Estimation Matrix Calibration of PMU Data-driven State Estimation Using Neural Network

Guanyu Tian*, Yingzhong Gu*, Xiao Lu†, Di Shi*, Qun Zhou†, Zhiwei Wang*, Jie Li†

*GEIRI North America, San Jose, CA 95134, USA
†Dispatch Center, State Grid Jiangsu Electric Power Company Ltd., Nanjing, Jiangsu, China
§University of Central Florida, Orlando, FL 32816, USA

Email: tiang@knights.ucf.edu, yingzhong.gu@geirina.net, di.shi@geirina.net

Abstract—Linear state estimation (LSE) is a phasor measurement unit (PMU) data-based power system state estimation that incorporates a linear measurement model in rectangular coordinates. Due to the high computational efficiency and high observational time-resolution, LSE can act as a supplementary state estimation in a wide-area monitoring system (WAMS). The performance of LSE is relatively sensitive to noise in measurements. Therefore, the estimation accuracy relies heavily on the accuracy of the estimation matrix, which is directly influenced by the measurement weight matrix. This paper proposes two novel calibration method of the estimation matrix using neural networks. One is based on the minimum absolute network loss (ANL), and the other is based on the minimum average squared network loss (ASNL). Both methods are tested and compared with LSE algorithms on the IEEE 14-bus system.

Index Terms—State estimation, parameter calibration, neural network, PMU.

I. INTRODUCTION

State estimation is considered one of the fundamental functions supporting modern power grids. Many system stability analysis and electricity market transactions are carried out based on accurate system states [1]. Correspondingly, the requirements for system monitoring in terms of accuracy and time efficiency become progressively demanding [2]. Unfortunately, traditional SCADA-based state estimation is not capable of capturing the changing states due to the relatively low time-resolution of SCADA data. Hence, LSE by using PMU data, can be adopted as supplementary monitoring. The measurement model of LSE is a set of linear functions, which enables the estimated states to be solved in a non-iterative manner and makes LSE much faster than the recursive methods developed upon the quadratic measurement model compatible with the SCADA data [3], [4].

However, LSE methods are relatively sensitive to measurement noise. In order to attenuate the impact of noisy measurements collected from PMU with comparably poor data quality, the weights of measurements need to be carefully selected. A common practice is assigning weights upon the reciprocal of residuals’ variance. Theoretically, the expected weighted average measurement residual under normal-distributed noise can be minimized [3].

In practice, the measurement noise is affected by many factors, including communication interference, inaccuracy of device calibration, etc. [5]. Therefore, distribution of measurement noise is varying across the time. In order to achieve better estimation results, it is necessary to tune the weight matrix across the time as well. Reference [6] introduces an online tuning method for SCADA data, and a modified version is applied to the PMU data in [7]. The initial weights can be normalized across all measurements or assigned based on experience. Then the weight matrix is updated by the standard deviation of new measurements. The estimation and the update process will repeat until the weights converge.

In this paper, we propose a novel neural network-based calibration method that differentiates us from the conventional statistics-based methods. The neural network-based linear state estimation (NNLSE) is applied to the noisy measurements within a time window. Features of noises are learned and filtered by the neural network through a stochastic gradient descent (SGD) based training process. The converged network is an approximation to the estimation matrix in LSE that contains an implicit weight matrix inside. The estimation matrix can be derived from the estimated states and the corresponding measurements within the time window. Two algorithms are derived and introduced in this paper. The first one is derived from the single-sample-point with the minimum absolute network loss (ANL). The other one is to minimize the average squared network loss (ASNL) of all sample points within the time window. Based on their properties, they are suitable for different convergence scenarios. The minimum ANL-based calibration algorithm only considers one sample point per calculation, and therefore it is less affected by the convergence performance. Nevertheless, the minimum ASNL-based calibration algorithm takes all points within the time window into consideration, and therefore it is more accurate when the standard deviation of the loss is lower.

Major contributions of this paper lie in that two novel estimation matrix calibration methods using neural networks are proposed to improve the performance of LSE by enhancing its robustness against measurement noise.

This paper is organized as follows: The proposed method and the related techniques are introduced in section II, including the LSE formulation, the NNLSE architecture, and
the two estimation matrix calibration algorithms. Section III presents the case study of the proposed methods. Conclusions and future work are discussed in section IV.

II. METHODOLOGY

A. Linear States Estimation

The PMUs are usually installed at the terminals of lines, and their measurements include the 3-phase current and voltage phasors in polar coordinates. Generally speaking, the 3-phase data stream can be fed into a 3-phase state estimation directly especially for small systems such as smart distribution systems [8]. The transmission system can be simplified as three-phase-balanced systems. Hence, 3-phase data can be transformed into sequence data through the Clarke transformation, and the state estimation can be implemented upon the positive sequence measurements [9].

Equations (1) and (2) represent the state vector and measurement vector. The state vector consists of the voltage phasors and current phasors measured.

\[ \vec{x} = [V_1 \angle \theta_1 \quad V_2 \angle \theta_2 \cdots V_n \angle \theta_n]^T \]  
\[ \vec{z} = [V_1 \angle \theta_1 \quad V_2 \angle \theta_2 \cdots V_n \angle \theta_n \quad I_1 \angle \theta_1 \quad I_2 \angle \theta_2 \cdots I_m \angle \theta_m]^T \]  

(1)  
(2)

The measurement model of PMU data can be derived from Ohm’s law as formulated in Equation (3). The full measurement model is inserted in between the network output and current injection admittance matrix that used to calculate the current injection at the “from” end of lines. Variables \( \vec{V} \) and \( \vec{I} \) are the voltage and current phasors of the entire system. The measurement model is on the right of the arrow, where the \( \vec{V}' \) and \( \vec{I}' \) are the measured voltage and current phasors. Variables \( I' \) and \( V' \) are the corresponding rows from \( I \) and \( V \) respectively. By combining the voltage and current measurements into one formulation, the measurement model of PMU data can be represented by the \( H \) matrix in Equation (4).

\[ \begin{bmatrix} \vec{V}' \\ \vec{I}' \end{bmatrix} = \begin{bmatrix} \vec{V} \\ \vec{I} \end{bmatrix} \quad \begin{bmatrix} \vec{V}' \\ \vec{I}' \end{bmatrix} = \begin{bmatrix} \vec{V} \\ \vec{I} \end{bmatrix} = H \vec{x} \]  

(3)

(4)

Although the model in Equation (4) is linear, its components are complex numbers. It can be further expanded into the rectangular-coordinate formulation in Equation (5). The corresponding measurement model becomes Equation (6), where \( H_{real} \) and \( H_{imag} \) are the real and imaginary parts of the \( H \) matrix [4]. Matrix \( H' \) represents the final linear measurement model for linear state estimation in rectangular form.

\[ x = \begin{bmatrix} \text{real}(\vec{x}) \\ \text{imag}(\vec{x}) \end{bmatrix}, \quad z = \begin{bmatrix} \text{real}(\vec{z}) \\ \text{imag}(\vec{z}) \end{bmatrix} \]  

\[ z = \begin{bmatrix} H_{real} & -H_{imag} \\ H_{imag} & H_{real} \end{bmatrix} x = H' x \]  

(5)  
(6)

Based on the formulation in Equation (6), it is possible to solve for the states directly. The solution of \( x \) is given in Equation (7), where the weighted pseudo-inverse of \( H' \) is calculated using the Moore-Penrose method [10]. Matrix \( W \in \mathbb{R}^{(n+m) \times (n+m)} \) is a diagonal matrix, of which the diagonal components are weights for the corresponding measurements. Matrix \( E \) is the linear estimation matrix, whose physical meaning is the weighted pseudo-inverse of \( H \). The proposed calibration methods aim to calibrate the \( E \) matrix.

\[ \hat{x} = (H'^T W^{-1} H')^{-1} H'^T W^{-1} z = E z \]  

(7)

B. Neural Network-based LSE

Figure 1 shows the flowchart of the NNLSE. The architecture selected for the linear state estimation neural network (LSE-net) is a 3-layer feed-forward neural network, because it has a powerful universal function approximation capability and, as a shallow network, it has fast computation speed to meet the requirement of online estimation [11], [12]. It is expected to approximate the LSE equations through online SGD training. The input of the LSE-net is a measurement vector \( z \) and output is an estimated state vector \( \hat{x} \). The estimated measurements \( \hat{z} \) are calculated base on the estimated states and then used for loss calculation, where the loss function is the L2 norm of the measurement residuals. Unlike the standard neural network training, in which the output is directly compared with target values to calculate the loss. The measurement model is inserted in between the network output and the loss function input. The reason to do so is that the true states are unknowable so that only the measurements and the estimated measurements are comparable. Minimizing the measurement residual can minimize the difference between the estimated states and their true values. Due to the insertion of the measurement model, the defined loss is not equivalent to the loss of the neural network itself. The gradient of loss to the network output \( \hat{x} \) needs to be derived through the measurement model first and then applied to the backpropagation inside the network to update its parameters [13]. This process is formulated in Equation (8), where \( H'^T \) is the pseudo inverse of the measurement model \( H' \).

\[ \frac{\partial \text{Loss}}{\partial \hat{x}} = \frac{\partial ||z - \hat{z}||^2}{\partial \hat{x}} \]  
\[ \frac{\partial \text{Loss}}{\partial \hat{x}} = \frac{\partial \text{Loss}}{\partial \hat{z}} \frac{\partial \hat{z}}{\partial \hat{x}} \]  
\[ \frac{\partial \hat{x}}{\partial \hat{z}} = H'^T \]  

(8)

The training process of the LSE-net is an SGD because the estimation and update are performed on each data point. The
batch size is one, and the average gradient of the subset is the gradient itself. Moreover, the batch size is adjustable depending on the accuracy and speed requirement. When the batch size is greater than 1, the update still follows SGD, but the gradient becomes the average gradient of all samples within the batch.

C. Calibration Algorithms

The real-time application of the proposed method is to provide LSE with an estimation matrix that is robust against noise at each interval. The flowchart of the estimation matrix calibration process is presented in Figure 2. Vector \( z^t \) denotes the PMU measurement vector at step \( t \). Matrices \( E_{cali} \) and \( E'_{cali} \) are the estimation matrices generated upon two different rules. The NNLSE is performed on the noisy measurements, from which the converged part of the estimated states and the corresponding measurements are the sample points. The estimation matrix can be calibrated upon two criteria, the minimum ANL, and the minimum ASNL. Details of each method are introduced below.

![Fig. 2. Estimation matrix calibration flowchart](image)

1) **Minimum Absolute Network Loss-based Calibration:**

The calibrated estimation matrix \( E_{cali} \) can be derived by Equation (9) from Equation (7). Variable \( t^* \) is the time stamp with the minimum ANL, and \( x^{t^*} \) contains the estimated states by NNLSE. Variable \((z^{t^*})^\dagger \) is the pseudo inverse of the measurement vector. If the solved pseudo inverse satisfies the two requirements in Equation (10), then \((z^{t^*})^\dagger \) is a unique pseudo inverse, so that the calibrated estimation matrix \( E_{cali} \) is also unique [14].

\[
E_{cali} = \hat{x}^{t^*} (z^{t^*})^\dagger \tag{9}
\]

\[
\begin{align*}
(z^{t^*} (z^{t^*})^\dagger)^T &= z^{t^*} (z^{t^*})^\dagger \\
((z^{t^*})^\dagger z^{t^*})^T &= (z^{t^*})^\dagger z^{t^*} \tag{10}
\end{align*}
\]

2) **Minimum Average Squared Network Loss-based Calibration:**

The solution of estimation matrix \( E'_{cali} \) that yields the minimum ASNL for all samples is given in Equation (11), where \( \hat{\chi} \) and \( \zeta \) are the NNLSE estimated state vector set and the corresponding measurement vector set, respectively. Sample points are stacked horizontally as formulated in Equation (12). The uniqueness criteria are the two requirements in Equation (10) as well.

\[
E'_{cali} = \hat{\chi} (\zeta)^\dagger \tag{11}
\]

\[
\hat{\chi} = [\hat{x}^1, \hat{x}^2, \ldots, \hat{x}^T], \zeta = [z^1, z^2, \ldots, z^T] \tag{12}
\]

### III. CASE STUDY

The proposed methods are tested and compared with traditional LSE methods in the IEEE 14-bus system for performance validation. This section includes four parts. Firstly, simulation settings are listed in III-A. Secondly, the convergence of the NNLSE is presented in III-B. Then the performance of the calibrated estimation matrices is tested in two case studies. The first case is discussed in III-C, which applies the same distributed noise to all measurements. Here we call this test the homogeneous noise case. The other test case discussed in III-D applies noise with different distributions to measurements, which is named the heterogeneous noise case.

A. Settings

1. PMUs are placed at Bus 2, 6, 7 and 9 according to the optimal PMU placement in [15]. The current phasors of the connected lines are also measured.
2. The PMU reporting rate is set to 50 Hz.
3. The neural network estimator is a 3-layer feed-forward neural network with 108 neurons in the hidden layer. The dimension of the input and output layers are 36 and 28 respectively, which is determined by the dimension of the measurement vector and state vector. The learning rate is 0.01.
4. The measurement noise complies with the zero-mean Gaussian distribution.
5. The simulation duration is 1 second.
6. The noise of training data and the testing data are generated independently, but with the same distribution and standard deviation.
7. The algorithms are implemented in MATLAB 2018a under the Microsoft Windows 10 environment. The simulation is conducted on a computer with Intel(R) Core(TM) i5-6400 CPU@3.00GHz Processor and 8 GB of RAM.
8. The evaluation metric of estimation accuracy is the root mean square error (RMSE) [16].

B. NNLSE Convergence and Time Efficiency

Figure 3 shows the convergence curves of the losses of neural networks. Under the noise-free measurements, the results are ideally converged to near 0. However, with measurement noise of \( N(0,0.01^2) \), the loss converges to the magnitude of \( 10^{-2} \). The convergence takes approximately 0.4 seconds (20 iterations) in both cases. Convergence time does not affect the online implementation of NNLSE because the LSE-net used for online NNLSE is pre-trained, and the pre-training process is carried out off-line. Based on the convergence result, the collected samples for the estimation matrix calibration are from the 21st step to the 50th step.

According to [17], typical PMU reporting rate currently used in the industry is between 30-60 Hz, which requires the time consumption of the estimation algorithm to be lower than 0.0167 seconds to achieve real-time implementation. The
estimation time consumption of NNLSE is recorded for every step and the average time consumption per step is $5.338 \times 10^{-5}$ seconds, satisfies the real-time implementation requirement of time efficiency.

C. Homogeneous Noise Case

In this test case, the noise applied to all measurements has the same distribution. Hence, the weights for all measurements are the same. The diagonal components of the weight matrix are normalized to 1. The noise distribution is $N(0, \sigma^2)$, where $\sigma$ is the standard deviation of the Gaussian distribution. The $E$ and $E'$ are the estimation matrices calibrated using the minimum ANL-based algorithm and the minimum ASNL-based algorithm, respectively. The LSE with homogeneous weight matrix and the calibrated estimation matrices are applied to the test data set. Their estimation error-indexes under different noise standard deviations are summarized in Table I.

<table>
<thead>
<tr>
<th>$\sigma^2$</th>
<th>$1 \times 10^{-6}$</th>
<th>$1 \times 10^{-5}$</th>
<th>$1 \times 10^{-4}$</th>
<th>$1 \times 10^{-3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSE</td>
<td>0.0006</td>
<td>0.0017</td>
<td>0.0051</td>
<td>0.0174</td>
</tr>
<tr>
<td>LSE-E</td>
<td>0.0005</td>
<td>0.0015</td>
<td>0.0030</td>
<td>0.0096</td>
</tr>
<tr>
<td>LSE-E'</td>
<td>0.0004</td>
<td>0.0011</td>
<td>0.0037</td>
<td>0.0130</td>
</tr>
</tbody>
</table>

D. Heterogeneous Noise Case

The noise applied to the voltage and current phasors are different in this test case. The distribution of noise on the current phasors complies with $N(0, \sigma_1^2)$ and that for the voltage phasors complies with $N(0, \sigma_2^2)$, where $3\sigma_1 = \sigma_2$. Based on this setting, the relative relationship between the current phasor measurement weight $w_I$, and the voltage phasor measurement weight $w_V$ should be $w_I = 9w_V$. The resulting weight matrix is labeled as $W$. Here we set the diagonal components of $W$ to be either 1 or 9 depending upon its corresponding measurements. The estimation results of the four algorithms under different noise deviations are summarized in Table II, where the LSE method is using the homogeneous weights, and the LSE-W uses the $W$ as weight matrix, the LSE-E and LSE-E’ are the estimation matrix calibration-based methods.

Figure 6 visualizes the estimation error against the deviation of noise. The relative accuracy is similar to the homogeneous test case in that both estimation calibration-based methods are more accurate than the weight matrix-based methods under different noise levels. The minimum network loss-based
being affected by the network convergence performance. The

The heterogeneous weighted LSE yields the same estimation error with the homogeneous weighted LSE when the noise magnitude is low. Nevertheless, with the increase of noise level, the heterogeneous weighted LSE demonstrates better performance than the homogeneous weighted LSE. However, the advantage is marginal, especially when compared with the estimation matrix calibration-based methods.

The comparison of fitted distribution of the four methods is shown in Figure 7, of which the variance of noises is $\sigma_1^2 = 1 \times 10^{-4}$ and $\sigma_2^2 = 9 \times 10^{-4}$.

IV. CONCLUSION

This paper proposes two neural network-based estimation matrix calibration algorithms for LSE. Both methods reduce the estimation error of the traditional weight matrix-based LSE under all the tested cases. Their difference is the extent of being affected by the network convergence performance. The minimum network loss-based algorithm is more accurate under the scenarios with more substantial noise and network loss, while the minimum square error-based algorithm yields lower error at scenarios with less noise and network loss. This work can be further extended. The weight matrix is hidden in the estimation matrix. If it could be derived from the calibrated estimation matrix, the calibrated weights are the measurement accuracy indices. Although the proposed neural network-based state estimation and weight matrix calibration method shows good performances, there still exist limitations. For instance, the proposed methods are vulnerable to malicious data and system topology errors. Therefore, the proposed method needs to be aided by malicious data removal and system parameter calibration algorithms as data pre-processing in real-world applications.

REFERENCES