



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# Compressing Phasor Measurement Data

P. Top, J. Breneman

December 6, 2012

PES General Meeting  
Vancouver, Canada  
July 21, 2013 through July 25, 2013

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

# Compressing Phasor Measurement Data

Philip Top and John Breneman  
Engineering Department  
Lawrence Livermore National Lab  
Livermore, CA USA

**Abstract**—Phasor Measurement units (PMU's) are becoming standard equipment in electrical grids around the world. They are capable of generating a significant amount of data on a continuous basis from distribution and transmission networks. Several months of data from 2 PMU's situated on a distribution network were captured in raw format. This data was separated by measurement type and compressed using a number of standard compression algorithms. The results show the compressibility of PMU data for these algorithms and are used to estimate the space requirements for storing a day of data from a single PMU.

**Index Terms**—PMU, compression, szip, synchrophasors.

## I. INTRODUCTION

In recent years there has been a significant increase in the number and availability of Phasor Measurement Units (PMUs) for power system measurements. Applications include enhanced state estimation, wide area monitoring, and stability assessments; and numerous other uses are being researched. PMUs are synchronized using a GPS clock and capable of providing measurement updates at up to 60 Hz. The units can be connected via ethernet and the data is transmitted using the C37.118 standard [1]. Data packets commonly include: a timestamp, frequency, frequency rate of change, 3 phase voltages and 3 phase currents. The voltage and current data can be represented by real and imaginary components or as a magnitude and phase. Additional digital or analog channels can be defined depending on the measurement system.

The North American SynchroPhasor Initiative (NASPI), a consortium of various government, industrial, and academic institutions, has sought to improve power system reliability through the use of wide area measurement systems [2]. As an organization, they are promoting increased use of PMU technology in the North American power grids. At present, there are hundreds of the devices in operation across the country and within a decade there is potential for this number to increase to several thousands. While the data management issues for this amount of PMU data are within the realm of current technology, managing it effectively will require significant thought and effort. This is particularly true when conceiving of long term storage and accessibility.

In typical practice, PMU data is stored in a database such as OSISoft PI or another structured database such as Oracle or MySQL. Projects such as OpenPDC can also store data to various other file formats [3]. Typically, these formats store the data as binary values and can become very large quickly. PI has an optional compression algorithm that can be very effective at reducing the size of the data but it is a lossy compression that may be unsuitable depending on the technical and legal requirements for the data. In many cases, therefore, it is desirable to be able to archive the data in a lossless fashion.

In order to reduce expense and efficiently utilize available storage resources, the data must be compressed. Various lossy compression algorithms have been proposed for PMU and power system data. These make use of various compression techniques such as principle component analysis [4], and wavelets [5]. The applicability of these algorithms depends on the application and what characteristics of signal degradation are acceptable. Since future applications are unknown, a lossless compression might be better suited for data archival. Some research on the topic of lossless compression of PMU data [6] has been done. However, a more detailed exploration using a large quantity of real PMU data is required for gauging the potential impact of compression on storage for PMU data and as a benchmark for any specialized techniques.

## II. COMPRESSION ALGORITHMS

This study compares a set of compression algorithms and their performance on the various types of PMU data. The programs tested use freely available software to perform the compression. Five compression formats were tested: gzip, bzip2, 7-zip, szip, and szip using an optional flag that encodes the incremental values instead of the original data. These represent distinct compression methodologies and are available in easy-to-use software.

Gzip is the oldest algorithm and is based on the Lempel-Ziv 77 algorithm [7]. It creates a dictionary of common values and it can store the values or patterns based on the dictionary to reduce the size. The bzip2 compressor uses the Burrows-Wheeler algorithm [8]. The 7-zip algorithm utilizes the Lempel-Ziv-Markov algorithm designed for that program [9]. Finally, the szip algorithm is targeted at scientific data and uses the extended-rice algorithm for data

---

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

compression [10]. It has been used for earth science data with good results [11].

### III. DATA COLLECTION

Two Arbiter 1133a PMU units were set up to collect data from a 3-phase 120 V distribution circuit. The arbiter units were set up to collect voltage and frequency data at 60 Hz using the IEEE C37.118 protocol. The data was collected using a LabView application and stored as individual messages in pcap format. Each pcap file typically consisted of a single day of data. The data used in this research comprised of a generally continuous data set from July 20, 2012 to November 18, 2012 and in total comprised over 59 GB in the raw pcap format. The distribution circuits monitored are connected to the US Western Interconnect.

The circuit itself is in an office/lab building and contains a number of different loads, including computers, lighting, machinery and various motor loads for air conditioning and pumping. The voltage signal can range from very noisy with a lot of loads switching on and off to quiescent and very stable. Fig. 1 shows some samples of typical voltage patterns.

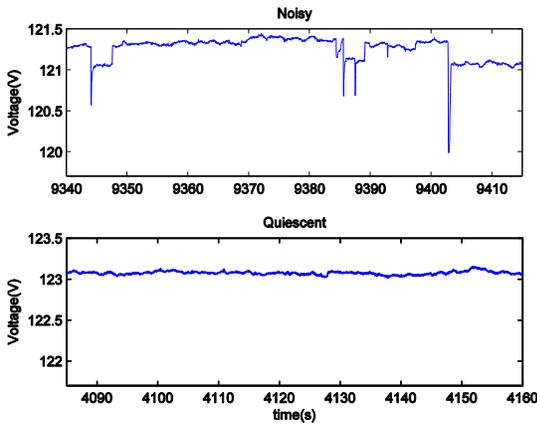


Figure 1 Typical Voltage signals

The voltage data is represented in the C37.118 packet as 32-bit floating point numbers in real and imaginary components. The frequency is also recorded as a 32-bit floating point number. The Arbiter 1133a uses a 16-bit sampler and collects 1,024 samples per measurement giving it a resolution limit of 26 bits on the voltage data. The data limits are set to record data between  $\pm 150$  V with a resulting precision of approximately  $4.47\mu\text{V}$ . Filtering inside the device should produce a slightly lower quantization level in the floating point data. For the frequency data the resolution is specified as 7 decimal digits of precision, in the data this encoded with a quantization level of  $3.81\mu\text{Hz}$  in the floating point number which is comparable to approximately 24 bits of resolution.

### IV. METHODOLOGY

For each component a separate binary file was created from data extracted from the pcap files. These files represented the frequency along with each of the 3-phase real and imaginary voltages. Additionally, the real and imaginary components were transformed to magnitude and phase components and stored as separate files. Four additional files were also created by merging the 3 separate voltage phases into a single data file, in either row-major or column-major format. Finally, all the data was converted to int32 values from the single precision floating point values and saved to separate files. The conversion process captures the same precision as the floating point representation as indicated from the device specification. The purpose of the conversion was to test the idea that integer based numbers are easier to compress than floating point data.

The compression testing was done using command line programs from 7-zip namely 7za.exe for the 7-zip, bzip2 and gzip formats and szip.exe for the szip compression methods. All compression was performed on a Windows 7 computer. To ensure a fair test, all files converted using 7za were converted using the “-mx9” flag which specifies maximum compression. For the szip program, two sets of options were used. The first, “-r4”, specifies that the dominant unit size in the file is 4 bytes. The second, “-r4i”, indicates that the compressor should compress the increment values between successive units instead of the original value.

A typical file consisted of around 5.18 million records, and the raw binary files were typically 20 MB in size. Both the integer based and the floating point files were identical in size. A number of partial files were also included in the test which were created as a result of computer reboots, or network glitches.

### V. RESULTS

Fig. 2 through Fig. 5 represents the results from the five compression methodologies for each of the corresponding data types: frequency, Voltage, voltage magnitude, and voltage phase. The box in the plots represents the 25<sup>th</sup> to 75<sup>th</sup> percentile. The dashed error bars represent 3 times the range of the 25<sup>th</sup> to 50<sup>th</sup> and 50<sup>th</sup> to 75<sup>th</sup> above and below the median line respectively. Any outliers are highlighted as points on the chart. The aggregate results are summarized in Table 1.

Table 1 Summary results for floating point compression

	7z	gz	bz2	sz	sz -i
V	1.57	1.15	1.09	1.90	2.18
Vmag	2.33	1.86	2.33	2.64	2.77
Vphase	1.64	1.15	1.07	2.00	2.39
freq	2.95	2.33	3.01	3.27	3.38

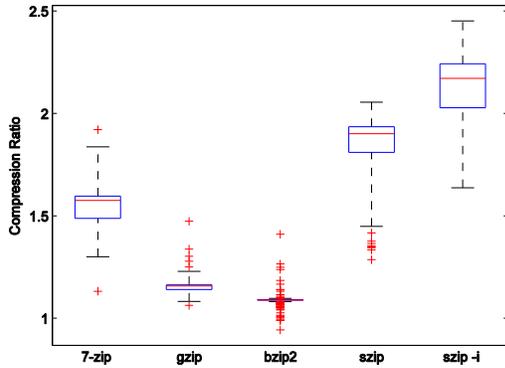


Figure 2 Compression ratios of real and imaginary voltage components

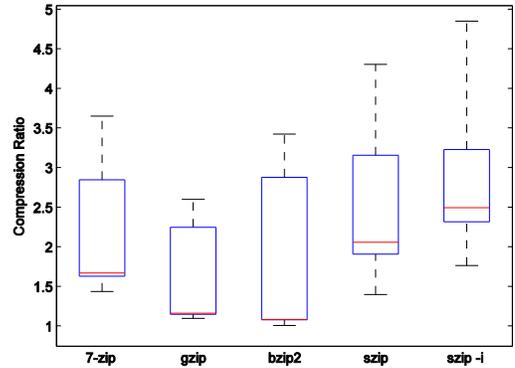


Figure 5 Compression ratios of frequency data

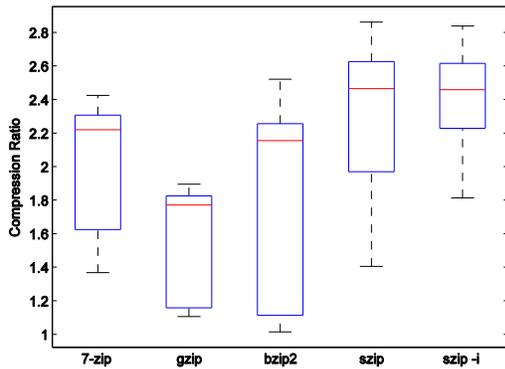


Figure 3 Compression ratios of Voltage magnitude

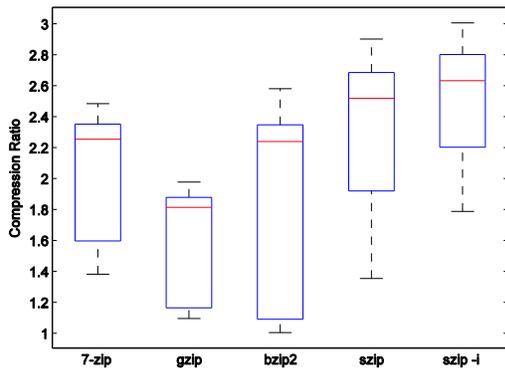


Figure 4 Compression ratios of Voltage Phase

### A. Integer vs. Floating Point Compression

Common conception is that integer data will compress more easily than floating point data. In this case, that proved to be generally untrue. The actual results depended on the data type and the compression algorithm. Table 2 shows the relative comparison of the floating point vs. integer compression.

Table 2 Integer vs. floating point compression results

	7z	gz	bz2	sz	sz -i
V	1.5%	-0.5%	2.1%	2.6%	1.4%
Vmag	-1.8%	-2.7%	-3.7%	-2.2%	-6.5%
Vphase	0.4%	-0.3%	1.2%	1.5%	-0.6%
freq	16.2%	13.0%	15.5%	12.8%	11.4%

These results indicate slight improvement for real and imaginary voltage components, slightly worse results for voltage magnitude compression, and mixed results for the voltage phase information. For frequency, the conversion to integers may involve the truncation of precision that is present in the floating point values that is not specified in the result. In the case of frequency, 7 digits of decimal precision were kept. Translating to integer based representations may have removed some unspecified precision from the results, thereby improving the compression when operating on an integer basis.

### B. Clumped Results

The binary data from the 3 different voltage phases and magnitudes was also clustered into a single file. The file was written in two ways with the individual data sets stacked one after another in the file. As might be expected, the stacked method produced identical compression results as compressing the data sets independently. The other method of interleaving the data sets actually reduced the compression ratios by 10 to 15%. This result indicates no advantage to

merging the 3 phases together for either phase or magnitude data.

## VI. DISCUSSION

The szip algorithm was the best performing algorithm on PMU data. In all the data sets in question, it outperformed the 7-zip algorithm. Using szip with the incremental flag further improved the compression performance with final compression ratios of 2.77 for the voltage magnitude data and 2.39 for the phase data. Compression performance for the frequency data was higher than for the voltage data with a ratio of 3.38.

The general order of performance put gzip as the lowest performing of the algorithms tested, then bzip2, followed by 7-zip with szip performing the best. On the real and imaginary voltage data, gzip slightly outperformed bzip2, though both performed poorly in compression results. These results are generally comparable to the results reported in [6] when examining their unprocessed results. The results achieved in this study indicate slightly higher compression ratios, perhaps due to the use of the maximum compression flags in the file creation. The szip algorithm is one of the compression algorithms that can be used with the HDF5 file format, a common data format for sharing of large scientific data sets. Follow on work might be warranted in examining the applicability of utilizing an HDF5 file format for the storage and transfer of PMU data sets for use in the broader scientific community.

Converting the voltage data to integers, while maintaining the same precision, does not change the compression ratios significantly. For voltage magnitude data this process reduces the compression ratios by around 2%. Converting to integer based data may have some advantage for frequency data. However, a significant fraction of these gains may be coming from a reduction in apparent precision. The conversion to integer was done using the specifications of the measurement system and not the actual precision found in the raw data. It is, therefore, debatable whether this process can be considered a true lossless compression.

There were minor differences in the compressibility of the different voltage phases and between the different PMU units. The main difference occurred in the compression of the voltage magnitude data and was typically within 5% of the aggregate values. These variations are likely due to variations on the loading patterns on certain phases and the noise level on the different units. There also appeared to be a weak temporal correlation in the compressibility of the voltage signals. It is unclear what the implications of this observation are.

A PMU measuring at 60 Hz would produce 257 MB of frequency, voltage and current data in a day. If we were to assume that current data would compress at an equivalent rate as the voltage data. Using compression this data could be reduced to about 98 MB. The timestamp data and status data are highly compressible since very little information is contained in these fields. They are often constant for a long

period of time or are spaced in very regular intervals. If the timestamps are truly regularly spaced, they can be encoded using only a few bytes by simply recording a starting time, an interval and a count. Otherwise, they can be compressed with high ratios. A few tests indicate compression ratios of greater than 20 on time data sets with missing packets or irregularly spaced measurements. Given the relatively infrequent changes in any status field it is expected this would also be highly compressible. Given this, we expect that, accounting for file format overhead, PMU device measurements including 3-phase voltage, 3-phase current, and frequency operating at 60 Hz can be compressed using available lossless compression technologies to about 105MB per day.

## VII. CONCLUSION

In this paper, we discuss the compressibility of a number of fields of data from a synchrophasor using real data collected on a distribution network over a period of several months. The results indicate that storing voltage data as magnitude and phase makes it significantly easier to compress and that frequency data is generally more compressible than the voltage data. The best compressor in terms of compression ratio for lossless storage of PMU data is the szip program using the `-i` flag.

## REFERENCES

- [1] *IEEE Standard for Synchrophasors for Power Systems*, IEEE standard C37.118-2005, Mar. 2006
- [2] North American Synchrophasor Initiative, "Synchrophasor technology roadmap", March 13, 2009. [Online]. Available: [http://www.naspi.org/resources/2009\\_march/phasortechnologyroadmap.pdf](http://www.naspi.org/resources/2009_march/phasortechnologyroadmap.pdf).
- [3] OpenPDC, "about", [Online]. Available: <http://openpdc.codeplex.com/#about>.
- [4] S. Das and P. S. N. Rao, "Principle component analysis based compression scheme for power system steady state operational data", in *Proc 2011 IEEE PES Innovative Smart Grid Technologies -India*, June 2011.
- [5] J. Ning, J. Wang, W. Gao, and C. Liu, "A wavelet based data compression technique for smart grid", *IEEE Transactions on Smart Grid*, Vol. 2, No 1, Mar 2011.
- [6] R. Klump, P. Argawal, J. E. Tate, and H. Khurana, "Lossless Compression of Synchronized Phasor Measurements", *Power and Energy Society General Meeting, 2010 IEEE*, vol., no., pp.1-7, 25-29 July 2010
- [7] J. Ziv and A. Lempel, "A Universal algorithm for sequential data compression", *IEEE Transactions on Information Theory*, Vol. IT-23, no 3, May 1977, pp 337-342.
- [8] M. Burrows, D. J. Wheeler, "A block-sorting lossless data compression algorithm", Digital Equipment Corporation, Palo Alto, CA, Tech. Rep. SRC-RR-124, May 1994.
- [9] I. Pavlov, "7z Format", [Online], <http://www.7-zip.org/7z.html>.
- [10] M. Schindler, "szip homepage", [Online], <http://www.compressconsult.com/szip>.
- [11] P.-S. Yeh, W. Xia-Serafino, L. Miles, B. Kobler, and D. Menasce, "Implementation of CCSDS Lossless Data Compression in HDF", *Earth Science Technology Conference-2002*, 11-13 June 2002, Pasadena, California.