DISTRIBUTED ONLINE ALGORITHMS FOR ENERGY MANAGEMENT IN SMART GRIDS

By

Shailesh Wasti

Vahid R. Disfani
Assistant Professor of Electrical Engineering
(Chair)

Abdelrahman A. Karrar
Associate Professor of Electrical Engineering
(Committee Member)

Abdul R. Ofoli
UC Foundation Associate Professor of Electrical Engineering
(Committee Member)

Raga Ahmed
Associate Professor of Electrical Engineering
(Committee Member)

DISTRIBUTED ONLINE ALGORITHMS FOR ENERGY MANAGEMENT IN SMART GRIDS

By

Shailesh Wasti

A Thesis Submitted to the Faculty of the University of
Tennessee at Chattanooga in Partial
Fulfillment of the Requirements of the Degree
of Master of Science: Engineering

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

December 2020

ABSTRACT

The 21st-century electric power grid is transitioning from a centralized structure designed for bulk-power transfer to a distributed paradigm that integrates the variable renewable energy (VRE) resources spatially distributed across the grid. This work proposes algorithmic solutions for distributed economic dispatch based on Subgradient method and Alternating Direction Method of Multipliers (ADMM), both designed to be agnostic with any initialization vector. The proposed distributed online solutions leverage a dynamic average consensus algorithm to track the time-variant linearly coupled constraint that allows an abrupt change in power demand of the network because of the high penetration of VRE resources. The problems are modeled as discrete dynamic systems to investigate the stability and convergence of the algorithm. The update procedures are designed such that the iterates converge to the optimal solution of the original optimization problem, steered by the gain parameter corresponding to the second largest eigenvalue of the system matrix.

DEDICATION

In the loving memory of my grandmother...and the ‘lost Einsteins’ of my generation.

ACKNOWLEDGEMENTS

I am grateful to my advisor Dr. Vahid Disfani for his extraordinary mentorship throughout my graduate study. Before I met you, I had a naïve understanding of research and was not sure what I really wanted to pursue for the rest of my career. The lectures on Power System Optimization during my first semester truly awakened my research interests. I would have never known what I wanted to study without those incredible lectures covering topics from the primal-dual algorithm to consensus theory. I am always amazed how quickly you understand my questions, be it coding related or conceptual, and come up with elegant solutions. I thank you for patiently listening to all my questions, sometimes for hours. I am beyond words to describe how great your mentorship has been and how much I cherish it. I will always be indebted to you for everything you have done for me.

I am very thankful to all my committee members: Dr. Abdelrahman A. Karrar, Dr. Abdul R. Ofoli, and Dr. Raga Ahmed for being kind to spare time for my work. Dr. Karrar, you truly have amazed me with your teaching and inspired me with your passion for the power system, not to mention that your door has always been open to students. Dr. Ofoli, I was fortunate to be in your class on Fuzzy Logic and Intelligent Control that has instilled in me the enthusiasm for control theory. Dr. Ahmed, it cannot be stressed enough how much you care about each of your students. Thank you Dr. Ahmed H. Eltom for your great leadership and constant encouragement.

I want to thank Pablo, my friend and collaborator, for enthusiastically helping me throughout my thesis. Thank you for the early morning coffee and for teaching me some invaluable life lessons. I want to thank Saroj, my friend since college, for his support throughout and want to especially acknowledge him for providing me the template of LaTeX. It would have taken days if I had to make one from scratch. Thank you to all my friends from ConnectSmart Lab: Ahmed, Farog, James, and Shahab. Each one of you is special to me. Thank you to my friends in Cherryton Christopher and Jackson. Thank you Sambuddha, Bikash, Kamal, Sudip, Kanchan, Mahim, Rohan, Hafiz, and all my dear friends for always cheering me up.

Last but not the least, I want to thank my family for being with me during the highs and the lows. I could have never dreamt big without the sacrifice of my parents. Thank you to Bishal and Shailaja for their kindness, Satish and Samiksha for growing up with me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 General Background

The steady growth of distributed energy resources (DER), due to the society's increasing commitment to a low-carbon economy along with the remarkable progress in technology, is questioning the relative benefit of large-scale electricity generation which is supposed to stem from high economies of scale. The disruption at the grid edge of the physical layers of the power system–generation, transmission, and distribution–has profoundly transformed the 21st-century power systems, and the time has come to revisit the classic categorization of these physical layers. The explosion of data with the proliferation of DER has allowed researchers to look for new ways to control, monitor, and optimize resources. The dispatch of generators has become challenging because of the number of players in the electricity markets and their hesitation to share their own control variables and economic data due to privacy and economic reasons.

Economic Dispatch (ED) is a typical resource allocation problem in power system, where each generator finds its optimum strategy in order to ensure power balance in the network [2]. In traditional ED problem, generators share their control variables to global control center, which then implements the centralized dispatch algorithms and sends back the information to all the generating units. However, microgrids and DERs owned and managed by private sectors urge the ED problem to be solve in different fashion.

Distributed Energy Resources (DER) are the resources spatially distributed in the physical structure of the grid, and often tend to be smaller in size.

Smart Grid is often used to contrast the traditional electric grid. Smart Grid is basically defined as the grid with computational intelligence that uses two-way communication [3].

Distributed Economic Dispatch is an optimization problem where each agent finds their optimal strategy based on their own and that of its directly connected neighbors' information. Figure 1.1 shows the structure of centralized and distributed structure of ED in power grids.

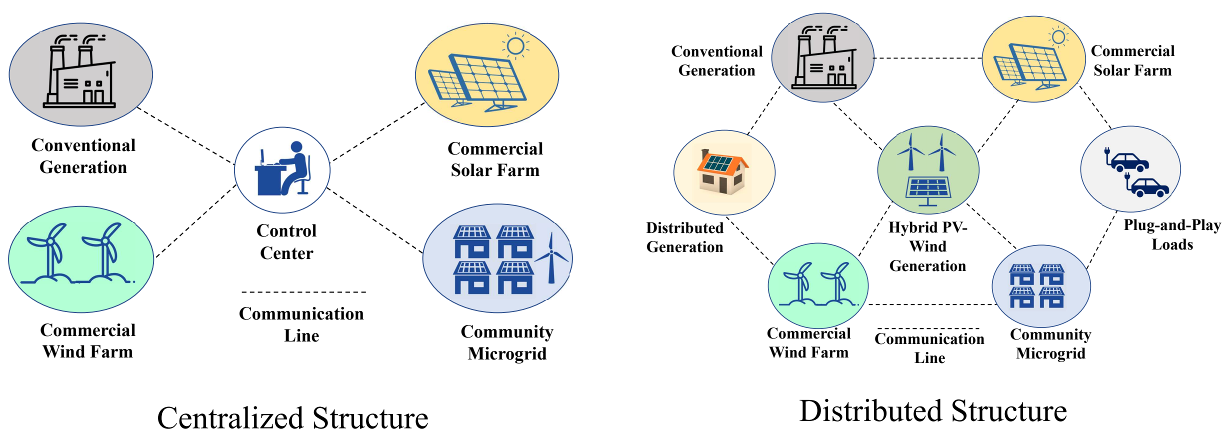

Figure 1.1 Different approaches of economic dispatch in power grids

Agents are the players of the market in a typical optimization problem. Each bus or node is an agent in the formulation. It can be the bus with only demand, or generating unit, or both.

Online Algorithms are designed such that they can track the real-time changes and allocate resource. They are sometimes described by real-time algorithms.

Consensus based gradient algorithms are often exploited to realize the distributed version of economic dispatch. The algorithms, however, are designed for constant load demand, which do not capture the change in the demand pronounced by the penetration of variable energy resources in the grid.

1.2 Statement of Problems

The high penetration of DERs in the grid has raised some serious concerns on the overall operation of the power systems. This work investigates the current issues with the alternatives proposed to optimally dispatch the generating units. I discuss each one of them below:

- **Distributed Algorithms for Time-Variant Load Demand**: In recent years, popular algorithms based on (sub)gradient method and Alternating Direction Method of Multipliers (ADMM) have been extensively employed to provide distributed solutions to energy management system, and in particular ED and optimal power flow (OPF) problems. Both of the algorithms can solve distributed ED problem in presence of a coordinator (also referred to as a leader or a master node) which addresses power mismatch in the network. Several solutions have been proposed in recent years to remove the coordinator [4–7], but none of the works addresses the fluctuations of demand in real-time without some adjustments.

- **Comprehensive Study on the Convergence Speed**: Speed of algorithm is governed by the gain parameter; gain is often called gradient step in (sub)gradient algorithm, and penalty parameter in ADMM algorithm. While the convergence of algorithm is proven theoretically exploiting different approaches such as Lyapunov function, there is a research gap in providing comprehensive study on the role of gain parameter in the speed of distributed algorithm.

- **Privacy Preserving Algorithm**: Confidentiality of data is the basic tenet of distributed algorithms. Sharing of decision variables and cost parameters of the optimization problem would question the implementation of fair electricity market. Most of the distributed approaches in the literature are based on the sharing of decision variables. Authors in [8] introduce power mismatch for data privacy as a sharing variable, but is designed with the strict assumption that power balance has already been ensured in the network in the first place.

1.3 Objectives

In this thesis, the algorithmic solutions to distribute ED problem are investigated, where the total demand of the network fluctuates in granularity of seconds because of the high penetration of Variable Renewable Energy (VRE) sources. ED problem is a constrained optimization problem with an equality constraint that linearly couples the decision variables of all agents. While ED problem may not necessarily be convex, the study is limited to convex problem with smooth cost functions. ED is a resource allocation problem in which all agents cooperate to collectively minimize a global objective function, which is a sum of all local objective functions. The algorithms are designed to attain the same objective in distributed fashion without the need of a master node or a coordinator. The design is distinct in the sense that algorithms proposed are agnostic to any initialization vector, and can track the time-variant constraint violation in real-time, thus called online algorithms. Besides, the proposed design does not discriminate between the type of agents (bus with or without generating unit).

This work leverages average consensus theory and its dynamic aspect in that agents communicate through the underlying communication network in distributed fashion. The network is distributed if the agents can communicate only with their directly connected neighbors. Even though consensus problems can handle communication imperfections such as network splitting, time delays [9, 10], the study here is restricted to perfect communication. The contribution of this thesis are as follows:

- **Distributed Online ED for *Any* Convex Problem**: First, a consensus based algorithm for ED problem is proposed in chapter 3. In a departure from the existing literature where the all the decision variable are coupled to meet certain time-invariant constraint set, the proposed online algorithm finds consensus on the time-varying estimate of the average power-mismatch in a purely distributed setting via dynamic average consensus algorithm detailed in chapter 2, thus ensuring power-balance in the network. The equivalent dual problem of the Lagrange relaxed problem is first formulated, and solved using consensus theory and Sub-gradient method. It is shown that the iterates provably go to optimal points using the famous Karush–Kuhn–Tucker conditions. As the algorithm tracks the real-time changes in demand using dynamic average consensus algorithm, it is agnostic to any initialization vector. In other words, optimization can start from any random points without any apology.

  Second, the distributed online solution, in chapter 4, is extended to modify ADMM algorithm to address the performance requirement. ADMM algorithms are widely known for its ability to handle any convex problem without limiting to differentiable and strictly convex problem. The ADMM is different from a regular Lagrange relaxation in that it has a penalty term added in the relaxed version which vanishes to zero at primal feasibility. This brings robustness in the problem. From the optimization point of view, the augmented penalty term makes the dual problem differentiable under mild conditions. In other words, it removes the assumption of the strictly convex problem to have a equivalent differentiable dual problem. While I simulate the case for quadratic cost curve based on the law of diminishing marginal returns [11], the future grid looks for algorithm that can only handle real-time changes in cost curve but also may not be differentiable. Chapter 4 proposes a novel distributed online ADMM algorithm to solve the ED problem, where the optimization problem is fully decomposed between participating agents and solved online without any need to initialize the solution process. The primal and dual feasibility of distributed-ADMM problem are proved, and shown that the primal and dual iterates provably go to the optimal points. In addition, the algorithm ensures privacy of data as agents communicate the estimate of the average power mismatch to their neighbors, which quickly goes to zero, instead of the generated power and the demand of the node. Privacy preserving is fundamental for fair electricity trading in the market.

- **Convergence Speed using Modal Analysis**: The algorithms are modeled as discrete dynamic systems to investigate the stability and convergence of the algorithm detailed in chapter 3 and chapter 4. The optimum gain parameters are calculated based on the study of the state matrix and modal analysis, thus ensuring stability and convergence. The update procedures are designed such that the iterates converge to the optimal solution of the original optimization problem, steered by the gain parameter corresponding to the second largest eigenvalue of the system matrix. Lastly, the parallels between two distributed algorithms in terms of their speed are drawn.

1.4 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 introduces some background on graph theory, consensus theory, and convex optimization. Chapter 3 discusses a privacy preserving consensus algorithm for ED for distributed optimization. Chapter 4 proposes a distributed algorithm based on ADMM that handles any convex problem, and draws parallel between subgradient and ADMM algorithm in terms of their convergence speed. Chapter 5 concludes with directions ahead for future research.

CHAPTER 2

PRELIMINARIES

This chapter introduces some basic concept on graph theory, convex optimization, optimality conditions, and stability of discrete dynamic system. While the intent is not to detail the concepts that are quite basics, this background serves as the foundation to the algorithms presented in coming chapters.

2.1 Distributed Consensus Algorithms

2.1.1 Graph Theory

Notation: Let $\mathcal{G}$ denote a graph with the set of vertices $\mathcal{V} = \{1, \cdots, N\}$ and the edges $\mathcal{E} \subseteq \mathcal{V} \times V$; $\mathcal{G}$ is restricted to *simple, undirected* graph without multi-edges and self-edges. For a set $\mathcal{V}$, its cardinality is represented by $|\mathcal{V}|$. I define $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}$ and $j \neq i\}$ as the set of all the neighbors of the agent i and $d_i = |\mathcal{N}_i|$ as degree of any node i; $d_i = |\{j : (i, j) \in \mathcal{E}\}|$. $\mathbb{R}$ denotes the set of all real numbers, $\mathbb{0} \in \mathbb{R}^{N \times N}$ is a null matrix, $\boldsymbol{I} \in \mathbb{R}^{N \times N}$ is an identity matrix, $\mathbf{1} \in \mathbb{R}^{N \times N}$ is a unit vector, and γ is the eigenvalue. $(\cdot)^T$ is the transpose. Matrices and vectors are written in bold throughout the thesis.

Spectral Graph theory is the study of eigenvalues and eigenvectors to understand the interesting properties of graph. Let $\boldsymbol{A}$ be the *adjacency matrix* of graph $\mathcal{G}$, and $\mathbf{D}$ be the degree matrix with the vertex degree along its diagonal. Then, the graph Laplacian $\mathbf{L} \in \mathbb{R}^{N \times N}$ can be expressed in matrix form as:

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{2.1}$$

In full form:

$$l_{ij} := \begin{cases} \mathrm{d_i} & i = j, \\ -1 & i \neq j \text{and there is an edge (i,j)} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

The Laplacian consensus dynamics is given by the equation

$$\dot{\boldsymbol{x}} = -\boldsymbol{L}\boldsymbol{x} \tag{2.3}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is the values of corresponding vertices of the network. The spectral analysis on the Laplacian graph shows that

$$0 = \gamma_1(\boldsymbol{L}) \leq \gamma_2(\boldsymbol{L}) \leq \cdots \leq \gamma_N(\boldsymbol{L}) \tag{2.4}$$

where γ_i denotes eigenvalue

Observe that $\mathbf{L}$ is a symmetric positive semi-definite matrix, and $\boldsymbol{L}\mathbf{1}_N = 0$. For a connected graph $\mathcal{G}$, the connectivity is given by $\gamma_2(\boldsymbol{L})$, also called algebraic connectivity [12]. Consensus is achieved if and only if $\gamma_2(\boldsymbol{L})$ is greater than zero. The convergence speed to consensus is governed by $\gamma_2(\boldsymbol{L})$, *i.e.,* the slowest mode [13]. With these conditions, according to [9], any initial condition leads to a consensus and any consensus is an equilibrium, *i.e.,*

$$\frac{1}{N}\mathbf{1}^T\mathbf{x}(0) = \frac{1}{N}\sum_i x_i(0) \tag{2.5}$$

where i denotes the initial value $x_i(0) \in \mathbb{R}$, and $x(0) = (x_1(0), \cdots, x_N(0))$ denotes the vector of the initial values of the network. Below are some Lemmas on communication network, which I will employ later in coming chapters.

Lemma 1 *Communication topology between all the agents **A(t)** is connected at all times* $t \geq 0$*. i.e.,* $\gamma_2(\boldsymbol{L}) \geq 0$

Lemma 2 *Any bus* i *can exchange information only with its neighboring agents i.e.* $\mathcal{N}_i = \{j \in \mathcal{V} | (j,i) \in \mathcal{E}$ *and* $j \neq i\}$*. The coloring scheme of the network is available.*

Lemma 3 *Laplacian Matrix* $\boldsymbol{L}$ *is positive semidefinite,* $\sum_j \mathbf{L_{ij}} = 0$*, and* $\gamma_2(\boldsymbol{L})$ *is the algebraic connectivity of the network. The speed of convergence to reach the consensus in the iterative process is governed by* $\gamma_2(\boldsymbol{L})$ *for it represents the convergence rate of the slowest mode [13].*

2.1.2 Decentralized and Distributed Algorithms

Optimization algorithms are iterative process of searching for some desired iterates. They are usually tailored to address some specific need. Based on their architectural framework, they are categorized in three types: centralized, decentralized, and distributed. Figure 2.1 shows three different design architectures of algorithms.

In a centralized framework, one central entity runs the optimization by collecting information from all the agents, and communicate back to the agents. This algorithm has been employed

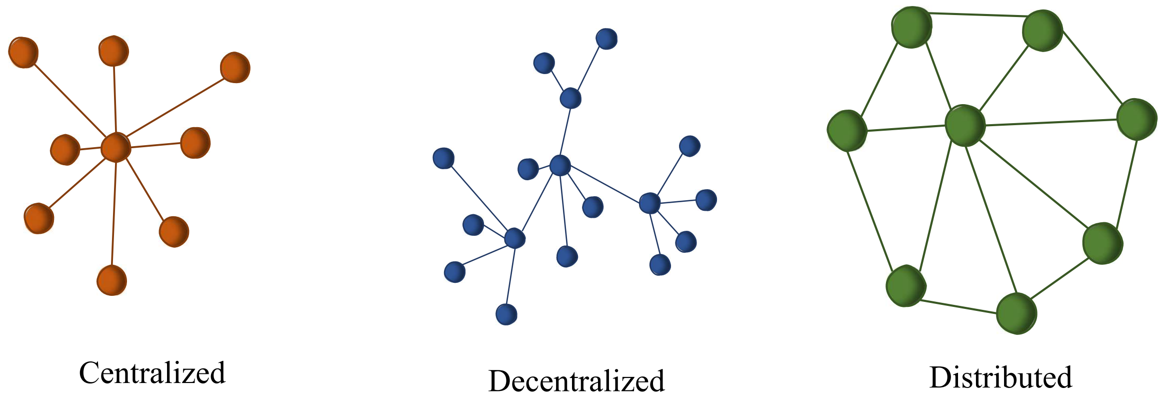

Figure 2.1 Figure showing the schematic of algorithms

for a long time, and still extensively used. While it is often characterized by its simplicity, the computation and communication challenges are quite serious. In decentralized algorithms, all agent can individually run the optimization in the presence of a master node(s) for tight coordination. This coordinator are often called by the name of aggregators. While the large problem is divided among agents, but this is still vulnerable to single point failure, and the concern for the confidentiality of data still alarms the participating agents.

In distributed algorithms, each agent optimizes it resources based on its and neighbor's information. Some of the characterizing features of distributed algorithms are worth highlighting here [14].

- *Scalibility*: The size of optimization problem doesn't grow with the network size, unlike centralized framework.

- *Robustness*: Most of the distributed algorithms can be modeled to handle communication delays, packet drop, plug-and-play etc.

- *Data Privacy*: Distributed algorithms do not need to trust a single entity to collect data, and facilitate the optimization process. This is essential to fair market design problems.

2.1.3 Dynamic Average Consensus

Let subscript i denote the initial value $\mathbf{x_i^0} \in \mathbb{R}^\mathbf{N}$, and $x(0) = [x_1(0), \cdots, x_N(0)]^T$ be the vector of the initial values of the network. The discussion below is on the computation of average, $\frac{1}{N}\sum_{i=1}^{N} x_i^0$, in distributed fashion, where each node in the graph communicates with only its neighbors. Let us consider *distributed linear iteration* of the form

$$x_i^{k+1} = w_{ii}x_i^k + \sum_{j \in \mathcal{N}_i} w_{ij}x_j^k, \quad i = 1, \ldots, N \tag{2.6}$$

where $k = 0, 1, 2, \ldots$ and w_{ij} is the weight on x_j at node i, and $\mathcal{N}_i$ denotes the set of all neighbors of agent i. Setting $w_{ij} = 0$ for $j \notin \mathcal{N}_i$, this iteration can be written as

$$\mathbf{x}^{k+1} = \mathbf{W}\mathbf{x}^k \tag{2.7}$$

or equivalently as

$$\mathbf{x}_i^{k+1} = \mathbf{W_i^T}\mathbf{x}^k \tag{2.8}$$

where $\mathbf{W} = [w_{ij}] = [W_1, \cdots, W_N] \in \mathbb{R}^{N \times N}$ with $W_i \in \mathbb{R}^{N \times 1}$. According to [15], the constraint on the sparsity pattern of the matrix $\mathbf{W}$ can be expressed as $\mathbf{W} \in \mathcal{S}$, where

$$\mathcal{S} = \left\{ W \in \mathbb{R}^{N \times N} | W_{ij} = 0 \text{ if } \{i, j\} \notin \mathcal{E} \text{ and } i \neq j \right\} \tag{2.9}$$

and (2.7) can be written as

$$\mathbf{x}^k = \mathbf{W}^k \mathbf{x}^0, \quad k = 0, 1, 2, \cdots \tag{2.10}$$

I am interested in a matrix $\mathbf{W}$ such that

$$\lim_{k \to \infty} \mathbf{W}^k = \frac{\mathbf{1}\mathbf{1}^T}{N} \quad \text{as} \tag{2.11}$$

This brings to state the following Lemma on weight matrix $\mathbf{W}$.

Lemma 4 *The following conditions are necessary to guarantee the convergence [15]*

$$\begin{cases} \mathbf{1}^T \mathbf{W} = \mathbf{1}^T \\ \mathbf{W}\mathbf{1} = \mathbf{1} \\ \zeta(\mathbf{W} - \mathbf{1}\mathbf{1}^T / N) < 1 \end{cases} \tag{2.12}$$

where $\zeta(\cdot)$ is the spectral radius of the matrix, $\mathbf{1} = [1, \cdots, 1]^T$ is the eigenvector associated with weight matrix $\mathbf{W}$, and $\mathbf{1}^T$ is the transpose of $\mathbf{1}$.

The choice of the coefficients of matrix **W** is detrimental to the speed of the convergence. *Constant edge weight* and *local degree weight* matrices are discussed in [15]. The *local degree weight* matrix **W** with coefficients depending only on the degree of the incident node is

$$w_{ij} = \begin{cases} \frac{1}{\max\{d_i,d_j\}} & \{i,j\} \in \mathcal{E} \\ 1 - \sum_{j\in\mathcal{E}} \frac{1}{\max\{d_i,d_j\}} & i = j \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

This method is implied from the Metropolis-Hastings algorithm and often called *Metropolis* method. The improved *Metropolis* called *Mean Metropolis* is proposed in [16] where

$$w_{ij} = \begin{cases} \frac{2}{\{d_i+d_j+\epsilon\}} & \{i,j\} \in \mathcal{E} \\ 1 - \sum_{j\in\mathcal{E}} \frac{2}{\{d_i+d_j+\epsilon\}} & i = j \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

where ϵ is a very small number. The average consensus described by (2.7) can be extended to on a consensus on general time varying signal [9].

In case the average of a dynamically changing signal **z** is desired, some modifications in the algorithm are required. Let $\Delta\mathbf{z}$ be the input bias (the difference of z in two consecutive time steps) applied to *average consensus system*. It can be claimed that the following modification to (2.7) makes the *dynamic consensus* algorithm tracks the time-varying *average consensus*:

$$\mathbf{x}^{k+1} = \mathbf{W}\mathbf{x}^k + \Delta\mathbf{z}^{\mathbf{k+1}} \tag{2.15}$$

where the bias $\Delta \mathbf{z^{k+1}} = \mathbf{z}^{k+1} - \mathbf{z}^{k}$. The extension in (2.15) is still fully distributed as each agent need to obtain the information from its directly connected neighbors. Figure 2.2 shows an example of communication network for IEEE bus.

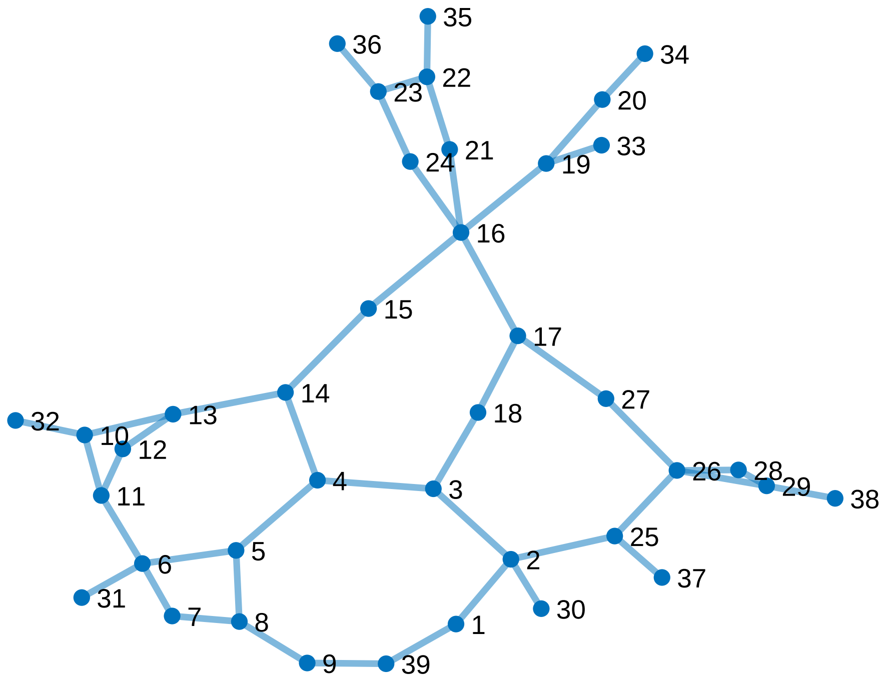

Figure 2.2 Communication Network for IEEE 39-bus; the value in each node can be thought as changing over time and each node estimates the average of the network in a distributed fashion

Lemma 5 *Dynamic average consensus algorithm in* (2.15) *follows the conservation property.*

Proof: The modified algorithm (2.15) tracks the time varying average consensus. In order to show that any consensus is an equilibrium, I show that (2.15) holds the conservation property.

If $\mathbf{x}(k)$ is subtracted from both sides of (2.15), we have:

$$\mathbf{x}^{k+1} - \mathbf{x}^{k} = \mathbf{W}\mathbf{x}^{k} + \Delta\mathbf{z} - \mathbf{x}^{k} \tag{2.16}$$

$$\mathbf{1}^{T}(\mathbf{x}^{k+1} - \mathbf{x}^{k}) = \mathbf{1}^{T}(\mathbf{W}\mathbf{x}^{k} + \Delta\mathbf{z} - \mathbf{x}^{k}) \tag{2.17}$$

From the property of matrix **W** stated in (2.12), I can write:

$$\sum \Delta\mathbf{x} = \mathbf{1}^{T}\mathbf{x}^{k} - \mathbf{1}^{T}\mathbf{x}^{k} + \mathbf{1}^{T}\Delta\mathbf{z} \tag{2.18}$$

$$\sum \Delta\mathbf{x} = \sum \Delta\mathbf{z} \tag{2.19}$$

Note that dynamic average consensus drives any initial points to consensus, and the consensus is stable. Thus, the conservation property at iteration is proven.■

2.2 Convex Optimization

2.2.1 Constrained Optimization

A constrained optimization is of the form: [17]

$$\underset{x\in R^{n}}{\text{minimize}}\, f(x) \tag{2.20a}$$

$$\text{subject to,} \quad h_i(x) = 0, \quad \forall i \epsilon \mathcal{E} \tag{2.20b}$$

$$g_i(x) \leq 0, \quad \forall i \in \mathcal{I} \tag{2.20c}$$

Preliminarily, the Lagrange function of the problem (2.20):

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i \in \mathcal{E}} \lambda_i h_i(x) + \sum_{i \in \mathcal{I}} \mu_i g_i(x) \tag{2.21}$$

2.2.2 Convex Functions

A set $S \in \mathbb{R}^N$ is a convex set if the straight line segment connecting two points in S lie entirely inside S. Mathematically,

$$\alpha x + (1-\alpha) y \in S \quad \forall \alpha \in [0, 1]$$

The function f is a convex function in its convex set domain S if it satisfy:

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) \quad \forall \alpha [0, 1] \tag{2.22}$$

$\forall\ x, y \in S$ [18]. A continuously differential function $f : \mathbb{R}^N \to \mathbb{R}$ is smooth if it has a globally Lipschitz gradient, $\exists L > 0 \ni \|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\| \quad \forall x, y \in \mathbb{R}^N$. Readers can refer [18] for more insight on convexity.

2.2.3 Optimality Conditions: KKT Conditions

I first define first order necessary conditions for the optimality, also know as Karush-Kuhn-Tucker (KKT) conditions.

Theorem 1 *Suppose x^* is a local solution of* (2.20), *and the fuctions f, h_i, g_i are continuously differentiable, and linearly independent constraint qualification holds at x^*. Then there is a vector of Lagrange multipliers $\lambda_i \in \mathcal{E}, \mu_i \in \mathcal{I}$, such that the following conditions hold at (x^*, λ^*, μ^*): [17]*

Stationarity: $\nabla \mathcal{L}(x^*, \lambda^*, \mu^*) = \nabla f(x) + \sum_{i\in\mathcal{E}} \lambda_i \nabla h_i(x) + \sum_{i\in\mathcal{I}} \mu_i \nabla g_i(x) = 0$

Primal feasibility: $h_i(x^*) = 0, \quad \forall i \in \mathcal{E} \quad g_i(x^*) \leq 0, \quad \forall i \in \mathcal{I}$

Dual feasibility: $\mu_i^* \geq 0, \quad \forall i \in \mathcal{I}$

Complementary slackness: $\sum_{i\in\mathcal{I}} \mu_i g_i(x^*) = 0$

For convex problem, KKT conditions are sufficient for optimality.

Lemma 6 *For any convex optimization problem with diffferentiable objective and constraint functions, any points that satisfy the **Karush–Kuhn–Tucker** (KKT) conditions are primal and dual optimal and have zero duality gap, and vice versa. [18].*

2.3 Distributed Optimization

2.3.1 Primal and Dual Decomposition

Some optimization problems are inherently decomposed. The optimization problem is decomposable if the decision variables are decoupled for each subproblem. The problem is non-decomposable if the subproblems share some decision variables [19].

- Decomposable Problem:

$$\min_{x,y} f(x) + g(y) = \min_{x} f(x) + \min_{y} g(y) \tag{2.23}$$

- Non decomposable Problem:

$$\min_{x,y,z} f(x,z) + g(y,z) \neq \min_{x,z} f(x,z) + \min_{y,z} g(y,z) \tag{2.24}$$

Problem (2.24), however, can be decomposed using primal and dual decomposition.

Primal Decomposition:

$$\min_{x,y} f(x,z) + g(y,z) = \min_{x} f(x,z) + \min_{y} g(y,z)$$

Algorithm:

1. Solve sub-problems for a fixed value of z

2. Update z

Dual decompostion: Let us rewrite the problem in (2.24):

$$\min_{x,y,z} f(x,z) + g(y,z) = \boldsymbol{p^*} \tag{2.25}$$

where $\boldsymbol{p^*}$ is the optimal solution for non-empty feasible domain of (2.25). Correspondingly,

$$\min_{x,y,z_1,z_2} f(x,z_1) + g(y,z_2) \quad \text{subject to, } z_1 = z_2 \tag{2.26}$$

The equality constrained can be relaxed using Lagrange Relaxation. The Lagrange function for (2.26) is:

$$\mathcal{L}(x, y, z_1, z_2; \lambda) = f(x, z_1) + g(y, z_2) + \lambda(z_1 - z_2) \tag{2.27}$$

Corresponding Lagrangian dual is:

$$\begin{aligned} D(\lambda) &= \min_{x,z}[f(x, z) + \lambda z] + \min_{y,z}[g(y, z) - \lambda z] \\ &= \max_{\lambda}\{\min_{x,z}[f(x, z) + \lambda z] + \min_{y,z}[g(y, z) - \lambda z]\} \end{aligned} \tag{2.28}$$

Readers can refer [19] for the detail explanation on decomposition. Note that (2.28) is a unconstrained problem. Constructing a dual problem from the original problem, sometimes called primal problem, can provide more insight such as finding the lower bound (upper bound) for minimization (maximization) problem. Once the constrained problem is transformed to equivalent unconstrained dual problem, all optimization algorithms for unconstrained can easily applied. Chapter 3 employs this method to solve distributed economic dispatch problem .

2.3.2 Distributed Gradient Algorithm

For a system with N agents in a network, a typical unconstrained optimization problem can be written as:

$$\underset{x \in \mathbb{R}^N}{\text{minimize}} \sum_{i \in \mathcal{V}} f_i(x) \tag{2.29}$$

Any agent i can find consensus in the underlying communication network with descent step along the local (sub) gradient direction of its own convex objective function [20]. Mathematically,

$$\mathbf{x}_i^{k+1} = \mathbf{W_i^T}\mathbf{x}^k - \rho(k) g_i(k) \tag{2.30}$$

where $x_i(k) \in \mathbb{R}^N$ is agent i's estimate of the optimal solution at time k, $\rho(k)$ is a diminishing step size, $\mathbf{W}$ is a stochastic weight matrix, and $g_i(k)$ is the (sub)gradient of local objective function $f_i(x)$ which is convex.

2.3.3 Augmented Lagrangian and ADMM

Let us write a typical constrained optimization problem:

$$\text{minimize } f(\boldsymbol{x}) + g(\boldsymbol{y}) \quad \text{Subject to,} \quad \boldsymbol{Ax} + \boldsymbol{By} = \boldsymbol{c} \tag{2.31}$$

The equivalent Lagrange relaxed function is:

$$\max_{\boldsymbol{\lambda}} \quad \min_{\boldsymbol{x},\boldsymbol{y}} L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \boldsymbol{\lambda}^{\boldsymbol{T}}(\boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c}) \tag{2.32}$$

A penalty term is added (hence the name augmented) in the objective function in order to increase the convergence speed to form the augmented Lagrangian function.

$$\max_{\boldsymbol{\lambda}} \quad \min_{\boldsymbol{x},\boldsymbol{y}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \boldsymbol{\lambda}^{\boldsymbol{T}}(\boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c}) + \underbrace{\frac{\rho}{2}\left\|(\boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c})\right\|_2^2}_{\text{penalty term}} \tag{2.33}$$

While the function in (2.33) is convergent without strict convexity and can take the value ∞, it comes at a cost of compromising the decomposibility feature from Lagrange relaxed problem (2.32) [21].

Alternative Direction Method of Multipliers (ADMM) introduces decomposibility in augmented Lagrangian without compromising the ability to take not strictly convex function, where primal iterates are updated in sequential fashion and a global dual updater collects all information to adjust the dual variable by pulling all decision variables to the optimum points. The unscaled problem (2.33) can be simplified to write in a compact form with the following sequence of iterations:

$$
\begin{aligned}
\boldsymbol{x}^{k+1} &:= \operatorname*{argmin}_{\boldsymbol{x}} \mathcal{L}_{\rho}(\boldsymbol{x}, \boldsymbol{y}^{k}, \boldsymbol{\lambda}^{k}) \\
\boldsymbol{y}^{k+1} &:= \operatorname*{argmin}_{\boldsymbol{y}} \mathcal{L}_{\rho}(\boldsymbol{x}^{k+1}, \boldsymbol{y}, \boldsymbol{\lambda}^{k}) \\
\boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^{k} + \rho(\boldsymbol{A}\boldsymbol{x}^{\boldsymbol{k+1}} + \boldsymbol{B}\boldsymbol{y}^{\boldsymbol{k+1}} - \boldsymbol{c})
\end{aligned}
$$

The penalty parameter ρ is equivalent to proportional gain in control theory, and λ can be interpreted as the integrator of the algorithm that integrates the error.

2.4 Algorithm Stability: Discrete Dynamic Systems

A system is discrete if the time variables have been quantized. The state-space representation of a discrete-dynamic system is

$$\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k + \boldsymbol{B}\boldsymbol{u}^k = \boldsymbol{f}(\boldsymbol{x}^k, \boldsymbol{u}^k) \tag{2.34}$$

$$\boldsymbol{y}^{k+1} = \boldsymbol{C}\boldsymbol{x}^k + \boldsymbol{D}\boldsymbol{u}^k = \boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{u}^k) \tag{2.35}$$

where $\boldsymbol{x}(\cdot)$ is a state vector, and $\boldsymbol{u}(\cdot)$ is an input vector. The natural response of state equation (2.34) is

$$\boldsymbol{x}^k = \boldsymbol{A}^k \boldsymbol{x}(0) \tag{2.36}$$

where $\boldsymbol{x}(\mathbf{0})$ is the initial condition. The stability of the system exclusively depends on matrix A.

Theorem 2 *Let $\gamma_1, \cdots, \gamma_m \quad m \leq n$ be the eigenvalues of $\boldsymbol{A} \in \mathbb{R}^{N \times N}$. The system is [22]*

- *asymptotically stable iff $|\gamma_i| < 1, \forall i = \{1, \cdots, m\}$*
- *stable if $|\gamma_i| \leq 1, \forall i = \{1, \cdots, m\}$*
- *unstable if $\exists$ i such that $|\gamma_i| > 1$*

The spectral radius of a matrix **A** is defined as the maximal modulus of all of its real and complex numbers; $\zeta(\boldsymbol{A}) = \max\{\gamma_1, \cdots, \gamma_N\}$.

Correspondingly,

$$1 = \gamma_1 \geq |\gamma_2| \geq \cdots \geq |\gamma_N| \tag{2.37}$$

Lemma 7 *A fixed point* $\boldsymbol{x}$ *of an iterative system* $\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k$ *is called stable if for every* $\epsilon > 0$ *there exists a* $\delta > 0$ *such that whenever* $\|\boldsymbol{x}^0 - \boldsymbol{x}^*\| < \delta$*, then the resulting iterates satisfy* $\|\boldsymbol{x}^0 - \boldsymbol{x}^*\| < \epsilon$ *for all* k *[23].*

CHAPTER 3

DISTRIBUTED ECONOMIC DISPATCH VIA SUBGRADIENT

3.1 Introduction

Economic Dispatch (ED) is a typical resource allocation problem of power systems. Traditionally, power plants share their generation cost function and generator limit with the Energy Management System (EMS), which runs the centralized optimization and shares the information back to power plants. The growth of Residential Energy Storage (RES), power-responsive demand, and distributed generation, however, has dramatically increased the size of the optimization problem, and added complexity on the communication and computation front. In view of these challenges, several decentralized and distributed algorithms are proposed as an alternative paradigm to solve the ED problem.

3.1.1 Motivation and Related Work

The study on distributed computation over networks can be traced back to seminal work for distributed decision making and parallel computation [24], and ”consensus” in multi-agent systems [25]. Distributed algorithms have recently gained renewed interest because of their ability to handle large problems using parallel computation as well as protect the privacy and confidentiality of data [14]. Authors in [20, 26, 27] review distributed algorithms for the optimization in power systems such as economic dispatch, coordination of Distributed Energy Resources (DERs) and optimal power flow. Most of these algorithms are based on dual decomposition such as Alternative Direction Method of Multipliers (ADMM) [21, 28], primal-dual [29], subgradient [5] among others. In these algorithms, the Lagrange multiplier (dual variable) associated with equality constraint couples all the decision variables, and each agent finds its optimum strategy in tight coordination with a master node(s). In the economic dispatch problem, the dual variable of the power balance constraint is identified as the market energy price which is updated by the central coordinator to eliminate any power mismatch between total demand and generation. The update procedure is as simple as increasing energy price in case of excess demand in order to attract more generations and vice versa. It can be presented as $\lambda(k+1) = \lambda(k) + \alpha(\sum p_{di} - \sum p_{gi})$ where λ is the market price, α is the gradient step, p_{di} is the demand at node i and p_{gi} is the generation at node i. The equation is referred to as subgradient (SG) update. The caveat of these algorithms is their need for a central coordinator to gather and broadcast information to all the agents which adds a communication burden and force to disclose private information of generators [26, 30].

There are several solutions to remove the central coordinator, which include the use of primal-dual [29, 31] and center-free algorithms [5]. The idea behind these algorithms is to let all agents have their own estimates of the dual variables, and to design some iterative algorithms for them to communicate with their neighbors only and reach a consensus about these variables. In a distributed economic dispatch problem, for example, agents have their own estimates of the market price at each iteration which must i) have to converge to a single consensus price eventually and ii) reflect the update procedure previously carried out by the removed central coordinator to balance total demand and generation. In addition to the market price, the total power imbalance at each moment must be known by all the agents. A distributed sub-gradient based solution has been proposed to coordinate among distributed renewable generators in [5] where the local frequency measurements are used as a proxy for power balance constraint. Although the frequency measurements are local, a central coordinator (named power system model) is still required to determine the local measurements. Authors in [6] proposed a gradient-based algorithm, with a coordinator agent to ensure power balance in the network.

In [7], an average consensus algorithm is developed for decentralized economic dispatch with the need of a master node(s) to obtain the primal and dual variables of the system in order to address power imbalance of the entire network. The authors in [4, 32, 33] eliminate the need for a master node in the consensus algorithm and introduce ”innovation” to address power mismatch. Two major shortcomings are identified in their proposed solutions. First, the convergence of the algorithm depends on the trade-off between the update coefficients chosen for consensus and innovation. The choice of diminishing step size makes the algorithm slow, and the fixed step-size would settle for sub-optimal solution points. Second, the update coefficients go to zero after enough number of iterations, and thence the algorithm is not suitable for economic dispatch to meet dynamically changing power demands. Authors in [34] modified [4] with fixed step-size, but it is restricted to communication topology with quadratic cost function. If the demand is changed in the bus without the generating unit, this algorithm cannot capture the change. In addition, a consensus based ADMM is proposed in [35, 36] to solve ED in a distributed fashion, but for a constant load demand. This algorithm also cannot capture the continuous change in the demand of the network because of the penetration of VRE resources.

Algorithms based on finite-time consensus is proposed to address the changing demand in [37], and the convergence speed is improved in [38]. In [38], the bus without the generating unit has to find a way to communicate its demand to its nearest generator bus as the design of algorithm restricts the communication network to the bus with generating unit. Authors in [8] propose a consensus-based algorithm, and they introduce mismatch as a feedback variable. The algorithm is designed with the strict assumption that power balance has already been ensured in the network, which is not a practical assumption for grids with high penetration of DER. While the algorithm can handle ”slight demand change” in the bus if it has the generating unit to supply, any demand change on purely demand buses or on generating buses operating at their generation limit can’t be addressed.

In [39], a distributed economic dispatch is realized using projected gradient and finite-time average consensus method. In this method, the update coefficients of agents are defined based on the eigenvalues of their Laplacian graph. Such a choice for updating coefficients is challenging since it simply contradicts the fact that the design of distributed algorithms requires all agents to determine their updating coefficients based on their own and, at most, their neighbors’ shared information. The need for every agent to know the Laplacian graph, calculate all eigenvalues in the same order, and determine what eigenvalue to be used at each iteration, makes the algorithm too challenging for large networks. In addition, if an agent(s) voluntarily leaves the participation, which in turn changes the eigenvalues of the network, the algorithm cannot address the configuration change.

In a departure from the existing literature, a consensus based optimization algorithm based on dynamic average consensus that allows each agent track the time-varying coupling constraint set is proposed. The algorithm proposed is agnostic to any initialization vector and is not restricted to agents with generating units. This feature gives freedom to the purely demand bus to respond by increasing or decreasing the demand based on its own estimate of market price. In addition, modal analysis is presented to determine the optimum value of the gradient step of the algorithm.

In the proposed Distributed-subgradient (D-subgradient) algorithm, at each iteration, agents first solve their own optimization problems given the market price, then they update their estimate of total power mismatch via dynamic average consensus, and finally they update their estimate of market price using average consensus and subgradient step to find consensus while ensuring the power balance in the network. Four desirable properties of the proposed algorithm are discussed below:

1. **Distributed Consensus** : Different from the existing distributed algorithm where the all the decision variables are coupled to meet certain time-invariant constraint, our proposed online algorithm finds consensus on the time-varying estimate of the average power-mismatch in a purely distributed setting via dynamic average consensus algorithm detailed in [9, 14] , thus ensuring power-balance in the network.

2. **Optimality**: The KKT conditions are leveraged to show that the generator's output and market price are the saddle points of the *Lagrangian* function.

3. **Stability and Convergence**: The proposed algorithm has been modeled as a discrete time dynamic system in order to study the characteristics of the system. The optimum gradient step was calculated based on the study of the state matrix and modal analysis, thus ensuring stability and convergence.

4. **Privacy**: Our algorithm ensures privacy of data as agents communicate the estimate of the average power mismatch to their neighbors, which quickly goes to zero, instead of the generated power and the demand of the node. Privacy is fundamental to fair electricity trading in the market.

The efficacy of the proposed algorithm was tested against different IEEE 300-bus network with dynamic demand profiles in order to show that the algorithm dynamically responds to the real-time changes.

The remainder of the chapter is presented as follows. Section 3.2 details the design of online distributed economic dispatch based on dual decomposition . Section 3.3 discusses the numerical stability and convergence of the proposed Distributed-subgradient (D-subgradient) algorithm. Section 3.4 presents and discusses the simulation results. Section 3.5 concludes the paper.

3.1.2 Notation

Let $\mathcal{G}$ denote a graph with the set of vertices $\mathcal{V} = \{1, \cdots, N\}$ and the edges $\mathcal{E} \subseteq \mathcal{V} \times V$; $\mathcal{G}$ is restricted to *simple, undirected* graph without multi-edges and self-edges. For a set $\mathcal{V}$, its cardinality is represented by $|\mathcal{V}|$. I define $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}$ and $j \neq i\}$ as the set of all the neighbors of the agent i and $d_i = |\mathcal{N}_i|$ as degree of any node i; $d_i = |\{j : (i, j) \in \mathcal{E}\}|$. $\mathbb{R}$ denotes the set of all real numbers, $\mathbf{0} \in \mathbb{R}^{N \times N}$ is a null matrix, $\boldsymbol{I} \in \mathbb{R}^{N \times N}$ is an identity matrix, $\mathbf{1}_N \in \mathbb{R}^{N \times N}$ is a unit vector, and γ is the eigenvalue. $(\cdot)^T$ is the transpose. D stands for distributed in D-subgradient, DED stands for distributed economic dispatch, SG stands for subgradient. Matrices and vectors are written in bold throughout the thesis.

3.2 Economic Dispatch Problem

Economic Dispatch is a typical resource allocation problem in power system, where each generator finds its optimum strategy in order to ensure power balance in the network [2].

Let each agent $i \in \mathcal{V} = \{1, \dots, N\}$ be the bus of the power network. The economic dispatch is, then, modeled as

$$\underset{\boldsymbol{p_g^k}}{\text{minimize}} \sum_{i \in \mathcal{V}} \underbrace{a_i (p_{gi}^k)^2 + b_i p_{gi}^k + c_i}_{C_i(p_{gi}^k)} \tag{3.1a}$$

$$\text{s.t} \sum_{i \in \mathcal{V}} p_{gi}^k = \sum_{i \in \mathcal{V}} p_{di}^k \tag{3.1b}$$

$$p_{gi}^{\text{min}} \leq p_{gi}^k \leq p_{gi}^{\text{max}} \qquad \forall i \in \mathcal{V} = \{1, \dots, N\} \tag{3.1c}$$

with the following details.

1. $\boldsymbol{p_g^k} = \{p_{gi}^k, \dots, p_{gN}^k\}$ denotes the power generation of all buses at the time-step k, and is the set of decision variables

2. (3.1a) represents the objective function as minimization of the total generation cost at any time-step k with $c_i(\cdot)|\mathbb{R} \to \mathbb{R}$ equal to the generation cost function of agent i. To preserve the convexity of (3.1), I assume $c_i(p_{gi})$ is Lipschitz continuous and convex, defined as $C_i(\cdot) = a_i p_{gi}^2 + b_i p_{gi} + c_i$ if agent i has a generating unit, and 0 otherwise; the coefficients a_i, b_i, c_i are the characterstic cost parameters of generator i and p_{gi} is its generation,

3. (3.1b) represents the power balance constraint; it ensures that the network's total generation and total demand are equal at any time, where p_{di} denotes the demand at bus i.

4. (3.1c) guarantees all generations operate within the minimum and maximum generation capacities at any time.

The optimization problem (3.1) has local objective function (3.1a) and local constraint (3.1c), coupled by the global constraint (3.1b) on the network's total generation and total demand. In this chapter, I aim to model problem (3.1) as a completely distributed optimization problem that allow each agent to determine the value of its decision variables $p_{gi} \in \boldsymbol{p_g}$.

Let us define agent i's power mismatch by $p_i = p_{gi} - p_{di}$ and its feasible set by

$$\Omega_i = [p_i^{\min}, p_i^{\max}] = [p_{gi}^{\min} - p_{d_i}, p_{gi}^{\max} - p_{d_i}]$$

Substituting $p_{gi} = p_i + p_{di}$ in the cost function $c_i(p_{gi})$ defined in the problem formulation (3.1), $C_i(p_{gi})$ modifies to $f_i(p_i)$ where

$$f_i(p_i) = a_i p_i^2 + (2a_i p_{di} + b_i)p_i + a_i p_{di}^2 + b_i p_{di} + c \tag{3.2}$$

Then, (3.1) can be written as

$$\underset{\boldsymbol{p} \in \boldsymbol{\Omega}}{\text{minimize}} \quad f(\boldsymbol{p}) := \sum_{i \in \mathcal{V}} f_i(p_i) \tag{3.3a}$$

$$\text{subject to,} \quad \sum_{i \in \mathcal{V}} p_i = 0 \tag{3.3b}$$

where $f_i : \mathbb{R} \rightarrow \mathbb{R}$ is a local convex objective function of agent i, $\boldsymbol{\Omega} = [\Omega_1; \cdots ; \Omega_N] \in \mathbb{R}^N$ is compact and convex constraint sets, and $\boldsymbol{p} \in \mathbb{R}^N$ is the set of the decision variables. (3.3b) represents a global equality constraint, and I assume that the set of feasible points is non-empty. I formalize the assumptions in the following Lemma.

Lemma 8 *The function f_i and thus $\boldsymbol{f} : \mathbb{R}^N \rightarrow \mathbb{R}$ is closed, proper, and convex. The interior of the feasible region is non-empty. This is known by Slater's condition for convex optimization problems [18]*

3.2.1 Dual Decomposition of Economic Dispatch

The dual representation of (3.1) is

$$\max_{\lambda \in \mathbb{R}} \sum_{i \in \mathcal{V}} \underbrace{\left\{ \min_{\Omega_i} \; f_i(p_i) - \lambda p_i \quad \right\}}_{\Phi_i^{\dagger}(\lambda)}$$

where the optimal solution of the minimization problem for given λ can be explicitly defined as

$$\Phi_i^{\dagger}(\lambda) := f_i(p_i^{\dagger}) \tag{3.4}$$

with

$$p_i^{\dagger}(\lambda) := \left[MC_i^{-1}(\lambda) \right]_{p_i{}^{min}}^{p_i^{max}} \tag{3.5}$$

where $MC_i(p_i) = \frac{\partial f_i}{\partial p_i}$ is the marginal cost corresponding to bus i, MC_i^{-1} denotes the inverse of marginal cost of power output, and $[\cdot]_a^b$ denotes max$\{$min$\{\cdot, b\}, a\}$ for $a, b \in \mathbb{R}$, $a \leq b$.

The dual variable λ is updated as

$$\lambda^{k+1} = \lambda^k - \alpha \sum_{i \in \mathcal{V}} p_i^k = \lambda^k - \alpha N \overline{p}^k = \lambda^k - \rho \overline{p}^k \quad \forall i \tag{3.6}$$

to ensure that the power balance constraint is met where α and ρ are the feedback gain parameters for average mismatch, and N is the total number of agents of the network. ρ is chosen below certain critical value to ensure stability of convergence, which will be discussed later in the chapter. The price update reflects the fact that with a demand excess, the market price must increase to attract more generators to supply power, and the opposite otherwise.

Let $\mathcal{X}_g \subset \mathcal{V}$ be the set of buses with the generating units, For the cost function defined in the problem formulation (3.1), the marginal cost of generation is

$$df_i(p_i)/dp_i = 2a_i p_i + 2a_i p_{di} + b_i \doteq \lambda_i \tag{3.7}$$

in its feasible region. The feasible region of any agent i is bounded by the lower and the upper limit of its power mismatch. Consequently,

$$p_i^* = \left[\frac{\lambda^* - b_i - 2a_i p_{di}^k}{2a_i} \right]_{p_i^{min}}^{p_i^{max}} \tag{3.8}$$

where superscript p_i^* and λ^* denote the corresponding parameters values at the optimal solution. Thus, (3.8) can be written in a generalized form as

$$\boldsymbol{p}^{k+1} = \left[\boldsymbol{\Lambda}\boldsymbol{\lambda}^k - \boldsymbol{p}_d^k - \boldsymbol{\Lambda}\boldsymbol{b}\right]_{\boldsymbol{p^{min}}}^{\boldsymbol{p^{max}}} \tag{3.9}$$

where $\boldsymbol{a} = \text{diag}\{a_i\}, \boldsymbol{b} = \text{diag}\{b_i\}, \boldsymbol{\Lambda} = \text{diag}\{1/(2a_i)\}$. For $i \notin \mathcal{X}_g$, $a_i = b_i = \Lambda_i = 0$.

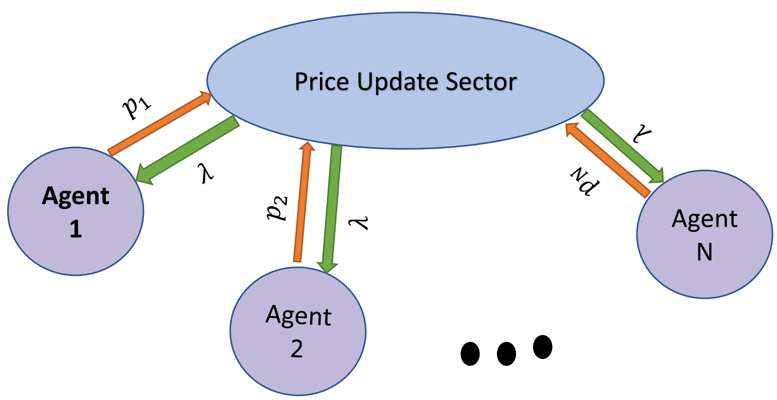

Figure 3.1 Information flow in decentralized platform

Decentralized Algorithm for Economic Dispatch: The algorithm for all the agents in the network can be summarized in vector format as

$$\begin{cases} \boldsymbol{p}^{k+1} = \left[\Lambda\boldsymbol{\lambda}^k - \boldsymbol{p}_d^k - \boldsymbol{\Lambda}\boldsymbol{b}\right]_{\boldsymbol{p^{min}}}^{\boldsymbol{p^{max}}} \\ \lambda^{k+1} = \lambda^k - \rho\overline{p}^k \end{cases} \tag{3.10}$$

In a decentralized framework, a central coordinator (price updater) tightly coordinates between all agents to ensure supply and demand balance in the network. In the semantics of optimization, λ is a dual variable in Lagrange relaxed function associated with equality constraint, and is updated based on the information from all the participating agents. It should be noted that in an optimization problem with an equality constraint(s) binding all the agents, the feasible region of an individual agent's decision variable is influenced by other agents' decision variables [20]. In addition to the concerns on data privacy and vulnerability to single point failure, the central question is, ”Who adjust the market price?”

Remark: Price adjustment process in general equilibrium theory is known by *tatonnement* process, which is a simple way to model supply and demand of any market. The fundamental idea is that all generators find optimum strategy to adjust their output based on their own cost function and local sets of constraints for the given market price. With all agents in perfect coordination, the market goes to equilibrium after some trial and error process. That is, the price updater increases the market price when total generation falls short and decreases for surplus in supply. The equilibrium price is in locally stable equilibrium if the initial price in the small neighborhood of λ^* converges to optimum price , and it is globally stable if any initial value of market price λ converges to λ^* [40]. This raises a very important question of who can possibly be that benign updater for price adjustment. This following section presents the formulation without the price updater.

3.2.2 Distributed Economic Dispatch

The objective function $\Phi_i^\dagger(\lambda)$ is not separable across buses $i \in \mathcal{V}$ since it has a scalar variable λ which is common between all $i \in \mathcal{V}$. The variables λ_i are constrained to be equal for all $i \in \mathcal{V}$

at the optimal solution i.e. $\lambda_i = \lambda_j \;\; \forall i, j \in \mathcal{V}$. In a connected graph, it suffices to define the constraint $\lambda_i = \lambda_j$ just across the neighboring agents i.e. $\forall (i, j) \in \mathcal{E}$. With these modifications, the *Distributed Economic Dispatch* (DED) problem can be written as below.

$$\max_{\lambda} \;\; \Phi(\lambda) := \sum_{i \in \mathcal{V}} \Phi_i^{\dagger}(\lambda_i) \quad \text{s. t.} \;\; \lambda_i = \lambda_j \quad \forall (i, j) \in \mathcal{E} \tag{3.11}$$

The following two results suggest solving the distributed optimization problem (3.11) is equivalent to solving problem (3.3) . The unique optimal point (p_i^*) will then be recovered from the unique dual optimal λ^*. A proof has been provided in Appendix B of [41] for these two results.

Lemma 9 *The objective function* Φ *of DED in* (3.11) *is strictly concave over* $\mathbb{R}^N$.

Lemma 10 *Optimization problem* (3.11) *has a unique optimal point* (λ^*) *with* $\lambda_i^* = \lambda_j^* = \lambda^*$ *for all* $(i, j) \in \mathcal{E}$. *DED has a unique optimal point* (p^*) *where* $p_i^* = p_i^{\dagger}(\lambda^*)$.

Similar to (3.4)-(3.5), any agent i can find its optimal solution point given λ_i as

$$\Phi_i^{\dagger}(\lambda_i) := f_i(p_i^{\dagger}) \tag{3.12}$$

with

$$p_i^{\dagger}(\lambda_i) := \left[MC_i^{-1}(\lambda_i) \right]_{p_i{}^{min}}^{p_i^{max}}. \tag{3.13}$$

In order to realize this, all the market prices should reach to one consensus value while power balance constrained is ensured in the network. Based on Lemma 1, 2, and Lemma 4, all participating agents reach consensus on market price given as:

$$\lambda_i^{k+1} = W_i^T \boldsymbol{\lambda}^k - \rho \overline{p}_i^k \tag{3.14}$$

where $\boldsymbol{\lambda} = [\lambda_1; \cdots; \lambda_N] \in \mathbb{R}^N$, and $W_i \in \mathbb{R}^{N\times 1}$ is the vector of agent i's update weights as defined in (2.14). Notice that the global variable average power mismatch $\overline{p}$ is replaced with the local variable $\overline{p_i}$, defined as the agent i's estimate of the average power mismatch. For this, any agent i estimates the network's average mismatch using dynamic average consensus algorithm in (2.15).

$$\overline{p}_i^{k+1} = W_i^T \overline{\boldsymbol{p}}^k + (p_i^{k+1} - p_i^k) \tag{3.15}$$

where $\overline{\boldsymbol{p}} = [\overline{p}_1; \cdots; \overline{p}_N] \in \mathbb{R}^N$.

Consequently, fully distributed form of (3.8) is:

$$p_i^{\dagger}(\lambda_i) := \left[MC_i^{-1}(\lambda_i)\right]_{p_i{}^{min}}^{p_i^{max}} = \left[\frac{\lambda_i^* - b_i - 2a_i p_{di}^k}{2a_i}\right]_{p_i^{min}}^{p_i^{max}} \tag{3.16}$$

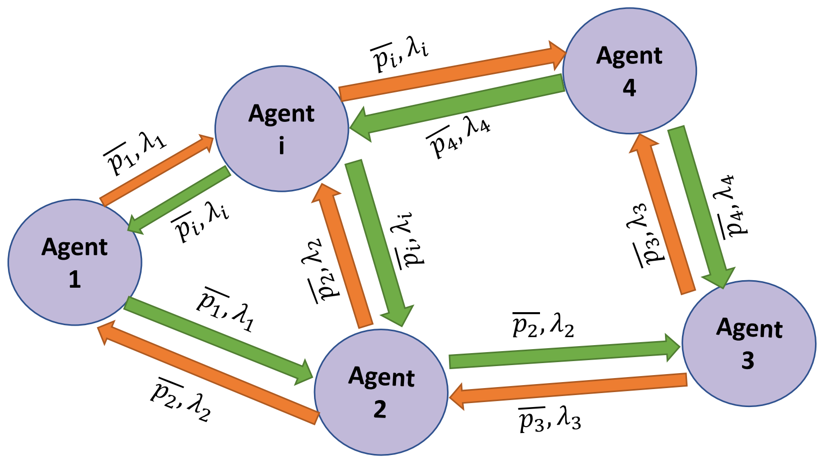

Figure 3.2 Information flow in distributed platform

The distributed algorithm for all the agents in the network can be summarized in vector format as

$$\begin{cases} \boldsymbol{p}^{k+1} \in \left[\boldsymbol{\Lambda}\boldsymbol{\lambda}^k - \boldsymbol{p}_d^k - \boldsymbol{\Lambda}\boldsymbol{b}\right]_{\boldsymbol{p^{min}}}^{\boldsymbol{p^{max}}} \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{W}\boldsymbol{\lambda}^k - \rho\overline{\boldsymbol{p}}^k \\ \overline{\boldsymbol{p}}^{k+1} = \boldsymbol{W}\overline{\boldsymbol{p}}^k + (\boldsymbol{p}^{k+1} - \boldsymbol{p}^k) \end{cases} \tag{3.17}$$

where $\boldsymbol{\lambda} = [\lambda_1; \cdots; \lambda_N] \in \mathbb{R}^N$.

With reference to Lemma 8, I state the following Lemma to investigate the **optimality** of our proposed DED Algorithm (3.17).

Lemma 11 *For any convex optimization problem with diffferentiable objective and constraint functions, any points that satisfy the* ***Karush–Kuhn–Tucker*** *(KKT) conditions are primal and dual optimal and have zero duality gap, and vice versa. [18].*

Let us reintroduce the DED optimization problem to investigate the optimality. The Lagrange relaxation of the DED for any agent i is:

$$\inf_{p_i\in\Omega_i}\sup_{\lambda_i\in\mathbb{R}}\Phi_i^{\dagger}(p_i,\lambda_i) := \inf_{p_i\in\Omega_i}\sup_{\lambda_i\in\mathbb{R}}\left\{f_i(p_i)-\lambda_i(p_i-p_i^k+N\overline{p}^k)\right\} := \inf_{p_i\in\Omega_i}\sup_{\lambda_i\in\mathbb{R}}\left\{f_i(p_i)-\lambda_i p_i\right\} \tag{3.18}$$

where $p_i - p_i^k + N\overline{p}^k = \mathbf{1}^T\boldsymbol{p} = 0$ is the power balance constraint of the network.

Consequently, based on strong duality of the problem

$$\underbrace{\inf_{\boldsymbol{p}\in\boldsymbol{\Omega}}\sup_{\boldsymbol{\lambda}\in\mathbb{R}^N}\boldsymbol{\Phi}^{\dagger}(\boldsymbol{p},\boldsymbol{\lambda})}_{\text{primal problem}} = \underbrace{\sup_{\boldsymbol{\lambda}\in\mathbb{R}^N}\inf_{\boldsymbol{p}\in\boldsymbol{\Omega}}\boldsymbol{\Phi}^{\dagger}(\boldsymbol{p},\boldsymbol{\lambda})}_{\text{dual problem}} \tag{3.19}$$

where $\boldsymbol{\Phi}^{\dagger} = [\Phi_1^{\dagger};\cdots;\Phi_N^{\dagger}]$.

I need to show that a fixed point $(\boldsymbol{p},\boldsymbol{\lambda})$ is the solution of (3.19). In other words, $(\boldsymbol{p},\boldsymbol{\lambda})$ are the KKT points. Using stationary condition for (3.18) in vector format,

$$\nabla_{\boldsymbol{p}}\boldsymbol{\Phi} \in \partial f(\boldsymbol{p}^*) - \boldsymbol{\lambda}^* \in \boldsymbol{0} \tag{3.20}$$

$$\nabla_{\boldsymbol{\lambda}}\boldsymbol{\Phi} \in \boldsymbol{p}^* - \boldsymbol{p}^{\boldsymbol{k}} + \overline{\boldsymbol{p}}^k \in \boldsymbol{0} \tag{3.21}$$

where ∇ is the partial differentiation operator. The update process of market price $\boldsymbol{\lambda}$ in (3.17) can be written as $\boldsymbol{W}\boldsymbol{\lambda}^{\boldsymbol{k}} - \boldsymbol{\lambda}^{k+1} = \rho\overline{\boldsymbol{p}}^k$. It is pretty straight forward that each market price finds stable point once the mismatch term goes to zero. Then, the consensus on λ is guaranteed based on the properties of $\boldsymbol{W}$ in (2.12). So, for a fixed point $(\boldsymbol{p}, \boldsymbol{\lambda})$, $\boldsymbol{p^*} - \boldsymbol{p^k} = 0$ and $\bar{\boldsymbol{p}}^k = 0$, hence the left hand side of (3.21) goes to zero, and all market prices settle in their optimum points. This can now be readily interpreted as price adjustment process in general equilibrium theory but in distributed fashion. All players find optimum strategy to adjust its output based on its own objective function and local sets of constraints for their estimated market price, which in turn depends on its estimated mismatch of the network.

Distributed Algorithm for Economic Dispatch: The proposed distributed Economic Dispatch in (3.17) now can be designed to solve in an iterative procedure. Each agent after receiving the information on price, estimate of the average generation and demand of the network from its directly connected neighbors $\mathcal{N}_i$ finds the optimum value of generation. It then updates its price and the estimate of network's power generation and demand and broadcasts to its connected neighbors.

Algorithm 1 elaborates the iterative processes for our consensus based economic dispatch problem.

Algorithm 1: Distributed-Subgradient (D-Subgradient) Algorithm from the perspective of agent i

---

Input: cost-coefficients $\{a_i, b_i\}$, and control parameter ρ

Initialization: $\lambda_i{}^0$, $p_i{}^0 = \overline{p_i}{}^0 = -p_d{}^0$

for $k = 0$ *to* ∞ **do**

Receive $\overline{\boldsymbol{p}}$ and $\boldsymbol{\lambda}$. Each agent receives data from its neighboring agents only.

Compute power mismatch p^{k+1} as:

$$p^{k+1} \in \left[\Lambda_i \lambda_i{}^k - p_{di}^k - \Lambda_i b_i\right]_{p^{min}}^{p^{max}}$$

Update $\lambda_i^{k+1} = W_i^T \boldsymbol{\lambda}^k - \rho \overline{p}_i^k$

Update $\overline{p}_i^{k+1} = W_i^T \overline{\boldsymbol{p}}^k + (p_i^{k+1} - p_i^k)$

Broadcast $\overline{p_i}{}^{k+1}$, $\lambda_i{}^{k+1}$. Each agent broadcasts its data to its neighboring agents only.

3.3 Numerical Stability and Convergence

A system is discrete if the time variables have been quantized. I model the optimization problem in (3.17) to investigate the stability and convergence of our proposed DED algorithm.

The state-space representation of a discrete-dynamic system is

$$\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k + \boldsymbol{B}\boldsymbol{u}^k = \boldsymbol{f}(\boldsymbol{x}^k, \boldsymbol{u}^k) \tag{3.22}$$

$$\boldsymbol{y}^{k+1} = \boldsymbol{C}\boldsymbol{x}^k + \boldsymbol{D}\boldsymbol{u}^k = \boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{u}^k) \tag{3.23}$$

where $\boldsymbol{x}(\cdot)$ is a state vector, and $\boldsymbol{u}(\cdot)$ is an input vector. Recall that the natural response of state equation (3.22) is

$$\boldsymbol{x}^k = \boldsymbol{A}^k\boldsymbol{x}(0) \tag{3.24}$$

where $\boldsymbol{x}(\boldsymbol{0})$ is the initial condition. The stability of the system exclusively depends on state matrix A.

The state equation for market price vector is

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{W}\boldsymbol{\lambda}^k - \rho\overline{\boldsymbol{p}}^k \tag{3.25}$$

Similarly, the state equation of power output is:

$$\boldsymbol{p}^{k+1} = \boldsymbol{\Lambda}\boldsymbol{\lambda}^k - \boldsymbol{p}_d^k - \boldsymbol{\Lambda}\boldsymbol{b} \tag{3.26}$$

The state equation for average power mismatch is:

$$\overline{\boldsymbol{p}}^{k+1} = \boldsymbol{W}\overline{\boldsymbol{p}}^k + (\boldsymbol{p}^{k+1} - \boldsymbol{p}^k)$$

Consequently,

$$\bar{\boldsymbol{p}}^{k+1} = \boldsymbol{W}\bar{\boldsymbol{p}}^{k} + (\boldsymbol{\Lambda}\boldsymbol{\lambda}^{k} - \boldsymbol{p}_{d}^{k} - \boldsymbol{\Lambda}\boldsymbol{b} - \boldsymbol{p}^{k}) \tag{3.27}$$

The state equations (3.25), (3.26), (3.27) can be written in matrix form as

$$\begin{bmatrix} \boldsymbol{\lambda}^{k+1} \\ \boldsymbol{p}^{k+1} \\ \bar{\boldsymbol{p}}^{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{W} & \mathbb{0} & -\rho\boldsymbol{I} \\ \boldsymbol{\Lambda} & \mathbb{0} & \mathbb{0} \\ \boldsymbol{\Lambda} & -\boldsymbol{I} & \boldsymbol{W} \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} \boldsymbol{\lambda}^{k} \\ \boldsymbol{p}^{k} \\ \bar{\boldsymbol{p}}^{k} \end{bmatrix}}_{\boldsymbol{x}} + \underbrace{\begin{bmatrix} \mathbb{0} & \mathbb{0} \\ -\boldsymbol{I} & -\boldsymbol{b}\boldsymbol{\Lambda} \\ -\boldsymbol{I} & -\boldsymbol{b}\boldsymbol{\Lambda} \end{bmatrix}}_{\boldsymbol{B}} \underbrace{\begin{bmatrix} \mathbf{P_d}^{k} \\ \mathbf{1}_{N} \end{bmatrix}}_{\boldsymbol{u}} \tag{3.28}$$

Output Matrix:

$$\begin{bmatrix} \boldsymbol{\lambda}(k) \\ \mathbf{P_g}(k) \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{I} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \boldsymbol{I} & \mathbb{0} \end{bmatrix}}_{\boldsymbol{C}} \underbrace{\begin{bmatrix} \boldsymbol{\lambda}^{k} \\ \boldsymbol{p}^{k} \\ \bar{\boldsymbol{p}}^{k} \end{bmatrix}}_{\boldsymbol{x}} + \underbrace{\begin{bmatrix} \mathbb{0} & \mathbb{0} \\ \boldsymbol{I} & \mathbb{0} \end{bmatrix}}_{\boldsymbol{D}} \underbrace{\begin{bmatrix} \mathbf{P_d}^{k} \\ \mathbf{1}_{N} \end{bmatrix}}_{\boldsymbol{u}} \tag{3.29}$$

Comparing (3.28) and (3.29) with the canonical form in (3.22), $\boldsymbol{A} \in \mathbb{R}^{3N\times3N}$ is the state matrix, $\boldsymbol{B} \in \mathbb{R}^{3N\times2N}$ is the input matrix, $\boldsymbol{C} \in \mathbb{R}^{2N\times3N}$ is the output matrix, and $\boldsymbol{D} \in \mathbb{R}^{2N\times2N}$ is the feed-forward matrix.

Theorem 3 *Let $\gamma_1, \cdots, \gamma_m \quad m \leq n$ be the eigenvalues of $\boldsymbol{A} \in \mathbb{R}^{3N\times3N}$. The system* (3.22) *is [22]*

- *asymptotically stable iff* $|\gamma_i| < 1, \forall i = \{1, \cdots, m\}$
- *stable if* $|\gamma_i| \leq 1, \forall i = \{1, \cdots, m\}$
- *unstable if* $\exists$ *i such that* $|\gamma_i| > 1$

The spectral radius of a matrix **A** is defined as the maximal modulus of all of its real and complex numbers; $\zeta(\boldsymbol{A}) = \max\{\gamma_1, \cdots, \gamma_{3N}\}$. Based on theorem (5), the stability can be examined by the study of spectral radius. I am interested in the stability to a non-zero fixed point, thus

$$\zeta(\boldsymbol{A}) = 1 \tag{3.30}$$

Correspondingly,

$$1 = \gamma_1 \geq |\gamma_2| \geq \cdots \geq |\gamma_{3N}| \tag{3.31}$$

Lemma 12 *A fixed point* $\boldsymbol{x}$ *of an iterative system* $\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k$ *is called stable if for every* $\epsilon > 0$ *there exists a* $\delta > 0$ *such that whenever* $\|\boldsymbol{x}^0 - \boldsymbol{x}^*\| < \delta$*, then the resulting iterates satisfy* $\|\boldsymbol{x}^0 - \boldsymbol{x}^*\| < \epsilon$ *for all* k *[23].*

The solution of the natural response (3.24) is:

$$\boldsymbol{x}(k) = \sum_{i=1}^{3N} c_i \gamma_i^k \xi_i \tag{3.32}$$

where c_i is the scalar prescribed by initial condition $\boldsymbol{x}(0)$, γ_i is the eigenvalue of $\boldsymbol{A}$, and ξ_i is linearly independent eigenvector. Consequently,

$$\lim_{k\to\infty} \boldsymbol{x}(k) = c_1\xi_1 \tag{3.33}$$

Note that the stability means the stability of all the fixed points, which in fact are the elements of eigenspace xi_1 corresponding to $\gamma = 1$.

Corollary 1 *All solutions of linear iterative solution $\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k$ converges to a vector ξ that lies in the $\gamma_1 = 1$ eigenspace provided* (3.30) *holds true. Moreover, the rate of convergence of the solution is governed by the modulus $|\gamma_2|$ of the subdominant eigenvalue. [23].*

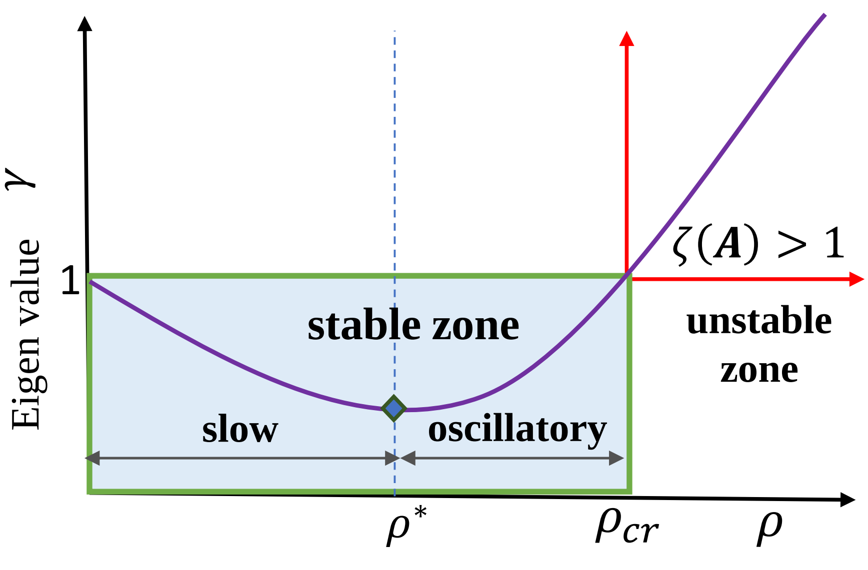

Figure 3.3 Plot showing the locus of eigenvalues for different feedback gain parameter ρ; $\gamma = \max|\gamma_j|, j \notin i; |\gamma_i| = 1$

I am interested in investigating the correlation of gain parameter ρ with the convergence speed of the algorithm. Figure 3.3 shows the plot of second highest eigenvalues of the state matrix $\mathbf{A}$ for different input of ρs. The algorithm becomes unstable beyond the critical value ρ_{cr}. As stated in the corollary 2, smaller the second largest eigenvalue, faster the convergence. As depicted in the Figure 3.3, the subdominant eigenvalue quickly decreases as ρ is decreased. The system has oscillatory behavior at the margin of stability, and the oscillation is controller with lower gain. Recall that the update process of the market price λ has a consensus part and a gradient part. With the higher ρ, the mismatch of the network goes to acceptable limit before the state variables settle to a fixed point. A trade off can be sought by choosing ρ in between ρ_{cr} and ρ corresponding to minimum subdominant eigenvalue.

3.4 Results and Discussions

This section presents the distributed online economic dispatch results for IEEE-300, followed by the discussion on the choice of optimum gain parameter ρ.

3.4.1 Simulation Setup

The **Algorithm** 1 is implemented in MATLAB R2020a environment to test the efficacy of the optimization problem (3.10) for IEEE network ranging from IEEE-14 bus to IEEE-300 bus. The cost-coefficients a_i and b_i , generator limit, and the initial values of power demand at each bus are adopted from [42]. The power demands of all buses shift in three different positions during the iterative process to mimick the dynamic nature of the load. This is accomplished by uniform distribution functions, that are used to randomize the drop percentages for each individual load.

In all cases, as the algorithm is robust enough to drive any initial value to convergence, I initialize power generations at zero, and prices at a uniform distribution around the initial optimal value. I use *Mean Metropolis* algorithm with $\epsilon = 1$ to set up the weight matrix $\boldsymbol{W}$ assuming that each bus of the network is an agent and the communication topology follows the electrical connection between buses. All the case study results are benchmarked against MATPOWER 7.0.

3.4.2 Algorithm Performance

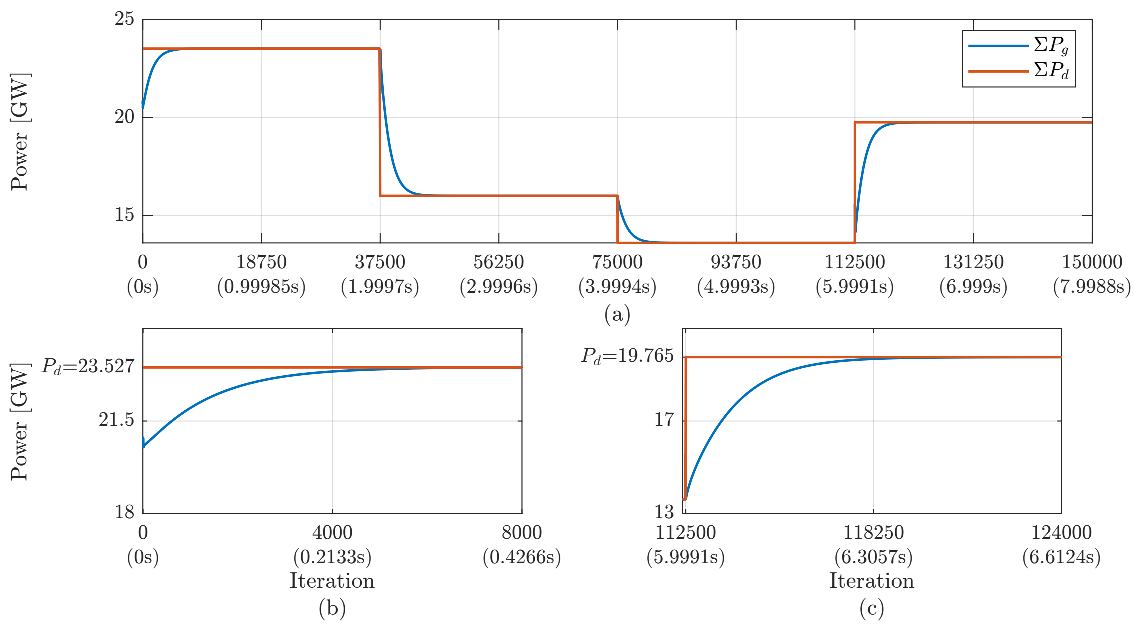

Figure 3.4 Results for IEEE 300-bus case; (a) displays the total generated power and total demand of the network, (b) shows the magnified version of the convergence process when all the units start from 0 MW, (c) shows the generating units moving together from one equilibrium point to another after the change in demand

Figure 3.4 (a) shows that all 69 generating units in IEEE 300-bus network collectively ensuring power balance in the network, while finding their optimal strategies. Figure. 3.4 (b) is the

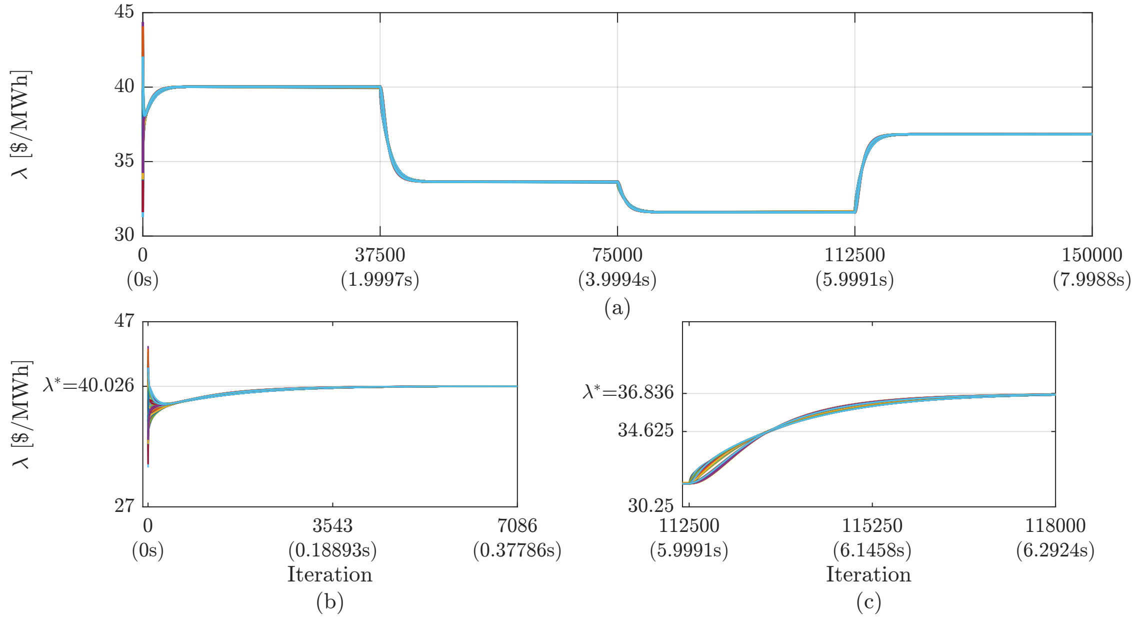

Figure 3.5 Results for IEEE 300-bus case; (a) displays the convergence process for the market price (λ); (b) shows the detail of the initial convergence process; (c) shows the detail of the last convergence process

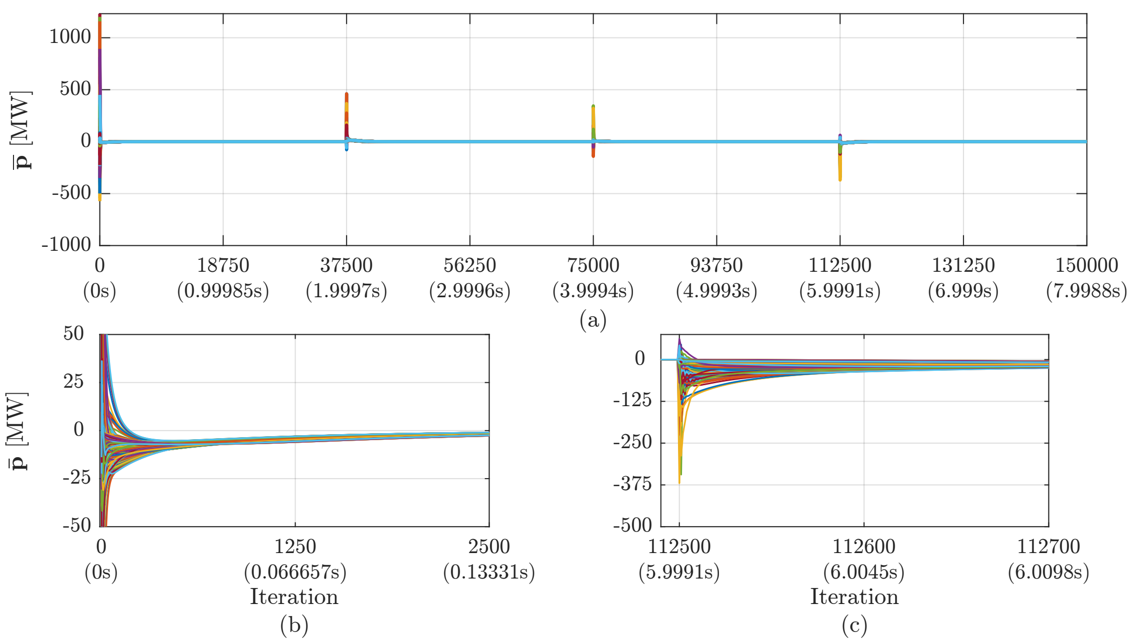

Figure 3.6 Results for IEEE 300-bus case; (a) displays $\bar{P}_i$ converging to zero; (b) details of the initial convergence process; (c) details of the last convergence process

magnified version showing the convergence at 0.4266s (8000 iterations), and Figure. 3.4 (c) shows the magnified version when all the generating units move from one equilibrium point to another. The demand fluctuates in every 2 seconds, and the generating units quickly find their optimal points based on the demand of the network. While the number of iteration is higher, the time for each iteration is quite small because all nodes participate in the update process and the optimization problem doesn't need any sophistic ted computation.

Figure 3.5 (a) shows the convergence of market price λ (dual variable) of all the nodes. Similar to Figure 3.4, all agents reach one consensus value, and any consensus that is stable is the optimal dual variable. Figure 3.5 (b) depicts the magnified version of the initial consensus process, where all agents take $0.37786s(7086)$ iterations to find the settling point. In the iterative process, agents calculate their iterates based on their estimation of average and gradient descent to find the market equilibrium. In Figure3.5 (c) all agent increase λ based on their estimation as the network had sudden deficit of power because of the increment in demand.

Our algorithm preserves the privacy of the agents participating in the market. Figure 3.6 (a) shows the estimate of total power mismatch by each agent. With the the change in demand in the network in different nodes, all agents quickly estimate the power imbalance of the network in order to adjust their market price. Figure 3.6 (b) and Figure 3.6 (c) are the enlarged version of the individual estimates of price mismatch. In our algorithm, any agent i share just two information to their directly connected neighbors, λ_i and $\overline{p_i}$. As displayed, the estimate of price quickly vanish to zero as the iteration proceeds which makes difficult for any other agent to track the data of its neighbors. This feature of the algorithm is necessary for any fair market operation.

3.4.3 Convergence and Stability Results

The computer specification used to implement the **Algorithm** 1 is Desktop PC with Intel Core i7 processor (3.6 GHz) 64 GB RAM. I investigate the convergence speed of our proposed algorithm using modal analysis. Based on the state matrix $\mathbf{A}$, I first calculated the value of gain parameter ρ such that the spectral radius of the matrix is unity. The value of ρ was further decreased in small step by looking at the second largest eigenvalue γ_2. I am interested in the fixed point stability of state variables $\boldsymbol{x}$, and the oscillation of the dynamics depend on γ_2. Figure 3.7 (a) shows the plot of absolute value of γ_2 for different ρs from 0 to critical value. Recall that critical value of ρ corresponds to the border of stability zone. In the plot, red color shows the critical eigenvalue and yellow shows the pri-critical value. The value in green is obtained using time domain simulation in MATLAB Simulink, where the oscillation of λ for given ρ is looked for any disturbance, and final value is obtained for 2% of oscillation of market price around its steady value. The value is blue corresponds to the minimum of second largest eigenvalues for the range of ρs. I see in Figure 3.7 (a) that $|\gamma_2|$ decreases at first with decreasing ρs, and diverges from a knee point in blue. Finally, I randomly select a point in purple that is for lower value of ρ than the optimum.

In order to illustrate the effect of different ρs, Figure 3.7 (b) is plotted. It shows all $3N$ eigenvalues for each value of ρ shown in the same purple, blue, green, yellow, and red color. Eigenplot shows that higher the value of ρ, farther the points go from real-axis. There is always one eigenvalue that has unity real part, which in fact takes care of the stability at fixed point.

Remark 1.: The algorithm has its equivalence with the dynamic system in control theory. The mismatch gain ρ corresponds to the proportional controller, which is exactly what I can see in

Figure 3.7 (b); with the larger value of proportional gain , higher will be the overshoot and hence oscillations in the system. The update of market price λ corresponds to integral controller, where it is adding all the errors set from the optimum point.

Figure 3.8 shows the convergence process for different residuals. First I define the following residuals:

$$\Delta P = \frac{\left\|\boldsymbol{p}(k) - \boldsymbol{p}^*\right\|_2}{\left\|\boldsymbol{p}(0) - \boldsymbol{p}^*\right\|_2}; \qquad \Delta\lambda = \frac{\left\|\boldsymbol{\lambda}(k) - \boldsymbol{\lambda}^*\right\|_2}{\left\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}^*\right\|_2}; \quad \sum \Delta P = \frac{\left\| \sum_{i\in\mathcal{V}} \left(p_i(k) - p_i{}^*\right)\right\|_2}{\left\| \sum_{i\in\mathcal{V}} \left(p_i(k) - p_i{}^*\right)\right\|_2};$$

Where ΔP is the residual of primal iterates, $\Delta\lambda$ is the residual of dual iterates, and $\sum \Delta P$ is the residual of power mismatch.

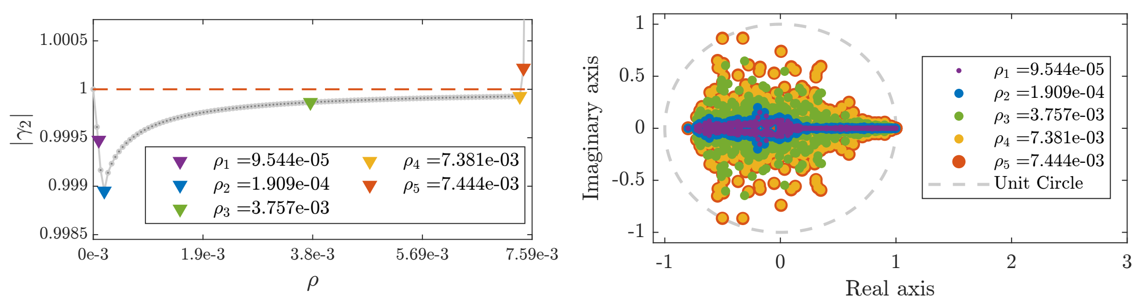

Figure 3.7 Results for IEEE 300-bus case. (a) the locus illustrates the relationship between various values of ρ, admitted in the system matrix (3.28), and the matrix's second largest eigenvalue; (b) displays, on the imaginary plane, the set of eigenvalues that were obtained for each value of ρ shown in the legend

Figure 3.8 (a), (b), (c) show the convergence for different ρs depicted in Figure 3.7. Figure 3.8 (a) shows the convergence of primal iterates with iterations (time). It is seen that the ρ corresponding to lowest of the second largest eigenvalues in blue color takes any initialization of $\boldsymbol{p}$ to

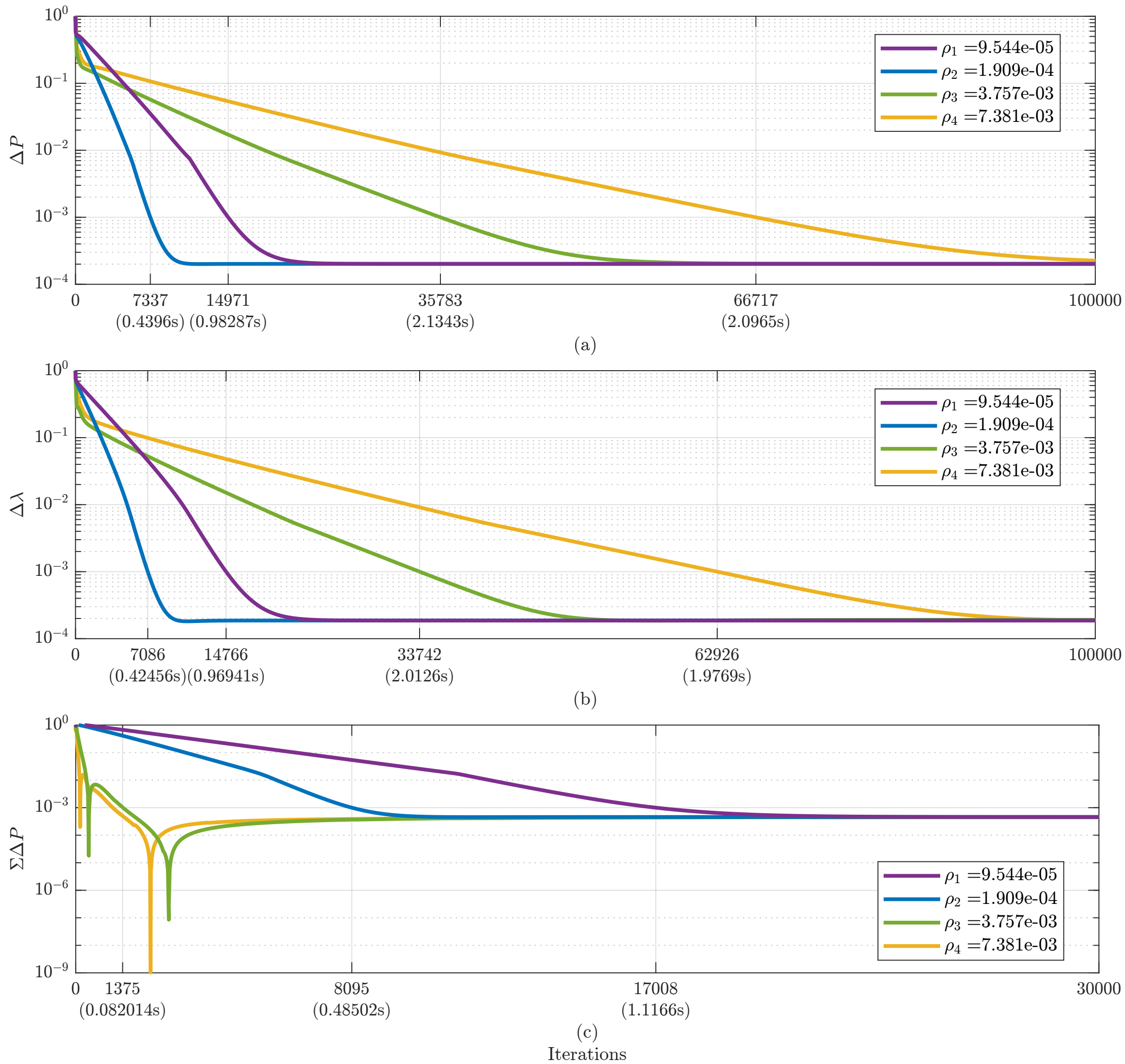

Figure 3.8 Results for IEEE 300-bus case. Based on different values of ρ, the figure displays the convergence process for: (a) the primal residual (R_{SG}); (b) displays the convergence process for the dual residual (S_{SG}); (c) displays the convergence process for the total power mismatch

the fixed optimal solution faster than any other values. We see that for 300 bus network, $\boldsymbol{p}$ settles in less than half a second. Similar convergence is displayed in Figure 3.8 (b) for dual iterates $\boldsymbol{\lambda}$. Figure 3.8 (c), on the other hand, shows the ρ corresponding to green has faster convergence. This shows that the mismatch of power in the network can be cleared faster with higher gain parameter, while compromising the time for the agents to reach their optimal primal and dual points. Note that with all the choices of ρ, the algorithm will eventually find the optimal solution points.

Remark 2: In the hierarchy of control, stability of power system always comes at first to the economics of operation. Any power mismatch is the network is first responded by the inertia, primary control, secondary control, and tertiary control (economic dispatch) in the same order as the name suggests. This motivated us to look for the gain parameter in green using time domain simulation.

3.5 Summary

In this chapter, a fully distributed algorithm is proposed for economic dispatch problem where all agents' computations are just based on their own data and their neighbors' shared information. The algorithm employs dual decomposition and dynamic average consensus algorithms to develop the update procedures. Use of dynamic average consensus has enabled the algorithm to track time varying constraint set of the optimization problem, which in fact translates to tracking of demand change in the network. Modal analysis is presented to speed up the convergence to a fixed point solution. Performance of the proposed solution including convergence and computation speed are tested against different IEEE 300 bus test case. Simulations demonstrate promising results for the algorithm to solve real-time economic dispatch problems with dynamic load variations. It would be interesting to study the communication imperfection in data transfer as the future work. It is assumed that the communication network is perfect ,but time delays and packet drops are ubiquitous. In addition, an optimization problem (semi definite programming) to calculate the optimum gain is an interesting work to pursue in the future.

CHAPTER 4

DISTRIBUTED ECONOMIC DISPATCH VIA ADMM

4.1 Introduction

4.1.1 Motivation and Related Work

Future electric grid will likely be dominated by small-scale distributed energy resources (DER) mostly comprising of variable energy resource (VRE), which are characterized by their unpredictability and variability [43–45]. This necessitates new ways to control, monitor, and optimize resources. With the proliferation of DERs and their participation at the grid-edge, a central entity to dictate the decision to all the players of the market to allocate resources is simply impractical given the unprecedented size of data and the challenges involved in computation and communication [27, 46]. Against this backdrop, distributed algorithms that would dynamically adapt to real-time changes are projected as a promising solution for economic dispatch [28, 32, 47, 48]. More, high penetrations of VRE in the future power grids leads to dynamically changing net power demand on the grid, which requires online (real-time) solutions for its optimal operation–e.g. *online economic dispatch*–without a need for initialization of the optimization process.

Inspired by the multi-agent systems from control theory [25] and a seminal work on distributed computation [24], a plethora of papers have investigated the relevance of decentralized and distributed algorithms for economic dispatch including analytical target cascading (ATC), proximal message passing (PMP) [26], auxiliary proximal message passing (APMP) [49], and Alternating

Direction Method of Multipliers (ADMM) [21, 50, 51]. ADMM algorithm is widely reported in the literature due to its capabilities to handle large-scale problems, and its $O(1/k)$ convergence rate [52]. The fundamental idea of this optimization method is that the central coordinator gathers the information including decision variables from all agents to update the global dual variable, and broadcasts necessary information back to all the agents. The central coordinator adjusts the dual variable until the market is cleared, and optimum resource is allocated. An application of this pristine ADMM algorithm with model predictive control is used in [28] to control and schedule DERs in real-time in a micro-grid. This and all the decentralized algorithms, however, are vulnerable to single point failure, and even might have data-privacy concerns.

Different from decentralized algorithms, distributed algorithms are designed to remove the need for a coordinator (master node) by letting each agent communicate only with its directly connected neighbors to compute its optimum decision variables [53]. Gradient and sub-gradient based algorithms are well established distributed algorithms [54]. Gradient-based algorithm is proposed in [6] to solve economic dispatch problem, but with an additional agent to communicate with all the agents in order to ensure power-balance of the network. A distributed subgradient based algorithm is introduced in [5], where frequency is used as a proxy for power balance to coordinate among renewable generations. Besides, [29, 55] exploit primal-dual algorithm to implement distributed economic dispatch.

A consensus based ADMM is proposed in [35, 36] to solve distributed economic dispatch problem in a microgrid. An average consensus algorithm and projection methods are exploited to realize a distributed ADMM by seeking the consensus of the primal variables, *i.e.* the agents' power generation. While both of these algorithms are effective for a constant demand, none of them is practical for real-time implementation in the grid with high penetration of variable energy resources as they cannot capture the dynamics of real-time changes in demand and renewable resources. Besides, [20] identified the need of distributed algorithms to solve the online convex optimization problem with time-varying constraint sets.

The convergence rate of (sub)gradient and ADMM algorithm have been studied. Authors in [56] investigate the convergence of decentralized gradient descent algorithm for a proper closed convex function with Lipschitz gradient. The paper shows that for a strongly convex problem, solution converge to the global minimizer at a linear rate. Similarly, the convergence rate of ADMM algorithm for different functions have been explored given the popularity of this algorithm to solve large scale optimization problem. A Survey paper [21] explicitly states that the algorithm is slow to converge to high accuracy compared to its counterparts, but converges to modest accuracy within a few iterations. Authors in [57] study the convergence of ADMM for strongly convex and have Lipschitz continuous gradient, and show that the sequence generated by the algorithm converges linearly to the optimal solution. They further show that the convergence depend on the connectivity of communication matrix and the weights associated with the each edge of the graph. Authors in [58] add to the conclusion on [57] and demonstrate with the simulation results that average consensus is faster in convergence for highly connected graph, quantified in terms of spectral radius of graph.

This chapter proposes a completely distributed algorithm called distributed Alternating Direction Method of Multipliers (D-ADMM) with an application in *online economic dispatch*, where each agent updates its time-varying local estimate of the average of power mismatch using *dynamic average consensus* algorithm [9, 14], and updates its dual variable via a combination of average consensus and subgradient step. Based on the updated information, all agents find their optimal decision projected onto its local constraint set from its augmented *Lagrangian* objective function. Preliminary result was published by the author in [59]. The proposed D-ADMM algorithm possess the following five desirable properties:

i. *Distributed Consensus*: It ensures Consensus on average power-mismatch and market price via a purely distributed setting—as opposed to one central coordinator calculating and sharing these global values with all agents.

ii. *Optimality*: The optimal solution sought is proven optimal by leveraging the KKT conditions to show its primal and dual feasibility.

iii. *Stability and Convergence*: Its stability and convergence to the optimal solution are guaranteed via eigensystem analysis. In addition, D-ADMM is compared with D-Subgradient in terms of convergence speed.

iv. *Scalability*: It fully decomposes the optimization problem between agents, and its computational complexity does not grow with the network size; its scalability is demonstrated by testing it against IEEE 1354-bus network.

v. *Data Privacy*: It ensures minimum sharing of data between agents as it lets agents communicate their power mismatch with their neighbors only—rather than sharing their generation and demand data with all the agents or a central coordinator.

4.1.2 Notation

Let $\mathcal{G}$ denote a graph with the set of vertices $\mathcal{V} = \{1, \cdots, N\}$ and the edges $\mathcal{E} \subseteq \mathcal{V} \times V$; $\mathcal{G}$ is restricted to *simple, undirected* graph without multi-edges and self-edges. For a set $\mathcal{V}$, its cardinality is represented by $|\mathcal{V}|$. I define $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E} \text{ and } j \neq i\}$ as the set of all the neighbors of the agent i and $d_i = |\mathcal{N}_i|$ as degree of any node i; $d_i = |\{j : (i, j) \in \mathcal{E}\}|$. $\mathbb{R}$ denotes the set of all real numbers, $\mathbb{0} \in \mathbb{R}^{N \times N}$ is a null matrix, $\boldsymbol{I} \in \mathbb{R}^{N \times N}$ is an identity matrix, $\mathbf{1}_N \in \mathbb{R}^{N \times N}$ is a unit vector, and γ is the eigenvalue. $(\cdot)^T$ is the transpose. DED stands for distributed economic dispatch, SG stands for subgradient, and ADMM is Alternating Direction Method of Multipliers. Matrices and vectors are written in bold throughout the thesis.

4.2 Distributed Economic Dispatch

Economic Dispatch is the determination of the optimal output of each generator based on the marginal cost of production in order to meet the demand of the network.

Let each agent $i \in \mathcal{V} = \{1, \ldots, N\}$ be the bus of the power network. The economic dispatch is, then, modeled as

$$\min_{\boldsymbol{p_g^k}} \sum_{i \in \mathcal{V}} C_i(p_{gi}^k) \tag{4.1a}$$

$$\text{s.t} \sum_{i \in \mathcal{V}} p_{gi}^k = \sum_{i \in \mathcal{V}} p_{di}^k \tag{4.1b}$$

$$p_{gi}^{\min} \leq p_{gi}^k \leq p_{gi}^{\max} \qquad \forall i \in \mathcal{V} \tag{4.1c}$$

where $\boldsymbol{p_g^k} = \{p_{gi}^k, \ldots, p_{gN}^k\}$ denoting the power generation of all buses at the time-step k is the set of decision variables; (4.1a) represents the objective function as minimization of the total generation cost at any time-step k with $C_i(\cdot)|\mathbb{R} \to \mathbb{R}$ equal to the generation cost function of agent i; (4.1b) represents the power balance constraint, with p_{di} denoting the demand at bus i, to ensure the network's total generation and total demand are equal at any time; (4.1c) guarantees all generations operate within the minimum and maximum generation capacities at any time.

The optimization problem (4.1) has local objective function (4.1a) and local constraint (4.1c), coupled by the global constraint (4.1b) on the network's total generation and total demand. In this chapter, I aim to model problem (4.1) as a completely distributed optimization problem that allow each agent to determine the value of its decision variables $p_{gi} \in \boldsymbol{p_g}$.

4.2.1 Economic Dispatch Problem in a Decentralized Setting

Let us define agent i's power mismatch by $p_i = p_{gi} - p_{di}$ and its feasible set by

$$\Omega_i = [p_i^{\min}, p_i^{\max}] = [p_{gi}^{\min} - p_{d_i}, p_{gi}^{\max} - p_{d_i}]$$

Substituting $p_{gi} = p_i + p_{di}$ in the cost function $C_i(p_{gi})$ defined in the problem formulation (4.1), $C_i(p_{gi})$ modifies to $f_i(p_i)$ where

$$f_i(p_i) = a_i p_i^2 + (2a_i p_{di} + b_i)p_i + a_i p_{di}^2 + b_i p_{di} + c \tag{4.2}$$

Then, (4.1) can be written as

$$\min_{\boldsymbol{\Omega}} \quad \sum_{i \in \mathcal{V}} f_i(p_i) \tag{4.3a}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{V}} p_i = 0 \tag{4.3b}$$

where $\boldsymbol{\Omega} = [\Omega_1; \cdots; \Omega_N] \in \mathbb{R}^N$ is the set of the decision variable. The problem (4.3) is as an exchange ADMM problem as detailed in [21] in which each agent participates in the competitive market and the market equilibrium is reached.

Another canonical form of the exchange problem (4.3) is the *sharing problem* [21, 60]

$$\min_{\boldsymbol{\Omega}} \quad f(\boldsymbol{p}) + g(\boldsymbol{z}) \quad \text{s.t.} \quad p_i = z_i; \quad i \in \mathcal{V} \tag{4.4}$$

where $\boldsymbol{p} = [p_1; \cdots; p_N]$, $\boldsymbol{z} = [z_1; \cdots; z_N]$ is an auxiliary vector,$g(\boldsymbol{z})$ is a shared objective term defined as

$$g(\boldsymbol{z}) = \begin{cases} 0 & \text{if } \sum_{i \in \mathcal{V}} z_i = 0 \\ \infty & \text{otherwise} \end{cases} \tag{4.5}$$

and $f = \sum_{i \in \mathcal{V}} f_i : \mathbb{R}^N$ is the total cost function.

Assumption 1 *The function $f : \mathbb{R}^N \to \mathbb{R} \bigcup \{+\infty\}$ is closed, proper, and convex.*

I consider the *augmented Lagrangian $\mathcal{L}_\rho$* associated with (4.4):

$$\mathcal{L}_\rho(p, z, \lambda) = f(\boldsymbol{p}) + g(\boldsymbol{z}) - \lambda^T(p - z) + \frac{\rho}{2}\left\|p - z\right\|_2^2 \tag{4.6}$$

where $\rho > 0$ is the penalty parameter, $\|\cdot\|_2$ is the usual euclidean norm and $\boldsymbol{\lambda} = [\lambda_1; \cdots; \lambda_N]$ is a vector of dual variables associated with equality constraints.

A classic ADMM runs the iterations in the following sequence

$$p_i^{k+1} = \underset{\Omega_i}{\operatorname{argmin}}\, f_i(p_i) - \lambda_i^k p_i + \frac{\rho}{2}\left\|p_i - z_i^k\right\|_2^2 \tag{4.7}$$

$$\boldsymbol{z}^{k+1} = \underset{\boldsymbol{z}}{\operatorname{argmin}}\, g(\boldsymbol{z}) + \boldsymbol{\lambda}^{k^T}\boldsymbol{z} + \frac{\rho}{2}\left\|\boldsymbol{p}^{k+1} - \boldsymbol{z}\right\|_2^2 \tag{4.8}$$

$$\lambda_i^{k+1} = \lambda_i^k - \rho\left(p_i^{k+1} - z_i^{k+1}\right) \tag{4.9}$$

Based on definition of $g(z)$ in (4.5), optimization problem (4.8) can be formulated as

$$\min_{z} \quad \sum_{i\in\mathcal{V}} \left\{\lambda_i^{k^T} z_i + \frac{\rho}{2}\left\|p_i^{k+1} - z_i\right\|_2^2\right\} \tag{4.10a}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{V}} z_i = 0; \tag{4.10b}$$

Correspondingly, the *Lagrangian* of (4.10) is equal to

$$\sum_{i\in\mathcal{V}} \mathcal{L}(z_i, y) = \sum_{i\in\mathcal{V}} \left\{\lambda_i^{k^T} z_i + \frac{\rho}{2}\left\|p_i^{k+1} - z_i\right\|_2^2 + y z_i\right\} \tag{4.11}$$

where y is the Lagrangian variable.

Proposition 1 *For any convex optimization problem with diffferentiable objective and constraint functions, any points that satisfy the* ***Karush–Kuhn–Tucker*** *conditions are primal and dual optimal and have zero duality gap [18].*

Thus, using the stationary condition i.e., $\Delta\mathcal{L}(z_i, y) = 0$ of KKT conditions:

$$\Delta_{zi}\mathcal{L}(z_i, y) = 0 \rightarrow z_i = p_i^{k+1} - \frac{\lambda_i^k}{\rho} - \frac{y}{\rho} \tag{4.12}$$

$$\Delta_{y_i}\mathcal{L}(z_i, y) = 0 \rightarrow \sum_{i \in \mathcal{V}} z_i = 0 \tag{4.13}$$

Solving (4.12) and (4.13) yields:

$$y = \rho\left(\overline{p}^{k+1} - \frac{\overline{\lambda}^k}{\rho}\right) \tag{4.14}$$

where $\overline{p}^{k+1} = 1/N \sum_{i \in \mathcal{V}} p_i^{k+1}$ is the network's average mismatch of power and $\overline{\lambda}^k = 1/N \sum_{i \in \mathcal{V}} \lambda_i^k$ is the network's average price of generations.

Substituting (4.14) in (4.12) leads to

$$z_i^{k+1} = p_i^{k+1} - \overline{p}^{k+1} - \frac{{\lambda_i}^k}{\rho} + \frac{\overline{\lambda}^k}{\rho} \tag{4.15}$$

and substituting the gradient term in (4.9) with (4.15) gives

$${\lambda_i}^{k+1} = {\lambda_i}^k - \rho\overline{p}^{k+1} + \overline{\lambda}^k - \lambda_i^k = \overline{\lambda}^k - \rho\overline{p}^{k+1} \tag{4.16}$$

which implies that all the individual prices of the agents are in fact in consensus as the average market price and average mismatch of power are constant terms. In other words, $\lambda_i = \lambda = \overline{\lambda} \quad \forall i \in \mathcal{V}$.

Consequently, (4.15) can be written as

$$z_i^{k+1} = p_i^{k+1} - \overline{p}^{k+1} \tag{4.17}$$

and substituting (4.17) on (4.7), the decentralized ADMM algorithm can be summarized as:

$$p_i^{k+1} := \underset{\Omega_i}{\text{argmin}} \left(f_i(p_i) - \lambda^k p_i + \frac{\rho}{2} \left\| p_i - \left(p_i^k - \overline{p}^k \right) \right\|_2^2 \right) \tag{4.18}$$

$$\lambda^{k+1} := \lambda^k - \rho \overline{p}^{k+1} \tag{4.19}$$

In this *exchange ADMM* algorithm, the central coordinator receives the average mismatch of power from all the agents, which then adjusts the market price to ensure the supply-demand balance of the network. From the semantics of optimization, in (4.19), λ is constantly pulling the decision variables $p_i \in \boldsymbol{p}$ toward the optimal value projected onto the feasible space Ω_i. Each agent, then, updates their p_i independently based on the broadcasted values λ and $\overline{p}$ from the central coordinator.

4.2.2 Economic Dispatch Problem in a Distributed Setting

I am interested in a platform which lacks any coordinator that has access to all the agents' information and the agents communicate exclusively with their directly connected neighbors. In order to make the *exchange ADMM* algorithm summarized in (4.18) and (4.19) completely distributed, (i) each agent must be able to obtain a meaningful estimate of the network average power mismatch $\overline{p}$ and the dual variable λ by communicating with its neighboring agents, and (ii) all the agents must reach consensus about their estimates. Let us denote these estimates for any agent i by $\overline{p}_i$ and λ_i, respectively.

To achieve this, I use (2.15) to leverage the concept of dynamic average consensus discussed in chapter 2 to convert these global variables to local ones. Localization of these two parameters are achieved by

$$\overline{p}_i^{k+1} = W_i^T \overline{\boldsymbol{p}}^k + (p_i^{k+1} - p_i^k) \tag{4.20}$$

$$\lambda_i^{k+1} = W_i^T \boldsymbol{\lambda}^k - \rho \overline{p}_i^{k+1} \tag{4.21}$$

with $\overline{\boldsymbol{p}} = [\overline{p}_1; \cdots; \overline{p}_N] \in \mathbb{R}^N$, $\boldsymbol{\lambda} = [\lambda_1; \cdots; \lambda_N] \in \mathbb{R}^N$, and where $W_i \in \mathbb{R}^{N\times 1}$ is the vector of agent i's update weights as defined for (2.8).

Note that although each agent communicates its own estimate of the network power mismatch which might include some information about its own local power mismatch, they never communicate their actual generation and demand data with any other agents. This makes it difficult for any agent to track the private information–*e.g.* the generation cost coefficients–of any other agents, which leads to a higher level of *preservation of data privacy*.

Using (4.20) and (4.21), every agent can estimate the average power mismatch power and the dual variable by communicating exclusively with its directly connected neighbors. A fully distributed form of a decentralized ADMM in (4.18) becomes

$$\mathcal{L}_\rho(p_i, \lambda_i) = f_i(p_i) - \lambda_i^k p_i + \frac{\rho}{2} \left\| p_i - \left(p_i^k - \overline{p_i}^k \right) \right\|_2^2 \tag{4.22}$$

and the distributed ADMM algorithm of all the agents in the network can be summarized in vector format network as

$$
\begin{cases}
\boldsymbol{p}^{k+1} \in \operatorname{argmin}_{\Omega} \left\{ \boldsymbol{f}(\boldsymbol{p}) - \boldsymbol{\lambda}^{k^T} \boldsymbol{p} + \frac{\rho}{2} \left\| \boldsymbol{p} - \left(\boldsymbol{p}^k - \overline{\boldsymbol{p}}^k \right) \right\|_2^2 \right\} \\
\overline{\boldsymbol{p}}^{k+1} = \boldsymbol{W} \overline{\boldsymbol{p}}^k + (\boldsymbol{p}^{k+1} - \boldsymbol{p}^k) \\
\boldsymbol{\lambda}^{k+1} = \boldsymbol{W} \boldsymbol{\lambda}^k - \rho \overline{\boldsymbol{p}}^{k+1}
\end{cases}
\tag{4.23}
$$

where $\boldsymbol{f}(\boldsymbol{p}) = \left[f_1(p_1); \cdots ; f_N(p_N)\right] \in \mathbb{R}^N \to \mathbb{R} \bigcup \{+\infty\}$ and $\boldsymbol{\lambda} = [\lambda_1; \cdots ; \lambda_N] \in \mathbb{R}^N$.

4.2.3 Distributed Economic Dispatch Algorithm

The distributed ADMM (D-ADMM) now can be designed to solve in an iterative procedure. At each iteration, each agent receives the information on price and average power mismatch from its connected neighbors $\mathcal{N}_i$, and finds the optimum value of its primal variable. It then updates its price and the estimate of network's power mismatch and broadcasts to its connected neighbors.

Algorithm 2: Distributed ADMM Algorithm

Input: cost-coefficients $\{\mathbf{a},\mathbf{b}\}$, and penalty parameter ρ

Initialization: Let $\boldsymbol{\lambda}^0 = \mathbb{0}_N, \boldsymbol{p}^0 = \overline{\boldsymbol{p}}^0 = -\boldsymbol{p_d}^0$

for $k = 0$ *to* ∞ **do**

- Receive $\overline{\boldsymbol{p}}$ and $\boldsymbol{\lambda}$. Each agent receives data from its neighboring agents only.
- Compute power mismatch $\boldsymbol{p}^{k+1}$ as:

$$
\boldsymbol{p}^{k+1} = \left[\boldsymbol{\Lambda} \boldsymbol{\lambda}^k + \rho \boldsymbol{\Lambda} \boldsymbol{p}^k - \rho \boldsymbol{\Lambda} \overline{\boldsymbol{p}}^k - (2 \boldsymbol{a} \boldsymbol{\Lambda} + \boldsymbol{\psi}) \boldsymbol{p}_d^k - \boldsymbol{b} \boldsymbol{\Lambda} \right]_{\boldsymbol{p}^{min}}^{\boldsymbol{p}^{max}}
$$

- Update $\overline{\boldsymbol{p}}^{k+1} = \boldsymbol{W} \overline{\boldsymbol{p}}^k + (\boldsymbol{p}^{k+1} - \boldsymbol{p}^k)$
- Update $\boldsymbol{\lambda}^{k+1} = \boldsymbol{W} \boldsymbol{\lambda}^k - \rho \overline{\boldsymbol{p}}^{k+1}$
- Broadcast $\overline{\boldsymbol{p}}^{k+1}$, $\boldsymbol{\lambda}^{k+1}$. Each agent broadcasts its data to its neighboring agents only.

Differentiating (4.22) with respect to p_i leads to

$$\frac{\partial}{\partial p_i}\mathcal{L}_\rho(p_i,\lambda_i)=\frac{\partial f_i(p_i)}{\partial p_i}-\lambda_i^k+\rho(p_i-p_i^k+\overline{p_i}^k)=0 \tag{4.24}$$

in order for any agent i to define its optimal value of p_i at each iteration $k+1$. Letting $\mathcal{X}_g\subset\mathcal{V}$ be the set of buses with the generating units, (4.24) for any agent $i\in\mathcal{X}_g$ leads to

$$p_i^{k+1}=\frac{\lambda_i^k+\rho p_i^k-\rho\overline{p_i}^k-2a_ip_{di}^k-b_i}{2a_i+\rho} \tag{4.25}$$

For any agent $i\notin\mathcal{X}_g$, the optimal value of p_i is simply

$$p_i^{k+1}=p_{gi}^{k+1}-p_{di}^{k+1}=-p_{di}^{k+1} \tag{4.26}$$

For a more concise presentation, (4.25) and (4.26) is written in a generalized form of

$$\boldsymbol{p}^{k+1}=\boldsymbol{\Lambda}\boldsymbol{\lambda}^k+\rho\boldsymbol{\Lambda}\boldsymbol{p}^k-\rho\boldsymbol{\Lambda}\overline{\boldsymbol{p}}^k-(2\boldsymbol{a}\boldsymbol{\Lambda}+\boldsymbol{\psi})\boldsymbol{p}_d^k-\boldsymbol{b}\boldsymbol{\Lambda} \tag{4.27}$$

where $\boldsymbol{a}=\text{diag}\{a_i\}$, $\boldsymbol{b}=\text{diag}\{b_i\}$, $\boldsymbol{\Lambda}=\text{diag}\{1/(2a_i+\rho)\}$, and $\boldsymbol{\psi}=\text{diag}\{\psi_i\}$. If $i\notin\mathcal{X}_g$, $\psi_i=1$ and $a_i=b_i=\Lambda_i=0$; otherwise, $\psi_i=0$.

Algorithm 2 elaborates the iterative processes for the distributed ADMM method proposed for the economic dispatch problem. Note that the update processes are presented in vector format and preserve the distributed characteristics of the .

4.2.4 Optimality Conditions

This section sheds light on the optimality of the proposed distributed ADMM algorithm in (4.23). I make the following assumption on the optimization problem (4.3).

Assumption 2 *Any saddle point $(\boldsymbol{p}^*, \lambda^*)$ of $\mathcal{L}_0$ is a pair of primal and Lagrangian dual optimal solutions. This assumption, which is a standard result on Lagrangian duality, implies that*

$$\mathcal{L}_0(\boldsymbol{p}^*, \lambda^*) = \inf_{\boldsymbol{p} \in \boldsymbol{\Omega}} \mathcal{L}_0(p, \lambda^*); \quad \mathcal{L}_0(\boldsymbol{p}^*, \lambda^*) = \sup_{\lambda} \mathcal{L}_0(\boldsymbol{p}^*, \lambda)$$

where $\mathcal{L}_0$ is the unaugmented Lagrangian.

Theorem 4 *The necessary and sufficient optimality conditions for the D-ADMM are primal and dual feasibility.*

Proof: If the optimization problem (4.23) is primal and dual feasible, the optimality of the problem is guaranteed.

(a) Primal Feasibility: The equality constraint (4.3b) in a generalized form $Ax = b$ is:

$$p_i + \sum_{j \in \mathcal{V}, j \neq i} p_j = 0; \quad Ap_i = -\sum_{j \in \mathcal{V}, j \neq i} p_j; \quad A = 1 \tag{4.28}$$

Using Farkas Lemma [17], only one of the two alternatives holds

1. $\exists\, p_i \geq 0$ such that $Ap_i = -\sum_{j \in \mathcal{V}, j \neq i} p_j$

2. $\exists\, \mu$ such that $\mu^T A \geq 0$ and $\mu^T(-\sum_{j \in \mathcal{V}, j \neq i} p_j) < 0$

if (1) is true, then suppose $\mu^T A \geq 0$, then using (1), $\mu^T A p_i = -\mu^T \sum_{j \in \mathcal{V}, j \neq i} p_j \geq 0$ which shows that (2) cannot be true.

(b) Dual Feasibility: Here, I show that the optimization problem in (4.23) is **dual feasible**, that is, $0 \in \partial f(\boldsymbol{p}^*) - \boldsymbol{\lambda}^*$. Applying Stationarity of **Karush–Kuhn–Tucker** conditions to the augmented Lagrangian $\mathcal{L}_\rho$ (4.22) for agent i; that is

$$\Delta_p \mathcal{L}_\rho(p, \lambda) = 0 \in \partial f_i(p_i)^{k+1} - \lambda_i^k + \rho(p_i^{k+1} - p_i^k + \overline{p_i}^k)$$

Correspondingly,

$$0 \in \partial f_i(p_i)^{k+1} - \lambda_i^{k+1} + \lambda_i^{k+1} - \lambda_i^k + \rho(p_i^{k+1} - p_i^k + \overline{p_i}^k)$$

Substituting λ_i^{k+1} from (4.21),

$$0 \in \partial f_i(p_i)^{k+1} - \lambda_i^{k+1} + W_i^T \boldsymbol{\lambda}^k - \rho \overline{p_i}^{k+1} - \lambda_i^k + \rho(p_i^{k+1} - p_i^k + \overline{p_i}^k)$$

Substituting $\overline{p_i}^{k+1}$ from (4.20) and arranging,

$$0 \in \partial f_i(p_i)^{k+1} - \lambda_i^{k+1} - (\lambda_i^k - W_i^T \boldsymbol{\lambda}^k) + \rho(p_i^{k+1} - p_i^k + \overline{p_i}^k - W_i^T \overline{\boldsymbol{p}}^k - p_i^{k+1} + p_i^k)$$

which leads to the dual feasibility condition for agent i

$$(\lambda_i^k - W_i^T \boldsymbol{\lambda}^k) - \rho(\overline{p_i}^k - W_i^T \overline{\boldsymbol{p}}^k) \in \partial f_i(p_i)^{k+1} - \lambda_i^{k+1} \tag{4.29}$$

where left hand side quantity is the dual residual. For the complete network, the dual residual S can be written in vector format as:

$$\boldsymbol{S}^{k+1} = (\boldsymbol{I} - \boldsymbol{W})\boldsymbol{\lambda}^k - \rho(\boldsymbol{I} - \boldsymbol{W})\overline{\boldsymbol{p}}^k \tag{4.30}$$

where $\boldsymbol{S} \in \mathbb{R}^{\mathbb{N}}$ is the dual residual at iteration $k+1$. The matrix $(\boldsymbol{I} - \boldsymbol{W})$ is a Laplacian matrix and $\boldsymbol{S}$ converges to zero as soon as all agents reach consensus on $\boldsymbol{\lambda}$ and $\overline{\boldsymbol{p}}$. Recall that, under Lemma 1-2, the consensus on average mismatches and on market prices is guaranteed because of the conditions imposed for the choice of W as detailed in (2.12).

4.3 Numerical Stability and Convergence

The optimization problem in (4.23) is modeled as a discrete-time linear system to study the stability of our proposed D-ADMM algorithm. The state-space representation of a discrete-dynamic system is

$$\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k + \boldsymbol{B}\boldsymbol{u}^k = \boldsymbol{f}(\boldsymbol{x}^k, \boldsymbol{u}^k) \tag{4.31}$$

$$\boldsymbol{y}^{k+1} = \boldsymbol{C}\boldsymbol{x}^k + \boldsymbol{D}\boldsymbol{u}^k = \boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{u}^k) \tag{4.32}$$

where $\boldsymbol{x}(\cdot)$ is a state vector, and $\boldsymbol{u}(\cdot)$ is an input vector. Recall that the natural response of state equation (4.31) is

$$\boldsymbol{x}^k = \boldsymbol{A}^k\boldsymbol{x}(0) \tag{4.33}$$

where $\boldsymbol{x}(\mathbf{0})$ is the initial condition. This means the stability of the system exclusively depends on $\boldsymbol{A}$, which motivates modeling of D-ADMM algorithm as a discrete state-space model. With reference to (4.23) and (4.27), one can formulate the state equations of the proposed model as

$$\boldsymbol{p}^{k+1} = \boldsymbol{\Lambda}\boldsymbol{\lambda}^k + \rho\boldsymbol{\Lambda}\boldsymbol{p}^k - \rho\boldsymbol{\Lambda}\overline{\boldsymbol{p}}^k - (2\boldsymbol{a}\boldsymbol{\Lambda} + \boldsymbol{\psi})\boldsymbol{p}_d^k - \boldsymbol{b}\boldsymbol{\Lambda} \tag{4.34}$$

$$\overline{\boldsymbol{p}}^{k+1} = \boldsymbol{W}\overline{\boldsymbol{p}}^k + (\boldsymbol{p}^{k+1} - \boldsymbol{p}^k)$$

$$= \boldsymbol{W}\overline{\boldsymbol{p}}^k + (\boldsymbol{\Lambda}\boldsymbol{\lambda}^k + \rho\boldsymbol{\Lambda}\boldsymbol{p}^k - \rho\boldsymbol{\Lambda}\overline{\boldsymbol{p}}^k - (2\boldsymbol{a}\boldsymbol{\Lambda} + \boldsymbol{\psi})\boldsymbol{p}_d^k - \boldsymbol{b}\boldsymbol{\Lambda} - \boldsymbol{p}^k)$$

Correspondingly,

$$\overline{\boldsymbol{p}}^{k+1} = \boldsymbol{\Lambda}\boldsymbol{\lambda}^k + (\rho\boldsymbol{\Lambda} - \boldsymbol{I}_{N\times N})\boldsymbol{p}^k - (\rho\boldsymbol{\Lambda} - \boldsymbol{W})\overline{\boldsymbol{p}}^k$$
$$- (2\boldsymbol{a}\boldsymbol{\Lambda} + \boldsymbol{\psi})\boldsymbol{p}_d^k - \boldsymbol{b}\boldsymbol{\Lambda} \tag{4.35}$$

Finally, the state equation for price vector is

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{W}\boldsymbol{\lambda}^k - \rho\overline{\boldsymbol{p}}^{k+1}$$

Correspondingly,

$$\boldsymbol{\lambda}^{k+1} = (\boldsymbol{W} - \rho\boldsymbol{\Lambda})\boldsymbol{\lambda}^k + (\rho\boldsymbol{I}_{N\times N} - \rho^2\boldsymbol{\Lambda})\boldsymbol{p}^k - (\rho\boldsymbol{W} - \rho^2\boldsymbol{\Lambda})\overline{\boldsymbol{p}}^k$$
$$+ (2\rho\boldsymbol{a}\boldsymbol{\Lambda} + \rho\boldsymbol{\psi})\boldsymbol{p}_d^k + \rho\boldsymbol{b}\boldsymbol{\Lambda} \tag{4.36}$$

The state equations (4.3) can be written in matrix form as

$$\begin{bmatrix} \boldsymbol{p}^{k+1} \\ \bar{\boldsymbol{p}}^{k+1} \\ \boldsymbol{\lambda}^{k+1} \end{bmatrix} = \begin{bmatrix} \rho\boldsymbol{\Lambda} & -\rho\boldsymbol{\Lambda} & \boldsymbol{\Lambda} \\ (\rho\boldsymbol{\Lambda} - \boldsymbol{I}) & -(\rho\boldsymbol{\Lambda} - \boldsymbol{W}) & \boldsymbol{\Lambda} \\ (\rho\boldsymbol{I} - \rho^2\boldsymbol{\Lambda}) & -(\rho\boldsymbol{W} - \rho^2\boldsymbol{\Lambda}) & (\boldsymbol{W} - \rho\boldsymbol{\Lambda}) \end{bmatrix} \begin{bmatrix} \boldsymbol{p}^{k} \\ \bar{\boldsymbol{p}}^{k} \\ \boldsymbol{\lambda}^{k} \end{bmatrix} + \begin{bmatrix} -(2\boldsymbol{a}\boldsymbol{\Lambda} + \boldsymbol{\psi}) & -\boldsymbol{b}\boldsymbol{\Lambda} \\ -(2\boldsymbol{a}\boldsymbol{\Lambda} + \boldsymbol{\psi}) & -\boldsymbol{b}\boldsymbol{\Lambda} \\ (2\rho\boldsymbol{a}\boldsymbol{\Lambda} + \rho\boldsymbol{\psi}) & \rho\boldsymbol{b}\boldsymbol{\Lambda} \end{bmatrix} \begin{bmatrix} \mathbf{P_d}^{k} \\ \mathbf{1}_N \end{bmatrix} \tag{4.37}$$

with an output matrix equal to

$$\begin{bmatrix} \boldsymbol{\lambda}(k) \\ \mathbf{P_g}(k) \end{bmatrix} = \begin{bmatrix} \mathbb{0} & \mathbb{0} & \boldsymbol{I} \\ \boldsymbol{I} & \mathbb{0} & \mathbb{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}^{k} \\ \bar{\boldsymbol{p}}^{k} \\ \boldsymbol{\lambda}^{k} \end{bmatrix} + \begin{bmatrix} \mathbb{0} & \mathbb{0} \\ \boldsymbol{I} & \mathbb{0} \end{bmatrix} \begin{bmatrix} \mathbf{P_d}^{k} \\ \mathbf{1}_N \end{bmatrix} \tag{4.38}$$

Comparing (4.37) and (4.38) with the canonical form in (4.31), we see $\boldsymbol{A} \in \mathbb{R}^{3N\times 3N}$ is the state matrix, $\boldsymbol{B} \in \mathbb{R}^{3N\times 2N}$ is the input matrix, $\boldsymbol{C} \in \mathbb{R}^{2N\times 3N}$ is the output matrix, and $\boldsymbol{D} \in \mathbb{R}^{2N\times 2N}$ is the feed-forward matrix.

Theorem 5 *Let* $\gamma_1, \cdots, \gamma_m \quad m \leq n$ *be the eigenvalues of* $\boldsymbol{A} \in \mathbb{R}^{3N\times 3N}$. *The system* (4.31) *is [22]*

- *asymptotically stable iff* $|\gamma_i| < 1, \forall i = \{1, \cdots, m\}$
- *stable if* $|\gamma_i| \leq 1, \forall i = \{1, \cdots, m\}$
- *unstable if* $\exists$ *i such that* $|\gamma_i| > 1$

From theorem (5), we see that the spectral radius $\zeta(\boldsymbol{A})$ is detrimental to the stability of the iterative process. Besides, I am interested in the stability of non-zero fixed point for the linear iterative system. In other words, for the stability to a non-zero fixed point

$$\zeta(\boldsymbol{A}) = 1 \tag{4.39}$$

Also, the solution of the natural response (4.33) is:

$$\boldsymbol{x}(k) = \sum_{i=1}^{3N} c_i \gamma_i^k \xi_i \tag{4.40}$$

where c_i is the scalar prescribed by initial condition $\boldsymbol{x}(0)$, γ_i is the eigenvalue of $\boldsymbol{A}$, and ξ_i is linearly independent eigenvector. From (4.39) and (4.40), I can write

$$1 = \gamma_1 \geq |\gamma_2| \geq \cdots \geq |\gamma_{3N}| \tag{4.41}$$

Correspondingly,

$$\lim_{k \to \infty} \boldsymbol{x}(k) = c_1 \xi_1 \tag{4.42}$$

which can be interpreted as the state variables $\boldsymbol{x} \in \mathbb{R}^{3N}$ convergences to a fixed point that is the multiple of eigenvector ξ_1.

Corollary 2 *All solutions of linear iterative solution $\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k$ converges to a vector ξ that lies in the $\gamma_1 = 1$ eigenspace provided* (4.39) *holds true. Moreover, the rate of convergence of the solution is governed by the modulus $|\gamma_c|$ of the subdominant eigenvalue. [23].*

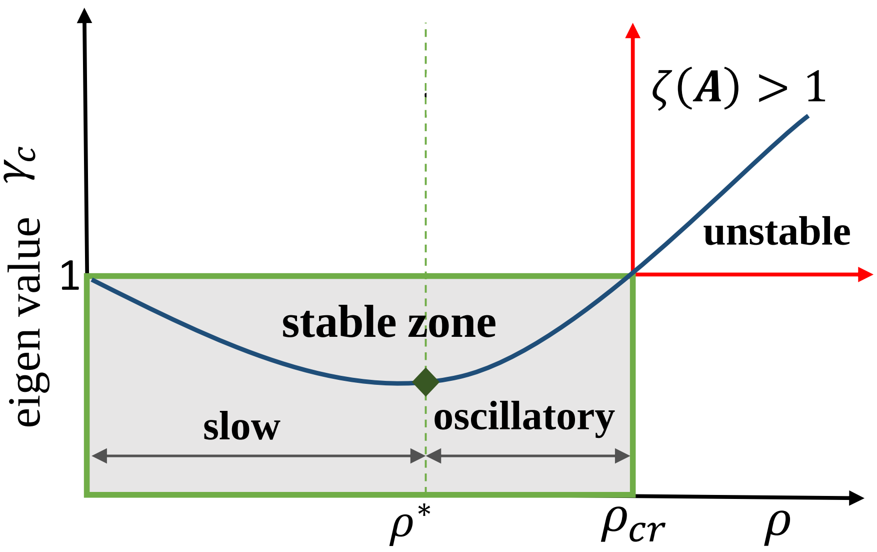

Figure 4.1 Plot of of γs for different control gains ρ; $\gamma_c = \max|\gamma_j|, j \notin i; |\gamma_i| = 1$

With corollary 2, also note that the smaller the $|\gamma_c|$, the faster the convergence. The penalty parameter ρ is equivalent to the step size in gradient methods and proportional controller in control literature. A close observation on $\boldsymbol{A}$ reveals that for a given network , ρ is the governing parameter for the stability and convergence of the iterative system. The value of ρ is computed for $\zeta(\boldsymbol{A}) = 1$ is the critical value ρ_{cr} beyond which the system is unstable. The reduction of ρ would decrease the subdominant eigenvalue $|\gamma_c|$, which corresponds to faster convergence of the fixed points. It is because the behavior of the system was stable but oscillatory at the margin of stability. With decreasing ρ from ρ_{cr}, the oscillation of the iterative process was controlled. When the ρ is further continuously decreased, $|\gamma_c|$ starts to increase as depicted in fig 4.1, which translates to slowing of the speed of convergence. The value of ρ before $|\gamma_c|$ starts to increase is the optimum value ρ^*.

4.4 Results and Discussions

4.4.1 Simulation Setup

The **Algorithm** 2 is implemented in MATLAB R2020a environment to test the efficacy of the optimization problem (4.23) for IEEE network ranging from IEEE-30 bus to IEEE-1354 bus. The cost-coefficients a_i and b_i , generator limit, and the initial values of power demand at each bus are adopted from [42]. The power demands of all buses drop by up to 50% along the half-way of the simulations. Uniform distribution functions are used to randomize the drop percentages.In all cases, as the algorithm is robust enough to drive any initial value to convergence, I initialize power generations and prices at zero, $\{\boldsymbol{P}_g^0, \boldsymbol{\lambda}^0\} = \{\mathbf{0}\}$. I use *Mean Metropolis* algorithm with $\epsilon = 1$ to set up the weight matrix W assuming that each bus of the network is an agent and the communication topology follows the electrical connection between buses.

4.4.2 Algorithm Performance

Figure 4.2 illustrates the IEEE-30 bus test case where the iteration is carried up to 1200 iterations and the demand of the network abruptly changes at 600^{th} iteration. Figure 4.2(a) shows the primal residuals of all the buses converging to zero, which is detrimental to ensure power balance. The residuals go to zero as iteration k proceeds. A compassion with the plot on network's generation and demand Figure 4.2(d) illustrates that the generation meets the demand once all the residuals converge to zero. At iteration 600, the net demand is changed and the generators adjust to new optimal value in order to ensure power balance in the network. Here, the primal residuals quickly vanish to zero as the buses were already in one equilibrium point.

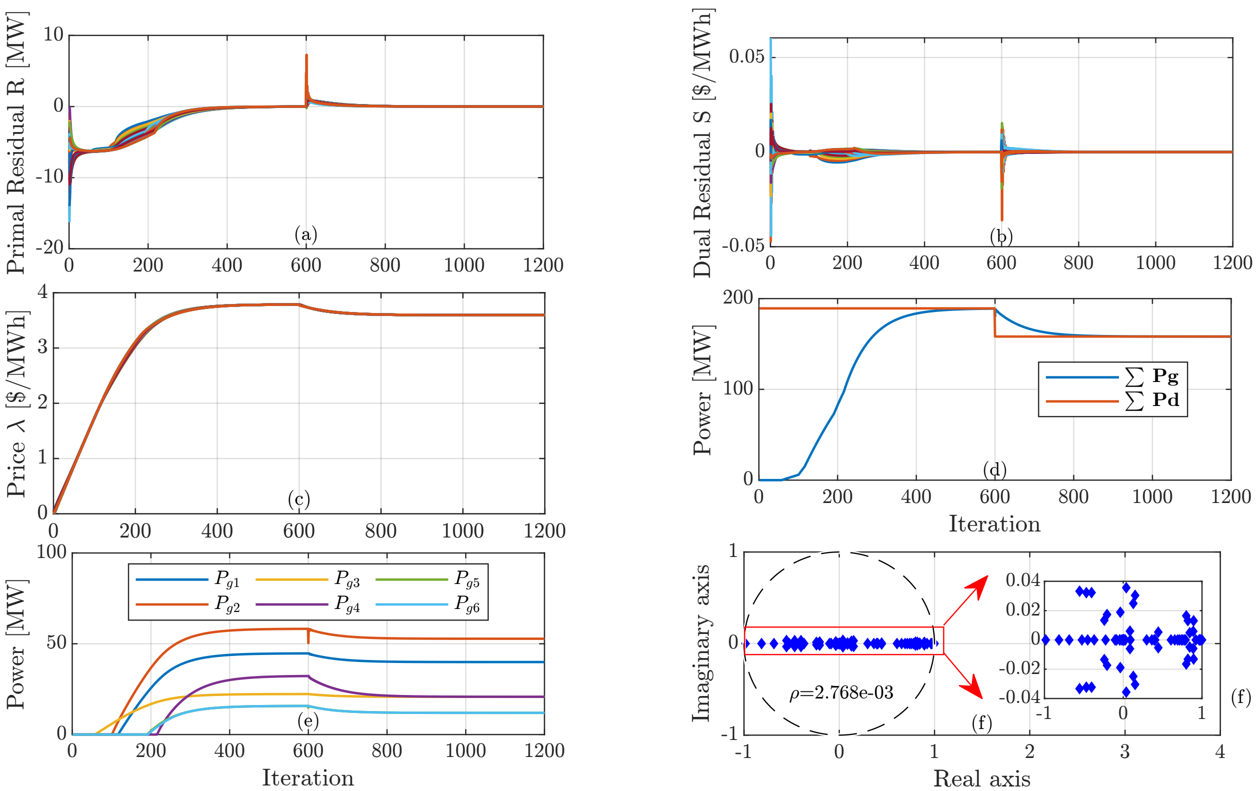

Figure 4.2 Results for IEEE 30-bus case. (a), (b) and (c) show the convergence process of primal residual, dual residual and market price respectively; (d) displays the total generation following the total demand of the network; (e) illustrates the optimum generation profile of all the plants in the network; (f) shows the eigenvalues of the state-space given in (4.37), for $\rho = 2.768e - 3$

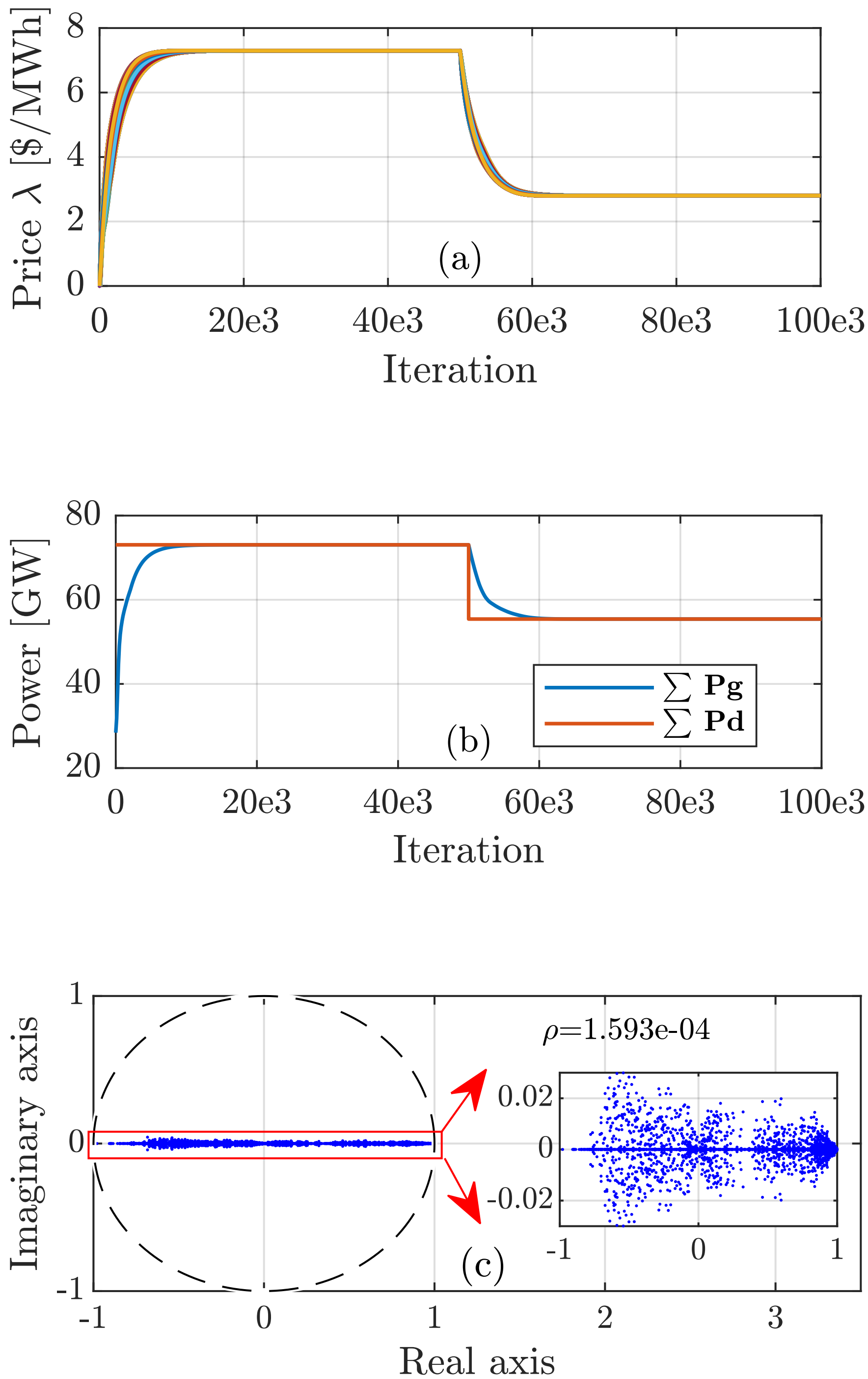

Figure 4.3 Results for IEEE 1354-bus case; (a) displays the convergence process for the market price (λ); (b) displays the total generation following the total demand on the system; (c) shows the eigenvalues of the state-space given in (4.37), for $\rho = 1.593e - 4$

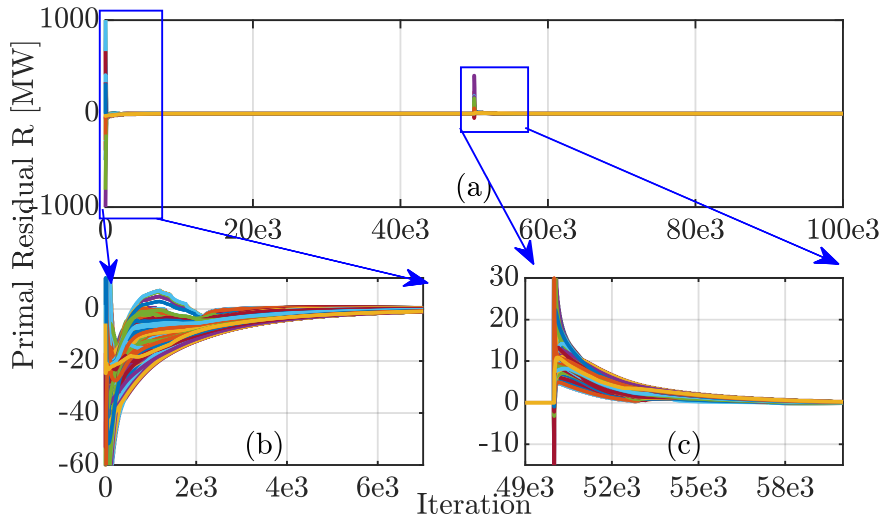

Figure 4.4 Results for Primal Residual Convergence:(a) displays the convergence process for primal residual along the whole simulation. (b) and (c) show in details the first and second iterative dynamics of this process respectively

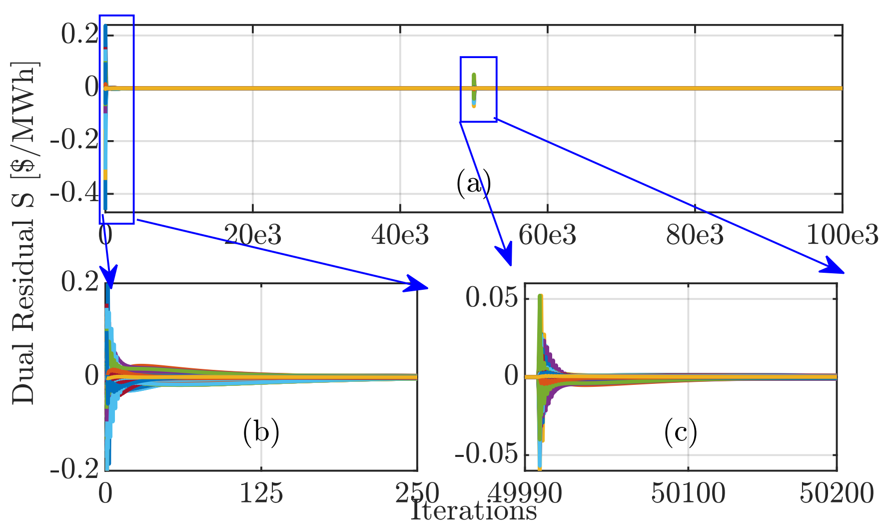

Figure 4.5 Results for dual residual convergence: (a) displays the convergence process for dual residual along the whole simulation. (b) and (c) show in details the first and second iterative dynamics of this process respectively

The dual residuals converge to zero once all the buses have consensus on market price and the estimate of power mismatch of the network. In other words, the market is cleared once the primal residuals and dual residuals converge to zero, thus dispatching all the generators in their optimum value. The optimum decision of all the generators is shown in Figure 4.2(e). Observe that the generations start from zero as initialized and quickly find their optimum points.

Figure 4.2(c) shows the convergence of all the prices to one market price. The price in all agents are travelling together as it is is set zero at the start. When the demand is reduced, market price finds consensus to a lower value, which comply with the supply-demand curve.

In order to demonstrate the stability of the proposed algorithm, Figure 4.2 shows the eigenvalues of state matrix $\boldsymbol{A}$ from the state-space aforementioned in (4.37); for the gradient step $\rho = 2.768e - 03$, visibly the eigenvalues are all within the unit that guarantees the stability. Furthermore, the predominance of eigenvalues closer to the abscissa announces the system reaches the steady-state without oscillating between operation states which in real time operation is a valuable characteristic. Recall that any eigenvalue outside the unity circle invites instability in the algorithm. The choice of optimum ρ is discussed in subsection 4.4.3.

Figure 4.3,4.4 and 4.5 illustrate the similar results but for a larger system, IEEE 1354-bus. This network has 260 generators and for this reason the individual generation profiles are not shown in the figures. The convergence of primal residuals and dual residuals to zero is shown in Figure 4.4, 4.5 respectively. This shows that the algorithm is scalable.

4.4.3 Computation Time

The computer specification used to implement the **Algorithm** 2 is Desktop PC with Intel Core i7 processor (3.6 GHz) 64 GB RAM. The the total number of iterations and time for test cases IEEE-30 bus, IEEE-118 bus, IEEE-300 bus and IEEE-1354 bus are displayed on Figure 4.6. I see that the larger the network, the longer time taken for the convergence of dual residuals, primal residuals and market price.

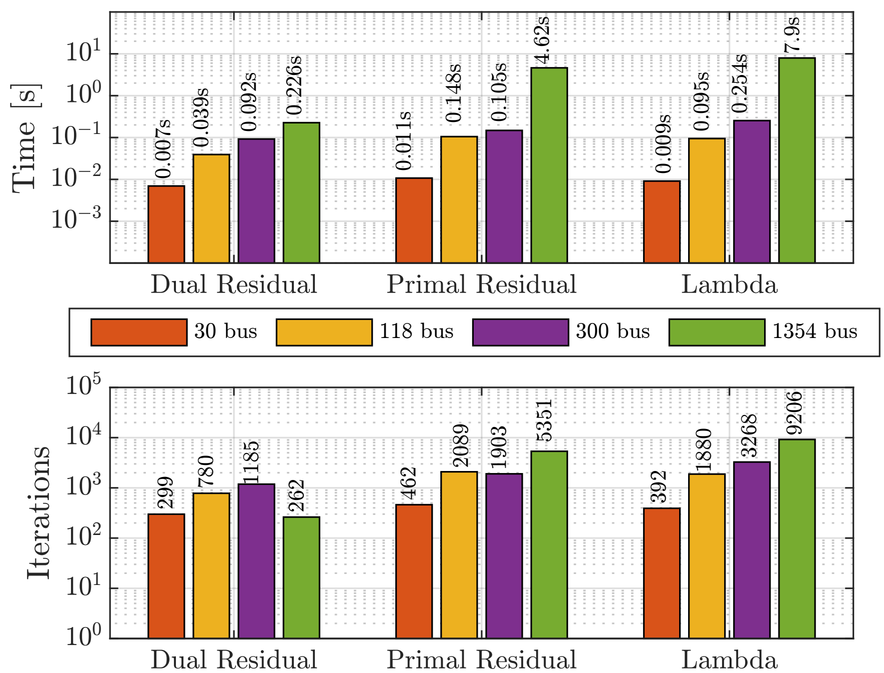

Figure 4.6 Figure shows (a) the time, and (b) the number of iterations necessary for the algorithm to converge within the limits of ϵ = 1e-5 of the final value. For visualization purpose, the convergence time and iterations are shown on top of each bar of the figures

Observe that the number of iterations for the convergence dual residuals of IEEE 1354-bus is lower than that of smaller networks. This inconsistency can be explained using the definition of dual residual S in (4.30). Note that S has two terms, the first term in which the Laplacian matrix drives the consensus on the market price to zero and the second term with in which the gradient ρ and Laplacian matrix together drives the consensus on the estimate of power mismatch to zero. As ρ is very small $(= 1.593e - 04)$, the effect second term quickly vanishes to zero ,and Figure 4.3 shows that market price λ find consensus quite early and move together to settle in its optimum value.

4.4.4 Convergence and Stability Results

Chapter 3 discusses the choice of optimal gain parameter ρ, and demonstrates the effect, in simulations, for the proposed D-subgradient algorithm. Section 4.3 builds on the same conceptual framework to quantify the value of optimal gain parameter (penalty parameter) ρ. This section intends to compare the convergence of two algorithms by extensive simulations.

In order to have a fair comparison, I simulate the case for strongly convex with quadratic cost function to compare the convergence speed between two algorithms. Table 4.1 describes the IEEE networks in terms of their connectivity, and weight of edges.

Table 4.1 IEEE Graph Network Characterstics

Test Case	$\lvert\gamma_c(\boldsymbol{L})\rvert$	max $\lvert\gamma_j(\boldsymbol{W})\rvert$ $j \neq i, \lvert\gamma_i(\boldsymbol{W})\rvert = 1$	Maximum Degree	Average Degree
Case 39	0.076186	0.020896	5	2.359
Case 300	0.0093838	0.0021002	11	2.7267

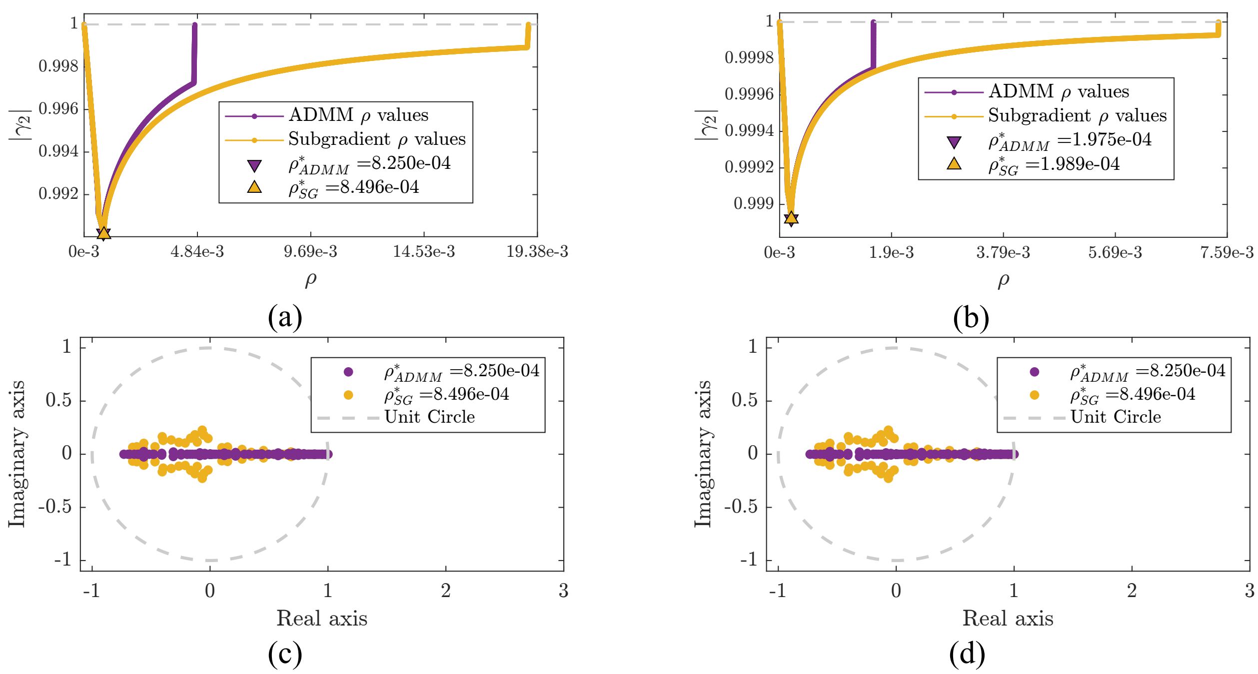

Figure 4.7 Results for: the locus of the second largest eigenvalues for corresponding gain parameter for (a) IEEE 39-bus network for (b) IEEE 300-bus network;displays the set of all eigenvalues on imaginary plane for (c)IEEE 39-bus network (d) IEEE 300-bus network

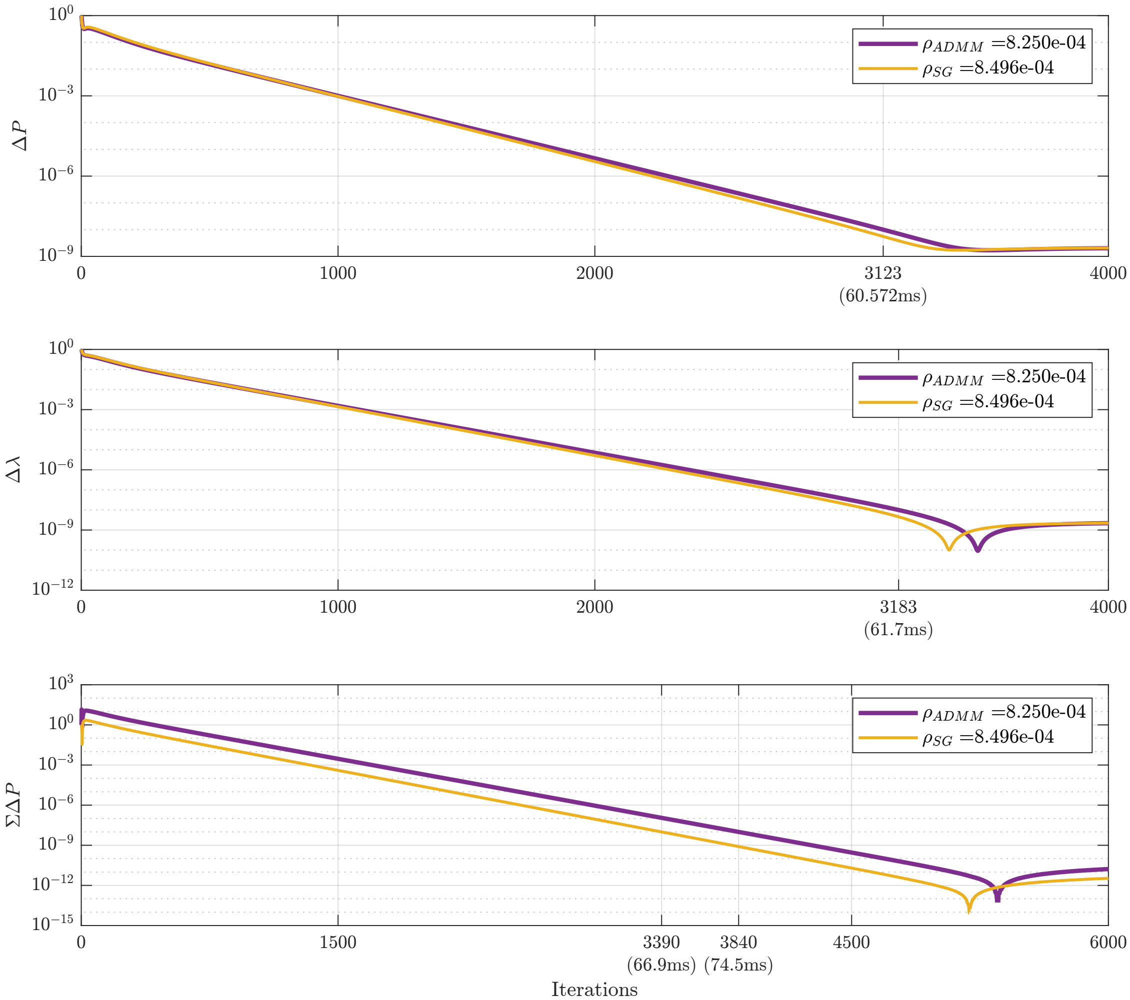

Figure 4.8 Results for IEEE 39-bus case. Based on different values of ρ, the figure displays the convergence process for residuals

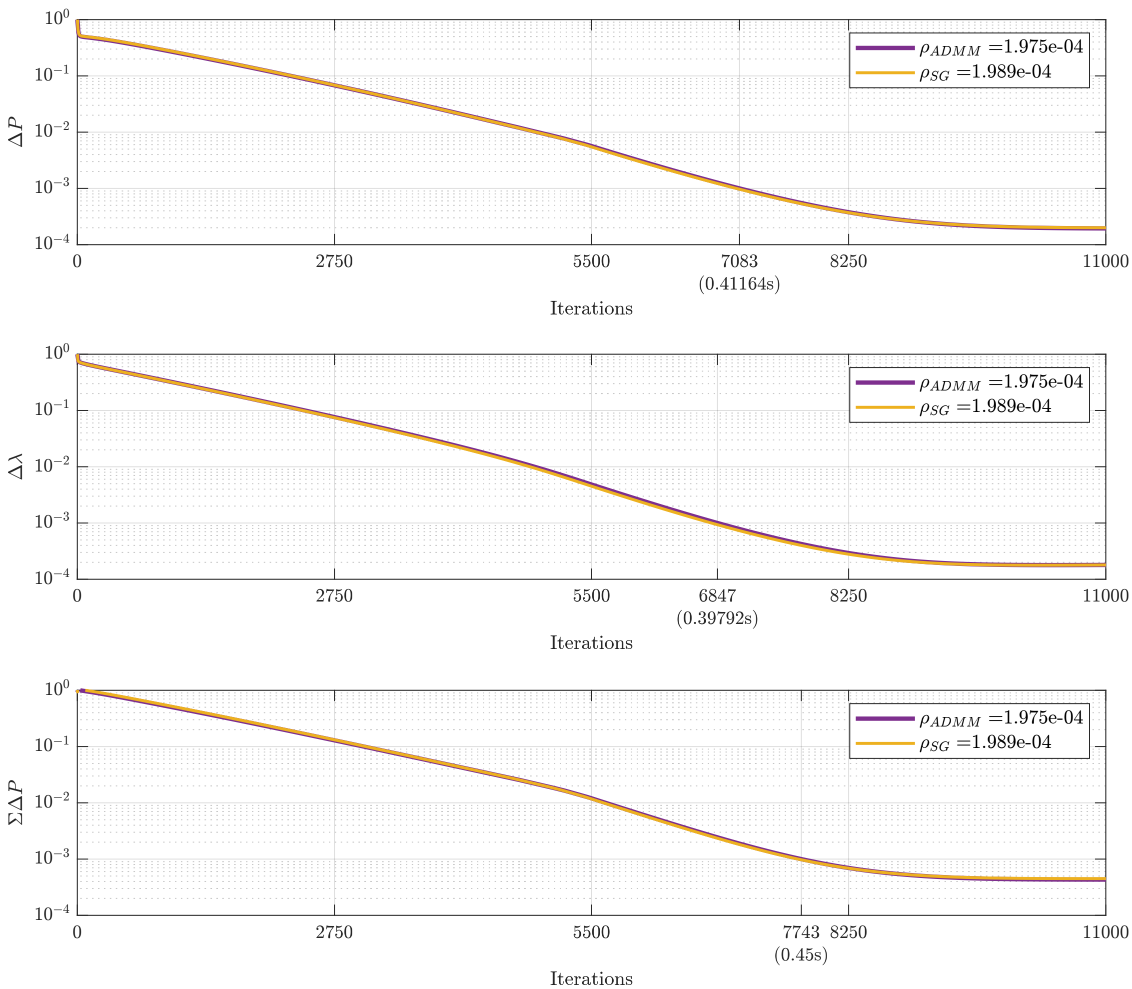

Figure 4.9 Results for IEEE 300-bus case. Based on different values of ρ, the figure displays the convergence process for residuals

Figure 4.7 (a) and (b) compares the response of second largest eigenvalues for different gain parameters in D-subgradient and D-ADMM algorithm. The second largest eigenvalue keeps decreasing with the decrease in ρ, and departs from the knee point. The simulation demonstrates that it is not algorithm, but the graph network that defines the convergence of iterates to the optimal solution.

Similarly, Figure 4.7 (c) and (d) compares the response of all eigenvalues in complex plane for the change in gain parameter for IEEE 39-bus network and IEEE 300-bus network. The purple area corresponding to ADMM algorithm are more to real plane than subgradient algorithm. This can be interpreted as D-ADMM is more stable than D-subgradient algorithm, which supports the fundamental idea behind augmented part in the Lagrange relaxation of D-ADMM algorithm.

First I define the following residuals in order to show the convergence of different iterate. The metric of convergence speed is same as in chapter 3.

$$\Delta P = \frac{\left\|\boldsymbol{p}(k) - \boldsymbol{p}^*\right\|_2}{\left\|\boldsymbol{p}(0) - \boldsymbol{p}^*\right\|_2}; \qquad \Delta\lambda = \frac{\left\|\boldsymbol{\lambda}(k) - \boldsymbol{\lambda}^*\right\|_2}{\left\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}^*\right\|_2}; \quad \sum \Delta P = \frac{\left\|\sum_{i\in\mathcal{V}}\left(p_i(k) - {p_i}^*\right)\right\|_2}{\left\|\sum_{i\in\mathcal{V}}\left(p_i(k) - {p_i}^*\right)\right\|_2};$$

Where ΔP is the residual of primal iterates, $\Delta\lambda$ is the residual of dual iterates, and $\sum \Delta P$ is the residual of power mismatch.

Figure 4.8 and 4.9 demonstrate the convergence of primal iterates, dual iterates, and power mismatch. The results show that subgradient and ADMM algorithm converge with the same rate for strongly convex functions.

4.5 Summary

In this chapter, the ADMM algorithm is fully distributed to solve the economic dispatch problem in real-time without a need for a master node. The proposed solution exploits dual decomposition and *dynamic average consensus* algorithms to develop the update procedures with minimal information shared between the directly connected neighbors. The performance of the proposed solution including optimality, stability, convergence, and computation speed is investigated against standard IEEE test cases. Simulations demonstrate that the algorithm dynamically responds to the real-time change in the demand of the network.

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

In this thesis, consensus based distributed algorithms to allocate resources in real-time have been proposed to solve economic dispatch problems. First, a decentralized problem was transformed into a fully distributed problem using consensus theory and dual decomposition, where all agents' computations are just based on their own data and their neighbors' shared information. The net load (passive load minus renewable generation) in the future grid is expected to fluctuate quickly, and the distributed algorithms must be able to track the real-time changes in the constraint. The dynamic average consensus algorithm was embedded in the Subgradient algorithm to track the time-variant constraint, and distributed online algorithm was proposed. It has been further proved that the primal and dual iterates converges to the optimal points using KKT conditions of optimality. The algorithm proposed can continuously perform without a need for agents to collect information overtime to run the optimization since the algorithm is agnostic to any initialization vector. Thus, the algorithm can be embedded in the any intelligent device. For example, plug-and-play loads like electric vehicles can independently decide how to control their charging and discharging process, any price responsive load can be programmed for features like demand response and auxiliary services like frequency regulation with no need for coordination with other entities.

Similarly, a decentralized version of ADMM was modified to handle fully distributed optimization problems. The ADMM is different from a regular Lagrange relaxation in that it has a penalty term added in the relaxed version which vanishes to zero at primal feasibility. This brings robustness in the problem. From the optimization point of view, augmenting penalty term makes the dual problem differentiable under mild conditions. In other words, this removes the assumption of the need for strictly convex problem to have a equivalent differentiable dual problem. In the future electric grid, then, the objective functions (cost function, utility function) whose objective function is not necessarily strictly convex can be modeled. The dynamic average consensus algorithms has been embedded in order to transform the solution to a fully distributed version. It has been further shown that the optimal solution is successfully sought by the distributed algorithm developed in this thesis. Using KKT conditions, It has been demonstrated that the problem is both primal- and dual-feasible, and it still has a saddle point. Our proposed distributed ADMM algorithm decomposes the optimization problem into subproblems for individual agents where each agent calculates the primal minimizer (cost minimization) and the dual maximizer (subgradient update) at the same time, leading to the saddle point of the Lagrange surface. The proposed ADMM algorithm–similar to the proposed subgradient algorithm–possess the properties of starting from any initialization vector and handling the demand change in real-time in a distributed fashion with any risk to deviate from the optimal solution. In addition, the algorithm has been designed in a way such that the agent's information is kept confidential, which in turn helps a fair electricity trading. The test cases are simulated from IEEE 30 bus network to the network as large as 1354 bus, thus demonstrating the scalibility of the algorithm. Second, the stability and the convergence were investigated by modeling both algorithms as discrete dynamic systems. In the dynamics of the proposed algorithms–*i.e.* minimization over primal variables and maximization over dual variables in alternating sequence–large gradient step might lead to instability of the algorithm. The optimal and stable gain parameter was calculated based on the study of the state matrix and modal analysis. It has been shown that the convergence rate is governed by the gradient step corresponding to the second largest eigenvalue of the state matrix. Lastly, the parallels have been drawn between two algorithms for a given quadratic cost function via simulations. Through simulations of IEEE test cases for various gradient steps, the convergence speed of the Subgradient and ADMM algorithms are similar for equivalent gradient steps. More, increasing the network connectivity level significantly increases the convergence speed.

5.2 Future Directions

There are ample research directions to pursue. Some of them are briefly summarized in bullets below:

- Security Constraint Economic Dispatch: While, in this work, the primal iterates are projected to its feasible domain in every iteration, the optimization can be expanded to problems with inequality constraints. For economic dispatch, this is equivalent to adding reserve limit in the problem. Development of distributed algorithms for this problem has not been targeted in the literature yet.

- Holistic Control Framework: The traditional power system has layers of hierarchical frequency control including primary control (droop control), secondary control (Automatic Generation Control), and tertiary control (static Economic Dispatch). With the future smart power grids dominated by VRE resources in imminent future, there is a need for a new control platform to control frequency and perform power system optimization simultaneously.

- Communication Imperfections: It is suggested to study the communication imperfection in data transfer as the future work. In this thesis, it has been assumed the communication network is perfect and real-time while time delays and packet drops are ubiquitous.

- Analytical Solutions for Distributed Gain Calculation: In this work, the stability analysis and defining the optimal gain parameters are done in a central fashion. Given that the algorithm is distributed in nature and agents may join or drop at any time, it is recommended to develop algorithms to calculate the optimal gain parameter in a distributed fashion, too.

With the power grid going through an unprecedented transformation because of the penetration of DER, there is an imminent need to design the market and address some pressing issues that have recently surfaced. The algorithms proposed can set the foundation for electricity market design with high penetration of DERs.

REFERENCES

[1] I. Dabbagchi and R. Christie, "Power systems test case archive," *University of Washington*, 1993.

[2] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power generation, operation, and control*. John Wiley & Sons, 2013.

[3] G. Arnold, D. Bakken, A. Bose, C. Hauser, D. Whitehead, G. Zweigle, A. Gómez-Expósito, A. Abur, A. de la Villa Jaén, C. Gómez-Quiles, *et al.*, "Smart grid: The electric energy system of the future," *Proceedings of the IEEE*, vol. 99, no. 6, 2011.

[4] G. Hug, S. Kar, and C. Wu, "Consensus+ innovations approach for distributed multiagent coordination in a microgrid," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1893–1903, 2015.

[5] Y. Xu, W. Zhang, W. Liu, X. Wang, F. Ferrese, C. Zang, and H. Yu, "Distributed subgradient-based coordination of multiple renewable generators in a microgrid," *IEEE Transactions on Power Systems*, vol. 29, no. 1, pp. 23–33, 2013.

[6] F. Guo, G. Li, C. Wen, L. Wang, and Z. Meng, "An accelerated distributed gradient-based algorithm for constrained optimization with application to economic dispatch in a large-scale power system," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[7] Z. Zhang, X. Ying, and M.-Y. Chow, "Decentralizing the economic dispatch problem using a two-level incremental cost consensus algorithm in a smart grid environment," in *2011 North American Power Symposium*, pp. 1–7, IEEE, 2011.

[8] R. Wang, Q. Li, B. Zhang, and L. Wang, "Distributed consensus based algorithm for economic dispatch in a microgrid," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3630–3640, 2018.

[9] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *IFAC world congress*, pp. 1–6, Citeseer, 2005.

[10] R. O. Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," 2003.

[11] D. S. Kirschen and G. Strbac, *Fundamentals of power system economics*. John Wiley & Sons, 2018.

[12] M. Newman, *Networks*. Oxford university press, 2018.

[13] L. Kempton, G. Herrmann, and M. di Bernardo, "Distributed optimisation and control of graph laplacian eigenvalues for robust consensus via an adaptive multilayer strategy," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 9, pp. 1499–1525, 2017.

[14] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 40–72, 2019.

[15] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[16] Y. Xu and W. Liu, "Novel multiagent based load restoration algorithm for microgrids," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 152–161, 2011.

[17] N. Jorge and J. W. Stephen, "Numerical optimization," *Springerverlang, USA*, 2006.

[18] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[19] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, "Notes on decomposition methods," *Notes for EE364B, Stanford University*, pp. 1–36, 2007.

[20] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[22] W. L. Brogan, *Modern control theory*, pp. 348–349. Upper Saddle River: Prentice Hall, third ed., 1991.

[23] P. J. Olver, C. Shakiban, and C. Shakiban, *Applied linear algebra*, vol. 1, p. 493. Springer, 2006.

[24] J. N. Tsitsiklis, "Problems in decentralized decision making and computation.," tech. rep., Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.

[25] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[26] 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 Transactions on Smart Grid*, vol. 9, no. 4, pp. 2574–2594, 2016.

[27] 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 Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[28] T. Wang, D. O'Neill, and H. Kamath, "Dynamic control and optimization of distributed energy resources in a microgrid," *IEEE transactions on smart grid*, vol. 6, no. 6, pp. 2884–2894, 2015.

[29] V. R. Disfani, L. Fan, and Z. Miao, "Distributed dc optimal power flow for radial networks through partial primal dual algorithm," in *2015 IEEE Power & Energy Society General Meeting*, pp. 1–5, IEEE, 2015.

[30] V. Disfani, Z. Miao, L. Fan, B. Zeng, *et al.*, "Dual decomposition-based privacy-preserving multi-horizon utility-community decision making paradigms," *arXiv preprint arXiv:1503.09087*, 2015.

[31] V. R. Disfani, *Optimization and control for microgrid and power electronic converters*. University of South Florida, 2015.

[32] S. Kar and G. Hug, "Distributed robust economic dispatch in power systems: A consensus+ innovations approach," in *2012 IEEE Power and Energy Society General Meeting*, pp. 1–8, IEEE, 2012.

[33] S. Kar, G. Hug, J. Mohammadi, and J. M. Moura, "Distributed state estimation and energy management in smart grids: A consensus + innovations approach," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1022–1038, 2014.

[34] Z. Tang, D. J. Hill, and T. Liu, "A novel consensus-based economic dispatch for microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3920–3922, 2018.

[35] G. Chen and Q. Yang, "An admm-based distributed algorithm for economic dispatch in islanded microgrids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3892–3903, 2017.

[36] X. He, Y. Zhao, and T. Huang, "Optimizing the dynamic economic dispatch problem by the distributed consensus-based admm approach," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3210–3221, 2019.

[37] G. Chen, J. Ren, and E. N. Feng, "Distributed finite-time economic dispatch of a network of energy resources," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 822–832, 2016.

[38] M. Baranwal, K. Garg, D. Panagou, and A. O. Hero, "Robust distributed fixed-time economic dispatch under time-varying topology," *IEEE Control Systems Letters*, 2020.

[39] F. Guo, C. Wen, J. Mao, and Y.-D. Song, "Distributed economic dispatch for smart grids with random wind power," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1572–1583, 2015.

[40] J. Tuinstra, *Price Dynamics in Equilibrium Models: The Search for Equilibrium and the Emergence of Endogenous Fluctuations*, vol. 16. Springer Science & Business Media, 2001.

[41] C. Zhao, U. Topcu, N. Li, and S. Low, "Design and stability of load-side primary frequency control in power systems," *Automatic Control, IEEE Transactions on*, vol. 59, pp. 1177–1189, May 2014.

[42] 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 Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2010.

[43] T. Mai, D. Sandor, R. Wiser, and T. Schneider, "Renewable electricity futures study. executive summary," tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2012.

[44] B. A. Frew, S. Becker, M. J. Dvorak, G. B. Andresen, and M. Z. Jacobson, "Flexibility mechanisms and pathways to a highly renewable us electricity future," *Energy*, vol. 101, pp. 65–78, 2016.

[45] S. Afshar, S. Wasti, and V. Disfani, "An admm-based miqp platform for the ev aggregation management," *arXiv preprint arXiv:2010.03475*, 2020.

[46] Y. Wang, S. Wang, and L. Wu, "Distributed optimization approaches for emerging power systems operation: A review," *Electric Power Systems Research*, vol. 144, pp. 127–135, 2017.

[47] H. Pourbabak, J. Luo, T. Chen, and W. Su, "A novel consensus-based distributed algorithm for economic dispatch based on local estimation of power mismatch," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5930–5942, 2017.

[48] M. Baranwal, K. Garg, D. Panagou, and A. O. Hero, "Distributed fixed-time economic dispatch under time-varying topology and uncertain information," *arXiv preprint arXiv:1910.14214*, 2019.

[49] S. Chakrabarti and R. Baldick, "Look-ahead scopf (lascopf) for tracking demand variation via auxiliary proximal message passing (apmp) algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 116, p. 105533, 2020.

[50] Y. Wang, L. Wu, and S. Wang, "A fully-decentralized consensus-based admm approach for dc-opf with demand response," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2637–2647, 2016.

[51] T. Erseghe, "Distributed optimal power flow using admm," *IEEE transactions on power systems*, vol. 29, no. 5, pp. 2370–2380, 2014.

[52] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "A proof of convergence for the alternating direction method of multipliers applied to polyhedral-constrained functions," *arXiv preprint arXiv:1112.2295*, 2011.

[53] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2011.

[54] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[55] C. Li, X. Yu, T. Huang, and X. He, "Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2407–2418, 2017.

[56] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[57] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed admm over networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5082–5095, 2017.

[58] N. Bof, R. Carli, and L. Schenato, "Is admm always faster than average consensus?," *Automatica*, vol. 91, pp. 311–315, 2018.

[59] S. Wasti, P. Ubiratan, S. Afshar, and V. Disfani, "Distributed dynamic economic dispatch using alternating direction method of multipliers," *arXiv preprint arXiv:2005.09819*, 2020.

[60] J. Rivera, C. Goebel, and H.-A. Jacobsen, "Distributed convex optimization for electric vehicle aggregators," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1852–1863, 2016.

APPENDIX A

IEEE TEST CASES

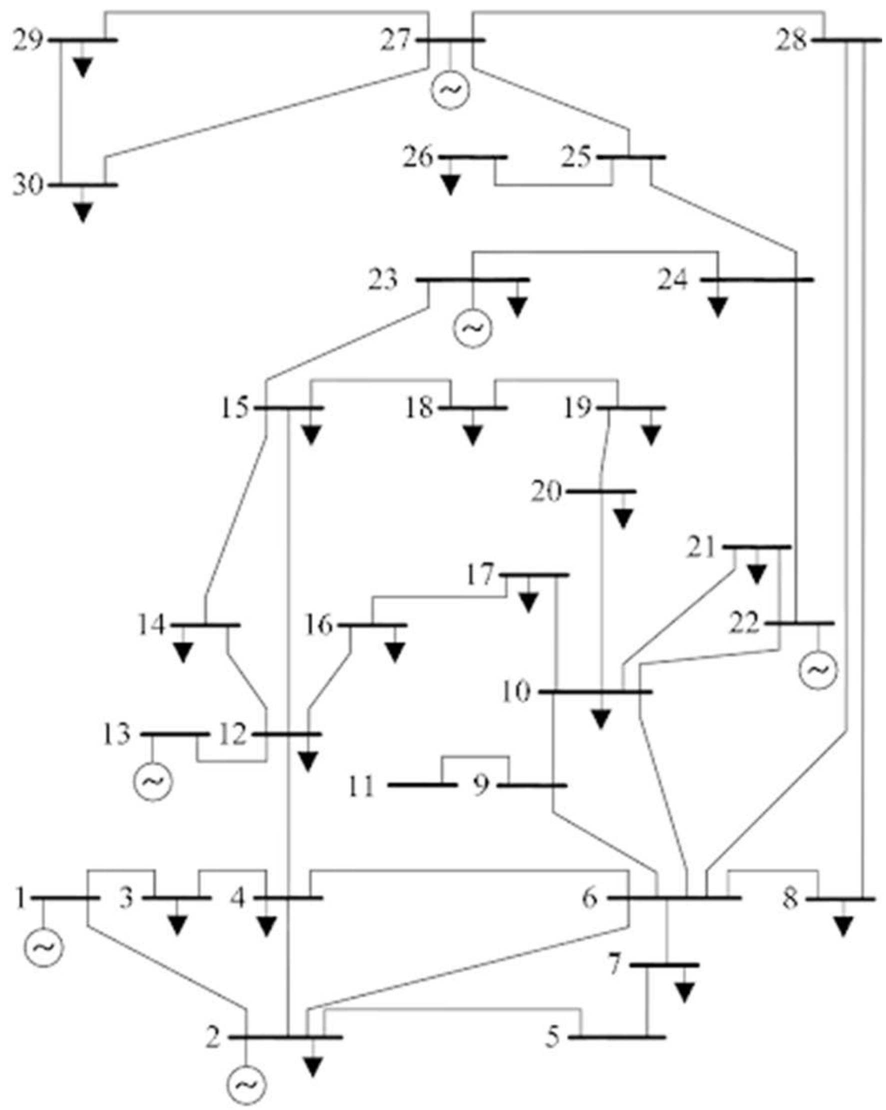

Figure A.1 IEEE 30-bus Network [1]

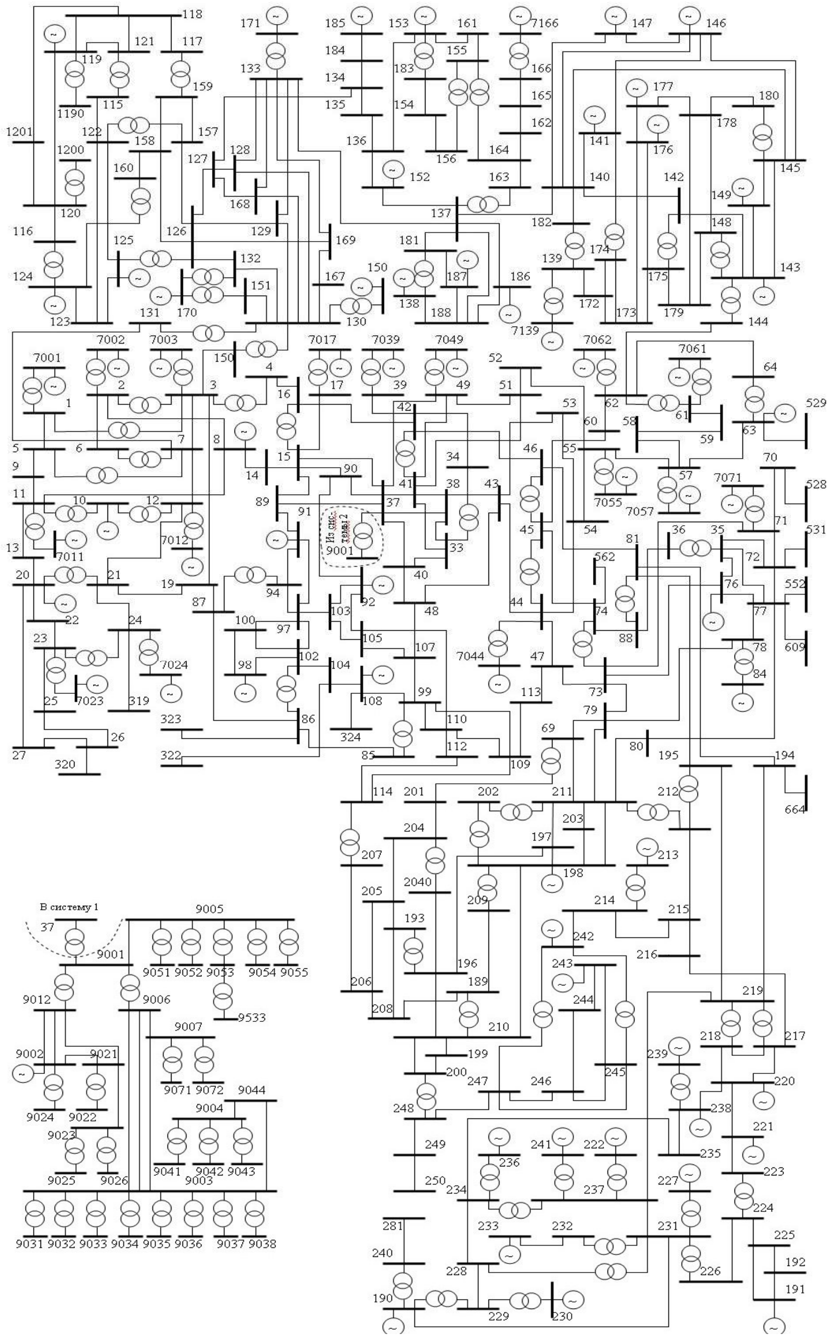

Figure A.2 IEEE 300-bus Network [1]

VITA

Shailesh Wasti received his Bachelor's degree in Electrical Engineering (2015) from Tribhuvan University, Nepal, and is currently working toward an MS degree in Electrical Engineering at the University of Tennessee at Chattanooga under the supervision of Dr. Vahid R. Disfani. Shailesh has worked on a market design of electric vehicles, distributed optimization for resource allocation problems, allocation of inertia in the power grid dominated with renewables, and modular multilevel converters for grid integration of solar photovoltaic systems.

His research interest lies in the study of distributed algorithms, market design, control and optimization of the power system, and the economics of power system. He is passionate, and quite serious about the welfare of society in general while transcending the technicalities of electricity generation and distribution.