Lossless Compression of Synchrophasor Data in Wide Area Monitoring (WAM) of Power System

Mahesh Yenagimath[1], Shekhappa G. Ankaliki[2]
[1]*Department of Electrical and Electronics Engineering, Hirasugar Institute of Technology, Nidasoshi, Karnataka, India.*
[2]*Department of Electrical and Electronics Engineering, SDM College of Engineering & Technology, Dharwad, Karnataka, India*

***Abstract* - Indian Power System is spreading at a fast pace to meet the growing requirement. This widely spreading grid has also increased the complexity towards monitoring and control. Deployment of Phasor Measurement Units (PMU) in Indian power system has brought a new data stream to be processed and which has also created an opportunity to improve situational awareness on the grid. This high-ratedata collection helps to minimize widespread outages. The placement of Phasor Measurement Units (PMUs) in Wide Area Monitoring Systems (WAMS) results huge amount of data to be analyzed and stored, making efficient storage of such data a priority. Hence it is necessary to use a compression technique that mitigates the increased storage requirements of synchrophasor data. This paper presents a lossless compression technique that utilizes the inherent correlation within PMU data with reasonable accuracy and within limited resource of memory.**

***Index Terms* - Synchrophasor, WAMS, Smart Frid.**

I.INTRODUCTION

The Indian power grid is undergoing a major modernization effort as it transforms into the “Smart Grid”. Key smart grid applications such as wide area situational awareness will need to make effective use of these components in realizing the capabilities of the next-generation power grid. These new applications and capabilities will certainly involve the creation, transmission, storage, and processing of large amounts of data. The major challenge is addressing this data explosion which will require research and development of data management and processing tools. This paper focuses on new approach of improving power system monitoring by compressing massive synchrophasor data in efficient manner.

The existing SCADA/EMS system provides only the steady state view of the power system. But these systems will take more time to send a snapshot of a power system whose characteristics are changing very fast. Conventional SCADA system will acquire voltage, current and frequency data from RTUs, but Wide Area Monitoring system will acquire time synchronized current, voltage measurements with phase angle and frequency measurement by PMUs by using GPS receiver with time resolution of less than 1 microsecond. Hence the better analyzation, condition monitoring and state measurement of dynamic power system is possible. This will help to keep the power system safe, stable, and secured.

Wide area measurement systems (WAMS) for the Indian power grid are recognized as one of the key functionalities of the new Smart Grid. But the August 2003 blackout report [1] strongly recommended the deployment of phasor measurement units (PMUs) in Wide area monitoring systems. PMU will provide GPS-time synchronized data whose data rates are much higher than traditional SCADA measurement systems. Because every PMU will provide accurate measurements from the entire interconnection which will certainly increase situational awareness of grid operators in WAMS.

Indian power grid is one of the largest power grids in the world. Operation of Indian power grid is monitored and coordinated through national load dispatch centre, five regional load dispatch centers, and state load dispatch centers. PMU brought new opportunities for more intelligent and secure control from time-synchronized measurements which make the grid behavior comparable between different locations [16]. The PMUs reports the data to PDC situated at the RLDCs then to NLDC as shown in figure 1.

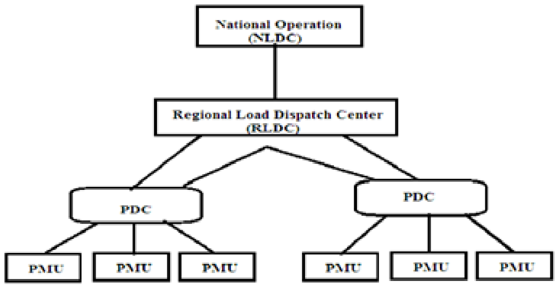

Figure 1: External Architecture of PMU

A PhasorData Concentrator (PDC) is basically used to collect data from multiple PMUs and other PDCs [15]. A PDC also aligns data by time tag to create a time synchronized dataset and transmit this dataset to other information system.

The high volume of synchrophasor data usually 30 to 60 samples/second enabled the application of data mining and machine learning techniques in both operations as well as in planning environment [9]. Huge amount of recorded and stored PMU data can be used for offline applications. Big data is a big concern in grid networks which is shown in table 1.As an example, with only 100 PMUs installed, the Indian power authority needs to manage more than 50GB/ day data at 30 HZ sampling rate [10]. Then as per the survey the actual recorded disturbance data to be preserved for only 10 calendar days in NERC PRC 002-2 device.Large volume of PMU data need to be stored at PDC or transmitted to super PDCs.All PMU data are finally archived by utilities for model validation, testing new wide area protection and control applications, or training/testing disturbance classifiers typically used in system visualization applications.Hence data compression methods are necessary to archive this data.

Around 100 PMUs at 20 Measurements at	
30 Hz sampling rate	50 GB data/day
60 Hz sampling rate	100 GB data/day
120 Hz sampling rate	200 GB data/day

Table 1: Big data in Grid

Synchrophasor data as well as historical data can be used in real time applications such as event detection, stability monitoring etc.[2]. Further data mining techniques are applied to support grid operators in assessing the operating condition of the power grid and it also provides immediate guidance to grid operators for potential mitigation actions in case of system security threats.

II. LOSSLESS DATA COMPRESSION USING PCA

Presently the existing SCADA based data management systems do not support management of huge power system operational data. Hence the power utilities are looking for solutions to store large volume of operational power system data for long periods. But some utilities are using compact discs or tape drives to keep the data for linger time, but which are not reliable. Data compression is the possible solution to limit the stored data size. The usefulness of compression technology for storing power system-data has been highlighted by Werner and Hermansson in [3]. Application of wavelet transform is proposed in [4] for disturbance data compression. In [5] minimum description length criterion (MDL) based compression method is used. Wavelet based compression scheme for PMU data is proposed in [6]. Compression methods are applied on PMU data in such a way that they must give maximum CR and minimum loss of data. But parameters chosen for any compression technique heavily depends on the nature of the data. Most of the data collected by PMU remains constant or slowly varying when power system is in normal condition but at the time of disturbance data will vary rapidly based on type of disturbance. While research has been done to find suitable compression technique to store PMU data effectively [12].

Principal Component Analysis (PCA)

PCA is a very popular dimension reduction technique suitable for large data sets with interrelated variables. PCA is basically a dimension reduction technique which is especially suitable for large data sets with interrelated variables. The original correlated variables in data set transforms into a smaller set of uncorrelated ones, these transformed variables are called as principal components (PC's) [11]. The first principal component accounts for most significant variability in the original data and the next components accounts in the order of reducing variability.

PCA is a dimensionality reduction technique that enables to identify correlations and patterns in a data set so that it can be transformed into a data set of significantly lower dimension without loss of any important information. The main idea behind PCA is to figure out patterns and correlations among various features in the data set. On finding a strong correlation between different variables, a final decision is made about reducing the dimensions of the data in such a way that the significant data is still retained [7].

Step by Step Computation of PCA

Step 1: Standardization of the data: Standardization is all about scaling PMU data in such a way that all the variables and their values lie within a similar range. Comparing the data with large data will have biased impact on output therefore, standardizing the data into a comparable range is very important. Standardization is carried out by subtracting each value in the data from the mean and dividing it by the overall deviation in the data set.

Z= [(Variable value – Mean) / (Standard deviation)]

Step 2: Computing the Covariance Matrix: PCA helps to identify the correlation and dependencies among the features in a data set. Covariance matrix expresses the correlation between the different variables in the data set [13].It is essential to identify heavily dependent variables because they contain biased and redundant information which reduces the overall performance of the model. Mathematically, a covariance matrix is a p × p matrix, where p represents the dimensions of the data set. Each entry in the matrix represents the covariance of the corresponding variables. The covariance value shows how one variable is co-dependent on another variable. If the covariance value is negative, then it indicates that the respective variables are indirectly proportional to each other. A positive covariance denotes that the respective variables are directly proportional to each other. Step by step computation of PCA is as shown in figure 2.

Step 3:Calculating the Eigenvectors and Eigen values: Eigenvectors and Eigen values are calculated from the covariance matrix to determine the principal components of the original data set [13].Eigen values are the scalars of the respective Eigen vector also values shows how big are the variation and Eigen vectors indicates the direction of variance.

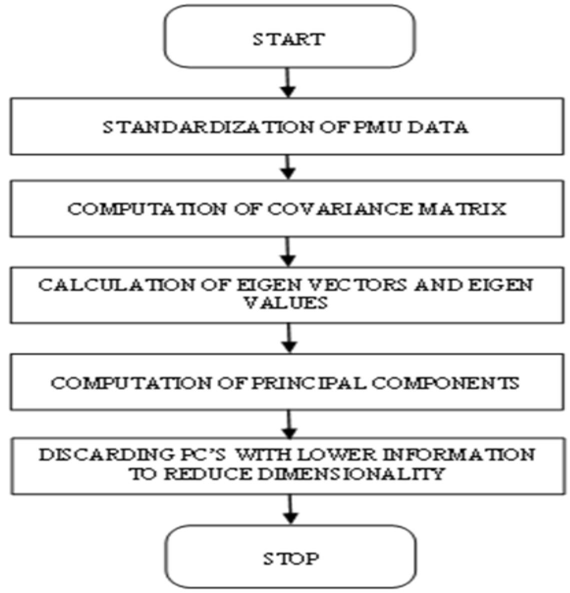

Figure 2: Detailed steps of PCA

Step 4: Computing the Principal Components (PC's): After computation of the Eigenvectors and Eigen values we must order them in the descending order. The eigenvector with the highest eigen value is the most significant and thus forms the first Principal Component. Similarly, the principal components of lesser significances can be removed to reduce the dimensions of the data.The final step in computing the Principal Components is to form a matrix known as the feature matrix that contains all the significant data variables that possess maximum information about the data.

Step 5: Reducing the dimensions of the data set: The last step in performing PCA is to re-arrange the original data set with the final principal components which represents the maximum and the most significant information of the data set. To replace the original data axis with the newly formed Principal Components, we simply multiply the transpose of the original data set by the transpose of the obtained feature vector.

III. CASE STUDY & RESULTS

Data compression algorithm is used to make a data package smaller, in other it is used to decompress the package into its original form. In our work we are using Principal Component Analysis(PCA) algorithm to compress PMU data. Initially for testing purpose we used available data set from Kaggle website. Kaggle allows users to find and publish data sets, and gives platform to work with other data scientists, researchers, and machine learning engineers.

Initially we imported available PMU data set from kaggle website, the same data sheet we uploaded to Google Colab. Google Colab is a product from Google research, which allows programmers to write and execute python code from browser and is especially suitable for data analysis and machine learning applications. Then we applied Principal component analysis (PCA) on the imported dataset using Scikit-learn to find out the dimensions (also referred as components) with maximum variance(where the data is spread out). Basically Scikit-learn is a freely available,simple, and efficient software machine learning library especially suitable for the Python programming language for predictive data analysis.

Features with smaller variance in the data are then projected into new lower dimension. Next step is to compute Eigen vectors and ordering them by their Eigenvalues in descending order, which allows us to find the principal components in order of significance. The covariance value denotes how co-dependent two variables are with respect to each other. If the covariance value is negative, it denotes the respective variables are indirectly proportional to each other. A positive covariance denotes that the respective variables are directly proportional to each other.

Continued from the previous section for principal component analysis, in this section we will standardize the data, construct the covariance matrix, obtain the Eigen values and eigenvectors of the covariance matrix, and sort the Eigen values by decreasing order to rank the eigenvectors.

```
pca.get_covariance()
```

```
array([[ 1.03909111e+00, -3.06583180e-01, -2.19610043e-01,
        -3.35815762e-01,  2.08823796e-02,  9.25329558e-03,
        -1.56711800e-02,  8.04107561e-01,  8.03501968e-01,
         8.01777119e-01,  2.39245211e-02, -4.24132237e-05,
        -1.30499322e-02,  1.24673304e-03],
       [-3.06583180e-01,  1.18684912e+00,  8.98461180e-01,
         9.08607598e-01, -1.62409271e-02, -8.61860405e-03,
         1.61861843e-02, -5.45809014e-01, -5.44773778e-01,
        -5.45950072e-01, -1.43157896e-02,  2.88310486e-03,
         1.51296255e-02, -1.52382017e-03],
       [-2.19610043e-01,  8.98461180e-01,  1.17558882e+00,
         8.93921846e-01, -1.53822889e-02, -7.23023422e-03,
         1.55163186e-02, -4.60763470e-01, -4.59773473e-01,
        -4.61204015e-01, -1.30184927e-02,  3.71544475e-03,
         1.42870380e-02, -1.40401199e-03],
       [-3.35815762e-01,  9.08607598e-01,  8.93921846e-01,
         1.18653734e+00, -1.79112506e-02, -8.52549763e-03,
         1.72635672e-02, -5.73097180e-01, -5.72051604e-01,
        -5.73131808e-01, -1.61580177e-02,  3.41630829e-03,
         1.57418524e-02, -1.54858533e-03],
       [ 2.08823796e-02, -1.62409271e-02, -1.53822889e-02,
        -1.79112506e-02,  9.29946561e-01, -3.33032351e-01,
        -3.31266498e-01,  1.33503470e-02,  1.34538245e-02,
         1.33960040e-02,  6.23919441e-01, -4.55706470e-01,
        -8.02350137e-02, -4.78058875e-03],
       [ 9.25329558e-03, -8.61860405e-03, -7.23023422e-03,
        -8.52549763e-03, -3.33032351e-01,  9.64514563e-01,
        -3.14260780e-01,  1.72223941e-02,  1.72757805e-02,
         1.76684956e-02, -1.37832889e-01,  6.71337485e-01,
        -5.10468076e-01,  4.86927197e-03],
       [-1.56711800e-02,  1.61861843e-02,  1.55163186e-02,
         1.72635672e-02, -3.31266498e-01, -3.14260780e-01,
         8.97596398e-01, -1.42605611e-02, -1.44257218e-02,
        -1.47610339e-02, -4.85617010e-01, -1.78796343e-01,
         5.56314805e-01,  1.82787164e-04],
```

Figure 2: Computing Covariance Matrix

In this step, what we need to do is, to take proper decision whether to keep all these components or discard those of lesser significance or of low Eigen values, after that we form matrix of vectors with the remaining eigen values that we call as Feature vector. Feature vector is simply a matrix that contains columns with the eigenvectors of the components that we decide to keep for further analysis. Formation of feature vector matrix is the first step towards dimensionality reduction, because if we decide to keep only M eigenvectors or components out of n, the final data set will have only M dimensions. We can form feature vector matrix with all Eigenvectors, or we can discard the Eigen vectors with lesser significance. Discarding Eigen vectors of lesser significance will reduce dimensionality which will also cause loss of information in the final data set. If Eigen vector having lesser information carrying only 4% of the information, this loss is not so important because still we have 96% of the information is carried by other components

```
pca=PCA(n_components=4)
X_new=pca.fit_transform(X)
X_new
```

```
array([[-0.41945784, -0.28206274,  2.29302112, -0.23080954],
       [-0.44723214, -0.28308421,  2.29387832, -0.22843133],
       [-0.45312574, -0.28407317,  2.2933874 , -0.23526108],
       ...,
       [ 2.30195677,  1.20497811,  0.81343848, -1.21146008],
       [ 2.30523967,  1.20440416,  0.81327727, -1.2039413 ],
       [ 2.28858526,  1.20434311,  0.81449487, -1.1828891 ]])
```

Figure 3:Feature Vector

In this step, the aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components hence the name is referred as Principal Components Analysis. This process can be completed by multiplying the transpose of the original data set by the transpose of the feature vector.

After the pre-processing [8], we have applied that data to the PCA. Principal Component Analysis computes new set variables called "Principal Components" and this will express the data in terms of these new variables. As we have 14 data labels or variables in uploaded PMU data sheet such as R, Y, B phase voltage and current magnitudes and respective angles and time, so totally there are 14 variables in uploaded

PMU data hence PCA will generate 14 Principal components which is as shown in Figure 4.

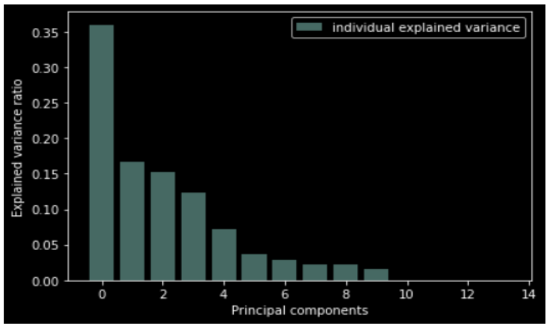

Figure 4: Principal components

The main aim of PCA is to compress the PMU data without loss of any data. As shown in the figure, first 4 Principal components are carrying maximum information, hence for the compression purpose we are considering only first 4 Principal components. But we have liberty to choose number of PC's for the compression based on information.

The explained variance is the percentage of variance that is usually attributed by each of the selected components. Ideally, we would choose the number of components to include in our model by adding the explained variance ratio of each component. Figure 5 shows the Explained variance ratio of 4 Principal Components where first 4 principal components will explain most variance. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 14-dimensional data gives you 14 principal components, but PCA tries to put maximum possible information in the first component then maximum remaining information in the second component and so on.

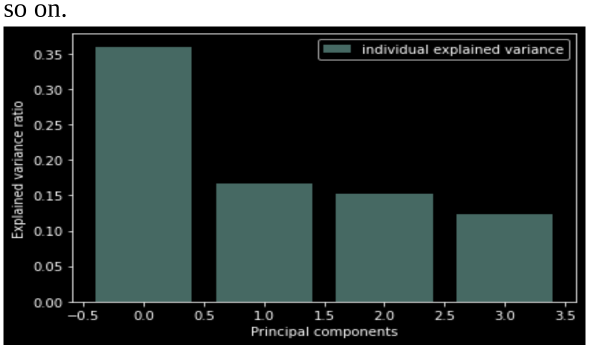

Figure 5: Implementation using first four Principal Components

In this way organizing information in principal components will allow you to reduce dimensionality without losing much information, and this is done by discarding the components with low information and considering the remaining components as new variables. An important thing to realize here is that the principal components are less interpretable and don't have any real meaning since they are constructed as linear combinations of the initial variables. Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance the lines that capture most information of the data.

The relationship between variance and information here is that the larger the variance carried by a line, the larger the dispersion of the data points along it and the larger the dispersion along a line, the more the information it has.

The Figure 3 shows that, the first principal component alone accounts for 40 percent of the variance. Also, we can see that the first two principal components combined explain almost 60 percent of the variance in the data. After implementing PCA on synchrophasor data we are getting compression ratio of around 46% which is better as mentioned in reference paper [12].

```
[ ] import os
    data_size_orig = os.path.getsize("_LBNL_a6_bus1_2015-10-01.csv")
    data_size_comp = os.path.getsize("copressed_data.csv")
    print("Original data size = " + str(data_size_orig) + " Bytes")
    print("Compressed data size = " + str(data_size_comp) + " Bytes")

    Original data size = 1950580927 Bytes
    Compressed data size = 888717062 Bytes
```

Figure 6. Data set sizes before and after compression

```
[ ] Compr_ratio = (data_size_comp/data_size_orig)*100
    print("Total Compression Ratio = " + str(round(Compr_ratio)) + "%")

    Total Compression Ratio = 46%
```

Figure 7. Percentage of compression ratio

Original Data size	Compressed Data size
1950580927 Bytes	888717062 Bytes

Table 2. Data Compression Result

IV. CONCLUSION

Big data produced by modern power system in day-to-day operation has become a valuable resource for the development of advanced applications for both real time operations, as well as for offline planning environment [2]. Here we implemented an efficient data compression algorithmi.e., Principal Component Analysis (PCA) on synchrophasor data to reduce physical size. The proposed data compression algorithm is very general technique that will use massive Synchrophasor data to reduce the dimensionality. The same algorithm can be applied to a Phasor data concentrator (PDC) fed from any number of PMUs and does not need any modifications.

REFERENCES

[1] U.S.-Canada Power System Outage Task Force. (2004) Final Report on the August 14, 2003, Blackout in the United States and Canada: Causes and Recommendations. [Online]. Available: https://reports.energy.gov.

[2] Mahesh Yanagimath, et al., “Review on Synchrophasor based Data Mining Techniques and Tools”, in i-manager’s Journal on Power Systems Engineering, Vol. 7, No. 4, November 2019 - January 2020.

[3] Werner.M. and Hermansson, U, "Integrated Utility Data Warehousing - A Prerequisite to Keep Up With Competition On Electricity Markets," Power System Management and Control 2002 Fifth International Conference on, 17-19 April 2002, pp 130-135.

[4] Littler,T.B., Morrow,D.J., “Wavelets for the analysis and compression of power system disturbances,” IEEE Transaction on Power Delivery., vol. 14, no. 2, Apr. 1999, pp. 358-364.

[5] M. V. Ribeiro, S. H. Park, J. M. T. Romano, and S. K. Mitra, “A novel MDL-based compression method for power quality applications,” IEEE Trans. Power Del., vol. 22, no. 1, pp. 27–36, Jan. 2007.

[6] Jiaxin Ning, Jianhui Wang, Wenzhong Gao, and Cong Liu, “A WaveletBased Data Compression Technique for Smart Grid” IEEE Transactions on Smart Grid, Vol. 2, No. 1, March 2011.

[7] Phani Harsha Gaddi, et al., “Efficient Compression of PMU Data in WAMS”, IEEE TRANSACTIONS ON SMART GRID, VOL. 7, NO. 5, SEPTEMBER 2016.

[8] Yang Zhang, Tao Huang and Ettore Francesco Bompard, “Big data analytics in smart grids: a review”, Springer Open access Zhang et al. Energy Informatics (2018) 1: 8,https: // doi. Org / 10.1186 / s42162-018-0007-5.

[9] Philip Top and John Breneman, “Compressing Phasor Measurement Data”, DOI:10.1109/NAPS .2013.6666959, https://ieeexplore.ieee.org/ document/6666959

[10] L. Xie, Y. Chen, and P. R. Kumar, “Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis,” IEEE Trans. Power Syst., vol.29, no. 6, pp. 2784–2794,Nov. 2014.

[11] Y. Geet al., “Power system real-time event detection and associated data archival reduction based on synchrophasors,” IEEE Trans. Smart Grid, vol. 6, no. 4, pp. 2088–2097, Jul. 2015.

[12] R. Klump, P. Agarwal, J. E. Tate, and H. Khurana, “Lossless compression of synchronized phasor measurements,” in Proc. IEEE Power Energy Soc. Gen. Meeting, Minneapolis, MN, USA, Jul. 2010, pp. 1–7.

[13] Yang Chen, Student Member, IEEE, Le Xie, Member, IEEE, and P. R. Kumar, Fellow, IEEE, “Dimensionality Reduction and Early Event Detection Using Online Synchrophasor Data” https://ieeexplore.ieee.org/document/6808416.

[14] Daniel Dotta, Member, IEEE, Joe H. Chow, Fellow, IEEE, Luigi Vanfretti, Member, IEEE Muhammad S. Almas, Student Member, IEEE, and Marcelo N. Agostini, Member, IEEE, “A MATLAB-based PMU Simulator”, https://ieeexplore.ieee.org/document/6672629.

[15] Kedar V. Khandeparkar, et.al, “Design of a Phasor Data Concentrator for Wide Area Measurement System”, https://www.research gate.net/publication/317664931.

[16] M. N. Aravind et.al, “Application of Compressed Sampling to Overcome Big Data Issues in Synchrophasors” https://ieeexplore.ieee.org/ document/7584142,DOI: 10.1109/ICPES.2016.7584 142.