Assessing the Impacts of Nonideal Communications on Distributed Optimal Power Flow Algorithms

Mohannad Alkhraijah, Carlos Menendez, and Daniel K. Molzahn
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA, USA

***Abstract*—Power system operators are increasingly looking toward distributed optimization to address various challenges facing electric power systems. To assess their capabilities in environments with nonideal communications, this paper investigates the impacts of data quality on the performance of distributed optimization algorithms. Specifically, this paper compares the performance of the Alternating Direction Method of Multipliers (ADMM), Analytical Target Cascading (ATC), and Auxiliary Problem Principle (APP) algorithms in the context of DC Optimal Power Flow (DC OPF) problems. Using several test cases, this paper characterizes the performance of these algorithms in terms of their convergence rates and solution quality under three data quality nonidealities: (1) additive Gaussian noise, (2) bad data (large error), and (3) intermittent communication failure.**

***Index Terms*—Distributed Optimization, Nonideal Communication, Optimal Power Flow.**

I. Introduction

Traditionally, power system operations are primarily conducted centrally, where the tasks of modeling the system's components, solving an optimization problem, and dispatching setpoints are performed by the system operator. However, as power systems move toward a more decentralized paradigm with many locally controlled microgrids, scaling these tasks becomes challenging when using centralized methods. Further, modeling connected systems is difficult, especially for microgrids in a distribution network.

Distributed algorithms address these challenges by allowing interconnected systems with local controllers to cooperatively solve large optimization problems. This increases flexibility as each microgrid solves a local subset of the original optimization problem. Distributed algorithms thus have various potential advantages, such as allowing parallel computations, reducing the requisite communication infrastructure, and possibly maintaining the subsystems' autonomy and data privacy.

These advantages motivate solving Optimal Power Flow (OPF) problems in a distributed fashion. OPF problems seek optimal setpoints for a power system. The decision variables are typically the generators' power outputs and the voltage phasors. Several distributed optimization algorithms have been proposed for solving OPF problems [1]. Numerical comparisons among different distributed OPF algorithms are presented in [2], [3]. Further, a numerical analysis presented in [4] shows that distributed optimization algorithms might converge to suboptimal solutions for nonconvex problems, depending on the initialization.

Communication plays a major role in implementing distributed optimization since regions share data with their neighbors [5]. The shared data between interconnected regions can be subject to communication errors and malicious attacks. Reference [6] provides a detailed study on the communication infrastructure requirement for distributed optimization that solves power system problems. The authors investigate the communication impact of two distributed optimization algorithms in terms of time delays. Other types of communication nonidealities can occur to the shared data between neighboring region while using distributed optimization. The authors of [7] investigate the impact of injecting false values into the shared data of the Analytical Target Cascading (ATC) algorithm solving network constraint unit commitment problem and show that the error injection increases the convergence time. Nevertheless, it is difficult to derive a conclusion about the robustness of the algorithms when considering one type of communication nonideality with a single erroneous value injection.

In more general distributed optimization settings, the impacts of communication noise due to quantization effects have been investigated in [8]–[10]. The authors of [11] provide analytic upper and lower bounds on the Alternating Direction Method of Multipliers (ADMM) algorithm's performance in the presence of random errors and validate the results using randomly generated networks. The authors of [12] study the robustness of distributed algorithms based on dual averaging to additive communication noise. The authors of [13] investigate the ADMM algorithm's convergence under additive communication noise and recommend modifications to the underlying optimization problem. Packet loss is another communication issue that has been discussed in the literature. The impact of packet loss on unconstrained convex distributed optimization is investigated in [14].

In this paper, we compare the performance of three distributed algorithms, Alternating Direction Method of Multipliers (ADMM), Analytical Target Cascading (ATC), and Auxiliary Problem Principle (APP), in solving OPF problems in the presence of nonideal communications. We characterize the algorithms' performance in terms of convergence speed and solution quality. To serve as a benchmark, we first compare the performance of the three algorithms with ideal communications. We then consider three noise models that impact the shared data between neighboring regions: (1) additive Gaussian noise, (2) bad data (large error), and (3) intermittent communication loss. We use three test cases to evaluate the performance of the algorithms with nonideal communication. This paper considers the DC OPF formulation which uses the

DC power flow linearization [15]. Thus, this paper's main contribution is an extensive empirical study regarding the impacts of nonideal communication on various distributed solution algorithms for DC OPF problems.

We note that the convexity of the DC OPF formulation studied in this paper provides advantages in terms of theoretical convergence guarantees for the distributed algorithms. These guarantees are useful in the context of this paper as they ensure that the distributed algorithms should all converge to the same solution, thus permitting consistent comparisons that primarily focus on the speed and robustness of the algorithms. However, there are applications where the DC power flow approximation is inappropriate, thus requiring alternative power flow representations [16]. This paper forms a basis for future extensions of our analyses for these applications. Further, this paper only shows the main results and findings of the study. Due to the limitation on the content length, we refer the reader to [17] for the full empirical results and figures.

The rest of the paper is organized as follows. Section II introduces the DC OPF problem. Section III formulates the problem decomposition and the distributed solution algorithms. Section IV describes the noise models used in the analysis. Section V presents results and discusses the distributed algorithms' performance under nonideal communication. Section VI summarizes our findings and discusses future work.

Remark on Notation

We use bold letters to indicate vectors. We denote a local subproblem and regions with the letter m and let $\mathcal{M}$ indicate the set of all subproblems. The set $\mathcal{N}_s^m$ contains the boundary variables for region m. We use the letters n and c to denote variables from the neighboring regions and the central coordinator. We use $||\cdot||$ to denote the vector l_2-norm. We use the hat in $\hat{x}$ to indicate that the variables x are evaluated with constant values from the previous iteration as received from neighboring regions or obtained from the prior local solution.

II. DC Optimal Power Flow Formulation

OPF problems typically minimize generation costs while satisfying the power flow equations and limits on generator outputs, line flows, etc. The optimization variables are the generators' outputs and the bus voltages. We consider the DC power flow approximation [15] to obtain convergence guarantees for the distributed algorithms, which are discussed in Section III. The DC OPF formulation used in this paper is:

$$\min_{\boldsymbol{\theta},\boldsymbol{p}} \quad \sum_{i\in\mathcal{G}} f_i(p_i) \tag{1a}$$

$$\text{s.t.} \quad p_i - d_i = \sum_{(i,j)\in\mathcal{L}} B_{ij}(\theta_i - \theta_j), \qquad \forall i \in \mathcal{B} \tag{1b}$$

$$P_i^{min} \le p_i \le P_i^{max}, \qquad \forall i \in \mathcal{G} \tag{1c}$$

$$-P_{ij}^{max} \le B_{ij}(\theta_i - \theta_j) \le P_{ij}^{max}, \quad \forall (i,j) \in \mathcal{L} \tag{1d}$$

$$\theta^{ref} = 0 \tag{1e}$$

where the sets $\mathcal{B}$, $\mathcal{G}$, and $\mathcal{L}$ denote the buses, generators, and lines, respectively. The generation costs are denoted by f_i. The decision variables are $\boldsymbol{\theta}$, the bus voltage angles, and $\boldsymbol{p}$, the generators' active power outputs. We denote the power demands by d, and B denotes the lines' susceptances. Bounds on the power output of generator i are P_i^{max} and P_i^{min}, and P_{ij}^{max} defines the flow limit of the line between buses i and j. The objective function in (1a) minimizes the total cost of the generators' power outputs. Constraint (1b) is the DC approximation of the power flow equations [15]. Constraints (1c) and (1d) limit the generators' power outputs and the line flows. Constraint (1e) sets the reference angle, θ^{ref}.

III. Distributed Optimization Algorithms

This section overviews the distributed algorithms that we consider in this paper. These algorithms decompose the centralized DC OPF problem into subproblems corresponding to a partition of the original system. In the DC OPF formulation (1), tie-lines between regions couple the power flow equations (1b) and the line flow limits (1d). For each tie-line, we duplicate the boundary variables and assign a local copy to each region as shown in Fig. 1.

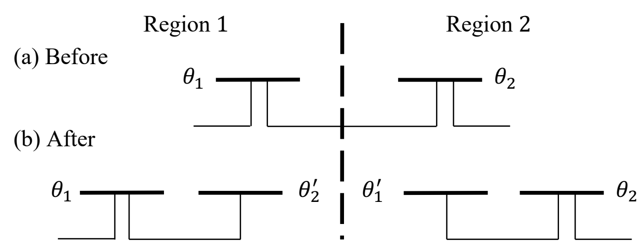

Fig. 1. Tie-line model before (a) and after (b) applying the decomposition.

To obtain a feasible solution to the centralized problem, we need to ensure consistency between the duplicated variables in each subproblem. To accomplish this, we use an Augmented Lagrangian method to enforce consistency using a penalized objective function:

$$L_\rho(\boldsymbol{p},\boldsymbol{\theta},\boldsymbol{\lambda}) := \sum_{i\in\mathcal{G}} f_i(p_i) + \sum_{i\in\mathcal{N}_s} \lambda_i(\theta_i - \theta_i') + \frac{\rho}{2}||\theta_i - \theta_i'||_2^2, \tag{2}$$

where $\boldsymbol{\lambda}$ are the Lagrange multipliers of the consistency constraints, $\boldsymbol{\theta}'$ are copies of the boundary variables, and ρ is a parameter. The set $\mathcal{N}_s$ contains all copies of the bus voltage angle variables corresponding to the tie-lines between regions.

The three distributed algorithms we consider are similar in their application of an Augmented Lagrangian to decompose the original problem. However, they use different processes for evaluating the objective function and updating the Lagrange multiplier λ. Generally, the three algorithms use the shared variables received from neighboring regions to update the Lagrange multipliers and evaluate the relaxed consistency constraints in the objective function. This makes the subproblems separable and independent from each other. The algorithms repeat these steps until all regions reach a consensus on the shared variables' values. We use the l_2-norm of the shared variables' mismatch, denoted by $||\Delta\theta||$, to measure the consensus on the shared variables. We next provide more detail for these distributed algorithms.

A. Alternating Direction Method of Multipliers (ADMM)

The ADMM is a well-known algorithm for solving large optimization problems [18], [19]. The ADMM algorithm solves the augmented Lagrangian problem in a distributed fashion that is similar to the Gauss-Siedel iterative method. Solving the OPF problem using ADMM in a distributed way involves a central coordinator to ensure achieving consensus among the local controllers. The local problem of region m at iteration $k+1$ is:

$$\min_{\boldsymbol{p}^{k+1},\boldsymbol{\theta}^{m,k+1}} \sum_{i\in\mathcal{G}^m} f_i(p_i^{k+1}) + \sum_{i\in\mathcal{N}_s^m} \lambda_i^k(\hat{\theta}_i^{c,k} - \theta_i^{m,k+1}) + \frac{\rho}{2}||(\hat{\theta}_i^{c,k} - \theta_i^{m,k+1})||_2^2 \tag{3}$$

subject to the DC OPF constraints (1b)–(1e),

where ρ is a tuning parameter. The decision variable $\boldsymbol{\theta}^m$ includes the local variables, i.e., the voltage angles for both the local and shared buses. The variable $\hat{\boldsymbol{\theta}}^c$ denotes the shared variables evaluated with values received from the central coordinator, and $\boldsymbol{\lambda}$ are the Lagrange multipliers.

After solving the subproblem (3), the local controllers share their solutions with the central coordinator. Then, the central coordinator solves an unconstrained optimization problem:

$$\min_{\boldsymbol{\theta}^{c,k+1}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{N}_s^m} \lambda_i(\theta_i^{c,k+1} - \hat{\theta}_i^{m,k+1}) + \frac{\rho}{2}||(\theta_i^{c,k+1} - \hat{\theta}_i^{m,k+1})||_2^2, \tag{4}$$

where $\boldsymbol{\theta}^c$ are the decision variables and $\hat{\boldsymbol{\theta}}^m$ are the shared variables received from the local controllers. The shared variable values obtained from the central coordinator's problem are then sent to the local controllers to update the Lagrange multipliers according to (5):

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\boldsymbol{\theta}^{c,k+1} - \boldsymbol{\theta}^{m,k+1}), \tag{5}$$

where ρ is the same parameter used in the local optimization problem (3). Note that $\boldsymbol{\theta}^m$ and $\boldsymbol{\theta}^c$ are the solutions of (3) and (4), respectively. The local controllers then use the updated Lagrange multipliers and the central coordinator's solution to update the local problems. This process is repeated until the shared variables agree to within a specified tolerance.

There are several variants of this ADMM implementation that have been proposed for solving OPF problems. These include extensions that eliminate the need for a central coordinator [20], [21]. A proximal massage passing (PMP) method proposed in [22] also permits a fully distributed ADMM implementation. For the calculations in this paper, we implemented ADMM using a modified version of PMP where local controllers share variable values with all other local controllers as this was easier to implement with the data structures that we used. However, calculations from each local controller are only dependent on variable values from their own regions and adjacent regions. This means that the extra shared information is unused and therefore, our implementation of ADMM is mathematically and structurally identical to PMP. The ADMM algorithm has been proven to converge to the optimal solution if the subproblems are convex. See [23] for more detail regarding ADMM implementations and convergence guarantees.

B. Analytical Target Cascading (ATC)

ATC is another distributed algorithm based on augmented Lagrangian relaxation that solves a large optimization problem via dividing it into hierarchically connected subproblems with multiple levels [24]. Two levels of subproblems are connected if they share coupling variables. To solve the OPF problem, we use a two-level ATC structure. The first level consists of a central coordinator. The regions' local subproblems are in the second level. The local subproblem m for iteration $k+1$ is:

$$\min_{\boldsymbol{p}^{k+1},\boldsymbol{\theta}^{m,k+1}} \sum_{i\in\mathcal{G}^m} f_i(p_i) + \sum_{i\in\mathcal{N}_s^m} \lambda_i(\hat{\theta}_i^{c,k} - \theta_i^{m,k+1}) + ||\beta(\hat{\theta}_i^{c,k} - \theta_i^{m,k+1})||_2^2 \tag{6}$$

subject to DC OPF constraints (1b)–(1e),

where $\boldsymbol{\lambda}$ are the Lagrange multipliers and β is a tuning parameter. Shared variables that are fixed to their values from the central coordinator are denoted as $\hat{\boldsymbol{\theta}}^c$. The local controllers communicate the resulting shared variable values with a central coordinator. The coordinator solves an unconstrained optimization problem that minimizes the differences between the boundary variables for neighboring regions:

$$\min_{\boldsymbol{\theta}^{c,k+1}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{N}_s^m} \lambda_i(\theta_i^{c,k+1} - \hat{\theta}_i^{m,k+1}) + ||\beta(\theta_i^{c,k+1} - \hat{\theta}_i^{m,k+1})||_2^2. \tag{7}$$

The coordinator shares the target results $\boldsymbol{\theta}^{c,k+1}$ with the local controllers. Next, the local controllers update the Lagrange multipliers and the parameters β using the target variables:

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + 2(\beta^k)^2(\boldsymbol{\theta}^{c,k+1} - \boldsymbol{\theta}^{m,k+1}), \tag{8a}$$
$$\beta^{k+1} = \alpha\beta^k. \tag{8b}$$

where β is the same parameter used in the local optimization problem (6). After updating the Lagrange multipliers the local controllers and the centralize coordinator repeated the process until the shared variables are within a specified tolerance.

Variants of ATC use different functions besides the Augmented Lagrangian to relax the consistency constraints [25]. Further, the ATC variant proposed in [7] is fully distributed, eliminating the need for a central coordinator. In this paper, we use the ATC implementation described above, but the same analysis is applicable for other ATC variants. The ATC algorithm is proven to converge for convex problems if the interaction is limited to subproblems in different levels [24].

C. Auxiliary Problem Principle (APP)

The APP algorithm is also based on augmented Lagrangian decomposition [26]. The APP algorithm solves a sequence of auxiliary problems in a distributed fashion without the need for a central coordinator. In contrast to ADMM and ATC which directly use the Augmented Lagrangian, APP linearizes the quadratic term in the augmented Lagrangian around the previous iteration and introduces a regularization term in the objective function [27]. Using the APP algorithm, the OPF formulation for region m and iteration $k+1$ is:

$$\min_{\boldsymbol{p}^{m,k+1},\boldsymbol{\theta}^{k+1}} \quad \sum_{i\in\mathcal{G}} f_i(p_i^{k+1}) + \sum_{i\in\mathcal{N}_s^m} \frac{\beta}{2}||\theta_i^{m,k+1} - \hat{\theta}_i^{n,k}||_2^2 + \gamma\theta_i^{m,k+1}(\hat{\theta}_i^{m,k} - \hat{\theta}_i^{n,k}) + \lambda\theta_i^{m,k+1} \tag{9}$$

subject to DC OPF constraints (1b)–(1e),

where α, β, and γ are tuning parameters. We define $\hat{\boldsymbol{\theta}}^{m,k}$ as the value of the shared variable obtained by the region m in the previous iteration, and $\hat{\boldsymbol{\theta}}^{n,k}$ denotes the values of the shared variables received from neighboring regions. After each region solves its associated problem and exchanges the results with the neighboring regions, each region updates the values of the Lagrange multipliers as follows:

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha(\boldsymbol{\theta}^{m,k+1} - \boldsymbol{\theta}^{n,k+1}). \tag{10}$$

To summarize, each region iteratively solves (9), shares the results with neighboring regions, and updates the Lagrange multiplier using (10) until reaching consensus on the shared variables. If the local objectives are convex and differentiable, selecting parameters satisfying the condition $\alpha < 2\gamma < \beta$ guarantees the convergence of the APP algorithm [28].

IV. Nonideal Communications Models

The communication requirements for a distributed algorithm depend on the shared variables. During each iteration, each region shares the results of its local optimization by communicating with the neighboring regions. Since communication networks are not ideal, the shared data may suffer from data quality issues or interruptions that impact the distributed algorithms' performance. In this section, we introduce three models for nonideal communications: noisy data, bad data, and intermittent loss of communication.

A. Noisy Communications

The data shared between connected regions may be subject to noise resulting from imperfect communication. This noise could be due to quantization error [29] or added to enforce data privacy requirements [30]. To model noisy shared data, we inject additive Gaussian noise into the shared variables:

$$\boldsymbol{\theta}_{noisy} = \boldsymbol{\theta}_{noiseless} + \boldsymbol{N}(0, \sigma_{noise}), \tag{11}$$

where $\boldsymbol{\theta}_{noisy}$ is the data that is actually communicated to the neighbors, $\boldsymbol{\theta}_{noiseless}$ is the true data, and $\boldsymbol{N}(0, \sigma_{noise})$ is a vector of normally distributed random numbers with zero mean and standard deviation of σ_{noise}.

B. Bad Data

Neighboring regions may occasionally receive "bad data", i.e., data with large errors. Bad data may be due to an instantaneous bit error [31] or a malicious adversarial agent [32]. We model a random injection of bad data at a specified occurrence probability as shown in the following model:

$$\boldsymbol{\theta}_{noisy} = \boldsymbol{\theta}_{noiseless} + 2\ M\ \boldsymbol{r}, \quad \text{where} \quad \boldsymbol{r} = \begin{cases} \boldsymbol{X}_1 - 0.5 & \text{if } \boldsymbol{X}_2 < p, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

In (12), M is the maximum magnitude of the error and $\boldsymbol{r}$ is a vector of multipliers that randomly selects the error magnitude. The vectors $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ contain uniformly distributed random numbers between $[0, 1]$, and p is the probability of bad data occurrence per iteration.

C. Intermittent Communication Loss

Communications between the agents may occasionally fail entirely for multiple iterations. For instance, communication transmission collisions or instability of the communication link can cause packet loss preventing the agents from sharing data for a number of iterations [14]. We consider a simple two-state model with success and fail states to represent the communication loss. If the communication channel is in a success state, then the data will be transmitted while the fail state means the data will be lost. The transition from one state to another occurs with a constant probability. This model is similar to Gilbert-Elliott Erasure channel used to model pocket loss [33]. Although communication loss can have more complicated occurrence behaviour, this model is used to model pocket loss due to its tractability and reasonable behaviour with respect to experimental data [34]. We define the *failure probability*, denoted λ_f, as the transition probability of the communication channel from success to fail states per iteration given that the channel was in success state during the previous iteration. Similar to the failure probability, we use the *repair probability*, denoted λ_r, to denote the transition probability from fail to success state. The *communication availability*, i.e., the probability that the system will be in a successful state at any iteration, is:

$$Availability\ (A) = \frac{\lambda_r}{\lambda_f + \lambda_r} \times 100\ [\%]. \tag{13}$$

To model intermittent communication loss, we introduce a state variable s, where $s = 1$ is a success state and $s = 0$ is a fail state. We also use an indicator function $\mathbf{1}\{x\}$, which equals 1 if x is true and 0 otherwise. We model the state of the communication channel by sampling a uniformly distributed random number at each iteration. If the controller detects an interruption from a neighboring region, it uses the value from the last successful data transmitted from this region. The following pseudocode describes our intermittent communication loss model:

Model for Intermittent Communication Loss

1: generate a random number X
2: **if** $s = 1$ **then** $s = 1 - \mathbf{1}\{\lambda_f \geq X\}$
3: **else** $s = \mathbf{1}\{\lambda_r \geq X\}$
4: **if** $s = 0$ **then** set $\theta_{shared}^{k+1} = \theta_{shared}^{k}$

V. Performance Analyses

This section presents the numerical results of solving the DC OPF problem using the three selected distributed algorithms. We first present the results of each algorithm with ideal communication. We then compare the performance of the distributed algorithms with the nonideal communication models in Section IV. We use three case studies: the 5-bus system WB5 from [35] and the IEEE 14- and 118-bus systems from MATPOWER [36], each decomposed into two regions. We present figures for the IEEE 118-bus system results that are representative of the results for the other cases.

A. Performance with Ideal Communications

We use the distributed algorithms assuming each region shares the exact results with neighboring regions at each iteration. We use the results of the ideal communication to tune the parameters and evaluate the convergence rates of the algorithms.

1) Alternating Direction Method of Multipliers (ADMM): The ADMM algorithm's performance depends on the value selected for the parameter ρ. Fig. 2 shows the convergence of the ADMM algorithm for the IEEE 118-bus system. For this test case, we observe that the subsystems reach consensus on the shared variables the fastest when the value of ρ is around 10^6. For WB5 and the IEEE 14-bus system, we tune the value of ρ to be 10^3 and $10^{4.6}$, respectively. Selecting smaller values for ρ increases the number of iterations to achieve consensus, while selecting significantly larger values cause oscillations that prevent the subsystems from reaching consensus.

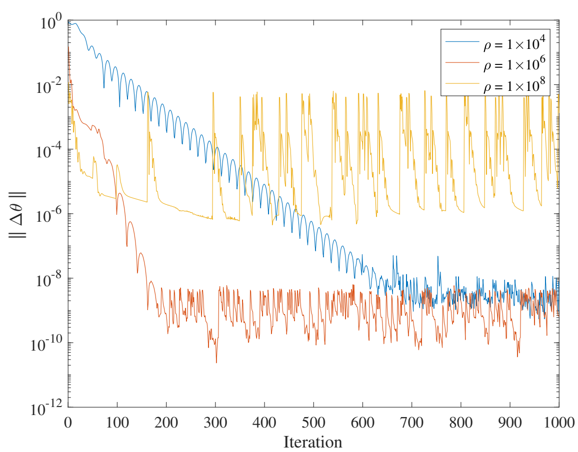

Fig. 2. ADMM convergence with different parameters for the IEEE 118-bus system with ideal communications.

2) Analytical Target Cascading (ATC): Similar to ADMM, the ATC algorithm requires tuning one parameter α. The convergence of the shared variables for the IEEE 118-bus system is shown in Fig. 3. We observe that setting the parameter $\alpha = 1.01$ increases the convergence time by a factor of five compared to $\alpha = 1.05$. We observe similar behaviour for the WB5 and IEEE 14-bus systems. Furthermore, the consensus on the shared variables diverges if the stopping criteria is not met. This behaviour occurs due to the update criteria (8b), which increases the penalty on the shared variable consistency term as the number of iterations increases.

3) Auxiliary Problem Principle (APP): Unlike ADMM and ATC, the APP algorithm contains three parameters that need to be tuned. We adopt the condition $\alpha = \gamma = \frac{1}{2}\beta$ from prior literature [28] in order to simplify the parameter tuning. For the IEEE 118-bus system with two regions, the APP algorithm converges when $\alpha = 10^4$ as shown in Fig. 4. The

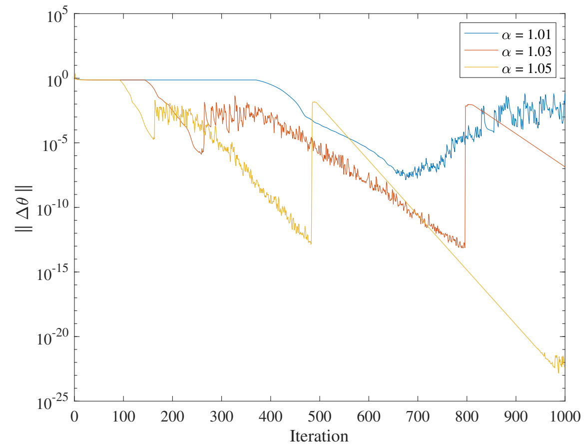

Fig. 3. ATC convergence with different parameters for the IEEE 118-bus system with ideal communications.

algorithm does not converge with the value of $\alpha = 10^3$ after 1000 iterations. For the WB5 and IEEE 14-bus systems, we find a value of α equal to 10^2 and 10^4, respectively, yield a fast convergence rate. While selecting larger parameter values improves the convergence rate, this also degrades the mismatch error in the shared variables achieved by the algorithm.

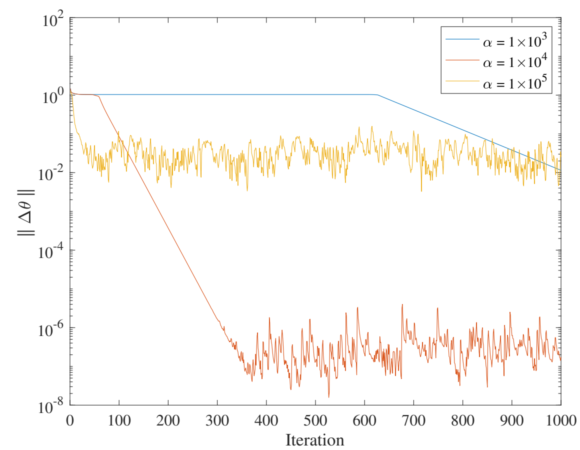

Fig. 4. APP convergence with different parameters for the IEEE 118-bus system with ideal communications.

4) Remark about the Algorithms' Performance: Along with the convergence rate, the quality of the final solution is another metric for selecting parameters. We use the *Relative Gap* (RG), defined as relative difference between the cost of the distributed and centralized solutions, to evaluate the solution quality. We also use the *Power Imbalance* (PI), defined as the relative difference between the total power generation and the total load demand, to capture violations of the power balance constraints. In the lossless DC power flow approximation, the value of PI should be zero upon convergence.

Table I summarizes the results obtained from the three algorithms in terms of the convergence rate and solution quality after reaching consensus on the shared variables, with

TABLE I
CONVERGENCE RESULTS WITH IDEAL COMMUNICATION

Case	Measure	ADMM	ATC	APP
WB5	**Time (s)**	2.62	0.79	0.36
	Iterations	29	54	30
	RG (%)	1.1×10^{-3}	5.1×10^{-3}	5.1×10^{-3}
	PI (%)	-1.1×10^{-3}	5.1×10^{-3}	4.4×10^{-3}
14-Bus	**Time (s)**	4.67	1.42	0.74
	Iterations	41	119	79
	RG (%)	-3.0×10^{-3}	9.3×10^{-3}	-5.0×10^{-2}
	PI (%)	3.0×10^{-3}	7.0×10^{-3}	-3.8×10^{-2}
118-Bus	**Time (s)**	7.09	3.02	4.85
	Iterations	59	149	214
	RG (%)	-3.0×10^{-3}	8.1×10^{-4}	8.9×10^{-3}
	PI (%)	3.0×10^{-3}	-6.8×10^{-4}	6.7×10^{-3}

tolerance equal to 1×10^{-4} radians (5.7×10^{-3} degrees). We observe that parameter selection strongly impact the convergence rate of the algorithms. Comparing the results of the three algorithms with the tuning parameters that we select, APP has the faster convergence rate for the WB5 and 14-bus cases, while ATC has the fastest convergence rate for the 118-bus case. Furthermore, APP has the largest RG compared to the other two algorithms, which is especially noticeable for the IEEE 14-bus test case.

B. Performance with Noisy Communications

We assume the regions send shared variable information with additive Gaussian noise as described in (11). We vary the standard deviation of the noise σ_{noise} to compare the performance of the algorithms under different noise levels. The consensus on the voltage angles achieved by the three algorithms for the IEEE 118-bus system with standard deviation $\sigma = 1 \times 10^{-3}$ is shown in Fig. 5. The results indicate that small noise levels do not significantly impact the convergence rate of the algorithms. Further, the three distributed algorithms converge to a similar level of accuracy as the level of the injected noise. However, the ATC algorithm's convergence pattern exhibits a sudden ramp on the norm of the mismatch during the iterations.

The mean and standard deviation of the shared variables' mismatches in the final solution, $\mu_{||\Delta\theta||}$ and $\sigma_{||\Delta\theta||}$, for the three algorithms with three noise levels are shown in Table II. To visualize the performance differences, Fig. 6 shows the mean of the shared variable mismatches, $\mu_{||\Delta\theta||}$, in the final solutions for the IEEE 118-bus system as the noise standard deviation, σ_{noise}, varies from 10^{-5} to 10^{-3}. The three algorithms achieve consensus on the shared variables with an error that increases approximately linearly with the standard

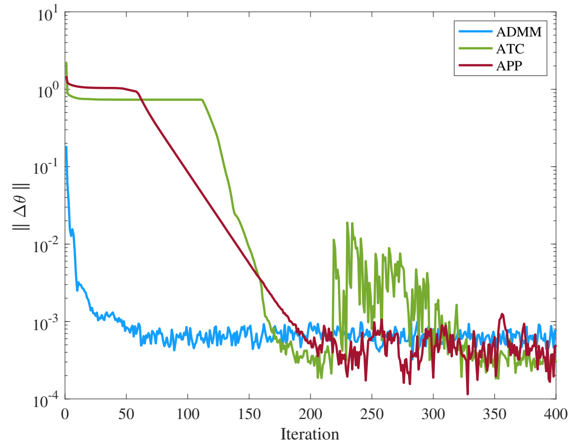

Fig. 5. The three algorithms' convergence for the IEEE 118-bus system with noisy data ($\sigma = 1 \times 10^{-3}$).

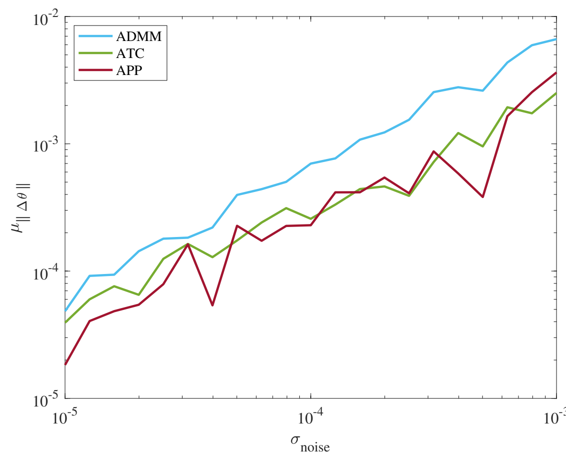

Fig. 6. Mean of the mismatch in the final solution for the IEEE 118-bus system with different noise levels.

deviation of the added noise, with a slightly lower mismatch observed for APP and ATC compared to ADMM.

Note that the solution quality for both ADMM and APP is not significantly impacted by the noise, while the ATC algorithm's final solution can have a high relative gap from the optimal solution if consensus is not achieved, i.e., the stopping criteria are not met after many iterations. This happens due to the parameter update for the ATC algorithm (8b), which increases the penalty on the consistency term in the objective as the number of iterations increase. This degrades the solution quality if the regions do not reach consensus.

TABLE II
MEAN ($\mu_{solution}$) AND STANDARD DEVIATION ($\sigma_{solution}$) OF THE MISMATCH IN THE FINAL SOLUTION WITH DIFFERENT VALUES OF NOISE (σ_{noise})

Algorithm		ADMM			ATC			APP		
Case	σ_{noise}	10^{-5}	10^{-4}	10^{-3}	10^{-5}	10^{-4}	10^{-3}	10^{-5}	10^{-4}	10^{-3}
WB5	$\mu_{\|\|\Delta\theta\|\|}$	4.4×10^{-5}	4.3×10^{-4}	4.6×10^{-3}	2.3×10^{-5}	2.5×10^{-4}	2.7×10^{-3}	4.1×10^{-5}	3.4×10^{-4}	3.3×10^{-3}
	$\sigma_{\|\|\Delta\theta\|\|}$	1.1×10^{-5}	1.1×10^{-4}	1.2×10^{-3}	7.6×10^{-6}	6.1×10^{-5}	5.4×10^{-4}	1.8×10^{-5}	1.1×10^{-4}	5.9×10^{-4}
14-Bus	$\mu_{\|\|\Delta\theta\|\|}$	4.8×10^{-5}	4.9×10^{-4}	4.9×10^{-3}	3.1×10^{-5}	2.7×10^{-4}	2.3×10^{-3}	4.6×10^{-5}	4.9×10^{-4}	4.2×10^{-3}
	$\sigma_{\|\|\Delta\theta\|\|}$	1.2×10^{-5}	1.2×10^{-4}	1.1×10^{-3}	7.6×10^{-6}	6.5×10^{-5}	5.2×10^{-4}	9.7×10^{-6}	1.4×10^{-4}	7.8×10^{-4}
118-Bus	$\mu_{\|\|\Delta\theta\|\|}$	6.5×10^{-5}	6.2×10^{-4}	6.5×10^{-3}	5.7×10^{-5}	3.6×10^{-4}	3.3×10^{-3}	6.4×10^{-5}	6.8×10^{-4}	4.8×10^{-3}
	$\sigma_{\|\|\Delta\theta\|\|}$	1.2×10^{-5}	1.2×10^{-4}	1.1×10^{-3}	1.2×10^{-6}	5.6×10^{-5}	5.9×10^{-4}	1.2×10^{-5}	1.8×10^{-4}	1.7×10^{-3}

C. Performance with Bad Data

For the second type of noise, we inject bad data into the shared variables as described in (12). We set the value for the bad data magnitude $M = 2\pi$ radians and vary the probability of the injected errors. Fig. 7 shows the mismatches of the shared variables with probabilities of bad data occurrence $p = 0.1\%$ for the IEEE 118-bus system.

We observe that the mismatches return to the same convergence pattern after a large error is injected in all three algorithms. However, we observe that even when consensus is achieved on the shared variables for ADMM and ATC, bad data can cause a large relative gap in the resulting solution.

To quantitatively compare the algorithms' performance, we estimate the probability of achieving the optimal solution using 100 runs of the algorithm for varying probabilities of bad data occurrence. We consider an algorithm to have achieved the optimal solution if both RG and PI are below 1% within a maximum of 1000 iterations. Table III summarizes these results. The three algorithms may fail to attain the optimal solution even with bad data probabilities as low as 0.1%. Further, Fig. 8 shows the success rates when varying the bad data probability from 0.1% to 1% for the IEEE 118-bus system. Generally, the convergence success rate decreases approximately linearly as the probability of bad data increases.

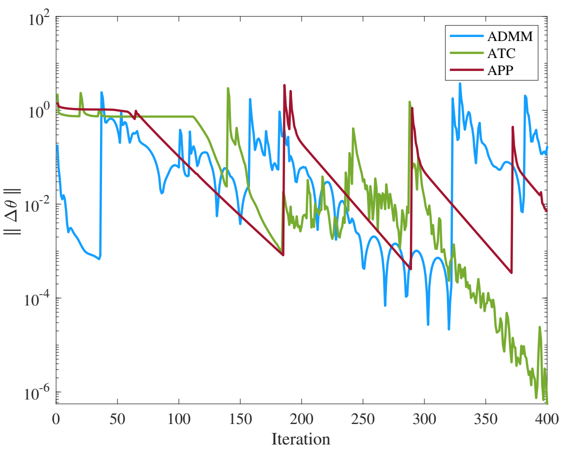

Fig. 7. The three algorithms convergence for the IEEE 118-bus system with bad data.

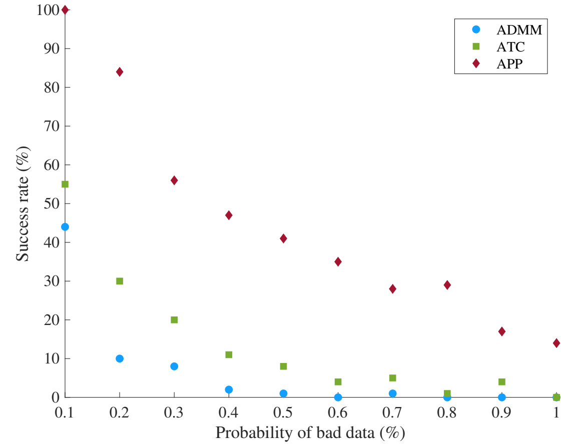

Fig. 8. Success rates for the IEEE 118-bus system with different rates of bad data injection.

D. Performance with Intermittent Communication Loss

This section compares the distributed algorithms' performance under the intermittent communication loss model in Section IV-C. Fig. 9 shows the convergence characteristics for the shared variable mismatches for the three algorithms with failure probability $\lambda_f = 5\%$ and repair probability $\lambda_r = 10\%$.

The results indicate that the APP algorithm fails to achieve consensus for the IEEE 118-bus case for failure probability above 1% per iteration or availabilities below 50%. The ADMM and ATC algorithms, conversely, achieve consensus on the values of the shared variables and the mismatch almost always decreases as the number of iterations increases for all cases. However, the final solutions can be far from optimal. To qualitatively compare algorithmic performance during intermittent communication loss, Table IV describes how the probability of achieving the optimal solution changes with failure probability equal to 1 and 5% per iteration. Each of the results in this table are computed from 100 runs of the algorithm with a constant repair rate of 10% per iteration. We again consider an algorithm to have achieved the optimal solution if both RG and PI are below 1% within a maximum of 1000 iterations.

Focusing on the IEEE 118-bus case, Fig. 10 compares the algorithms' performance when subjected to intermittent communication loss by varying the failure probability from 0.5

TABLE III
PERFORMANCE OF THE DISTRIBUTED ALGORITHMS WITH BAD DATA

Algorithm		**ADMM**		**ATC**		**APP**	
Probability of bad data (%)		**0.1**	**1.0**	**0.1**	**1.0**	**0.1**	**1.0**
WB5	**Success rate (%)**	88	30	100	97	100	100
	Avg. Iterations	31	53	61	78	21	28
	Avg. RG (%)	8.7×10^{-4}	-1.4×10^{-7}	4.2×10^{-3}	3.0×10^{-3}	-5.1×10^{-3}	-2.3×10^{-3}
	Avg. PI (%)	-8.7×10^{-4}	2.3×10^{-7}	-3.9×10^{-3}	-1.6×10^{-3}	-3.9×10^{-4}	-1.8×10^{-4}
14-Bus	**Success rate (%)**	68	15	98	42	42	10
	Avg. Iterations	42	370	145	164	991	985
	Avg. RG (%)	-4.1×10^{-3}	-4.6×10^{-2}	1.5×10^{-2}	5.5×10^{-2}	-2.1×10^{-1}	9.2×10^{-2}
	Avg. PI (%)	3.9×10^{-3}	1.2×10^{-2}	-9.6×10^{-3}	-5.8×10^{-4}	-1.7×10^{-1}	6.3×10^{-2}
118-Bus	**Success rate (%)**	44	0	55	0	100	14
	Avg. Iteratiosn	61	NC	193	NC	404	1000
	Avg. RG (%)	1.0×10^{-2}	NC	1.68×10^{-2}	NC	1.0×10^{-1}	2.2×10^{-1}
	Avg. PI (%)	3.0×10^{-3}	NC	-2.2×10^{-3}	NC	-6.6×10^{-3}	8.8×10^{-2}

NC: Not converged.

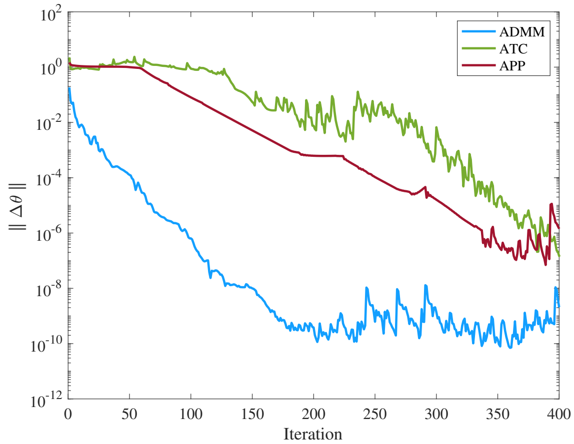

Fig. 9. The three algorithms convergence for the IEEE 118-bus system with intermittent communication loss.

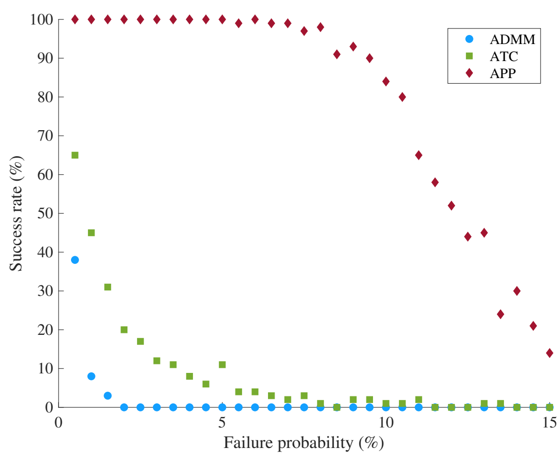

Fig. 10. Success rates for the IEEE 118-bus system with different communication failure probability. Communication repair probability are held constant at 10%.

to 15%. The APP algorithm starts to fail to converge when the communication failure probability is above 8% (corresponding to an availability below 55%). While the ADMM and ATC algorithms fail to obtain the optimal solution around 50% of the time when the failure probability is 0.5% (corresponding to an availability of 95%), they both reach consensus on the shared variables. Overall, the results suggest that the APP algorithm outperforms the other two algorithms with the intermittent communication loss model for the three case studies.

E. Discussion and Comparison

Parameter tuning plays a major role in the performance of distributed algorithms as it strongly impacts the convergence rate. The parameters in the selected algorithms are associated with the penalty terms for the relaxed consistency constraints. We observe that all three algorithms converge for a certain range of parameter values. Generally speaking, selecting large parameter values will prevent the algorithm from achieving the optimal solution and, in some cases, large values might cause the algorithm to diverge. On the other hand, small values reduce the convergence speed and, in extreme cases, the algorithm can diverge. Furthermore, the impacts of the parameter selection differ among the three algorithms. For both ADMM and APP, the algorithms converge toward the optimal solution if consensus on the shared variables is achieved. On the other hand, the ATC algorithm can have a relative gap that increases even if the mismatches for the values of the shared variables are decreasing. We also observe that different systems and cost functions might require repeating the parameter tuning step.

The results shown in this paper indicate the importance of data integrity on the performance of the distributed algorithms. We observe various responses from the distributed algorithms to the error models. With Gaussian communication noise, all three algorithms converge to comparable accuracy of the standard deviation of noise. Although ATC is least impacted by Gaussian noise, this algorithm converges to the wrong values if the stopping criterion is not met. This suggests that reliable performance of the ATC algorithm in the presence of Gaussian communication noise requires either using another stopping criterion or modifying the update step in the algorithm.

Among the three algorithms considered in this paper, the APP algorithm has the best performance with respect to bad data and intermittent communication loss in terms of the final solution quality. Further, the ADMM algorithm has the lowest success rate compared to the other two algorithms with respect to these two noise types. Although the ADMM and ATC algorithms have a more stable convergence pattern for these two noise types, they might converge to a suboptimal solution.

TABLE IV
PERFORMANCE OF THE DISTRIBUTED ALGORITHMS WITH INTERMITTENT COMMUNICATION LOSS

Algorithm		**ADMM**		**ATC**		**APP**	
Failure probability (%)		**1**	**5**	**1**	**5**	**1**	**5**
Availability (%)		**91**	**67**	**91**	**67**	**91**	**67**
WB5	**Success rate (%)**	97	70	59	100	100	100
	Avg. Iterations	43	81	66	94	29	98
	Avg. RG (%)	1.0×10^{-2}	2.2×10^{-5}	3.9×10^{-3}	4.0×10^{-3}	-2.7×10^{-3}	4.4×10^{-4}
	Avg. PI(%)	-7.5×10^{-4}	-2.1×10^{-5}	-5.5×10^{-4}	9.5×10^{-4}	-8.3×10^{-4}	1.4×10^{-4}
14-Bus	**Success rate (%)**	61	8	100	99	100	100
	Avg. Iterations	50	68	154	185	997	1000
	Avg. RG (%)	1.1×10^{-2}	7.9×10^{-2}	2.2×10^{-2}	4.4×10^{-3}	-1.9×10^{-1}	-1.5×10^{-1}
	Avg. PI (%)	3.5×10^{-3}	-8.0×10^{-3}	-6.4×10^{-3}	-3.6×10^{-3}	-1.3×10^{-1}	-1.1×10^{-1}
118-Bus	**Success rate (%)**	8	0	45	11	100	100
	Avg. Iterations	56	NC	193	206	248	435
	Avg. RG (%)	4.7×10^{-1}	NC	-1.3×10^{-3}	-1.2×10^{-2}	1.1×10^{-1}	-1.3×10^{-3}
	Avg. PI (%)	1.5×10^{-3}	NC	-3.1×10^{-3}	-1.1×10^{-2}	1.1×10^{-1}	3.7×10^{-4}

NC: Not converged.

VI. Conclusion and Future Work

Distributed algorithms have many attractive features for solving power system optimization problems, especially for systems with many independent microgrids. Distributed algorithms allow interconnected systems to cooperatively solve large optimization problems while maintaining their autonomy. However, the performance of a distributed algorithm strongly depends on the quality of the shared data. In this paper, we show that different distributed algorithms have various responses to data quality issues. The ADMM and APP algorithms have superior convergence patterns with respect to communication noise compared to ATC. After we stop injecting noise, both the ADMM and APP algorithms return to the same convergence pattern and achieve the optimal solution. In contrast, solutions obtained from the ATC algorithm might be suboptimal even when the neighboring regions achieve consensus on the values of the shared variables.

As extensions to this work, there are other communication and data integrity issues requiring detailed investigations. For power systems applications, these include asynchronous data sharing between neighboring regions and communication latency. Another direction for future work is studying how different power flow representations affect the convergence rates for problems where the DC approximation is inapplicable.

References

[1] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[2] B. H. Kim and R. Baldick, "A comparison of distributed optimal power flow algorithms," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 599–604, 2000.

[3] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, and R. Baldick, "Toward distributed/decentralized DC optimal power flow implementation in future electric power systems," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2574–2594, 2016.

[4] I. Murzakhanov, A. Malakhov, and E. Gryazina, "Suboptimality of decentralized methods for OPF," in *IEEE Milan PowerTech*, 2019.

[5] J. Mohammadi, G. Hug, and S. Kar, "Role of communication on the convergence rate of fully distributed DC optimal power flow," in *IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, 2014, pp. 43–48.

[6] J. Guo, G. Hug, and O. K. Tonguz, "On the role of communications plane in distributed optimization of power systems," *IEEE Trans. Industr. Inform.*, vol. 14, no. 7, pp. 2903–2913, 2018.

[7] A. Kargarian, M. Mehrtash, and B. Falahati, "Decentralized implementation of unit commitment with analytical target cascading: A parallel approach," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 3981–3993, 2018.

[8] H. Li, C. Huang, G. Chen, X. Liao, and T. Huang, "Distributed consensus optimization in multiagent networks with time-varying directed topologies and quantized communication," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 2044–2057, 2017.

[9] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *47th IEEE Conf. Decis. Control (CDC)*, 2008, pp. 4177–4184.

[10] D. Yuan, S. Xu, H. Zhao, and L. Rong, "Distributed dual averaging method for multi-agent optimization with quantized communication," *Syst. Control Lett.*, vol. 61, no. 11, pp. 1053–1061, 2012.

[11] L. Majzoobi, F. Lahouti, and V. Shah-Mansouri, "Analysis of distributed ADMM algorithm for consensus optimization in presence of node error," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1774–1784, 2019.

[12] C.-X. Shi and G.-H. Yang, "Distributed optimization under unbalanced digraphs with node errors: Robustness of surplus-based dual averaging algorithm," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 1, pp. 331–341, 2021.

[13] H. Li, B. Jin, and W. Yan, "Distributed model predictive control for linear systems under communication noise: Algorithm, theory and implementation," *Automatica*, vol. 125, p. 109422, 2021.

[14] S. Chen, W. Lan, J. Ma, and X. Yu, "Distributed optimization design of multi-agent systems with packet losses," in *IEEE 16th Int. Conf. Control Autom.*, 2020, pp. 1241–1246.

[15] B. Stott, J. Jardim, and O. Alsac, "DC power flow revisited," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1290–1300, 2009.

[16] D. K. Molzahn and I. A. Hiskens, "A Survey of Relaxations and Approximations of the Power Flow Equations," *Found. Trends Electric Energy Syst.*, vol. 4, no. 1-2, pp. 1–221, February 2019.

[17] M. Alkhraijah, C. Menendez, and D. K. Molzahn, "Evaluating the performance of distributed optimal power flow algorithms with nonideal communication," *arXiv preprint arXiv:2106.00135*, 2021.

[18] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Math. Appl.*, vol. 2, no. 1, pp. 17–40, 1976.

[19] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, vol. 9, no. R2, pp. 41–76, 1975.

[20] A. X. Sun, D. T. Phan, and S. Ghosh, "Fully decentralized AC optimal power flow algorithms," in *IEEE Power & Energy Society General Meeting*, 2013, pp. 1–5.

[21] T. Erseghe, "Distributed optimal power flow using ADMM," *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2370–2380, 2014.

[22] M. Kraning, E. Chu, J. Lavaei, and S. Boyd, "Dynamic network energy management via proximal message passing," *Found. Trends Optimiz.*, vol. 1, no. 2, pp. 70–122, 2013.

[23] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[24] N. Michelena, H. Park, and P. Y. Papalambros, "Convergence properties of analytical target cascading," *AIAA J.*, vol. 41, no. 5, pp. 897–905, 2003.

[25] S. Tosserams, L. Etman, P. Papalambros, and J. Rooda, "An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers," *Struct. Multidiscip. Optimiz.*, vol. 31, no. 3, pp. 176–189, 2006.

[26] G. Cohen, "Auxiliary problem principle and decomposition of optimization problems," *J. Optimiz. Theory App.*, vol. 32, no. 3, pp. 277–305, 1980.

[27] L. Zhao, D. Zhu, and B. Jiang, "Auxiliary problem principle of augmented Lagrangian with varying core functions for large-scale structured convex problems," *arXiv:1512.04175*, 2015.

[28] B. H. Kim and R. Baldick, "Coarse-grained distributed optimal power flow," *IEEE Trans. Power Syst.*, vol. 12, no. 2, pp. 932–939, 1997.

[29] D. Marco and D. L. Neuhoff, "The validity of the additive noise model for uniform scalar quantizers," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1739–1755, 2005.

[30] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 172–187, 2017.

[31] M. Jeruchim, "Techniques for estimating the bit error rate in the simulation of digital communication systems," *IEEE J. Sel. Areas Commun.*, vol. 2, no. 1, pp. 153–170, 1984.

[32] F. Pasqualetti, R. Carli, and F. Bullo, "A distributed method for state estimation and false data detection in power networks," in *IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, 2011, pp. 469–474.

[33] G. Haßlinger and O. Hohlfeld, "The gilbert-elliott model for packet loss in real time services on the internet," in *14th GI/ITG Conference-Measurement, Modelling and Evalutation of Computer and Communication Systems*. VDE, 2008, pp. 1–15.

[34] E. Martinian and C. Sundberg, "Decreasing distortion using low delay codes for bursty packet loss channels," *IEEE Trans. Multimedia*, vol. 5, no. 3, pp. 285–292, 2003.

[35] W. A. Bukhsh, A. Grothey, K. I. M. McKinnon, and P. A. Trodden, "Local solutions of the optimal power flow problem," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4780–4788, 2013.

[36] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, 2011.