Theory and Applications. Lisbon: VISPAP Portugal, 2014, pp. 641–647.

20. Sa Y. Improved Bilinear Interpolation Method for Image Fast Processing, 2014 7th International Conference on Intelligent Computation Technology and Automation. Changsha, China: IEEE, 2014, pp. 308–311.

21. Huang W, Xue Y, Hu L, et al. S-EEGNet: electroencephalogram signal classification based on a separable convolution neural network with bilinear interpolation. IEEE Access 2020; 8: 131636–131646.

22. Razmjooy N, Deshpande A, Khalilpour M, et al. Optimal bidding strategy for power market based on improved world cup optimization algorithm. In: “Optimal Bidding Strategy for Power Market Based on Improved World Cup Optimization Algorithm.” Metaheuristics and Optimization in Computer and Electrical Engineering. Lecture Notes in Electrical Engineering, 2020, pp. 137–152.

23. Wang Z, Zhang Z and Razmjooy N. A new configuration of autonomous CHP system based on improved version of marine predators algorithm: a case study. Int Trans On Electr Energ Syst 2021; 31(4): 1–22.

24. Yang Z, Liu Q, Zhang L, et al. Model parameter estimation of the pemfc using improved barnacles mating optimization algorithm. Energy 2020; 212(1): 1–10.

25. Razmjooy N and Ramezani M. Robust optimal control of two-wheeled self-balancing robot using chebyshev inclusion method. Majlesi J Electr Eng 2018; 12(1): 13–21.

26. Razmjooy N, Madadi A and Ramezani M. Robust control of power system stabilizer using world cup optimization algorithm. Int J Inf Security Syst Manag 2016; 5(1): 519–526.