Learning to Control Home Batteries in the Smart Grid

by

Baihong Qi

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

Master of Science

Department of Computing Science

University of Alberta

Abstract

Modern residential buildings are complex cyber-physical systems housing energy systems with numerous sensors and actuators. In recent years, the falling costs of battery storage and photovoltaic systems have substantially increased the number of "solar-plus-battery" installations in these buildings. The solar-plus-battery system enables homeowners to protect their homes during a power outage and save on their electricity bills by stacking multiple value streams that battery storage can provide (e.g., energy arbitrage, and maximizing the self-use of solar power). However, controlling a solar-plus-battery system is quite challenging, mainly due to the wide range of variability and uncertainty associated with the building energy demand, electricity price, and meteorological factors affecting solar generation. Physical constraints, such as energy and power ratings of the lithium-ion battery and solar micro-inverter, only exacerbate this problem.

This thesis aims to investigate how to model the solar-plus-battery system and the stochastic environment, and how to design learning-based control policies for operating batteries in the smart grid to cut the monthly electricity bills for customers. We seek to develop control policies that are adaptive, optimal, and suitable for real-world applications.

We model different components of the solar-plus-battery system based on first principles, and the surrounding environment utilizing historical data about building energy demand, electricity price, and weather condition in Chapter 3. In particular, we develop and evaluate various supervised learning

models for predicting the available solar energy and household demand over the next 24 hours. We propose four learning-based methods for the *optimal control* of the solar-plus-battery system, under various operating conditions in Chapter 4, and study their effectiveness in terms of maximizing the revenue of homeowners. The control methods developed and discussed in this thesis are Model Predictive Control (MPC), Advantage Actor-Critic (A2C), Proximal Policy Optimization (PPO), and Direct Learning-based Control (DLC) using a neural network. The battery control is optimal in the sense that it minimizes the monthly electricity bills for customers. We implement these algorithms and integrate them into EnergyBoost, a Python program that runs on a Raspberry Pi and controls the battery. This allows us to compare their performance with specific baselines under various pricing schemes.

Experiments presented in Chapter 5 are based on real traces of solar irradiance and power consumption of 70 homes located in the same jurisdiction. We investigate how these sophisticated control policies compare with simple policies that are being used today to control battery storage systems. We further study whether it makes sense economically to install a battery controlled by the proposed algorithms in different jurisdictions with distinct tariff structures.

Preface

This thesis is original work of Baihong Qi. Some results presented in this thesis are based on a conference publication which is co-authored by the author of this thesis: "Baihong Qi, Mohammad Rashedi, Omid Ardakanian, *EnergyBoost: Learning-based Control of Home Batteries*, In Proceedings of the 10th ACM International Conference on Future Energy Systems (ACM e-Energy), pp.239-250, June 2019"

As the first author, Baihong was responsible for developing supervised learning models, modelling the optimization problem in CVXPY, implementing learning-based control techniques and baselines, and carrying out simulations to obtain the results. The second author, Dr. Mohammad Rashedi, performed the return-on-investment calculations and built the battery degradation model. The third author, Dr. Omid Ardakanian, formulated the mixed integer linear program and edited the paper.

Acknowledgements

I would like to thank my supervisor, Professor Omid Ardakanian, for being a good mentor, teacher and friend. Besides reviewing this thesis, he has given me invaluable advice during my graduate studies. I am impressed by his talent, diligence and patience. Being a student of him is one of the most valuable experiences in my life.

I would also like to thank my parents for their endless support.

Contents

1 Introduction **1**
- 1.1 Strategies for Controlling a Home Battery . . . 2
- 1.2 Challenges . . . 4
- 1.3 Objectives & Contributions . . . 6
- 1.4 Outline of the Thesis . . . 8

2 Related Work **9**
- 2.1 Control Objectives . . . 9
  - 2.1.1 Cutting the Electricity Bill . . . 9
  - 2.1.2 Increasing the Battery Lifespan . . . 12
  - 2.1.3 Supporting the Microgrid . . . 14
  - 2.1.4 Peak Shaving . . . 14
  - 2.1.5 Shaping the Demand of a Neighbourhood . . . 15
- 2.2 Control Methods . . . 15
  - 2.2.1 Linear Programming . . . 15
  - 2.2.2 Model Predictive Control . . . 17
  - 2.2.3 Dynamic Programming . . . 17
  - 2.2.4 Reinforcement Learning . . . 18
  - 2.2.5 Supervised Learning . . . 21

3 Modelling **23**
- 3.1 System Architecture . . . 23
- 3.2 System Models . . . 25
  - 3.2.1 Battery Model . . . 25
  - 3.2.2 Solar Inverter Model . . . 26
- 3.3 Environment Models . . . 26
  - 3.3.1 Data Sets . . . 27
  - 3.3.2 Data Prepossessing . . . 28
  - 3.3.3 Feature Selection . . . 29
  - 3.3.4 Overview of Supervised Learning Models . . . 29
  - 3.3.5 Choosing the Best Model . . . 34

4 Optimal Control Methods **37**
- 4.1 Optimization Problem . . . 37
  - 4.1.1 Constraints . . . 38
  - 4.1.2 Mixed Integer Linear Program . . . 39
- 4.2 Model Predictive Controller . . . 41
- 4.3 Sample-Based Predictive Controller . . . 42
  - 4.3.1 Feature Representation . . . 44
  - 4.3.2 Linear Function Approximation . . . 45
  - 4.3.3 Simulator . . . 46
  - 4.3.4 Actor-Critic Algorithm . . . 47
  - 4.3.5 Proximal Policy Optimization Algorithm . . . 47

4.4 Direct Learning-based Controller (DLC) 50
4.5 Rule-based Controllers 50
4.5.1 Performing Tariff Optimization (RBC-T) 51
4.5.2 Maximizing Self-use of Solar Energy (RBC-S) 51

5 Experimental Results **53**
5.1 Evaluation 53
5.1.1 Scenarios 53
5.1.2 Renewable Energy Initiatives 54
5.1.3 Evaluation Metrics 55
5.2 Results 55
5.2.1 Effect of the System Size 57
5.2.2 Pricing Schemes 60
5.2.3 Financial Analysis 62
5.2.4 Practical Considerations 65

6 Conclusion **68**

References **71**

List of Tables

3.1 Time-of-use prices ($/kWh) 28

5.1 EnergyBoost's estimated cost breakdown 63

List of Figures

1.1 Power producers, consumers, and prosumers in the context of a solar-powered home equipped with a battery. 3

2.1 Interaction between the critic, the actor, and the environment [71] 21
2.2 Structure of neural networks 22

3.1 A grid-connected home with behind-the-meter rooftop PV and battery storage controlled by a system called EnergyBoost. Solid arrow represent the direction of power flow and dashed arrows represent the direction of data/control flow. 24
3.2 nRMSE of the next 24-hour home load and PV output predictions using different models. Error bars represent one standard error. 35
3.3 nRMSE of PV output and home load predictions over the next 24 hours averaged over all homes. 35

4.1 Episode reward versus the number of episodes. 10 runs of PPO are shown using different colors. 49
4.2 Structure of DLC neural networks 51

5.1 Comparing annual bills obtained by different controllers in an example home with a 4.4kWp PV system and a Tesla Powerwall battery (left column: Model 1; right column: Model 2). The solar tariff is 0.03$/kWh (top row) and 0.154$/kWh (bottom row). 56
5.2 Comparing different policies in an example home with a 4.4kWp PV system and a Tesla Powerwall 1. The solar tariff is 0.03$/kWh. The on-peak and mid-peak intervals are highlighted in red and yellow. The lower plot shows polices during the same week. It is not overlaid on the upper figure for legibility. 58
5.3 Distribution of annual bills obtained by MPC for different sizes of the PV system and battery. The caption shows the solar export tariff in each case. 59
5.4 Comparing distributions of the annual electricity bill of homes equipped with a 8.8kWp PV system under hourly pricing scheme and solar export tariff of 0.03$/kWh. 62
5.5 Distributions of annual electricity bill of homes equipped with a 4.4kWp PV system and a Tesla Powerwall 1 under TOU pricing scheme. The solar export tariff is 0.03$/kWh, 0.061$/kWh, 0.077$/kWh, and 0.154$/kWh (in order). 65
5.6 Violin plot for the payback period (# years) of homes when the solar export tariff is 0.154$/kWh. 67

5.7 Violin plot for the payback period (# years) of homes when the solar export tariff is 0.077$/kWh. 67

Chapter 1

Introduction

Solar power is the fastest-growing source of renewable energy worldwide. The installed cost of solar power has fallen dramatically over the past decade due to the continuing decline in photovoltaic (PV) module and inverter prices, improved module efficiency, and lower labor cost. A recent study reports a 61% reduction in the residential PV system cost (from US$7.24 to US$2.8 per Watt DC) from 2010 to 2017 [53]. This along with renewable energy subsidies has encouraged homeowners to install their own rooftop PV systems or lease their roofs to companies that install and operate PV systems. Despite the rise in residential rooftop solar installations, homeowners do not currently utilize their PV system to the fullest extent. This is because solar generation usually peaks around noon and does not always coincide with the peak demand period when the aggregate household demand is the highest. Hence, solar power could exceed the local electricity consumption at times. The excess production must be exported back to the grid at a predetermined rate — as in *net metering* and *feed-in-tariff* programs[1] — if it cannot be stored locally. This export rate does not currently reflect the varying value of solar power, which depends on the time and location of its production.

With the rapid decline in the price of battery storage [54][2], homeowners increasingly consider installing batteries as a secure and lucrative investment opportunity. Combined with rooftop PV, home battery storage offers even

[1] These programs are introduced in Section 5.1.2.

[2] The cost per kilowatt-hour of battery storage is expected to fall between 50% and 60% by 2030 [33].

greater cost-reduction potential. In particular, the surplus solar power can be stored in the battery to serve the home load during peak demand periods. Moreover, the battery can be charged during off-peak times when electricity is cheaper and discharged during peak times when it is most expensive. This highlights the need for a control strategy to maximize the benefits offered by the solar-plus-battery system to the homeowners.

1.1 Strategies for Controlling a Home Battery

As the number of residential *solar-plus-battery* installations increases in the smart grid, the need for optimal and adaptive control strategies for the battery grows. Figure 1.1 shows a solar-plus-battery system installed in a home which is mainly supplied by the power distribution grid. This system consists of a PV system, a solar micro-inverter, a lithium-ion battery, and a battery inverter. The arrows indicate the direction of power flow between different power generators, consumers, and prosumers which can consume and produce power. We consider two types of generators, namely the rooftop solar energy system and the grid which collectively represent various types of conventional power plants that are connected to the grid. The lithium-ion battery is a prosumer because it can inject and consume power depending on its operation mode. The domestic load is the only consumer we have in this figure.

The energy management system, which is the core of the solar-plus-battery system and is depicted in the middle of Figure 1.1, decides on the charge or discharge power of the battery based on the available data (i.e., real-time and historical measurements). The battery can be charged with renewable power from the rooftop solar system or with conventional power from the grid. The conventional power is purchased from the grid at a predetermined rate which varies over time in many jurisdictions. The battery can also be discharged to meet the demand of the domestic load or export energy back to the grid at another rate, determined the solar export tariff. The unmet demand of the domestic load (which cannot be currently met by solar generation and battery) is always supplied from the grid.

The control algorithm is executed by the energy management system to determine the charge or discharge power of the battery. The objective we consider in this work is to minimize the monthly electricity bill of the homeowner. Given that the demand and supply must in balance at all times:

$$HouseholdDemand + BatteryCharge = Grid + Solar + BatteryDischarge$$

and that the household demand and solar generation are either observed or predicted, determining the charge or discharge power of the battery yields the amount of power that must be imported from the grid. Thus, a separate control variable is not needed for the amount of power purchased from the grid.

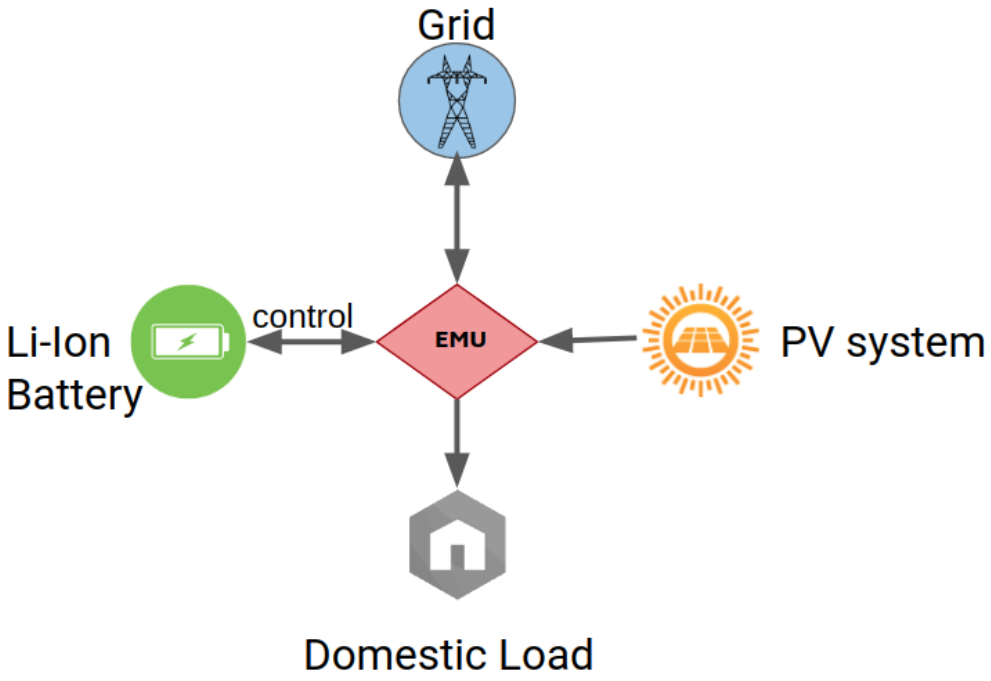


Figure 1.1: Power producers, consumers, and prosumers in the context of a solar-powered home equipped with a battery.

The optimal control of a battery energy storage system to reduce the homeowner's monthly electricity bill is an example of a constrained and continuous control problem which can be solved using several techniques we study and benchmark in this thesis. Rule-based control is the most commonly used method for operating a battery in this context. The main benefit of this control

strategy is that the controller can instantly respond to changes in the environment since updating the control variable does not require any sophisticated computation [30]. Nevertheless, rule-based control is typically sub-optimal because it utilizes simple rules to decide on the battery operation without taking forecasts into account. Another control strategy is to cast the problem into a linear optimization problem considering the system constraints [42]. The optimal control strategy is then found by solving this optimization problem given the day-ahead forecasts. Supervised leaning techniques, such as neural networks [35], have been recently employed to learn the optimal control of the system. To train such a model, the idea is to use the optimal control found offline by solving an optimization problem. Thus, this model maps input variables (e.g., home load and solar generation) to the optimal control. Besides these techniques, some attempts have been made to apply a Reinforcement Leaning technique to solve this continuous control problem [23], but they don't consider the diversity of buildings and pricing schemes.

1.2 Challenges

The optimal control of battery charge and discharge operations (to reduce the homeowners' electricity bill) in the presence of stochastic demands and supply sources is a complex and difficult problem due to several reasons. First, solar generation and household demand are highly variable and difficult to predict with high accuracy in advance. For example, the amount of solar power that can be generated depends on several factors, including the cloud cover, wind speed, temperature, time of the day, and season. On a cloudy day, the PV output might peak at a time other than the solar noon. This variability introduces noise to predictions which are used by a receding horizon control technique like model predictive control. It could also make the environment non-stationary, making it difficult for reinforcement learning techniques to converge to the optimal policy. Myopic control of this system would also lead to sub-optimal operation. For example, the battery may get full (or depleted), thus it cannot be charged (or discharged) further in the future.

Solar energy may not be available during the day (for example due to the effect of passing clouds) and the electricity price may also change over time, especially in locations with an hourly pricing scheme.

Second, the lithium-ion battery has restricted capacity, and charge and discharge power rates. These physical constraints together with battery imperfections, such as self-discharge and charge/discharge efficiency, complicate the optimal operation of the battery. While in an ideal situation, the total household demand over the peak hours would be stored in the battery during off-peak hours, in reality the total amount of energy a home consumes during the peak hours can exceed the effective capacity of the battery[3].

Third, the charge or discharge power of a lithium-ion battery is a continuous variable restricted by a set of time-varying constraints. The time-varying constraints are introduced because the remaining capacity of the battery changes over time and the battery cannot be charged or discharged past certain limits. Moreover, the difference between the buy and sell prices, and battery imperfections make the problem non-convex and difficult to solve through model predictive control or reinforcement learning. The model predictive controller has to solve a convex relaxation of this problem and the reinforcement learning agent may not converge to the optimal policy in a polynomial number of episodes. Tackling this problem is much harder than a convex optimal control problem with a finite set of actions.

Fourth, with time-of-use pricing and feed-in tariffs, there are multiple value streams the battery can provide, such as promoting solar self-consumption and taking advantage of time-of-use pricing. These value streams must be traded off against one another to maximize the homeowner's revenue. Due to the intrinsic difficulty of modelling various sources of uncertainty and the physical constraints of batteries and solar inverters in a computationally efficient way, most home batteries are currently controlled using simple rule-based control mechanisms [30], which are myopic and suboptimal. Existing rule-based con-

[3]The price of a Tesla Powerwall is around US$3,000 for the 7kWh battery [75], while an average home consumes about 24kWh per day in North America, about half of which is consumed during off-peak times as discussed in Chapter 2.

trol strategies often lead to suboptimal battery operation. At higher penetration levels, such mechanisms can contribute to distribution network problems (e.g., voltage sag and swell, and reverse power flow) and might even increase the peak demand. Furthermore, customers and system operators often have competing objectives which cannot be satisfied at the same time using simple control strategies.

1.3 Objectives & Contributions

In this thesis, we explore how to model the stochastic environment (i.e., weather, household demand, electricity price) and different components of the system depicted in Figure 1.1 by incorporating their operating limits and imperfections. We specifically adopt physical models that closely approximate the output of a solar inverter [29] and the state of charge (SoC) of a lithium-ion battery [36], where the SoC is defined as the ratio of the energy stored in the battery to its capacity. The first objective of this thesis is to study how the operation of a solar-plus-battery system can be controlled using learning-based methods (which may utilize the developed models) to effectively reduce the homeowner's electricity bill and investigate whether these methods outperform simple rule-based control methods that have been widely used in practice. The second objective of this thesis is to explore whether it makes sense economically to install this system in different jurisdictions, assuming that it is controlled by a specific algorithm, and whether the payback period of this investment can be further reduced by improving the control method. The contributions of this thesis are as follows:

- Developing and evaluating various models obtained by supervised learning for the environment, which are trained using the available historical data, in addition to physics-based models for the solar-plus-battery system. The environment model is used to predict the available solar energy and household demand in the future.
- Formulating the optimal control of the lithium-ion battery's charge and discharge operations as a mixed integer linear program (MILP) in CVXPY,

which is a Python-based convex optimization framework [14], and solving it using a solver (Gurobi [24]). This problem is solved assuming that there is an *oracle*, providing perfect information about the future (i.e., solar productions, household demands, and tariffs). This optimal control is used to evaluate the learning-based control algorithms.

- Developing four learning-based control algorithms for operating the battery in order to leverage the physics-based and data-driven models of the system and optimally control the solar-plus-battery system to save more than rule-based controllers in eletricity bills . These algorithms are Model Predictive Control (MPC) with a time horizon of 24 hours that leverages the physics-based and data-driven models of the system and the environment, Advantage Actor-Critic (A2C) and Proximal Policy Optimization (PPO) which try to learn the optimal control policy by interacting with the system in a simulated environment, and Direct Learning-based Control (DLC) which uses a neural network to learn the mapping from the state variables to the control action directly.
- Implementing the proposed learned-based controllers and the two baseline (rule-based) controllers in Python, and evaluating them on a Raspberry Pi in terms of their ability to reduce the average monthly electricity bill of homeowners. A software package, called EnergyBoost, is created for this purpose and is available at `https://github.com/sustainable-computing/EnergyBoost`. The evaluation is carried out using real electricity demand of 70 homes in the city of Austin in Texas, and weather data from a nearby site.
- Analyzing the financial feasibility of a suitably sized solar-plus-battery system when it is controlled using one of the proposed algorithms in different jurisdictions which are characterized by their pricing schemes. This analysis involves calculating the return on investment (ROI) and finding the break-even period.

1.4 Outline of the Thesis

Chapter 2 surveys related work on different methods for controlling a home battery and control applications that have been considered in the literature. Chapter 3 introduces the system model (i.e., the lithium-ion battery and solar micro-inverter) derived from first principles and the environment model (i.e., the household demand and solar irradiance) developed using historical data. Several supervised learning models are proposed for modelling how the environment changes over time. Chapter 4 formulates the optimal battery control problem as a non-convex constrained optimization problem, explains how it can be solved assuming perfect information about the future, and proposes four learning-based and two baseline rule-based methods for controlling the charge and discharge operations. Chapter 5 describes our simulation scenarios and a set of metrics that are used for evaluation. Furthermore, it presents the performance evaluation results for different system sizes and tariff structures, and discusses the economic feasibility of installing solar-plus-battery systems in different jurisdictions. Chapter 6 explains limitations of this work, summarizes the contribution of this thesis, and provides avenues for future work.

Chapter 2

Related Work

The optimal control of energy storage has been extensively studied in the past in the context of a residential building, a small neighbourhood, or the power distribution system using various optimization and control techniques. This chapter summarizes related work on controlling energy storage systems. Section 2.1 introduces the objectives that are considered in previous work for operating a battery. Section 2.2 discusses different methods that have been employed to control energy storage, including convex programming, model predictive control, dynamic programming, reinforcement leaning, and supervised learning.

2.1 Control Objectives

Related work on controlling an energy storage system falls into different categories based on the control objective. These objectives are (a) reducing the customer's electricity bill, (b) supporting the power distribution grid or microgrid operation, (c) shaving or shifting the peak demand, and (d) shaping the aggregate demand of a neighbourhood comprised of a small number of homes. Note that the first objective is the one considered in this thesis.

2.1.1 Cutting the Electricity Bill

Cutting the electricity bill is the most common objective when it comes to controlling an energy storage system installed in a home as it enables the homeowner to pay back the initial investment. Most related work assumes perfect

information about the future when formulating an optimization problem which is solved several hours in advance to identify the strategy that maximizes the revenue generated by distributed energy storage co-located with PV systems.

Babacan, et at. [5] develop a scheduling algorithm based on convex optimization for charging or discharging distributed energy storage co-located with solar PV systems. The optimization-based scheduling algorithm incentivizes self-consumption of solar generation using a new supply charge tariff. The proposed algorithm minimizes the customer energy costs while providing ancillary services to the grid. Comparing this algorithm against two algorithms proposed in the literature reveals that the proposed algorithm can successfully restrict the reverse power flow without increasing the customer energy costs. Ratnam, et at. [65] formulate a quadratic program to maximize the daily operational savings that accrue to customers, while penalizing voltage swings in the power grid. The authors have shown through simulations that their algorithm penalizes reverse power flow and peak demands.

Kazhamiaka et al. [34] study the profitability of residential solar-plus-storage systems in three different jurisdictions in Germany, Canada, and United States, considering various factors such as solar radiation and typical residential load profiles, the system's installed cost, electricity pricing schemes, and government incentives. The authors set up an integer linear program to determine the battery operation policy that maximizes the 20-year return on investment and explore how the choice of a jurisdiction can affect profitability of solar-plus-storage systems.

These papers do not take the uncertainty of renewable generation and electricity consumption into account, assuming perfect information and accurate forecasts. We address this shortcoming by building data-driven models to predict the future household demand and solar generation.

Leveraging Predictions

Residential solar-plus-battery systems can fully utilize the renewable energy produced, while reducing the household electricity bill, especially for customers who have surplus solar production that can be exported back to the grid.

In this respect, incorporating the predicted solar generation and household demand can further increase the cost saving.

Solar generation and household demand are stochastic processes, and modelling and predicting them has been the focus of several studies. In [23], TD(λ)-learning is employed to minimize the homeowner's electricity bill by taking an action that yields the best expected reward. The proposed reinforcement learning algorithm is deemed model-free as it does not need models to accurately predict the household demand and solar generation. Even then the system model is required to estimate the remaining energy in the energy storage system. Despite the novelty of this work, the authors ignore several important system constraints which could make the learned policy infeasible. Another shortcoming of this work is that the obtained results are not compared with other control strategies.

In another line of work [15], a reinforcement learning algorithm that takes advantage of neural network function approximators is used to minimize the amount of energy (and therefore the bill) received from the grid. This work relies on the data obtained from only a single home and develops a finite-horizon Markov decision process to determine an optimal control policy for cost minimization. However, it does not consider various pricing schemes nor performs return-on-investment (ROI) calculations. Furthermore, their system does not include a PV system.

Reference [30] uses dynamic programming to determine the optimal battery operation schedule over a finite horizon, and converts this optimal schedule to simple rule-based controllers, each representing a particular value stream that battery storage can offer. Despite the novelty of this approach, the authors do not thoroughly evaluate the performance of rule-based controllers on a large number of homes with different system sizes and pricing schemes. Stochastic dynamic programming is also employed in [1] to optimize battery operation in a receding horizon while taking the battery lifetime into account. This work complements the work presented in this thesis in that it mainly focuses on optimizing the lifetime of the battery under a specific tariff structure, and does not compare its proposed controller with the optimal controllers.

SmartCharge [42] predicts the next day energy usage and determines the energy storage control strategy for minimizing the bill given the next day electricity prices. The problem is formulated as a linear optimization problem using the next day price and energy consumption forecasts, and is solved only once a day. In contrast, the model predictive controller we present in this thesis works with predictions that are updated at the beginning of each time interval of 24 hours. Moreover, their model neither includes a PV system nor considers the possibility of selling stored energy to the grid. Reference [43] is an extension of [42] which incorporates distributed solar generation and predicts the available solar energy and demand. The authors consider different electricity pricing schemes for solar generation and energy storage to incentivize distributed generation.

Reference [58] uses the day-ahead pricing scheme for scheduling appliances to reduce the electricity bills. It considers energy storage, solar generation, and electric vehicles. Reference [40] studies the optimal control of energy storage when there is local renewable generation and it is possible to buy/sell electricity from/to the grid. It develops a real-time control strategy based on Lyapunov optimization to determine a time-averaged solution over a 24-hour time horizon. But simulation results are insufficient to show the value of this controller in real-world scenarios.

Most of these studies are limited in that they explore the problem given a specific tariff structure. Comparing to these approaches, in this thesis we take a large number of factors into account, such as battery imperfections, rated capacity and charge/discharge powers of the battery, the solar inverter model, and consider different system sizes and multiple tariff structures. Specifically, we evaluate our method using both time-of-use, hourly, and tiered pricing schemes to determine the cost saving potential in each case.

2.1.2 Increasing the Battery Lifespan

A variety of operating factors could impact the battery lifespan, i.e., degrading its usable capacity over time. However, battery degradation is often not considered when solving the optimal control problem. This is because bat-

tery degradation is a highly nonlinear process governed by several factors, and developing a closed-form cost function suitable for convex optimization is a challenging task.

A battery cycle life estimation method would be beneficial for various applications. In [79], the cycle life of lithium-ion batteries is estimated using a combination of infrared thermography and supervised learning techniques, including artificial neural networks (ANNs) and support vector machines (SVMs). It is found that the ANN can estimate the current cycle life with less than 10% error in less than 10 minutes.

Reference [37] defines a degradation cost function for optimal control of a battery energy storage system. The battery degradation is parameterized by the Depth of Discharge(DOD), which is the fraction of the capacity which has been removed from the fully charged battery, the charge rate, and the state of charge and is incorporated into the cost function. This model is suitable for arbitrary battery load patterns and captures nonlinearities of the battery, making it an appropriate cost function for mixed-integer quadratic programming or model predictive control. Reference [49] presents a battery life prediction methodology to optimally control a battery. The proposed methodology can incorporate a multitude of dynamically changing cycling parameters considering the following factors: charging and discharging currents, minimum and maximum cycling limits, and the operating temperature. The authors develop four independent models which are customized using experimental battery data. Finally they implement the methodology in different applications to maximizing the benefits offered by lithium-ion batteries.

More dynamic models are introduced in [1]. This work proposes a stochastic dynamic programming approach based on the rainflow counting algorithm, which extracts closed cycles of battery operation, to approximate the battery degradation and optimally operate an energy storage system over some time horizon. The proposed method utilizes energy storage to deliver maximal lifetime value, taking into account the operational impacts and several other factors. The authors find that an average residential customer operating the battery using the proposed algorithm could increase the lifetime of the battery

by 160%. Shi, et at. [69] compute battery degradation as a complex material fatigue process based on different stress cycles. They prove that the rainflow cycle-based cost is convex. This convexity result allows for the battery degradation model to be incorporated in different optimization problems. Moreover, the authors provide a subgradient algorithm minimizing a non-differentiable convex function to study the effectiveness of the proposed degradation model in maximizing the battery's operating cost and its lifetime.

The above lines of work mainly focus on operating battery energy storage systems so as to maximize the customer's profit and the battery lifetime at the same time. But they do not study the possibility of selling the stored energy back to the grid and do not incorporate accurate models of the system and the environment in the design of optimal controllers.

2.1.3 Supporting the Microgrid

Battery storage can offer many other benefits apart from cutting the elasticity bill of customers in the smart grid. Several attempts have been made to study energy sharing in a microgrid [32], [48], [76], [80]. Owing to the success of solar-plus-battery systems, many studies focus on how to operate a battery so that the home can disconnect from the grid and be supplied by the solar power generated locally at all times. Reference [62] provides an assessment of solar self-consumption with respect to solar PV and battery requirements in different regions and provides a database of household profiles. It also develops a simulation tool to predict self-consumption and optimally size such systems.

2.1.4 Peak Shaving

There is a growing body of work on demand side management strategies to incentivize residential customers to consume less energy during peak hours. This includes price-based methods [47], [63], and direct load control strategies for operating energy storage systems [44], plug-in electric vehicles [4], and thermostatically controlled loads [41].

In [31], the authors design a load control system called Smart Home Energy Management System (SHEMS) to achieve dynamic price response considering

both the interests of residential customers and the grid. They utilize sensor data to predict activities which are then used to determine a strategy for alleviating the peak demand. In [57] the optimal battery storage capacity for peak load shaving is investigated.

2.1.5 Shaping the Demand of a Neighbourhood

Reference [19] studies load scheduling in a local neighborhood by formulating the problem as a distributed constraint optimization problem (DCOP). It schedules the time of use of specific appliances from multiple smart homes so as to minimize the energy use overlaps, thereby reducing the aggregate demand during peak hours. The main limitation of this approach is that the demand can only be shifted based on the number of elastic (*i.e.*, controlled) loads which can be rescheduled without having any impact on their operation or user comfort. Our approach achieves the same goal of cutting the electricity bill by controlling the battery charge and discharge operations.

2.2 Control Methods

In this section we introduce the methods that have been used to control solar-plus-battery systems In particular, we focus on linear programming, model predictive control, dynamic programming, reinforcement learning and deep learning, and explain how each method can be applied to solve a control problem.

2.2.1 Linear Programming

Convex optimization is a type of mathematical optimization which concerns minimizing (maximizing) a convex (concave) function over a convex set. Linear programming is a special case of *convex optimization* where the objective function is linear and the constraints are of linear equality and inequality

forms [66]. A standard linear program can be expressed as follows

$$\min_{x} \quad c^\top x \tag{2.1}$$

$$\text{subject to} \quad a^\top x \leq b, \forall i \in \{1, ..., m\} \tag{2.2}$$

Here $x \in \mathbb{R}^n$ is the vector of decision variables, and $c, a_1, \cdots, a_m \in \mathbb{R}^n$ and $b_1, \cdots, b_m \in \mathbb{R}$ are constant parameters. A linear program can be solved quite efficiently as it is a convex problem. Specifically, there are many polynomial-time algorithms and solvers for tackling linear programming problems, e.g., simplex, ellipsoid [22] and interior point method [2].

Given the vast number of algorithms and solvers that exist for linear programming, the key challenge is to cast an optimization problem into a linear program. This involves identifying the decision variables of the problem, writing the objective function in the linear form, and defining the constraints. Decision variables are the variables that will determine the objective, which could be the electricity bill of a customer as in [42]. Constraints are restricting the decision variables. Often times multiple transformations have to be performed to ensure that the problem is linear and convex.

Linear programming is widely used and appears in many problem areas, such as telecommunications networks [55], cellular networks [18] and power systems [61]. Controlling a solar-plus-battery system can also be formulated as a linear program assuming perfect information about the future [42].

Integer linear programming is similar to linear programming but some or all decision variables are constrained to take on integer values. This makes the optimization problem non-convex. If only some of the decision variables are integer, the problem is called a mixed integer program (MIP), and when the objective function and constraints are also linear, it is called a mixed-integer linear program (MILP). Since MILP is a non-convex problem, we often solve a convex relaxation of it using the same algorithms and solvers that are built for linear programming. We note that the integer variables can be used to represent binary decisions, such as turning a system on or off.

2.2.2 Model Predictive Control

Model predictive control (MPC) is a method to solve a multivariable constrained control problem over a finite horizon [78]. It determines an optimal control action for the current time slot while taking future time slots into account. To this end, an optimization problem is solved repeatedly by predicting how the control decisions change the state variables using a dynamic model of the system. The first optimal control action is implemented and the same optimization problem is then solved for the next time slot.

When the system model is linear and the cost function is convex, MPC solves a convex optimization problem at every time slot. This optimization problem can be a linear program if the cost function is linear and constraints are affine.

MPC has become used extensively to control processes that appear in various applications, including the energy management. For example, MPC is used to control battery operations to maximize the power generated in a wind microgrid system [25].

2.2.3 Dynamic Programming

Dynamic programming is another polynomial-time method for solving optimization problems using recursion. It divides the problem into overlapping subproblems, and the results of these subproblems are combined to solve the original problem. Since subproblems are encountered multiple times, their solution is cached so that they do not need to be solved again. Reference [6] introduces dynamic programming and explains how it can be used for optimal control.

There are two approaches to understand and solve a dynamic programming problem. The first approach is called the top-down approach as it starts from the original problem and recursively solves smaller cases of the original problem until we have the results needed to solve the original problem. Another approach is called the bottom-up approach, which solves the problem in the opposite way: it solves the basic cases of the problem first and combines them

to solve bigger cases until the original problem is solved.

Dynamic programming could be used to solve a problem if it has two key properties. First, the optimal solution of the original problem can be determined by finding the optimal solution of its subproblems. Second, the number of unique subproblems is polynomial rather than exponential. A variant of dynamic programming known as stochastic dynamic programming is used in previous work to control battery storage under uncertainty. For example, it is used in [1] to operate a battery considering its degradation cost and in [30] to find the optimal control of a battery.

2.2.4 Reinforcement Learning

Reinforcement learning (RL) is an area of machine learning which involves learning through trial-and-error interaction with the environment based on a 'reward' signal which is determined based on the observations from the environment [71]. The decision maker is called *agent* and everything else it interacts with is called *environment*. An action of the agent takes it from the current state to the next state, and it receives a reward from the environment that corresponds to this action. The goal of the agent is to find a policy that maximizes its expected total reward over some time horizon.

A reinforcement learning problem can be formulated as a Markov Decision Process (MDP). The set of all states of the environment is denoted by $\mathcal{S}$ and the set of all action at one state is denoted by $\mathcal{A}$. The transition probability, denoted by $p(S_{t+1} = s'|S_t = s, A_t = a)$, is the probability of going from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ after taking action $a \in \mathcal{A}$. This transition results in a reward denoted by $r(s, a, s')$. The agent takes an action $\mathrm{A_t}$ at state $\mathrm{S_t}$, which takes it to state $\mathrm{S_{t+1}}$, and gains the reward $\mathrm{R_{t+1}}$. A policy, π, is what the agent learns; it is a function that maps states to actions. We denote the expected *discounted return* of a policy as the sum of discounted rewards the agent receives over time, that is:

$$G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k,$$

where $0 \leq \gamma \leq 1$ is the discount rate, determining the present value of future rewards. The agent becomes more farsighted as γ approaches 1. To evaluate π, we define a value function which is the expected sum of discounted rewards of that policy computed by the agent to determine which actions are best to take in which states:

$$V_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s].$$

The optimal policy is the one with the highest expected sum of discounted rewards:

$$V^*(s) = \max_\pi V_\pi(s).$$

When the number of states or actions is large, evaluating all state-action pairs becomes computationally expensive. To address this, the value function is estimated for each state based on a limited number of experiments in the environment. One way to learn the value function is using the temporal difference (TD) method, which updates the value function based on the difference between temporally successive predictions of the states. The simplest TD method, TD(0) updates the value function as described below:

$$V_{\mathrm{t}+1}(s) = V_{\mathrm{t}}(s) + \alpha\Big[r + \gamma V_{\mathrm{t}}(s^{'}) - V_{\mathrm{t}}(s)\Big],$$

when the agent takes action a which causes a transition from state s to state $s^{'}$ with reward r. In the above equation, α is the learning rate and γ is the discount rate. One of the most popular TD methods is Q-learning, which is an off-policy temporal difference learning algorithm. The agent learns the optimal policy using an absolute greedy policy, and behaves using other policies such as ϵ-greedy policy. Because the update policy is different from the behavior policy, Q-learning is called off-policy. The action at each state is chosen based on the state action value, and the value will be updated after a new reward is obtained. However, this algorithm has several shortcomings, especially when the state space or the action space is continuous. In particular, the value of each state-action pair must to be recorded in a table, which will be scanned later to look up or update a state-action value. It is impossible to store the values of all state-action pairs when the state space or the action space is continuous. When

the state space and action space are continuous, continuous state spaces may be discretized into a set of binary features, using a coarse coding method, or into a set of continuous-valued features by radial basis functions [71], and the value function is approximated as a linear combination of these features. Non-linear function approximation, for example, using artificial neural networks, may also be employed. In [20] Q-learning has been extended to deal with continuous states and continuous actions using a neural network coupled with a novel interpolator. Unfortunately continuous Q-learning also exhibits poor performance in a benchmark continuous control problem [39]. In general, a continuous action space may be discretized and dealt with using temporal difference methods such as Q-learning. It can also be dealt with directly using a policy gradient method to determine the policy. However, the problem with the policy gradient method is the noisy gradient and high variance, since the policy parameters are updated through random samples, introducing high variability in log probabilities (the logarithm of the policy distribution) and cumulative reward values. The high variability leads to noisy gradient and directs policy distribution to a non-optimal direction, thereby contributing to instability and slow convergence.

Actor-critic methods can address this problem. They are TD methods that represent the policy independent of the value function. As shown in Figure 2.1, a policy is included to choose the action. The policy is the actor part the algorithm; it selects an action based on the policy. Once the action is chosen, the critic part gets the TD-error, which evaluates the improvement compared to the average the action taken at that state, and updates the value function and the policy accordingly.

Since the policy is already stored, choosing the best action does not require going through the whole set of state-action values. This makes the actor-critic method efficient, especially when dealing with problems with continuous state and action spaces. In Chapter 4 we introduce two families of actor-critic methods, namely basic Actor critic (A2C) and Proximal Policy Optimization (PPO), and describe how these methods are applied to solve our continuous control problem.

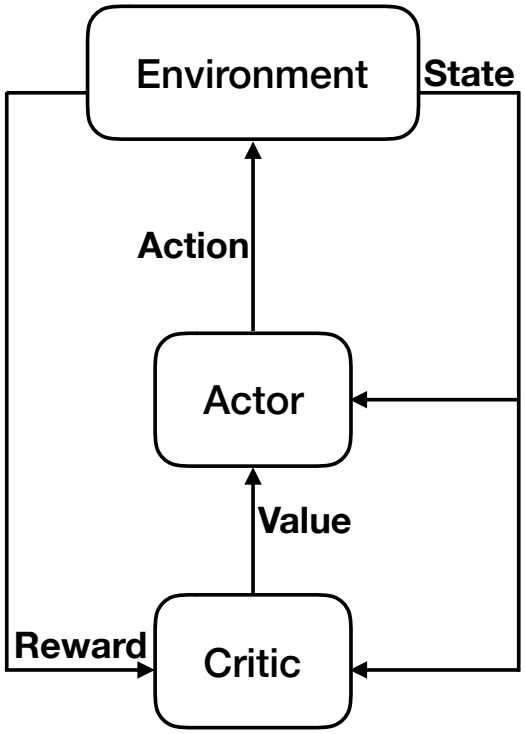


Figure 2.1: Interaction between the critic, the actor, and the environment [71]

Reinforcement learning can be applied to the problem of controlling a solar-plus-battery system. It is a well-suited approach because it can handle the uncertainty of the environment and can represent the customer's electricity bill as a cumulative reward. For instance, Reference [23] uses a reinforcement learning algorithm to control the battery in the power system.

2.2.5 Supervised Learning

Deep learning is a popular supervised learning technique based on an artificial neural network (ANN) which has several hidden layers. Reference [11] defines a neural network as "a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs". Figure 2.2 shows the structure of an example neural network. It is characterized with three kinds of layers, i.e., input layer, hidden layer, and output layer, where each layer is made up of interconnected nodes. The input layer takes in the input and sends it to hidden layers through weighted connections. The hidden layers are then con-

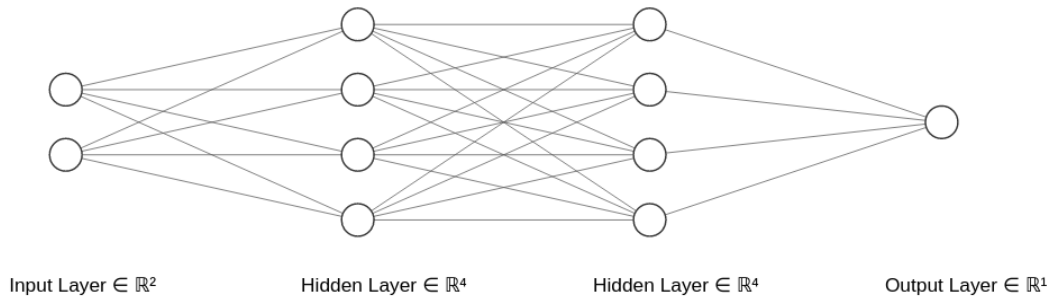


Figure 2.2: Structure of neural networks

nected to the output layer through weighted connections which produces the output. Training an ANN involves learning the weight of theses connections. There are several ways to learn these weights, one is called the Backpropagation rule [27] which helps to refine connections within networks, and modify the connection weights between inputs and outputs with layers of neurons. In a Backpropagation, when the network is given a new input, the connection weights are randomly initialized. It then gets the output by a forward activation, and the errors are used to readjust the weight. Backpropagatin utilizes a gradient descent towards a global minimum which is the solution with lowest possible error.

Training neural networks requires a large number of individual training runs to determine the best solution with the lowest error. The training rate is the rate of convergence to the global minimum with the training runs. Once a neural network is trained, it can be used as a model to perform prediction. Thanks to the predictive power of deep neural network models, they have been used in numerous applications, such as pattern recognition, playing game, and nonlinear system identification [7]. They have also been applied to various control problems, e.g., autonomous driving and process control. Reference [35] uses neural networks to learn an optimal control strategy of a battery energy storage system.

Chapter 3

Modelling

In this chapter, we discuss how we build physics-based models of the system and data-driven models of the stochastic environment. Section 3.1 presents the overall system architecture according to [60]. Section 3.2 introduces the models built for the lithium-ion battery and solar micro-inverter. Section 3.3 develops several supervised learning methods describing how the environment changes over time.

We note that time-dependent variables are denoted with a subscript t throughout this chapter.

3.1 System Architecture

Figure 3.1 depicts a grid-connected home with a rooftop PV system, a solar micro-inverter, a lithium-ion battery, and an energy management unit (EMU) which controls battery charge and discharge operations and monitors its state of charge (SoC). The rooftop PV system and battery are connected behind the home's standard meter which measures the amount of energy consumed (or exported to the grid). The micro-inverter converts the DC output of the PV system to AC. The EMU also comprises an integrated internal inverter performing AC/DC conversion for the battery. The home may have an additional meter installed in front of the PV system measuring the amount of renewable energy generated by the system. This meter is necessary for participating in some renewable energy programs.

The system runs on a smart home gateway, i.e., a personal computer or a

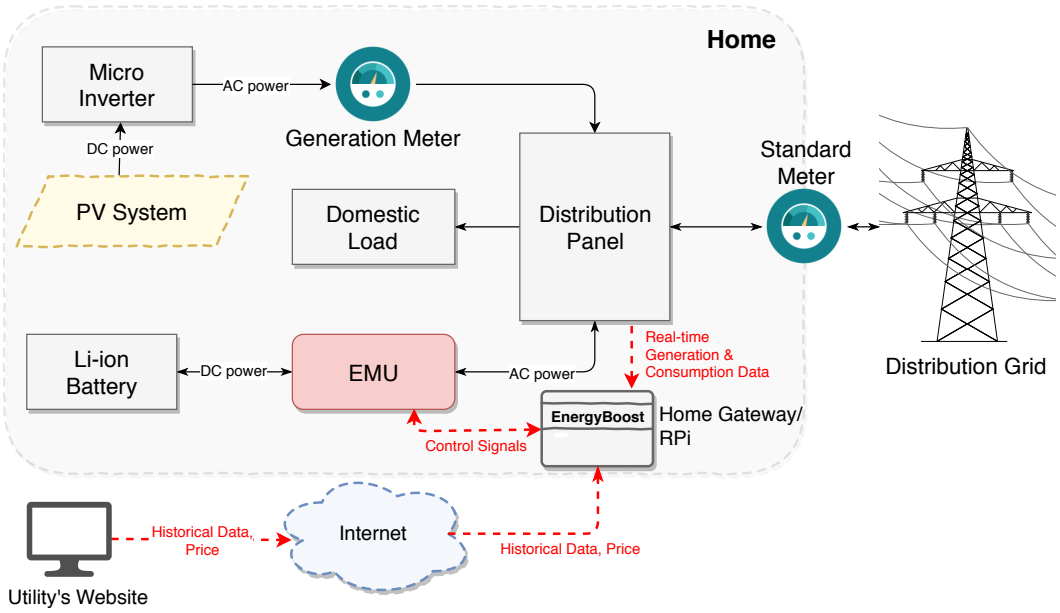


Figure 3.1: A grid-connected home with behind-the-meter rooftop PV and battery storage controlled by a system called EnergyBoost. Solid arrow represent the direction of power flow and dashed arrows represent the direction of data/control flow.

Raspberry Pi, which is responsible for home automation and runs optimization solvers and Python code. Additionally, the gateway logs generation and consumption data, gathers current and historical electricity and solar export prices from the utility's website, and computes the policy for the home battery by running one of the optimal control algorithms proposed in this work. Real-time solar generation and electricity consumption data can be pulled in directly from the meters or measured at the distribution panel through an energy monitor, e.g., an eGauge sub-metering device [17]. In the beginning of each time slot, the home gateway communicates a feasible control point for the time slot to the EMU system which adjusts the battery charge/discharge power in the best interest of the homeowner, i.e., reducing their electricity bill. We do not study the optimal control of the solar micro-inverter in this paper and assume that it simply runs the Maximum Power Point Tracking (MPPT) algorithm which extracts the maximum power from a PV module under different conditions.

3.2 System Models

3.2.1 Battery Model

We adopt the battery model introduced in [36] (Model 1*), which is a tractable and accurate linear approximation of the physical model of a lithium-ion battery[1]. Let B^{cap} be the capacity of the battery, $E_t = SoC_t \times B^{\text{cap}}$ be the energy content of the battery, AC_t be the battery charge power at time t, AD_t be the battery discharge power at time t, and T^u be the unit of time, i.e., the length of each time slot. According to this model, the energy content of the battery evolves as follows:

$$E_{t+1} = E_t(1 - \eta^{\text{p\_leak}}) + \Delta E_t - T^u \eta^{\text{c\_leak}}, \tag{3.1}$$

$$\Delta E_t = \begin{cases} T^u AC_t \eta^c, & \text{if charging,} \\ \frac{-T^u AD_t}{\eta^d}, & \text{if discharging,} \end{cases} \tag{3.2}$$

where $\eta^c, \eta^d, \eta^{\text{p\_leak}}, \eta^{\text{c\_leak}}$ denote respectively the charge efficiency, the discharge efficiency, the leakage rate per time unit as a fraction of the SoC, and the constant leakage rate.

We also denote the charge and discharge power ratings (i.e., maximum rates possible) of the battery by α^c, α^d ($\alpha^c, \alpha^d > 0$), and its minimum and maximum capacity by $E^{\min}$ and $E^{\max} \leq B^{\text{cap}}$, respectively. The battery operations are subject to the following constraints:

$$0 \leq AC_t \leq \alpha^c, \tag{3.3}$$

$$0 \leq AD_t \leq \alpha^d, \tag{3.4}$$

$$E^{\min} \leq E_t \leq E^{\max}. \tag{3.5}$$

The first two constraints ensure that the battery operation is within the maximum charge and discharge rates supported by the battery, while the third constraint ensures that the battery energy content cannot be outside a certain range, preventing overflow and underflow. We assume $E^{\min}$ and $E^{\max}$ are linear functions of the current, i.e., $\frac{|A_t|}{V^{\text{nom}}}$ where V^{nom} is the nominal charge/discharge

[1]The control methods proposed in this work can be extended to lead-acid batteries or other energy storage technologies by substituting (3.1) and (3.2) with an accurate model of the corresponding technology.

voltage and A_t is the battery charge or discharge power at t. The parameters of the battery are set according to the lithium-ion battery specifications as discussed in Section 5.1.1.

3.2.2 Solar Inverter Model

We use the PVLib Toolbox [29] to translate Global Horizontal Irradiance (GHI) readings to the inverter's AC output power on a particular day of the year in Austin, Texas. The Sandia PV array performance model (SAPM) is used to generate the PV module's direct current (DC) I-V curve. The model is built in [38] and is widely used in the PV industry. This model assumes the temperature of the photovoltaic cells is 25°. We can compute the DC power by applying Ohm's law to the I-V curve. To convert DC I-V curve to AC power, we adopt Sandia's grid-connected PV inverter model [8].

The power output of a given PV system with a specific size in Austin at time t, denoted by G_t, can be calculated as a function of global horizontal irradiance GHI_t, outdoor temperature $Temp_t$, and time of day ToD_t:

$$G_t = \mathcal{F}^I\Big(GHI_t, Temp_t, ToD_t\Big), \tag{3.6}$$

where $\mathcal{F}^I$ is a known, non-linear function defined in PVLib. Other parameters of the PV module are specified in Section 5.1.1.

3.3 Environment Models

Considering the stochasticity of the environment due to intermittent weather conditions and fluctuations in household demand, the solar-plus-battery system relies on predictions of the future home loads and solar productions to determine a sequence of actions over some time horizon. These actions should result in the lowest electricity bill. Since we do not have the knowledge of the future at a decision-making epoch, we must develop models to predict the future so that we can find the optimal control. These models are different from the battery and the solar micro-inverter models which were closed-form, physics-based expressions for the evolution of home electricity consumption

and solar generation. We use historical data about electricity consumption of a home to develop supervised learning models for predicting its future electricity consumption. Similarly, we use historical data about the output of a PV inverter to predict its future output power.

In [42] the authors use different models to predict the future, including Exponentially Weighted Moving Averages (EWMA), Linear Regression (LR), and Support Vector Machines (SVMs) with various kernel functions, which take data as input and transform it into the required form including Linear, Polynomial, and Radial Basis Function (RBF) kernels. In addition to these models, we develop other supervised learning and time-series models in the following.

3.3.1 Data Sets

We use four public data sets to train and validate the models that are necessary for developing learning-based battery control strategies. These data sets contain real world traces of household electricity consumption, meteorological factors (i.e., temperature, wind speed, cloud , and incoming solar radiation), time-of-use (TOU) rates, and hourly electricity prices. All data are collected hourly between Jan. 1 2016 and Dec. 31, 2017 (2 years) and are cleaned properly to address data quality issues by removing outliers and imputing the missing values. We train supervised learning models of electricity consumption and solar generation using data from 2016 and utilize these models to control the battery storage using real electricity consumption and solar production data, and electricity prices in 2017. The electricity consumption, outdoor temperature, and cloud data are pulled in from the Pecan Street Dataport [59]. The overcast index is a real number between 0 and 1, where 0 and 1 indicate no cloud cover and full cloud cover, respectively. Although this repository contains a large number of homes located across the United States, the temperature and overcast index are only available from Colorado, California, and Texas. Thus, in this paper we use data from Austin, Texas which comprises the largest number of monitored homes. Precisely, we utilize electricity consumption of 70 individual homes reported in 1-hour intervals

Table 3.1: Time-of-use prices ($/kWh)

TOU price	7am-11am	11am-5pm	5pm-7pm	7pm-7am
Nov. to Apr.	0.101	0.072	0.101	0.05
May to Oct.	0.072	0.101	0.072	0.05

(15-minute consumption data is also available for some homes).

The second data set contains Global Horizontal Irradiance (GHI) data from the NREL measurement and instrumentation data center [52]. GHI is the total amount of solar radiation received by a surface horizontal to the ground. We obtained irradiance measurements from the Solar Radiation Lab at the University of Texas Pan American (UTPA), the nearest station to the selected homes. The third data set contains TOU electricity prices. Since the TOU pricing scheme is not implemented for residential customers in Austin, we used TOU rates from Ontario [56] and converted them to US dollars using 0.77 for the exchange rate as shown in Table 3.1. The on-peak period is 11am-5pm in summer, and 7am-11am and 5pm-7pm in winter. The mid-peak period is 11am-5pm in winter, and 7am-11am and 5pm-7pm in summer. The off-peak period is 7pm-7am on weekdays and all day on weekends. The TOU rates change every six months in Ontario. The last data set contains hourly electricity prices implemented in several jurisdictions in the United States. These prices are obtained for the same time period using an API [12]. We used prices that correspond to the selected time window.

3.3.2 Data Prepossessing

Multiple data quality problems exist in smart metering and meteorological data which must be detected and addressed before they can be used to train supervised learning models. In particular, we found several instances of missing data, erroneous data, and timestamp issues. For example, the NREL data set contains large negative values for GHI measurement in some time windows. We identified these data points using a threshold and treated them similar to the missing data. We found that they most frequently fell on the month of November, and that on certain days there was no reading from some homes

or meters reported zero throughout the day. We did not include these data points since the electricity consumption of a home is expected to be non-zero even if it is unoccupied (the baseload demand is usually higher than zero). A total of 70 homes remained at the end of the preprocessing step and were used in our study.After preprocessing, all data sets are merged based on their timestamps and a boolean field is added to each record to distinguish between weekend and weekday.

3.3.3 Feature Selection

Previous work has shown that electricity consumption (L) and solar generation (G) are autoregressive time series, meaning that their values at a given time slot depends on their previous time slot values [3]. Furthermore, there are several exogenous variables that may affect electricity consumption and solar generation of a home. These variables are outdoor temperature ($Temp$), wind speed (WS), overcast index (O), hour of day (HoD), day of week (DoW), and month (MoY). We favor parsimonious models as they are easier to interpret and have lower variance. To build such models, we must select a subset of these variables comprising the most discriminating ones from the original set of variables. To this end, we use ANOVA F-score [77] to rank the features that can be used for predicting the home load and PV output. We then select the features with highest F-values (larger than a specific threshold) to develop supervised learning models. These features are $L_{t-1}, L_{t-2}, L_{t-23}, L_{t-24}$, HoD_t, MoY_t, DoW_t and $Temp_t$ for predicting L_t, and $G_{t-1}, G_{t-24}, G_{t-168}, O_t$, WS_t, HoD_t, MoY_t, and $Temp_t$ for predicting G_t.

3.3.4 Overview of Supervised Learning Models

We develop and compare a large suite of supervised learning methods to predict future values of household demand and solar generation. These models include Ridge Regression, Lasso, Bayesian Ridge (BR), Lasso with Least Angle Regression (LassoLars), Linear Regression (LR), Random Forest regression (RF), Decision Tree Regression (DTR), MultiLayer Perceptron network (MLP), and Gaussian Process Regression (GPR). Most of these models have been used

previously to predict the household demand [42]. We describe these models in the following.

Linear Regression

Linear regression is used to find the linear relationship between a target and one or more predictors [50]. It fits a straight line to the relationship between dependent variables, denoted by y, and one or more independent variables, denoted by X:

$$y = \alpha + X\beta + e$$

where α is an intercept term, β is the slope of the line (or hyperplane), and e is the error term. Once α and β are determined, this equation can be used to predict the target variable given the predictors. The best-fit line can be obtained by the method of Least Squares, which minimizes the sum of the squares of the vertical deviations of each data point from the line. Despite its simplicity, linear regression yields accurate results when it is used to model the relationship between the household demand and external variables, such as temperature [42].

Ridge Regression

Ridge regression is a regression method that is used when independent variables are highly linearly correlated [28]. In the presence of such multicollinearity, the least squares estimates may have high variance. Ridge regression addresses this problem by adding bias, i.e., it decreases the variance through a shrinkage parameter λ:

$$\hat{\beta} = arg \min_{\beta \in R^P} \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2$$

The first term in the above equation is the least squares term, while the second term is a regularization term which shrinks the value of coefficients, β, towards zero.

Lasso

Lasso stands for the Least Absolute Shrinkage and Selection Operator. This method enforces sparsity by adding ℓ_1 norm of the regression coefficients to least squares [74]:

$$\hat{\beta} = arg \min_{\beta \in R^P} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

The first term in the above equation is the loss and the second term is the penalty. The lasso penalty will force some of the coefficients to be zero. This means that some variables are removed from the model, hence the sparsity. Unlike ridge regression, LASSO uses absolute values in the penalty function instead of the squares of the penalty function. This may cause some of the parameters to become exactly zero. The larger the penalty is, the more it will be shrunk to zero. This way Lasso selects some features from the set of correlated features. In general, if some variables are highly correlated, Lasso picks only one of them and shrinks the others to zero.

Lasso LARS

Least-angle regression (LARS) is also an algorithm for fitting linear regression models to high-dimensional data [81]. When variables are correlated, the LARS algorithm decides on which variables to include and calculates their coefficients.

It is a particular method to fit a Lasso model and works better than solving a quadratic programming problem [16]. The solution of LARS consists of a curve including the ℓ_1 norm of each parameter vector instead of a vector result.

Bayesian Linear Regression

In Bayesian linear regression, we use probability distributions rather than point estimates to formulate the linear regression. Hence, the response Y is not estimated as a single value but is sampled from a probability distribution. The Bayesian linear regression model with the response sampled from a normal distribution is:

$$y \sim N(\beta^T X, \sigma^2 I)$$

The output, y is generated from a normal distribution characterized by a mean $\beta^T X$ and a variance σ^2. The basic idea of Bayesian linear regression is to find the distribution of the parameters instead of one single best value. Both the response and model parameters are generated by a distribution:

$$P(\beta|y, X) = \frac{P(y|\beta, X) \times P(\beta|X)}{P(y|X)}$$

This distribution $P(\beta|y, X)$ given by inputs and outputs is called the posterior distribution. It is equal to the likelihood of the data, $P(\beta|y, X)$, multiplied by the prior probability of the parameters and divided by a normalization constant (Bayes Theorem).

Decision Tree Regression

Decision tree models are another type of models which are commonly used for classification and regression. Reference [9] introduces different tree regression methods. Decision tree models offer several advantages over other supervised learning models. First, they are easy to be understood, implemented, and visualized. They run fast on large data sets. Second, they can handle both numerical data and categorical data. This makes them suitable for our problem because we deal with numerical data, e.g., temperature and humidity, and categorical data, e.g., if it is cloudy or not and it is a weekend or a weekday.

A decision tree model can suffer from overfitting especially when it is deep. It means that it gives highly accurate output on training data, but low accurate output on test data.

Random Forest

To address the overfitting problem of the decision tree model, a random forest uses a collection of decision trees whose results are aggregated into one final result to reduce the variance [9]. These decision trees can be trained on different slices of data or using a random subset of features. For example, in each tree we can utilize five random features. If we use many trees in a forest, all the features will be considered eventually. A random forest is more robust

than a single decision tree, and limits the error due to bias and the error due to variance.

Multilayer Perceptron

Multilayer perceptron is a class of feed-forword artificial neural networks [26]. Perceptron is a feed-forward neuron that performs binary classification using a weight and a bias:

$$y = f(W^\top x + b)$$

where W is the weight vector, x is the input vector which could be the output of the previous layer, b is the bias, and f is the activation function describing the (nonlinear) input-output relation. Popular activation functions are sigmoid, rectifier linear unit, hyperbolic tangent, etc. A multilayer perception consists of multiple linear layers of such neurons, and approximates an arbitrary function that maps an input x to an output y. The neurons in one layer are connected to all the neurons in the previous layer and possess a unique set of weights. The layers can be the input layer, the output layer, and hidden layers. We feed data into the input linear and take the output from output layer. The model becomes more complex as the number of hidden layers increases. We use two hidden layers, 50 units (neurons) each, in our implementation.

To train the network, an optimization problem is solved to find W that minimizes the loss function by matching the target (actual) value and predicted value. Specifically, W is updated in the direction defined by the gradient of the loss function, until convergence. A learning rate is used to adjust the amount by which the algorithm changes W in every iteration.

Gaussian Process Regression

Gaussian process regression is a nonparametric, kernel-based probabilistic model which takes a Bayesian approach to regression [64]. It calculates the probability distribution of all functions that fit the data instead of calculating the distribution of parameters of a specific function.

Gaussian process regression assumes a Gaussian process prior distribution, which can be specified using a mean function, $m(x)$, and a covariance function

$k(x, x')$

$$f(x) \sim GP(m(x), k(x, x'))$$

A Gaussian process is like an infinite-dimensional multivariate Gaussian distribution, where any collection of the labels of the data set are joint Gaussian distributed. With this Gaussian process prior, we can incorporate prior knowledge about the space of functions through the selection of the mean and covariance functions.

3.3.5 Choosing the Best Model

We perform 10-fold cross validation, which is a re-sampling procedure used to evaluate machine learning models on a limited data sample, to tune parameters of each model described in the previous section to predict the home electricity consumption and solar generation in the next time slot. The 10 results expressed by normalized Root-Mean-Square Error (nRMSE), which measures the differences between predicted values predicted and the observed values observed. To compute nRMSE, we normalize RMSE values with respect to the difference between the maximum and the minimum of the target value. are then averaged to produce a single result. We compare these results and pick the best parameter setting.

To build a model for the home electricity consumption and a model for the solar generation, we use the most discriminating features identified previously. We feed these futures into each model and predict the electricity consumption and solar production of the next hour. Figure 3.2 depicts the prediction error of different models over the next 24 hours, averaged over all days and homes in our data set. Decision tree regression turns out to be the most accurate model for both prediction tasks. Thus, we incorporate this model in model predictive control to obtain future household demands and solar productions.

We use the same set of features used to predict the next value of home electricity consumption and solar generation to predict their values multiple hours in the future. Since some of these features are not observed at the time we run these models, we have to use the predicted values of previous

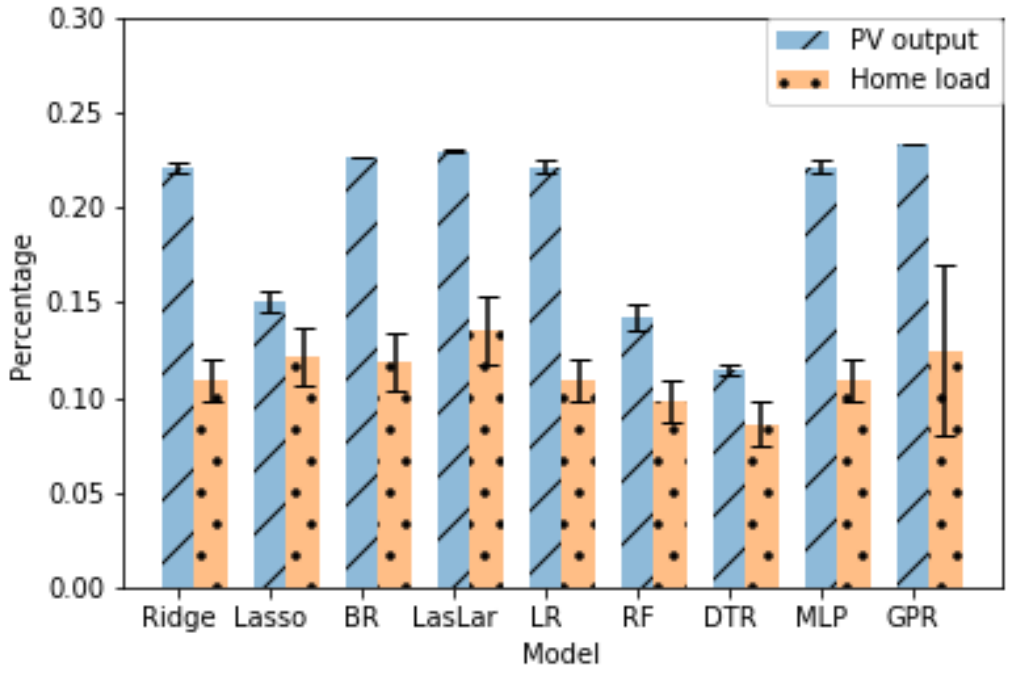


Figure 3.2: nRMSE of the next 24-hour home load and PV output predictions using different models. Error bars represent one standard error.

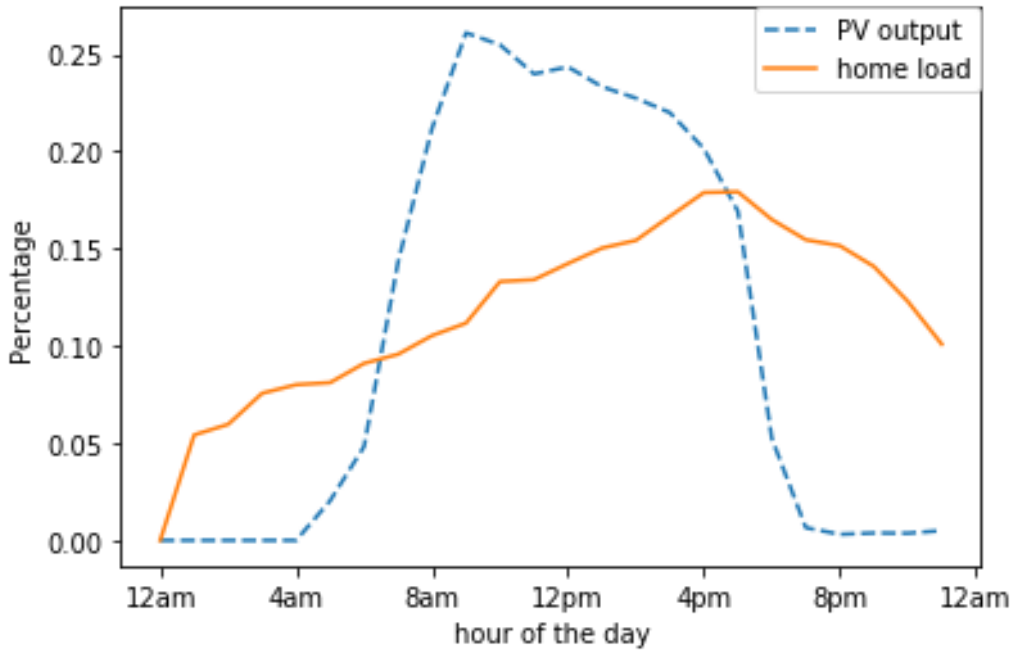


Figure 3.3: nRMSE of PV output and home load predictions over the next 24 hours averaged over all homes.

time slots as features. This causes the prediction error to accumulate over time. Figure 3.3 depicts the nRMSE of home electricity consumption and PV

output over one day. Note that the nRMSE values shown for every hour are averaged over 70 homes and 365 days. Taking the average of these nRMSE values over 24 hours of the day, we get 15.0% and 14.8% error for home load and PV output, respectively. In general, prediction errors are smaller when the demand and solar production are less variable; for example, the PV output can be accurately predicted after sunset and before sunrise next day as the PV output is zero during this time. Furthermore, we observe that the prediction error of home electricity consumption is accumulated over time since we use predicted values instead of observed values as features when we predict the demand multiple hours in advance.

Chapter 4

Optimal Control Methods

In this chapter, we introduce different methods for solving the optimal control problem. We cast the optimal control of the battery as a discrete-time optimization problem, and present model-based and model-free control schemes which can solve this problem. Each control decision represents the amount of energy charged into or discharged from the battery in a given time slot. Once the battery operation is fixed, the amount of conventional power that must be bought from the grid or the amount of solar power that must be exported to the grid can be easily determined. We consider 1 hour time slots because it provides a good trade-off between complexity and optimality of the controller. We also discuss how the optimal control changes as a result of taking control actions at a faster time scale.

The optimal control methods described in this chapter are implemented in Python and are released together with physics-based and data-driven models of the system and the environment as open-source software, called EnergyBoost.

4.1 Optimization Problem

The objective of the battery controller is to minimize the homeowner's annual electricity bill by regulating the charge or discharge power of their battery within certain limits imposed by the battery and its charger. The electricity bill is the net payment to the grid, i.e., the difference between the electricity cost and the revenue generated by selling solar power to the grid. Thus, we

can simply formulate it as an optimization problem over a finite time horizon (for example one year), where the objective function is the total electricity bill including the battery degradation over this horizon. Since homeowners can sell their excess solar power to the grid, the bill would be the difference between the price paid for buying electricity from the grid and the credit received for selling electricity to the grid. Hence, the bill can be negative if they receive more credits than what they pay. We define this optimization problem below.

4.1.1 Constraints

As discussed in Section 3.2.1, battery operations are subject to a set of constraints. In particular, the battery cannot be charged (or discharged) past a certain limit to prevent overflow (or underflow), or at a rate higher than the maximum supported charge (or discharge) power. In this section we discuss three additional constraints. First, the battery cannot be simultaneously charged and discharged at any given point in time. Hence, if the charge rate, AC_t, is positive in a given time slot, the discharge rate, AD_t, must be zero in that time slot, and vice-versa. Second, electricity cannot be bought from and sold to the grid at the same time. Hence, at most one of the two variables $W_t^s, W_t^b \geq 0$, which respectively denote the power sold to the grid and the power bought from the grid, can be positive at any point in time. Third, the power exported to the grid, W_t^s, cannot surpass the instantaneous solar production, G_t. Note that the last two constraints are introduced to prevent homeowners from creating a "money pump" under the feed-in-tariff program introduced in Section 5.1.2, i.e., selling the conventional power which is bought from the grid in the current or a previous time slot back to the grid at a higher price![1]

Finally, just as other power systems, we have to ensure that supply and demand are in balance at all times:

$$W_t^b - W_t^s = L_t + AC_t - AD_t - G_t, \tag{4.1}$$

[1]This can be enforced in different ways (e.g., using a separate generation meter) which are outside the scope of this work.

where L^t is the household demand and G_t is the solar generation. Observe that in each time slot W^s is upper bounded by the total power generated by the solar panel, and W^b is upper bounded by the sum of the household demand and the maximum feasible charge power of the battery.

4.1.2 Mixed Integer Linear Program

Let the solar export tariff be C_t^s and the residential electricity tariff be C_t^b at time t ($C_t^s, C_t^b \geq 0$). The objective function can be written as:

$$J = \sum_{t=t_0}^{T+t_0} W_t^p \tag{4.2}$$

where t_0 is the initial time slot, T is the length of the optimization horizon, and

$$W_t^p = \begin{cases} W_t^b C_t^b & \text{if buying from grid,} \\ -W_t^s C_t^s & \text{if selling to grid,} \end{cases} \tag{4.3}$$

Equation (4.2) can be reformulated by introducing a binary variable M_t^1. This binary variable is 0 when conventional power is bought from the grid to meet the local demand, and is 1 when excess solar power is sold to the grid. Similarly, since the battery is either in the charge or discharge mode at each time slot, we use another binary variable M_t^2 to represent the battery's operating mode in that interval. Note that we have to separate the optimization variables for the charge and discharge rates because they affect the battery SoC in different ways (refer to Equation (3.2)). Putting it all together, the optimization problem

can be formulated as:

$$\underset{W^b,W^s,AC,AD,M^1,M^2}{\text{minimize}} \quad J = \sum_{t=t_0}^{T+t_0} W_t^b C_t^b - W_t^s C_t^s \tag{4.4a}$$

$$\text{s.t.} \quad E_{t+1}{=}E_t(1{-}\eta^{\text{p\_leak}}){+}\Delta E_t{-}T^u\eta^{\text{c\_leak}} \tag{4.4b}$$

$$\Delta E_t = T^u(AC_t\eta^c - AD_t/\eta^d) \tag{4.4c}$$

$$0 \leq W_t^s \leq G_t M_t^1 \tag{4.4d}$$

$$0 \leq W_t^b \leq (L_t + \alpha^c)(1 - M_t^1) \tag{4.4e}$$

$$W_t^b - W_t^s = L_t + AC_t - AD_t - G_t \tag{4.4f}$$

$$0 \leq AD_t \leq \alpha^d M_t^2 \tag{4.4g}$$

$$0 \leq AC_t \leq \alpha^c(1 - M_t^2) \tag{4.4h}$$

$$E^{min} \leq E_{t+1} \leq E^{max} \tag{4.4i}$$

$$M_t^1, M_t^2 \in \{0, 1\} \tag{4.4j}$$

The first two equality constraints express how the battery's SoC evolves over time as described in Section 3.2.1. The third and fourth constraints limit the maximum amount of power that can be possibly sold to or bought from the grid, respectively, and ensure that power cannot be bought from and sold to the grid simultaneously. The fifth constraint is the power balance equation, and constraints 4.4g 4.4h ensure that battery charge and discharge rates are bounded and the battery cannot be charged and discharged simultaneously. Constraint 4.4i defines the upper and lower bounds on the state of the charge of the battery, where the upper bound is the battery's nominal capacity. Note that all these constraints are convex except for the last one which defines the binary decision variables.

Given the binary variables and linearity of the objective function, Problem (4.4) is a mixed-integer linear program (MILP), which can be solved using the branch-and-bound algorithm, after relaxing the integrality constraints. The solution of the relaxed problem gives a lower bound on the solution of the original problem[2] The solution to this problem determines, in each interval,

[2]Based on our experiments for various parameter settings, the relaxation gap to the true optimal solution is always less than 0.1% for homes in our data set. We defer the study of finding a better relaxation to future work.

the strategy to control battery operations and how much power it must sell to or buy from the grid. We note that this problem cannot be solved in practice, unless we assume the knowledge of G_t, L_t for every t in the planning horizon.

MILP with Oracle (Optimal)

To obtain a lower bound on the electricity bill, we implement a hypothetical controller which takes advantage of an oracle to obtain future household demands and solar productions. we formulate the optimization problem assuming that future values for home load and solar generation are available. We use actual data instead of the predicted data for one year period and solve the optimization problem to find the best policy for battery operation in this single snapshot. Indeed, this consideration contributes to obtaining the best battery charge/discharge rates leading to the minimum energy received from the grid during the whole year of 2017. The solution to this problem makes a baseline to evaluate the results of the problem in the presence of predictive models. After plugging in these predictions, this controller solves the MILP problem presented in Section 4.1 over one year to find the minimum bill that can be possibly achieved. We compare the bills of other controllers with this bill to understand how far they are from the optimal.

4.2 Model Predictive Controller

In this section, we propose a model predictive controller with a horizon of 24 hours to minimize the electricity bill. The proposed control algorithm utilizes a learned model of the system to predict the future control outputs given the current control inputs and system states. An optimal control over the specified time horizon is then determined by solving Problem (4.4) based on these predictions and is only implemented in the current time slot. This process repeats in every time slot with a model that is updated in an online fashion as described in Algorithm 1, below. The Model Predictive Control algorithm solves an optimization problem similar to the one described in Section 4.1 by using the predicted values of G_t and L_t in the optimization horizon. To pre-

dict G_t and L_t during the optimization horizon, we utilize data-driven models described in Section 3.3:

$$L_t = \mathcal{F}^L(L_{t-1}, L_{t-2}, L_{t-23}, L_{t-24}, HoD_t, MoY_t, DoW_t, Temp_t), \tag{4.5}$$

$$G_t = \mathcal{F}^G(G_{t-1}, G_{t-24}, G_{t-168}, Temp_t, O_t, WS_t, HoD_t, MoY_t). \tag{4.6}$$

where L,HoD,MoY,DoW,$Temp$,G,O,WS are homeload, home of a day, day of week, external temperature, solar generation,cloud observation, and wind speed respectively.

Remark. *MPC uses the predicted values of L_t and G_t to calculate the optimal decision variables, referred to as $\overline{AC}_t, \overline{AD}_t, \overline{W}^s_t, \overline{W}^b_t, \overline{M}^1_t, \overline{M}^2_t$. We argue that $\overline{AC}_t$ and $\overline{AD}_t$ are always feasible for the next time slot as they satisfy all the constraints despite the prediction errors of L_t and G_t. This is simply because the battery constraints for the next time slot do not depend on these predictions. That said, the error of predicting L_t and G_t affects the actual amount of electricity which must be bought from or sold to the grid at t. Hence, before we can obtain the bill, W^s_t, W^b_t must be recalculated based on the observed values of L_t and G_t, denoted by $\widetilde{L}_t$ and $\widetilde{G}_t$, as follows:*

$$W^b_t = \max(\widetilde{L}_t + \overline{AC}_t - \overline{AD}_t - \widetilde{G}_t, 0), \tag{4.7}$$

$$W^s_t = \max(\overline{AD}_t + \widetilde{G}_t - \widetilde{L}_t - \overline{AC}_t, 0). \tag{4.8}$$

Here $W^{t^*}_b$ and $W^{t^*}_s$ are the actual amount of electricity bought from or sold to the grid at t, respectively. It should be noted that due to strict bounds of AC^t and AD^t, they cannot be back calculated, otherwise, they may be recalculated out of their feasibility region.

4.3 Sample-Based Predictive Controller

Reinforcement learning provides an alternative approach to solving the optimal control problem. In this framework a decision making *agent* takes a sequence of actions (i.e., charge and discharge operations) in a number of episodes in a stochastic environment and learns from the outcome of these actions, i.e.,

Algorithm 1: Model Predictive Control

Input: $s_0, \alpha^c, \alpha^d, \eta^c, \eta^d, \eta^{\text{p\_leak}}, \eta^{\text{c\_leak}}$
learn data-driven models to predict future values of L, G;
while *True* **do**
  solve a relaxation of Optimal Control Problem (4.4);
  apply $\overline{AC}_t$ or $\overline{AD}_t$ in $[t, t+1)$ and update E_{t+1};
  observe $\widetilde{L}_t$ & $\widetilde{G}_t$ and recalculate W_t^s & W_t^b from (4.7-4.8);
  $t \leftarrow t+1$;

rewards returned by the environment, to maximize the *expected* cumulative reward [71]. In our problem, the system we built namely EnergyBoost is the agent, each action is the battery charge/discharge power in a time slot, and the cumulative reward is equivalent to negative of the homeowner's electricity bill. Unlike MPC which relies on a model describing dynamics of the environment, the RL agent uses a *policy* to interact with the environment and learns from these interactions to gradually converge to an optimal policy which determines a sequence of actions maximizing the expected cumulative reward. We denote the state space by $\mathcal{S}$, the action space by $\mathcal{A}$, and the reward function by $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Given the state $s \in \mathcal{S}$, the action $a \in \mathcal{A}$ is selected according to a policy π; this action causes a transition to the next state $s' \in \mathcal{S}$ and earns the agent a reward of $r(s, a)$.

To model our optimal control problem, we must first define the state, the policy function, and the reward function. We define the agent's state at t, which includes the agent's observations of the environment, as a quadruple consisting of these features:

$$s = (L^t, G^t, SoC^t, t^t_{index}), \tag{4.9}$$

where t_{index} compactly represents all time-related features, i.e., time of the day, day of the week, and month. Note that the first three state variables are continuous-valued features and the action (i.e., the battery charge/discharge power) is also continuous-valued. Thus, we need to adopt an appropriate *feature representation* to incorporate low dimensional features [45].

The stochastic policy, denoted by $\pi_\theta : \mathcal{S} \to P(\mathcal{A})$, is defined as the condi-

tional probability density at a,

$$\pi_\theta(a|s) \sim G(\mu_\theta(s,a), \sigma^2_\theta(s,a)) \tag{4.10}$$

where G is the Gaussian distribution, $P(\mathcal{A})$ is the set of probabilities on $\mathcal{A}$, $\mu_\theta(s,a) = x_a^\top \theta_\mu$, $\sigma_\theta(s,a) = \log(\exp(x_a^\top \theta_\sigma) + 1)$, and parameters $\theta_\mu, \theta_\sigma \in \mathbb{R}^n$. x_a is a low dimensional feature of state and action which will be defined in the following section.

4.3.1 Feature Representation

Appropriate representation of states in terms of features is needed for the implementation of a reinforcement learning algorithm. State representation involves characterizing raw data with particular features in low dimension. The incorporation of informative low dimensional features are very helpful in representation of continuous action space in RL problems. The learned features evolve over time and capture aspects of states that are useful in learning in the environment, leading to improved performance and speedup in policy learning algorithms.Indeed, by using feature representation instead of raw data, reinforcement learning can incorporate informative low dimensional features which can provide the ability to learn controllers directly from observations too [45].

Although the states can be sorted in various forms to represent the state space, radial basis functions (RBF) provide a natural generalization to continuous features [71]. With a large data set of n transitions (s_i, a_i, $s_i^{'}$, and r_i), the implementation of RBF for d randomly selected transitions at each time step results in the feature vectors

$$x(s) = [k(s,s1), \ldots, k(s,s_d)] \tag{4.11a}$$

$$x_a(s) = [k(s,s_1)k(a,a_1), \ldots, k(s,s_d)k(a,a_d)] \tag{4.11b}$$

where $k(s,s_i) = exp(-\frac{\|s-s_i\|}{2\sigma_i^2})$ and σ_i is the feature width. Now that a reduced dimension of data is obtained through the RBF representation, we can produce smooth and differentiable approximate functions.

4.3.2 Linear Function Approximation

In value-based reinforcement learning algorithms, the value function is represented using a parameterized function approximator with weight vectors $\mathbf{w}, \mathbf{u} \in \mathbb{R}^d$, especially when the state space is large. The approximation of true value function under policy π given the weights $\mathbf{w}$ and $\mathbf{u}$ can be represented as

$$\hat{V}(x; \mathbf{w}) \approx V^{\pi}(x) \tag{4.12}$$

$$\hat{Q}(x_a; \mathbf{u}) \approx Q^{\pi}(x_a) \tag{4.13}$$

where x and x_a are the feature representation of state and action spaces as defined in (4.11a) and (4.11b), respectively. The value function may be approximated with linear or nonlinear function approximators. Nonlinear function approximators such as neural networks may cause instability or even divergence due to the correlation in the sequence of observations or the correlation among the action values ($Q(x_a)$) and their target values ($r + \gamma \text{max } Q(x'_a)$), where x'_a is the successor of the feature x_a [45]. Another problem with nonlinear function approximators is that it is difficult to identify the features. Consequently, linear approximation of the value function would be a more reliable alternative to converge to the true value function. We define the linear approximate value functions as

$$\hat{V}(x; \mathbf{w}) = \mathbf{w}^T x = \sum_{i=1}^{d} w_i x_i \tag{4.14}$$

$$\hat{Q}(x_a; \mathbf{u}) = \mathbf{u}^T x_a = \sum_{i=1}^{d} u_i x_{a_i} \tag{4.15}$$

where the value function can be updated in each step when the new data is stored. In order to find the best linear approximation, the weights must be updated at each step in such a way that the overall error between the true and approximated value functions following the policy π is minimized. This error can be defined as a quadratic function of parameters $\mathbf{w}$ and $\mathbf{u}$ as follows:

$$J(\mathbf{w}) = \mathbb{E}_{\pi}[(V^{\pi}(x) - \mathbf{w}^T x)^2] \tag{4.16}$$

$$J(\mathbf{u}) = \mathbb{E}_{\pi}[(Q^{\pi}(x_a) - \mathbf{u}^T x_a)^2] \tag{4.17}$$

where the optimal weight vector can be obtained using the solution of the least square algorithm.

4.3.3 Simulator

A reinforcement learning agent can learn by interacting with the environment in a number of episodes. The more the agent interacts with the environment, the better the resulting policy could be. In our control problem, interacting with the real environment in the form of charging and discharging the battery at arbitrary rates could be detrimental for the battery lifetime and may even cause overheating of the integrated inverter. To overcome this obstacle, we develop a simulator which stores the current state of the real environment, implements the action selected by the agent, updates the state using physics-based models, and returns a reward in lieu of the real environment, thereby enabling the agent to explore without worrying about the consequence of its actions. For our implementation, an episode is defined by a time interval of a certain length and once we reach the predefined maximum number of interactions per episode, the simulator resets the state to the current state of the system and starts over for the next episode.

Building the simulator requires updating the state every time an action is taken. Among the elements of the state quadruple (4.9), SoC_t is the only element which depends on the action and can be updated based on Eq.(3.1-3.2). t_t^{index} should be simply incremented every time a new action is simulated. The other two elements of the state, i.e., L_t and G_t, must be simulated. To this end, we exploit electricity consumption and solar generation data in 2016 to simulate electricity consumption and solar generation on a given day in 2017. Specifically, the simulator takes a random sample from a day in the same month in 2016 and perturbs it with noise to increase exploration for policy evaluation. We consider a white noise with the standard deviation equal to 5% of the average of the value we want to predict. The L_t and G_t samples are then used to update the state. This process is repeated for every episode until the maximum number of episodes, denoted by τ, is reached.

4.3.4 Actor-Critic Algorithm

Policy gradient algorithms are promising approaches to solving reinforcement learning problems with a continuous action space [70]. The basic idea is to improve the performance of a policy by updating its parameter vector θ in the direction of the performance gradient $\nabla_\theta J(\pi_\theta)$, which is given by [72]:

$$\begin{aligned} \nabla_\theta J(\pi_\theta) &= \sum_{a \in A} \nabla_\theta \pi_\theta(a|\phi(s))\ Q^\pi(\phi(s,a)) \\ &= \mathbb{E}_{a \sim \pi_\theta}[\nabla_\theta \log\ \pi_\theta(a|\phi(s))\ Q^\pi(\phi(s,a))] \end{aligned} \tag{4.18}$$

We just need to learn the action-value function $Q^\pi(\phi(s,a))$. We leverage a linear approximation of the action-value function

$$Q^\pi(\phi(s,a);\mathbf{u}) = \mathbf{u}^\top \phi(s,a) \tag{4.19}$$

and update its parameter vector $\mathbf{u}$ every time we update the policy. This gives rise to the actor-critic algorithm [13] which is a policy gradient method. The actor part of the algorithm uses gradient ascent algorithm to update θ. The critic part uses an appropriate policy evaluation algorithm such as temporal-difference (TD) learning [71] to update $\mathbf{u}$. Algorithm 2 describes the basic actor-critic (A2C) algorithm [46], which is an online, model-free controller. This algorithm can effectively deal with problems with continuous and constrained action spaces. It estimates the advantage of the state-action pair, which is defined as

$$A^\pi(\phi(s,a);\mathbf{w}) = Q^\pi(\phi(s,a);w_u) - V^\pi(\phi(s);w_v) \tag{4.20}$$

instead of the action-value function to reduce the variance. TD error is an approximation of the advantage function. The parameter vector of the value function, $\mathbf{w}$, is then updated using the TD error. We use an episodic version of the basic actor-critic algorithm and set the episode length to one day. We also set $\tau = 500$, $\gamma = 1$, $\lambda = 0.001$, and $\beta = 0.1$.

4.3.5 Proximal Policy Optimization Algorithm

The Actor-Critic algorithm presented in the previous section is empirically finicky, and has poor data efficiency and robustness in many real-world control problems, including the control problem studied in this thesis. Thus, we

Algorithm 2: One-step Actor-Critic

Input: a differentiable policy parameterization $\pi(a|s,\theta)$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
for *each episode* **do**
  Initialize S (first state of episode)
  $I \leftarrow 1$
  for *each time step* **do**
    $A \sim \pi(\cdot|S,\theta)$
    Take action A, observe S', R
    $\delta \leftarrow R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$ (if S' is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}}\delta\nabla\hat{v}(S,\mathbf{w})$
    $\theta \leftarrow \theta + \alpha^{\theta} I\delta\nabla ln\pi(A|S,\theta)$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$

attempt to control battery operations using another policy gradient algorithm, namely Proximal Policy Optimization (PPO) [68], which has shown superior performance in several continuous control problems. This algorithm alternates between sampling data through interaction with the environment and optimizing a surrogate objective function with clipped probability ratios.

There are several approaches with neural network function approximators that have been proposed for reinforcement learning. However, Q-learning with function approximation [71] fails on continuous control benchmarks such as those in OpenAI Gym [10], and the vanilla policy gradient method, which is a generic Policy Gradient algorithm with a baseline, has poor data efficiency and robustness. Trust region policy optimization (TRPO) [67] maximizes an objective function subject to a constraint on the size of the policy update. However the constraint in TRPO is relatively complicated and is not compatible with noisy architecture and parameter sharing, meaning that a certain parameter does not perform well across different problems.

PPO draws upon the idea of TRPO but is relatively simple and easy to implement. It could scale to large models and parallel implementation, and offers improved data efficiency and robustness in problems with continuous action space. Algorithm 3 shows different steps of PPO. $\hat{A}_t$ is an estimator of the advantage function at time step t and $rt(\theta)$ denotes the probability ratio

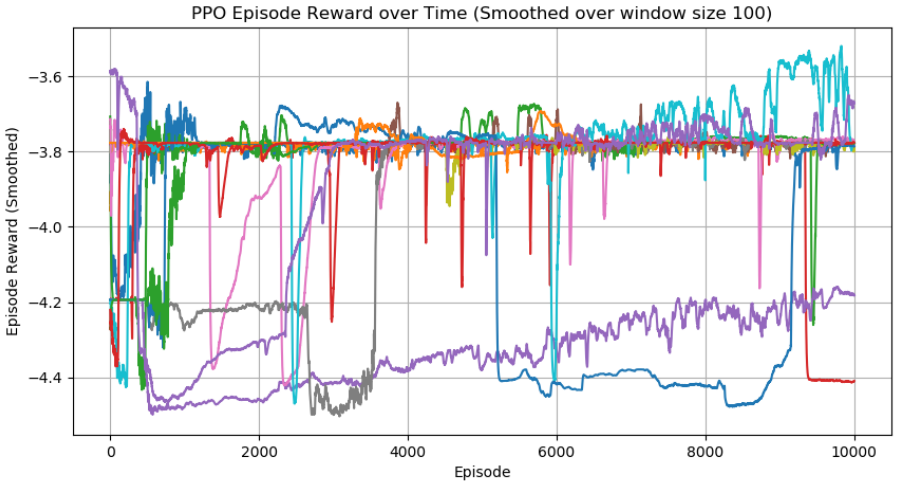


Figure 4.1: Episode reward versus the number of episodes. 10 runs of PPO are shown using different colors.

of policy:

$$rt(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{4.21}$$

where $r(\theta_{old}) = 1$. Similar to TRPO, PPO maximizes an objective function

$$L(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \hat{A}_t \right]. \tag{4.22}$$

Without a constraint, the maximization of $L(\theta)$ would lead to a large policy update, hence the objective function needs to be modified (clipped) to prevent $r_t(\theta)$ from moving away from 1. Note that SGD stands for Stochastic Gradient Descent in Algorithm 3. Instead of performing computations on the whole dataset, it only computes on a small subset or a random selection of data samples. Adam also stands for Adaptive Moment Estimation, an algorithm for gradient-based optimization of stochastic objective functions [51].

Figure 4.1 illustrates how the episode reward changes as we increase the number of episodes. We find that it is relatively easier for the agent to converge to the optimal control policy using PPO compared to A2C. Considering 10 rounds of simulation each with 10,000 episodes, PPO exhibits a converging pattern most of times, whereas A2C does not converge in most cases.

Algorithm 3: PPO with Clipped Objective

Input: initial policy parameters θ_0, clipping threshed ϵ

for *k = 0,1,2,...* **do**

Collect a set of partial trajectories D_k on policy $\pi_k = \pi(\theta_k)$

Estimate advantage $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm

Compute the policy update

$$\theta_{k+1} = arg\max_{\theta} \mathcal{L}_{\theta_k}^{CLIP}(\theta)$$

by taking K steps of minibatch SGD (via Adam), where

$$\mathcal{L}_{\theta_k}^{CLIP}(\theta) = \mathbb{E}_{\tau\sim\pi_k}\left[\sum_{t=0}^{T}[\min(r_t(\theta)\hat{A}_t^{\pi_k}, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^{\pi_k}]\right]$$

4.4 Direct Learning-based Controller (DLC)

This controller utilizes a multi-layer perceptron model to directly learn the relationship between a set of features and the optimal control without solving an optimization problem in an online fashion. The features are $G_{t-1}, L_{t-1}, O_{t-1}$, WS_{t-1}, $Temp_{t-1}$, HoD_{t-1}, MoY_{t-1}, and DoW_{t-1}, and the output is $AC_t - AD_t$ which can be split into AC_t and AD_t based on its sign.

Figure 4.2 shows the structure of the neural network used by DLC. The neural network model consists of one hidden layer with 100 neurons and ReLU activation function. It is trained using the stochastic gradient descent algorithm, given the features and the corresponding optimal policy obtained in the previous year (i.e., the result of solving the MILP problem with an oracle in 2016). The output of the neural network model is projected onto the feasible set before it is applied to control the battery.

4.5 Rule-based Controllers

We now explain two rule-based controllers which we use as baselines.

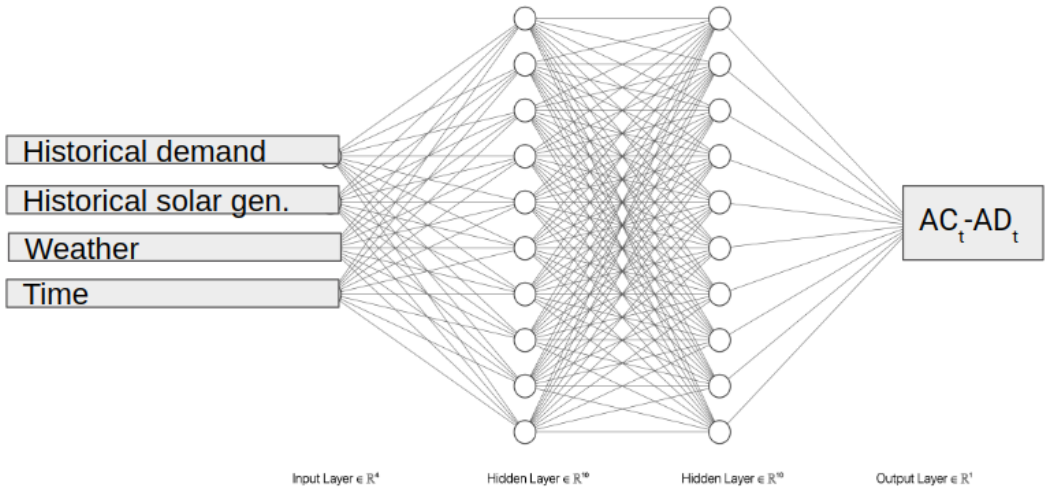


Figure 4.2: Structure of DLC neural networks

4.5.1 Performing Tariff Optimization (RBC-T)

This rule-based controller simply adjusts the battery charge/discharge power based on the retail electricity prices. In particular, it charges the battery at the maximum charge rate when electricity is cheaper (during the off-peak period) such that it gets full just before the electricity price starts to increase (this is to minimize the energy loss due to self-discharge). Similarly, it discharges the battery at the maximum feasible rate when electricity is most expensive (in the beginning of the on-peak period) until the battery is depleted. During the mid-peak period, it neither charges nor discharges the battery.

This strategy does not perform sophisticated forecasting of intermittent solar generation or battery and inverter modelling; thus, it can be easily implemented on a microcontroller. Nevertheless, this controller is myopic and only benefits from energy arbitrage without utilizing the battery to support self-use of solar power. Due to its sheer simplicity, it is widely adopted today for operating grid-tied batteries, but it does not guarantee optimal operation of the battery, especially when the solar export tariff is high.

4.5.2 Maximizing Self-use of Solar Energy (RBC-S)

Another rule-based controller is the one that maximizes self-consumption of the generated solar power irrespective of electricity prices and export tariffs.

Specifically, this controller uses all solar energy to meet the household demand, storing the excess solar energy in the battery. When solar generation falls short of the household demand, the battery discharges at the maximum rate, unless it is either fully discharged or the household demand is smaller than its maximum discharge rate.

Chapter 5

Experimental Results

In this chapter, we evaluate the proposed learning-based control methods and discuss the results. Section 5.1 describes our performance evaluation methodology and introduces different simulation scenarios and performance metrics. Section 5.2 explains the performance evaluation results in terms of the monthly electricity bills, and studies the effects of the system size and pricing scheme. It also discusses the economic feasibility of the solar-plus-battery system when the battery is controlled by one of the proposed learning-based methods and concludes the discussion by presenting important practical considerations.

5.1 Evaluation

We evaluate the performance of the proposed optimal controllers through extensive numerical simulations. Our code is written in Python. We use OpenAI Gym and TensorFlow to implement the A2C, PPO and DLC algorithm, and CVXPY and Gurobi Python API to solve the MILP problem required for MPC. We discuss simulation scenarios below and introduce two baseline methods which help us better understand how close we can possibly get to the true optimal and how the proposed controllers compare to the widely used controllers.

5.1.1 Scenarios

We consider different system sizes, residential retail electricity prices, and solar export tariffs to compare different control algorithms. In particular, we try

three battery sizes $B^{cap} = 0$, 6.4, and 13.5 kWh and three sizes of solar panel with 0, 2, and 4 strings of PV module in parallel. We assume each string is an array of 10 modules in series. Hence, the nameplate rating of PV systems (i.e., the maximum amount of power they can produce) would be $G^{cap} = 0$, 4.4, 8.8 kWp, respectively. Hence kWp represents 'peak power' in kilowatt. Moreover, to analyze the cost saving in various jurisdictions, we implement TOU and hourly pricing schemes for residential customers and consider four different solar export tariffs, namely $3, 6.1, 7.7$, and $15.4¢/kWh$. The solar export rates are chosen such that we have at least one rate below and above every TOU rate.

We set parameters of the battery according to the specifications of Tesla Powerwall Li-ion battery. Hence, α^c, α^d are set to 2kW for a 6.4kWh battery, and to 5kW for a 13.5kWh battery for Powerwall model 1 and model 2, respectively. The roundtrip efficiency (i.e., the ratio of the energy put into and later retrieved from the battery) of this battery is around 90% so we set η_c, η_d to 95%, and its depth of discharge is reported as 100% so we set $E^{min} = 0$, $E^{max} = B^{cap}$ [73]. The other parameters of the battery are set as follows: $\eta^{p\_leak} = 0$, $\eta^{c\_leak} = B^{cap} \times 10^{-4}$, $T^u = 1$, and $SoC_{t_0} = 0.5 \times B^{cap}$.

5.1.2 Renewable Energy Initiatives

Feed-in tariff (FIT) and net energy metering (NEM) are two methods by which a utility company compensates homeowners for the renewable energy (solar, wind, etc.) they generate and export to the grid. Net metering can be easily implemented as it requires only one meter measuring the current flowing in both directions. Under the NEM program, homeowners are billed for the difference between their renewable energy production and demand. Unlike net metering, homeowners may need to install an additional meter for measuring their renewable generation separately under the FIT program. This allows for using two different prices for power consumption and generation. Specifically, homeowners pay for their electricity demand at one price and are paid for the renewable energy they produce at a predefined price which gradually reduces over the years.

There is a variety of approaches to reimbursing customers who produce more electricity than they use in the NEM and FIT programs. In some jurisdictions, they receive credits at full retail value, while in other jurisdictions they are reimbursed at a predetermined rate. We compare different solar tariffs and explore how they impact ROI calculations in Section 5.2.3.

5.1.3 Evaluation Metrics

Annual electricity bill and peak-to-average ratio (PAR) are two metrics used to evaluate the performance of different controllers. We define PAR as the ratio of the maximum grid power consumed by all homes in our data set to the average grid power consumed by these homes over one year. These metrics reflect the benefits the system offers to homeowners and to the grid, respectively. Since PAR is not incorporated in the objective function of our optimal controller, this metric merely indicates whether the EnergyBoost system exacerbates the already high peak-to-average ratio if it is adopted by more than a certain percentage of residential customers.

5.2 Results

We first compare the annual electricity bill of each home and the system's PAR under different control algorithms and TOU pricing scheme. To illustrate the effect of control algorithm on the bill, we compare the cumulative electricity bill computed by RBC-S, RBC-T, DLC, A2C, PPO, and MPC with the optimal bill under various scenarios and plot them in Figure 5.5

Figure 5.1 shows the comparison of the bill obtained by different control algorithms in a randomly sampled home under two different battery sizes and solar tariffs. We assume that the installed solar system is 4.4kWp and the bill is computed using TOU rates. Our results indicate that MPC outperforms RBC, DLC, and A2C controllers in terms of the annual electricity bill on average by 88.6%, 62.2%, and 89.6%, respectively, and yields a bill that closely follows the optimal bill (is only 7.6% higher). We attribute the poor performance of both A2C and PPO to the small number of episodes we tried, low sample efficiency

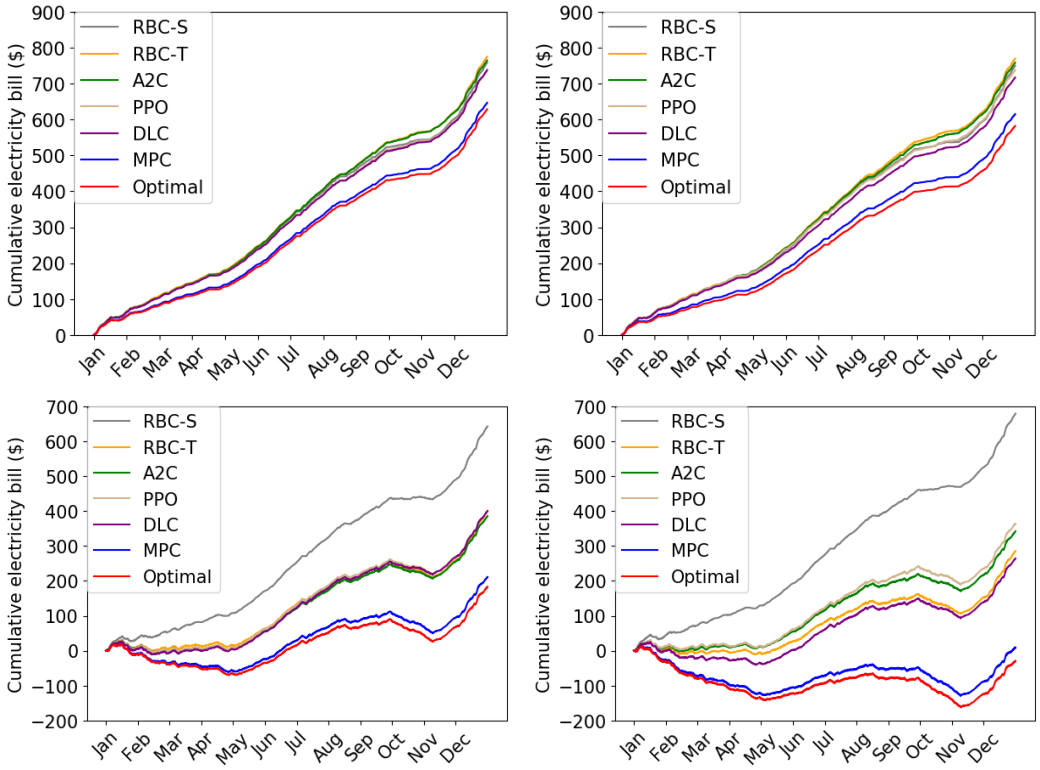


Figure 5.1: Comparing annual bills obtained by different controllers in an example home with a 4.4kWp PV system and a Tesla Powerwall battery (left column: Model 1; right column: Model 2). The solar tariff is 0.03$/kWh (top row) and 0.154$/kWh (bottom row).

of the on-policy actor-critic method, and the non-stationary environment. The bill obtained by the two RL methods (i.e., A2C and PPO) improves gradually as we increase the number of episodes since it allows for more exploration. But in practice we cannot try as many episodes as it takes to converge to the optimal policy due to the time constraint.

The policy curves of the four control algorithms are depicted for an example home over one week in Figure 5.2. The y-axis shows the household demand, solar production, and battery rate in kilowatts. A positive (negative) rate implies that the battery is charged (discharged) at that rate. Observe that the policy found by MPC is the most similar policy to the optimal policy, supporting the conclusion drawn from Figure 5.1. Moreover, except for the two RL methods, other policies do not lead to extreme changes in battery charge/discharge rates in successive time slots, which could accelerate battery degradation.

The MPC policy increases PAR of the aggregate load of all homes in our data set by 37% on average compared to the case that there was no battery, while RBC-T increases it by 65% on average. That said, PAR of the entire system does not increase as long as the the penetration rate of batteries controlled by EnergyBoost is moderate. This is because the peak introduced by simultaneous charging of batteries does not coincide with the existing demand peak, and it does not also contribute to a new peak unless the penetration rate of this system is really high.

5.2.1 Effect of the System Size

Figure 5.3 compares distributions of annual electricity bill for different sizes of battery and PV systems, and four different solar export tariffs. In all cases, TOU pricing scheme is used and the battery is controlled by MPC. It can be readily seen that for all solar export tariffs, the size of the PV system has more impact on the distributions compared to the size of the battery. This is mainly because the difference between the energy produced by 4.4kWp and 8.8kWp PV systems could be 20kWh on a sunny day, which is much higher than the difference between the battery capacities we considered.

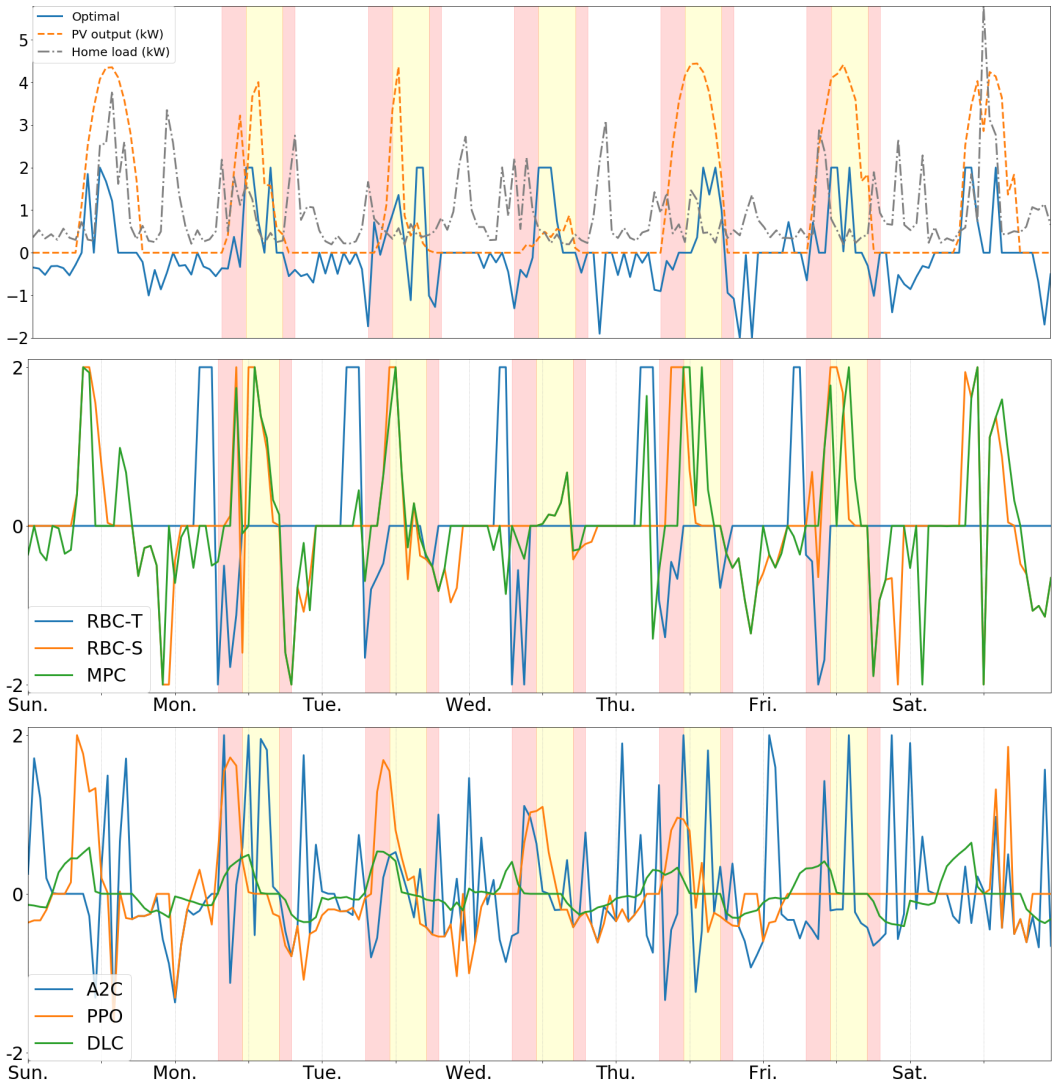
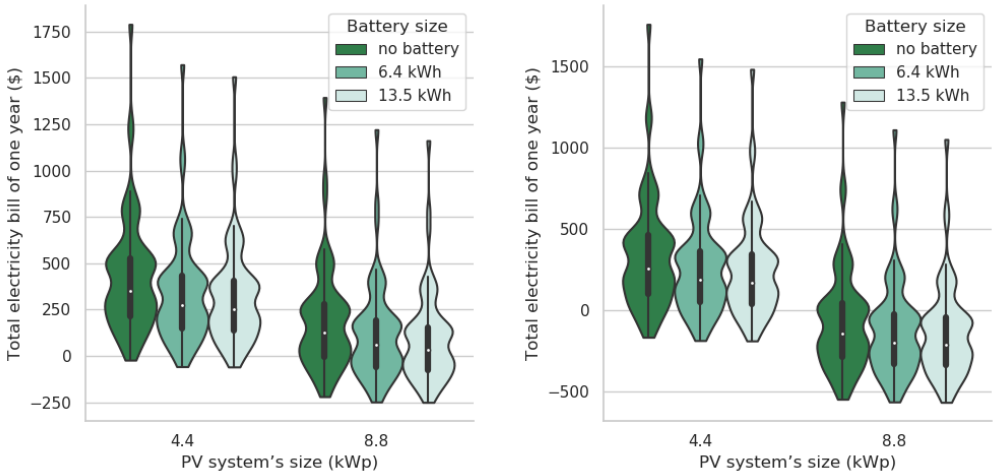
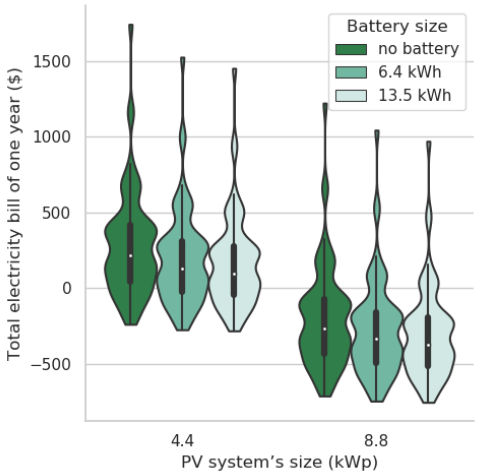


Figure 5.2: Comparing different policies in an example home with a 4.4kWp PV system and a Tesla Powerwall 1. The solar tariff is 0.03$/kWh. The on-peak and mid-peak intervals are highlighted in red and yellow. The lower plot shows polices during the same week. It is not overlaid on the upper figure for legibility.

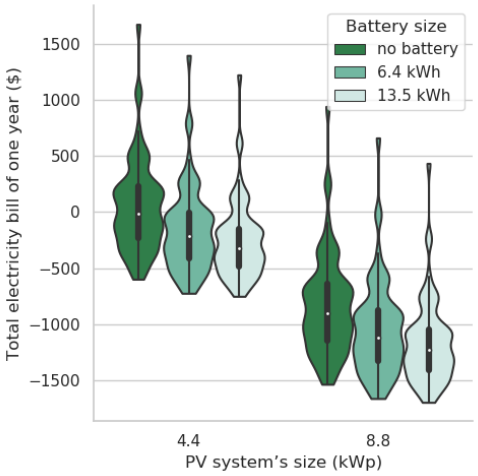


(a) 0.03$/kWh

(b) 0.061$/kWh



(c) 0.077$/kWh



(d) 0.154$/kWh

Figure 5.3: Distribution of annual bills obtained by MPC for different sizes of the PV system and battery. The caption shows the solar export tariff in each case.

It can also be seen that installing a larger battery reduces the median of the annual bill distribution more drastically when the solar export tariff is high. We attribute this to the fact that when the solar export tariff is low, EnergyBoost mostly uses the battery for energy arbitrage rather than shifting solar production to peak times. Thus, installing a larger battery makes a smaller difference in the bill. Nevertheless, when the solar export tariff is high (0.154$/kWh), EnergyBoost utilizes the battery to increase self-use of solar energy, thereby taking advantage of the excess battery capacity. The same observation can be made in Figure 5.1.

5.2.2 Pricing Schemes

With the growing adoption of smart meters measuring electricity consumption at a fine granularity (e.g., once every 15 minutes), utilities have begun to implement dynamic and market-based pricing schemes for residential customers who used to receive monthly (or bi-monthly) electricity bills. Time-of-use (TOU) electricity pricing, hourly electricity pricing, and demand charges are examples of such pricing schemes designed to reflect the costs of producing electricity at different times of the day[1]. Time-of-use has been introduced in many jurisdictions to date to encourage residential customers to shift their loads to off-peak hours, thereby lowering the peak to average ratio of the system while reducing their own electricity bill. There are typically three time-of-use periods that vary across seasons: off-peak when the cost and demand are low; mid-peak when the cost and demand are moderate; on-peak when the cost and demand are high. Nighttimes, weekends, and holidays are considered off-peak as the cost and demand for electricity are low during these times.

Hourly electricity pricing is another new pricing scheme. It is currently offered by a small number of utilities (e.g., in Illinois[12]) and requires meters capable of measuring and recording electricity consumption in hourly intervals. The hourly rate is usually determined by taking the average of the twelve 5-

[1]While TOU pricing reflects the electricity production costs more accurately than conventional fixed-rate pricing, it is different from real-time pricing in which the electricity price varies continuously over the course of a day, tracking fluctuations in supply and demand.

minute prices from that hour.

In contrast to TOU and hourly pricing schemes which adjust the volumetric prices ($/kWh) that customers pay for electricity, with demand charges, customers generally pay lower volumetric prices and are instead charged based on their peak power usage (measured in kW) during a billing period. Demand charges could represent a large fraction of a customer's total electricity bill and studies suggest that PV installations alone could even lead to more expensive bills overall. This is because the portion of the bill that can be cut with solar production may be reduced with demand charges, while the peak power usage does not change because it does not coincide with the peak solar production. However, solar-plus-battery installations can significantly reduce customer demand charges by reducing the peak power usage.

In demand charge pricing scheme, the utility bill is based on the customer's average peak power usage within a defined period (e.g. within 15 minutes interval) during the billing period. If the customer uses much power in the short time interval, then the demand charge will constitute large part of the bill. The demand charge will be reduced for homes that are equipped with the PV system when the peak power usage coincides with the peak PV output.

There are several methods by which a utility company can compensate homeowners for the renewable energy they generate and export to the grid. In some jurisdictions, customers receive credits at full retail value, while in other jurisdictions they are reimbursed at a predetermined rate. We compare different solar export tariffs and explore how they impact ROI calculations in Section 5.2.3.

We compare the performance of MPC and optimal controller under the hourly pricing scheme. In this case, the MPC controller utilizes an additional data-driven model to predict the next day's hourly prices given today's hourly prices. Figure 5.4 shows the annual bill distribution when the battery is controlled following the MPC and optimal policies. We find that the average annual electricity bill (over all homes and system sizes) by MPC is only 3.4% and 4.5% higher than the optimal bill for Tesla Powerwall 1 and Powerwall 2 batteries, respectively. This is particularly interesting since the simple rule-

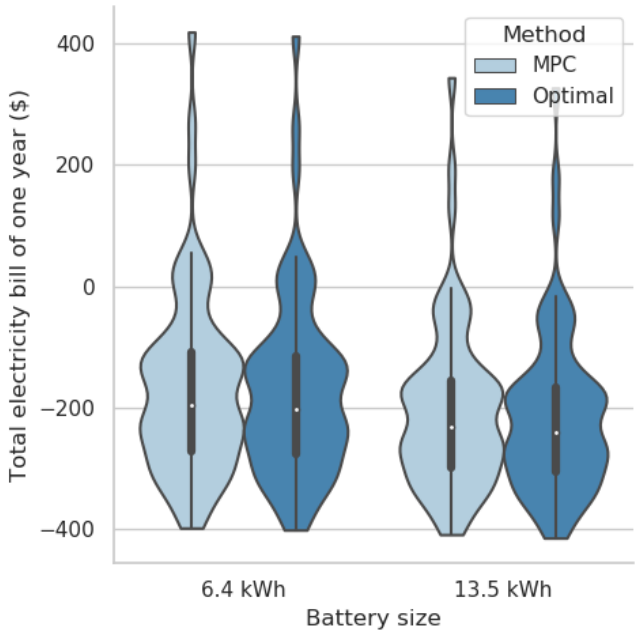


Figure 5.4: Comparing distributions of the annual electricity bill of homes equipped with a 8.8kWp PV system under hourly pricing scheme and solar export tariff of 0.03$/kWh.

based controller performs poorly under the hourly pricing scheme as it cannot figure out when to charge/discharge the battery without prediction. The DLC policy performs well under the hourly pricing scheme but it does not outperform MPC.

5.2.3 Financial Analysis

We investigate whether it makes sense financially to adopt one of the proposed optimal controllers by calculating ROI and break-even-period for a lithium-ion battery controlled by EnergyBoost. Table 5.1 summarizes the total installed cost (i.e., sum of the equipment price and installation costs) of each component of this system. As discussed in Section 5.1.1, we consider 2 sizes for the solar system and 2 sizes for the battery in addition to the scenario that no battery or no solar system is installed. The smaller solar system ($4.4kWp$) is comprised of 20 Canadian Solar CS5P-220M PV modules and the larger one ($8.8kWp$) is comprised of 40 of these modules. The smaller battery is a Tesla Powerwall 1 ($6.4kWh/2kW$) and the larger one is a Tesla Powerwall 2 ($13.5kWh/5kW$), both come with an integrated inverter. The total installed

Table 5.1: EnergyBoost's estimated cost breakdown

Component	**Size**	**Installed Cost (USD)**
Solar System & Micro Inverter	20 Modules 40 Modules	\$16,000 \$30,000
Battery & Inverter	$6.4kWh/2kW$ $13.5kWh/5kW$	\$6,500 \$9,000
Controller	—	\$65

cost of the PV system and inverter is estimated based on quotes from Google's Project Sunroof [21]. We assume a total of $\$6,250$ and $\$11,500$ in utility incentives and federal tax credits for residential PV systems of size $4.4kWp$ and $8.8kWp$, respectively, in Austin, Texas.

Similar to [34], we calculate ROI over a 20-year period for all homes in our data set. A positive ROI suggests that the initial investment is profitable over 20 years, whereas a negative ROI suggests the opposite. The ROI is defined as $\frac{Bill_{NS}-Bill_S-Cost}{Cost}$ where $Cost$ represents the capital expenditure for buying this system, and $Bill = Pay - Rev$ represents the difference between the amount paid to grid to buy electricity and the credit received from the grid for selling electricity. Subscripts NS and S represent the bill for the case with no system and for the case with a system, respectively. The system refers to either a solar system and a battery (Case A), or just a battery (Case B). In Case B, we assume the solar system was already installed and is considered as part of the no system case.

Both $Bill_{NS}$ and $Bill_S$ are calculated for the TOU pricing scheme, assuming that the TOU rates will not increase over the 20 years as suggested by the projection in [34]. We do the ROI calculation for different system sizes and solar export tariffs. To this end, we assume that each home consumes nearly the same amount of electricity each year and solar production is roughly the same each year. Since the projected electricity price remains constant, it is possible to calculate Pay_{NS} and Pay_S for the next 20 years in advance. This is done by multiplying Pay_{NS} and Pay_S of the first year by 20. Rev_{NS} and Rev_S are computed using $Rev = \sum_{i=1}^{20} \frac{Rev_c}{(1+f)^i}$ where f denotes the annual inflation rate

and Rev_c is the revenue of the first year. Clearly, Rev_{NS} is zero in Case A, while it would take a non-zero value in Case B. Assuming an annual inflation rate of 2%, we simply compute the ROI for each control method with and without government subsidies in both cases.

We first examine Case A where we aim to understand whether installing a battery and a solar system makes sense financially in a home which was not previously equipped with any of these systems. Our ROI results indicate that government incentives play a key role in making this installation profitable in 20 years since even assuming a solar export tariff as high as $15.4¢/kWh$, only 82% of homes equipped with a 4.4kWp solar system and a Tesla Powerwall 2 controlled by MPC have a positive ROI without the incentives. Nevertheless, all these homes will have a positive ROI with the same solar export tariff after including incentives. Considering the setting where homes are equipped with an 8.8kWp solar system and Tesla Powerwall 1 with the same solar export tariff as before, 95% of homes have a positive ROI if their battery is controlled by MPC; this reduces to 10% of homes if the battery is controlled by RBC-T. For the same setting but this time with a Tesla Powerwall 2 battery, all homes with the MPC controller and 32% of homes with the RBC-T controller have a positive ROI. Interestingly, if the solar export tariff is $7.7¢/kWh$, only five homes have a positive ROI with incentives if the battery is controlled by MPC.

Turning our attention to Case B, since there is no government incentives in Texas for installing home batteries, our results indicate that adding a battery to an existing PV installation is not profitable in most cases. Specifically, we witness that even with a solar export tariff as high as $15.4¢/kWh$, almost 20% (25%) of homes with an existing PV installation and a Tesla Powerwall 1 (Powerwall 2) have a positive ROI if the MPC policy is adopted. This implies that at current price points, installing a battery does not make sense economically for most customers unless new incentives are put in place to encourage integration of home battery.

In both cases, we observe that MPC always increases the profit made from installing a battery compared to RBC-T. We also calculate the break-even-period for different system sizes, tariff structures, and control policies. The

break-even-period is defined as the number of years required to operate the system to pay off the initial investment. Figures 5.6 and 5.7 this result.

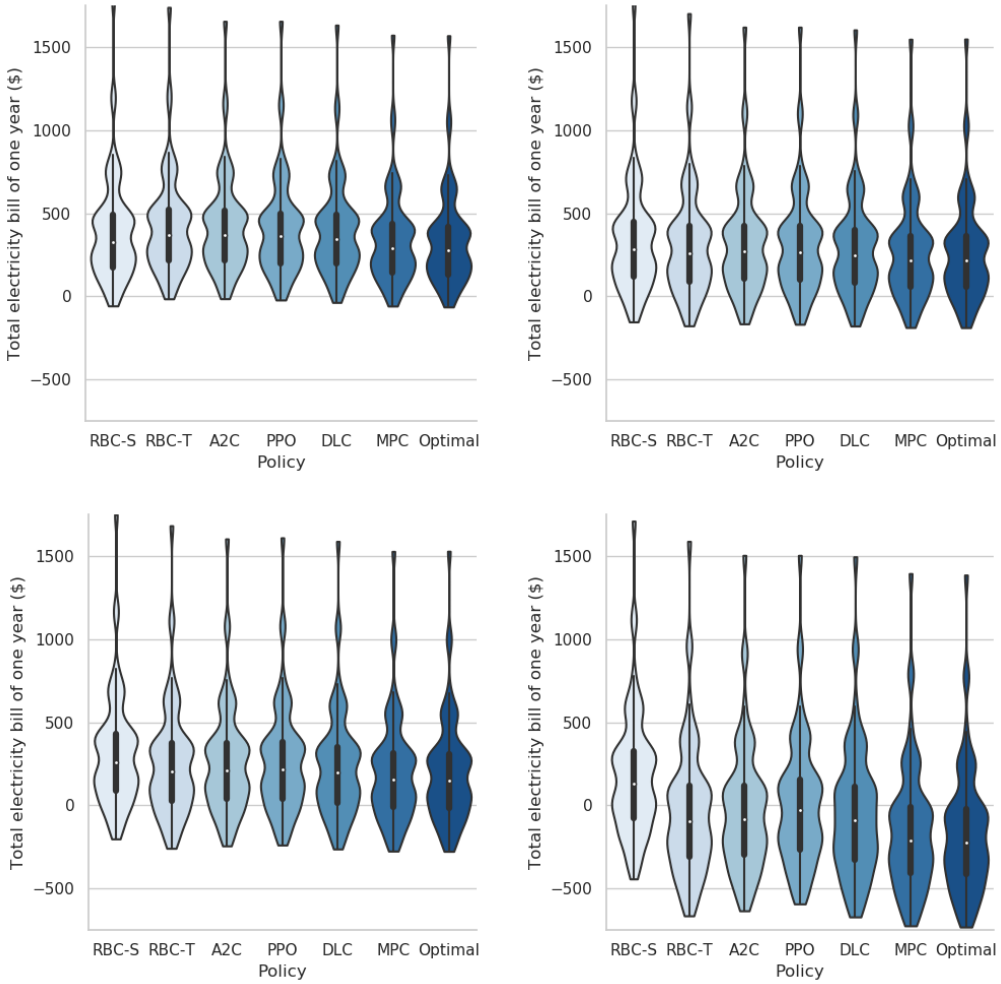


Figure 5.5: Distributions of annual electricity bill of homes equipped with a 4.4kWp PV system and a Tesla Powerwall 1 under TOU pricing scheme. The solar export tariff is 0.03$/kWh, 0.061$/kWh, 0.077$/kWh, and 0.154$/kWh (in order).

5.2.4 Practical Considerations

Runtime

We run EnergyBoost on a Raspberry Pi 3 Model B and measure the runtime of different controllers. We find that computing AC and AD for the next time slot takes on average 53.8, 65.7, 0.87, and 0.02 seconds for MPC, A2C, DLC, and RBC-T controllers, respectively[2]. Thus, actions can be taken at a faster

[2]We did not include the time it takes to train the neural network model for DLC.

timescale (up to around 1 minute) using any of these controllers, though in practice this may not be necessary as we discuss below.

Timescale of Control

To understand whether it makes sense to take control actions at a faster timescale we use 15-minute electricity consumption data which was available for most homes in our data set and compute the annual bill under different system sizes and solar export tariffs when the battery is controlled by MPC. We find that the annual electricity bill is 2.5% lower on average if we take control actions every 15 minutes compared to every 1 hour. We argue that this is the fastest timescale we can control a home battery in practice because weather and climate data are usually unavailable at a faster timescale, unless homeowners install their own sensors which would be costly.

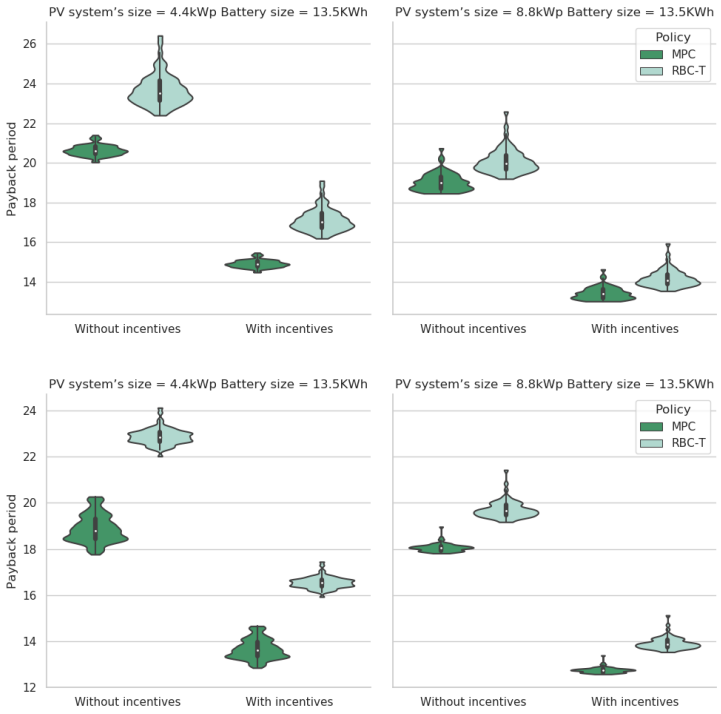


Figure 5.6: Violin plot for the payback period (# years) of homes when the solar export tariff is 0.154$/kWh.

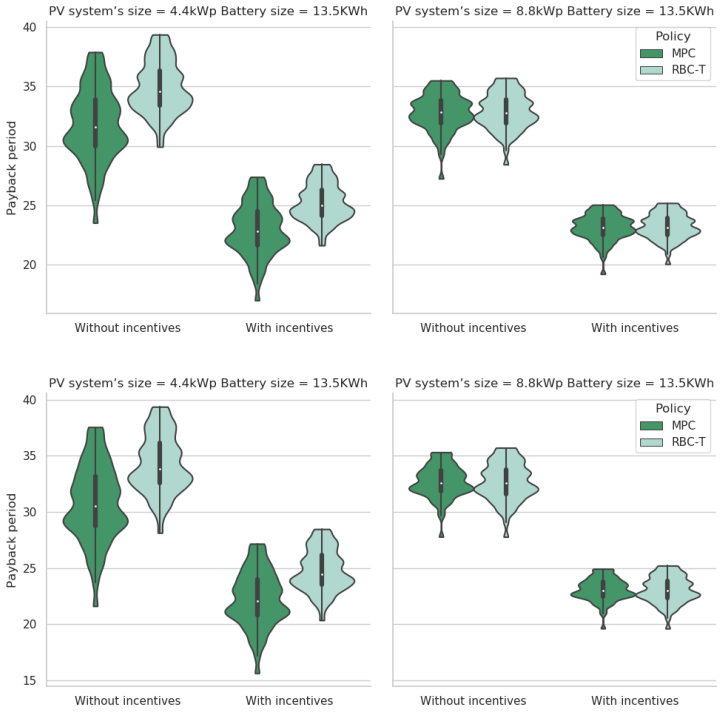


Figure 5.7: Violin plot for the payback period (# years) of homes when the solar export tariff is 0.077$/kWh.

Chapter 6

Conclusion

Solar-plus-battery systems are becoming increasingly popular around the world. And yet most of these systems are currently being controlled using simple rule-based methods which appear to be suboptimal. In this thesis, we studied the application of learning-based methods to determine the charge or discharge power of the lithium-ion battery, which is part of the solar-plus-battery system, to reduce the monthly electricity bill. In particular, we focused on model-based and model-free control methods, and empirically evaluated their performance.

The proposed learning-based control methods utilize real-time measurements and historical data about solar generation, home energy consumption, and electricity and solar export tariffs to determine the optimal operation of the battery. To evaluate the performance of different controllers in terms of their ability to cut the monthly electricity bill, we solved the mixed integer linear program assuming perfect information about future household demands and solar energy productions, and implemented two baseline controllers which rely on simple rules to operate the battery. We analyzed the economic feasibility of the system by calculating the return on investment (ROI) and break-even period for different system sizes and electricity prices with or without the government subsidies. We packaged and integrated all developed models and proposed control algorithms into EnergyBoost and ran this software on a Raspberry Pi connected to the Internet to empirically validate our results.

Our results indicated strengths and weaknesses of different learning-based methods when it comes to controlling a physical system with several con-

straints and imperfections. We showed that the underlying optimization problem is non-convex and therefore inherently complex. The environment is also non-stationary. Nevertheless, we showed that the best learning-based controller (i.e., MPC) outperforms baseline controllers in terms of the annual electricity bill, and closely tracks the minimum bill that can be theoretically achieved. It also reduces the payback period by more than 44 months on average compared to the rule-based controller. We provided insight about the size of battery storage that could generate revenue under various tariff structures and discussed the impact of changing the control timescale on the homeowner's bill. We found that only in a certain scenario with a sizable battery and solar panel, the system could be profitable for the homeowner in less than 20 years. This is assuming that the battery will not reach its end of life in 20 years.

Our work has some limitations that warrant further investigation:

- We only studied how the system could benefit the homeowner without considering how it could benefit the grid. Specifically, we did not study how optimal controllers can be modified to lower the system's peak-to-average ratio or offer other services (such as voltage support) to the grid. In future work, we will take these objectives into account and explore how we can benefit the homeowner and grid at the same time.

- We did not compare the proposed controllers under the tiered pricing scheme. Under tiered pricing, customers pay a flat rate for buying electricity from the grid as long as their demand is below a certain level in some time period. If their demand exceeds this level, they will be charged at a higher rate. Controlling the battery in a jurisdiction that implements this pricing scheme is more challenging as it requires changing the objective function and setting the control horizon to 1 month. This will significantly increase the running time of all algorithms and will create a credit assignment issue for the reinforcement learning technique.

- We did not discuss how the solar micro-inverter can be jointly controlled with the battery to stabilize the distribution voltage, while maximizing

the revenue of homeowners. Considering the possibility of controlling both systems, the problem can be solved for a neighbourhood. We also intend to study what will happen if all homes in a neighbourhood decide to control their system using one of the proposed methods.

- We found that the reinforcement learning methods are not performing quite well in most cases. We attribute this to the non-stationary environment and the small number of episodes we considered when training the reinforcement learning agent. The primary reason for the poor performance of the policy gradient methods is that they did not get sufficient time to gain enough experience. Increasing the number of episodes could improve the result, but it comes at the cost of increasing the runtime of the algorithm which is a concern for the real-world applications. Moreover, only online learning algorithms are considered in this thesis. We will explore how the reinforcement learning methods can be improved in terms of their execution time and performance.

Despite the above-mentioned limitations, this thesis describes one of the first attempts to create a comprehensive framework for controlling battery energy storage systems in the smart grid. We hope that this framework is useful for other researchers in the community, and could facilitate developing and evaluating new control methods.

References

[1] K. Abdulla *et al.*, "Optimal operation of energy storage systems considering forecasts and battery degradation," *Transactions on Smart Grid*, vol. 9, no. 3, pp. 2086–2096, May 2018. 11, 13, 18

[2] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM journal on Optimization*, vol. 5, no. 1, pp. 13–51, 1995. 16

[3] O. Ardakanian *et al.*, "Computing electricity consumption profiles from household smart meter data," in *Proceedings of the Workshops of the EDBT/ICDT Joint Conference*, 2014, pp. 140–147. 29

[4] O. Ardakanian, S. Keshav, and C. Rosenberg, *Integration of Renewable Generation and Elastic Loads into Distribution Grids*, ser. SpringerBriefs in Electrical and Computer Engineering. Springer International Publishing, 2016, ISBN: 9783319399843. 14

[5] O. Babacan, E. L. Ratnam, V. R. Disfani, and J. Kleissl, "Distributed energy storage system scheduling considering tariff structure, energy arbitrage and solar PV penetration," *Applied Energy*, vol. 205, pp. 1384–1393, 2017. 10

[6] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*, 2. Athena scientific Belmont, MA, 1995, vol. 1. 17

[7] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013. 22

[8] W. E. Boyson, G. M. Galbraith, D. L. King, and S. Gonzalez, "Performance model for grid-connected photovoltaic inverters.," Sandia National Laboratories, Tech. Rep., 2007. 26

[9] L. Breiman, *Classification and regression trees*. Routledge, 2017. 32

[10] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016. 48

[11] M. Caudill, "Neural networks primer, part i," *AI expert*, vol. 2, no. 12, pp. 46–52, 1987. 21

[12] ComEd, *Hourly Pricing program API*, Online `https://hourlypricing.comed.com/hp-api/`, 2019, Retrieved. 28, 60

[13] T. Degris, P. M. Pilarski, and R. S. Sutton, "Model-free reinforcement learning with continuous action in practice," in *2012 American Control Conference (ACC)*, Montreal, QC, Canada, 2012, pp. 2177–2182. 47

[14] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016. 7

[15] N. Ebell, F. Heinrich, J. Schlund, and M. Pruckner, "Reinforcement learning control algorithm for a pv-battery-system providing frequency containment reserve power," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, IEEE, Oct. 2018, pp. 1–6. 11

[16] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004. 31

[17] eGauge, *Home Energy Metering*, Online `https://www.egauge.net/eos/energy-meters/`, 2018, Retrieved. 24

[18] A. Eisenblätter, M. Grötschel, and A. M. Koster, "Frequency planning and ramifications of coloring," 2000. 16

[19] F. Fioretto, W. Yeoh, and E. Pontelli, "A multiagent system approach to scheduling devices in smart homes," in *Proc. 16th Conference on Autonomous Agents and MultiAgent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 981–989. 15

[20] C. Gaskett, D. Wettergreen, and A. Zelinsky, "Q-learning in continuous state and action spaces," *Advanced Topics in Artificial Intelligence*, pp. 417–428, 1999. 20

[21] Google, *Project Sunroof*, Online `https://www.google.com/get/sunroof`, 2018, Retrieved. 63

[22] M. Grötschel, L. Lovász, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, no. 2, pp. 169–197, 1981. 16

[23] C. Guan *et al.*, "Reinforcement learning-based control of residential energy storage systems for electric bill minimization," in *Proc. 12th Consumer Communications and Networking Conference*, IEEE, 2015, pp. 637–642. 4, 11, 21

[24] L. Gurobi Optimization, *Gurobi optimizer reference manual*, 2019. [Online]. Available: `http://www.gurobi.com`. 7

[25] J. Han, S. K. Solanki, and J. Solanki, "Coordinated predictive control of a wind/battery microgrid system," *IEEE Journal of emerging and selected topics in power electronics*, vol. 1, no. 4, pp. 296–305, 2013. 17

[26] S. S. Haykin *et al.*, *Neural networks and learning machines/Simon Haykin.* New York: Prentice Hall, 2009. 33

[27] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, Elsevier, 1992, pp. 65–93. 22

[28] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970. 30

[29] W. F. Holmgren, R. W. Andrews, A. T. Lorenzo, and J. S. Stein, "Pvlib python 2015," in *Photovoltaic Specialist Conference (PVSC), 2015 IEEE 42nd*, IEEE, 2015, pp. 1–5. 6, 26

[30] J. de Hoog, K. Abdulla, R. R. Kolluri, and P. Karki, "Scheduling fast local rule-based controllers for optimal operation of energy storage," in *Proc. 9th International Conference on Future Energy Systems (e-Energy)*, ACM, 2018, pp. 168–172. 4, 5, 11, 18

[31] Q. Hu and F. Li, "Hardware design of smart home energy management system with dynamic price response," *Transactions on Smart grid*, vol. 4, no. 4, pp. 1878–1887, 2013. 14

[32] Z. Huang, T. Zhu, Y. Gu, D. Irwin, A. Mishra, and P. Shenoy, "Minimizing electricity costs by sharing energy in sustainable microgrids," in *Proc. 1st Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys)*, ACM, 2014, pp. 120–129. 14

[33] International Renewable Energy Agency, *Electricity storage and renewables: Costs and markets to 2030*, Online `http://www.irena.org/publications/2017/Oct/Electricity-storage-and-renewables-costs-and-markets`, 2017, Retrieved. 1

[34] F. Kazhamiaka, P. Jochem, S. Keshav, and C. Rosenberg, "On the influence of jurisdiction on the profitability of residential photovoltaic-storage systems: A multi-national case study," *Energy Policy*, vol. 109, pp. 428–440, 2017. 10, 63

[35] F. Kazhamiaka, S. Keshav, and C. Rosenberg, "Adaptive battery control with neural networks," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ACM, 2019, pp. 536–543. 4, 22

[36] F. Kazhamiaka, C. Rosenberg, S. Keshav, and K.-H. Pettinger, "Li-ion storage models for energy system optimization: The accuracy-tractability tradeoff," in *Proc. 7th International Conference on Future Energy Systems (e-Energy)*, ACM, 2016, 17:1–17:12. 6, 25

[37] M. Koller, T. Borsche, A. Ulbig, and G. Andersson, "Defining a degradation cost function for optimal control of a battery energy storage system," in *2013 IEEE Grenoble Conference*, Jun. 2013, pp. 1–6. 13

[38] J. A. Kratochvil, W. E. Boyson, and D. L. King, "Photovoltaic array performance model.," Sandia National Laboratories, Tech. Rep., 2004. 26

[39] A. Lazaric, M. Restelli, and A. Bonarini, "Reinforcement learning in continuous action spaces through sequential Monte Carlo methods," in *Advances in neural information processing systems*, 2008, pp. 833–840. 20

[40] T. Li and M. Dong, "Residential energy storage management with bidirectional energy control," *Transactions on Smart Grid*, pp. 1–1, 2018. 12

[41] J. L. Mathieu, M. Kamgarpour, J. Lygeros, and D. S. Callaway, "Energy arbitrage with thermostatically controlled loads," in *Proc. European Control Conference (ECC)*, 2013, pp. 2519–2526. 14

[42] A. Mishra *et al.*, "Smartcharge: Cutting the electricity bill in smart homes with energy storage," in *Proc. 3rd International Conference on Future Energy Systems (e-Energy)*, ACM, 2012, 29:1–29:10. 4, 12, 16, 27, 30

[43] ——, "Greencharge: Managing renewableenergy in smart buildings," *Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1281–1293, 2013. 12

[44] A. Mishra, D. Irwin, P. Shenoy, and T. Zhu, "Scaling distributed energy storage for grid peak reduction," in *Proc. 4th International conference on Future energy systems (e-Energy)*, ACM, 2013, pp. 3–14. 14

[45] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015. 43–45

[46] ——, "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd International Conference on Machine Learning*, vol. 48, PMLR, 2016, pp. 1928–1937. 47

[47] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *Transactions on Smart Grid*, vol. 1, no. 2, pp. 120–133, 2010. 14

[48] A.-H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *Transactions on Smart Grid*, vol. 1, no. 3, pp. 320–331, 2010. 14

[49] V. Muenzel, J. de Hoog, M. Brazil, A. Vishwanath, and S. Kalyanaraman, "A multi-factor battery cycle life prediction methodology for optimal battery management," in *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*, ser. e-Energy '15, Bangalore, India, 2015, pp. 57–66. 13

[50] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied linear statistical models*. Irwin Chicago, 1996, vol. 4. 30

[51] W. K. Newey, "Adaptive estimation of regression models via moment restrictions," *Journal of Econometrics*, vol. 38, no. 3, pp. 301–339, 1988. 49

[52] NREL, *Measurement and instrumentation data center*, Online `https://midcdmz.nrel.gov/`, 2018, Retrieved. 28

[53] ——, *U.S. Solar Photovoltaic System Cost Benchmark: Q1 2017*, Online `https://www.nrel.gov/docs/fy17osti/68925.pdf`, 2018, Retrieved. 1

[54] B. Nykvist and M. Nilsson, "Rapidly falling costs of battery packs for electric vehicles," *Nature Climate Change*, vol. 5, no. 4, pp. 329–332, 2015. 1

[55] E. Oki, *Linear programming and algorithms for communication networks: a practical guide to network design, control, and management*. CRC Press, 2012. 16

[56] Ontario Energy Board, *Ontario TOU Pricing*, Online `https://www.oeb.ca/rates-and-your-bill/electricity-rates/historical-electricity-rates`, 2018, Retrieved. 28

[57] A. Oudalov, R. Cherkaoui, and A. Beguin, "Sizing and optimal operation of battery energy storage system for peak shaving application," in *Power Tech*, IEEE, 2007, pp. 621–625. 15

[58] N. G. Paterakis, O. Erdinc, A. G. Bakirtzis, and J. P. Catalão, "Optimal household appliances scheduling under day-ahead pricing and load-shaping demand response strategies," *Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1509–1519, 2015. 12

[59] Pecan Street, *Dataport*, Online `https://dataport.cloud`, 2018, Retrieved. 27

[60] B. Qi, M. Rashedi, and O. Ardakanian, "Energyboost: Learning-based control of home batteries," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ACM, 2019, pp. 239–250. 23

[61] V. H. Quintana, G. L. Torres, and J. Medina-Palomo, "Interior-point methods and their applications to power systems: A classification of publications and software codes," *IEEE Transactions on power systems*, vol. 15, no. 1, pp. 170–176, 2000. 16

[62] S. Quoilin, K. Kavvadias, A. Mercier, I. Pappone, and A. Zucker, "Quantifying self-consumption linked to solar home battery systems: Statistical analysis and economic assessment," *Applied Energy*, vol. 182, pp. 58–67, 2016. 14

[63] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. Jennings, "Agent-based control for decentralised demand side management in the smart grid," in *Proc. 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, ser. AAMAS, International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 5–12. 14

[64] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, Springer, 2003, pp. 63–71. 33

[65] E. L. Ratnam, S. R. Weller, and C. M. Kellett, "An optimization-based approach to scheduling residential battery storage with solar PV: Assessing customer benefit," *Renewable Energy*, vol. 75, pp. 123–134, 2015. 10

[66] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998. 16

[67] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, 2015, pp. 1889–1897. 48

[68] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *ArXiv*, vol. abs/1707.06347, 2017. 48

[69] Y. Shi, B. Xu, Y. Tan, and B. Zhang, "A convex cycle-based degradation model for battery energy storage planning and operation," in *2018 Annual American Control Conference (ACC)*, IEEE, 2018, pp. 4590–4596. 14

[70] D. Silver *et al.*, "Deterministic policy gradient algorithms," in *Proc. 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14, Beijing, China: JMLR.org, 2014, pp. I-387–I-395. 47

[71] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1. 18, 20, 21, 43, 44, 47, 48

[72] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, ser. NIPS'99, Denver, CO: MIT Press, 1999, pp. 1057–1063. 47

[73] TESLA, *Powerwall*, Online `https://www.tesla.com/sites/default/files/pdfs/powerwall/Powerwall%202_AC_Datasheet_en_northamerica.pdf`, 2018, Retrieved. 54

[74] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. 31

[75] C. N. Truong, M. Naumann, R. C. Karl, M. Müller, A. Jossen, and H. C. Hesse, "Economics of residential photovoltaic battery systems in germany: The case of tesla's powerwall," *Batteries*, vol. 2, no. 2, p. 14, 2016. 5

[76] Z. Wang, C. Gu, F. Li, P. Bale, and H. Sun, "Active demand response using shared energy storage for household energy management," *Transactions on Smart Grid*, vol. 4, no. 4, pp. 1888–1897, 2013. 14

[77] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006. 29

[78] P. Zhang, *Advanced industrial control technology*. William Andrew, 2010. 17

[79] X. Zhou, S.-J. Hsieh, B. Peng, and D. Hsieh, "Cycle life estimation of lithium-ion polymer batteries using artificial neural network and support vector machine with time-resolved thermography," *Microelectronics Reliability*, vol. 79, pp. 48–58, 2017. 13

[80] T. Zhu, Z. Huang, A. Sharma, J. Su, D. Irwin, A. Mishra, D. Menasche, and P. Shenoy, "Sharing renewable energy in smart microgrids," in *Proc. International Conference on Cyber-Physical Systems (ICCPS)*, IEEE, 2013, pp. 219–228. 14

[81] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American statistical association*, vol. 101, no. 476, pp. 1418–1429, 2006. 31