University of California
Santa Barbara

# Stochastic Control of Energy Systems: A Statistical Learning Framework

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Statistics & Applied Probability

by

Aditya Maheshwari

Committee in charge:

Professor Michael Ludkovski, Chair
Professor Jean-Pierre Fouque
Professor Nils Detering

December 2019

The Dissertation of Aditya Maheshwari is approved.

---

Professor Jean-Pierre Fouque

---

Professor Nils Detering

---

Professor Michael Ludkovski, Committee Chair

June 2019

Stochastic Control of Energy Systems: A Statistical Learning Framework



Aditya Maheshwari

To my parents and sister.

## Acknowledgements

Before we jump into any of the technical matter of this thesis, I would like to express my gratitude to my advisor and role model, Prof. Michael Ludkovski. This thesis would not be possible without his help, advice and hours he has invested in me. His clarity of thought and ability to envision key goals of a project right at the start has been indispensable in finishing the projects we have worked together. Besides teaching me how to write mathematically precise research papers, I hope to take-away strong work ethic he exemplifies in his life and work.

I am also thankful to Prof. Jean-Pierre Fouque and his seminar classes which have been key contributors to my understanding of financial mathematics. Prof. Raya Feldman's class on Probability theory and stochastic processes has been instrumental in instigating my interest in the subject. I am indebted to her for a clean and rigorous introduction of the subject matter and her class notes were nothing less than Bhagavad Gita to me. Her help throughout my first year and during preparation of the qualifying exams were paramount in developing my confidence in the graduate program. I am thankful to Prof. Nils Detering for agreeing to be in my committee and Prof. Tomoyuki Ichiba with whom I worked most as a TA.

My collaborators have been my key asset throughout my graduate program and I thank them for helping me improve as a researcher. I am also grateful to the financial support from National Science Foundation.

This thesis would be incomplete without expressing my gratitude towards my friends who made my stay in UCSB a home away from home. Thank you Neeraj Kumar for being a fun housemate and go-to-guy for all technical issues. Nhan Huynh for standing by my side all through last four years and many insightful

discussions on Gaussian Processes. Pratik Soni for being a friend I can always count on. Brian Wainwright for hours of conversations on life, science and research. Andrea Angiuli for being the point person on probability theory, real and functional analysis questions. Achyuthan J.R. for being an exceptional friend at all times for past 6 years.

Finally, I want to thank my parents for supporting me in all my decisions. The constant motivation from them, my sister and brother-in-law have been the key force which kept me going though both, good and bad times.

# Curriculum Vitæ
Aditya Maheshwari

## Education

| | |
|---|---|
| 2019 | Ph.D. in Statistics and Applied Probability (Expected), University of California, Santa Barbara. |
| 2018 | M.A. in Statistics, University of California, Santa Barbara. |
| 2015 | M.Sc. in Mathematics, McMaster University, Canada. |
| 2013 | M.Sc.(Integrated) in Economics, Indian Institute of Technology, Kanpur. |

## Working paper

- Maheshwari, A., Heleno, M., Ludkovski, M., *The Effect of Rate Design on Power Distribution Reliability Considering Adoption of DERs,* 2019

## Preprint

- Balata, A., Ludkovski, M., Maheshwari, A. and Palczewski, J. , *Statistical Learning for Probability-Constrained Stochastic Optimal Control,* arXiv:1905.00107, 2019

## Publications

- Ludkovski, M. and Maheshwari, A. *Simulation Methods for Stochastic Storage Problems: A Statistical Learning Perspective*, to appear in Energy Systems, 2019
- Alasseur, C., Balata, A., Ben Aziza, S., Maheshwari, A., Tankov, P., Warin, X. *Regression Monte Carlo for Microgrid Management*, ESAIM: Proceedings and Surveys, Vol 65, 46-67, 2019
- Maheshwari, A. and Sarantsev, A. *Modeling Systemic Risk with Interbank Flows, Borrowing and Investing*, Risks, Special Issue "Systemic Risk in Finance and Insurance", Vol 6, Issue 4, 2018.
- Grasselli, M. and Maheshwari, A. *Testing a Goodwin model with general capital accumulation rate*, Metroeconomica, Vol 69, Issue 3, 619-643, 2018.
- Grasselli, M. and Maheshwari, A. *A comment on "Testing Goodwin: growth cycles in ten OECD countries"*, Cambridge Journal of Economics, Vol 41, Issue 6, 1761-7166, 2017.

## Abstract

Stochastic Control of Energy Systems: A Statistical Learning Framework

by

Aditya Maheshwari

The overarching theme of this dissertation is to develop algorithms to efficiently solve finite horizon stochastic optimal control (SOC) problems. These problems naturally arise in the context of microgrid management where a controller is trying to optimally dispatch diesel generator or battery storage to maintain reliable supply of power. A popular approach is to formulate the microgrid management as a deterministic optimal control (DOC) and solve it using mixed integer linear program. However, this formulation fails to incorporate the stochasticity in the models and crucially relies on linearization of the objective/constraints. As a result, we formulate it as a SOC and consider two variants of it, first with explicit constraints on no-blackouts and second with implicit constraints on the probability of blackouts.

We investigate Regression Monte Carlo (RMC), a simulation based approach to recursively solve the Bellman's dynamic programming equation (DPE) for SOC. The proposed algorithms convert the SOC into a recursive sequence of statistical learning tasks. In addition to estimating the conditional expectation encapsulated in the Bellman's DPE, they also find the set of admissible controls for probability constrained SOC. One of our main contributions is the bridge between statistical learning and numerical methods for SOC. The algorithms presented in this dissertation also generalize existing approaches within the RMC paradigm, pro-

vide additional features for efficient implementation and extends RMC to include probabilistic constraints. Besides microgrid management, we also benchmark the performance of the algorithms for the valuation of natural gas storage.

In the final part of this dissertation we study the link between electricity tariffs and reliability of the distribution network. We assume the consumers at each node in the distribution network invest in behind-the-meter resources such as photovoltaic (PV) system and electrical storage. An industry model, *Distributed Energy Resources–Customer Adoption Model (DER–CAM)*, based on DOC is used to compute the optimal size of investments and dispatch of PV and storage. We use PG&E 69-bus distribution network to assess several different aspects of electricity tariffs that can impact the reliability; such as homothetic change in the electricity purchase rate, change in the magnitude of the peak purchase rate, and the time-of-day of peak purchase rate. The work provides a new tool to the regulators for improving reliability of the distribution network.

# Contents

**Curriculum Vitae** **vii**

**Abstract** **viii**

**1 Introduction** **1**
1.1 Permissions and Attributions . . . . . 9

**2 Stochastic Optimal Control and Regression Monte Carlo** **10**
2.1 Stochastic Optimal Control . . . . . 10
2.2 Dynamic Programming Equation . . . . . 13
2.3 Other Methods . . . . . 19
2.4 Additional Applications . . . . . 20

**3 Regression Monte Carlo for Microgrid Management** **23**
3.1 Introduction . . . . . 24
3.2 Microgrid . . . . . 26
3.3 Stochastic control formulation . . . . . 31
3.4 Numerical Resolution . . . . . 35
3.5 Numerical Experiments . . . . . 43
3.6 Summary . . . . . 54

**4 Simulation Methods for Stochastic Storage Problems: A Statistical Learning Perspective** **55**
4.1 Introduction . . . . . 56
4.2 Problem description . . . . . 59
4.3 Dynamic Emulation Algorithm . . . . . 66
4.4 Approximation Spaces . . . . . 73
4.5 Simulation Design . . . . . 79
4.6 Natural Gas Storage Facility . . . . . 87
4.7 Microgrid Balancing under Stochastic Net Demand . . . . . 102

4.8 Summary . . . . . 105

**5 Statistical Learning for Probability-Constrained Stochastic Optimal Control 108**
5.1 Introduction . . . . . 109
5.2 Problem formulation . . . . . 112
5.3 Probability Constrained-DEA . . . . . 121
5.4 Admissible set estimation . . . . . 130
5.5 Case Studies . . . . . 143
5.6 Summary . . . . . 157

**6 Impact of Electricity Tariffs on Distribution Network Reliability with Behind-the-meter Investments 159**
6.1 Introduction . . . . . 160
6.2 Behind-the-meter Investments and Optimal Control . . . . . 162
6.3 Distribution System . . . . . 166
6.4 Reliability Evaluation . . . . . 172
6.5 Numerical Example . . . . . 175
6.6 Summary . . . . . 183

**Bibliography 186**

# Chapter 1

# Introduction

Twentieth century was transformative due to substantial increase in our capability for generation, transmission and distribution of electric power at large distances. In the twenty first century, the technology for generation and distribution of electric power is poised to change. The momentum is now shifting to "decentralization" and "decarbonization" of electricity grids through higher integration of renewable generation units (e.g., solar photovoltaic (PV) arrays and wind turbines) combined with battery energy storage system. Microgrids will play an important role in this new transition due to their ease of adoption, ability to reduce the carbon footprint of the community, and additional economic incentives through reduced electricity purchase from utility. They can also help improve reliability by reducing the dependency on the main grid and acting as a back-up power during natural disasters, such as hurricanes [1].

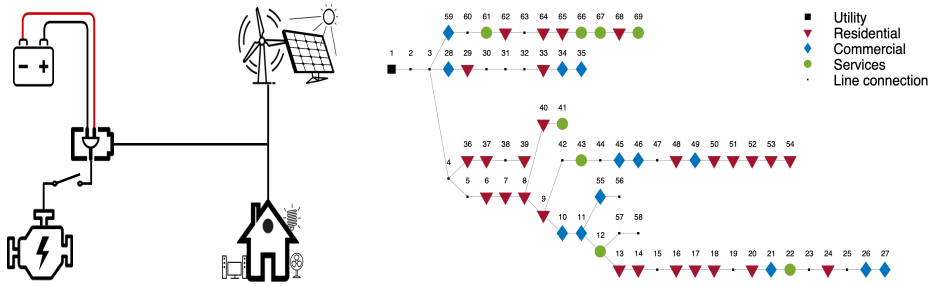
US Department of Energy defines microgrid as "*a group of interconnected loads and distributed energy resources within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid. A microgrid*

[1] https://www.princeton.edu/news/2014/10/23/two-years-after-hurricane-sandy-recognition-princetons-microgrid-still-surges

*can connect and disconnect from the grid to enable it to operate in both grid-connected or island-mode*" [1]. The elementary purpose of a microgrid is to provide a continuous electricity supply from the power produced by renewable generators while minimizing the installation and running costs. In this kind of systems, the variability of both the load and the renewable production is high and its negative effect on the system stability can be mitigated by including a battery energy storage system in the microgrid. Energy storage devices ensure power quality, including frequency and voltage regulation (see [2]) and provide backup power in case of any contingency. A dispatchable unit in the form of diesel generator is also used as a backup solution and to provide baseload power. In figure 1.1a we present an example of a microgrid topology that we will consider in this dissertation. It contains renewable generation through solar PV, a dispatchable battery energy storage system and a back-up diesel generator, together they will be used to meet demand of electric power from the consumers. Being an isolated microgrid it has no connection to the main grid.

Next, we take a macro view and present a typical structure of a distribution network in Figure 1.1b. Each node in Figure 1.1b represents a consumer (or connection node) who may or may not own a microgrid. In contrast with Figure 1.1a, here the consumer can buy (sell) power from (to) the main grid (node 1, represented via black square) and the corresponding microgrid can either work independently or in coordination with other microgrids in the network. Presence of microgrid at the location of the consumer also illustrates the current trend from centralized to decentralized generation where a consumer can use a mix of power purchased from the utility and generated locally to meet the demand of electricity.

An inspiration for our work is a microgrid in Chile, where "*the University of*




(a) Microgrid (b) Distribution Grid

Figure 1.1: *Left panel:* Topology of a microgrid comprising of diesel generator, battery storage, PV and households. *Right panel:* Line diagram of modified PG&E 69-bus system with three customer types. Commercial customers include seven different building categories including restaurants, super markets, hotels and malls. Services customers include schools, hospitals and government offices. Residential customers include mid-rise apartments.

*Chile has developed Chile's first microgrid project in a remote Andes Mountains community of 150 residents (mostly miners and their families) called Huatacondo. Prior to the microgrid installation, the community had its own electric network (operating independently from the macro-grid) operating 10 hours per day with power provided from a single diesel generator. The vision of the microgrid was to continue using that diesel generator but supplement it with distributed energy resources, namely solar PV, wind, and a battery system. The microgrid includes a 150 kW diesel generator, 22 kW tracking solar PV system, a 3 kW wind turbine, a 170 kWh battery, and an energy management system."*[2] In Figure 1.2, we present the load demand and power supply from solar PV using the data for this microgrid. Notice two features in the data, first the stochasticity observed in the load demand and in solar PV output. Second, during the afternoon the generation from PV exceeds the load demand, resulting in either curtailment of energy or storing it in battery for later use, motivating "smart" utilization of storage devices.

[2]https://building-microgrid.lbl.gov/huatacondo

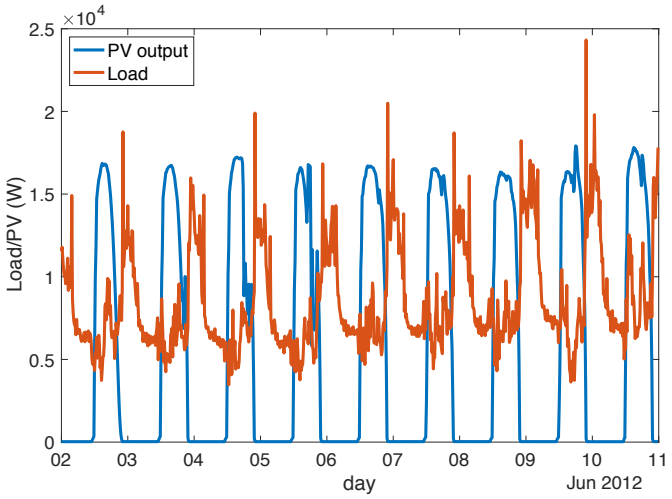



Figure 1.2: Historical load profile and PV output from a microgrid Huatacondo, Chile.

As microgrids become more ubiquitous, several engineering and mathematical questions arise. We consider three problems that naturally arise in the context of microgrid management:

- **Optimal usage:** Supply of power from the renewables and demand of electricity from consumers is stochastic (evident from figure 1.2), as it is difficult to claim with certainty when the sun will shine or wind will blow. This uncertainty raises practical questions on reliability of the power supply and optimal operational policy for the microgrid. Optimal operational policy is derived as a solution of a constrained optimization problem which minimizes the expected cost of running the diesel generator subject to physical constraints on the system components. Herein lie our most important contributions; in Chapters 3, 4 and 5 we propose algorithms to find optimal operational policy of microgrid considering the stochasticity in net demand (demand net of solar output) and constraints on reliability.

- **Investment size:** Investment costs in the renewables and storage increase linearly with the size of installation capacity, however, the benefits are often non-linear and present diminishing return. High investment costs in these technologies makes it important to compute the "optimal" battery capacity, PV size and diesel generator. In Chapters 3 and 6 we discuss optimal size of the storage for the microgrid.

- **Reliability:** As the penetration of PV increases, two engineering problems arise: (i) overgeneration during the day leading to voltage increase on the transmission lines and (ii) large ramp-up required during the evening to match demand [3]. One of the strategies to mitigate these challenges is to provide economic incentive to the consumers to alter their demand pattern. As a result, regulators often change tariff structure, such as peak time-of-use rate or demand charges, in an attempt to shift the peak demand and improve the reliability. In Chapter 6 we develop a framework to assess the impact of electricity tariffs on the reliability of a distribution network. Operational policy of a microgrid can be modified to improve reliability even in absence of external economic incentive. One such technique would be to add probabilistic constraints on the imbalance between demand and supply in the optimization problem; this method is explored in Chapter 5.

The contributions of this dissertation can be divided into two main themes: (i) application of theory of stochastic optimal control to energy systems with special attention to control of microgrid, and (ii) developing new tools to efficiently solve high dimensional stochastic optimal control problems. We divide the discussion into 6 chapters, content of which are briefly summarized below.

In Chapter 2 we provide a brief introduction to stochastic optimal control and discuss its connection to microgrid control. We present the dynamic programming principle and summarize the regression Monte Carlo (RMC) paradigm to solve it. We provide sketch of our main algorithm (dubbed *probability constrained dynamic emulation algorithm* (PC-DEA)) and how different chapters in this dissertation fit together to explain the key ideas of PC-DEA.

In Chapter 3 we model the optimal control of an islanded microgrid as a stochastic optimal control problem and solve it using RMC. This formulation of microgrid control was inspired by [4], however, unlike them we use Monte Carlo simulations to solve the dynamic programming equation. Applications of RMC for microgrid control is borrowed from mathematical finance where it is used for variety of problems, such as valuation of natural gas storage [5, 6, 7] and American option pricing [8, 9]. RMC algorithms are easily scalable to higher dimensions and can be adapted to large range of stochastic processes (e.g., Lévy processes, latent factors, piecewise linear processes, etc.). Scalability and adaptability are both absent when using methods in [4]. We also provide a technique to infer optimal size of the battery storage system.

Chapters 4 and 5 form the backbone of this dissertation. In Chapter 4, we discuss the challenges in the traditional implementation of RMC presented in Chapter 3 and borrow several techniques from statistical learning to improve its efficiency. Particularly, we propose a marriage of statistical learning with experimental design in the context of stochastic control via our Dynamic emulation algorithm (DEA). It synthesizes variety of existing approaches in a single modular template. Two important attributes of the DEA, input design and regression, are discussed in detail. We use natural gas storage valuation and microgrid control

as examples to illustrate DEA.

In Chapter 5 we extend the DEA to solve stochastic optimal control with probabilistic constraints. Thus, the algorithm in this chapter is referred to as PC-DEA. The motivation again is microgrid management, where the controller tries to find optimal usage of the diesel generator while maintaining low probability of blackouts. Compared to DEA, here the admissible set of controls (i.e. set of controls for which the probability constraint is respected) is implicit and not known a priori. We develop statistical models, taking as input the state vector and output as the set of admissible controls, to provide functional representation of the admissible sets for the continuous state space. This chapter provides a new application of statistical learning and uncertainty quantification techniques which we employ to approximate and then provide statistical guarantees regarding admissibility of state-action pairs.

In Chapter 6 we shift focus from developing tools for stochastic optimal control of a microgrid to understand the effect of electricity tariffs on the reliability of a distribution network. We rely on an industry model *Distributed Energy Resources – Customer Adoption Model* (DER–CAM) [10] which incorporates several practical aspects of a microgrid (e.g. efficiency of battery storage, purchase and sale of power to the main grid, etc.) that were ignored in Chapters 3–5. DER–CAM takes as input the load profile, tariff rate, solar irradiance, etc. and computes the optimal size of microgrid components and their dispatch policy via deterministic optimal control. We assume that the consumers at each node of the distribution network invest in microgrid to minimize the long-run economic cost of purchase of electricity. As a consequence, the distribution network comprises of a collection of microgrids operating locally to match demand with supply of power. This is

in contrast to the isolated microgrid of Chapters 3–5. The stochasticity in this chapter is due to potential random failure in the lines connecting the consumer to the main grid. Through a combination of deterministic optimal control via DER-CAM with the Monte Carlo simulation of line failures, we compute reliability metrics for the distribution network and determine their sensitivity to changes in electricity tariffs.

## 1.1 Permissions and Attributions

1. The content of Chapter 3 is the result of a collaboration with Clemence Alasseur, Alessandro Balata, Sahar Ben-Aziza, Peter Tankov and Xavier Warin, and has previously appeared in ESAIM: Proceedings and Surverys [11]. Authors started to work on this project during the $22^{\text{nd}}$ edition of CEMRACS which took place at CIRM, Marseille, France from July-August, 2017. The primary work for this paper was equally divided between Alessandro Balata and Aditya Maheshwari.

2. The content of Chapter 4 is the result of a collaboration with Michael Ludkovski, and has previously appeared in the Energy Systems [12].

3. The content of Chapter 5 is the result of a collaboration with Alessandro Balata, Michael Ludkovski and Jan Palczewski. The paper [13] is under review in the SIAM/ASA Journal on Uncertainty Quantification. The primary work was equally divided between Alessandro Balata and Aditya Maheshwari.

4. The content of Chapter 6 is the result of a collaboration with Miguel Heleno and Michael Ludkovski. The initial work on this project started with internship of Aditya Maheshwari at Lawrence Berkeley National Laboratory between September, 2018 and December, 2018. The paper [14] is under review in the IEEE Transactions on Power Systems.

# Chapter 2

# Stochastic Optimal Control and Regression Monte Carlo

Operating a dynamical system in the presence of stochastic noise to minimize a performance criterion is the subject of the theory of stochastic optimal control (SOC). The objective function of these problems typically involves minimizing the expected cost of operating the dynamical system in a finite or infinite time horizon. Under Markovian assumption for the system dynamics, the variable of optimization is a function in space and time representing the optimal control. In this chapter we first describe a general formulation of a SOC problem and its key ingredients. This is followed by a specific example on microgrid control. We conclude with a sketch of the algorithm to solve these control problems and alternates available in the literature.

## 2.1 Stochastic Optimal Control

Throughout this thesis, we will only consider SOC problems described over a finite horizon $[0, T]$. A SOC problem is described via the following quantities:

- State Variable $\mathbf{X}(t) \in \mathcal{X} \subset \mathbb{R}^d$: represents the state of the system at time

$t$ and we formalize its evolution through time via a controlled stochastic Markov process. Typically, we assume the dynamics of the state process via stochastic differential equation (SDE) of the form:

$$\mathrm{d}\mathbf{X}(t) = b(t, \mathbf{X}(t), u(t))\mathrm{d}t + \sigma(t, \mathbf{X}(t), u(t))\mathrm{d}B(t), \tag{2.1}$$

where $B(t)$ is a $m-$dimensional Brownian motion, $b : \mathbb{R}_+ \times \mathcal{X} \times \mathcal{W} \to \mathbb{R}^d$, $\sigma : \mathbb{R}_+ \times \mathcal{X} \times \mathcal{W} \to \mathbb{R}^{d\times m}$ and $u(t)$ is the control. The algorithms developed in this dissertation are agnostic to the specification of $\mathbf{X}(t)$ as long as we can simulate its trajectories. Thus, the dynamics of $\mathbf{X}(t)$ could include latent factors, Lévy processes, piecewise-linear processes, etc. Chapters 3 and 4 consider discrete time version of the SDE in Equation (2.1), i.e., the dynamics of $\mathbf{X}(t)$ is written via corresponding difference equations defined at discrete epochs $\{t_0, t_1, \ldots, t_N = T\}$.

- Control $u(t)$: represents the decision variable that drives the distribution of the state process $\mathbf{X}(t)$. We consider Markovian controls adapted to the filtration of $(\mathbf{X}(t))$, i.e., $u(t) \equiv u(t, \mathbf{X}(t))$. Besides being adaptive, the control may have to satisfy some problem specific constraints which restricts the choice of possible controls. The set of admissible controls is thus described through $\mathcal{U}$. Throughout this dissertation, we will assume that the control decisions are made at pre-determined discrete epochs $\{t_0, t_1, \ldots, t_N = T\}$. Thus, the control process is piecewise constant in time and described via vector of functions $\{u(t_n, \mathbf{X}(t_n))\}_{n=0}^N$. For the sake of brevity, we will sometimes write $\mathbf{X}(t_n) \equiv \mathbf{X}_n$ and $u(t_n, \mathbf{X}(t_n)) \equiv u_n$ at discrete epochs $t_n$. At every other time-point $s \in (t_n, t_{n+1})$ we continue to use the standard nota-

tion i.e., $\mathbf{X}(s), u(s)$.

- Value function $V(t, \mathbf{X}(t))$: The SOC problem is formulated as:

$$V(t_n, \mathbf{X}(t_n)) = \inf_{(u_s)_{s=n}^{N} \in \mathcal{U}_{n:N}(\mathbf{X}_n)} \Big\{ \mathbb{E}\Big[ \sum_{k=n}^{N-1} \int_{t_k}^{t_{k+1}} \pi_s(\mathbf{X}(s), u_k) \mathrm{d}s + K(\mathbf{X}_n, u_n) + W(\mathbf{X}(t_N)) \Big| \mathbf{X}_n \Big] \Big\} \tag{2.2}$$

where $W(\cdot)$ represents the terminal penalty, $\pi_t(\cdot, \cdot)$ the running cost, $K(\cdot, \cdot)$ the switching cost that incurs only at discrete time epochs when the controls are chosen and $\mathcal{U}_{n:N}(\mathbf{X}_n)$ represents the set of admissible controls in the interval $[t_n, t_N]$, given the state of the system $\mathbf{X}_n$ at time $t_n$. Remember that the state process $\mathbf{X}(t)$ is controlled and is affected by the dynamics of $u(t)$. Thus, formally we should write $\mathbf{X}(t) \equiv \mathbf{X}^u(t)$, however, we drop the superscript $u$ for brevity. A sequence of controls $\{u_k^*\}_{k=n}^{N} \in \mathcal{U}_{n:N}(\mathbf{X}_n)$ is optimal if

$$V(t_n, \mathbf{X}(t_n)) = \mathbb{E}\Big[ \sum_{k=n}^{N-1} \int_{t_k}^{t_{k+1}} \pi_s(\mathbf{X}(s), u_k^*) \mathrm{d}s + K(\mathbf{X}_n, u_k^*) + W(\mathbf{X}(t_N)) \Big| \mathbf{X}_n \Big]. \tag{2.3}$$

As an example, in the context of microgrid the state variable $\mathbf{X}(t)$ is three-dimensional comprising of the net demand $L(t)$ (demand net of solar output), state of charge of the battery storage system $I_t$ and the regime/state of the diesel generator $m(t) \in \{\text{off}, \text{on}\} \equiv \{0, 1\}$. Thus, $\mathbf{X}(t) = (L(t), I(t), m(t))$. The dynamics of the net demand $L(t)$ is exogenous i.e. not dependent upon the control, on the other hand, both $I(t)$ and $m(t)$ are endogenous and fully or partially controlled. The control $u(t) \in \{0\} \cup [\underline{u}, \overline{u}]$ represents the power output from diesel generator with $\underline{u}$ ($> 0$) and $\overline{u}$ ($> \underline{u}$) representing the minimum and maximum power output.

The controller incurs a positive switching cost $K > 0$ when the diesel generator is switched on $u(t) > 0$ from $m(t) = 0$, and zero switching cost $K = 0$ otherwise. The value function comprises of cost of using the diesel generator, the switching cost and the cost due to mismatch of demand and supply.

**Challenges:** The main challenge of SOC problems arises because the trajectory of the control $\{u_n\}_{n=0}^{N}$ affects the distribution of the state process $(\mathbf{X}(t))_{t\geq 0}$ as evident via equation (2.1). This in turn affects the expected running cost, which needs to be minimized. A naive approach from stochastic optimization literature will be to estimate expected running cost via nested simulations for a fixed trajectory of control and then optimize over the possible set of such trajectories. However, since the optimization in equation (2.2) is over the space of functions (piece-wise constant in time), the set of possible trajectories is "large" and thus makes this naive approach computationally intractable. To make matters worse, in some problems the admissible set $\mathcal{U}_{n:N}(\mathbf{X}_n)$ is implicitly defined and not known a priori. Thus, whether the control is admissible i.e. $\{u_n\}_{n=0}^{N-1} \in \mathcal{U}_{0:N}(\mathbf{X}_0)$ is not obvious. Furthermore, the admissible set $\mathcal{U}_{n:N}(\mathbf{X}_n)$ depends upon the state $\mathbf{X}_n$, which in turn depends upon the control decisions until time $t_n$ i.e. $\{u_k\}_{k=0}^{n-1}$. In sum, together these challenges make it non-trivial to estimate the sequence of optimal controls in equation (2.2).

## 2.2 Dynamic Programming Equation

The SOC problem described in equation (2.2) can be solved recursively using the Bellman's dynamic programming principle. Since the control $u(t)$ is piecewise constant, the dynamic programming equation corresponding to equation (2.2) at

step $n$ is given as:

$$
\begin{aligned}
V_n(\mathbf{X}_n) &= \inf_{u_n \in \mathcal{U}_n(\mathbf{X}_n)} \Big\{ \mathcal{C}_n(\mathbf{X}_n, u_n) \Big\}, \\
\text{where} \quad \mathcal{C}_n(\mathbf{X}_n, u_n) &= \mathbb{E}\left[ \pi^{\Delta}(\mathbf{X}_{n:n+1}, u_n) + V_{n+1}(\mathbf{X}(t_{n+1})) \Big| \mathbf{X}_n, u \right], \qquad (2.4) \\
\text{and} \quad \pi^{\Delta}(\mathbf{X}_{n:n+1}, u_n) &= \int_{t_n}^{t_{n+1}} \pi_s(\mathbf{X}(s), u_n) \mathrm{d}s + K(\mathbf{X}_n, u_n).
\end{aligned}
$$

Above $\mathcal{C}_n(\mathbf{X}_n, u_n)$ is the continuation value, i.e. reward-to-go plus expectation of future rewards, from using the control $u_n$ over $[t_n, t_{n+1})$ and $\mathcal{U}_n(\mathbf{X}_n) = \mathcal{U}_{n:n}(\mathbf{X}_n)$ represents the set of admissible controls satisfying the constraints at a single decision epoch $t_n$ conditional on $\mathbf{X}_n$. Moreover, given the state $\mathbf{X}_n$, we say that $u_n^* \in \mathcal{U}_n(\mathbf{X}_n)$ is an optimal control if $V_n(\mathbf{X}_n) = \mathcal{C}_n(\mathbf{X}_n, u_n^*)$ and since the state dynamics is Markovian, the optimal control is also of feedback type, i.e., $u_n^* \equiv u^*(t_n, \mathbf{X}_n)$. Thus, the DPE (2.4) reduces the problem of searching the sequence of optimal controls $\{u_k^*\}_{k=n}^N$ into two-step procedure. First step computes the value function $V_{n+1}(\mathbf{x})$ for any arbitrary $\mathbf{x}$, and second optimal control $u_n^* : (t_n, \mathbf{X}_n) \to \mathcal{W}$. The procedure is repeated iteratively backward in time starting from $n = N - 1$ until $n = 0$ .

### 2.2.1 Regression Monte Carlo

In this dissertation we focus on simulation-based techniques to solve (2.2). The overall framework is based on solving equation (2.4) through backward induction on $n = N-1, N-2, \ldots$, replacing the true $V_n(\mathbf{x})$ with an estimate $\hat{V}_n(\mathbf{x})$. Since neither the conditional expectation, nor the admissibility constraint are generally available explicitly those terms must also be replaced with their estimated counterparts. As a result, we work with the approximate Dynamic Programming

recursion

$$\begin{aligned} \hat{V}_n(\mathbf{X}_n) &= \inf_{u_n \in \hat{\mathcal{U}}_n(\mathbf{X}_n)} \left\{ \hat{\mathcal{C}}_n(\mathbf{X}_n, u_n) \right\}, \\ \text{where} \quad \hat{\mathcal{C}}_n(\mathbf{X}_n, u_n) &:= \hat{\mathbb{E}} \left[ \pi^{\Delta}(\mathbf{X}_{n:n+1}, u_n) \mathrm{d}s + \hat{V}_{n+1}(\mathbf{X}(t_{n+1})) \middle| \mathbf{X}_n, u_n \right]. \end{aligned} \tag{2.5}$$

Above, $\hat{\mathbb{E}}$ is the approximate projection operator and needs to be estimated along with the set of admissible controls $\hat{\mathcal{U}}_n$. The estimated optimal control $\hat{u}_n \in \hat{\mathcal{U}}_n(\mathbf{X}_n)$ satisfies $\hat{V}_n(\mathbf{X}_n) = \hat{\mathcal{C}}_n(\mathbf{X}_n, \hat{u}_n)$. In problems where $\mathcal{U}_n$ is explicitly defined (e.g. Chapters 3 and 4 in this dissertation), the Regression Monte Carlo (RMC) procedure only requires estimating the continuation value function $\mathcal{C}(\cdot, \cdot)$.

The key idea underlying our algorithms and defining the RMC paradigm is that $\hat{\mathbb{E}}$ and $\hat{\mathcal{U}}$ are implemented through empirical regressions based on Monte Carlo simulations. In other words, we construct *random*, probabilistically defined approximations based on realized paths of $\mathbf{X}$. This philosophy allows to simultaneously handle the numerical integration (against the stochastic shocks in $\mathbf{X}$) and the numerical interpolation (defining $\hat{V}_n(\mathbf{x})$ for arbitrary $\mathbf{x}$) necessary to solve (2.5).

**Sketch of the algorithm:** Without delving into the algorithmic details, let us briefly enumerate the sequence of steps required to solve the dynamic programming equation (2.5) at time step $t_n$ i.e., find $(\hat{\mathcal{C}}_n(\cdot, \cdot), \hat{\mathcal{U}}_n(\cdot))$, assuming we know the pairs $\{\hat{\mathcal{C}}_j(\cdot, \cdot), \hat{\mathcal{U}}_j(\cdot)\}_{j=n+1}^{N-1}$ through backward iteration:

1. The procedure starts by choosing an input dataset $\mathcal{D}_n := \{(\mathbf{x}_n^j, u_n^j)\}_{j=1}^{M_c}$, where $(\mathbf{x}_n^j, u_n^j) \in \mathcal{X} \times \mathcal{W}$.

2. Simulate the path $(\mathbf{x}^j(s))_{t_n \le s \le t_{n+1}}$ starting from $\mathbf{x}_n^j$ and using the control $u_n^j$ for each $j = 1, \ldots, M_c$.

3. Evaluate one-step ahead realization of the value function:

$$\hat{V}_{n+1}(\mathbf{x}_{n+1}^j) = \inf_{u \in \hat{\mathcal{U}}_{n+1}(\mathbf{x}_{n+1}^j)} \left\{ \hat{\mathcal{C}}_{n+1}(\mathbf{x}_{n+1}^j, u) \right\} \text{ for } j = 1, \ldots, M_c.$$

4. Compute the path-wise rewards $y_n^j = \int_{t_n}^{t_{n+1}} \pi_s(\mathbf{x}^j(s), u_n^j)\mathrm{d}s + K(\mathbf{x}_n^j, u_n^j) + \hat{V}_{n+1}(\mathbf{x}_{n+1}^j)$.

5. Regress for estimating the continuation value function $\hat{\mathcal{C}}_n(\cdot,\cdot)$ by projection of $\{y_n^j\}$ on the approximation space $\mathcal{H}_n^c$

$$\hat{\mathcal{C}}_n(\cdot,\cdot) := \arg\min_{f_n^c \in \mathcal{H}_n^c} \sum_{n=1}^{M_c} |f_n^c(\mathbf{x}_n^j, u_n^j) - y_n^j|^2.$$

6. Estimate $\hat{\mathcal{U}}_n(\cdot)$, if not available explicitly. For brevity, we will postpone the discussion on estimation of admissible set $\hat{\mathcal{U}}_n(\cdot)$ until chapter 5.

The sequence of steps (1-6) described above is not the traditional implementation [5, 6] of RMC to solve the dynamic programming equation (2.5). This algorithm (dubbed, *probability constrained Dynamic Emulation Algorithm* (PC-DEA)) is one of the fundamental contributions of this dissertation.

RMC algorithms were popularized by [15, 16, 17] for optimal stopping problems arising in pricing of American options. Optimal stopping is a special case of stochastic control problem (2.2) as far as computational methods are concerned, with $\mathbf{X}(t)$ comprising of only exogenous variables (stock prices) and binary regime $m(t) \in \{\text{continue, stop}\}$. The control $u_t$ determines whether the next regime is to continue or exercise the option (thus stop). In this context, the design points $\mathcal{D}_n$ are chosen only in the space of exogenous variables. The value function $\hat{V}_{n+1}(\mathbf{x})$ at

any location $\mathbf{x}$ is computed by comparing the continuation value function $\hat{\mathcal{C}}_{n+1}(\mathbf{x})$ with the payoff by exercising the option immediately.

RMC was then extended to solve optimal switching problems in the context of scheduling of gas-fired power plant [18]. In comparison to the American option pricing, the number of regimes for optimal switching problems are typically greater than two and the controller continues to switch between different regimes until the time horizon $T$. RMC was further refined to solve optimal control problems arising in the valuation of gas storage [6, 5, 7, 19, 20]. An important difference between scheduling of gas-fired power plant as discussed in [18] and the valuation of gas storage is the presence of an additional endogenous variable (on a continuous domain) whose dynamics is controlled. A typical approach for valuation of gas storage is to first partition the state variables into exogenous and endogenous components. This is followed by simulation of the trajectories of the exogenous variable from time $[0, T]$. Since the dynamics of the endogenous variables is not known a priori, its domain is discretized into finite levels. The control problem is solved for each level of the endogenous variable. Simulation of both endogenous and exogenous state variables was considered in [21] for portfolio optimization problems.

Development of DEA is motivated from the recent work in optimal stopping [8], where the author reformulated optimal stopping as a sequence of statistical learning tasks and proposed several techniques for its efficient implementation. DEA and PC-DEA together extend the framework in [8] from optimal stopping to stochastic optimal control and probability constrained stochastic optimal control. DEA and PC-DEA also provide several advantages over existing methods: (1) They neatly wrap existing approaches in the realm of RMC into a single modular

template with several plug-and-play components. (2) In contrast to the current literature, DEA and PC-DEA can avoid simulating the full trajectory of the state variables and provide significant memory savings especially in high dimensional problems. (3) They can provides additional user flexibility to efficiently choose the type of design and the regression to run. (4) Finally, they can also solve stochastic control problems with noisy constraints which were previously not considered in the RMC literature.

**Approximation errors:** The main numerical challenge for the implementation of the RMC algorithms is that the errors recursively propagate backward in time. At every time step, each approximation in the sequence (1-6) above adds to the error and can affect the final quality of solution $\{\mathcal{C}_j, \mathcal{U}_j\}_{j=0}^{N-1}$. At step 1, the choice of input dataset $\mathcal{D}_n$ can quite literally make or break the algorithm. In Chapter 4 we discuss several choices for $\mathcal{D}_n$ and how they affect the solution. At step 2, simulation of the paths $\mathbf{x}^j(s)$ can add to the bias due to time-discretization. At step 3, the optimization will be non-trivial if both or either the continuation value function $\hat{\mathcal{C}}_n(\cdot,\cdot)$ and the admissible set $\hat{\mathcal{U}}_n(\cdot)$ are non-convex. At step 4, the bias in $y_n^j$ may arise due to poor approximation of $\hat{\mathcal{C}}_{n+1}(\cdot,\cdot)$. Other approximations of the path-wise rewards to reduce bias has been studied in [12, 5, 22]. At step 5, distance between the true $\mathcal{C}_n(\cdot)$ and the closest element in $\mathcal{H}_n^c$ determines the accuracy of $\hat{\mathcal{C}}_n(\cdot)$. Thus choice of $\mathcal{H}_n^c$ strongly affects the quality of the solution. In Chapter 4 we propose Gaussian process regression for non-parametric approximation of the continuation value and compare its performance with alternates in the literature. Finally at step 6, incorrect approximation of the admissible set can lead to making inadmissible decisions and thus violating the constraints of the problem. In Chapter 5, we provide detailed discussion on admissible sets de-

fined through local probabilistic constraints on the system state for the microgrid control.

## 2.3 Other Methods

There are several other approaches that have been discussed in the literature to solve stochastic optimal control problems.

**Markov Decision Process:** This is a classic method to solve SOC problems assuming finite state space and actions; textbooks on the subject include [23, 24, 25]. The dynamics of the state process is thus defined through a Markov chain. Since the state space in SOC problems is often continuous, discretization (to achieve finite states) is a key part of the problem. However as the dimension of state variables increase, discretization makes this approach extremely prohibitive. A recent work utilizing MDPs in the context of gas storage valuation is [26].

**Hamilton Jacobi Bellman Equation (HJB):** Value function $V$ of a stochastic optimal control problem for Markov processes in continuous time is a solution of a partial differential equation (PDE), called HJB equation [27, 28]. HJB represents the local behavior of $V$ and can be considered as an infinitesimal version (as $|t_{k+1} - t_k| \to 0$) of the dynamic programming equation (2.4). However, as with any PDE approach they lack scalability, thus become prohibitive when the state space increases beyond three or four dimensions. Application of HJB equation to solve SOC problems arising in gas storage valuation and microgrid management are [29, 30].

**Stochastic Dual Dynamic Programming (SDDP):** High dimensional stochastic control problems with linear constraints and linear system dynamics

can be solved very efficiently using SDDP. The efficient implementation is due to assumptions of linearity, resulting in value function which is also a supremum over affine hyperplanes. It was devised to find operational policies for large number of interconnected hydroelectric systems [31, 32], but has also found its way into optimal control of microgrids [33]. A recent work using piecewise linear regressions for estimating the conditional expectations in SDDP was proposed in [34] and available in the library [35].

**Reinforcement learning:** Stochastic optimal control with no assumptions on the dynamics of the controlled state process $\mathbf{X}^u(t)$ and random rewards $\pi_s(\mathbf{X}^u(s), u(s))$ is considered within reinforcement learning literature. A textbook level discussion is given in [36] and its application in the context of energy management is presented in [37].

## 2.4 Additional Applications

The DEA and PC-DEA are applicable in numerous other contexts; in this section we mention some additional examples.

**Generalized Microgrid.** The microgrid example described above is highly simplified. More realistic models would consider separate stochastic factors for the different components (e.g. renewable output ($R_t$) and demand load ($D_t$)). Moreover, the controller can typically control both the diesel generator, as well as the battery, adding further states. In a long-term planning context, battery degradation due to repeated charge/discharge becomes important and needs to be captured by an additional "age" variable. Thus, an industrial-grade implementation of microgrid management would call for a high-dimensional formulation

$d \gg 4$, where DEA aspects, such as choice of design $\mathcal{D}$, become critical for performance.

**Hydropower optimization.** Control of a hydropower dam [38, 39, 40, 34] is a classic problem in energy systems and is often formulated as stochastic optimal control. Within this setup, the controller observes random inflows from precipitation, as well as fluctuating electricity prices. Her objective is to control the downstream outflow from the dam to maximize profit from power sales. RMC has been used for this problem in [20]. Additional probabilistic constraints, such as minimum (daily) dam capacity [38], will require estimating the admissible set of controls akin to statistical estimation of $\mathcal{U}$ in PC-DEA.

**Robot/Drone Motion Control.** Dynamic path optimization is a classical engineering problem that is frequently formulated as a stochastic control. The robot motion is subject to random shocks (e.g. wind disturbance for an unmanned aerial vehicle) and the control is velocity and/or acceleration. Objectives might include target-tracking, obstacle avoidance, fuel use optimization, etc. The robot location/speed are the endogenous state components, while the external shocks (wind speed, obstacle motion) are the stochastic factors. See e.g. [22, 41] for approaches where a modification of DEA could be beneficial. Constraints on the path of the robot, such as avoiding collision with objects that obstruct its path [42], will require computing admissible sets; thus PC-DEA will be useful.

**Portfolio Optimization.** Another application of DEA could be for portfolio optimization, with inventory corresponding to the current wealth that is driven by the investment strategy $u_t$. The objective is to maximize expected utility of terminal wealth, subject to various market and investment constraints. The typical state is then the asset price $P_t$ and the current wealth $I_t$. Zhang et al. [43]

recently used RMC method for this problem. PC-DEA will be useful to handle additional constraints, such as bounded value-at-risk.

# Chapter 3

# Regression Monte Carlo for Microgrid Management

*This chapter is the result of a collaboration with Clemence Alasseur, Alessandro Balata, Sahar Ben Aziza, Peter Tankov and Xavier Warin. It is based on the work [11].*

In this chapter we study an islanded microgrid system designed to supply a small village with the power produced by photovoltaic panels, wind turbines and a diesel generator. A battery storage system device is used to shift power from times of high renewable production to times of high demand. We build on the mathematical model introduced in [30] and optimize the diesel consumption under a no-blackout constraint. We introduce a methodology to solve microgrid management problem using different variants of RMC algorithms and use numerical simulations to infer results about the optimal design of the grid.

## 3.1 Introduction

In this chapter, we consider a traditional microgrid (cf. Figure 1.1a) serving a small group of customers in islanded mode, meaning that the network is not connected to the main national grid. The system consists of an intermittent renewable generator unit, a conventional dispatchable generator, and a battery storage system. Both the load and the intermittent renewable production are stochastic, and we use a stochastic differential equation (SDE) to model directly the net demand, that is, the difference between the load and the renewable production. We then set up a stochastic optimal control problem, whose goal is to minimize the cost of using the diesel generator plus the cost of curtailing renewable energy in case of excess production, subject to the constraint of ensuring reliable energy supply. We use RMC to solve this stochastic control problem numerically. Two variants of the regression algorithm, called Regress Now and Regress Later are proposed and compared in this chapter. The numerical examples illustrate the performance of the optimal policies, provide insights on the optimal sizing of the battery, and compare the policies obtained by stochastic optimization to the industry standard, which uses deterministic policies.

The optimization problem arising from the search for a cost-effective control strategy has been extensively studied. Three recent survey papers [44, 45, 46] summarize different methods used for optimal usage, expansion and voltage control for the microgrids. Authors in [4, 30] transform the optimization problem associated with the microgrid management into an optimal control framework and solve it using the corresponding Hamilton Jacobi Bellman equation. Besides proposing an optimal strategy, the authors also compare the solution of the deter-

ministic and stochastic representation of the problem. However, similarly to most PDE methods, this approach suffers from the curse of dimensionality and as a result, it is difficult to scale. The main contribution of this chapter is to solve the microgrid control problem using Regression Monte Carlo algorithms. In contrast to existing approaches, the method used in this chapter is more easily scalable and works well in moderately large dimensions [9]. It is fair to mention here that the problem we study in the following is however low dimensional as it displays one source of randomness and one degenerate controlled process.

Identifying the optimal mix, the size and the placement of different components in the microgrid is an important challenge to its large scale use. The papers [47, 10] use mixed-integer linear programming to address the design problem and test their model on a real data set from a microgrid in Alaska. In a similar work, [48] studied the economically optimal mix of PV, wind, batteries and diesel for rural areas in Nigeria. In [49], optimal battery storage sizing is deduced from the autocorrelation structure of renewable production forecast errors. In this chapter, we propose an alternative approach for the optimal sizing of the battery energy storage system, assuming stochastic load dynamics and fixed lifetime of the battery. Our in-depth analysis of the system behavior leads to practical guidelines for the design and control of islanded microgrids.

Finally, several authors [50, 51, 52] used stochastic control techniques to determine optimal operation strategies for wind production – storage systems with access to energy markets. In contract to these papers, in this chapter, energy prices appear only as constant penalty factors in the cost functional, and the main focus is on the stable operation of the microgrid without blackouts.

The rest of the chapter is organized as follows: In Section 3.2 we describe the

microgrid model and introduce the different components of the system, in Section 3.3 we translate the problem of managing the microgrid in a stochastic control problem and present the dynamic programming equation that we intend to solve numerically. Section 3.4 introduces the numerical algorithms used to solve the control problem, we give a general framework for solving the dynamic programming equation and we then provide three algorithms for the approximation of conditional expectations. In Section 3.5 we illustrate the results of the numerical experiments, identify the best algorithm among those we studied and then employ it to analyze the system behavior.

## 3.2 Microgrid

In this section, we will formalize the dynamics of different components of the microgrid (Figure 1.1a) from Chapter 1. The microgrid serves a small, isolated village; most of the power to the village is supplied by generating units whose output has zero marginal cost, is intermittent and uncontrolled. Additional power is supplied by a controlled generator whose operations come alongside a cost for the microgrid owner (either the community itself or a power utility). Often the intermittent units include PV panels and wind turbines, while the controlled unit is often a diesel generator. The battery energy storage system in Figure 1.1a allows for inter-temporal transfer of energy from times when demand is low, to times when it is higher, but also introduces an element of strategic behavior that can be employed by the system controller, to minimize the operational costs. Without an energy storage, diesel had to be run at all times demand exceeded production. When a battery is installed, intensity and timing of output from the

diesel generator can be adjusted to move the level of charge of the battery towards the most cost effective levels.

**Remark 1** *Note that for convenience, in the following, we will work in discrete time only and divide the time interval* $[0, T]$ *into discrete time points* $\{t_0, t_1, \ldots, t_N\}$. *For simplicity of notation, we continue to denote any stochastic process* $\mathbf{X}(t_n)$ *at time* $t_n$ *via* $\mathbf{X}_n$, *thus* $\mathbf{X}(t_n) \equiv \mathbf{X}_n$. *We also consider a finite optimization horizon represented by the number of periods over which we want to optimize the system operations indicated by* $T \equiv t_N$.

### 3.2.1 Net Demand

Consider two stochastic processes $D_n$ and $R_n$, the former represents the demand/load and the latter the production through the renewable generators. Notice that both processes are uncontrolled and they represent, respectively, the unconditional withdrawal and injection of power in the system (constant during time step). For the purpose of managing the microgrid, the controller is interested only in the net effect of the two processes denoted by the process $L_n$:

$$L_n = D_n - R_n \; ; \quad n \in \{0, 1, \ldots, N\}. \tag{3.1}$$

**Remark 2** *The state variable* $L_n$ *represents the net demand of power at each time* $t_n$, *such that for* $L_n > 0$, *we should provide power through the battery or diesel generator and for* $L_n < 0$ *we can store the extra power in the battery.*

For simplicity, we model the net demand as an AR(1) process, the discrete equivalent of an OrnsteinUhlenbeck process. In practical applications we expect

$L_n$ to be an $\mathbb{R}$-valued mean reverting process with many different sources of noise and time dependent random parameters; our choice of using an AR(1) avoids the cumbersome notation coming from multiple noise sources still providing scope for generalization. The process $L_n$ is driven by the following difference equation, starting from an initial point $L_0$:

$$L_{n+1} = L_n + b(\Lambda_n - L_n)\Delta t + \sigma\sqrt{\Delta t_n}\,\xi_n\,; \quad n \in \{0, 1, \ldots, N\} \tag{3.2}$$

where $\xi_n \sim \mathcal{N}(0,1)$, $\Delta t$ is the amount of time before new information is acquired, $b$ is the mean reversion speed, $\sigma$ the volatility of the process and $\Lambda_n$ is the mean reversion level (typically deterministic function of time).

**Remark 3** *In real applications the deterministic function $\Lambda_n$ should represent the best forecast available for future net demand at the time of the estimation of the policy.*

### 3.2.2 Diesel generator

The Diesel generator represents the controlled dispatchable unit. The state of the generator is represented by $m_n = \{0, 1\}$. If $m_n = 0$ then the diesel generator is OFF, while it is ON when $m_n = 1$. When the engine is ON, it produces a power output denoted by $u_n \in [\underline{u}, \overline{u}]$ at time $t_n$, for $\underline{u} > 0$.

Notice that, in addition, when the engine is turned ON, an extra amount of fuel is burned in order for the generator to warm up and reach working regime. We model the cost of switching the diesel generator from state $i$ to $j$ via $K(i, j)$. Thus, the controller pays $K(0, 1)$ every time the generator is switched on $m_{n+1} = 1$ from $m_n = 0$. We assume the cost to switch off the generator (when it is running)

is zero i.e. $K(1,0) = 0$ and continuing in the same regime incurs no cost i.e. $K(1,1) = K(0,0) = 0$. The fuel consumption of the diesel generator is modeled by an increasing function $\rho(u_n)$ which maps the power $u_n$ produced during one time step into the quantity of diesel necessary for such output. Denoting by $P_n$ the price of fuel at time $t_n$, the cost of producing $u_n$ kW of power at one time step is $P_n\rho(u_n)$; for simplicity we take a constant price of the fuel $P_n = p$.

### 3.2.3 Dynamics of the Battery

The storage device is directly connected to the microgrid and therefore its output is equal to the imbalance between net demand $L_n$ and diesel generator output $u_n$, when this is allowed by the physical constraint. The battery therefore is discharged in case of insufficiency of the diesel output and charged when the diesel generator and renewables provide a surplus of power.

Let us denote the state of charge of the battery at time $t_n$ as $I_n$ and its maximum capacity as $I_{max}$. If the power rating of the battery is given by $B^{\max}$ and $B^{\min}$, where $B^{\max}$ and $B^{\min}$ represent respectively the maximum output and input with $B^{\min} < 0 < B^{\max}$, its power output $B_n$ at time $t_n$ is defined as:

$$B_n = \frac{I_n - I_{\max}}{\Delta t_n} \vee \left(B^{\min} \vee (L_n - u_n) \wedge B^{\max}\right) \wedge \frac{I_n}{\Delta t_n}. \tag{3.3}$$

Intuitively, $B_n < 0$ refers to the charging of the battery and $B_n > 0$ refers to the supply of power from the battery. The inner terms in equation (3.3) capture the constraints due to maximum power output/input to the battery and the outer terms capture the effect of the capacity constraints on the power output of the battery. Notice then that an energy storage has a limited amount of capacity after

which it can not be charged further, as well as an "empty" level below which no more power can be provided from the battery. The dynamics of the controlled process $I_n$ is described by the following equation:

$$I_{n+1} = I_n - B_n \Delta t_n, \quad n \in \{0, 1, \ldots, N-1\}, \tag{3.4}$$

here $I_n \in [0, I_{max}]$ and $B_n \in [B^{\min}, B^{\max}]$. For simplicity we assume that the battery is 100% efficient. Notice that the battery output $B$ and state of charge in the battery $I$ depend on the controlled diesel output $u_n$.

Intuition tells us that the bigger the battery, the less diesel will be needed to run the operations of the microgrid. This is true because a bigger battery would allow to store for later use a bigger proportion of the excess power produced by the renewables. Batteries however are very expensive, and the cost per kWh of capacity scales almost linearly for the kind of devices we consider in this chapter (parallel connection of smaller batteries), hence it is important to find the optimal size of battery for the needs of each specific microgrid.

### 3.2.4 Management of the Microgrid

The purpose of the microgrid is to provide a cheap and reliable source of power supply to at least match the demand. Therefore, we search for a control policy for the diesel generator which minimizes the operating cost and produces enough electricity to match the net demand. In order to assess how well we are doing in supplying electricity, we introduce the controlled imbalance process $S_n$ defined as follows:

$$S_n = L_n - B_n - u_n \quad n \in \{0, 1, \ldots, N\}. \tag{3.5}$$

Ideally, the owner of the Microgrid would like to have $S_n = 0 \quad \forall\, n$. This situation represents the perfect balance of demand and generation. When $S_n > 0$ we observe a *blackout*, net demand is greater than the production meaning that some loads are automatically disconnected from the system. The situation $S_n < 0$ is defined as a curtailment of renewable resources and takes place when we have a surplus of electricity.

We treat the two scenarios, blackout and curtailment asymmetrically. To ensure no-blackout $S_n \leq 0$ and regular supply of power, we impose a constraint on the set of admissible controls:

$$\begin{aligned} S_n &\leq 0 \\ \text{i.e. } u_n &\geq L_n - B_n. \end{aligned} \tag{3.6}$$

However, for $S_n < 0$ i.e. surplus of electricity, we penalize the microgrid using a proportional cost denoted by $C_1$. Large penalty would lead to low level of curtailment and can be thought of as a parameter in the subsequent optimization problem.

A rigorous mathematical description of the microgrid management problem follows in section 3.3.

## 3.3 Stochastic control formulation

We state now the stochastic control problem for the diesel generator operating in a microgrid system as described in section 3.2. In practice we seek a control that minimizes the cost of diesel usage $p\rho(u)$, the switching cost $K(0,1)$ and the curtailment cost $C_1|S_n|\mathbb{1}_{\{S_n<0\}}$, under the no black-out constraint $S_n \leq 0$.

Note that, given the type of control we have on the diesel generator, we can frame the optimization problem as a special case of stochastic control problems known as optimal switching problems.

Let us denote by $(\mathcal{F}_n)_{n\geq 0}$ the filtration generated by the random variables $\{\xi_n\}_{n\geq 0}$, which represent the only randomness in the system, that is, we define $\mathcal{F}_n = \sigma(\xi_i, i < n)$ for $n \geq 1$, and $\mathcal{F}_0$ to be the trivial $\sigma$-field. We require the control process $(u_n)_{n\geq 0}$ to be adapted to this filtration or, in other words, no future information should be used to determine its value. Under this assumption, the net demand process $(L_s)_{s=0}^n$, the state of charge process $(I_s)_{s=0}^n$ and the current regime $m_n$, become adapted to $(\mathcal{F}_n)_{n\geq 0}$. The objective of the controller is to minimize the following cost functional

$$\mathbb{E}\left[\sum_{s=0}^{N-1} \mathbb{1}_{\{m_{s+1}-m_s=1\}} K(0,1) + p\rho(u_s) + C_1|S_s|\mathbb{1}_{\{S_s<0\}} + W(I_N)\right],$$

where $W$ is a terminal condition which might be linked with situations where the battery has been rented and has to be returned with the same level of charge otherwise a penalty might be applied. The minimization is carried out over the set of admissible strategies $\mathcal{U}$, containing all $(\mathcal{F}_n)_{n\geq 0}$-adapted controls $(u_n)_{n\geq 0}$ such that

$$u_n \geq L_n - B_n \quad \forall n \tag{3.7}$$

$$u_n \in [\underline{u}, \overline{u}] \cup \{0\}. \tag{3.8}$$

$$B_n = \frac{I_n - I_{\max}}{\Delta t_n} \vee \left(B^{\min} \vee (L_n - u_n) \wedge B^{\max}\right) \wedge \frac{I_n}{\Delta t_n} \tag{3.9}$$

where (3.7) represents the no-blackout constraints translated for the power pro-

duced by the diesel generator, (3.8) represents the minimum and maximum power output of the generator and (3.9) models the physical constraints of the battery: maximum input/output power and maximum capacity.

Since the state dynamics is Markovian, the optimal control is of feedback type and can be computed using the dynamic programming approach (see [53, Chapter 8]). To formulate this approach, we define the pathwise value $\mathcal{J}_n$ starting from time $t_n$, given by

$$\mathcal{J}_n = \sum_{s=n}^{N-1} \mathbb{1}_{\{m_{s+1}-m_s=1\}} K(0,1) + p\rho(u_s) + C_1|S_s|\mathbb{1}_{\{S_s<0\}} + W(I_N). \tag{3.10}$$

The value function is then defined as follows.

$$V_n(L,I,m) = \min_{u\in\mathcal{U}_n} \left\{ \mathbb{E}\left[ \mathcal{J}_n \middle| L_n = L, I_n = I, m_n = m \right] \right\}, \tag{3.11}$$

where the class $\mathcal{U}_n$ contains admissible controls "starting from time $t_n$": processes $(u_s)_{s=n}^{N-1}$ adapted to the filtration $\mathcal{F}_s^n := \sigma(\xi_u, n \le u < s$ and satisfying the constraints (3.7), (3.8) and (3.9) between $n$ and $N-1$.

The dynamic programming principle associated to (3.11), decomposes the problem on a single interval into two optimal control problems: an optimal switching problem between being in the regime ON or OFF, and another absolutely continuous control problem assuming the regime is ON. The equation reads as

follows:

$$V_n(L,I,m) = \min_u \Big( \pi^{\Delta}(L,I,m,u) + \mathcal{C}_n(L,I,m;u) \Big), \tag{3.12}$$

$$\text{subject to} \quad u \geq L - B, \quad u \in \{0\} \cup [\underline{u}, \overline{u}], \tag{3.13}$$

$$\text{where} \quad \pi^{\Delta}(L,I,m,u) = \mathbb{1}_{\{\mathbf{1}_{u\neq 0} - m = 1\}} K(0,1) + p\rho(u) + C_1 |S| \mathbb{1}_{\{S<0\}}, \tag{3.14}$$

$$B = \frac{I - I_{\max}}{\Delta t_n} \vee \big( B^{\min} \vee (L-u) \wedge B^{\max} \big) \wedge \frac{I}{\Delta t_n} \quad , \quad S = L - B - u, \tag{3.15}$$

$$\text{and} \quad \mathcal{C}_n(L,I,m;u) = \mathbb{E}[V_{n+1}(L_{n+1}, I_{n+1}, \mathbf{1}_{u\neq 0}) | L_n = L, I_n = I, m_n = m]. \tag{3.16}$$

In order to ensure that the set of admissible controls is nonempty we introduce the following assumption:

**Assumption 1** *The diesel generator is powerful enough to supply demand at all times, i.e there is always a control $u$ that satisfies the blackout constraint.*

**Remark 4** *We enforce assumption 1 by redefining the net demand process with a truncated version of* (3.1)*, such that $\tilde{L}_n = \min(L_n, L_{\max})$ is the net demand. In practice this is reasonable because the maximum power that could be required from the microgrid is known a priori and the diesel generator is generally sized to the maximum capacity installed on the system. For the sake of notational simplicity, we will drop the $\sim$ on the variable $\tilde{L}_n$ from the following sections.*

Note that (3.12) provides a direct technique to solve problem (3.11), iterating backward in time from a known terminal condition and solving a static, one period, optimization problem at each time step. The only difficulty in this procedure lies in the estimation of conditional expectations of future value function, which can not be computed exactly. In the next section 3.4 we will focus on the numerical solution of (3.11).

## 3.4 Numerical Resolution

In this section we describe the algorithm which we want to employ in the solution of the energy management problem for the Microgrid system described in Section 3.3. The main mathematical difficulty comes from the approximation of conditional expectations in (3.12), which we will tackle using a family of methods called Regression Monte Carlo. For our purposes we assume that the one step optimization problem can be solved either by extensive search, or by any more efficient method preferred by the reader. Here we discretize the set of possible controls into a finite collection, as a result the optimization is straightforward.

In the dynamic programming equation (3.12), the conditional expectation is not available analytically and needs to be estimated. As a result, equation (3.12) is replaced by the corresponding approximate dynamic programming equation (3.17) and our algorithm fully exploits this formulation. Similar to Section 2.2.1, $\hat{\mathbb{E}}$ represents the approximate projection operator. We start by generating a set of simulations (scenarios) of the process $L$, which we will refer to as training points, then we optimize our policy so that it performs well, on average (weighted on the probability of each scenario), on the different scenarios.

$$\hat{V}_n(L, I, m) = \min_u \Big( \pi^{\Delta}(L, I, m, u) + \hat{\mathcal{C}}_n(L, I, m; u) \Big), \tag{3.17}$$

$$\text{where} \quad \hat{\mathcal{C}}_n(L, I, m; u) = \hat{\mathbb{E}}[\hat{V}_{n+1}(L_{n+1}, I_{n+1}, \mathbf{1}_{u \neq 0}) | L_n = L, I_n = I, m_n = m]. \tag{3.18}$$

In practice, we initialize the value function at last time step in the backward procedure to be equal to the terminal condition $W$. We then iterate backward in time and at each time step over each training point we choose the control that minimizes the sum of one step cost function and the estimated conditional expectation of the future costs $\hat{\mathcal{C}}_n(L, I, m; u)$. Note that, as expected, the conditional expectation is a function of time,

the state of the system $(L, I)$ and the state of the diesel generator, represented by the ON/OFF switch $m$ and the control $u$.

As the iteration reaches the initial time point we collect a set of optimal actions for each time step and many different scenarios; in addition, since the problem is Markovian, we can summarize such strategies in the form of control maps: best action at each time $t_n$ given a pair of state variables $(L_n, I_n)$ and state of the diesel generator $m_n$. We propose three different techniques to compute $\hat{\mathcal{C}}$ in Section 3.4.1.

A fair assessment of the quality of the control policies approximated by the algorithm just introduced is obtained by running a number of forward Monte Carlo simulations of the net demand, controlling the system using such policies and then taking the average performance.

We give a general description of the pseudo code in algorithms 1 and 4.

**Remark 5** *Notice that it is typical of Regression Monte Carlo algorithms to provide the optimal policy only implicitly, in the form of minimizer of an explicit parameterized function. The outputs of the algorithm are therefore the parameters (regression coefficients) of such function.*

### 3.4.1 Regression for continuation value

In this section we present the numerical techniques we use to estimate conditional expectations $\mathcal{C}_n(L, I, m; u)$ in algorithm 1. These techniques belong to the realm of Regression Monte Carlo methods, and in particular these specifications allow to deal with degenerate controlled processes (the inventory). We focus on two main variants: Regress Now (RN) and Regress Later (RL). Regress Now require projection of the value function at $n + 1$ on $\mathcal{F}_n$ measurable basis functions, Regress Later requires an $\mathcal{F}_{n+1}$ projection. Within Regress Now there are two techniques, piecewise continuous approximation and global polynomial approximation, differing in how the regression is performed to esti-

mate the conditional expectation. Piecewise continuous approximation is characterized by a one dimensional projection in the net demand dimension repeated at different inventory points. The approximation is extended to the full domain using linear interpolation in the inventory dimension. Global polynomial approximation, on the other hand, use a two dimensional regression in net demand and inventory. Since we will use polynomial basis for regressions, we will refer piecewise continuous approximation and global polynomial approximation as PR-1D and PR-2D respectively. In total, we will compare the performance of three methods i.e. PR-1D, PR-2D and RL. For details on these techniques, see [20] for Regress Later, [6, 7] for PR-1D and [5] for PR-2D. Note that in the three methods we repeat the regression approximation for both values of $m$. An open source platform has also been developed to numerically solve wide variety of stochastic optimization problems in [35].

Let us denote by $\{L_n^j\}_{j=1}^M$ the collection of training points at time $t_n$, similar notation is used for the inventory $\{I_n^j\}_{j=1}^M$.

#### 3.4.1.1 Regress Now

**Piecewise Continuous Approximation (PR-1D)**: This is characterized by a one dimensional approximation of the conditional expectation repeated at different levels of inventory. Let $\Upsilon_I = \{I^0 = 0, \ldots, I^{M_I} = I_{\max}\}$ be a discretisation of the state space of the inventory and $\{L_n^j\}_{j=1,n=1}^{M,N}$ be generated from a forward simulation of the dynamics of $L$. We define the approximation of the continuation value on the grid $\Upsilon_I$ by regressing the set of value functions $\{V_{n+1}(L_{n+1}^j, I^i)\}_{j=1}^M$ over the basis functions $\{\phi_k(L)\}_{k=1}^K$ for each $\{I^i\}_{i=0}^{M_I}$, obtaining:

$$\tilde{\mathcal{C}}_n(L, I^i, m) = \sum_{k=1}^{K} \alpha_n^{k,i,m} \phi_k(L)\,, \quad i = 0,\, 1,\, \ldots,\, M_I,$$

where we compute a collection of regression coefficients through least square minimization

$$\boldsymbol{\alpha}_n^{i,m} = \arg\min_{a\in\mathbb{R}^K} \Big\{ \frac{1}{M} \sum_{j=1}^{M} \big(V_{n+1}(L_{n+1}^j, I^i, m) - \sum_{k=1}^{K} a^k \phi(L_n^j)\big)^2 \Big\},$$

where we define $\mathbb{R}^K \ni \boldsymbol{\alpha}_n^{i,m} = (\alpha_n^{1,i,m}, \ldots, \alpha_n^{K,i,m})$.

Note that the least square projection is a sample estimation of the $L^2$ projection induced by the conditional expectation, for this reason we can approximate the function $\mathcal{C}_n(\cdot)$ using a least square projection of the value function at time $t_{n+1}$. However, as we have not included the inventory in the basis functions, we need to interpolate between values of $\tilde{\mathcal{C}}_n(L, I^i; m)$ in order to obtain an estimation of the value function for $I_{n+1} \in (I^i, I^{i+1})$. Let us define $\hat{\mathcal{C}}_n(L, I, m; u)$ by the linear interpolation

$$\hat{\mathcal{C}}_n(L, I, m; u) = \omega_n(I, u)\tilde{\mathcal{C}}_n(L, I^i, m) + \big(1-\omega_n(I, u)\big)\tilde{\mathcal{C}}_n(L, I^{i+1}, m),\ I - B_n\Delta t_n \in [I^i, I^{i+1}),$$

where $\omega_n(I, u) = \frac{I^{i+1} - I + B_n\Delta t_n}{I^{i+1} - I^i}$ and $i = 0, \ldots, M_I$.
Details of the algorithms are given in the pseudocode 2.

**Global polynomial approximations (PR-2D)** Contrary to PR-1D, here we approximate the conditional expectation jointly as a function of both net demand $L$ and inventory $I$, without the need for interpolation. We generate training points $\{L_n^j\}_{j=1,n=1}^{M,N}$ from a forward simulation of the dynamics of $L$ and $\{I_n^j\}_{j=1,n=1}^{M,N}$ from a distribution $\mu_N$ on $[0, I_{max}]$ independently. We choose $\mu_N$ to be the Lebesgue measure on $[0, I_{\max}]$.

The regression coefficients for the global polynomial approximation are computed by least-square projection as:

$$\boldsymbol{\alpha}_n^m = \arg\min_{a\in\mathbb{R}^K} \Big\{ \mathbb{E}\Big[ \big(\hat{V}_{n+1}(L_{n+1}^j, I_{n+1}^j, m) - \sum_{k=1}^{K} a_k \phi(L_n^j, I_{n+1}^j)\big)^2 \Big] \Big\},$$

where we define $\mathbb{R}^K \ni \boldsymbol{\alpha}_n^m = (\alpha_n^{1,m}, \ldots, \alpha_n^{K,m})$. The conditional expectation is then

estimated as:

$$\hat{\mathcal{C}}_n(L, I, m; u) = \mathbb{E}\Big[\sum_{k=1}^{K} \alpha_n^{k,m} \phi_k(L_n, I_{n+1}) \Big| L_n = L,\, I_n = I,\, u_n = u\Big]$$
$$= \sum_{k=1}^{K} \alpha_n^{k,m} \phi_k(L, I - B_n \Delta t_n).$$

**Remark 6** *Although in this chapter we chose $\mu_N$ to be independent of the dynamics of $(L_n)_{n\geq 0}$, in Chapter 4 we discuss in detail the effect of the correlation between $\{L_n^j\}_{j=1}^{M_c}$ and $\{I_{n+1}^j\}_{j=1}^{M_c}$ on the performance of Regress Now algorithm.*

#### 3.4.1.2 Regress Later

In this framework, the value function $\hat{V}_{n+1}(\cdot, \cdot, m)$ at time $t_{n+1}$ is parameterized via regression and then the conditional expectation $\mathcal{C}_n(\cdot, \cdot, \cdot)$ is evaluated analytically. We start by generating the samples $\{L_{n+1}^j, I_{n+1}^j\}_{j=1}^{M}$ from an appropriate distribution $\mu_L$, we chose $\mu_L$ to be Lesbegue measure on $[0, I_{\max}] \times [-L_{\max}, L_{\max}]$. The value function is then approximated via least square projection as:

$$\boldsymbol{\alpha}_n^m = \arg\min_{a \in \mathbb{R}^K} \Big\{\mathbb{E}\Big[\big(\hat{V}_{n+1}(L_{n+1}^j, I_{n+1}^j, m) - \sum_{k=1}^{K} a_k \phi(L_{n+1}^j, I_{n+1}^j)\big)^2\Big]\Big\},$$

where we define $\mathbb{R}^K \ni \boldsymbol{\alpha}_n^m = (\alpha_n^{1,m}, \ldots, \alpha_n^{K,m})$. Let us recall, denoting by $\boldsymbol{\phi}$ the vector $\big(\phi_1(\cdot), \ldots, \phi_K(\cdot)\big)$, that the coefficients $\boldsymbol{\alpha}_n^m$ can be computed explicitly by

$$\boldsymbol{\alpha}_n^m = \Big(\mathbb{E}_\mu\big[\boldsymbol{\phi}\boldsymbol{\phi}^T\big]\Big)^{-1} \mathbb{E}_\mu\Big[\hat{V}_{n+1}(L_{n+1}, I_{n+1}, m)\boldsymbol{\phi}\Big]^T$$
$$\approx \Big(\sum_{j=1}^{M} \boldsymbol{\phi}\boldsymbol{\phi}^T\Big)^{-1} \sum_{j=1}^{M} \hat{V}_{n+1}(L_{n+1}^j, I_{n+1}^j, m)\boldsymbol{\phi}^T$$

and therefore, even though the regression coefficients are random (sample average approximation of expectations with respect to the measure $\mu$) they are independent of $\mathcal{F}_n$.

Given the previous remark we can estimate the conditional expectation of future value through:

$$\begin{aligned}\hat{\mathcal{C}}_n(L,I;m,u) &= \mathbb{E}\Big[\sum_{k=1}^{K}\alpha_n^{k,m}\phi_k(L_{n+1},I_{n+1})\Big|L_n=L,\,I_n=I,\,u_n=u\Big]\\ &= \sum_{k=1}^{K}\alpha_n^{k,m}\mathbb{E}\Big[\phi_k(L_{n+1},I_{n+1})\Big|L_n=L,\,I_n=I,\,u_n=u\Big].\end{aligned}$$

Now, we need to compute the expectation with respect to the randomness contained in the transition function from $L_n$ to $L_{n+1}$ and we simply write

$$\mathbb{E}\Big[\phi_k(L_{n+1},I_{n+1})\Big|\mathcal{F}_n\Big] = \mathbb{E}_\xi\Big[\phi_k(L+b(\Lambda_n-L)\Delta t_n+\sigma\sqrt{\Delta t_n}\xi, I-B_n\Delta t_n)\Big] =: \hat{\phi}_k(L,I,u).$$

For polynomial basis functions, i.e. $\phi_k(L_{n+1},I_{n+1}) := L_{n+1}^p I_{n+1}^q$, the conditional expectation $\hat{\phi}_k(L,I,u)$ can be written in closed form as:

$$\begin{aligned}\hat{\phi}_k(L,I,u) &= \mathbb{E}\big[L_{n+1}^p I_{n+1}^q\big|L_n=L,I_n=I,u_n=u\big]\\ &= I_{n+1}^q\sigma^p dt^{\frac{p}{2}}\sum_{k=0}^{p}\mathbb{I}_{\{(p-k)\text{ is odd}\}}\binom{p}{k}\Big(L\frac{1-\lambda dt}{\sigma\sqrt{dt}}\Big)^k\prod_{j=1}^{\frac{p-k}{2}}(2j-1)\end{aligned}$$

**Remark 7** *Notice that RL does not require us to simulate the path of the net demand $(L_n)_{n\geq 0}$ process, rather it uses the transition probability $L_{n+1}|L_n$ to estimate the conditional expectation.*

**Out-of-sample evaluation** To compare different algorithms, we compute the out-of-sample estimate of the value function at $t_0 = 0$ and state $(L, I, m)$. We start by simulating fixed $M'$ paths of the $\{L_n^j\}_{j,n=1}^{M',N}$ starting from $L_0^j = L$. We then iteratively update the trajectory of the controlled process $I_n^j$ and state of the diesel generator $m_n^j$ starting from $I_0 = I$ and $m_0 = m$. Assuming we know the state of the system at time $t_n$ as $(L_n^j, I_n^j, m_n^j)$, for each path $j = 1, \ldots, M'$ and at any time step $t_n$, we compute the

**Algorithm 1:** Regression Monte Carlo algorithm for Microgrid management

**input:** number of basis $K$, number of training points $M$, discretisation of the inventory $D$, time-steps $N$.
**if** Regress Later **then**
  Generate $\{X_n^j, I_n^j\}_{j,n=1}^{M,N}$ accordingly to a distribution $\mu$;
**else if** Regress Now **then**
  **if** PR $-$ 2D **then**
    Simulate $\{X_n^j\}_{j,n=1}^{M,N}$ according to its dynamics.
    Generate $\{I_{n+1}^j\}_{j,n=0}^{M,N-1}$ according to a distribution $\mu$;
  **else if** PR $-$ 1D **then**
    Generate a customary grid $\{I^0, \dots, I^{M_I}\}$ over the domain $[0, I_{\max}]$.
    Simulate $\{L_n^j\}_{j,n=1}^{M_L,N}$ according to its dynamics where $M_L = M/(M_I + 1)$;
    Define $\{L_n^j, I_{n+1}^j\}_{j=1}^{M}$ as cross product of $\{L_n^j\}_{j=1}^{M_L}$ and $\{I^j\}_{j=0}^{M_I}$ for $\forall n$.
Initialize the value function
$V_N(X_N^j, I_N^j, 1) = V_N(X_N^j, I_N^j, 0) = W(I_N^j), \quad \forall j = 1, \dots, M;$
**for** $n = N-1$ *to* $1$ **do**
  Compute $\hat{\mathcal{C}}_n$ using Algorithms 2 or 3.
  **for** $m = 0$ *to* $1$ **do**
    $\hat{V}_n(L_n^j, I_n^j, m) = \min_{u \in \mathcal{U}_n} \Big( \pi^{\Delta}(L, I, m, u) + \hat{\mathcal{C}}_n(L, I, m; u) \Big), \quad j = 1, \dots, M$
  **end**
**end**
**output:** continuation value function $\{\hat{\mathcal{C}}_n(\cdot,\cdot,\cdot,\cdot)\}_{n=1}^{N-1}$.

**Algorithm 2:** Regression technique for continuation value: PR $-$ 1D

**input:** $\{\hat{V}_{n+1}(L_{n+1}^j, I_{n+1}^j, m)\}_{j=1}^{M}$, $\{\phi_k\}_{k=1}^{K}$. **for** $i = 0$ *to* $M_I$ **do**
  $\boldsymbol{\alpha}_n^m = \arg\min_a \Big\{ \sum_{j=1}^{M} \Big( \hat{V}_{n+1}(L_{n+1}^j, I^i, m) - \sum_{k=1}^{K} a_k \phi_k(L_n^j) \Big)^2 \Big\}$;
  Define $\tilde{\mathcal{C}}_n(L, I^i, m) = \sum_{k=1}^{K} \alpha_t^{k,i,m} \phi_k(x)$, $m = 0, 1$;
**end**
Define
$\hat{\mathcal{C}}_n(L, I, m; u) = \frac{I^{i+1} - I + B_n \Delta t_n}{I^{i+1} - I^i} \tilde{\mathcal{C}}_n(L, I^i, m; u) + \frac{I - B_n \Delta t_n - I^i}{I^{i+1} - I^i} \tilde{\mathcal{C}}_n(L, I^{i+1}, m; u)$, $I \in [I^i, I^{i+1})$, $m = 0, 1$.
**output:** $\hat{\mathcal{C}}_n(\cdot,\cdot,\cdot,\cdot)$.

**Algorithm 3:** Regression technique for continuation value: RL and PR − 2D

**input:** $\{\hat{V}_{n+1}(L^j_{n+1}, I^j_{n+1}, m)\}_{j=1}^{M}$, $\{\phi_k\}_{k=1}^{K}$.
**if** Regress Later **then**
  $r = n + 1$ ;
**else if** Regress Now **then**
  $r = n$ ;
$\boldsymbol{\alpha}_n^m = \arg\min_a \Big\{ \sum_{j=1}^{M} \Big( \hat{V}_{n+1}(L^j_r, I^j_{n+1}, m) - \sum_{k=1}^{K} a_k \phi_k(L^j_r, I^j_{n+1}) \Big)^2 \Big\}$, $m = 0, 1$
Define $\hat{\mathcal{C}}_n(L, I, m, u) = \sum_{k=1}^{K} \alpha_n^{k,m} \mathbb{E}[\phi_k(L_r, I_{n+1}) | L, I, m, u]$.
**output:** $\hat{\mathcal{C}}_n(\cdot, \cdot, \cdot, \cdot)$.

**Algorithm 4:** Out of sample simulation

**input:** simulation budget $M'$, continuation value function $\{\mathcal{C}_n(\cdot, \cdot, \cdot, \cdot)\}_{n=1}^{N-1}$.
**for** $n = 1$ *to* $N - 1$ **do**
  **for** $j = 1$ *to* $M'$ **do**
    $u^j_n = \arg\min_{u \in \mathcal{U}_n} \Big( \pi^{\Delta}(L^j_n, I^j_n, m^j_n, u) + \hat{\mathcal{C}}_n(L^j_n, I^j_n, m^j_n; u) \Big)$.
    Set $I^j_{n+1} = I^j_n - B^j_n \Delta t_n$ and $m^j_{n+1} = \mathbb{1}_{u^j_n > 0}$.
    Simulate $L^j_{n+1}$ using its dynamics.
    $J^j_{n+1} = J^j_n + \pi^{\Delta}(L^j_n, I^j_n, m^j_n, u^j_n)$.
  **end**
**end**
$V_0(L, I, m) = \frac{1}{M'} \sum_{j=1}^{M'} (J^j_N + W(I^j_N))$.
**output:** Value function $V$.

optimal control $u^j_n$ using the estimated conditional expectation $\hat{\mathcal{C}}_{n+1}(\cdot, \cdot, \cdot)$ and update the value of the controlled processes $(I^j_{n+1}, m^j_{n+1})$. Repeating this until the final time $t_N$, we compute the out-of-sample valuation as:

$$\hat{V}_0(L, I, m) = \frac{\sum_{j=1}^{M'} \Big[ \sum_{n=1}^{N} \pi^{\Delta}(L^j_n, I^j_n, m^j_n, u^j_n) + W(I^j_N) \Big]}{M' \cdot N}$$

Algorithm 4 summarizes the sequence of steps for out-of-sample evaluation.

## 3.5 Numerical Experiments

In this section we use the algorithms introduced in section 3.4 to solve a simple instance of the microgrid management problem. We fix some base parameters and test the three algorithms; the one performing best is then used to study the sensitivity of the control policy and of the operational costs on changes in system parameters, hoping to gain some insight on the optimal design of the microgrid.

We now list the base parameters chosen for the numerical experiments. For the meaning of the parameters refer to section 3.2.

| |
|---|
| $b = 0.5,\ \Lambda_n = 0,\ \sigma = 2, T = 100$ (hours), $\Delta t = 0.25$ (hours) |
| $I_{\max} = 10$ (kWh), $B_{\min} = -6, B_{\max} = 6$ (kW) |
| $\rho(u) = [(u-u^*)^3 + (u^*)^3 + u]/10$ (litre/kW), $u^* = 6$ (kW), $p = 1$ €, $[\underline{u}, \overline{u}] = [1, 10]$ (kW) |
| $C_1 = 0$ €, $K(0,1) = 5$ €, $W(\cdot) = 0$ |

Table 3.1: Parameters for the Microgrid.

According to the parameters table above, and recalling remark 4 the net demand has the following dynamics:

$$L_{n+1} = \left(L_n(1 - 0.5\Delta t_n) + \sigma\sqrt{\Delta t_n}\xi_n\right) \wedge 10, \quad n \in \{0, 1, \ldots, N-1\}, \tag{3.19}$$

where $\xi_n \sim \mathcal{N}(0,1)$.

We decided to use such simple dynamics for illustrative purposes in order to make the sensitivity of the optimal control policy to the remaining parameters more straight forward to understand.

Consider now that for the parameters listed above, the problem is time homogeneous. We have also observed empirically that the estimated continuation values tend to forget the terminal condition rather quickly. We show in Figure 3.1 that the regression coefficients for all algorithms converge to a stationary value time steps, suggesting that optimization ran for longer time horizons would not bring any noticeable effect

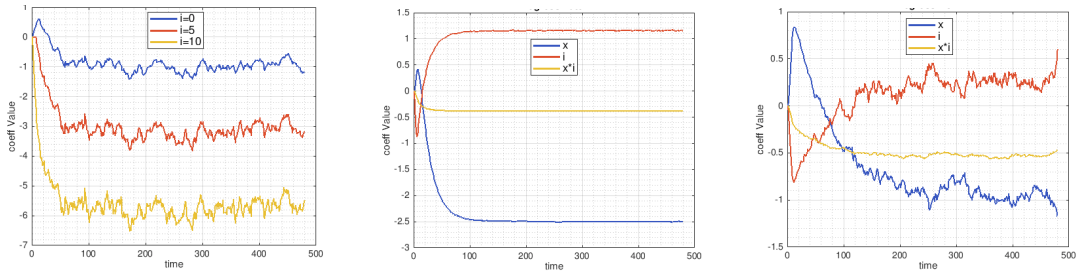



Figure 3.1: Regression Coefficients for the three methods. *Left panel:* regression coefficients for $\{L\}$ at three different inventory levels for PR-1D for $m_n = 1$. *Center and Right Panel:* Estimated regression coefficients corresponding to the basis $\{L, I, L\,I\}$ for RL (center panel) and PR-2D (right panel). Although we used basis function up to polynomial degree 2, we present few coefficients for clarity of presentation. Notice that the time axis is inverted to show the number of time steps computed backward. Remarkable smooth coefficients are computed by the Regress Later algorithm.

to control policy. Since all three methods use polynomial basis of degree two for the projection, it also allows for easy comparison of the dynamics of the coefficients across methods. For example, at inventory level $I = 0$ the dynamics of the coefficient for $L$ achieves same stationary level for both PR-1D and PR-2D. Although an exact comparison is not possible between PR-2D and RL, we continue to observe similar sign and dynamics for each of the coefficients. However, getting away with almost no noise in the dynamics of the estimated coefficients of Regress Later compared to Regress Now is essentially magical.

As a result, we define a stationary policy $u(L, I, m)$ to be used in a longer time horizon than the one employed for its estimation which performance are comparable to the time dependent policy $u_n(L, I, m)$.

We finally tested the value of both stationary and time dependent policy and found that the performance of the stationary policy is comparable to that of the time dependent policy.

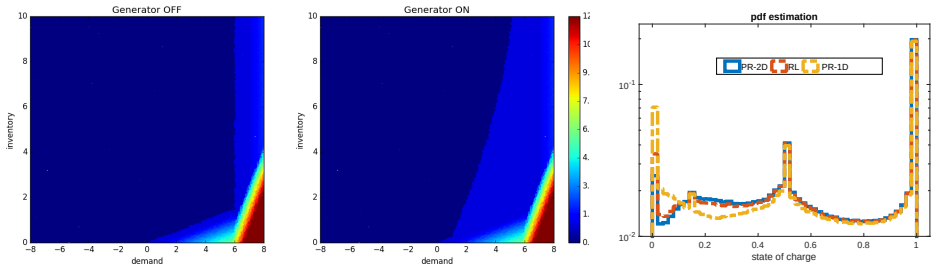



Figure 3.2: *Left/Center Panel:* Control policy $\hat{u}(0, L, I, m)$ for Regress Later algorithm at time $n = 0$. *Right Panel:* Estimated probability density of state of charge of the battery associated with the use of the three algorithms. Notice that RL and PR-2D induce very similar distributions.

### 3.5.1 Analysis of the controllers

In this section we compare the control policies estimated by the three algorithms and we try to assess whether one of the approaches is preferable.

#### 3.5.1.1 Control maps

We compare now the stationary control policies produced by the different algorithms; recall that these policies are feedback to the state, i.e. can be written as function $u^m(L, I)$. Figure 3.2 displays an example of the feedback control policy in the form of control map, a graphical representation of the value of the optimal control for each pair $(L, I)$.

We observed that the three policies agree with the intuition that the diesel generator should produce more power when net demand is high and inventory is low. We can also notice that the switching cost influences the policy, forcing the diesel to keep running for longer in order to charge the battery sufficiently and avoid turning ON and OFF the generator too often. Just by observation of the control maps little difference can be found among the algorithms, we display in Figure 3.2 the effect of the control policy on a the state of charge of the battery. It can be observed from the estimated unconditional probability density of the process $I$ that the policies induced by PR-2D and RL are

very similar. Both seem to induce a peculiar mass of probability around $I_n = 2.5$, differentiating the behavior of the inventory compared to PR-1D. The distribution of the state of charge, obtained by plotting the histogram of all simulations over all time steps, shows that PR-2D and RL does not fully exploit the whole inventory but rather they are more conservative, saving energy to avoid to turn ON the diesel generator in the future. In the next section we will investigate the value associated to this control maps.

#### 3.5.1.2 Performance of the policies

In order to assess the performance of each policy in an unbiased manner, we select a collection of simulated paths of the net demand process $L$, and record the costs associated with managing the microgrid as indicated by each control map.

We first study how the quality of each policy improves when we increase the computational budget $M$ (and the complexity of the projection $K$) for each algorithm to compute the stationary policy. In Figure 3.3, we show the estimated value of the policy when the initial state of the system is $(L, I, m) = (0, 5, 0)$ for polynomial basis functions of increasing degree, for PR-2D. In case of PR-1D we increase the number of discretisation points for the inventory. In particular we make the computational time increase by providing the problem with more training points and more parameters to use in the definition of $\mathcal{C}$ as increasing the number of basis functions. In the case of PR-2D and RL, surprisingly, we notice that the performance of the estimated control improves only when polynomials of even degree are added, and the effect is more prominent for Regress Later.

We notice from the comparison that PR-1D converges quickly, resulting in the best algorithm in terms of trade off between running time and precision. Among the PR-2D and RL (not displayed in order to maintain clear presentation, but available on request),

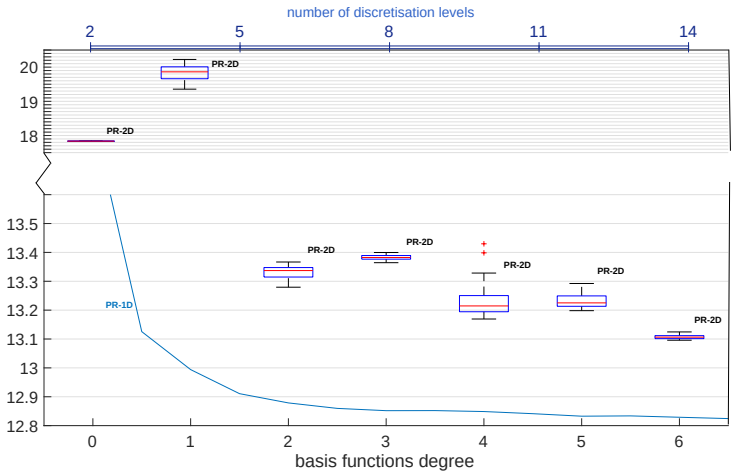



Figure 3.3: Value function $\hat{V}_0$ for PR-2D (lower x-axis represents the polynomial degree) and PR-1D (upper x-axis axis represents the number of inventory levels). Notice the peculiar behaviour of even/odd degree of basis functions in the PR-2D regressions. Similar analysis was performed for Regress Later.

we observe similar bias, however latter has lower standard error. This is not surprising because Regress Later has only one element of approximation error due to finite basis functions while PR-2D has error attributed to two sources, first, due to finite basis function and second, pathwise estimation of the conditional expectation.

### 3.5.2 System behavior

In the previous section we found PR-1D to be the best performing algorithm by our criteria. In the following we shall always employ PR-1D to conduct our study of the sensitivity of the control policy and the associated cost of managing the grid to some of the parameters of the model.

The aim of the section is to build a solid understanding of the behavior of the microgrid in order to get an insight into the optimal design of the system. We decided to study the following aspects of the grid: battery capacity, represented by $I_{max}$; different proportion of renewable production, via the volatility $\sigma$ and the mean reversion $b$;

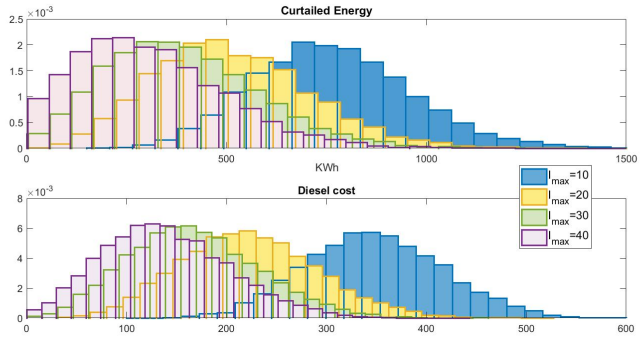



Figure 3.4: Empirical distribution of curtailed energy (top panel) and diesel cost (bottom panel) for different levels of battery capacity. Notice that the decrease in cost and curtailed energy per kWh of additional capacity is smaller for high capacity batteries.

tenable behavior of the policy, via the switching cost $K$ and curtailment cost $C_1$.

In order to be able to carry out our analysis, without introducing cumbersome economic and engineering details regarding the microgrid components, we have to make very simplistic assumptions. Our aim is however to guide the reader through a methodology that can be replicated to study real world microgrid systems.

#### 3.5.2.1 Battery capacity

We study first the behaviour of the system relatively to changes in the capacity of the battery. We would expect to observe negative correlation between the quantity of diesel consumed and the battery size. We display in Figure 3.4 both the quantity of energy curtailed and the cost of running the diesel generator for different values of the battery capacity. We can observe that, as expected, increasing the size of the battery leads to lower diesel usage thanks to the higher proportion of renewable energy that is retained within the system. As the capacity of the battery reaches 30/40 kWh, we start observing a decrease in the cost-reduction per kWh of additional capacity suggesting that further analysis should be run in order to understand up to which size it is worth to pay to add storage capacity to the system.

We show now how to infer information about the optimal sizing of the battery, minimizing the trade off between the installation cost of a bigger battery and the reduced use of the diesel generator. Consider however that including battery ageing in the stochastic control problem is outside the scope of this chapter but rather in this section we present only a post-optimization analysis. Assuming that the microgrid runs under similar conditions for the next 10 years, we can quickly estimate the total throughput of energy for the different battery capacities. Consider now that a battery does not have an infinite lifetime, but rather it should be scrapped after equivalent 4000 cycles (amount of energy for one full charge and discharge). Under the previous assumptions, we can compute how many batteries would be necessary to cover the next 10 years of operations. Similarly, using the data relative to the usage of diesel generator for different levels of capacity, we can compute the operating cost of the diesel generator over the same time period. Further exploiting the assumption about the lifetime of a battery, we obtain the cost of running the grid for 10 years as a function of the number of batteries. To conclude, assuming a linear cost of 400 €/kWh of capacity, we work out the installation cost of the different-size storage devices.

Once this information is collected we search for the minimum of the sum of installation and running cost and, in turn, we compute the optimal capacity. Figure 3.5, on the left, displays a graphical summary of the procedure just described and shows that in our problem the optimal size of the battery is 14 kWh under the current set of assumptions. Further, we study how much our result is affected by the cost per KWh of capacity, repeating the procedure above. We find that, as expected, as cost increases the size of the optimal battery decreases. Figure 3.5, on the right, displays such behaviour.

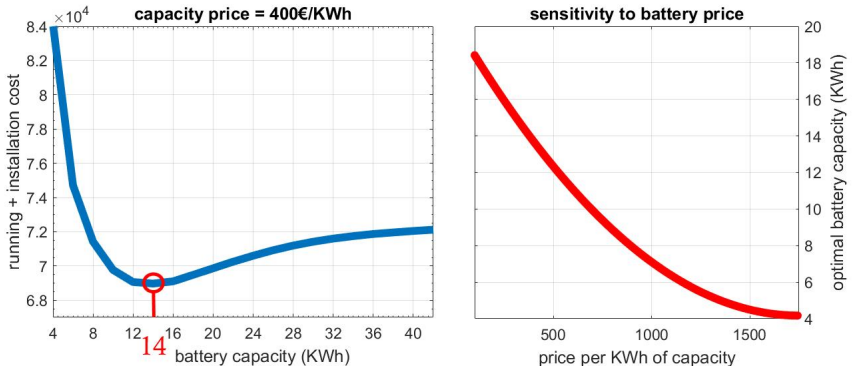



Figure 3.5: *Left panel:* Total cost of installing and running the grid for ten years, assuming we replace the battery every 4000 cycles, against the battery capacity. *Right panel:* Sensitivity of optimal battery capacity with respect to the price of battery energy storage system.

#### 3.5.2.2 Renewable penetration

In this section we want to investigate how robust the microgrid is to higher penetration of renewable generation, or, in other words, to what extent the algorithm can cope with increasing randomness and decreasing predictability of the system. To model this phenomena we assume that greater penetration of renewables can be modeled by increasing both the parameters for volatility $\sigma$ and the mean reversion rate $\lambda$. Increasing these two parameters makes the problem more difficult to solve, given that the control policy can rely less and less on the statistical properties of the process $L$ which approaches white noise as high variance and high mean reversion make the current position of the process not very informative to predict its next one.

In order to establish the real added value provided by our stochastic optimization algorithm, we compare the estimated policy with an heuristic myopic control which can be reproduced in our model solving the dynamic programming equation (3.12) taking constant conditional expectation with respect to the control (greedy policy with respect to the current cost), particularly $\hat{\mathcal{C}} = 0$. We plot the value of the two control policies as function of the increasing learning difficulty in Figure 3.6 where we observe that

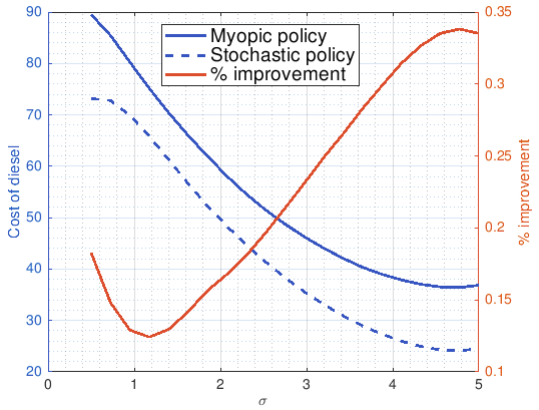



Figure 3.6: The blue lines (solid and dashed, mostly decreasing with y-axis on the left) represents the cost of the diesel usage for myopic and stochastic policy as a function of $\sigma$. The orange curve (mostly increasing with y-axis on the right) represents the percentage improvement in cost when using stochastic policy as a proportion of cost of myopic policy.

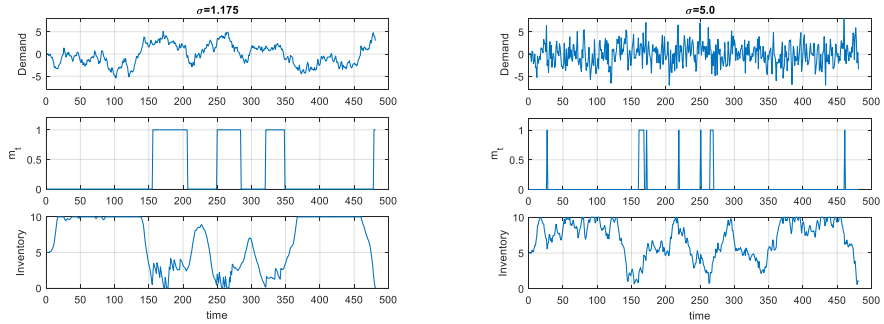



Figure 3.7: A sample path for the dynamics of load demand $L_t$, diesel usage $m_t$ and inventory $I_t$ for low (left panel) and high (right panel) volatility $\sigma$. Mean reversion rate was chosen as $\lambda := \sigma^2/8$, in order to ensure a constant volatility of the process regardless of $\sigma$. Notice the low usage of the diesel generator in the figure on the right compared to the one on the left.

the importance of accounting for the future conditional expectations $\hat{\mathcal{C}}$ increases as the predictability of $L$ decreases.

In Figure 3.6 we present cost of diesel (solid and dashed blue, mostly decreasing with y-axis on the left) as a function of $\sigma$ for myopic and stochastic policy. The orange line (mostly increasing and y-axis on the right) represents the percentage improvement. Since increasing $\sigma$ alters the volatility of the distribution of the process $L$, we define the mean reversion rate $\lambda := \sigma^2/(2c)$ in order to ensure that the volatility of the process is constant while we increase $\sigma$. The stochastic policy leads to at least 12% reduction in the cost of the diesel usage, compared to the myopic policy, and the difference magnifies with increasing "fluctuations" in the process. The decreasing relationship of the cost with $\sigma$ signifies the importance of the battery storage system in the microgrid which absorbs the sharp change in the demand. In Figure 3.7 we compare the demand for two different levels of the $\sigma$, the dynamics of the diesel generator and the inventory. Notice significantly less usage of the diesel for high fluctuations, $\sigma = 5$, compared to $\sigma = 1.175$.

The results of this experiment are affected by the over-pessimistic assumption of modeling greater penetration of renewables with an increasingly unpredictable, and eventually completely random, net demand process. This sort of analysis can however provide insight into how much (weather and load) forecasting capability will be necessary for a given level of renewable penetration.

#### 3.5.2.3 Switching and curtailment

We conclude this section by analyzing the dependence of the system behavior on two key parameters in the model: switching cost $K(0, 1)$ and curtailment cost $C_1$. Switching cost is a system's property and the microgrid controller has little freedom over, however the controller can significantly reduce the amount of curtailed energy by choosing the appropriate curtailment cost. In Figure 3.8, we observe that increasing the curtailment

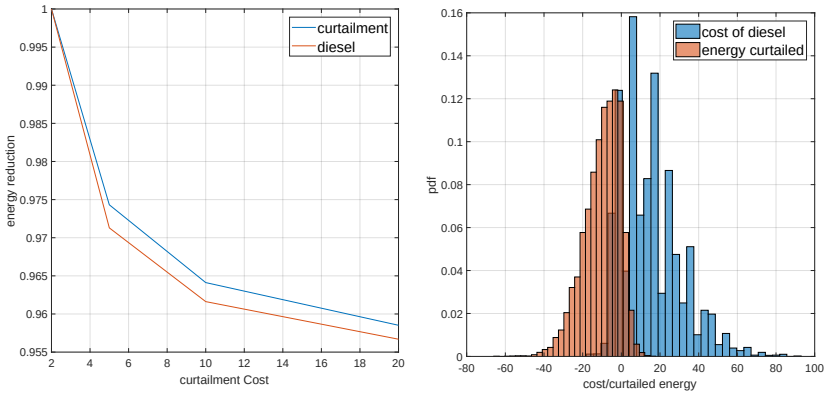



Figure 3.8: Sensitivity to $C_1$. *Left panel:* Total curtailed energy as a proportion of curtailed energy at $C_1 = 2$. *Right panel:* Empirical distribution of difference in the cost of the diesel used (blue color) and curtailed energy (orange) when $C_1 = 20$ against $C_1 = 2$. The histograms represent the differences between the two cases, $C_1 = 20$ and $C_1 = 2$. Notice that higher curtailment cost leads to reduced curtailed energy but at the expense of inefficient diesel usage.

cost reduces the total curtailed energy by approximately 4%. However, it comes at the cost of inefficient usage of the diesel generator, which is represented on the right in the Figure 3.8. The blue historgam represents the difference between the cost of diesel usage for $C_1 = 20$ and $C_1 = 2$. Similarly the orange histogram represents the difference between the energy curtailed for the two cases. Positive diesel cost depicts inefficient usage of the diesel at $C_1 = 20$ compared to $C_1 = 2$. Depending upon the specific cost functional for the diesel, the controller can use an artificial $C_1$ as a parameter in the algorithm to achieve better quality of the optimization.

The optimal policy when the generator is ON $m_t = 1$ is significantly altered depending upon the switching cost. For example, in Figure 3.9, we present the control maps associated with $K(0,1) = 2$ and $K(0,1) = 5$. As expected, larger switching cost disincentivise the controller to switch OFF the diesel generator once it's ON. However, we don't observe "significant" change in the control policy due to increase in switching cost when the generator is OFF.

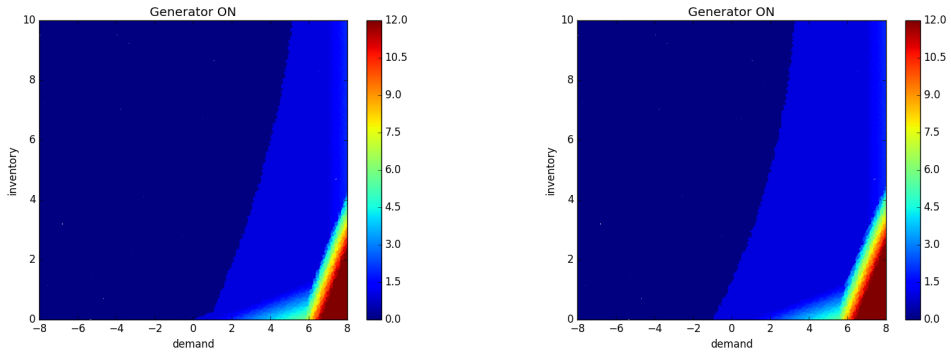



Figure 3.9: Control map $\hat{u}(\cdot, \cdot, 1)$ for different switching costs when the diesel generator is ON. *Left panel:* Switching cost $K(0, 1) = 2$. *Right panel:* Switching cost $K(0, 1) = 5$. Notice the increase in area for light blue (corresponding to $u = 1$) in the figure on the right because of increased switching cost.

## 3.6 Summary

In this chapter we solved the problem of optimal management of a microgrid by employing two algorithms from the Regression Monte Carlo literature, namely: Regress Now and Regress Later. We also evaluated the performance of two different approximation methods (piecewise continuous and global polynomial) for Regress Now algorithm. We find that piecewise continuous approximation for Regress Now significantly outperforms the other methods. Besides algorithm design, we propose a methodology to optimize the design of the grid and determine the optimal sizing of the battery. In addition, we perform a thorough sensitivity analysis to some of the key parameters, showing the robustness of our solution.

# Chapter 4

# Simulation Methods for Stochastic Storage Problems: A Statistical Learning Perspective

*This chapter is the result of a collaboration with Michael Ludkovski and is based on the work [12].*

In this chapter we take a statistical learning perspective to develop the dynamic emulation algorithm (DEA) that unifies the different existing approaches of RMC methods in a single modular template. We then investigate the two central aspects of regression architecture and experimental design that constitute DEA. For the regression piece, we discuss various non-parametric approaches, in particular introducing the use of Gaussian process regression in the context of stochastic storage. For simulation design, we compare the performance of traditional design (grid discretization), against space-filling, and several adaptive alternatives. The overall DEA template is illustrated with multiple examples drawing from natural gas storage valuation and optimal control of back-up generator in a microgrid.

## 4.1 Introduction

Stochastic storage problems concern the optimal use of a limited inventory capacity under uncertainty, motivated by models from commodity management, energy supply, operations research and supply chains. The common thread in all these settings is deciding how to optimally add and reduce inventory as the system state stochastically fluctuates over time. For example, in the gas storage version [26, 6, 54, 5, 29, 55, 19, 56, 34, 7], the objective is to manage an underground cavern through buying and selling natural gas, with the principal stochastic factor being the commodity price. In the context of microgrid (Chapter 3), the objective is to deliver electricity at the lowest cost by maximizing inter-temporal storage linked to a renewable intermittent power source (say from solar or wind), so as to minimize the use of non-renewable backup generator. In the hydropower pumped storage setup, the objective is to match upstream inflows and downstream energy demands at the lowest cost [38, 57, 58].

### Contributions

In this chapter we present a unified treatment of stochastic storage problems from the statistical learning perspective. We recast simulation-based dynamic programming approaches as an iterative sequence of machine learning tasks, corresponding to approximating the value functions, indexed by the time-step parameter $t$. Equivalently, the tasks can be thought of as learning the underlying $\mathcal{C}$-values, or the optimal feedback control $u_t$.

With the machine learning perspective, the storage model is viewed as a stochastic simulator that produces noisy pathwise observations, and the aim is to recover the latent *average* behavior, i.e. the conditional expectation, by judiciously selecting which simulations to run. This framework naturally emphasizes computational complexity and offers an abstract modular template that accommodates a variety of approximation

techniques. Indeed, our template involves three major pieces: (i) experimental design to determine which simulations to run; (ii) approximation technique for the conditional expectation; (iii) optimization step to recover optimal feedback control. While these sub-problems have been treated variously elsewhere, to our knowledge we are the first ones to fully modularize and distill them in this context. In particular, emphasizing the design aspect of RMC methods was only recently taken up in [8, 59]. We borrow the design framework introduced in [8] for American options, and to our knowledge, this is the first work to explore experimental designs in context of storage problems.

We show that existing proposals for Regression Monte Carlo for storage problems all fit neatly into this template, and moreover our setup furnishes a variety of further improvements. Specifically, we address the following three enhancements:

(A) New simulation designs. To better explore the input space we consider space-filling designs based on quasi Monte Carlo sequences or Latin hypercube sampling. To focus the emulators on the regions of interest, we consider probabilistic design. We also explore various adaptive versions, such as mixtures of space-filling and probabilistic designs, and designs that vary across $t$;

(B) Non-parametric regression architectures for learning the value function. In particular, we document the advantages of using Gaussian Process regression;

(C) Implementations that vary RMC ingredients across time-steps, either deterministically or adaptively, during the backward recursion. This includes alternating between joint-$(P, I)$ and discretized-inventory regression schemes, as well as changing the design size as $t$ changes. We showcase one such approach, addressing a non-smooth terminal condition.

We emphasize the mixing-and-matching aspect, which is especially attractive for implementation in a broad-scope software library where the different methods are ex-

pressed as subroutines accessible via a uniform interface.

The developed Dynamic Emulation Algorithm is applicable in a wide range of stochastic storage settings, being scalable in the dimension of the state variable and potential state dynamics. We illustrate DEA with 4 different extended case-studies. The first three case-studies consider valuation of natural gas storage facilities, starting from a standard benchmark first introduced in Forsyth and Chen [29]. This benchmark is then extended to add switching costs (that make control regime part of the state), and to consider simultaneously optimizing two storage facilities (leading to a 3D state space). Last but not least, we consider an example from microgrid management, solving for the optimal dispatch of backup generator to balance an intermittent renewable power source coupled to a limited battery.

Our developments parallel the recent literature on *optimal stopping*, especially in the context of Bermudan option pricing, where a wealth of strategies have been proposed and investigated [8]. It has been a well known folklore result that storage problems, especially with discrete controls, are "essentially" optimal stopping as far as computational methods are concerned. Nevertheless, the respective knowledge transfer is non-trivial and there remain substantive gaps, which we address herein, between the respective numerical algorithms. Thus, the present chapter "lifts" optimal stopping techniques to the setting of stochastic storage (i.e. optimal switching), and can be seen as a step towards similar treatment of further stochastic control problems, such as optimal impulse, or continuous control.

The rest of this chapter is organized as follows. Section 4.2 describes the classical storage problem and the key ingredients to its solution. Section 4.3 describes the algorithm developed to modularize the solution steps for regression Monte Carlo into a sequence of statistical learning tasks. Section 4.4 discus the mathematics for Gaussian process regression and other popular regression methods used in the literature in the

context of storage problems. In Section 4.5, we introduce different design alternates such as space-filling, adaptive and dynamic to exploit the spatial information and efficiently implement non-parametric regression methods (particularly, Gaussian process regression) utilizing batched design. Sections 4.6 and 4.7 are devoted to numerical illustrations, taking up the gas storage and microgrid management, respectively. Finally, Section 4.8 concludes.

## 4.2 Problem description

A storage problem is exemplified by the presence of *stochastic risk factors* together with an inventory state variable. The risk factors have autonomous dynamics, while the inventory is (fully) controlled by the operator via the storage policy; thus the latter dynamics are endogenized. A second feature of storage problems we consider is their *switching* property: the controller actions consist of directly toggling the storage *regime*. In turn the storage regime drives the dynamics of the inventory. Depending on the setup, the regime is either a control variable, or a part of the system state.

To make our presentation concrete, we focus on the classical storage problem with a stochastic price. Namely, there are two main state variables: $P_t$ and $I_t$. $P_t \in \mathbb{R}_+$ represents the price of the stored commodity; $I_t \in [0, I_{\max}]$ is the inventory level. We present the storage dynamics in discrete-time on a time interval $[0, T]$ discretized into a finite grid $0 = t_0 < \dots t_n \dots < t_N = T$, such that $t_n = n\Delta t = n\frac{T}{N}$.

For the price, we assume exogenous Markovian dynamics of the form

$$P_{n+1} = P_n + b(n, P_n)\Delta t_n + \sigma(n, P_n)\,\Delta W_n, \tag{4.1}$$

where $(\Delta W_n)$ are exogenous i.i.d. stochastic shocks. For the rest of the article we take $\Delta W_n \sim \mathcal{N}(0, \sqrt{\Delta t})$ representing Brownian motion dynamics; any other (time-

dependent) shocks could be straightforwardly utilized too. We denote by $\mathcal{F}_n$ the $\sigma$-algebra generated by price process up until time $t_n$ and by $\mathbb{F} = (\mathcal{F}_{t_n})$ the corresponding filtration. The inventory level $I_t$ follows

$$I_{n+1} = I_n + a(u_n)\,\Delta t, \tag{4.2}$$

where $u_n$ is the inventory control, representing the rate of storage injection $u > 0$, withdrawal $u < 0$ or holding $u = 0$. The control is linked to the storage regime $m_n$. We assume that there are three regimes $m_n \in \mathcal{J} := \{+1, -1, 0\}$ representing injection, withdrawal and do-nothing respectively. The regime and control are determined by the joint state:

$$m_{n+1} = \mathcal{M}_n(P_n, I_n, m_n), \tag{4.3}$$

$$u_n(m_{n+1}) = \mathcal{A}_n(P_n, I_n, m_n; m_{n+1}). \tag{4.4}$$

Note that the above form implies that at each time-step $t_n$ and state $(P_n, I_n, m_n)$ the controller picks her next regime $m_{n+1}$ which in turn determines her control $u_n(m_{n+1})$. It also directly restricts controls to be of Markovian feedback form, making the policies $(u_n, m_{n+1})$ $(\mathcal{F}_n)$-adapted. As a result, $(P_n, I_n, m_n)$ is a Markov process, adapted to the price filtration $(\mathcal{F}_n)$.

Let $\pi(P, u)$ be the instantaneous profit rate earned by using control $u$ when price is $P$, and $K(i, j) \geq 0$ be the switching cost for switching from regime $i$ to $j$. Then

$$\pi^{\Delta}(P_n, m_n, m_{n+1}) := \pi(P_n, u_n(m_{n+1}))\Delta t_n - K(m_n, m_{n+1}),$$

is the net profit earned during one time-step $[t_n, t_{n+1})$. To denote the cumulative profit of the controller on $[t_n, T]$ along the path specified by $\mathbf{P}_n = P_{t_n:t_N}$ and a selected

sequence of regimes $\mathbf{m}_n := (m_{t_n:t_N})$ (and consequently the control $\mathbf{u}_n := (u_{t_n:t_N})$) we use

$$v_n(\mathbf{P}_n, I_n, \mathbf{m}_n) := \sum_{s=n}^{N-1} e^{-r(t_s-t_n)}\pi^{\Delta}(P_s, m_s, m_{s+1}) + e^{-r(T-t_n)}W(P_N, I_N), \tag{4.5}$$

where $r \geq 0$ is the discount rate and $W(P, I)$ is the terminal condition (typically concerning the final inventory $I$) at the contract expiration. Note that $I_T$ is determined recursively based on $\mathbf{m}_n$ using (4.2). Similarly, because $v(\cdot)$ depends on the initial regime $m_n$ through the switching costs $K(m_n, m_{n+1})$, $m_n$ is part of the current state, impacting the next decision to be made, cf. Figure 4.7 in Section 4.6.4. The goal of the controller is to maximize discounted expected profits on the horizon $[t_n, T]$,

$$V_n(P, I, m) = \sup_{\mathbf{m}_n} \mathbb{E}\left[v_n(\mathbf{P}_n, I_n, \mathbf{m}_n)\middle|\ P_n = P, I_n = I, m_n = m\right], \tag{4.6}$$

$$\text{subject to } I_n \in [I_{min}, I_{max}] \quad \forall s. \tag{4.7}$$

Problem (4.6) belongs to the class of stochastic optimal control, and satisfies the dynamic programming principle (DPP, also known as the Bellman equation) [27]. The DPP implies that $V_n(\cdot)$ satisfies the one-step recursion

$$V_n(P_n, I_n, m_n) = \max_{m \in \mathcal{J}} \mathbb{E}\left[\pi^{\Delta}(P_n, m_n, m) + e^{-r\Delta t_n}V_{n+1}(P_{n+1}, I_{n+1}, m)\middle|P_n\right], \tag{4.8}$$

where the expectation is over the random variable $P_{n+1}$ since the inventory $I_{n+1}$ is fully determined by $I_n$ and the chosen regime $m$ on $[t_n, t_{n+1})$. See [11, 20, 5] for further discussion and proof of DPP in related storage problems. Note that due to the inventory constraints, some of the controls might not be admissible for different initial conditions, so that formally the maximum in (4.8) is over $\mathcal{J} = \mathcal{J}_n(P_n, I_n, m_n) \subseteq \{+1, -1, 0\}$. For instance, if the inventory is zero, $I_n = 0$ further withdrawal is ruled out and

$\mathcal{J}_n(P_n, 0, m_n) = \{0, 1\}$. Such constraints could even be time- or price-dependent, for instance in hydropower management.

**Remark 8** *In our main setup the choice of the control is pre-determined given the stochastic state and the regime. More generally, conditional on the regime there might be a set of admissible controls $\mathcal{U}_n(m_{n+1})$, adding an additional optimization sub-step. For example, we might have $\mathcal{U}_n(+1) = (0, u_{\max}(I_n)]$ and $\mathcal{U}_n(-1) = [u_{\min}(I_n), 0)$, where $u_{max} > 0$ is the maximum injection rate and $u_{min} < 0$ is the largest withdrawal rate (i.e. minimal, negative injection rate). When $|\mathcal{U}| > 1$, the optimization problem (4.8) requires first to find the optimal regime $m_{n+1}$, and secondly to find the optimal control $u_n(m_{n+1})$ admissible to this regime. The original formulation corresponds to $\mathcal{U}$ being a singleton and can be interpreted as a trivial optimization over $\mathcal{U}$, e.g. due to a bang-bang structure. Abstractly, we may always write $u_n(m_{n+1}) = \mathcal{U}_n(P_n, I_n, m_n; m_{n+1})$ subsuming the inner optimizer.*

### 4.2.1 Solution Structure

Due to the Markovian structure, at each time-step $t_n$ and state $(P_n, I_n, m_n)$ the controller picks her next regime $m_{n+1}$ (consequently control $u_n(m_{n+1})$) according to

$$m_{n+1} = m^*(t_n, P, I, m) = \arg\max_{j \in \mathcal{J}} \left\{ \pi^{\Delta}(P, m, j) + e^{-r\Delta t_n} \mathcal{C}_n(P, I + a(u_n(j))\Delta t_n, j) \right\} \tag{4.9}$$

where the continuation value of switching to regime $m$ is

$$\mathcal{C}_n(P, I, m) := \mathbb{E}\left[ V_{n+1}(P_{n+1}, I, m) \middle| P_n = P \right]. \tag{4.10}$$

Conceptually, we have a map from the value function $V$ to $m^*$ and $u^*$, encoded as $m_n^* : (P, I, m) \mapsto \mathcal{J}$. The dependence of the continuation value $\mathcal{C}$, value function $V$ and

the control map $m^*$ on the current regime $m$ is a consequence of the switching costs $K(m_n, m_{n+1})$, see Remark 11.

Conversely, equation (4.6) provides the representation of the value function as a conditional expectation of future profits based on an optimal policy $\mathbf{m}^*$. Therefore, any estimate $\widehat{m} : (t_n, P, I, m) \mapsto \mathcal{J}$ of the control map naturally induces a corresponding estimate $\hat{V}$ of the value function. Specifically, $\widehat{m}$ yields the dynamics

$$\hat{I}_{n+1} = \hat{I}_n + a\big(u_n(\widehat{m}_{n+1}(t_n, P, I, \widehat{m}_n))\big)\Delta t_n,$$

and induces

$$\hat{V}_0(P_0, I_0, m_0) = \mathbb{E}\left[\sum_{n=0}^{N-1} e^{-rt_n}\pi^{\Delta}(P_n, \widehat{m}_n, \widehat{m}_{n+1}) + e^{-rT}W(P_N, \hat{I}_N)\right]. \tag{4.11}$$

While $\hat{I}_n$ does not appear explicitly above, it is crucially driving $\widehat{m}_{n+1}(t_n, P_n, \hat{I}_n, \widehat{m}_n)$.

Figure 4.1 illustrates this dual link by showing a trajectory of $(P_t)$ and several corresponding trajectories of $(\hat{I}_t)$ indexed by their initial inventories $I_0$ (viewed as an external parameter) for a gas storage facility (see section 4.6.1 for more details). One interesting observation is that the dependence of time-$t$ inventory $\hat{I}_t$ on $I_0 = i$ is rather weak, i.e. the inventory levels coalesce: $\hat{I}^i_t = \hat{I}^{i'}_t$ after an initial "transient" time period. Note that because the controls are specified in feedback form, once $\hat{I}^i_t = \hat{I}^{i'}_t$ we have $\widehat{m}^i_s = \widehat{m}^{i'}_s$ for all $s \geq t$ and the inventory paths will stay together forever. The Figure also illustrates the underlying maxim of "buy low, sell high": when $P_t$ is low, controlled inventory $\hat{I}_t$ is high (and increasing), and when $P_t$ is high, $\hat{I}_t$ is low (and shrinking). As a result, we see a clustering of $\hat{I}_t$ around the minimum and maximum storage levels $I_{min}, I_{max}$, indicating the strong constraint imposed by the bounded storage capacity.

To visualize the estimated optimal policy $\widehat{m}$, Figure 4.2a plots the control map $(P, I) \mapsto \widehat{m}(t, P, I)$ at a fixed time step $t$, namely with $T - t = 0.3$ years for the gas

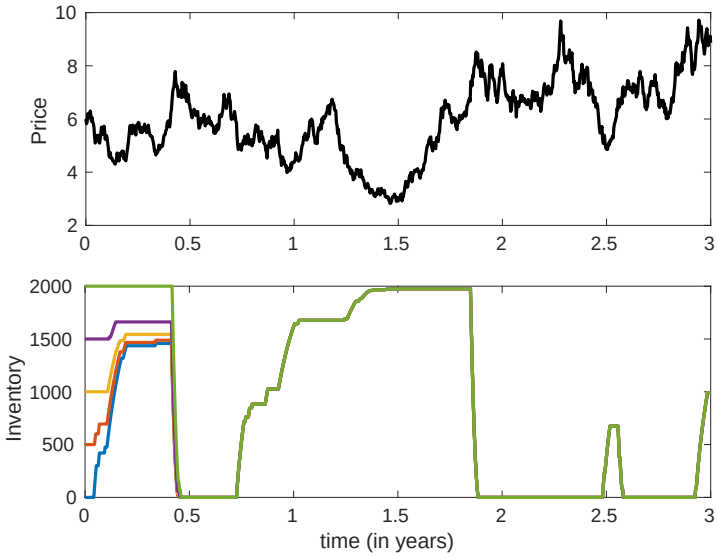



Figure 4.1: *Top panel:* a given trajectory of commodity price $(P_t)$ following logarithmic mean reverting process in (4.27). *Lower panel:* Corresponding trajectories of controlled inventory $\hat{I}_t$ starting at $\hat{I}_0 \in \{0, 500, 1000, 2000\}$. This figure is associated with the gas storage example of Section 4.6.1 using the PR-1D solution scheme.

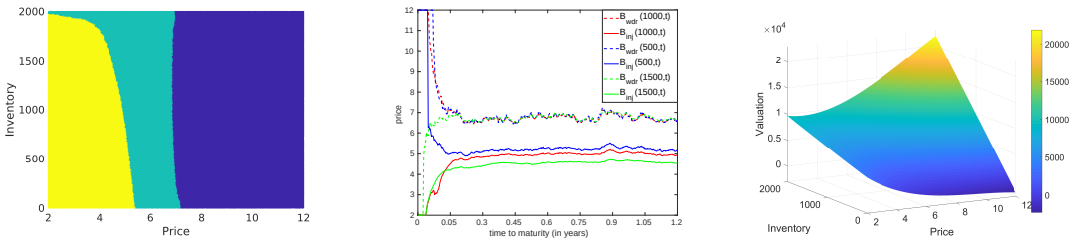
storage example of section 4.6.1. (In that example, there is no dependence on current regime $m$.) The state space is divided into three regions: when $P$ is high it is optimal to withdraw: $\widehat{m} = -1$; if $P$ is low, it is optimal to inject $\widehat{m} = +1$; in the middle, or if inventory is very large, it is optimal to do nothing $\widehat{m} = 0$. Typically, the control map is interpreted by fixing current inventory $I$ and looking at $\widehat{m}$ as a function of $P$. We can then summarize the resulting policy in terms of the *injection/withdrawal boundaries* $B_{inj}(I,t)$ and $B_{wdr}(I,t)$:

$$\begin{cases} B_{inj}(I,t) := \sup\{P : \widehat{m}(t,P,I) = +1\}, \\ B_{wdr}(I,t) := \inf\{P : \widehat{m}(t,P,I) = -1\}. \end{cases} \tag{4.12}$$

Since injection becomes profitable with low prices, $B_{inj}(I,t)$ represents the maximum price for which injection $(\widehat{m}(t,P_t,I) = +1)$ is the optimal policy. Similarly, $B_{wdr}(I,t)$

represents the minimum price for which withdrawal ($\widehat{m}(t, P_t, I) = -1$) is the optimal policy. The interval $[B_{inj}(I,t), B_{wdr}(I,t)]$ is the no-action region. These boundaries are plotted as a function of $T-t$ in Figure 4.2b at three different inventory levels. One prominent feature is the boundary layer as $T-t \to 0$ whereby the policy is primarily driven by the terminal penalty $W(P,I)$ than immediate profit considerations. In this example the "hockey-stick" $W(P_T, I_T)$ forces the controller to target the inventory level $I = 1000$, as $T-t \to 0$, so that injection becomes the optimal policy for $I < 1000$ independent of the price, and withdrawal becomes optimal for $I > 1000$ (the no-action region effectively disappears). Conversely, for large $T-t$, the boundaries $t \mapsto B_{inj}(I,t)$ and $t \mapsto B_{wdr}(I,t)$ are essentially time-stationary.

Finally, Figure 4.2c shows the value function $\hat{V}_0(P,I)$ as a 2-D surface in the price and inventory coordinates. As noted by previous studies, for a fixed price $P$, we observe a linear relationship between value and inventory. However, as a function of $P$, $V_0(\cdot, I)$ is non-linearly decreasing at low inventory levels, and increasing for large inventory.




(a) Control map (b) Prod./Inject. boundaries (c) Value function

Figure 4.2: *Left panel:* snapshot of the control map $\widehat{m}(t, P, I)$ at $t = 2.7$ years. *Middle:* injection $B_{inj}(I, \cdot)$ and withdrawal $B_{wdr}(I, \cdot)$ boundaries as a function of time for $I \in \{500, 1000, 1500\}$. *Right:* value function $\hat{V}_0(P, I)$ at $t = 0$. Results were obtained with conventional design and PR-1D regression.

## 4.3 Dynamic Emulation Algorithm

The numerical algorithms we consider provide approximations, denoted as $\hat{\mathcal{C}}$, to the continuation value in (4.10). This is done via backward induction to estimate $\hat{\mathcal{C}}_n(\cdot)$ as the conditional expectation of $\hat{V}_{n+1}(\cdot)$ for $n = N-1$ to $n = 0$. For this induction, the simulation-based framework relies on a Monte Carlo approximation which consists of generating *pathwise* profits $v_{n+1}(\mathbf{P}_{n+1}, \mathbf{I}_{n+1}, \mathbf{m}_{n+1})$ and then inferring the input-output relationship between $(P_n, I_n, m_n)$ and $v_{n+1}(\cdot)$ via a statistical regression. Indeed, a conditional expectation, which is an $L^2$-projection, is naturally approximated as learning the mean response with a squared-loss criterion. Specifically, regression is implemented as empirical projection onto an approximation space $\mathcal{H}_n$.

Because these pathwise simulations of $P_n \to P_{n+1}$ are themselves part of the algorithm, we must also choose the respective experimental design $\mathcal{D}_n$, i.e. the inputs "$\mathbf{x}$" of the regression (with $v_{n+1}$'s being the corresponding "$\mathbf{y}$"). The core loop, which we dub Dynamic Emulation, thus consists of the following sequence of learning tasks that must be done at each time-step $t$:

Generate design $\to$ Generate pathwise profits $\to$ Estimate the continuation function $\hat{\mathcal{C}}$.

The nomenclature of the DEA emphasizes the associated recursive learning that can be dynamically adjusted. One non-standard feature is that while at each step DEA targets the emulation of the continuation value $\mathcal{C}(t, \cdot)$, the overall performance measure is given by the quality of the final answer $\hat{V}(0, \cdot, \cdot, \cdot)$ on an out-of-sample simulations utilizing the estimates of the continuation value $\mathcal{C}(t, \cdot)$.

The above perspective distills the following key properties of the RMC framework:

- Picking the simulation designs $\mathcal{D}_n$ underlying the emulation;
- Generating the pathwise profits $v_{n+1}$ underlying the regression;
- Picking the projection spaces $\mathcal{H}_n$ for the regression;

- Modularity in terms of $t$, which allows dynamic selections for the above sub-steps as the backward recursion unfolds.

The resulting template for solving the storage problem brings multiple advantages. First, DEA offers a wide latitude in selecting the approximation space $\mathcal{H}_n$. Existing schemes largely concentrated on parametric regression with a fixed set of basis functions. In contrast, DEA allows for any number of regression frameworks, including non-parametric approaches that are more expressive. It also includes the possibility to (adaptively) pick different $\mathcal{H}_n$ across timesteps $t_n$. Second, DEA eliminates the requirement to store global price paths in memory, rather all simulations are performed "online". Third, DEA introduces arbitrary experimental designs; since the latter is primal to maximizing the learning efficiency, there are significant gains to be exploited from judiciously selecting the shape of $\mathcal{D}_n$. Heretofore the literature concentrated on what we call "probabilistic" designs where the values of $P_n$ came from pre-computed price paths. Fourth, DEA allows to vary the number of simulations $M_n$ at time step $t_n$.

The DEA is based on the concept of stochastic simulation. Assuming that we have estimated the continuation function $\hat{\mathcal{C}}_{n+1}(\cdot), \ldots, \hat{\mathcal{C}}_{N-1}(\cdot)$, the objective is to estimate $\hat{\mathcal{C}}_n(\cdot)$ using a simulation design $\mathcal{D}_n := (P_n^j, I_{n+1}^j, m_{n+1}^j, j = 1, \ldots, M_n)$. Note that for conceptual clarity we re-label the current state via the next-step $I_{n+1}$ which is deterministic based on $I_n, m_{n+1}$. The simulator first generates one-step paths $P_n^j \mapsto P_{n+1}^j$, $j = 1, \ldots, M_n$. It then computes the one-step-ahead pathwise profits (cf. [15])

$$v_{n+1}^j := \max_{j' \in \mathcal{J}} \left\{ \pi^\Delta(P_{n+1}^j, m_{n+1}^j, j') + e^{-r\Delta t_n} \hat{\mathcal{C}}_{n+1}(P_{n+1}^j, I_{n+1}^j + a(u(j'))\Delta t_n, j') \right\} \quad (4.13)$$

$$= \pi^\Delta(P_{n+1}^j, m_{n+1}^j, m_{n+2}^j) + e^{-r\Delta t_n} \hat{\mathcal{C}}_{n+1}(P_{n+1}^j, I_{n+2}^j, m_{n+2}^j), \quad (4.14)$$

along each path $j = 1, \ldots, M_n$ where

$$m_{n+2}^j = \arg\max_{j' \in \mathcal{J}} \left\{ \pi^\Delta(P_{n+1}^j, m_{n+1}^j, j') + e^{-r\Delta t_n} \hat{\mathcal{C}}_{n+1}(P_{n+1}^j, I_{n+1}^j + a(u(j'))\Delta t_n, j') \right\}$$
$$I_{n+2}^j = I_{n+1}^j + a(u(m_{n+2}^j))\Delta t_{n+1}.$$

Observe that while (4.13) is based on the definition of (4.10) in terms of the value function, $\hat{V}$ actually never makes an appearance, and $v_{n+1}$ is defined solely from $\hat{\mathcal{C}}_{n+1}(\cdot)$. The approximation task is then to learn (by fitting a statistical model) the input-output relationship between $(P_n^j, I_{n+1}^j, m_{n+1}^j)_{j=1}^{M_n}$ and $(v_{n+1}^j)_{j=1}^{M_n}$ to extract the expected response (i.e. the mathematical expectation of $v_{n+1}$). Note that to evaluate the term on the RHS of (4.13) we need to predict continuation values at arbitrary next-step states $\hat{\mathcal{C}}_{n+1}(P_{n+1}, I_{n+1}, m_{n+1})$.The underlying operations of `fit` and `predict` are thus the two main workhorses of the statistical approximation procedure. In the `fit` step, we seek to $L^2$-project $v_{n+1}$ onto an approximation space $\mathcal{H}_n$:

$$\check{\mathcal{C}}_n(\cdot) := \arg\min_{h_n \in \mathcal{H}_n} \|h_n - v_{n+1}\|_2. \tag{4.15}$$

As a canonical setup, $\mathcal{H}_n = \text{span}(\phi_1, \ldots, \phi_R)$ is the linear space generated by basis functions $\phi_i$ and the approximation $\check{\mathcal{C}}_n(\cdot) = \sum_{i=1}^R \beta_i \phi_i(\cdot)$ is described through its coefficient vector $\vec{\beta}$. To estimate $\vec{\beta}$ we solve a discrete optimization problem based on experimental design $\mathcal{D}_n$ of size $M_n$ and the corresponding realized pathwise values $v_{n+1}^j, j = 1, \ldots, M_n$ from trajectories started at $(P_n^j, I_{n+1}^j, m_{n+1}^j)$:

$$\hat{\mathcal{C}}_n(\cdot,\cdot,\cdot) = \arg\min_{h_n \in \mathcal{H}_n} \sum_{j=1}^{M_n} \left| h_n(P_n^j, I_{n+1}^j, m_{n+1}^j) - v_{n+1}^j \right|^2. \tag{4.16}$$

Thus, $\hat{\mathcal{C}}$ is an empirical approximation of the projection $\check{\mathcal{C}}$, with the corresponding finite-sample error. In particular, subject to mild conditions on $\mathcal{D}_n$, we have $\hat{\mathcal{C}} \to \check{\mathcal{C}}$ as

$M_n \to \infty$.

Since the regime $m_{n+1}$ is a discrete covariate, we repeat the above emulation sub-problem separately for each $m \in \mathcal{J}$. In particular, we select a separate simulation design for different regimes, labeled as $\mathcal{D}_{n,m}$. To emphasize the resulting link between the design and the pathwise simulation we then write $P_n^{j,\mathcal{D}_{n,m}} \mapsto P_{n+1}^{j,\mathcal{D}_{n,m}}$. Also note that both the design and the simulators can be carried out *on-the-fly* during the backward recursion; there is no need to do any pre-simulations.

To complete the description of the algorithm, it remains to address the initialization and post-processing steps. Recall that DEA runs backward over $n$. It is initialized with the known terminal condition

$$\hat{\mathcal{C}}_N(P_N, I_N, m) = W(P_N, I_N) \quad \forall m.$$

The recursive construction then ensures that at step $t_n$ we know the continuation functions $\hat{\mathcal{C}}_n(\cdot,\cdot,\cdot)$ and hence can find $\widehat{m}(t_{n+1}, P, I, m)$ for any $(P, I, m)$ in the state space as in (4.9), as well as

$$\hat{V}_n(P, I, m) = \pi^{\Delta}(P, m, \widehat{m}_{n+1}) + e^{-r\Delta t_n}\hat{\mathcal{C}}_n(P, I + a(u(\widehat{m}_n))\Delta t_n, \widehat{m}(t_{n+1}, P, I, m)). \tag{4.17}$$

In the last step of DEA, we compute an *out-of-sample* estimate of the value function (see Algorithm 4) at $t_0 = 0, P_0, I_0, m_0$ by generating $M'$ new paths $(\mathbf{P}_{0:T}^{m'}, \hat{\mathbf{I}}_{0:T}^{m'}, \widehat{\mathbf{m}}_{0:T}^{m'}), m' = 1, \ldots, M'$, where the pathwise inventory $\hat{I}$ is based on the just-estimated control map $\widehat{m}_{n+1}(\cdot,\cdot,\cdot)$ matching (4.9), and consequently the control $c(\widehat{m})$, cf. Figure 4.1. Thus, $\hat{V}_0(P_0, I_0, m_0) = \frac{1}{M'}\sum_{m'=1}^{M'} v_{0:N}(\mathbf{P}_{0:T}^{m'}, \hat{\mathbf{I}}_{0:T}^{m'}, \widehat{\mathbf{m}}_{0:T}^{m'})$, where $v_{0:N}(\mathbf{P}_{0:T}^{m'}, \hat{\mathbf{I}}_{0:T}^{m'}, \widehat{\mathbf{m}}_{0:T}^{m'})$ is the total cumulative discounted profit over the $N$ time-steps using $\widehat{m}$ for each sample path, cf. (4.11). Since the policy $\widehat{\mathbf{m}}$ is necessarily sub-optimal, $\hat{V}_0(\cdot)$ is a lower bound for the true $V$, modulo the Monte Carlo error from the $M'$ trajectories used in the last

averaging.

Algorithm 5 presents the overall template for solving the storage problem. Assuming $N$ time-steps and a fixed design size of $|\mathcal{D}_n| = M \forall n$, its overall complexity is $\mathcal{O}(NM)$. It is purposely short and abstract, stressing the modular setup and the associated looping. Lines 1-3 initialize with the terminal condition; Line 6 is the emulation step, returning a fitted regression model for $\hat{\mathcal{C}}_n(\cdot)$. Line 7 is the experimental design sub-step. Line 8 is the stochastic simulator which is combined with Lines 10-13 to create the one-step-ahead profits $v_{n+1}$. We conclude this section with several remarks. In the next two sections we then provide menus for the two principal steps in DEA: selecting the approximation space $\mathcal{H}_n$ and the design $\mathcal{D}_{n,m}$.

**Remark 9 (Defining the Pathwise Profits)** *Above we view the continuation value as the conditional expectation of one-step-ahead value function, matching the original Tsitsiklis-van Roy [15] scheme. More generally, we can unroll the Dynamic Programming equation* (4.8) *using the optimal regime choices* $m^*_{n+1}$ *and corresponding controls* $u^*_n$ *for any* $w \geq 1$ *as*

$$
\begin{aligned}
V_n(P_n, I_n, m_n) &= \mathbb{E}[\pi^{\Delta}(P_n, m_n, m^*_{n+1}) + e^{-r\Delta t_n}\mathcal{C}_n(P_n, I_{n+1}, m^*_{n+1})\Big| P_n] \\
&= \mathbb{E}\Big[\pi^{\Delta}(P_n, m_n, m^*_{n+1}) + e^{-r\Delta t_n}\pi^{\Delta}(P_{n+1}, m^*_{n+1}, m^*_{n+2}) + e^{-2r\Delta t_n}\mathcal{C}_{n+1}(P_{n+1}, I^*_{n+2}, m^*_{n+2})\Big| P_n\Big] \\
&\quad \cdots \\
&= \mathbb{E}\Big[v_{n:n+w}(\pi, \mathcal{C})(\boldsymbol{P}_n, \boldsymbol{I}_n, \boldsymbol{m}_n)\Big| P_n, I_n, m^*_n\Big] \qquad (4.18)
\end{aligned}
$$

*in terms of the pathwise gains*

$$
\begin{aligned}
v_{n:n+w}(\pi, \mathcal{C})(\boldsymbol{P}_n, \boldsymbol{I}_n, \boldsymbol{m}_n) := \sum_{s=n}^{n+w-1} & e^{-r(s-n)\Delta t}\pi^{\Delta}(P_s, m_s, m^*_{s+1}) \\
& + e^{-rw\Delta t}\mathcal{C}_{n+w}(P_{n+w}, I_{n+w+1}, m_{n+w+1}). \qquad (4.19)
\end{aligned}
$$

**Algorithm 5:** Dynamic Emulation Algorithm (DEA)

**Data:** $N$ (time steps), $(M_n)$ (simulation budgets per step);

Generate design $\mathcal{D}_{N-1,m} := (\mathbf{P}_{N-1}^{\mathcal{D}_{N-1,m}}, \mathbf{I}_N^{\mathcal{D}_{N-1,m}})$ of size $M_{N-1}$ for each $m \in \mathcal{J}$.

Generate one-step paths $P_{N-1}^{j,\mathcal{D}_{N-1,m}} \mapsto P_N^{j,\mathcal{D}_{N-1,m}}$ for $j = 1, \ldots, M_{N-1}$, $m \in \mathcal{J}$

Terminal condition: $v_{N,m}^j \leftarrow W(P_N^{j,\mathcal{D}_{N-1,m}}, I_N^{j,\mathcal{D}_{N-1,m}})$ for $j = 1, \ldots, M_{N-1}$, $m \in \mathcal{J}$

**for** $n = N-1, \ldots, 1$ **do**

  **for** $m \in \mathcal{J}$ **do**

    Fit: $\hat{\mathcal{C}}_n(\cdot, \cdot, m) \leftarrow \arg\min_{h_n \in \mathcal{H}_n} \sum_{j=1}^{M_n} |h_n(P_n^{j,\mathcal{D}_{n,m}}, I_{n+1}^{j,\mathcal{D}_{n,m}}) - v_{n+1,m}^j|^2$

    Generate design $\mathcal{D}_{n-1,m} := (\mathbf{P}_{n-1}^{\mathcal{D}_{n-1,m}}, \mathbf{I}_n^{\mathcal{D}_{n-1,m}})$ of size $M_{n-1}$ for each $m \in \mathcal{J}$

    Generate one-step paths $P_{n-1}^{j,\mathcal{D}_{n-1,m}} \mapsto P_n^{j,\mathcal{D}_{n-1,m}}$ for $j = 1, \ldots, M_{n-1}$

  **end**

  **for** $j = 1, \ldots, M_{n-1}$ *and* $m \in \mathcal{J}$ **do**

    Predict: $\tilde{m} \leftarrow \arg\max_{j' \in \mathcal{J}} \{\pi^\Delta(P_n^{j,\mathcal{D}_{n-1,m}}, m, j') + e^{-r\Delta t_n} \hat{\mathcal{C}}_n(P_n^{j,\mathcal{D}_{n-1,m}}, I_n^{j,\mathcal{D}_{n-1,m}} + a(u_n(j'))\Delta t_n, j')\}$

    $v_{n,m}^j \leftarrow \pi^\Delta(P_n^{j,\mathcal{D}_{n-1,m}}, m, \tilde{m}) + e^{-r\Delta t_n} \hat{\mathcal{C}}_n(P_n^{j,\mathcal{D}_{n-1,m}}, I_n^{j,\mathcal{D}_{n-1,m}} + a(u_n(\tilde{m}))\Delta t_n, \tilde{m})$

  **end**

**end**

return $\{\hat{\mathcal{C}}_n(\cdot, \cdot, m)\}_{n=1, m \in \mathcal{J}}^{N-1}$

*Similarly, the continuation value* $\mathcal{C}_n(P_n, I_{n+1}, m_{n+1})$ *can be written as*

$$\mathcal{C}_n(P_n, I_{n+1}, m_{n+1}) = \mathbb{E}\Big[v_{n+1:n+w}(\pi, \mathcal{C})(\boldsymbol{P}_{n+1}, \boldsymbol{I}_{n+1}, \boldsymbol{m}_{n+1}) | P_n, I_{n+1}, m_{n+1}\Big]. \tag{4.20}$$

*Then one can use the* $v_{n:n+w}$ *to construct alternative stochastic simulators that would lead to further ways for estimating* $\hat{\mathcal{C}}_n(\cdot)$*. The partial-path construction in* (4.20) *can be traced back to [60, 61]. It encompasses the TvR choice* $w = 1$ *that we employ in this article, and the Longstaff-Schwartz (CLS) algorithm [16] where* $w = N - n - 1$*. Note that* (4.19) *also nests the final pathwise profits* $v_{0:N}$ *used for estimating* $\hat{V}_0(\cdot)$*.*

**Remark 10 (No More Global Paths)** *Traditionally RMC is implemented by gen-*

*erating $M$* global *paths $(P_n^j)$ for the exogenous (price) process starting at $t = t_0$ until maturity $t_N$, which are then stored permanently in memory for the entire backward induction, introducing significant overhead (see Chapter 3). Algorithm 5 replaces this with a design $\mathcal{D}_n$ and the associated one-step trajectories $(P_{n:n+1}^j, I_{n+1}^j)$.*

**Remark 11 (Dimension of the Regression Problem)** *In the case where there are no switching costs $K(i,j) \equiv 0 \ \forall i,j$, the dimensionality of the regression problem can be reduced from 3 to 2. Indeed, since the inventory process $I_t$ is completely controlled, the continuation value only depends on the inventory and regime $(I_n, m_n)$ through the next-step inventory $I_{n+1} = I_n + a(u(m_{n+1}))\Delta t$. Analogously, the value function is then independent of the current regime, $V_n(P_n, I_n)$. With a slight abuse of notation we then write $\mathcal{C}_n(P_n, I_{n+1})$, working with the projection subspace generated by $(P_n, I_{n+1})$.*

*When switching costs are present, the same reduction is possible during the regression, but the present regime $m_n$ remains a part of the state since it affects the continuation value $\mathcal{C}$, so no overall dimension savings are achieved. Practically, this is handled by solving a distinct regression for each $m \in \mathcal{J}$ as in* (4.16).

**Remark 12 (Regress Now vs Regress Later)** *An alternative to* (4.16) *is to use as state variables $(P_{n+1}, I_{n+1})$ during the projection, and then take the conditional expectation analytically:*

$$\check{\mathcal{C}}_n(P, I, m) = \mathbb{E}\left[ \arg\min_{h_{n+1} \in \mathcal{H}_{n+1}} \sum_{j=1}^{M} |h_{n+1}(P_{n+1}^j, I_{n+1}^j) - v_{n+1}^j|^2 \middle| \ P_n = P, I_n = I, m_n = m \right]. \tag{4.21}$$

*This is known as Regress Later Monte Carlo (RLMC) and lowers the variance in the estimated $\hat{\mathcal{C}}$ (see Chapter 3). However, the requirement of closed form expressions for the conditional expectation generally rules out use of RLMC with non-parametric approximation spaces.*

**Remark 13 (Role of the Testing Set)** *Note that the final estimate $\hat{V}_0(P_0, I_0, m_0)$ depends on the out-of-sample simulations $\boldsymbol{P}_{0:N}^{1:M'}$, as well as the in-sample simulations $(\mathbf{P}_{n:n+1})$. To facilitate comparison of different methods, where possible we fix*

*the test scenario database* ($\mathbf{P}_{0:N}^{m'}$)*, whereby we can directly evaluate the different controls/cumulative revenues obtained along a given sample path of the price process.*

**Remark 14 (DEA vs. Conventional RMC for Storage Problems)** *The DEA template nests essentially all existing RMC approaches (discussed in Chapter 3) to* (4.6)*. For example, inventory back-propagation can be interpreted as a specific recipe how to build* $\mathcal{D}_n$ *given* $\mathcal{D}_{n+1}$ *and the control map at step* $t_{n+1}$*. Inventory discretization (PR-1D in Chapter 3) is a specific combination of* $\mathcal{D}_n$ *and* $\mathcal{H}_n$ *that applies piecewise regression plus interpolation, see Section 4.4.1. In particular, the DEA emphasizes the unified treatment of* $P$ *and* $I$ *dimensions, with their stochastic/endogenous dynamics only implicitly appearing as a structural property of the pathwise* $v_{n+1}$ *simulator. Hence, we decouple the backward induction inherent to DPP from the emulation task that is based on the forward stochastic simulator. In turn, the template offers many new strategies (see below) that can (i) improve statistical efficiency, i.e. better use of the computational resources, such as speed, memory, and simulation budget; (ii) lead to more automated solvers that require less fine-tuning (e.g. no need to directly specify basis functions) and adapt to the problem structure; (iii) facilitate application of RMC on new problem instances through schemes that are less sensitive to dimensionality, model dynamics, and payoff format; (iv) create new links between RMC and existing emulation strategies in machine learning (including the ability to re-use existing code).*

## 4.4 Approximation Spaces

In this section we fix a given time step $t_n$ and consider the problem of approximating the continuation value $\mathcal{C}_n(P, I, m)$ viewed as a function $h_n(P, I)$, with $m$ treated as a discrete parameter ("factor" level). Below we generically use $\mathbf{x} = (P_n^j, I_{n+1}^j)_{j=1}^M$ and $\mathbf{y} = (v_{n+1}^j)_{j=1}^M$ to represent the dataset used during the regression.

The statistical assumption is that the input-output relationship is described by

$$y^j = h(x^j) + \sigma^2 \xi, \text{ where } \xi \sim \mathcal{N}(0,1), \tag{4.22}$$

where $h \in \mathcal{H}_n$ is the unknown function to be learned, and $\sigma^2\xi$ is the noise. In our case, the noise is due to the stochastic shocks in $(P_{n+1})$ which induce variation in the realized pathwise profit relative to the continuation value.

Selection of $\mathcal{H}_n$ is key because the intrinsic approximation error, i.e. the distance between the true $\mathcal{C}_n(\cdot)$ and the closest element in $\mathcal{H}_n$, strongly affects the quality of the solution. Among schemes that have been explored in the literature are global polynomial regression [20, 6, 5], radial basis functions [19], support vector regressions [55], kernel regressions [62], neural networks [63], and piecewise linear regression [7].

All of the above can be straightforwardly implemented within the DEA template, so that Algorithm 5 nests all these proposals. Below, we provide more details on three representative schemes which in our experience have been most promising. After reviewing the piecewise regression and the inventory-discretization approaches, we discuss non-parametric regressions where $\mathcal{H}_n$ is characterized through the design sites in $\mathcal{D}_n$. In particular, inspired by the recent work [8] on Bermudan options, we introduce Gaussian process (GP) regression for solving storage problems. To our knowledge, ours is the first paper to use GPs in such context.

### 4.4.1 Bivariate piecewise approximation

The classical regression framework is a linear parametric model where $\mathcal{H}_n$ is the vector space spanned by some basis functions $(\phi_i)$. Then the prediction at a generic $x_*$ is controlled by the regression coefficients $\vec{\beta}$: $h(x_*) = \vec{\beta}^T \vec{\phi}(x_*)$. A simple choice dating back to Longstaff-Schwartz's seminal work [16] is to use polynomial bases. Such polynomial regression (PR) has also been employed for storage problems in [20, 6, 5].

A degree-$r$ global polynomial approximation $h_n = \sum_i \beta_i \phi_i$, has $(r+1)(r/2+1)$ basis functions and takes $\phi_i(P,I) = P^{\alpha_1(i)} I^{\alpha_2(i)}$, where the total degree of the basis function is $\alpha_1 + \alpha_2 \leq r$. For example, a global quadratic approximation ($r = 2$) has 6 basis functions $\{1, P, P^2, I, I^2, P \cdot I\}$, and a cubic PR has 10 basis functions. Our experiments indicate that typically PR leads to poor performance due to the resulting stringent constraints on the shape of the continuation value and consequent back-propagation of error.

A popular alternative is to use piecewise approximations based on partitioning the space of $(P,I)$, restricted to $[\min_{1\leq j\leq M} P_i^j, \max_{1\leq j\leq M} P_i^j] \times [I_{\min}, I_{\max}]$, into $\tilde{M} = M_P \times M_I$ rectangular sub-domains $\mathcal{D}_{i_1,i_2}$, $i_1 = 1, 2, \ldots, M_P; i_2 = 1, 2, \ldots, M_I$. Piecewise regression allows to localize the projection errors (while global regression is apt to oscillate) and tends to be more stable empirically. Relative to PR, piecewise regressions are also more "robust" to fitting arbitrary shapes of the continuation value. We then consider basis functions of the form $\{\phi_g^{i_1,i_2}\}$, with support restricted to $\mathcal{D}_{i_1,i_2}$. For example, in piecewise linear approximation we have $g = 1, 2, 3$ with

$$\begin{aligned}
\phi_1^{i_1,i_2}(P,I) &= \mathbf{1}_{(P,I)\in\mathcal{D}_{i_1,i_2}} \\
\phi_2^{i_1,i_2}(P,I) &= P \cdot \mathbf{1}_{(P,I)\in\mathcal{D}_{i_1,i_2}} \\
\phi_3^{i_1,i_2}(P,I) &= I \cdot \mathbf{1}_{(P,I)\in\mathcal{D}_{i_1,i_2}}.
\end{aligned}$$

Overall we then have $3M_PM_I$ coefficients to be estimated. Higher-degree terms could also be added, e.g. a cross-term $P{\cdot}I$ or quadratic terms $P^2, I^2$. Piecewise regression offers a divide-and-conquer advantage with the overall fitting done via a loop across $\mathcal{D}_{i_1,i_2}$'s; in each instance only a small subset of the data is selected to learn a few coefficients. This decreases the overall workload of the regression substep, and allows for parallel processing. The main disadvantage of this approach is the inherent discontinuity in $\hat{\mathcal{C}}$ at the sub-domain boundaries, and the need to specify $M_P, M_I$ and then construct the rectangular sub-domains $\mathcal{D}_{i_1,i_2}$. We refer to [7] for several adaptive constructions,

including equal-weighted and equi-gridded.

### Piecewise continuous approximation

As discussed in Chapter 3, Section 3.4.1, another approach is to construct a piecewise approximation that is continuous through a linear interpolation. For example, after discretizing the endogenous inventory variable into $M_I + 1$ levels $I_0, I_1, \ldots, I_{M_I}$, we fit an independent degree-$M_P$ monomial in $P$ for each level, i.e. optimize for $\hat{h}^j(P) := \sum_i \beta_{ij}\phi_i(P)$ for $j = 0, \ldots, M_I$, giving a total of $(M_I + 1)M_P$ regression coefficients. (Conversely, one could also fit a piecewise linear model with $M_P$ sub-domains in the $P$-dimension.) The final interpolated prediction for arbitrary $I \in (I_j, I_{j+1})$ is then piece-wisely defined as

$$\widehat{h}_n(P, I) := \delta(I)\ \hat{h}^j(P) + (1 - \delta(I))\ \hat{h}^{j+1}(P), \tag{4.23}$$

where $\delta(I) = \frac{I - I_j}{I_{j+1} - I_j}$. This scheme leverages the fact that the stochastic shocks are only in $P$, and effectively replaces the problem of learning $\hat{\mathcal{C}}$ over $(P, I)$ with a collection of one-dimensional (hence simpler) regressions in $P$ only. In principle this allows to re-use $P$-simulations across different $I$-levels. It is intrinsically smooth in $P$ and piecewise linear in $I$.

## 4.4.2 Local polynomial regression (LOESS)

Local regressions minimize the worry regarding the choice of basis functions by constructing a non-parametric fit that solves an optimization problem at each predictive site. Given a dataset $\{\mathbf{x}, \mathbf{y}\}$ (as a reminder, $x \equiv (P, I)$ is 2-dimensional throughout this section), the prediction using LOESS at $x_*$ is $h(x_*) = \sum_{i=1}^{r} \beta_i(x_*)\phi_i(x_*)$ where the *local*

coefficients $\beta_i(x_*)$ are determined from the weighted least-squares regression

$$\vec{\beta}(x_*) = \arg\min_{\vec{\beta}\in\mathbb{R}^r} \frac{1}{M}\sum_{j=1}^{M} \kappa(x_*, x^j)\left[y^j - \vec{\beta}^T\vec{\phi}(x^j)\right]^2. \tag{4.24}$$

The weight function $\kappa(x_*, x) \in [0,1]$ gives more weight to $y^j$'s from inputs close to $x_*$, akin to kernel regression. Since a separate optimization is needed for each $x_*$, to make $M'$ predictions (4.24) has complexity of $\mathcal{O}(MM')$. Implemented in the context of real options for mining by [62], LOESS was enhanced through a "sliding trick" that reduces complexity to $\mathcal{O}((M + M')\log M)$.

### 4.4.3 Gaussian process regression (GPR)

Gaussian process regression is a non-parameteric technique widely used in spatial statistics and more recently for emulation tasks in machine learning. Besides its ability to efficiently reconstruct non-linear input-output relationships, its popularity is attributed to very few tunable hyperparameters, intrinsic smoothness of the obtained approximation, and symbiotic links to adaptive experimental designs. In our context, GPR offers a flexible alternative to parametric regression, obviating the need to directly specify regression bases or manually construct the mesh within the piecewise approach. GPR is superficially similar to LOESS, in that the prediction $h(x_*)$ is a weighted average of sampled outputs $\mathbf{y}$. The underlying relationship $h : x \mapsto y$ is taken to be a realization of a Gaussian random field, i.e. $\{h(x^i)\}_{i=1}^M$ is a sample from the multivariate Gaussian distribution with mean $\{m(x^i)\}_{i=1}^M$ and covariance function $\{\kappa(x^i, x^j)\}_{i,j=1}^M$. Among many options in the literature, arguably the most popular is the squared exponential kernel,

$$\kappa(x^i, x^j) = \sigma_f^2 \exp\Big(-\frac{1}{2}(x^i - x^j)^T\mathbf{\Sigma}^{-1}(x^i - x^j)\Big),$$

where $\mathbf{\Sigma}$ is a diagonal matrix. Traditionally, diagonal elements of the matrix $\mathbf{\Sigma}$ are known as the lengthscale parameters, and $\sigma_f^2$ as the signal variance. Together they are often referred to as hyper-parameters. While the lengthscale parameters determine the smoothness of the surface in the respective dimension, $\sigma_f^2$ determines the amplitude of the fluctuations. In figure 4.2c we observed that for a fixed price, the continuation value function is linear in the inventory dimension, and has non-linear behavior in the price dimension. GPR captures this difference through its scale parameters, which will be "large" for inventory (slow decay of correlation, so little curvature) and "small" for price dimension (fast correlation decay allow for wiggles in terms of $P$). The hyperparameters are estimated through likelihood maximization. For the prior mean we take $m(x) = \beta_0$ where $\beta_0$ is learned together with the other hyperparameters.

For any site $x_*$, $h(x_*)$ is a random variable whose conditional distribution given $\{\mathbf{x}, \mathbf{y}\}$ is:

$$h(x_*)|\mathbf{y} \sim \mathcal{N}\Big(m(x_*) + H_*\mathbf{H}^{-1}(\mathbf{y} - m(\mathbf{x})), \kappa(x_*, x_*) - \mathbf{H}_*(x_*)\mathbf{H}^{-1}\mathbf{H}_*(x_*)^T\Big) \quad (4.25)$$

where the $N \times N$ matrix covariance matrix $\mathbf{H}$ and the $N \times 1$ vector $\mathbf{H}_*(x_*)$ are

$$\mathbf{H} := \begin{bmatrix} \kappa'(x^1, x^1) & \kappa(x^1, x^2) & \dots & \kappa(x^1, x^N) \\ \kappa(x^2, x^1) & \kappa'(x^2, x^2) & \dots & \kappa(x^2, x^N) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(x^N, x^1) & \kappa(x^N, x^2) & \dots & \kappa'(x^N, x^N) \end{bmatrix}, \ \mathbf{H}_*(x_*)^T := \begin{bmatrix} \kappa(x_*, x^1) \\ \kappa(x_*, x^2) \\ \vdots \\ \kappa(x_*, x^N) \end{bmatrix}, \quad (4.26)$$

where $\kappa'(x^i, x^j) = \kappa(x^i, x^j) + \sigma^2$. Consequently, the prediction at $x_*$ is $\hat{h}(x_*) = m(x_*) + \mathbf{H}_*(x_*)\mathbf{H}^{-1}(\mathbf{y} - m(\mathbf{x}))$ and the posterior GP variance

$$s^2(x_*) := \kappa(x_*, x_*) - \mathbf{H}_*(x_*)\mathbf{H}^{-1}\mathbf{H}_*(x_*)^T$$

provides a measure of uncertainty (akin to standard error) of this prediction. GPR generally has $\mathcal{O}(M^3 + MM'^2)$ complexity, similar to kernel regression. GPR usually performs extremely well but becomes prohibitively expensive for $M \gg 1000$.

**Remark 15** *The above is the most basic version of GP emulation. There is an extensive GP ecosystem containing numerous extensions for optimizing GPR in a specific context. Some of the relevant aspects include more advanced prior mean specification for $m(\cdot)$; other (including adaptive) kernel families $\kappa(\cdot,\cdot)$; heteroskedastic models that can handle state-dependent simulation noise $\sigma^2(\cdot)$; further techniques for selecting GP hyperparameters; and piecewise models for allow for spatially non-stationary covariance. See [64, 8] and references therein.*

## 4.5 Simulation Design

The second central piece of Algorithm 5 concerns the designs $\mathcal{D}_n$. The choice of $\mathcal{D}_n$ directly affects the quality of $\hat{\mathcal{C}}_n(\cdot)$: the approximation will generally be better in regions where $\mathcal{D}_n$ has many input sites, and worse where $\mathcal{D}_n$ is sparse. Trying to make predictions of $\hat{\mathcal{C}}_n(\cdot)$ beyond $\mathcal{D}_n$, i.e. extrapolating, is especially prone to large errors. Thus, the shape of $\mathcal{D}_n$ is akin to introducing weights within the projection (4.15), emphasizing some sub-domains of the input space and de-emphasizing others. This effect is particularly strong for non-parametric or piecewise regression schemes where there is a close link between $\mathcal{D}_n$ and $\mathcal{H}_n$. Spatially, a good simulation design should (i) cover all regions containing potential $(P_n, I_n)$ pairs; (ii) target the region of interest that is most relevant for selecting the optimal action $\widehat{m}(t_n, P_n, I_n)$. Statistically, $\mathcal{D}_n$ ought to maximize the learning rate of $\hat{\mathcal{C}}_n(\cdot)$ and lead to stable regressions, i.e. low empirical sensitivity of $\hat{\mathcal{C}}_n(\cdot)$ across algorithm runs.

Like the regression sub-problem, the template gives the user a wide latitude in se-

lecting $\mathcal{D}_n$ given a simulation budget $M$. We identify four relevant aspects of potential designs: joint vs. product; adaptive vs. space-filling; deterministic vs. stochastic; and unique vs. replicated. Last but not least, we discuss the design size $M$. To summarize the range of simulation designs we use a short-hand nomenclature. Product designs are identified as $\mathcal{D}_P \times \mathcal{D}_I$, while joint designs are denoted with a single symbol. Subscripts are used where necessary to identify the dimensionality of the respective design. Different design types are identified by different letters.

To set the stage, let us summarize the "conventional" design [20, 6, 19]. Traditionally, the design to solve the storage problem relied on a mix of global paths together with inventory discretization. In the price dimension, it consists of choosing $M_P$ initial conditions for the price process $P_0$ at time $t_0$ and sampling $j = 1, \ldots, M_P$ paths $P_{n+1}^j$ following the conditional density $p(t_{n+1}, .|t_n, P_n)$, until terminal date $T$. The resulting collection $(P_n^j)$ is used for the design "mesh" at each $t_n$. For the endogenous inventory dimension, $I$ is discretized into $M_I$ levels: $I^l = l\Delta I$, $\Delta I = \frac{I_{\max} - I_{\min}}{M_I - 1}$, $l = 0, 1, \ldots, M_I - 1$. The overall design $\mathcal{D}_n$ has $M = M_P M_I$ sites and is constructed as the Cartesian product $\{P_n^1, P_n^2, ..., P_n^{M_P}\} \times \{I^0, I^1, ..., I^{N_I - 1}\}$. In our terminology, this is a product design that is adaptive and stochastic in $P$, space-filling and deterministic in $I$, and has no replication. We label it as $\mathcal{P} \times \mathcal{G}$, representing a density-based "probabilistic" design $\mathcal{P}$ in the first coordinate of $x$, and a gridded design $\mathcal{G}$ in the second coordinate.

The shape of a $\mathcal{P} \times \mathcal{G}$ design is convenient for the piecewise continuous regression scheme of Section 4.4.1 that treats the $P$ and $I$ coordinates separately, yielding stable empirical $\hat{\mathcal{C}}_n(\cdot)$. While this recipe can lead to competitive results, it is clearly more prescriptive than adaptive to the particular problem instance. The fact that it relies on a grid in the $I$-coordinate makes it poorly suited for scaling into higher dimension, while using a random sample drawn from the density of $P_{n+1}$ is prone to requiring extrapolation in subsequent `predict` calls. Through DEA we are able to search numerous

other feasible approaches to maximize performance.

### 4.5.1 Space Filling Designs

To achieve the goal of learning $(P, I) \mapsto \mathcal{C}_n(P, I, m)$ we need to explore continuation values throughout the input domain. A simple mechanism to achieve this is to spread out the design sites to fill the space. A gridded design, with design sites uniformly selected using a mesh size $\Delta$ like in the conventional approach above, is an example of such *space-filling* sequences. Exploration through "spreading out" $\mathcal{D}$ can be supported by more rigorous criteria, such as A- or D-optimality that quantify the optimal way to reduce the global $L^2(Leb)$ approximation error.

Space-filling can be done either deterministically or randomly. For the deterministic case, besides the grid $\mathcal{G}$ one may employ various Quasi Monte Carlo (QMC) sequences, for example the Sobol sequences $\mathcal{S}$. Sobol sequences are useful in dimension $d > 1$ where they can produce a $d$-variate space filling design of any size $M$, whereas a grid is limited to rectangular constructions of size $M_P \times M_I$. QMC sequences are also theoretically guaranteed to provide a good "uniform" coverage of the specified rectangular domain. We experimented with the following two setups:

- A 1-D Sobol sequence $\mathcal{S}_1$ of size $M_P$ for the price dimension, restricted to $[P_{\min}, P_{\max}]$ at each time-step $t_n$, $n = 0, \ldots, N$. We then discretize the inventory dimension as $\{I^1, I^2, ..., I^{M_I}\}$, similar to conventional design, and the final $\mathcal{D}$ is the product $\mathcal{S}_1 \times \mathcal{G}$.
- Alternatively, we generate $M$ design sites from the 2-D Sobol sequence $\mathcal{S}_2 = \{P^j, I^j\}_{j=1}^M$ on the restricted domain $[P_{\min}, P_{\max}] \times [I_{\min}, I_{\max}]$.

An example of a randomized space filling design is taking $P_n^j \sim \text{Unif}(P_{\min}, P_{\max})$ i.i.d. Because i.i.d. uniforms tend to cluster, there are variance-reduced versions, such as Latin hypercube sampling (LHS). In two dimensions, the LHS design $\mathcal{L}_2$ stratifies the

input space into a rectangular array and ensures that each row and column has exactly one design site.

Note that if the same deterministic space-filling method is employed across time-steps, the design $\mathcal{D}_n \equiv \mathcal{D}$ becomes identical in $t_n$. This may generate "aliasing" effects from the regression scheme, i.e. approximation artifacts around $(P^j, I^j)$ due to the repeated regressions and respective error back-propagation. Changing or randomizing $\mathcal{D}_n$ across $t_n$'s is one remedy and often preferred as an implementation default. Another challenge with space-filling designs is the need to specify the bounding box $[P_{\min}, P_{\max}] \times [I_{\min}, I_{\max}]$, which is easy in $I$ but not obvious in the unbounded $P$-coordinate.

### 4.5.2 Adaptive Designs

In contrast to space-filling designs that aim to explore the input space, adaptive designs exploit the observation that the quality of $\hat{V}_0$ depends on the correct prediction of storage actions along *controlled* paths $(P_n^{m'}, \hat{I}_n^{m'})$. Therefore, the region of interest at $t_n$ where we should target the best estimation of $\hat{\mathcal{C}}_n(\cdot)$ is the region where the $(P_{0:T}^{m'}, \hat{I}_{0:T}^{m'})$ trajectory is most likely to be. This suggests to use the distribution of $(P_n, \hat{I}_n)$ when constructing $\mathcal{D}_n$, cf. Figure 4.3(c). Information about the distribution of the system state was already leveraged in the conventional approach which used randomized, *probabilistic* design for $\mathcal{D}_P$. Similarly, [19] used a non-uniform discretization in $I$ to refine the mesh closer to $I_{\min}$ and $I_{\max}$, since inventory tends to be either 0% or 100% full, see Figure 4.1.

An ideal adaptive design would reflect the *bivariate* distribution of $(P, \hat{I})$. Of course, this is not directly feasible since $\hat{I}_n$ is endogenous to the controls on $[0, t_n]$. One strategy is to use a small part of the simulation budget to first create a policy using a traditional/space-filling design. In the second step, one runs a forward simulation of this policy to construct a proxy for the joint distribution of $(P_n, \hat{I}_n)$, and hence link $\mathcal{D}_n$

to the joint probabilistic design $\mathcal{P}_2$.

There are numerous variations on generating adaptive designs that reflect some target density $\mathfrak{p}(\cdot)$ (either bivariate or univariate for $P$ and $I$). Instead of a joint design, one could build a product design $\mathcal{P} \times \mathcal{P}$ that matches the respective marginal densities $P_n^j \sim \mathfrak{p}^P(t_n, \cdot)$ and $I_n^j \sim \mathfrak{p}^I(t_n, \cdot)$. Yet another approach is to deterministically quantize the target density $\mathfrak{p}$, similar to numerical quadrature methods, to return a discrete representation with $M_P$ sites.

### 4.5.3 Batched Designs

Non-parametric techniques like GPR and LOESS improve accuracy at the cost of increased regression overhead. Indeed, for both methods the complexity is at least quadratic in the number of sites $M$ which can become prohibitive for $M \gg 1000$. One solution to overcome this hurdle is to use replicates, i.e. re-use the same design site for multiple simulations. Thus, rather than having thousands of distinct design sites equivalent to the simulation budget $M$, we select only a few hundred distinct sites $M_s$ and generate $M_b := M/M_s$ paths from each design site. Formally this means that we distinguish between the $M$ initial conditions $(P_n^j, I_n^j)_{j=1}^M$ for simulating pathwise continuation values and the *unique* design sites $M_s \ll M$ which comprise the design $\bar{\mathcal{D}}$. The latter can then be of any type, space-filling, adaptive, etc. The use of such replicated designs is common in the design of experiments literature, but has been little explored in the RMC context.

Given a design site $(P_n^j, I_{n+1}^j)_{j=1}^{M_s}$, we make $M_b$ draws $P_{n+1}^{j,(m)}$ and evaluate the corresponding pathwise continuation value $v_{n+1}^{(m)}(P_{n+1}^{j,(m)}, I_{n+1}^j)$, $m = 1, \ldots, M_b$. For kernel-based techniques like GP and LOESS one may then work with pre-averaged values,

i.e. first evaluate the empirical average:

$$\bar{v}^j_{n+1}(P^j_n, I^j_{n+1}) = \frac{1}{M_b}\sum_{m=1}^{M_b} v^{(m)}_{n+1},$$

across the $M_b$ replicates. One then feeds the resulting dataset $(P^j_n, I^j_{n+1}, \bar{v}^j_{n+1})_{j=1}^{M_s}$ into the regression equations to estimate the continuation function.

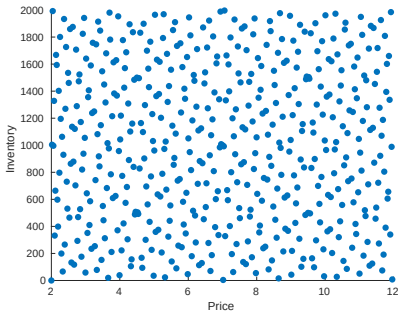
Besides reducing the overall time spent in regressions (which can easily be several orders of magnitude), a batched design has the advantage of reduced simulation variance of $\bar{v}$ at each design site, thus improving the signal-to-noise ratio. While a replicated design is sub-optimal (in terms of maximizing the quality of the statistical approximation), in practice for large $M$ (which are frequently in the hundreds of thousands) the loss of fidelity is minor and is more than warranted given the substantial computational gains.

Beyond uniform batching that uses a fixed $M_b$ number of replicates at each site, one could also employ adaptive batching with site-specific $M_b$ with more replications around the switching boundaries to gain better precision [64].

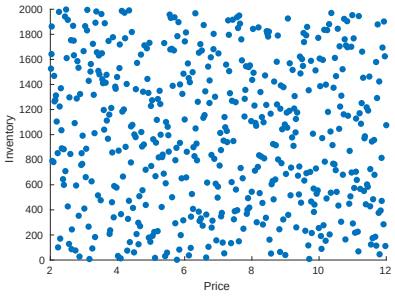
### 4.5.4 Dynamic Designs

DEA lets us easily combine different designs at various time steps. For example, one may vary the step-wise simulation budget, employing larger $M$ near maturity to effectively capture the effect of the terminal conditions in the continuation function, and lesser budget thereafter.

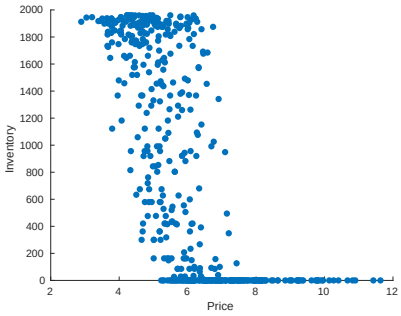
We conclude this section with two illustrations. In Figure 4.3 we display four representative designs: Sobol space-filling sequence in 2D $\mathcal{S}_2$, LHS in 2D $\mathcal{L}_2$, joint probabilistic $\mathcal{P}_2$, and conventional design $\mathcal{P}\times\mathcal{G}$. These will also be used in the numerical examples below. While the Sobol sequence fills the input space with a symmetric pattern, LHS is



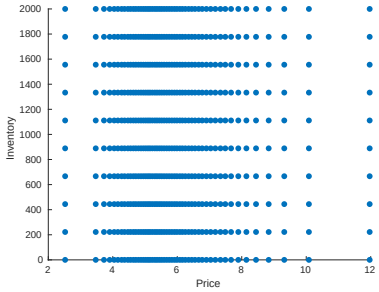

(a) 2D Sobol QMC sequence $\mathcal{S}_2$




(b) 2D LHS design $\mathcal{L}_2$




(c) Joint Probabilistic design $\mathcal{P}_2$




(d) Conventional product design $\mathcal{P} \times \mathcal{G}$

Figure 4.3: Illustration of different simulation designs $\mathcal{D}$. In all cases we take $M = 500$ design sites. Top row, left panel: 2D Sobol QMC sequence $\mathcal{S}_2$, Top row, right panel: 2D LHS design $\mathcal{L}_2$, Bottom row, left panel: Joint Probabilistic design $\mathcal{P}_2$, Bottom row, right panel: Conventional product design $\mathcal{P} \times \mathcal{G}$

randomized. The joint probabilistic design mimics the distribution of the state variables $(P(t), \hat{I}(t))$, putting most sites at the boundaries of the inventory $I_{min}, I_{max}$ and around the mean of the price. Note that this design is very aggressive and only explores a small subset of the input space; therefore, in the following numerical examples we blend (via a statistical mixture) probabilistic designs with other types. Such blending is a natural way to resolve the underlying exploration-exploitation trade-off. Finally, conventional design discretizes the inventory while maintaining the adaptive distribution in the price dimension.

Next, in Figure 4.4 we display the effect of regression and design on the continuation

value function and the corresponding control maps. The left panel compares the continuation value function of GP-2D and PR-2D at the last step before maturity $t = T - \Delta t$. We observe that PR-2D has a poor fit compared to GP-2D, which almost perfectly matches the "hockey-stick" penalty. Furthermore, -1D regressions have "non-smooth" switching boundaries due to the piecewise regressions in $I$. This is evident in the center panel of the figure, where the control map shows jumps at $I \in \{500, 1000, 1500\}$, the intermediate discretization levels of inventory (we used $M_I = 5$ with $\Delta I = 500$). This behaviour becomes less prominent when $M_I$ is increased. The right panel of the figure visualizes two control maps for PR-1D with LHS and conventional design. The effect of an oversized input domain (we used $[P_{min}, P_{max}] = [2, 20]$ which explores *too much*) for LHS is evident, as we notice that the Store region is much too wide in the latter variant. Thus, the controller does not benefit from withdrawing when prices are $P \in [7, 7.5]$, ultimately resulting in lower valuation. Overall, Figure 4.4 highlights the need to properly pick both $\mathcal{H}$ and $\mathcal{D}$ to obtain good performance.

The effect of design shape can be conveniently visualized with Gaussian process emulators which provide a proxy for local estimation standard error through the posterior GP standard deviation $s(x)$. Figure 4.5 compares $s(x)$ that results from Mixture (we used a 60/40 mix of $\mathcal{S}_2$ and $\mathcal{P}_2$) and Space-filling (Sobol QMC $\mathcal{S}_2$) 2D designs of same size $M = 500$. For a space-filling design we observe a constant posterior variance, i.e. GPR learns $\hat{\mathcal{C}}_n(\cdot)$ equally well across the interior of the regression domain. At the same time, the posterior standard deviation is very high around the edges of $[P_{\min}, P_{\max}] \times [I_{\min}, I_{\max}]$, which is problematic for correctly identifying the strategy when inventory is almost full or almost empty. For the Mixture design, the posterior variance reflects the concentration of the input sites along the diagonal, compare to the joint probabilistic design $\mathcal{P}_2$ in Figure 4.3c. In turn, this is beneficial for learning the control map, as the GPR prediction is most accurate along the switching boundaries,

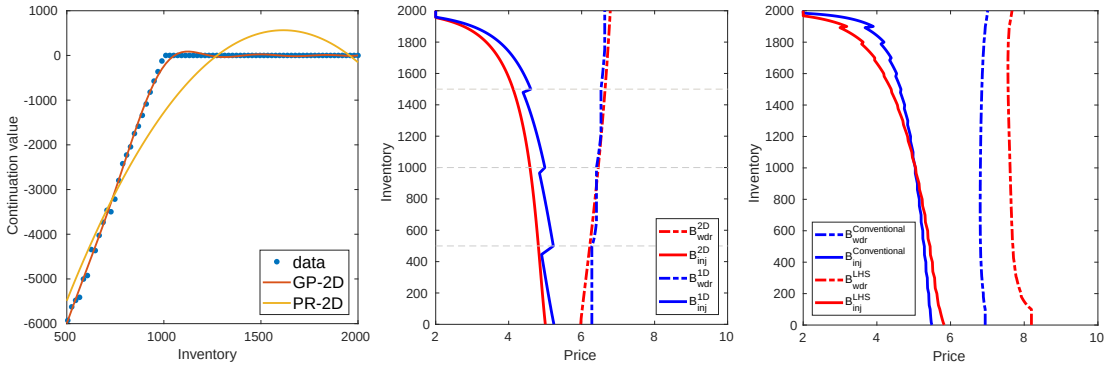



Figure 4.4: *Left panel:* Comparison of the continuation function $\mathcal{C}(t, P, \cdot)$ for different regressions at one step to maturity $t = T - \Delta t$ and $P = 6$. *Center panel:* the control maps $\widehat{m}(t, P, I)$ at $t = 2.7$ years corresponding to GP-1D and GP-2D regressions. For GP-1D we used $M_s = 100$, $M_b = 20$ and $M_I = 5$. For GP-2D we used $M_s = 500$ and $M_b = 20$. The horizontal lines represent the inventory discretization levels for the GP-1D design. *Right panel:* the control maps corresponding to conventional and LHS design with PR-1D regressions. We used $M_P = 2000$, $M_I = 21$ and price domain for LHS $[P_{min}, P_{max}] = [2, 20]$.

cf. Figure 4.2a. Higher precision around the switching boundaries allows the Mixture design to allocate the simulation budget to the regions where accuracy is needed most. We emphasize that such local inference quality is only available with non-parametric tools; global schemes like PR cannot directly benefit from focused designs.

## 4.6 Natural Gas Storage Facility

To illustrate DEA and its various ingredients, we present several numerical studies. In this section, we consider the gas storage problem which has been already explored in existing literature and hence affords a good testbed for comparison. Our main aims are to: (i) illustrate multiple implementations of DEA in terms of picking the regression spaces $\mathcal{H}$ and designs $\mathcal{D}$; (ii) explain the effect of the DEA modules on the ultimate problem solution, e.g. on the control maps; (iii) document the relative performance of

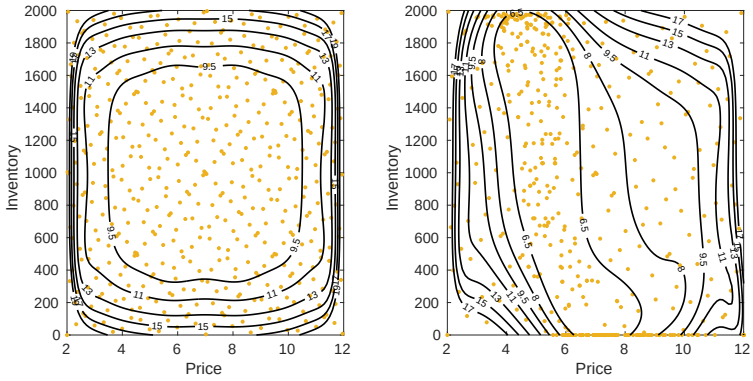



Figure 4.5: Impact of design on the posterior standard error $s(x)$ of GPR at $t = 1.5$ years. We show the contours of $x \mapsto s(x)$ with dots indicating the underlying respective designs with $M = 500$ sites. *Left panel:* Space-filling design (Sobol QMC $\mathcal{S}_2$) of Figure 4.3a leads to rectangular level sets of $s(\cdot)$. *Right panel:* under a Mixture design, $s(\cdot)$ resembles the shape of $\mathcal{P}_2$ in Figure 4.3c.

the different schemes to draw some conclusions on their merit in this context. While we present quite a lot of numeric experiments, we stress that we do not seek to provide a rigorous benchmarking, but rather view these as case studies. To this end, we do not concentrate on one single setup, instead exploring several formulations, including with and without switching costs, and with varying dimensionality. Our main message is that there is a lot of gains to be unlocked from fine-tuning DEA through carefully selecting its ingredients, in particular by mixing-and-matching existing proposals. Thus, the advertised flexibility of the template indeed leads to superior performance through optimizing the regression and design aspects, and exploiting the synergies between the two.

### 4.6.1 Market and Storage Description

In this section we revisit the gas storage problem in the setting of Chen and Forsyth [29]. The exogenous state process $P$ follows logarithmic mean-reverting dynamics

$$P_{n+1} - P_n = \alpha(\underline{P} - P_n)\Delta t_n + \sigma P_n \, \xi_{n+1}, \tag{4.27}$$

where $P_t$ is the spot price per unit of natural gas and is quoted in "dollars per million of British thermal unit ($/MMBtu)". The inventory $I_t$ is quoted in million cubic feet (MMCf). Since roughly 1000MMBtu = 1MMcf, we multiply $P_t$ by $10^3$ when calculating revenue or profit.

The penalty function $W(P_N, I_N)$ at maturity is $W(P_N, I_N) = -2P_N \max(1000 - I_N, 0)$. Thus, the target inventory is $I_N = 1000$ or 50% capacity. There is no compensation for excess inventory and a strong penalty (at 200% of the market price) for being short. As a result, the value function at maturity has a non-smooth hockey-stick shape in $I$, with zero slope for $I_N > 1000$ and a slope of $-2P_N$ otherwise.

The withdrawal and injection rates are inventory-dependent and given by

$$c_{wdr}(I) = -k_1\sqrt{I} \quad \text{and} \quad c_{inj}(I) = k_2\sqrt{\frac{1}{I + k_3} - \frac{1}{k_4}}.$$

These functional specifications are derived from the physical hydrodynamics of the gas storage facility, see [56]. The resulting dynamics of the inventory is:

$$I_{n+1} = I_n + \begin{cases} u_{wdr}(I_n)\Delta t, & \text{if } m_n = -1 \quad \text{(withdrawal)}; \\ 0 & \text{if } m_n = 0; \\ (u_{inj}(I_n) - k_5)\Delta t, & \text{if } m_n = +1 \quad \text{(injection)}. \end{cases} \tag{4.28}$$

The constant $k_5$ measures the cost of injection which is represented as "gas lost". It leads to a profit gap between production and injection whereby no-action ($m^* = 0$) will be the optimal action if the price is "close" to the mean level, $P_t \approx \underline{P}$.

In the numerical experiments below we discretize $T$ into $N = 1000$ steps, so that $\Delta t = 0.001T$, the rest of the parameters are listed in Table 4.1. The switching costs are taken to be zero $K(i, j) \equiv 0 \forall i, j$. Absence of switching cost reduces the state variables in the continuation function $\mathcal{C}_n(\cdot)$ in (4.16) to $(P_n, I_{n+1})$. As a result, in this example

every time step requires one projection of the 2D value function.

| |
|---|
| $\alpha = 2.38,\ \sigma = 0.59,\ \underline{P} = 6$ |
| $k_1 = 2040.41,\ k_2 = 7.3 \cdot 10^5,\ k_3 = 500,\ k_4 = 2500,\ k_5 = 620.5$ |
| $I_{\max} = 2000$ MMcf, $T = 3$, $\Delta t = 0.003$, $r = 10\%$ |

Table 4.1: Parameters for the gas storage facility in Section 4.6.

### 4.6.2 Benchmarking Setup

To benchmark the performance, we compare to two schemes utilizing conventional product design $\mathcal{P} \times \mathcal{G}$ (with inventory uniformly discretized): degree-3 global polynomial approximation over $(P, I)$ (PR-2D) and piecewise continuous approximation (with degree-3 polynomial regressions in $P$ at each inventory level $I_k$, PR-1D). These can be viewed as a "classical" TvR scheme with a joint regression [5, 65], and the discretized-$I$ version as used by [22, 20, 6, 19]. Additionally, we implement five regression approaches (PR-1D, GP-1D, PR-2D, LOESS-2D and GP-2D) on several different designs, with simulation budgets: $M \simeq 10K, 40K, 100K$:

- Randomized space filling design implemented via LHS ($\mathcal{L}_1 \times \mathcal{G}$ for piecewise-continuous regression and $\mathcal{L}_2$ otherwise). We use a large conservative input domain $P \in [2, 10]$.

- Mixture-2D design with 40% of sites from space-filling and the remaining 60% from the joint empirical distribution $\mathcal{P}_2$, $\mathcal{D}_M := \mathcal{P}_2(0.6M) \cup \mathcal{L}_2(0.4M)$. To implement $\mathcal{P}_2$, we need to estimate $\mathfrak{p}^P(t_n, \cdot)$ and $\mathfrak{p}^I(t_n, \cdot)$. This is done offline by first running the algorithm with a small budget and conventional product design. We then generate forward paths $(P_n^{m'}, \hat{I}_n^{m'})$ to estimate the joint $(\mathfrak{p}^P(t_n, \cdot), \hat{\mathfrak{p}}^I(t_n, \cdot))$ at $t_n$. Since the marginal distribution $\hat{\mathfrak{p}}^I(t_n, \cdot)$ starts to concentrate around $I = 1000$ as we get close to the maturity of the contract, the resolution at other parts of the domain is reduced and $\mathcal{L}_2$ is to used to compensate for this effect.

- Adaptive-1D: For 1D regressions, we estimate $\mathfrak{p}^P(t_n, \cdot)$ as above, and then non-uniformly discretize the inventory to incorporate the fact that the optimally controlled inventory process $\hat{I}_n$ concentrates around $I_{\min}, I_{\max}$ and $I = 1000$. Discretization levels for each simulation budget are detailed in Table 4.2.

- Dynamic time-dependent design that varies the simulation budget $M_n$ across $t_n$. We used the specification $M_n(M_{(1)}, M_{(2)}) := M_{(1)}\mathbf{1}_{\{n<900\}} + M_{(2)}\mathbf{1}_{\{n\geq 900\}}$ such that (approximately) $0.9M_{(1)} + 0.1M_{(2)} \in \{10K, 40K, 100K\}$. Exact specification is given in Table 4.3. We use two variants of it:

  - fixed projection space $\mathcal{H}$ (namely GP-1D) and conventional product design $\mathcal{D} = \mathcal{P} \times \mathcal{G}$.
  - time-dependent projection and designs, namely GP-2D and Mixture design for $n < 900$ and PR-1D and Conventional design for $n \geq 900$.

  The motivation for the above Dynamic scheme is to better handle the non-smooth terminal condition by devoting to it larger simulation budget, as well as using the -1D regression.

For GP we use `Matlab`'s in-built implementation `fitrgp`. For LOESS, we use the `curvefitting` toolbox again from `Matlab` that constructs local quadratic approximations based on the tri-cube weight function $\kappa(x_*, x^j) = \left(1 - (\frac{|x_* - x^j|}{\lambda(d, x_*)})^3\right)^3$. Above $\lambda(d, x_*)$ is the Euclidean distance from $x_*$ to the most distant $x^j$ within the span $d$. We use the default span of $d = 25\%$, keeping $P$ on its original scale and re-scaling $I$ to be in the range $[0, 2]$. To reduce the regression overhead of both GPR and LOESS we utilize batched designs with $M_b$ replicates (see Table 4.3) on top of the underlying design type.

In order to compare the performance of different designs/regressions, we use the estimate of the value function $\hat{V}_0(P_0, I_0)$ at $P_0 = 6, I_0 = 1000$ using a fixed set of $M' = 10,000$ out-of-sample paths (i.e. fixed $P^{m'}_{0:N}$) as a performance measure.

| $N_I$ | Specification |
|---|---|
| 11 | [0:100:200, 500:250:1500, 1800:100:2000] |
| 21 | [0:50:200, 400:200:800, 900:50:1100, 1200:200:1600, 1800:50:2000] |
| 31 | [0:25:100, 150, 200:100:900, 950:50:1050, 1100:100:1800, 1850, 1900:25:2000] |
| else | uniformly spaced |

Table 4.2: Discretized inventory levels used for Adaptive design for -1D methods.

| Design | Regression | Low | Medium | High |
|---|---|---|---|---|
| Conventional | PR-1D/-2D ($M_P \times M_I$) | $1050 \times 10$ | $2100 \times 20$ | $3400 \times 30$ |
| | GP-1D/-2D, LOESS ($M_s \times M_b \times M_I$) | $105 \times 10 \times 10$ | $210 \times 10 \times 20$ | $340 \times 10 \times 30$ |
| Space-filling | PR-1D ($M_P \times M_I$) | $1050 \times 10$ | $2100 \times 20$ | $3400 \times 30$ |
| | GP-1D ($M_s \times M_b \times M_I$) | $105 \times 10 \times 10$ | $210 \times 10 \times 20$ | $340 \times 10 \times 30$ |
| | PR-2D ($M$) | 10500 | 42000 | 102000 |
| | LOESS/GP-2D ($M_s \times M_b$) | $500 \times 21$ | $1000 \times 42$ | $2000 \times 51$ |
| Adaptive-1D | PR-1D ($M_P \times M_I$) | $950 \times 11$ | $2000 \times 21$ | $3300 \times 31$ |
| | GP-1D ($M_s \times M_b \times M_I$) | $95 \times 10 \times 11$ | $200 \times 10 \times 21$ | $330 \times 10 \times 31$ |
| Mixture-2D | PR-2D ($M$) | 10500 | 42000 | 102000 |
| | LOESS/GP-2D ($M_s \times M_b$) | $500 \times 21$ | $1000 \times 42$ | $2000 \times 51$ |
| Dynamic | GP-1D ($M_{(2)}$) | $150 \times 10 \times 21$ | $340 \times 10 \times 31$ | $440 \times 10 \times 41$ |
| | ($M_{(1)}$) | $74 \times 10 \times 11$ | $168 \times 10 \times 21$ | $300 \times 10 \times 31$ |
| | PR-1D + GP-2D ($M_{(2)}$) | $2000 \times 21$ | $3400 \times 31$ | $4400 \times 41$ |
| | ($M_{(1)}$) | $500 \times 21$ | $1000 \times 42$ | $2000 \times 51$ |

Table 4.3: Design construction for different methods in Table 4.4 of Section 4.6.1.

### 4.6.3 Results

Table 4.4 presents the performance of different designs and regression methods. We proceed to discuss the results focusing on three different aspects: (i) impact of different regression schemes, in particular parametric PR vs. non-parametric GP and LOESS approaches; (ii) impact of simulation design; (iii) joint -2D vs. interpolated -1D methods.

First, our results confirm that the interpolated -1D method performs extremely well in this classical example, perfectly exploiting the 2-dimensional setup with a 1-dimensional inventory variable. In that sense the existing state-of-the-art is already excellent. There are two important reasons for this. First PR-1D is highly flexible with lots degrees of freedom, allowing a good fit (with overfitting danger minimized due to a 1D setting). Second, PR-1D perfectly exploits the fact that the value function is almost

(piecewise) linear in inventory. We obtain a slight improvement by replacing PR-1D with GP-1D; another slight gain is picked up by replacing the equi-spaced inventory discretization with an adaptive approach that puts more levels close to the inventory boundaries. As a further enhancement, the Dynamic design utilizes a step-dependent simulation budget (to capture the boundary layer effect due to the non-smooth "hockey-stick" terminal condition that requires more effort to learn statistically), leading to significant improvement, highlighting the potential benefit of mixing-and-matching approximation strategies across time-steps. By taking $M$ time-dependent one may effectively save simulation budget (e.g. the valuation for Dynamic GP-1D with $M = 10^4$ is comparable to $M = 2 \cdot 10^4$ for Adaptive GP-1D). Nevertheless, as we repeatedly emphasize, the -1D methods do not scale well if more factors/inventory variables are added.

The much more generic bivariate -2D regressions give a statistically equitable treatment to all state variables and hence permit arbitrary simulation designs. The resulting huge scope for potential implementations is both a blessing and a curse. Thus, we document both some good and some bad choices in terms of picking a regression scheme, and picking a simulation design. We find that PR-2D tends to significantly underperform which is not surprising given that it enforces a strict parametric shape for the continuation value with insufficient room for flexibility. Similarly, we observe middling performance by LOESS; on the other hand GPR generally works very well.

Turning our attention to different 2D simulation designs, we compare the conventional $\mathcal{P} \times \mathcal{G}$ choice against 3 alternatives: conservative space-filling $\mathcal{L}_2$ on a large input domain; joint probabilistic design $\mathcal{P}_2$; a mixture design that blends the former two. We find that both plain space-filling and joint probabilistic do not work well; the first one is not targeted enough, spending too much budget on regions that make little contribution to $\hat{V}$; the second is too aggressive and often requires extrapolation produces inaccurate

predictions when computing $\widehat{m}$. In contrast, the mixture design is a winner, significantly improving upon the conventional one. In particular, GP-2D with Mixture design is the only bivariate regression scheme which performs neck-to-neck with GP-1D. This is significant because unlike GP-1D, GP-2D can be extended to higher dimensions in a straightforward manner and does not require a product design (or any interpolation which is generally slow). These findings highlight the importance of proposal density in design choice. To highlight the flexibility of our algorithm, we also present another dynamic design combining Adaptive PR-1D with Mixture GP-2D. This combination maintains the same accuracy but runs about 10-15% faster thanks to lower regression overhead of PR-1D.

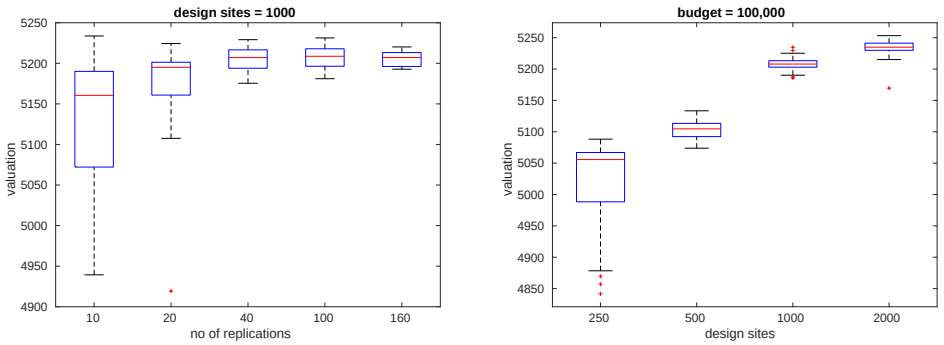
| Design | Regression Scheme | Simulation Budget Low | Medium | Large |
|---|---|---|---|---|
| Conventional | PR-1D | 4,965 | 5,097 | 5,231 |
| | GP-1D | 4,968 | 5,107 | 5,247 |
| | PR-2D | 4,869 | 4,888 | 4,891 |
| | LOESS-2D | 4,910 | 4,969 | 5,011 |
| | GP-2D | 4,652 | 5,161 | 5,243 |
| Space-filling | PR-1D | 4,768 | 4,889 | 5,028 |
| | GP-1D | 4,854 | 5,064 | 5,224 |
| | PR-2D | 4,762 | 4,789 | 4,792 |
| | LOESS-2D | 4,747 | 4,912 | 4,934 |
| | GP-2D | 4,976 | 5,080 | 5,133 |
| Adaptive 1D | PR-1D | 5,061 | 5,187 | 5,246 |
| | GP-1D | 5,079 | 5,195 | 5,245 |
| Dynamic | GP-1D | 5,132 | 5,225 | 5,266 |
| | Mixed | 5,137 | 5,205 | 5,228 |
| Mixture 2D | PR-2D | 4,820 | 4,835 | 4,834 |
| | LOESS-2D | 4,960 | 4,987 | 5,003 |
| | GP-2D | 5,137 | 5,210 | 5,233 |

Table 4.4: Valuation $\hat{V}_0(6, 1000)$ (in thousands) using different design-regression pairs and three simulation budgets: Low $M \simeq 10K$, Medium $M \simeq 40K$, Large $M \simeq 100K$, (cf. Table 4.3). The valuations are averages across 10 runs of each scheme except for LOESS-2D with large budget: due to the excessive overhead of LOESS only a single run was carried out.

**Simulation Designs for -1D Methods.** In the context of -1D regression methods that regress on $P$ only and discretize + interpolate in $I$, the design needs to be of product type. We find that a probabilistic $\mathcal{P} \times \mathcal{G}$ consistently outperforms a space-filling design, such as $\mathcal{L} \times \mathcal{G}$. We can also compare the performance of Conventional PR-1D and Adaptive PR-1D (Table 4.4) designs that share the same $\mathcal{P}$-design in $P$ based on the marginal distribution of $P_t$, but utilize different approaches for inventory discretization. The non-uniform discretization in the Adaptive version improves precision close to $I_{\min}$, $I_{\max}$ and $I = 1000$ and leads to a higher valuation relative to Conventional, Our take-away is that for the inventory discretization approach, one should concentrate on fine-tuning the $I$-mesh.

**Replicated Designs with GPR.** Implementation of GPR also requires to manage the tradeoff between the number of design sites $M_s$ and the replication amount $M_b$. The use of replication is necessitated since having more than $M_s > 2000$ distinct sites is significantly time consuming, at least for the off-shelf-implementation of GPR we used. In the left pane of Figure 4.6a we consider fixing $M_s$ and varying $M_b$ (hence $M$). While larger simulation budgets obviously improve results, we note that eventually increasing $M_b$ with $M_s$ fixed does not improve the regression quality (although it still reduces standard error). In the right panel of Figure 4.6b we present the impact of $M_s$ for fixed total budget $M = M_b \cdot M_s$. We find that replication in general sub-optimal and better results are possible when $M_s$ is larger (i.e. $M_b$ is smaller). To manage the resulting speed/precision trade-off, we recommend taking $M_b \in [20, 50]$; for instance when $M = 10^5$ we use $M_s = 2000, M_b = 50$ and when $M = 10^4$ we use $M_s = 500, M_b = 20$.

**Take-Aways:** Our experiments suggest the following key observations: (i) Among the inventory-discretized -1D methods, Gaussian process regression outperforms the standard polynomial regression in all cases, and is more robust to "poor" design or




(a) Impact of replication $M_b$ (b) Impact of distinct design sites $M_s$

Figure 4.6: *Left panel:* performance of Mixture GP-2D as a function of $M$. We fix the number of unique design sites $M_s = 1000$ and progressively increase the number of replicates $M_b$, reporting the resulting $\hat{V}_0(P, I)$ at $P = 6, I = 1000$. *Right:* effect of increasing $M_s$ for fixed $M = 10^5$. Results are for 20 runs of each algorithm. In each boxplot, the central mark represents the median, the box indicates the 25th and 75th percentiles, and the whiskers represent the most extreme runs.

low simulation budget. (ii) Within joint -2D schemes, we continue to observe superior performance of GPR compared to the alternate bivariate regression schemes (LOESS and PR). Moreover, GP-2D is neck-in-neck with the best-performing GP-1D. We emphasize that efficient implementation of GPR and LOESS relies on batched designs which is another innovation in our DEA implementations. (iii) We also find strong dependence between choice of design and performance. We confirm that best results come from designs, such as the Mixture and Dynamic versions we implemented, that balance filling the input space and targeting the domain where most of $(P_t, \hat{I}_t)$ trajectories lie. Otherwise, plain space-filling or conversely aggressive boundary-following degrade performance. (iv) Between the two parametric methods PR-1D and PR-2D, we find PR-1D to significantly outperform PR-2D irrespective of the design and simulation budget, indicating the advantage of piecewise continuous regression.

### 4.6.4 Gas Storage Modelization with Switching Costs

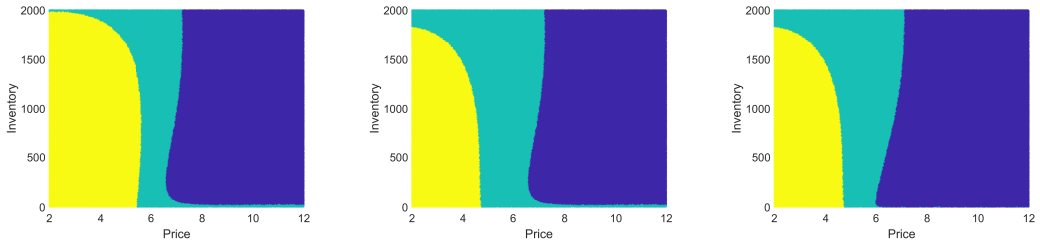
We generalize the previous example by incorporating switching costs. Switching costs make the control map depend on the current regime $m_n$ and induce inertia, i.e. preference to continue with the same regime so as to reduce overall costs. To handle the discrete $m$-dimension, we treat it as $|\mathcal{J}|$ distinct continuation functions, estimated through distinct regressions. Otherwise, the algorithm proceeds exactly the same way as before. This illustrates the flexibility of DEA to handle a range of problem formulations.

Besides the injection loss through $k_5$, we also add switching cost $K(i,j)$ with the following specification:

$$K(-1,1) = K(0,1) = 15000; \quad K(1,-1) = K(0,-1) = 5000; \quad K(1,0) = K(-1,0) = 0, \tag{4.29}$$

i.e. switching cost depends only on the regime the controller decides to switch to, with switching to injection the costliest and switching to no-action free. In Figure 4.7 we present the policy of the controller for different regimes. The inertia of being in regime $m_n = 0$ is evident as the corresponding control map has the widest Store region. Effect of $K(i,j)$ is also evident when comparing the Store region of left and center panels. If the controller moves from Inject to Store regime, she finds more resistance while trying to move back to injection due to the switching cost.

In Table 4.5 we present the performance of different design-regression pairs with a $M = 40K$ simulation budget. We dropped LOESS from this and the following case study to simplify the exposition and also because Matlab's implementation of LOESS does not support $d > 2$. As expected, introduction of the switching costs leads to lower valuation relative to Table 4.4. Moreover, the relative behaviour remains similar to the previous section i.e. space-filling design has the worst performance, Mixture design observes significant improvement, but Dynamic design finally wins the race.




(a) $m = +1$ (Inject) (b) $m = 0$ (Store) (c) $m = -1$ (Withdraw)

Figure 4.7: The control maps $\widehat{m}(t, P, I, m)$ at $t = 2.7$ years for the model with switching costs in (4.29), for $m \in \{+1, 0, -1\}$. The colors are $\widehat{m}_{t+\Delta t} = +1$ (inject, light yellow), $\widehat{m}_{t+\Delta t} = 0$ (store, medium cyan), $\widehat{m}_{t+\Delta t} = -1$ (withdraw, dark blue). The solution used GP-2D regression with Mixture design.

By comparing the valuation in Table 4.5 with Table 4.4 we may infer the impact and number of the switching costs. For example, previously Dynamic GP-1D produced valuation of $\hat{V}_0(6, 1000) = \$5,266$ (in thousands), however, with switching cost it is now $\$5,102K$. The respective loss of $\$166K$ can be interpreted as approximately 16 regime switches on a typical trajectory (assuming $\$10K$ as an average switching cost).

| Design | Regression | $\hat{V}_0(P_0, I_0)$ ('000s) |
|---|---|---|
| Conventional | PR-1D | 4,901 (12) |
| | PR-2D | 4,654 (11) |
| Space-filling | PR-1D | 4,663 (16) |
| | GP-1D | 4,757 (10) |
| | PR-2D | 4,594 (13) |
| | GP-2D | 4,879 (22) |
| Adaptive-1D | PR-1D | 4,978 (14) |
| | GP-1D | 5,058 (12) |
| Dynamic | PR-1D | 4,997 (17) |
| | GP-1D | 5,102 (20) |
| Mixture 2D | PR-2D | 4,602 (30) |
| | GP-2D | 4,978 ( 8) |

Table 4.5: Valuation of a gas storage facility with switching costs $\hat{V}_0(6, 1000)$ of different design-regression pairs with simulation budget of $M = 40,000$. Results are averages (standard deviations in brackets) across 10 runs of each algorithm.

Table 4.5 confirms the superior performance of GPR relative to PR, and the gains from using a Mixture (or even better a Dynamic) simulation design compared to the Conventional one. We remind the reader that implementing DEA in this setup is identical to the case where $K(i,j) \equiv 0$, except that a separate approximation $\hat{h}(\cdot, m)$ is constructed for each of the three levels of $m \in \mathcal{J}$. Thus, DEA immediately incorporates this extension and it is not surprising that the findings in Tables 4.4 and 4.5 are consistent with each other.

### 4.6.5 3D Test Case with Two Facilities

An important motivation for our work has been algorithm scalability in terms of the input dimension. In the classical storage problem the dimensionality is two: price $P$ and inventory $I$. However, in many contexts there might be multiple stochastic factors (e.g. the power demand and supply processes in the microgrid example below) or multiple inventories. The respective problem would then be conceptually identical to those considered, except that $\mathbf{X}$ has $d \geq 3$ dimensions.

Taking up such problems requires the numerical approach to be agnostic to the dimensionality. In terms of existing methods, the piecewise continuous strategy (such as PR-1D) has been the most successful, but it relies critically on interpolating in the single inventory variable. In contrast, joint polynomial regression is trivially scalable in $d$ but typically performs poorly. Thus, there is a strong need for other joint-$d$ methods that can improve upon PR.

In this section we illustrate the performance of DEA with a three-dimensional state variable. To do so, we consider a joint model of two gas storage facilities, which leads to three state space variables: price $P_t$, inventory of first storage $I_t^1$ and inventory of second storage $I_t^2$. Each storage facility is independently controlled and operated (so that there are 9 possible regimes $m_t \in \{-1, 0, 1\} \times \{-1, 0, 1\}$. Furthermore, the two

facilities have identical operating characteristics, each matching those of Section 4.6.1; it follows that the total value of two such caverns is simply twice the value of a single cavern.

To implement DEA in this setup we employ a direct analogue of the previous 2D problem. Namely, we use PR and GP regressions, together with Mixture and space-filling designs, highlighting the scalability of these choices. Two secondary changes are made: (i) For the space-filling design we switch to a 3D Sobol sequence; while we do not observe any significant difference in performance between LHS and Sobol sequences for 2D problems, in 3D LHS is less stable (higher variability of $\hat{V}_0(6, 1000, 1000)$ across runs) and yields estimates that are about \$80K-100K worse than from Sobol designs; (ii) for the mixing weights we take this time 50%/50% of sites from space-filling and from empirical distribution i.e. $\mathcal{D} = \mathcal{P}_3(0.5M) \cup \mathcal{S}_3(0.5M)$. The reason for both modifications is due to lower density of design sites per unit volume compared to previous examples , i.e. effectively lower budget. Adequate space-filling, best achieved via a QMC design, is needed to explore the relevant input space and fully learn the shape of the 3D continuation function.

In Figure 4.8, we present the performance of PR-3D and GP-3D for Mixture and space-filling designs. In addition, we also implement PR-1D with conventional design, meaning that we generate a probabilistic design in $P$ and do a two-dimensional gridded discretization (plus linear interpolation) in $I^1, I^2$. This approach reduces to doing $M_I^2$ one-dimensional polynomial regressions in $P$ and forces to take quite low $M_I, M_P$ values to fulfill an overall simulation budget of $M = M_P \times M_I \times M_I$, see Table 4.6. In contrast, -3D methods can borrow information from all $M$ paths, drastically improving statistical efficiency. To ease the comparison, we report half of total value of the two facilities, which should ideally match the original values in Table 4.4. Not surprisingly, the 3D problem is harder, so for the same simulation budget the reported valuations

are lower. For example, at $10^4$ budget, Adaptive GP-3D obtains a valuation $\$251K$ below that of Mixture GP-2D; this gap declines to $\$109K$ as we increase the simulation budget to $M = 10^5$. Moreover, difference between the performance of Mixture and space-filling design is evident even with polynomial regression (Sobol PR vs. Mixture PR). Mixture design with GP-3D further improves the valuation; we observe difference of over $\$300K$ comparing Mixture PR and Mixture GP at $M = 10^5$ budget. Consistent with previous examples, we again find that the valuation from conventional PR-1D is between the valuation obtained via PR-3D and Mixture GP-3D. However, the significantly larger standard errors of PR-1D is a sign of deteriorating stability of inventory discretization in the increased dimension, and highlight the limited scope of that technique. The take-away is that the gains from fine-tuning the DEA components grow as problem complexity increases. The emulation paradigm further suggests that non-parametric/adaptive approaches will be best able to maximize performance in "hard" contexts.

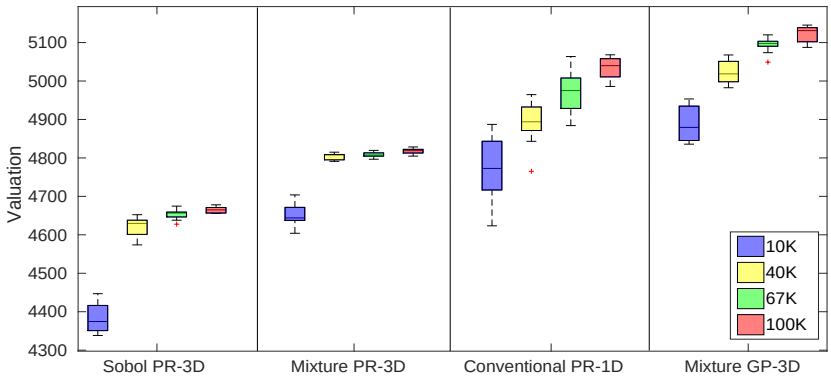



Figure 4.8: Half of estimated value $\hat{V}_0(6, 1000, 1000)/2$ for the 3D example with two storage caverns from Section 4.6.5. Results are for 10 runs of each algorithm across 4 different simulation budgets $M \in \{10K, 40K, 67K, 100K\}$. Description of the boxplots is same as in Figure 4.6.

| Design | Regression | 10k | 40k | 67k | 100k |
|---|---|---|---|---|---|
| Sobol/Adaptive | PR-3D ($N$) | 10500 | 42000 | 67500 | 102000 |
| | GP-3D ($N_s \times N_b$) | $500 \times 21$ | $1000 \times 42$ | $1500 \times 45$ | $2000 \times 51$ |
| Conventional | PR-1D ($N_P \times N_I^2$) | $215 \times 7^2$ | $347 \times 11^2$ | $400 \times 13^2$ | $450 \times 15^2$ |

Table 4.6: Design specifications for different DEA implementations of Section 4.6.5 in Figure 4.8.

## 4.7 Microgrid Balancing under Stochastic Net Demand

In this example, we use the framework of Section 4.2 in the context of a Microgrid, which is a scaled-down version of a power grid comprising of renewable energy sources, a diesel generator, and a battery for energy storage. The microgrid could be isolated or connected to a national grid and the objective is to supply electricity at the lowest cost by efficiently utilizing the diesel generator and the battery to match demand given the intermittent power production from the renewable sources. The topology of the microgrid considered here is similar to that in [11] and presented in Figure 1.1a.

The exogenous factor corresponds to net demand $L_n = D_n - R_n$, where $D_n$, $R_n$ are the demand and output from the renewable source, respectively. We assume that the microgrid controller bases his policy only on $L$, modeled as a discrete Ornstein-Uhlenbeck process:

$$L_{n+1} - L_n = \alpha(\underline{L} - L_n)\Delta t_n + \sigma\xi_n, \qquad \Delta\xi_n \sim \mathcal{N}(0, \Delta t_n). \tag{4.30}$$

On the supply side, the controller has two resources: a battery (energy storage) and a diesel generator. The state of the battery is denoted by $I_t \in [0, I_{\max}]$ with dynamics given by

$$I_{n+1} = I_n + a(u_n)\Delta t_n = I_n + B_n\Delta t_n, \tag{4.31}$$

where $a(u_n)$ is interpreted as battery output, driven by $u_n$ the diesel output. The latter has two regimes $m_n \in \{0, 1\}$. In the OFF regime $m_{n+1} = 0$, $u_n(0) = 0$. When the diesel is ON, $m_{n+1} = 1$, its power output is state dependent and given by:

$$u_n(1) = L_n \mathbf{1}_{\{L_n > 0\}} + B_{\max} \wedge \frac{I_{\max} - I_n}{\Delta t_n}, \tag{4.32}$$

where $B_{\max} > 0$ is the maximum power input to the battery. The power output from the battery is the difference between the net demand and the diesel output, provided it remains within the physical capacity constraints $[0, I_{\max}]$ of the battery:

$$B_n := a(u_n) = -\frac{I_n}{\Delta t_n} \vee \big(B_{\min} \vee (u_n - L_n) \wedge B_{\max}\big) \wedge \frac{I_{\max} - I_n}{\Delta t_n}. \tag{4.33}$$

To describe the cost structure, define an imbalance process $S_t = S(c_t, X_t)$:

$$S_n = u_n - L_n - B_n. \tag{4.34}$$

Normally the imbalance is zero, i.e. the battery absorbs the difference between production and demand. $S_n < 0$ implies insufficient supply of power resulting in a *blackout*; $S_n > 0$ leads to curtailment or waste of energy. We penalize both scenarios asymmetrically using costs $C_{1,2}$ for curtailment and blackout, taking $C_2 \gg C_1$ in order to target zero blackouts:

$$\pi(u, L) := -u^\gamma - |S| \Big[ C_2 \mathbf{1}_{\{S<0\}} + C_1 \mathbf{1}_{\{S>0\}} \Big]. \tag{4.35}$$

More discussion on the choice of this functional form can be found in [30, 11]. Furthermore, starting the diesel generator when it is OFF incurs a switching cost $K(0, 1) = 10$, but no cost is incurred to switch off the diesel generator, $K(1, 0) = 0$. The final optimization problem is starting from state $(L_n, I_n, m_n)$ and observing the net demand process $\mathbf{L}_n$, to maximize the pathwise value following the policy $\mathbf{m}_n$, exactly like in

(4.6).

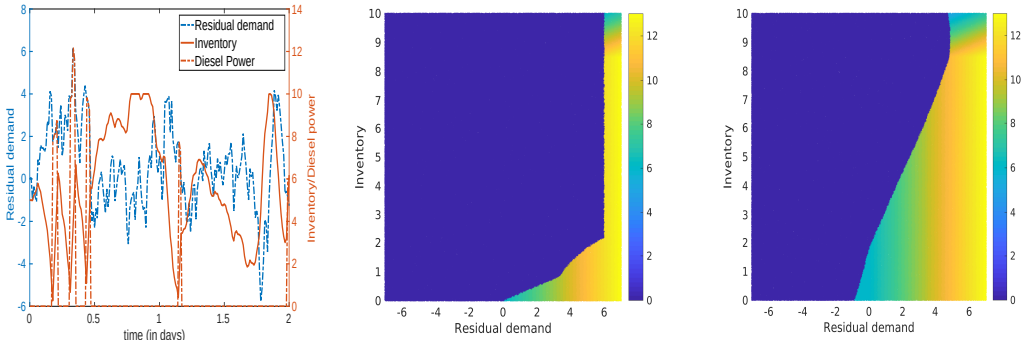
### 4.7.1 Optimal Microgrid Control

The parameters we use are given in Table 4.7. For the terminal condition we again force the controller to return the microgrid with at least the initial battery charge: $W(X_N, I_N) = -200 \max(I_0 - I_N, 0)$. The effect of this penalty is different compared to the gas storage problem in Section 4.6. Because the controller can only partially control the inventory, we end up with $\hat{I}_N \in [I_0, I_{\max}]$. We use simulation budget of $M = 10,000$ and out-of-sample budget of $M' = 200,000$. For simplicity of implementation, we use same simulation designs across both $m$-regimes (i.e. $\mathcal{D}_n$ is independent of $m$, cf. Algorithm 5).

| |
|---|
| $\alpha = 0.5,\ \underline{L} = 0,\ \sigma = 2$ |
| $I_{\max} = 10$ (kWh), $B_{\min} = -6, B_{\max} = 6$ (kW), $K(0,1) = 10$, $K(1,0) = 0$ |
| $C_1 = 5, C_2 = 10^6$, $\gamma = 0.9$, $T = 48$ (hours), $\Delta t = 0.25$ (hours) |

Table 4.7: Parameters for the Microgrid in Section 4.7.

Figure 4.9a illustrates the computed policy $(\widehat{m}_t)$ of the microgrid controller for a given path of net demand $(L_t)$. The left panel plots the joint trajectory of demand $L_{0:T}$ (left $y$-axis), inventory $\hat{I}_{0:T}$ and diesel output $u_{0:T}$ (both right $y$-axis). The diesel is generally off; the controller starts the diesel generator whenever the net demand $L_n$ is large, or the inventory $I_n$ is close to empty. When the generator is on, the battery gets quickly re-charged according to (4.32); otherwise $\hat{I}$ tends to be decreasing, unless $L_n < 0$. The center and right panels of the Figure visualize the resulting policy $u(t, L, I, m) = u(\widehat{m}(t, L, I, m))$. Due to the effect of the switching cost, when the generator is ON (Figure 4.9c), it continues to remain ON within a much larger region of the state space compared to when it is OFF.




(a) Pathwise trajectory (b) Generator OFF ($m = 0$) (c) Generator ON ($m = 1$)

Figure 4.9: *Left panel:* trajectory of the net demand ($L_t$), corresponding to policy ($u_t$) and the resultant inventory trajectory ($\hat{I}_t$). *Middle and right panels:* the control policy $\hat{u}(t, L, I, m)$ at $t = 24$ hours. Recall that $u(0) = 0$ whenever the diesel is OFF. All panels are based on GP-2D regression and Mixture design $\mathcal{D} = \mathcal{P}_2(0.5N) \cup \mathcal{L}_2(0.5N)$.

### 4.7.2 Numerical Results

Figure 4.10 shows the estimated value $\hat{V}_0(0, 5, 1)$ of the microgrid across different designs and regression methods at $M = 10,000$. Recall that in this setup, the controller only incurs costs so that $\hat{V} < 0$ and smaller (costs) is better. The relative performance of the schemes remains similar to Section 4.6. We continue to observe lower performance of space filling designs across regression methods. However, GP is more robust to this design change compared to traditional regression methods. Moreover, GPR dramatically improves upon PR-2D (whose performance is so bad it was left off Figure 4.10). Adaptive design with GP-2D once again produces the highest valuation (lowest cost), and substantially improves upon PR-1D.

## 4.8 Summary

The developed DEA template generalizes the existing methodologies used in the sphere of RMC methods for stochastic storage problems. The modularity of DEA al-

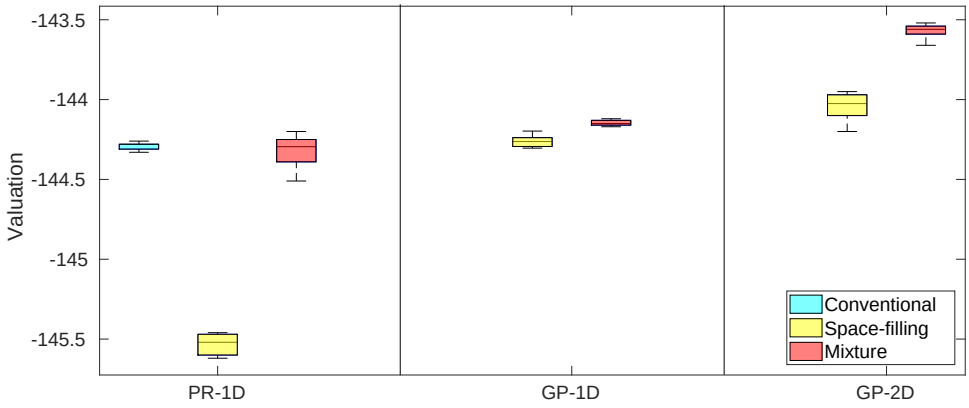



Figure 4.10: Estimated value $\hat{V}_0(0, 5, 1)$ for the microgrid example and different design-regression pairs. Results are for 10 runs of each algorithm. For comparison, PR-2D estimated mean valuation was significantly lower at $-153.6$, $-156.4$, $-152.3$ for Conventional, Space-filling and Mixture designs, respectively. Description of the boxplots is the same as in Figure 4.6.

lows a wide range of modifications that can enhance current state-of-the-art and improve scalability. In particular, we show several combinations of approximation spaces and simulation designs that are as good or better than any benchmarks (using both -1D and -2D approaches) reported in the literature. Emphasizing the experimental design aspect we show that there is wide latitude in removing memory requirements of traditional RMC by eliminating the need to simulate global paths. Similarly, non-parametric regression approaches like GPR, minimize the concern of picking "correct" basis functions. Furthermore, we stress the possibility to mix and match different methods. As an example, we illustrated DEA-based valuation of a gas storage facility using different designs, regressions, and budgets across the time-steps.

A natural extension is to consider the setting where the injection/withdrawal rates are continuous. In that case, one must optimize over $u_t$, replacing the arg max operator with a bonified arg sup over the admissible control set $u \in \mathcal{U}$. Depending on how switching costs are assessed, one might eliminate the regime $m_t$ entirely, or have a

double optimization

$$V(t, P_t, I_t, m_t) = \max_{m \in \mathcal{J}} \Big\{ \sup_{u \in \mathcal{U}(P_t, I_t, m)} \pi(P_t, u)\Delta t + \mathcal{C}(t, P_t, I_{t+\Delta t}(u), m) - K(m_t, m) \Big\}.$$

This could be interpreted as solving a no-bang-bang switching model, which would arise when there is some nonlinear link between profit $\pi$ and $u$, or a nonlinear effect of $u$ on $I_{t+\Delta t}$. Numerically carrying out the inner optimization over $u$ calls for joint regression schemes in order that $\hat{\mathcal{C}}(t, \cdot)$ is smooth in $I_{t+\Delta t}$.

A further direction afforded by our template is to move to look-ahead strategies using Remark 9 for generation of pathwise continuation values. By taking $w > 1$ one may interpolate between the Tsitsiklis-van Roy approach and the Longstaff-Schwartz one, ideally via a data-driven scheme that adaptively selects the look-ahead at each time step. To this end, GP regressions results can be used to quantify the single-step projection errors in $\hat{\mathcal{C}}_n(\cdot)$.

In a different vein, one may consider modifications where the autonomous/endogenous dichotomy between $(P_t)$ and $(I_t)$ is blurred. For example, a variant of the storage problem arises in the context of hydropower operations, i.e. controlling a double dammed reservoir that receives inflows from upstream and can release water downstream [38, 5]. Moreover, the reservoir experiences evaporation and/or natural drawdown, irrespective of the operations. In this setup, the inventory $I_t$ experiences stochastic shocks, either due to random inflows (due to precipitation) or random outflows (due to temperature-based evaporation, etc). Therefore, $I_{t+1}$ is a function of both $I_t, u_t$, and some outside noise (or factor) $O_t$. Moreover, if the dam is large, hydropower management has endogenous stochastic risk, i.e. the control $u_n$ also affects the distribution of the price process $P_{n+1}$ by modifying the regional supply of energy and hence affecting the supply-demand equilibrium that drives changes in $(P_t)$.

# Chapter 5

# Statistical Learning for Probability-Constrained Stochastic Optimal Control

*This chapter is the result of a collaboration with Alessandro Balata, Michael Ludkovski and Jan Palczewski. It is based on the work [13].*

We investigate Monte Carlo based algorithms for solving stochastic control problems with probabilistic constraints. Our motivation comes from microgrid management, where the controller tries to optimally dispatch a diesel generator while maintaining low probability of blackouts. The key question we investigate are empirical simulation procedures for learning the admissible control set that is specified implicitly through a probability constraint on the system state. We propose a variety of relevant statistical tools including logistic regression, Gaussian process regression, quantile regression and support vector machines, which we then incorporate into an overall Regression Monte Carlo (RMC) framework for approximate dynamic programming. Our results indicate that using logistic or Gaussian process regression to estimate the admissibility probability outperforms the other options. Our algorithms offer an efficient and reliable extension of RMC to probability-constrained control. We illustrate our findings with

two case studies for the microgrid problem.

## 5.1 Introduction

Stochastic control with probabilistic constraints is a natural relaxation of deterministic restrictions which tend to generate high costs forcing the avoidance of extreme events no matter their likelihood of occurrence. In contrast, with probabilistic constraints, constraint violation is tolerated up to a certain level offering a better trade-off between admissibility and cost. We refer to [66] for an overview of probability constrained problems and list below some of our motivating settings and references:

1. Microgrid management: In the context of microgrid control, since perfect balancing between fluctuating demand and supply is very expensive, it is common to allow for a small frequency of black-outs, i.e. occurrences where demand outstrips supply. A standard approach is to use mixed-integer linear programming by approximating the non-linear and non-convex probability constraints with more conservative convex constraints as in [67].

2. Hydro-power optimization: control of a hydro-power dam with probabilistic constraints was discussed in [38]. Within this setup, the controller observes random inflows from precipitation, as well as fluctuating electricity prices. Her objective is to control the downstream outflow from the dam to maximize profit from power sales, while ensuring a minimum dam capacity with high probability. Other related works are [39, 40].

3. Motion planning: finding the minimum-cost path for a robot from one location to another while avoiding colliding with objects that obstruct its path. Stochasticity in the environment implies that the robot motion is only partially controlled. Robust optimization that guarantees obstacle avoidance might be infeasible, making

probabilistic constraints a viable alternative. Dynamic programming methods for unmanned aerial vehicles were introduced in [41] and the probabilistic-constrained motion of a robot was solved in [42].

**Contribution.** In sum, in the stochastic context it is common and natural to impose probabilistic constraints. In contrast to deterministic constraints that are often simple to verify, probabilistic constraints are much harder to handle since admissibility of the control can generally only be estimated. Therefore, a numerical procedure to learn which actions are admissible is necessary *in addition* to the core optimization routine. In this chapter, we consider continuous-state, continuous-time models on infinite probability spaces. Therefore, probability constraints become a local expectation constraint at each system state. The canonical setup involves finite-horizon control of a stochastic process described through a stochastic differential equation of Itô type. The overarching solution paradigm involves the Bellman or Dynamic Programming equation, which works with discretized time-steps but with a smooth spatial variable. In this context, we develop algorithms to solve stochastic optimal control problems with probabilistic constraints using RMC. To make this highly nontrivial extension to RMC, we investigate tools from statistics and machine learning (including support vector machines (SVM), Gaussian process (GP) regression, parametric density estimation, logistic regression and quantile regression) to estimate the admissible set corresponding to the probability constraint and test them for a practical problem of energy management. Our algorithm allows us to estimate the two parts of the problem —the constraint and the approximation of the conditional expectation—in parallel and with significantly lower simulation budget compared to a *naive* implementation.

After proposing several approaches and benchmarking them on two case-studies, our main finding is to recommend logistic regression and GP-smoothed probability estimation as the best procedures. These methods are stable, relatively fast and allow for

a variety of further adjustments and speed-ups. In contrast, in our experience despite theoretical appeal, quantile regression and SVM are not well-suited for this task. On a higher level, our main take-away is that stochastic control with probabilistic constraints (SCPC) is well within reach of cutting-edge RMC methods. Thus, it is now computationally feasible to tackle such problems, opening the door for new SCPC models and applications.

**Solutions in literature.** Mixed-integer linear programming (MILP) is the standard tool used to solve SCPC (see [68] for an overview), however, there are several reasons why RMC or approximate dynamic programming methods may be a better choice. First, unlike MILP, RMC does not require any discretization of the state space, neither does it require linearizing the constraints. Approximating non-linear constraints by linearizing them can significantly affect the quality of the solution. Second, an important advantage of RMC is its ability to find optimal control dynamically for each time step and every state. This differs from MILP methods where the entire problem needs to be solved again for a new state. Third, MILP suffers from severe time-complexity constraints as the time horizon increases, RMC, on the other hand, has linear time complexity with respect to the horizon. In a recent work [69], the authors also find that the approximate dynamic programming methods like RMC have better solution quality and better runtime as the horizon of the problem increases.

A dual dynamic programming based approach for SCPC has been discussed in [70, 38]. The central idea is to incorporate the constraint in the objective function via Lagrange multiplier and iteratively solve for the optimal control and Lagrange multiplier. Although it is a popular approach, the final solution is sub-optimal due to the duality gap.

Another approach to SCPC is the *stochastic viability* framework for multi-period constraints developed in [38, 71]. In these works, the goal is to maximize the probability

of being admissible, which is defined both in terms of profit targets and satisfying constraints at every time step. Local probabilistic constraints of the type discussed in this chapter have been recently also studied in [72] to compute hedging price of a portfolio whose risk is defined in terms of its future value with respect to a set of stochastic benchmarks. Besides a local probabilistic constraint, authors also provide dynamic programming equations for multi-period constraints. However, their solution is driven by very specific loss functions and state processes. In contrast, we develop general purpose numerical schemes using statistical learning methods.

## 5.2 Problem formulation

We study numerical resolution of stochastic control problems on finite horizon $[0, T]$ with local implicit constraints, specifically we work with constraints defined through probabilistic conditions on the controlled state. The general formulation of the stochastic control problem we are interested in this chapter is of the form:

$$V_n(\mathbf{X}_n) = \inf_{(u_s)_{s=n}^N \in \mathcal{U}_{n:N}(\mathbf{X}_n)} \Big\{ \mathbb{E}\Big[ \sum_{k=n}^{N-1} \int_{t_k}^{t_{k+1}} \pi_s(\mathbf{X}(s), u_k) \mathrm{d}s + K(\mathbf{X}_n, u_n) + W(\mathbf{X}(t_N)) \Big| \mathbf{X}_n \Big] \Big\}, \tag{5.1}$$

where $W(\cdot)$ represents the terminal penalty, $\pi_t(\cdot, \cdot)$ the running cost, $K(\cdot, \cdot)$ the switching cost that incurs only at discrete time epochs when the controls are chosen and

$$\mathcal{U}_{n:N}(\mathbf{X}_n) = \Big\{ (u_k)_{k=n}^N : P_k(\mathbf{X}_k, u_k) \in \mathcal{A}_k \ \forall k \in \{n, \ldots, N-1\} \Big\}, \tag{5.2}$$

with $P_k : \mathcal{X} \times \mathcal{W} \to \mathbb{R}$ and $A_k \subset \mathbb{R}$. The admissible set $\mathcal{U}$ restricts potential choices of controls given the current state $\mathbf{X}_n$. A key assumption is that admissibility is defined implicitly, i.e. *a priori* it is not clear which control choices satisfy constraints and which do not. Thus, the controller must carry out the optimization while simultaneously

*learning* the feasibility of proposed actions. In other words, the mapping $P_k(\cdot,\cdot)$ is only given implicitly and inverting it to obtain $\mathcal{U}_{n:N}(\mathcal{X}_n)$ is numerically nontrivial. We will assume in the following that an admissible control always exists at any state, so that we may define $\mathcal{U}_n(\mathbf{X}_n) = \mathcal{U}_{n:n}(\mathbf{X}_n)$ to be the set of admissible controls satisfying the constraints at a single decision epoch $t_n$ conditional on $\mathbf{X}_n$. The dynamic programming equation at step $n$ corresponding to equation (5.1) is:

$$\begin{aligned} V_n(\mathbf{X}_n) &= \inf_{u\in\mathcal{U}_n(\mathbf{X}_n)} \Big\{\mathcal{C}_n(\mathbf{X}_n,u)\Big\}, \\ \text{where} \quad \mathcal{C}_n(\mathbf{X}_n,u) &= \mathbb{E}\left[\pi^{\Delta}(\mathbf{X}_{k:k+1},u_k) + V_{n+1}(\mathbf{X}(t_{n+1}))\Big|\mathbf{X}_n,u\right], \\ \text{and} \quad \pi^{\Delta}(\mathbf{X}_{k:k+1},u_k) &= \int_{t_k}^{t_{k+1}} \pi_s(\mathbf{X}(s),u_k)\mathrm{d}s + K(\mathbf{X}_k,u_k). \end{aligned} \tag{5.3}$$

As before, $\mathcal{C}_n(\mathbf{X}_n,u)$ is the continuation value from using the control $u$ over $[t_n,t_{n+1})$. The admissible set $\mathcal{U}_n(\mathbf{X}_n)$ is both time and state dependent. Thus, we need to estimate the continuation value $\mathcal{C}_n(\cdot,\cdot)$ and the admissible control set $\mathcal{U}_n(\cdot)$ at every time step.

Through the rest of the Chapter we will assume $P_n(\mathbf{X}_n,u_n)$ and $\mathcal{A}_n$ in (5.2) to be

$$P_n(\mathbf{X}_n,u_n) \equiv p_n(\mathbf{X}_n,u_n) := \mathbb{P}\Big(\mathcal{G}((\mathbf{X}(s))_{s\in[t_n,t_{n+1})}))>0|\mathbf{X}_n,u_n\Big) \;\text{ and }\; \mathcal{A}_n := [0,p). \tag{5.4}$$

In other words, we target the set of controls such that the conditional probability of the functional $\mathcal{G}(\cdot)$ of $\mathbf{X}$ being greater than zero is bounded by a threshold $p$, i.e.

$$\mathcal{U}_n(\mathbf{X}_n) := \Big\{u\in\mathcal{W}:\; p_n(\mathbf{X}_n,u_n)<p\Big\}. \tag{5.5}$$

For simplicity of notation, we define $G_n(\mathbf{X}_n,u_n)$ as the regular conditional distribution of the functional $\mathcal{G}(\cdot)$ given $(\mathbf{X}_n,u_n)$:

$$G_n(\mathbf{X}_n,u_n) := \mathcal{L}\Big(\mathcal{G}((\mathbf{X}(s))_{s\in[t_n,t_{n+1})})\Big|\mathbf{X}_n,u_n\Big). \tag{5.6}$$

When writing $\mathbb{P}\big(G_n(\mathbf{X}_n, u_n) > z\big)$ or $\mathbb{E}\big[g\big(G_n(\mathbf{X}_n, u_n)\big)\big]$ we mean the probability or the expectation with respect to this conditional distribution. The parameter $p$ in equation (5.5) is interpreted as relaxing the strong constraint $\mathcal{G} \leq 0$ which may not be appropriate in a stochastic environment. The random variable $G_n(\mathbf{X}_n, u_n)$ quantifies the riskiness of the controlled trajectory, and the controller is required to keep the former below some pre-specified level, taken without loss of generality to be zero. Typical values of $p$ would generally be small.

**Remark 16** *We may rewrite* (5.4) *through the corresponding* $(1-p)^{th}$ *quantile* $q(\mathbf{X}_n, u_n)$ *of* $G_n(\mathbf{X}_n, u_n)$*:*

$$q_n(\mathbf{X}_n, u_n) : (\mathbf{X}_n, u_n) \mapsto arg\inf_z \Big\{\mathbb{P}\Big(G_n(\mathbf{X}_n, u_n) > z\Big) \leq p\Big\}. \tag{5.7}$$

*Then using*

$$\mathcal{U}_n(\mathbf{X}_n) := \Big\{u : p_n(\mathbf{X}_n, u) < p\Big\} = \Big\{u : q_n(\mathbf{X}_n, u) < 0\Big\}, \tag{5.8}$$

*we can set* $P_n' := q_n$ *and* $\tilde{\mathcal{A}} = (-\infty, 0)$ *in* (5.2)*. We will exploit this equivalence to propose quantile-based methods (Section 5.4) for the admissible set.*

**Remark 17** *Assuming a one dimensional control* $u_n \in \mathcal{W} \subset \mathbb{R}$*, and the probability* $p_n(\mathbf{X}_n, u_n)$ *monotonically decreasing in* $u_n$*, estimating the admissible set* $\mathcal{U}_n(\mathbf{X}_n)$ *is equivalent to estimating the* minimum *admissible control*

$$u_n^{\min}(\mathbf{X}_n) := \inf_{u \in \mathcal{W}} \Big\{u : p_n(\mathbf{X}_n, u) < p\Big\}.$$

*The corresponding admissible set will be* $\mathcal{U}_n(\mathbf{X}_n) = \{u \in \mathcal{W} : u \geq u_n^{\min}(\mathbf{X}_n)\}$.

**Remark 18** *A more general version are implicit constraints of the form*

$$\Big\{u \in \mathcal{W} : \mathbb{E}\Big[g\Big(G_n(\mathbf{X}_n, u)\Big)\Big] < p\Big\},$$

*for a function $g : \mathbb{R} \to \mathbb{R}$, where of course the probability constraint* (5.5) *above arises when $g$ is an indicator function. Also notice that in principle $\hat{\mathcal{C}}$ is not monotone in $u$, and hence the admissibility set $\mathcal{U}$ might affect the optimal control even when $u_n^*(\mathbf{x}) > u_n^{\min}(\mathbf{x})$.*

**Remark 19** *Equation* (5.8) *describes admissible controls $u$ for a given state $\mathbf{x}$. The "dual" perspective is to consider the set of states $\mathcal{X}_n^a(u) \subset \mathcal{X}$ for which a given control $u$ is admissible:*

$$\mathcal{X}_n^a(u) := \Big\{\mathbf{x} \in \mathcal{X} : p_n(\mathbf{x}, u) < p\Big\}. \tag{5.9}$$

*Often the cardinality of $\mathcal{X}$ is infinite, while the control space $\mathcal{W}$ is finite, so that enumerating* (5.9) *over $u \in \mathcal{W}$ is considerably easier than enumerating the uncountable family of sets $\mathbf{x} \mapsto \mathcal{U}_n(\mathbf{x})$ in equation* (5.5)*. Furthermore, if $u \mapsto p_n(\mathbf{x}, u)$ is decreasing for all $\mathbf{x} \in \mathcal{X}$, then we obtain an ordering $\mathcal{X}_n^a(u_1) \subseteq \mathcal{X}_n^a(u_2)$ for $u_1 \leq u_2$. The latter nesting feature greatly helps to estimate the various $\mathcal{X}_n^a$'s. In other words, frequently one may rank the controls in terms of their "riskiness" with respect to $G_n$, so that the safest control will have a very large $\mathcal{X}_n^a(u)$ (possibly all of $\mathcal{X}$), while the riskiest control will have a very small admissibility domain.*

**Remark 20** *Notice that the reward between time $[t_n, t_{n+1})$, $\int_{t_n}^{t_{n+1}} \pi_s(\mathbf{X}(s), u)\mathrm{d}s$ is random at time $t_n$. As a result, we incorporate it in the definition of our continuation value $\mathcal{C}_n(\mathbf{X}_n, u)$.*

### 5.2.1 Regression Monte Carlo

We continue to focus on simulation-based techniques to solve (5.1). As a result, we work with the approximate Dynamic Programming recursion

$$\begin{aligned} \hat{V}_n(\mathbf{X}_n) &= \inf_{u_n \in \hat{\mathcal{U}}_n(\mathbf{X}_n)} \Big\{ \hat{\mathcal{C}}_n(\mathbf{X}_n, u_n) \Big\}, \\ \text{where} \quad \hat{\mathcal{C}}_n(\mathbf{X}_n, u_n) &:= \hat{\mathbb{E}} \left[ \pi^{\Delta}(\mathbf{X}_{n:n+1}, u_n) + \hat{V}_{n+1}(\mathbf{X}(t_{n+1})) \Big| \mathbf{X}_n, u_n \right]. \end{aligned} \tag{5.10}$$

The set of admissible controls $\hat{\mathcal{U}}_n$ is approximated via either $\hat{p}_n(\cdot,\cdot)$, i.e., $\hat{\mathcal{U}}_n(\mathbf{X}_n) := \big\{u : \hat{p}_n(\mathbf{X}_n, u) < p\big\}$, or $\hat{q}_n(\cdot,\cdot)$, i.e., $\hat{\mathcal{U}}_n(\mathbf{X}_n) = \big\{u : \hat{q}_n(\mathbf{X}_n, u) < 0\big\}$, see (5.8).

Recall from Section 3.4 in Chapter 3 and Section 4.3 in Chapter 4 that specifying $\hat{\mathbb{E}}$ is equivalent to approximating the conditional expectation map

$$(\mathbf{x}, u) \mapsto \mathbb{E}[\psi\big((\mathbf{X}(s))_{s\in[t_n,t_{n+1}]}\big) | \mathbf{X}_n = \mathbf{x}, u_n = u] =: f(\mathbf{x}, u)$$

where we will specifically substitute

$$\psi\big((\mathbf{X}(s))_{s\in[t_n,t_{n+1}]}\big) = \int_{t_n}^{t_{n+1}} \pi_s(\mathbf{X}(s), u_n)\mathrm{d}s + K(\mathbf{X}_n, u_n) + \hat{V}_{n+1}(\mathbf{X}(t_{n+1})).$$

To do so, we consider a dataset consisting of inputs $(\mathbf{x}_n^1, u_n^1), \ldots, (\mathbf{x}_n^{M_c}, u_n^{M_c})$ and the corresponding pathwise realizations $y^1, \ldots, y^{M_c}$ with $y^j = \psi\big((\mathbf{x}(s))^j_{s\in[t_n,t_{n+1}]}\big)$, where $(\mathbf{x}(s))^j_{s\in[t_n,t_{n+1}]}$ is an independent draw from the distribution of process $(\mathbf{X}(s))_{s\in[t_n,t_{n+1}]} | (\mathbf{x}_n^j, u_n^j)$. Then we use the training set $\{\mathbf{x}_n^j, u_n^j, y^j\}_{j=1}^{M_c}$ to learn $\hat{f}$, an estimator of $f$, via regression. In contrast to Chapters 3 and 4, here we include the cumulative cost from $[t_n, t_{n+1})$ when approximating the conditional expectation map. Also, we project the realizations in the joint state-action space.

Similarly, estimating $\mathcal{U}_n$ is equivalent to learning the conditional probability map $p_n(\mathbf{x}, u)$ (or the conditional quantile map $q_n(\mathbf{x}, u)$ in (5.5)) and then comparing to the

threshold value $p$ (zero, respectively). This statistical task, whose marriage with RMC is our central contribution, is discussed in Section 5.4.

The technique of using regressions for the approximation of the continuation value is very well developed (see Chapters 3 and 4). In contrast to estimating continuation value function, we are not aware of any existing works to estimate the set of admissible controls $\hat{\mathcal{U}}_n(\mathbf{X}_n)$ which requires approximating $p(\mathbf{x}, u)$ (or $q(\mathbf{x}, u)$) in equation (5.5). A naive approach is to estimate $\hat{\mathcal{U}}_n(\mathbf{X}_n)$ for every state realized during the backward induction through nested Monte Carlo. Namely for each pair $(\mathbf{x}, u)$ encountered, we may estimate the probability of violating the constraint by simulating $M_b$ samples from the conditional distribution $G_n(\mathbf{x}, u)$ as $\{g_n^b(\mathbf{x}, u)\}_{b=1}^{M_b}$. We then set $u \in \hat{\mathcal{U}}_n(\mathbf{x})$ if $\bar{p}_n(\mathbf{x}, u) < p$, where

$$\bar{p}_n(\mathbf{x}, u) := \sum_{b=1}^{M_b} \frac{\mathbf{1}_{g_n^b(\mathbf{x},u)>0}}{M_b} \tag{5.11}$$

is the empirical probability. Although extremely easy to implement, this Nested Monte Carlo (NMC) method is computationally intractable for even the easiest problems. As an example, a typical RMC scheme employs $M_c \approx 100,000$ and assuming $M_b = 1000$ for inner simulations, which is necessary for good estimates of small probabilities $p \leq 0.1$, would require $10^8$ simulation budget at every time-step to implement NMC. Note furthermore that NMC returns only the local estimates $\bar{p}(\mathbf{x}, u)$; no *functional* estimate of $\mathcal{U}_n(\mathbf{x})$ or $\mathcal{X}_n^a(u)$ is provided for an arbitrary $\mathbf{x}$. As a result, any out-of-sample evaluation (i.e. on a future sample path of $\mathbf{X}$.) requires further inner simulations, making this implementation even more computationally prohibitive.

An important challenge in using $\hat{\mathcal{U}}$ is verifying admissibility. Since we are employing random Monte Carlo samples to decide whether $u$ is admissible at $\mathbf{x}$, this is a probabilistic statement and admissibility can never be guaranteed. We may use statistical tools to quantify the accuracy of $\mathcal{U}$, for example, by applying classical Central Limit Theorem tools for the estimator $\bar{p}(\mathbf{x}, u)$ of the true $p(\mathbf{x}, u)$. In particular, to provide

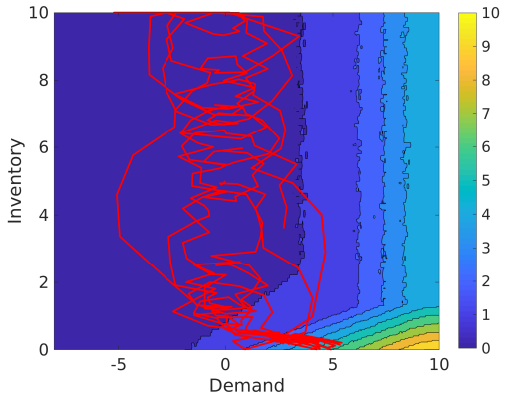



Figure 5.1: Contour plot for minimum admissible diesel output $(L, I, m) \mapsto u_n^{\min}(L, I, m)$ (see Remark 17). Generally for $L < 0$, the constraint is not binding and $u_n^{\min}(L, I, m) = 0$. As demand increases, the constraint becomes more stringent, i.e. $u_n^{\min}(L, I, m)$ increases in $L$. Red curve represents a path of the controlled demand-inventory pair $(L_n^{u^*}, I_n^{u^*}, m_n^{u^*})$ following a myopic strategy choosing the minimum admissible control $u_n(L_n, I_n, m_n) = u_n^{\min}(L_n, I_n, m_n)$.

better statistical guarantees regarding $\hat{\mathcal{U}}$ we develop statistical tools in order to make statements (with asymptotic guarantee) such as $u \in \mathcal{U}$ with 95% confidence (equivalent to $p(\mathbf{x}, u) < p$ with 95% probability). As we show, without such "conservative" estimates based on confidence levels, estimates of $\mathcal{U}$ might be highly unreliable, frequently causing decisions that are inadmissible with respect the imposed probability constraint. Thus, the related construction of $\hat{\mathcal{U}}^{(\rho)}$ with specified confidence level $\rho$ is a running theme in Section 5.4, where we propose several statistical methods.

### 5.2.2 Motivation: controlling blackout probability in a microgrid

To make our presentation concrete, in this section we go back to the microgrid management problem to illustrate the application of the framework (5.3). Recall our microgrid topology from figure 1.1a where the microgrid controller is in charge of the

diesel generator through the control $u(t)$. Simultaneously, she faces the constraint of avoiding blackouts, whereby demand is not met. We reiterate that the control decisions are made at discrete epochs $\{t_0, t_1, \ldots, t_{N-1}\}$, however these decisions affect the state of the system continuously. As a result, choosing the control $u(t_n) \equiv u_n$ at time $t_n$ involves minimizing the cost of running the microgrid, as well as controlling the probability of blackout (i.e. controller fails to match the net demand) at intermediate intervals $[t_n, t_{n+1})$. The blackout is described through the imbalance process $S(s) := L(s) - u_n - B(s)$, $\forall s \in [t_n, t_{n+1})$, representing the difference between the demand and supply, while the diesel output is held constant ("zero-order-hold") over the time step. The power output from the battery is a deterministic function of net-demand, inventory and the control $B(s) = \varphi(L(s), I(s), u_n)$ constrained by the physical limitations of the battery. Furthermore, $B(s) > 0$ implies supply of power from the battery and $B(s) < 0$ implies charging of the battery. The set of admissible controls is thus defined as:

$$\mathcal{U}_n(L_n, I_n, m_n) := \left\{ u : \mathbb{P}\Big( \sup_{s \in [t_n, t_{n+1})} S(s) > 0 \Big| (L_n, I_n, m_n, u) \Big) < p \right\}. \tag{5.12}$$

Thus in the context of microgrid, the conditional distribution $G_n$ of equation (5.6) and the corresponding $p_n(L_n, I_n, C_n)$ are:

$$\begin{aligned} G_n(L_n, I_n, m_n, u_n) &= \mathcal{L}\Big( \sup_{s \in [t_n, t_{n+1})} S(s) \Big| (L_n, I_n, m_n, u_n) \Big), \\ p_n(L_n, I_n, m_n, u_n) &= \mathbb{P}(G_n(L_n, I_n, m_n, u_n) > 0). \end{aligned} \tag{5.13}$$

Because $p_n$ is not (in general) available analytically, the admissibility condition $p_n(L_n, I_n, m_n) < p$ is implicit. Recall that we denote by $\mathcal{W} = 0 \cup [\underline{u}, \bar{u}]$ the unconstrained control set. We assume that $u(t) = 0$ means that the diesel is OFF, while $u(t) > 0$ means that it is ON, and at output level $u(t)$. Thus, we define $m(s) = \mathbf{1}_{\{u_n > 0\}}$ $\forall s \in (t_n, t_{n+1}]$ with the time interval left-open in order to allow for identification of

switching on and off of the diesel generator at times $t_n$. Notice also that the process $m(t)$ does not satisfy the controlled diffusive dynamics, but this slight extension of the framework does not impact on the methods and results presented. We then look at the following formulation of the general problem:

$$\begin{aligned} V_n(L_n, I_n, m_n) = \quad & \min_{\{u_k\}_{k=n}^{N-1}} \Big\{ \mathbb{E}\Big[ \textstyle\sum_{k=n}^{N-1} \Big[ \rho(u_k)\Delta t_k + \mathbf{1}_{\{m_k=0, u_k>0\}} K(0,1) \Big] \\ & \qquad + W(L_N, I_N, m_N) \Big| (L_n, I_n, m_n) \Big] \Big\}, \\ \text{subject to} \quad & \mathbb{P}\Big( \sup_{s\in[t_n, t_{n+1})} S(s) > 0 \Big| (L_n, I_n, m_n) \Big) < p \quad \forall\, n \end{aligned} \tag{5.14}$$

where $\Delta t_k = t_{k+1} - t_k$, $\rho(u_k)$ is the instantaneous cost of running the diesel generator to produce power output $u_k$ and $K(0,1)$ is the cost of switching on the diesel generator. We continue to assume zero cost to turn off the generator i.e. $K(1,0) = 0$. The DPE corresponding to (5.14) is the same as in (5.3) with the integral running cost $\int_{t_n}^{t_{n+1}} \pi_s(\mathbf{X}(s), u_n)\mathrm{d}s$ replaced by $\rho(u_n)\Delta t_n$ and the switching cost $K(\mathbf{X}_n, u_n)$ by $\mathbf{1}_{\{m_n=0, u_n>0\}} K(0,1)$.

**Remark 21** *The admissible set $\mathcal{U} \subseteq \mathcal{W}$ for this problem has the special structure of being an interval: if $u \in \mathcal{U}(\mathbf{x})$, then $\forall\ \tilde{u} > u, \tilde{u} \in \mathcal{U}(\mathbf{x})$. Hence, we may represent $\mathcal{U}(\mathbf{x}) = [u_n^{\min}(\mathbf{x}), \bar{u}]$ in terms of the minimal admissible diesel output $u_n^{\min}(\mathbf{x})$. Conversely, the admissibility domains for a fixed $u \in \mathcal{W}$ are nested: if $u_1 \leq u_2$ then $\mathcal{X}_n^a(u_1) \subseteq \mathcal{X}_n^a(u_2)$. This suggests to compute $\mathcal{X}_n^a(u)$ sequentially as $u$ is increased and then invert to get $\mathcal{U}(\mathbf{x})$.*

To visualize the minimum admissible control $u_n^{\min}(\mathbf{x})$, the right panel of Figure 5.1 presents the map $\mathbf{x} \to u_n^{\min}(\mathbf{x})$ under a constraint of $p = 0.01$ probability of blackout. We also present a path for $(L(t), I(t), m(t))_{t\geq 0}$ using a *myopic* strategy where the controller employs the minimum admissible control at each point, $u_n := u_n^{\min}(L_n, I_n, m_n)\ \forall n$. Notice how for the most part, $u_n^{\min} = 0$ is trivially admissible so that $\mathcal{U}(\mathbf{x}) = [0, \bar{u}]$

and the blackout constraint is not binding. This is not surprising, as blackouts are only possible when $L(t) \gg 0$ is strongly positive and the battery is close to empty, $I(t) \simeq 0$. Thus, except for the lower-right corner, any control is admissible. As a result, only a small subset of the domain $\mathcal{X}$ actually requires additional effort to estimate the admissible set $\mathcal{U}(\mathbf{x})$. In our experience this structure, where the constraint is not necessarily binding and where we mostly perform unconstrained optimization, is quite common in applications.

## 5.3 Probability Constrained-DEA

In this section we present our Probability constrained dynamic emulation algorithm (PC-DEA) which provides approximation for the admissible set $\hat{\mathcal{U}}_n(\cdot)$ and the continuation value function $\hat{\mathcal{C}}_n(\cdot,\cdot)$. The main steps of the algorithm can be summarized using the following two steps, implemented in parallel at every time-step:

$$\begin{aligned}&\text{Generate design} \rightarrow \text{Generate 1-step paths \& statistic for admissibility} \rightarrow \text{Estimate } \mathcal{U}\\ &\text{Generate design} \rightarrow \text{Generate 1-step paths \& pathwise profits} \rightarrow \text{Estimate } \mathcal{C}\end{aligned} \tag{5.15}$$

To estimate $\hat{\mathcal{C}}_n(\cdot,\cdot)$'s and $\hat{\mathcal{U}}_n(\cdot)$'s, we proceed iteratively backward in time starting with known terminal condition $W(\mathbf{X})$ and sequentially estimate $\hat{\mathcal{U}}_n$ and $\hat{\mathcal{C}}_n$ for $n = N-1,\ldots,0$. Assuming we have estimated $\hat{\mathcal{U}}_{n+1},\ldots,\hat{\mathcal{U}}_{N-1}$ and $\hat{\mathcal{C}}_{n+1},\ldots,\hat{\mathcal{C}}_{N-1}$, we first explain the estimation procedure for $\hat{\mathcal{U}}_n$ and $\hat{\mathcal{C}}_n$. This corresponds to a `fit` task. In the subsequent backward recursion at step $n-1$ we also need the `predict` task to actually *evaluate* $\hat{V}_n(\mathbf{X}_n)$ which requires evaluating $\hat{\mathcal{C}}_n(\cdot)$ at new ("out-of-sample") inputs $\mathbf{X}_n, u_n$ which of course do not coincide with the training inputs $(\mathbf{x}_n^1, u_n^1)$, ..., $(\mathbf{x}_n^{M_c}, u_n^{M_c})$.

### 5.3.1 Estimating the set of admissible controls

To estimate the set of admissible controls $\hat{\mathcal{U}}_n(\cdot)$ at time-step $n$, we choose design $\mathcal{D}_n^a := (\mathbf{x}_n^i, u_n^i, i = 1, \ldots, M_a)$ and simulate trajectories of the state process $(\mathbf{X}(s))_{s\in[t_n,t_{n+1})}^i$ starting from $\mathbf{X}^i(t_n) = \mathbf{x}_n^i$ and driven by control $u_n^i$. To evaluate the functional $\mathcal{G}\big((\mathbf{X}(s))_{s\in[t_n,t_{n+1})}^i\big)$, we discretize the time interval $[t_n, t_{n+1})$ into $K$ finer sub-steps with $\Delta n_k := t_{n_{(k+1)}} - t_{n_k}$ and define the discrete trajectory $\mathbf{x}_n^i = \mathbf{x}_{n_0}^i, \mathbf{x}_{n_1}^i, \ldots, \mathbf{x}_{n_{(K-1)}}^i, \mathbf{x}_{n_K}^i$. We then record

$$w_n^i := \mathbf{1}\Big(\mathcal{G}((\mathbf{x}_{n_k}^i)_{k\in\{0,\ldots,K-1\}}) > 0\Big), \qquad i = 1, \ldots, M_a, \tag{5.16}$$

where, formally, we extend $(\mathbf{x}_{n_k}^i)_{k\in\{0,\ldots,K-1\}}$ to a piecewise constant trajectory on $[t_n, t_{n+1})$.

Analogous to standard RMC, we now select an approximation space $\mathcal{H}_n^a$ to estimate the probability $\hat{p}_n$ or the quantile $\hat{q}_n$, using the loss function $\mathcal{L}_n^a$ and apply empirical projection:

$$\hat{p}_n := \arg\min_{f_n^a\in\mathcal{H}_n^a} \sum_{i=1}^{M_a} \mathcal{L}_n^a(f_n^a, w_n^i; \mathbf{x}_n^i, u_n^i). \tag{5.17}$$

See Section 5.4 for concrete examples of $\mathcal{H}^a$ and $\mathcal{L}^a$. Note that the approximations $\hat{p}_n$ and $\hat{q}_n$ must be trained on joint state-control datasets $\{\mathbf{x}_n^i, u_n^i, w_n^i\}_{i=1}^{M_a}$ with $w_n^i$ dependent on the method of choice and moreover yield random estimators ($\hat{p}_n$ is a random variable).

Using the distribution of $\hat{p}_n(\mathbf{x}, u)$ we may obtain a more conservative estimator that provides better guarantees on the ultimate admissibility of $(\mathbf{x}, u)$. As a motivation, recall the NMC estimator $\bar{p}_n(\mathbf{x}, u)$ from (5.11); for reasonably large $M_b \gg 20$, the distribution of $\bar{p}_n(\mathbf{x}, u)$ is approximately Gaussian with mean $p_n(\mathbf{x}, u)$ and variance

$\sqrt{\frac{p_n(\mathbf{x},u)(1-p_n(\mathbf{x},u))}{M_b}}$. Defining

$$\hat{p}_n^{(\rho)}(\mathbf{x},u) := \bar{p}_n(\mathbf{x},u) + \xi_n^{(\rho)}(\mathbf{x},u) \tag{5.18}$$

$$:= \bar{p}_n(\mathbf{x},u) + z_\rho \sqrt{\frac{\bar{p}_n(\mathbf{x},u)(1-\bar{p}_n(\mathbf{x},u))}{M_b}}, \tag{5.19}$$

where $z_\rho$ is the standard normal quantile at level $\rho$ and $\xi_n^{(\rho)}(\mathbf{x},u)$ represents a "safe" margin of error for $\bar{p}_n$ at confidence level $\rho$. The corresponding admissible set with confidence $\rho$ is

$$\hat{\mathcal{U}}_n^{(\rho)}(\mathbf{x}) := \hat{\mathcal{U}}_n^{\xi^{(\rho)}}(\mathbf{x}) = \left\{u : \hat{p}_n(\mathbf{x},u) + \xi_n^{(\rho)}(\mathbf{x},u) < p\right\}. \tag{5.20}$$

More generally, we set the admissible set for a site $\mathbf{x} \in \mathcal{X}$ to

$$\hat{\mathcal{U}}_n^{\xi}(\mathbf{x}) = \left\{u : \hat{p}_n(\mathbf{x},u) + \xi_n(\mathbf{x},u) < p\right\}, \tag{5.21}$$

where $\xi_n(\mathbf{x},u)$ ensures "stronger" guarantee for the admissibility of $u$ at $\mathbf{x}$. The margin of estimation error can also be fixed, $\xi_n(\mathbf{x},u) = c \; \forall (\mathbf{x},u) \in \mathcal{X} \times \mathcal{W}$, which can be applied when the sampling distribution of $\hat{p}_n(\mathbf{x},u)$ is unknown or cannot be approximated using a Gaussian distribution. The corresponding admissible set

$$\hat{\mathcal{U}}_n^{\xi=c}(\mathbf{x}) = \left\{u : \hat{p}_n(\mathbf{x},u) + c < p\right\}. \tag{5.22}$$

is equivalent to estimating $\hat{\mathcal{U}}_n^{\xi=0}(\mathbf{x})$ with a shifted lower probability threshold $p-c$. For simplicity of notation, throughout this article we use $\hat{\mathcal{U}}_n(\mathbf{x})$ to denote the unadjusted admissible set,

$$\hat{\mathcal{U}}_n(\mathbf{x}) := \hat{\mathcal{U}}_n^{\xi=0}(\mathbf{x})$$

in the context of NMC. As mentioned in Remark 16, equations (5.16)-(5.17) based on

learning the quantile $q_n(\mathbf{x}, u)$ could also be adjusted analogously to (5.21) to add a margin of error, $\hat{\mathcal{U}}_n^{\xi}(\mathbf{x}) = \{u : \hat{q}_n(\mathbf{x}, u) + \xi_n(\mathbf{x}, u) < 0\}$.

### 5.3.2 Estimating the continuation value

To estimate the continuation value $\mathcal{C}_n(\cdot, \cdot)$, we choose a simulation design $\mathcal{D}_n^c := (\mathbf{x}_n^j, u_n^j, j = 1 \ldots, M_c)$ (which could be independent or equivalent to $\mathcal{D}_n^a$) and generate one-step paths for the state process $(\mathbf{X}(s))_{s\in[t_n,t_{n+1})}^j$ starting from $\mathbf{X}^j(t_n) = \mathbf{x}_n^j$ and driven by control $u_n^j$, comprising again of finer sub-steps $\mathbf{x}_n^j = \mathbf{x}_{n0}^j, \mathbf{x}_{n1}^j, \ldots, \mathbf{x}_{n(K-1)}^j, \mathbf{x}_{nK}^j$ (in principle the sub-steps could differ from the time discretization for $\hat{\mathcal{U}}_n$). Next, we compute the pathwise cost $y_n^j$, $j = 1 \ldots M_c$:

$$\begin{aligned} y_n^j &= \sum_{k=0}^{K-1} \pi_{n_k}(\mathbf{x}_{n_k}^j, u_n^j)\Delta n_k + K(\mathbf{x}_n^j, u_n^j) + v_{n+1}^j, \\ \text{where } v_{n+1}^j &= \inf_{u\in\hat{\mathcal{U}}_{n+1}(\mathbf{x}_{nK}^j)} \hat{\mathcal{C}}_{n+1}(\mathbf{x}_{nK}^j, u), \end{aligned} \tag{5.23}$$

and we replace the time integral in (5.3) with a discrete sum over $t_{n_k}$'s. At the key step, we project $\{y_n^j\}_{j=1}^{M_c}$ onto an approximation space $\mathcal{H}_n^c$ to evaluate the continuation value $\mathcal{C}_n(\cdot, \cdot)$:

$$\hat{\mathcal{C}}_n(\cdot, \cdot) := \arg\min_{f_n^c\in\mathcal{H}_n^c} \sum_{n=1}^{M_c} |f_n^c(\mathbf{x}_n^j, u_n^j) - y_n^j|^2. \tag{5.24}$$

The design sites $\{\mathbf{x}_n^j, u_n^j\}_{j=1}^{M_c}$ could be same or different from those used for learning the admissible sets in the previous subsection. Two standard approximation spaces $\mathcal{H}_n^c$ used in this context are: global polynomial approximation and piecewise continuous approximation.

**Remark 22** *In the microgrid example of Section 5.2.2 the running cost over $[n, n+1)$ is known once the control $u_n$ is chosen. Thus it can be taken outside the conditional expectation and the data to be regressed is simply $y^j = v_{n+1}^j$.*

**Global polynomial approximation:** As described in Chapters 3 and 4, this is a classical regression framework where $\hat{\mathcal{C}}_n^{\boldsymbol{\alpha}}(\mathbf{x}, u) := \sum_k \alpha_k \phi_k(\mathbf{x}, u)$, where $\phi_k(\cdot, \cdot)$ is a polynomial basis and the coefficients $\boldsymbol{\alpha}$ are fitted via

$$\hat{\boldsymbol{\alpha}} := \arg\min_{\boldsymbol{\alpha}} \sum_{j=1}^{M_c} \left| \sum_k \alpha_k \phi_k(\mathbf{x}^j, u^j) - y^j \right|^2 . \tag{5.25}$$

Notice that in contrast to Chapters 3 and 4, here we have explicitly incorporated the control $u$ in the basis functions. As an illustration, for the microgrid example of Section 5.2.2 we construct a quadratic polynomial approximation when diesel generator is ON, $u > 0$, using 10 bases $\{1, L, I, u, L^2, I^2, u^2, LI, Iu, LI\}$ and a separate quadratic approximation with the 6 basis functions $\{1, L, I, L^2, I^2, LI, LI\}$ when diesel generator is OFF, $u = 0$. Polynomial approximation is easy to implement but typically requires many degrees of freedom (lots of $\phi$'s) to properly capture the shape of $\mathcal{C}$ and can be empirically unstable, especially if there are sharp changes in the underlying function (see left panel of figure 4.4 in Chapter 4 and [62]).

**Piecewise continuous approximation:** This is a state-of-art tool in low dimensions, $d \leq 3$ (see Chapter 3). The main idea is to employ polynomial regression in a single dimension and extend to the other dimensions via linear interpolation. As an example, for the microgrid with diesel generator ON, we have three dimensions $(L, I, u)$. We discretize inventory $I$ as $\{I^0, I^1, \ldots, I^{M_I}\}$ and control $u$ as $\{u^1, u^2, \ldots, u^{M_u}\}$ and fit independent cubic polynomials in $L$ for each pair $(I^l, u^e)$ with $l \in \{0, 1, \ldots, M_I\}$ and $e \in \{0, 1, \ldots, M_u\}$, i.e., $f_n^{l,e}(L) = \sum_k \alpha_k^{l,e} \phi_k(L)$. For any $(I, u) \in [I^l, I^{l+1}] \times [u^e, u^{e+1}]$ we then provide the interpolated approximation $\hat{\mathcal{C}}_n(L, I, u)$ as

$$\hat{\mathcal{C}}_n(L, I, u) = \frac{\begin{bmatrix} I^{l+1} - I & I - I^l \end{bmatrix} \begin{bmatrix} f_n^{l,e}(L) & f_n^{l,e+1}(L) \\ f_n^{l+1,e}(L) & f_n^{l+1,e+1}(L) \end{bmatrix} \begin{bmatrix} u^{e+1} - u \\ u - u^e \end{bmatrix}}{(u^{e+1} - u^e)(I^{l+1} - I^l)}. \tag{5.26}$$

**Nonparametric approximation:** These avoid the problem of choosing a basis function and thus creating a bias in the approximation of the continuation value function. In Chapter 4, Section 4.4 we presented several non-parametric regression methods including Gaussian process regression, local polynomial regression and piecewise multivariate linear regression.

### 5.3.3 Evaluation

We analyze the quality of the solution by computing three quantities on the out-of-sample dataset:

- estimate of the value function $V_0(\mathbf{x}_0)$ at $t = 0$ and state $\mathbf{x}_0$;
- empirical frequency of inadmissible decisions on the controlled trajectories $\mathbf{x}_{\cdot}^{\hat{u}}$;
- statistical test for the realized number of constraint violations (blackouts for the microgrid).

Good solutions should minimize costs and not apply inadmissible controls. However, since we employ empirical estimators, $\mathcal{U}$ is never known with certainty and we must handle the possibility that constraints are violated with probability more than $p$. In turn this leads to the trade-off between complying with (5.2) and optimizing costs. Similar treatment of constraints in the context of sample average approximation of probabilistic constrained optimization problems have been discussed in [73, 74]. Moreover, our

framework implies that the whole algorithm is stochastic: multiple runs will lead to different results since both $\hat{p}_n$ and $\hat{\mathcal{C}}_n$ are impacted by the random samples $y_n^j$ and $w_n^i$.

**Estimate of the value function:** We evaluate the value function $\hat{V}_0(\mathbf{x}_0)$ at time $t_0 = 0$ and state $\mathbf{x}_0$ using $M'$ out-of-sample paths $(\mathbf{x}_{0:N}^{\hat{u},m'}), m' = 1, \ldots, M'$. Each trajectory $(\mathbf{x}_{0:N}^{\hat{u},m'})$ is generated by applying the estimated optimal control $\hat{u}_{0:N-1}$ based on the estimated counterparts of both, the continuation value function and admissible sets $(\hat{\mathcal{C}}_n, \hat{\mathcal{U}}_n)_{n=0}^{N-1}$ leading to the realized pathwise cost:

$$v_0(\mathbf{x}_{0:N}^{\hat{u},m'}) := \sum_{n=0}^{N-1} \sum_{k=0}^{K-1} \pi_{n_k}(\mathbf{x}_{n_k}^{\hat{u},j}, \hat{u}_n^j)\Delta n_k + K(\mathbf{x}_n^{\hat{u},j}, \hat{u}_n^j) + W(\mathbf{x}_N^{\hat{u},j}).$$

The resulting empirical Monte Carlo estimate is

$$\hat{V}_0(\mathbf{x}_0) \simeq \frac{1}{M'} \sum_{m'=1}^{M'} v_0(\mathbf{x}_{0:N}^{\hat{u},m'}) \tag{5.27}$$

and represents an unbiased estimation of the value of the control policy and an upper bound estimation of the value function, provided all controls used are admissible.

The sequence of steps for evaluating Equation (5.27) is also described in Algorithm 4. Therein we consider explicit admissible sets. Since this chapter focuses on admissible sets which are implicit in nature, $\mathcal{U}_n$ in Line 4 of Algorithm 4 should be replaced by the corresponding estimate $\hat{\mathcal{U}}_n$.

**Empirical frequency of inadmissible decisions on the controlled trajectories**: For the $M'$ out-of-sample paths, we compare the estimated optimal control $\{\hat{u}_n(\mathbf{x}_n^{\hat{u},m'})\}_{n=m=1}^{N-1,M'}$ against the minimum admissible control $\{u_n^{\min}(\mathbf{x}_n^{\hat{u},m'})\}_{n=m=1}^{N-1,M'}$ assumed for a second to be known. Namely, for each path we compute the number of inadmissible decisions $w_0(\mathbf{x}_{0:N}^{\hat{u},m'})$ and the empirical frequency of inadmissible decisions

$w_{freq}$ as:

$$w_0(\mathbf{x}_{0:N}^{\hat{u},m'}) := \sum_n \mathbb{1}_{\hat{u}_n(\mathbf{x}_n^{\hat{u},m'})<u_n^{\min}(\mathbf{x}_n^{\hat{u},m'})} \quad \text{and} \quad w_{freq} := \frac{1}{N \cdot M'} \sum_{m'=1}^{M'} w_0(\mathbf{x}_{0:N}^{\hat{u},m'}), \quad (5.28)$$

respectively. We employ these metrics in Section 5.5, where we obtain the "gold standard" $\{u_n^{\min}(\mathbf{x}_n^{\hat{u},m'})\}_{n=m'=1}^{N-1,M'}$ by brute force, utilizing a simulation budget $10^5$ larger than for the actual methods we are comparing. Empirical gold standard is a common technique when analytical benchmark is unavailable, see e.g. [75]. A good estimation method should yield $w_{freq} \simeq 0$.

**Statistical test:** Next we propose statistical tests using the controlled trajectories to validate different methods for admissible set estimation. Such a test is essential to affirm the use of a numerical scheme for $\mathcal{U}_n$ in the absence of a benchmark. As an example, in the context of microgrid we want to test the null hypothesis $H_0$ that the realized probability of blackouts is bounded to the required level against the alternative $H_1$ that their probability is too high. Let

$$B_n^{m'} = \mathbb{1}\Big(\mathcal{G}(\mathbf{x}_{s\in[t_n,t_{n+1})}^{\hat{u},m'}) > 0\Big), \quad n = 0,\dots,N-1 \text{ and } m' = 1,\dots,M'. \quad (5.29)$$

Ignoring the correlation due to the temporal dependence in $\mathbf{x}_n$, we assume that $B_n^{m'} \sim \text{Bernoulli}(\tilde{p})$, i.i.d. We want to test:

$$H_0 : \tilde{p} \le p \quad \text{vs.} \quad H_0 : \tilde{p} > p. \quad (5.30)$$

A common approach to such composite null hypothesis is to replace $H_0$ with a more conservative hypothesis $\tilde{p} = p$ leading to the test statistic

$$\mathcal{T} := \frac{\sum_{m',n}(B_n^{m'} - p)}{\sqrt{M' \cdot N \cdot p \cdot (1-p)}} \sim \mathcal{N}(0,1). \quad (5.31)$$

Hence, $H_0$ is rejected at a confidence level $\alpha$ if $\mathcal{T} > z_\alpha$ with $z_\alpha = \Phi^{-1}(\alpha)$, e.g. $z_\alpha = 1.65$ for $\alpha = 95\%$.

**Remark 23** *The above test assumes independence and identical distribution of $B_n^{m'}$'s. In the context of the microgrid example, neither of the two assumptions are valid; $B_n^{m'}$ have different distribution because the state of the system affects the probability of a blackout, thus $\tilde{p}$ varies with $n, m'$. Furthermore, $B_n^{m'}$ are not independent as they are derived from a single, sequentially controlled trajectory.*

**Remark 24** *In the microgrid setup, the blackout constraint is frequently not binding (the net demand is negative half of the time). Therefore, $\mathcal{T}$ as defined in equation* (5.31) *is most likely negative leading to accept the $H_0$ even when the method fails to choose the admissible control when the constraint is binding. We fix this by evaluating the sum only when the constraint is binding, i.e.*

$$\tilde{\mathcal{T}} := \frac{\sum_{m',n}(B_n^{m'} - p)\mathbb{1}_{u_n^{\min}(\mathbf{x}_n^{\hat{u},m'})>0}}{\sqrt{p\cdot(1-p)\cdot M'\cdot N\cdot w_{bind}}} \quad \textit{where} \quad w_{bind} = \frac{\sum_{m',n}\mathbb{1}_{u_n^{\min}(\mathbf{x}_n^{\hat{u},m'})>0}}{M'\cdot N}. \tag{5.32}$$

To wrap up this section, Algorithm 6 (dubbed Dynamic Emulation due to similarities with Algorithm 5 for stochastic control with explicit constraints) summarizes the overall sequence of steps. Lines 1-6 can be thought of as part of a stochastic simulator which generates designs and one-step paths for each design site. Line 8 (and again Line 18) computes pathwise one-step costs. Line 10 is the admissible set estimation. Line 11 is the estimation of the continuation value. Lines 12-17 call the stochastic simulator for generating new design and one-step paths.

Algorithm 6 carries several advantages. First and foremost it is very general, as the method does not assume any restriction on the distribution $G_n(\mathbf{X}_n, u)$ defining $\mathcal{U}_n$ or the form of the payoffs $\pi(\mathbf{x}, u)$. Hence it can be generically applied across a wide spectrum of SCPC problems. Second, the same template (in particular based on having two

essentially independent sub-modules) accommodates a slew of potential techniques for learning $\mathcal{C}$ and $\mathcal{U}$ bringing plug-and-play functionality, such as straightforward switching from probability estimation to quantile estimation. Third, it allows for computational savings either through parallelizing the estimation of $\mathcal{U}$ and $\mathcal{C}$, or by re-using the same design and simulations $\mathcal{D}_n^a \equiv \mathcal{D}_n^c$ for the computation of the two sub-modules.

**Remark 25** *The challenge of RMC methods is that the errors recursively propagate backward. As a result, poor estimation at one step can affect the overall quality of the solution. In our algorithm, the errors at every step occur due to:*

- *Approximation architecture $\mathcal{H}_n^a$ for $\hat{\mathcal{U}}_n \Rightarrow$ Projection error in admissible control set estimation;*
- *Approximation architecture $\mathcal{H}_n^c$ for $\hat{\mathcal{C}}_n \Rightarrow$ Projection error in estimating continuation value;*
- *Designs $\mathcal{D}_n^a$ and $\mathcal{D}_n^c \Rightarrow$ Finite-sample Monte Carlo errors (difference between empirical estimates and theoretical projection-based ones)*
- *Discretization of the time interval $[t_n, t_{n+1})$ using $\Delta n_k \Rightarrow$ Integration error in approximating the integral $\int_{t_n}^{t_{n+1}} \pi_s(\mathbf{X}(s), u)\mathrm{d}s$ and the admissible set $\mathcal{U}_n$.*
- *Numerical approximation of the solution of the controlled dynamics of $\mathbf{X}(t)$.*
- *Optimization errors in maximizing for $\hat{u}$ over $\hat{\mathcal{U}}$, especially when the control set $\mathcal{W}$ is continuous.*

## 5.4 Admissible set estimation

In this section we propose two different approaches to estimate the admissible set of controls $\mathcal{U}_n$ in equation (5.5):

**Algorithm 6:** PC-DEA

**Data:** $N$ (time steps), $M_c$ (simulation budget for conditional expectation), $M_a$ (simulation budget for admissible set estimation)

Generate designs:

  $\mathcal{D}^a_{N-1} := (\mathbf{x}^{\mathcal{D}^a_{N-1}}_{N-1}, u^{\mathcal{D}^a_{N-1}}_{N-1})$ of size $M_a$ for estimating $\hat{\mathcal{U}}$.

  $\mathcal{D}^c_{N-1} := (\mathbf{x}^{\mathcal{D}^c_{N-1}}_{N-1}, u^{\mathcal{D}^c_{N-1}}_{N-1})$ of size $M_c$ for estimating $\hat{\mathcal{C}}$.

Generate one-step paths:

  $\mathbf{x}^{i,\mathcal{D}^a_{N-1}}_{N-1} \mapsto \mathbf{x}^{i,\mathcal{D}^a_{N-1}}_{N}$ using $u^{\mathcal{D}^a_{N-1}}_{N-1}$ for $i = 1, \ldots, M_a$

  $\mathbf{x}^{j,\mathcal{D}^c_{N-1}}_{N-1} \mapsto \mathbf{x}^{j,\mathcal{D}^c_{N-1}}_{N}$ using $u^{\mathcal{D}^c_{N-1}}_{N-1}$ for $j = 1, \ldots, M_c$

Terminal condition:

  $y^j_{N-1} \leftarrow \sum_{k=0}^{K-1} \pi_{(N-1)_k}(\mathbf{x}^{j,\mathcal{D}^c_{N-1}}_{(N-1)_k}, u^{j,\mathcal{D}^c_{N-1}}_{(N-1)_k}) + W(\mathbf{x}^{j,\mathcal{D}^c_{N-1}}_{N})$ for $j = 1, \ldots, M_c$

**for** $n = N-1, \ldots, 1$ **do**

  Estimate $\hat{\mathcal{U}}_n(\cdot)$ using methods in Section 5.4 and paths $\mathbf{x}^{i,\mathcal{D}^a_n}_n \mapsto \mathbf{x}^{i,\mathcal{D}^a_n}_{n+1}$

  $\hat{\mathcal{C}}_n(\cdot,\cdot) \leftarrow \underset{f_n \in \mathcal{H}^c_n}{\arg\min} \sum_{j=1}^{M_c} |f_n(\mathbf{x}^{j,\mathcal{D}^c_n}_n, u^{j,\mathcal{D}^c_n}_n) - y^j_n|^2$

  Generate designs:

    $\mathcal{D}^a_{n-1} := (\mathbf{x}^{\mathcal{D}^a_{n-1}}_{n-1}, u^{\mathcal{D}^a_{n-1}}_{n-1})$ of size $M_a$ for estimating $\hat{\mathcal{U}}$.

    $\mathcal{D}^c_{n-1} := (\mathbf{x}^{\mathcal{D}^c_{n-1}}_{n-1}, u^{\mathcal{D}^c_{n-1}}_{n-1})$ of size $M_c$ for estimating $\hat{\mathcal{C}}$.

  Generate one-step paths:

    $\mathbf{x}^{i,\mathcal{D}^a_{n-1}}_{n-1} \mapsto \mathbf{x}^{i,\mathcal{D}^a_{n-1}}_{n}$ using $u^{\mathcal{D}^a_{n-1}}_{n-1}$ for $i = 1, \ldots, M_a$

    $\mathbf{x}^{j,\mathcal{D}^c_{n-1}}_{n-1} \mapsto \mathbf{x}^{j,\mathcal{D}^c_{n-1}}_{n}$ using $u^{\mathcal{D}^c_{n-1}}_{n-1}$ for $j = 1, \ldots, M_c$

  $y^j_{n-1} \leftarrow \sum_{k=0}^{K-1} \pi_{(n-1)_k}(\mathbf{x}^{j,\mathcal{D}^c_{n-1}}_{(n-1)_k}, u^{j,\mathcal{D}^c_{n-1}}_{(n-1)_k}) + \max_{u \in \hat{\mathcal{U}}_n(\mathbf{x}^{j,\mathcal{D}^c_{n-1}}_n)} \Big\{\hat{\mathcal{C}}(n, \mathbf{x}^{j,\mathcal{D}^c_{n-1}}_n, u)\Big\}$ $\forall j$

**end**

return $\{\hat{\mathcal{C}}_n(\cdot,\cdot), \hat{\mathcal{U}}_n(\cdot)\}_{n=1}^{N-1}$

- **Probability estimation:** Given a state $\mathbf{X}_n = \mathbf{x}$ and $u \in \mathcal{W}$, we estimate, via simulation, the probability of violating the constraint

$$\hat{p}_n(\mathbf{x}, u) \simeq \mathbb{P}\Big(G_n(\mathbf{x}, u) > 0\Big).$$

It follows that $u \in \hat{\mathcal{U}}_n(\mathbf{x}) \;\Leftrightarrow\; \hat{p}_n(\mathbf{x}, u) < p$. Particularly, to compute $\hat{p}_n(\mathbf{x}, u)$ we

consider Gaussian process smoothing of empirical probabilities, logistic regression and parametric density fitting.

- **Quantile estimation:** We approximate the quantile $q_n(\mathbf{x}, u)$ of $G_n(\mathbf{x}, u)$ via empirical ranking, support vector machines and quantile regression methods. The admissible sets $\mathcal{U}_n(\mathbf{x})$ and $\mathcal{X}_n^a(u)$ are then defined as:

$$\hat{\mathcal{U}}_n(\mathbf{x}) := \Big\{ u : \hat{q}_n(\mathbf{x}, u) < 0 \Big\} \text{ and } \quad \hat{\mathcal{X}}_n^a(u) := \Big\{ \mathbf{x} : \hat{q}_n(\mathbf{x}, u) < 0 \Big\}.$$

To implement all of the above techniques we use Monte Carlo simulation, specifying first the simulation design and then sampling (independently across draws) the $G$'s or $Y$'s to be used as training data. We work in a flexible framework where samples of $G_n(\mathbf{x}, u)$ are generated in batches of $M_b$ simulations from each design site $\{\mathbf{x}^i, u^i\}_{i=1}^{M_a}$. The case of $M_b = 1$ corresponds to a classical regression approach, while large $M_b \gg 1$ can be interpreted as nested Monte Carlo averaging along $M_b$ inner samples.

**Remark 26** *In section 5.3.2, we parameterized the elements of the approximation space $\mathcal{H}_n^c$ for estimation of the continuation value function $\hat{\mathcal{C}}(\cdot, \cdot)$ via vectors $\boldsymbol{\alpha}$ i.e. $f_n^c(\mathbf{x}, u) \equiv f_n^c(\mathbf{x}, u; \boldsymbol{\alpha})$ (see equations* (5.24) *and* (5.25)*). To maintain distinct notations, in the following sections we will use $\boldsymbol{\beta}$ to generically parameterize the elements of the approximation space $\mathcal{H}_n^a$ for estimation of the admissible set, so that $f_n^a(\mathbf{x}, u) \equiv f_n^a(\mathbf{x}, u; \boldsymbol{\beta})$ in equation* (5.17)*. The meaning and dimension of $\boldsymbol{\beta}$ will vary from method to method.*

## 5.4.1 Probability estimation

### Interpolated nested Monte Carlo (INMC)

Recall the NMC method from Section 5.2.1 where we select $M_a$ design sites of state-action pairs and simulate multiple paths from each site to *locally* assess the probability

of $G_n(\mathbf{x}, u) > 0$ (in what follows, we suppress in the notation the dependence on $n$). Specifically, for each design site $(\mathbf{x}^i, u^i),\ i = 1, \ldots, M_a$, we simulate $M_b$ batched samples from the distribution $G(\mathbf{x}^i, u^i)$ as $\{g^b(\mathbf{x}^i, u^i)\}_{b=1}^{M_b}$. The unbiased point estimator of $p(\mathbf{x}^i, u^i)$ is:

$$\bar{p}(\mathbf{x}^i, u^i) := \sum_{b=1}^{M_b} \frac{\mathbf{1}_{g^b(\mathbf{x}^i, u^i)>0}}{M_b}. \tag{5.33}$$

Since (5.33) only yields $M_b$ local estimates $\bar{p}(\mathbf{x}^i, u^i)$, for Algorithm 6 we have to extend them to an arbitrary state-action $(\mathbf{x}, u) \mapsto \hat{p}_{\text{INMC}}(\mathbf{x}, u)$. This is achieved by interpolating $\bar{p}(\mathbf{x}^i, u^i)$'s, e.g. linearly. The admissible set with confidence level $\rho$ becomes:

$$\hat{\mathcal{U}}_{\text{INMC}}^{(\rho)}(\mathbf{x}) := \left\{ u : \hat{p}_{\text{INMC}}(\mathbf{x}, u) \leq p - z_\rho \sqrt{\frac{\hat{p}_{\text{INMC}}(\mathbf{x}, u)(1 - \hat{p}_{\text{INMC}}(\mathbf{x}, u))}{M_b}} \right\}.$$

However, especially for $M_b$ small, interpolation performs poorly because the underlying point estimates $\bar{p}(\mathbf{x}^i, u^i)$ are *noisy*. Therefore, smoothing should be applied leading to consideration of statistical *regression* models. Regression allows to borrow information cross-sectionally to remove the above estimation noise and hence lower both the bias and variance of $\bar{p}$.

### Gaussian process regression (GPR)

One flexible non-parametric regression method we propose is Gaussian process regression (GPR). Recall, in Chapter 4 we introduced GPR in the context of estimating $\mathcal{C}$, while here the objective is to estimate $\mathcal{U}$. GPR assumes that the map $(\mathbf{x}, u) \to p(\mathbf{x}, u)$ is a realization of a Gaussian random field so that $\{p(\mathbf{x}, u) | (\mathbf{x}, u) \in \mathcal{X} \times \mathcal{W}\}$ is a collection of random variables with any finite subset being multivariate Gaussian. For any $n$ design sites $\{(\mathbf{x}^i, u^i)\}_{i=1}^n$, GPR posits that

$$p(\mathbf{x}^1, u^1), \ldots, p(\mathbf{x}^n, u^n) \sim \mathcal{N}(\vec{m}_n, \mathbf{K}_n)$$

with mean vector $\vec{m}_n := [m(\mathbf{x}^1, u^1; \boldsymbol{\beta}), \ldots, m(\mathbf{x}^n, u^n; \boldsymbol{\beta})]$ and $n \times n$ covariance matrix $\mathbf{K}_n$ comprised of $\kappa(\mathbf{x}^i, u^i, \mathbf{x}^{i'}, u^{i'}; \boldsymbol{\beta})$, for $1 \leq i, i' \leq n$. The vector $\boldsymbol{\beta}$ represents all the hyperparameters for this model.

Given the training dataset $\{(x^i, u^i), \bar{p}^i\}_{i=1}^{M_a}$ (where $\bar{p}^i$ is a shorthand for $\bar{p}(\mathbf{x}^i, u^i)$), GPR infers the posterior of $p(\cdot, \cdot)$ by assuming an observation model of the form $\bar{p}(\mathbf{x}, u) = p(\mathbf{x}, u) + \epsilon$ with a Gaussian noise term $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Conditioning equations for multivariate normal vectors imply that the posterior predictive distribution $p(\mathbf{x}, u) | \{(x^i, u^i), \bar{p}^i\}_{i=1}^{M_a}$ at any arbitrary site $(\mathbf{x}, u)$ is also Gaussian with the posterior mean $\hat{p}_{\text{GPR}}(\mathbf{x}, u)$ that is the proposed estimator of $p(\mathbf{x}, u)$:

$$\hat{p}_{\text{GPR}}(\mathbf{x}, u) := m(\mathbf{x}, u) + K^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}(\vec{p} - \vec{m}) = \mathbb{E}\Big[p(\mathbf{x}, u) \big| \vec{x}, \vec{u}, \vec{p}\Big] \tag{5.34}$$

$$\begin{aligned} \text{where} \quad & \vec{x} = [\mathbf{x}^1, \ldots, \mathbf{x}^{M_a}]^T, \quad \vec{u} = [u^1, \ldots, u^{M_a}]^T, \quad \vec{p} = [\bar{p}^1, \ldots, \bar{p}^{M_a}]^T, \\ & K^T = [\kappa(\mathbf{x}, u, \mathbf{x}^1, u^1; \boldsymbol{\beta}), \ldots, \kappa(\mathbf{x}, u, \mathbf{x}^{M_a}, u^{M_a}; \boldsymbol{\beta})], \\ & \vec{m} = [m(\mathbf{x}^1, u^1; \boldsymbol{\beta}), \ldots, m(\mathbf{x}^{M_a}, u^{M_a}; \boldsymbol{\beta})], \end{aligned} \tag{5.35}$$

and $\mathbf{K}$ is $M_a \times M_a$ covariance matrix described through the kernel function $\kappa(\cdot, \cdot; \beta)$.

The mean function is often assumed to be constant $m(\mathbf{x}, u; \boldsymbol{\beta}) = \beta_0$ or described using a linear model $m(\mathbf{x}, u; \boldsymbol{\beta}) = \sum_{k=1}^{K} \beta_k \phi(\mathbf{x}^i, u^i)$ with $\phi(\cdot, \cdot)$ representing a polynomial basis. A popular choice for the kernel $\kappa(\cdot, \cdot, \cdot, \cdot)$ is squared exponential (see equation (5.36)) with $\{\{\beta_{\text{len},k}\}_{k=1}^{d}, \beta_{\text{len},u}\}$ termed the lengthscales and $\sigma_p^2$ the process variance of $p(\cdot, \cdot)$:

$$\kappa(\mathbf{x}^i, u^i, \mathbf{x}^{i'}, u^{i'}) = \sigma_p^2 \exp\Big( - \sum_{k=1}^{d} \frac{(x^{i,k} - x^{i',k})^2}{\beta_{\text{len},k}} - \frac{(u^i - u^{i'})^2}{\beta_{\text{len},u}} \Big). \tag{5.36}$$

The set of the hyperparameters $\boldsymbol{\beta} := (\{\beta_k\}_{k=1}^{K}, \{\beta_{\text{len},k}\}_{k=1}^{d}, \beta_{\text{len},u}, \sigma_p^2, \sigma_\epsilon^2)$ is estimated by maximizing the log-likelihood function using the dataset $\{(x^i, u^i), \bar{p}^i\}_{i=1}^{M_a}$. Besides squared exponential kernel described above, other popular kernels include Matern-3/2

and Matern-5/2 [76].

A conservative estimate $\hat{p}^{(\rho)}_{\text{GPR}}(\mathbf{x}, u)$ at confidence level $\rho$ is obtained by explicitly incorporating the (estimated) standard error of $\bar{p}(\mathbf{x}^i, u^i)$ into the GPR smoothing. Namely, we adjust the training dataset to $\{(x^i, u^i), \bar{p}^i_\rho\}_{i=1}^{M_a}$, where $\bar{p}^i_\rho := \bar{p}(\mathbf{x}^i, u^i) + z_\rho \sqrt{\frac{\bar{p}(\mathbf{x}^i,u^i)(1-\bar{p}(\mathbf{x}^i,u^i))}{M_b}}$. The resulting $\hat{p}^{(\rho)}_{\text{GPR}}(\mathbf{x}, u)$ is the counterpart of (5.34) using $\{(x^i, u^i), \bar{p}^i_\rho\}_{i=1}^{M_a}$.

In Figure 5.2b we present the dataset $\{L^i, I^i, 0, \bar{p}^i\}_{i=1}^{M_a}$ (background colormap) for the microgrid case study. The thick red line indicates the contour $\{\hat{p}_{GPR} = 5\%\}$, dividing the state space $\mathcal{X}$ for $u = 0$ into admissible $\mathcal{X}^a(0)$ (left of red line) and inadmissible region $(\mathcal{X}^a(0))^c$ (right of red line).

### Logistic regression (LR)

In the previous section, we first created local batches to estimate $p(\mathbf{x}^i, u^i)$ pointwise and then regressed these estimates to build a global approximator. A classical alternative is to directly learn the probability of $G(x, u) > 0$ using a logistic regression model. This setup uses a single sample $g(\mathbf{x}^i, u^i)$ from $G(\mathbf{x}^i, u^i)$ from each design site $(\mathbf{x}^i, u^i)$ and transforms it to a binary variable $y^i = \mathbf{1}_{g(\mathbf{x}^i,u^i)>0}$. The probability $\hat{p}(\mathbf{x}, u)$ can then be directly modeled as a generalized linear model with a logit link function

$$\mathbb{P}\Big(Y = 1|\mathbf{x}, u\Big) = \frac{1}{1 + e^{-\boldsymbol{\beta}^T \phi(\mathbf{x},u)}} =: \hat{p}_{LR}(\mathbf{x}, u; \boldsymbol{\beta}). \tag{5.37}$$

The basis functions $\phi(\mathbf{x}, u)$ could be polynomials, e.g. quadratic or cubic in coordinates of $(\mathbf{x}, u)$. The regression coefficients $\boldsymbol{\beta}$ are fitted using the dataset $\{\mathbf{x}^i, u^i, \mathbf{y}^i\}_{i=1}^{M_s}$, as the solution to

$$\arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{M_s} \left\{ y^i \log p_{LR}(\mathbf{x}^i, u^i; \boldsymbol{\beta}) + (1 - y^i) \log(1 - p_{LR}(\mathbf{x}^i, u^i; \boldsymbol{\beta})) \right\}. \tag{5.38}$$

We may again create a more conservative estimate $\hat{\mathcal{U}}_{LR}^{(\rho)}(\mathbf{x})$ of $\hat{\mathcal{U}}_{LR}(\mathbf{x})$ at confidence level $\rho$ by utilizing the standard error for $\hat{p}_{LR}$ using the Delta method [77]:

$$\hat{\mathcal{U}}_{LR}^{(\rho)}(\mathbf{x}) := \left\{ u : \hat{p}_{LR}(\mathbf{x}, u, \boldsymbol{\beta}) \leq p - z_\rho \sqrt{\hat{p}_{LR}(\mathbf{x}, u)(1 - \hat{p}_{LR}(\mathbf{x}, u))\phi^T \mathrm{Var}(\boldsymbol{\beta})\phi} \right\}.$$

In Figure 5.2a, we present the original realizations $y^i \in \{0, 1\}$ (in blue) for a design in the input subspace $(L, I, u = 0)$ of the microgrid case study. The figure indicates the resulting logistic regression fit $\hat{p}_{LR}(L, I, 0)$ at levels 1%, 5% and 10% (i.e. contour lines of $\hat{p}_{LR}(\hat{\boldsymbol{\beta}}) \in \{0.01, 0.05, 0.1\}$). The admissibility set for $u = 0$, $\mathcal{X}_n^a(0)$ is the region to the left of the thick red contour.

**Remark 27** *Similar to INMC, we can simulate batched samples from each design site for the logistic regression, leading to "binomial" observation likelihood rather than the likelihood function in equation* (5.38)*.*

**Remark 28** *A non-parametric variant of equation* (5.37) *is kernel logistic regression, where the basis functions are $\phi_j(\mathbf{x}, u) = \kappa(\mathbf{x}, u, \mathbf{x}^j, u^j)$ for a kernel function $\kappa$ centered at $(\mathbf{x}^j, u^j)$. One common choice are radial basis functions (RBF) where $\kappa(\mathbf{x}, u, \mathbf{x}^j, u^j) = \exp\left(-\gamma_1 \|\mathbf{x} - \mathbf{x}^j\|_2^2 - \gamma_2 \|u - u^j\|_2^2\right)$. RBF can be interpreted as the squared-exponential kernel for a logistic Gaussian Process model, with a fixed bandwidth parameter $\gamma_i$. In contrast, in GPR the bandwidths are estimated through MLE.*

### Parametric density fitting (PF)

This approach aims to fit the distribution $G(\mathbf{x}, u)$, and then analytically infer the probability $\mathbb{P}\Big(G(\mathbf{x}, u) > 0\Big)$ from the corresponding cumulative distribution function. This is done by proposing a parametric family $\{f(\cdot; \Theta)\}$ of densities, fitting the underlying parameters $\Theta$ based on an empirical sample from $G$ and then evaluating the

resulting analytical probability $\bar{p}_{PF}(\mathbf{x}, u) := \int_0^\infty f_{G(\mathbf{x},u)}(z|\hat{\Theta}(\mathbf{x}, u))dz$. This approach yields a "universal" solution across a range of constraint levels $p$.

At a design site $(\mathbf{x}, u)$, the probability $p(\mathbf{x}, u)$ is estimated in a two-step procedure: first estimated locally over a design $\mathcal{D}^a = \{\mathbf{x}^i, u^i\}$ and then regressed/interpolated over the full input domain $\mathcal{X} \times \mathcal{W}$. For the first step, we apply nested Monte Carlo to generate a collection of realized $\{g^b(\mathbf{x}^i, u^i)\}_{b=1}^{M_b}$ that is used to construct a parametric density via the maximum likelihood estimate:

$$\hat{\Theta}^i := \arg\max_{\Theta} \sum_{b=1}^{M_b} \log f_G(g^b(\mathbf{x}_i, u_i)|\Theta). \tag{5.39}$$

In the second step, we evaluate $\tilde{p}_{PF}(\mathbf{x}^i, u^i) := \int_0^\infty f_G(z|\hat{\Theta}(\mathbf{x}^i, u^i))$ and extend it to the full domain $\mathcal{X} \times \mathcal{W}$ based on the computed $\{\mathbf{x}^i, u^i, \tilde{p}_{PF}(\mathbf{x}^i, u^i)\}_{i=1}^{M_a}$ using $\mathcal{L}_2$ projection:

$$\hat{p}_{PF} = \arg\min_{\hat{p} \in \mathcal{M}_T} \|\hat{p}(\mathbf{x}^i, u^i) - \tilde{p}_{PF}(\mathbf{x}^i, u^i)\|^2, \tag{5.40}$$

where $\mathcal{M}_T$ is an approximation space chosen for regression. The admissible set $\mathcal{U}(\mathbf{x})$ is estimated as:

$$\hat{\mathcal{U}}_{PF}(\mathbf{x}) := \{u : \hat{p}_{PF}(\mathbf{x}, u) \leq p\}.$$

A transformation of the distribution $G(\mathbf{x}, u)$ might be important for above distribution fitting. For example, in the context of microgrid, in Section 5.2.2, $G = \mathcal{L}\big(\sup_{s\in[t_n,t_{n+1})} S(s)\big)$ has a point mass at 0 and thus, any continuous distribution will lead to poor statistical estimation. Using a transformation that preserves the probability of the target event,

$$\mathbb{P}\Big(\sup_{s\in[t_n,t_{n+1})} S(s) > 0|\mathcal{F}_n\Big) = \mathbb{P}\left(\sup_{s\in[t_n,t_{n+1})} [L(s) - u_n - \frac{I(s)}{\delta s} \wedge B_{\max}] > 0 \,\Big|\mathcal{F}_n\right), \tag{5.41}$$

we work with $G'(L_n, I_n, u_n) := \mathcal{L}\big(\sup_{s\in[t_n,t_{n+1})}[L(s) - u_n - \frac{I(s)}{\delta s} \wedge B_{\max}]\big)$. In Figure 5.2c

we present the empirical and estimated probability $z \mapsto \mathbb{P}(G'(L_n, I_n, u_n) > z)$ when $L_n = 5.5, I_n = 1.48$ and $u_n \in \{0, 1\}$ for the microgrid example. We model the distribution $G'$ using a truncated normal distribution, $\mathbb{P}(G' \leq g) = \Phi(\frac{g-\theta_2}{\theta_3})\mathbb{1}_{g \geq \theta_1}$, with parameters $\Theta = (\theta_1, \theta_2, \theta_3)$ representing the location of censoring, the mean and the standard deviation respectively. At $L_n = 5.5, I_n = 1.48, u_n = 1.0$ and inner simulation budget $M_b = 100$, the estimated parameters $(\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3) = (-1.5, -1.12, 0.53)$ result in probability $\tilde{p}_{PF}(5.5, 1.48, 1.0) = 0.016$. The corresponding probability after $\mathcal{L}_2$ projection (equation (5.40)) is $\hat{p}_{PF}(5.5, 1.48, 1.0) = 0.017$. Thus at $p = 0.05$, the control $u = 1.0 \in \hat{\mathcal{U}}_n$ is admissible. However, at $u_n = 0$, $(\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3) = (-0.5, -0.12, 0.55)$, $\tilde{p}_{PF}(5.5, 1.48, 0.0) = 0.414$ and $\hat{p}_{PF}(5.5, 1.48, 0.0) = 0.429$, thus the control $u = 0 \notin \hat{\mathcal{U}}_n$ is inadmissible.

### 5.4.2 Quantile estimation

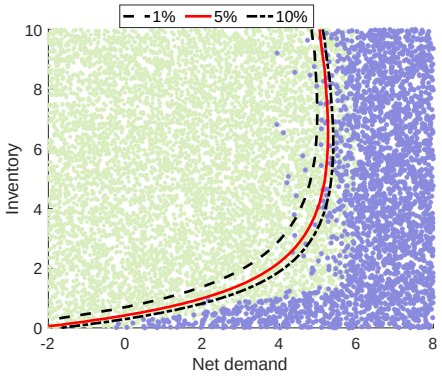
In this section we consider methods for modeling and estimating $q(\mathbf{x}^i, u^i)$, the $(1-p)$-th quantile of the distribution $G(\mathbf{x}^i, u^i)$. Admissibility corresponds to the quantile being negative.

#### Empirical percentiles (EP)

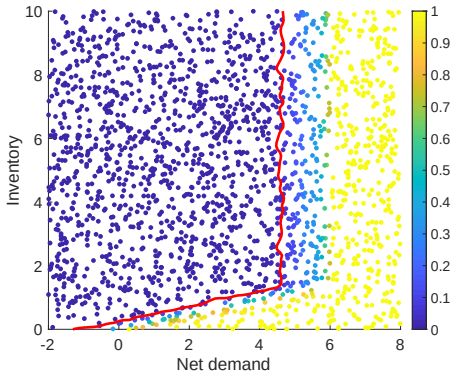
As before, we start by choosing $M_a$ design sites of state-action pairs and generate batched samples $\{g^b(\mathbf{x}^i, u^i)\}_{b=1}^{M_b}$ from each design site $(\mathbf{x}^i, u^i)$. The empirical estimate of $q(\mathbf{x}^i, u^i)$ is simply the $(1-p)^{th}$ percentile of the realized $\{g^b\}_{b=1}^{M_b}$ (which requires $M_b > p^{-1}$):

$$\bar{q}(\mathbf{x}^i, u^i) = \text{percentile}\left(\{g^b\}_{b=1}^{M_b}, 100(1-p)\%\right).$$

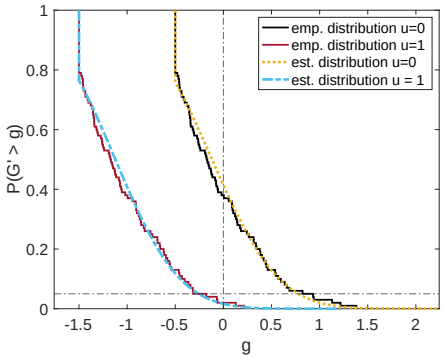
Similar to previous methods, we extend to arbitrary $(\mathbf{x}, u) \mapsto \hat{q}(\mathbf{x}, u)$ using regression on the dataset $\{\mathbf{x}^i, u^i, \bar{q}(\mathbf{x}^i, u^i)\}_{i=1}^{M_a}$ and an approximation space $\mathcal{M}_q$. The set of admissible




(a) LR ($M_a = 10000, M_b = 1$)




(b) GPR ($M_a = 2000, M_b = 50$)



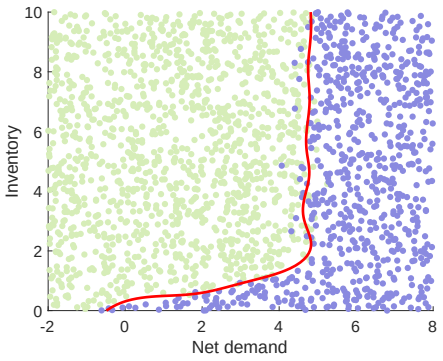

(c) PF ($M_b = 100$)

Figure 5.2: Training data and fitted models for the probability estimation methods of Section 5.4.1 at $u = 0$. *Top/left panel:* Training set $\{L^i, I^i, y^i\}_{i=1}^{M_a}$ for the LR model, color-coded according to the value of $y^i \in \{0, 1\}$, along with the estimated contours for $\hat{p}_{LR}(L, I)$ at levels $\{1\%, 5\%, 10\%\}$. *Top/right:* Training set $\{L^i, I^i, \bar{p}^i\}_{i=1}^{M_a}$ color-coded according to $\bar{p}^i$ for GPR along with the contour $\{\hat{p}_{GPR}(L, I) = 5\%\}$. *Bottom:* parametric density fitting at $L_0 = 5.5, I_0 = 1.48$ and $u \in \{0, 1\}$. We show the empirical and fitted inverse cdf $\mathbb{P}(G' > g)$ based on a truncated Gaussian distribution.

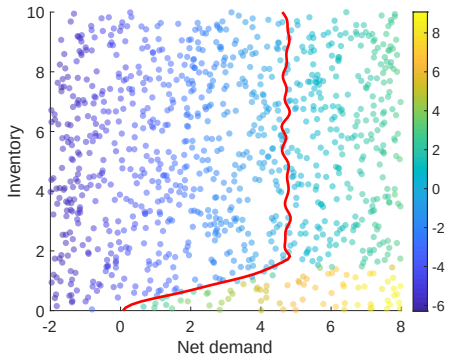
controls for $\mathbf{x}$ is:

$$\hat{\mathcal{U}}_{EP}(\mathbf{x}) := \Big\{u : \hat{q}(\mathbf{x}, u) < 0\Big\}.$$

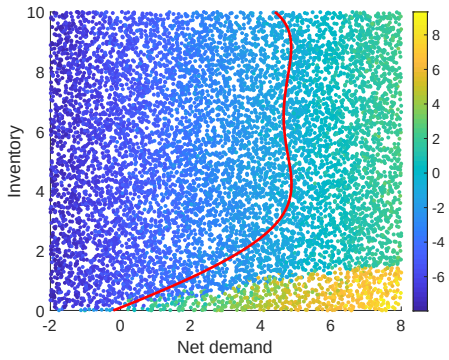
**Remark 29** *This approach is similar to the INMC approach discussed in Section 5.4.1, however, here we model the quantile rather than the probability of exceeding zero. Fur-*




(a) SVM ($M_a = 2000, M_b = 50$)




(b) EP ($M_a = 1000, M_b = 100$)




(c) QR ($M_a = 10000, M_b = 1$)

Figure 5.3: Training data and fitted models for the quantile estimation methods of Section 5.4.2 at $u = 0$. *Top/left panel:* Training set $\{L^i, I^i, y^i\}_{i=1}^{M_a}$ for SVM (color-coded according to $y^i \in \{-1, 1\}$) and the decision boundary in red. *Top/right:* Training set $\{L^i, I^i, \bar{q}^i\}_{i=1}^{M_a}$ color-coded according to $\bar{q}^i$ for EP and the contour $\{\hat{q} = 0\}$. *Bottom:* Training set $\{L^i, I^i, g^i\}_{i=1}^{M_a}$ color-coded according to $g^i$ for QR along with the contour $\{\hat{q}_{QR}(L, I) = 0\}$. All models share the same ground truth, so the red contours are identical up to model-specific estimation errors.

*thermore, we can use the regression standard error of $\hat{q}(\cdot, \cdot)$ to construct a more conservative estimate of the admissible set $\mathcal{U}_{EP}(\mathbf{x})$.*

In Figure 5.3b we show the estimated $\hat{q}(\cdot, \cdot, 0)$ indicated via the background colormap. The thick red line indicates the zero-contour $\hat{q} = 0$, so that the admissibility set for $u = 0$, $\mathcal{X}_n^a(0)$, is the region to the left of the contour.

A popular alternative to adjusting $\bar{q}$'s via regression standard errors is to replace the empirical percentile with the empirical conditional tail expectation (CTE):

$$\overline{\mathrm{CTE}}(\mathbf{x}^i, u^i) := \frac{\sum_{b=1}^{M_b} g^b \mathbf{1}_{g^b \geq \bar{q}(\mathbf{x}^i, u^i)}}{\sum_{b=1}^{M_b} \mathbf{1}_{g^b \geq \bar{q}(\mathbf{x}^i, u^i)}},$$
$$\hat{\mathcal{U}}_{CTE}(\mathbf{x}) := \big\{u : \widehat{\mathrm{CTE}}(\mathbf{x}, u) < 0\big\},$$

where $\widehat{\mathrm{CTE}}(\mathbf{x}, u)$ is the CTE surface fitted via a regression on the training set $(\mathbf{x}^i, u^i, \overline{\mathrm{CTE}}(\mathbf{x}^i, u^i))$. This idea is similar to regularizing the Value-at-Risk estimation with the Conditional VaR.

### Support Vector Machines (SVM)

For a fixed control $u$, finding the admissible set $\mathcal{X}_n^a(u)$ in (5.9) can be interpreted as classifying each input $\mathbf{x}$ as being in $\mathcal{X}_n^a(u)$ or not. Therefore, we consider the use of classification techniques, specifically support vector machines (SVM). This approach does not estimate the $(1-p)$-quantile $q(\mathbf{x}, u)$, but rather its 0-level set with respect to $(\mathbf{x}, u)$. The starting point is to use the nested Monte Carlo simulations to compute $\bar{p}(\mathbf{x}^i, u^i)$ with much smaller batch size $M_b$ compared to INMC. Next, we construct a binary classification objective with a training dataset $\{\mathbf{x}^i, u^i, y^i\}_{i=1}^{M_a}$ where the $\pm 1$-labels are

$$y^i := \begin{cases} 1, & \text{if } \bar{p}(\mathbf{x}^i, u^i) < p; \\ -1, & \text{otherwise.} \end{cases} \tag{5.42}$$

The boundary separating the two classes is evaluated by solving the optimization problem:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^K} \Big\{ \sum_{i=1}^{M_a} \Big(1 - y^i [\boldsymbol{\beta}^T \phi(\mathbf{x}^i, u^i) + \beta_0]\Big)_+ + \frac{C}{2 \cdot M_a} ||\boldsymbol{\beta}||^2 \Big\}, \tag{5.43}$$

where $\phi(\mathbf{x}, u) = \big[\phi_1(\mathbf{x}, u), \phi_2(\mathbf{x}, u), \ldots, \phi_K(\mathbf{x}, u)\big]^T$ are the $K$ basis functions and $C$ is the penalty parameter. We estimate the set of admissible controls corresponding to $\mathbf{x}$ as:

$$\hat{\mathcal{U}}_{SVM}(\mathbf{x}) := \Big\{u : \hat{\boldsymbol{\beta}}^T \phi(\mathbf{x}, u) + \hat{\beta}_0 > 0\Big\}.$$

Figure 5.3a displays the estimated $\hat{\mathcal{X}}_n^a(u)$ and the corresponding dataset $(L^i, I^i, 0, y^i)$ ($u = 0$ is fixed). The region where $u = 0$ is admissible is to the left of the decision boundary (represented through the thick red line).

**Remark 30** *A conservative estimate $\hat{\mathcal{U}}_{SVM}^{(\rho)}$ of $\hat{\mathcal{U}}_{SVM}$ is obtained by re-labeling the training points in* (5.42) *via:*

$$y^i = \begin{cases} 1, & \text{if } \bar{p}(\mathbf{x}^i, u^i) + z_\rho \sqrt{\frac{\bar{p}(\mathbf{x}^i, u^i)(1-\bar{p}(\mathbf{x}^i, u^i))}{M_b}} < p \\ -1, & \text{otherwise,} \end{cases} \tag{5.44}$$

*i.e. biasing the decision boundary to the left.*

### Quantile Regression (QR)

Quantile regression directly constructs a parametric model for $q(\mathbf{x}, u)$:

$$\hat{q}(\mathbf{x}, u; \boldsymbol{\beta}) := \sum_k \beta_k \phi_k(\mathbf{x}, u).$$

To estimate the coefficients $\boldsymbol{\beta} \in \mathbb{R}^K$, we use the dataset $\{\mathbf{x}^i, u^i, g^i\}_{i=1}^{M_a}$ (where $g^i$ is a sample from the distribution $G(\mathbf{x}^i, u^i)$) to maximize the negative log likelihood:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^K} \Big\{ \sum_{i=1}^{M_a} \mathcal{L}^{(p)}\Big(g^i - \sum_{k=1}^{K} \beta_k \phi_k(\mathbf{x}^i, u^i)\Big)\Big\},$$

$$\text{with} \quad \mathcal{L}^{(p)}(y) = y(p - 1_{\{y<0\}}) = p y_+ + (1-p) y_-.$$

As for the parametric density fitting, a transformation of $G(\mathbf{x}, u)$ might be beneficial when applying quantile regression. Figure 5.3c presents the dataset $\{L^i, I^i, 0, g^i\}_{i=1}^{M_a}$ in the background colormap and the estimated contour line $\{\hat{q}_{QR}(L, I) = 0\}$ with thick red line. The region to the left of the red line is the estimate of the admissible set $\hat{\mathcal{X}}^a(0)$.

Relying on the Delta method again to compute the variance of the estimated quantile $\hat{q}(\mathbf{x}, u; \hat{\boldsymbol{\beta}})$ as $\phi(\mathbf{x}, u)' Var(\hat{\boldsymbol{\beta}})\phi(\mathbf{x}, u)$, the admissible set at $\mathbf{x}$ at confidence level $\rho$ is:

$$\widehat{\mathcal{U}}_{QR}^{(\rho)}(\mathbf{x}) := \Big\{ u : \hat{q}(\mathbf{x}, u; \hat{\boldsymbol{\beta}}) + z_\rho \sqrt{\phi(\mathbf{x}, u)^T Var(\hat{\boldsymbol{\beta}})\phi(\mathbf{x}, u)} < 0 \Big\}.$$

## 5.5 Case Studies

Recall the problem introduced in Section 5.2.2 where we aim to control the operations of a diesel generator in order to supply power to match demand at minimal cost maintaining the probability of blackout between each decision epoch below a given threshold $p$. In this section, we will discuss two variants of such microgrid control. In the first example, we assume a time-homogeneous net-demand process which reduces the problem of estimating admissible set to a pre-processing step. In the second example, we use time-dependent net demand process calibrated to data obtained from a microgrid in Huatacondo, Chile. Time-inhomogeneity requires to estimate the admissible set at every step. The microgrid features a perfectly efficient battery directly connected to it, so that the respective power output at $t_{n_k}$ (recall $t_{n_k}$ is a generic time instance on the finely discretized time grid) is given by:

$$B_{n_k} = \frac{I_{n_k} - I_{\max}}{\Delta n_k} \vee \big( B_{\min} \vee (L_{n_k} - u_n) \wedge B_{\max} \big) \wedge \frac{I_{n_k}}{\Delta n_k}. \tag{5.45}$$

Table 5.1 lists other microgrid parameters, i.e. capacity of the battery $I_{\max}$, maximum charging rate $B_{\min}$, maximum discharging rate $B_{\max}$ and diesel switching cost $\mathcal{K}$.

| $I_{\max} = 10$ (kWh), $B_{\min} = -6, B_{\max} = 6$ (kW), $\mathcal{K} = 5$ |
|---|
| $T = 48$ (hours), $\Delta t = 0.25$ (hours) |

Table 5.1: Parameters for the Microgrid example.

### 5.5.1 Implementation details

**Numerical Gold Standard:** In the absence of analytic benchmark, we use empirical gold standard to compare the output from the models discussed in Section 5.4. For each fixed time-step $t_n$ we discretize the domain $\mathcal{X} = (L, I)$ into $10,000$ design sites over a grid of $100 \times 100$. For each design site $(L^i, I^j)$, $i, j \in \{1, \ldots, 100\}$ and $u^k \in 0 \cup \{1 = u_1, \ldots, u_{101} = 10\}$, we evaluate $\hat{p}(L^i, I^j, u^k)$ using (5.33) with batch size $M_b = 10,000$. Thus, the total simulation budget is $100 \times 100 \times 102 \times 10000 \approx 10^{10}$. We then evaluate the local minimal admissible control

$$u_n^{\min}(L^i, I^j) = \min\left\{u : \hat{p}(L^i, I^j, u) < p\right\}.$$

To evaluate $u_n^{\min}(L, I)$ at new sites we employ linear interpolation on the dataset $\{L^i, I^j,\ u_n^{\min}(L^i, I^j)\}_{i,j=1}^{100}$. Finally, to estimate the continuation function, we use piecewise continuous approximation of Section 5.3.2 with $M_I = 15, M_u = 15$ and 1500 sites in $L$.

**Low budget policies:** We approximate the continuation value function $\mathcal{C}$ using a piecewise continuous approximation with three degrees in $(L)$ combined with interpolation in other dimensions (with discretizations $M_I = 10, M_u = 10$). For the estimation of the admissible set $\mathcal{U}$, we approximate it using the methods described in Section 5.4. We discretize the control space $[1, 10]$ into 51 values. We compare the performance of each method by using a fixed set of $M' = 20,000$ out-of-sample simulations.

To address the discontinuity in $\mathcal{W} = 0 \cup [\underline{u}, \overline{u}]$, we implement two separate statistical

| Method | Budget ($M_a \times M_b$) | Further parameters |
|---|---|---|
| Gaussian Process (GPR) | $2000 \times 50$ | Matern-3/2 kernel |
| Logistic Regression (LR) | $10^5 \times 1$ | Degree-2 polynomials |
| Parametric Density Fitting (PF) | $2000 \times 50$ | Truncated Gaussian, Matern-3/2 kernel |
| Empirical Percentile (EP) | $1000 \times 100$ | Squared exponential kernel |
| Conditional Tail Expectation (CTE) | $1000 \times 100$ | Squared exponential kernel |
| Quantile Regression (QR) | $10^5 \times 1$ | Degree-4 polynomials |
| Support Vector Machine (SVM) | $2000 \times 50$ | C =1, RBF kernel |
| Gold Standard (GS) | $10^6 \times 10^4$ | budget = $10^{10}$ |

Table 5.2: Parameters for the estimation of the admissible sets for each method. We use total simulation budget of $10^5$ for all models except the Gold Standard.

models to learn $\mathcal{U}_n(\cdot)$. As an example, with logistic regression of Section 5.4.1 we estimate two sets of parameters in equation (5.37): the first one uses one-step paths generated from $u = 0$ and a two-dimensional regression of $y^{i,(1)}$ in terms of $(L^i, I^i)$. The second one uses design sites in the three-dimensional space $(L, I, u)$ where $u \in [1, 10]$ and a 3-D regression of $y^{i,(2)}$ against $(L^i, I^i, u^i)$.

Additional parameters used for each method are specified in Table 5.2. We found that Matern-3/2 kernels work better than (5.36) for smoothing $\bar{p}(L, I, u)$ (GPR) and $\tilde{p}(L, I, u)$ (PF) because the respective input-output maps feature steep transitions as a function of $(L, i, u)$.

It is known that "rougher" kernels are better suited for such learning tasks compared to the $C^\infty$-smooth squared exponential kernel (5.36) by allowing the fitted $\hat{p}$ to have more "wiggle room". On the other hand, in the context of EP and CTE the input observations of $\hat{q}(L, I, u)$ and $\overline{\text{CTE}}(L, I, u)$ are quite smooth in $(L, i, u)$ and both GP kernel families perform equally well.

The algorithms are implemented in `python 2.7`. We used "`GaussianProcessRegressor`" and "`SVM.SVC`" functions from `sklearn` library for GPR and SVM respectively. For LR and QR we used "`Logit`" and "`quantile_regression`" functions from `statsmodels` library.

### 5.5.2 Example 1: Microgrid with Stationary Net Demand

In this subsection, we assume time-homogeneous Ornstein-Uhlenbeck dynamics of the net demand process

$$dL(t) = -\lambda L(t)dt + \sigma \mathrm{d}B(t) \implies L(t) = L(0)e^{-\lambda t} + \sigma \int_0^t e^{-\lambda(t-s)} \mathrm{d}B(s), \qquad (5.46)$$

where $(B(t))$ is a standard Brownian motion. This scenario reduces the complexity of learning the probability constraints since we need to estimate the admissible set $\mathcal{U}_0(\cdot)$ only once as a pre-processing step before starting the ADP estimation scheme for the continuation values. The simplified setting offers a good testbed to evaluate the performance of different admissible set estimation methods of Section 5.4; we show that the relative performance remains similar as we extend to more realistic dynamics in Section 5.5.3. For this example, we assume the mean reversion parameter $\lambda = 0.5$ and volatility $\sigma = 2$.

Figure 5.4a plots the resulting costs $\hat{V}_0(0,5)$ versus the frequency of inadmissible decisions $w_{freq}$ for different methods of Section 5.4. We show the result both for the original setting of $p = 0.05$ (dark blue) as well as for $p = 0.01$ (light grey). In both cases we benchmark each scheme against the numerical gold standard (indicated by diamonds). Since the probabilistic constraints form the crux of the problem, we require schemes to maintain $\hat{u} \in \mathcal{U}_n$ as much as possible, i.e., $w_{freq} \approx 0$. At $p = 5\%$, we observe 0.09% , 0.54% and 1.36% estimated frequency of inadmissible decisions with logistic regression (LR), Gaussian process regression (GPR) and parametric density fitting (PF), respectively. The corresponding frequency jumps up to 5.9% for quantile regression (QR), 7.8% for conditional tail expectation (CTE), 8.4% for empirical percentiles (EP) and 5.3% for support vector machines (SVM). While all the methods are a priori consistent, admissible set estimation via probability-based methods clearly seems to outperform quantile-based ones. Our experiments suggest that at low simulation

budget, estimators of $p(\mathbf{x}, u)$ have significantly lower bias compared to estimators of $q(\mathbf{x}, u)$, thus partially explaining the difference. For a more conservative probability threshold $p = 1\%$, we find the cost of all the methods to increase, without significant difference in the frequency of inadmissible decisions $w_{freq}$. Indeed, Figure 5.4 illustrates the trade-off between lower costs and lower $w_{freq}$ (i.e. more conservative estimate of the constraints).

Table 5.3 expands Figure 5.4 by also reporting the corresponding $\tilde{\mathcal{T}}$ statistic, the average inadmissibility margin $w_{avm}$ and realized frequency of violations (i.e. blackouts) $w_{rlzd}$ defined as:

$$w_{avm} := \frac{1}{N \cdot M'} \sum_{n,m} |\hat{u}_n(\mathbf{x}_n^{\hat{u},m'}) - u_n^{\min}(\mathbf{x}_n^{\hat{u},m'})| \mathbb{1}_{\hat{u}_n(\mathbf{x}_n^{\hat{u},m'}) - u_n^{\min}(\mathbf{x}_n^{\hat{u},m'}) < 0}; \tag{5.47}$$

$$w_{rlzd} := \frac{1}{N \cdot M'} \sum_{n,m'} \mathbb{1}_{\sup_{s \in [t_n, t_{n+1})} S^{m'}(s) > 0}. \tag{5.48}$$

We find the realized frequency of violations $w_{rlzd}$ to be lowest for LR, GPR and PF. The average inadmissibility margin $w_{avm}$ is also lowest for GPR and PF (the large value of $w_{avm}$ for LR is attained in very small region as evident from $w_{freq} \approx 0$). The $\tilde{\mathcal{T}}$ statistic is negative for LR, GPR and PF and positive for the rest, meaning that all other methods fail to statistically respect the probability constraints when binding. Due to small frequency of inadmissible decisions $w_{freq}$, cost $\hat{V}_0(0, 5)$ similar to the numerical gold standard and negative test statistic $\tilde{\mathcal{T}}$, we recommend LR, GP and PF methods for the problem at hand.

Next, we test the sensitivity of the cost in terms of the probability threshold $p$ (employing logistic regression $\hat{\mathcal{U}}_{LR}$) in Figure 5.4b. Increasing the probability threshold $p$ decreases $V$ as the set of admissible controls $\mathcal{U}$ monotonically increases in $p$. For example, any admissible control at $p = 1\%$ threshold is also feasible for $p > 1\%$, thus the cost at 1% threshold should be greater than or equal to cost at, say, 10% threshold.

As previously discussed, the constraint is binding for only approximately 10% of time-steps. In fact, that probability varies across the methods which happens because the estimate of $\hat{\mathcal{U}}$ affects the choice of $\hat{u}_n$ and ultimately the *distribution* of $\hat{X}_n$. Intuitively, the realized system states are driven by the estimates of the probabilistic constraints. Typically, more conservative estimates of $\mathcal{U}$ will push $\hat{X}_{0:N}$ away from the "risky" regions. This is also confirmed in Figure 5.4 where as $p \to 1$, $w_{rlzd} \to 20\% = w_{bind}$ while in Table 5.3 $w_{bind} \simeq 10\%$.

The variables $w_{freq}$ (equation (5.28)), $w_{bind}$ (equation (5.32)), $w_{rlzd}$ (equation (5.48)) are closely related to each other. As the inadmissible decisions can occur only when the constraint is binding, $u^{\min} > 0$, we expect $w_{freq} \leq w_{bind}$ and $w_{freq} \approx w_{bind}$ for a method with a bias in overestimating the admissible set (e.g. $\mathcal{X}^{a,EP}(u) \supset \mathcal{X}^{a,GS}(u)\ \forall u \in \mathcal{W}$ ). The realized violations (blackouts) $w_{rlzd}$ can be represented as a sum of three:

$$w_{rlzd} = p_1 w_{freq} + p_2(w_{bind} - w_{freq}) + p_3(1 - w_{bind}), \qquad p_1 + p_2 + p_3 = 1,$$

where the weights $p_1, p_2, p_3$ depend on the distribution of the controlled trajectories. The first term represents the instances when the constraint is binding but the controller chooses an inadmissible control (i.e. mis-estimates $\hat{\mathcal{U}}$). The second term represents instances when the constraint is binding and correctly estimated, but due to random shocks violations take place (with a conditional frequency below the specified $p = 0.05$). The last term represents instances when the constraint is not binding but some violations still occur with the intrinsic conditional frequency strictly less than $p$. Note that due to $w_{bind} \ll 1$, most of the violations are of the latter type, i.e. take place when $u^* = 0$ and the conditional violation probability is below $p$. We illustrate these scenarios in Figure 5.4c using the LR model. The red triangles represent the $(L, I)$-location of realized violations, circles represent the locations of inadmissible decisions

| Method | $\hat{V}_0(0,5)$ (\$) | $w_{freq}$ (%) | $w_{avm}$ (kW) | $w_{rlzd}$ (%) | $\tilde{\mathcal{T}}$ | $w_{bind}$ (%) |
|---|---|---|---|---|---|---|
| GS | 26.79 | 0.00 | 0.00 | 0.37 | - | - |
| LR | 26.83 | 0.09 | 0.82 | 0.03 | -125 | 8.69 |
| GPR | 26.89 | 0.53 | 0.16 | 0.11 | -98 | 8.10 |
| PF | 26.79 | 1.36 | 0.27 | 0.21 | -69 | 8.51 |
| SVM | 26.68 | 5.26 | 0.55 | 1.83 | 388 | 9.67 |
| QR | 27.04 | 5.95 | 0.33 | 0.98 | 145 | 9.49 |
| CTE | 26.99 | 7.79 | 0.43 | 1.63 | 320 | 9.93 |
| EP | 26.36 | 8.39 | 0.49 | 1.98 | 403 | 10.45 |

Table 5.3: Cost of running the microgrid $\hat{V}_0(0,5)$, frequency of inadmissible decisions $w_{freq}$, average inadmissibility margin $w_{avm}$, realized frequency of violations (i.e. blackouts) $w_{rlzd}$, test statistic $\tilde{\mathcal{T}}$ and frequency of binding constraint $w_{bind}$ for the example in Section 5.5.2.

(with color representing the inadmissibility margin) and the grey region represents when the constraint is not binding. Thus, the first term counts the instances when violations occur at the same time as controller makes an inadmissible decision (circle encircling the triangle), the second term counts the triangles when $I \approx 0$, and the third term the triangles in the grey region (violations when $u^{\min} = 0$).

Although we observed poor performance of quantile based methods, asymptotically (with respect to the simulation budget) we expect them to perform similar to the probability based methods. As an example, in Table 5.4, we present the performance of SVM for thresholds $p = 5\%$ and $p = 1\%$ with increasing budget. For $p = 5\%$ and by increasing the simulation budget from $10^5$ to $10^8$, we find the frequency of inadmissible decisions $w_{freq}$ to drop from 5.93% to 1.5%, average inadmissibility amount $w_{avm}$ from 0.78 kW to 0.27 kW, frequency of realized blackouts $w_{rlzd}$ from 2.80% to 0.30% and the test statistic which rejected the method at $10^5$ simulation budget ($\mathcal{T} \gg 0$) suggests to accept it ($\mathcal{T} \ll 0$) at $10^8$ simulation budget. We observe similar behavior at $p = 1\%$.

**Conservative estimators for $\mathcal{U}$.** Algorithms for SCPC are expected to respect the probabilistic constraints, so that it is critical to minimize the occurrence of inadmissible

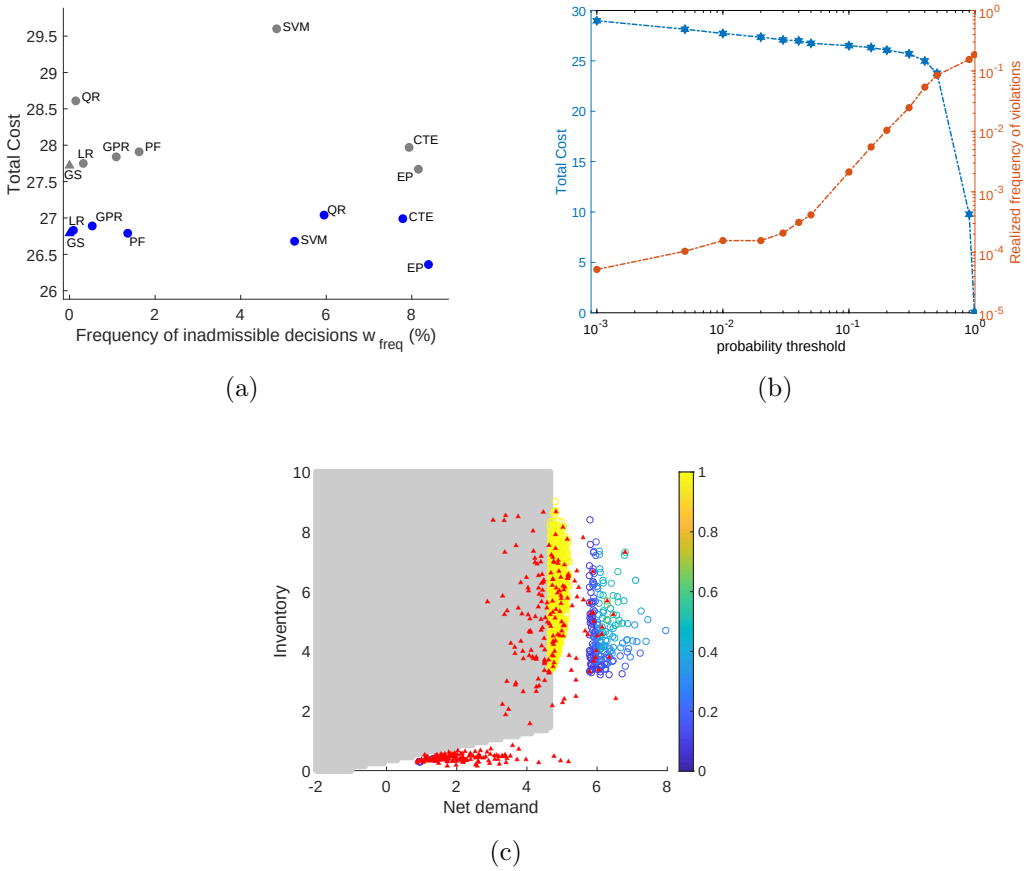



Figure 5.4: *Top/left panel:* Trade-off between cost $\hat{V}_0(0,5)$ and frequency of inadmissible decisions $w_{freq}$ for the stationary model. Dark (blue colored) dots correspond to $p = 5\%$ probability constraint threshold and light (grey colored) dots to $p = 1\%$. *Top/right panel:* Total cost $\hat{V}_0(0,5)$ (left axis, line with stars) and realized frequency of violations $w_{rlzd}$ (right axis, line with circles) as functions of $p$ employing the LR model. *Bottom panel:* Locations $(L, I)$ of realized violations $\sup_{s\in[t_n,t_{n+1})} S^{m'}(s) > 0$ (red triangles), inadmissible decisions $\hat{u}(n, \mathbf{x}_n^{\hat{u},m'}) - u_n^{\min}(\mathbf{x}_n^{\hat{u},m'}) < 0$ (circles with color representing the inadmissibility margin) on 5000 out-of-sample simulations using LR model. The constraint is binding in the white region and is not binding in the grey region.

decisions. As discussed in Section 5.4, one way to raise the statistical guarantee for admissibility of all controls in $\hat{\mathcal{U}}$ is by adding a margin of error $\xi(\mathbf{x}, u)$. The margin of error yields a more conservative (i.e. smaller) $\hat{\mathcal{U}}$ and hence lowers $w_{freq}$. In Table 5.5 we examine three scenarios for $\xi(\mathbf{x}, u)$ sorted from least to most conservative (in all cases

| $p$ | Budget | $\hat{V}_0(0,5)$ (\$) | $w_{freq}$ (%) | $w_{avm}$ (kW) | $w_{rlzd}$ (%) | $\tilde{\mathcal{T}}$ | $w_{bind}$ (%) |
|---|---|---|---|---|---|---|---|
| 5% | $10^5$ | 26.38 | 5.93 | 0.78 | 2.80 | 665 | 9.73 |
| | $10^6$ | 26.55 | 5.28 | 0.55 | 1.84 | 386 | 9.77 |
| | $10^7$ | 26.68 | 4.96 | 0.53 | 1.64 | 330 | 9.75 |
| | $10^8$ | 26.79 | 1.50 | 0.27 | 0.30 | -51 | 9.22 |
| 1% | $10^5$ | 28.32 | 6.63 | 0.93 | 2.43 | 1,460 | 9.87 |
| | $10^6$ | 28.26 | 5.17 | 0.66 | 1.09 | 631 | 9.56 |
| | $10^7$ | 28.52 | 0.55 | 0.24 | 0.03 | -39 | 8.78 |
| | $10^8$ | 28.41 | 0.15 | 0.22 | 0.01 | -51 | 8.82 |

Table 5.4: Impact of simulation budget ($M_b \times M_a$) on performance of SVM for the case study in Section 5.5.2 and probability thresholds $p = 5\%$ and $p = 1\%$. The reported values are averages over 10 runs of each scheme. The total simulation budget is divided into batch size $M_b$ and number of design sites $M_a$. For total budget $10^5$: $(M_b, M_a) = (100, 1000)$; for $10^6$: $(M_b, M_a) = (500, 2000)$; for $10^7$: $(M_b, M_a) = (2000, 5000)$; for $10^8$: $(M_b, M_a) = (10000, 10000)$.

we keep the probability constraint at $p = 5\%$):

- Scenario 1: unadjusted $\xi = 0\%$ (same as Table 5.3);
- Scenario 2: $\xi^{(\rho)}(\mathbf{x}, u)$ at 95% confidence level, $z_\rho = 1.96$;
- Scenario 3: fixed $\xi = 4\%$, which is equivalent to lowering the violation threshold to $p - \xi = 1\%$.

Table 5.5 confirms the intuition that the frequency of inadmissible decisions $w_{freq}$ should be decreasing from scenario 1 to 3. This is further illustrated in Figure 5.5 that shows how the minimum admissible control is affected by $\xi(\mathbf{x}, u)$. Although adding a margin of error does lower $w_{freq}$ we note that this mechanism does not really alter the relative performance of the different methods. Thus, for all three scenarios, we find SVM, CTE and EP to be performing poorly (unreliable estimation of $\mathcal{U}$ since $\tilde{\mathcal{T}} \gg 0$). An exception is QR which yields high $w_{freq}$ for $\xi = 0$ but does become acceptable ($\tilde{\mathcal{T}} < 0$) in scenario 3. In contrast, LR, GPR and PF perform well throughout. Table 5.6 lists further comparison as we set the confidence level to $\rho = 90\%, 99\%$ and $99.95\%$, with

| | $\xi = 0\%$ | | | $\xi^{(0.95)}(\mathbf{x}, u)$ | | | $\xi = 4\%$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | $\hat{V}_0(0,5)$ | $w_{freq}$ | $\tilde{\mathcal{T}}$ | $\hat{V}_0(0,5)$ | $w_{freq}$ | $\tilde{\mathcal{T}}$ | $\hat{V}_0(0,5)$ | $w_{freq}$ | $\tilde{\mathcal{T}}$ |
| GS | 26.79 | 0.00 | - | - | - | - | - | - | - |
| LR | 26.83 | 0.09 | -125 | 26.95 | 0.08 | -124 | 27.86 | 0.04 | -112 |
| GPR | 26.89 | 0.53 | -98 | 28.00 | 0.01 | -110 | 28.12 | 0.00 | -107 |
| PF | 26.79 | 1.36 | -69 | - | - | - | 27.91 | 0.44 | -96 |
| SVM | 26.68 | 5.26 | 388 | 29.65 | 3.41 | 225 | 29.60 | 3.41 | 225 |
| QR | 27.04 | 5.95 | 145 | 26.89 | 5.17 | 72 | 28.61 | 0.00 | -117 |
| CTE | 26.99 | 7.79 | 320 | 27.36 | 7.52 | 274 | 28.44 | 6.83 | 248 |
| ER | 26.36 | 8.39 | 403 | 26.97 | 7.78 | 225 | 28.13 | 7.08 | 283 |

Table 5.5: Impact of margin of error $\xi$ on the estimated cost of running the microgrid $\hat{V}_0(0,5)$, frequency of inadmissible decisions $w_{freq}$, and test statistic $\tilde{\mathcal{T}}$ from (5.31). The probabilistic constraint is $p = 5\%$.

the same general conclusions. (Observe that driving $w_{freq}$ all the way to zero might be non-ideal since it likely implies that $\hat{\mathcal{U}} \subset \mathcal{U}$ is strictly smaller so the controller is overly conservative and rules out some admissible actions.)

We generally expect the ultimate cost $\hat{V}_0(0,5)$ to increase as $\hat{\mathcal{U}}$ becomes more conservative, see the estimated $\hat{V}$'s across each row of Table 5.5. The increase in costs arises due to two factors: when the diesel generator is started sooner (due to $u = 0$ becoming inadmissible as $\xi$ is raised) and the higher level of $\hat{u}$ once the diesel is ON. This can be seen in Figure 5.5 where in Scenarios 2 and 3 the controller switches the diesel generator at a lower net demand and once the diesel is running picks a higher power output ($\hat{u}^{\min}(\cdot, I; p = 5\%, \xi) - \hat{u}^{\min}(\cdot, I; p = 5\%, \xi = 0) > 0$). It is important to note however that the link between $\hat{\mathcal{U}}$ and $\hat{V}$ is complicated by the fact that as $\hat{\mathcal{U}}$ changes, so does the distribution of the controlled paths. So for example in Table 5.5 the cost for QR falls in Scenario 2, although it remains within two Monte Carlo standard errors.

**Take-aways.** Our experiments demonstrate the following: (i) Admissible sets of the form (5.5) are more accurately estimated via LR, GPR and PF which all model the underlying probability of violations $p(\mathbf{x}, u)$. Although asymptotically equivalent,

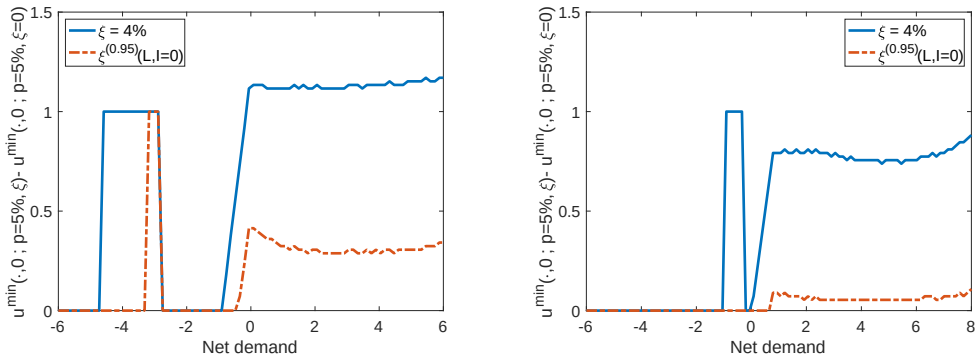
| | $\rho = 90\%$ | | | $\rho = 99\%$ | | | $\rho = 99.95\%$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | $\hat{V}_0(0,5)$ | $w_{freq}$ | $w_{rlzd}$ | $\hat{V}_0(0,5)$ | $w_{freq}$ | $w_{rlzd}$ | $\hat{V}_0(0,5)$ | $w_{freq}$ | $w_{rlzd}$ |
| LR | 26.74 | 0.090 | 0.034 | 26.87 | 0.085 | 0.032 | 27.04 | 0.085 | 0.026 |
| GPR | 27.34 | 0.012 | 0.055 | 28.06 | 0.007 | 0.037 | 28.06 | 0.005 | 0.029 |
| SVM | 27.35 | 4.975 | 1.732 | 29.20 | 3.481 | 1.117 | 29.72 | 3.395 | 1.088 |
| QR | 27.20 | 5.373 | 0.793 | 27.18 | 4.880 | 0.676 | 27.04 | 4.409 | 0.591 |
| CTE | 27.93 | 7.158 | 1.153 | 28.31 | 6.766 | 0.888 | 28.61 | 6.163 | 0.714 |
| EP | 26.78 | 7.990 | 1.497 | 27.17 | 7.629 | 1.183 | 27.96 | 7.102 | 0.956 |

Table 5.6: Impact of conservative $\mathcal{U}^{(\rho)}$ estimators for the case study in Section 5.5.2. The probabilistic constraint is set at $p = 5\%$.

the approach of quantile estimation leads to poor estimation of the admissible sets for practical budgets. Thus LR, GPR and PF are our recommended choices. (ii) Frequency of inadmissible decisions can be controlled by using a more conservative estimate of the admissible sets. Such conservatism will tend to raise costs. We find that even a conservative $\hat{\mathcal{U}}^{\xi}$ fails to make quantile-based methods acceptable, except for QR. (iii) For a new application, our suggested approach is to first evaluate the test statistic $\tilde{\mathcal{T}}$ at $\xi = 0\%$ using one of the recommended methods. Depending on how close is $\tilde{\mathcal{T}}$ to zero, one can then adjust $\hat{\mathcal{U}}$'s by adding in $\xi$ (or $\xi^{(\rho)}$) to improve the statistical guarantees on the frequency of inadmissible decisions $w_{freq}$.

### 5.5.3 Example 2: Microgrid with seasonal demand

Unlike the previous example, where we assumed time-homogeneous net demand, in practice there is seasonality: during the day renewable generation is high and net demand is often negative; during morning/evening demand exceeds supply making $L(t) > 0$. To incorporate this seasonality we use time-dependent Ornstein Uhlenbeck




(a) LR (b) QR

Figure 5.5: Impact of the margin of error $\xi(\cdot,\cdot)$ on minimum admissible control $\hat{u}^{\min}$. We plot the difference between minimum admissible control for scenario 2 ($\hat{u}^{\min}(\cdot, I; \xi^{(0.95)}(L, I))$) and scenario 3 ($\hat{u}^{\min}(\cdot, I; \xi = 4\%)$) with respect to scenario 1 ($\hat{u}^{\min}(\cdot, I; \xi = 0\%)$) using LR (left panel) and QR (right panel) models.

process (see [4] for a similar microgrid control problem):

$$dL(t) = \left[\frac{\partial \mu}{\partial t}(t) + \lambda\big(\mu(t) - L(t)\big)\right] dt + \sigma(t)\mathrm{d}B(t). \tag{5.49}$$

Here, $\lambda$ represents the speed of mean reversion towards the seasonal mean $\mu(t)$, while $\sigma(t)$ represents the time-varying volatility. Using the transformation $L(t)e^{\lambda t}$ followed by Ito's lemma and integration by parts one can prove that

$$L(t) = \mu(t) + e^{-\lambda t}\big(L(0) - \mu(0)\big) + \int_0^t e^{-\lambda(t-s)}\sigma(s)\mathrm{d}B(s).$$

Thus,

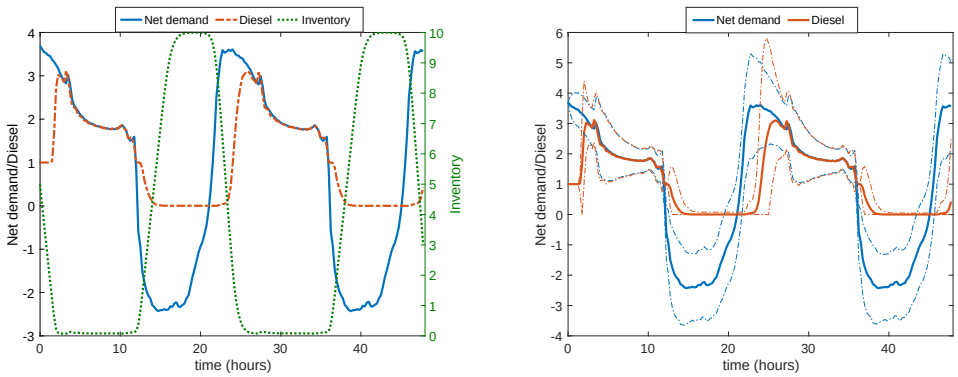
$$\mathbb{E}[L(t)] = \mu(t) + e^{-\lambda t}(L(0) - \mu(0)).$$

We calibrate $\mu(t)$ and $\sigma(t)$ in (5.49) using iterative methodology described in [4] and the data from a solar-powered microgrid in Huatacondo, Chile. Specifically, we compute the mean and variance of the net demand over 24 hours at 15-minute intervals using data

from Spring 2014, i.e. compute $\{\mu_1, \mu_2, \ldots, \mu_{96}\}$ and $\{\sigma_1, \sigma_2, \ldots, \sigma_{96}\}$. The estimated $\mu(t)$ can be seen in the left panel of Figure 5.6 that plots the empirical average of $L(t)$. As expected, during the day, i.e., $t \in [12, 20]$ (noon-8:00 pm), the expected net-demand is negative ($\mu(t) < 0$) while it is positive ($\mu(t) > 0$) in the morning and during the night. The volatility $\sigma(t)$ is higher during the day due to the intermittent and unpredictable nature of solar irradiance. The mean reversion parameter was estimated to be $\lambda = 0.3416$.

To visualize the interplay of the net demand, inventory and optimal control, the left panel of Figure 5.6 presents the average trajectories of the three processes over 48 hours. During the morning hours when the demand $L(t)$ is high and the battery is empty, the controller uses the diesel generator. Similarly, during the day when the renewable output is high and $L(t)$ is negative, the controller switches off the diesel generator and the battery charges itself. However, the non-trivial region is when the average net-demand changes sign, either from positive to negative around noon or negative to positive in the evening. During the former time-interval, the optimal control process is in $\{0, 1\}$ (recall that minimum diesel output is 1). Similarly, during the evening when the net demand becomes positive (as the renewable output declines), the controller quickly ramps up the diesel output to match $L(t) \gg 0$. The right panel of Figure 5.6 repeats the average control and inventory curves, but also shows their 2-standard deviation bands (in terms of the out-of-sample trajectories of $\hat{L}^{\hat{u}}_{0:T}$). As expected, the time periods around ramp-up or ramp-down of the diesel generator is when $\hat{u}_n$ experiences the greatest path-dependency and dispersion and differs most from the demand curve.

Comparing Table 5.7, which lists the estimated cost $\hat{V}_0(\mu(0), 5)$ along with related statistics, with Figure 5.4 indicates that incorporating seasonal net-demand process does not change the relative order of performance between the methods. The cost goes up as the diesel generator has to be used throughout the mornings and the evenings to match




(a) (b)

Figure 5.6: Model parameters, average trajectory of the state variables, control and their variance. Left panel: Average values of net demand $\frac{1}{M'}\sum_{m'=1}^{M'} L_n^{\hat{u},m'}$, inventory $\frac{1}{M'}\sum_{m'=1}^{M'} I_n^{\hat{u},m'}$ and optimal control (diesel) $\frac{1}{M'}\sum_{m'=1}^{M'} \hat{u}_n^{m'}$ processes using the gold standard strategy. Right panel: 95% confidence bands for net demand $L_n^{\hat{u}}$ and realized optimal diesel control $\hat{u}_n$.

demand.

As in the previous example, the performance of LR, GPR and PF almost matches the gold standard despite significantly lower simulation budget. In this setting the constraint is binding approximately 45% of the time (except for GPR and PF where it is 30% and 25% of the time). Frequency of inadmissible decisions $w_{freq}$ is 0.03% for LR, 1.17% for GPR, and 0.02% for PF. In contrast $w_{freq}$ is 43% for QR, 22% for EP, 43% for SVM and 22% for CTE, implying that all these schemes are highly unreliable for learning $\hat{\mathcal{U}}$. The average inadmissibility margin $w_{avm}$ is also significantly lower for GPR (0.14 kW) and PF (0.26 kW) compared to the rest of the methods. Here again we observe larger inadmissibility margin and very low frequency of inadmissible decisions for logistic regression. Similar behavior is also evident for the test statistic $\tilde{\mathcal{T}}$ and realized frequency of violations $w_{rlzd}$.

To illustrate the typical behavior over a trajectory, Figure 5.7 plots the average control $Ave(\hat{u}_n) := \frac{1}{M'}\sum_{m=1}^{M'} \hat{u}_n(\mathbf{x}_n^{\hat{u},m'})$ corresponding to different methods and the average

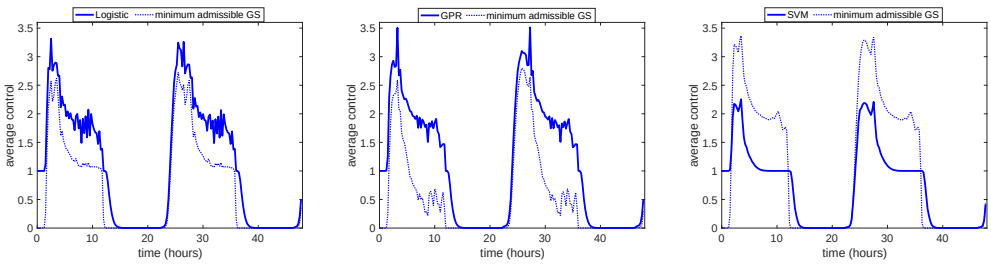
| Method | $\hat{V}_0(\mu(0), 5)$ | $w_{freq}$ (%) | $w_{avm}$ (kW) | $w_{rlzd}$ (%) | $\tilde{\mathcal{T}}$ | $w_{bind}$ (%) |
|---|---|---|---|---|---|---|
| GS | 53.38 | 0 | 0 | 0.30 | - | - |
| LR | 53.78 | 0.03 | 0.79 | 0.01 | -301 | 45.2 |
| GPR | 54.04 | 1.17 | 0.14 | 0.19 | -220 | 31.0 |
| PF | 54.55 | 0.02 | 0.26 | 0.01 | -226 | 25.7 |
| SVM | 40.52 | 43.37 | 0.91 | 43.37 | 5,306 | 46.4 |
| QR | 52.56 | 42.87 | 0.28 | 38.41 | 4,772 | 46.3 |
| CTE | 53.02 | 21.62 | 0.21 | 10.43 | 1,079 | 46.0 |
| EP | 52.82 | 21.91 | 0.23 | 11.57 | 1,227 | 46.1 |

Table 5.7: Cost of running the microgrid $\hat{V}_0(\mu(0), 5)$, frequency of inadmissible decisions $w_{freq}$, average inadmissibility margin $w_{avm}$, realized frequency of violations $w_{rlzd}$ and frequency of the constraint being binding $w_{bind}$ for the case study in Section 5.5.3.

minimum admissible control $Ave(u_n^{\min}) := \frac{1}{M'} \sum_{m'=1}^{M'} u_n^{\min}(\mathbf{x}_n^{\hat{u},m'})$ computed using the gold standard. Notice that the latter is dependent upon the controlled trajectories $\mathbf{x}_n^{\hat{u}}$ derived for each method, resulting in a different trajectory of $Ave(u_n^{\min})$ across methods. We expect $Ave(\hat{u}_n)$ above $Ave(u_n^{\min})$ if a given method does not violate the constraint most of the time. This is true for LR and GPR, but SVM quite obviously fails, as the dashed line (Figure 5.7c) is significantly higher than the solid line at numerous time steps. Furthermore, the conservative nature of GPR is also evident via the large difference between the average minimum admissible control and the average optimal control. This is also evident through $w_{bind} \approx 30\%$ for GPR compared to approximately 45% for the rest of the methods.

## 5.6 Summary

We developed a statistical learning framework to solve stochastic optimal control with local probabilistic constraints. The key objective of our algorithm is to efficiently estimate the set of admissible controls $\mathcal{U}(\cdot)$ and the continuation value function $\mathcal{C}(\cdot, \cdot)$




(a) (b) (c)

Figure 5.7: Average control $Ave(\hat{u}_n)$ for LR, GPR and SVM and the average minimum admissible control $Ave(u_n^{\min})$ using Gold Standard across forward controlled trajectories.

covering a general formulation regarding the state process dynamics and rewards. Since stochastic control problems require estimating the admissible set repeatedly during the backward induction, we use regression based functional representation of $\mathbf{x} \mapsto \mathcal{U}(\mathbf{x})$. This perspective also provides a natural way of uncertainty quantification for admissibility, in particular offering conservative estimates that bring statistical guarantees regarding $\hat{\mathcal{U}}$. At the same time, our dynamic emulation algorithm allows parallel computation of $\mathcal{U}$ and $\mathcal{C}$ for additional computational efficiency.

Thanks to the plug-and-play functionality of the PC-DEA, it was straightforward to test a large variety of schemes for learning $\mathcal{U}$. Our numerical results suggest that estimating probabilistic constraints via logistic regression, Gaussian process smoothing and parametric density fitting is more accurate than estimating the corresponding quantile (empirical ranking, SVM or quantile regression). A future line of research would be to additionally parametrize (e.g. using another GP model) the optimal control map $\mathbf{x} \mapsto \hat{u}_n(\mathbf{x})$ [75] which would speed-up the algorithm in the context of continuous action spaces.

# Chapter 6

# Impact of Electricity Tariffs on Distribution Network Reliability with Behind-the-meter Investments

*This chapter is the result of a collaboration with Miguel Heleno and Michael Ludkovski. It is based on the paper [14].*

Electricity rates are a main driver for adoption of DERs by private consumers and, consequently, they will impact the reliability of energy access in the long run. Defining reliability indices in a paradigm where energy is generated both behind and in front of the meter is part of an ongoing discussion about the future role of utilities and system operators with many regulatory implications. This paper contributes to that discussion by analyzing the effect of rate design on the long term reliability of power distribution. A methodology to quantify this effect is proposed and a case study involving PV and storage adoption in California is presented.

## 6.1 Introduction

To achieve emission targets, countries need to increase generation share from renewable energy sources, not only as part of the bulk generation system but also at the level of the distribution network [78], where private owned Photovoltaic (PV) systems installed behind the meter and coupled with electric storage and control technologies have been seen as an efficient way to increase renewables penetration in a decentralized form. Ambitious policy targets have been announced to promote the adoption of Distributed Energy Resources (DERs) by private consumers. For example, the state of California added a new amendment to the Building Energy Efficiency standard, requiring new residential buildings to have a rooftop PV unit installed, starting in 2020 [79]. Undoubtedly, this type policy orientations promoting PV in new buildings will continue bringing solar technology costs down, creating conditions for the mass adoption of PV and storage systems.

Besides the cost of technologies, another main driver for the behind the meter adoption of DERs comes from the electricity tariffs. In fact, the magnitude and structure of the electricity rates - including demand charges, energy costs and PV feed-in remuneration - strongly affect the internal rate of return of PV and storage investments, changing adoption decisions of private consumers. Hence, recent works have been analyzing the impact of electricity rate design on the adoption of DERs. For example, the authors of [80] discuss tariff cost allocation approaches for networks with large penetration of distributed generation, while [81] and [82] propose new rate design mechanisms to facilitate the integration of DERs. The dynamics between PV adoption and retail electricity rates are modeled in [83]. The authors of [84] develop a socio-economic model for PV adoption at the residential level that takes feed-in tariffs as an input.

The increasing penetration of DERs, both behind and in front of the meter, can affect the reliability of the distribution system and new methodologies have been developed

to evaluate its positive and negative impacts [85] [86]. Recent studies have been looking at DERs operation as a resource to improve reliability of the systems. This is the case of storage (both grid-connected and in EVs) and demand response (DR). For example, the potential of battery stations and EV parking lots to enhance reliability indices of the distribution network is evaluated in [87] and [88], respectively. Similarly, in residential feeders, the role of vehicle-to-home and vehicle-to-grid capabilities to improve reliability is presented in [89]. The authors of [90] discuss centralized and decentralized approaches for outage management in distribution grids when DERs are organized in microgrids. A real-case scenario illustrating the benefits of DR to the reliability of the distribution network is presented in [91]. To take advantage of these benefits, the authors of [92] propose a methodology allowing utilities to use DR programs to improve reliability metrics and increase the expected return in performance-based regulatory frameworks. In [93], time-differentiated prices are offered to end-users to incentivize DR behaviour that improves system reliability. In that model, DR participation is defined by an optimal residential energy management strategy considering customer satisfaction.

In this chapter, we develop a framework to assess the reliability of a distribution network with behind-the-meter investments in DERs and test its sensitivity to the magnitude and structure of electricity tariffs. Specifically, we assume that the consumers make private investment in DERs to minimize the long run economic cost of purchase of electricity. The electricity rates are an input to the optimization problem of the consumers, driving the size of investments and the operational policy of the DERs, and consequently the reliability. The focus of this study is thus to understand the link between the electricity tariff structure and reliability, quantified via three indices: average energy not supplied (AENS), average energy not consumed (AENC) and system average interruption duration index (SAIDI). We evaluate several different aspects of electricity tariffs that can affect the reliability; impact of homothetic change in the

purchase rate, change in the magnitude of the peak purchase rate, and the time-of-day of peak purchase rate.

Our results clearly indicate that the choice of electricity tariffs has a significant effect on the reliability. We find non-obvious tariff options, arising due to the combination of storage policy, PV generation and load profile, where the AENC is lowest. For example, having a peak tariff time of 8:00 am-1:00 pm for the residential consumers leads to AENC to be lower than the standard peak times of noon-5:00 pm or 4:00 pm-9:00 pm. The results also indicate that for improving reliability, increasing just the peak purchase rate is a more cost efficient alternative than homothetic change in the purchase rate. We also find that different times of the peak rate for different consumers lead to better reliability compared to the same time of peak rates for all consumers, confirming the current market practice.

We have organized the rest of this chapter in five sections. Section 6.2 describes the optimization framework for the optimal investments and control policy for PV and storage. Section 6.3 describes the distribution system, its representation, Monte Carlo simulation for simulation of system states and the storage model during failure. Section 6.4 describes the computation of the reliability indices. Section 6.5 contains the numerical example using PG&E 69 bus network. Section 6.6 concludes.

## 6.2 Behind-the-meter Investments and Optimal Control

In this section we describe the framework to compute optimal behind-the-meter investment in DERs by a consumer. We start by assuming that the consumer can invest in PV, storage, or purchase power from the utility to meet her demand for electric power. We further assume that any excess power generated from PV can also be exported

back to the utility with feed-in remuneration (or export rate) set by the latter. Given these assumptions, along with the electricity tariffs set by the utility and the economic parameters for the DERs, the objective for the consumer is to find optimal investments in PV and/or storage and their operational policy. The consumer formulates this as an optimization problem which minimizes the sum of fixed/variable costs for investments in PV and storage, operational and maintenance costs for PV, and power purchases/sales from the utility. The cost function is given as:

$$\begin{aligned} C = & \sum_{k \in \{s,pv\}} \Big[ (\mathrm{CFix_k} \cdot \mathrm{pur_k} + \mathrm{CVar_k} \cdot \mathrm{Cap_k}) \mathrm{Ann_k} + \mathrm{Cap_k} \cdot \mathrm{DERMFx_k} \Big] \\ & + \sum_{t \in \mathcal{T}_{yr}} \mathrm{PGen_t} (\mathrm{DERGnCst_{pv}} + \mathrm{DERMVr_{pv}}) \\ & + \sum_{t \in \mathcal{T}_{yr}} (\mathrm{UtilPur_t} \cdot \mathrm{PurRt_t} - \mathrm{UtilExp_t} \cdot \mathrm{ExpRt_t}), \end{aligned} \tag{6.1}$$

where $\mathcal{T}_{yr} := \{0, 1, \ldots, 8760\}$ represents the hourly time-steps for the year. The optimization variables are (i) binary decision variable for purchase of battery and PV, $\mathrm{pur_k},\ k \in \{s, pv\}$; (ii) capacity installed $\mathrm{Cap_k},\ k \in \{s, pv\}$ for each technology; (iii) binary electricity purchase/sell decision $\mathrm{psb_t}$; (iv) amount of utility power purchased or exported at each time step $\mathrm{UtilPur_t}, \mathrm{UtilExp_t}$; (v) PV generation at every time step $\mathrm{Gen_{pv,t}}$; (vi) energy input and output to the battery $\mathrm{SIn_t}, \mathrm{SOut_t}$. The optimization of $C$ is subject to the following constraints.

**Storage Constraints:**

$$\mathrm{SOC_t} = \mathrm{SOC_{t-1}} + \mathrm{SIn_t} \cdot \mathrm{SCEff} - \frac{\mathrm{SOut_t}}{\mathrm{SDEff}}; \tag{6.2}$$

$$\underline{\mathrm{SOC}} \leq \mathrm{SOC_t} \leq \overline{\mathrm{SOC}}; \tag{6.3}$$

$$\mathrm{SIn_t} \leq \mathrm{Cap_s} \cdot \overline{\mathrm{SCRt}}; \tag{6.4}$$

$$\mathrm{SOut_t} \leq \mathrm{Cap_s} \cdot \overline{\mathrm{SDRt}}; \tag{6.5}$$

$$\mathrm{SCap} \leq \mathrm{pur_s} \cdot \mathrm{M}. \tag{6.6}$$

**PV Constraints:**

$$\mathrm{PGen_t} = \mathrm{Cap_{pv}} \frac{\mathrm{SolEff_t}}{\mathrm{ScPkff}} \mathrm{Solar_t} \tag{6.7}$$

$$\mathrm{Cap_{pv}} \leq \mathrm{pur_{pv}} \cdot \mathrm{M} \tag{6.8}$$

$$\mathrm{PGen_t} \leq \mathrm{Cap_{pv}}. \tag{6.9}$$

**Import and Export Constraints:**

$$\mathrm{UtilPur_t} \leq \mathrm{psb_t} \cdot \mathrm{M} \tag{6.10}$$

$$\mathrm{UtilExp_t} \leq (1 - \mathrm{psb_t}) \cdot \overline{\mathrm{UtExp}}. \tag{6.11}$$

**Imbalance Constraint:**

$$\mathrm{Ld_t} = \mathrm{UtilPur_t} - \mathrm{UtilExp_t} + \mathrm{SOut_t} - \mathrm{SIn_t} + \mathrm{PGen_t}. \tag{6.12}$$

The last Equation (6.12) ensures that the demand matches the supply. Dimensionality of Equation (6.1) can be reduced by reformulating it using a typical year, where for each month we assume up to three hourly load profiles: week day, weekend and peak day. This yields 12x3x24 = 864 unique time-steps rather than $T := |\mathcal{T}_{yr}| = 8760$.

In Figure 6.1 we present input data (load profile and tariff) and model output of this section (PV generation, purchase of power from the utility, and SOC of the storage) for a residential consumer, to showcase the sensitivity of the optimization routine for different input tariffs. The two panels have different input for the peak purchase rates; in the top panel, the peak purchase time is from 3:00pm - 8:00pm, while in the lower panel, it is from 5:00am - 10:00am. Notice that the profiles for the SOC in the two panels are starkly different. For the top panel, the demand exceeds the power supply during the evening when electricity purchase rate is high. Thus, the storage unit supplies energy until it is empty and remains so until the afternoon on the following day. In contrast, when the peak electricity purchase rate is between 5:00am-10:00am (lower panel), the storage policy changes so as to supply power during the morning. The SOC increases in the afternoon due to excess generation from the PV. The storage remains fully charged for the rest of the day as there is no incentive to supply power during the evening. The difference in the profiles of the SOC has a significant impact on reliability (see Section 6.5). For example, if there is a failure in the line connecting the consumer to the utility during the night, storage in the lower panel can act as a back-up as it has available energy; however, storage in the top panel will be empty and will fail to back-up.

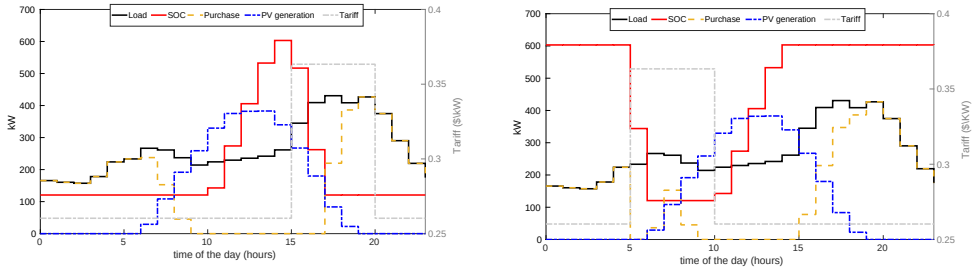



Figure 6.1: Optimization output by changing tariff rates for a residential consumer.

## 6.3 Distribution System

### 6.3.1 Representation and Simulation

We consider a radial distribution network with $B$ buses (denoted $\{b_1, b_2, b_3 \ldots, b_B\}$). Each bus (except $b_1$ which denotes the utility) contains a consumer with private investments in DERs. Each consumer runs the optimization module (Section 6.2) to find investments in DERs and their dispatch policies to locally match demand and supply. For each bus $b \in \{b_2, b_3 \ldots, b_B\}$, the optimization problem is solved locally, independent of the other consumers in the distribution network, considering the data (such as load profile $(\mathrm{Ld}^b_t)_{t\in\mathcal{T}_{yr}}$ and tariff $\mathbf{c}^b_t := (\mathrm{PurRt}^b_t, \mathrm{ExpRt}^b_t, \mathrm{DmdRt}^b_{m,p})_{t\in\mathcal{T}_{yr}}$) only for the consumer at the the bus $b$. As before, the output of the optimization includes the optimal investments $(\mathrm{Cap}^b_{pv}, \mathrm{Cap}^b_s)$, optimal dispatch policy for the storage $(\mathrm{SOut}^b_t, \mathrm{SIn}^b_t)_{t\in\mathcal{T}_{yr}}$, renewable output $(\mathrm{PGen}^b_t)_{t\in\mathcal{T}_{yr}}$ by the PV, purchase (export) of power from (to) the utility $(\mathrm{UtilPur}^b_t, \mathrm{UtilExp}^b_t)_{t\in\mathcal{T}_{yr}}$, for the consumer at bus $b$.

**Distribution network as a graph.** A distribution network can be represented as a graph with vertices $\{v_1, \ldots, v_B\}$ representing the buses $\{b_1, \ldots, b_B\}$ and the edges $\{e_1, e_2, \ldots, e_L\}$ representing the distribution lines $\{l_1, l_2, \ldots, l_L\}$. This transformation is useful when considering large networks with thousands of buses and distribution lines. Since we only consider radial networks in this chapter, the corresponding graph is an acyclic tree. A failure at any distribution line is equivalent to "breaking" the edge in the corresponding tree, partitioning it into 2 compartments. At any time $t$, we represent the total number of compartments as $A_t \geq 1$ and $\mathcal{I}_{i,t}, i = 1 \ldots A_t$ representing the set of vertices in compartment $i$. Finally, the set of compartments at $t$ is denoted by $\mathcal{A}_t = \{\mathcal{I}_{1,t}, \ldots, \mathcal{I}_{A_t,t}\}$. We denote the set of all vertices which have a path to $v_1$ as $\mathcal{I}_{1,t}$; it corresponds to buses with uninterrupted supply of power from the utility. The buses corresponding to vertices $v \notin \mathcal{I}_{1,t}$ have no connection to the utility (islanded) at time $t$,

as a result the DERs installed at these buses have to work in isolation to supply power.

**Monte Carlo simulations.** In this chapter, we only consider power interruptions due to failure in the distribution lines and assume the rest of the system components, e.g. storage and PV, to work without any failures. Extending our approach to incorporate respective failures would be mathematically straightforward, although computationally more expensive. At any time $t$, we assume that each of the distribution lines $l \in \{1, 2, \ldots, L\}$ could be in one of the two states, described through the variable $\delta_t^l$, *connected* ($\delta_t^l = 1$) and *disconnected* (or failed, $\delta_t^l = 0$). We further assume that each distribution line transitions from one state to the other independently, following an Exponential distribution for the transition times $\tau_t^l$:

$$f_{\tau_t^l}(s) = \tilde{\lambda}_t^l e^{-\tilde{\lambda}_t^l s},$$

where $\tilde{\lambda}_t^l = \lambda_f^l \delta_t^l + \lambda_r^l (1 - \delta_t^l)$. $\lambda_f^l$ represents the line failure rate i.e. rate of transition from state $\delta_t^l = 1$ to $\delta_{t+\tau_t^l}^l = 0$, and $\lambda_r^l$ represents the repair rate, i.e. rate of transitions from $\delta_t^l = 0$ to $\delta_{t+\tau_t^l}^l = 1$.

The state of the distribution network is defined via $L_t = [\delta_t^1, \delta_t^2, \ldots, \delta_t^L]$, which is a vector of states for each distribution line. The time for the transition of the distribution network from state $L_t$ is defined via $\tau_t := \min_l \tau_t^l$ with density

$$f_{\tau_t}(s) = \tilde{\lambda}_t e^{-\tilde{\lambda}_t s}, \tag{6.13}$$

where $\tilde{\lambda}_t = \sum_{l=1}^{L} \tilde{\lambda}_t^l$ is additive thanks to properties of Exponential random variables. At the transition epoch $t + \tau_t$, the distribution network may transition to $L$ possible states. The probability that the distribution network changes states due to a change in

the $k^{th}$ distribution line is:

$$\mathbb{P}(\tau_t^k = \tau_t | L_t) = \frac{\tilde{\lambda}_t^k}{\sum_{l=1}^{L} \tilde{\lambda}_t^l}. \tag{6.14}$$

To assess reliability of the distribution network, we simulate Monte Carlo samples of the transition times and transition states of the distribution network using Equations (6.13) and (6.14). For a total of $N_s$ Monte Carlo samples, we denote by $L_{0:T}^n, n = 1, \ldots, N_s$ the $n^{th}$ sample of the sequence of transition states in the time interval $[0, T]$ and with $\mathcal{T}_{fr}^n, n = 1, \ldots, N_s$ the corresponding sequence of transition times for the distribution network.

Notice that the output of the optimization in Section 6.2 was defined on the set $\mathcal{T}_{yr}$ containing only hourly time steps, however, the transition times $\mathcal{T}_{fr}^n \in [0, T]$ in any Monte Carlo sample are continuous. As a result, computation of the reliability indices requires analyzing the system $\forall t \in \mathcal{T}^n := \mathcal{T}_{yr} \bigcup \mathcal{T}_{fr}^n$. Thus, we extend the output of the optimization problem from $t \in \mathcal{T}_{yr}$ to $t \in [0, T]$ using piecewise constant functions for all the variables, except $\text{SOC}_t$ which is linearly interpolated:

$$\text{SOC}_t = \text{SOC}_{t_j} + (\text{SIn}_{t_j} \cdot \text{SCEff} - \frac{\text{SOut}_t}{\text{SDEff}}) \cdot (t_j - t),$$

$\forall t \in [t_j, t_{j+1})$ and $t_j \in \mathcal{T}_{yr}$.

### 6.3.2 Storage model

Failure of any of the distribution lines connecting bus $b$ to the utility interrupts its power supply. This forces the DERs installed at the bus to work in islanded mode to meet the needs of the consumer. Remember that the optimal policies $(\text{SIn}_t^b, \text{SOut}_t^b)_{t \geq 0}$ (superscript $b$ to emphasize that it is specific to the storage at bus $b$) for the storage are derived from the optimization problem which is oblivious to any failures. Thus, in this

section we propose heuristics for the operation of the storage during such islanded operation. Ideally, for every bus, we should either redo the optimization at every islanding instance and again at the time of reconnection; or, explicitly incorporate probability of such failures in the optimization problem itself. The former is computationally expensive, and the latter requires tools that are beyond the scope of this chapter. As a result, we rely on heuristics which work to keep the operational policy for the storage as close as possible to the optimal dispatch from Section 6.2.

Let us assume that the time points in the set $\mathcal{T}^n$ are arranged in sorted order. Since the rest of the subsection is devoted to computing the operational storage policy for any arbitrary $n$, for brevity we drop the index $n$ and denote it simply as $\mathcal{T}$.

We denote the operational policy for the charge and discharge of the storage at bus $b$ via $(\widehat{\mathrm{SIn}}^b_{t_j})_{t_j\in\mathcal{T}}$ and $(\widehat{\mathrm{SOut}}^b_{t_j})_{t_j\in\mathcal{T}}$ respectively ($\hat{}$ to remind that it is different from the optimal); and the corresponding state of charge as $(\widehat{\mathrm{SOC}}^b_{t_j})_{t_j\in\mathcal{T}}$. Let us define an additional state variable $(m^b_{t_j})_{t_j\in\mathcal{T}}$, which determines the operational policy $(\widehat{\mathrm{SIn}}^b_{t_j}, \widehat{\mathrm{SOut}}^b_{t_j})_{t_j\in\mathcal{T}}$ of the storage:

$$m^b_{t_j} = \begin{cases} 1 & \text{if } b \in \mathcal{I}_{1,t_j} \text{ and } \widehat{\mathrm{SOC}}^b_{t_j} = \mathrm{SOC}^b_{t_j} \\ 2 & \text{if } b \notin \mathcal{I}_{1,t_j} \\ 3 & \text{if } b \in \mathcal{I}_{1,t_j} \text{ and } \widehat{\mathrm{SOC}}^b_{t_j} \neq \mathrm{SOC}^b_{t_j}. \end{cases} \tag{6.15}$$

Equation (6.15) determines the three modes: normal, active and recovery mode for the storage:

- Normal mode ($m^b_{t_j} = 1$): During this mode, the bus is connected to the utility $b \in \mathcal{I}_{1,t_j}$ and state of charge is same as the optimal $\widehat{\mathrm{SOC}}_{t_j} = \mathrm{SOC}_{t_j}$. As a result, the storage follows the optimal control policy as derived from the optimization in Section 6.2 i.e.

$$\widehat{\mathrm{SIn}}^b_{t_j} = \mathrm{SIn}^b_{t_j}, \qquad \widehat{\mathrm{SOut}}^b_{t_j} = \mathrm{SOut}^b_{t_j}. \tag{6.16}$$

- Back-up mode ($m_{t_j} = 2$): storage transitions to the back-up mode when the bus $b$ has no connection to the utility $b \notin \mathcal{I}_{1,t_j}$. During back-up the storage unit acts to balance the net demand (load - PV generation), supplying power when net demand $> 0$ and charging when net demand $< 0$ while being constrained by the physical limits of the storage. We assume that the storage control at time $t_j$ is determined by the information available only prior to $t_j$, i.e. at the time of failure, the policy does not depend on the time-to-repair and assumes the failure will continue at least until the nearest hourly time-step. Thus, the storage policy is:

$$\widehat{\mathrm{SOut}}^{b}_{t_j} = \left((\mathrm{Ld}^{b}_{t_j} - \mathrm{PGen}^{b}_{t_j}) \wedge \widehat{\mathrm{SOut}}^{b,\max}_{t_j}\right)^{+},$$
$$\widehat{\mathrm{SIn}}^{b}_{t_j} = \left((\mathrm{PGen}^{b}_{t_j} - \mathrm{Ld}^{b}_{t_j}) \wedge \widehat{\mathrm{SIn}}^{b,\max}_{t_j}\right)^{+},$$

where,

$$\widehat{\mathrm{SOut}}^{b,\max}_{t_j} = \min \begin{cases} \mathrm{SCap}^{b} \cdot \overline{\mathrm{SDRt}}; \\ (\mathrm{SOC}^{b}_{t_j} - 0.2 \cdot \mathrm{SCap}^{b}) \frac{\mathrm{SDEff}}{\lceil t_j \rceil - t_j}; \end{cases}$$
$$\widehat{\mathrm{SIn}}^{b,\max}_{t_j} = \min \begin{cases} \mathrm{SCap}^{b} \cdot \overline{\mathrm{SCRt}} \\ \frac{\mathrm{SCap}^{b} - \mathrm{SOC}^{b}_{t_j}}{\overline{\mathrm{SCEff}} \cdot (\lceil t_j \rceil - t_j)} \end{cases}.$$

- Recovery mode ($m^{b}_{t_j} = 3$): When the bus $b$ is connected to the utility $b \in \mathcal{I}_{1,t}$, but the operational SOC of the storage is different from the optimal SOC, $\widehat{\mathrm{SOC}}^{b}_{t_j} \neq \mathrm{SOC}^{b}_{t_j}$, we define it as recovery mode. Within this mode the storage unit chooses the policy for charge and discharge such that the operational SOC soon achieves the optimal. Namely, if the SOC of the storage is more than the optimal SOC, the customer sells the energy to the grid to attain optimal SOC and vice-versa.

Thus,

$$\widehat{\mathrm{SIn}}^{b}_{t_j} = \begin{cases} \left(\frac{\mathrm{SOC}^{b}_{\lceil t_j \rceil} - \widehat{\mathrm{SOC}}^{b}_{t_j}}{\mathrm{SCEff}\cdot(\lceil t_j \rceil - t_j)} \wedge \widehat{\mathrm{SIn}}^{b,\max}_{t_j}\right)^{+} & \text{if } \mathrm{SOC}^{b}_{\lceil t_j \rceil} - \widehat{\mathrm{SOC}}^{b}_{t_j} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6.17}$$

$$\widehat{\mathrm{SOut}}^{b}_{t_j} = \begin{cases} 0 & \text{if } \mathrm{SOC}^{b}_{\lceil t_j \rceil} - \widehat{\mathrm{SOC}}^{b}_{t_j} > 0 \\ \left(\frac{(\widehat{\mathrm{SOC}}^{b}_{t_j} - \mathrm{SOC}^{b}_{\lceil t_j \rceil})\cdot \mathrm{SDEff}}{(\lceil t_j \rceil - t_j)} \wedge \widehat{\mathrm{SOut}}^{b,\max}_{t_j}\right)^{+} & \text{otherwise.} \end{cases} \tag{6.18}$$

Finally, given the operational policy of the storage, the state of charge updates via:

$$\widehat{\mathrm{SOC}}_{t_{j+1}} = \widehat{\mathrm{SOC}}_{t_j} + \left(\widehat{\mathrm{SIn}}_{t_j} \cdot \mathrm{SCEff} - \frac{\widehat{\mathrm{SOut}}_{t_j}}{\mathrm{SDEff}}\right) \cdot (t_{j+1} - t_j).$$

To illustrate the three modes and the operational policy for the storage, we present 5 different fault scenarios for a hypothetical example. We assume the net demand ($\mathrm{Ld}_t - \mathrm{PGen}_t$) and the optimal SOC is given as an input. For ease of illustration, in Figure 6.2 we assume that the net demand achieves only two values (+253kW and -253kW).

1. At the first fault, operational SOC is at the minimum. As a result storage fails to supply power.

2. At the second line fault, storage shifts to back-up mode to supply power. Once the line gets repaired, storage moves to recovery mode and pushes the operational SOC to return back to the optimal.

3. The optimal policy before the third fault was to charge the storage. However, line failure shifts the storage to the back-up mode and makes it supply power until it is empty. After the repair, the storage control matches the optimal policy because the operational and optimal SOC are the same.

4. The net demand is negative at the fourth line failure, thus the storage maintains the balance of power by charging. After the repair, storage is discharged to bring the operational SOC to the optimal.

5. After the fourth failure, the operational SOC exactly follows the optimal SOC.

Next, we define the reliability indices for the distribution network, given the operational policy of the storage and the Monte Carlo sample sequences of the failure-repair times of the distribution lines.

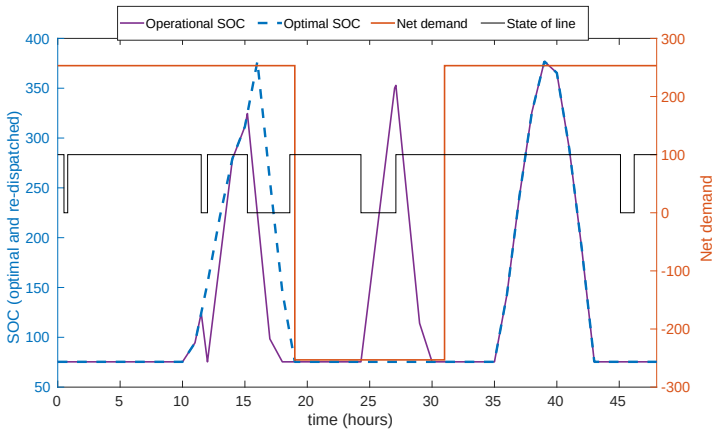



Figure 6.2: Operational policy for the storage for a hypothetical example with two bus network.

## 6.4 Reliability Evaluation

In this section we discuss the computation of the reliability indices. We only consider active power, with no losses, and assume that the voltage at each bus would be within the pre-defined limits. These assumptions are common in reliability studies of distribution networks, see for example [90, 93].

We define two sets of reliability indices, one from the perspective of the utility (named average energy not supplied, AENS) and another from the perspective of the

consumers (named average energy not consumed (AENC) and system average interruption duration index (SAIDI)). AENS is a standard metric used in reliability studies, however, as we will see in Section 6.5, it fails to fully encapsulate the impact on reliability when consumers make behind-the-meter investments in storage. We find that AENC and SAIDI are better metrics to assess reliability when consumers make private DER investments.

Given a Monte Carlo sample of the transition times $\mathcal{T}^n$ and states $L^n_{0:T}$, the first step is to evaluate the loss of load $C^{b,n}_{t_j}$ for the bus $b$ at time $t \in [0, T]$. We define:

$$C^{b,n}_t := \begin{cases} 0 & \text{if } b \in \mathcal{I}^n_{1,t} \\ \mathrm{Ld}^b_t & \text{if } b \notin \mathcal{I}^n_{1,t}, \mathrm{Cap}^b_s = 0 \\ (\mathrm{Ld}^b_t - \mathrm{PGen}^b_t - \widehat{\mathrm{SOut}}^{b,n}_t + \widehat{\mathrm{SIn}}^{b,n}_t)^+ & \text{if } b \notin \mathcal{I}^n_{1,t}, \mathrm{Cap}^b_s > 0. \end{cases} \tag{6.19}$$

According to Equation (6.19), the consumer at bus $b$ has zero loss of load $C^{b,n}_t = 0$ if the bus is connected to the utility $b \in \mathcal{I}^n_{1,t}$. However, if the bus is not connected $b \notin \mathcal{I}^n_{1,t}$ and there is no local storage available $\mathrm{Cap}^b_s = 0$, then the loss of load is same as load demand $\mathrm{Ld}^b_t$. Here we make an assumption that PV requires either connection to the utility or additional microelectronics (e.g. storage) to work. If neither is available, the circuit breaker connecting the PV to the consumer trips and the loss of load is the load demand. Finally, if the node is not connected but has storage installed locally $\mathrm{Cap}_s > 0$, the loss of load will be the net demand after storage re-dispatch.

We define energy not supplied $\mathrm{ENS}^{b,n}(\mathbf{c}^b)$, energy not consumed $\mathrm{ENC}^{b,n}(\mathbf{c}^b)$ and

failure duration $\mathrm{FD}^{b,n}(\mathbf{c}^b)$ at the bus $b$ for the $n^{th}$ Monte Carlo sample as:

$$\mathrm{ENS}^{b,n}(\mathbf{c}^b) := \int_0^T \mathrm{UtilPur}_t^b \mathbf{1}_{b \notin \mathcal{I}_{1,t}^n} \mathrm{dt}, \tag{6.20}$$

$$\mathrm{ENC}^{b,n}(\mathbf{c}^b) := \int_0^T C_t^{b,n} \mathrm{dt}, \tag{6.21}$$

$$\mathrm{FD}^{b,n}(\mathbf{c}^b) := \int_0^T \mathbf{1}_{\{C_t^{b,n}>0\}} \mathrm{dt}. \tag{6.22}$$

$\mathrm{ENS}^{b,n}$ accounts for the total energy that the utility could not supply to bus $b$ during the period $[0, T]$. $\mathrm{ENC}^{b,n}$ and $\mathrm{FD}^{b,n}$ calculate the loss of load and the failure duration for the consumer after incorporating the re-dispatch from the storage. Equations (6.20)-(6.22) emphasize the dependence of these metrics on the tariffs $\mathbf{c}^b$ (see Section 5.5). In Table 6.1 we summarize the relationship between $\mathrm{ENS}^{b,n}(\mathbf{c}^b)$ and $\mathrm{ENC}^{b,n}(\mathbf{c}^b)$ for different investment scenarios at node $b$.

Table 6.1: Reliability and investments.

| Investment | Relationship |
|---|---|
| None | $\mathrm{ENS}^{b,n}(\mathbf{c}^b) = \mathrm{ENC}^{b,n}(\mathbf{c}^b)$ |
| Only PV | $\mathrm{ENS}^{b,n}(\mathbf{c}^b) \leq \mathrm{ENC}^{b,n}(\mathbf{c}^b)$ |
| PV and storage | $\mathrm{ENS}^{b,n}(\mathbf{c}^b) \geq \mathrm{ENC}^{b,n}(\mathbf{c}^b)$ |

Finally, we define our reliability indices. A generic reliability index with $B$ buses in the distribution network is defined as:

$$\mathrm{Index} = \frac{1}{N_s} \sum_{n=1}^{N_s} \Big[ \frac{\sum_{i=2}^{B} F^{b_i,n}}{B} \Big], \tag{6.23}$$

$$\sigma(\mathrm{Index}) = \frac{\sqrt{\sum_{n=1}^{N_s} \left[ \frac{\sum_{i=2}^{B} F^{b_i,n}}{B} - \mathrm{Index} \right]^2}}{N_s}, \tag{6.24}$$

where $F^{b_i,n}$ is a test function. Thus, Index computes the average of the test function over the buses and the $N_s$ Monte Carlo samples and $\sigma(\mathrm{Index})$ is the standard error in

estimating the index. If the test function is $\text{ENS}^{b_i,n}(\mathbf{c}^{b_i})$, then the corresponding index is AENS; if it is $\text{ENC}^{b_i,n}(\mathbf{c}^{b_i})$, then the index is AENC; and if it is $\text{FD}^{b_i,n}(\mathbf{c}^{b_i})$, then the index is SAIDI.

We reiterate that AENS accounts for the energy that the utility was supposed to supply but could not due to failures in the distribution lines. Any re-dispatch from the storage to meet demand at times of islanding of the bus will *not* be considered in the definition of AENS, however, it will be part of the AENC and SAIDI. Thus, AENS captures the reliability from the perspective of the utility; in contrast, SAIDI and AENC are defined from the perspective of the consumers.

## 6.5 Numerical Example

In this section we discuss the effect of different tariff structures on the adoption of the PV and storage, along with its impact on the reliability indices. We consider the modified PG&E 69-bus [94] network (Figure 1.1b) for the case studies.

The distribution network in Figure 1.1b contains a mix of residential (red triangles), commercial (blue diamonds) and public services (green circles) consumers. Commercial consumers comprise of restaurants, supermarkets, hotels, malls and retail stores. Public services include schools, hospitals and government offices. The load profiles are scaled using active power data for the network in [95]. The total load demand for the network at 8:00am on a typical week in January is 3,802 kW and the total annual energy consumption is $\approx$ 23GWh. Unless otherwise specified, we use the following parameters for the storage and PV.

**Storage Parameters:** We assume storage charging/discharging efficiency of 0.9, maximum charge/discharge rate of 0.3, and minimum state of charge of 0.2. We take the fixed cost $\text{CFix}_s$ and variable cost $\text{CVar}_s$ as 250\$ and 250\$/kWh respectively. Lifetime of the storage is assumed to be 10 years.

**PV Parameters:** We assume the fixed cost $\text{CFix}_{pv}$ and variable $\text{CVar}_{pv}$ cost as 2500$ and 2500$/kWh respectively, PV lifetime of 20 years and mo operation and maintenance costs, $\text{DERMVr}_{\text{pv}} = 0$.

**Tariff Data:** We consider the base purchase rate $\overline{\text{PurRt}}_t$ as given in Tables 6.2, 6.3 and 6.4 for residential, services and commercial consumers respectively. We assume the base electricity export rate $\overline{\text{ExpRt}}_t = 0.3 \cdot \overline{\text{PurRt}}_t$, i.e. 30% of the base purchase rates $\overline{\text{PurRt}}_t$. Unless otherwise mentioned, we use the electricity purchase rate $\text{PurRt}_t = \overline{\text{PurRt}}_t$ and electricity export rate $\text{ExpRt}_t = \overline{\text{ExpRt}}_t$ as an input to the optimization module of Section 6.2. Notice the difference in the structure of purchase rates between residential consumers and service or commercial consumers. Residential consumers have only two types of time blocks during the day: on-peak and off-peak. On the other hand, service and commercial consumers have a more segmented tariff structure, divided into on-peak, mid-peak and off-peak hours. The peak purchase rate for the residential consumers is during the evening from 4:00pm - 9:00pm, while service or commercial consumers have peak purchase rate during the day from noon-6:00pm.

**Software:** The optimization problem described through Equations (6.1)-(6.12) is solved using the software *Distributed Energy Resources - Customer Adoption Model* (DER-CAM) developed at Lawrence Berkeley National Laboratory [96, 10]. For computing the compartments $\mathcal{A}_t$ of the graph (Section 6.3), we use the Python library `networkx`. The model is implemented in Python 3.

In Figure 6.3, we present the aggregate load, load net of PV output, and net of PV and storage output for the distribution network. If consumers are allowed to do investments only in PV, the net load profile corresponds to the "duck curve", with low net load during the day followed by a large ramp-up during the evening. In this example, ramp-up required by the utility to match the demand is 2000kW (200%) within four hours. Allowing for the investments in storage helps dampen the effect by reducing the

required ramp-up to 1600kW (160%) over 5 hours. It is important to mention that the load profile for individual consumers could be very different from the aggregate profile and it changes significantly with time of day and month of the year.

Table 6.2: Base tariff rates and periods for residential consumers. Summer: June-September, Winter: October-May.

| Type | Weekdays | Weekends | Summer ($/kWh) | Winter ($/kWh) |
|---|---|---|---|---|
| On-peak | 4:00pm - 9:00pm | — | 0.36335 | 0.22588 |
| Off-peak | Other times | All times | 0.26029 | 0.20708 |

Table 6.3: Base tariff rates and periods for service consumers. Summer: May-October and Winter: November-April.

| Type | Weekdays | Weekends | Summer ($/kWh) | Winter ($/kWh) |
|---|---|---|---|---|
| On-peak | noon - 6:00pm | — | 0.14726 | 0.10165 |
| Mid-peak | 8:00am - noon<br>6:00pm - 9:00pm | — | 0.10714 | 0.10165 |
| Off-peak | 9:00 pm - 8:00 am | All times | 0.08057 | 0.08717 |

Table 6.4: Base tariff rates and periods for commercial consumers. Summer: May-October and Winter: November-April

| Type | Weekdays | Weekends | Summer ($/kWh) | Winter ($/kWh) |
|---|---|---|---|---|
| On-peak | noon - 6:00 pm | — | 0.21471 | 0.1309 |
| Mid-peak | 8:00am - noon<br>6:00pm - 9:00pm | — | 0.15958 | 0.1309 |
| Off-peak | 9:00pm - 8:00am | All times | 0.13151 | 0.11384 |

Before presenting the results for the base parameters, let us summarize the sequence of steps to compute the reliability indices. First, we run the optimization module locally (Section 6.2) for consumers at each bus in Figure 1.1b. Next, we simulate $N_s$ Monte

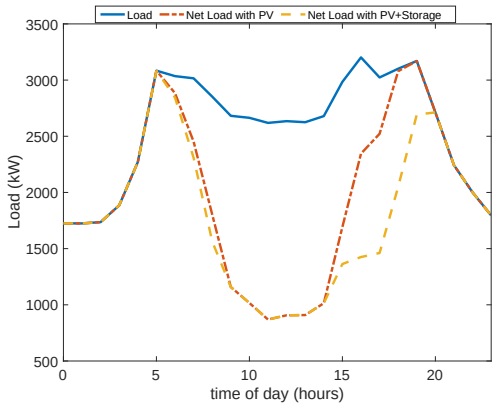



Figure 6.3: Aggregate load profile for the modified PG&E 69-bus network on a typical day in April along with net load profile with investments in only PV and PV/Storage. No export of power was allowed.

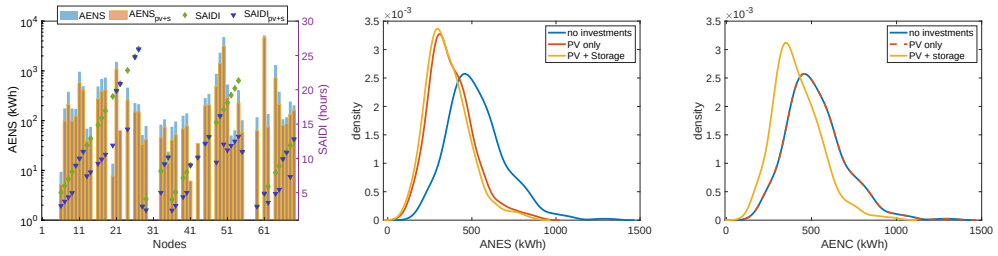



Figure 6.4: Load point indices for the base case. Center and Right panel: Distribution of AENS and AENC for the base case in the three investment scenarios.

Carlo samples for the failures and repair times of the distribution lines and the state of the distribution network as described in Section 6.3. Given the investments in DERs and the operational policy for the storage, we compute the network reliability indices as described in Section 6.4.

### Base case

In Table 6.5 we present the network reliability indices for three investment scenarios: (A) No investments are allowed, (B) Investments only in PV, (C) Investments in PV and storage are allowed. We observe a reduction in SAIDI by 2.6 hours when moving from

scenario (A) to (C). AENS and AENC are reduced from 526.3 kWh to 355.8 kWh and 417 kWh, respectively. Notice that as we move from scenario (A) to (B), AENS declines but AENC remains the same. The reduction in AENS is due to the installation of PV at the load points, which decreases the dependence of the consumer on the utility. Because of the absence of storage, the loss of load is equivalent to the load demand during the failure times; thus, the AENC remains the same. We emphasize that for scenario (C), while the investments in PV are distributed across all consumer types, the investments in storage are fully concentrated to residential consumers.

To understand the variability of the reliability across load points, we show in the left panel of Figure 6.4 the load point indices for scenarios (A) and (C). Due to the heterogeneity of the consumers in the network, we observe AENC at load points to vary from 10kWh to 10,000 kWh. Investments in PV and storage improve the load point indices across the network. The independent failure timings for each line is evident via green triangles representing SAIDI for scenario (A). Consumers further away from the utility have higher SAIDI as the local islanding rate is additive due to independent exponential line failures. However, allowing for investments in PV and storage implies $\text{SAIDI}_{pv+s} \leq \text{SAIDI}$ and $\text{AENS}_{pv+s} \leq \text{AENS}$.

Next, we present the distribution of AENS and AENC for the three investment scenarios in the center and right panel of Figure 6.4. Besides reducing the AENS or AENC, we also notice the reduction in variance of the indices for scenario (C) compared to scenario (A).

| scenario | AENS (kWh) | AENC (kWh) | SAIDI (hours) | PV (kW) | Storage (kWh) |
|---|---|---|---|---|---|
| No PV or storage | 526.3 | 526.3 | 12.1 | 0 | 0 |
| Only PV allowed | 378.8 | 526.3 | 12.1 | 3,812 | 0 |
| Both PV and storage | 355.8 | 417.4 | 9.5 | 3,812 | 3,852 |

Table 6.5: Reliability and total investments in PV and storage for the base case.

### Sensitivity

We discuss three case studies: (A) Effect of homothetic change in electricity purchase rate, (B) Effect of increasing the peak purchase rate (PPR), and (C) Effect of changing time of PPR.

**Homothetic change in purchase rate.** Here we increase/decrease the purchase rate by the same factor for all times i.e. $\mathrm{PurRt}_t = \gamma_{\mathrm{pur}} \cdot \overline{\mathrm{PurRt}}_t \quad \forall t \in [0, T]$, where the purchase factor $0.7 \leq \gamma_{\mathrm{pur}} \leq 1.3$. In the left panel of Figure 6.5 we present the adoption in PV and storage, and in the right panel its impact on reliability. Higher tariffs provide more incentive for the consumers to invest in PV and storage, thus, higher $\gamma_{\mathrm{pur}}$ results in higher investments and lower values for the reliability indices. Investments in PV increase from 2,700 kW to 4,200 kW and storage from 0 kWh to 5000 kWh as $\gamma_{\mathrm{pur}}$ changes from 0.7 to 1.3.

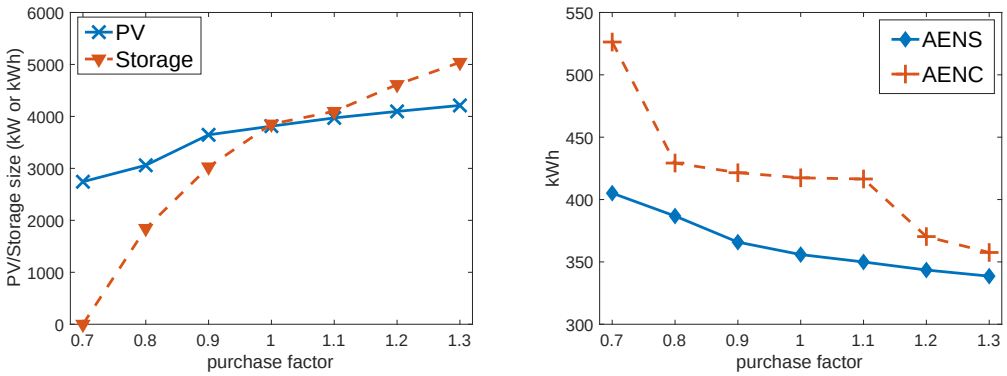



Figure 6.5: Effect of homothetic change in purchase rate. *Left Panel:* Investments in PV and storage due to change in purchase factor $\gamma_{\mathrm{pur}}$. *Right Panel:* AENS and AENC as a function of $\gamma_{\mathrm{pur}}$.

**Peak purchase rate.** Next, we present the effect of changing just the on-peak

rate. The purchase rate is thus,

$$\text{PurRt}_t = \begin{cases} \gamma_{\text{pk}} \cdot \overline{\text{PurRt}}_t & \text{if } t \in \mathcal{T}_{\text{pk}}; \\ \overline{\text{PurRt}}_t & \text{if } t \notin \mathcal{T}_{\text{pk}}, \end{cases} \tag{6.25}$$

where $\mathcal{T}_{\text{pk}}$ is the set of times corresponding to on-peak period of the base purchase rate and $\gamma_{\text{pk}}$ refers to the peak factor. We consider $1.0 \leq \gamma_{\text{pk}} \leq 2.5$.

In Figure 6.6 we present the effect of increasing $\gamma_{\text{pk}}$ on investments in PV and storage (left panel) and the reliability indices (right panel). Similar to $\gamma_{\text{pur}}$, higher $\gamma_{\text{pk}}$ leads to increased investments in both PV and storage. Unlike $\gamma_{\text{pur}}$, this time the investments in storage increase by 400% (3,852 kWh to 16,162 kWh) and dwarf the increase in PV investments (3,812 kW to 4,255 kW). There are also noticeable jumps in the capacity of the storage investments as we increase $\gamma_{\text{pk}}$. The massive increase in storage capacity leads to decline in AENC from 420 kWh to 270 kWh. The AENS, on the other hand, remains relatively flat because the storage re-dispatch during line failures is captured in AENC but not in AENS.

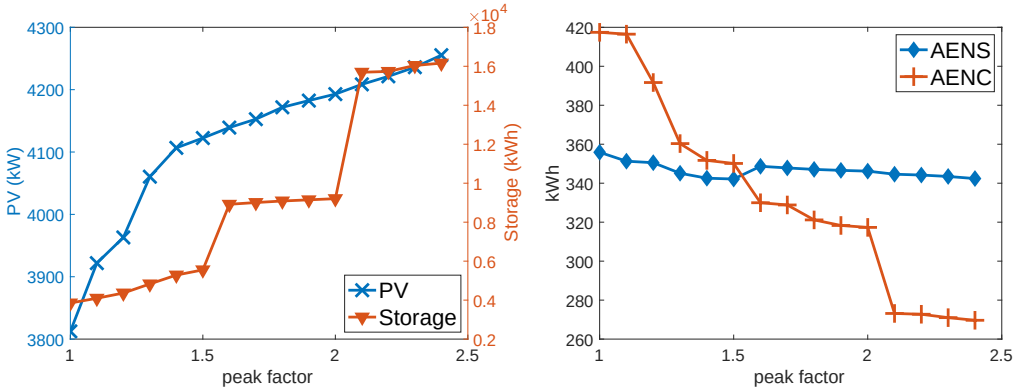



Figure 6.6: Effect of peak factor $\gamma_{\text{pk}}$. *Left Panel:* Investments in PV and storage due to increase in peak factor. *Right Panel:* AENS and AENC as a function of $\gamma_{\text{pk}}$.

On comparing the cost of investments against changes in reliability for different values of $\gamma_{\text{pur}}$ and $\gamma_{\text{pk}}$, we find that $\gamma_{\text{pk}}$ is a more efficient tool for improving network

reliability. Increasing the peak rate provides more incentive for the storage to keep relatively higher SOC before the peak time, thus providing more cushion from the storage in case of islanding of the bus. In Figure 6.7, we present the total annualized costs (Equation (6.1)) against the reliability indices by changing $\gamma_{\text{pk}}$ (blue circles) and $\gamma_{\text{pur}}$ (orange stars). In the left panel, we present the AENC—representing the perspective of the consumers, and in the right panel AENS—representing the perspective of the utility. First, higher costs due to increased investments in PV and storage leads to lower reliability indices. Second, increasing $\gamma_{\text{pk}}$ reduces AENC with relatively lower increase in cost to the consumers. Thus, for the same level of reliability, the peak factor is a better tool than the purchase factor. From the perspective of the utility (left panel with AENS on $x$-axis), the peak factor and the purchase factor are interchangeable.

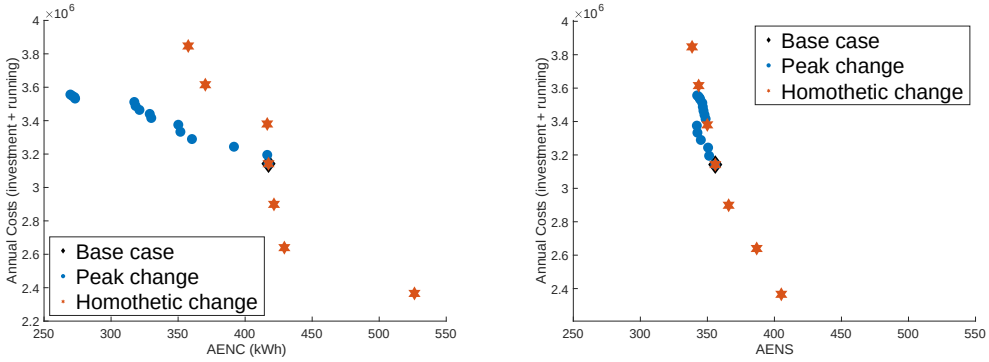



Figure 6.7: Effectiveness of peak factor $\gamma_{\text{pk}}$ and purchase factor $\gamma_{\text{pur}}$.

**Peak purchase time.** As renewables penetration increases, regulators may move the time of PPR in attempt to change the behavior of the consumers, aligning it with the needs of the grid. As a result, we discuss the effect of changing the time of PPR on the investments and the reliability indices. In Figure 6.8, we present the effect of changing the start hour for the peak rate (SHPR) of the residential, services, and commercial consumers. We continue to keep the duration for the PPR as 5 hours a day for residential, and 6 hours a day for services and commercial consumers. Shifting the

SHPR for the residential consumers from 8:00am to 4:00pm results in approximately 60% (from 2400 kWh to 3800 kWh) increase in the investments in storage but has no impact on the investments in PV. Intuitively, PPR during the evening provides more incentive for the consumers to store PV output, resulting in higher storage investments. In comparison, moving the SHPR for the services and commercial consumers from 8:00am to 4:00pm marginally reduces the investments in PV, from 3840kW to 3755 kW, but results in no change for the storage capacity. The reduction in PV investments can be explained through the overlap between the periods of peak solar irradiance and PPR. The advantage of PV installation is maximized when the two periods overlap. Because the period of peak solar irradiance is from 8:00am to 6:00pm, the investment in PV is economically more beneficial if the time window of 8:00am - 2:00pm is chosen for PPR compared to 2:00pm - 8:00pm.

The impact of these investments on the reliability indices is presented in the lower panel of Figure 6.8. The lowest AENS is attained when the SHPR for commercial and services consumers is at 8:00am and for residential consumers is at 4:00pm. The level of AENS is similar to the base tariff (blue diamonds). This suggests that differentiated peak time in the tariffs for different types of consumers is a better alternative than the same peak time for everyone.

Turning the attention to AENC, by changing the SHPR of the residential consumers to 8:00am, we can improve the AENC by another 3.5% due to change in storage SOC (cf. Figure 6.1). This reduction in AENC is attained with lowest investment in storage capacity.

## 6.6 Summary

In this chapter we propose a model for evaluating the effect of tariffs on the reliability of distribution networks considering optimal behind-the-meter investments in

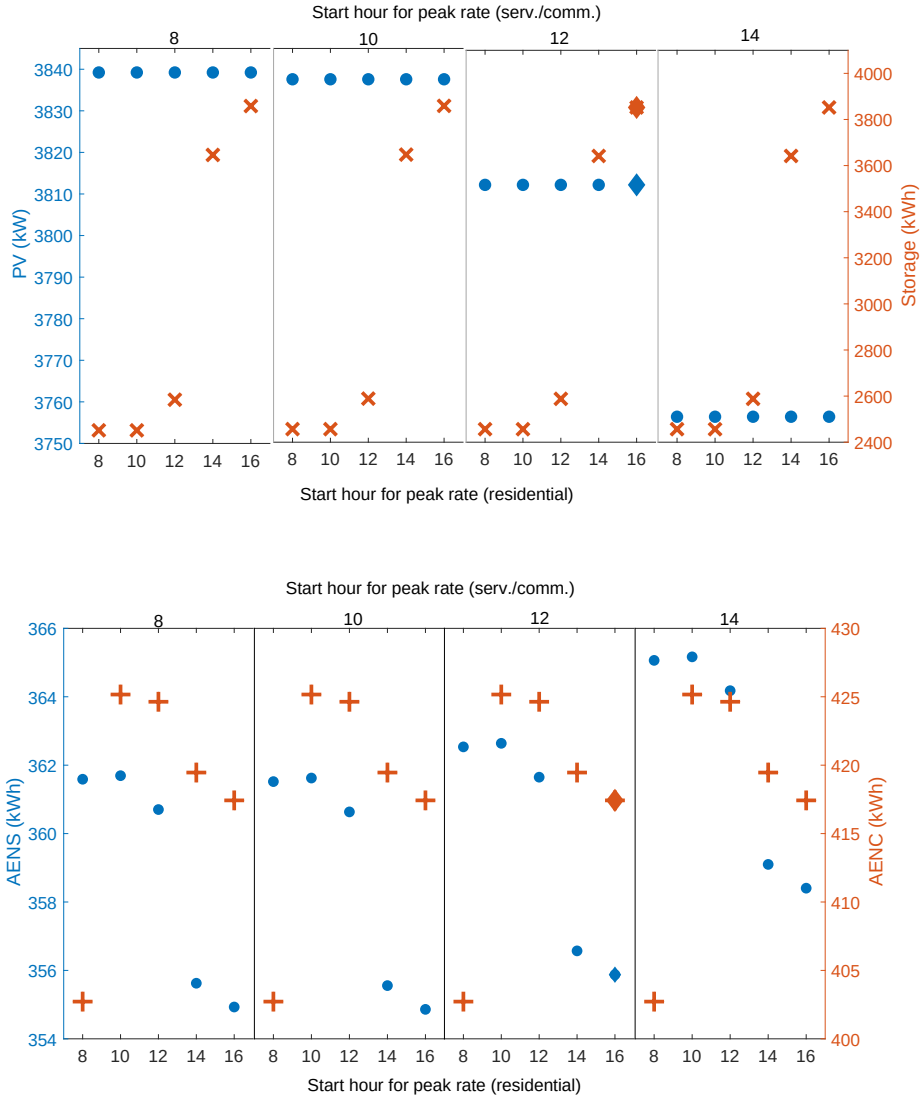



Figure 6.8: Effect of time-of-day of peak purchase rate, SHPR. *Top panel:* Investments in PV (blue circles) and Storage (orange crosses). *Bottom panel:* reliability indices, AENS (blue circles) and AENC (orange crosses). Scatter plot with diamonds represents the base tariffs.

PV and storage. We demonstrate the model on a PG&E-69 bus network with building energy-load data for San Francisco with PG&E's electricity tariff rate. Our contributions are threefold: (1) We overlay deterministic optimal control with simulation-based

methods to assess reliability of a distribution network. To the best of our knowledge, we are not aware of other works considering reliability with "optimal" behind-the-meter investments in PV and storage. (2) We provide two sets of reliability indices, including AENS which is an industry standard metric based on the perspective of the utility, as well as AENC and SAIDI, overlooking the perspective of the consumers. Our results indicate that the sensitivity of these indices to changes in electricity tariffs is very different. With behind-the-meter resources becoming more ubiquitous, we find AENC to better capture the loss of load at the consumer end. (3) We assess the impact of changes in tariffs, both in time and magnitude, on the reliability of the distribution network. We find that if the regulator's perspective is reliability, changing only the peak rate is more cost efficient than changing the tariff homothetically. Furthermore, we find significant change in storage dispatch policy and investments in PV/storage as the peak time for the purchase rate is changed, resulting in different reliability.

# Bibliography


[1] J. Yu, C. Marnay, M. Jin, C. Yao, X. Liu, and W. Feng, *Review of Microgrid Development in the United States and China and Lessons Learned for China*, *Energy Procedia* **145** (2018) 217 – 222.

[2] N. Hayashi, M. Nagahara, and Y. Yamamoto, *Robust AC Voltage Regulation of Microgrids in Islanded Mode with Sinusoidal Internal Model*, *SICE Journal of Control, Measurement, and System Integration* **10** (2017), no. 2 62–69.

[3] P. Denholm, M. O'Connell, G. Brinkman, and J. Jorgenson, *Overgeneration from Solar Energy in California. A Field Guide to the Duck Chart*, *Technical report* (2015) [doi=NREL/TP-6A20-65023].

[4] B. Heymann, J. F. Bonnans, F. Silva, and G. Jimenez, *A Stochastic Continuous Time Model for Microgrid Energy Management*, in *2016 European Control Conference (ECC)*, pp. 2084–2089, 2016.

[5] R. Carmona and M. Ludkovski, *Valuation of Energy Storage: An Optimal Switching Approach*, *Quantitative Finance* **10** (2010), no. 4 359–374.

[6] A. Boogert and C. de Jong, *Gas Storage Valuation Using a Monte Carlo Method*, *The Journal of Derivatives* **15** (2008), no. 3 81–98.

[7] X. Warin, *Gas Storage Hedging*, in *Numerical Methods in Finance: Bordeaux, June 2010* (R. A. Carmona, P. Del Moral, P. Hu, and N. Oudjane, eds.), pp. 421–445. Springer, Berlin, Heidelberg, 2012.

[8] M. Ludkovski, *Kriging Metamodels and Experimental Design for Bermudan Option Pricing*, *Journal of Computational Finance* **22** (2018), no. 1 37–77.

[9] B. Bouchard and X. Warin, *Monte Carlo Valuation of American Options: Facts and New Algorithms to Improve Existing Methods*, in *Numerical Methods in Finance: Bordeaux, June 2010* (R. A. Carmona, P. Del Moral, P. Hu, and N. Oudjane, eds.), pp. 215–255. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[10] S. Mashayekh, M. Stadler, G. Cardoso, and M. Heleno, *A Mixed Integer Linear Programming Approach For Optimal DER Portfolio, Sizing, And Placement In Multi-Energy Microgrids*, *Applied Energy* **187** (2017) 154 – 168.

[11] C. Alasseur, A. Balata, S. Ben-Aziza, A. Maheshwari, P. Tankov, and X. Warin, *Regression Monte Carlo for Microgrid Management*, *ESAIM: ProcS* **65** (2019) 46–67.

[12] M. Ludkovski and A. Maheshwari, *Simulation Methods for Stochastic Storage Problems: A Statistical Learning Perspective*, *Energy Systems (to Appear)* (2019) [arXiv:1803.11309].

[13] A. Balata, M. Ludkovski, A. Maheshwari, and J. Palczewski, *Statistical Learning for Probability-Constrained Stochastic Optimal Control*, *ArXiv e-prints* (2019) [arXiv:1905.00107].

[14] A. Maheshwari, M. Heleno, and M. Ludkovski, *The Effect of Rate Design on Power Distribution Reliability Considering Adoption of DERs*, *Unpublished manuscript* (2019).

[15] J. N. Tsitsiklis and B. Van Roy, *Regression Methods for Pricing Complex American-style Options*, *IEEE Transactions on Neural Networks* **12** (2001), no. 4 694–703.

[16] F. A. Longstaff and E. S. Schwartz, *Valuing American Options by Simulation: A Simple Least-Squares Approach*, *The Review of Financial Studies* **14** (2001), no. 1 113–147.

[17] J. F. Carriere, *Valuation of the early-exercise price for options using simulations and nonparametric regression*, *Insurance: Mathematics and Economics* **19** (1996), no. 1 19 – 30.

[18] R. Carmona and M. Ludkovski, *Pricing asset scheduling flexibility using optimal switching*, *Applied Mathematical Finance* **15** (2008), no. 5-6 405–447.

[19] D. Mazieres and A. Boogert, *A Radial Basis Function Approach to Gas Storage Valuation* , *Journal of Energy Markets* **6** (2013), no. 2 19–50.

[20] A. Balata and J. Palczewski, *Regress-Later Monte Carlo for Optimal Inventory Control with Applications in Energy*, *ArXiv e-prints* (2018) [arXiv:1703.06461].

[21] I. Kharroubi, N. Langrené, and H. Pham, *A Numerical Algorithm for Fully Nonlinear HJB Equations: An Approach by Control Randomization*, *Monte Carlo Meth. and Appl.* **20** (2014) 145–165.

[22] A. Balata and J. Palczewski, *Regress-Later Monte Carlo for Optimal Control of Markov Processes*, *ArXiv e-prints* (2017) [arXiv:1712.09705].

[23] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II.* Athena Scientific, 3rd ed., 2007.

[24] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st ed., 1994.

[25] N. Bäuerle and U. Rieder, *Theory of Finite Horizon Markov Decision Processes*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[26] N. Bäuerle and V. Riess, *Gas Storage Valuation with Regime Switching*, *Energy Systems* **7** (2016), no. 3 499–528.

[27] H. Pham, *Optimal Switching and Free Boundary Problems*, in *Continuous-time Stochastic Control and Optimization with Financial Applications*, pp. 95–137. Springer, Berlin, Heidelberg, 2009.

[28] N. Touzi, *Stochastic Control Problems, Viscosity Solutions, and Application to Finance*. Sc. Norm. Super. di Pisa Quaderni, Scuola Normale Superiore, Pisa, 2004.

[29] Z. Chen and P. A. Forsyth, *A Semi-Lagrangian Approach for Natural Gas Storage Valuation and Optimal Operation*, *SIAM Journal on Scientific Computing* **30** (2008), no. 1 339–368.

[30] B. Heymann, J. F. Bonnans, P. Martinon, F. J. Silva, F. Lanas, and G. Jiménez-Estévez, *Continuous Optimal Control Approaches to Microgrid Energy Management*, *Energy Systems* **9** (2018), no. 1 59–77.

[31] M. V. F. Pereira and L. M. V. G. Pinto, *Stochastic Optimization of a Multireservoir Hydroelectric System: A Decomposition Approach*, *Water Resources Research* **21** (1985), no. 6 779–792.

[32] M. V. F. Pereira and L. M. V. G. Pinto, *Multi-stage Stochastic Optimization Applied to Energy Planning*, *Mathematical Programming* **52** (1991), no. 1 359–375.

[33] A. Bhattacharya, J. P. Kharoufeh, and B. Zeng, *Managing energy storage in microgrids: A multistage stochastic programming approach*, *IEEE Transactions on Smart Grid* **9** (Jan, 2018) 483–496.

[34] W. Van-Ackooij and X. Warin, *On Conditional Cuts for Stochastic Dual Dynamic Programming*, *ArXiv e-prints* (2017) [arXiv:1704.06205].

[35] H. Gevret, N. Langrené, J. Lelong, X. Warin, and A. Maheshwari, *STochastic OPTimization library in C++*, research report, EDF Lab, May, 2018.

[36] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st ed., 1998.

[37] S. Kim and H. Lim, *Reinforcement Learning Based Energy Management Algorithm for Smart Energy Buildings*, *Energies* **11** (2018), no. 8.

[38] J.-C. Alais, P. Carpentier, and M. De Lara, *Multi-usage Hydropower Single Dam Management: Chance-Constrained Optimization and Stochastic Viability*, *Energy Systems* **8** (2017), no. 1 7–30.

[39] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati, *Joint Chance Constrained Programming for Hydro Reservoir Management*, *Optimization and Engineering* **15** (2014), no. 2 509–531.

[40] L. Andrieu, R. Henrion, and W. Rmisch, *A model for dynamic chance constraints in hydro power reservoir management*, *European Journal of Operational Research* **207** (2010), no. 2 579 – 589.

[41] S. A. P. Quintero, M. Ludkovski, and J. P. Hespanha, *Stochastic Optimal Coordination of Small UAVs for Target Tracking using Regression-based Dynamic Programming*, *Journal of Intelligent & Robotic Systems* **82** (2016), no. 1 135–162.

[42] L. Janson, E. Schmerling, and M. Pavone, *Monte Carlo Motion Planning for Robot Trajectory Optimization Under Uncertainty*, in *Robotics Research: Volume 2* (A. Bicchi and W. Burgard, eds.), pp. 343–361. Springer International Publishing, Cham, 2017.

[43] R. Zhang, N. Langrené, Y. Tian, Z. Zhu, F. Klebaner, and K. Hamza, *Dynamic Portfolio Optimization with Liquidity Cost and Market Impact: A Simulation-and-Regression Approach*, *ArXiv e-prints* (2017) [arXiv:1610.07694].

[44] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, *et. al.*, *Trends in microgrid control*, *IEEE Transactions on smart grid* **5** (2014), no. 4 1905–1919.

[45] S. S. Reddy, V. Sandeep, and C.-M. Jung, *Review of stochastic optimization methods for smart grid*, *Frontiers in Energy* **11** (Jun, 2017) 197–209.

[46] H. Liang and W. Zhuang, *Stochastic Modeling and Optimization in a Microgrid: A Survey*, *Energies* **7** (2014), no. 4 2027–2050.

[47] S. Mashayekh, M. Stadler, G. Cardoso, M. Heleno, S. C. Madathil, H. Nagarajan, R. Bent, M. Mueller-Stoffels, X. Lu, and J. Wang, *Security-Constrained Design of Isolated Multi-Energy Microgrids*, *IEEE Transactions on Power Systems* **33** (2018), no. 3 2452–2462.

[48] L. Olatomiwa, S. Mekhilef, A. Huda, and O. S. Ohunakin, *Economic evaluation of hybrid energy systems for rural electrification in six geo-political zones of nigeria*, *Renewable Energy* **83** (2015) 435 – 446.

[49] P. Haessig, B. Multon, H. B. Ahmed, S. Lascaud, and P. Bondon, *Energy Storage Sizing for Wind Power: Impact of the Autocorrelation of Day-Ahead Forecast Errors*, *Wind Energy* **18** (2015), no. 1 43–57.

[50] H. Ding, Z. Hu, and Y. Song, *Stochastic optimization of the daily operation of wind farm and pumped-hydro-storage plant*, *Renewable Energy* **48** (2012) 571–578.

[51] H. Ding, Z. Hu, and Y. Song, *Rolling optimization of wind farm and energy storage system in electricity markets*, *IEEE Transactions on Power Systems* **30** (2015), no. 5 2676–2684.

[52] J. Collet, O. Féron, and P. Tankov, *Optimal management of a wind power plant with storage capacity*, *HAL preprint* (2017) [hal:01627593].

[53] D. P. Bertsekas, *Stochastic Optimal Control*. Academic Press, New York, 1978.

[54] A. Boogert and C. de Jong, *Gas storage valuation using a multi-factor price process*, *Journal of Energy Markets* **4** (2011) 29–52.

[55] A. M. Malyscheff and T. B. Trafalis, *Natural Gas Storage Valuation via Least Squares Monte Carlo and Support Vector Regression*, *Energy Systems* **8** (2017), no. 4 815–855.

[56] M. Thompson, M. Davison, and H. Rasmussen, *Natural Gas Storage Valuation and Optimization: A Real Options Application*, *Naval Research Logistics (NRL)* **56** (2009), no. 3 226–238.

[57] M. Davison and G. Zhao, *Optimal Control of Two-Dam Hydro Facility*, *Systems Engineering Procedia* **3** (2012) 1 – 12.

[58] G. Zhao and M. Davison, *Optimal Control of Hydroelectric Facility Incorporating Pump Storage*, *Renewable Energy* **34** (2009), no. 4 1064 – 1077.

[59] R. B. Gramacy and M. Ludkovski, *Sequential Design for Optimal Stopping Problems*, *SIAM Journal on Financial Mathematics* **6** (2015), no. 1 748–775.

[60] D. Egloff, *Monte Carlo Algorithms for Optimal Stopping and Statistical Learning*, *The Annals of Applied Probability* **15** (2005), no. 2 1396–1432.

[61] D. Egloff, M. Kohler, and N. Todorovic, *A Dynamic Look-Ahead Monte Carlo Algorithm for Pricing Bermudan Options*, *The Annals of Applied Probability* **17** (2007), no. 4 1138–1171.

[62] N. Langrené, T. Tarnopolskaya, W. Chen, Z. Zhu, and M. Cooksey, *New Regression Monte Carlo Methods for High-dimensional Real Options Problems in Minerals industry*, in *21st International Congress on Modelling and Simulation*, pp. 1077–1083, 2015.

[63] J. Han and W. E, *Deep Learning Approximation for Stochastic Control Problems*, *ArXiv e-prints* (2016) [arXiv:1611.07422].

[64] M. Binois, R. B. Gramacy, and M. Ludkovski, *Practical Heteroscedastic Gaussian Process Modeling for Large Simulation Experiments*, *Journal of Computational and Graphical Statistics* **27** (2018), no. 4 808–821, [arXiv:1611.05902].

[65] M. Denault, J.-G. Simonato, and L. Stentoft, *A Simulation-and-Regression Approach for Stochastic Dynamic Programs with Endogenous State Variables*, *Computers & Operations Research* **40** (2013), no. 11 2760 – 2769.

[66] A. Geletu, M. Klppel, H. Zhang, and P. Li, *Advances and Applications of Chance-Constrained Approaches to Systems Optimisation Under Uncertainty*, *International Journal of Systems Science* **44** (2013), no. 7 1209–1232.

[67] C. Liu, X. Wang, Y. Zou, H. Zhang, and W. Zhang, *A probabilistic chance-constrained day-ahead scheduling model for grid-connected microgrid*, *2017 North American Power Symposium (NAPS)* (2017) 1–6, [doi=10.1109/NAPS.2017.8107180].

[68] S. Ahmed and A. Shapiro, *Solving Chance-Constrained Stochastic Programs via Sampling and Integer Programming*, in *INFORMS TutORials in Operations Research* (Z.-L. Chen and S. Raghavan, eds.), pp. 261–269. INFORMS, 2014.

[69] C. Keerthisinghe, G. Verbi, and A. C. Chapman, *A Fast Technique for Smart Home Management: ADP With Temporal Difference Learning*, *IEEE Transactions on Smart Grid* **9** (2018), no. 4 3291–3303.

[70] M. Ono, M. Pavone, Y. Kuwata, and J. Balaram, *Chance-Constrained Dynamic Programming with Application to Risk-Aware Robotic Space Exploration*, *Autonomous Robots* **39** (2015), no. 4 555–571.

[71] L. Doyen and M. D. Lara, *Stochastic viability and dynamic programming*, *Systems & Control Letters* **59** (2010), no. 10 629 – 634.

[72] Y. Jiao, O. Klopfenstein, and P. Tankov, *Hedging under multiple risk constraints*, *Finance and Stochastics* **21** (2017), no. 2 361–396.

[73] A. Nemirovski and A. Shapiro, *Convex Approximations of Chance Constrained Programs*, *SIAM Journal on Optimization* **17** (2007), no. 4 969–996.

[74] J. Luedtke and S. Ahmed, *A Sample Approximation Approach for Optimization with Probabilistic Constraints*, *SIAM Journal on Optimization* **19** (2008), no. 2 674–699.

[75] M. P. Deisenroth, C. E. Rasmussen, and J. Peters, *Gaussian Process Dynamic Programming*, *Neurocomputing* **72** (2009), no. 7 1508 – 1524.

[76] O. Roustant, D. Ginsbourger, and Y. Deville, *DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization*, *Journal of Statistical Software* **51** (2012), no. 1 1–55.

[77] J. Xu and J. S. Long, *Confidence intervals for predicted outcomes in regression models for categorical outcomes*, *Stata Journal* **5** (2005), no. 4 537–559.

[78] IPCC, *Special Report on Renewable Energy Sources and Climate Change Mitigation*, ch. 8, pp. 609–706. Cambridge University Press, 2011.

[79] California Energy Commission, *Building Energy Efficiency Standards for Residential and Nonresidential Buildings*, 2019.

[80] P. M. Sotkiewicz and J. M. Vignolo, *Towards a Cost Causation-Based Tariff for Distribution Networks With DG*, *IEEE Transactions on Power Systems* **22** (2007), no. 3 1051–1060.

[81] R. Sioshansi, *Retail electricity tariff and mechanism design to incentivize distributed renewable generation*, *Energy Policy* **95** (2016) 498 – 508.

[82] A. Picciariello, J. Reneses, P. Frias, and L. Sder, *Distributed generation and distribution pricing: Why do we need new tariff design methodologies?*, *Electric Power Systems Research* **119** (2015) 370 – 376.

[83] D. W. Cai, S. Adlakha, S. H. Low, P. D. Martini, and K. M. Chandy, *Impact of Residential PV Adoption on Retail Electricity Rates*, *Energy Policy* **62** (2013) 830 – 843.

[84] S. Candas, K. Siala, and T. Hamacher, *Sociodynamic modeling of small-scale PV adoption and insights on future expansion without feed-in tariffs*, *Energy Policy* **125** (2019) 521 – 536.

[85] D. Issicaba, J. A. Pecas Lopes, and M. A. da Rosa, *Adequacy and Security Evaluation of Distribution Systems With Distributed Generation*, *IEEE Transactions on Power Systems* **27** (2012), no. 3 1681–1689.

[86] A. M. Leite da Silva, L. C. Nascimento, M. A. da Rosa, D. Issicaba, and J. A. Peas Lopes, *Distributed energy resources impact on distribution system reliability under load transfer restrictions*, *IEEE Transactions on Smart Grid* **3** (2012), no. 4 2048–2055.

[87] H. Farzin, M. Moeini-Aghtaie, and M. Fotuhi-Firuzabad, *Reliability studies of distribution systems integrated with electric vehicles under battery-exchange mode*, *IEEE Transactions on Power Delivery* **31** (Dec, 2016) 2473–2482.

[88] H. Farzin, M. Fotuhi-Firuzabad, and M. Moeini-Aghtaie, *Reliability studies of modern distribution systems integrated with renewable generation and parking lots*, *IEEE Transactions on Sustainable Energy* **8** (2017), no. 1 431–440.

[89] N. Z. Xu and C. Y. Chung, *Reliability evaluation of distribution systems including vehicle-to-home and vehicle-to-grid*, *IEEE Transactions on Power Systems* **31** (Jan, 2016) 759–768.

[90] H. Farzin, M. Fotuhi-Firuzabad, and M. Moeini-Aghtaie, *Role of outage management strategy in reliability performance of multi-microgrid*

*distribution systems*, *IEEE Transactions on Power Systems* **33** (May, 2018) 2359–2369.

[91] A. Safdarian, M. Z. Degefa, M. Lehtonen, and M. Fotuhi-Firuzabad, *Distribution network reliability improvements in presence of demand response*, *IET Generation, Transmission Distribution* **8** (2014), no. 12 2027–2035.

[92] K. I. Sgouras, D. I. Dimitrelos, A. G. Bakirtzis, and D. P. Labridis, *Quantitative risk management by demand response in distribution networks*, *IEEE Transactions on Power Systems* **33** (2018), no. 2 1496–1506.

[93] M. Rastegar, *Impacts of Residential Energy Management on Reliability of Distribution Systems Considering a Customer Satisfaction Model*, *IEEE Transactions on Power Systems* **33** (2018), no. 6 6062–6073.

[94] K.-Y. Liu, W. Sheng, Y. Liu, X. Meng, and Y. Liu, *Optimal sitting and sizing of DGs in distribution system considering time sequence characteristics of loads and DGs*, *International Journal of Electrical Power & Energy Systems* **69** (2015) 430 – 440.

[95] M. E. Baran and F. F. Wu, *Optimal capacitor placement on radial distribution systems*, *IEEE Transactions on Power Delivery* **4** (1989), no. 1 725–734.

[96] G. Cardoso, M. Stadler, M. Bozchalui, R. Sharma, C. Marnay, A. Barbosa-Pvoa, and P. Ferro, *Optimal investment and scheduling of distributed energy resources with uncertainty in electric vehicle driving schedules*, *Energy* **64** (2014) 17 – 30.