Real-Time Lossless Compression for Ultra-High-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, Yilu Liu, Fellow, IEEE

Abstract—Modern advanced Phasor Measurement Units (PMUs) are developed with ultra-high 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 Ultra-High-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 paper 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 TSSC algorithm to compress the UHD phase angle data. Finally, a cyclic 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—lossless data compression, ultra-high density, synchrophasor data, point on wave data, Improved--TSSC method, Huffman encoding method

I. INTRODUCTION

With the increasing demand to analyze the complex power system dynamics, the Phasor Measurement Unit (PMU) is required to provide Ultra-High-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 Compression Ratio (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 is transferred to the intermediate form with low entropy, so that it can be encoded 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 Point on Wave (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. [19] proposes 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 paper 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 the N and the 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 Cylcical High-order Delta Modulation (CHDM) method is proposed for the UHD POW data due to its cyclicity. 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, a summary of the proposed methods is presented in Section VI.

II. IMPROVED-TSSC 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

| TSSC Method sets \( N = 3 \), while a larger \( N \) makes the processes RV more likely to match one of the IPres, thus a higher Compression Ratio (CR) could reach as no BD is calculated. In this case, the TSSC method sets \( M = 4 \) bits, 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 five bits, 11101, are valid. However, the recorded BDs will be 00011101, when \( M = 4 \), because it records the 3 prefix zeros to make up 4 digits. As opposed to it, when \( M = 2 \), the BDs will be 011101, which reduces 2 bits. 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.

To efficiently compress the UHD frequency data in real time, this paper proposes an Improve-TSSC (ITTS) 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 flow chart of the ITSSC method. As illustrated, this method transforms each value into a fixed-length CW and the BDs. Note that the ITSSC method

\[
\text{Fig. 1 UHD frequency data}
\]

\[
\text{Fig. 2 ITSSC compression algorithm diagram}
\]
compresses Raw Values (RVs) one after another due to the dependency of the current value and its predecessors. The specific steps to compress the \( i \)-th RV, \( R_V \), are as follows:

a) Step 1: Compare with immediate predecessors.

There is a vector of \( N \) IPres (\( \text{IPres}_0, \text{IPres}_1, \ldots, \text{IPres}_{SN-1} \)) in the allocated buffer, where \( \text{IPres}_0 \) 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 \( R_V \). In this way, \( R_V \) is transformed into a fixed-length CW with fewer bits.

b) Step 2: Calculate BDs.

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

\[
BD = R_V \oplus \text{IPres}_0
\]

where \( \oplus \) 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.

c) Step 3: Valid bits (VBs) of BDs determination, and VB-based CW assignment.

Per its definition, the BD indicates the variation between \( R_V \) and \( \text{IPres}_k \). For the UHD frequency data, the BDs are supposed to be small. However, BDs are, by default, 32-bit integers, which means only some lower bits can represent the difference. Discarding the meaningless prefix zeros to get the VBs of the BDs (\( \text{BD}_{\text{valid}} \)) 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, \( \text{BD}_{\text{valid}} \) is determined by the criterion as follows

\[
2^\left(k+1\right)M \leq BD \leq 2^{(k+1)^M}
\]

where \( k=0,1,2, \ldots, L(\text{BD}_{\text{valid}})/M-1 \), \( L(*) \) is the bit length calculation function. Once the criterion (2) is satisfied, the last \((k+1)M\) bits of the BDs are \( \text{BD}_{\text{valid}} \), and the corresponding CW will be stored according to the \( \text{BD}_{\text{valid}} \).

C. Compression Performance

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

\[
\text{CR} = \frac{n \times L(RV) + L(\text{FrameHeader})}{n \times L(CW) + L(\text{FrameHeader}) + \sum_{i=0}^{M} L(BD)_{\text{valid}}}
\]

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

\[
\lim_{n \to \infty, BD_{\text{valid}} \to 0} \left[ \frac{n \times L(RV) + L(\text{FrameHeader})}{n \times L(CW) + L(\text{FrameHeader}) + \sum_{i=0}^{M} L(BD)_{\text{valid}}} \right] = \frac{L(RV)}{L(CW)}
\]

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

\[
L(CW) = \left\lceil \log_2(N + \frac{L(RV)}{M}) \right\rceil
\]

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 \( N=9 \), \( \text{IPres}_0 = 60.000000 \), \( \text{IPres}_1 = 59.999962 \), \( \text{IPres}_2 = 0 \) \( (j=2,3, \ldots, 8) \); \( M=8 \). \( L(RV) = 32 \) bits, and \( L(CW) = 4 \) bits. 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 bits contain the valid information of the BDs, the last 8 bits are recorded as \( \text{BD}_{\text{valid}} \). According to the criterion (2), then, \( RV_3 \) is added to the PVs buffer as the new \( \text{IPres}_0 \). \( RV_4 \) has a similar compression process as \( RV_3 \), which has a longer \( \text{BD}_{\text{valid}} \). 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 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 \( \Delta \phi(n) \)

\[ \Delta \phi(n) = \phi(n) - \phi(n-1) \quad (6) \]

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

It is mentioned in the Introduction section 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 paper 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 (PLOT), and the difference encoding [8]-[12]. The SRE uses one node as a reference to predict the others, which is only applicable for a multi-nodes system but not for independent PMU data online compression. SD and PLOT are lossy methods that replace raw data with a set of parameters under given error limits; these methods cannot maintain data accuracy. Besides, [6] proposes the 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 occurred 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

![Flow chart of the DE-Huffman-TSSC compression method.](image)

![The Huffman-TSSC compression example.](image)

\[ \lim_{n \to \infty} \left( \sum_{i=0}^{n} \ell(CW_{i,Huffman}) \right) = \ell(RV) \] (7)

where \( CW_{i,Huffman} \) 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 is first processed to \( \Delta \phi(n) \) that could repeat. Then, a Huffman table is built for \( \Delta \phi(n) \). After that, the \( \Delta \phi(n) \) that occurs the most frequently are compressed to the shortest CWs based on the Huffman tree. Finally, the Huffman table is compressed to CWs and BDSvalid via the TSSC method, so that the Huffman table can take up fewer bits. In this example, the CR is increased by 20%.
IV. POINT ON WAVE DATA COMPRESSION

A. UHD Point on Wave Data Characteristics

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

\[ x(k) = A \sin(\omega k + \phi) \]  

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

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 paper proposes a CHDM-VB method. The Cyclical High-order Delta Modulation (CHDM) model that takes \( A, \omega, \) and \( \phi \) as inconstant parameters is deducted. Based on the model, a criterion to determine the optimal differential order is established, thus the optimal high-order delta modulation 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 flow chart of the algorithm, where the compression steps are as follows:

1. **Step 1: CHDM**
   
   Assuming \( x_{0}(k-1) = A_{0(k-1)} \sin[\omega_{0(k-1)}(k-1) + \phi_{0(k-1)}] \) is the POW value at time \( t_{k-1} \) and \( x_{0}(k) = A_{0(k)} \sin[\omega_{0(k)} + \phi_{0(k)}] \) 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
   
   \[ x_{1}(k) = A_{1(k)} \cos[\omega_{1(k)} + \phi_{1(k)}] \]  

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

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

   Similarly, the 2nd DM data set can be calculated by

   \[ x_{2}(k) = x_{1}(k) - x_{1}(k-1) \]  

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

   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 has more VBs, which requires more CWs and decreases the CR.

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

   \[ \overline{X}_{m} = \frac{v_{n-m}}{n-m} \]  

   where \( L(\overline{x}_{m}(k)|_{valid}) \) is the VBs of the absolute value of \( x_{m}(k) \).

   The smaller the \( \overline{x}_{m}(k) \), the smaller the \( \overline{X}_{m} \), the higher the CR.

   Therefore, if \( \overline{X}_{m} > \overline{X}_{m-1}, \overline{x}_{m-1}(k) \) is considered as the optimal differential data set that has fewer VBs.

   \( x_{m-1}(k) \) 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 data set \( x_{m-1}(k) \) can be processed by a periodically differential operation as follows.
work. For the \( x \) th data by the entropy Router-T, T-Router communication protocol and L(FrameHeader)=32 bytes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2021.3057034, IEEE

Since the data frame complies with the standard protocol, PMU communication protocol and L(FrameHeader)=32 bytes.

National Laboratory (ORNL) modes, using sign bit length CWs, which makes it difficult to be compressed.

intermediate form is transferred to the intermediate form that is colored in the blue background.

b) 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 raw values. Therefore, they can be efficiently compressed by discarding the prefixed meaningless zeros and assigning fixed length CWs, which is similar to Step 3 of the ITSSC method. Note that the CWs for POW data consists of two parts, i.e. a sign bit that indicates if \( x_{m-1}(k) \) is positive or negative and several bits that represent the VBs of \([x_{m-1}(k)]\).

Table 1: UHD Frequency Compression Ratio of Collected Data

<table>
<thead>
<tr>
<th>M</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>24</th>
<th>28</th>
<th>30</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.17</td>
<td>3.31</td>
<td>3.35</td>
<td>3.38</td>
<td>3.39</td>
<td>3.40</td>
<td>3.43</td>
<td>3.45</td>
<td>3.45</td>
<td>3.46</td>
</tr>
<tr>
<td>2</td>
<td>3.65</td>
<td>3.72</td>
<td>3.76</td>
<td>3.82</td>
<td>3.84</td>
<td>3.85</td>
<td>3.85</td>
<td>3.40</td>
<td>3.40</td>
<td>3.41</td>
</tr>
<tr>
<td>3</td>
<td>3.82</td>
<td>3.90</td>
<td>4.00</td>
<td>3.60</td>
<td>3.61</td>
<td>3.62</td>
<td>3.69</td>
<td>3.31</td>
<td>3.33</td>
<td>3.34</td>
</tr>
<tr>
<td>4</td>
<td>3.38</td>
<td>3.50</td>
<td>3.24</td>
<td>3.31</td>
<td>3.02</td>
<td>3.04</td>
<td>3.10</td>
<td>3.12</td>
<td>2.62</td>
<td>2.62</td>
</tr>
<tr>
<td>5</td>
<td>2.25</td>
<td>2.18</td>
<td>2.12</td>
<td>2.18</td>
<td>2.20</td>
<td>2.07</td>
<td>2.13</td>
<td>2.15</td>
<td>2.16</td>
<td>2.03</td>
</tr>
</tbody>
</table>

Note that each of the frame header has only one timestamp. This timestamp is related to the first measurement point of the data set. Since the data in each frame is 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 1000Hz and the reporting rate is 10Hz, 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. 11 The CHDM table](image)

\[
x_{m-1}(k) = x_{m-1}(k + \frac{1}{x_{m-1}}) - x_{m-1}(k)
\]

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

b) 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 raw values. Therefore, they can be efficiently compressed by discarding the prefixed meaningless zeros and assigning fixed-length CWs, which is similar to Step 3 of the ITSSC method. Note that the CWs for POW data consists of two parts, i.e. a sign bit that indicates if \(x_{m-1}(k)\) is positive or negative and several bits that represent the VBs of \([x_{m-1}(k)]\).

V. DATA COMPRESSION TESTS

In this paper, experiments are carried out in offline and online modes, using the field-collected UHD data of the wide-area monitoring system (WAMS), FNET/GridEye and Oak Ridge National Laboratory (ORNL) [20]-[22]. In the offline tests, the frame header is not considered, i.e. \(L(\text{FrameHeader})=0\). In the online tests, the algorithms are implemented in IEEE C37.118.2 PMU communication protocol and \(L(\text{FrameHeader})=32\) bytes. 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 data set. Since the data in each frame is consecutive in time.
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 Tab. 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 Tab. I is greater than that shown in Tab. II. This is mostly because the frame size is 0.

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 the proposed methods can achieve a good CR under these scenarios.

B. UHD Phase Angle Compression Tests

a) Different algorithms comparison

Tab. 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 Tab. III, DE, 1st LP, and DEFC perform similarly with the TSSC-based method for UHD phase angle data, where 1st 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 2 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.

b) Online compression experiments for phase angle data

The proposed Huffman-TSSC method is tested for several days in PMUs. The results are shown in Tab. IV. As is seen from Tab. 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

a) 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 paper, the ACR of the UHD synchrophasor data is 4.90, which exceeds the previously reported state-of-the-art by the time the paper was written.

b) 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 0.004Hz, 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

---

**TABLE II**

<table>
<thead>
<tr>
<th>Method</th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
<th>Day 6</th>
<th>Day 7</th>
<th>Day 8</th>
<th>Day 9</th>
<th>Day 10</th>
<th>ACR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITSSC</td>
<td>3.23</td>
<td>3.51</td>
<td>3.11</td>
<td>3.51</td>
<td>3.92</td>
<td>3.16</td>
<td>4.28</td>
<td>3.47</td>
<td>3.62</td>
<td>3.57</td>
<td>3.51</td>
</tr>
<tr>
<td>Huffman-based</td>
<td>3.05</td>
<td>3.20</td>
<td>3.15</td>
<td>3.45</td>
<td>3.68</td>
<td>3.05</td>
<td>4.31</td>
<td>3.47</td>
<td>3.56</td>
<td>3.07</td>
<td>3.34</td>
</tr>
</tbody>
</table>

**TABLE III**

<table>
<thead>
<tr>
<th>Stage 1</th>
<th>Stage 2&amp; Stage 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>1st LP</td>
</tr>
<tr>
<td>ITSSC</td>
<td>3.78</td>
</tr>
<tr>
<td>Huffman</td>
<td>8.10</td>
</tr>
<tr>
<td>Huffman-TSSC</td>
<td>8.90</td>
</tr>
</tbody>
</table>

**TABLE IV**

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.68</td>
<td>7.02</td>
<td>7.28</td>
<td>7.36</td>
<td>7.15</td>
</tr>
<tr>
<td>8.10</td>
<td>6.25</td>
<td>7.81</td>
<td>6.41</td>
<td>7.67</td>
</tr>
</tbody>
</table>

---

0278-0046 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information. This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2021.3057034, IEEE Transactions on Industrial Electronics.
paper

This paper 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 0.118s, which is lower than the IEEE standard limits 0.2s and 0.7s for the P-class and M-class PMUs, respectively.

D. UHD POW Data Compression Tests

a) FNET/GridEye data tests

The POW data is 16-bit and sampled at 6000Hz. 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 compression ratio of the CHDM-VB method is about 3.2, while that of the other methods are 1.2~2.

Tab. 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 non-repeatability 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 0.125s, which is lower than the IEEE standards limits 0.2 s and 0.7s for the P-class and M-class PMUs.

b) 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 0.06s and disappears at around 0.28s. Use CHDM-VB method to compress the POW data, and compare it with other reference methods. The compression performance is listed in Tab. VI. As is seen from Tab. VI, the CHDM-VB method also has a good compression performance for the 3840 Hz POW data, whose ACR is 3.33, while 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 data, 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. SUMMARY

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

The proposed methods are tested with the field-collected data, and the algorithms are implemented in a high-speed PMU to compress data in real time. The results show that the proposed methods have extraordinary compression efficiency, where the average compression ratio of the synchrophasor data is 4.9, and that of the POW data is 3.3.
VII. REFERENCES


