FAST NONCONVEX SDP SOLVER FOR LARGE-SCALE POWER SYSTEM STATE ESTIMATION

Yu Lan[†⋆] *Hao Zhu*[⋆] *Xiaohong Guan*[†]

[†] Institute of Systems Engineering, Xi'an Jiaotong University
[⋆] Department of ECE, The University of Texas at Austin

ABSTRACT

Fast power system state estimation (SE) solution is indispensable to achieve real-time decision making in power grid management. Semidefinite programming (SDP) reformulation has shown powerful to approach the global optimum of the nonlinear SE problem, while suffering from high computational complexity. Thus, we leverage the recent advances in nonconvex SDP reformulation that can allow first-order updates to potentially solve the original SDP problem. We further adopt the accelerated gradient descent (AGD) method for the resultant unconstrained problem for improved convergence speed. Numerical tests have demonstrated that AGD can achieve comparable SE performance as the globally optimal SDP solution at improved computational efficiency.

Index Terms— Power system state estimation, semidefinite programming, nonconvex reformulation, accelerated gradient descent.

1. INTRODUCTION

Power system state estimation (SE) aims to obtain the operating condition of the grid, namely nodal complex voltages, from noisy measurements taken at buses and branches. The SE problem is of paramount importance for reliable control and economic operation of power systems; see e.g., [1, 2].

Due to a nonlinear measurement model, SE is traditionally formulated as a nonlinear least-squares (LS) problem and solved by Gauss-Newton (GN) iterations [3, Ch. 2]. The GN method iteratively updates the variables by minimizing an approximate objective through linearization. Albeit computationally efficient per iteration, convergence of GN to global optimum is generally not guaranteed. To tackle the nonlinearity, recent work [4] has proposed a semidefinite programming (SDP) reformulation of the SE problem using rank relaxation. To promote low-rank solutions, [5] suggested a nuclear norm based penalization. General penalization terms are designed in [6, 7] for guaranteed exact recovery and quantifiable estimation error of the SDP-SE formulation. Although the global optimum to SDP-SE can be obtained by generic algorithms such as the interior-point method based SeDuMi solver [8], the high-order polynomial complexity therein could be a computational burden for large-scale power systems [9]. Therefore, recent work has focused on using conic relaxation or composite optimization techniques for the SE problem [6, 10]. A parallelizable SDP solution was developed in [11] using graph-specific decomposition and alternating projection schemes. However, it remains open to develop fast solvers specifically for the SDP-SE formulation.

Recently, a nonconvex approach to solving SDPs by representing the solution as matrix factorization has become popular [12, 13]. The idea is to eliminate the positive semidefinite (PSD) constraint. The latter can be enforced through eigen-decomposition, yet at high computational complexity. Instead, the nonconvex reformulation leads to computational gains in practice based on gradient descent updates. The so-termed factored gradient descent (FGD) algorithm was developed in [14] for the nonconvex SDP formulation. If the original SDP objective function is convex and smooth, FGD can converge sublinearly to the optimal SDP solution of low rank, assuming the initialization is sufficiently accurate. However, the choice of good initialization as well as the convergence rate for any initialization are unclear for the FGD updates in general SDP problems.

The goal of our work is to develop a fast solution technique for the nonconvex SDP-SE formulation. Towards this end, we first reformulate the SDP-SE problem using the matrix factorization idea. The FGD method is presented to obtain low-rank solution to the nonconvex reformulation. To tackle the slow convergence speed of FGD, we propose to adopt the popular accelerated gradient descent (AGD) method by Nesterov [15] as a heuristic alternative to FGD. AGD has achieved fast convergence rate for convex problems with applications to matrix trace norm minimization [16] and binary classification [17]. Recently, AGD has been shown to potentially provide accelerated performance for even nonconvex problems in neural networks [18]. Furthermore, AGD has been advocated to better avoid saddle points in certain nonconvex problems [19]. To the best of our knowledge, AGD has not yet been used for the nonconvex SDP reformulation. Therefore, future work remains to investigate the

This work was supported in part by the National Science Foundation under Award ECCS-1802319 and the Key R&D Project of China 2016YFB0901904.

performance guarantees of AGD, especially for our nonconvex SDP-SE problem. We use the low-rank solutions from both FGD and AGD to recover a feasible SE solution, further improved by GN iterations to reduce the relaxation gap. Numerical results have demonstrated that AGD can significantly reduce the computational time while achieving better solutions than FGD. Compared to the globally optimal SDP solution, the AGD method produces competitive SE performance more efficiently.

Notation: Upper (lower) boldface symbols stand for matrices (vectors); $|\cdot|$ stands for the magnitude; $(\cdot)^{\mathcal{T}}$ denotes transposition; $(\cdot)^{\mathcal{H}}$ complex-conjugate transposition; $\Re(\cdot)$/$\Im(\cdot)$ the real/imaginary part; $\mathrm{Tr}(\cdot)$ the matrix trace; $\mathrm{rank}(\cdot)$ the matrix rank; $\|\cdot\|_F$ the Frobenius norm; $\|\cdot\|_2$ the spectral norm; and $\mathrm{vec}(\cdot)$ the column-wise vectorization operator for a matrix.

2. PROBLEM FORMULATION

A power network can be modeled as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, with the set of buses (nodes) in $\mathcal{N} := \{1, ..., N\}$ and the set of lines (edges) in $\mathcal{E} := \{(n, m)\}$. The complex voltage phasor V_n per bus $n \in \mathcal{N}$ can be expressed in the rectangular coordinate as $V_n = \Re(V_n) + \mathtt{j}\Im(V_n)$. All nodal voltages form the full system state vector $\mathbf{v} := [V_1, ..., V_N]^{\mathcal{T}} \in \mathbb{C}^N$. To estimate nodal voltages in $\mathbf{v}$, a subset of the following system variables are measured:

- $P_n(Q_n)$: the active (reactive) power injection at bus n;
- $P_{mn}(Q_{mn})$: the active (reactive) power line flow from bus m to bus n;
- $|V_n|$: the voltage magnitude at bus n.

Based on the well-known AC power flow model [1, Ch. 4], the power variables are nonlinearly (quadratically) related to the system state $\mathbf{v}$. Collecting the noisy measurements in vector $\mathbf{z} \in \mathbb{R}^L$, where L denotes the total number of measurements, one can write the ℓ-th measurements

$$z_\ell = h_\ell(\mathbf{v}) + \epsilon_\ell,\ \forall \ell = 1, \ldots, L \tag{1}$$

where $h_\ell(\cdot)$ stands for the nonlinear transformation from $\mathbf{v}$, while ϵ_ℓ accounts for the additive measurement error. Assuming zero-mean random noise ϵ_ℓ of variance σ_ℓ^2, we can cast the state estimation (SE) problem with the weigthed least-squares (WLS) error objective, given by

$$\hat{\mathbf{v}} = \arg \min_{\mathbf{v} \in \mathbb{C}^N} \textstyle\sum_{\ell=1}^L w_\ell \big[z_\ell - h_\ell(\mathbf{v})\big]^2 \tag{2}$$

where parameters $\{w_\ell\}$ scale inverse proportionally with σ_ℓ^2. The Gauss-Newton method has been the workhorse solution for the nonlinear WLS-SE formulation; see e.g., [20]. It iteratively approximates the objective by linearizing (1) at the latest solution. This iterative linearization procedure, though computationally efficient if convergent, can be potentially divergent or stuck at saddle-points.

To tackle the nonlinearity in (2), one approach is to introduce the outer-product matrix $\mathbf{V} := \mathbf{v}\mathbf{v}^{\mathcal{H}} \in \mathbb{C}^{N \times N}$, consisting of all quadratic terms involving $\mathbf{v}$. This way, each measurement is linearly related to $\mathbf{V}$, as given by

$$z_\ell = \mathrm{Tr}(\mathbf{H}_\ell \mathbf{V}) + \epsilon_\ell,\ \forall \ell = 1, \ldots, L \tag{3}$$

where matrix $\mathbf{H}_\ell \in \mathbb{C}^{N \times N}$ is Hermitian matrix, which is defined in [4] based on the network topology and line parameters. Note that the voltage magnitude can be easily converted to its squared $|V_n|^2$ which also admits to (3). Reformulating the WLS objective in (2) using (3) leads to the following SDP problem:

$$\hat{\mathbf{V}} = \arg \min_{\mathbf{V} \in \mathbb{C}^{N \times N}} f(\mathbf{V}) := \sum_{\ell=1}^L w_\ell \big[z_\ell - \mathrm{Tr}(\mathbf{H}_\ell \mathbf{V})\big]^2 \tag{4a}$$

$$\text{s.t. } \mathbf{V} \succeq \mathbf{0} \tag{4b}$$

where the positive semi-definite (PSD) constraint in (4b) together with $\mathrm{rank}(\mathbf{V}) = 1$ can guarantee the existence of $\mathbf{v}$ that satisfies $\hat{\mathbf{V}} = \mathbf{v}\mathbf{v}^{\mathcal{H}}$. Due to the nonconvexity of rank constraint, it is dropped through a well-appreciated semidefinite relaxation (SDR) procedure that leads to a convex SDP-SE formulation (4). The latter can achieve a near-optimal performance for the SE problem as (4) typically admits to very low-rank solutions; see the tutorial overview in [9]. Furthermore, there exist a variety of general convex solvers that can obtain the optimal $\hat{\mathbf{V}}$ in polynomial time, such as the popular interior-point method based solver SeDuMi [8]. Nonetheless, these solution methods can scale unfavorably as the number of buses or measurements increases [9], with worst-case complexity at $\mathcal{O}(N^{4.5})$. Thus, it is necessary to develop accelerated algorithms for solving large-scale SDP-SE in real-time.

Motivated by recent work on nonconvex SDP solvers [14], we consider an equivalent formulation of (4) under low-rank conditions. The main challenge in SDP solution lies in the complexity of the PSD conic constraint (4b). Assuming $\mathrm{rank}(\hat{\mathbf{V}}) = r$ with $r \leq N$, one can express it as $\mathbf{V} = \mathbf{U}\mathbf{U}^{\mathcal{H}}$ with $\mathbf{U} \in \mathbb{C}^{N \times r}$ and reformulate (4) as an unconstrained one involving $\mathbf{U}$, namely,

$$\hat{\mathbf{U}} = \arg \min_{\mathbf{U} \in \mathbb{C}^{N \times r}} g(\mathbf{U}) := f(\mathbf{U}\mathbf{U}^{\mathcal{H}}). \tag{5}$$

Note that the case of $r = 1$ boils down to the original WLS problem (2). The advantage of using $\mathbf{U}$ for a general rank-r solution is two-fold: i) the relaxed SDP problem (4) is likely to attain a low-rank solution, motivating to search over the lower-dimensional space with small r; ii) by using a fixed r, the number of optimization variables would scale linearly with the problem dimension N, achieving computational gains. Note that nonconvexity is introduced to the objective function of (5) to bypass the complex PSD conic constraint. Thus, its solution techniques need more discussions as compared to general convex problems.

3. FACTORED GRADIENT DESCENT FOR SDP-SE

Leveraging the unconstrained structure and convenient gradient computation of the nonconvex reformulation, the so-termed FGD method has been proposed based on first-order updates for general SDP problems [14]. Under standard convexity and smoothness assumptions on f, it has been shown that FGD could converge sublinearly to the optimum with sufficiently accurate initialization. This convergence rate result is similar to that of regular gradient descent method for smooth convex functions [21, Ch. 1].

To invoke the FGD updates for SDP-SE, recall that $f(\mathbf{V})$ is a quadratic function, and thus the gradient

$$\begin{aligned}\nabla g(\mathbf{U}) =& 2\nabla f(\mathbf{U}\mathbf{U}^{\mathcal{H}}) \times \mathbf{U} \\ =& \textstyle\sum_{\ell=1}^{L} 4w_\ell \big[\mathrm{Tr}(\mathbf{U}^{\mathcal{H}}\mathbf{H}_\ell\mathbf{U}) - z_\ell\big]\mathbf{H}_\ell\mathbf{U}. \end{aligned} \quad (6)$$

For given step-size η, the iterative FGD updates is

$$\mathbf{U}_{k+1} = \mathbf{U}_k - \eta\nabla g(\mathbf{U}_k),\ \forall k = 0, 1, \ldots. \quad (7)$$

To select the η value, let M denote the smoothness parameter of $f(\cdot)$ such that $\|\nabla f(\mathbf{V}) - \nabla f(\mathbf{V}')\|_F \leq M \cdot \|\mathbf{V} - \mathbf{V}'\|_F$ holds for any two PSD matrices $\mathbf{V}$ and $\mathbf{V}'$. As f is quadratic, the lower bound of M is given by

$$M \geq \Big\| \textstyle\sum_{\ell=1}^{L} 2w_\ell \left[\mathrm{vec}(\mathbf{H}_\ell)\mathrm{vec}(\mathbf{H}_\ell)^{\mathcal{H}}\right] \Big\|_2 \quad (8)$$

As given in [14], the step-size can be set according to the initialization $\mathbf{V}_0 = \mathbf{U}_0\mathbf{U}_0^{\mathcal{H}}$, as

$$\eta = \frac{1}{16(M\|\mathbf{V}_0\|_2 + \|\nabla f(\mathbf{V}_0)\|_2)} \quad (9)$$

For the FGD algorithm as tabulated in Algorithm 1, its local convergence property can be established using the results in [14] for smooth functions.

Proposition 1. *Let the choice of rank $r = \mathrm{rank}(\hat{\mathbf{V}})$ with $\hat{\mathbf{V}} = \hat{\mathbf{U}}\hat{\mathbf{U}}^{\mathcal{H}}$ for the M-smooth f in* (4)*. Suppose the initialization $\mathbf{U}_0$ satisfies $\mathrm{Dist}(\mathbf{U}_0, \hat{\mathbf{U}}) \leq \rho\sigma_r(\hat{\mathbf{U}})$, where* Dist *denotes the minimum Frobenious error between the two matrices up to any rotational transformation and σ_r the r-th largest singular value, while ρ is a constant inversely scaling with the condition number of $\hat{\mathbf{V}}$. Under this constant relative error condition, the objective function sequence $f(\mathbf{U}_k\mathbf{U}_k^{\mathcal{H}})$ converges sublinearly to $f(\hat{\mathbf{V}})$ at the rate of $\mathcal{O}(1/k)$.*

The success of FGD critically depends on the choice of rank r and initialization $\mathbf{U}_0$. For functions that are only smooth, the condition for $\mathbf{U}_0$ is rather restrictive, and thus its choice is less well understood as compared to the case of strongly convex functions in [14]. There exist general initialization methods for SE, as detailed in the next section. Nonetheless, our numerical tests suggest that they fail to hold for Proposition 1. As a result, the empirical convergence speed for FGD would gradually decrease for (5). To tackle this, we will develop an acceleration scheme in the ensuring section for our SDP-SE problem.

Algorithm 1 Factored Gradient Descent (FGD)

Input: Function f, rank r, maximum iteration number K.
Output: $\mathbf{U}$ and $\mathbf{V} = \mathbf{U}\mathbf{U}^{\mathcal{H}}$

1: Initialize $\mathbf{U}_0 \in \mathbb{C}^{n\times r}$ and set $\mathbf{V}_0 = \mathbf{U}_0\mathbf{U}_0^{\mathcal{H}}$.
2: Set the step-size η as in (9).
3: **for** $k = 0$ to K **do**
4: $\quad \mathbf{U}_{k+1} = \mathbf{U}_k - \eta\nabla g(\mathbf{U}_k)$
5: **end for**
6: **return** $\mathbf{U} = \mathbf{U}_K$ and $\mathbf{V} = \mathbf{U}\mathbf{U}^{\mathcal{H}}$

4. ACCELERATED GRADIENT DESCENT METHOD

To tackle the slow convergence issue of FGD, we propose to adopt the popular acceleration scheme by Nesterov [15] to develop the AGD method as a heuristic alternative for the nonconvex SDP-SE problem (5).

Different from only using the instantaneous gradient information in FGD, AGD leverages the data from the past two iterations to accelerate the updates. Per iteration k, a time-varying interpolation is first performed to obtain

$$\mathbf{U}^{+} = \mathbf{U}_k + \left(\frac{k-2}{k+1}\right)(\mathbf{U}_k - \mathbf{U}_{k-1}), \quad (10)$$

which is used to compute the next iterate $\mathbf{U}_{k+1}$ as the gradient descent update at $\mathbf{U}^{+}$. For $\mathbf{U}_0$ at $k = 0$, the iterate $\mathbf{U}_1$ is simply computed using the FGD update (7). The AGD method is tabulated in Algorithm 2.

Remark 1. *(*AGD for nonconvex g.*)* The acceleration achieved by AGD for a convex g is guaranteed at the rate of $\mathcal{O}(1/k^2)$, as established in the classical work [15]. If g is nonconvex, the accelerate of AGD is less understood. There exist some recent results on the convergence rate guarantees for generally accelerated schemes under nonconvex objective functions. For example, under certain Lipschitz continuous conditions for the gradient and Hessian, [18] has developed an accelerated scheme that can find a stationary point with gradient norm less than ζ in $\mathcal{O}(\zeta^{-(7/4)})$ number of iterations. In addition, [19] has advocated AGD can better escape saddle points of nonconvex functions than gradient descent updates. It is challenging to establish the convergence guarantees for AGD in the SDP-SE problem. Instead, numerical studies have been used to demonstrate AGD outperforms FGD in terms of convergence time and also the optimality performance.

To implement the FGD and AGD updates, we set the same values for rank r, initial point $\mathbf{U}_0$, and step-size η using (9). As the rank relaxation is nearly optimal for (4), we set $r = 2$ to encourage rank-1 solutions while having low computational order. Based on our emipirical experience, the solution obtained with $r = 2$ exhibits better performance compared to the case of $r = 1$. The first column of $\mathbf{U}_0$ can follow from the initialization of classical GN methods. Specifically, the voltage magnitude is initialized as the corresponding meter

Algorithm 2 Accelerated Gradient Descent (AGD)

Input: Function f, rank r, iterations K.
Output: $\mathbf{U}$ and $\mathbf{V} = \mathbf{U}\mathbf{U}^{\mathcal{H}}$

1: Initialize $\mathbf{U}_0 \in \mathbb{C}^{n\times r}$ and set $\mathbf{V}_0 = \mathbf{U}_0\mathbf{U}_0^{\mathcal{H}}$.
2: Set the step-size η as in (9).
3: Compute $\mathbf{U}_1 = \mathbf{U}_0 - \eta\nabla g(\mathbf{U}_0)$.
4: **for** $k = 1$ to K **do**
5: Update $\mathbf{U}^+$ as (10) and $\mathbf{U}_{k+1} = \mathbf{U}^+ - \eta\nabla g(\mathbf{U}^+)$
6: **end for**
7: **return** $\mathbf{U}$ and $\mathbf{V} = \mathbf{U}\mathbf{U}^{\mathcal{H}}$

measurement, if available, while voltage phase angles are initialized as the solution to the linear DC flow model [20, Ch. 2]. As for the second column, it is randomly generated to be of unit Euclidean norm and orthogonal to $\mathbf{v}_0$.

In general, the solution $\hat{\mathbf{V}}$ achieved by the SDP problem is not exactly of rank 1, neither is that by FGD or AGD. Hence, it is necessary to recover the estimator $\hat{\mathbf{v}}$ from $\hat{\mathbf{V}}$. Following from earlier work [4], we use the eigen-decomposition $\hat{\mathbf{V}} = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i \mathbf{u}_i^{\mathcal{H}}$, where $\lambda_1 \geq \cdots \geq \lambda_r$ denote the positive ordered eigenvalues and $\mathbf{u}_i$ is the corresponding eigenvector of λ_i. The state estimator is set as $\hat{\mathbf{v}} = \sqrt{\lambda_1}\mathbf{u}_1$, corresponding to the best rank-one approximation of $\hat{\mathbf{V}}$. To reduce the optimality gap due to convex relaxation, the recovered vector $\hat{\mathbf{v}}$ is further used to initialize the Gauss-Newton updates for the nonlinear WLS formulation (2).

5. NUMERICAL RESULTS

The FGD and AGD methods have been tested on an Intel® CPU @ 2.2GHz (8GB RAM) computer using MATLAB® R2017a. They are compared with the SDP solution to (4) using the MATLAB-based optimization modeling package CVX [22] together with SeDuMi [8]. Two power transmission system test cases, the IEEE 118-bus and 300-bus systems, are used with the pertinent power flow solver and Gauss-Newton based SE iterations implemented by the MATLAB-based toolbox MATPOWER [23]. To generate the measurements, random Gaussian noise is added to the power flow output, with $\sigma_\ell = 0.02$ at power meters and 0.01 at voltage meters. Empirical estimation error $||\mathbf{v} - \hat{\mathbf{v}}||_2$ is computed by averaging over 300 Monte-Carlo realizations. For each realization, the actual bus voltage voltage magnitude follows from the Gaussian distribution $\mathcal{N}(0, 0.01)$, with the phase angle uniformly distributed over $[-0.35\pi, 0.35\pi]$.

The IEEE 118-bus case is first tested, with active/reactive power flows measured at all 186 lines and voltage magnitudes at 118 buses. The stop criteria for FGD/AGD iterations are the same, based on the consecutive change of the iterate and its objective value. The average run time for FGD is 77.64s, and that for AGD is 0.90s. Clearly, AGD has significantly reduced the number of iterations, as the per-iteration time is very close for the two. With the rank $r = 2$, we can compare

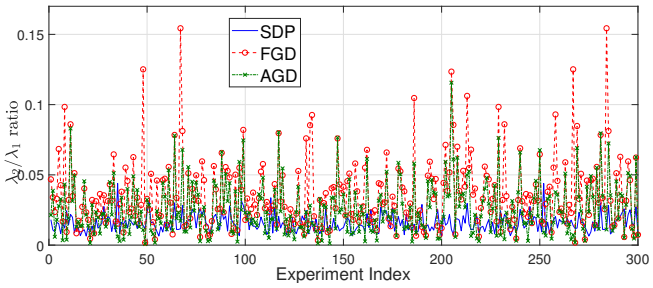

Fig. 1. The ratio (λ_2/λ_1) in 300 experiments for the IEEE 118-bus case.

Table 1. SE Error and GN Convergence Rate

SE Error	DC-GN	SDP-GN	AGD-GN
118-bus	0.288 (58.7%)	0.004 (100%)	0.019 (100%)
300-bus	1.46 (40.8%)	0.019 (100%)	0.086 (100%)

Table 2. Average Run Time of SDP and AGD

Time	SDP	AGD (per iteration)
118-bus	6.36s	0.90s (0.21ms)
300-bus	60.32s	15.02s (1.1ms)

the ratio between the second largest eigenvalue to the largest one (λ_2/λ_1) for the output solution $\mathbf{V}$, as plotted in Fig. 1. This ratio for AGD is smaller than that of FGD in almost all experiments, while the SDP solution is always the smallest. A smaller (λ_2/λ_1) ratio indicates the solution is more nearly rank-1, preferred for recovering the estimated $\hat{\mathbf{v}}$. This result suggests that AGD is better at approximating the SDP solution. Similar observations based on the achieved objective value and gradient norm corroborate that AGD has improved the performance of FGD while requiring less computational time. Thus, we have chosen the AGD method only for further comparisons of the SE performance to the SDP solution.

We compare the SE performance achieved by the AGD method to that by DC-SE and SDP solutions with each of the three as initialization for GN iterations. The average estimation error $||\mathbf{v} - \hat{\mathbf{v}}||_2$ is listed in Table 1, along with the percentage of convergence for the respective GN iterations. The AGD-GN is very competitive to the benchmark SDP-GN performance, with slightly higher estimation error probably due to numerical accuracy. Both can effectively address the divergence issue of GN iterations encountered by the DC-SE initialization. Moreover, the corresponding average run time as shown in Table 2 has corroborated the computational time improvement of AGD over with SDP solver. Note the per-iteration time in parenthesis shows good scalability of the AGD updates. These results demonstrate that AGD can efficiently solve the SDP-SE problem with good approximation accuracy. In future, we plan to investigate the analytical guarantee for the AGD method, while improve its computational efficiency by further leveraging the sparse problem structure.

6. REFERENCES

[1] Allen J Wood and Bruce F Wollenberg, *Power generation, operation, and control*, John Wiley & Sons, 2012.

[2] Georgios B Giannakis, Vassilis Kekatos, Nikolaos Gatsis, Seung-Jun Kim, Hao Zhu, and Bruce F Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 107–128, 2013.

[3] A Monticelli, "Electric power system state estimation," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 262–282, 2000.

[4] Hao Zhu and Georgios B Giannakis, "Power system nonlinear state estimation using distributed semidefinite programming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1039–1050, 2014.

[5] Yang Weng, Marija D Ilić, Qiao Li, and Rohit Negi, "Convexification of bad data and topology error detection and identification problems in ac electric power systems," *IET Generation, Transmission & Distribution*, vol. 9, no. 16, pp. 2760–2767, 2015.

[6] Yu Zhang, Ramtin Madani, and Javad Lavaei, "Conic relaxations for power system state estimation with line measurements," *IEEE Transactions on Control of Network Systems*, 2017.

[7] Ramtin Madani, Javad Lavaei, and Ross Baldick, "Convexification of power flow equations for power systems in presence of noisy measurements," *Preprint*, 2016.

[8] Jos F Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[9] Vassilis Kekatos, Gang Wang, Hao Zhu, and Georgios B Giannakis, "PSSE redux: Convex relaxation, decentralized, robust, and dynamic approaches," *arXiv preprint arXiv:1708.03981*, 2017.

[10] Gang Wang, Georgios B Giannakis, and Jie Chen, "Robust and scalable power system state estimation via composite optimization," *arXiv preprint arXiv:1708.06013*, 2017.

[11] Ramtin Madani, Abdulrahman Kalbat, and Javad Lavaei, "A low-complexity parallelizable numerical algorithm for sparse semidefinite programming," *IEEE Transactions on Control of Network Systems*, 2017.

[12] Samuel Burer and Renato DC Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003.

[13] Samuel Burer and Renato DC Monteiro, "Local minima and convergence in low-rank semidefinite programming," *Mathematical Programming*, vol. 103, no. 3, pp. 427–444, 2005.

[14] Srinadh Bhojanapalli, Anastasios Kyrillidis, and Sujay Sanghavi, "Dropping convexity for faster semi-definite optimization," in *Conference on Learning Theory*, 2016, pp. 530–582.

[15] Yurii Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," in *Soviet Mathematics Doklady*, 1983, vol. 27, pp. 372–376.

[16] Shuiwang Ji and Jieping Ye, "An accelerated gradient method for trace norm minimization," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 457–464.

[17] Naoki Ito, Akiko Takeda, and Kim-Chuan Toh, "A unified formulation and fast accelerated proximal gradient method for classification," *Journal of Machine Learning Research*, vol. 18, no. 16, pp. 1–49, 2017.

[18] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford, "Accelerated methods for nonconvex optimization," *SIAM Journal on Optimization*, vol. 28, no. 2, pp. 1751–1772, 2018.

[19] Chi Jin, Praneeth Netrapalli, and Michael I Jordan, "Accelerated gradient descent escapes saddle points faster than gradient descent," *arXiv preprint arXiv:1711.10456*, 2017.

[20] Antonio Gomez-Exposito and Ali Abur, *Power system state estimation: theory and implementation*, CRC press, 2004.

[21] Yurii Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2013.

[22] Michael Grant, Stephen Boyd, and Yinyu Ye, "Cvx: Matlab software for disciplined convex programming," 2008.

[23] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2011.