2020

Distributed Multi-agent Optimization and Control with Applications in Smart Grid

Towfiq Rahman
University of Central Florida

DISTRIBUTED MULTI-AGENT OPTIMIZATION AND CONTROL WITH APPLICATIONS IN SMART GRID

by

TOWFIQ RAHMAN
B.Sc. Islamic University of Technology, 2012

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2020

Major Professor: Zhihua Qu

ABSTRACT

With recent advancements in network technologies like 5G and Internet of Things (IoT), the size and complexity of networked interconnected agents have increased rapidly. Although centralized schemes have simpler algorithm design, in practicality, it creates high computational complexity and requires high bandwidth for centralized data pooling. In this dissertation, for distributed optimization of networked multi-agent architecture, the Alternating Direction Method of Multipliers (ADMM) is investigated. In particular, a new adaptive-gain ADMM algorithm is derived in closed form and under the standard convex property to greatly speed up the convergence of ADMM-based distributed optimization. Using the Lyapunov direct approach, the proposed solution embeds control gains into a weighted network matrix among the agents uses and those weights as adaptive penalty gains in the augmented Lagrangian. For applications in a smart grid where system parameters are greatly affected by intermittent distributed energy resources like Electric Vehicles (EV) and Photo-voltaic (PV) panels, it is necessary to implement the algorithm in real-time since the accuracy of the optimal solution heavily relies on sampling time of the discrete-time iterative methods. Thus, the algorithm is further extended to the continuous domain for real-time applications and the convergence is proved also through Lyapunov direct approach. The algorithm is implemented on a distribution grid with high EV penetration where each agent exchanges relevant information among the neighboring nodes through the communication network, optimizes a combined convex objective of EV welfare and voltage regulation with power equations as constraints. The algorithm falls short when the dynamic equations like EVs state of charge is taken into account. Thus, the algorithm is further developed to incorporate dynamic constraints and the convergence along with control law is developed using Lyapunov direct approach. An alternative approach for convergence using passivity-short properties is also shown. Simulation results are included to demonstrate the effectiveness of proposed schemes.

ACKNOWLEDGMENTS

The work in this dissertation was supported in part by the U.S. Department of Energy under awards DEEE0006340, DE-EE0007327 and DE-EE0007998, by the U.S. National Science Foundation under grant ECCS-1308928, by the U.S. Department of Transportation under award DTRT13GUTC51.

TABLE OF CONTENTS

LIST OF FIGURES viii

LIST OF TABLES x

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: BACKGROUND 4

Convex sets and functions 4

Unconstrained Convex Optimization 6

Gradient Descent 6

Constrained Optimization: Lagrangian dual function 7

Dual ascent and Augmented Lagrangian 9

Alternating Direction Method of Multipliers 12

CHAPTER 3: DISTRIBUTED MULTI AGENT ADMM FRAMEWORK 14

Distributed ADMM formulation 16

Network of Agents 17

Distributed Adaptive-Gain ADMM 18

Real Time Iterative Laws of ADMM 25

Improvement of convergence rate via adaptive gain 29

Simulation Results 32

CHAPTER 4: CONTINUOUS TIME ADMM ALGORITHM FOR SMART GRID 40

Problem Formulation 43

Architecture of EV Aggregators 43

Branch Flow Model of Power Distribution Network 45

Objective Functions 48

Continuous-Domain Distributed ADMM Algorithm 49

Network of Agents 50

Distributed Real-Time Continuous Domain ADMM 51

Real-Time ADMM Solution to EV Management 57

Simulation results 61

CHAPTER 5: ADMM ALGORITHM WITH DYNAMIC CONSTRAINT 67

A Distributed Multi-Agent Networked ADMM 68

Networked Mutli-Agent System 69

Distributed Dynamic ADMM 69

Dynamic ADMM Application . . . 86
Branch Flow Model . . . 87
EV Parameters . . . 89
Objective Functions . . . 90
Problem Formulation . . . 91
Solution Procedure . . . 93
Simulation Results . . . 95
CHAPTER 6: CONCLUSION . . . 99
APPENDIX A: LIST OF PUBLICATIONS . . . 102
LIST OF REFERENCES . . . 103

LIST OF FIGURES

3.1 Networked cyber-physical system of multi agents. 17

3.2 Agent i's information flow for local variable update. 28

3.3 Concept illustration of the proposed algorithm 31

3.4 Primal residual $(\hat{x}_i^k - \hat{z}_{ij}^k)$: fixed versus adaptive penalty 34

3.5 Dual residual $(\hat{z}_{ij}^{k+1} - \hat{z}_{ij}^k)$: fixed versus adaptive penalty 34

3.6 Dual residual $(\hat{\mu}_{ij}^{k+1} - \hat{\mu}_{ij}^k)$: fixed versus adaptive penalty 35

3.7 Resultant vector from update law: agent 1 35

3.8 Resultant direction comparison between fixed gain versus adaptive penalty: agent 1 . 36

3.9 The resultant trajectory taken by agent 1 to reach optimality 36

4.1 The communication structure of aggregators. 44

4.2 A distribution network. 45

4.3 Voltage profile of the base case 63

4.4 Voltages at node 111,113 and 114 63

4.5 Total active power injection by EVs into the grid 64

4.6 Steady state voltage output after using the algorithm 64

5.1 A networked multi-agent system with both physical and communication layer 70

5.2 The overall update dynamics 78

5.3 The multi-layer representation of distribution grid with associated variables . 86

5.4 A distribution network. 87

5.5 Voltage at Node 111,113 and 114 96

5.6 Total active power injection by EVs into the grid 97

5.7 Active power injection by 3 EVs 97

5.8 The SOC evolution of the 3 EVs 98

LIST OF TABLES

3.1 Comparative analysis of the algorithms 38

4.1 EV charging categories 62

4.2 EV Types 65

NOMENCLATURE

Symbol/Abbreviation	Description
x_i	State of agent i.
N	Set of agents in the network.
N_i	Set of neighboring agents of agent i.
$f_i(x_i)$	Objective function of agent i.
A_{ij}	Matrix representing physical interconnection.
$\nabla_{x_i} f_i(x_i)$	Gradient operator w.r.t variable x_i.
S	Communication matrix.
k	Discrete time step.
z_{ij}	Auxiliary variable representing the estimate of agent j's variables by agent i.
L_D	Augmented Lagrangian
D	Row-stochastic gain matrix.
d_{ij}^k	Regularized time-varying penalty parameter.
w_i	Penalty parameter.
β_{ij}^k	Scalar control gains.
λ_{ij}, μ_{ij}	Dual variables.
∂	Partial derivative operator.
α_i	step size of agent i.
M_f	Lipschitz constant.
φ	Uniformly bounded disturbance.
E_i^k	Lyapunov function of agent i.
l_i, m_i	Indices for max and min,respectively.

ϵ_i^k	Gain adjustment in network for agent i .
$\|x_i^k - z_{ij}^k\|$	Primal residual.
$\|z_{ij}i^{k+1} - z_{ij}^k\|, \|\mu_{ij}^{k+1} - \mu_{ij}^k\|$	Dual residuals.
$x_i^*, z_{ij}^*, \lambda_{ij}^*, \mu_{ij}^*$	Optimal solution vectors.
p_i, q_i	Active and reactive power injection by EV aggregators at node i.
G	Directed graph.
$\mathcal{E}$	Set of distribution lines.
V_i, I_i	Voltage and current at node i.
P_{d_i}, Q_{d_i}	Load demand at node i.
R_{ij}, X_{ij}	Impedance of branch $i \to j$.
$U_i(p_i, q_i)$	Utility function.
ψ	Price of electricity.

CHAPTER 1: INTRODUCTION

In any kind of practical problem which involves decision making, either by the user or by the device itself, an optimization routine is run to come up with an optimal solution that will solve the problem in the most desired fashion. Most of this problem can be cast into a mathematical framework and can then be solved using available optimization algorithms. The optimization routine can be run centrally where all the data involving the system and its constraints are pooled into one central entity which runs the necessary steps to optimize the whole system. But in recent times, due to several shortcomings of centralized structure like scaling and robustness, distributed optimization has gained a lot of attention. A distributed optimization algorithm must be capable of collecting localized data across a network of interconnected agents, and each agent must be able to solve their optimization problem without requiring any centralized coordination. Among all these algorithms, the Alternating Direction Method of Multipliers (ADMM) has gained a lot of popularity due to its ability to decompose a complex optimization problem into a sequence of simpler sub-problems. It combines this decomposability with the superior convergence property of augmented Lagrangian [1]. ADMM was first introduced by Glowinski & Marroco [2] and by Gabay & Mercier [3]. Most recently, it has been applied to many applications in such areas as image processing [4], machine learning [5], resource allocation [6], power system optimization [7] etc. Due to this diverse range of applications, researchers dug deep into the convergence properties of ADMM. In this dissertation, a distributed multi-agent setting of ADMM is investigated and an adaptive penalty method for faster convergence based on the information received by each agent from its neighboring agents through the communication protocol is developed. In most of the existing ADMM literature, penalty parameter are set to be constant and identical [8, 9, 10, 11]. The advantage of using the adaptive penalty is noted in [12, 13, 1], but those results either require global information or have convergence and scalability issues. Thus in this work, these shortcomings are tried to overcome and develop an algorithm where each agent can solve its optimization problem locally while making the architecture scalable. The problem is setup into a multi-agent problem where agents are connected among themselves and can communicate with each other. Each agent runs an optimization routine where it tries to solve its objective function and satisfy its constraints based on the information it receives from its neighbors. While the optimization routine is running, some of the information received by an agent is more useful than some other information for converging. So, the algorithm determines the important information and increases the gain on those and vice versa. In this way, the proposed adaptive penalty method can be made to converge faster.

For the next step in this work, the algorithm is applied to a smart grid setting where a distribution system with heavy electric vehicle (EV) penetration is chosen. EVs have gained massive popularity in the recent past due to attention from leading automotive industries and the government's push to reduce its emission of greenhouse gases. Although this is good for the environment, the distribution system is facing problems due to the penetration of electric vehicles. The presence of a large number of EVs is causing a substantial degradation of the quality and reliability of the power grid. Despite the adverse effects, EVs can be controlled and used as a power source to benefit the grid. Also, this control of EVs cannot be done by a centralized body since that would be non-scalable and would require huge communication bandwidth. To tackle this ever scaling problem, in this dissertation, a continuous domain multi-agent distributed ADMM algorithm is developed. The objective is to control the EVs and utilize them to maintain the grid voltage within the normal operating range while also satisfying the consumers by maximizing their welfare. The developed algorithm is an extension of the distributed multi-agent ADMM algorithm mentioned above. The algorithm was developed in the continuous domain since in most of the other algorithms which are iterative and in discrete time, the accuracy of the optimal solution greatly depends on the sampling time, thus it is not robust to changes that are prevalent with distributed energy resources like EVs. We took the update dynamics from the previous algorithm which were in discrete time, converted them to continuous-time, and proved the convergence through the Lyapunov direct method. The algorithm is implemented in an IEEE 123 bus test distribution system to show its effectiveness.

One of the drawbacks of the continuous-domain algorithm is that it is unable to tackle dynamic constraints such as the state of charge (SOC) of the EVs. To tackle this shortcoming, the algorithm is further developed to incorporate this dynamic constraint. The dynamic update laws are developed and the convergence with the control law is proven through the Lyapunov approach. An alternative approach to convergence was also shown using passivity short properties. It was shown that the dynamics can be expressed in state-space and can be broken down into several subsystems. Each subsystem has its properties and they can be interconnected together and the whole system can be proved to be passivity-short. The algorithm is implemented into a smart grid with EV penetration. The problem was formulated into a multi-layer multi-period optimization problem where the distribution system operator (DSO) solves the optimal power flow and utilizes the EV potential to regulate the voltage close to 1p.u. Aggregators, who work in tandem with the DSO, provides input signals to the EVs. The EVs in turn allows the aggregators to use them for grid ancillary services in exchange for financial compensation. They also have their optimization problem to solve with the dynamic constraint of SOC to satisfy. Simulation studies were run on the system with the algorithm in effect and the results were illustrated to plots and figures.

For future work, the algorithm can be further extended to incorporate the temporal dependence of the objective function due to the dynamic nature of the constraints. With the introduction of dynamic constraints, static objective functions can no longer be used. So the problem must be developed into a multi-stage ADMM problem. More simulation scenarios must be run in the smart grid setting to test the algorithm and measure its scalability and stability. Finally, a comparative study can be performed with the other distributed ADMM algorithm in the literature with the algorithm presented in the dissertation and measure the effectiveness of their performance.

CHAPTER 2: BACKGROUND

In this chapter, a brief review of some of the existing optimization techniques, their definitions, and properties are presented which acts as a precursor to the ADMM. These concepts act as a baseline for some of the developments which will be carried out in the subsequent chapters.

Convex sets and functions

Our ability to solve an optimization problem depends considerably on several factors. One such factor is the form of the objective function and constraint functions. Even when they are smooth and proper, most of the time it is difficult to solve. There are classes of problems like least-square problems and linear programming which are easier to solve even when the size of the problem is huge with thousands of variables and constraints. One such class of problems is convex optimization which has structured and effective algorithms for solving them reliably and efficiently. To understand these problems with solution algorithms, we need a basic understanding of convexity properties.

Definition 1 *A set C is said to be a convex set if for any two points $x_1, x_2 \in C$ and any θ with $0 \leq \theta \leq 1$, we have:*

$$\theta x_1 + (1-\theta)x_2 \in C. \tag{2.1}$$

Thus a set is said to be convex if any two points in the set can be connected by an unobstructed straight line which also lies in the set. Below are definitions of a convex function.

Definition 2 *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if domain of f is convex set and for all x_1, x_2 in* *domain of f, and $\theta \in [0, 1]$, we have*

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2) \tag{2.2}$$

The above definition of a convex function is sometimes referred to as the "0-th order condition" for convexity. If the function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable, then we also have the following definitions.

Definition 3 *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if for all x_1, x_2 in the domain of f, we have*

$$f(x_2) \geq f(x_1) + \nabla^T f(x_1)(x_2 - x_1) \tag{2.3}$$

where

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1} \cdots \frac{\partial f(x)}{\partial x_n} \right] \tag{2.4}$$

The definition states that for a convex function, the first order Taylor approximation is a global under-estimator of the function and hence it is called first order condition. The second order condition is defined by the following definition.

Definition 4 *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if for all x_1, x_2 in the domain of f,*

$$\nabla^2 f(x) \geq 0, \tag{2.5}$$

where Hessian matrix is

$$\nabla^2 f(x) = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right] \tag{2.6}$$

The above definition says that a convex function's derivative is non-decreasing and the graph of the function has positive (upward) curvature at x.

Unconstrained Convex Optimization

An unconstrained convex minimization problem can be stated as below

$$\min_{x} \quad f(x) \tag{2.7}$$

Let X denotes a feasible set for the problem. Suppose that the function is differentiable over a chosen domain and that partial derivatives of at-least first order exists. Then x^* is optimal if and only if $x^* \in X$ and

$$\nabla f(x^*)^T(x - x^*) \geq 0 \text{ for all } x \in X \tag{2.8}$$

The first order necessary and sufficient condition for a relative minimum point x^* of $f(x)$ is

$$\nabla f(x^*) = 0 \tag{2.9}$$

where $\nabla f(x)$ is the gradient of the first variant. Since $f(x)$ is differentiable, its domain is open, so all x sufficiently close to x^* are feasible. Thus the necessary and sufficient condition is automatically derived from (2.8). In the next section, we discuss the gradient descent algorithm to find the solution to the problem iteratively.

Gradient Descent

The objective of the gradient method is a to generate a sequence of vectors, $x^0, x^1 \cdots x^n$ such that : $f(x^{k+1}) < f(x^k)$. Thus the iterative method can be mathematically expressed as :

$$x^{k+1} = x^k + \alpha^k \Delta x^k \tag{2.10}$$

where $\alpha^k > 0$. Here Δx is called the search direction and $k = 0, 1 \cdots$ denotes the iteration number. From convexity and from (2.8), the search direction in the descent method must satisfy

$$\nabla f(x^k) \Delta x^k < 0 \tag{2.11}$$

A natural choice for the search direction is the negative gradient $\Delta x = -\nabla f(x)$ and the resulting algorithm is called the gradient descent method. The algorithm is given below

---

Algorithm 1: Gradient descent method

---

given a starting point $x \in X$ **repeat**

1. $\Delta x := -\nabla f(x)$.

2. Choose a step size α.

3. update $x := x + \alpha \Delta x$

until stopping criteria is satisfied.

The stopping criteria is usually of the form $||\nabla f(x)||_2 < \epsilon$ where ϵ is small and positive.

Constrained Optimization: Lagrangian dual function

In the previous sections, we have shown how to solve unconstrained optimization and use gradient descent method to solve it numerically. In this section, we will tackle convex optimization with both equality and inequality constraints. Let us consider the following optimization problem

$$\begin{aligned} \min \quad & J(x) \\ \text{s.t.} \quad & g_i(x) \le 0 \qquad i = 1, \cdots, m \\ & h_i(x) = 0 \qquad i = 1, \cdots p \end{aligned} \tag{2.12}$$

with variable $x \in \mathbb{R}^n$. We assume the domain $\mathcal{D} = \cap_{i=0}^{m} \textbf{dom } g_i \cap \cap_{i=1}^{p} \textbf{dom } h_i$ is non-empty and the optimal value of (2.12) is p^*. The basic idea in the Lagrangian duality is to take the constraints into account by augmenting the objective function with weighted sum of the constraint functions. We define the Lagrangian L: $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ as follows:

$$L(x, \lambda, \mu) = J(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{p} \mu_i h_i(x) \tag{2.13}$$

where λ_i is the Lagrange multiplier associated with the ith inequality constraint $g_i(x) < 0$ and μ_i is the Lagrange multiplier associated with the ith equality constraint $h_i(x) = 0$. The vectors λ and μ are called the dual variables. From here, we can define the Lagrangian dual function $G : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ as the minimum value of Lagrangian over $x : \lambda \in \mathbb{R}^m, \mu \in \mathbb{R}^p$.

$$G(\lambda, \mu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \mu) = \inf_{x \in \mathcal{D}} \left(J(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{p} \mu_i h_i(x) \right) \tag{2.14}$$

The dual function also yields the lower bound on the optimal value p^*. That is

$$G(\lambda, \mu) \le p^* \tag{2.15}$$

Let us define x^* and λ^*, μ^* as the primal and dual optimal point of the problem (2.12). We assume the functions $J, g_1 \cdots g_m, h_1, \cdots h_p$ are convex and differentiable. The necessary and sufficient conditions for the minimization of problem (2.12) can be given as the Karush-Kuhn-Tucker (KKT) conditions which are stated below

$$\nabla J(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{i=1}^{p} \mu_i^* \nabla h_i(x^*) = 0 \tag{2.16}$$

$$g_i(x^*) \leq 0 \qquad i = 1, \cdots, m \tag{2.17}$$

$$h_i(x^*) = 0 \qquad i = 1, \cdots, p \tag{2.18}$$

$$\lambda_i^* \leq 0 \qquad i = 1, \cdots, m \tag{2.19}$$

$$\lambda_i^* g_i(x^*) \leq 0 \qquad i = 1, \cdots, m \tag{2.20}$$

Many algorithms of convex optimization are basically interpreted as methods for solving the KKT conditions.

Dual ascent and Augmented Lagrangian

So far, we have shown how we can use gradient descent to solve a problem with primal variable by following the negative direction of the gradient. In this section, we will show that the same primal problem can be solved by solving the Lagrangian dual problem using gradient ascent, that is following the positive direction of the gradient for the dual update. Let us consider the following problem with an equality constraint:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Ax = b \end{aligned} \tag{2.21}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. The Lagrangian for the problem (2.21) is given as

$$L(x, \lambda) = f(x) + \lambda^T (Ax - b) \tag{2.22}$$

and the dual function is

$$g(\lambda) = \inf_x L(x, \lambda) = -f^*(-A^T \lambda) - b^T \lambda \tag{2.23}$$

where λ is the dual variable and f^* is the convex conjugate of f. The dual problem is

$$\max g(\lambda) \tag{2.24}$$

Assuming strong duality holds, the optimal values of primal and dual problems are the same. Thus, we can recover the primal optimal point from the dual optimal point as

$$x^* = \arg\min_x L(x, \lambda^*) \tag{2.25}$$

We solve the dual problem using the gradient ascent. The whole algorithm consists of the following iterations

$$x^{k+1} := \quad \arg\min_x L(x, \lambda^k) \tag{2.26}$$

$$\lambda^{k+1} := \quad \lambda^k + \alpha^k(Ax^{k+1} - b) \tag{2.27}$$

where $\alpha^k > 0$ is the step size and k is the iteration counter. To further improve robustness of the dual ascent algorithm, augmented Lagrangian methods were developed. The augmented Lagrangian for (2.21) is given as

$$L_\rho(x, \lambda) = f(x) + \lambda^T(Ax - b) + \frac{\rho}{2}||Ax - b||_2^2 \tag{2.28}$$

where $\rho > 0$ is the penalty parameter. The above problem is similar to problem (2.21) since for any feasible x, the term added to the objective is zero. The augmented Lagrangian can be solved similar to dual ascent as follows

$$x^{k+1} := \quad \arg\min_x L_\rho(x, \lambda^k) \tag{2.29}$$

$$\lambda^{k+1} := \quad \lambda^k + \rho(Ax^{k+1} - b) \tag{2.30}$$

The algorithm is similar to dual ascent for solving problem (2.21) except x-minimization step uses augmented Lagrangian and the penalty term is used as the step size for dual update. The optimality conditions for (2.21) can be easily derived from the KKT conditions as

$$Ax^* - b = 0 \qquad \nabla f(x^*) + A^T\lambda^* = 0 \tag{2.31}$$

which are termed primal and dual feasibility, respectively.

Alternating Direction Method of Multipliers

The ADMM algorithm blends the decomposability of dual ascent with the superior convergence of the method of multipliers. Consider the following problem

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c \end{aligned} \tag{2.32}$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$ and $c \in \mathbb{R}^p$. We assume that f and g are convex. The only difference between the problem above and problem (2.21) is that the variable x is split into two variables, namely x and z, with objective separable across this splitting. Similar to method of multipliers, we define the following augmented Lagrangian

$$L_\rho = (x, z, \lambda) = f(x) + g(z) + \lambda^T(Ax + Bz - c) + \frac{\rho}{2}||Ax + Bz - c||_2^2 \tag{2.33}$$

and it consists of the following iterations

$$x^{k+1} \quad := \arg\min_x L_\rho(x, z^k, \lambda^k) \tag{2.34}$$

$$z^{k+1} \quad := \arg\min_z L_\rho(x^{k+1}, z, \lambda^k) \tag{2.35}$$

$$\lambda^{k+1} \quad := \lambda^k + \rho(Ax + Bz - c) \tag{2.36}$$

where $\rho > 0$ is the penalty parameter. The algorithm is similar to dual ascent and the method of multipliers: it consists of a x-minimization step (2.34), z-minimization step (2.35) and a dual variable update (2.36). The necessary and sufficient optimality conditions of ADMM problem (2.32) are primal feasibility

$$Ax^* + Bz^* - c = 0 \tag{2.37}$$

and the dual feasibility

$$0 \in \partial f(x^*) + A^T \lambda^* \tag{2.38}$$

$$0 \in \partial g(z^*) + B^T \lambda^* \tag{2.39}$$

Assuming the functions f and g are convex and the corresponding unaugmented Lagrangian has a saddle point, ADMM guarantees the following

- Residual convergence. $r^k \to 0$ as $k \to \infty$, where $r = Ax + Bz - c$.
- Objective convergence. $f(x^k) + g(z^k) \to p^*$ as $k \to \infty$ where p^* is the optimal solution to problem (2.32)
- Dual variable convergence $\lambda^k \to \lambda^*$.

CHAPTER 3: DISTRIBUTED MULTI AGENT ADMM FRAMEWORK

In the last decade, the Alternating Direction Method of Multipliers (ADMM) has received much attention due to its ability of decomposing complex optimization problems into a sequence of simpler sub-problems that can be solved asymptotically under certain convex properties [15]. Most recently, it has been applied to many applications in such areas as image processing [4], machine learning [5], resource allocation [6], power system optimization [7] etc. These diverse applications also demand a detailed study of ADMM convergence properties [16, 17].

The convergence speed of ADMM relies on the selection of penalty parameters [18], which is often manually chosen by the user for a specific problem setup. Convergence rate of ADMM is studied, and earlier work include [16, 19]. It is now well established in the literature that, if the objective functions are strongly convex and have Lipschitz-continuous gradients, the basic ADMM algorithms have global linear convergence[8, 9]. The strong convexity conditions are relaxed in [10], and a constant $O(1/n)$ convergence rate is achieved under mild convex assumptions. It is shown in [11] that convergence can be achieved in $O(1/n^2)$ time if at least one of the objective functions is strongly convex. These specific results all use constant penalty parameters and, in practical applications, efficiency of ADMM is highly sensitive to parameter choices and could be improved via adaptive penalty selection [12, 13, 1].

The first approach that comes intuitively is to use different penalty parameters in each iteration. In He et al. [12], an adaptive penalty based on the relative magnitude of primal and dual residuals is proposed to balance their magnitudes. In [18], primal and dual residuals are also used to improve a defined convergence factor while solving a class of quadratic optimization problem using ADMM. In both these cases, the ADMM algorithm is shown to converge, but global computation

Most of the contents of this Chapter have appeared in [14] of primal and dual residuals are required and hence the resulting algorithm is no longer distributed. In [20], distributed ADMM is implemented to minimize locally known convex functions over a network, and the effect of communication weights on the convergence rate is investigated. In [21], the weighted network matrix is adaptively tuned to improve convergence in a consensus-based distributed problem framework using cooperative control. This idea is used in [22] where a consensus based distributed ADMM is formulated with a predefined network structure, for which primal and dual residuals are balanced locally by each agent. However, their adaptive penalty needs to be reset after several iterations to guarantee convergence, which results in much weakened convergence conditions. More recently, adaptive penalty parameters are used in [15] to improve convergence speed by estimating the local curvatures of the dual functions. However, as pointed out in [23], an increase in the number of nodes causes the local curvature estimation to be inaccurate and possibly unstable.

A Lyapunov-based analytical design methodology is proposed to synthesize adaptive penalty parameters for ADMM to ensure convergence and improve convergence time, all in a multi-agent setting. The proposed distributed ADMM algorithm is designed in four steps. First, distributed control gains are embedded into a row-stochastic weighted network connectivity matrix to ensure consistency of the ADMM between its constraints and network connectivity. Second, the entries of weighted network matrix are embedded as the penalty parameters into the augmented Lagrangian for ADMM so they can be adjusted in a distributed manner for each agent to use its local information and optimize its local objective function. Third, utilizing the convex property of the individual agents' objective functions, the standard ADMM formulation is applied to the newly formulated augmented Lagrangian, the resulting ADMM algorithm with adaptive gains is shown to be asymptotically convergent, and its iterative ADMM updating laws are derived. Fourth, using the Lyapunov direct method, adaptive gain updating laws are analytically synthesized, and the improvement of convergence is proven.

Distributed ADMM formulation

Let us consider the following distributed optimization problem with a conforming communication topology among agents:

$$\min \quad \sum_{i\in\mathcal{N}} f_i(x_i) \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}_i} A_{ij}x_j = 0 \quad \text{for} \quad i \in \mathcal{N}, \tag{3.1b}$$

where $\mathcal{N} = \{1, 2, \cdots, N\}$ is the set of agents. For agent i, $\mathcal{N}_i$ denotes the set of its neighbors including itself, $x_i \in \mathbb{R}^n$ is its state vector, $f_i(x_i)$ is its objective function, and A_{ij} are matrices of appropriate dimensions in the linear constrained equations representing the interconnection of the physical layer. It should be noted that the number of columns for matrix A_{ij} should be greater than the number of rows for the problem to be feasible. The problem can be perceived as each agent i trying to optimize its own objective function while satisfying the network level interconnection constraint of (3.1b).The following assumption is made on the individual objective functions.

Assumption 1: *Functions f_i, $i \in \mathcal{N}$, are strictly convex and differentiable, and their gradients denoted by $\nabla_{x_i} f_i(x_i)$ are Lipschitz continuous. The set of optimal solutions to* (3.1) *is not empty, and the corresponding minimum of* (3.1a) *is finite.*

Network of Agents

Local communication in the network is characterized by a bidirectional graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$; specifically, its sensing/communication matrix is binary and of form [24]:

$$S = \begin{bmatrix} 1 & s_{12} & \cdots & s_{1N} \\ s_{21} & 1 & \cdots & s_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ s_{N1} & s_{N2} & \cdots & 1 \end{bmatrix}, \tag{3.2}$$

where $s_{ij} = 1$ if and only if $e_{ij} \in \mathcal{E}$. Matrix S has 1 in the diagonal as every agent knows its own information, and it is equal to the sum of the adjacency matrix and identity matrix. The following assumption ensures conformity and connectivity of the network, and the multi-agent system is visualized in figure 3.1.

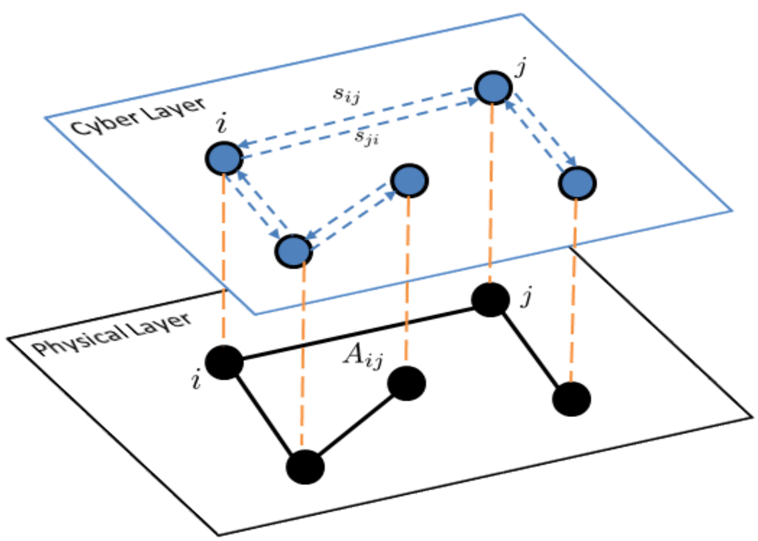


Figure 3.1: Networked cyber-physical system of multi agents.

Assumption 2: *The communication graph conforms with system constraints in the sense that, if* $A_{ij} \neq 0$ *or* $A_{ji} \neq 0$, $s_{ij} = s_{ji} = 1$. *And, the communication graph is connected, i.e., matrix* S *is irreducible.*

Conformity requires that, if agent j is linked to agent i in the ith constraint (3.1b) and through matrix A_{ij}, the two agents have a bidirectional communication channel between them. Connectivity is known [24] to be equivalent to irreducibility. Hence, assumption 2 ensures that the communication network is connected and each agent can optimize its objective function while satisfying all the constraints.

Distributed Adaptive-Gain ADMM

To solve (3.1) using ADMM, we introduce a set of auxiliary variables, z_{ji}, which are the estimates of agent j's variables by agent i [7]. Then, problem (3.1) can be restated as

$$\min \quad \sum_{i \in \mathcal{N}} f_i(x_i) \tag{3.3a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji} = 0 \quad \text{for} \quad i \in \mathcal{N} \tag{3.3b}$$

$$x_i = z_{ij} \qquad j \in \mathcal{N}_i, \;\; i \in \mathcal{N} \tag{3.3c}$$

where z_{ij} are relaxation variables used in the standard ADMM [1]. It should be noted that several existing formulations of consensus based ADMM problem, including those in [25] and [26], are more restrictive than the above. In those setups, each agent tries to reach consensus to a global value which is often the average of the states over the whole network and that common value is assumed to be the global optimal point. In our problem formulation, there is a network level constraint (3.1b) to be satisfied, each agent tries to minimize its own objective function under the constraints, and the resulting optimal solutions for agents are different in general. For this reason, we introduced the variable z_{ij} which is the observation of the state of agent i at agent j. All the agents related to the ith agent try to make their estimates z_{ij} of x_i reach consensus so that a solution to (3.3) converges to an optimal solution x^* of the original problem (3.1). The goal of this reformulation is to solve the optimization problem in a distributed fashion that agent i solves its own optimization sub-problem by exchanging information with its neighboring nodes in set $\mathcal{N}_i$. To this end, we form the so-called augmented Lagrangian as:

$$L_D(x, z, \lambda, \mu) = \sum_{i \in \mathcal{N}} L_i(x_i, z_{ij}, \lambda_{ij}, \mu_i), \tag{3.4}$$

where λ_{ij} and μ_i $(j \in \mathcal{N}_i, i \in \mathcal{N}_i)$, are the Lagrange multipliers (dual variables) associated with the constraints,

$$L_i = f_i(x_i) + \sum_{j \in \mathcal{N}_i} \left[\lambda_{ij}^T (x_i - z_{ij}) + \frac{d_{ij}}{2} ||x_i - z_{ij}||^2 \right] + \mu_i^T \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji} + \frac{w_i}{2} \Big|\Big| \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji} \Big|\Big|^2,$$

and $d_{ij} \geq 0$ are regularized but time-varying penalty parameters from a row-stochastic gain matrix D^k, and $w_i > 0$ is the penalty parameter associated with constraint (3.3b). The augmented Lagrangian reduces to a standard Lagrangian L_0 when the penalty terms are removed (i.e., $d_{ij} = 0$

for all $i \in \mathcal{N}$ and $j \in \mathcal{N}_i$).

To ensure that the proposed adaptive scheme is consistent with local communication network, gain matrix $D^k \in \mathbb{R}^{N \times N}$ are locally calculated (by row) as

$$D^k = \left[d_{ij}^k\right], \quad d_{ij}^k = \frac{s_{ij}\beta_{ij}^k}{\sum_{l=1}^{N} s_{il}\beta_{il}^k}, \tag{3.5}$$

where $k \in \aleph^+$ is the discrete time step, and $\beta_{ij}^k \geq 0$ as local scalar gains (with initial gain values of $\beta_{ij}^0 > 0$). Entries d_{ij}^k (or equivalently β_{ij}^k) will be updated real-time according to the proposed design. Clearly, gain matrix D^k is non-negative, row-stochastic and diagonally positive. The proposed adaptive ADMM approach naturally lends itself to distributed optimization and gives us the flexibility of adjusting the gains on received information. Should all d_{ij}^k become a constant penalty parameter ρ, the proposed design reduces to the standard ADMM algorithm [1].

The ADMM algorithm consists of an x-minimization step, a z-minimization step, and an update of dual variables. The proposed ADMM algorithm is obtained by applying these steps to the above reformulation, that is,

1. For any $i \in \mathcal{N}$, x_i is updated according to

$$x_i^{k+1} := \underset{x_i \in \mathbb{R}^n}{\arg\min}\, L_D(x, z^k, \mu^k, \lambda^k) \tag{3.6a}$$

2. For any $i \in \mathcal{N}$ and for $j \in \mathcal{N}_i$, z_{ji} is solved as

$$z_{ij}^{k+1} := \underset{z_{ij} \in \mathbb{R}^n}{\arg\min} \quad L_D(x^{k+1}, z, \mu^k, \lambda^k) \tag{3.6b}$$

3. For any $i \in \mathcal{N}$ and for $j \in \mathcal{N}_i$, μ_i and λ_{ij} evolves as

$$\mu_i^{k+1} := \mu_i^k + w_i \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji}^{k+1} \tag{3.6c}$$

$$\lambda_{ij}^{k+1} := \lambda_{ij}^k + d_{ij}^k \left[x_i^{k+1} - z_{ij}^{k+1} \right] \tag{3.6d}$$

Convergence property of the proposed ADMM algorithm (3.6) for primal-dual sequences of $\{x_i^k, z_{ij}^k\}$ and $\{\lambda_{ij}^k, \mu_i^k\}$ is summarized as the following lemma:

Lemma 1 *Under assumptions 1 and 2, the distributed ADMM algorithm* (3.6) *is convergent to an optimal solution provided that, for all* i, j

$$\frac{\partial d_{ij}}{\partial x_i} = 0. \tag{3.7}$$

Proof: Let's begin with defining the following error terms: for any $i \in \mathcal{N}$, for $j \in \mathcal{N}_i$, and for $k \in \aleph$,

$$r_{ij}^{k+1} \triangleq [x_i^{k+1} - z_{ij}^{k+1}], \tag{3.8}$$

$$q_i^{k+1} \triangleq \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji}^{k+1}. \tag{3.9}$$

Under assumption 1, problem (3.3) has at least one optimal solution, denoted by $(x_i^*, z_{ij}^*, \mu_i^*, \lambda_{ij}^*)$ for $i \in \mathcal{N}$ and $j \in \mathcal{N}_i$. Since it satisfies the KKT conditions [27], we have

$$L_0(x^*, z^*, \lambda^*, \mu^*) \leq L_0(x^{k+1}, z^{k+1}, \lambda^*, \mu^*),$$

or equivalently,

$$\sum_{i\in\mathcal{N}}\Big[f_i(x_i^*)+\sum_{j\in\mathcal{N}_i}(\lambda_{ij}^*)^T(x_i^*-z_{ij}^*)+(\mu_i^*)^T\sum_{j\in\mathcal{N}_i}A_{ij}z_{ji}^*\Big]$$
$$\leq\sum_{i\in\mathcal{N}}\Big[f_i(x_i^{k+1})+\sum_{j\in\mathcal{N}_i}(\lambda_{ij}^*)^T r_{ij}^{k+1}+(\mu_i^*)^T q_i^{k+1}\Big].$$

Since the optimal solution satisfies the constraints, we know $x_i^*-z_{ij}^*=0$ and $\sum_{j\in\mathcal{N}_i}A_{ij}z_{ji}^*=0$. Hence, the above inequality becomes

$$p^*\leq p^{k+1}+\sum_{i\in\mathcal{N}}\Big[\sum_{j\in\mathcal{N}_i}(\lambda_{ij}^*)^T r_{ij}^{k+1}+(\mu_i^*)^T q_i^{k+1}\Big],\tag{3.10}$$

where $p^*=\sum_{i\in\mathcal{N}}f_i(x_i^*)$ and $p^{k+1}=\sum_{i\in\mathcal{N}}f_i(x_i^{k+1})$. Also, it follows that

$$\lambda_{ij}^{k+1}-\lambda_{ij}^k = (\lambda_{ij}^{k+1}-\lambda_{ij}^*)-(\lambda_{ij}^k-\lambda_{ij}^*),\tag{3.11}$$
$$z_{ij}^{k+1}-z_{ij}^* = (z_{ij}^{k+1}-z_{ij}^k)+(z_{ij}^k-z_{ij}^*),\tag{3.12}$$
$$z_{ij}^{k+1}-z_{ij}^k = (z_{ij}^{k+1}-z_{ij}^*)-(z_{ij}^k-z_{ij}^*),\tag{3.13}$$
$$\mu_{ij}^{k+1}-\mu_{ij}^k = (\mu_{ij}^{k+1}-\mu_{ij}^*)-(\mu_{ij}^k-\mu_{ij}^*).\tag{3.14}$$

It follows from (3.6a) that, under (3.7), the optimality condition for agent i is

$$0=\nabla_{x_i}f_i(x_i^{k+1})+\sum_{j\in\mathcal{N}_i}\lambda_{ij}^k+\sum_{j\in\mathcal{N}_i}d_{ij}^k\Big[x_i^{k+1}-z_{ij}^k\Big].$$

Substituting (3.6d) into the above equation yields

$$0=\nabla_{x_i}f_i(x_i^{k+1})+\sum_{j\in\mathcal{N}_i}\lambda_{ij}^{k+1}+\sum_{j\in\mathcal{N}_i}d_{ij}^k\Big[z_{ij}^{k+1}-z_{ij}^k\Big].$$

The above equation implies that x_i^{k+1} also minimizes

$$f_i(x_i^{k+1}) + \sum_{j \in \mathcal{N}_i} \Big[\lambda_{ij}^{k+1} + d_{ij}^k (z_{ij}^{k+1} - z_{ij}^k)\Big]^T x_i^{k+1}. \tag{3.15}$$

Similarly, it follows from (3.6b) that

$$0 = -\lambda_{ij}^k - d_{ij}^k \Big[x_i^{k+1} - z_{ij}^{k+1}\Big] + A_{ji}^T \mu_j^k + w_j A_{ji}^T \sum_{\phi \in \mathcal{N}_i} A_{\phi i} z_{i\phi}^{k+1}.$$

Substituting (3.6c) and (3.6d) in the above equation yields

$$0 = -\lambda_{ij}^{k+1} + A_{ji}^T \mu_j^{k+1}. \tag{3.16}$$

Applying Lagrange duality to (3.15), we have

$$\begin{aligned} &\sum_{i \in \mathcal{N}} \left[f_i(x_i^{k+1}) + \sum_{j \in \mathcal{N}_i} [\lambda_{ij}^{k+1} + d_{ij}^k (z_{ij}^{k+1} - z_{ij}^k)]^T x_i^{k+1} \right] \\ \le &\sum_{i \in \mathcal{N}} \left[f_i(x_i^*) + \sum_{j \in \mathcal{N}_i} [\lambda_{ij}^{k+1} + d_{ij}^k (z_{ij}^{k+1} - z_{ij}^k)]^T x_i^* \right]. \end{aligned}$$

It follows from (3.16) that

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \left[-\lambda_{ij}^{k+1} + A_{ji}^T \mu_j^{k+1} \right]^T z_{ij}^{k+1} = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \left[-\lambda_{ij}^{k+1} + A_{ji}^T \mu_j^{k+1} \right]^T z_{ij}^*.$$

Adding the above two expressions together and performing simple manipulations, we obtain

$$\begin{aligned} p^{k+1} - p^* \le &\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \Big[-\lambda_{ij}^{k+1} r_{ij}^{k+1} - (\mu_j^{k+1})^T A_{ji} z_{ij}^{k+1} \\ &- d_{ij}^k (z_{ij}^{k+1} - z_{ij}^k)^T (r_{ij}^{k+1} + z_{ij}^{k+1} - z_{ij}^*) \Big]. \end{aligned} \tag{3.17}$$

Combining (3.10) and (3.17) together with (3.9) and multiplying both sides by 2 yield

$$0 \geq \sum_{i\in\mathcal{N}} \Bigg\{ \sum_{j\in\mathcal{N}_i} \Bigg[2(\lambda_{ij}^{k+1} - \lambda_{ij}^{*})^T \underbrace{r_{ij}^{k+1}}_{(5.12)} + 2d_{ij}^k (z_{ij}^{k+1} - z_{ij}^k)^T r_{ij}^{k+1} + 2d_{ij}^k (z_{ij}^{k+1} - z_{ij}^k)^T \underbrace{(z_{ij}^{k+1} - z_{ij}^{*})}_{(3.12)} \Bigg] + 2(\mu_i^{k+1} - \mu_i^{*})^T \underbrace{(A_{ij} z_{ji}^{k+1})}_{(4.23c)} \Bigg] \Bigg\}.$$

Performing the substitutions indicated by the underbraces above, we have

$$0 \geq \sum_{i\in\mathcal{N}} \Bigg[\sum_{j\in\mathcal{N}_i} \Big[\frac{2}{d_{ij}^k} (\lambda_{ij}^k - \lambda_{ij}^{*})^T \underbrace{(\lambda_{ij}^{k+1} - \lambda_{ij}^k)}_{(3.11)} + d_{ij}^k || \underbrace{r_{ij}^{k+1}}_{(5.12)} ||^2 + d_{ij}^k || r_{ij}^{k+1} - (z_{ij}^{k+1} - z_{ij}^k) ||^2 + d_{ij}^k || \underbrace{z_{ij}^{k+1} - z_{ij}^k}_{(3.13)} ||^2 + 2d_{ij}^k \underbrace{(z_{ij}^{k+1} - z_{ij}^k)^T}_{(3.13)} (z_{ij}^k - z_{ij}^{*}) \Big] + \frac{2}{w_i} (\mu_i^{k+1} - \mu_i^{*})^T \underbrace{(\mu_i^{k+1} - \mu_i^k)}_{(3.14)} \Bigg].$$

Using the substitutions indicated by the underbraces above, we obtain

$$0 \geq \sum_{i\in\mathcal{N}} \Bigg\{ \sum_{j\in\mathcal{N}_i} \Bigg[\frac{1}{d_{ij}^k} \Big[||\lambda_{ij}^{k+1} - \lambda_{ij}^{*}||^2 - ||\lambda_{ij}^k - \lambda_{ij}^{*}||^2 \Big] + \frac{1}{w_i} \Big[||\mu_i^{k+1} - \mu_i^{*}||^2 - ||\mu_i^k - \mu_i^{*}||^2 \Big] + d_{ij}^k \Big[||z_{ij}^{k+1} - z_{ij}^{*}||^2 - ||z_{ij}^k - z_{ij}^{*}||^2 \Big] + \frac{1}{w_i} ||\mu_i^{k+1} - \mu_i^k||^2 + d_{ij}^k \Big[||r_{ij}^{k+1} + (z_{ij}^{k+1} - z_{ij}^k)||^2 \Big] \Bigg] \Bigg\}. \tag{3.18}$$

Considering the following Lyapunov function

$$V(k) = \sum_{i\in\mathcal{N}} \Bigg\{ \sum_{j\in\mathcal{N}_i} \Bigg[d_{ij}^k ||z_{ij}^k - z_{ij}^{*}||^2 + \frac{1}{d_{ij}^k} ||\lambda_{ij}^k - \lambda_{ij}^{*}||^2 \Bigg] + \frac{1}{w_i} ||\mu_i^k - \mu_i^{*}||^2 \Bigg] \Bigg\}, \tag{3.19}$$

we can rewrite inequality (3.18) in terms of the Lyapunov function as

$$V^{k+1} - V^k \leq - \sum_{i\in\mathcal{N}} \Bigg[\sum_{j\in\mathcal{N}_i} d_{ij}^k ||r_{ij}^{k+1} + (z_{ij}^{k+1} - z_{ij}^k)||^2 + \frac{1}{w_i} ||\mu_i^{k+1} - \mu_i^k||^2 \Bigg]. \tag{3.20}$$

Inequality (3.20) shows that consensus (3.3c) is ensured and that μ_i converges. Convergence of μ_i ensures constraint (3.3b) is also satisfied. It also follows from (3.6d) that consensus (3.3c) implies convergence of λ_{ij}. These conclude the proof. ■

Real Time Iterative Laws of ADMM

Under assumption 1, the ith agent can choose to solve the sub-optimization problems in (3.6) using a distributed gradient descent technique so solution (3.6a) can be solved real-time but asymptotically. In particular, by taking partial derivatives of the augmented Lagrangian (3.4) with respect to x_i for the x-minimization step, the negative direction of the gradient is followed to real-time iteratively solve for the minimization problem (3.6a). The z-minimization step (3.6b) has a closed form solution since the Lagrangian is quadratic with respect to z_{ij}. The dual variables μ_i and λ_{ij} are also updated using the gradient ascent technique, whose structure is identical to (3.6c) and (3.6d), respectively. Hence, the iterative ADDM (I-ADMM) algorithm for real-time and asymptotic implementation becomes: for agent i, aa

$$\hat{x}_i^{k+1} = \hat{x}_i^k - \alpha_i \Big[\nabla_{\hat{x}_i} f_i(\hat{x}_i^k) + \sum_{j \in \mathcal{N}_i} \big[\hat{\lambda}_{ij}^k + d_{ij}^k (\hat{x}_i^k - \hat{z}_{ij}^k) \big] \Big] \tag{3.21a}$$

$$\hat{z}_{ji}^{k+1} = \hat{x}_j^{k+1} + \frac{1}{d_{ji}^k} \Big[\hat{\lambda}_{ji}^k - A_{ij}^T \hat{\mu}_i^k - w_i^k A_{ij}^T \sum_{\phi \in \mathcal{N}_i} A_{i\phi} \hat{z}_{\phi i}^k \Big] \tag{3.21b}$$

$$\hat{\mu}_i^{k+1} = \hat{\mu}_i^k + w_i \sum_{j \in \mathcal{N}_i} A_{ij} \hat{z}_{ji}^{k+1} \tag{3.21c}$$

$$\hat{\lambda}_{ji}^{k+1} = \hat{\lambda}_{ji}^k + d_{ji}^k (\hat{x}_j^{k+1} - \hat{z}_{ji}^{k+1}) \tag{3.21d}$$

where $0 < \alpha_i << 1$ are the step sizes. For simplicity, α_i are chosen to be small and satisfy $\alpha_1 = \alpha_2 = \cdots = \alpha_N = \alpha$. The real time iterative algorithm (3.21) is a gradient-based optimization and, under convexity, it is guaranteed that its trajectory moves closer to an optimal solution. Convergence results of algorithm (3.21) are summarized in the following lemma:

Lemma 2 *Assume that function $\sum_i f_i(x)$ is strictly convex and has a unique optimal x^*. Then, the following properties hold:*

(i) Dynamic system

$$\sum_i \hat{x}_i^{k+1} = \sum_i \hat{x}_i^k - \alpha \sum_i \nabla_{\hat{x}_i} f_i(\hat{x}_i^k), \tag{3.22}$$

is globally asymptotically stable and convergent to x^* for all sufficiently small α provided that

$$\sup_{\hat{x}_i^k \in \Re^n} \frac{\| \sum_i \nabla_{\hat{x}_i} f_i(\hat{x}_i^k) - \sum_i \nabla_{x^*} f_i(x^*) \|}{\| \sum_i \hat{x}_i^k - x^* \|} < M_f \tag{3.23}$$

is uniformly bounded for a positive Lipschitz constant $M_f > 0$.

(ii) Perturbed dynamic system

$$\sum_i \hat{x}_i^{k+1} = \sum_i \hat{x}_i^k - \alpha \sum_i \nabla_{\hat{x}_i} f_i(\hat{x}_i^k) + \varphi^k, \tag{3.24}$$

is input-to-state stable for all uniformly bounded "disturbance" φ^k.

(iii) The I-ADMM algorithm (3.21) *is convergent to the constrained optimal solution provided that* (3.7) *holds.*

Proof: (i) It follows from (3.22) that

$$e^{k+1} = e^k - \alpha_i \sum_i [\nabla_{\hat{x}_i} f_i(\hat{x}_i^k) - \nabla_{x^*} f_i(x^*)],$$

where $e^k = \sum_i \hat{x}_i^k - x^*$. Considering Lyapunov function $V_e^k = \|e^k\|^2$ and using the concept of Jacobian system (Lemma 1 in [28]), we have

$$
\begin{aligned}
V_e^{k+1} - V_e^k = & \; -2\alpha(e^k)^T \textstyle\sum_i [\nabla_{\hat{x}_i} f_i(\hat{x}_i^k) - \nabla_{x^*} f_i(x^*)] + \alpha^2 \left\| \sum_i \nabla_{\hat{x}_i} f_i(\hat{x}_i^k) - \sum_i \nabla_{x^*} f_i(x^*) \right\|^2 \\
= & \; -2\alpha(e^k)^T \left[\textstyle\sum_i \nabla^2_{\xi^k} f_i(\xi^k) \right] e^k \Big|_{\xi^k = e^k - \tau^k x^*} + \alpha^2 \frac{\| \sum_i \nabla_{\hat{x}_i} f_i(\hat{x}_i^k) - \sum_i \nabla_{x^*} f_i(x^*) \|^2}{\|e^k\|^2} \|e^k\|^2,
\end{aligned}
$$

where $\tau^k \in (0, 1)$ is a constant parameter for any fixed k. Applying strict convexity of $\sum_i f_i(\cdot)$ and the uniform boundedness property (3.23), we know that $[V_e^{k+1} - V_e^k]$ is negative definite for small values of α. Hence, global asymptotic stability and convergence to x^* is established for all sufficiently small values of α.

(ii) It is straightforward to show using the same Lyapunov argument that, under uniformly bounded disturbance φ^k, perturbed dynamic system (3.24) is uniformly bounded.

(iii) It has been shown in lemma 1 that x^* is the optimal solution to algorithm (3.6) and hence to its gradient version, algorithm (3.21). It follows from (ii) that, during the transient, the algorithm remains to be bounded. As the gradient algorithm (3.21) evolves, error φ^k becomes small and diminishing. Thus, asymptotic convergence to x^* can be established by invoking (i). ■

In the implementation, agent i updates not only its own state vector $\hat{x}_i$ but also estimates $\hat{z}_{ji}$ of its neighboring agents' states as well as the associated Lagrange multipliers $\hat{\lambda}_{ji}$ and $\hat{\mu}_i$. This information flow for updating the iterates are shown in figure 3.2.

In most of the existing ADMM literature, penalties d_{ij}^k are set to be constant and identical[8, 9, 10, 11]. Advantage of using adaptive penalty is noted in [12, 13, 1], but those results either require global information or have convergence and scalability issues. These motivate us to develop the proposed adaptive penalty algorithm whose gain matrix is chosen constructively to retain the ADMM's distributed nature while enhancing its scalability and convergence. The dynamic update

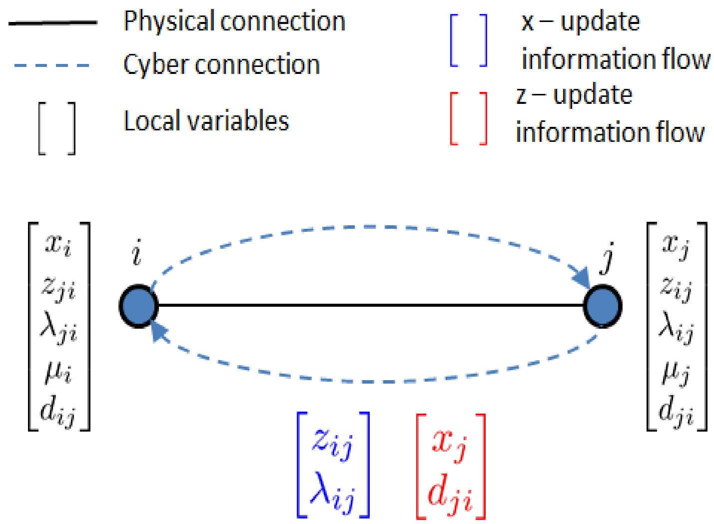


Figure 3.2: Agent i's information flow for local variable update.

laws in (3.21) are the gradients of the augmented Lagrangian with respect to the primal and dual states of each agent. In particular, the ith agent updates $\hat{x}_i$ according to a composite expression consisting of the gradient of its own objective function and the summation of its dual updates and the estimation errors received from its neighbors. Due to the gradient-based optimization under convexity, the algorithm ensures that the solution trajectory moves closer to an optimal solution by following the decreasing direction of the gradient. On the other hand, individually at the ith agent, the ith row entries of matrix D can be adjusted using the control gains β_{ij} embedded therein and, since their adjustments alter the motion direction of trajectory, their choices can also be designed to improve convergence time, and the proposed design is to make the value of an appropriate Lyapunov function decrease more at each of the iteration steps. Specifically, the ith agent dynamically adjusts its penalties to improve convergence time of I-ADMM. This idea was first applied successfully to cooperative control among a network of cooperative agents in [21]. An application of this idea to ADMM is pursued in the next section.

At each of the iteration steps, convergence of I-ADMM algorithm (3.21) can be measured using the following Lyapunov function by agent i:

$$E_i^k = ||\tilde{x}_i^k||^2 + ||\tilde{\mu}_i^k||^2 + \sum_{j \in \mathcal{N}_i} \Big[||\tilde{z}_{ji}^k||^2 + ||\tilde{\lambda}_{ji}^k||^2 \Big], \tag{3.25}$$

where $\tilde{x}^k = \hat{x}^k - \hat{x}^{k-1}$, $\tilde{z}^k = \hat{z}^k - \hat{z}^{k-1}$, $\tilde{\lambda}^k = \hat{\lambda}^k - \hat{\lambda}^{k-1}$ and $\tilde{\mu}^k = \hat{\mu}^k - \hat{\mu}^{k-1}$ are incremental residues of the primal and dual variables. The following theorem provides the proposed distributed adaptive-gain algorithm:

Theorem 1 *Convergence of I-ADMM algorithm* (3.21) *is improved if Lyapunov function* E_i^{k+1} *is made to be more negative through locally and adaptively choosing* β_{ij}^k*. Specifically, for each of* $k \in \aleph^+$*, only two of the penalties* d_{ij}^k *(equivalently, gains* β_{ij}^k*) are adaptively adjusted as*

$$d_{il_i}^k = d_{il_i}^{k-1} + \epsilon_i^k \qquad d_{im_i}^k = d_{im_i}^{k-1} - \epsilon_i^k, \tag{3.26}$$

where indices l_i *and* m_i *are determined according to*

$$l_i \in \mathcal{N}_i \Longrightarrow [\hat{x}_i - \hat{z}_{il_i}] = \max_{j \in \mathcal{N}_i} [(\nabla_{\hat{x}_i} f_i(\hat{x}_i^k))^T (\hat{x}_i^k - \hat{z}_{ij}^k)],$$
$$m_i \in \mathcal{N}_i \Longrightarrow [\hat{x}_i - \hat{z}_{im_i}] = \min_{j \in \mathcal{N}_i} [(\nabla_{\hat{x}_i} f_i(\hat{x}_i^k))^T (\hat{x}_i^k - \hat{z}_{ij}^k)],$$

quantity

$$h_i(\tilde{x}, \tilde{z}) = 2\alpha_i \tilde{x}_i^k \Big[(\tilde{x}_i^k - \tilde{z}_{il_i}^k) - (\tilde{x}_i^k - \tilde{z}_{im_i}^k) \Big] \tag{3.27}$$

is calculated using the locally-available information, and adjustment ϵ_i *is chosen to be: for some*

$0 < \gamma_i < 1$,

$$\epsilon_i^k = \begin{cases} \gamma_i d_{im_i}^{k-1} & \textit{if } h_i(\tilde{x},\tilde{z}) > 0 \\ -\gamma_i d_{il_i}^{k-1} & \textit{if } h_i(\tilde{x},\tilde{z}) < 0 \\ 0 & \textit{otherwise.} \end{cases} \tag{3.28}$$

Proof: First, note that (3.7) holds under the gain adaptations in (3.26) and that the inequality (3.17) still holds and remains the same under gain adaptation because matrix D^k still remains row-stochastic. Hence, both lemmas 1 and 2 holds.

It follows from (3.21) that, for agent i, adaptive gains d_{ij} only appear in (3.21a) but not (3.21b) or (3.21c) or (3.21d). Hence, the expansion of Lyapunov function (3.25) with fixed d_{ij}^k is given by:

$$\begin{aligned} E_i^{k+1} \mid_{d_{ij}^k \text{ not updated}} =& ||\tilde{x}_i^k||^2 - 2\alpha_i(\tilde{x}_i^k)^T\Big[\nabla_{\tilde{x}_i} f_i(\tilde{x}_i^k) + \sum_{j\in\mathcal{N}_i}\tilde{\lambda}_{ij}^k\Big] - 2\alpha_i(\tilde{x}_i^k)^T\sum_{j\in\mathcal{N}_i} d_{ij}^k(\tilde{x}_i^k - \tilde{z}_{ij}^k) \\ &+ ||\tilde{\mu}_i^{k+1}||^2 + \sum_{j\in\mathcal{N}_i}\Big[||\tilde{z}_{ji}^{k+1}||^2 + ||\tilde{\lambda}_{ji}^{k+1}||^2\Big], \end{aligned} \tag{3.29}$$

in which all the terms with α_i^2 are neglected due to $0 < \alpha_i << 1$. Similarly, the expansion of Lyapunov function (3.25) under the adaptive law presented in theorem 1 has the expression that:

$$\begin{aligned} E_i^{k+1}\Big|_{d_{ij}^k \text{ updated}} =& ||\tilde{x}_i^k||^2 - 2\alpha_i(\tilde{x}_i^k)^T\Big[\nabla_{\tilde{x}_i} f_i(\tilde{x}_i^k) + \sum_{j\in\mathcal{N}_i}\tilde{\lambda}_{ij}^k\Big] - 2\alpha_i(\tilde{x}_i^k)^T\Bigg[\sum_{\substack{j\in\mathcal{N}_i \\ j\neq l_i,m_i}} d_{ij}^k(\tilde{x}_i^k - \tilde{z}_{ij}^k) \\ &- (d_{il_i}^k + \epsilon_i)(\tilde{x}_i^k - \tilde{z}_{il_i}^k) - (d_{im_i}^k - \epsilon_i)(\tilde{x}_i^k - \tilde{z}_{im_i}^k)\Bigg] + ||\tilde{\mu}_i^{k+1}||^2 \\ &+ \sum_{j\in\mathcal{N}_i}\Big[||\tilde{z}_{ji}^{k+1}||^2 + ||\tilde{\lambda}_{ji}^{k+1}||^2\Big]. \end{aligned} \tag{3.30}$$

Hence, E_i^{k+1} can assume two different values: one with d_{ij}^k updated according to gain adaptation law (3.26), and another with no adaptation (i.e., $d_{ij}^k = d_{ij}^{k-1}$ for all $j \in \mathcal{N}_i$). The difference in their respective Lyapunov function can be calculated as

$$\Delta E_i^{k+1} := E_i^{k+1}\Big|_{d_{ij}^k \text{ updated using (3.26)}} - E_i^{k+1}\Big|_{d_{ij}^k \text{ not updated}}$$

which after simplification becomes

$$\begin{aligned} &= -\epsilon_i 2\alpha_i \tilde{x}_i^k \Big[(\tilde{x}_i^k - \tilde{z}_{il_i}^k) - (\tilde{x}_i^k - \tilde{z}_{im_i}^k) \Big] \\ &\triangleq -\epsilon_i h_i(x, z) \end{aligned}$$

where $h_i(x,z)$ is defined in (3.27). Thus, the proof is completed by noting that $\Delta E_i^{k+1} < 0$ under choice ϵ_i^k of (3.28). ■

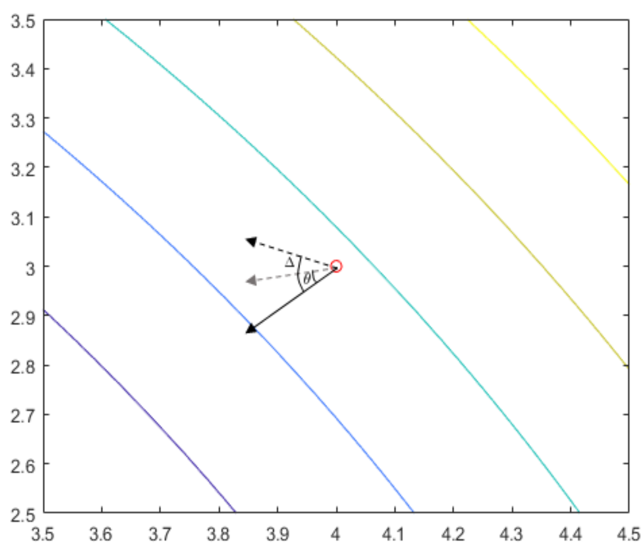


Figure 3.3: Concept illustration of the proposed algorithm

The proposed gain adaptation defined by (3.26) and (3.28) ensures that condition (3.7) holds and the gains with non-zero initial values will remain positive. It should also be noted that all the information required for the ith agent to locally calculate and evaluate $h_i(\tilde{x}, \tilde{z})$ are already gathered from its neighbors. The proposed concept and the proof of theorem 1 can further be illustrated through figure 3.3. For agent i, the solid arrow represents an agent's objective function gradient vector (i.e., $\nabla_{\hat{x}_i} f_i(\hat{x}_i)$). The dotted black and grey arrows represent the vectors of the differences between agent i's state ($\hat{x}_i$) and the observations by the neighbors ($\hat{z}_{ji}$) based on the information received. The black and grey dotted arrows have angles Δ and θ with the gradient vector, respectively, where $\Delta > \theta$. In the proposed algorithm, the vector products (i.e., dot products) between the gradient vector and the difference vectors ($\hat{x}_i - \hat{z}_{ij}$) are calculated, and their maximum (in this case, the vector of smaller angle θ) and their minimum (in this case, the vector of larger angle Δ) are evaluated. Among the ith row entries in the D matrix, the element to be increased by a small amount ϵ_i corresponds to the maximum dot product so that, in solving for the current iteration, that maximum term gets a higher weight. To maintain the convex combination (i.e., the row stochastic property of matrix D), the element to be decreased by the same small amount ϵ_i corresponds to the minimum dot product. This way, the vector most aligned with the gradient vector has more weight than others, and consequently the time derivative of the system Lyapunov function becomes more negative to yield faster convergence.

Simulation Results

In this section, the proposed gain adaptation technique is illustrated through simulations and in two parts. First, the time trajectory of convergence error measure under the proposed adaptive-gain ADMM is compared to that under fixed penalties for a 5-agent network. Second, comparative studies are done for scaled-up networks up to 100 agents and for different network topologies so improvements of convergence speed are established together with scalability.

In all the simulations, agent i aims to minimize its estimation error between the measured state $x_i^m \in \mathbb{R}^2$ and the state estimate $\hat{x}_i \in \mathbb{R}^2$, where $x_i^m = x_i + n_i$ and n_i is the measurement noise in a Gaussian normal distribution. Thus, the individual objective function of agent i can be expressed in terms of the least square error minimization as $f_i(x_i) = \frac{1}{2}||x_i^m - \hat{x}_i||^2$ which is convex and only known to agent i. The primal residual is measured by $\max_i \sum_{j \in \mathcal{N}_i} |\hat{x}_i^k - \hat{z}_{ij}^k|$, and the two dual residuals are given by $\max_i \sum_{j \in \mathcal{N}_i} |\hat{z}_{ij}^{k+1} - \hat{z}_{ij}^k|$ and $\max_i \sum_{j \in \mathcal{N}_i} |\hat{\mu}_{ij}^{k+1} - \hat{\mu}_{ij}^k|$, respectively. These incremental residuals are chosen to measure the convergence speed as they converge to 0 at the optimality. In the implementation of algorithms, the following choices are made: stepsize α being either 0.1 or 0.3, and $w_i(t_0) = 1$ for all $i \in \mathcal{N}$. Tolerance threshold is chosen to be either 1×10^{-3} or 1×10^{-4} [1] in stopping the simulation and determining the number of iterations needed for convergence when comparing various algorithms.

First, let's consider a 5-agent ring network (whose connectivity matrix is cyclic). The I-ADMM algorithm (3.21) is implemented for each agent, and simulations are run twice: one with fixed gains (in which case d_{ij} is computed using (3.5) at $k = 0$ and then kept constant), and another with adaptive gains (whose initial values are calculated the same way and afterwards the gains are updated over time according to theorem 1). Convergence comparison under adaptive-gain ADMM versus fixed-penalty ADMM is shown in figures 3.4, 3.5 and 3.6, respectively, with respect to with respect to primal residual and dual residuals. It should be noted that the number of iterations needed for converge is determined by the primal residual since both dual residuals tend to converge faster.

Fig. 3.7 shows the magnitude (scaled for the purpose of visualization) and direction of the vectors $\nabla_{x_i} f_i(x_i)$, λ_{ij}, and $(x_i - z_{ij})$ in equation (3.21a) as well as the resultant vector, all for agent 1 and super-imposed at the initial point. The primary direction is set by the gradient of the individual

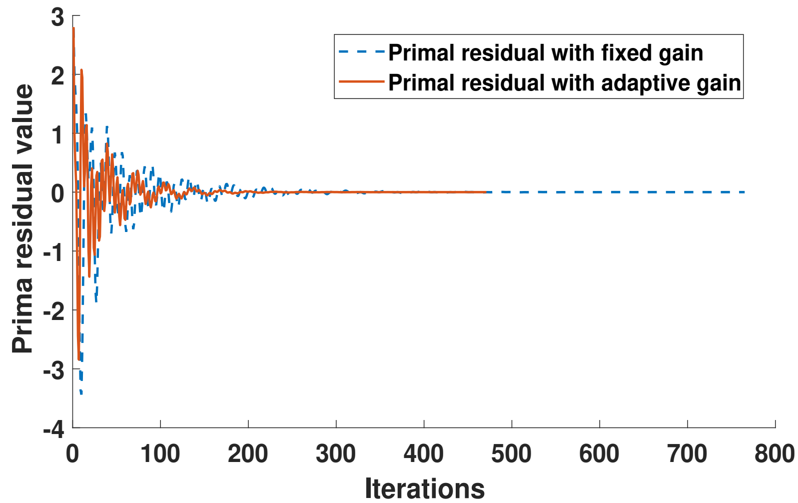


Figure 3.4: Primal residual $(\hat{x}_i^k - \hat{z}_{ij}^k)$: fixed versus adaptive penalty

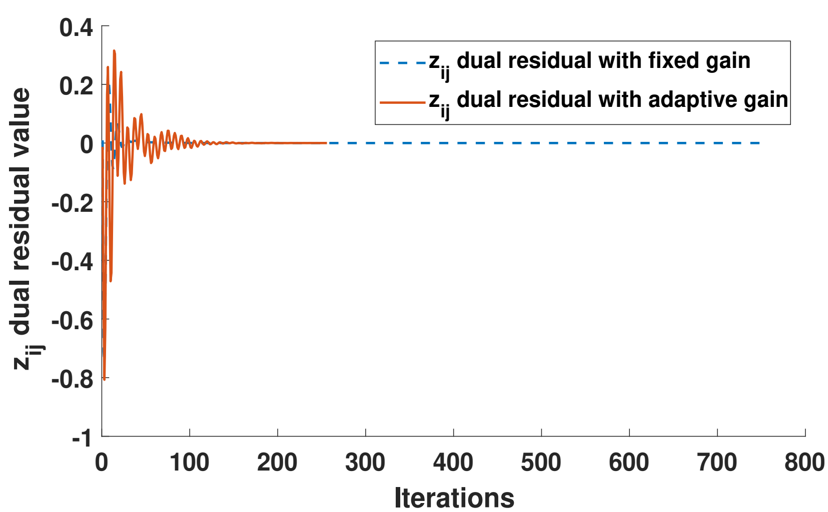


Figure 3.5: Dual residual $(\hat{z}_{ij}^{k+1} - \hat{z}_{ij}^k)$: fixed versus adaptive penalty

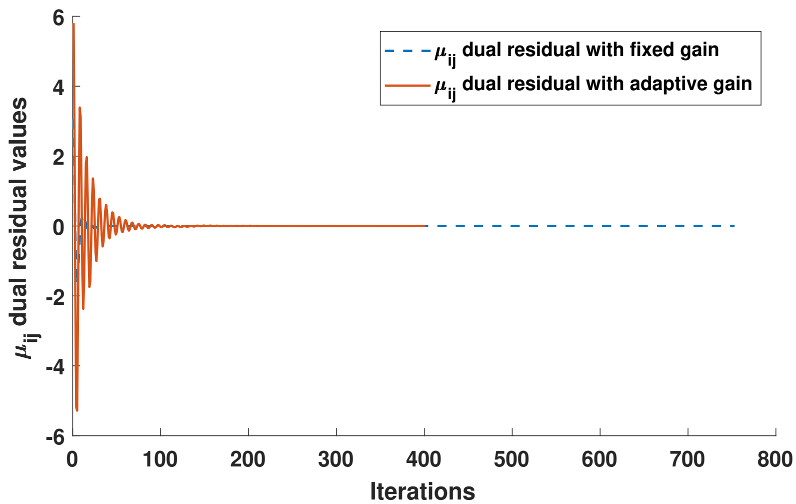


Figure 3.6: Dual residual $(\hat{\mu}_{ij}^{k+1} - \hat{\mu}_{ij}^{k})$: fixed versus adaptive penalty

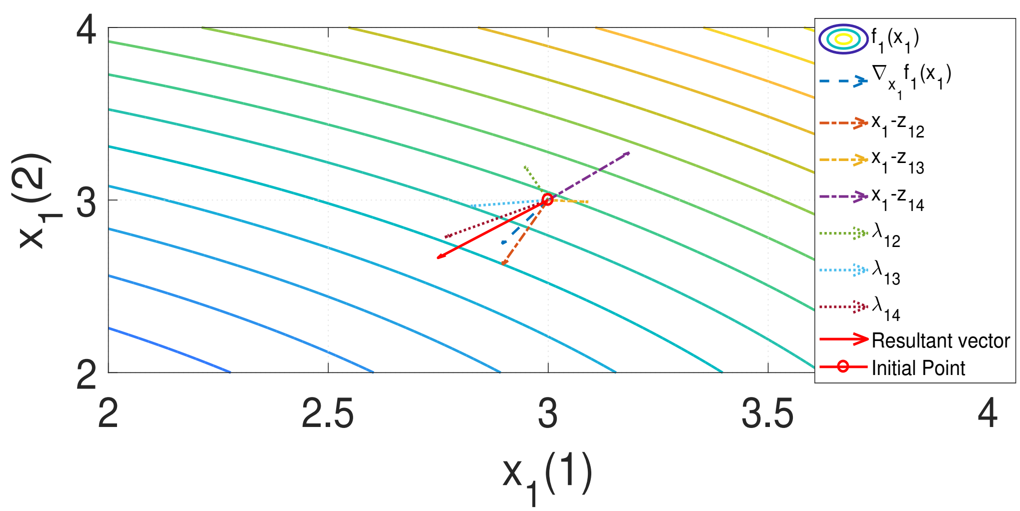


Figure 3.7: Resultant vector from update law: agent 1

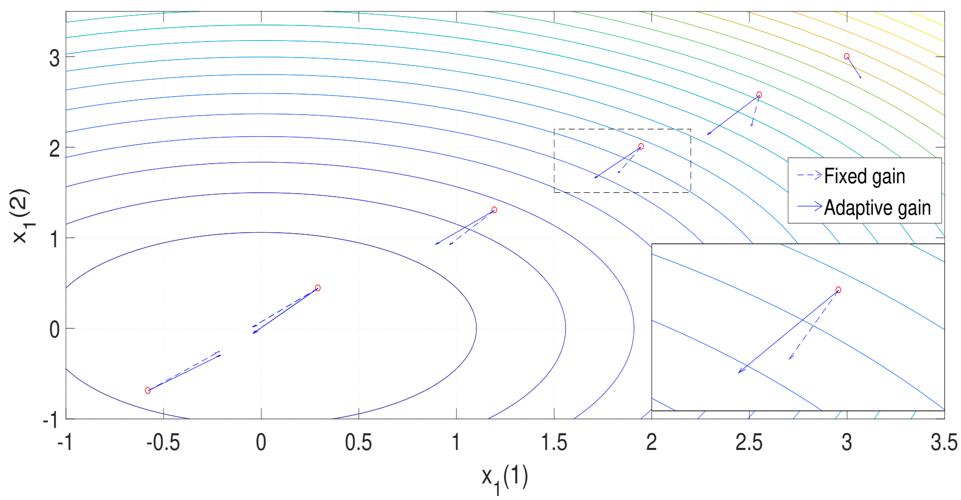


Figure 3.8: Resultant direction comparison between fixed gain versus adaptive penalty: agent 1

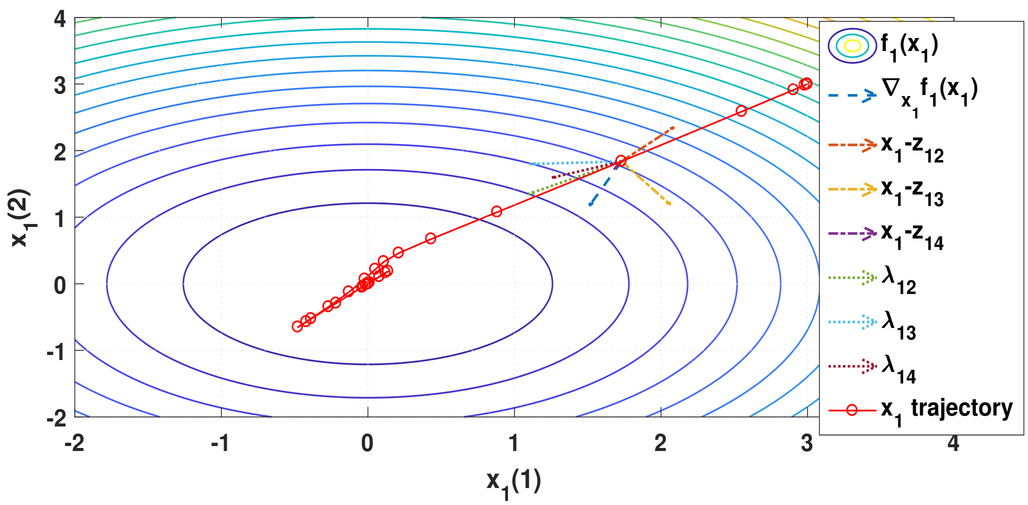


Figure 3.9: The resultant trajectory taken by agent 1 to reach optimality

agent's objective function as each agent is trying to minimize its own objective but the resultant direction is formed by the vector addition given by expression (3.21a). By modifying the weights (corresponding row entries of the D^k matrix) on the information received by agent 1, the proposed algorithm alters the direction of the resultant vector more towards the optimal solution for faster convergence, which is illustrated by figure 3.8. In particular, figure 3.8 shows both the resultant vectors obtained under the fixed penalty gain and the proposed adaptive penalty gain at different stages of the evolution of agent 1's state. It is observed that, under the adaptive algorithm, the resultant vector is closer to the perpendicular of the level curves of the objective function than that under a fixed penalty gain (see the zoomed-in portion of fig. 3.8), which corresponds to the value of Lyapunov function being decreased more (see the proof of theorem 1 in Appendix C). Figure 3.9 shows the resultant vector trajectory taken by agent 1 towards the optimal solution (which is shifted to the origin as $x_1^* = [0, 0]$).

In the second set of simulation studies, the 5-agent network and its scaled-up versions up to 100 agents are simulated for various initial conditions and of different connected network topologies (that are generated by starting with the ring network and then randomly inter-linking each of the agents to up to a maximum of five other agents). Once again, each simulation setup is run twice: one with constant penalty gains, and another with the proposed adaptive penalty gains. For all the networks of certain agents, their convergence times are recorded and their average is recorded/reported for different step sizes and tolerance limits. The maximum iteration limit is set to $30,000$. Table 3.1 provides the comparative summary of the results for networks of different sizes and with different step sizes and tolerance limits. As in any of numerical methods, step size needs to be chosen to be relatively small in order to avoid oscillations around the optimal solution, but it should not be too small to slow the convergence. It is shown that the proposed adaptive gain method does improve the convergence rate no matter what step size is chosen (so long as it ensures convergence). As expected, when a large step size and a very small tolerance ($\alpha_i = 0.3$

and tol= 10^{-4}) are used, the algorithm also fails to converge within the maximum iteration limit.

Table 3.1: Comparative analysis of the algorithms

Iterations required for convergence $\alpha_i = 0.1$, tol = 10^{-3}		
Number of agents	Fixed penalty	Adaptive penalty
5	765	471
10	1138	573
25	5934	1924
50	11968	7342
100	Exceeds iteration limit	20369
Iterations required for convergence $\alpha_i = 0.1$, tol = 10^{-4}		
5	1493	554
10	3412	913
25	23113	5765
50	Exceeds iteration limit	14117
100	Exceeds iteration limit	23429
Iterations required for convergence $\alpha_i = 0.3$, tol = 10^{-3}		
5	886	253
10	2355	862
25	7958	2876
50	11205	7263
100	23429	16173
Iterations required for convergence $\alpha_i = 0.3$, tol = 10^{-4}		
5	1578	442
10	3740	864
25	13379	5615
50	Exceeds iteration limit	24654
100	Exceeds iteration limit	Exceeds iteration limit

The results of our two-part simulation studies clearly demonstrate the convergence improvement under the proposed adaptive-gain ADMM algorithm.

In summary, a distributed multi-agent ADMM algorithm with adaptive gains is developed. The convex properties are utilized to obtain closed form iterative dynamics for the optimization sub-problems. In contrast to the standard ADMM which uses a fixed penalty gain in the augmented

Lagrangian, the proposed algorithm embeds control gains into a row-stochastic matrix based on network connectivity, utilizes the matrix coefficients as the penalty parameters in ADMM, and uses information received by each agent from its neighbors to adaptively adjust these penalties. The proposed adaptive algorithm is both distributed and of closed form, and it substantially improves the rate of ADMM agents' convergence to an optimal solution. The improvement is analytically shown by the Lyapunov direct approach. Numerical simulation demonstrates the effectiveness of the proposed adaptive-gain ADMM.

CHAPTER 4: CONTINUOUS TIME ADMM ALGORITHM FOR SMART GRID

In recent times, the number of Electric Vehicles (EVs) in the distribution grid has increased in large numbers due to the attention from leading automotive industries and the government's push to reduce its greenhouse gas emissions. Consumers are also choosing EVs as their primary vehicle due to the development of charging infrastructure, upgraded battery capacity, and comparable price [29]. Due to this, EV sales in the United States have tripled between 2014 and 2018[30]. As a consequence, a high number of EV penetration has caused substantial degradation in efficiency and reliability of the distribution power grid [31]. As shown in [32, 33], a large number of EV charging at the same time without proper control can lead to abrupt energy peaks and an overall reduction in power quality. This scenario can also lead to an increase in the power loss during transmission and significant voltage drop in the distribution buses. Despite all these adverse effects, most researchers agree that a large number of EVs is a valuable resource that can be controlled to benefit the grid [34].

Over the years, different control schemes have been proposed in the literature. In [35], a centralized optimization scheme was presented to minimize the charging rates and facilitate voltage regulation, but it poses a scalability issue since an increase in the number of vehicles increases the number of control variables adversely. The authors in [36] proposed a centralized optimization framework to minimize the total charging cost based on time-of-use price. Although shown to have good results, however, the authors did not consider the appearance of new load peaks in low price regions causing disruption in the grid operation. Though the centralized schemes presented in [37, 38, 39] have simpler algorithm design, in practicality, it creates high computational complexity and requires high communication bandwidth for centralized data pooling. For these reasons, researchers in recent studies started looking into distributed architectures for controlling EVs.

In most of the distributed control strategies, third party entities called aggregators are considered which acts as a bridge between the Distribution System Operators (DSO) and the EVs [40]. The major purpose of the aggregators is to collect EV information, communicate and send input signals to the EVs based on control strategies, thus taking care of the scaling issues as the number of EVs keep increasing in the grid [41, 42]. In [43], to make the problem into a distributed one, decomposition techniques are applied in the joint optimization of optimal power flow and EV charging and is solved in a nested fashion. A primal-dual subgradient method is implemented in [44] for EV control in a residential distribution network. Although the aforementioned works provide us with a good EV charging algorithm, they somewhat fail to address the network level constraints like voltage fluctuation due to EV control in the distribution grid. In [45], a multi-agent system is presented for EV charging control. The authors investigate the bidding strategy for EV energy injection into the grid and propose energy management strategies based on it. Another multi-agent based control structure is proposed in [46] where the authors design EV charging based on several study factors such as driver behavior, location of charging station, electricity price, etc. In both cases above, the authors focused on the EVs and their charging strategy but did not focus on the distribution grid and how their algorithm is affecting the grid. The work in [47] presents a chance-constrained energy management system based on ADMM where the authors tackle the stochastic randomness of the EVs and solve an EV charging scheduling optimization problem. The solution presented tackles the randomness of the EVs well but failed to show how the proposed charging solution affects the distribution grid. The authors in [48] present a distributed ADMM based multi-period problem where the DSO solved optimal power flow problems with high penetration of EV which are controlled by the aggregators. In their problem, they also ensure that grid security parameters like voltage bound and transmission line limits are maintained. A similar kind of objective of EV charging while maintaining voltage in the grid is also proposed in [49]. In this case, the authors tackled non-separable objective function with coupled power flow constraints and solved it in a decentralized way by providing a hierarchical method based on ADMM. In the above-cited works based on ADMM, iterative methods were used which are in discrete time, and thus the accuracy of the results greatly depends on the sampling time. Due to the intermittent nature of EV charging in the distribution grid, a very small sampling time is preferred to solve for the optimization dynamics in real-time otherwise the optimal solution obtained for a time step might not remain optimal anymore. The work in [31] presented a real-time solution to EV charge scheduling through a dynamic non-cooperative game approach. The authors used ADMM to decompose the centralized problem and realize the solution in real-time. The results are very interesting considering real-time application but then again, the impact of the charging scheduling on the distribution grid parameters like the voltage is neglected in the study.

Given the shortcomings of the existing research, the paper studies a real-time ADMM algorithm to be used to schedule EV charging through aggregators while maintaining the voltage profile of the grid. The contribution of this paper is two-fold: First, we present a novel continuous-domain real-time ADMM algorithm to solve networked multi-agent systems' control problems distributively. With the advent of 5G technologies and smart sensing devices, it is practically possible to obtain data from the field in one-second resolution [50] and implement a control structure in real-time. The existing literature of EV charging combined with a power flow problem is solved iteratively which takes several minutes, thus not utilizing a fair share of the data available as well as not robust to changes that are prevalent in the distribution system with distributed energy resources. Also, since all the parameters in a power grid are in the continuous domain, we can use the proposed algorithm to implement a real-time control. Secondly, we formulate an optimization problem in the distribution system with EV penetration. By using the bi-directional energy transfer capabilities of EVs using a built-in DC-AC converter, EVs can be treated as any other distributed energy resources present in the grid [51, 52]. We set up a distributed optimization problem with power flow equations as the constraint where aggregators establish a contract with EV owners enabling them to use EVs as energy resources for voltage regulation while compensating the owners financially. Since the algorithm is in continuous domain and real-time, the aggregators can have a real-time update of each EVs state of charge and only use them when their state of charge is above a certain predefined threshold which is agreed through the contract. We solve the problem using our developed algorithm and show that the EV charge scheduling can be handled in real-time while keeping the distribution grid parameters within tolerable limits.

Problem Formulation

In this section, we develop the aggregator architecture with EVs and establishes a communication protocol between them. We also present the branch flow model of the distribution grid with aggregators and formulate the optimization problem.

Architecture of EV Aggregators

In the distribution system, we have a hierarchical communication structure. On the top is the Distribution System Operator (DSO) who is responsible for Optimal Power Flow (OPF) calculations and voltage regulation at each node of the grid. Then, on the second layer, we assume the existence of a third party entity, generally termed as ”aggregator” in the literature, which can collect and dispatch the aggregate information of the EVs (cite the wuhan paper). The EV owners can establish a contract with a local aggregator allowing them to utilize the EVs for grid purposes in real-time if it is between a certain State of Charge (SOC) range and in return, they will be compensated financially. The aggregators works in tandem with the DSO for OPF calculation and voltage regulation and works as a ”middleman” between DSO and fleet of EVs. And finally, in the bottom layer, we have the EVs. When they are connected to a charging station, either commercial or residential, can chose to connect with aggregaotr sensor network for ancillary services and share only their SOC and power input/output, thus maintaining the privacy of the owners. The aggregators provide input signals through the same sensor network and ask the EV fleet to inject power into the grid based on the OPF calculations. The communication structure between all the entities including the variables that they handle is illustrated in figure 4.1.

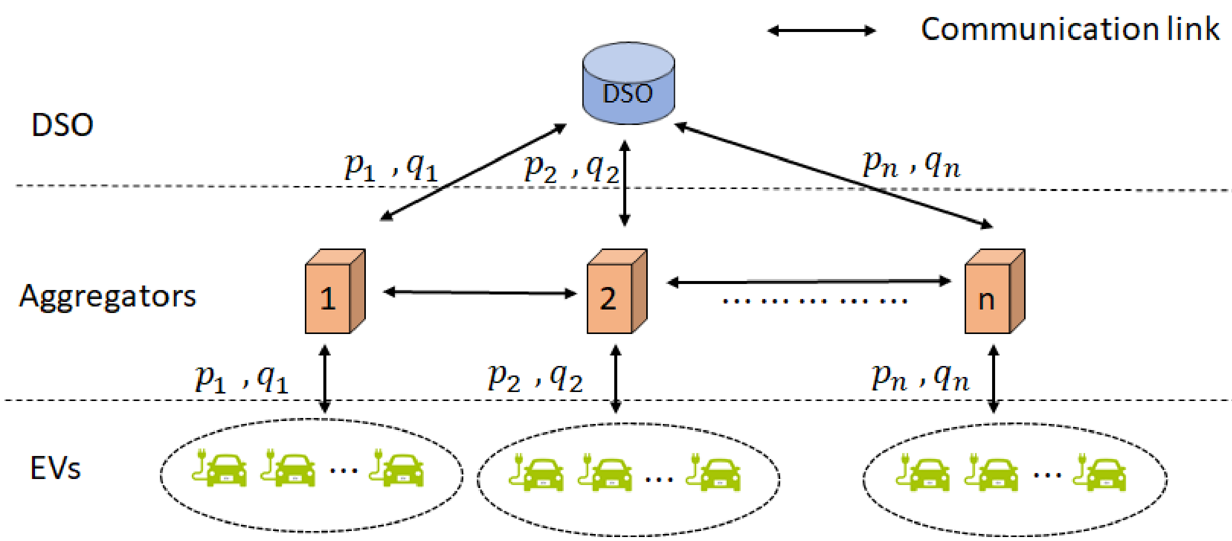


Figure 4.1: The communication structure of aggregators.

It should be noted that the EVs can decide to connect to any aggregator whose sensor network is in range of the EV based on the compensation rates the aggregator is offering. We assume that the aggregators can communicate among themselves, thus, if there are multiple EVs under the same node connected to different aggregators, they can exchange information and can obtain the total aggregated power injection by EVs at that particular node.

The branch flow model was first proposed in [53] which has better numerical stability than the branch injection model. Consider a radial distribution network by a directed graph $G = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} := \{1, \cdots, N\}$ represents the set of buses and $\mathcal{E}$ represents the set of distribution lines connecting the buses in $\mathcal{N}$. Without any loss of generality, the substation of the radial network is indexed by 1. Each node $i \in \mathcal{N} \setminus 1$ has an unique parent node Γ_i and a set of children nodes, denoted by $\mathcal{C}_i$ as shown in figure 4.2. We assume each directed line points towards its children, i.e., power flows from parent Γ_i to node i. We also assume all the parameters are exact and are free of uncertainty.

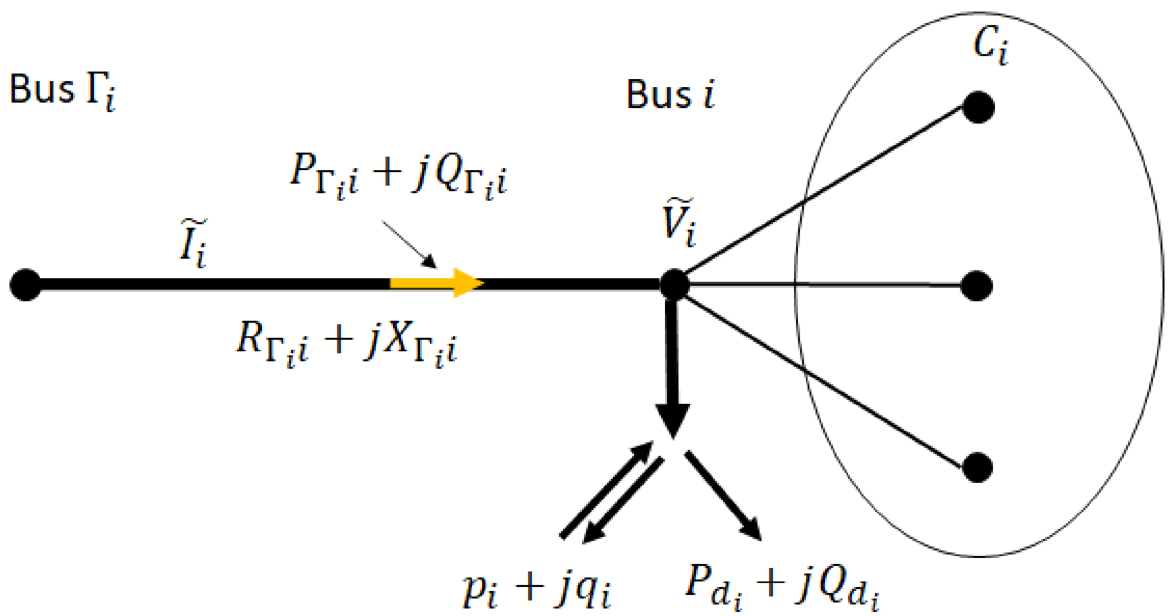


Figure 4.2: A distribution network.

For each bus $i \in \mathcal{N}$, V_i is its voltage with $P_{d_i} + jQ_{d_i}$ being the load demand and $p_i + jq_i$ is the aggregated active and reactive power injection by the aggregated EV since EVs are capable of producing both active and reactive power using four-quadrant charging [54]. For the branch

$\Gamma_i \rightarrow i$, I_i is the current flowing through it with $R_{\Gamma_i i} + jX_{\Gamma_i i}$ being the impedance of the line and $P_{\Gamma_i i} + jQ_{\Gamma_i i}$ being the complex power flowing from the parent Γ_i to node i. For bus $i \in \mathcal{N} \setminus 1$, the power balance equations are given as

$$P_{\Gamma_i i} = p_i + P_{d_i} + \sum_{j \in \mathcal{C}_i} (P_{ij} + R_{ij} l_{ij}) \qquad i \in \mathcal{N} \tag{4.1}$$

$$Q_{\Gamma_i i} = q_i + Q_{d_i} + \sum_{j \in \mathcal{C}_i} (Q_{ij} + X_{ij} l_{ij}) \qquad i \in \mathcal{N} \tag{4.2}$$

where $l_{ij} = I_i^2$ is the square of current magnitude which is defined as

$$l_{ij} = \frac{P_{ij}^2 + Q_{ij}^2}{v_i} \tag{4.3}$$

where $v_i = V_i^2$ is the square of the voltage magnitude. Equation (4.3) is a non-linear equation which can be linearized [55] around current operating point P_{ij}^o, Q_{ij}^o, v_i^o and l_{ij}^o as below:

$$l_{ij} - l_{ij}^o = \frac{2P_{ij}^o}{v_i^o}(P_{ij} - P_{ij}^o) + \frac{2Q_{ij}^o}{v_i^o}(Q_{ij} - Q_{ij}^o) - \frac{(P_{ij}^o)^2 + (Q_{ij}^o)^2}{(v_i^o)^2}(v_i - v_i^o) \tag{4.4}$$

For the substation bus, equations (4.1) and (4.2) takes the following form

$$0 = p_1 + P_{d_1} + \sum_{j \in \mathcal{C}_1} (P_{1j} + R_{1j} l_{1j}) \tag{4.5}$$

$$0 = q_1 + Q_{d_1} + \sum_{j \in \mathcal{C}_1} (Q_{1j} + X_{1j} l_{1j}) \tag{4.6}$$

The power flow on all lines $(i, j) \in \mathcal{E}$ are expressed as

$$v_i - v_j = 2(R_{ij}P_{ij} + X_{ij}Q_{ij}) + (R_{ij}^2 + X_{ij}^2)l_{ij} \tag{4.7}$$

The voltage magnitude limits are given by

$$v_i^{min} \leq v_i \leq v_i^{max} \qquad \forall i \in \mathcal{N} \tag{4.8}$$

From the study done by the authors in [56], the distribution line maximum current flow constraint should take into account the current that flows in the charging susceptance of the line. For the line $(i, j) \in \mathcal{E}$, the thermal limit constraint is given as

$$l_{ij} + \frac{1}{4}B_{ij}^2 v_i + B_{ij}Q_{ij} \leq l_{ij}^{max} \tag{4.9}$$

$$l_{ij} + \frac{1}{4}B_{ij}^2 v_j + B_{ij}(X_{ij}l_{ij} - Q_{ij}) \leq l_{ij}^{max} \tag{4.10}$$

Let us define set of feasible points which satisfy the inequality (4.8) as follows:

$$\mathcal{S}_v = \{v_i : \text{(4.8)} \qquad \forall i \in \mathcal{N}\}, \tag{4.11}$$

and the inequalities (4.9) and (4.10) as follows:

$$\mathcal{S}_l = \{l_{ij} : \text{(4.9)} - \text{(4.10)} \qquad \forall j \in \mathcal{N}_i, i \in \mathcal{N}\}. \tag{4.12}$$

Objective Functions

The EV owners sign a contract with the aggregators to maximize their monetary gain by providing voltage regulation as well as minimize the cost incurred due to charging. Let us define a utility function $U_i(p_i, q_i), i \in \mathcal{N}$ which is concave in nature and expresses the monetary satisfaction of all the EVs connected at the ith bus. Thus, we can define the EV owners welfare function $W_i(p_i, q_i)$ at each bus as follows:

$$W_i(p_i, q_i) = \psi p_i - U_i(p_i, q_i) \qquad i \in \mathcal{N} \tag{4.13}$$

where ψ is the price of electricity. The meaning of the welfare function $W_i(p_i, q_i)$ simple. The EVs incur expense ψp_i by charging their batteries from the grid, but by providing services like voltage regulation, they can earn revenue defined by the concave function $U_i(p_i, q_i)$. The overall welfare function $\sum_{i \in \mathcal{N}} W_i(p_i, q_i)$ is convex in nature since its an addition of a straight line and a negative concave function. It should be noted that the welfare function is a function of aggregated power by the aggregators at each node. The aggregators uses only those EVs whose SOCs are inside a predefined range. The aggregator then compensate the owners based on the contract agreement. Since this work only deals with the DSO and aggregator layer, it is beyond the scope of this paper to define the aggregator-EV relationship and how aggregator compensates the EVs. This issue would be addressed in our future work. For the grid, the DSO and the aggregators want to use the EVs to maintain the voltage at each node close to 1p.u. which can be expressed by the following penalty function which is quadratic and is thus convex:

$$H_i(v_i) = (1 - v_i)^2 \qquad i \in \mathcal{N} \tag{4.14}$$

Therefore, we can define the optimization problem for the whole network as follows:

$$J(p, q, v) = \sum_{i \in \mathcal{N}} J_i(p_i, q_i, v_i) \tag{4.15}$$

$$\begin{aligned} \text{where} \quad & J_i(p_i, q_i, v_i) = k_i W_i(p_i, q_i) + (1 - k_i) H_i(v_i) \\ \text{s.t.} \quad & v_i \in \mathcal{S}_v, l_{ij} \in \mathcal{S}_l \qquad (4.16) \\ & \text{(4.1), (4.2), (4.4)} - \text{(4.7)} \end{aligned}$$

where $< 0k_i < 1$ is the weight on each objective function. The aggregators sum up all the EV power potential at each node and try to minimize the welfare function $\sum_{i \in \mathcal{N}} W_i(p_i, q_i)$ for their customers. The DSO works with the aggregators to obtain EV injection information at each node and wants to maintain a unity voltage profile $\sum_{i \in \mathcal{N}} H_i(v_i)$ in the grid while solving the power flow in the process. In the next section, we will develop a continuous-domain distributed ADMM algorithm to solve problem (4.15) in real time.

Continuous-Domain Distributed ADMM Algorithm

In this section, we develop the continuous-domain real-time ADMM algorithm which can be implemented to a broad class of networked multi-agent distributed optimization and control problems. Then, we show our optimization problem, which is a special case, can be solved distributively using the developed ADMM method in the continuous domain.

Network of Agents

Let us consider a networked multi-agent system which is characterized by a bidirectional graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \{1, 2, \cdots, N\}$ is the number of agents and $\mathcal{E}$ represents the set of edges between them. Also, let node 1 be the virtual leader of the network which means, through the communication network, node 1 can obtain the information of all the nodes in the network if necessary. We can define the network interconnection with the following binary matrix [24]:

$$S = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & s_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & s_{N2} & \cdots & 1 \end{bmatrix}, \tag{4.17}$$

where $s_{ij} = 1$ if and only if $(i \leftrightarrow j) \in \mathcal{E}$, and $s_{ij} = 0$ if otherwise. The matrix S has 1 in the diagonal as every agent knows its own information. We have the following assumption which ensures the connectivity of the network.

Assumption 1: *The matrix S is irreducible, i.e., the communication graph is strongly connected.*

Using the communication matrix S, let us also define a gain matrix D whose values are calculated according to the following equation:

$$D = [d_{ij}] \in \mathbb{R}^{N \times N}, \quad d_{ij} = \frac{s_{ij}\beta_{ij}}{\sum_{l=1}^{N} s_{il}\beta_{il}}, \tag{4.18}$$

where $\beta_{ij} > 0$ are piecewise-constant scalar gains. The matrix D is a non-negative, row stochastic and diagonally positive matrix.

Let us consider the following optimization problem

$$\min \quad \sum_{i\in\mathcal{N}} f_i(x_i) \tag{4.19a}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}_i} A_{ij}x_j = 0 \quad \text{for} \quad i \in \mathcal{N}, \tag{4.19b}$$

For agent i, $\mathcal{N}_i$ denotes the set of its neighboring agents including itself, $x_i \in \mathbb{R}^n$ is its state vector, $f_i(x_i)$ is its objective function, and A_{ij} are matrices of appropriate dimensions which represents the interconnection between the agents. The following assumption is made on the individual objective functions.

Assumption 2: *Functions f_i, $i \in \mathcal{N}$, are convex and differentiable, and their gradients denoted by $\nabla_{x_i} f_i(x_i)$ are Lipschitz continuous. The set of optimal solutions to* (4.19) *is not empty, and the corresponding minimum of* (4.19a) *is finite.*

The goal is to develop a distributed algorithm so that each agent can solve problem (4.19a) while satisfying the linear constraint (4.19b) by exchanging relevant information with its neighboring agents. The problem (4.19) can be solved using ADMM by reformulating it using a secondary variable z_{ji}, which represents the observation of the variables of agent j at agent i. We can then reformulate (4.19) as follows [14]:

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{N}} f_i(x_i) && \text{(4.20a)} \\ \text{s.t.} \quad & \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji} = 0 \quad \text{for} \quad i \in \mathcal{N} && \text{(4.20b)} \\ & x_i = z_{ij} \qquad j \in \mathcal{N}_i, \;\; i \in \mathcal{N} && \text{(4.20c)} \end{aligned}$$

where x and z represents the two set of variables in standard ADMM [1]. In the consensus constraint (4.20c), the observations z_{ij} are forced to equal to the state variable x_i, thus the optimal solution to the problem (4.20) is also optimal to the original problem (4.19). It should be noted that in problem (4.20), (4.20b) involves only z but (4.20c) contains both x and z. Hence, we form the so called "augmented Lagrangian" as follows:

$$\begin{aligned} L(x, z, \lambda) &= \sum_{i \in \mathcal{N}} L_i(x_i, z_{ij}, \lambda_{ij}), \qquad \text{(4.21)} \\ L_i &= f_i(x_i) + \sum_{j \in \mathcal{N}_i} \left[d_{ij} \lambda_{ij}^T (x_i - z_{ij}) + \frac{d_{ij}}{2} ||x_i - z_{ij}||^2 \right] \end{aligned}$$

where λ_{ij} is the dual variable. In the proposed multi-agent ADMM algorithm, the usual constant penalty term is replaced with d_{ij} from D matrix defined in (4.18). This enables the penalty term to conform with the actual physical interconnection of the agents[14]. Because of this reason, the Lagrange multiplier λ_{ij} is also scaled by it. The following theorem provides the continuous-domain solution to the augmented Lagrangian (4.21) whose proof is provided in the Appendix.

Theorem 1: *The augmented Lagrangian* (4.21) *can be solved in the continuous domain using the following dynamics:*

$$\dot{x}_i = -\alpha_i^{'}\left[\nabla_{x_i} f_i(x_i) + \sum_{j\in\mathcal{N}_i} d_{ij}\left[\lambda_{ij} + (x_i - z_{ij}^{-})\right]\right] \tag{4.22a}$$

$$\dot{z}_{ji} = \alpha_i^{'}[d_{ji}\lambda_{ji} + d_{ji}(x_j - z_{ji}) - A_{ij}^T\mu_i] \tag{4.22b}$$

$$\dot{\mu}_i = \sum_{j\in\mathcal{N}_i} A_{ij}z_{ji} \tag{4.22c}$$

$$\dot{\lambda}_{ji} = d_{ji}(x_j - z_{ji}), \tag{4.22d}$$

where $z_{ij}^{-}(t) \triangleq z_{ij}(t-\Delta)$ *for some* $\Delta > 0$ *is a delayed version of* z_{ij}*.*

Proof: ADMM consists of first the x-minimization and then z-minimization, followed by the updates of the dual variables[1]. Following from the augmented Lagrangian (4.21), The algorithm is as follows:

1. x_i is updated according to

$$x_i^{k+1} := \underset{x_i\in\mathbb{R}^n}{\arg\min}\, L_D(x, z^k, \lambda^k, \mu^k), \tag{4.23a}$$

2. For $j \in \mathcal{N}_i$, z_{ji} is solved as

$$z_{ji}^{k+1} := \underset{z_{ji} \in \mathbb{R}^n}{\arg\min} \quad L_D(x^{k+1}, z, \lambda^k, \mu^k)$$
$$\text{s.t.} \qquad \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji} = 0; \tag{4.23b}$$

3. For $i \in \mathcal{N}$ and $j \in \mathcal{N}_i$, μ_i evolves as

$$\mu_i^{k+1} := \mu_i^k + \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji}^{k+1} \tag{4.23c}$$

4. For $j \in \mathcal{N}_i$, λ_{ji} evolves as

$$\lambda_{ji}^{k+1} := \lambda_{ji}^k + d_{ji}^k \Big[x_j^{k+1} - z_{ji}^{k+1} \Big]. \tag{4.23d}$$

where μ_i is the dual variable used to relax the constraint (4.20b) in the z-minimization step [20]. The x-minimization step can be solved iteratively while the other steps have an explicit solution [14]. Under assumption 2, the ith agent can solve the x-minimization sub-problem using the gradient descent technique, that is

$$x_i^{k+1} = x_i^k - \alpha_i \left[\nabla_{x_i} f_i(x_i^k) + \sum_{j \in \mathcal{N}_i} d_{ij}^k \Big[\lambda_{ij}^k + (x_i^k - z_{ij}^{k-}) \Big] \right] \tag{4.24a}$$

where k is the time step and $\alpha_i > 0$ is the step size. The variable z_{ij}^{k-} is a previous solution of z-minimization step and is held constant until x_i^{k+1} reaches optimal solution. For x-minimization step, agent i gathers z_{ij}, λ_{ij} $(j \in \mathcal{N}_i)$ information through the communication network from its neighbors. Then z_{ji} has a closed form solution using the updated x_i as follows

$$z_{ji}^{k+1} = x_j^{k+1} + \frac{1}{d_{ji}^k} \Big[d_{ij}^k \lambda_{ji}^k - A_{ij}^T \mu_i^k \Big], \tag{4.24b}$$

For z-minimization step, agent i obtains x_j, d_{ji} $(j \in \mathcal{N}_i)$ form its neighbors. Finally, the dual updates are given as follows

$$\mu_i^{k+1} = \mu_i^k + \sum_{j\in\mathcal{N}_i} A_{ij} z_{ji}^{k+1}, \tag{4.24c}$$

$$\lambda_{ji}^{k+1} = \lambda_{ji}^k + d_{ji}^k \Big[x_j^{k+1} - z_{ji}^{k+1}\Big]. \tag{4.24d}$$

The dynamic update laws defined in (4.24) is a gradient-based optimization, and assuming assumption 2 holds, the trajectory generated by the primal and dual variables moves closer to the optimal solution. The primal variables generate a trajectory towards the negative direction of the gradient and the dual variables produce trajectory in the positive direction of the gradient. From (4.24a), the amount of distance covered by the trajectory towards optimality in each iteration can be calculated as follows:

$$x_i^{k+1} - x_i^k = -\alpha_i \Bigg[\nabla_{x_i} f_i(x_i^k) + \sum_{j\in\mathcal{N}_i} d_{ij}^k \Big[\lambda_{ij}^k + (x_i^k - z_{ij}^{k-})\Big]\Bigg]$$

If we divide both side by a small number Δt, we get

$$\frac{x_i^{k+1} - x_i^k}{\Delta t} = -\alpha_i^{'} \Big[\nabla_{x_i} f_i(x_i^k) + \sum_{j\in\mathcal{N}_i} d_{ij}^k \Big[\lambda_{ij}^k + (x_i^k - z_{ij}^{k-})\Big]\Big]$$

where $\alpha_i^{'} = \frac{\alpha_i}{\Delta t}$. By taking the limit, we can approximate the discrete time steps into the continuous domain. For x update, we have

$$\lim_{\Delta t \to 0} \frac{x_i^{k+1} - x_i^k}{\Delta t} = \dot{x}_i = -\alpha_i^{'} \left[\nabla_{x_i} f_i(x_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\lambda_{ij} + (x_i - z_{ij}^{-}) \right] \right] \tag{4.25}$$

Following similar procedure for z, λ and μ, we obtain the equations provided in theorem 1. ■

The above solution to the dynamical equation set (4.22) provides us with the optimal solution to the problem (4.20). The convergence results of theorem 1 are summarized in lemma 1.

Lemma 1: *Under assumptions 1 and 2, the distributed ADMM algorithm* (4.22) *is convergent to an optimal solution.*

Proof: Let us consider $x_i^*, z_{ji}^*, \lambda_{ji}^*$ and μ_i^* as the primal-dual optimizer of (4.22) which satisfies the KKT conditions [27]. Thus, we can define the error states as $\tilde{x}_i = x_i - x_i^*$, $\tilde{z}_{ij} = z_{ij} - z_{ij}^*$, $\tilde{\mu}_{ij} = \mu_{ij} - \mu_{ij}^*$ and $\tilde{\lambda}_{ij} = \lambda_{ij} - \lambda_{ij}^*$ and redefine the update law as:

$$\dot{\tilde{x}}_i = -\alpha_i^{'} \left[\eta_i(x_i, x_i^*) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\tilde{\lambda}_{ij} + (\tilde{x}_i - \tilde{z}_{ij}^{-}) \right] \right] \tag{4.26a}$$

$$\dot{\tilde{z}}_{ji} = \alpha_i^{'} [d_{ji} \tilde{\lambda}_{ji} + d_{ji}(\tilde{x}_j - \tilde{z}_{ji}) + A_{ij}^T \tilde{\mu}_i] \tag{4.26b}$$

$$\dot{\tilde{\mu}}_i = w_i \sum_{j \in \mathcal{N}_i} A_{ij} \tilde{z}_{ji} \tag{4.26c}$$

$$\dot{\tilde{\lambda}}_{ji} = d_{ji}(\tilde{x}_j - \tilde{z}_{ji}). \tag{4.26d}$$

where $\eta_i(x_i, x_i^*) = \nabla_{x_i} f_i(x_i) - \nabla_{x_i^*} f_i(x_i^*)$. Consider the Lyapunov function

$$V = \sum_{i=1}^{n} \frac{1}{2\alpha_i^{'}} \left\{ ||\tilde{x}_i||^2 + \alpha_i^{'} ||\tilde{\mu}_i||^2 + \sum_{j \in \mathcal{N}_i} ||\tilde{z}_{ij}||^2 + \alpha_i^{'} ||\tilde{\lambda}_{ij}||^2 + \alpha_i d_{ij} \int_{t-\delta}^{t} ||\tilde{z}_{ij}(\tau)||^2 d\tau \right\}$$

Taking the time-derivative of V yields

$$\begin{aligned}\dot{V} = &\sum_{i=1}^{n} \Bigg\{ - \eta_i(x_i, x_i^*) + \tilde{\mu}_i^T \sum_{j \in \mathcal{N}_i} A_{ij} \tilde{z}_{ji} + \sum_{j \in \mathcal{N}_i} \Bigg[\tilde{x}_i^T \Big[- d_{ij} \tilde{\lambda}_{ij} - d_{ij}(\tilde{x}_i - \tilde{z}_{ij}^-) \Big] \\ &+ \tilde{z}_{ij}^T \Big[d_{ij} \tilde{\lambda}_{ij} + d_{ij}(\tilde{x}_i - \tilde{z}_{ij}) - A_{ij}^T \tilde{\mu}_i \Big] + d_{ij} \tilde{\lambda}_{ij}^T \Big[\tilde{x}_i - \tilde{z}_{ij} \Big] + \frac{1}{2} d_{ij} ||\tilde{z}_{ij}||^2 - \frac{1}{2} d_{ij} ||\tilde{z}_{ij}^-||^2 \Bigg] \Bigg\}\end{aligned}$$

$\eta_i(x_i, x_i^*)$ is positive definite with respect to $\tilde{x}_i$ according to the global under-estimator property of the convex function. Hence we have

$$\begin{aligned}\dot{V} &= \sum_{i=1}^{n} \Bigg\{ - \eta_i(x_i, x_i^*) + \sum_{j \in \mathcal{N}_i} \Bigg[- d_{ij} \tilde{x}_i^T \tilde{x}_i + d_{ij} \tilde{x}_i^T \tilde{z}_{ij}^- + d_{ij} \tilde{z}_{ij}^T \tilde{x}_i - \frac{1}{2} d_{ij} \tilde{z}_{ij}^T \tilde{z}_{ij} - \frac{1}{2} d_{ij} ||\tilde{z}_{ij}||^2 \Bigg] \Bigg\}. \\ &= \sum_{i=1}^{n} \Bigg\{ - \eta_i(x_i, x_i^*) + \sum_{j \in \mathcal{N}_i} \Bigg[\frac{d_{ij}}{2} ||\tilde{x}_i - \tilde{z}_{ij}||^2 + \frac{d_{ij}}{2} ||\tilde{x}_i - \tilde{z}_{ij}^-||^2 \Bigg] \Bigg\}.\end{aligned}$$

which is negative semi-definite with respect to all the variables and is negative definite with respect to $\tilde{x}_i$ as well as both $(\tilde{x}_i - \tilde{z}_{ij})$ and $(\tilde{x}_i - \tilde{z}_{ij}^-)$. ■

Real-Time ADMM Solution to EV Management

In this subsection, we show that the optimization problem (4.15) can be solved in a distributed manner using the above method. The aggregators at each bus and DSO can be considered as agents in the distribution power network, where DSO at the substation is node 1 who is the virtual leader of the network. They communicate relevant information among them for OPF calculation and maintaining voltage regulation. Thus the communication topology between them can be represented by the communication matrix (4.17). Agent i has the information of its own voltage v_i, the aggregated active and reactive power injection p_i, q_i. It also has the information of the amount of active power, reactive power and current that is coming from its parent node. Thus, at agent i, we can define the state variables as $x_i = [v_i, p_i, q_i, P_{\Gamma_i i}, Q_{\Gamma_i i}, l_{\Gamma_i i}]^T$. To solve the problem using continuous-domain ADMM, we introduce an observation vector z_{ji}, which represents the variables of node j observed at node i as $z_{ji} = [v_j^i, p_j^i, q_j^i, P_{\Gamma_j j}^i, Q_{\Gamma_j j}^i, l_{\Gamma_j j}^i]^T$ [14]. With these vector definitions, we can redefine the problem (4.15) according to our developed distributed ADMM problem (4.20) as follows

$$\min_{x_i} \quad \sum_{i \in \mathcal{N}} J_i(x_i) + \mathcal{I}_{v_i} + \mathcal{I}_{l_{ij}} \tag{4.27a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji} + m_{ji} = 0 \qquad \forall i \in \mathcal{N} \tag{4.27b}$$

$$x_i = z_{ij} \qquad \forall i \in \mathcal{N} \tag{4.27c}$$

$$v_i \in \mathcal{S}_v, l_{ij} \in \mathcal{S}_l \tag{4.27d}$$

where $\mathcal{N}_i \triangleq \{\Gamma_i\} \cup \{i\} \cup \mathcal{C}_i$. $\mathcal{I}_{v_i}$ anr $\mathcal{I}_{l_{ij}}$ are indicator functions defined as :

$$\mathcal{I}_{v_i} = \begin{cases} 0, & v_i \in \mathcal{S}_{v_i} \\ \infty & \text{otherwise.} \end{cases} \tag{4.28}$$

$$\mathcal{I}_{l_{ij}} = \begin{cases} 0, & l_{ij} \in \mathcal{S}_{l_{ij}} \\ \\ \infty & \text{otherwise.} \end{cases} \tag{4.29}$$

Equation (4.27b) is based on equations (4.1)-(4.7). Tus, we can define the matrix A_{ij}, based on which agent j represents, as follows

$$A_{ii} = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 1 & 2R_{\Gamma_i i} & 2X_{\Gamma_i i} & 0 & 0 & (R^2_{\Gamma_i i} + X^2_{\Gamma_i i}) \\ 0 & 0 & 0 & \frac{2P^o_{\Gamma_i i}}{v^o_{\Gamma_i}} & \frac{2Q^o_{\Gamma_i i}}{v^o_{\Gamma_i}} & -1 \end{bmatrix}$$

$$A_{ij} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & R_{ij} \\ 0 & 0 & 0 & 0 & 1 & X_{ij} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad j \in \mathcal{C}_i,$$

$$A_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{(P^o_{ij})^2 + (Q^o_{ij})^2}{(v^o_{\Gamma_i})^2} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad j = \Gamma_i, \tag{4.30}$$

and a vector of constants m_{ji} as $m_{ii} = [P_{d_i} \;\; Q_{d_i} \;\; 0; 0]^T$, $m_{ji} = \{[R_{ji} l_{ji} \;\; X_{ji} l_{ji} \;\; 0; 0]^T, j \in \mathcal{C}_i\}$ and $m_{ij} = \{[0 \;\; 0 \;\; (R^2_{ji} + X^2_{ji}) l_{ji}]^T, -\frac{2(P^o_{ji})^2 v^o_j + 2(Q^o_{ji})^2 v^o_j + (P^o_{ji})^2 - (Q^o_{ji})^2}{(v^o_j)^2} - l^o_{ji}, j \in \Gamma_i\}$. Following the procedure in Section III-B, we form the augmented Lagrangian using dual vector $\lambda_{ij} = [\lambda_{ij}(1), \lambda_{ij}(2), \cdots, \lambda_{ij}(6)]^T$ and using theorem 1, obtain the following continuous-domain dynamics for agent i:

x_i update dynamics for each variable:

$$\dot{v}_i = -\alpha_i^{'} \Bigg[\nabla_{v_i} J_i(x_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \Big[\lambda_{ij}(1) + (v_i - v_i^{j-}) \Big] \Bigg]$$

$$\dot{p}_i = -\alpha_i^{'} \Bigg[\nabla_{p_i} J_i(x_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \Big[\lambda_{ij}(2) + (p_i - p_i^{j-}) \Big] \Bigg]$$

$$\dot{q}_i = -\alpha_i^{'} \Bigg[\nabla_{q_i} J_i(x_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \Big[\lambda_{ij}(3) + (q_i - q_i^{j-}) \Big] \Bigg]$$

$$\dot{P}_{\Gamma_i i} = -\alpha_i^{'} \Bigg[\nabla_{P_{\Gamma_i i}} J_i(x_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \Big[\lambda_{ij}(4) + (P_{\Gamma_i i} - P_{\Gamma_i i}^{j-}) \Big] \Bigg]$$

$$\dot{Q}_{\Gamma_i i} = -\alpha_i^{'} \Bigg[\nabla_{Q_{\Gamma_i i}} J_i(x_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \Big[\lambda_{ij}(5) + (Q_{\Gamma_i i} - Q_{\Gamma_i i}^{j-}) \Big] \Bigg]$$

$$\dot{l}_{\Gamma_i i} = -\alpha_i^{'} \Bigg[\nabla_{l_{\Gamma_i i}} J_i(x_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \Big[\lambda_{ij}(6) + (l_{\Gamma_i i} - l_{\Gamma_i i}^{j-}) \Big] \Bigg]$$

The auxiliary primal variable z_{ji} update dynamics is given as:

$$\begin{bmatrix} \dot{v}_j^i \\ \dot{p}_j^i \\ \dot{q}_j^i \\ \dot{P}_{\Gamma_j j}^i \\ \dot{Q}_{\Gamma_j j}^i \\ \dot{l}_{\Gamma_j j}^i \end{bmatrix} = \alpha_i^{'} d_{ji} \begin{bmatrix} \lambda_{ji}(1) \\ \lambda_{ji}(2) \\ \lambda_{ji}(3) \\ \lambda_{ji}(4) \\ \lambda_{ji}(5) \\ \lambda_{ji}(6) \end{bmatrix} + d_{ji} \begin{bmatrix} v_j - v_j^i \\ p_j - p_j^i \\ q_j - q_j^i \\ P_{\Gamma_j j} - P_{\Gamma_j j}^i \\ Q_{\Gamma_j j} - Q_{\Gamma_j j}^i \\ l_{\Gamma_j j} - l_{\Gamma_j j}^i \end{bmatrix} - A_{ij}^T \begin{bmatrix} \mu_i(1) \\ \mu_i(2) \\ \mu_i(3) \\ \mu_i(4) \\ \mu_i(5) \\ \mu_i(6) \end{bmatrix}$$

The dual updates are given as follows:

$$\begin{bmatrix} \dot{\mu}_i(1) \\ \dot{\mu}_i(2) \\ \dot{\mu}_i(3) \\ \dot{\mu}_i(4) \\ \dot{\mu}_i(5) \\ \dot{\mu}_i(6) \end{bmatrix} = \sum_{j \in \mathcal{N}_i} A_{ij} \begin{bmatrix} v_j^i \\ p_j^i \\ q_j^i \\ P_{\Gamma_j j}^i \\ Q_{\Gamma_j j}^i \\ l_{\Gamma_j j}^i \end{bmatrix}$$

$$\begin{bmatrix} \dot{\lambda}_{ji}(1) \\ \dot{\lambda}_{ji}(2) \\ \dot{\lambda}_{ji}(3) \\ \dot{\lambda}_{ji}(4) \\ \dot{\lambda}_{ji}(5) \\ \dot{\lambda}_{ji}(5) \end{bmatrix} = d_{ji} \begin{bmatrix} v_j - v_j^i \\ p_j - p_j^i \\ q_j - q_j^i \\ P_{\Gamma_j j} - P_{\Gamma_j j}^i \\ Q_{\Gamma_j j} - Q_{\Gamma_j j}^i \\ l_{\Gamma_j j} - l_{\Gamma_j j}^i \end{bmatrix}$$

where A_{ij} are defined in (4.30).

Simulation results

In this section, the proposed distributed continuous-domain real-time ADMM algorithm is implemented on the IEEE 123 bus distribution system with EV penetration. First, we simulated a base case with no EV penetration and control to set up a reference point for the simulation. Then we included EVs into the system and implemented our proposed algorithm for EV welfare and voltage optimization with power flow equations as constraints.

First, we set up the IEEE 123 bus distribution system in OpenDSS and ran the base case without any control whose voltage profile is shown in figure 4.3. Then we introduced 200 EVs of different capacities and types into the grid which can be aggregated among 40 buses. Table 4.1 summarises the different types of input power injection range for EVs and table 4.2 shows the types of EVs used in the study and their corresponding battery capacity [57]. In the simulation, the welfare function is defined as a quadratic function: $W_i(p_i, q_i) = a_p p_i^2$ with the value of coefficients taken from the literature [58] and the price of electricity was fixed at an average of 15 cents per kWh. Although the algorithm is capable of handling it, we assumed the EVs only inject active power in this scenario and the total aggregated reactive power injection is 0. To measure the convergence to steady-state, we defined the primal residual as $\max_i \sum_{j\in\mathcal{N}_i} |x_i - z_{ij}^-|$, and the dual residual as $\max_i \sum_{j\in\mathcal{N}_i} |z_{ij} - z_{ij}^-|$ [14, 1]. In the implementation of the algorithm, we chose $\alpha_i^{'}$ to be the same for each node with a value of 0.01 and weight k_i is chosen to be 0.2. In the simulation scenarios, we kept the generation from the sub-station at a fixed point equal to the minimum loading conditions. We introduced several intermittent renewable resources with a random generation profile. We also created a random loading profile where the spot loads at each node vary over time. We introduced our algorithm to all the 40 aggregators to compensate for the intermittent renewable generation and random loading with the objective of keeping the voltage profile within tolerable range as well as maximizing their utilization function. In our simulation, we assumed that there is always enough EVs with acceptable SOC range to compensate for voltage regulation. The simulation was run for 5 hours to observe the impact on the node voltages.

Table 4.1: EV charging categories

Charging method	Voltage	Max. current	Input power
AC level 1	120 V	12 A	1.4 kW
AC level 2	208 - 240 V	32 A	7.2-19.2 kW
DC charging	400 - 1000 V	300 A	50-150 kW

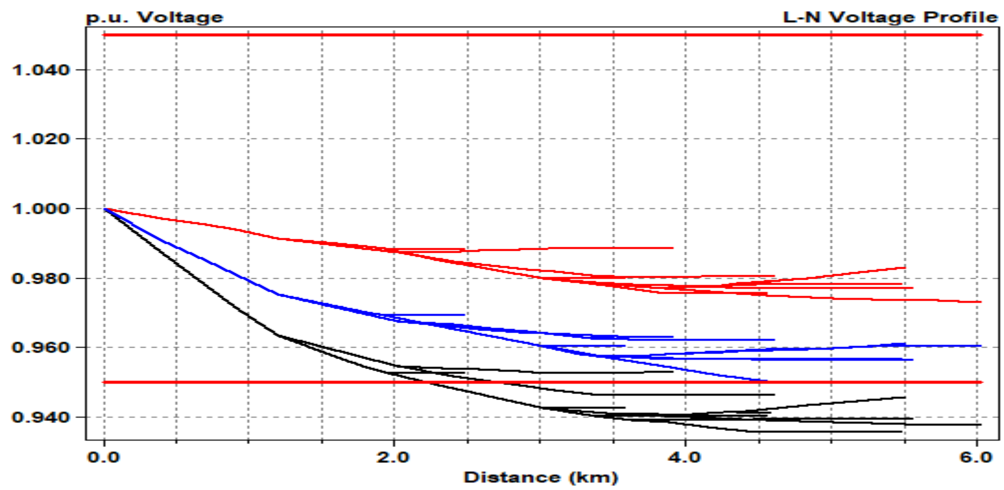


Figure 4.3: Voltage profile of the base case

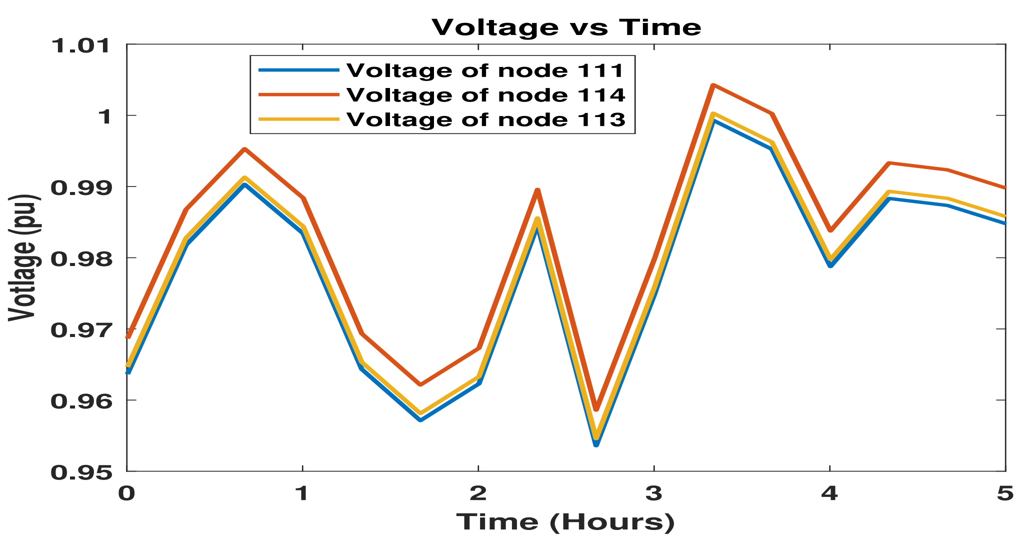


Figure 4.4: Voltages at node 111,113 and 114

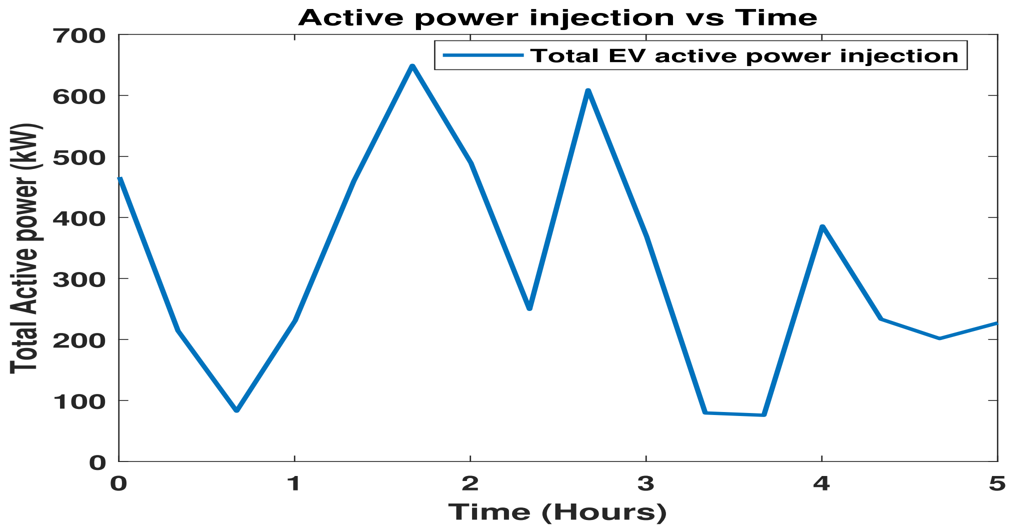


Figure 4.5: Total active power injection by EVs into the grid

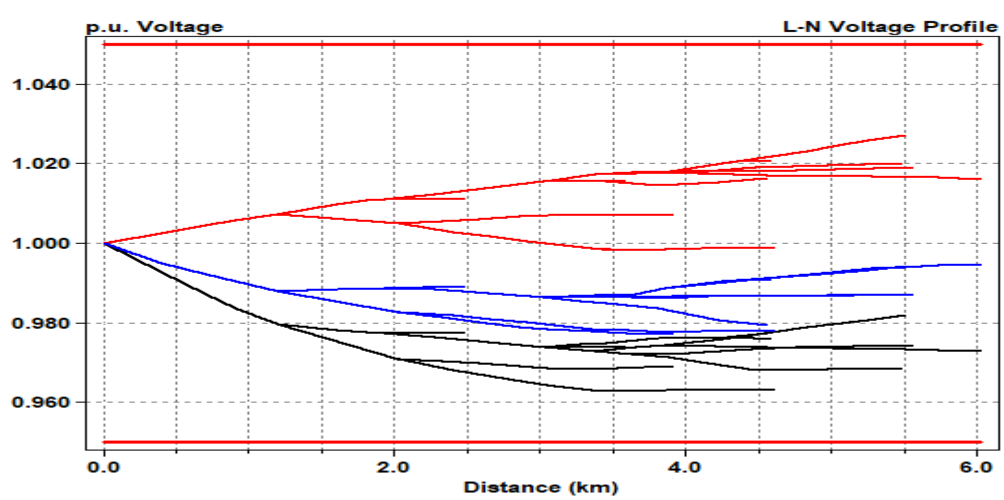


Figure 4.6: Steady state voltage output after using the algorithm

Table 4.2: EV Types

Brand name	Battery capacity
Nissan Leaf	40-62 kWh
Toyota RAV4-EV	41.8 kWh
BMW i3	42.2 kWh
Tesla model 3	75 kWh
Tesla model S	100 kWh
Tesla model X	100 kWh

From the base case, we identified three nodes, namely node 111,113 and 114, which has the three lowest nodal voltage without control in the circuit. Figure 4.4 shows the evolution of the voltages in those three nodes over the five hours where control was applied. Figure 4.5 shows the total injection of active power into the grid by the EVs. It can be seen from both the figure that when ever there is a rise in voltage (between time period 0-1 hour and time period 3-4 hour) due to more generation from the intermittent renewable resources or lower loading condition, the total EV active power injection decreases which means they are charging. The opposite happens whenever there is a loss in voltage due to higher loading condition or loss of renewable resource, the EVs start discharging which maintains the voltage above the lower limit of 0.95 pu. In this way the algorithm makes sure the EV injection follows the voltage profile. Figure 4.6 shows the steady-state voltage at each node of the test circuit.

In summary, a novel continuous-domain real-time distributed multi-agent ADMM algorithm is developed for real-time applications. A distribution system was modeled where EV owners sign contracts with aggregators to allow them to be used for ancillary services like voltage regulation in exchange for monetary gain. A convex optimization problem with power flow equations as constraints were set up where each aggregator tries to minimize the EV charging cost while contributing to ancillary services while DSO would maintain the voltage close to unity and solve the power flow. In the setup, we only consider DSO and aggregators as the acting agents in the grid and assume the aggregators can send signals to EVs to alter their power injection in the grid. We assume the SOC of EVs are within a predefined agreed upon range and there is enough EV reserve for ancillary services. In our future work, we would consider EV optimization with aggregator connection and also tackle SOC evolution which is dynamic in nature. The developed problem with DSO and aggregators was cast into a distributed ADMM framework and is solved using the proposed continuous-domain real-time algorithm by communicating relevant information among them. In contrast to usual discrete-time iterative solution techniques where the accuracy and optimality of the solution depend on the sampling and convergence time, the proposed continuous-domain algorithm can solve the optimization and control problems in real-time. Numerical simulations were carried out on IEEE 123 bus distribution grid to show the effectiveness of the proposed algorithm.

CHAPTER 5: ADMM ALGORITHM WITH DYNAMIC CONSTRAINT

In this chapter, we would focus on developing a multi-layer distributed Alternating Direction Method of Multiplier (ADMM) algorithm which can handle dynamic EV constraints like the state of charge (SOC) to optimize and control EVs in the grid to properly utilize them for voltage regulation and at the same time, ensure that they are compensated by maximizing their utility function and have desired state of charge at the end of their charging period. To analyze multi-layer distributed ADMM, one must understand the basics of original ADMM. Let us consider the problem below [1]

$$\begin{aligned} \min_{x,z} \quad & f(x) + g(z) \\ s.t. \quad & Ax + Bz = c \end{aligned} \tag{5.1}$$

where variables $x \in \mathcal{R}^n$ and $z \in \mathcal{R}^m$. $A \in \mathcal{R}^{p \times n}$, $B \in \mathcal{R}^{p \times m}$ and $c \in \mathcal{R}^p$ are the matrices and vector in the linear constraint. We will assume f and g are convex and differentiable and that their gradients are locally Lipschitz. The problem can be thought of as a general convex linear equality-constrained problem except for the fact that the main optimization variable has been split into two parts, namely x and z in this case, with objective function separable across this splitting, and thus the algorithm is distributed in nature. As with any primal-dual convex optimization, we form the Lagrangian as follows

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}||Ax + Bz - c||_2^2. \tag{5.2}$$

The Lagrangian in this case is called the ”Augmented Lagrangian” due to the fact that an additional penalty term with multiplier $\rho > 0$ is added to the objective function. The problem is solved using ADMM with the following iterations:

$$x^{k+1} := \arg\min_{x} L_\rho(x, z^k, y^k) \tag{5.3a}$$

$$z^{k+1} := \arg\min_{z} L_\rho(x^{k+1}, z, y^k) \tag{5.3b}$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \tag{5.3c}$$

The ADMM algorithm consists of a x-minimization step (5.3a), followed by a z-minimization step (5.3b) and a dual variable update (5.3c). The primal variables x and z are updated in an alternating fashion thus the name ”alternating direction”. The ADMM algorithm of this type can only deal with static linear constraints but falls short when the constraint becomes dynamic in nature. This is very true when we deal with a distribution network with EV penetration since EV itself has a State of Charge (SOC) parameter which is dynamic in nature and evolves with time. To tackle the problem of this nature, in the next section, we will develop our proposed ADMM algorithm with dynamic constraints in the continuous-time domain.

A Distributed Multi-Agent Networked ADMM

In this section of the chapter, we develop a continuous-domain real-time ADMM algorithm with a dynamic constraint that can be implemented to a broad class of networked multi-agent distributed optimization and control problems.

Networked Mutli-Agent System

Let us consider a networked multi-agent system which is characterized by a bidirectional graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \{1, 2, \cdots, N\}$ is the number of agents and $\mathcal{E}$ represents the set of edges between them. Also, let node 1 be the virtual leader of the network which means, through the communication network, node 1 can obtain the information of all the nodes in the network if necessary. We can define the network interconnection with the following binary matrix [24]:

$$S = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & s_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & s_{N2} & \cdots & 1 \end{bmatrix}, \tag{5.4}$$

where $s_{ij} = 1$ if and only if $(i \leftrightarrow j) \in \mathcal{E}$, and $s_{ij} = 0$ if otherwise. That means that if agent i can communicate with agent j and vice versa, we have an entry of 1 in the position (i, j) in the communication matrix. The matrix S has 1 in the diagonal as every agent knows its information. We assume that the communication matrix S is irreducible, i.e., the communication graph is strongly connected.

Figure 5.1 shows the network of multi-agent system. The agents can be connected physically as we will see in the next subsection and it will also be required when we implement the algorithm in a power distribution system.

Distributed Dynamic ADMM

In this subsection, we will formulate a distributed networked multi-agent problem with a dynamic constraint. The work in [14] formulated a similar kind of problem without the dynamic constraint

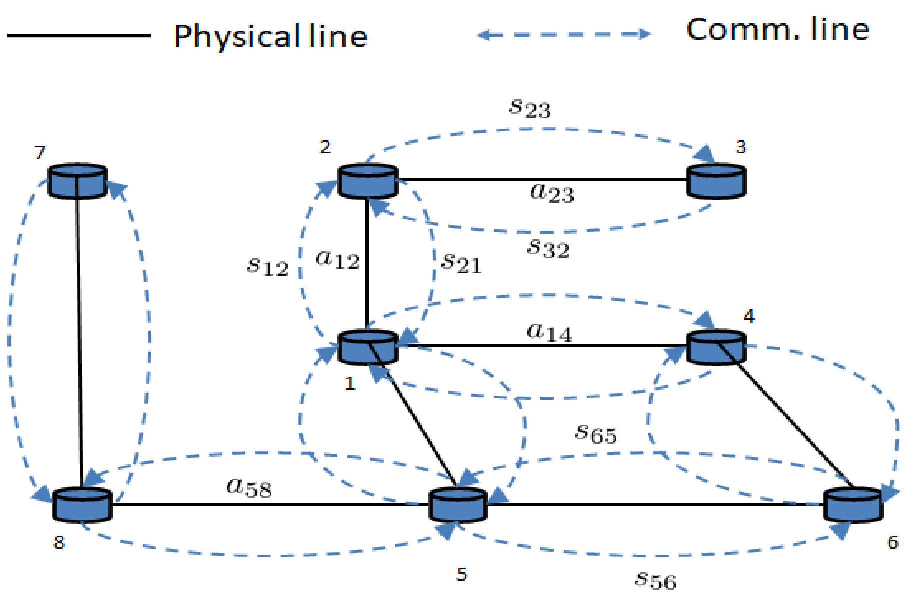


Figure 5.1: A networked multi-agent system with both physical and communication layer

and the solution with convergence proof was provided in the discrete-time domain. In this chapter, we will add a dynamic constraint and tackle the problem in the continuous-time domain.

Consider the following distributed optimization problem:

$$\begin{aligned} \min_{y_i} \quad & \sum_{i\in\mathcal{N}} f_i(y_i) \\ s.t. \quad & \sum_{j\in\mathcal{N}_i} A_{ij} z_{ji} = 0, \quad \forall i \in \mathcal{N}; \quad y_i = z_{ij}, \forall i \in \mathcal{N},\ j \in \mathcal{N}_i, \\ & \dot{x}_i = F_i(x_i) + G_i(x_i)u_i, \quad y_i = H_i(x_i) \end{aligned} \tag{5.5}$$

where

- $x_i \in \mathcal{R}^{n_i}$ is the state of the ith agent
- $y_i \in \mathcal{R}^{l_i}$ is the output of the ith agent
- $f_i(y_i) : \mathcal{R}^{l_i} \to \in \mathcal{R}$ is the objective function of agent i
- $u_i \in \mathcal{R}^{m_i}$ is the control (or decision variable) vector.
- A_{ij} are constant matrices of appropriate dimension.

We assume that the functions $f_i(y_i)$ are convex and differentiable, and their gradients are locally Lipschitz. In the above problem, y and z are the two primal variables used in the original ADMM. The matrix A_{ij} represents the physical interconnection between the agents as shown in figure 5.1. To tackle the problem, let us define the following design principles:

(i) Let

$$u_i = \mathcal{U}_i(x_i) + \omega_i$$

where local feedback control $\mathcal{U}_i$ is designed so that subsystem of x_i is input-to-state stable and that, if $\omega_i \to c_i$ for any constant $c_i, y_i \to c_i$.

(ii) Input ω_i is chosen as the ADMM law.

(iii) Using the communication matrix S, let us also define a gain matrix D whose values are calculated according to the following equation:

$$D = [d_{ij}] \in \mathbb{R}^{N \times N}, \quad d_{ij} = \frac{s_{ij}\beta_{ij}}{\sum_{l=1}^{N} s_{il}\beta_{il}}, \tag{5.6}$$

where $\beta_{ij} > 0$ are piecewise-constant scalar gains. The matrix D is a non-negative, row stochastic and diagonally positive matrix.

With the above design principle, we can redifne the optimization problem as follows:

$$\min_{\omega_i} \quad \sum_{i \in \mathcal{N}} f_i(y_i) \tag{5.7a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji} = 0 \qquad j \in \mathcal{N}_i, i \in \mathcal{N} \tag{5.7b}$$

$$y_i = z_{ij} \qquad j \in \mathcal{N}_i, i \in \mathcal{N} \tag{5.7c}$$

$$\dot{x}_i = F_i(x_i) + G_i(x_i) u_i \tag{5.7d}$$

$$u_i = \mathcal{U}_i(x_i) + \omega_i \tag{5.7e}$$

Now we can form the so called augmented Lagrangian as follows:

$$L(u, z, \lambda) = \sum_{i \in \mathcal{N}} L_i(u_i, z_{ij}, \lambda_{ij}) \tag{5.8}$$

$$L_i = f_i(y_i) + \sum_{j \in \mathcal{N}_i} \left[d_{ij} \lambda_{ij}^T (y_i - z_{ij}) + \frac{d_{ij}}{2} ||y_i - z_{ij}||^2 \right]$$

It should be noted that only the consensus constraint (5.7c) is used in the augmented Lagrangian since it is the only constraint that contains both the primal variables. The rest of the constraints are taken into account while solving the individual sub-problems. Also, the penalty parameter term in the augmented Lagrangian is replaced with entries d_{ij} from the D matrix defined in the design principle [14]. This enables the penalty term to conform with the actual interconnection of the agents. Because of this reason, the dual variable λ is also scaled by the d_{ij} term. The augmented Lagrangian (5.8) is solved using ADMM by solving the following sub-problems in an alternating sequential manner:

1. For any $i \in \mathcal{N}$, ω_i is updated according to

$$\dot{\omega}_i = \arg\min_{x_i \in \mathbb{R}^n} L(y, z^-, \lambda) \tag{5.9}$$

2. For any $i \in \mathcal{N}$ and for $j \in \mathcal{N}_i$, z_{ji} is solved as

$$\dot{z}_{ji} = \arg\min_{z_{ji} \in \mathbb{R}^n} \quad L(y, z, \lambda) \tag{5.10}$$

$$\text{s.t.} \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji} = 0 \tag{5.11}$$

3. For any $i \in \mathcal{N}$ and for $j \in \mathcal{N}_i$, λ_{ji} evolves as

$$\dot{\lambda}_{ji} = \arg\max_{\lambda_{ji} \in \mathbb{R}^n} L(y, z, \lambda) \tag{5.12}$$

where $z^- \triangleq z(t^-)$ is the immediate past solutions to the problems of (5.11). We use the delayed version of z to mimic the alternating behavior of the ADMM algorithm. Using the convex properties and techniques from [14], we obtain the continuous-time dynamics including dynamic constraints for the solution of sub-problems (5.9) - (5.12) as follows:

$$\dot{\omega}_i = -\alpha_i \left[\bigtriangledown_{y_i} f_i(y_i) + \sum_{j\in\mathcal{N}_i} d_{ji}\lambda_{ij} + \sum_{j\in\mathcal{N}_i} d_{ij}(\xi_i - z_{ij}^-) \right] \tag{5.13a}$$

$$\dot{z}_{ji} = -\alpha_i[-d_{ij}\lambda_{ji} - d_{ji}(\xi_j - z_{ji}) + A_{ij}^T\mu_i] \tag{5.13b}$$

$$\dot{\mu}_i = w_i \sum_{j\in\mathcal{N}_i} A_{ij} z_{ji} \tag{5.13c}$$

$$\dot{\lambda}_{ji} = d_{ji}(\xi_j - z_{ji}). \tag{5.13d}$$

where μ_i is the dual variable associated with the constraint (5.11) of z-minimization sub-problem. The variable ξ_i is replacing ω_i in all the dynamics equations which is to be designed using the passivity-short framework[59]. Let us also define the error states as $\tilde{e}_i = e_i - e_i^*$ where $e = \{u_i, z_{ij}, \lambda_{ij}, \mu_i, x_i, y_i\}$. With this, the error dynamics are given as

$$\dot{\tilde{\omega}}_i = -\alpha_i \left[(\bigtriangledown_{y_i} f_i(y_i) - \bigtriangledown_{y_i^*} f_i(y_i^*)) + \sum_{j\in\mathcal{N}_i} d_{ij}\tilde{\lambda}_{ij} + \sum_{j\in\mathcal{N}_i} d_{ij}(\tilde{\xi}_i - \tilde{z}_{ij}^-) \right] \tag{5.14a}$$

$$\dot{\tilde{z}}_{ji} = -\alpha_i[-d_{ji}\tilde{\lambda}_{ji} - d_{ji}(\tilde{\xi}_j - \tilde{z}_{ji}) + A_{ij}\tilde{\mu}_i] \tag{5.14b}$$

$$\dot{\tilde{\mu}}_i = w_i \sum_{j\in\mathcal{N}_i} A_{ij}\tilde{z}_{ji} \tag{5.14c}$$

$$\dot{\tilde{\lambda}}_{ji} = d_{ji}(\tilde{\xi}_j - \tilde{z}_{ji}). \tag{5.14d}$$

We present theorem 1 below which shows that the error state dynamic equations (5.13) obtained converges to the optimal solutions.

Theorem 1: *Consider the statically and dynamically constrained optimization problem* (5.7)*. Sup-*

pose that $\mathcal{U}_i$ is designed such that, for constant matrices C_{il}, there is an individual positive definite Lyapunov function $V_i^{'}(x_i, \omega_i)$ satisfying the following inequality:

$$\begin{aligned}&\Big[\nabla_{x_i} V_i^{'}(x_i, \omega_i)\Big]^T \Big[F_i(x_i) + G_i(x_i)\mathcal{U}_i + G_i(x_i)\omega_i\Big] + \Big[\nabla_{\omega_i} V_i(x_i, \omega_i)\Big]^T \dot{\omega}_i \\ &\leq -\sum_l k_i||\omega_i - C_{il}x_i||^2 + \left[\sum_l (\omega_i - C_{il}x_i)\right]^T \dot{\omega}_i + (y_i - v_i)\dot{\omega}_i\end{aligned} \tag{5.15}$$

Then, continuous-time ADMM algorithm (5.14) *with*

$$\xi_i = y_i + \sum_l (\omega_i - C_{il}x_i)$$

is globally convergent to the optimal solution.

Proof: Consider the following properties of the convex objective function $f_i(y_i)$:

- The functions $f_i(y_i)$ is convex and differentiable. In particular, the gradient of a convex function is a global under-estimator as:

  $$f_i(y_i) - f_i(y_i^*) \geq \nabla_{y_i^*} f_i(y_i^*)(\tilde{y}_i). \tag{5.16}$$

  and

  $$f_i(y_i^*) - f_i(y_i) \geq \nabla_{y_i}^T f_i(y_i)(-\tilde{y}_i) \tag{5.17}$$

  Adding the above two inequalities yields

  $$[-\nabla_{y_i}^T f_i(y_i) + \nabla_{y_i^*}^T f_i(y_i^*)]^T \tilde{y}_i \leq 0. \tag{5.18}$$

- The gradient of $f_i(y_i)$ (denoted by $\nabla_{y_i} f(y_i)$) is Lipschitz, i.e., $||\nabla_{y_i} f(y_i) - \nabla_{y_i^*} f(y_i^*)|| \leq$

$l_i||y_i - y_i^*||$, where l_i is Lipschitz constant.

From (5.18), we get $[\nabla_{y_i}^T f_i(y_i) - \nabla_{y_i^*}^T f_i(y_i^*)]^T \tilde{y}_i$ which is positive definite with respect to $\tilde{y}$.

Lemma 1: If the gradients denoted by $\nabla_{y_i} f_i(y_i)$ are locally Lipschitz, the sum

$$-\sum_l k_i||\epsilon_{il}||^2 - \alpha_i \left[\sum_l \epsilon_{il}\right]^T \nabla_{y_i} f_i(y_i) - \alpha \tilde{y}_i^T \nabla_{y_i} f_i(y_i) \tag{5.19}$$

is negative definite with respect to ϵ_{il} and $\tilde{y}_i$ for small stepsize α_i and for all gain $k_i > 0$ above a certain threshold.

Consider the Lyapunov function

$$\begin{aligned} V =& \frac{1}{2}\sum_{i=1}^{N}\left\{2V_i^{'}(x_i,\omega_i) + ||\tilde{\omega}_i||^2 + \frac{1}{w_i}||\tilde{\mu}_i||^2 + \sum_{j\in\mathcal{N}_i}\left[\frac{1}{\alpha_i}||\tilde{z}_{ij}||^2 + ||\tilde{\lambda}_{ij}||^2\right.\right. \\ &+ \left.\left. d_{ij}\int_{t-\delta}^{t}||\tilde{z}_{ij}(\tau)||^2 d\tau\right]\right\} \end{aligned}$$

It follows from (5.15) that

$$\dot{V} \leq \sum_{i=1}^{N} \Bigg\{ -\sum_{l} k_i ||\tilde{\omega}_i - C_{il}\tilde{x}_i||^2 + \Bigg[\sum_{l}(\tilde{\omega}_i - C_{il}\tilde{x}_i)\Bigg]^T \dot{\tilde{\omega}} + \tilde{y}_i^T \dot{\tilde{\omega}}_i + \tilde{\mu}_i^T \dot{\tilde{\mu}}_i$$
$$+ \sum_{j\in\mathcal{N}_i} \Bigg[\tilde{z}_{ij}^T \dot{\tilde{z}}_{ij} + \tilde{\lambda}_{ij}^T \dot{\tilde{\lambda}}_{ij} + \frac{1}{2} d_{ij}\Big[||\tilde{z}_{ij}||^2 - ||\tilde{z}_{ij}^-||^2\Big]\Bigg] \Bigg\}$$
$$= \sum_{i=1}^{N} \Bigg\{ -\sum_{l} k_i ||\omega_i - C_{il}\tilde{x}_i||^2 - \alpha_i \Bigg[\sum_{l} \omega_i - C_{il}\tilde{x}_i\Bigg]^T \nabla_{y_i} f_i(y_i)$$
$$- \alpha\tilde{y}_i^T \nabla_{y_i} f_i(y_i) + \sum_{j\in\mathcal{N}_i} \Bigg[\tilde{\mu}_i A_{ji}\tilde{z}_{ij} + \tilde{\xi}_i^T\Big[-d_{ij}\tilde{\lambda_{ij}} - d_{ij}(\tilde{\xi}_i - \tilde{z}_{ij}^-)\Big]$$
$$+ \tilde{z}_{ij}^T\Big[d_{ij}\tilde{\lambda}_{ij} + d_{ij}(\tilde{\xi}_i - \tilde{z}_{ij}) - A_{ji}^T\tilde{\mu}_i\Big] + d_{ij}\tilde{\lambda}_{ij}^T\Big[\tilde{\xi}_i - \tilde{z}_{ij}\Big]$$
$$+ \frac{1}{2} d_{ij}||\tilde{z}_{ij}||^2 - \frac{1}{2} d_{ij}||\tilde{z}_{ij}^-||^2\Bigg]\Bigg\}$$
$$= \sum_{i=1}^{N} \Bigg\{ \underbrace{-\sum_{l} k_i ||\omega_i - C_{il}\tilde{x}_i||^2 - \alpha_i \Bigg[\sum_{l} \omega_i - C_{il}\tilde{x}_i\Bigg]^T \nabla_{y_i} f_i(y_i) - \alpha\tilde{y}_i^T \nabla_{y_i} f_i(y_i)}_{\text{n.d. according to (5.19)}}$$
$$- \frac{1}{2} d_{ij}||\tilde{\xi}_i - \tilde{z}_{ij}||^2 - \frac{1}{2} d_{ij}||\tilde{\xi}_i - \tilde{z}_{ij}^-||^2$$

which is negative definite with respect to $(\tilde{\xi}_i - \tilde{z}_{ij})$ as well as $(\tilde{\xi}_i - \tilde{z}_{ij}^-)$. This concludes the proof. ■

The dynamic ADMM problem (5.5) can be also proved through the concept of passivity-short systems. It can be shown that equation set (5.14) can be cast into a passivity-short system. Consider the following state space diagram

The system in figure 5.2 represents the dynamics from (5.14). The whole system can be broken down into several subsystems and can be proved to be passivity short.

The gradient system can be presented as follows The above system is passive with input $\tilde{y}_i$ and

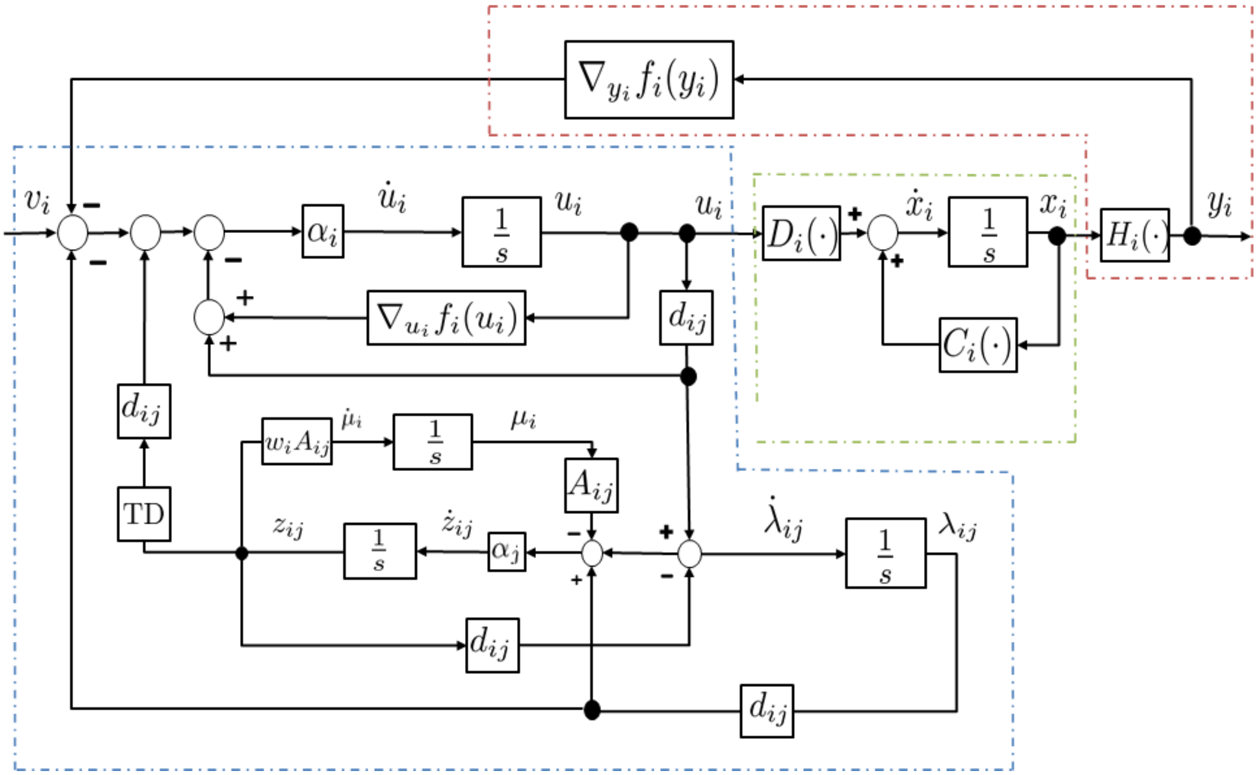
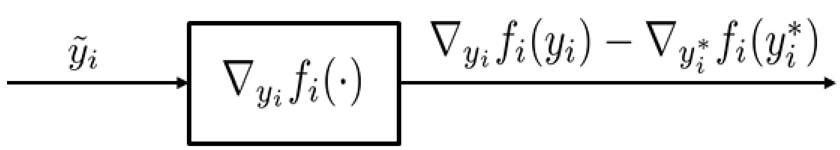


Figure 5.2: The overall update dynamics



output $\nabla_{y_i} f_i(y_i) - \nabla_{y_i^*} f_i(y_i^*)$ since the input-output relationship can be presented as

$$\tilde{y}_i^T [\nabla_{y_i} f_i(y_i) - \nabla_{y_i^*} f_i(y_i^*)]$$

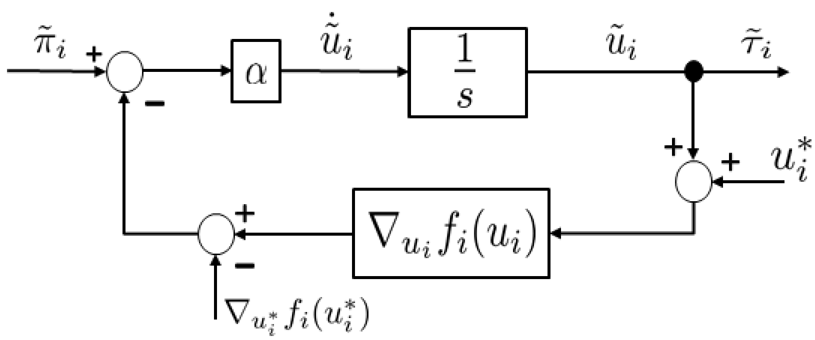
From the global under-estimator property of differentiable convex functions

$$\tilde{y}_i [\nabla_{y_i} f_i(y_i) - \nabla_{y_i^*} f_i(y_i^*)] \geq 0$$

which makes the system passive. Consider the following:

$$
\begin{aligned}
\dot{\tilde{u}}_i &= -\alpha_i[\nabla_{x_i} f_i(u_i) - \nabla_{u_i^*} f_i(u_i^*)] + \alpha\tilde{\pi}_i \\
\tilde{\tau}_i &= \tilde{u}_i
\end{aligned}
$$

The state space model of each agent can be drawn as below:



Consider the following Lyapunov function:

$$V_i = \frac{1}{2\alpha_i}||\tilde{u}_i||^2$$

Differentiating the Lyapunov candidate yields

$$\dot{V}_i = \frac{1}{\alpha_i}\tilde{u}_i^T\dot{\tilde{u}}_i = -\tilde{u}_i^T\nabla_{u_i} f_i(u_i) + \tilde{u}_i{}^T\nabla_{u_i^*} f_i(u_i^*) + \tilde{u}_i^T\tilde{\pi}_i$$

It follows from (5.18) that

$$\dot{V}_i = -\tilde{u}_i^T\nabla_{u_i} f_i(u_i) + \tilde{u}_i^T\nabla_{u_i^*} f_i(u_i^*) + \tilde{u}_i^T\tilde{\pi}_i \leq \tilde{\tau}_i^T\tilde{\pi}_i$$

Then, consider the following secondary system

$$
\begin{aligned}
\dot{\tilde{z}}_{ij} &= -\alpha_j \left[A_{ji}\tilde{\mu}_i + \tilde{\pi}_{ij}\right] \\
\dot{\tilde{\mu}}_j &= w_i \sum_{i \in \mathcal{N}_j} A_{ji}\tilde{z}_{ij} \\
\tilde{\tau}_{ij} &= \tilde{z}_{ij}
\end{aligned}
$$

The state space model

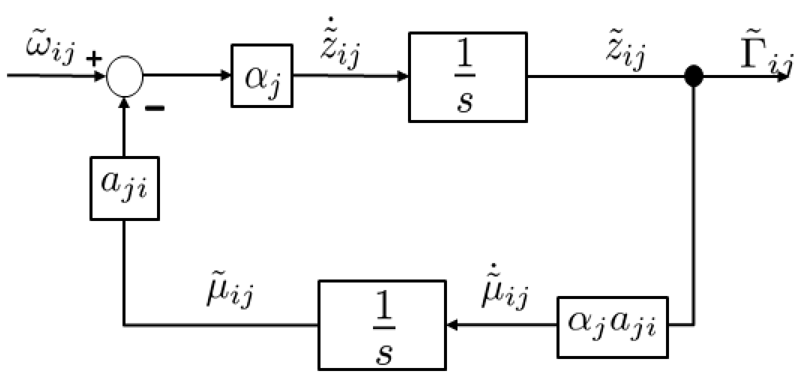
Consider the Lyapunov function

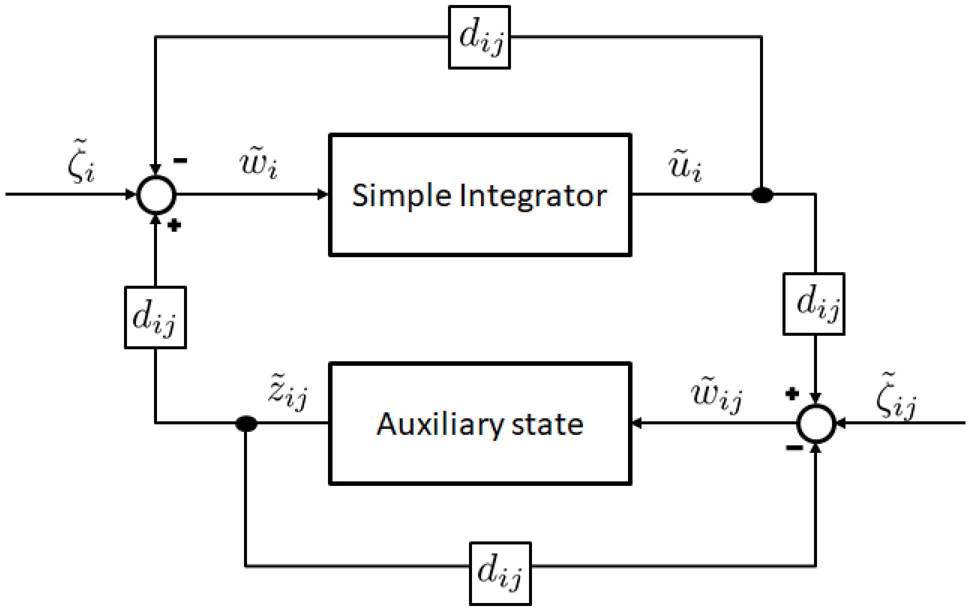
$$
V_{z_{ij}} = \frac{1}{2\alpha_j}||\tilde{z}_{ij}||^2 + \frac{1}{2w_j}||\tilde{\mu}_j||^2
$$

Taking time derivative

$$
\dot{V}_{z_{ij}} = \tilde{z}_{ij}^T\Big[\tilde{\pi}_{ij} - A_{ji}\tilde{\mu}_j\Big] + \tilde{\mu}_j^T\Big[A_{ji}\tilde{z}_{ij}\Big] = \tilde{\tau}_{ij}^T\tilde{\pi}_{ij}
$$

With the above two systems defined, we can interconnect them in the following ways





The control inputs are defined as

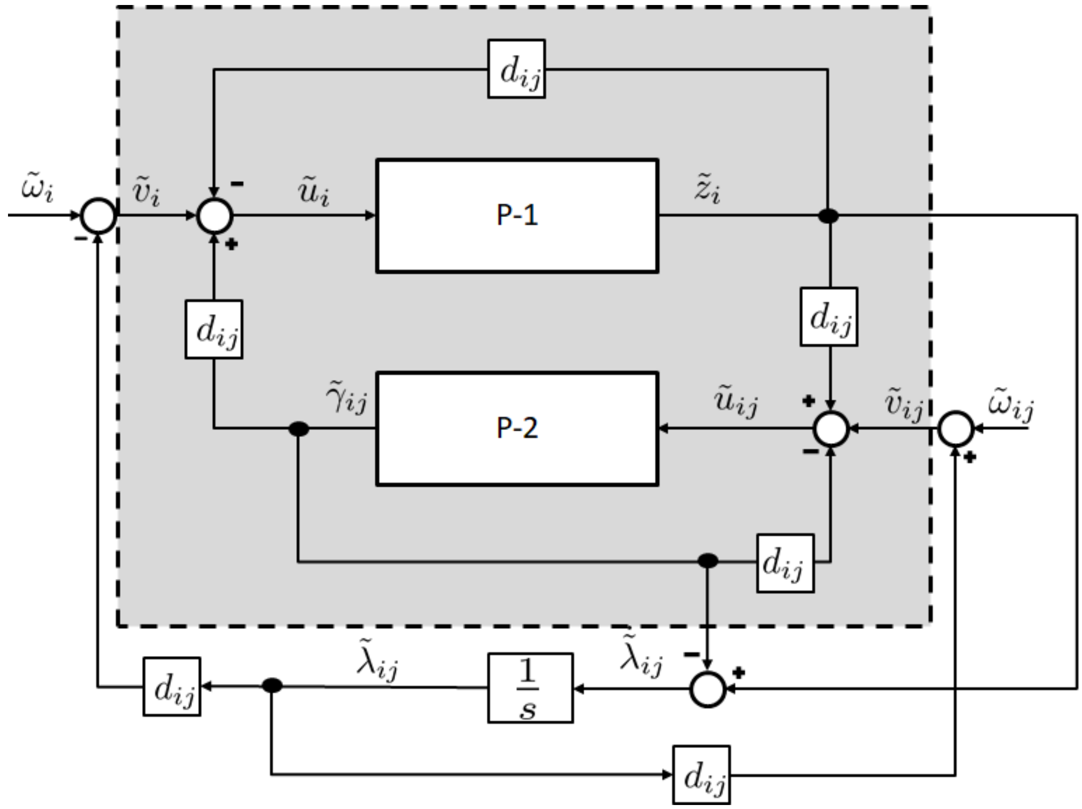
$$\tilde{\pi}_i = -d_{ij}(\tilde{u}_i - \tilde{z}_{ij}) + \tilde{\xi}_i \qquad \tilde{\pi}_{ij} = d_{ij}(\tilde{u}_i - \tilde{z}_{ij}) + \tilde{\xi}_{ij}$$

Defining $\tilde{\pi} = [\tilde{\pi}_i \; \tilde{\pi}_{ij}]$, $\tilde{\xi} = [\tilde{\xi}_i \; \tilde{\xi}_{ij}]$ and $\tilde{\Gamma} = [\tilde{u}_i \; \tilde{z}_{ij}]$, it follows that

$$\begin{aligned} \dot{V}_{c_i} = \dot{V}_{u_i} + \dot{V}_{z_{ij}} &\leq \tilde{\pi}^T \tilde{\Gamma} \\ &= \tilde{u}_i[-d_{ij}(\tilde{u}_i - \tilde{z}_{ij}) + \tilde{\xi}_i] + \tilde{z}_{ij}[d_{ij}(\tilde{u}_i - \tilde{z}_{ij}) + \tilde{\xi}_{ij}] \\ &= -d_{ij}||(\tilde{u}_i - \tilde{z}_{ij})||^2 + \tilde{u}_i^T \tilde{\xi}_i + \tilde{z}_{ij}^T \tilde{\xi}_{ij} \leq \tilde{\xi}^T \tilde{\Gamma} \end{aligned}$$

Finally we can complete the ADMM system by adding the Lagrange multiplier subsystem as follows: With the control laws

$$\tilde{\zeta}_i = -d_{ij}\tilde{\lambda}_{ij} + \tilde{\psi}_i \qquad \tilde{\zeta}_{ij} = d_{ij}\tilde{\lambda}_{ij}$$



Using the Lyapunov function:

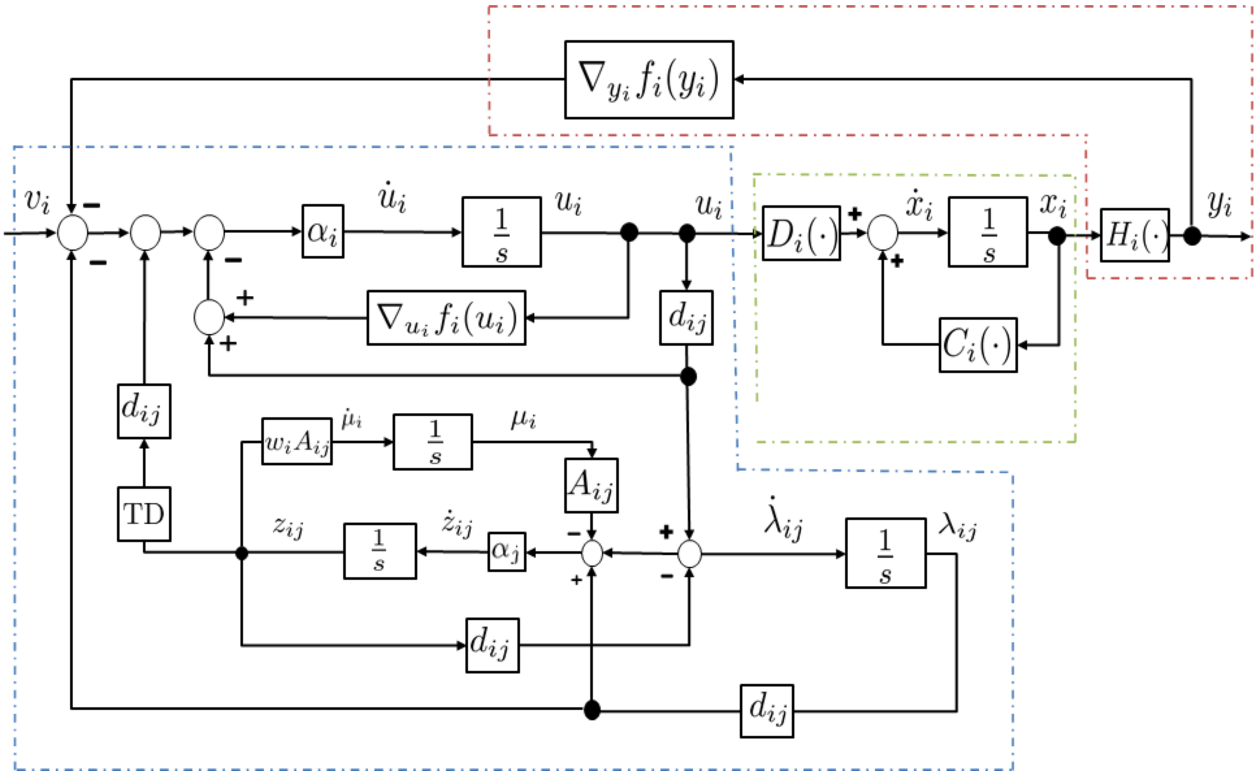
$$V_i = V_{u_i} + V_{z_{ij}} + \frac{1}{2}||\tilde{\lambda}||^2$$

Taking time derivative, it follows that:

$$\begin{aligned}
\dot{V}_i &\leq -d_{ij}||\tilde{u}_i - \tilde{z}_{ij}||^2 + \tilde{u}_i^T[\tilde{\psi}_i - d_{ij}\tilde{\lambda}_{ij}] + d_{ij}\tilde{z}_{ij}^T\tilde{\lambda}_{ij} + d_{ij}\tilde{\lambda}_{ij}^T[\tilde{u}_i - \tilde{z}_{ij}] \\
&= -d_{ij}||\tilde{u}_i - \tilde{z}_{ij}||^2 + \tilde{u}_i^T\tilde{\psi}_i
\end{aligned}$$

For visualization, the overall system is presented again below The system can be broken down into 3 smaller subsystems.

1. The blue box is the ADMM subsystem which is output strictly passive.

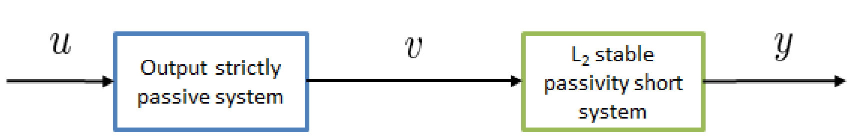


2. The red box is the gradient subsystem which is also passive.

3. The green box is the dynamic system which is assumed to be passivity-short and l_2 stable.

Now to prove the whole system to be passivity short, consider the following interconnected system which represents the passive ADMM subsystem in series connection with the dynamic subsystem. The time derivative of the Lyapunov functions for both the systems can be written as

$$\dot{V}_1 \leq u^T v - \gamma ||v||^2 \qquad \dot{V}_2 \leq v^T y + \epsilon ||v||^2 - \delta ||y||^2$$



Adding the functions

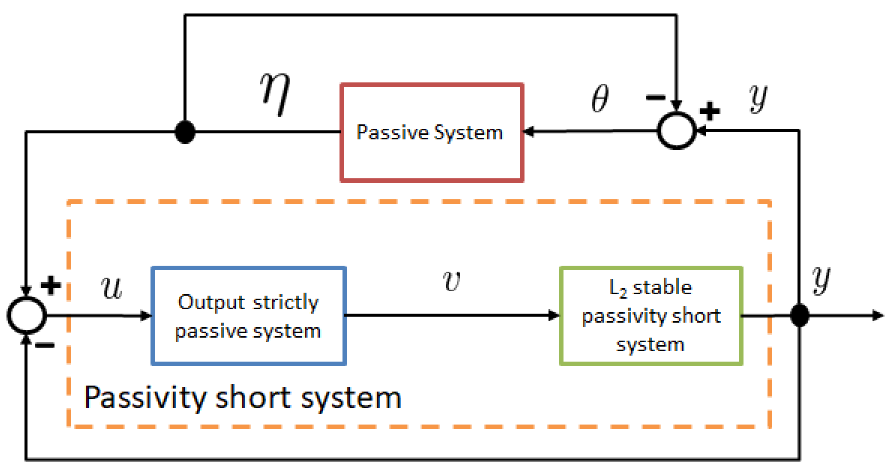
$$
\begin{aligned}
\dot{V}_1 + \dot{V}_2 &\leq u^T v - \gamma||v||^2 + v^T y + \epsilon||v||^2 - \delta||y||^2 \\
&\leq \frac{\gamma}{2}||u||^2 + \frac{1}{2\gamma}||v||^2 - \gamma||v||^2 + \frac{1}{2\epsilon}||v||^2 + \frac{\epsilon}{2}||y||^2 + \epsilon||v||^2 - \delta||y||^2
\end{aligned}
$$

Simplifying further, we get

$$
\begin{aligned}
\dot{V}_1 + \dot{V}_2 &\leq \frac{\gamma}{2}||u||^2 + \frac{\epsilon}{2}||y||^2 - \delta||y||^2 - \frac{2\gamma^2\epsilon - \gamma - \epsilon - 2\gamma\epsilon^2}{2\gamma\epsilon}||v||^2 \\
&\leq \frac{\gamma}{2}||u||^2 - \frac{2\delta - \epsilon}{2}||y||^2 - \frac{2\gamma^2\epsilon - \gamma - \epsilon - 2\gamma\epsilon^2}{2\gamma\epsilon}||v||^2 \\
&\leq u^T y + \frac{\gamma(2\delta - \epsilon) + 1}{2(2\delta - \epsilon)}||u||^2 - \frac{2\gamma^2\epsilon + \gamma + \epsilon + 2\gamma\epsilon^2}{2\gamma\epsilon}||v||^2
\end{aligned}
$$

which shows the system is passivity-short. Next, consider the following system The time derivative of the both the systems can be written as

$$
\dot{V}_3 \leq u^T y + \frac{\gamma(2\delta - \epsilon) + 1}{2(2\delta - \epsilon)}||u||^2 \qquad \dot{V}_4 \leq \theta^T \eta
$$



Define $u = \eta - y$ and $\theta = y - \eta$, we get

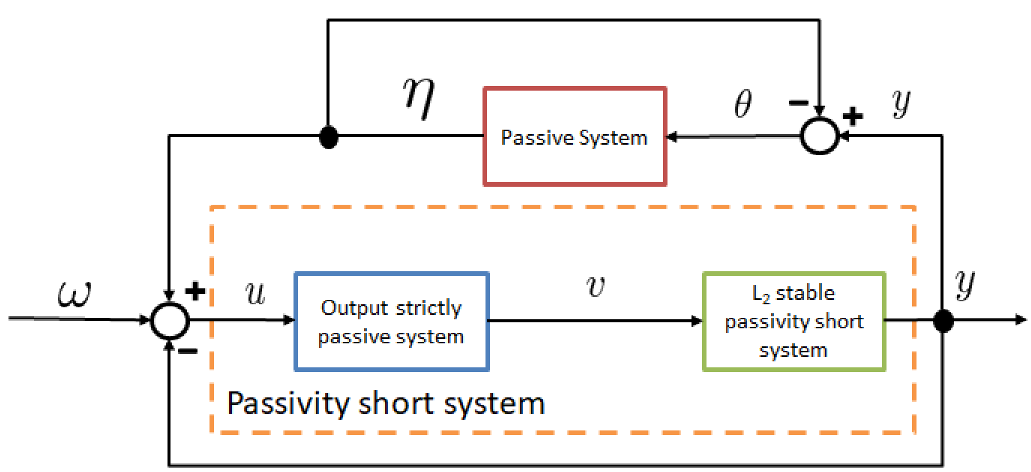
$$\begin{aligned}
\dot{V}_3 + \dot{V}_4 &= \leq (\eta - y)^T y + k(\eta - y)^T(\eta - y) + \eta^T(y - \eta) \\
&\leq \eta^T y - ||y||^2 + k||\eta||^2 - 2k\eta^T y + k||y||^2 + \eta^T y - ||\eta||^2 \\
&\leq 2(1-k)\eta^T y - (1-k)||y||^2 - (1-k)\eta^2 \\
&\leq -(1-k)||y - \eta||^2
\end{aligned}$$

where $k = \frac{\gamma(2\delta-\epsilon)+1}{2(2\delta-\epsilon)}$. Since $k \leq 1$, The derivative of the Lyapunov is negative semidefinite, which makes the system stable. Then, with the input,

Taking the time derivative and defining $u = (\eta - y) + \omega$ and $\theta = y - \eta$

$$\begin{aligned}
\dot{V}_3 + \dot{V}_4 &\leq u^T y + k||u||^2 + \theta^T \eta \\
&\leq \omega^T y + (\eta - y)^T y + k||(\eta - y) + \omega||^2 + (y - \eta)\eta \\
&\leq \omega^T y + k\omega^2 + k||\eta - y||^2 + 2k(\eta - y)^T \omega - (\eta - y)^2
\end{aligned}$$

which makes the system input passivity short and L2 stable.



In our proposed approach, we have divided the distribution system into three separate layers. On the top layer is the Distribution System Operator (DSO) who is in charge of the overall distribution grid. The DSO is responsible for maintaining optimal power flow in the grid as well as maintaining system parameters like voltage within the desired tolerable range. In the next layer, we assume the existence of a third party entity, generally termed as the aggregator in the literature, which collects EV information at each node and can also send/receive input/output signals. And in the final layer, we have all the consumers with the EVs who agree and signs a contract with the aggregators, allowing them to use the EVs for grid services in exchange for financial compensation. We assume at any instance of time, if the EV owners permit through the contract, aggregators can gather information on EV's active and reactive power injection at each bus through the existing sensor network between them in real-time. Figure 5.3 shows the structure of the multi-layer distribution grid and the associated variables which are defined in the next sub-section.

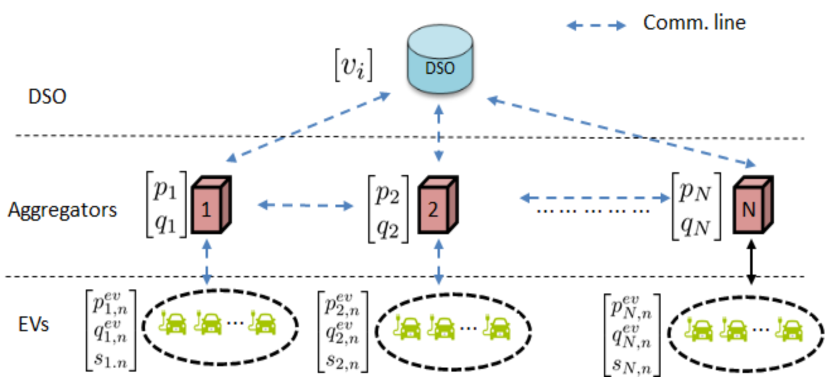


Figure 5.3: The multi-layer representation of distribution grid with associated variables

Branch Flow Model

For the branch flow model, consider a radial distribution network by a directed graph $G = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} := \{1, \cdots, N\}$ represents the set of buses and $\mathcal{E}$ represents the set of distribution lines connecting the buses in $\mathcal{N}$. Without any loss of generality, the substation of the radial network is indexed by 1. Each node $i \in \mathcal{N} \setminus 1$ has an unique parent node Γ_i and a set of children nodes, denoted by $\mathcal{C}_i$ as shown in figure 5.4. We assume each directed line points towards its children, i.e., power flows from parent Γ_i to node i.

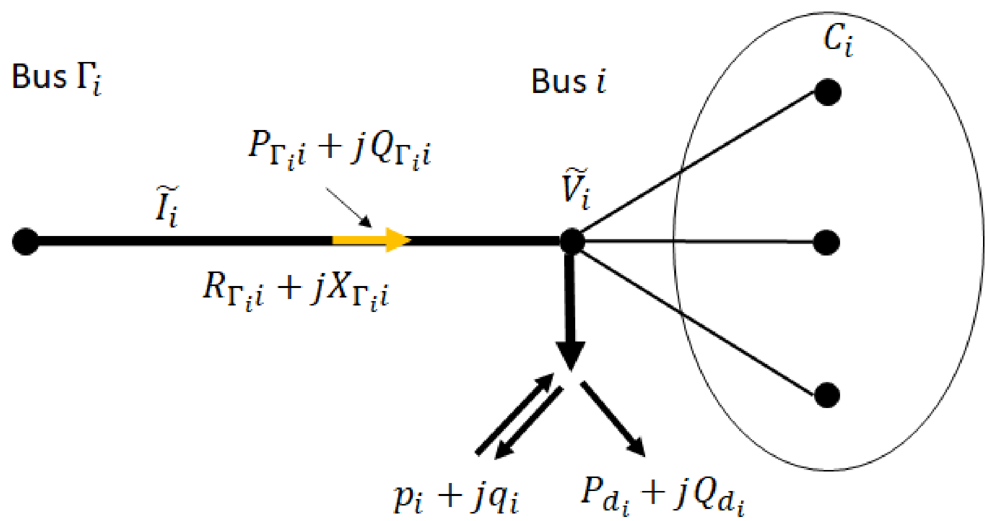


Figure 5.4: A distribution network.

For each bus $i \in \mathcal{N}$, let $\mathcal{N}_i$ bet the set of neighboring nodes including node i and its parent, i.e., $\mathcal{N}_i = \Gamma_i \cup \{i\} \cup \mathcal{C}_i$. Also, let V_i be its voltage with $P_{d_i} + jQ_{d_i}$ being the load demand and $p_i + jq_i$ is the aggregated active and reactive power injection by the EVs. For the branch $\Gamma_i \to i$, let I_i be the current flowing through it with $R_{\Gamma_i i} + jX_{\Gamma_i i}$ being the impedance of the line and $P_{\Gamma_i i} + jQ_{\Gamma_i i}$ being the complex power flowing from the parent Γ_i to node i. For bus $i \in \mathcal{N}$, the power balance equations are given as

$$P_{\Gamma_i i} = p_i + P_{d_i} + \sum_{j \in \mathcal{C}_i} (P_{ij} + R_{ij} l_{ij}) \qquad i \in \mathcal{N} \tag{5.20a}$$

$$Q_{\Gamma_i i} = q_i + Q_{d_i} + \sum_{j \in \mathcal{C}_i} (Q_{ij} + X_{ij} l_{ij}) \qquad i \in \mathcal{N} \tag{5.20b}$$

where $l_{ij} = I_i^2$ is the square of current magnitude which is defined as

$$l_{ij} = \frac{P_{ij}^2 + Q_{ij}^2}{v_i} \tag{5.21}$$

where $v_i = V_i^2$ is the square of the voltage magnitude. Equation (5.21) is a non-linear equation which can be linearized around current operating point $P_{ij}^o, Q_{ij}^o, v_i^o$ and l_{ij}^o as below:

$$l_{ij} - l_{ij}^o = \frac{2P_{ij}^o}{v_i^o}(P_{ij} - P_{ij}^o) + \frac{2Q_{ij}^o}{v_i^o}(Q_{ij} - Q_{ij}^o) - \frac{(P_{ij}^o)^2 + (Q_{ij}^o)^2}{(v_i^o)^2}(v_i - v_i^o) \tag{5.22}$$

The power flow on all lines $(i, j) \in \mathcal{E}$ are expressed as

$$v_i - v_j = 2(R_{ij} P_{ij} + X_{ij} Q_{ij}) + (R_{ij}^2 + X_{ij}^2) l_{ij} \tag{5.23}$$

EV Parameters

In our structure, EV owners, who are interested to sign up for grid services, are connected to aggregators at each node. Thus, The aggregated power of all EVs connected to node i at time t is

$$p_i(t) = \sum_{n=1}^{E_i} p_{i,n}^{ev}, \qquad \forall\, i \in \mathcal{N} \tag{5.24}$$

where E_i denotes the number of EVs controlled by the aggregator in charge of bus i and $p_{i,n}^{ev} \in \{0, \overline{p}_{i,n}^{ev}, \underline{p}_{i,n}^{ev}\}$ is the charging/discharging power of EV n. Let us denote the time of arrival of nth EV as $t_{a,n}$ and departure time as $t_{d,n}$. The EV owner registers this time with the aggregator so that they can be used in the continuous-domain real-time optimization framework. The nth EV only charge/discharge itself between this time-frame. Thus we can write:

$$p_{i,n}^{ev} = \begin{cases} 0 & t \notin [t_{a,n}, t_{d,n}] \\ \left[\overline{p}_{i,n}^{ev}, \underline{p}_{i,n}^{ev}\right] & t \in [t_{a,n}, t_{d,n}] \end{cases} \tag{5.25}$$

The state of charge dynamics $s_{i,n}$ of EV can be represented by the following first order differential equation:

$$\dot{s}_{i,n} = \frac{\eta_n p_{i,n}^{ev}}{B_{i,n}} \tag{5.26}$$

At the end of the charging period, the EV consumers wants their EVs to be charged to a specific state of charge, specifically

$$s_{i,n}(t_{d,n}) = s_{i,n}^{d}, \qquad n \in \mathcal{E}_i,\ i \in \mathcal{N} \tag{5.27}$$

where $s_{i,n}^d$ is the desired SOC.

Objective Functions

For the Distribution System Operator (DSO), the objective would be to minimize the generation cost and maintain the voltage close to unity at every bus over the time horizon, mathematically:

$$f_i(p_i, v_i) = C_i(p_i) + H_i(v_i) \qquad i \in \mathcal{N} \tag{5.28}$$

where $C_i(p_i(t))$ is the cost function for generation production and $H_i(v_i(t))$ is the voltage regulation penalty function.

As for the EV owners, the want to minimize their charging cost and maximize their utility. Thus we can express their welfare function as follows

$$W_{i,n}(p_{i,n}^{ev}) = \psi_{i,n} p_{i,n}^{ev} - U_i(p_{i,n}^{ev}) \qquad n \in \mathcal{E}_i,\ i \in \mathcal{N} \tag{5.29}$$

We also have a terminal condition, where at the end of a charging period, the SOC of the EV should be at the desired SOC, that is

$$S_{i,n}(s_{i,n}^d) = k_{i,n}(s_{i,n} - s_{i,n}^d)^2 \tag{5.30}$$

where $k_{i,n} > 0$ is the weight on terminal condition.

Let us stack all the state variables into a vector ω_i, that is

$$\omega_i = \left[v_i, p_i, q_i, P_{\Gamma_i i}, Q_{\Gamma_i i}, l_{\Gamma_i i}, p^{ev}_{i,n}, q^{ev}_{i,n}, s_{i,n}\right]^T \qquad \forall n = [1, \cdots, E_i],\ i \in \mathcal{N}$$

we also introduce an observation vector z_{ji} which represents the variables of node j at node i, that is $z_{ji} = [v^i_j, p^i_j, q^i_j, P^i_{\Gamma_j j}, Q^i_{\Gamma_j j}, l^i_{\Gamma_j j}, p^{ev,i}_{j,n}, q^{ev,i}_{j,n}, s^{ev,i}_{j,n}]^T$. With these definitions, we can formulate our optimization problem according to the developed dynamic ADMM as follows :

$$\min_{\omega_i} \quad \sum_{i=1}^{N} \sum_{n=1}^{E_i} \left[\phi_i(\omega_i)\right] \tag{5.31a}$$

$$\tag{5.31b}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{N}_i} A_{ij} z_{ji} + m_{ij} = 0 \qquad \forall i \in \mathcal{N} \tag{5.31c}$$

$$\dot{x}_i = A_i(x_i) + B_i(x_i) u_i \tag{5.31d}$$

$$y_i = z_{ij} \tag{5.31e}$$

$$y_i = \omega_i \tag{5.31f}$$

$$u_i = \mathcal{U}_i(x_i) + \omega_i \tag{5.31g}$$

where $\phi_i(\omega_i) = \left[S_{i,n}(s_{i,n}) + f_i(p_i, v_i) + W_{i,n}(p^{ev}_{i,n})\right]$, $x_i = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ s_i]^T$ and $\mathcal{U}_i(x_i) = 1$. It should be noted that the objective function is not summed over time since the problem is solved in real time with continuous-domain dynamics where the arrival and departure time can be tackled by the individual EV dynamics. Based on what agent j represents, the matrix A_{ij} and vector m_{ij} takes the following form:

$$A_{ii} = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \\ 1 & 0 & 0 & 2R_{\Gamma_i i} & 2X_{\Gamma_i i} & (R_{\Gamma_i i}^2 + X_{\Gamma_i i}^2) & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & \frac{2P_{\Gamma_i i}^o}{v_{\Gamma_i}^o} & \frac{2Q_{\Gamma_i i}^o}{v_{\Gamma_i}^o} & -1 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \end{bmatrix},$$

$$A_{ij} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & R_{ij} & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 0 & 1 & X_{ij} & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \end{bmatrix}, \quad j \in \mathcal{C}_i,$$

$$A_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \\ -\frac{(P_{ij}^o)^2 + (Q_{ij}^o)^2}{(v_{\Gamma_i}^o)^2} & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \end{bmatrix}, \quad j = \Gamma_i.$$

and $m_{ii} = [P_{d_i} \; Q_{d_i} \; 0 \; 0]^T$, $m_{ij} = \{[0\;0\;0\;0]^T, j \in \mathcal{C}_i\}$ and
$m_{ij} = \left\{ \left[0\;0\;0 \left(-\frac{2(P_{ji}^o)^2 v_j^o + 2(Q_{ji}^o)^2 v_j^o + (P_{ji}^o)^2 - (Q_{ji}^o)^2}{(v_j^o)^2} - l_{ji}^o \right) \right]^T, j \in \Gamma_i \right\}.$

Following the procedures developed in section 2 and dynamic equation set (5.13), we obtain the following update dynamics for the system

$$\dot{v}_i = -\alpha_i \left[\nabla_{v_i} \phi_i(\omega_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\lambda_{ij}(1) + (v_i - v_i^{j-}) \right] \right] \tag{5.32a}$$

$$\dot{p}_i = -\alpha_i \left[\nabla_{p_i} \phi_i(\omega_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\lambda_{ij}(2) + (p_i - p_i^{j-}) \right] \right] \tag{5.32b}$$

$$\dot{q}_i = -\alpha_i \left[\nabla_{q_i} \phi_i(\omega_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\lambda_{ij}(3) + (q_i - q_i^{j-}) \right] \right] \tag{5.32c}$$

$$\dot{P}_{\Gamma_i i} = -\alpha_i \left[\nabla_{P_{\Gamma_i i}} \phi_i(\omega_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\lambda_{ij}(4) + (P_{\Gamma_i i} - P_{\Gamma_i i}^{j-}) \right] \right] \tag{5.32d}$$

$$\dot{Q}_{\Gamma_i i} = -\alpha_i \left[\nabla_{Q_{\Gamma_i i}} \phi_i(\omega_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\lambda_{ij}(5) + (Q_{\Gamma_i i} - Q_{\Gamma_i i}^{j-}) \right] \right] \tag{5.32e}$$

$$\dot{l}_{\Gamma_i i} = -\alpha_i \left[\nabla_{l_{\Gamma_i i}} \phi_i(\omega_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\lambda_{ij}(6) + (l_{\Gamma_i i} - l_{\Gamma_i i}^{j-}) \right] \right] \tag{5.32f}$$

$$\dot{p}_{i,n}^{ev} = -\alpha_i \left[\nabla_{p_{i,n}^{ev}} \phi_i(\omega_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\lambda_{in}(7) + (p_{i,n}^{ev} - p_{i,n}^{ev,j-}) \right] \right] \tag{5.32g}$$

$$\dot{q}_{i,n}^{ev} = -\alpha_i \left[\nabla_{q_{i,n}^{ev}} \phi_i(\omega_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\lambda_{in}(8) + (q_{i,n}^{ev} - q_{i,n}^{ev,j-}) \right] \right] \tag{5.32h}$$

$$\dot{s}_{i,n} = -\alpha_i \left[\nabla_{s_{i,n}} \phi_i(\omega_i) + \sum_{j \in \mathcal{N}_i} d_{ij} \left[\lambda_{in}(9) + (s_{i,n} - s_{i,n}^{j-}) \right] \right] \tag{5.32i}$$

where $n \in [1, \cdots, E_i]$, $j \in \mathcal{N}_i$ and $i \in \mathcal{N}$. The z_{ji} is given as

$$\begin{bmatrix} \dot{v}_j^i \\ \dot{p}_j^i \\ \dot{q}_j^i \\ \dot{P}_{\Gamma_j j}^i \\ \dot{Q}_{\Gamma_j j}^i \\ \dot{l}_{\Gamma_j j}^i \\ \dot{p}_{j,n}^{ev,i} \\ \dot{q}_{j,n}^{ev,i} \\ \dot{s}_{j,n}^i \end{bmatrix} = \alpha_i d_{ji} \begin{bmatrix} \lambda_{ji}(1) \\ \lambda_{ji}(2) \\ \lambda_{ji}(3) \\ \lambda_{ji}(4) \\ \lambda_{ji}(5) \\ \lambda_{ji}(6) \\ \lambda_{jn}(7) \\ \lambda_{jn}(8) \\ \lambda_{jn}(9) \end{bmatrix} + d_{ji} \begin{bmatrix} v_j - v_j^i \\ p_j - p_j^i \\ q_j - q_j^i \\ P_{\Gamma_j j} - P_{\Gamma_j j}^i \\ Q_{\Gamma_j j} - Q_{\Gamma_j j}^i \\ l_{\Gamma_j j} - l_{\Gamma_j j}^i \\ p_{j,n}^{ev} - p_{j,n}^{ev,i} \\ q_{j,n}^{ev} - q_{j,n}^{ev,i} \\ s_{j,n} - s_{j,n}^i \end{bmatrix} - A_{ij}^T \begin{bmatrix} \mu_i(1) \\ \mu_i(2) \\ \mu_i(3) \\ \mu_i(4) \\ \mu_i(5) \\ \mu_i(6) \\ \mu_i(7) \\ \mu_i(8) \\ \mu_i(9) \end{bmatrix}$$

The dual updates are given as

$$\begin{bmatrix} \dot{\mu}_i(1) \\ \dot{\mu}_i(2) \\ \dot{\mu}_i(3) \\ \dot{\mu}_i(4) \\ \dot{\mu}_i(5) \\ \dot{\mu}_i(6) \\ \dot{\mu}_i(7) \\ \dot{\mu}_i(8) \\ \dot{\mu}_i(9) \end{bmatrix} = \sum_{j \in \mathcal{N}_i} A_{ij} \begin{bmatrix} v_j^i \\ p_j^i \\ q_j^i \\ P_{\Gamma_j j}^i \\ Q_{\Gamma_j j}^i \\ l_{\Gamma_j j}^i \\ p_{i,n}^{ev,j} \\ q_{i,n}^{ev,j} \\ s_{i,n}^j \end{bmatrix}$$

$$
\begin{bmatrix} \dot{\lambda}_{ji}(1) \\ \dot{\lambda}_{ji}(2) \\ \dot{\lambda}_{ji}(3) \\ \dot{\lambda}_{ji}(4) \\ \dot{\lambda}_{ji}(5) \\ \dot{\lambda}_{ji}(6) \\ \dot{\lambda}_{jn}(7) \\ \dot{\lambda}_{jn}(8) \\ \dot{\lambda}_{jn}(9) \end{bmatrix} = d_{ji} \begin{bmatrix} v_j - v_j^i \\ p_j - p_j^i \\ q_j - q_j^i \\ P_{\Gamma_j j} - P^i_{\Gamma_j j} \\ Q_{\Gamma_j j} - Q^i_{\Gamma_j j} \\ l_{\Gamma_j j} - l^i_{\Gamma_j j} \\ p^{ev}_{j,n} - p^{ev,i}_{j,n} \\ q^{ev}_{j,n} - q^{ev,i}_{j,n} \\ s_{j,n} - s^i_{j,n} \end{bmatrix}
$$

These dynamics were implemented on the IEEE 123 bus distribution system in a co-simulation with Matlab and OpenDSS.

Simulation Results

In the simulation of the proposed algorithm on the distribution system, we defined $C_i(p_i) = a_i^p p_i^2$, $H_i(v_i) = (1 - v_i)^2$ and $U_i(p^{ev}_{i,n}) = a_{p_n} log(p^{ev}_{i,n} + 1)$ and the parameters of the coefficients were taken from [58]. The price of electricity was fixed at an average of 15 cents per kWh. We randomly placed 150 EVs among 40 buses in the system and we fixed that by the end of their charging period, the EV owners want their vehicles to be charged to at least 90% of its capacity. The value of step-size α_i was fixed at 0.01 for all the nodes. We first ran a base case where the algorithm was not implemented and figured that bus 111,113 and 114 had the lowest voltage in the whole grid. So we observed the voltages of those 3 nodes after implementing the algorithm and the results are shown

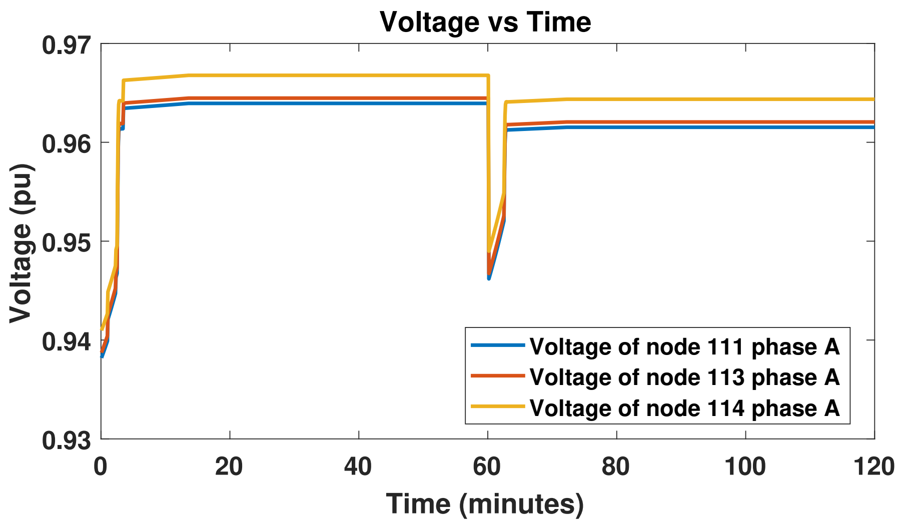


Figure 5.5: Voltage at Node 111,113 and 114

in figure 5.5. Figure 5.6 shows the total amount of active power injection by the EVs over 5 hours responding to the voltage fluctuations in the grid. We selected 3 random EVs around node the three selected nodes and plotted their active power injection and how their SOC evolved which are presented in figure 5.7 and 5.8, respectively. The figures shows the arrival and departure time of the EVs and we can observe that the EVs react to the voltage fluctuations in the grid and eventually at the end of their charging time, they get their SOC above 90%.

This chapter presents a novel continuous-domain distributed multi-agent ADMM algorithm with dynamic constraint. In contrast to usual discrete-time iterative solution techniques where the accuracy and the optimality of the solution depend on sampling and convergence time, the proposed algorithm can solve the problem in real-time. It is also capable of handling dynamic constraints that are pretty prevalent in distribution power systems with intermittent energy resources like EVs present. It is proven to converge which is shown using the Lyapunov direct method and the analytical results were tested on IEEE 123 bus test system. The results obtained are also included for the illustration of the proposed algorithm.

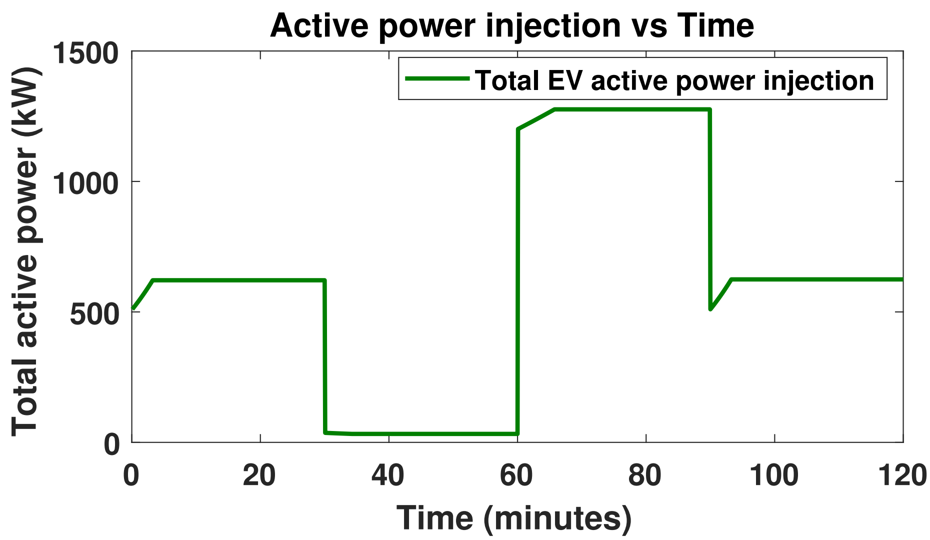


Figure 5.6: Total active power injection by EVs into the grid

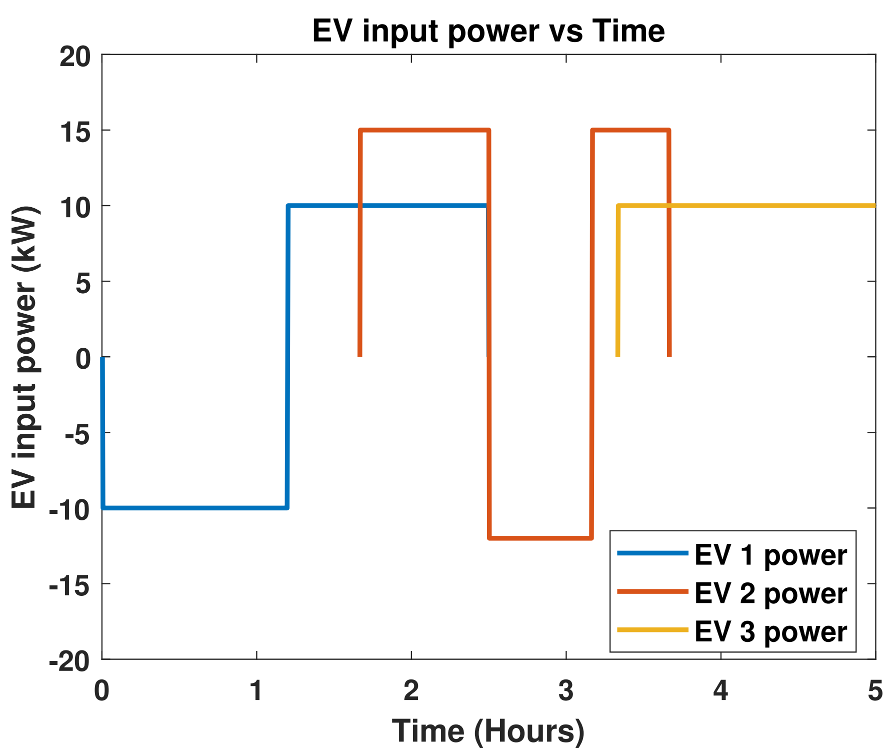


Figure 5.7: Active power injection by 3 EVs

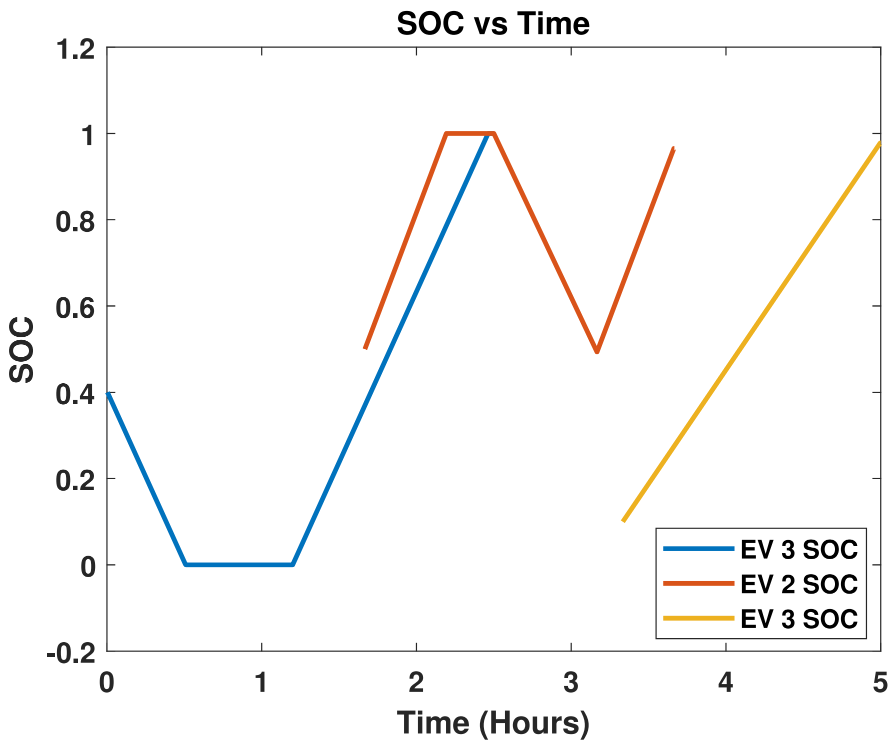


Figure 5.8: The SOC evolution of the 3 EVs

CHAPTER 6: CONCLUSION

With the advent of technologies like 5G and IoT, the number of connected devices is rising exponentially while the network is getting more sophisticated. As a result, the data availability and its interchange in the network are escalating in both size and complexity. Centralized management of the network has failed since it is not efficient to gather all the information for single-point communication. The recent research is moving towards distributed optimization because of its flexibility and scalability and does not require any centralized coordination. Also, for modern interconnected systems, convergence time and sampling speed of the data matter a lot since an optimal solution based on a set of data is no longer optimal after few moments because of the rapid changing of the system parameters. Keeping these in mind, this dissertation focuses mainly on developing tangible distributed algorithms that can be implemented in real-time. In this dissertation, a network of interconnected devices is considered where each device acts as an agent. Each agent has its local objective function to satisfy by exchanging minimum information between its neighbors to satisfy the global objective. While doing so, they need to also satisfy a network-level constraint which is defined by the physical interconnection between them. This is a distributed optimization setting with network-level constraints and it was converted into ADMM by introducing the second primal variable which is the observation of the neighboring agent's state at each agent. Several assumptions were made on the convexity of the local objective functions as well as the strong connectivity of the network. Utilizing these assumptions, a closed-form iterative dynamics in discrete time is developed for ADMM optimization sub-problems. Each agent optimizes its local objective functions based on the information it receives from its neighbors, this information can be utilized to design an algorithm for faster convergence. In contrast to the standard ADMM which uses a fixed penalty gain in the augmented Lagrangian, the proposed algorithm embeds control gains into a row-stochastic matrix based on network connectivity, utilizes the matrix coefficients as the penalty parameters in ADMM, and uses information received by each agent from its neighbors to adaptively adjust these penalties. One drawback of this algorithm was that it was solved in discrete time. With a modern networked system that consists of agents that are intermittent and whose parameters change very rapidly, an optimal solution a few moments ago does not remain optimal anymore. So, because of this, the algorithm needs to handle the problem in real-time. For this reason, the algorithm was further developed into a continuous domain with real-time applicability. To illustrate its effectiveness, it was implemented in a smart grid setting. A distribution system was considered with distributed energy resources as well as electric vehicles (EVs). The objective of the problem was to maintain voltage close to unity at each node by using electric vehicles. EVs would establish a contract with a third party, known as aggregators, who will give them input signal to charge or discharge based on the energy need of the grid to maintain the voltage. A convex optimization problem with power flow equations as constraints were set up where each aggregator tries to minimize the EV charging cost while contributing to ancillary services while DSO would maintain the voltage close to unity and solve for the optimal power flow. The developed problem with DSO and aggregators was cast into a distributed ADMM framework and is solved using the developed continuous-domain real-time algorithm by communicating relevant information among them. In contrast to usual discrete-time iterative solution techniques where the accuracy and optimality of the solution depend on the sampling and convergence time, the proposed continuous-domain algorithm can solve the optimization and control problems in real-time. In this problem, we solved the problem concerning the DSO and aggregators and left the EV parts out. EVs have their objectives to fulfill like maintaining a certain level of state of charge. Tackling this problem would be difficult since the state of charge is dynamic. For this reason, the algorithm was further developed to accommodate dynamic equations. It was shown that the algorithm can be shown to be the interconnection of several subsystems, which when added together, can be shown to be passivity-short. It was again implemented in the same smart grid scenario, except this time the EVs work as agents of the system as well. The objective for the DSO was to use EVs, through aggregators, for voltage regulation in the grid in exchange for financial compensation. For EVs, the objective was to maximize their gain but in the meantime, also make sure that when they leave, they leave with the state of charge of their battery above a certain threshold. The setup was simulated and the results illustrate the effectiveness of the algorithms developed. For future work, since the algorithm is capable of handling dynamic constraints, the next step would be to solve the problem where the objective function has temporal dependence.

APPENDIX A: LIST OF PUBLICATIONS

Journal

T. Rahman, Z. Qu and T. Namerikawa, "Improving Rate of Convergence via Gain Adaptation in Multi-Agent Distributed ADMM Framework," in *IEEE Access*, vol. 8, pp. 80480-80489, 2020.

T. Rahman, Y. Xu and Z. Qu, "Continuous-Domain Real-Time ADMM Algorithm for Distributed EV Charging and Voltage Stability in Distribution Network" submitted, *IEEE Transaction on Automatic Science and Engineering*.

Conference

T. Rahman and Z. Qu,"The role of electric vehicles for frequency regulation during grid restoration," *2017 IEEE Power & Energy Society General Meeting*, IEEE 2017, pp. 1-5.

T. Rahman, R. Harvey, Z. Qu and M. A. Simaan, "A distributed cooperative load control approach for ancillary services in smart grid," *2017 American Control Conference (ACC)*, IEEE, 2017, pp. 1401-1406.

Book Chapter

T. Rahman and Z. Qu, "Autonomous Intelligent Charging/Discharging of Electric Vehicles using Distributed Multi-Agent ADMM Framework for Grid Ancillary Services," *Intelligent Control and Smart Energy Management: Renewable Resources and Transportation*, submitted, Springer Optimization and Its Applications (SOIA).

LIST OF REFERENCES

[1] 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, pp. 1–122, 2011.

[2] R. Glowinski and A. Marroco, “Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de dirichlet non linéaires,” *Revue Française d’Automatique, Informatique, Recherche Opérationnelle. Analyse Numérique*, vol. 9, pp. 41–76, 1975.

[3] D. Gabay and B. Mercier, “A dual algorithm for the solution of nonlinear variational problems via finite element approximation,” *Computers & Mathematics with Applications*, vol. 2, pp. 17–40, 1976.

[4] M. A. Figueiredo and J. M. Bioucas-Dias, “Restoration of poissonian images using alternating direction optimization,” *IEEE Transactions on Image Processing*, vol. 19, pp. 3133–3145, 2010.

[5] P. A. Forero, A. Cano, and G. B. Giannakis, “Consensus-based distributed support vector machines,” *Journal of Machine Learning Research*, vol. 11, pp. 1663–1707, 2010.

[6] S. Joshi, M. Codreanu, and M. Latva-aho, “Distributed SINR balancing for MISO downlink systems via the Alternating Direction Method of Multipliers,” in *Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks*, 2013, pp. 318–325.

[7] Q. Peng and S. H. Low, “Distributed optimal power flow algorithm for radial networks 1: Balanced single phase case,” *IEEE Transactions on Smart Grid*, vol. 9, pp. 111–121, 2018.

[8] W. Deng and W. Yin, “On the global and linear convergence of the generalized Alternating Direction Method of Multipliers,” *Journal of Scientific Computing*, vol. 66, pp. 889–916, 2016.

[9] P. Giselsson and S. Boyd, “Linear convergence and metric selection for Douglas-Rachford splitting and ADMM,” *IEEE Transactions on Automatic Control*, vol. 62, pp. 532–544, 2017.

[10] B. He and X. Yuan, "On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers," *Numerische Mathematik*, vol. 130, pp. 567–577, 2015.

[11] M. Kadkhodaie, K. Christakopoulou, M. Sanjabi, and A. Banerjee, "Accelerated alternating direction method of multipliers," in *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 497–506.

[12] B. He, H. Yang, and S. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *Journal of Optimization Theory and Applications*, vol. 106, pp. 337–356, 2000.

[13] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, "A general analysis of the convergence of ADMM," *arXiv preprint arXiv:1502.02009*, 2015.

[14] T. Rahman, Z. Qu, and T. Namerikawa, "Improving rate of convergence via gain adaptation in multi-agent distributed ADMM framework," *IEEE Access*, 2020.

[15] Z. Xu, M. A. Figueiredo, and T. Goldstein, "Adaptive ADMM with spectral penalty parameter selection," *arXiv:1605.07246*, 2016.

[16] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall Englewood Cliffs, NJ, 1989.

[17] D. Boley, "Local linear convergence of the Alternating Direction Method of Multipliers on quadratic or linear programs," *SIAM Journal on Optimization*, vol. 23, pp. 2183–2207, 2013.

[18] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems," *IEEE Transactions on Automatic Control*, vol. 60, pp. 644–658, 2015.

[19] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[20] 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.

[21] Z. Qu, C. Li, and F. Lewis, "Cooperative control with distributed gain adaptation and connectivity estimation for directed networks," *International Journal of Robust and Nonlinear Control*, vol. 24, pp. 450–476, 2014.

[22] C. Song, S. Yoon, and V. Pavlovic, "Fast ADMM algorithm for distributed optimization with adaptive penalty." in *Association for the Advancement of Artificial Intelligence*, 2016, pp. 753–759.

[23] Z. Xu, G. Taylor, H. Li, M. Figueiredo, X. Yuan, and T. Goldstein, "Adaptive consensus ADMM for distributed optimization," *Proceedings of the 34th International Conference on Machine Learning, vol.70, JMLR.org*, 2017.

[24] Z. Qu, *Cooperative Control of Dynamical Systems*. Springer, 2009.

[25] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "Dlm: Decentralized linearized alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 4051–4064, 2015.

[26] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[28] Z. Qu, "Robust state observer and control design using command-to-state mapping," *Automatica*, vol. 41, no. 8, pp. 1323–1333, 2005.

[29] L. Cai, J. Pan, L. Zhao, and X. Shen, "Networked electric vehicles for green intelligent transportation," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 77–83, 2017.

[30] I. Wagner, *The U.S. Electric Vehicle Industry - Statistics and Facts*, 2020 (April 30). [Online]. Available: https://www.statista.com/topics/4421/the-us-electric-vehicle-industry/

[31] L. Chen, T. Yu, Y. Chen, W. Guan, Y. Shi, and Z. Pan, "Real-time optimal scheduling of large-scale electric vehicles: A dynamic non-cooperative game approach," *IEEE Access*, vol. 8, pp. 133 633–133 644, 2020.

[32] M. Li, J. Gao, N. Chen, L. Zhao, and X. Shen, "Decentralized pev power allocation with power distribution and transportation constraints," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 1, pp. 229–243, 2019.

[33] H. Liang, Y. Liu, F. Li, and Y. Shen, "Dynamic economic/emission dispatch including pevs for peak shaving and valley filling," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 2880–2890, 2018.

[34] B. Khaki, C. Chu, and R. Gadh, "A hierarchical admm based framework for ev charging scheduling," in *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*. IEEE, 2018, pp. 1–9.

[35] P. Richardson, D. Flynn, and A. Keane, "Optimal charging of electric vehicles in low-voltage distribution systems," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 268–279, 2011.

[36] Y. Cao, S. Tang, C. Li, P. Zhang, Y. Tan, Z. Zhang, and J. Li, "An optimized EV charging model considering tou price and soc curve," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 388–393, 2011.

[37] K. Clement-Nyns, E. Haesen, and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid," *IEEE Transactions on power systems*, vol. 25, no. 1, pp. 371–380, 2009.

[38] S. Deilami, A. S. Masoum, P. S. Moses, and M. A. Masoum, "Real-time coordination of plug-in electric vehicle charging in smart grids to minimize power losses and improve voltage profile," *IEEE Transactions on Smart Grid*, vol. 2, no. 3, pp. 456–467, 2011.

[39] S. Shao, M. Pipattanasomporn, and S. Rahman, "Challenges of PHEV penetration to the residential distribution network," in *2009 IEEE Power & Energy Society General Meeting*. IEEE, 2009, pp. 1–8.

[40] A. Asrari, M. Ansari, J. Khazaei, and P. Fajri, "A market framework for decentralized congestion management in smart distribution grids considering collaboration among electric vehicle aggregators," *IEEE Transactions on Smart Grid*, 2019.

[41] Z. Pan, T. Yu, J. Li, K. Qu, L. Chen, B. Yang, and W. Guo, "Stochastic transactive control for electric vehicle aggregators coordination: A decentralized approximate dynamic programming approach," *IEEE Transactions on Smart Grid*, 2020.

[42] S. Gao and H. Jia, "Integrated configuration and optimization of electric vehicle aggregators for charging facilities in power networks with renewables," *IEEE Access*, vol. 7, pp. 84 690–84 700, 2019.

[43] N. Chen, C. W. Tan, and T. Q. Quek, "Electric vehicle charging in smart grid: Optimality and valley-filling algorithms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1073–1083, 2014.

[44] M. Liu, P. K. Phanivong, Y. Shi, and D. S. Callaway, "Decentralized charging control of electric vehicles in residential distribution networks," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 266–281, 2017.

[45] H. K. Nunna, S. Battula, S. Doolla, and D. Srinivasan, "Energy management in smart distribution systems with vehicle-to-grid integrated microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4004–4016, 2018.

[46] K. Chaudhari, N. K. Kandasamy, A. Krishnan, A. Ukil, and H. B. Gooi, "Agent-based aggregated behavior modeling for electric vehicle charging load," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 856–868, 2018.

[47] B. Wang, P. Dehghanian, and D. Zhao, "Chance-constrained energy management system for power grids with high proliferation of renewables and electric vehicles," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2324–2336, 2019.

[48] H. Fan, C. Duan, C.-K. Zhang, L. Jiang, C. Mao, and D. Wang, "Admm-based multiperiod optimal power flow considering plug-in electric vehicles charging," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 3886–3897, 2017.

[49] X. Zhou, S. Zou, P. Wang, and Z. Ma, "Voltage regulation in constrained distribution networks by coordinating electric vehicle charging based on hierarchical admm," *IET Generation, Transmission & Distribution*, vol. 14, no. 17, pp. 3444–3457, 2020.

[50] M. Sengupta and A. Andreas, "Oahu solar measurement grid (1-year archive): 1-second solar irradiance; oahu, hawaii (data)," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2010.

[51] K. Kaur, N. Kumar, and M. Singh, "Coordinated power control of electric vehicles for grid frequency support: Milp-based hierarchical control design," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3364–3373, 2018.

[52] H. Liu, Y. Zhang, S. Ge, C. Gu, and F. Li, "Day-ahead scheduling for an electric vehicle pv-based battery swapping station considering the dual uncertainties," *IEEE Access*, vol. 7, pp. 115 625–115 636, 2019.

[53] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Transactions on power Delivery*, vol. 4, no. 1, pp. 725–734, 1989.

[54] N. Mehboob, M. Restrepo, C. A. Canizares, C. Rosenberg, and M. Kazerani, "Smart operation of electric vehicles with four-quadrant chargers considering uncertainties," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2999–3009, 2018.

[55] C. Ai, G. Zhou, W. Gao, J. Guo, G. Jie, Z. Han, and X. Kong, "Research on quasi-synchronous grid-connected control of hydraulic wind turbine," *IEEE Access*, vol. 8, pp. 126 092–126 108, 2020.

[56] M. Nick, R. Cherkaoui, J.-Y. Le Boudec, and M. Paolone, "An exact convex formulation of the optimal power flow in radial distribution networks including transverse components," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 682–697, 2017.

[57] H. Wang, D. Zhao, Q. Meng, G. P. Ong, and D.-H. Lee, "A four-step method for electric-vehicle charging facility deployment in a dense city: An empirical study in singapore," *Transportation Research Part A: Policy and Practice*, vol. 119, pp. 224–237, 2019.

[58] Y. Okawa, T. Namerikawa, and Z. Qu, "Passivity-based stability analysis of dynamic electricity pricing with power flow," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 813–818.

[59] R. Harvey and Z. Qu, "Cooperative control and networked operation of passivity-short systems," in *Control of Complex Systems*. Elsevier, 2016, pp. 499–518.