Real-Time Lossless Compression for Ultrahigh-Density Synchrophasor and Point-on-Wave Data

Chang Chen, *Student Member, IEEE*, Weikang Wang, *Student Member, IEEE*, He Yin, *Member, IEEE*, Lingwei Zhan, *Member, IEEE*, and Yilu Liu, *Fellow, IEEE*

***Abstract*—Modern advanced phasor measurement units are developed with ultrahigh reporting rates to meet the demand for monitoring the power systems dynamics in detail. Due to the large volume of data, the communication and storage systems are seriously challenged with the presence of ultrahigh-density (UHD) synchrophasor and point-on-wave (POW) data. Therefore, it is an urgent task to compress the UHD data for more efficient communication and data storage. This article proposes several methods to compress the synchrophasor and POW data in a lossless manner. First, an improved time-series special compression (ITSSC) method is proposed to compress the UHD frequency data. Second, a delta-difference Huffman method is combined with the time-series special compression algorithm to compress the UHD phase angle data. Finally, a cyclical high-order delta modulation method is proposed to compress the UHD POW data. The proposed models are extensively tested and compared with different existing lossless compression algorithms using the field-collected synchrophasor and POW data at different reporting rates. The results indicate that the proposed algorithms are efficient in performing lossless compression for the UHD synchrophasor and POW data in real time.**

***Index Terms*—Huffman encoding method, improved time-series special compression (ITSSC) method, lossless data compression, point-on-wave (POW) data, synchrophasor data, ultrahigh density.**

I. Introduction

WITH the increasing demand to analyze the complex power system dynamics, the phasor measurement unit (PMU) is required to provide ultrahigh-density (UHD) data. Recently, the advanced PMUs developed by FNET/GridEye are reported to provide synchrophasor and point-on-wave (POW) data at an extremely high reporting rate, e.g., 6000 Hz, which is much faster than other devices that have been recorded [1]–[3]. However, the UHD data will significantly challenge the underlying communication network and the server storage due to its massive data size [4]. Therefore, intelligent online compression methods are essential for UHD data to maintain the efficiency of communication and storage.

The compression techniques generally are classified into two fields: lossy compression and lossless compression. Lossy compression approaches can have a better compression performance though sacrificing the accuracy of the data [5]. However, online compression of UHD data needs to maintain data accuracy for future analyses. Therefore, the lossless compression algorithms that can exactly reconstruct the raw data are preferred for the online compression of UHD data. Lossless compression can be categorized into dictionary-based and prediction-based algorithms. The dictionary-based method compresses the repetitive patterns into a shorter codeword. The computational time of some dictionary-based methods, such as LZ77 and LZW, is a great concern, though they can achieve a high compression ratio (CR) for archived data. On the other hand, the prediction-based methods need prior knowledge of the data characteristics, which allows the algorithm to transfer the raw data into some residuals. Then, some methods can be adopted to compress the residuals.

Some methods have been proposed to compress synchrophasor data in a lossless manner [6]–[14]. However, these methods are optimal only for archived data, and the online compression is mostly implemented by lossy compression methods [15], [16]. The time-series special compression (TSSC) method is a dedicated algorithm developed for the streaming telemetry transport protocol. It is a potential tool for the UHD frequency data compression, which changes slightly due to the high sampling rate. However, this method does not consider the effect of the number of the immediate predecessors (IPres) N and the size of the moving window where the valid bits (VBs) are recorded M on the CR, making it not the optimal solution for the UHD frequency data. The approaches to achieve high CR for the phase angle data generally consist of two stages, preprocessing and encoding [6]. By preprocessing, the raw data are transferred to the intermediate form with low entropy, so that it can be encoded

Manuscript received September 2, 2020; revised December 21, 2020; accepted January 17, 2021. Date of publication February 15, 2021; date of current version October 27, 2021. This work was supported primarily by the Engineering Research Center Program of the National Science Foundation and the Department of Energy under NSF Award Number EEC-1041877 and the CURENT Industry Partnership Program. *(Corresponding author: Weikang Wang.)*

Chang Chen, Weikang Wang, and He Yin are with the University of Tennessee, Knoxville, TN 37996 USA (e-mail: cchen75@utk.edu; wwang72@utk.edu; hyin8@utk.edu).

Lingwei Zhan is with the Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA (e-mail: zhanl@ornl.gov).

Yilu Liu is with the University of Tennessee, Knoxville, TN 37996 USA, and also with the Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA (e-mail: liu@utk.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2021.3057034.

Digital Object Identifier 10.1109/TIE.2021.3057034

by an entropy compression algorithm in the second stage. The Huffman method is one of the most widely used entropy method, which minimizes the average code length in bits by constructing a Huffman table for the data. However, regarding the real-time UHD phase angle compression, the Huffman table information should be reconstructed for every next frame, which introduces more information to each frame, restricting the CR.

For the UHD POW data, traditional algorithms use tools, such as the fast Fourier transform (FFT) to obtain the signal frequency, constructing an estimated POW signal [17], [18]. Lossless compression can be achieved by combining these tools with lossless encoding algorithms, such as Huffman encoding. However, the running time of the FFT is the concern for the real-time PMU compression. Zhang *et al.* [19] propose a high-order delta modulation (HDM) algorithm to preprocess the POW data to some residuals, and it employs the Huffman encoding method afterward. However, the HDM signal does not repeat frequently in one frame, making Huffman method hard to compress well.

This article proposes some methods to efficiently compress the UHD synchrophasor and POW data in real time. First, an improved-TSSC (ITSSC) method is proposed to quickly compress the UHD frequency data. This method set N and M as variables. In this way, the optimal algorithm for the UHD frequency data compression can be obtained at runtime. For the UHD phase angle data, the Huffman-TSSC method is proposed, where the Huffman table information is compressed in stage 3 by the TSSC method to increase the compression performance. Finally, the cyclical HDM (CHDM) method is proposed for the UHD POW data due to its cyclicality. Considering that the CHDM signal does not occur frequently but has fewer VBs in one frame, it is compressed by assigning code words based on the length of the VBs. According to field testing, the proposed methods show satisfying compression performances.

In Section II, the ITSSC method is proposed to compress UHD frequency data based on its characteristics. In Section III, the Huffman compression method is combined with the TSSC algorithm to compress the UHD phase angle data. In Section IV, the cyclical characteristics of the UHD POW data are discussed, and the CHDM-VB method is proposed. Section V tests the proposed methods with both the collected data and a PMU variation, high-speed universal grid analyzer (UGA). Finally, Section VI concludes this article.

II. ITSSC Frequency Compression Method

A. UHD Frequency Data Characteristics

A PMU data frame contains a batch of GPS-synchronized measurement data. It benefits the PMU communication efficiency by including synchrophasors in a single frame to reduce the data streaming rate. Fig. 1 shows one sample frame of the UHD frequency data, where the reporting rate is 1440 Hz. In this case, the PMU streams 10 frames per second, where each frame has $n = 144$ frequency measurements.

As is seen from Fig. 1, although the trend of the UHD frequency data is like a random walking, the high reporting rate reduces the difference between any two consecutive measurements. Traditional compression methods, such as entropy encoding, generally require the raw data to be repetitive, which is not suitable for the UHD frequency data. The TSSC method is applicable to compress the erratic but small-variation values. However, it is not the optimal solution to compress the real-time UHD frequency data. First, this method sets $N = 3$ while a larger N makes the processes raw value (RV) more likely to match one of the IPres, thus a higher CR could reach as no BD is calculated. Second, the TSSC method sets $M = 4$ b, which could lead to extraneous zero prefix, thus it decreases CR. For example, assume that the BDs is 00011101 in binary, where only the last 5 b, 11101, are valid. However, the recorded BDs will be 00011101, when $M = 4$, because it records the three prefix zeros to make up four digits. As opposed to it, when $M = 2$, the BDs will be 011101, which reduces 2 b. However, it is not optimal to naively use the largest N and the smallest M. This is because these parameters will increase the bit length of the CW, thus decreasing the CR as well.

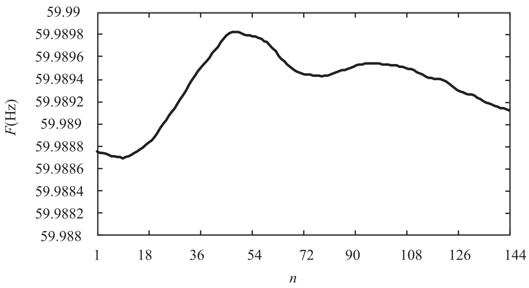


Fig. 1. UHD frequency data.

To efficiently compress the UHD frequency data in real time, this article proposes an ITSSC method. The ITSSC sets N and M as variables, so that the optimal parameters for the UHD frequency compression can be obtained at runtime.

B. ITSSC Method for UHD Frequency Data

Fig. 2 shows the flowchart of the ITSSC method. As illustrated, this method transforms each value into a fixed-length CW and the BDs. Note that the ITSSC method compresses RVs one after another due to the dependence of the current value and its predecessors. The specific steps to compress the ith RV, RV$_i$, are as follows.

Step 1: Compare with immediate predecessors.

There is a vector of N IPres (IPres = [IPres$_0$, IPres$_1$, …, IPres$_{N\text{-}1}$]) in the allocated buffer, where IPres$_0$ is the most recent immediate predecessor. These values are kept so that they can be compared with the current RV. If the RV equals any value in the IPres, a CW will be generated to directly represent RV$_i$. In this way, RV$_i$ is transformed into a fixed-length CW with fewer bits.

Step 2: Calculate BDs.

If no value in the IPres matches RV$_i$, an XOR algorithm is used to calculate the BD of the RV$_i$ (BD$_i$) by

$$BD_i = RV_i{}^{\wedge} IPres_0 \tag{1}$$

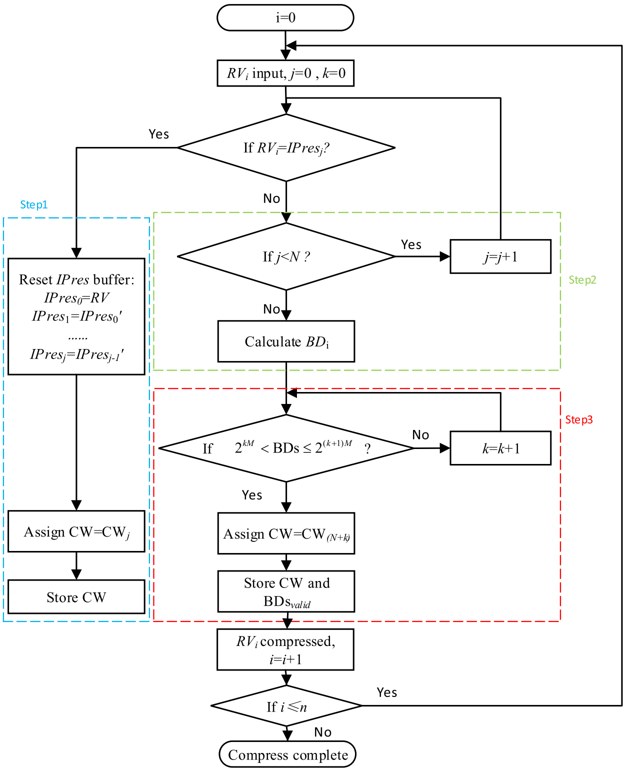


Fig. 2. ITSSC compression algorithm diagram.

where ^ is the XOR operator.

As demonstrated, the ITSSC method is an XOR bitwise compression approach and operates only in binary rather than in floating values, making the operation faster than arithmetic-based methods.

Step 3: VBs of BDs determination, and VB-based CW assignment.

Per its definition, the BD indicates the variation between RV_i and IPres_0. For the UHD frequency data, the BDs are supposed to be small. However, BDs are, by default, 32-b integers, which means only some lower bits can represent the difference. Discarding the meaningless prefix zeros to get the VBs of the BDs ($\mathrm{BD}_{\mathrm{valid}}$) is the next step for ITSSC to reduce redundancy.

However, discarding all the prefix zeros can create more scenarios than only discarding a fixed number of prefix zeros, which requires more CWs. Therefore, instead of discarding all the prefix zeros, $\mathrm{BD}_{\mathrm{valid}}$ is determined by the criterion as follows:

$$2^{kM} < BD_i 2^{(k+1)M} \tag{2}$$

where $k = 0, 1, 2, \ldots, \mathcal{L}(\mathrm{BD}_{\mathrm{valid}})/\mathrm{M}-1$, $\mathcal{L}(\bullet)$ is the bit length calculation function. Once the criterion (2) is satisfied, the last $(k+1)M$ b of the BDs are $\mathrm{BD}_{\mathrm{valid}}$, and the corresponding CW will be stored accompany by $\mathrm{BD}_{\mathrm{valid}}$.

C. Compression Performance

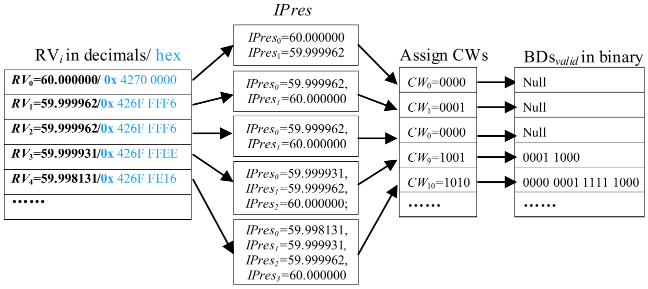


Fig. 3. ITSSC frequency compression process example ($M = 8$, $N = 9$).

The CR is defined as the ratio between the original and the compressed sizes of a data frame, which can be expressed as follows for the ITSSC method:

$$\mathrm{CR} = \frac{n \times \mathcal{L}(\mathrm{RV}) + \mathcal{L}(\mathrm{FrameHeader})}{n \times \mathcal{L}(\mathrm{CW}) + \mathcal{L}(\mathrm{FrameHeader}) + \sum_{i=0}^{n-1} \mathcal{L}(\mathrm{BD}_i)}. \tag{3}$$

Theoretically, the limit of the CR for the proposed ITSSC method can be estimated by

$$\lim_{n\to\infty,\mathrm{BD}=0} \times \left[\frac{n \times \mathcal{L}(\mathrm{RV}) + \mathcal{L}(\mathrm{FrameHeader})}{n \times \mathcal{L}(\mathrm{CW}) + \mathcal{L}(\mathrm{FrameHeader}) + \sum_{i\,=\,0}^{n-1} \mathcal{L}(\mathrm{BD}_i)} \right] = \frac{\mathcal{L}(\mathrm{RV})}{\mathcal{L}(\mathrm{CW})}. \tag{4}$$

As mentioned, the compression performance of the ITSSC method is affected by N and M. As is seen from (4), to pursue a higher CR, a minimized $\mathcal{L}$(CW) should be determined by

$$\mathcal{L}\ (\mathrm{CW}) = \log_2 \left(N + \frac{\mathcal{L}(\mathrm{RV})}{M} \right) \tag{5}$$

where $\lceil \cdot \rceil$ is the operator that rounds a number up to the next largest integer.

D. Compression Example

Fig. 3 shows an example for the ITSSC method to compress a batch of RVs (RV_i, $i = 0, 1, 2, 3, 4$), where $N = 9$, $\mathrm{IPres}_0 = 60.000000$, $\mathrm{IPres}_1 = 59.999962$, $\mathrm{IPres}_j = 0\,(j = 2, 3, \ldots, 8)$; $M = 8$; $\mathcal{L}(\mathrm{RV}) = 32$ b, and $\mathcal{L}(\mathrm{CW}) = 4$ b. In this example, RV_0–RV_2 equals one of the IPres, so they are only assigned with CWs in order, but no BDs information is required. For RV_3, since it does not match any of the IPres, the BDs are calculated. Although only the last 5 b contain the valid information of the BDs, the last 8 b are recorded as $\mathrm{BDs}_{\mathrm{valid}}$ according to the criterion (2). Then, RV_3 is added to the PVs buffer as the new IPres_0. RV_4 has a similar compression process as RV_3, which has a longer $\mathrm{BDs}_{\mathrm{valid}}$. The CR of this example is 3.63, which means that 72.5% of the space was saved after the compression.

III. Phase Angle Compression Method

A. UHD Phase Angle Data Characteristics

Fig. 4 shows one sample frame of UHD phase angle data. The reporting rate of this example is 1440 Hz, where 10 frames are

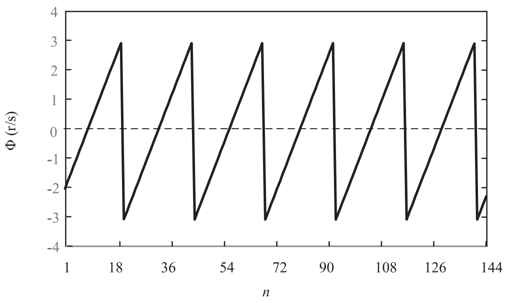


Fig. 4. UHD phase angle data sample frame.

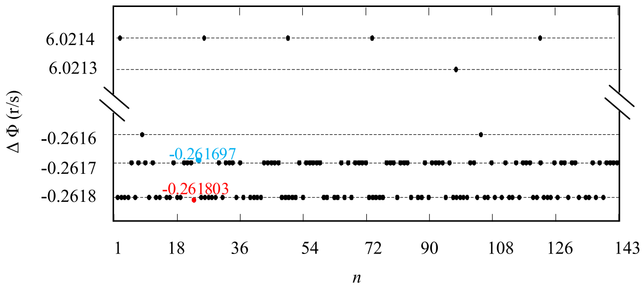


Fig. 5. $\Delta\phi(n)$ of UHD phase angle data.

sent per second and each has $n = 144$ measurement points. As can be seen from Fig. 4, the phase angle data has a relatively fixed slope rate in each cycle.

Fig. 5 shows a slope distribution of UHD phase angle data, which is the difference value between two consecutive values $\Delta\phi(n)$

$$\Delta\phi(n) = \phi(n) - \phi(n-1). \tag{6}$$

As is seen from Fig. 5, UHD $\Delta\phi(n)$ data are highly repetitive. However, the repetition is intermittent.

It is mentioned in Section I that the Huffman table information should be reconstructed for every next frame if the Huffman encoding method is exploited to compress the UHD phase angle data. This is because each frame has a unique Huffman table, and the loss of a single frame will lead to the decompression failure for the subsequent frames if their Huffman tables are dependent. However, including the Huffman table information introduces more information to each frame, restricting the CR. Considering that the different values in the Huffman table variate slightly due to the high sampling rate, this article exploits the TSSC method as stage 3 to compress the Huffman table information, forming the Huffman-TSSC compression method.

B. Preprocessing Algorithms

There are several methods proposed for the preprocessing, including the slack reference encoding (SRE), the swinging door (SD), the piecewise linear online trending, and the difference encoding [8]–[12]. The SRE uses one node as a reference to predict the others, which is only applicable for a multinodes system but not for independent PMU data online compression. SD and PLOP are lossy methods that replace raw data with a set of parameters under given error limits; these methods cannot maintain data accuracy. Besides, Tate [6] proposes the Difference Encoding With Variable Frequency Compensation (DEFC) method to predict phase angle values by compensating a frequency-related element. As UHD $\Delta\phi(n)$ has lower entropy than raw data, delta encoding (DE) that transforms raw data into the differences between sequential data via (6) shows the potential of compressing UHD phase angle data [11].

C. Encoding Techniques

Various algorithms have been proposed to encode the preprocessed data. These algorithms include run-length encoding (RLE), the Huffman compression method, etc.

The RLE simplifies the consecutive repeated values into a single value and the number of its repetition. This method is efficient when the same data value occurs consecutively [13]. However, it is unable to have a good performance for UHD phase angle data as the preprocessed data occurs intermittently.

The Huffman compression method is one of the most widely used techniques. It first builds the Huffman table, which consists of the different values and their occurrence frequency. Then, a Huffman tree is constructed, which assigns the CWs with fewer bits to the values that occur more frequently [11]. Unlike the RLE, Huffman encoding does not require the data to be consecutively repeated. It compresses the data efficiently if it has some frequently observed values. Additionally, it is easy to be implemented and faster than dictionary encoding methods.

Technically, the limit of the CR for the Huffman-based method is

$$\begin{aligned}\lim_{n\to\infty}(\mathrm{CR}) &= \frac{n \times \mathcal{L}(\mathrm{RV}) + \mathcal{L}(\mathrm{FrameHeader})}{\sum_{i=0}^{n-1}\mathcal{L}(\mathrm{CW}_{i\_\mathrm{Huff}}) + \mathcal{L}(\mathrm{FrameHeader})}\\ &= \mathcal{L}(\mathrm{RV})\end{aligned} \tag{7}$$

where $\mathrm{CW}_{i\_\mathrm{Huff}}$ is the Huffman CW.

D. Huffman-TSSC Phase Angle Compression Algorithm

Since the Huffman table introduces more information to each frame, restricting the CR, the TSSC method is exploited to compress the different values in the Huffman table. Fig. 6 shows the stages of the Huffman-TSSC method for phase angle compression. Stage 1 is to preprocess the raw data to the intermediate form $\Delta\phi(n)$ by DE, and then the Huffman table is built for $\Delta\phi(n)$. Then, in stage 2, a Huffman tree is built, which assigns shorter CWs to $\Delta\phi(n)$ that occurs more frequently, so that $\Delta\phi(n)$ can be compressed to some CWs with less bits. In stage 3, the TSSC method is used to compress the Huffman table, where $N = 1$, $M = 4$, and XOR are operated between any two consecutive different values of the Huffman table.

Fig. 7 shows an example of the proposed Huffman-TSSC method. As is seen, the data are first preprocessed to $\Delta\phi(n)$ that could repeat. Then, a Huffman table is built for $\Delta\phi(n)$. After that $\Delta\phi(n)$ that occurs the most frequently is compressed to the shortest CWs based on the Huffman tree. Finally, the Huffman table is compressed to CWs and $\mathrm{BDs}_{\mathrm{valid}}$ via the TSSC method, so that the Huffman table can take up fewer bits. In this example, the CR is increased by 20%.

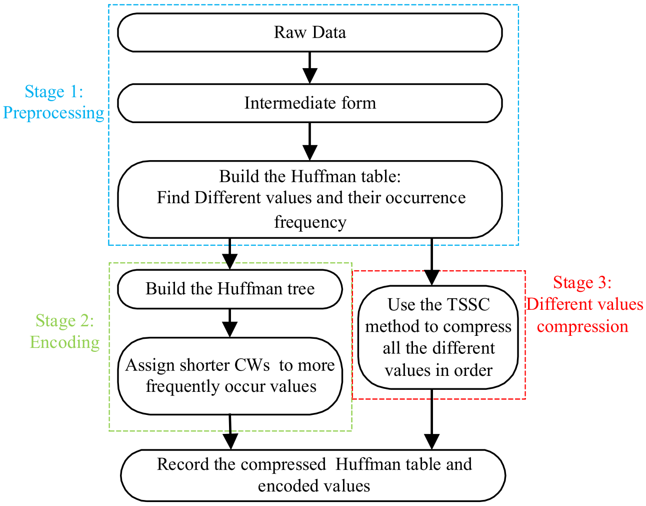
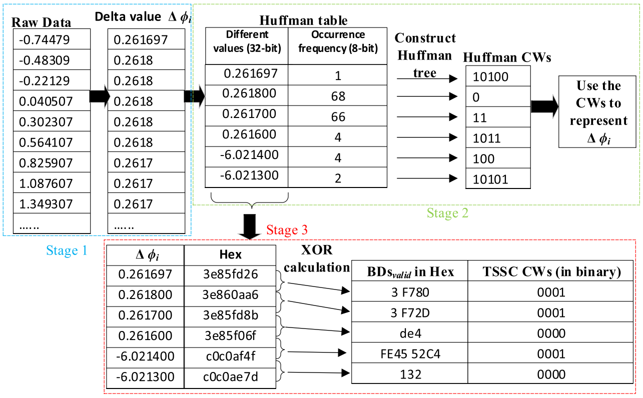

Stage 1:
Preprocessing
Raw Data
Intermediate form
Build the Huffman table:
Find Different values and their occurrence frequency
Stage 2:
Encoding
Build the Huffman tree
Assign shorter CWs to more frequently occur values
Use the TSSC method to compress all the different values in order
Stage 3:
Different values compression
Record the compressed Huffman table and encoded values

Fig. 6. Flowchart of the DE-Huffman-TSSC compression method.



Raw Data	Delta value Δ ϕ_i
-0.74479	0.261697
-0.48309	0.2618
-0.22129	0.2618
0.040507	0.2618
0.302307	0.2618
0.564107	0.2618
0.825907	0.2617
1.087607	0.2617
1.349307	0.2617
……	……

Huffman table

Different values (32-bit)	Occurrence frequency (8-bit)	Huffman CWs
0.261697	1	10100
0.261800	68	0
0.261700	66	11
0.261600	4	1011
-6.021400	4	100
-6.021300	2	10101

Δ ϕ_i	Hex
0.261697	3e85fd26
0.261800	3e860aa6
0.261700	3e85fd8b
0.261600	3e85f06f
-6.021400	c0c0af4f
-6.021300	c0c0ae7d

BDs$_{valid}$ in Hex	TSSC CWs (in binary)
3 F780	0001
3 F72D	0001
de4	0000
FE45 52C4	0001
132	0000

Fig. 7. Huffman-TSSC compression example.

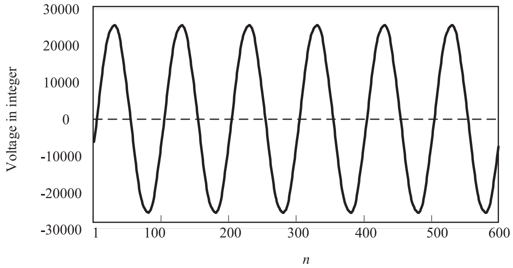


Fig. 8. UHD POW (voltage) data sample.

IV. POW Data Compression

A. UHD POW Data Characteristics

Fig. 8 shows a UHD POW example, where the values are expressed in 16-b integers. As is seen, the POW signal is almost a sinusoidal wave, which can be expressed as

$$x\left(k\right) \ = A\mathrm{sin}\left(\omega T_s k + \varphi\right) \tag{8}$$

where A, ω, and φ are the amplitude, angular frequency, and the phase angle of the POW. A, ω, and φ can change between every two points in practical. However, since the sampling rate is quite high, the changes of A and ω can be very small.

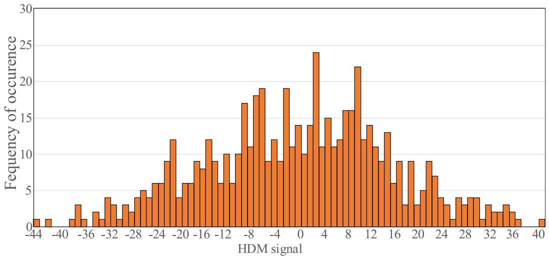


Fig. 9. Example of frequency occurrence of HDM signal.

Fig. 9 shows a sample frame of the HDM signal, which contains 600 points. As mentioned in Section III, the Huffman table needs to be reconstructed for every frame if Huffman method is used. However, it can be seen from Fig. 9 that the residuals do not tend to repeat frequently in one frame, making Huffman algorithm unable to compress it well.

To compress the UHD POW data in real time, this article proposes a CHDM-VB method. The CHDM model that takes A, ω, and φ as inconstant parameters is deducted. Based on the model, a criterion to determine the optimal differential order is established, thus the optimal HDM signal is obtained. Differential operation is then carried out cyclically for the obtained data to further reduce the VBs. Since the data processed by the aforementioned approaches have fewer VBs, it can be better compressed via the VB-based CW assignment method. Since this algorithm simply employs arithmetic operations, its running speed is higher than the FFT-based methods.

B. CHDM-VB Compression Method

Fig. 10 shows a flowchart of the algorithm, where the compression steps are as follows.

Step 1: CHDM.

Assuming $x_0(k-1) = A_{0(k-1)}\mathrm{sin}[\omega_{0(k-1)}(k-1) + \varphi_{0(k-1)}]$ is the POW value at time $t_{\mathrm{k}-1}$ and $x_0(k) = A_{0(k)}\mathrm{sin}[\omega_{0(k)}k + \varphi_{0(k)}]$ is the next point of value, the difference value between these two values, $x_1(k) = x_0(k) - x_0(k-1)$, can be derived as

$$x_1\left(k\right) A_{1(k)}\cos\left[\omega_{1(k)}k + \varphi_{1(k)}\right] \tag{9}$$

where $A_{1(k)} = 2\mathrm{sin}[(\varphi_{0(k)} - \varphi_{0(k-1)})/2] \times A_{0(k)}$, $\omega_{1(k)} = \omega_{0(k)}$ and $\varphi_{1(k)} = (\varphi_{0(k)} + \varphi_{0(k-1)})/2$.

As is seen from (9), by calculating the difference value between any two adjacent points, the POW data are transferred to a new set of data, where the amplitude $A_{1(k)}$ is much smaller than $A_{0(k)}$. The dataset $x_1(k)$ is termed as the first DM.

Similarly, the second DM dataset can be calculated by

$$x_2\left(k\right) \ = x_1\left(k\right) - x_1\left(k-1\right). \tag{10}$$

$x_2(k)$ can also be expressed as a sinusoidal wave $x_2(k) \approx A_{2(k)}\mathrm{sin}[\omega_{2(k)}k+\varphi_{2(k)}]$, where $\varphi_{2(k)} = (\varphi_{1(k)} + \varphi_{1(k-1)})/2$, $A_{2(k)} = -2\mathrm{sin}[(\varphi_{0(k)} - \varphi_{0(k-2)})/4] \times A_{1(k)}$, and $\omega_{2(k)} = \omega_{0(k)}$.

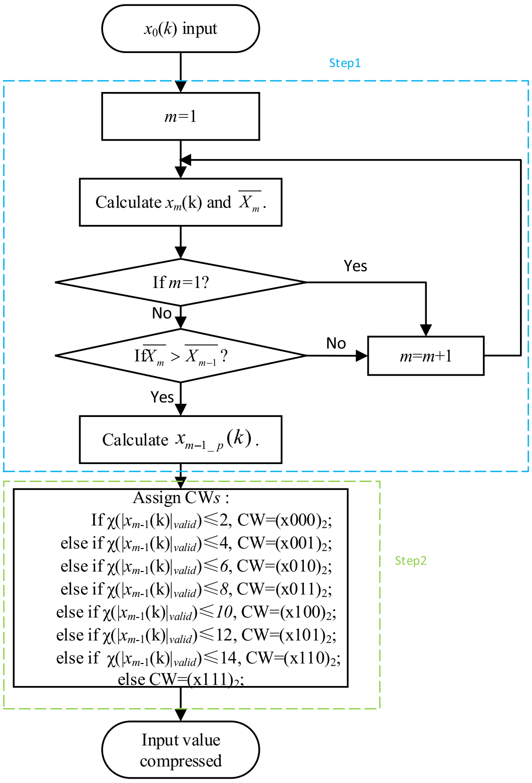

$x_0(k)$ input
Step1
m=1
Calculate x_m(k) and $\overline{X_m}$.
If m=1?
Yes
No
If $\overline{X_m} > \overline{X_{m-1}}$?
No
m=m+1
Yes
Calculate $x_{m-1\_p}(k)$.
Step2
Assign CWs:
If $\chi(|x_{m\text{-}1}(\mathrm{k})|_{valid})\leqslant 2$, CW=$(\mathrm{x}000)_2$;
else if $\chi(|x_{m\text{-}1}(\mathrm{k})|_{valid})\leqslant 4$, CW=$(\mathrm{x}001)_2$;
else if $\chi(|x_{m\text{-}1}(\mathrm{k})|_{valid})\leqslant 6$, CW=$(\mathrm{x}010)_2$;
else if $\chi(|x_{m\text{-}1}(\mathrm{k})|_{valid})\leqslant 8$, CW=$(\mathrm{x}011)_2$;
else if $\chi(|x_{m\text{-}1}(\mathrm{k})|_{valid})\leqslant 10$, CW=$(\mathrm{x}100)_2$;
else if $\chi(|x_{m\text{-}1}(\mathrm{k})|_{valid})\leqslant 12$, CW=$(\mathrm{x}101)_2$;
else if $\chi(|x_{m\text{-}1}(\mathrm{k})|_{valid})\leqslant 14$, CW=$(\mathrm{x}110)_2$;
else CW=$(\mathrm{x}111)_2$;
Input value compressed

Fig. 10. CHDM-FLA algorithm flowchart.

Such differential operations can be repeated many times. However, it can be seen that $A_{2(k)}$ is not necessarily smaller than $A_{0(k)}$. When the differential order increases to a certain number, the amplitude could increase. In this case, the compressed data have more VBs, which requires more CWs and decreases the CR.

A criterion is set to determine the optimal differential order when the dataset has fewer VBs, which is expressed as

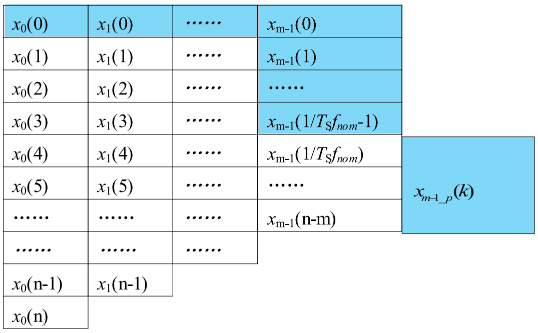
$$\overline{X_m} = \frac{\sum_{k=0}^{n-m} \mathcal{L}\left(|x_m(k)|_{\text{valid}}\right)}{n-m} \tag{11}$$

where $\mathcal{L}(|x_m(k)|_{\text{valid}})$ is the VBs of the absolute value of $x_m(k)$. The smaller the $|x_m(k)|$, the smaller the $\overline{X_m}$, the higher the CR. Therefore, if $\overline{X_m} > \overline{X_{m-1}}$, $x_{m\text{-}1}(k)$ is considered as the optimal differential dataset that has fewer VBs.

$x_{m\text{-}1}(k)$ is a sinusoidal wave that has the same cycle as $x_0(k)$, so the values tend to be cyclical. Therefore, the high-order difference dataset $x_{m\text{-}1}(k)$ can be processed by a periodically differential operation as follows:

$$x_{m-1_p}(k) = x_{m-1}\left(k + \frac{1}{T_s f_{nom}}\right) - x_{m-1}(k). \tag{12}$$

Fig. 11 shows the CHDM table that indicates the CHDM compression process. As is seen, the high-order DM datasets (x_1, …, $x_{m\text{-}1}$, x_m) are calculated. Based on the criterion in (11), the optimal high-order differential dataset $x_{m\text{-}1}$ is determined. Then, the differential operation is processed periodically for $x_{m\text{-}1}$. The original data are therefore transferred to the intermediate form that is colored in the blue background.



$x_0(0)$	$x_1(0)$	……	$x_{\text{m-1}}(0)$	
$x_0(1)$	$x_1(1)$	……	$x_{\text{m-1}}(1)$	
$x_0(2)$	$x_1(2)$	……	……	
$x_0(3)$	$x_1(3)$	……	$x_{\text{m-1}}(1/T_S f_{nom}\text{-}1)$	
$x_0(4)$	$x_1(4)$	……	$x_{\text{m-1}}(1/T_S f_{nom})$	$x_{m\text{-}1\_p}(k)$
$x_0(5)$	$x_1(5)$	……	……	
……	……	……	$x_{\text{m-1}}(\text{n-m})$	
……	……	……		
$x_0(\text{n-1})$	$x_1(\text{n-1})$			
$x_0(\text{n})$				

Fig. 11. CHDM table.

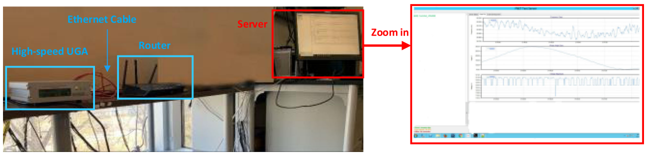


Fig. 12. Online experiment system setup.

Step 2: VB-based CW assignment.

The CHDM-preprocessed values are not highly repeated, which makes it difficult to be compressed by the entropy methods. However, they have fewer VBs compared with RVs. Therefore, they can be efficiently compressed by discarding the prefixed meaningless zeros and assigning fix-length CWs, which is similar to Step 3 of the ITSSC method. Note that the CWs for POW data consist of two parts, i.e., a sign bit that indicates if $x_{m\text{-}1}(k)$ is positive or negative and several bits that represent the VBs of $|x_{m\text{-}1}(k)|$.

V. Data Compression Tests

In this article, experiments are carried out in offline and online modes, using the field-collected UHD data of the wide-area monitoring system, FNET/GridEye, and Oak Ridge National Laboratory (ORNL) [20]–[22]. In the offline tests, the frame header is not considered, i.e., $\mathcal{L}(\text{FrameHeader}) = 0$. In the online tests, the algorithms are implemented in IEEE C37.118.2 PMU communication protocol and L(FrameHeader) = 32 B. Since the data frame complies with the standard protocol, there is no need to change the original circuit of the measurement device.

Note that each of the frame header has only one timestamp. This timestamp is related to the first measurement point of the dataset. Since the data in each frame are consecutive in time series, the rest timestamps of the data can be easily obtained when the sampling rate is known. For example, assuming that the sampling rate is 1000 Hz and the reporting rate is 10 Hz, each of a data frame consists of 100 measurement values. If the timestamps of the 100 values are “2020-01-01 00:00:00.000,” “2020-01-01 00:00:00.001,” “2020-01-01 00:00:00.002,” …, “2020-01-01 00:00:00.099,” respectively, then the timestamp of the frame will be “2020-01-01 00:00:00.000.”

Fig. 12 shows the experiment system setup, where the compression methods are implemented into a PMU variation, UGA. The UGA is connected to a router via an ethernet cable, and the

TABLE I
UHD FREQUENCY CR OF COLLECTED DATA

N M	2	4	8	12	14	16	24	28	30	32
1	3.17	3.31	3.35	3.38	3.39	3.40	3.43	3.45	3.45	3.46
2	3.65	3.72	3.76	3.82	3.84	3.85	3.40	3.40	3.41	3.41
4	3.82	3.90	**4.00**	3.60	3.61	3.63	3.69	3.31	3.33	3.34
8	3.38	3.50	3.24	3.31	3.02	3.04	3.10	3.12	2.62	2.62
16	2.25	2.18	2.12	2.18	2.20	2.07	2.13	2.15	2.16	2.03

2 b	3 b	4 b	5 b	6 b

Note: The shading represents $\mathcal{L}$(CW).

TABLE II
REAL-TIME CR OF DIFFERENT ALGORITHMS FOR UHD FREQUENCY DATA

Method	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	ACR
ITSSC	3.23	3.51	3.11	3.51	3.92	3.16	4.28	3.47	3.62	3.57	3.51
Huffman-based	3.05	3.20	3.15	3.45	3.68	3.05	4.31	3.47	3.56	3.07	3.34

router is connected to the remote router that the server is connected. The remote testing server decompresses the compressed data frames and shows the data plot in real time.

The frequency and phase angle data are 32-b values and their reporting rates are 1440 Hz, where 10 frames are sent per second and each has $n = 144$ measurement points.

A. UHD Frequency Compression Tests

1) Effect of Different Algorithm Settings: To find the optimal parameters (N and M) for the ITSSC method, tests are conducted with the data collected by the high-speed UGAs. The results are shown in Table I, where different shading indicates different $\mathcal{L}$(CW).

As is seen from Table I, when M is fixed, the CR increases with N if $\mathcal{L}$(CW) does not change as more values are in the IPres. However, if $\mathcal{L}$(CW) becomes larger with N, the CR becomes smaller. This is because the CW, in this case, takes up more bits, which offsets the positive effects of the increase of N on the compression performance. For M, when $\mathcal{L}$(CW) is fixed, the CR decreases with it as more meaningless prefix zeroes are recorded. When the increase of M decreases $\mathcal{L}$(CW), the CW takes up less space while more prefix values are recorded. Therefore, the CR could become either larger or smaller.

The CR is 4.0 when $M = 4$, $N = 8$, which is the highest among all the test results and is marked in bold in Table I. Therefore, they are the optimal parameters of the ITSSC method.

2) Online Compression Experiments: The proposed ITSSC method with optimal parameters is tested by implementing the code into the UGA. As a reference, the traditional Huffman compression method combined with the DE preprocessing is also tested, where the results are shown in Table II. As is seen, the average CR (ACR) of the ITSSC method is 3.51, exceeding that of the Huffman-based method, 3.34. Notice that the CR of the ITSSC method shown in Table I is greater than that shown in Table II. This is mostly because the frame size is 0.

TABLE III
REAL-TIME ACR OF DIFFERENT ALGORITHMS FOR UHD PHASE ANGLE DATA

Stage 1 Stage 2& Stage 3	DE	1st LP	DEFC
TSSC	3.78	3.83	3.56
Huffman	8.10	5.32	2.13
Huffman-TSSC	8.90	5.71	2.23

Note that the experiments are carried out under day-to-day operations, where events (generation trip, load disconnection, oscillation) and noises (mainly spikes) are constantly observed. Based on the performance evaluation, we believe that the proposed methods can achieve a good CR under these scenarios.

B. UHD Phase Angle Compression Tests

1) Different Algorithms Comparison: Table III shows the compression performance of the Huffman-TSSC method compression performance. As references, the TSSC-based and the Huffman-based methods are also tested. Additionally, different preprocessing algorithms are combined. As is seen from Table III, DE, first LP, and DEFC perform similarly with the TSSC-based method for UHD phase angle data, where first LP has the highest CR 3.83 among them. For the Huffman-based encoding algorithms, the DE combined method has the highest CR 8.1 among all the tests. This method is two times of the TSSC-based method. However, when combined with DEFC, the Huffman-based method has the lowest CR 2.13, which means that the DEFC is not suitable to preprocess UHD phase angle data. When using the TSSC method to compress the Huffman table information, the ACR can be increased by 10%, where the DE-processed algorithm has the highest ACR, which is 8.9. Therefore, the proposed Huffman-TSSC method that is preprocessed by DE is optimal for UHD phase angle data compression.

2) Online Compression Experiments for Phase Angle Data: The proposed Huffman-TSSC method is tested for several

TABLE IV
REAL-TIME CR OF THE HUFFMAN-TSSC METHOD FOR UHD PHASE ANGLE DATA

Day 1	Day 2	Day 3	Day 4	Day 5
6.68	7.02	7.28	7.36	7.15
Day 6	Day 7	Day 8	Day 9	Day 10
8.10	6.25	7.81	6.41	7.67

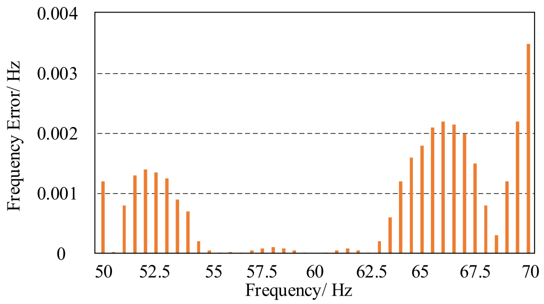


Fig. 13. Frequency error of the high-speed UGAs.

days in PMUs. The results are shown in Table IV. As is seen from Table IV, the proposed method compresses the UHD phase angle data efficiently. The ACR of experiments is 7.2, which is less than the offline CR due to the frame header.

C. UHD Synchrophasor Data Online Compression Tests

1) Compression Ratio: Since the frequency and the phase angle data are usually monitored simultaneously, they will be sent out at the same time as well. Therefore, it is more efficient for communication to compress them into one frame. To confirm the ACR for the UHD UGAs, experiments are also carried out to compress both the frequency and phase angle data.

The ITSSC method is used for the UHD frequency compression, where the parameters are $M = 4$ and $N = 8$. For UHD phase angle data, the proposed Huffman-TSSC method is used. Although the literature on the lossless compression of streaming data for power system monitoring is limited [23], it is reported that the ACR of the lossless compression methods that has been reported is 3 [24]. In the experiments of this article, the ACR of the UHD sychrophasor data is 4.90, which exceeds the previously reported state of the art by the time this article was written.

2) PMU Reporting Latency: The high-speed UGA is designed for both measuring and protecting power systems. As shown in Fig. 13, the frequency error of the high-speed UGA is less than 0.004 Hz, which satisfies the IEEE C37.118.1-2014 standard for both P-class (protection) and M-class (measurement) PMUs. It is also stated in the standard that the PMU real-time output reporting latency shall below $2/F_s$ and $7/F_s$ for P-class and M-class PMUs, respectively, where F_s is the reporting rate and $F_s = 10$ Hz in this article

This article measures the reporting latency of the high-speed UGAs with the proposed compression method implemented. The reporting latency during different days is shown in Fig. 14. As is seen, the maximum reporting latency is 0.118 s, which is lower than the IEEE standard limits 0.2 and 0.7 s for the P-class and M-class PMUs, respectively.

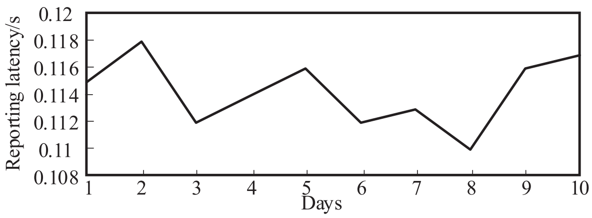


Fig. 14. Reporting latency for high-speed UGA with sychronphasor data compression methods implemented.

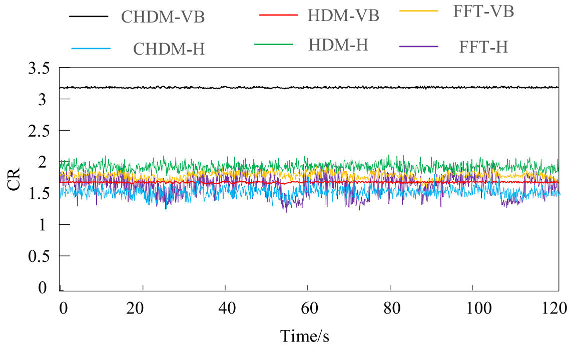


Fig. 15. CR of different algorithms for UHD POW data.

TABLE V
ACR OF DIFFERENT METHOD FOR FNET UHD POW DATA IN REAL TIME

CHDM-VB	CHDM-H	HDM-VB	HDM-H	FFT-VB	FFT-H
3.19	1.53	1.91	1.62	1.80	1.61

D. UHD POW Data Compression Tests

1) FNET/GridEye Data Tests: The POW data are 16 b and sampled at 6000 Hz. Fig. 15 shows the compression performance of the proposed CHDM-VB method, and several combined methods are also tested as references. As is seen from Fig. 15, the CHDM-VB method can always compress POW data more aggressively than the others. The CR of the CHDM-VB method is about 3.2, whereas that of the other methods are 1.2–2.

Table V lists the ACRs of the proposed and reference methods, where the Huffman method is abbreviated as "H." As can be seen, the proposed cyclical difference calculation can dramatically increase the ACR from 1.8 (FFT-VB) to 3.19 (CHDM-VB). Additionally, the Huffman coding does not perform well for online UHD POW data compression compared with the VB method due to the nonrepeatability of the preprocessed POW data. It is also noticed that the ACR of the proposed CHDM-VB method is higher than the ACR that has been reported, which is 3, thus further confirming the compression performance of the CHDM-VB method.

The reporting latency of the high-speed UGA with the CHDM-VB implemented is tested for the UHD POW data. The maximum reporting latency during the ten-day experiment is 0.125 s, which is lower than the IEEE standards limits 0.2 and 0.7 s for the P-class and M-class PMUs.

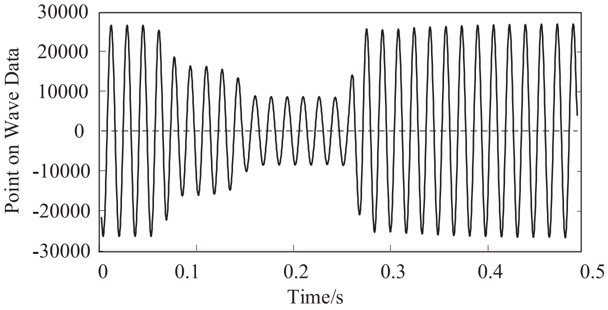


Fig. 16. ORNL POW data.

TABLE VI
ACR OF DIFFERENT METHOD FOR ORNL UHD POW DATA

CHDM-FL	CHDM-H	HDM-FL	HDM-H	FFT-FL	FFT-H
3.33	1.60	1.58	1.40	1.20	0.90

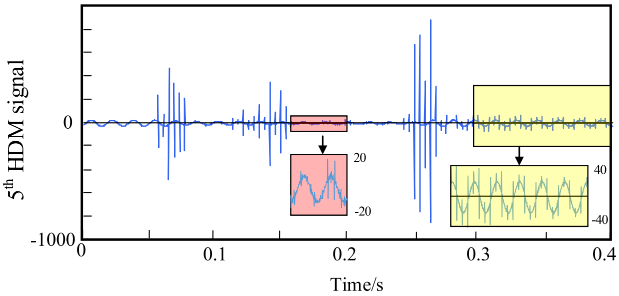


Fig. 17. HDM signals.

2) ORNL Data Tests: Fig. 16 shows a sample of the UHD POW data collected by ORNL at a sampling rate of 3840 Hz, where an event occurs at 0.06 s and disappears at around 0.28 s. Use CHDM-VB method to compress the POW data, and compare it with other reference methods. The compression performance is listed in Table VI. As is seen from Table VI, the CHDM-VB method also has a good compression performance for the 3840-Hz POW data, whose ACR is 3.33, whereas the ACRs of other methods are less than 1.6.

Fig. 17 shows the optimal HDM differential signals for the ORNL-collected POW data shown in Fig. 16, which contains an event and has an ACR of 3.12. As is seen, the high-order DM signal has some spikes due to the magnitude change of the raw POW dada, which may affect the compression performance of the method. However, most of the points still have less VBs, making it possible to be compressed well.

VI. Conclusion

This article proposed several methods to compress the UHD synchrophasor and POW data in a lossless and real-time manner. The methods were designed for advanced PMUs to compress data packets efficiently and quickly before its transmission to the server. The proposed methods addressed the communication efficiency and memory space issues on the server side.

The proposed methods were tested with the field-collected data, and the algorithms were implemented in a high-speed PMU to compress data in real time. The results showed that the proposed methods have extraordinary compression efficiency, where the average CR of the synchrophasor data was 4.9, and that of the POW data was 3.3.

References

[1] P. H. Gadde, M. Biswal, S. Brahma, and H. Cao, “Efficient compression of PMU data in WAMS,” *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2406–2413, Sep. 2016.

[2] M. Cui, J. Wang, J. Tan, A. R. Florita, and Y. Zhang, “A novel event detection method using PMU data with high precision,” *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 454–466, Jan. 2019.

[3] W. Wang, C. Chen, W. Yao, K. Sun, W. Qiu, and Y. Liu, “Synchrophasor data compression under disturbance conditions via cross-entropy-based singular value decomposition,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2716–2726, Apr. 2021.

[4] M. H. F. Wen and V. O. K. Li, “Optimal phasor data compression unit installation for wide-area measurement systems—An integer linear programming approach,” *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2644–2653, Nov. 2016.

[5] F. Zhang, L. Cheng, X. Li, Y. Sun, W. Gao, and W. Zhao, “Application of a real-time data compression and adapted protocol technique for WAMS,” *IEEE Trans. Power Syst.*, vol. 30, no. 2, pp. 653–662, Mar. 2015.

[6] J. E. Tate, “Preprocessing and Golomb–Rice encoding for lossless compression of phasor angle data,” *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 718–729, Mar. 2016.

[7] H. Li, N. Sheng, and L. Zhi, “WAMS/PMU data pre-processing and compression,”*Adv. Mater. Res.*, vol. 986/987, pp. 1700–1703, Jul. 2014.

[8] F. Xiaodong, C. Changling, L. Changling, and S. Huihe, “An improved process data compression algorithm,” in *Proc. 4th World Congr. Intell. Control Autom.*, Shanghai, China, 2002, vol. 3, pp. 2190–2193.

[9] J. D. A. Correa, A. S. R. Pinto, C. Montez, and E. Leão, Swinging door trending compression algorithm for IoT environments,” in *Proc. Companion Proc. 9th SBESC*, 2019, pp. 143–148.

[10] M. A. Khan, J. W. Pierre, J. I. Wold, D. J. Trudnowski, and M. K. Donnelly, “Impacts of swinging door lossy compression of synchrophasor data,” *Int. J. Elect. Power Energy Syst.*, vol. 123, 2020, Art. no. 106182, doi: 10.1016/j.ijepes.2020.106182.

[11] J. Uthayakumar, T. Vengattaraman, and P. Dhavachelvan, “A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications,” *J. King Saud Univ. - Comput. Inf. Sci.*, 33, 2021, pp. 119–140.

[12] Z. Jellali, L. Najjar Atallah, and S. Cherif, “Linear prediction for data compression and recovery enhancement in wireless sensors networks,” in *Proc. Int. Wireless Commun. Mobile Comput. Conf.*, Paphos, Cyprus, 2016, pp. 779–783.

[13] D. Nagamalai, F. Renault, and M. Dhanuskodi, “Advances in digital image processing and information technology,” in *Proc. 1st Int. Conf. Digit. Image Process. Pattern Recognit.*, Tirunelveli, India, 2011, pp. 23–25.

[14] R. Klump, P. Agarwal, J. E. Tate, and H. Khurana, “Lossless compression of synchronized phasor measurements,” in *Proc. IEEE PES General Meeting*, Providence, RI, USA, 2010, pp. 1–7.

[15] R. Wenyu, Y. Timothy, and N. Klara, “ISAAC: Intelligent synchrophasor data real-time compression framework for WAMS,” in *Proc. IEEE Int. Conf. Smart Grid Commun .*, 2017, pp. 430–436,.

[16] K. Gibson, D. Lee, J. Choi, and A. Sim, “Dynamic online performance optimization in streaming data compression,” in *Proc. IEEE Int. Conf. Big Data*, Seattle, WA, USA, 2018, pp. 534–541.

[17] A. Unterweger and D. Engel, “Lossless compression of high-frequency voltage and current data in smart grids,” in *Proc. IEEE Int. Conf. Big Data*, 2016, pp. 3131–3139.

[18] K. Zhiwu, X. Rui, L. Xianling, and Y. Rui, “Research on lossless compression technique basing on running-data of nuclear power plant,” in *Proc. Int. Conf. Comput. Intell. Commun. Netw.*, 2015, pp. 956–959.

[19] D. Zhang, Y. Bi, and J. Zhao, “A new data compression algorithm for power quality online monitoring,” in *Proc. Int. Conf. Sustain. Power Gener. Supply*, 2009, pp. 1–4.

[20] W. Yao *et al.*, “A fast load control system based on mobile distribution-level phasor measurement unit,” *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 895–904, Jan. 2020.

[21] W. Wang *et al.*, “Frequency disturbance event detection based on synchrophasors and deep learning,” *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3593–3605, Jul. 2020.

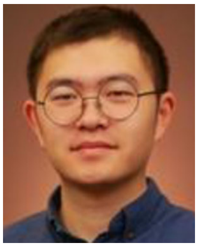
[22] W. Wang, W. Yao, C. Chen, X. Deng, and Y. Liu, "Fast and accurate frequency response estimation for large power system disturbances using second derivative of frequency data," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2483–2486, May 2020.

[23] X. Wang, Y. Liu, and L. Tong, "Adaptive Subband Compression for Streaming of Continuous Point-on-Wave and PMU Data," 2021. [Online]. Avaoilable: https://arxiv.org/abs/2008.10092.

[24] R. Jumar, H. Maaß, and V. Hagenmeyer,"Comparison of lossless compression schemes for high rate electrical grid time series for smart grid monitoring and analysis," *Comput. Elect. Eng.*, vol. 71, pp. 465–476, 2018.

Chang Chen (Student Member, IEEE) received the B.S. and Ph.D. degree in electrical engineering from Sichuan University, Sichuan, China, in 2015 and 2020, respectively.

Her research interests include wide-area power system measurement, harmonic modeling and analysis, power quality, etc.

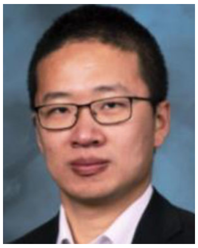


Weikang Wang (Student Member, IEEE) received the B.S. degree in computer science from the School of Control and Computer Engineering, North China Electric Power University, Beijing, China, in 2016. He is currently working toward the Ph.D. degree in computer engineering with the University of Tennessee, Knoxville, TN, USA.

His research interests include wide-area monitoring, situation awareness, big data, and machine learning.

He Yin (Member, IEEE) received the B.S. and Ph.D. degrees in electrical and computer engineering from the University of Michigan–Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai, China, in 2012 and 2017, respectively.

He is currently a Postdoctoral Researcher with the Center for Ultra-Wide-Area Resilient Electric Energy Transmission Networks, University of Tennessee, Knoxville, TN, USA. His research interests include optimization and decentralized control of microgird and PMU design.



Lingwei Zhan (Member, IEEE) received the B.S. and M.S. degrees from Tongji University, Shanghai, China, in 2008 and 2011, respectively, and the Ph.D. degree from University of Tennessee, Knoxville, TN, USA, in 2015, all in electrical engineering.

He is currently a Research and Development Staff with the Oak Ridge National Laboratory, Oak Ridge, TN, USA. His research interests include advanced grid sensors, PMU, synchrophasor measurement algorithm, wide-area power system monitoring, renewable energy sources, FACTs, and HVdc.

Yilu Liu (Fellow, IEEE) received the B.S. degree from Xian Jiaotong University, Xian, China, in 1982, and the M.S. and Ph.D. degrees from The Ohio State University, Columbus, OH, USA, in 1986 and 1989, respectively, all in electrical engineering.

She is currently the Governor's Chair with the University of Tennessee, Knoxville, TN, USA and Oak Ridge National Laboratory, Oak Ridge, TN, USA. Her current research interests include power system wide-area monitoring and control, large interconnection-level dynamic simulations, electromagnetic transient analysis, and power transformer modeling and diagnosis.

Dr. Liu is elected as the member of the National Academy of Engineering in 2016. She is also the Deputy Director of the DOE/NSF cofunded engineering research center CURENT.