# Decentralized State Estimation using ADMM in Networks

Saurabh Sihag (RIN: 661679538)

## 1 Introduction

The problem of estimating an unknown set of states or parameters forms the basis of several applications like power systems, wireless sensor networks, internet of things etc. Performing state estimation in large networks is computationally complex. Also, a centralized estimation scheme, which collects the data from the whole network, has high communication overhead and not desirable in privacy-sensitive applications where it is not desirable for the information of the network to be available to a central entity. This motivates multi-area approach to state estimation, where the network is divided into multiple areas and each area is deemed to have some computational power. In this project, a decentralized estimation approach using ADMM is implemented for multi-area estimation.

Multi-area state estimation has been studied in the domain of power systems in [1, 2, 3, 4, 5]. State estimation is performed by coordination between local estimators and a centralized estimator in [2] and [3]. A distributed approach is adopted in [4] and [5], where the areas exchange their estimates for the shared states to converge to a common estimate. Specifically, in [4], an ADMM based approach is considered, which motivates the theory used in this project.

## 2 Network Model

In this project, the network shown in Fig. 1 is considered. The network consists of 19 edges and 14 nodes. This topology is borrowed from an IEEE-14 bus power system, however this network is not a power system. An unknown parameter is associated with an edge and the unknown state vector is denoted by $\boldsymbol{X} \in \mathbb{R}^{19\times 1}$. A measurement is taken at each node and each edge, which results in a measurement vector $\boldsymbol{Y} \in \mathbb{R}^{33\times 1}$. The relationship between the measurements and the state vector is given by

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{N} \ , \tag{1}$$

where $\boldsymbol{H} \in \mathbb{R}^{33\times19}$ is a known matrix and $\boldsymbol{N} \in \mathbb{R}^{33\times1}$ represents noise.

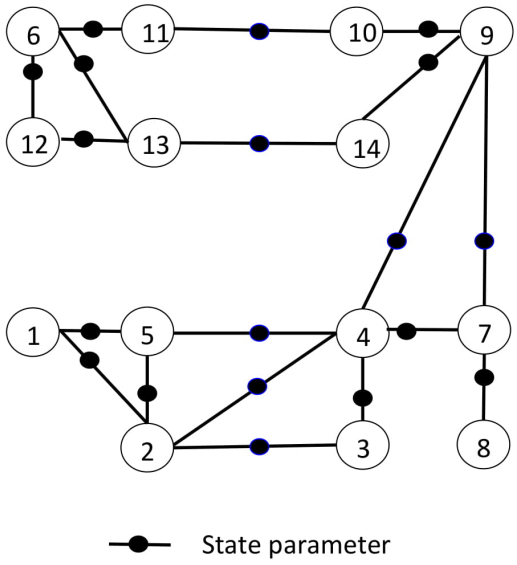


Figure 1: Network Model

The objective is to form an estimate for $\boldsymbol{X}$ by leveraging the measurements $\boldsymbol{Y}$. Furthermore, for the sake of clarity, it is assumed that the measurement at each node is the sum of the states corresponding to the edges that it is connected to with addition of noise. Also, the measurement at each edge is a noisy mapping of the state corresponding to that edge. Therefore, the elements of matrix $\boldsymbol{H}$ are either 0 or 1.

## 2.1 Multi-Area Network Model

In this project, a multi-area approach is adopted to lower the complexity of state estimation and illustrate the application of ADMM. The network model in Fig. 1 can be divided into multiple areas, as shown in Fig. 2.

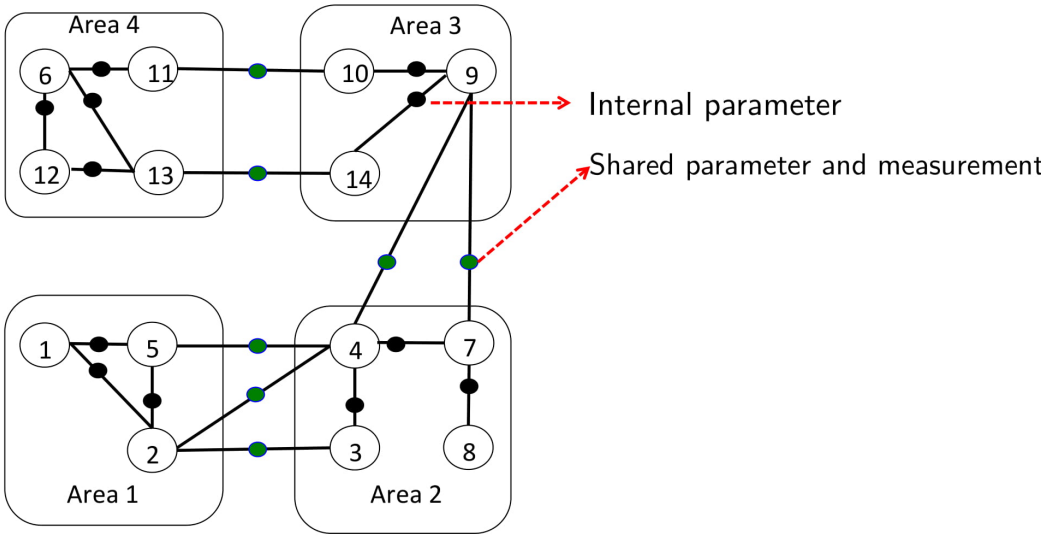


Figure 2: Multi-Area Network Model

For each Area $k \in \{1, 2, 3, 4\}$, its set of observed parameters can be classified as internal parameters, i.e., the parameters that are exclusively observed by that area, and external parameters, i.e., the parameters corresponding to tie lines or edges connecting any two areas. It is assumed that the measurement corresponding to a shared parameter is available to both areas that are observing it. For example, the parameter corresponding to edge $7-9$ in Fig. 2 is observed by Area 3 and Area 2, and the corresponding measurement is available to both Area 3 and Area 2. This allows both Area 3 and Area 2 to form an estimate for the shared parameter based on their respective set of measurements. Let $\boldsymbol{Y}^k$ be defined as the set of measurements collected by Area $k$, such that,

$$\boldsymbol{Y}^k = \boldsymbol{H}^k \boldsymbol{X}^k + \boldsymbol{N}^k \,, \tag{2}$$

where $\boldsymbol{X}^k$ is the set of parameters observed by Area $k$, $\boldsymbol{H}^k$ is the matrix that maps the unknown parameters to the measurements, and $\boldsymbol{N}^k$ is noise. For example, for Area 1, $\boldsymbol{Y}^1 \in \mathbb{R}^{9\times 1}$ and $\boldsymbol{X}^1 \in \mathbb{R}^{6\times 1}$.

# 3 State Estimation Model

Least squares estimation strategy is commonly used for state estimation, where the unknown parameter $\boldsymbol{X}$ is evaluated by solving

$$\hat{\boldsymbol{X}} = \arg\min_{\boldsymbol{X}} \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}\|_2^2 \,, \tag{3}$$

where $\hat{\boldsymbol{X}}$ is the estimate for $\boldsymbol{X}$. For small networks, solving the above optimization problem is trivial and can be done by calculating the exact solution or by using any of the existing first order or second order optimization methods. However, for reasons discussed in the introduction, adopting this centralized estimation scheme may neither be feasible or desirable.

Multi-area estimation approach helps in working around these issues. One naive estimation approach could be to solve

$$\hat{\boldsymbol{X}}^k = \arg\min_{\boldsymbol{X}^k} \frac{1}{2}\|\boldsymbol{Y}^k - \boldsymbol{H}^k\boldsymbol{X}^k\|_2^2 \,. \tag{4}$$

independently for each area and form the corresponding estimates of the unobserved subset of parameters. However, this approach is sub-optimal because of the presence of shared parameters whose information is available in the form of measurements at multiple areas. Secondly, adopting this independent estimation strategy may lead to different areas forming different estimates of the same parameter, which may not be desirable in sensitive applications like power systems or tracking/surveillance etc where the future decisions are based on the inference from the measurements. Therefore, it is desirable that the areas form the same estimates for the shared parameters.

Let $\hat{\boldsymbol{X}}^k = \{\hat{X}_j^k$ for all $X_j$ observed by Area $k\}$. Also, let $\{\hat{\boldsymbol{X}}^k\}$ denote the set of all estimates formed by all areas. Forming the estimates by solving the following optimization problem

$$\begin{aligned}\{\hat{\boldsymbol{X}}^k\} &= \arg\min_{\{\hat{\boldsymbol{X}}^k\}} \sum_{j=1}^{4} \frac{1}{2}\|\boldsymbol{Y}^j - \boldsymbol{H}^j\hat{\boldsymbol{X}}^j\|_2^2 \\ \text{s.t. } \hat{X}_i^l &= \hat{X}_i^m, \quad \forall\text{states } X_i \text{ shared by areas } l \text{ and } m\end{aligned} \tag{5}$$

allows eliminating the discrepancy between the estimates of the same parameter formed by multiple areas. However, note that the constraint $\hat{X}_i^l = \hat{X}_i^m$ couples the estimation process across the areas in the sense that each area cannot independently process its measurements without having access to the estimates of the shared parameters from other areas. This implies joint processing of the measurements which has the similar computational complexity issues as discussed before. Therefore, the problem in (5) is modified by introduction of the auxiliary variables, such that,

$$\begin{aligned}\{\hat{\boldsymbol{X}}^k\} &= \arg\min_{\{\hat{\boldsymbol{X}}^k\},\{x_k\}} \sum_{j=1}^{4} \frac{1}{2}\|\boldsymbol{Y}^j - \boldsymbol{H}^j\hat{\boldsymbol{X}}^j\|_2^2 \\ \text{s.t. } \hat{X}_i^l &= x_i, \quad \forall\text{states } X_i \text{ shared by areas } l\end{aligned} \tag{6}$$

where $x_i$ are the auxiliary variables and $\{x_k\}$ denotes the set of all auxiliary variables. This modification to the optimization problem is motivated from the theory in [4]. Using ADMM

approach to solve (6) helps in decentralizing the estimation process because the estimates $\hat{\boldsymbol{X}}^l$ at all areas $l$ can be updated locally and independently of other areas based on the values of auxiliary variables in the current iteration. However, a central entity is required for updating the values of the auxiliary variables, which does not have high computational complexity. Solving the problem in 6 using ADMM involves forming the augmented Lagrangian function

$$\ell_\beta(\{\hat{\boldsymbol{X}}^k\},\{x_k\},\{\lambda_i^k\}) = \sum_{j=1}^{4}\left[\frac{1}{2}\|\boldsymbol{Y}^j - \boldsymbol{H}^j\hat{\boldsymbol{X}}^j\|_2^2 + \sum_{m\in\mathcal{S}^j}\left(\lambda_m^j(\hat{X}_m^j - x_m) + \frac{\beta}{2}(\hat{X}_m^j - x_m)^2\right)\right] \tag{7}$$

where $\lambda_j^l$ is a Lagrange multiplier and $\{\lambda_i^k\}$ is the set of all Lagrange multipliers. $\beta$ is a penalty parameter. $\mathcal{S}^j$ is the set of all parameters that are shared by Area $j$ with other areas. The ADMM steps to solve (6) are summarized below:
At iteration $t+1$,

$$1.\{\hat{\boldsymbol{X}}^k\}^{(t+1)} = \arg\min_{\{\hat{\boldsymbol{X}}^k\}} \ell_\beta(\{\hat{\boldsymbol{X}}^k\},\{x_k\}^{(t)},\{\lambda_i^k\}^{(t)}) \tag{8}$$

$$2.\{x_k\}^{(t+1)} = \arg\min_{\{x_k\}} \ell_\beta(\{\hat{\boldsymbol{X}}^k\}^{(t+1)},\{x_k\},\{\lambda_i^k\}^{(t)}) \tag{9}$$

$$3.\{\lambda_i^k\}^{(t+1)} = \{\lambda_i^k\}^{(t)} + \beta(\{\hat{X}_i^k\}^{(t+1)} - \{X_i\}^{(t+1)}) \tag{10}$$

Note that step 1 can be decentralized by performing independent local processing at individual areas i.e. at every iteration, the step 1 is equivalent to solving

$$\hat{\boldsymbol{X}}^j = \arg\min_{\hat{\boldsymbol{X}}^j} \frac{1}{2}\|\boldsymbol{Y}^j - \boldsymbol{H}^j\hat{\boldsymbol{X}}^j\|_2^2 + \sum_{m\in\mathcal{S}^j}\left(\lambda_m^j(\hat{X}_m^j - x_m) + \frac{\beta}{2}(\hat{X}_m^j - x_m)^2\right) \tag{11}$$

at every area. Step 2 is equivalent to solving

$$\{x_k\} = \arg\min_{\{x_k\}} \sum_{j=1}^{4}\sum_{m\in\mathcal{S}^j}\left(\lambda_m^j(\hat{X}_m^j - x_m) + \frac{\beta}{2}(\hat{X}_m^j - x_m)^2\right) . \tag{12}$$

Steps 1 and 2 can be simply solved using gradient descent. Although with proper analysis, a closed form optimal solution can be found to Step 2.

# 4 Robust Estimation

In practical settings, the network model could deviate from (1) and (2) because of external disturbance or errors in the sensing units that are collecting measurements. In this project, two types of disturbances or deviations in the nominal model are considered.

## 4.1 Bad Data

The bad data is modeled as an additive unknown disturbance in the measurements that can compromise estimation quality. Let the bad data affecting the measurements of Area $k$ be denoted by $\boldsymbol{Z}^k$. In general, only a small number of measurements undergo such additive disturbances, and therefore, $\boldsymbol{Z}^k$ is expected to be sparse. This motivates the modification in the problem (6) to form

$$\{\hat{\boldsymbol{X}}^k\} = \arg \min_{\{\hat{\boldsymbol{X}}^k\},\{x_k\},\{\boldsymbol{Z}^k\}} \sum_{j=1}^{4} \frac{1}{2}\|\boldsymbol{Y}^j - \boldsymbol{H}^j\hat{\boldsymbol{X}}^j - \boldsymbol{Z}^j\|_2^2 + \gamma_j\|\boldsymbol{Z}^j\|_1$$
$$\text{s.t. } \hat{X}_i^l = x_i, \quad \forall \text{states } X_i \text{ shared by areas } l \tag{13}$$

where $\gamma_j\|\boldsymbol{Z}^j\|_1$ is a regularizer term to control the sparsity of $\boldsymbol{Z}^j$ in Area $j$. The problem in (13) can be solved using ADMM. The augmented Lagrangian function for this problem is given by

$$\ell_\beta(\{\hat{\boldsymbol{X}}^k\},\{x_k\},\{\boldsymbol{Z}^k\},\{\lambda_i^k\}) = \sum_{j=1}^{4}\Bigg[\frac{1}{2}\|\boldsymbol{Y}^j - \boldsymbol{H}^j\hat{\boldsymbol{X}}^j - \boldsymbol{Z}^j\|_2^2 + +\gamma_j\|\boldsymbol{Z}^j\|_1 + \sum_{m\in\mathcal{S}^j}\left(\lambda_m^j(\hat{X}_m^j - x_m) + \frac{\beta}{2}(\hat{X}_m^j - x_m)^2\right)\Bigg] \tag{14}$$

and the corresponding ADMM steps are given by:
At iteration $t+1$,

$$1.\{\hat{\boldsymbol{X}}^k\}^{(t+1)} = \arg\min_{\{\hat{\boldsymbol{X}}^k\}} \ell_\beta(\{\hat{\boldsymbol{X}}^k\},\{x_k\}^{(t)},\{\boldsymbol{Z}^k\}^{(t)},\{\lambda_i^k\}^{(t)}) \tag{15}$$

$$2.\{x_k\}^{(t+1)} = \arg\min_{\{x_k\}} \ell_\beta(\{\hat{\boldsymbol{X}}^k\}^{(t+1)},\{x_k\},\{\boldsymbol{Z}^k\}^{(t)},\{\lambda_i^k\}^{(t)}) \tag{16}$$

$$3.\{\lambda_i^k\}^{(t+1)} = \{\lambda_i^k\}^{(t)} + \beta(\{\hat{X}_i^k\}^{(t+1)} - \{X_i\}^{(t+1)}) \tag{17}$$

$$4.\{\boldsymbol{Z}^k\}^{(t+1)} = \arg\min_{\{\boldsymbol{Z}^k\}} \ell_\beta(\{\hat{\boldsymbol{X}}^k\}^{(t+1)},\{x_k\}^{(t+1)},\{\boldsymbol{Z}^k\},\{\lambda_i^k\}^{(t+1)}) \tag{18}$$

Step 1 and Step 4 can be decentralized to individual areas in the similar way as done before. Step 1 is equivalent to solving

$$\hat{\boldsymbol{X}}^j = \arg\min_{\hat{\boldsymbol{X}}^j} \frac{1}{2}\|\boldsymbol{Y}^j - \boldsymbol{H}^j\hat{\boldsymbol{X}}^j - \boldsymbol{Z}^j\|_2^2 + \sum_{m\in\mathcal{S}^j}\left(\lambda_m^j(\hat{X}_m^j - x_m) + \frac{\beta}{2}(\hat{X}_m^j - x_m)^2\right) \tag{19}$$

at all Areas $j$. Step 4 is equivalent to solving

$$\boldsymbol{Z}^j = \arg\min_{\boldsymbol{Z}^j} \frac{1}{2}\|\boldsymbol{Y}^j - \boldsymbol{H}^j\hat{\boldsymbol{X}}^j - \boldsymbol{Z}^j\|_2^2 + \gamma_j\|\boldsymbol{Z}^j\|_1 \tag{20}$$

at all Areas $j$. Step 4 can be solved by using proximal gradient descent.

## 4.2 Error in Sensing Units

The second source of deviation in the nominal model ((1) and (2)) is assumed to be the error in the sensing units, i.e., when there is a change in the matrix $\boldsymbol{H}$ and $\boldsymbol{H}^k$ that link the unknown state parameters to the set of measurements. Let the deviated model at area $j$ be given by

$$\boldsymbol{Y}^j = \boldsymbol{H}^j \boldsymbol{X}^j + \boldsymbol{N}^j - \boldsymbol{H}_e^j \boldsymbol{X}^j \tag{21}$$

where the matrix $\boldsymbol{H}_e^j$ is the difference between the true matrix that maps $\boldsymbol{X}^j$ to $\boldsymbol{Y}^j$ and the matrix $\boldsymbol{H}^j$ under the nominal model. If there is no change in the nominal model, the matrix $\boldsymbol{H}_e^j$ is a zero matrix. For the sake of simplicity, it is assumed that the sensing units that are observing shared parameters continue to do so even in the deviated model, i.e., the columns of $\boldsymbol{H}_e^j$ corresponding to shared parameters are 0-columns. The sensing units that are observing a non-empty set of unknown unshared parameter in an area under nominal model may be erroneous, such that, they stop observing a subset of the corresponding observed unshared parameters in the deviated model. Therefore, some elements of $\boldsymbol{H}_e^j$ corresponding to sensing units that are observing unshared parameters could be 1. All the 0 elements of $\boldsymbol{H}^j$ are also 0 in $\boldsymbol{H}_e^j$. Thus, $\boldsymbol{H}_e^j$ is expected to be a sparse matrix as only a small subset of sensing units are expected to develop faults. $\boldsymbol{H}_e^j$ can be determined while performing estimation by solving the problem

$$\begin{aligned} \{\hat{\boldsymbol{X}}^k\} = \arg \min_{\{\hat{\boldsymbol{X}}^k\},\{x_k\},\{\boldsymbol{H}_e^k\}} \sum_{j=1}^{4} \frac{1}{2} \|\boldsymbol{Y}^j - \boldsymbol{H}^j \hat{\boldsymbol{X}}^j + \boldsymbol{H}_e^j \boldsymbol{X}^j\|_2^2 + \eta_j \|\boldsymbol{H}_e^j\|_1 \\ \text{s.t. } \hat{X}_i^l = x_i, \quad \forall \text{states } X_i \text{ shared by areas } l \end{aligned} \tag{22}$$

where $\eta_j \|\boldsymbol{H}_e^j\|_1$ is a regularizer term to control sparsity of $\boldsymbol{H}_e^j$. The augmented Lagrangian function for the above problem is defined as

$$\begin{aligned} \ell_\beta(\{\hat{\boldsymbol{X}}^k\}, \{x_k\}, \{\boldsymbol{H}_e^k\}, \{\lambda_i^k\}) = \sum_{j=1}^{4} \Bigg[ \frac{1}{2} \|\boldsymbol{Y}^j - \boldsymbol{H}^j \hat{\boldsymbol{X}}^j + \boldsymbol{H}_e^j \hat{\boldsymbol{X}}^j\|_2^2 + + \eta_j \|\boldsymbol{H}_e^j\|_1 + \\ \sum_{m \in \mathcal{S}^j} \left( \lambda_m^j (\hat{X}_m^j - x_m) + \frac{\beta}{2} (\hat{X}_m^j - x_m)^2 \right) \Bigg] \end{aligned} \tag{23}$$

The ADMM steps are summarized below:
At iteration $t+1$,

$$1.\{\hat{\boldsymbol{X}}^k\}^{(t+1)} = \arg\min_{\{\hat{\boldsymbol{X}}^k\}} \ell_\beta(\{\hat{\boldsymbol{X}}^k\}, \{x_k\}^{(t)}, \{\boldsymbol{H}_e^k\}^{(t)}, \{\lambda_i^k\}^{(t)}) \tag{24}$$

$$2.\{x_k\}^{(t+1)} = \arg\min_{\{x_k\}} \ell_\beta(\{\hat{\boldsymbol{X}}^k\}^{(t+1)}, \{x_k\}, \{\boldsymbol{H}_e^k\}^{(t)}, \{\lambda_i^k\}^{(t)}) \tag{25}$$

$$3.\{\lambda_i^k\}^{(t+1)} = \{\lambda_i^k\}^{(t)} + \beta(\{\hat{X}_i^k\}^{(t+1)} - \{X_i\}^{(t+1)}) \tag{26}$$

$$4.\{\boldsymbol{H}_e^k\}^{(t+1)} = \arg\min_{\{\boldsymbol{H}_e^k\}} \ell_\beta(\{\hat{\boldsymbol{X}}^k\}^{(t+1)}, \{x_k\}^{(t+1)}, \{\boldsymbol{H}_e^k\}, \{\lambda_i^k\}^{(t+1)}) \tag{27}$$

Solving steps 1-3 is straightforward and discussed before. Note that only a subset of all elements of the matrix $\boldsymbol{H}_e^j$ are not fixed and can be either 0 or 1, which makes the problem in step 4 an integer programming problem. This can be solved using two approaches:

- Convex relaxation and thresholding: The feasible set of $\boldsymbol{H}_e^j$ is assumed to be all matrices $\boldsymbol{H}_e$, s.t. $\|\boldsymbol{H}_e\|_2 \leq 1$. Under this assumption, the problem in step 4 can be solved using projected gradient descent. This followed by thresholding the elements of the solution such that the uncertain elements of $\boldsymbol{H}_e^j$ below that threshold are set to 0 and above the or equal to the threshold are set to 1. This threshold is determined by running multiple experiments.

- Greedy approach: The matrix $\boldsymbol{H}_e^j$ for Area $j$ is initialized as a zero matrix. Start with the first uncertain element in $\boldsymbol{H}_e^j$ and calculate the objective value of the problem in (22) by setting the uncertain element to 0 ( case 0) and 1 (case 1). If the objective value corresponding to case 0 is less than the one corresponding to case 1, the uncertain element is set to 0. Otherwise, it is set to 1. Keeping the value of this uncertain element fixed, the next uncertain element can be chosen randomly or according to a predefined sequence and the procedure is replicated for all uncertain elements, till all uncertain elements are exhausted.

# 5 Simulations

## 5.1 ADMM based estimation

- Convergence of ADMM
  The objective function of (5) is plotted versus the number of iterations in 3.

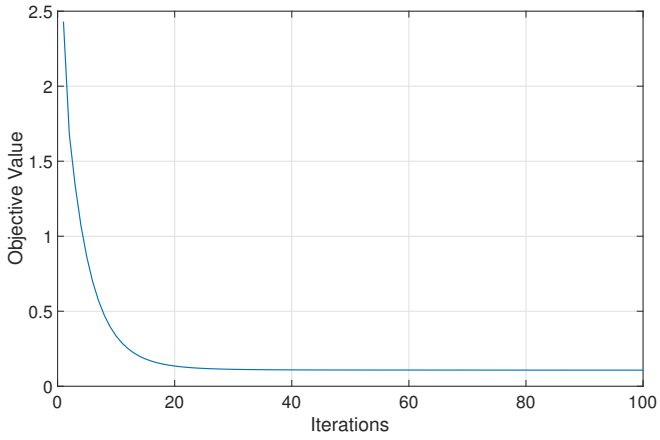


Figure 3: Objective value vs Number of iterations

For fig. 3, the measurements are generated by first generating the unknown parameters randomly with distribution $\mathsf{Unif}[-1, 1]$ that are added with the noise generated with the normal distribution $\mathcal{N}(0, 0.01)$. The state estimation is performed using ADMM based approach.

- Discrepancy between shared parameter estimates:
  Each area forms an estimate for its observed parameters, which implies that there are multiple estimates formed for shared parameters. The constraints in the probelm (22) aim to minimize this discrepancy, which is illustrated with the help of simulations. After 100 iterations, the average variance of the estimates of shared parameters is $1.7 \times 10^{-4}$ for ADMM based estimation, and it is $8 \times 10^{-4}$ for independent local estimation.

- Empirical estimation error evaluation:
  The main objective of state estimation strategy is to recover the set of unknown parameters at each Area. Therefore, lowering of discrepancy between the multiple estimates of the shared parameters is meaningful only if it also provides an improvement in the estimation quality as compared to that obtained using independent local processing. To illustrate this improvement in estimation quality, an empirical estimation error is evaluated for each area, which is given by

$$\text{Empirircal Error for Area } j = \sum_{n=1}^{M} \frac{\|\hat{\boldsymbol{X}}^j(n) - \boldsymbol{X}_{\mathsf{T}}^j\|_2^2}{M} \tag{28}$$

  where $\hat{\boldsymbol{X}}^j(n)$ is the estimate for $\boldsymbol{X}^j$ formed in the simulation $n$ and $\boldsymbol{X}_{\mathsf{T}}^j$ is the true value of the unknown parameters that were used to generate the measurements. The

number of Monte-Carlo simulations $M$ is set to 100 and this empirical error is evaluated till 500 iterations for all areas, and plotted in Fig. 4.

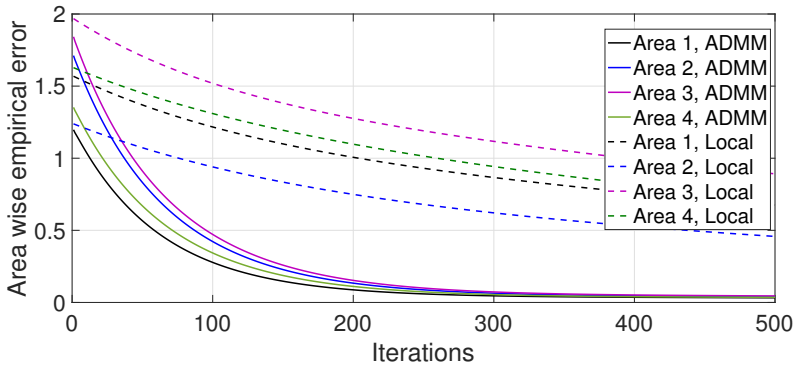


Figure 4: Empirical estimation error vs Number of iterations

It is observed that the estimates formed using ADMM based estimation strategy are closer to the ground truth as compared to those formed using independent local estimation strategy.

## 5.2 Robust Estimation

### 5.2.1 Bad Data

Bad data $\boldsymbol{Z}^k$ is generated for each area $k$ by generating a random sparse vector with density 0.1. The regularization parameter $\gamma_k$ is tuned accordingly to control the sparsity of the estimated bad data. Figure 5 shows the plots of $\sum_{k=1}^{4} \frac{1}{2}\|\boldsymbol{Y}^k - \boldsymbol{H}^k\hat{\boldsymbol{X}}^k - \boldsymbol{Z}^k\|_2^2$ for robust estimation and $\sum_{k=1}^{4} \frac{1}{2}\|\boldsymbol{Y}^k - \boldsymbol{H}^k\hat{\boldsymbol{X}}^k\|_2^2$ for non robust estimation (ignoring the effect of $\boldsymbol{Z}^k$) versus the number of iterations. Clearly robust estimation strategy is able to achieve better performance as compared to a non robust estimation strategy.

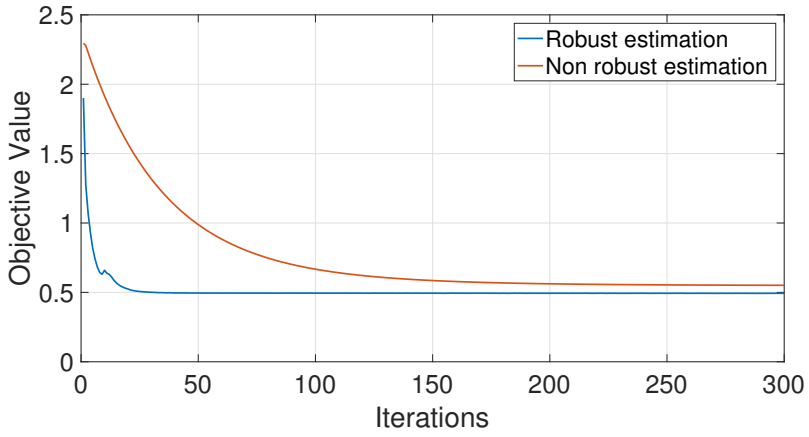


Figure 5: Objective value vs Number of iterations

Figure 6 shows the plot for empirical error for Area 1 for robust and non robust estimation.

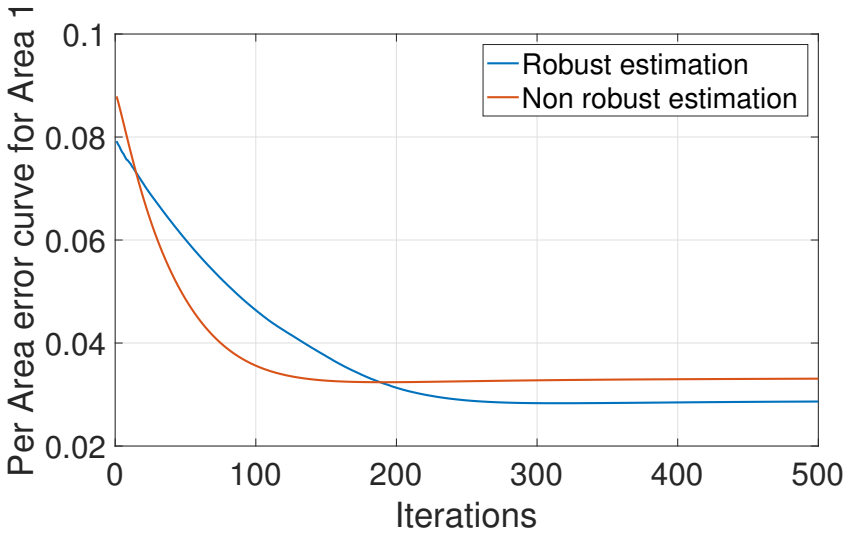


Figure 6: Empirical estimation error vs Number of iterations

### 5.2.2 Error in Sensing Units

4 sensing errors are introduced in the network. The regularization parameters $\eta_k$ are tuned to control the sparsity of estimated sensing error matrices $\boldsymbol{H}_e^k$. Figure 7 shows the plots of $\sum_{k=1}^{4} \frac{1}{2}\|\boldsymbol{Y}^k - \boldsymbol{H}^k\hat{\boldsymbol{X}}^k + \boldsymbol{H}_e^k\hat{\boldsymbol{X}}^k\|_2^2$ for two robust estimation approaches and $\sum_{k=1}^{4} \frac{1}{2}\|\boldsymbol{Y}^k - \boldsymbol{H}^k\hat{\boldsymbol{X}}^k\|_2^2$ for non robust estimation (ignoring the effect of sensing errors) versus the number of iterations. As expected, the robust estimation strategies are able to achieve better performance as compared to non robust estimation strategy. Also, the greedy approach performs better than the convex relaxation approach for robust estimation.

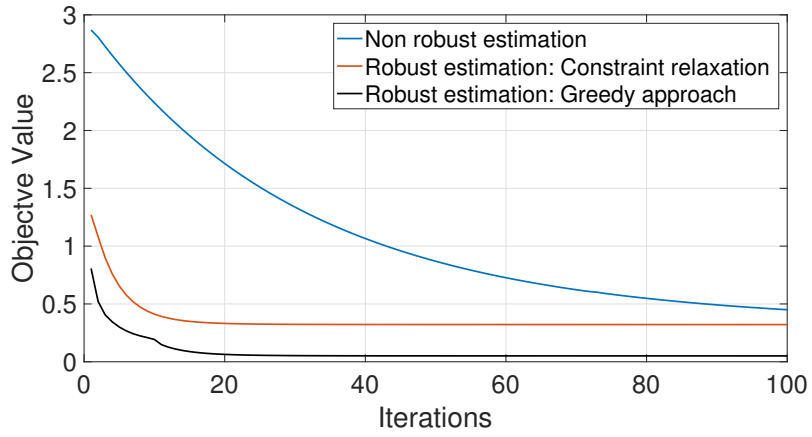


Figure 7: Objective value vs Number of iterations

Figure 8 shows the plot for empirical error for Area 2 for robust estimation using convex relaxation and non robust estimation.

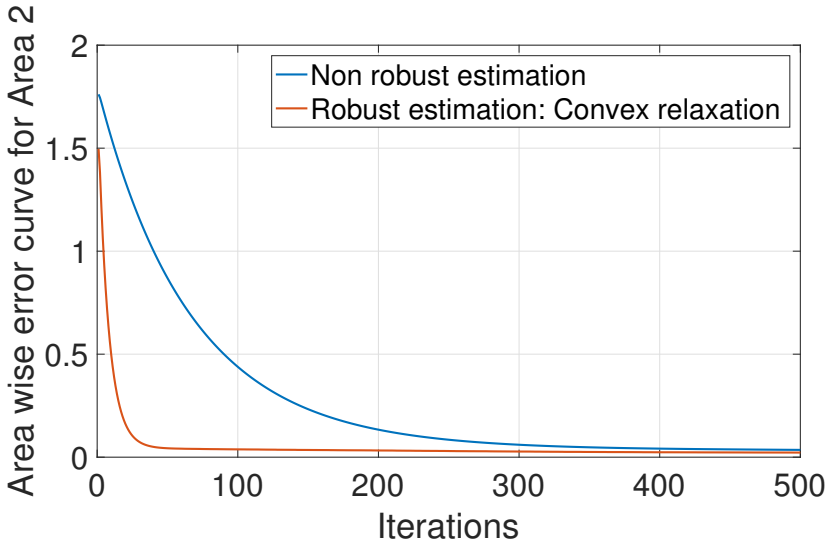


Figure 8: Empirical estimation error vs Number of iterations

Figure 9 shows the plot for empirical error for Area 3 for robust estimation using greedy approach and non robust estimation.

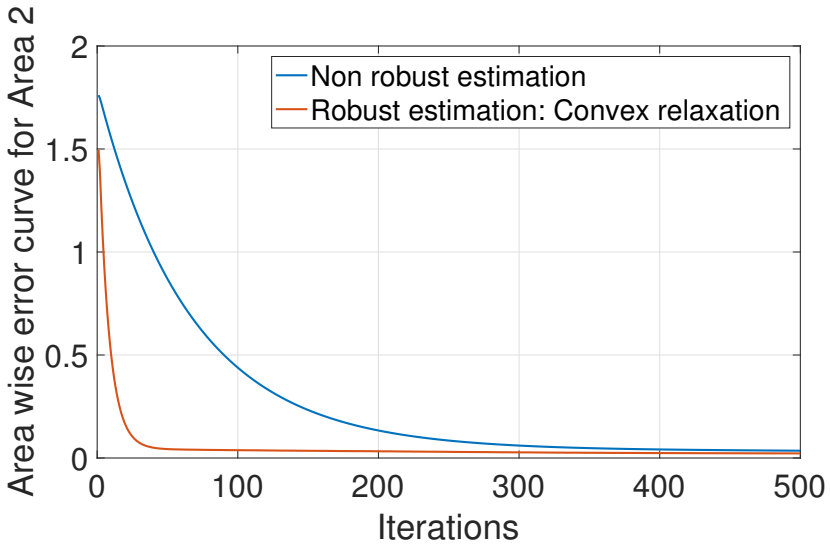


Figure 9: Empirical estimation error vs Number of iterations

# 6 Discussion

To deal with the computational complexity issue in state estimation in large networks, an ADMM based decentralized estimation is evaluated on a network. The network is divided into multiple areas and the estimation problem is posed as an optimization problem with constraints that encourage consistency between the estimates of shared parameters formed by multiple areas. From the simulations, it is observed that the ADMM based estimation strategy allows for better estimation quality at all areas as compared to performing independent local estimation.

In practical situations, the network model can deviate from the true model because of external disturbances or errors in the sensing units. Two possible sources of model deviation: bad data and sensing unit errors, are considered and robust estimation strategies are developed. It is observed that the robust estimation strategies provide better performance as compared to a non robust estimation strategy that ignores the effect of these imperfections in the nominal model.

# References

[1] X. Li and A. Scaglione, "Robust decentralized state estimation and tracking for power systems via network gossiping." *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1184–1194, Jul. 2013. 1

[2] L. Zhao and A. Abur, "Multi area state estimation using synchronized phasor measurements." *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 611–617, May 2005. 1

[3] G. N. Korres, "A distributed multiarea state estimation." *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 73–84, Apr. 2011. 1

[4] V. Kekatos and G. B. Giannakis., "Decentralized power system state estimation," *Global Communications Conference*, Dec. 2012. 1, 4

[5] A. J. Conejo, S. de la Torre, and M. Canas, "An optimization approach to multiarea state estimation," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 213–221, Feb. 2007. 1