

# An MMC Submodular Capacitor Voltage Balancing Scheme Based on Improved Bucket Sorting

Haoyu Liu

State Key Laboratory for Alternate Electrical Power System  
With Renewable Energy Sources  
North China Electric Power University  
Beijing, China  
haoyu\_l@126.com

Chongru Liu

State Key Laboratory for Alternate Electrical Power System  
With Renewable Energy Sources  
North China Electric Power University  
Beijing, China  
chongru.liu@qq.com

**Abstract**—Submodular capacitor voltage balancing problem is becoming increasingly prominent with the rapid development of high voltage and large capacity modular multilevel converter (MMC). The traditional submodular capacitor voltage balancing strategy based on bubble sorting has high time complexity, quantities of comparisons, and consumes a lot of computing resources. A new submodular capacitor voltage balancing method based on improved bucket sorting is proposed in this paper. The scheme utilizes bucket sorting to divide capacitor voltage into different buckets, and then sort the elements in only one bucket by improved quick sorting which sets the  $M$ th data as the reference value and just sorts on one side of benchmark. The strategy can accurately find sub-modules with the smallest or largest capacitor voltage, while time complexity of the algorithm is  $O(n)$ . Finally, a 227 level one-terminal MMC is used to verify the effectiveness of proposed method.

**Keywords**—Submodular Voltage Balancing Scheme, Improved Bucket Sorting, Improved Quick Sorting, Time Complexity

## I. INTRODUCTION

In 2001, R. Marquardt proposed the topology of modular multilevel converter (MMC) [1]. MMC bridge arms are formed by Sub-Modules (SM). As the switching frequency and the harmonic content of voltage waveform have significantly reduced, MMC becomes a research hotspot [2-4]. Since energy of MMC is stored in sub-module capacitors, how to coordinate the on and off state of different SMs and balance internal energy of capacitors is of importance.

In order to achieve energy balance of SMs, the SMs with lower capacitor voltage have priority to be charged and the SMs with higher voltage have priority to be discharged [5]. However, the conventional submodular capacitor voltage balancing method [6] has high time complexity and low computational efficiency which makes it difficult to deal with high-voltage and large-capacity MMC possessing hundreds of SMs. The maximum deviation of SM capacitor voltage in [7] and the retention factor in [8] are intended to reduce SMs switching losses as much as possible through reducing the switching frequency. However, the consistency of capacitor voltage still remains problems. To reduce computation of sorting, the prime factorization method for determining the number of SM packets is employed in [9], and the SMs are sorted within groups. Based on prime factorization grouping, the Hill sorting algorithm is introduced to sort within and between packets [10]. However, the process is complex and difficult to debug.

In view of the above-mentioned problems, a submodular capacitor voltage balancing method based on improved bucket sorting is proposed in this paper. The method has high accuracy in sorting SM capacitor voltage, and the consistency of voltages is high. Moreover, the time complexity of the method is linear, which can significantly reduce computation amount of sorting process. A 227 level MMC model in PSAD/EMTDC is used to verify the proposed method.

## II. FUNDAMENTAL PRINCIPLE AND MODULATION METHOD OF MMC

### A. Fundamental Principle of MMC

The topology of MMC is shown in Fig.1.

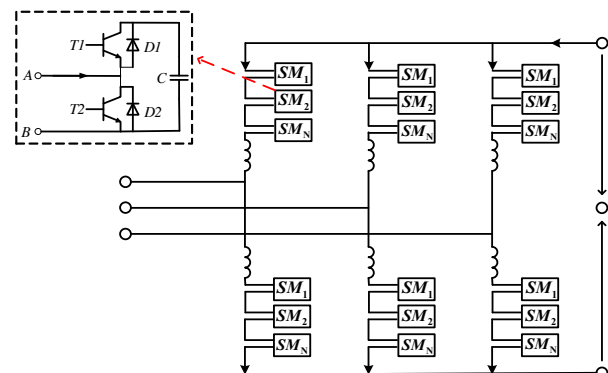


Fig. 1. Principle Topology of MMC

MMC is constituted by 6 bridge arms and each bridge arm has  $N$  SMs. Taking the half bridge sub-module (HBSM) as an example, its structure is shown in Fig.1. IGBT  $T1$  and  $T2$  alternately conduct. When  $T1$  turns on and  $T2$  turns off, the SM will be inserted; when  $T1$  turns off and  $T2$  turns on, the SM will be bypassed. MMC controllers maintain DC voltage stability and generate three-phase AC voltage by controlling the number of SMs conducted on the upper and lower bridge arm of every phase.

### B. Modulation Method of MMC

There are two basic modulation methods of MMC: Pulse Width Modulation (PWM) and Staircase Modulation. PWM was regarded having higher switching frequency and more loss which has not been used in practice projects with high-voltage and large-capacity MMC possessing hundreds of SMs. The Nearest Level Modulation (NLM) is a kind of Staircase Modulation, which is widely used in commercial

projects. When the number of levels is large, the harmonic content will be very low, which makes it quite suitable for high-voltage and large-capacity MMC.

The principle of NLM is shown in Fig.2.

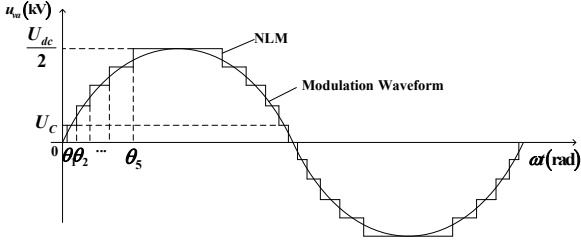


Fig. 2. NLM of MMC

The basic principle of NLM is to approximate the sinusoidal modulated wave with the nearest voltage or the closest level instantaneous value. NLM calculates the number of SMs that is required to be inserted in the upper and lower bridge arm by rounding the ratio of modulated wave to the average value of SM capacitor voltage. On this basis, submodular voltage balancing control determines which SMs would be inserted in the upper and lower bridge arm through sorting algorithm. Therefore, for high-voltage with over 500 voltage-levels, fast and accurate sorting method is of very importance in Staircase Modulation.

### III. IMPROVED BUCKET SORTING THEORY

#### A. Bucket Sorting Theory

Bucket sorting uses a nearly complete divide and conquer thought. It commonly assumed that data is evenly distributed and divided into several buckets. A mapping function is used to distribute the data into each bucket and then the data is sorted within each bucket and merge all barrels to form an ordered arrangement of all data. The basic realization process of bucket sorting can be shown in Fig. 3.

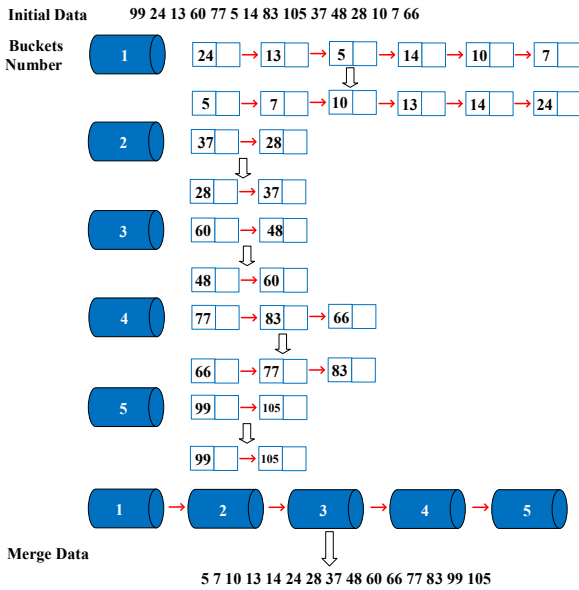


Fig. 3. Basic Implementation Process of Buckets Sort

Assuming that initial data are [99 24 13 60 77 5 14 83 105 37 48 28 10 7 66].

Step 1: Find the maximum value and the minimum value of the data, and then divide all the data into several buckets with equal interval.

In Fig.3, the maximum and minimum values are 5 and 105. It is divided into 5 buckets. Hence, the ranges of the 5 buckets are [5, 25), [25,45), [45,65), [65,85) and [85,105].

Step 2: Distribute all data into different bucket using mapping function  $f(x)$  in (1).

$$f(x) = \begin{cases} \text{floor}\left(\frac{x-x_{\min}}{B_{\text{size}}} + 1\right), & x < x_{\max} \\ B_N, & x = x_{\max} \end{cases} \quad (1)$$

Where  $x$  is the data vector needed to be sorted.  $x_{\min}$  and  $x_{\max}$  are the minimum and maximum value of  $x$ .  $B_{\text{size}}$  is the interval between buckets and  $B_N$  is buckets number.

For the data in Fig.3, bucket 1 includes [24 13 5 14 10 7], bucket 2 includes [37 28] and bucket 3 includes [60 48]. Data in bucket 4 and bucket 5 are [77 83 66] and [99 105].

Step 3: Sort within each bucket to obtain internal ordered permutation.

After sorting, data in bucket 1 become [5 7 10 13 14 24]. Data in bucket 2 become [28 37]. Data in bucket 3 become [48 60]. Data in bucket 4 become [66 77 83]. Data in bucket 5 become [99 105].

Step 4: Combine the data in sequence to obtain the final sorting results.

By splicing the data from each bucket in turn, a new ordered sequence [5 7 10 13 14 24 28 37 48 60 66 77 83 99 105] is obtained in Fig.3.

#### B. Principle of Improved Bucket Sorting

The ultimate objective of bucket sorting is to rank all data orderly, so data are required internally sorting after distributed into each bucket. However, the process of sorting in the bucket actually generates a large number of comparison operations, which increases the complexity of algorithm.

It is well known that submodular voltage balancing method usually inserts SMs with lower voltage when charge and inserts ones with higher voltage when discharge. That is, it is not necessary to sort all the SMs and only  $N_{ON}$  SMs with the smallest or largest capacitor voltage need to be found and the exact order of the chosen  $N_{ON}$  SMs is not concerned. Consequently, the bucket sorting can be improved as follows.

Assuming that data is evenly distributed and the number of data in each bucket is  $NUMBER(i)$ . The number of SMs to be inserted should satisfy (2).

$$\sum_{i=1}^k NUMBER(i) < N_{ON} \leq \sum_{i=1}^{k+1} NUMBER(i) \quad (2)$$

Data in the first  $k$  buckets and partial data after sorted in the  $(k+1)th$  bucket constitute the smallest  $N_{ON}$  data from all data.

Aiming at improving computational efficiency, this paper introduces the improved quick sorting to sort in the  $(k+1)th$  bucket.

Quick sorting still uses idea of divide and conquer. The first data is used as a benchmark, and all the data are divided into two regions. Data on the left side of base value are not larger than reference value and data on the right side are not less than reference value. The two regions are further divided using idea of divide and conquer. And so on, until each region is divided into only one data, the final ranked sequence is obtained.

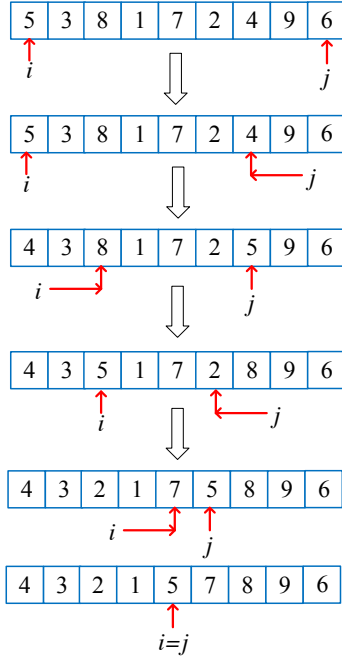


Fig. 4. Example of Quick Sorting

An example of quick sorting is shown in Fig.4.

1)  $i$  points to the position of the first data while  $j$  points to the last one.

2)  $j$  searches from back toward front until the first data less than the reference value is found.

3) The data pointed by  $j$  is exchanged with the data pointed by  $i$ . Then  $i$  searches from toward back until the first data larger than the reference value is found.

4) Then the data pointed by  $i$  is exchanged with the data pointed by  $j$ . Then  $j$  searches toward front until the first data less than the reference value is found.

5) Back to step 3) and 4) till  $i$  and  $j$  meets and the quick sorting ends.

As can be seen from above process, benchmark of quick sorting is the first data of a sequence. Meanwhile, quick sorting is still a kind of entire arrangement. The target of improved quick sorting is to find the smallest  $M$  data in all data of the  $(k+1)th$  bucket, where  $M$  is,

$$M = N_{ON} - \sum_{i=1}^k NUMBER(i) \quad (3)$$

In order to reduce comparison times, it always sets the  $Mth$  data as the reference value and just sorts on one side of benchmark.

Basic process of improved quick sorting is shown in Fig. 5. Initial data are [5 7 8 3 6 1 2 4 9] and the smallest 6 data should be found out.

1) Region to be sorted is all data. Select the 6th data in the region to be sorted as reference value and exchange it with the first data in the region. Then perform a quick sort, and the result is shown in Figure 5(b). There is no data on the left side of reference value 1, and 8 data on the right side. It also needs to find 5 data from the right side of benchmark.

2) Region to be sorted is [7 8 3 6 5 2 4 9]. Select the 5th data in the region to be sorted as reference value and exchange it with the first data in the region. Then perform a quick sort, and the result is shown in Figure 5(c). There are 4 data on the left side of reference value 5, and 4 data on the right side of benchmark. It also needs to find 1 data from the right side of benchmark.

3) Region to be sorted is [7 6 8 9]. Select the first data in the region to be sorted as reference value. Then perform a quick sort, and the result is shown in Figure 5(d). There are 6 data on the left side of reference value 7, and sorting ends.

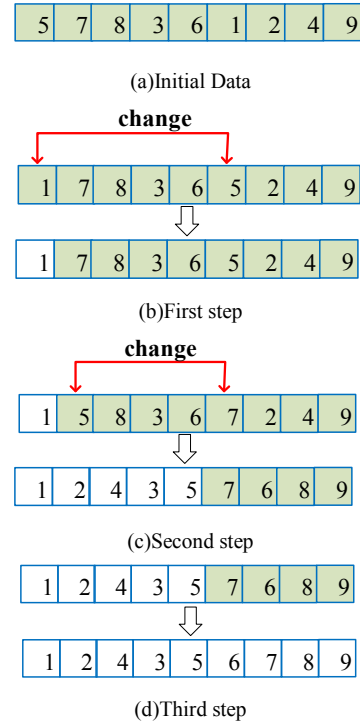


Fig. 5. Improved Quick Sorting

Improved quick sorting do not need to sort all data in the  $(k+1)th$  bucket. It just sets the  $Mth$  data as the reference value and sorts on one side of benchmark. Therefore, improved quick sorting greatly reduces the scale of data to be sorted and improves computational efficiency.

### C. Time complexity

Because data is mapped to each bucket by mapping function, time complexity of bucket sorting is mainly composed of two parts. One part of time complexity is finding the maximum and minimum value of data, and the other is sorting within each bucket. Assuming that there are  $n$

data needed to be sorted, and these data are evenly distributed. The improved bucket sorting divides these data into  $L$  buckets. Then the average number of data in each bucket is  $n/L$ . So, the first part of time complexity is

$$T_1(n) = 2n \quad (4)$$

Because improved bucket sorting just needs to sort in the  $(k+1)$ th bucket, the second part of time complexity [11] is,

$$T_2(n) < 4 \times \frac{n}{L} \quad (5)$$

Therefore, the time complexity of entire algorithm is,

$$T(n) = T_1(n) + T_2(n) < 2n + 4 \times \frac{n}{L} \quad (6)$$

If there is only one bucket, then it is no need to calculate the maximum and minimum values of data. In this case, improved bucket sorting is completely degraded to improved quick sorting, and the time complexity is

$$T(n) < 4n \quad (7)$$

In summary, the time complexity of improved bucket sorting is

$$\begin{cases} T(n) < 4n, L = 1 \\ T(n) < 2n + 4 \times \frac{n}{L}, L > 1 \end{cases} \quad (8)$$

That is to say, the time complexity of improved sorting is  $O(n)$ .

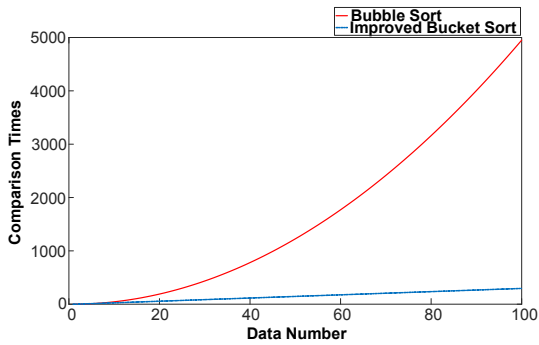


Fig. 6. Time Complexity of Bubble Sort and Improved Bucket Sort

The comparison of time complexity between bubble sorting and improved bucket sorting when bucket number is 3 is shown in Fig.6. The abscissa is the number of data and the ordinate is the comparison times. As the amount of data increases, the comparison times required for bubble sorting will increase rapidly, while the comparison times for improved bucket sorting is always linear with the number of data.

The relationship of calculation time and number of data to be sorted is illustrate in Fig.7, where the abscissa is the number of random data and the ordinate is the running time.

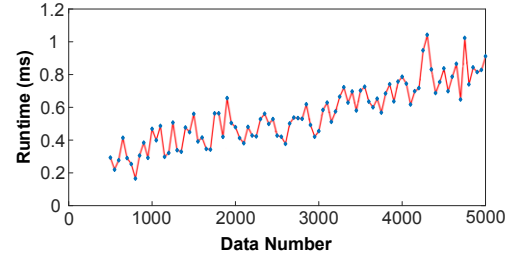


Fig. 7. Simulation Output of Improved Bucket Sort

The simulation time for improved bucket sorting is significantly positively correlated with the number of data.

#### IV. CAPACITOR VOLTAGE BALANCING METHOD BASED ON IMPROVED BUCKET SORTING

The capacitor voltage balancing control method in this paper by using the proposed improved bucket sorting theory combined with NLM is as follows.

1) Read SM capacitor voltage  $V_C(t)$ , bridge arm current  $I_{br}(t)$  and the number of SMs required to be inserted in bridge arm  $N_{ON}$  by NLM. Meanwhile, load the number of SMs in bridge arm  $N$  and the bucket number  $B_N$ .

2) Calculate the maximum value  $V_{C_{MAX}}$  and the minimum value  $V_{C_{MIN}}$ . Then calculate the interval  $B_{size}$  between buckets.

$$B_{size} = \frac{V_{C_{MAX}} - V_{C_{MIN}}}{B_N} \quad (9)$$

Determine boundaries of each bucket using  $B_{size}$  and  $B_N$ . Then map  $V_C(t)$  into different buckets by mapping function

$$f(x) = \begin{cases} \text{floor}\left(\frac{x - V_{C_{MIN}}}{B_{size}} + 1\right), x < V_{C_{MIN}} \\ B_N, x = V_{C_{MIN}} \end{cases} \quad (10)$$

3) Charge SMs with the smallest  $N_{ON}$  capacitor voltage when  $I_{br}(t)$  is greater than 0 and discharge SMs with the largest  $N_{ON}$  capacitor voltage when  $I_{br}(t)$  is negative. Calculate the  $N_{DP}$  according to  $I_{br}(t)$ .

$$N_{DP} = \begin{cases} N_{ON}, I_{br} > 0 \\ N - N_{ON}, I_{br} < 0 \end{cases} \quad (11)$$

4) Calculate the position of the  $(k+1)$ th bucket and the number of data required to be inserted in the  $(k+1)$ th bucket according to  $N_{DP}$ . Then perform an improved quick sorting in the  $(k+1)$ th bucket.

5) Merge data in each bucket, select SMs with the smallest or largest  $N_{ON}$  capacitor voltage and generate switching signal for SMs.

The flowchart of capacitor voltage balancing control based on improved bucket sorting is shown in Fig.8.

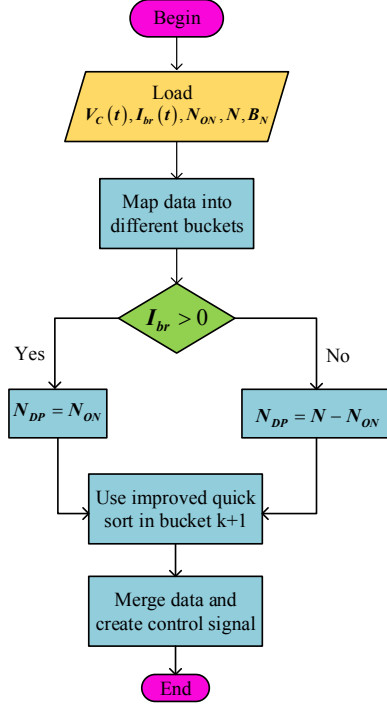


Fig. 8. Flowchart of Voltage Balance Control

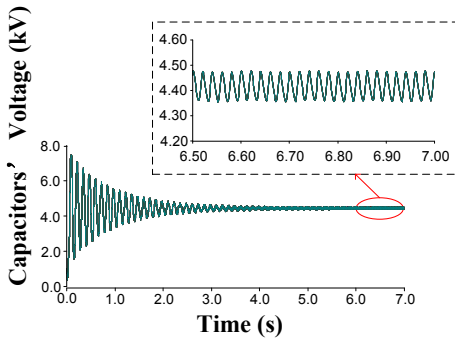
## V. SIMULATION VERIFICATION

A one-terminal 227 level MMC is built in PSCAD/EMTDC to verify the proposed method. The model uses active and reactive power control, and parameters of the case are shown in Table 1.

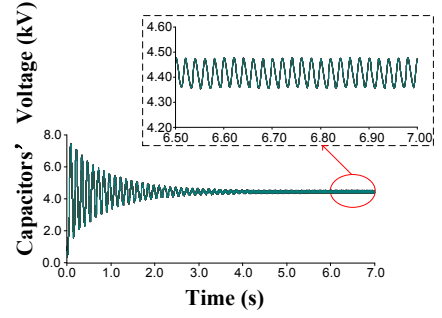
TABLE I. MODEL PARAMETERS

Parameters	Quantity (Unit)
RMS voltage of AC source	525 (kV)
Inductor in AC side	0.02913 (H)
Resistor in AC side	0.8 ( $\Omega$ )
Arm inductor	0.075 (H)
Number of SMs	226
Capacitance of SM	0.015 (F)
Active power reference	750 (MW)
Reactive power reference	1 (MVar)
DC voltage	1000 (kV)

The SM capacitor voltage in the upper bridge arm of phase *a* is shown in Fig.9.



(a) Capacitors' Voltage by Bubble Sorting



(b) Capacitors' Voltage by Improved Bucket Sorting

Fig. 9. Capacitors' Voltage by Bubble Sort and Improved Bucket Sort

Compared the bubble sorting and the improved bucket sorting, the capacitors' voltage obtained by improved bucket sorting is consistent with that by bubble sorting, however, the running time of improved bucket sorting is greatly shorter than that of bubble sorting.

$$\varepsilon = \frac{V_{CMAX} - V_{CMIN}}{(V_{CMAX} + V_{CMIN}) / 2} \quad (12)$$

The unbalance degree of capacitor voltage proposed in [12], as shown in (12), is used to get the curve in Fig.10.

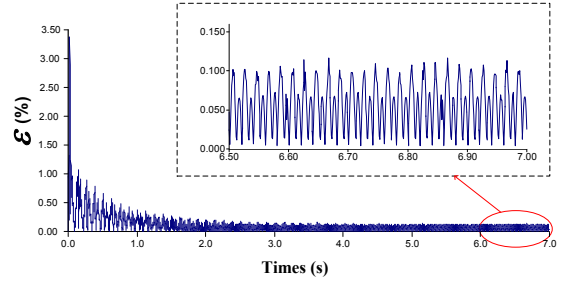


Fig. 10. Unbalance Degree of Capacitors' Voltage by Improved Bucket Sort

It can be seen from Fig.10 that the unbalance degree of SM capacitor voltage after stabilization is less than 0.15%, which indicates the method proposed in this paper can effectively improve the consistency of SM capacitor voltage.

## VI. CONCLUSION

This paper proposes an improved strategy of submodular capacitor voltage balancing. Compared with bubble sorting, the method can significantly improve computation efficiency on the basis of guaranteeing the effect of voltage balancing.

An improved bucket sorting combining with characteristic of MMC is proposed. The method first divides SM capacitor voltage value into buckets, avoiding the entire arrangement of all data. Based on this, data in one bucket are sorted by improved quick sort to accurately find the smallest or largest  $N_{ON}$  capacitors' voltage. The time complexity of the method proposed in this paper is  $O(n)$ , which is of high computation efficiency.

Simulation results show that the capacitors' voltage completely consistent with bubble sorting, while the running

time significantly decreases and the unbalance degree of voltage is low, which indicates the effectiveness of proposed method.

#### REFERENCES

- [1] Glinka, M and R. Marquardt, "A new AC/AC multilevel converter family." *IEEE Transactions on Industrial Electronics*, vol. 52, no. 3, pp. 662-669, 2005.
- [2] Li Gengyin et al, "Development and prospects for HVDC light." *Automation of Electric Power Systems*, vol. 27, no. 4, pp. 77-81, 2003.
- [3] Tang Guangfu, He Zhiyuan and Pang Hui, "Research, Application and Development of VSC-HVDC Engineering Technology." *Automation of Electric Power Systems*, vol. 37, no. 15, pp. 3-14, 2013.
- [4] Xu Zheng and Chen Hairong, "Review and Application of VSC HVDC." *High Voltage Engineering*, vol. 33, no. 1, pp. 1-10, 2007.
- [5] Ding Guanjun et al, "Submodule Capacitance Parameter and Voltage Balancing Scheme of a New Multilevel VSC Modular." *Proceedings of the CSEE*, vol. 29, no. 30, pp. 1-6, 2009.
- [6] Saeedifard, Maryam, and R. Iravani. "Dynamic Performance of a Modular Multilevel Back-to-Back HVDC System." *IEEE Transactions on Power Delivery*, vol. 25, no. 4, pp. 2903-2912, 2010.
- [7] Tu Ruiqing et al, "An Optimized Voltage Balancing Method for Modular Multilevel Converter." *Transactions of China Electrotechnical Society*, vol. 26, no. 5, pp. 15-20, 2011.
- [8] Guan Minyuan and Xuzheng, "Optimized Capacitor Voltage Balancing Control for Modular Multilevel Converter Based VSC-HVDC System." *Proceedings of the CSEE*, vol. 31, no. 12, pp. 9-14, 2011.
- [9] Peng Maolan et al, "An Optimized Capacitor Voltage Balancing Control Algorithm for Modular Multilevel Converter Employing Prime factorization Method." *Proceedings of the CSEE*, vol. 34, no. 33, pp. 5846-5853, 2014.
- [10] He Zhipeng et al, "A Capacitor Voltage Balancing Strategy Adopting Prime Factorization Method and Shell Sorting Algorithm for Modular Multilevel Converter." *Proceedings of the CSEE*, vol. 35, no. 12, pp. 2980-2988, 2015.
- [11] Su Shiping et al, "A Modified-sorting Balancing Method of Capacitor Voltage for Modular Multilevel Converter." *Proceedings of the CSEE*, vol. 37, no. 13, pp. 3874-3882, 2017.
- [12] Jiang Xirui et al, "A Capacitor Voltage Balancing Control Strategy for High-voltage Large-capacity VSC-HVDC Systems Based on Tabu Search Hard Optimization Algorithm." *Proceedings of the CSEE*, vol. 33, no. 21, pp. 71-80, 2013.