On the Control of Electric Vehicle Charging in the Smart Grid

by

Abdullah Al Zishan

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Over the last decade, the demand for electric vehicles (EVs) has surged across the globe. This spurred an increase in the installation of public and private EV charging points which are typically connected to low-voltage power distribution feeders. A high penetration of plug-in EVs in distribution networks is anticipated to cause several problems, such as transformer overloading, voltage limits violations, and increased heat losses. Hence, a demand-side management strategy is needed to control the real power drawn by the charging points. These control strategies can be classified as model-based and model-free depending on whether they rely on a model of the distribution network (i.e., the admittance matrix). This thesis aims to investigate how to control the charge power of EVs from user-centric and grid centric perspectives. We design and evaluate two control frameworks that are suitable for the smart grid.

Given an approximate model of the distribution grid, we first propose a reputation-based framework for allocating power to EVs in the smart grid. In this framework, the available capacity of the distribution network measured by distribution-level phasor measurement units is divided in a proportionally fair manner among connected EVs, considering their demands and self-declared deadlines. To encourage users to estimate their deadlines more precisely and conservatively, a weight is assigned to each deadline based on the user's reputation, which comprises two kinds of evidence: the deadlines declared before and after the actual departure times in the recent past. We design a decentralized algorithm that achieves quadratic convergence under specific conditions and evaluate it empirically on a test distribution network by comparing it with state of the art algorithms.

In the second framework, we relax the assumption of having a model and propose a model-free, adaptive additive-increase multiplicative-decrease (AIMD)-like algorithm for controlled charging of EVs. This control algorithm is decentralized and merely relies on congestion signals generated by sensors deployed across the network. To dynamically adjust the parameter of this congestion control algorithm, we cast the problem as multi-agent reinforcement learning where each charging point is an independent agent which learns this parameter using an off-policy actor-critic deep reinforcement learning algorithm. Simulation results of both control algorithms on a test distribution network and on a parking station corroborate that the proposed algorithms track the available capacity of the network in real-time, prevents transformer overloading and voltage limits violation problems for an extended period of time, and outperforms several other decentralized feedback control algorithms proposed in the literature.

Preface

This thesis is original work by Abdullah Al Zishan. Some of the chapters are based on conference publications which are co-authored by the author of this thesis.

Specifically, Chapter 4 and Chapter 5 were published as a conference paper: Abdullah Al Zishan, Moosa Moghimi Haji, Omid Ardakanian, "Reputation-based Fair Power Allocation to Plug-in Electric Vehicles in the Smart Grid", in Proceedings of the 11th ACM/IEEE Conference on Cyber-Physical Systems (ICCPS'20), pp.63–74, 2020. As the first author, I was responsible for reviewing the literature, developing the methodology, analyzing the theoretical aspects of the methodology and conducting simulation. The second author, Moosa Moghimi Haji helped to build the power flow simulator and described the system model and simulation environment.

Parts of Chapter 6 and 7 appeared in the conference paper: Abdullah Al Zishan, Moosa Moghimi Haji, Omid Ardakanian, "Adaptive Control of Plug-in Electric Vehicle Charging with Reinforcement Learning", in the proceedings of the 11th ACM International Conference on Future Energy Systems (e-Energy), pp.116-120, 2020. The remaining part has been submitted to *IEEE Transactions on Smart Grid* and is currently under review (with the same authors as the e-Energy paper). I was responsible for reviewing the literature partially, developing the methodology, running simulation and producing results. Moosa Moghimi Haji wrote the literature review and the description of system model and helped to edit the manuscript.

For all of the above mentioned papers, the third author and my supervisor, Dr. Omid Ardakanian edited the manuscripts and provided excellent supervision.

Acknowledgements

I am enormously thankful to my supervisor, Dr. Omid Ardakanian, for giving me expert guidance and invaluable suggestion throughout my graduate studies. Without his endless support and impressive supervision, the completion of this thesis and the research papers would be impossible. Besides academic supervision, he gave me career advice and inspired me to become a polite, diligent and patient researcher like him. I am really lucky to have him as my teacher, mentor and advisor.

Special thanks to Dr. Moosa Moghimi Haji whose expert suggestion and insight in power systems helped me a lot to publish the papers and to write this thesis.

Contents

1 Introduction **1**
1.1 Widespread Adoption of Electric Vehicles 2
1.2 Control Strategies for EV Charging 3
1.2.1 Addressing Congestion in the Smart Grid 3
1.2.2 Incorporating Deadline in Fair Allocation 4
1.3 Beyond Model-based Control Strategies 5
1.3.1 Adaptive, Learning-based Congestion Control 6
1.4 Control Objectives and Different Perspectives 6
1.5 Summary of Contributions 7
1.6 Outline 9

2 Literature Review **10**
2.1 Grid-centric vs. User-centric Approaches 10
2.2 Centralized vs. Decentralized Strategies 11
2.3 Deadline-Aware Scheduling 12
2.4 Gradient-Projection Algorithms for Decentralized Control . . . 13
2.5 Model-based vs Model-free Strategies 14
2.5.1 EV Charging Control with Reinforcement Learning . . 14
2.5.2 Rule-based, TCP-like Congestion Control Algorithms for EV Charging 15

3 System Model **17**
3.1 System Overview 17
3.2 Distribution Network 18
3.2.1 Characterization of the EV Charging Load 19
3.2.2 Logical System View 21
3.3 Parking Station in ACN-Sim 21
3.4 Power Flow Model 22
3.5 Reinforcement Learning 23
3.5.1 Modeling Agents and Environment 23

4 Reputation-based EV Charging Control **27**
4.1 Methodology 27
4.1.1 Centralized Control of EV Charging Stations 28
4.1.2 Dual Problem and Projected Gradient Descent Algorithm 29
4.2 Scaled Gradient Projection Algorithm (SGPA) 29
4.3 Proof of Quadratic Convergence 32
4.3.1 Vector and Matrix Norms 32
4.3.2 Projections 32
4.3.3 Quadratic Convergence of SGPA 33
4.3.4 Values of d and L 35
4.3.5 Proof of Descent Direction Update 36
4.3.6 Convergence Rate Analysis 37

4.4 Baseline Algorithms 37

5 Experimental Results for Model-based Control Strategies 39

5.1 Case Studies 39

5.1.1 Test System 40

5.1.2 Performance Metrics 43

5.1.3 User Types 44

5.2 Results 45

5.2.1 Scenario A 45

5.2.2 Scenario B 46

5.2.3 Scenario C & D 48

5.2.4 Analyzing the Rate of Convergence 49

5.2.5 Excursion from the Rated Capacity 49

6 Model-Free Control Strategies 51

6.1 Methodology Overview 51

6.2 Centralized Controller 52

6.2.1 33-bus system 52

6.2.2 ACN Testbed 53

6.3 Adaptive AIMD 54

6.3.1 Imitation Learning 56

6.3.2 Offline Direct Policy Learning 56

6.3.3 Soft Actor-Critic (SAC) 57

6.4 Baseline Algorithms 59

7 Experimental Results for Model-Free Control Strategies 61

7.1 Simulation Environment 61

7.1.1 Performance Metrics 62

7.2 Case Study 1: Distribution Network 62

7.2.1 Addressing Transformer Overloading Problem 64

7.2.2 Voltage Control 64

7.2.3 Fairness & User Satisfaction 65

7.2.4 Adaptive Nature of A-AIMD 66

7.2.5 Resource Utilization 66

7.2.6 Adaptability to Changes in Distribution Network . . . 67

7.3 Case Study 2: ACN-Sim 70

7.3.1 Congestion Avoidance 70

7.3.2 Performance Analysis 71

7.3.3 Adaptive Nature of A-AIMD 71

7.3.4 Resource Utilization 73

8 Conclusion 74

References 77

List of Tables

2.1 Taxonomy of related work 10

5.1 Energy supplied above the rated capacity of a transformer. In Scenarios A, C and D, the values are calculated for the substation; In Scenario B, it is computed for the first distribution transformer. 49

7.1 Energy (Amp-minute) supplied above the rated capacity of different lines. 73

7.2 Energy (Amp-minute) supplied below the rated capacity through different lines. Here, phase 1, 2, 3 indicate the phase currents of the secondary side 73

List of Figures

1.1 A high level overview of this thesis 7

3.1 A distribution system with primary network (Nodes 1-33) and secondary feeders (one of them is shown inside the dashed rectangle) supplying residential loads and EV chargers. 20

3.2 The logical view of a radial distribution network. 21

3.3 Network topology of Caltech ACN testbed [33]. 22

3.4 The agents-environment interplay in a multi-agent Markov Decision Process (MDP). 24

5.1 Overview of the simulation framework. 40

5.2 Distribution of arrival times. 42

5.3 Predicting and adjusting deadlines. μ and ϵ represent the actual and predicted deadlines respectively. Conservative (risk-taking) users add (subtract) δ to (from) their prediction. 43

5.4 Substation loading at different time slots for centralized, GPA and SGPA. The horizontal (purple) dashed line represents the rating of the substation transformer. 45

5.5 Comparing performance of different algorithms for conservative and risk-taking users (Scenario A). 46

5.6 Transformer loading over one day using centralized, SGPA and GPA. The top plot is for a distribution transformer and the bottom one is for the substation transformer. 47

5.7 Transformer loading over one day using centralized, LLF and EDF. The top plot is for a distribution transformer and the bottom one is for the substation transformer. 47

5.8 Comparing performance of different algorithms for conservative and risk-taking users (Scenario B). 48

5.9 Comparing performance of different algorithms for conservative and risk-taking users (Scenarios C and D). 48

5.10 95% convergence analysis of SGPA and GPA for a specific time slot (7:00 pm) for scenario A. Initial points for both algorithms are the same. 49

6.1 An overview of the proposed methodology. 52

7.1 Distribution of arrival times. 62

7.2 Power loading of a distribution transformer under the centralized, A-AIMD and AIMD algorithms (right figure) and under DRL, LLF and EDF control policies (left figure). The dashed horizontal line is the rated capacity and the red curve is the base load. 63

7.3 The voltage of one of the primary nodes of the distribution network using A-AIMD with and without considering voltage congestion signals. The horizontal lines indicate ±5% of the nominal voltage. 65
7.4 Comparing performance of A-AIMD with the baseline algorithms. 65
7.5 Comparing rate changes under AIMD and A-AIMD. 66
7.6 Voltage profile of one of the primary nodes under A-AIMD and AIMD algorithms. In the left figure the substation voltage is increased between 5:00pm and 9:00pm (marked by dashed vertical lines). In the right figure the configuration is changed at 5:00pm (marked by the dashed vertical line). 67
7.7 Performances of A-AIMD and AIMD before and after voltage change (left panel) and reconfiguration (right panel). Note that the y-axis is truncated. 68
7.8 Average reward per episode (in MVA) of A-AIMD achieved with and without imitation learning. Here the average is taken over all time steps of an episode and all active agents. 69
7.9 The charge current of two charging pods for different algorithms. The dashed lines indicate their nameplate capacity which is $80A$. 70
7.10 Performance comparison of different control algorithms in ACN-Sim. 71
7.11 The adaptive nature of A-AIMD. 72
7.12 The currents flowing in three phases conductors on the secondary side of the transformer for different algorithms. The dashed lines indicate the thermal limit which is $416.16A$ for each phase conductor. 72

Chapter 1

Introduction

The traditional power grid has gone through substantial changes in the last few decades. These changes include the modernization of power transmission and distribution systems to accommodate future demand growth and the increased adoption of information and communications technology to achieve a sustainable, reliable and efficient electric grid, i.e., the "smart grid".

Besides residential, commercial and industrial loads, a smart grid includes elastic loads whose demands can be shaped (e.g., electric vehicles and thermostatically controlled loads), distributed energy resources (e.g., storage systems) and renewable energy systems (e.g., solar panels). The integration of these components provides several benefits to the grid and customers. For example, electric vehicles reduce carbon emission and renewable energy sources offer a sustainable and safe alternative to fossil fuels.

However, without proper control and management, the mobility of electric vehicles and the intermittency of renewable sources may lead to severe reliability and stability issues in the grid. Thus, today's smart grid requires smart and efficient control strategies to safely integrate these resources. Not surprisingly, these changes are happening in the last mile of distribution, or on the demand side, encouraging the research community to mainly focus on the distribution system in the smart grid.

In recent years, distribution grids has been equipped with high-precision measurement devices, such as distribution-level phasor measurement units (DPMUs), in addition to smart meters. These sensors can measure the magnitude and phase angle of voltage and/or current at a node with a very high temporal resolution. They have memory to store past behavior of each user and can perform basic arithmetic operations to generate a congestion signal. Moreover, broadband communication supports high bandwidth and low latency communication between these measurement devices, distributed energy resources and elastic loads. These measurement and control capabilities enable distributed optimization and feedback control of elastic loads and storage systems in the smart grid.

This thesis addresses one of the emerging challenges in the smart grid, that is how to control elastic loads, in particular charge powers of electric vehicles, to improve user satisfaction and grid's reliability at the same time.

1.1 Widespread Adoption of Electric Vehicles

Over the last decade, electric vehicle (EV) fleet has grown significantly all across the globe as EVs are essential to reduce petroleum dependence and carbon emissions in developed and developing countries [15]. At the same time, the installation of both public and private EV charging points has been expanded rapidly. According to the IEA's Global EV Outlook 2020 [3], the number of private electric vehicle charging points, which are mostly connected to low-voltage distribution networks, was approximately 6.5 millions worldwide in 2019 showing a 40% increase compared to the year before. This trend is expected to continue in the future with increased global demand for electric passenger cars and electric light and medium-duty trucks.

There are several types of electric vehicles [71] like battery electric vehicle (BEV), plug-in hybrid electric vehicle (PHEV), plug-in electric vehicle (PEV), hybrid electric vehicle (HEV), etc. In this thesis we consider only PEVs. They charge their batteries from the grid.

A high penetration of plug-in EVs in distribution networks is anticipated to give rise to several problems, such as transformer overloading, voltage limits violations, and increased heat losses [42], [72]. This necessitates a demand-side management strategy to control the real power drawn by charging stations as residential and workplace electric vehicle charging stations increase in number.

1.2 Control Strategies for EV Charging

Controlling the EV charging demand is possible due to its *elasticity*, i.e., the charge power of an EV connected to a charging station can be adjusted within certain bounds as long as its battery is charged to a desired level by a deadline [17]. Hence, the EV charging load is characterized with a fixed demand that must be met by a deadline.

To accommodate a higher penetration of EVs safely and efficiently in today's distribution networks, many control methods have been proposed. The EV charging control algorithms proposed in related work can be divided into two categories: centralized [42], [45], [63] and decentralized [4], [8], [32]. In the first category, a central controller determines an admissible charge power for every connected EV after taking into account various constraints. These charge powers are then sent to the respective charging points. Since all EV related data and other measurements must be sent to the control centre and decisions must be sent back to charging points, these methods have high communication overhead and do not usually scale with the size of the network and the number of charging points. Additionally, the central node will become a single point of failure and sending data about individual EVs to a third party could raise privacy concerns. Decentralized EV charging control methods can address these shortcomings [36], [40], [76]. In this case the control problem is decomposed into several subproblems which are solved by individual charging stations or neighbourhood controllers to determine the charge powers of the respective connected EVs.

1.2.1 Addressing Congestion in the Smart Grid

In a power grid, a congestion event occurs when a component carries more load (power or current) than its rated capacity. When the loading of a transformer or line exceeds its rated capacity we call this congestion as transformer or line overloading.

The above definition of congestion can easily be extended to voltage. The voltage, measured at a node, should be within an accepted range. If the measured value deviates from the accepted range, we say a congestion (voltage or frequency) event has occurred. In this thesis, we consider three types of congestion events, namely transformer overloading, line current overloading and voltage deviation (from ±0.05 pu), and design control strategies that can deal with congestion inspired by the design of congestion control algorithms developed for the Internet.

1.2.2 Incorporating Deadline in Fair Allocation

Despite the extensive literature on EV charging, existing methods ignore the user-specified deadlines entirely or assume they must be met at all times. The former leads to the design of a best-effort service, as in the Internet, which is not acceptable in the power grid.

Incorporating the user-specified deadlines in a scheduling policy is non-trivial. Specifically, prioritizing EV charging according to the user-specified deadlines [14] will only work if there is a well-structured billing mechanism in place, which ensures users with earlier deadlines pay a higher price for electricity. Without such a mechanism, this prioritization opens the door for manipulation in any scheduling problem [26], [49]. For example, users may claim an earlier deadline in an attempt to receive better service compared to their neighbours. This could strain the distribution grid and increase losses if the deadlines were taken into account without any adjustment. What makes scheduling with deadlines even more challenging is the non-deterministic nature of trips especially in the context of residential charging. Users can at best report their estimated deadline after adding (or subtracting) a value to it, leading to two different types of users: conservative and risk-taking users.

This thesis addresses how to handle deadlines, fairness, and efficiency requirements simultaneously by proposing a novel framework for EV charging centred around the notion of reputation. It also explains how network constraints can be taken into account to avoid straining the power grid.

The proposed framework ensures that no EV starves for charge power and all the EVs are treated fairly [8], [31]. In the context of resource allocation, fairness is defined in different ways depending on the application. This thesis adopts a fairness criterion called *weighted proportional fairness*. A proportionally fair allocation is an allocation that maximizes the utilization and satisfies the axioms of fairness in game theory. Weighted proportional fairness, on the other hand, extends the basic proportional fairness by considering weighted allocation.

We propose a deadline-aware mechanism which takes into account the user's reputation in the recent past, thereby preventing users from gaming the system by claiming false deadlines repeatedly. The reputation comprises two kinds of evidence: deadlines declared prior and past the actual departure times. Furthermore, to minimize the communication overhead and protect user privacy we design a decentralised control algorithm which allows the charging stations to compute their fair share independently based on coordination signals they receive from sensors in the distribution network.

1.3 Beyond Model-based Control Strategies

The above mentioned centralized and decentralized methods rely on a model (i.e., the admittance matrix) which relates EV charge powers to voltages and real power flows in different parts of the distribution network. But the distribution system model is often inaccurate or nonexistent in practice [6]. In the absence of an accurate model, it is possible to incorporate an approximate model which ignores losses and reactive power flows [8]. Nevertheless, due to these approximations, the resulting model cannot be used to control voltage and does not guarantee that available resources are fully utilized.

A promising alternative, inspired by the design of Internet congestion control algorithms, is to employ a feedback control algorithm which infers *congestion* (i.e., overloading and voltage problems) based on measurements of local or remote sensors rather than using the network model to determine if there is congestion in some part of the network. Specifically, the charging points can adjust their power based on the received feedback such that the overall EV charging demand tracks the available capacity of the network in real-time [5], [64]. These algorithms draw upon a rule-based control scheme, called Additive-Increase Multiplicative-Decrease (AIMD), which has been successfully adopted in TCP congestion control [19]. In AIMD, the control rules are fixed regardless of the changes in the network condition [35]. Employing fixed rules is not a major problem for fast-timescale control, but can lead to inefficiencies when control decisions are made on a slower timescale, e.g., on the order of seconds in the distribution grid. Thus, we explore how to update the control rules dynamically to increase the overall network utilization and responsiveness [75].

1.3.1 Adaptive, Learning-based Congestion Control

Inspired by [35], we propose an Adaptive AIMD-like method (dubbed A-AIMD) for controlled EV charging. Specifically, we use the Multi-Agent Reinforcement Learning framework to adjust the additive and multiplicative rates of AIMD at each charging point assuming that it is a Reinforcement Learning (RL) agent. The RL agents are trained such that they can react to different conditions of the grid given the congestion signals they receive. Our control method is model-free and requires reliable transmission of congestion signals only. It scales with the size of the distribution network and has the same communication overhead as other decentralized feedback control algorithms that mimic TCP congestion control [5], [64].

1.4 Control Objectives and Different Perspectives

The objectives of this thesis and different perspectives (user-centric versus grid-centric) that are taken to control EV charging are depicted in Figure 5.1. The objectives are achieved using two approaches: 1) model-based where having an approximate model of the grid is assumed, and 2) model-free where we do not have such a model but there are sensors installed in the network to enable a feedback control mechanism.

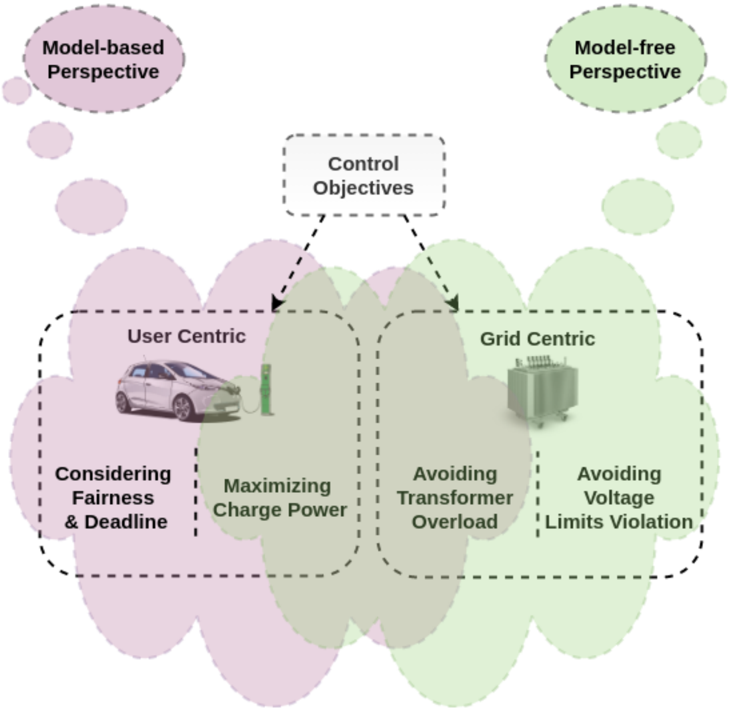

Figure 1.1: A high level overview of this thesis

From the model-based perspective, we address the fairness, deadline and charge power maximization to achieve user-centric objectives. For the grid-centric objectives, our optimization-based methods (both centralized and distributed) maximize network utilization considering the rated capacity of distribution transformers to avoid transformer overloading.

On the other hand, from the model-free perspective, our adaptive AIMD algorithm learns to maximize the network utility to earn user satisfaction while avoiding two types of congestion, namely transformer overloading and voltage limits violations.

1.5 Summary of Contributions

In this thesis we explore the distributed control of EV charge powers in the smart grid from two perspectives. First, a model-based approach where we have an approximate model (i.e. the admittance matrix) of the grid. The admittance matrix defines the network topology matrix and parameters of lines and transformers. We consider an approximate model which is a simple DC model that ignores line losses and reactive power flows.

The second approach is model-free as it does not require the admittance matrix. Instead of a model it requires a feedback system (for sending a reward signal) installed in the grid. Whenever a power or voltage congestion happens, this system sends a congestion signal to the controller. The controller adapts its charging rate based on the received feedback signal.

The contributions of this thesis are as follows:

- We propose a reputation-based mechanism for charging plug-in EVs that fairly allocates charge powers to prevent overloading of substation and distribution transformers;

- We present a scaled gradient-projection algorithm for decentralized control of EV charging and prove that it has super-linear convergence when its parameters are tuned carefully;

- We simulate our method on a test distribution network with real EV data and compare it with different baseline algorithms to show its superior performance and that it favors conservative users.

- We propose an adaptive AIMD-like algorithm for controlled EV charging in the smart grid. To train RL agents to adapt the parameters of AIMD to the varying network conditions, we use imitation learning [73] (in the offline phase) and soft actor-critic [23] (in the online phase).

- We evaluate the efficacy of our adaptive model-free control method (i.e., adaptive AIMD) in preventing transformer overloading and voltage limit violation problems through simulations carried out on the ACN simulation testbed [34] and on the 33-bus distribution network [11] with real EV and household demand data.

- We compare our adaptive model-free control method with various EV charging control methods that avoid congestion and show that it outperforms these baselines.

1.6 Outline

Chapter 2 presents related work on EV charging algorithms from different points of view, e.g., centralized vs. decentralized, and model-based vs. model-free approaches. Chapter 3 introduces the system model. The assumptions are stated, the physical and logical views of the system are presented, and the power flow equations and sensor types and specification are discussed in this chapter. Chapter 4 formulates the centralized, optimal control strategy considering reputation and deadlines. Then a distributed version of this control method is presented with its convergence analysis. A rigorous mathematical proof of convergence is presented and the super-linear (i.e. quadratic) rate of convergence is shown. Chapter 5 describes the simulation environment, performances metrics, and baseline algorithms for evaluating the control strategies presented in Chapter 4. It also presents the performances of our proposed methods and discusses how they can avoid congestion and outperform other baseline algorithms.

Chapter 6 presents the model-free, reinforcement learning-based control algorithms that prevent network congestion. It first discusses how multi-agent reinforcement learning is used in a distribution network to control the charge power of EVs. It then presents the adaptive version of the AIMD algorithm that maximizes the network utilization while avoiding two types of congestion: transformer overloading and voltage limits violations. This chapter also formulates the optimal control method for ACN-Simulator and a large distribution network. Chapter 7 includes details about the simulation environment for the reinforcement learning based controller. It also presents the performance metrics and graphs for different algorithms proposed in Chapters 6. Chapter 8 concludes the thesis with a discussion about the limitations of the work presented in this thesis and potential future extensions of this thesis.

Chapter 2

Literature Review

There has been many efforts to control the charge power of EVs in the smart grid [70]. These strategies can be classified and discussed from different viewpoints which are summarized in Table 2.1 and outlined in this chapter.

Table 2.1: Taxonomy of related work

Reference	**Model-based/ Model-free**	**Centralized/ Decentralized**	**Real-time/ Day-ahead**
[28], [30], [36], [55]	Model-based	Decentralized	Real-time
[41], [74]	Model-based	Centralized	Day-ahead
[44], [57], [60]	Model-based	Centralized	Real-time
[38], [39], [58]	Model-based	Decentralized	Real-time
[20], [59], [69]	Model-free	Centralized	Real-time
[69]	Model-free	Centralized	Day-ahead
[1]	Model-free	Decentralized	Day-ahead
[61]	Model-free	Decentralized	Real-time

2.1 Grid-centric vs. User-centric Approaches

The optimal scheduling of EV charging has been extensively studied in recent years. The scheduling objectives can be broadly divided into two categories. The main objective of the first category is to provide voltage and frequency support to the grid. For example, [36], [55] focus on voltage regulation using the EV charging load, while [28], [30] focus on providing secondary frequency support to the grid. In the second category, the emphasis is on optimizing some performance metric for EV owners rather than considering the services that can be offered to the grid. References [41], [74] propose day-ahead scheduling methods to maximize the revenue of EV owners considering varying electricity prices. A method is proposed in [29] for secondary frequency support, while reducing the battery degradation and maximizing the revenue of EV owners.

2.2 Centralized vs. Decentralized Strategies

The control algorithms proposed in related work can be broken down into two categories: centralized and decentralized. In the first category, a central controller calculates the admissible charge power of every connected EV for the next time slot. All user data and measurements are sent to the control center which usually solves an optimization problem and sends back control signals to individual charging stations [57], [60]. These control signals or charge powers are derived to achieve some goals. For example, centralized scheduling methods are proposed to maintain the voltage profiles [10], to minimize losses in the grid besides improving the voltage profiles [44], and to prevent congestion in the grid [25]. Since all EV related data and network measurements must be sent to a control centre and the decisions must be sent back to charging points, these methods have high communication overhead and do not scale with the size of the network and the number of charging points. When there are many sensors and charging stations in the system, there might be congestion in the communication network as well. Additionally, the central node will become a single point of failure, and sending EV data to a third party might raise privacy concerns. Furthermore, sending data to a control center raises privacy concerns, especially if this data contains EV arrival and departure times.

To address these shortcomings, several decentralized EV charging control methods have been proposed. [38], [58]. These methods are compelling when the EV penetration level is high, but they typically suffer from slow convergence in large distribution systems. References [36] propose decentralized shrunken-primal-dual sub-gradient (SPDS) algorithms for EV charging control to improve the demand profile while meeting heterogeneous individual charging requirements of EVs and satisfying distribution grid power quality requirements. This method is a variant of regularized Lagrangian which requires two-way explicit communication and exhibits a slow convergence rate. In [40], a SPDS algorithm is developed for decentralized control. However, this method also requires two-way communications between charging stations and operators, whereas the method proposed in this thesis requires only one-way communication by leveraging local information and the physical system.

In [5] authors propose an EV charging controller inspired by the TCP slow start mechanism used in the Internet. The authors use the traffic light model, a congestion notification mechanism to report the status of the grid in real time. However, this method does not solve an optimization problem to establish the optimal control, and therefore fairness is not theoretically guaranteed. In recent work [76], a decentralized algorithm is proposed based on dual decomposition. This algorithm requires one-way communication between the sensors and downstream EV chargers, and exhibits faster convergence than the previous methods. However, the method assumes the knowledge of the distribution network model and still needs several iterations to converge to the optimal charge powers.

2.3 Deadline-Aware Scheduling

Incorporating deadlines in scheduling has been extensively studied over the past three decades in the context of real-time scheduling of computer systems, including operating system, distributed systems, and data centers. Reference [24] studies the minimum laxity policy (ML) with multiple processors/servers and the preemptive earliest deadline (ED) policy with a single server, and presents different performance bounds for these policies. In [46], the impatient customers queue model is introduced where jobs (customers) request to complete execution or, at least, reach the server before a given deadline. The problem is modelled as a stochastic recursive sequence that keeps track of all service and patience times of the customers present in the system. The Earliest Progressive Deadline First is proposed in [22] as a class of policies considering server-side job scheduling with progressive deadlines.

However, little work has been done on EV scheduling considering inaccuracies in the reported deadlines. In an expandable service system, variability in service capacity incurs infrastructural and operational costs. To address this problem, an optimization framework is proposed in [48] to reduce service variance while maintaining deadlines and demands of jobs. Although it primarily focuses on cloud services, the authors have considered an EV charging scenario. Our work (chapter 4) differs from this work in several ways. First, their proposed method solves the optimization problem once at the beginning of the time horizon based on some statistics about future behavior, whereas our approach is real time and makes decisions based on the currently available information. Second, they assume that the user-specified deadlines are precise, whereas in our model, the specified deadlines may be different from the actual deadlines. The closest work to the method proposed in Chapter 4 formulates the EV charging control with deadline as a feasibility problem [47]. However, inspired by the best effort service in the Internet, our proposed method uses a "soft deadline" which means that it does its best to meet the demand of urgent EVs first. Additionally, we safeguard the system from manipulation that could result from claiming urgency in a deceitful manner.

2.4 Gradient-Projection Algorithms for Decentralized Control

The first-order gradient-projection algorithm (GPA) has been applied to many resource allocation problems to find the solution in a distributed fashion. For example, this method is employed in [43] to solve the network utility maximization problem. The authors present the synchronous and asynchronous versions of GPA. The synchronous version of GPA is also used in [7], [8] to control charge powers of EVs in a power distribution system. In these papers the power grid is used as a medium for sending implicit messages from the end nodes to distributed controllers, thereby reducing the communication overhead. Although GPA exhibits stable convergence, its rate of convergence is linear, taking many iterations to converge especially in large distribution systems.

The second-order gradient-projection algorithm converges faster than the first-order algorithm. The scaled gradient-projection algorithm was discussed by Bertsekas and Tsitsiklis [12]. We use thin algorithm in Chapter 4 to control EV charging in decentralized fashion.

2.5 Model-based vs Model-free Strategies

The above mentioned centralized and decentralized approaches assume the knowledge of the distribution network model. These methods require an approximate model (i.e. the admittance matrix) of the grid to map the charge powers of EVs to the congestion states of the grid. However, in practice, such model may not be available. Thus, researches are encouraged to design model-free control strategies.

In the literature, two types of model-free EV controllers are discussed: rule-based, TCP-like congestion control algorithms [64] and reinforcement learning (RL) based control methods [20], [59], [69].

In the first type, the charge powers are adjusted based on some simple predefined rules. For example, when there is no power/voltage congestion, the charge power of the EVs are increased additively. However, as soon as congestion is detected, the charge powers are decreased multiplicatively. In the later type, EV chargers are enabled to learn the rules to maximize the network utilization while avoiding congestion.

This is achieved via reinforcement learning (RL) algorithms that learn an optimal policy through interaction with the grid or a grid simulator.

2.5.1 EV Charging Control with Reinforcement Learning

Many reinforcement learning-based EV charging methods have been proposed in the literature. These methods have attracted much attention since they do not require the system model. One particular line of work focuses on scheduling a group of EVs, typically connected to an aggregator [20], [59], [69]. Although the results reported in these studies are satisfactory, they attempt to solve the problem in a centralized fashion, hence the proposed methods suffer from the same drawbacks as other centralized control methods. Another line of work focuses on charging a single EV considering different objectives, such as providing vehicle-to-grid support or reducing long-term electricity costs [1], [61]. The proposed methods are decentralized, but they do not discuss the coordination among multiple EVs connected to a power system. Reinforcement Learning has also been used to control electric buses (EBs). In [18], a Double Q-learning based EB control strategy is proposed to minimize the battery degradation cost. In contrast to these methods, we focus on decentralized and coordinated scheduling of EV charging without making any specific assumption about how EVs are connected to the power system.

2.5.2 Rule-based, TCP-like Congestion Control Algorithms for EV Charging

Congestion control in communication networks is a well-studied problem. As shown in [19] a simple AIMD algorithm satisfies the sufficient conditions for convergence to an efficient and fair state regardless of the starting state. Reference [9] proposes an AIMD algorithm that captures the key features of the TCP congestion control protocol. The authors discuss the fairness of their method in the steady state when all sessions have the same mean throughput.

The success of AIMD-based methods in TCP congestion control inspired researchers to apply them to control congestion in the power system. In [64] the authors propose a decentralized charging strategy based on the AIMD algorithm to maximize power utilization of EVs while allocating power to them in a fair fashion. However, they do not consider practical power system constraints, such as the rated capacity of transformers and voltage constraints. In [37], the authors propose modifications to the basic AIMD algorithm of [64] to incorporate these constraints assuming a hierarchical communication network. In addition, they propose a price-adjusted available power heuristic to encourage shifting of the EV charging load to off-peak times.

More recently, a fully decentralized AIMD-like EV charging algorithm in proposed in [68]. This algorithm relies on local voltage measurements only and can be tuned to have near-optimal decentralized operation. However, fine-tuning the parameters is a nontrivial problem. Since the method relies on local information, it may not be able to fully utilize the available capacity of the network. Moreover, the method does not consider fairness in allocating power to the EVs. The authors of [68] have also proposed AIMD-like EV charging approaches that considers fairness [67]. The proposed methods require the knowledge of the system model which is not typically available in practice. Reference [5] proposes a EV charging controller inspired by the TCP slow start mechanism used in the Internet. The method uses the traffic light model which is a congestion notification mechanism to report the status of the grid in real time. Despite the novelty of this approach, it cannot adjust the amount of increase or decrease in the charge power of EVs based on real-time dynamics of the power system.

Reference [62] proposes a modified AIMD algorithm for communication networks to ensure high network utilization regardless of the level of flow synchronization or the number of flows traversing the link. In a recent study [35] the authors discuss the limitations of the rule-based design of TCP congestion control, noting that it is unable to adapt to changes in the network condition. In addition, the rules are built upon standard assumptions and network models, making them unable to adjust intelligently by learning from experience. To address these problems, they propose an RL-based congestion control protocol, called QTCP (Q-learning based TCP), and show that it outperforms traditional rule-based TCP congestion control protocol. Inspired by this work and given the nature of EV charging control problem in the smart grid, we propose an RL-based adaptive AIMD method (in Chapter 6). Our RL agents learn how to react to congestion by interacting with the environment.

Chapter 3

System Model

In this chapter, we discuss the system models and state assumptions for our proposed control strategies. Section 3.1 describes the overview of two power systems: a distribution network and a parking station. Section 3.2 and Section 3.3 discuss the distribution grid and the parking station's power system respectively. Section 3.4 describes the power flow equations that are used by the model-based algorithms in Chapter 4. Section 3.5 presents the reinforcement learning system that is used to study the model-free controllers in Chapter 6.

Note that, the distribution network is used to study both the model-based (Chapter 4) and model-free (Chapter 6) control strategies. However, the parking station system is used to evaluate the model-free controllers only.

3.1 System Overview

We study the EV charging problem in the context of a distribution network [11] (shown in Figure 3.1), and a grid-tied parking station with several EV charging points [34] (shown in Figure 3.3). These power systems have been used in previous work to study controlled EV charging, yet they differ in several aspects, including size, operational structure, types of loads, and physical constraints. For example, EV charging points are the only type of loads connected to subfeeders in the parking station, whereas feeders in the distribution network supply a mix of residential loads and EV chargers. We consider both of these systems to perform a thorough evaluation of the proposed EV charging method.

We use a discrete-time modelling approach and make the following assumptions in the analysis of both systems. We assume that a measurement device is installed at the secondary side of every distribution transformer that is likely to be overloaded. Furthermore, we assume that there is a voltage sensor at every bus that is susceptible to undervoltage. These sensing nodes are assumed to have enough computation power and memory to perform simple arithmetic operations on the measured quantities and a network interface card for sending out the computed values. This capability is necessary for computing the *congestion signal* and communicating it to charging points. The congestion signal indicates that a constraint is violated, e.g., the voltage level has gone below 0.95 *pu* or the instantaneous load on a transformer has exceeded its short-term rating. We also assume that there is an overlay communication network which connects the sensors installed in the power system to downstream charging points. This neywork has low latency and high bandwidth.

3.2 Distribution Network

To accurately model the 33-bus distribution feeder, we consider both the primary and secondary (low voltage) feeders supplying the EV chargers as well as inelastic loads (e.g., residential loads). Each low-voltage feeder is connected to the primary distribution network via a step-down transformer. Figure 3.1 shows the low-voltage feeder connected to Node 25 of the primary distribution network (similar low-voltage feeders are connected to nodes 2 to 33 of the primary network via distribution transformers but are not shown in the figure). The primary distribution network is in turn connected to the substation via the substation transformer. We assume that both primary and secondary feeders have a radial operational structure. All loads, including EV chargers and home loads, are connected to leaf nodes (i.e., the end nodes) in the secondary network.

We aim to control the charge power of EV charging points to maximize the utility of the grid, avoid overloading of distribution transformers, and keep nodal voltages at the primary distribution system between 1.05 and 0.95 *pu* at all times. We define the utility of the grid operator as the sum of charge powers of all EVs at any point in time. To control EV charging, we rely on real-time measurements provided by distribution-level phasor measurement units (DPMUs) installed on the secondary side of each transformer. The DPMUs can synchronously measure the three-phase voltage and current phasors at regular intervals. Once voltage and current phasors measured at the secondary-side of the transformer are available, it is possible to compute the transformer loading and subtract it from the transformer's short-term rating to estimate the congestion state in volt-ampere. For voltage congestion, the congestion state is expressed in per unit nodal voltage minus the lower voltage limit (i.e., 0.95). In either case, the congestion state indicates the available capacity of the distribution network. In our proposed control method, each charging point needs to receive a congestion signal once the transformer is overloaded or a nodal voltage drops below the lower voltage limit. We assume that each EV charging point is connected via an overlay network to the sensors installed at its upstream transformers (i.e., the substation and distribution transformers that supply its demand) and receives a congestion signal if they are overloaded.

3.2.1 Characterization of the EV Charging Load

Each EV, is characterized by a tuple $\langle U_e, \alpha_e, d_e, \sigma_e \rangle$ where $U_e(x_e)$ represents the utility of its owner (a measure of their satisfaction) which is an increasing function of its charge power x_e at any time, α_e represents its arrival time, d_e represents its energy demand (the difference between its current state-of-charge and its desired state-of-charge), and σ_e which is the user-specified deadline (the latest time by which its energy demand must be met). We denote the remaining charging duration by $\widetilde{\sigma}_e = \sigma_e - \alpha_e$. The specified deadline of an EV may not be precise, meaning that it may depart before or after the specified deadline. We use a discrete time model and assume that EVs arrive or depart at the beginning of a time slot. We denote the length of time slots by τ.

We assume that each charging station keeps track of discrepancies between the claimed and true deadlines of the corresponding EV. Each time an EV is plugged in, the charging station records the specified deadline which is then

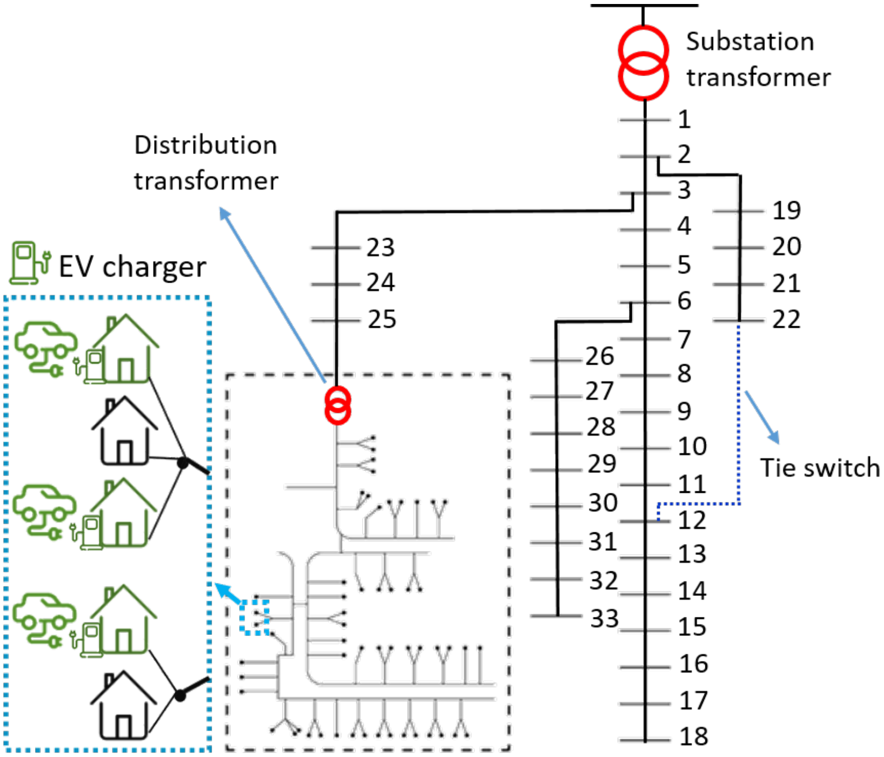

Figure 3.1: A distribution system with primary network (Nodes 1-33) and secondary feeders (one of them is shown inside the dashed rectangle) supplying residential loads and EV chargers.

compared with the time that the EV is unplugged from the charger. The charging station computes the difference between these two values, which we refer to as *historical discrepancy*, and stores it for every user (EV driver). The historical discrepancies collected in a fixed-size window are used by the charging station to update the reputation of a user at the beginning of each time slot. This reputation is used as a weight in the optimization problem discussed in the next section. In case of the centralized method, user reputation is sent to the central controller which solve the optimization problem. For decentralized methods, the reputation is used by the charging station to independently compute its charge power for the next time slot in a number of iterations.

The charging station is assumed to be tamper-resistant and can charge one EV at a time. It delays charging of an EV for some time, i.e., known as the *idle time*, if it is unplugged and plugged again after a short period of time. This prevents users from reporting an unrealistically early deadline and simulating a departure event at that deadline to maintain a high reputation.

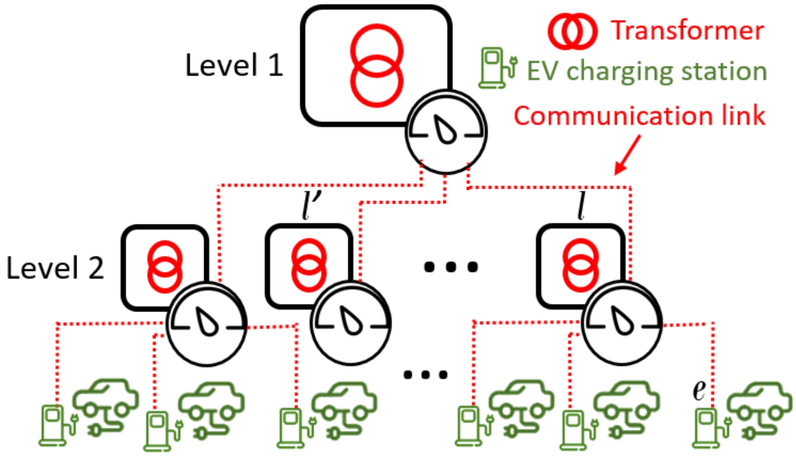

Figure 3.2: The logical view of a radial distribution network.

3.2.2 Logical System View

We denote the set of transformers installed in the distribution network by $\mathcal{L}$, the set of all connected EVs in the network by $\mathcal{E}$. We use matrix T to encode the point of connection of EV charging stations. In particular, $T_{el} = 1$ for every transformer l which supplies the charging station indexed by e. This element of the matrix is zero otherwise. Figure 3.2 shows an EV connected to a charging station e. This EV is supplied by distribution transformer l but not distribution transformer l'. Hence we have $T_{el} = 1$ and $T_{el'} = 0$. We denote the available capacity (i.e., the difference between its rated capacity and the total demand of inelastic loads downstream of the transformer) of transformer l by a_l. Note that we use a ballpark estimate of losses in primary and secondary distribution feeders and use a fixed power factor to calculate the loading of a transformer from the net demand of its downstream inelastic loads. This approximation leads to a small error as we discuss later.

3.3 Parking Station in ACN-Sim

We consider the EV parking station, simulated in ACN-Sim, as one of our systems. ACN-Sim [34] provides a simulation framework for benchmarking control algorithms, an evaluation library, and a data set, called ACN-Data, which is comprised of over 35,000 real EV charging sessions from the Caltech Adaptive Charging Network (ACN) and similar sites in California [33].

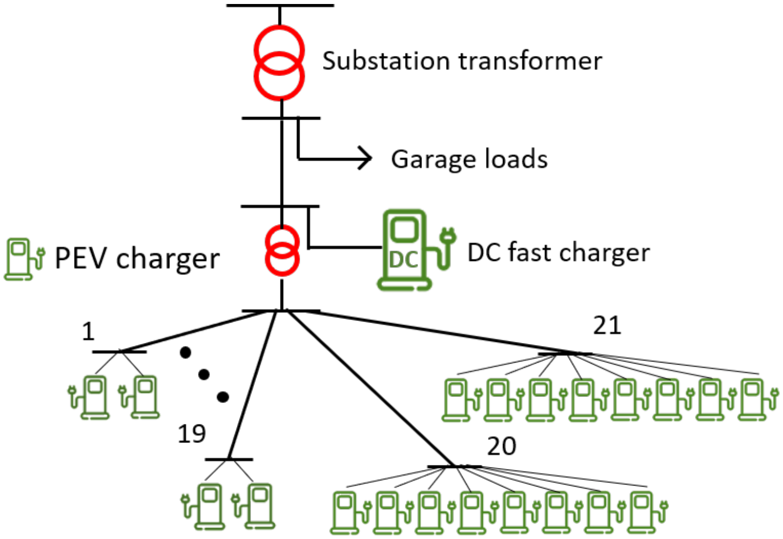

Figure 3.3: Network topology of Caltech ACN testbed [33].

Figure 3.3 shows the structure of the power system that corresponds to this testbed. According to [33], this parking station is equipped with a DC fast charger rated at 50kW and 54 level-2 Electric Vehicle Supply Equipment (EVSE) each rated at 6.6kW. These components are fed from a 150kVA transformer. Each line originating from the EV switch panel has a capacity of 80A. A total of 19 lines feed pairs of two 32A AeroVironment EVSE, and 2 lines feed pods of 8 EVSE each, one pod of AeroVironment (AV) stations and one pod of Clipper Creek (CC) stations. We assume all these lines are equipped with sensors that measure the line currents at regular intervals. The 50kW DC fast charger is a 3-phase load connected at the primary side of the transformer. The EV charging circuit is connected to the main switch panel via a three-phase connection with each line capable of carrying up to 180A.

3.4 Power Flow Model

The model-based algorithms, described in Chapter 4, require a model of the distribution grid. Here, this model is the admittance matrix that relates the power in a bus to the corresponding nodal voltages and currents. There are two different power flow models: the bus injection model and the branch flow model.

In [65], authors show that these two models are equivalent. We present the bus injection next:

Let $G = (N, E)$ be an undirected graph that represents the power network. Here, $N = \{0, 1, 2, \ldots, n\}$ and $E \subseteq N \times N$ and $(i, j) \in E$ if and only if there is a direct link between bus i and j. We define the $(n+1) \times (n+1)$ admittance Y as

$$Y_{ij} = \begin{cases} \sum_{(k,i)\in E} y_{ik}, & \text{if } i = j \\ -y_{ij}, & \text{if } i \neq j \text{ and } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

Here, $y_{ij} = 1/z_{ij}$ is the admittance (i.e. reciprocal of impedance) between bus i and j (if they are connected).

For $i \in N$, let s_i and V_i be the complex net power and voltage respectively on bus i. Let I_i^* and S_i be the complex current (conjugate transpose) and power injections respectively from bus i to the rest of the network. Then the following set of equations define the bus injection power flow model:

$$\begin{aligned} \text{Kirchhoff law :} & I = YV \\ \text{Power definition :} & S_i = V_i I_i^*, && \forall i \in N \\ \text{Power balance :} & s_i = -S_i, && \forall i \in N \end{aligned}$$

Given the admittance matrix Y, we can easily solve power flow equations to determine bus voltages and power flows. We can also construct the topology matrix, T, defined in Section 3.2 .

3.5 Reinforcement Learning

For model-free control of EV chargers, discussed in Chapter 6, we use reinforcement learning. This section describes the multi-agent reinforcement learning framework and a supervised-learning method to improve sample efficiency of the reinforcement learning algorithm.

3.5.1 Modeling Agents and Environment

A reinforcement learning agent learns a *policy* – a sequence of actions – that maximizes the expected reward received over time from interaction with the

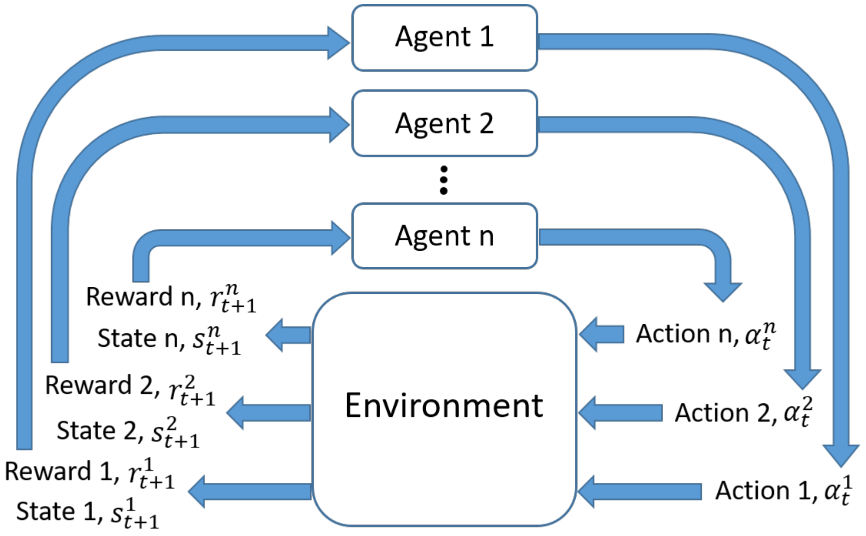

Figure 3.4: The agents-environment interplay in a multi-agent Markov Decision Process (MDP).

environment at discrete time steps. The optimal policy learned by the agent is the solution to a stochastic optimal control problem which can be framed using a Markov Decision Process (MDP). Specifically, in each time step, the agent receives some representation of the environment's state, s_t, and on that basis takes an action, a_t, which could change the environment's state, s_{t+1}. It then receives a numerical reward, r_{t+1}, from the resulting environment.

In multi-agent reinforcement learning, N independent agents try to maximize a reward signal they simultaneously receive from the environment. Figure 3.4 shows the agent-environment interaction in the multi-agent setting. Note that the rewards received by different agents are not necessarily the same and the agents cannot directly communicate with each other.

In our reinforcement learning problem, the action-space is continuous and each charging point is an agent that takes an action which determines the parameter of the A-AIMD (Adaptive Additive-Increase/Multiplicative-Decrease) algorithm (described in chapter 6) that modulates its charge power. Specifically, agent i samples an action $-1 \leq \alpha_t^i \leq 1$ at time step t as follows:

$$\alpha_t^i \sim \pi_\theta^i(\alpha|s_t^i) = \mathrm{P}(A_t^i|S_t^i = s_t^i; \theta),$$

where π^i is the policy parameterized by θ, and s_t^i is the environment state perceived by the agent. We consider a Gaussian policy with parameters $\mu_\theta, \sigma_\theta$

given by neural networks:

$$\pi_\theta^i(\alpha|s_t^i) = \frac{1}{\sqrt{2\pi\sigma_\theta^2}} \exp\left(-\frac{(\alpha-\mu_\theta)^2}{2\sigma_\theta^2}\right)$$

To update the policy parameter, θ, we use the soft actor-critic algorithm (introduced later). We divide the agent-environment interaction into a number of episodes where each episode spans one day. Upon convergence, we find a policy that maximizes the expected reward.

We define the partial state of the environment perceived by agent i at time t as a tuple:

$$s_t^i = (t, x_t^i, \Lambda_t^i), \tag{3.1}$$

where x_t^i is the charge power of the charging point controlled by agent i, and Λ_t^i compactly represents the congestion signals received by agent i at time t:

$$\Lambda_t^i = \{\lambda_j | j \in \mathcal{L}_i\} \text{ and,} \tag{3.2}$$

$$\lambda_j = \begin{cases} 1, \text{if congestion is detected} \\ 0, \text{ otherwise} \end{cases} \tag{3.3}$$

Note that agent i will receive (binary) congestion signals only from the sensors located on its path to the substation (i.e., members of the set $\mathcal{L}_i$). The sensors detect congestion either by comparing the transformer (or line) loading with its nominal rating, or by comparing the node voltage with the lower voltage limit. In both cases, exceeding the limit is deemed as congestion. We model the state transition probability, $\mathrm{P}(S_{t+1}^i | S_t^i = s_t^i, A_t^i = \alpha_t^i)$, using a neural network which is trained based on interaction with the environment (described later).

When an agent takes action α, it is translated into a new charge power for the next time slot following the rules of the A-AIMD algorithm. Charging the EV at this rate causes a transition from s_t to s_{t+1}. The agent then receives a reward, r_{t+1}, from the sensing node installed at the substation. Due to the Markov property, the reward and the next state are independent of previous states given the current state. In our problem formulation, the reward received by each agent is the instantaneous loading of the substation transformer when there is no congestion, and the negative of that value when at least one of the sensors on its path to the substation detects congestion:

$$r = \begin{cases} Load, & \text{if no congestion} \\ -Load, & \text{otherwise} \end{cases} \tag{3.4}$$

Note that the measured substation load is a function of the total power drawn by all EV charging points (i.e., $\sum_i x_t^i$). Thus, the agents receive a higher reward when the substation's utilization increases assuming that there is no transformer overloading or undervoltage problem on their path to the substation. When at least one of these problems occurs, the agents who might have contributed to the problem (i.e., located downstream of the congested resource) will receive a negative reward.

Chapter 4

Reputation-based EV Charging Control

In this chapter we propose and formulate the control methods that incorporate the reputation of EV owners. Section 4.1 presents the centralized, optimization-based control strategy. Section 4.2 proposes the decentralized, second-order gradient projection (SGPA) algorithm, and Section 4.3 analyzes its rate of convergence and derives appropriate values of some important parameters. Finally, Section 4.4 lists the baseline control algorithms that are used for performance comparison in the next chapter.

4.1 Methodology

A possible approach to charging EVs with different urgency parameters is to design a mechanism that allows EV drivers to buy electricity at a higher price if their charging demand is more urgent. This approach assumes the existence of a billing system and is therefore impractical in today's distribution grids, because existing electric vehicle supply equipment (EVSE) installed in residential buildings has a simple plug-and-charge interface. An alternative approach, which we take in this thesis, is to design a real-time scheduling mechanism that relies on user-specified deadlines. These deadlines are modified according to the reputation of EV drivers, where the reputation is a numerical value calculated based on historical discrepancies as explained in this section.

4.1.1 Centralized Control of EV Charging Stations

In the beginning of a time slot, t, each active charging station computes a weight, $w_e(t) = f(\Delta_e, \Lambda_e(t))$ for the corresponding EV, where Δ_e is a statistic of historical discrepancies (e.g., the mean) and $\Lambda_e(t)$ is the laxity defined as

$$\Lambda_e(t) = \widetilde{\sigma}_e(t) - \frac{d_e(t)}{M_e}$$

where M_e is the maximum charge power supported by EVSE. The function f should have some intuitive properties. Specifically, it should be (a) positive valued and bounded (otherwise the objective function (6.1) may lose concavity); (b) decreasing as discrepancy or laxity becomes larger; (c) robust to outliers (i.e., outliers in historical data should not greatly affect the weight); (d) numerically stable, i.e., does not return very small or large values. Hence, we choose the following function that satisfies all these properties:

$$F(\Delta_e, \Lambda_e) = exp\left(\frac{-(\Delta_e + \Lambda_e)}{\beta}\right) \tag{4.1}$$

Here β is a scaling factor to normalize the sum, $\Delta_e + \Lambda_e$.

Let $0 < \zeta \leq 1$ be an efficiency factor that approximates losses in the distribution network. Hence, the effective capacity of transformer l can be written as ζa_l. Solving the following optimization problem yields the charge power, $x_e(t)$, of every charger e for interval $[t, t+1)$:

$$\begin{aligned}
\text{maximize: } & \sum_{e \in E} w_e \log U_e(x_e) & & (4.2)\\
\text{subject to: } & \sum_{e \in E} T_{el} x_e \leq \zeta a_l, & \forall l \in \mathcal{L} \quad & (4.3)\\
& 0 \leq x_e \leq M_e, & \forall e \in \mathcal{E} \quad & (4.4)
\end{aligned}$$

Suppose $U_e(x_e) = x_e$ and $w_e = F(\Delta_e, \Lambda_e)$. For brevity the time index is dropped from all variables in the above problem. This convex optimization problem can be modelled in CVXPY and solved using Mosek software. It is shown in [8] that the solution of this problem is a *proportionally fair* allocation of the available network capacity to EV chargers. We refer to this solution as the centralized solution and use it as a baseline.

4.1.2 Dual Problem and Projected Gradient Descent Algorithm

The Lagrangian function of the primal problem is:

$$g(\lambda) = \max_{0 \preceq x \preceq \mathcal{M}} \left\{ \sum_e w_e \log x_e + \sum_l \lambda_l \left(\zeta a_l - \sum_e T_{el} x_e \right) \right\}$$

where, $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{|\mathcal{L}|}) \succeq 0$ are the Lagrangian multipliers for the constraints (6.2) and (6.3), respectively. Following [8], the Lagrangian function, $g(.)$ can be expressed more compactly as:

$$g(\lambda) = \sum_l \lambda_l \zeta a_l + \max_{0 \preceq x \preceq M} \left\{ \sum_e f_e(x_e; \lambda_e) \right\} \tag{4.5}$$

where

$$f_e(x_e; \lambda) = w_e \log x_e - \left(\sum_l \lambda_l T_{el} \right) x_e \tag{4.6}$$

Thus, the dual problem can be written as:

$$\begin{aligned} \text{minimize: } & g(\lambda) \\ \text{subject to: } & \lambda \succeq 0 \end{aligned} \tag{4.7}$$

By decomposing the dual problem into a number of subproblems that can be solved given the Lagrangian multipliers [51], it is possible to develop an iterative algorithm to compute the fair shares of EV chargers in a decentralized fashion. It requires synchronization among charging stations and sensors installed at transformers. This GPA-based method is described in Section 4.4 and is considered as a baseline in this paper. In the next section, we propose a synchronous algorithm that is faster and has lower communication overhead compared to baseline algorithms.

4.2 Scaled Gradient Projection Algorithm (SGPA)

We propose a projected Newton's method that computes the second-order partial derivatives of the dual objective function. Specifically, in each iteration, it updates the Lagrange multiplier λ as follows:

$$\lambda^{(k+1)} = \left[\lambda^{(k)} - \gamma \left[D\left(\lambda^{(k)}\right)\right]^{-1} \nabla g\left(\lambda^{(k)}\right)\right]^{+} \tag{4.8}$$

Here k denotes a specific iteration, $0 < \gamma \leq 1$ is the step-size parameter, $\nabla g(.)$ is the gradient, and $D(.)$ is a diagonal matrix consisting of the principal-diagonal elements of the Hessian matrix H projected onto the positive axis:

$$[D(\lambda)]_{ll} = \begin{cases} \eta, & \text{if, } [H(\lambda)]_{ll} < \eta \\ [H(\lambda)]_{ll}, & \text{otherwise} \end{cases} \tag{4.9}$$

Note that η is a small positive number used to make the matrix D invertible, and

$$\begin{aligned} [H(\lambda)]_{ll} &= \frac{\partial^2 g}{\partial \lambda_l \partial \lambda_l} \\ &= \frac{\partial}{\partial \lambda_l}\left(\zeta a_l - \sum_e T_{el} x_e^*\right) \\ &= -\sum_e T_{el} \frac{\partial x_e^*}{\partial \lambda_l} \\ &\approx \left| \frac{\sum_e T_{el} \Delta x_e^*}{\Delta \lambda_l} \right| \\ &= \left| \frac{\text{load}_l^{(k)} - \text{load}_l^{(k-1)}}{\lambda_l^{(k)} - \lambda_l^{(k-1)}} \right| \end{aligned} \tag{4.10}$$

The last derivation above holds if inelastic demands change at a slower rate than the rate at which we update the charge power of EV chargers. Note that $[H(\lambda)]_{ll}$ is a non-negative number (cf. Eq. (4.19) in the appendix). Thus, we take the absolute value in the above derivations.

This algorithm exhibits some desirable properties: (a) *fast convergence*: in Theorem 4.3.1 we show that, under some mild conditions, our algorithm converges super-linearly to the optimum point, which is an improvement over first-order projected gradient methods [8]; (b) *low computational cost*: computing the inverse of a diagonal matrix is easier than the inverse of Hessian and greatly reduces the computational cost of the Newton's method; (c) *low communication overhead*: this algorithm leverages only local information to compute the second-order partial derivatives; in Equation 4.10, each sensing node (installed at a transformer) only requires to track the previous loading level and λ.

Our decentralized EV charging algorithm is based on SGPA. Hence, it updates each Lagrangian multiplier and send this *congestion signal* to downstream charging stations. Algorithm 1, which is run at each sensing node (corresponding to a transformer) in the distribution network, describes the update that happens once in every iteration.

```
Algorithm 1: Algorithm for Transformer $l$
    input  : the rating and measured load of transformer $l$
    output: $\{\lambda_l^{(k+1)}\}$ which will be sent to downstream EV chargers
    $\frac{\partial g}{\partial \lambda_l} \approx \text{rating}_l - \text{load}_l$
    $\frac{\partial^2 g}{\partial \lambda_l^2} \approx \frac{\Delta load}{\Delta \lambda_l}$
    $\lambda_l^{(k+1)} = \max\left\{\lambda_l^{(k)} - \gamma\left(\frac{\partial^2 g}{\partial \lambda_l^2}\right)^{-1}\frac{\partial g}{\partial \lambda_l}, 0\right\}$
```

When a charging station receives the Lagrangian multipliers from the substation and the distribution transformer supplying its demand, it computes its charge power:

$$x_e^{(k+1)} = \min\left\{\max\left\{\frac{w_e}{\sum_l T_{el}\lambda_l^{(k)}}, 0\right\}, M_e\right\} \tag{4.11}$$

and begins charging the connected EV with a power that equals $x_e^{(k+1)}$. Algorithm 2 updates the charge power of each EV.

```
Algorithm 2: Algorithm for EV charging station $e$
    input  : $w_e$ and a sequence of $\{\lambda_l^{(k)}\}$ received from transformers
             supplying $e$ $(l : T_{el} = 1)$.
    output: the charge power of charger $e$: $x_e^{(k+1)}$
    $x_e^{(k+1)} = \min\left\{\max\left\{\frac{w_e}{\sum_{l:T_{el}=1}\lambda_l^{(k)}}, 0\right\}, M_e\right\}$
```

The change in the charge power of charging stations affects the loading of transformers, thereby enabling them to update the Lagrangian multiplier in every iteration. We assume that τ iterations take place in one time slot, i.e., in $[t, t+1)$, and the time between two consecutive iterations is on the order of hundred milliseconds.

4.3 Proof of Quadratic Convergence

One of the strongest contributions of this thesis is to prove that SGPA converges to optimum point with super-linear rate of convergence. In this section we first present some preliminary propositions about vector and matrix norms and projections. Then we present the proof of quadratic rate of convergence of SGPA. Finally, we derive the appropriate values of some parameters of SGPA for the distribution network.

4.3.1 Vector and Matrix Norms

Let x be a d dimensional vector, A be a symmetric $d \times d$ matrix, A' be its transpose and $\|.\|_k$ be a vector norm (or an induced norm for matrices). From Reference [12] we have these propositions:

1. $\|x\|_1 \leq \sqrt{d}\,\|x\|_2 \leq \sqrt{d}\,\|x\|_1$
2. $\|Ax\| \leq \|A\|\,\|x\|$
3. $\|AB\| \leq \|A\|\,\|B\|$ [Let B be a $d \times d$ matrix]
4. $\|A\|_1 \leq \sqrt{d}\,\|A\|_2$
5. $\|A\|_\infty = \|A'\|_1 = \|A\|_1$
6. $\|A\|_2^2 \leq \|A\|_\infty\,\|A\|_1$
7. $\|A\|_1 \leq \sqrt{d}\,\|A\|_2 \leq \sqrt{d}\,\|A\|_1$

Proposition 7 is derived from Propositions 4, 5, and 6.

4.3.2 Projections

In (4.9), we use a matrix projection which can be written as:

$$[A]^\eta = \arg\min_{D \in \mathcal{D}} \|A - D\|_2$$

where $\mathcal{D}$ is the set of diagonal matrices whose diagonal components are greater than or equal to a small positive number η. More precisely:

$$\mathcal{D} = \left\{ D \in \mathbb{R}^{d \times d} | D \text{ is diagonal and, } [D]_{ll} \geq \eta > 0, \forall l \right\}$$

It is easy to verify that $[A]^\eta$ is invertible, $\left[[A]^\eta\right]^{-1} \leq \frac{1}{\eta}$, and $\mathcal{D}$ is a nonempty and closed convex set. Hence, this projection is non-expansive [12]Page 211, Proposition 3.2.:(c):

$$\begin{aligned}
\left\|[A]^\eta - [B]^\eta\right\|_2 &\leq \|A - B\|_2 \implies \\
\sqrt{d}\left\|[A]^\eta - [B]^\eta\right\|_2 &\leq \sqrt{d}\,\|A - B\|_2 \implies \\
\left\|[A]^\eta - [B]^\eta\right\|_1 &\leq \sqrt{d}\,\|A - B\|_1
\end{aligned} \tag{4.12}$$

Similarly in (4.8), we use a projection that can be written as:

$$[\lambda]^+ = \arg\min_{\mu \in \mathcal{C}} \|\lambda - \mu\|_2$$

where $\mathcal{C} = \left\{\mu | \mu \in \mathbb{R}^d, \mu \succeq 0\right\}$. This projection is also non-expansive. Thus, we have:

$$\begin{aligned}
\left\|[\lambda]^+ - [\mu]^+\right\|_2 &\leq \|\lambda - \mu\|_2 \implies \\
\left\|[\lambda]^+ - [\mu]^+\right\|_1 &\leq \sqrt{d}\,\|\lambda - \mu\|_1
\end{aligned} \tag{4.13}$$

4.3.3 Quadratic Convergence of SGPA

Theorem 4.3.1. *Let $g(\lambda)$ be a twice continuously differentiable function in the domain $\mathcal{C} = \left\{\lambda \in \mathbb{R}^d | \lambda \succeq 0\right\}$, $\lambda^* \in \mathcal{C}$ be a point such that $\nabla g(\lambda^*) = 0$, $H(\lambda)$ be the Hessian matrix, $D(\lambda) = [H(\lambda)]^\eta$ be a diagonal matrix, and the following assumptions hold:*

(i) the Hessian matrix is diagonal and invertible at λ^*, i.e., $H(\lambda^*) = [H(\lambda^*)]^\eta$ for some small η;

(ii) there exist constants $L > 0$, $\theta > 0$ such that $\forall \lambda, \mu \in \mathcal{D}$ we have $\|H(\lambda) - H(\mu)\|_1 \leq L\,\|\lambda - \mu\|_1$ if $\|\lambda - \mu\|_1 \leq \theta$.

Now if $\lambda^{(k+1)} = \left[\lambda^{(k)} - \gamma D\left(\lambda^{(k)}\right)^{-1} \nabla g\left(\lambda^{(k)}\right)\right]^+$ and $\gamma \to 1$ then we have the following:

1. $\left\|\lambda^{(k+1)} - \lambda^*\right\|_1 \leq \frac{L}{\eta}\left(\frac{\gamma\sqrt{d}}{2} + d\right)\left\|\lambda^{(k)} - \lambda^*\right\|_1^2$

2. $\left\|\lambda^{(k+1)} - \lambda^*\right\| < \left\|\lambda^{(k)} - \lambda^*\right\|$, if $\left\|\lambda^{(k)} - \lambda^*\right\|_1 < \frac{\eta}{\left(\frac{\gamma\sqrt{d}}{2} + d\right)L}$

Proof. The proof idea is similar to [21]. For convenience we use the following notation: $\lambda := \lambda^{(k)}$, $H_t := H(\lambda^* + t(\lambda - \lambda^*))$, $H^* := H(\lambda^*)$, $D := D(\lambda)$, $D^* := D(\lambda^*)$, $\nabla g := \nabla g(\lambda)$ and $\nabla g^* := \nabla g(\lambda^*)$. Based on (4.13) and (4.12) we have the following derivation:

$$\begin{aligned}
&\left\|\gamma H_t - D\right\|_1 = \left\|\gamma H_t - \gamma H^* + \gamma H^* - D\right\|_1 \\
&\leq \gamma \left\|H_t - H^*\right\|_1 + \left\|\gamma H^* - D\right\|_1 \\
&\leq \gamma \left\|H_t - H^*\right\|_1 + \left\|H^* - D\right\|_1 + (1-\gamma)\left\|H^*\right\|_1 \\
&\approx \gamma \left\|H_t - H^*\right\|_1 + \left\|H^* - D\right\|_1 \qquad\qquad \text{as } \gamma \to 1 \\
&= \gamma \left\|H_t - H^*\right\|_1 + \left\|[H^*]^\eta - [H]^\eta\right\|_1 \\
&\leq \gamma \left\|H_t - H^*\right\|_1 + \sqrt{d}\left\|H^* - H\right\|_1 \\
&\leq \gamma L \left\|t(\lambda - \lambda^*)\right\|_1 + \sqrt{d} L \left\|\lambda - \lambda^*\right\|_1 \\
&= \left(\gamma t + \sqrt{d}\right) L . \left\|\lambda - \lambda^*\right\|_1
\end{aligned} \tag{4.14}$$

Let $\phi(t) = \nabla g\left(\lambda^* + t\left(\lambda - \lambda^*\right)\right)$. Then $\phi'(t) = H_t.(\lambda - \lambda^*)$ and

$$\nabla g - \nabla g^* = \phi(1) - \phi(0) = \int_0^1 H_t.(\lambda - \lambda^*) dt \tag{4.15}$$

Based on the propositions and equations, we have the following derivation:

$$\begin{aligned}
&\left\|\lambda^{(k+1)} - \lambda^*\right\|_1 \leq \sqrt{d}\left\|\lambda^{(k+1)} - \lambda^*\right\|_2 \\
&= \sqrt{d}\left\|\lambda^{(k+1)} - [\lambda^*]^+\right\|_2 \qquad\qquad \text{since } [\lambda^*]^+ = \lambda^* \\
&= \sqrt{d}\left\|\left[\lambda - \gamma D^{-1}\nabla g\right]^+ - [\lambda^*]^+\right\|_2 \\
&\leq \sqrt{d}\left\|\lambda - \gamma D^{-1}\nabla g - \lambda^*\right\|_2 \\
&= \sqrt{d}\left\|\lambda - \lambda^* - \gamma D^{-1}\left(\nabla g - \nabla g^*\right)\right\|_2 \\
&= \sqrt{d}\left\|\lambda - \lambda^* - D^{-1}\int_0^1 \gamma H_t.(\lambda - \lambda^*) dt\right\|_2 \\
&= \sqrt{d}\left\|D^{-1}\left(\int_0^1 (\gamma H_t - D).(\lambda - \lambda^*) dt\right)\right\|_2 \\
&\leq \sqrt{d}\left\|D^{-1}\left(\int_0^1 (\gamma H_t - D).(\lambda - \lambda^*) dt)\right)\right\|_1 \\
&\leq \sqrt{d}\left\|D^{-1}\right\| \int_0^1 \left\|\gamma H_t - D\right\|_1 \left\|\lambda - \lambda^*\right\|_1 dt
\end{aligned}$$

$$
\begin{aligned}
&= \sqrt{d}\left\|D^{-1}\right\| \left\|\lambda-\lambda^*\right\|_1 \int_0^1 \left\|\gamma H_t - D\right\|_1 dt \\
&\leq \sqrt{d}\left\|D^{-1}\right\| \left\|\lambda-\lambda^*\right\|_1 \int_0^1 \left(\gamma t + \sqrt{d}\right) L. \left\|\lambda-\lambda^*\right\|_1 dt \\
&= \sqrt{d}\left\|D^{-1}\right\| \left\|\lambda-\lambda^*\right\|_1^2 \int_0^1 \left(\gamma t + \sqrt{d}\right) L dt \\
&= \left\|D^{-1}\right\| \left\|\lambda-\lambda^*\right\|_1^2 \left(\frac{\gamma\sqrt{d}}{2} + d\right) L \\
&\leq \left(\frac{\gamma\sqrt{d}}{2} + d\right) \frac{L}{\eta} \left\|\lambda-\lambda^*\right\|_1^2 && (4.16) \\
&< \left\|\lambda-\lambda^*\right\|_1 && (4.17)
\end{aligned}
$$

Note that (4.16) and (4.17) prove the first and second parts of the theorem, respectively. □

4.3.4 Values of d and L

Here d is the dimension of λ or equivalently the number of transformers in the system. As we use the IEEE 33-bus system, $d = 33$ in our test system. To find the value of L, we first list the following definitions and propositions:

$$
\begin{aligned}
&g_k = \frac{\partial g}{\partial \lambda_k} = a_k - \sum_e T_{ek} x_e^* && (4.18) \\
&g_{kl} = \frac{\partial^2 g}{\partial \lambda_k \partial \lambda_l} = -\sum_e T_{ek} \frac{\partial x_e^*}{\partial \lambda_l} = \sum_e T_{ek} T_{el} \frac{(x_e^*)^2}{w_e} && (4.19) \\
&\nabla g_k = (g_{k1}, g_{k2}, \ldots, g_{kd}) && (4.20) \\
&g_{kll'} = \frac{\partial^3 g}{\partial \lambda_k \partial \lambda_l \partial \lambda_l'} = -\sum_e T_{ek} T_{el} T_{el'} \frac{2(x_e^*)^3}{w_e^2} && (4.21) \\
&M = \max_e \{M_e\} \qquad w = \min_e \{w_e\} && (4.22) \\
&\widetilde{L} = \max_e \sum_{l'} T_{el'} \qquad \widetilde{E} = \max_l \sum_e T_{el} && (4.23)
\end{aligned}
$$

Note that, the charge power, x_e (so the dual objective function, g) is not rigorously differentiable for all, $\lambda \succeq 0$. It is because of the projection we use on x_e (4.11) changes abruptly at $x_e = M_e$. We can approximate this sharp change by a smooth function. In that case, x_e and g will become differentiable at all points and (4.18), (4.19) and (4.21) will hold. Also from (4.11), g_{kl} and $g_{kll'}$ will be zero, if $\frac{w_e}{\sum_l T_{el}\lambda_l} > M_e$.

Now we show that ∇g_k is Lipschitz continuous (i.e. $\left\|\nabla g_k(\lambda) - \nabla g_k(\mu)\right\|_1 \leq L \left\|\lambda - \mu\right\|_1$) by showing that the corresponding Hessian, $[\mathcal{H}_k]_{ll'} = g_{kll'}$, is bounded, i.e., $\left\|\mathcal{H}_k\right\|_1 \leq L$:

$$\begin{aligned}
&\left\|\mathcal{H}_k\right\|_1 = \max_l \sum_{l'} |g_{kll'}| \leq \sum_e T_{ek} T_{el} T_{el'} \frac{2(x_e^*)^3}{w_e^2} \\
&\leq \frac{2}{w^2} \sum_e T_{el} T_{el'} (x_e^*)^3 && \text{since } T_{ek} \leq 1 \\
&\leq \frac{2M^3}{w^2} \max_l \sum_{l'} \sum_e T_{el} T_{el'} \\
&\leq \frac{2M^3}{w^2} \max_l \sum_e T_{el} \widetilde{L} \leq \frac{2M^3}{w^2} \widetilde{L}\widetilde{E} \implies \\
&\left\|\nabla g_k(\lambda) - \nabla g_k(\mu)\right\|_1 \leq \frac{2M^3}{w^2} \widetilde{L}\widetilde{E} \left\|\lambda - \mu\right\|_1
\end{aligned} \tag{4.24}$$

We have

$$\begin{aligned}
&\left\|H(\lambda) - H(\mu)\right\|_1 = \max_m \sum_l |g_{lm}(\lambda) - g_{lm}(\mu)| \\
&= \sum_l |g_{lk}(\lambda) - g_{lk}(\mu)| && \text{let, } k \text{ be the argmax} \\
&= \left\|\nabla g_k(\lambda) - \nabla g_k(\mu)\right\|_1 \\
&\leq \frac{2M^3 \widetilde{L}\widetilde{E}}{w^2} \left\|\lambda - \mu\right\|_1 \\
\implies & L = \frac{2M^3 \widetilde{L}\widetilde{E}}{w^2}
\end{aligned}$$

Hence in our test network, the parameters should be set as follows:

- $\gamma \to 1$
- $L = \frac{2M^3 \widetilde{L}\widetilde{E}}{w^2}$
- $d = 33$

4.3.5 Proof of Descent Direction Update

The projection in (4.8) can be written as

$$\lambda^{(k+1)} = \lambda^{(k)} - \gamma S D^{-1} \nabla g$$

Here, S is a diagonal matrix defined as follows:

$$S_{ii} = \begin{cases} 1, & \text{if } \gamma[D^{-1}\nabla g]_i \leq \lambda_i^{(k)} \\ \frac{\lambda_i^{(k)}}{\gamma[D^{-1}\nabla g]_i}, & \text{if } \gamma[D^{-1}\nabla g]_i > \lambda_i^{(k)} \end{cases} \tag{4.25}$$

This matrix scales each component of the vector, $\gamma D^{-1}\nabla g$, to project the difference (i.e., $\lambda^{(k)} - \gamma D^{-1}\nabla g$) on the positive orthant. Since SD^{-1} is a non-negative diagonal matrix, the direction, $-\gamma SD^{-1}\nabla g$, is a descent direction.

4.3.6 Convergence Rate Analysis

Definition: We say that a sequence of real vectors $\{v_k\}$ converging to v^*, has a super-linear rate of convergence (under a vector norm, $\|.\|$) if

$$\lim_{k\to\infty} \frac{\|v_{k+1} - v^*\|}{\|v_k - v^*\|} = 0$$

From (4.17) it is evident that, near λ^*, the sequence, $\left\{\lambda^{(k)}\right\}$ converges to λ^* (i.e., $\lim_{k\to\infty} \left\|\lambda^{(k)} - \lambda^*\right\|_1 \to 0$). From (4.16) we have:

$$\frac{\left\|\lambda^{(k+1)} - \lambda^*\right\|_1}{\left\|\lambda^{(k)} - \lambda^*\right\|_1} \leq \left(\frac{\gamma\sqrt{d}}{2} + d\right)\frac{L}{\eta}\left\|\lambda^{(k)} - \lambda^*\right\|_1$$

$$\implies \lim_{k\to\infty} \frac{\left\|\lambda^{(k+1)} - \lambda^*\right\|_1}{\left\|\lambda^{(k)} - \lambda^*\right\|_1} = 0$$

Moreover, the rate of convergence of a sequence is quadratic if

$$\lim_{k\to\infty} \frac{\|v_{k+1} - v^*\|}{\|v_k - v^*\|^2} < C$$

for some positive constant C. From (4.16) it is easy to show that this condition holds for the sequence, $\left\{\lambda^{(k)}\right\}$, and hence it converges quadratically to λ^*.

4.4 Baseline Algorithms

To evaluate our proposed scaled gradient projection algorithm, we consider the following baseline algorithms:

Gradient Projection Algorithm (GPA): this is the decentralized algorithm outlined in Section 4.1.2. It is basically updating the Lagrangian multiplier using the projected gradient descent algorithm:

$$\lambda_l^{(k+1)} = \left[\lambda_l^{(k)} - \gamma\frac{\partial g}{\partial \lambda_l}\right]^+ \tag{4.26}$$

The gradient $\frac{\partial g}{\partial \lambda_l} = \zeta a_l - \sum_e T_{el} x_e^*$ can be calculated by simply subtracting the transformer loading (including the demands of elastic and inelastic loads) from its rated capacity as shown in [8]. Thus, we do not need to separately measure the demand of each EV charger.

Earliest Deadline First (EDF): it simply priorities charging EVs based on their deadlines; thus, EVs with earliest deadline are charged at the maximum rate supported by their charging station first. To fully utilize the network, we charge as many EVs as possible at any given time without overloading the transformers.

Least Laxity First (LLF): it is similar to EDF except that charging EVs is prioritized based on their laxity. EVs with the lowest laxity are charged at the maximum rate supported by the charging station first. At any point int time, we charge as many EVs as possible until the network is congested.

Chapter 5

Experimental Results for Model-based Control Strategies

In this chapter, we analyze performance of the model-based control strategies that are proposed in Chapter 4. Section 5.1 describes the testbed and performance metrics for the simulations. Section 5.2 presents the simulations results for different user types and simulation scenarios.

5.1 Case Studies

We implement the proposed EV charging mechanism and baselines introduced in Section 4.4 in Python and use the simulation platform depicted in Figure 5.1 to compare their performance in four different simulation scenarios. The platform takes as input the structure of the distribution network, including points of connection of homes and EV charging stations, the household demands, the EV arrival and departure times, and the initial state-of-charge (SoC) of their batteries. To build the household demands, we take advantage of the real and reactive power consumption of several real homes. We use this data along with the charge powers of EVs determined by the algorithm to perform power flow analysis and obtain the loading levels of transformers. The sensor emulator module subsequently adds Gaussian noise to these values to imitate the measurement noise of sensors in real-world applications. These measurements are used to construct the congestion signal which is then sent to downstream EV charging stations via communication links. These signals are used by the con-

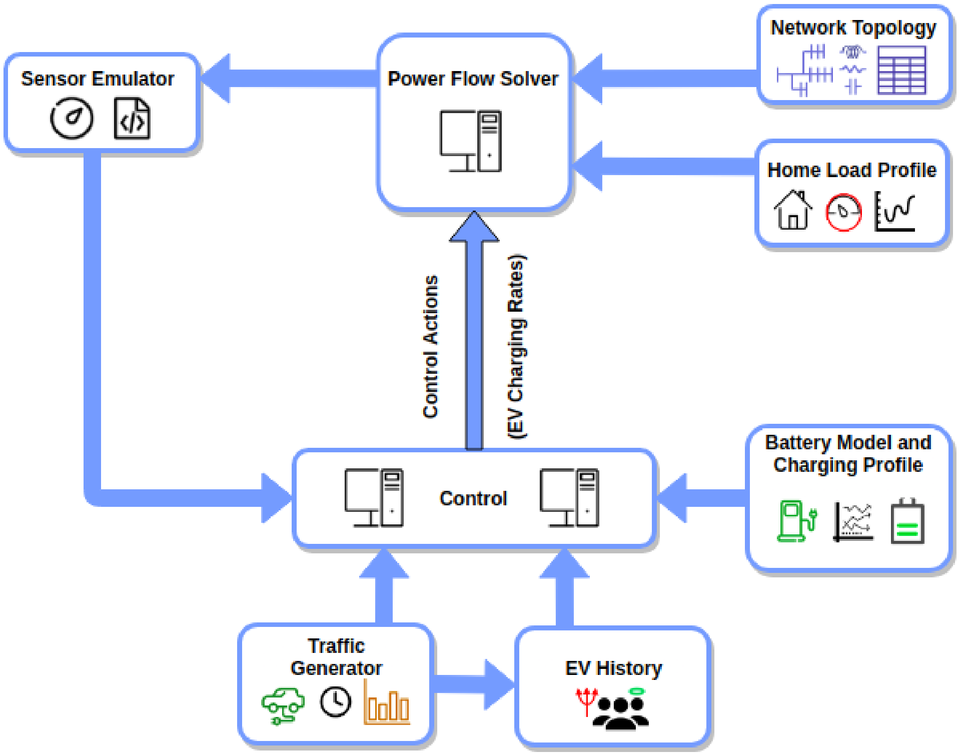

Figure 5.1: Overview of the simulation framework.

trol algorithm to update the charge power of EV charging stations in a single iteration.

The control algorithm needs to know which EVs will be connected to chargers in the next time step along with their demands and claimed deadlines. This data is produced by the traffic generator. We also store historical data about EVs to update their reputation throughout the simulation. Besides this data, the simulation platform adopts a battery model which expresses the capacity, and charge and discharge inefficiencies of the battery. This model is necessary to update the state-of-charge of the battery given its charge power.

5.1.1 Test System

We consider the 33-bus test system [11] shown in Figure 3.1 (Chapter 3) as the primary distribution network in our simulations. For modelling the secondary (low-voltage) distribution network, we assume that a feeder originates from every bus in the primary network (except Node 1). We adopt the IEEE European test feeder [27] as the secondary feeder. Figure 3.1 (Chapter 3) illustrates how a low-voltage network is connected to Node 25. We assume that a sensor, e.g., a distribution-level phasor measurement unit, is deployed at each of the transformer locations measuring the active and reactive powers in real time, and that there are communication links connecting the sensor at the substation to the sensors at distribution transformer locations, and these sensors to downstream charging stations.

There are 55 single-phase load connection points in each secondary feeder. Therefore, we have a total of 1760 (32×55) load connection points in the system. We assume a certain number of homes are connected to these points. To represent the household demand, we utilize sample household consumption data provided in the ADRES-CONCEPT project [2][1]. The data set contains real and reactive power consumption of 30 households with 1-second resolution over two weeks. Since our simulation spans one day, we split the data into 1-day segments. This yields 420 daily load profiles. We add white Gaussian noise with 10% standard deviation to each sample to increase the size of the data set; this way we create a total of 8400 unique household load data. We consider the active power consumption at the primary nodes provided in [11] to determine the appropriate level of aggregation at each low-voltage node (i.e., the houses connected to the same low-voltage node). We select houses randomly and connect them to a secondary node (i.e., a single-phase load connection point) until the sum of all loads in the low-voltage network under each primary node matches the load given in the specification of the 33-bus system.

To have a realistic EV schedule (i.e., arrival and departure times of the EVs), we exploit the EV usage data provided in the Pecan Street data set [53]. The data set contains the power usage of many home appliances with up to 1-second resolution. We use the data recorded in 2016. Since there are only about 80 EVs in this data set, we take the daily arrival times of these EVs over one year and fit a Gaussian Mixture Model (GMM) to the probability

[1] The data was generated in the research project "ADRES-Concept" (EZ-IF: Development of concepts for ADRES- Autonomous Decentralized Regenerative Energy Systems, project no. 815 674). This project was funded by the Austrian Climate and Energy Fund and performed under the program "ENERGIE DER ZUKUNFT".

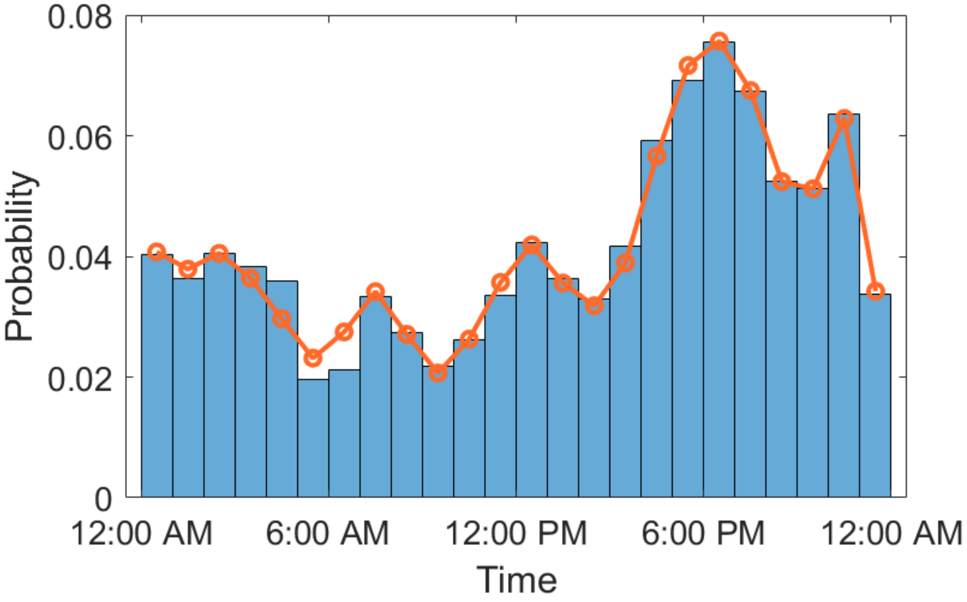

Figure 5.2: Distribution of arrival times.

distribution function (pdf). The distribution of arrival times and the fitted model with 7 Gaussian distributions are shown in Figure 5.2.

We assume that there are 500 EV chargers installed at random locations in our secondary distribution network. We sample the daily arrival times of the EVs from the fitted GMM. Since the Pecan Street data set does not provide the EV departure/disconnection times, we randomly sample the stay time of each EV from a Gaussian distribution with $\mu = 8$ and $\sigma = 2$ (in hours). The departure time is the sum of the arrival time and the stay time. The initial SoC of each EV when it arrives is sampled uniformly between 0 and 0.1. We consider 4 different battery sizes that belong to popular EVs in the market: a) $16kWh$ for Chevy Volt and Mitsubishi iMiEV, b) $30kWh$ for Nissan Leaf, c) $42kWh$ for BMW i3, and d) $75kWh$ for Tesla 3. For an EV, the battery size is selected randomly from the above types. We set the maximum charging rate to $\widetilde{M}_e = 7kW$ for all charging stations.

For SGPA and GPA, we define the step-size to be $\gamma = 0.008$ and 0.0008, respectively. We tune these values based on simulations, using the bounds suggested by our theoretical results as a hint. Furthermore, we assume 100 iterations in each time slot, which is 10 minutes long. Hence, the time between two iterations is 6 seconds.

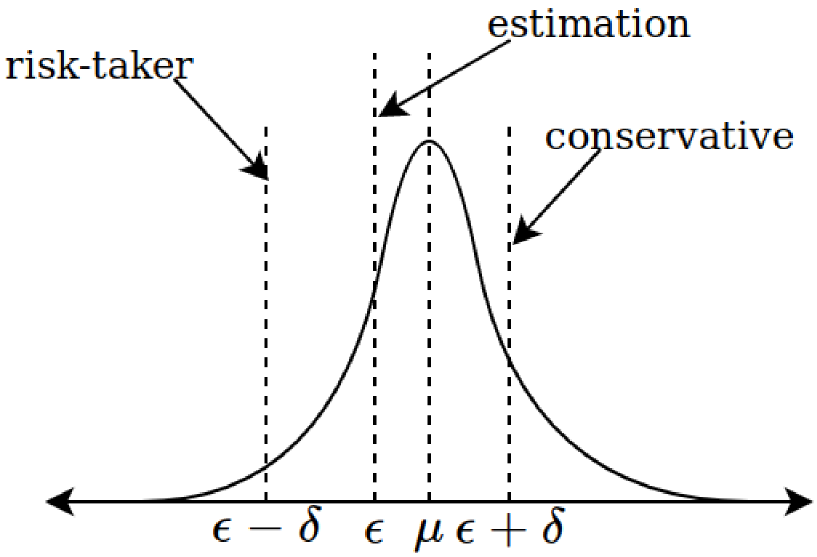

Figure 5.3: Predicting and adjusting deadlines. μ and ϵ represent the actual and predicted deadlines respectively. Conservative (risk-taking) users add (subtract) δ to (from) their prediction.

5.1.2 Performance Metrics

Jain index: is used to evaluate fairness of a power allocation scheme (a higher value suggests a fairer scheme):

$$J(x) = \frac{\left(\sum_i x_i\right)^2}{n \sum_i x_i^2}$$

where x is a vector containing charge powers of all chargers in a given time slot and n is the number of active chargers. Note that the Jain index is calculated for every time slot and then averaged over all time slots in one day.

Percentage of EVs with a certain SoC: it is the percentage of connected EVs with a SoC greater than or equal a certain threshold when they depart. We consider 90% as the threshold for Scenarios A, C and D (described below). For Scenario B, it is 80%.

Energy supplied above the rated capacity: it is the total amount of energy (in kWh) supplied by a transformer above its rated capacity for all the time slots in one day.

5.1.3 User Types

To study the efficacy of our charging mechanism, we consider conservative and risk-taking users. Both types estimate their duration of stay based on their information, but report their estimation after a slight modification. This modification is necessary as they may not know precisely when their EV will be used again. In particular, *conservative* users add a positive offset to their estimation so that they reduce the chance of declaring a deadline before their actual deadline when their estimation is not precise. This helps this type of users to maintain a better reputation in the system. Unlike conservative users, *risk-taking* users subtract a non-negative offset from their estimation, running the risk of ruining their reputation if the actual deadline is after their declared deadline.

Figure 5.3 presents the difference between these user types. We consider historical discrepancy data for the past 3 days and take their average to compute Δ_e. To obtain each discrepancy, we have to model how EV drivers predict their deadline. We sample their prediction from a Gaussian distribution which has the same mean as the true deadline and $\sigma = 0.5$. To simulate conservative (risk-taking) users, we add (subtract) a positive number, δ, to (from) the predicted deadline, where δ is sampled from a Gaussian distribution with $\mu = 3$ and $\sigma = 0.5$ (in hours). This gives us the user-specified deadline.

Based on these two user types, we develop four scenarios: (a) *Scenario A*: both types of EV drivers are present in the system (50% of population each) and all the other parameters (arrival times and locations) are sampled as described earlier, (b) *Scenario B*: is similar to Scenario A except that the arrival times and connection points of EVs are chosen in such a way that the first transformer is congested, (c) *Scenario C*: only conservative users are in the system, and (d) *Scenario D*: only risk-taking users are in the system. In the first two scenarios, we assume that risk-taking users have lower reputation compared to the conservative users when the simulation starts.

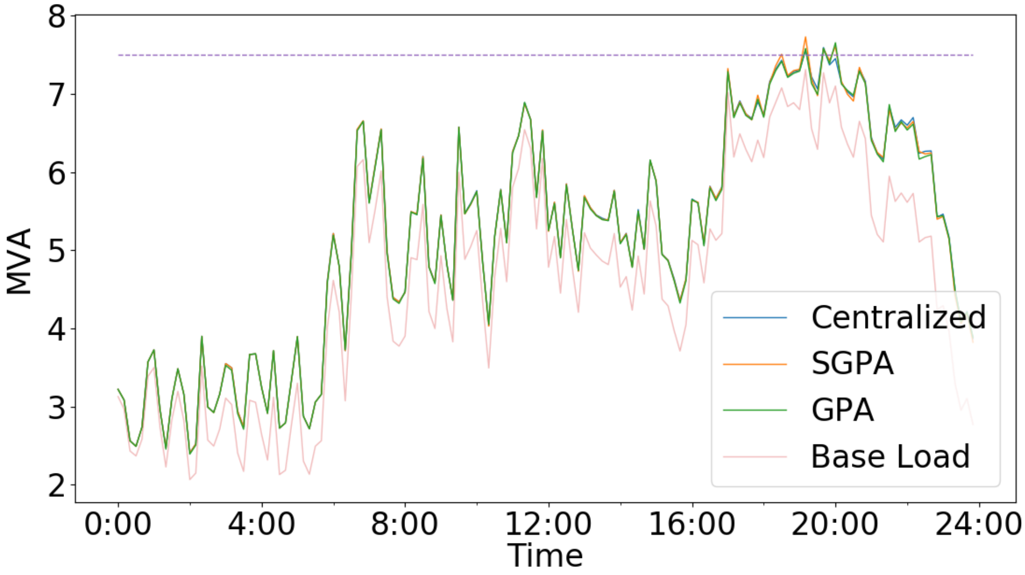

Figure 5.4: Substation loading at different time slots for centralized, GPA and SGPA. The horizontal (purple) dashed line represents the rating of the substation transformer.

5.2 Results

This section presents the simulation results for the four scenarios. It also presents the analysis of the rate of convergence of SGPA.

5.2.1 Scenario A

Figure 5.4 shows the substation transformer's loading for centralized, GPA, SGPA, and baseline algorithms. The congestion period is from 6:00pm to 9:00pm. It can be readily seen that, for most of the time, GPA and SGPA follow the centralized solution. Note that LLF and EDF policies are quite similar to the centralized solution so we do not show them in this figure.

Figure 5.5 compares the performance of different algorithms considering the first two performance metrics presented in Section 5.1.2. The centralized algorithm outperforms least-laxity-first and earliest-deadline-first both in terms of fairness and percentage of EVs with ≥ 90% SoC for both conservative and risk-taking users. Moreover, the performance gap between two types of users for the centralized algorithm is higher than baseline algorithms. This indicates that our centralized mechanism can better differentiate between the two types of users than baseline algorithms. Both of the distributed algorithms follow the

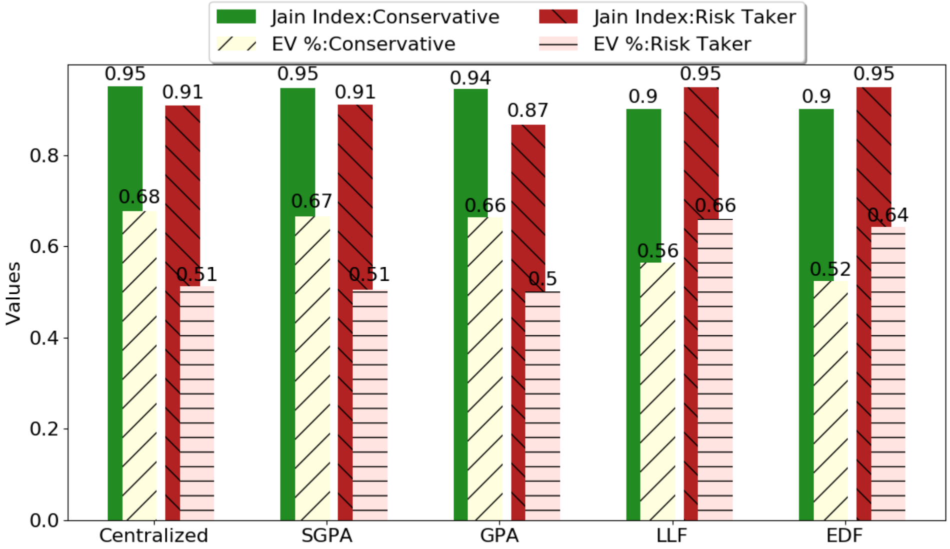

Figure 5.5: Comparing performance of different algorithms for conservative and risk-taking users (Scenario A).

power allocation of the centralized algorithm most of the time. That said, our mechanism (SGPA) follows the centralized solution more closely than GPA. We attribute this to the faster convergence of this algorithm.

5.2.2 Scenario B

To investigate the performance of our proposed methods during congestion, we study a scenario in which 54 EVs (27 conservative and 27 risk-taking) arrive at same time (i.e., 9:00am) to charging stations that are fed by the same distribution transformer (i.e., the first distribution transformer). The initial SoC (5%) and claimed charging duration (8 hours) are the same for all EVs. This ensures that the distribution transformer is congested for a relatively long period of time. However, conservative and risk-taking users stay in the system for 7 and 9 hours, respectively (hence, the discrepancy is 1 hour). Figure 5.6 and 5.7 depict the transformer loading for different algorithms during the congestion period. Both SGPA and GPA (as well as LLF and EDF) follow the centralized algorithm, though they exhibit some fluctuations.

Figure 5.8 shows the performance of all the algorithms in this scenario. The centralized algorithm, just as SGPA and GPA, assigns a higher priority to conservative users because of their better reputation. In fact for the centralized

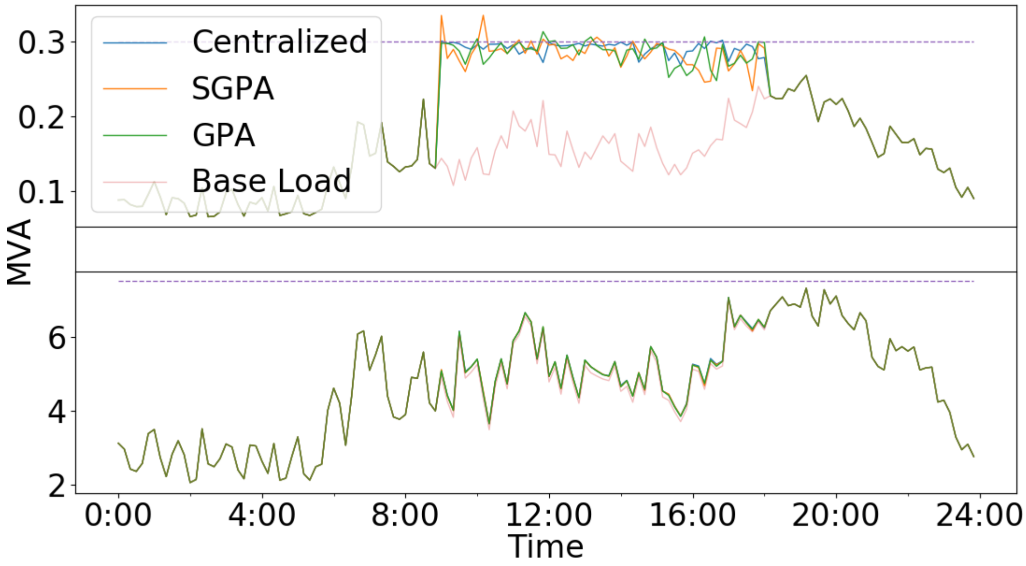

Figure 5.6: Transformer loading over one day using centralized, SGPA and GPA. The top plot is for a distribution transformer and the bottom one is for the substation transformer.

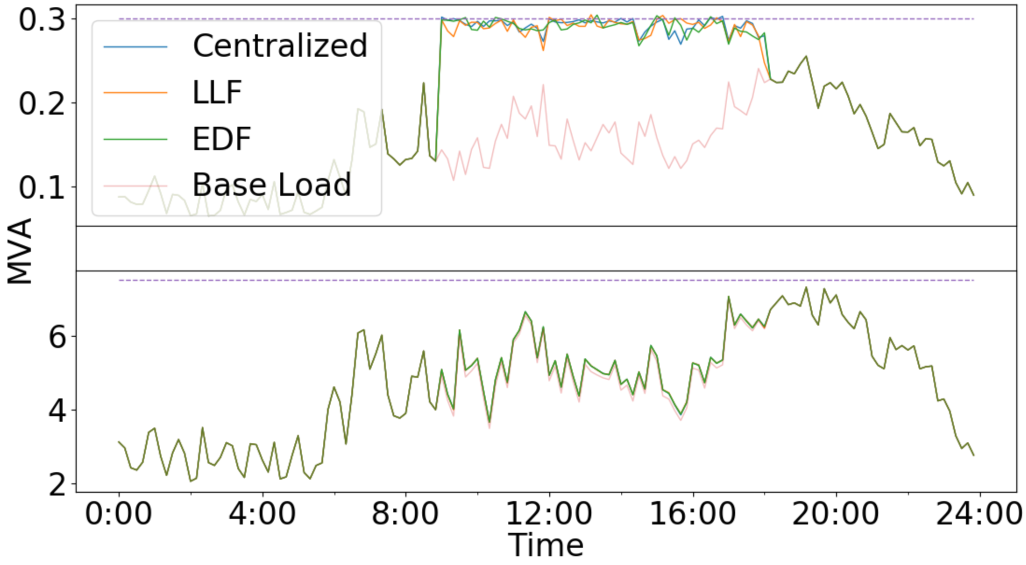

Figure 5.7: Transformer loading over one day using centralized, LLF and EDF. The top plot is for a distribution transformer and the bottom one is for the substation transformer.

algorithm, all the EVs are charged up to 80% of their capacity by the end of the simulation. However, LLF and EDF algorithms provide more power to risk-taking users than our proposed algorithms as they neglect their reputation.

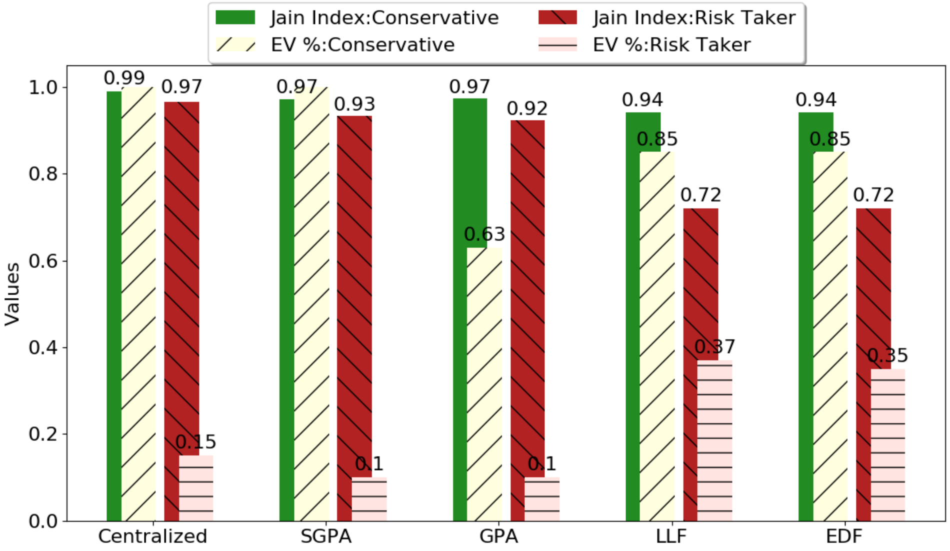

Figure 5.8: Comparing performance of different algorithms for conservative and risk-taking users (Scenario B).

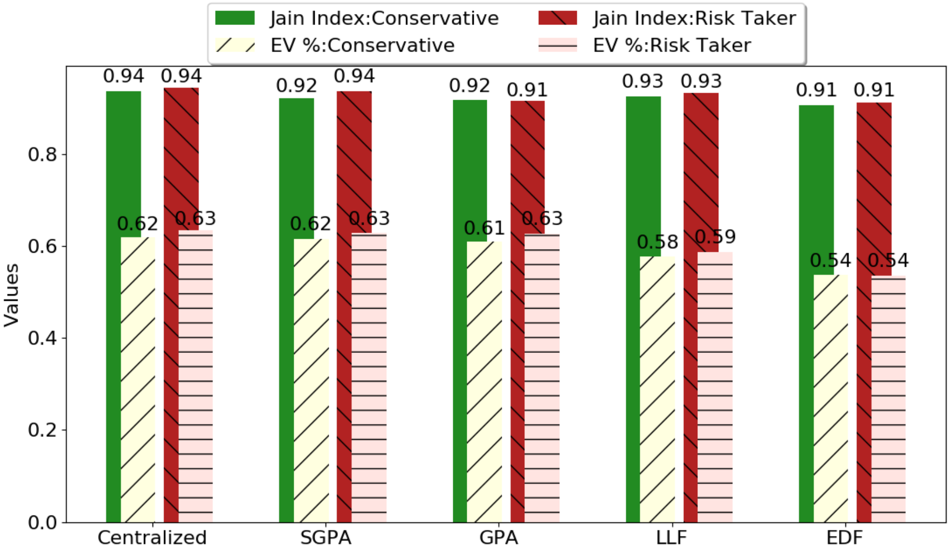

Figure 5.9: Comparing performance of different algorithms for conservative and risk-taking users (Scenarios C and D).

5.2.3 Scenario C & D

Figure 5.9 compares the performance of different algorithms in Scenarios C and D, which consist of only conservative and risk-taking users, respectively. As expected, when all the users belong to the same category, our mechanism does not differentiate between them. Here, the difference in performance between the two types is merely due to the different laxity values they have.

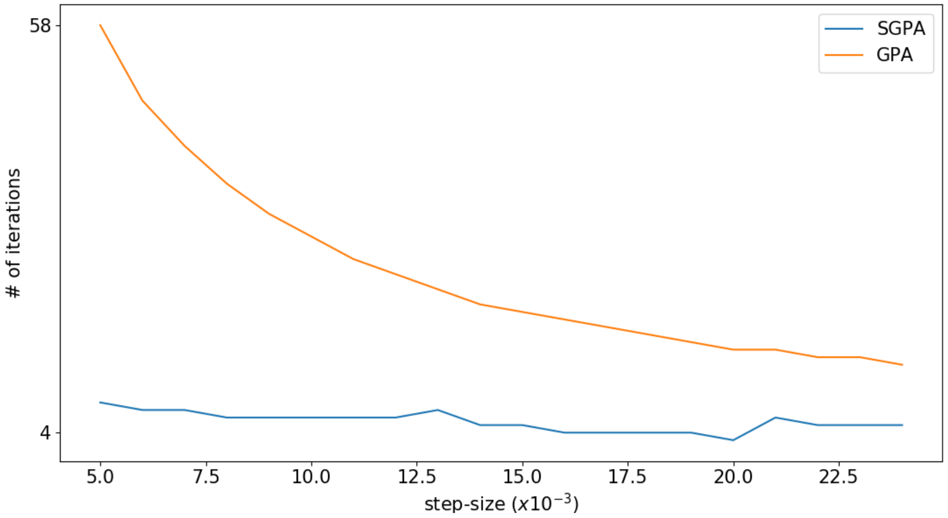

Figure 5.10: 95% convergence analysis of SGPA and GPA for a specific time slot (7:00 pm) for scenario A. Initial points for both algorithms are the same.

5.2.4 Analyzing the Rate of Convergence

Figure 5.10 compares the rate of convergence of SGPA and GPA. This analysis is done for a time slot (7:00pm) in Scenario A when the network is congested. Here, convergence is defined as the event when the total power allocated by SGPA (or by GPA) reaches 95% of the total power allocated by the centralized algorithm. As the step-size becomes larger, GPA converges in fewer iterations. Interestingly, SGPA converges after less than 8 iterations, irrespective of the step size.

Table 5.1: Energy supplied above the rated capacity of a transformer. In Scenarios A, C and D, the values are calculated for the substation; In Scenario B, it is computed for the first distribution transformer.

	Energy supplied above the rated capacity	
	SGPA	GPA
Scenario A	70.58 kWh	52.81 kWh
Scenario B	5.78 kWh	2.28 kWh
Scenario C	58.50 kWh	46.96 kWh
Scenario D	93.51 kWh	48.14 kWh

5.2.5 Excursion from the Rated Capacity

When the network is congested, SGPA and GPA may overshoot the target, thereby loading the transformer above its rated capacity temporarily. Table 5.1

shows the total energy (in kWh) supplied above the rated capacity of the substation transformer by centralized, SGPA and GPA for Scenarios A, C and D. For Scenario B, we examine the first distribution transformer which is expected to be overloaded. As expected, both SGPA and GPA overload the transformer for a short period of time; SGPA results in slightly higher overloading which we attribute to its higher responsiveness. In any case, this level of overloading is not problematic because transformers can be loaded above their rated capacity for a short period of time without being excessively overheated.

Chapter 6

Model-Free Control Strategies

In this chapter we present the model-free, reinforcement learning (RL) based control strategies that are applied in a distribution network and a parking station. Section 6.1 presents an overview of the RL-based methodologies. Section 6.2 describes the centralized control strategies for the distribution network and the parking station. These centralized methods require an approximate model of the system and we use them as baselines for the RL-based methods. Section 6.3 presents the Adaptive AIMD algorithm which is model-free and uses the reinforcement learning framework. Finally, Section 6.4 describes the baseline algorithms.

6.1 Methodology Overview

We propose an RL-based agent which learns the state-action mapping from interaction with the environment (i.e., the power system). The agent's action determines the parameter of the A-AIMD algorithm and therefore affects the charge power of the corresponding EV charging point.

As shown in Figure 6.1, RL agents are trained in two phases: offline direct policy learning and online interaction with environment. In the offline pretraining phase, the agent is initialized by imitating the centralized controller presented in the following. In the online (interactive) phase, the agent interacts with the environment and updates its policy using a policy gradient algorithm.

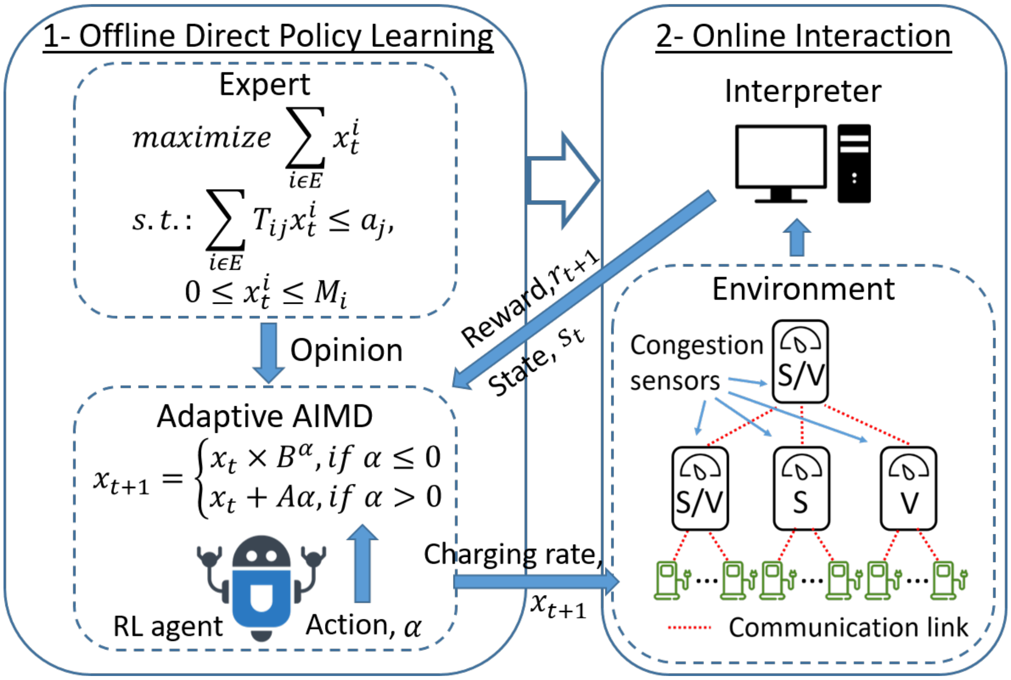
1- Offline Direct Policy Learning
Expert
$maximize \sum_{i\epsilon E} x_t^i$
$s.t.: \sum_{i\epsilon E} T_{ij} x_t^i \leq a_j,$
$0 \leq x_t^i \leq M_i$
Opinion
Adaptive AIMD
$x_{t+1} = \begin{cases} x_t \times B^{\alpha}, if\ \alpha \leq 0 \\ x_t + A\alpha, if\ \alpha > 0 \end{cases}$
RL agent
Action, α
Charging rate, x_{t+1}
Reward, r_{t+1}
State, s_t
2- Online Interaction
Interpreter
Environment
Congestion sensors
S/V
S/V
S
V
Communication link

Figure 6.1: An overview of the proposed methodology.

6.2 Centralized Controller

Controlling the charge power of EVs can be cast as a constrained optimization problem where the constraints largely depend on the structure and components of the power system (e.g., the distribution network or the parking station). In this section, we first present the centralized, optimization based control strategy for the distribution network. Then we present the centralized approach for the parking station described in ACN-Sim.

6.2.1 33-bus system

We use the optimization framework for controlled EV charging in a distribution network that was originally introduced in [8]. In this framework, an approximate model that ignores losses and reactive power flows is utilized to relate charge powers to real power flow. We solve this optimization problem in a centralized fashion for each time slot and use the solution as expert demonstration in offline training of the RL agents.

Let the available capacity (i.e., the rated capacity minus the current loading imposed by residential loads) of transformer j be a_j. Solving the following convex optimization problem yields the charge power, x_t^i, of every charger i for the interval $[t, t+1)$:

$$\text{maximize: } \sum_{i \in E} x_t^i \tag{6.1}$$

$$\text{subject to: } \sum_{i} T_{ij} x_t^i \leq a_j, \qquad \forall j \in \mathcal{L} \tag{6.2}$$

$$0 \leq x_t^i \leq M^i, \qquad \forall i \in \mathcal{E} \tag{6.3}$$

Here, x_t^i is the charge power of charging point i at time t, T_{ij} is a matrix that encodes the topology of the distribution network ($T_{ij} = 1$ if EV i is supplied by transformer j), $\mathcal{L}$ is the set of transformers installed in the distribution network, and M^i is the maximum charge power of charger i. The optimal solution is an allocation that uses up the available capacity of the network at time t. Note that this optimization framework cannot prevent undervoltage incidents from happening as it ignores reactive power flow in the distribution network. It is one of the problems of using an optimization based approach that can be partly resolved by model-free controllers.

We model this convex optimization problem in CVXPY and solve it using Mosek software to obtain a baseline. The problem is similar to the centralized approach used in [8] except that it maximizes the sum of charge powers rather than the sum of the logarithm of each charge power. Using the logarithmic utility function allows to achieve *proportionally fairness* while allocating power to EV charging points. However, it complicates the design of the RL-based controller. Throughout this paper, we refer to this (6.1) approach as the *centralized* control method.

6.2.2 ACN Testbed

We formulate the following constrained optimization problem which can be solved in each time slot to control EV charging in the parking station:

$$\text{maximize: } \sum_{i \in E} x_t^i \tag{6.4}$$

subject to:

$$\left(\sum_i C_{ij}\cos\phi_i x_t^i\right)^2 + \left(\sum_i C_{ij}\sin\phi_i x_t^i\right)^2 \leq R_j^2, \forall j \tag{6.5}$$

$$0 \leq x_t^i \leq M^i, \forall i \tag{6.6}$$

Here, x_t^i is the charge current of EVSE i at time t (expressed in Amps), M^i is the maximum charge current of charger i, C is the constraint matrix where $0 \leq C_{ij} \leq 1$ denotes the contribution of EVSE i to the aggregated current of feeder j, ϕ_i is the current phase angle and R_j is the current rating of feeder j. All constraints are expressed in terms of current (in Amps). Note that ϕ_i is a constant value, typically close or equal to zero.

We solve this convex optimization problem in a centralized fashion for each time slot and use the solution as the expert demonstration in offline training of the RL agents.

6.3 Adaptive AIMD

In this section we propose the Adaptive AIMD (A-AIMD) algorithm – an RL-based controller which mimics the AIMD method used in TCP congestion control. However, unlike AIMD, A-AIMD does not have a fixed additive increase factor and a fixed multiplicative decrease factor. It learns these parameters from interaction with the environment (see Algorithm 3). In particular, each RL agent learns the mapping between the congestion signals and a continuous action, $-1 \leq \alpha \leq 1$. This action determines the additive increase factor and the multiplicative decrease factor when it is positive and non-positive, respectively.

Although RL agents are supposed to learn the state-action mapping through online interaction, previous work [16], [54] shows that prior offline training can significantly improve their performance during the online interaction, especially in real-world applications. Therefore, inspired by the RL-based HVAC controller proposed in [16], we use imitation learning (described in the next subsection) and expose our RL agents to the trajectories generated by an expert, i.e., the optimal control policy obtained by solving the centralized optimization problem which was presented earlier.

```
Algorithm 3: Adaptive AIMD: Online Interaction
   input : Hyperparameters: $A > 0$ and $B \geq 1$; Pre-trained policy, $\bar{\pi}^i_\theta$;
1  for $e = 1, 2, \ldots, \#$ of episodes do
2      for $t = 1, 2, \ldots, \#$ of steps do
3          $s^i_t \longleftarrow (t, x^i_t, \Lambda^i_t)$
4          $\alpha \longleftarrow \bar{\pi}^i_\theta(s^i_t)$
5          if $\sum_{\lambda \in \Lambda^i_t} \lambda \geq 1$ then                // if congested
6              $\alpha \longleftarrow -|\alpha|$
7              $x_{t+1} \longleftarrow x_t * B^\alpha$
8          else
9              $\alpha \longleftarrow |\alpha|$
10             $x_{t+1} \longleftarrow \min\{M, x_t + A\alpha\}$
11         end
12         $(r_{t+1}, \Lambda^i_{t+1}) \longleftarrow$ env.step$(x^i_{t+1})$       // charge at $x^i_{t+1}$
13         $s^i_{t+1} \longleftarrow (t + 1, x^i_{t+1}, \Lambda^i_{t+1})$
14         $\mathcal{D}^i \longleftarrow \mathcal{D}^i \cup \big\{(s^i_t, \alpha, s^i_{t+1}, r_{t+1})\big\}$
15         $\theta \longleftarrow SAC(\mathcal{D})$
16     end
17 end
```

Algorithm 3 presents the online interaction phase of A-AIMD for the agent i. After perceiving the state at time t (line 3), it samples action α (line 4) from the policy $\bar{\pi}_\theta$ trained offline. In line 5, it checks for congestion by computing the sum of binary congestion signals received from the transformers. If congestion is detected, it ensures that the action is negative (line 6) and reduces the charge power by multiplying it by $0 < B^\alpha \leq 1$. When there is no congestion, α is made positive (line 9) and the charge power is increased by an additive factor $A\alpha$ (line 10). Note that $B \geq 1$ and $A > 0$ are hyperparameters of this algorithm.

The algorithm then receives the next state and reward after charging an EV at rate x_{t+1} (lines 12-13). We use *soft actor-critic* (SAC) as our RL algorithm (line 15) which requires a replay buffer $\mathcal{D}$. The replay buffer is the collection of current state, action, next state, and reward (line 14). We explain the SAC algorithm later in this section.

6.3.1 Imitation Learning

Imitation learning is a supervised learning method for an RL agent to learn the policy from expert demonstration. The basic idea is to pre-train the agent from trajectories generated by a domain expert.

There are three types of imitation learning [73]: *behavioral cloning* (BC), *inverse reinforcement learning* (IRL), and *direct policy learning* (DPL). In BC, agents are provided with expert demonstrations as data, previously generated by the optimal policy. This approach is particularly useful when historical control data is available. For example, Chen et al. [16] use the data available through the Building Automation System (BAS) to train an agent to optimally control the HVAC system. In IRL, agents learn a reward function instead of a policy [50], [56]. This approach is useful when the reward function is unknown and is statistically easier to learn than a policy. DPL is an improved version of BC where an interactive expert is available during offline training. In this case, instead of using historical data, agents initialize the policy parameter through interaction with the optimal controller. DPL seems to be the most suitable imitation learning approach for our control problem; thus, we adopt it in the offline training of RL agents.

6.3.2 Offline Direct Policy Learning

Algorithm 4 describes the offline pre-training of an agent. In line 4, the optimal charge power, which is the solution of the optimization problem at time t, is compared with the current charge power. Based on the comparison, the action α is learned either as multiplicative decrease factor (lines 5-6) or as additive increase parameter (lines 8-9). The rest of the algorithm (lines 11-14) is similar to Algorithm 3.

Note that the action α is calculated based on the optimal charge power instead of sampling from π_θ. The solution of the optimization problem (6.1) or (6.4) is treated as expert demonstration. In each time step, the buffer keeps track of the current state, best action, next state, and reward (line 13). Note that once the optimal action is obtained from the centralized method, direct

Algorithm 4: Offline Direct Policy Learning

```
input : Optimal Control Policy, $\{z_t^i\}_t$; Initial policy; $\pi_{\theta_0}$
for $e = 1, 2, \dots, \#$ of episodes do
    for $t = 1, 2, \dots, \#$ of steps do
        $s_t \longleftarrow (t, x_t^i, \Lambda_t^i)$
        if $x_t^i > z_t^i$ then
            $\alpha \longleftarrow \max\left\{\log_B\left(\frac{z_t^i}{x_t^i}\right), -1.0\right\}$
            $x_{t+1} \longleftarrow x_t * B^\alpha$
        else
            $\alpha \longleftarrow \min\left\{\frac{z_t^i - x_t^i}{A}, 1.0\right\}$
            $x_{t+1} \longleftarrow \min\{M, x_t + A\alpha\}$
        end
        $(r_{t+1}, \Lambda_{t+1}^i) \longleftarrow$ env.step$(x_{t+1}^i)$
        $s_{t+1}^i \longleftarrow (t+1, x_{t+1}^i, \Lambda_{t+1}^i)$
        $\mathcal{D}^i \longleftarrow \mathcal{D}^i \cup \left\{(s_t^i, \alpha, s_{t+1}^i, r_{t+1})\right\}$
        $\theta \longleftarrow SAC(\mathcal{D})$
    end
end
```

policy learning can be done by training a supervised machine learning model, e.g., a neural network. We use soft-actor critic because it has its own neural network to learn policy parameters from the provided examples.

6.3.3 Soft Actor-Critic (SAC)

We use a policy gradient method for charging EVs as it is the most natural choice for continuous action and state space systems [66]. In a policy gradient method, the policy is presented by a parameterized function (possibly a neural network) π_θ. The agent updates the policy parameter towards the direction of the gradient of a performance function:

$$\theta \longleftarrow \theta + \gamma \nabla_\theta J(\theta). \qquad (6.7)$$

The Soft Actor-Critic (SAC) algorithm [23] is a recently published policy gradient method that incorporates the entropy measure of the policy into the reward to encourage exploration. It is an off-policy actor-critic model following the maximum entropy reinforcement learning framework which tries to learn a policy that acts as randomly as possible while still succeeding at the task.

The policy is trained with the objective to maximize the expected return and the entropy at the same time:

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}_{(s_t, \alpha_t) \sim \rho_\pi} [r_t + \eta \mathcal{H}(\pi(.|s_t))]. \tag{6.8}$$

Here ρ_π is the marginal distribution of state-action pairs, i.e., (s_t, α_t) induced by policy π, $\mathcal{H}$ is the entropy measure, and η determines the importance of the entropy term (aka the temperature parameter). The entropy maximization leads to policies that explore more and capture multiple modes of near-optimal strategies, i.e., if there are multiple options that seem to be equally good, the policy should assign each with an equal probability to be chosen.

The SAC aims to learn three functions: the policy π_θ, the soft Q-value function Q_ψ, and the soft state value function V_ϕ. Although V can be inferred theoretically from Q and π, a separate V function is shown to be effective in stabilizing training in practice [23]. The soft Q-value and the soft state value are defined below:

$$Q(s_t, \alpha_t) = r_t + \gamma \mathbb{E}_{s_{t+1} \sim \rho_\pi} [V(s_{t+1})] \tag{6.9}$$

$$V(s_t) = \mathbb{E}_{\alpha_t \sim \pi} [Q(s_t, \alpha_t) - \log \pi(\alpha_t | s_t)] \tag{6.10}$$

Thus, $Q(s_t, \alpha_t)$ can be written as:

$$Q(s_t, \alpha_t) = r_t + \gamma \mathbb{E}[Q(s_{t+1}, \alpha_{t+1}) - \log \pi(\alpha_{t+1} | s_{t+1})] \tag{6.11}$$

Both the Q-function and the state value function V can be implemented by neural networks [23]. The soft state value function is trained to minimize the mean squared error:

$$J_V(\phi) = \mathbb{E}_{s_t \sim \mathcal{D}} \left[\frac{1}{2} (V_\phi(s_t) - \mathbb{E}[Q_\psi(s_t, \alpha_t) - \log \pi_\theta(\alpha_t | s_t)])^2 \right] \tag{6.12}$$

with the stochastic gradient

$$\nabla_\phi J_V(\phi) = \nabla_\phi V_\phi(s_t)(V_\phi(s_t) - Q_\psi(s_t, \alpha_t) + \log \pi_\theta(\alpha_t | s_t)) \tag{6.13}$$

Here D is the distribution of previously visited (s_t, α_t) pairs.
Similarly, the Q-function is trained to minimize the mean squared loss:

$$J_Q(\psi) = \mathbb{E} \left[\frac{1}{2} (Q_\psi(s_t, a_t) - (r_t + \gamma \mathbb{E}[V_{\bar{\phi}}(s_{t+1})]))^2 \right] \tag{6.14}$$

Algorithm 5: Soft Actor Critic (SAC)

```
input  : Initialized parameters: $\phi, \bar{\phi}, \psi, \theta$, Mixture Coefficient: $\tau$
output: Policy Parameter: $\theta$
for each gradient step do
    $\phi \longleftarrow \phi - \gamma_v \nabla_\phi J_V(\phi)$
    $\theta \longleftarrow \theta - \gamma_\pi \nabla_\theta J_\pi(\theta)$
    $\psi \longleftarrow \psi - \gamma_q \nabla_\psi J_Q(\psi)$
    $\bar{\phi} \longleftarrow \tau\phi + (1-\tau)\bar{\phi}$
end
```

with the stochastic gradient

$$\nabla_\psi J_Q(\psi) = \nabla_\psi Q_\psi(s_t, \alpha_t)(Q_\psi(s_t, \alpha_t) - r_t + V_{\bar{\phi}}(s_{t+1})) \tag{6.15}$$

Here $\bar{\phi}$ is the target value function which is the exponential moving average of ϕ.

The policy could be a Gaussian with mean and covariance given by neural networks [23]. Instead of directly maximizing the performance (6.8), SAC uses KL-divergence to update its policy parameter θ:

$$\begin{aligned}\pi_{new} &= \arg\min_{\pi' \in \prod} D_{KL}\left(\pi(.|s_t) \| \frac{\exp(Q^{\pi_{old}}(s_t, .))}{Z^{\pi_{old}}(s_t)}\right) \qquad (6.16)\\ &= \arg\min_{\pi' \in \prod} D_{KL}\left(\pi(.|s_t) \| \exp((Q^{\pi_{old}}(s_t, .)) - \log Z^{\pi_{old}}(s_t))\right)\end{aligned}$$

to minimize the objective function $(J_\pi(\theta))$ defined as follows

$$\begin{aligned}\nabla_\theta J_\pi(\theta) &= \nabla_\theta D_{KL}\left(\pi(.|s_t) \| \exp((Q^{\pi_{old}}(s_t, .)) - \log Z^{\pi_{old}}(s_t))\right)\\ &= \mathbb{E}_{\alpha_t \sim \pi}[\log \pi_\theta(\alpha_t|s_t) - Q_\psi(\alpha_t, s_t)] \qquad (6.17)\end{aligned}$$

Here, $\prod$ is the set of all potential policies, $Z^{\pi_{old}}$ is the partition function to normalize the distribution; we discard this term in Equation (6.17) as it does not contribute to the gradient.

We adopt the soft actor-critic algorithm presented in [23] and call it in our proposed algorithms (Algorithm 3 and Algorithm 4). Algorithm 5 shows different steps of the SAC algorithm.

6.4 Baseline Algorithms

Centralized method: solves the centralized optimization problem explained in Section 6.2.

AIMD: is the basic AIMD algorithm with fixed rates for additive increase and multiplicative decrease parts. The algorithm is similar to Algorithm 3 except that α is fixed. In particular, when there is no congestion it increases the charge power by one unit. If congestion is detected, it halves the current charge power.

Direct Rate Learning: is an RL-based algorithm that tries to learn the charge power directly from interaction with the environment instead of learning the additive increase and multiplicative decrease rates. The action space for this algorithm is the charge power, $0 \leq x^i \leq M^i$ for the i^{th} EV.

This algorithm also has two phases which are similar to Algorithm 3 and Algorithm 4 but the learned action is the new charge power, $0 \leq x^i \leq M^i$ rather than the parameters of AIMD.

Earliest Deadline First (EDF) and **Least Laxity First (LLF)** as described in Chapter 4, Section 4.4.

Chapter 7

Experimental Results for Model-Free Control Strategies

This chapter evaluates the model-free, reinforcement learning-based control strategies described in Chapter 6 and discusses their performance results. Section 7.1 describes the simulation environment and performance metrics for the distribution network and parking station. Section 7.2 and Section 7.3 present the performance evaluation results for the distribution network and the parking station respectively.

7.1 Simulation Environment

Our RL agents are implemented in PyTorch [52]. We use OpenAI Gym [13] wrapper to implement the environment. For the parking station, we use 50 and 20 episodes for offline pre-training and online interaction, respectively. We witnessed that after only 15 episodes (i.e., 10×288 steps) A-AIMD learns the optimal policy. However, DRL needs more episodes to converge. For the distribution network, we considered 50 and 30 episodes for offline pre-training and online interaction, respectively. In this case, we witnessed that after only 25 episodes (i.e., 25×144 time steps) A-AIMD learns the optimal policy. For A-AIMD, we choose the hyperparameters, $A = B = 2$. For the distribution network, in all scenarios (unless otherwise stated) we run the simulation for 14 days and report the average result.

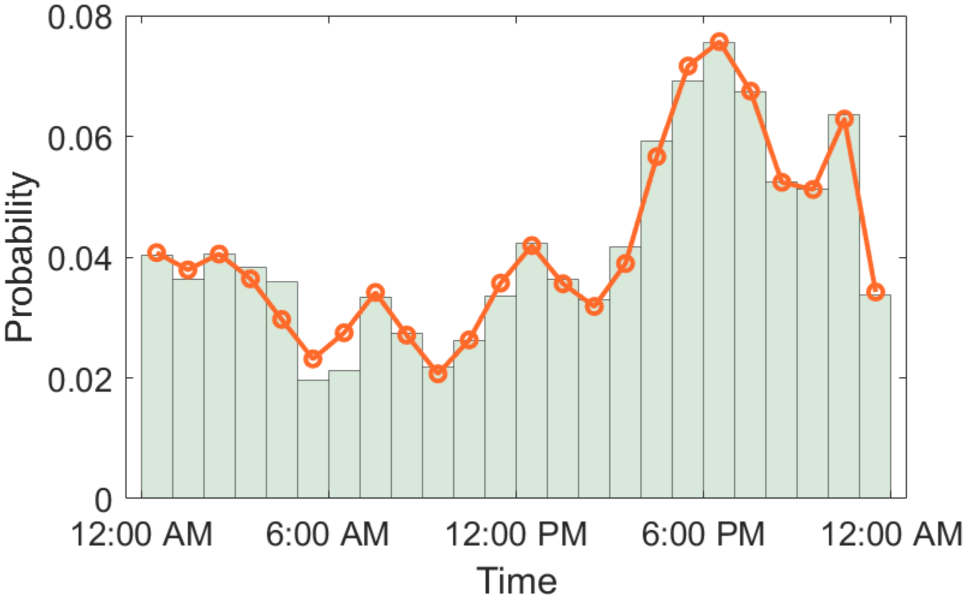

Figure 7.1: Distribution of arrival times.

7.1.1 Performance Metrics

We use the following performance metrics to evaluate the proposed algorithm:

Jain index: this index is used to evaluate fairness of a power allocation scheme (a higher value suggests a fairer scheme):

$$J(x) = \frac{\left(\sum_i x_i\right)^2}{n \sum_i x_i^2}$$

where x is a vector of charge powers (or currents) of all charging points in a given time slot and n is the number of active charging points. The index is calculated for every time slot and then averaged over all time slots in one day.

Percentage of EVs with a certain SoC: it is the percentage of connected EVs with a SoC greater than or equal to a certain threshold when they depart.

Resource utilization: it is the loading of a transformer (in MVA) or a line (in A) over one day.

7.2 Case Study 1: Distribution Network

We consider the 33-bus system [11] shown in Chapter 3 (Figure 3.1) as the primary distribution network, and adopt the IEEE European low voltage test feeder [27] as the secondary distribution network connected to nodes 2-33 of

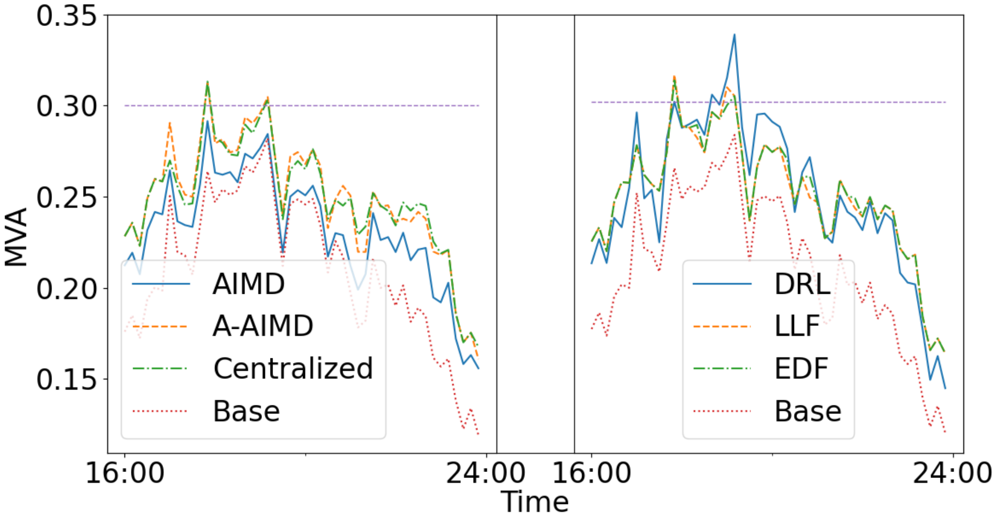

Figure 7.2: Power loading of a distribution transformer under the centralized, A-AIMD and AIMD algorithms (right figure) and under DRL, LLF and EDF control policies (left figure). The dashed horizontal line is the rated capacity and the red curve is the base load.

the primary system. We connect one or more homes to each of the end nodes of the low-voltage feeders to create the base load. Each of these homes can have one EV, hence multiple EVs may be connected to a single end node. To model the home loads, which are inelastic and cannot be controlled, we utilize sample household demands (with 1-second resolution) from the ADRES-CONCEPT project [2]. The data set contains real and reactive power consumption of 30 households over two weeks, hence we will get 420 unique load profiles when they are split into 1-day segments. To obtain more unique load profiles, we add white Gaussian noise with 10% standard deviation to each sample to create a total of 8400 unique household load data.

We consider a maximum of 500 EVs in our distribution system, although the actual number of connected EVs depends on their mobility and SoC. The arrival times of EVs are generated from the data provided in the Pecan Street data set [53]. Since there are only about 80 EVs in this data set, we take the daily arrival times of these EVs over one year (2016) and fit a Gaussian Mixture Model (GMM) to the probability distribution function and sample the daily arrival times from the fitted GMM. The distribution of arrival times and the fitted model with 7 Gaussian distributions are shown in Figure 7.1.

The data set does not include EV departure times, so we randomly sample the sojourn time of each EV from a Gaussian distribution with $\mu = 8$ and $\sigma = 2$ (in hours). We then calculate the departure time of each EV by adding their sojourn time to their arrival time. We disconnect EVs when they are fully charged or manually unplugged (whichever happens first). We randomly select the battery size of each EV from four of the popular models in the market: a) $16kWh$ for Chevy Volt and Mitsubishi iMiEV, b) $30kWh$ for Nissan Leaf, c) $42kWh$ for BMW i3, and d) $75kWh$ for Tesla Model 3. The initial SoC of each EV at the arrival time is sampled uniformly between 0 and 0.1. We consider 10-minute time intervals and run a day-long simulation for each episode.

7.2.1 Addressing Transformer Overloading Problem

Figure 7.2 shows loading of a distribution transformer during the peak time when charging points are controlled using the centralized, A-AIMD, AIMD, DRL, LLF, and EDF algorithms. We observe that in all cases except AIMD, there are some small overshoots which we attribute to changes in residential loads. The reason that there are overshoots in case of A-AIMD but no overshoots in case of AIMD is that when there is no congestion, A-AIMD tries to increase the charge power more aggressively than AIMD. However, as soon as the network is congested, A-AIMD quickly responds to the congestion signal and we do not have any sustained overloading in the system. We see larger overshoots for DRL, which we attribute to the large number of episodes required to train the RL agents.

7.2.2 Voltage Control

Figure 7.3 shows the voltage of one of the primary nodes under A-AIMD with and without voltage congestion signals. A congestion signal is sent to the RL agents when the nodal voltage level goes below 0.95 p.u. or above 1.05 p.u.. As shown in the figure, A-AIMD can effectively respond to voltage congestion signals by reducing the charge power of charging points downstream of the node with the voltage problem. As discussed earlier, the centralized optimization based controller is not capable of dealing with voltage problems as it ignores

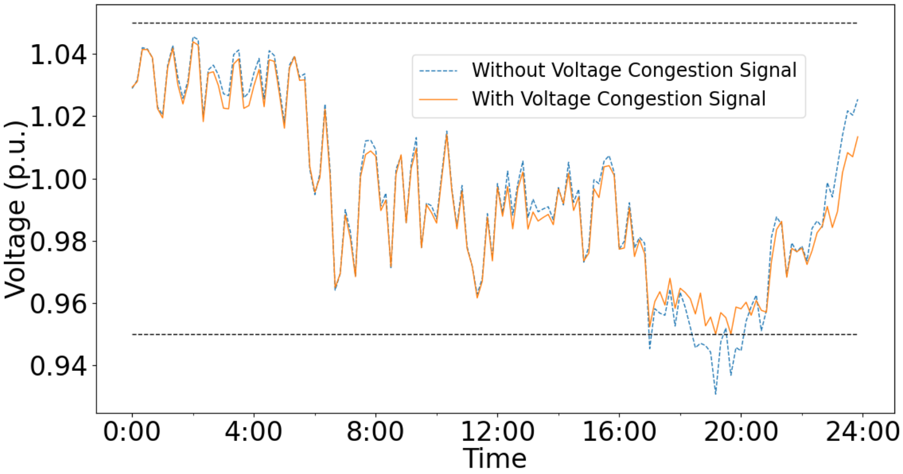

Figure 7.3: The voltage of one of the primary nodes of the distribution network using A-AIMD with and without considering voltage congestion signals. The horizontal lines indicate ±5% of the nominal voltage.

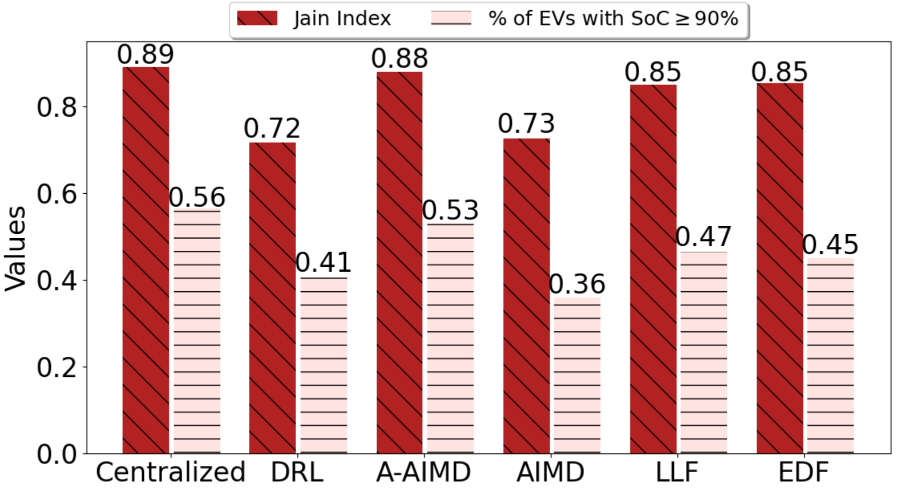

Figure 7.4: Comparing performance of A-AIMD with the baseline algorithms.

reactive power flows. The same observation can be made for EDF and LLF algorithms.

7.2.3 Fairness & User Satisfaction

Figure 7.4 presents the comparison between different algorithms in terms of fairness and the number of EVs charged up to 90% of their battery capacity. It can be seen that A-AIMD closely follows the centralized method and outperforms all the other methods. We also notice that both AIMD and DRL fail to outperform EDF and LLF, which merely rely on an admission control scheme.

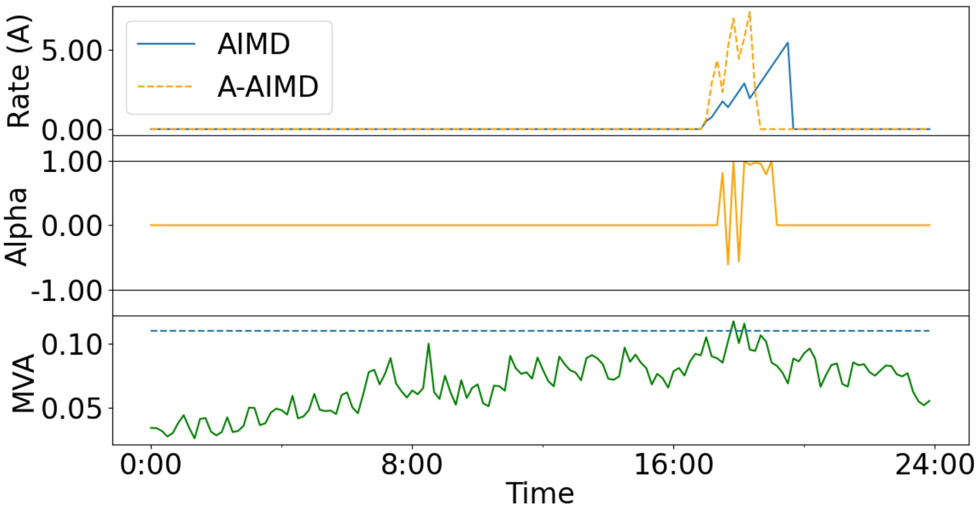

Figure 7.5: Comparing rate changes under AIMD and A-AIMD.

7.2.4 Adaptive Nature of A-AIMD

Figure 7.5 presents the adaptive nature of A-AIMD. The top figure shows the charge power of an EV when it is controlled by AIMD and A-AIMD algorithms. The middle figure shows the values of the parameter α learned by A-AIMD. The bottom figure shows the power loading and rated capacity (dashed line) of one of the phases of the transformer that feeds this EV and many other EVs. The EV arrives and connects to the charging station around 6:00pm. Given the available capacity, both A-AIMD and AIMD algorithms start to increase the charge power upon the arrival of the EV. However, the A-AIMD algorithm quickly increases its additive increase rate. We witness that when there is congestion, α becomes negative and A-AIMD reduces the charge power of the EV to relieve congestion. AIMD also decreases the charge power multiplicatively to avoid congestion. The differences in additive and multiplicative rates cause A-AIMD to fully charge the EV's battery much earlier than AIMD.

7.2.5 Resource Utilization

Results obtained for a day-long simulation period indicate that A-AIMD increases the total energy delivered to EVs by 4% on average compared to AIMD by better utilizing the available capacity of the system. Both methods overshoot the rated capacity of transformers for short periods of time, but quickly

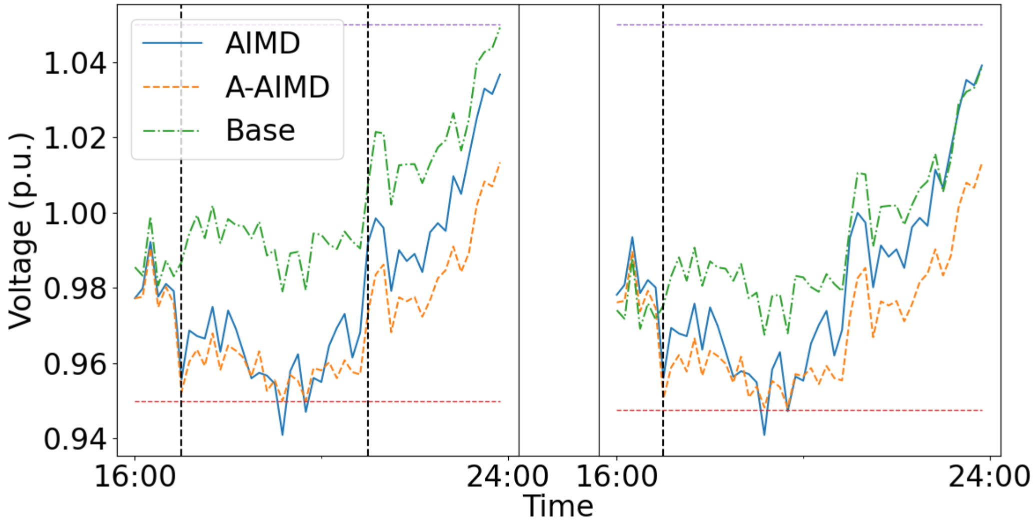

Figure 7.6: Voltage profile of one of the primary nodes under A-AIMD and AIMD algorithms. In the left figure the substation voltage is increased between 5:00pm and 9:00pm (marked by dashed vertical lines). In the right figure the configuration is changed at 5:00pm (marked by the dashed vertical line).

react to the received congestion signal by reducing the charge power of connected EVs. The A-AIMD algorithm results in a few more overshoots compared to AIMD, but none of them causes a sustained overloading which could damage the transformer.

7.2.6 Adaptability to Changes in Distribution Network

Distribution system operators frequently adjust the tap position of tap-changing transformers or change the configuration of the network for voltage regulation, loss reduction, etc. Thus, we consider two scenarios to study how our method can adapt to such changes in the network.

Scenario A – Adjusting Substation Voltage

We change the voltage level of the substation transformer during a certain period of time to see whether A-AIMD can still adapt its charging rate to avoid voltage problems. Figure 7.6 (left panel) shows the voltage profile of one of the primary nodes of the distribution network for A-AIMD and AIMD algorithms. The voltage of the substation is increased by 0.1 p.u. between 5:00pm and 9:00pm. We can see that both AIMD and A-AIMD can avoid

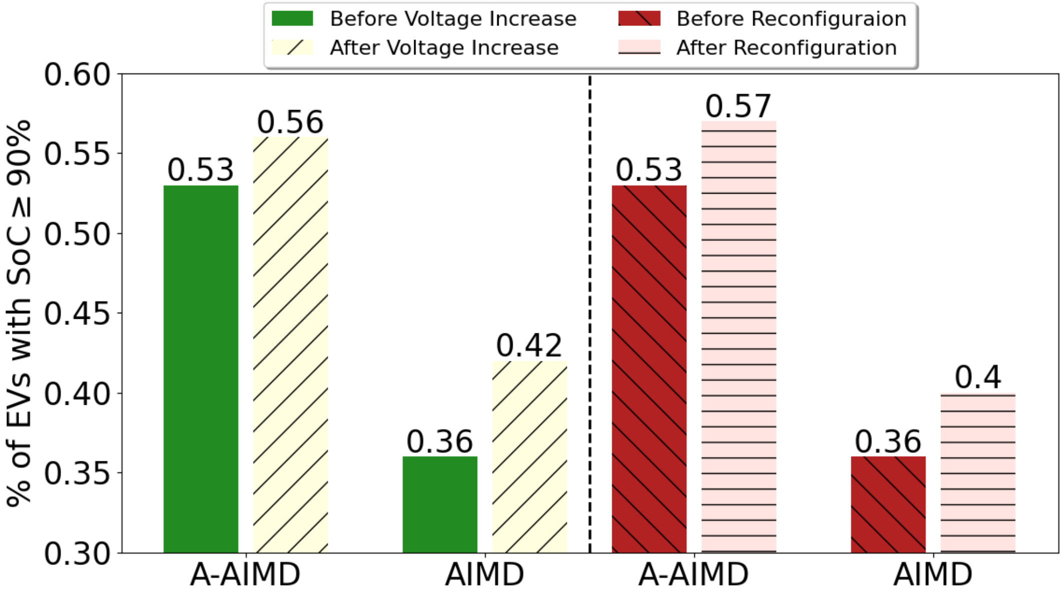

Figure 7.7: Performances of A-AIMD and AIMD before and after voltage change (left panel) and reconfiguration (right panel). Note that the y-axis is truncated.

congestion during this time. However, unlike AIMD, A-AIMD learns to update the charge power less aggressively.

Figure 7.7 (left panel) shows the performances of AIMD and A-AIMD algorithms with and without the voltage change at the substation. As the increase in voltage level creates more room for charging EVs, the performances of both algorithm improves in this case. Expectedly, A-AIMD has better performance than AIMD before and after the tap change.

Scenario B – Change in Configuration

To further evaluate adaptability of the A-AIMD, we consider a reconfiguration event. We perform a simple reconfiguration to merely evaluate the performance of our method when the current configuration of the system is different from the configuration on which we trained our RL agents. Recall that the tie switch we added between Nodes 12 and 22 is normally open. At 5:00pm, we close this tie switch and open the sectionalizing switch between Nodes 11 and 12 to maintain the tree structure.

Figure 7.6 (right panel) shows the voltage profile of one of the phases at Node 18 under AIMD and A-AIMD control algorithms. While both algorithms

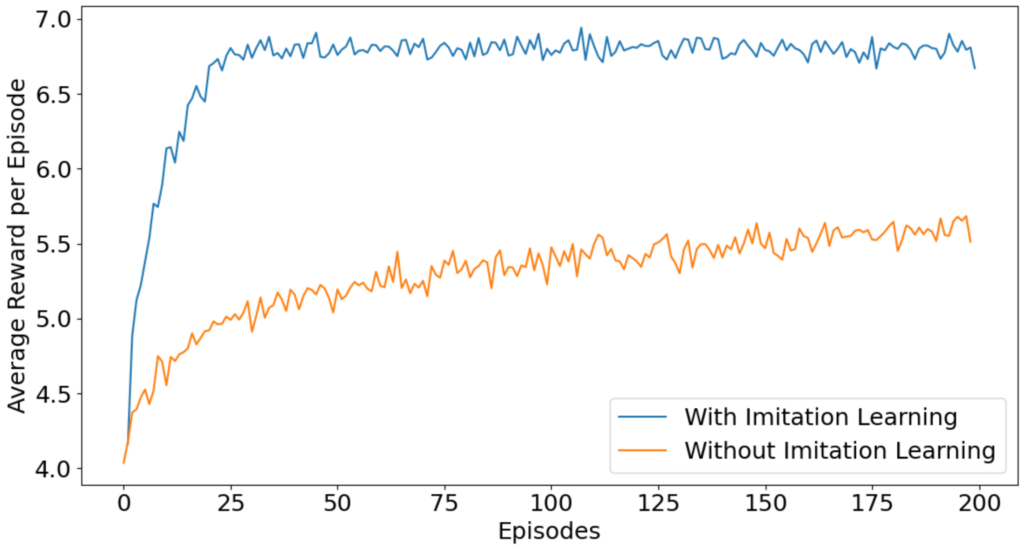

Figure 7.8: Average reward per episode (in MVA) of A-AIMD achieved with and without imitation learning. Here the average is taken over all time steps of an episode and all active agents.

avoid congestion successfully, A-AIMD quickly perceives the updated state and learns to update the charge power less aggressively. Figure 7.7 (right panel) depicts the performance of AIMD and A-AIMD algorithms before and after the reconfiguration. As the recongifuration increases the voltage level at Node 18, there is more available capacity for charging EVs. Thus, the performance of both algorithms improves.

Imitation Learning & Reducing Sample Complexity

Figure 7.8 shows the average reward per episode required by A-AIMD in the online interaction phase with and without imitation learning in the offline training phase. It can be seen that with imitation learning, A-AIMD requires around 25 episodes to converge. However, without using imitation learning in the offline training phase the agent does not achieve a stable reward after 200 episodes and many more episodes will be needed to achieve comparable performance. We conclude that using imitation learning can significantly speed up the learning process.

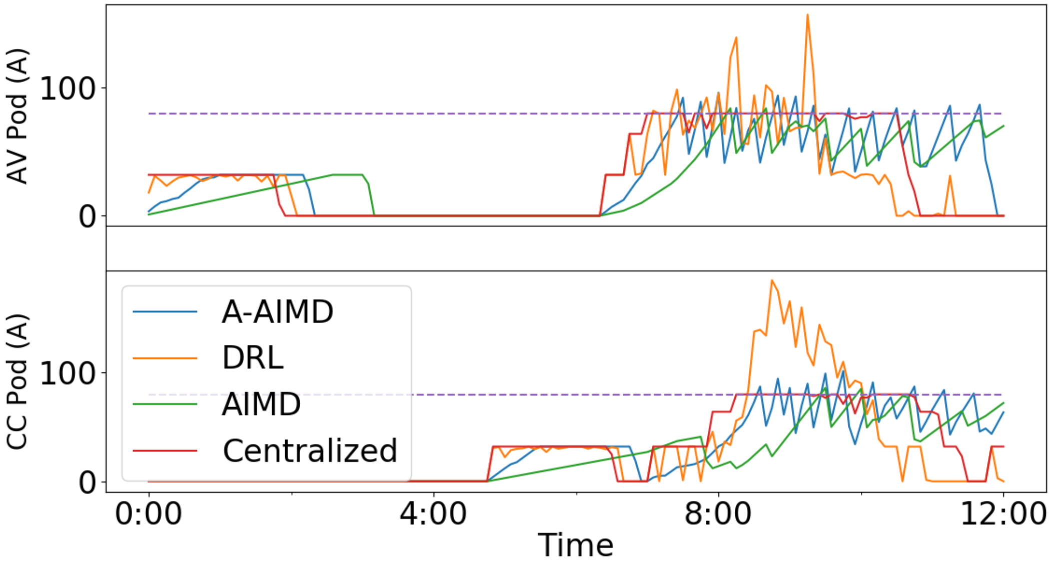

Figure 7.9: The charge current of two charging pods for different algorithms. The dashed lines indicate their nameplate capacity which is $80A$.

7.3 Case Study 2: ACN-Sim

We implement our RL based methods on top of ACN-Sim and use EV schedules that are generated according to ACN-Data. We consider a one-day long (from 2018/09/05 to 2018/09/06, LA Timezone) simulation with 5-minute time intervals. ACN-Sim utilizes a realistic *linear two-stage* [33] battery model for EVs. It is a piecewise linear model that consists of two stages: 1) bulk charging, for the first 80% SoC and 2) absorption, for the remaining 20% SoC. In the first stage, the current drawn is almost constant. In the second stage, it decreases linearly as the battery reaches the full charge. We use this battery model in our simulations.

7.3.1 Congestion Avoidance

Figure 7.9 shows the aggregated current loading of different algorithms for the two charging pods (AV: AeroVironment, CC: Clipper Creep). We can see that DRL exceeds the rated capacity significantly during the congestion period. This can be attributed to the insufficient number of episodes for training the RL agents. The A-AIMD alogrithm closely follows the optimum solution and although there are some small overshoots, it is much better than AIMD in

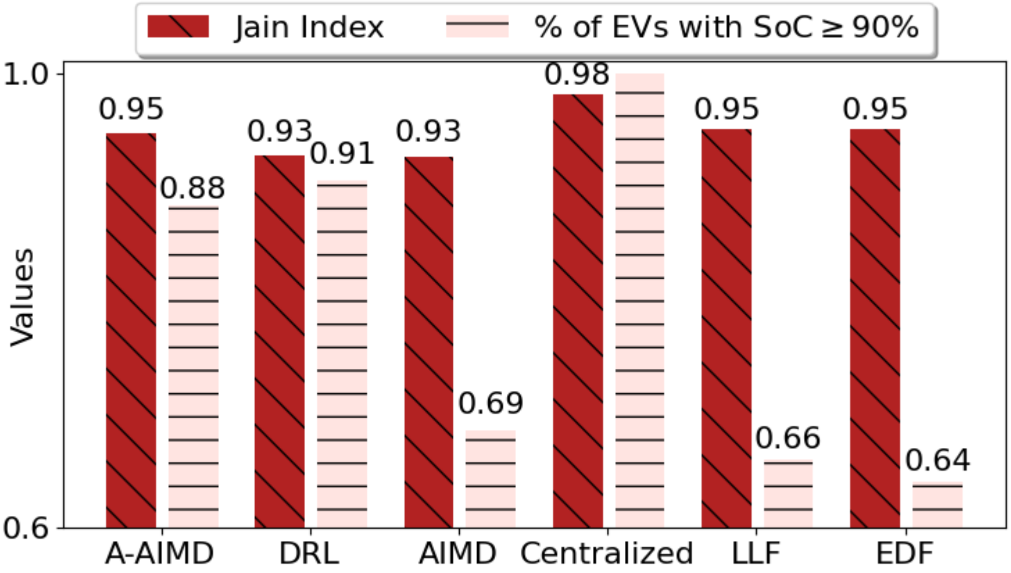

Figure 7.10: Performance comparison of different control algorithms in ACN-Sim.

terms of utilization.

7.3.2 Performance Analysis

Figure 7.10 compares the performances of different algorithms. We can see that A-AIMD has the closest performance to the centralized controller when both indices are considered. DRL outperforms A-AIMD in terms of the SoC index since it overshoots the targets many times and therefore does not respect the constraints (Figure 7.9). The absence of inelastic loads in this case study makes the system more predictable than the distribution network. Thus, RL agents can quickly learn the optimal policy.

7.3.3 Adaptive Nature of A-AIMD

Figure 7.11 illustrates the adaptive nature of A-AIMD. The top figure shows the charge power of a randomly selected charging point over the course of a day when it is controlled by AIMD and A-AIMD algorithms. The middle figure shows the values of the parameter α learned by A-AIMD. The bottom figure shows the current flow in the line that feeds this charging point along with several other charging points, and its rated capacity (the dashed horizontal line). Two EVs arrive and are plugged into this charging point at around 6:00am and

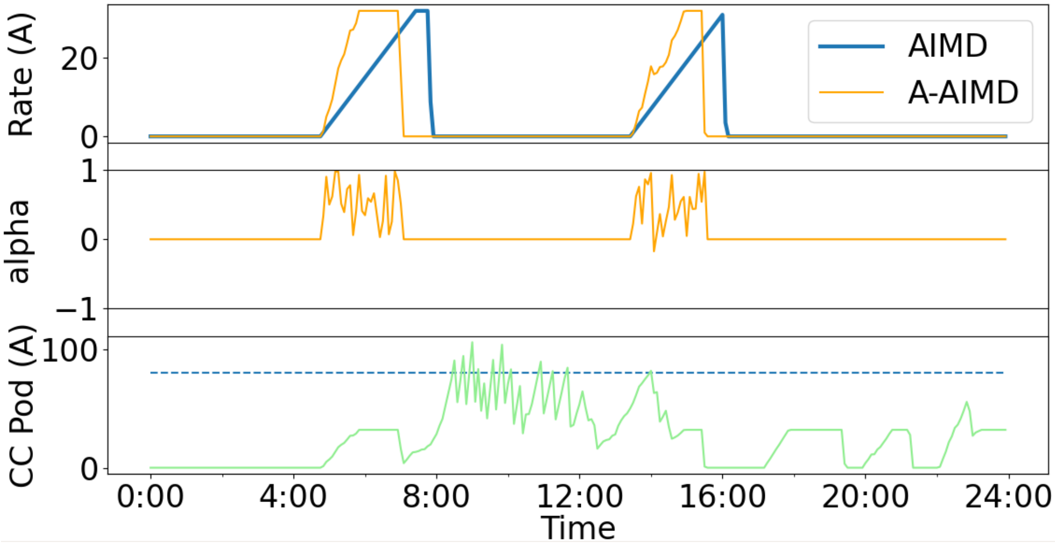

Figure 7.11: The adaptive nature of A-AIMD.

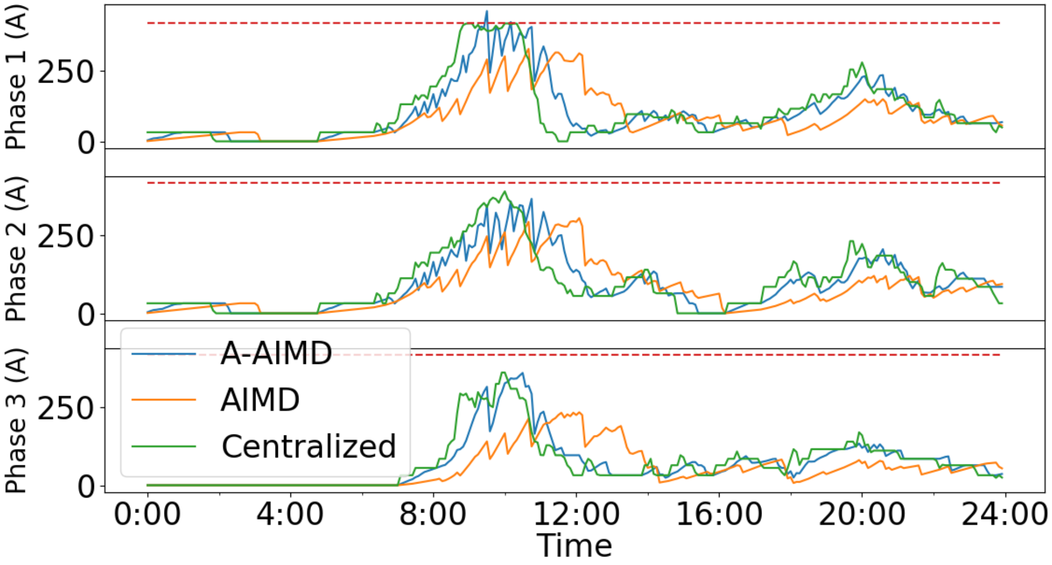

Figure 7.12: The currents flowing in three phases conductors on the secondary side of the transformer for different algorithms. The dashed lines indicate the thermal limit which is $416.16A$ for each phase conductor.

2:00pm respectively. Given the available capacity of this line, both A-AIMD and AIMD algorithms start to increase the charge power upon arrival of an EV. However, the A-AIMD algorithm quickly increases its additive increase rate of the charge power and manages to charge the EV's battery up to a high SoC before the line gets congested. The AIMD algorithm slowly increases its charge power. Once the line becomes congested, it has to decrease the charge

Table 7.1: Energy (Amp-minute) supplied above the rated capacity of different lines.

	Energy supplied above the rated capacity	
	A-AIMD	AIMD
CC Pod	126.318	104.625
AV Pod	168.71	160.0
Phase 1	44.33	0
Phase 2	0	0
Phase 3	0	0

Table 7.2: Energy (Amp-minute) supplied below the rated capacity through different lines. Here, phase 1, 2, 3 indicate the phase currents of the secondary side

	Energy supplied below the rated capacity	
	A-AIMD	AIMD
CC Pod	36865.135	35728.278
AV Pod	45671.128	43945.439
Phase 1	154018.220	136382.77
Phase 2	140331.096	126518.372
Phase 3	98558.347	81513.958

power multiplicatively several times. Note that current constraint of the CC Pod (bottom figure) is not the only binding constraint for the algorithms.

7.3.4 Resource Utilization

Figure 7.12 shows the current loading of each phase of the secondary side. It can be seen that A-AIMD follows the central controller more closely compared to AIMD. The results obtained for a one-day simulation period indicate that A-AIMD increases the total energy delivered to EVs by 15% on average compared to AIMD by better utilizing the available capacity of transformers and lines. Both algorithms overshoot the rated capacity only slightly before the charging points reduce their charge powers in response to a congestion signal.

Table 7.1 and 7.2 show the total amount of energy supplied above and below the rated capacity of the charging pods and the three phases of the transformer during the one-day simulation period. Similar to the previous case, A-AIMD violates the rated capacity slightly more, but it utilizes the capacity much better than AIMD.

Chapter 8

Conclusion

This thesis addresses several challenges that are or will be faced due to the large scale adoption of EVs in the smart grid. In particular, several control strategies are proposed to satisfy EV owners and enhance grid's reliability from two different perspectives: model-based and model-free.

Given an approximate model (i.e., the admittance matrix) of the grid, this thesis presents a deadline-aware, reputation-based, proportionally fair and efficient power allocation to EV charging stations connected to a power distribution network. This power allocation is determined in real time by individual charging stations using a decentralized algorithm which converges rapidly to the available capacity of the network. The proposed algorithm has a lower communication overhead as it utilizes the power system instead of sending explicit messages from charging stations to sensing nodes installed in the network. Hence, the only communication that is necessary is for sending congestion signals to the charging stations downstream. We show through extensive simulations that our algorithm tracks the available capacity of the network, favors conservative users (with better reputation), and is as efficient in utilizing network capacity as a centralized control algorithm. We also prove that the scaled gradient projection algorithm has quadratic convergence under some conditions.

We then relax the assumption about the availability of a model of the grid and develop a model-free framework to control the charge power of EVs. Model-free, rule-based algorithms mimicking the behaviour of TCP congestion control schemes have been successfully applied to control charging of EVs in the power grid. These control algorithms are decentralized, do not assume the knowledge of the distribution network model, and only require measurement of sensors installed beyond the substation and a trivial signalling mechanism; these properties make them ideal for real-world deployment. Yet there are fundamental differences between the Internet and the power grid which call for improving the control performance by adjusting the charge control rules dynamically. In this thesis, we have proposed a novel approach based on multi-agent reinforcement learning to adapt the parameter of the modified AIMD algorithm to the network condition. The proposed control method is model-free and decentralized, and therefore does not require the knowledge of the power system model. It instead relies on congestion signals sent to charging points by sensors installed in parts of the network that are likely to be congested. Our simulation results confirmed that the proposed algorithm outperforms several baseline algorithms presented in the literature, and can effectively deal with various types of congestion including transformer overloading and voltage limit violations.

From both perspectives, the work presented in this thesis has several limitations that we plan to address in future work:

- For the SGPA algorithm, we assume that the hessian matrix of the dual objective function is almost diagonal at the optimum point in Theorem 4.3.1. We have not verified whether this assumption holds in practice; nevertheless, our simulations have so far shown the superior rate of convergence over the first-order gradient projection algorithm. We intend to perform extensive experiments with data obtained from a real test system to further corroborate our theoretical results in future work.

- The weight function defined in Section 4.1 can have other forms. The design of an incentive compatible mechanism which promotes truthfulness is deferred to future work.

- In a model-based approaches, we use the DC approximation of power flow in distribution systems and therefore cannot address voltage problems that may occur as a result of charging a large number of EVs.

- For the model-free methods, we do not have theoretical results to show that adaptive AIMD yields a fair allocation to connected EVs. We also assume that a voltage congestion signal is only sent to downstream chargers; however, in practice other charging points can be notified to address an undervoltage (i.e., voltage dropping below the lower limit) problem.
- Lastly, we only considered voltage magnitudes dropping below the lower limit as our voltage congestion, but we believe that with a simple modification, this algorithm can deal with voltage magnitudes rising above the upper limit in distribution grids with a high penetration of solar PV system. We aim to design this algorithm and evaluate its performance in future work

Despite these limitations, this thesis studies the control of EV charge power from two different perspectives and brings together different areas of computer science (e.g. Internet, Reinforcement Learning, Mechanism Design, Smart-grid) to develop a comprehensive EV charge power controlling framework. We hope that this framework will help the research community to design advanced control strategies for achieving a green, safe and sustainable smart-grid.

References

[1] J. L. A. Chis and V. Koivunen, “Reinforcement learning-based plug-in electric vehicle charging with forecasted price,” *IEEE Trans. Vehicular Technology*, vol. 66, no. 5, pp. 3674–3684, 2017.

[2] (). ADRES dataset. [Online].

[3] I. E. Agency. (2020). Global EV Outlook 2019, [Online]. Available: `https://www.iea.org/reports/global-ev-outlook-2020` (visited on 02/01/2020).

[4] C. Ahn, C.-T. Li, and H. Peng, “Optimal decentralized charging control algorithm for electrified vehicles connected to smart grid,” *Journal of Power Sources*, vol. 196, no. 23, pp. 10 369–10 379, 2011.

[5] A. Alyousef and H. de Meer, “Design of a TCP-like smart charging controller for power quality in electrical distribution systems,” in *10th International Conference on Future Energy Systems (e-Energy)*, ACM, 2019, pp. 128–138.

[6] O. Ardakanian *et al.*, “On identification of distribution grids,” *IEEE Trans. Control of Network Systems*, vol. 6, no. 3, pp. 950–960, 2019.

[7] O. Ardakanian, S. Keshav, and C. Rosenberg, “Real-time distributed control for smart electric vehicle chargers: From a static to a dynamic study,” *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2295–2305, 2014.

[8] O. Ardakanian, C. Rosenberg, and S. Keshav, “Distributed control of electric vehicle charging,” in *Proceedings of the fourth international conference on Future energy systems*, ACM, 2013, pp. 101–112.

[9] F. Baccelli and D. Hong, “AIMD, fairness and fractal scaling of TCP traffic,” in *Twenty-First Annual Joint Conf. of the IEEE Computer and Communications Societies*, IEEE, 2002, pp. 229–238.

[10] S. Bansal, M. N. Zeilinger, and C. J. Tomlin, “Plug-and-play model predictive control for electric vehicle charging and voltage control in smart grids,” in *53rd IEEE Conference on Decision and Control*, 2014.

[11] M. E. Baran and F. F. Wu, “Network reconfiguration in distribution systems for loss reduction and load balancing,” *IEEE Transactions on Power Delivery*, vol. 4, no. 2, pp. 1401–1407, 1989.

[12] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

[13] G. Brockman *et al.*, "OpenAI Gym," *arXiv preprint arXiv:1606.01540*, 2016.

[14] Y. Cao *et al.*, "A decentralized deadline-driven electric vehicle charging recommendation," *IEEE Systems Journal*, no. 99, pp. 1–12, 2018.

[15] L. C. Casals *et al.*, "Sustainability analysis of the electric vehicle use in europe for CO2 emissions reduction," *Journal of Cleaner Production*, vol. 127, pp. 425–437, 2016.

[16] B. Chen, Z. Cai, and M. Bergés, "Gnu-rl: A precocial reinforcement learning solution for building hvac control using a differentiable mpc policy," in *Proceedings of the 6th ACM Int. Conf. on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 2019, pp. 316–325.

[17] S. Chen, Y. Ji, and L. Tong, "Deadline scheduling for large scale charging of electric vehicles with renewable energy," in *2012 IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, IEEE, 2012, pp. 13–16.

[18] W. Chen, P. Zhuang, and H. Liang, "Reinforcement learning for smart charging of electric buses in smart grid," in *2019 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2019, pp. 1–6.

[19] D. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1–14, 1989.

[20] B. J. Claessens *et al.*, "Peak shaving of a heterogeneous cluster of residential flexibility carriers using reinforcement learning," in *IEEE PES ISGT Europe*, IEEE, 2013, pp. 1–5.

[21] R. M. Freund, "Newton's method for unconstrained optimization," 2004.

[22] K. Gardner, S. Borst, and M. Harchol-Balter, "Optimal scheduling for jobs with progressive deadlines," in *IEEE Conference on Computer Communications (INFOCOM)*, IEEE, 2015, pp. 1113–1121.

[23] T. Haarnoja *et al.*, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, 2018.

[24] J. Hong, X. Tan, and D. Towsley, "A performance analysis of minimum laxity and earliest deadline scheduling in a real-time system," *IEEE Transactions on Computers*, vol. 38, no. 12, pp. 1736–1744, 1989.

[25] J. Hu, S. You, M. Lind, and J. Ostergaard, "Coordinated charging of electric vehicles for congestion prevention in the distribution grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 703–711, 2014.

[26] Z. Huang *et al.*, “Discovering valuations and enforcing truthfulness in a deadline-aware scheduler,” in *IEEE Conference on Computer Communications (INFOCOM)*, IEEE, 2017.

[27] (). IEEE PES distribution systems analysis subcommittee, radial test feeders, [Online]. Available: `http://sites.ieee.org/pes-testfeeders/resources/`.

[28] K. Kaur *et al.*, “A colored petri net based frequency support scheme using fleet of electric vehicles in smart grid environment,” *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4638–4649, 2016.

[29] 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*, 2019.

[30] K. Kaur, M. Singh, and Kumar, “Multiobjective optimization for frequency support using electric vehicles: An aggregator-based hierarchical control mechanism,” *IEEE Systems Journal*, vol. 13, no. 1, pp. 771–782, 2019.

[31] F. Kelly, “Charging and rate control for elastic traffic,” *European transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, 1997.

[32] M. C. Kisacikoglu, F. Erden, and N. Erdogan, “Distributed control of pev charging based on energy demand forecast,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, pp. 332–341, 2018.

[33] Z. Lee *et al.*, “Large-scale adaptive electric vehicle charging,” in *Int. Conf. on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, IEEE, 2018, pp. 1–7.

[34] Z. Lee, D. Johansson, and S. Low, “ACN-Sim: An open-source simulator for data-driven electric vehicle charging research,” in *Int. Conf. on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, IEEE, 2019, pp. 1–6.

[35] W. Li *et al.*, “QTCP: Adaptive congestion control with reinforcement learning,” *IEEE Trans. Network Science and Engineering*, vol. 6, no. 3, pp. 445–458, 2019.

[36] M. Liu *et al.*, “Decentralized charging control of electric vehicles in residential distribution networks,” *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 266–281, 2019.

[37] M. Liu and S. McLoone, “Enhanced AIMD-based decentralized residential charging of EVs,” *Trans. of the Institute of Measurement and Control*, vol. 37, no. 7, pp. 853–867, 2015.

[38] M. Liu, Y. Shi, and X. Liu, “Distributed MPC of aggregated heterogeneous thermostatically controlled loads in smart grid,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1120–1129, 2016.

[39] M. Liu *et al.*, "Decentralized charging control of electric vehicles in residential distribution networks," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 266–281, 2017.

[40] M. Liu, P. K. Phanivong, and D. S. Callaway, "Customer-and network-aware decentralized EV charging control," in *Power Systems Computation Conference (PSCC)*, IEEE, 2018, pp. 1–7.

[41] Z. Liu *et al.*, "Optimal day-ahead charging scheduling of electric vehicles through an aggregative game model," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5173–5184, 2018.

[42] J. A. P. Lopes, F. J. Soares, and P. M. R. Almeida, "Integration of electric vehicles in the electric power system," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 168–183, 2011.

[43] S. H. Low and D. E. Lapsley, "Optimization flow control. i. basic algorithm and convergence," *IEEE/ACM Transactions on networking*, vol. 7, no. 6, pp. 861–874, 1999.

[44] X. Luo and K. W. Chan, "Real-time scheduling of electric vehicles charging in low-voltage residential distribution systems to minimize power losses and improve voltage profile," *IET Generation, Transmission, Distribution*, vol. 8, no. 3, pp. 516–529, 2014.

[45] R. Mehta *et al.*, "Smart charging strategies for optimal integration of plug-in electric vehicles within existing distribution system infrastructure," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 299–312, 2018.

[46] P. Moyal, "On queues with impatience: Stability, and the optimality of earliest deadline first," *Queueing Systems*, vol. 75, no. 2-4, pp. 211–242, 2013.

[47] Y. Nakahira *et al.*, "Smoothed least-laxity-first algorithm for EV charging," in *8th International Conference on Future Energy Systems (e-Energy)*, ACM, 2017, pp. 242–251.

[48] Y. Nakahira, A. Ferragut, and A. Wierman, "Minimal-variance distributed deadline scheduling in a stationary environment," *ACM SIGMETRICS PER*, vol. 46, no. 3, pp. 56–61, 2019.

[49] M. M. Nejad, L. Mashayekhy, and D. Grosu, "Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds," *IEEE transactions on parallel and distributed systems*, vol. 26, no. 2, pp. 594–603, 2015.

[50] A. Y. Ng, S. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Icml*, vol. 1, 2000, pp. 663–670.

[51] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[52] A. Paszke *et al.*, “Automatic differentiation in PyTorch,” 2017.

[53] (). Pecan Street dataset. [Online].

[54] B. Qi, M. Rashedi, and O. Ardakanian, “Energyboost: Learning-based control of home batteries,” in *Proceedings of the Tenth ACM Int. Conf. on Future Energy Systems*, 2019, pp. 239–250.

[55] J. Quirós-Tortós *et al.*, “Control of EV charging points for thermal and voltage management of LV networks,” *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 3028–3039, 2016.

[56] B. Recht, “A tour of reinforcement learning: The view from continuous control,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019.

[57] 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, 2012.

[58] J. Rivera *et al.*, “Alternating direction method of multipliers for decentralized electric vehicle charging control,” in *52nd IEEE Conference on Decision and Control*, IEEE, 2013, pp. 6960–6965.

[59] N. Sadeghianpourhamami, J. Deleu, and C. Develder, “Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning,” *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 203–214, 2020.

[60] I. Sharma, C. Cañizares, and K. Bhattacharya, “Smart charging of PEVs penetrating into residential distribution systems,” *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1196–1209, 2014.

[61] W. Shi and V. Wong, “Real-time vehicle-to-grid control algorithm under price uncertainty,” in *Int. Conf. on Smart Grid Communications (SmartGridComm)*, IEEE, 2011, pp. 261–266.

[62] R. N. Shorten and D. J. Leith, “On queue provisioning, network efficiency and the transmission control protocol,” *IEEE/ACM Trans. Networking*, vol. 15, no. 4, pp. 866–877, 2007.

[63] E. Sortomme *et al.*, “Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses,” *IEEE transactions on smart grid*, vol. 2, no. 1, pp. 198–205, 2011.

[64] S. Stüdli, E. Crisostomi, R. Middleton, and R. Shorten, “AIMD-like algorithms for charging electric and plug-in hybrid vehicles,” in *IEEE Int. Electric Vehicle Conf.*, IEEE, 2012, pp. 1–8.

[65] B. Subhonmesh, S. H. Low, and K. M. Chandy, “Equivalence of branch flow and bus injection models,” in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2012, pp. 1893–1899.

[66] R. S. Sutton and A. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[67] E. Ucer *et al.*, “An internet-inspired proportional fair EV charging control method,” *IEEE Systems Journal*, vol. 13, no. 4, pp. 4292–4302, 2019.

[68] E. Ucer, M. Kisacikoglu, and M. Yuksel, “Analysis of AIMD algorithm for EV charging,” in *Proceedings of the Tenth ACM Int. Conf. on Future Energy Systems*, ACM, 2019, pp. 436–438.

[69] S. Vandael *et al.*, “Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market,” *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1795–1805, 2015.

[70] Q. Wang, X. Liu, J. Du, and F. Kong, “Smart charging for electric vehicles: A survey from the algorithmic perspective,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1500–1517, 2016.

[71] Wikipedia. (2020). Electric Vehicle, [Online]. Available: `https://en.wikipedia.org/wiki/Electric_vehicle` (visited on 02/01/2020).

[72] M. Yilmaz and P. T. Krein, “Review of the impact of vehicle-to-grid technologies on distribution systems and utility interfaces,” *IEEE Transactions on power electronics*, vol. 28, no. 12, pp. 5673–5689, 2013.

[73] Y. Yue and H. M. Le, *Imitation learning*, online, Tutorials of the Thirty-fifth Int. Conf. on Machine Learning, 2018.

[74] J. Zhao *et al.*, “Risk-based day-ahead scheduling of electric vehicle aggregator using information gap decision theory,” *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1609–1618, 2017.

[75] A. Zishan, M. Moghimi, and O. Ardakanian, “Adaptive control of plug-in electric vehicle charging with reinforcement learning,” in *Proceedings of the 11th Int. Conf. on Future Energy Systems*, ACM, 2020.

[76] ——, “Reputation-based fair power allocation to plug-in electric vehicles in the smart grid,” in *Proceedings of the 11th Int. Conf. on Cyber-Physical Systems*, ACM/IEEE, 2020, pp. 63–74.