Learning-Augmented Control and Decision-Making: Theory and Applications in Smart Grids

Thesis by
Tongxin Li

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2023
Defended June, 22nd, 2022

Tongxin Li
ORCID: 0000-0002-9806-8964

ACKNOWLEDGEMENTS

I am extremely fortunate to have spent five years at Caltech, under the supervision of my two advisors Steven Low and Adam Wierman, whose support, guidance, and encouragement during my PhD are invaluable. Their advice does not only pave the way for the results presented in this dissertation, but also deeply influenced my life in various aspects. As an advisor, Steven has done the best to help and mentor his students with his patience, expertise, passion, and enthusiasm in research that truly impacts real-world systems. He is also a genuine and perspicacious scholar who can always recognize and identify the key mathematical insights in complicated and challenging problems. What I learned from Steven throughout my PhD has motivated and will continue motivating me to pursue perfection in my academic career. Adam has set up a role model for me to be open-minded and conduct interdisciplinary research. He introduced me to a variety of areas in learning and control and his unique way of thinking has always inspired me in our every single meeting. I can never thank them enough for the time and efforts they spent on making me a mature researcher and I cannot imagine having had better PhD advisors.

I also would like to express my sincere gratitude to Yisong Yue and Eric Mazumdar for being my PhD candidacy and thesis committee chair and member, respectively. Their valuable feedback and insightful comments have greatly improved this work. Outside the thesis committee, I am also grateful to other Caltech faculty members, especially John Doyle, who dedicated his time on my candidacy exam and Anima Anandkumar, discussions with whom benefited me a lot.

This dissertation is a result of collaborations with many excellent people. I would like to sincerely acknowledge my collaborators and friends over the years, Yue Chen, Zachary Lee, Yiheng Lin, Guannan Qu, Bo Sun, Chenxi Sun, Lucien Werner, Ruixiao Yang, and Zixin Ye. I am grateful to have been working with and learning from them.

Furthermore, my PhD journey could have never been this enjoyable without my other colleagues and friends in Netlab and RSRG at Caltech. My great thanks to all current and past group members for creating and building an exceptional environment together. Especially, my thanks are due to Linqi Guo, for having been a good friend and life coach since my undergraduate years. He helped me settle in when I first came to Caltech and continue to motivate, inspire and support me in many ways.

The CMS administrative staff, especially Christine Ortega, has also been extremely helpful and supportive in various matters.

Finally, I would like to express my deepest gratitude to my family. This dissertation is dedicated to my parents and Stephanie for their unfaltering love and support.

ABSTRACT

Achieving carbon neutrality by 2050 does not only lead to the increasing penetration of renewable energy, but also an explosive growth of smart meter data. Recently, augmenting classical methods in real-world cyber-physical systems such as smart grids with data-driven black-box AI tools, forecasts, and ML algorithms has attracted a lot of growing interest. Integrating AI techniques into smart grids, on the one hand, provides a new approach to handle the uncertainties caused by renewable resources and human behaviors, but on the other hand, creates practical issues such as reliability, stability, privacy, and scalability, etc. to the AI-integrated algorithms.

This dissertation focuses on studying learning-augmented control and decision-making problems and their applications in smart grids.

The results presented in this dissertation are three-fold. The first part of this dissertation focuses on learning-augmented control problems. We study a problem in linear quadratic control, where imperfect/untrusted AI predictions of system perturbations are available. We show that it is possible to design a learning-augmented algorithm with performance guarantees that is aggressive if the predictions are accurate and conservative if they are imperfect. Machine-learned black-box policies are ubiquitous for non-linear control problems. Meanwhile, crude model information is often available for these problems from, e.g., linear approximations of non-linear dynamics. We next study the problem of equipping a black-box control policy with model-based advice for non-linear control on a single trajectory. We first show a general negative result that a naive convex combination of a black-box policy and a linear model-based policy can lead to instability, even if the two policies are both stabilizing. We then propose an *adaptive λ-confident policy*, with a coefficient λ indicating the confidence in a black-box policy, and prove its stability. With bounded non-linearity, in addition, we show that the adaptive λ-confident policy achieves a bounded competitive ratio when a black-box policy is near-optimal. Finally, we propose an online learning approach to implement the adaptive λ-confident policy and verify its efficacy in case studies about the Cart-Pole problem and a real-world electric vehicle (EV) charging problem with data bias due to COVID-19.

Aggregators have emerged as crucial tools for the coordination of distributed, controllable loads. To be used effectively, an aggregator must be able to communicate the available flexibility of the loads they control, known as the aggregate flexibility

to a system operator. However, most existing aggregate flexibility measures often are slow-timescale estimations and much less attention has been paid to real-time coordination between an aggregator and an operator. In the second part of this dissertation, we consider solving an online decision-making problem in a closed-loop system and present a design of *real-time* aggregate flexibility feedback, termed the *maximum entropy feedback* (MEF). In addition to deriving analytic properties of the MEF, combining learning and control, we show that it can be approximated using reinforcement learning and used as a penalty term in a novel control algorithm– the *penalized predictive control* (PPC) that enables efficient communication, fast computation, and lower costs. We illustrate the efficacy of the PPC using a dataset from an adaptive electric vehicle charging network and show that PPC outperforms classical model predictive control (MPC). In a theoretical perspective, a two-controller problem is formulated. A central controller chooses an action from a feasible set that is determined by time-varying and coupling constraints, which depend on all past actions and states. The central controller's goal is to minimize the cumulative cost; however, the controller has access to neither the feasible set nor the dynamics directly, which are determined by a remote local controller. Instead, the central controller receives only an aggregate summary of the feasibility information from the local controller, which does not know the system costs. We show that it is possible for an online algorithm using feasibility information to nearly match the dynamic regret of an online algorithm using perfect information whenever the feasible sets satisfy some criterion, which is satisfied by inventory and tracking constraints.

The third part of this dissertation consists of examples of learning, inference, and data analysis methods for power system identification and electric charging. We present a power system identification problem with noisy nodal measurements and efficient algorithms, based on fundamental trade-offs between the number of measurements, the complexity of the graph class, and the probability of error. Next, we specifically consider prediction and unsupervised learning tasks in EV charging. We provide basic data analysis results of a public dataset released by Caltech and develop a novel iterative clustering method for classifying time series of EV charging rates.

PUBLISHED CONTENT AND CONTRIBUTIONS

Tongxin Li contributed to the establishment of the foundational theory for the series of work below, proposing the methods, developing the applications, and designing, preparing, and running the simulations.

[1] Tongxin Li, Lucien Werner, and Steven H. Low. Learning graph parameters from linear measurements: Fundamental trade-offs and application to electric grids. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6554–6559, 2019. URL https://doi.org/10.1109/CDC40024.2019.9029949.

[2] Tongxin Li, Lucien Werner, and Steven H. Low. Learning graphs from linear measurements: Fundamental trade-offs and applications. *IEEE Transactions on Signal and Information Processing over Networks*, 6:163–178, 2020. URL https://doi.org/10.1109/TSIPN.2020.2975368.

[3] Tongxin Li, Steven H. Low, and Adam Wierman. Real-time flexibility feedback for closed-loop aggregator and system operator coordination. New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380096. URL https://doi.org/10.1145/3396851.3397725.

[4] Tongxin Li, Yue Chen, Bo Sun, Adam Wierman, and Steven H. Low. Information aggregation for constrained online control. 5(2), 2021. URL https://doi.org/10.1145/3460085.

[5] Tongxin Li, Bo Sun, Yue Chen, Zixin Ye, Steven H. Low, and Adam Wierman. Learning-based predictive control via real-time aggregate flexibility. *IEEE Transactions on Smart Grid*, 12(6):4897–4913, 2021. URL https://doi.org/10.1109/TSG.2021.3094719.

[6] Tongxin Li, Ruixiao Yang, Guannan Qu, Guanya Shi, Chenkai Yu, Adam Wierman, and Steven Low. Robustness and consistency in linear quadratic control with untrusted predictions. 6(1), 2022. URL https://doi.org/10.1145/3508038.

[7] Tongxin Li, Ruixiao Yang, Guannan Qu, Yiheng Lin, Steven Low, and Adam Wierman. Equipping black-box policies with model-based advice for stable nonlinear control. *Under review*.

Tongxin Li contributed to proposing the models and analyzing its performance, and designing, preparing, and running the simulations for the work below.

[1] Classification of electric vehicle charging time series with selective clustering. *Electric Power Systems Research*, 189:106695, 2020. ISSN 0378-7796. URL https://doi.org/10.1016/j.epsr.2020.106695.

[2] Zachary J. Lee, Tongxin Li, and Steven H. Low. Acn-data: Analysis and applications of an open ev charging dataset. New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366717. URL https://doi.org/10.1145/3307772.3328313.

TABLE OF CONTENTS

Acknowledgements . . . iii
Abstract . . . v
Published Content and Contributions . . . vii
Table of Contents . . . viii
List of Illustrations . . . xii
List of Tables . . . xviii
Chapter I: Introduction . . . 1
1.1 Key Problems and Challenges . . . 1
1.2 Related Work on Learning-Augmented Online Algorithms . . . 4
1.3 Learning-Augmented Control . . . 5
1.4 Large-Scale Learning-Augmented Decision-Making . . . 7
1.5 Learning, Inference, and Data Analysis in Smart Grids . . . 7
1.6 Dissertation Outline . . . 8

I Learning-Augmented Control 10
Chapter II: Linear Quadratic Control with Untrusted AI Predictions . . . 11
2.1 Introduction . . . 11
2.2 Model . . . 15
2.3 Consistent and Robust Control . . . 19
2.4 Self-Tuning λ-Confident Control . . . 22
2.5 Applications . . . 27
2.A Useful Lemmas . . . 36
2.B Competitive Analysis . . . 38
2.C Regret Analysis of Self-Tuning Control . . . 41
2.D Proof of Theorem 2.3.1 . . . 48
2.E Experimental Setup . . . 51
Chapter III: Non-linear Control with Black-box AI Policies . . . 53
3.1 Introduction . . . 54
3.2 Background and Model . . . 59
3.3 Warmup: A Naive Convex Combination . . . 61
3.4 Adaptive λ-Confident Control . . . 63
3.5 Practical Implementation and Applications . . . 65
3.A Experimental Setup and Supplementary Results . . . 69
3.B Notation and Supplementary Definitions . . . 74
3.C Useful Lemmas . . . 76
3.D Proof of Theorem 3.3.1 . . . 79
3.E Stability Analysis . . . 81
3.F Competitive Ratio Analysis . . . 91

II Learning-Augmented Decision-Making 96
Chapter IV: Learning-based Predictive Control: Formulation 97
4.1 Introduction 98
4.2 Related Literature. 100
4.3 Problem Formulation 103
4.4 Definitions of Real-Time Aggregate Flexibility: Maximum Entropy Feedback 109
4.5 Approximating Maximum Entropy Feedback via Reinforcement Learning 114
4.6 Penalized Predictive Control 117
4.7 Application 120
4.A Proof of Lemma 16 128
4.B Proof of Lemma 17 128
4.C Proof of Corollary 4.4.1 128
4.D Proof of Corollary 4.6.1 129
4.E Proof of Theorem 18 129
4.F Proof of Theorem 4.6.1 130
Chapter V: Learning-based Predictive Control: Regret Analysis 132
5.1 Introduction 132
5.2 Model 137
5.3 Information Aggregation 139
5.4 Penalized Predictive Control via Predicted MEF 143
5.5 Results 147
5.A Explanation of Definition 5.5.1 and Two Examples 152
5.B Proofs 154

III Learning, Inference, and Data Analysis in Smart Grids 161
Chapter VI: Learning Power System Parameters from Linear Measurements . 162
6.1 Introduction 162
6.2 Model and Definitions 168
6.3 Fundamental Trade-offs 173
6.4 Gaussian IID Measurements 180
6.5 Heuristic Algorithm 184
6.6 Applications in Electric Grids 186
6.A Proof of Theorem 6.3.1 192
6.B Proof of Theorem 6.3.2 193
6.C Proof of Corollary 6.3.1 194
6.D Proof of Lemma 22 194
6.E Proof of Lemma 23 196
6.F Proof of Lemma 24 196
6.G Proof of Theorem 6.4.1 197
6.H Proof of Theorem 6.4.2 198
Chapter VII: Electric Vehicle Charging Data Analysis 200
7.1 The ACN-DATA Dataset 200

7.2 Learning User Behavior 202
7.3 Predicting User Behavior 205
7.4 ACN-Data Charging Curves Analysis 209
7.5 Classification Method 214
7.6 Clustering, Applications, and Discussions 219

IV Impact and Future Directions **226**
Chapter VIII: Conclusions 227
8.1 Summary of Learning-Augmented Control Models 227
8.2 Summary of Chapters 227
8.3 Impact on Smart Grid Applications 230
8.4 Future Directions 230
Bibliography 232

LIST OF ILLUSTRATIONS

Number *Page*

1.1 Outline of this dissertation. Part I considers learning-augmented control problems for both linear and non-linear models. Part II of this dissertation focuses on large-scale learning-augmented decision making problems. The details of data, environments, predictions, and forecasts used in the applications presented in the first two parts are described in Part III. 9

2.1 System model of linear quadratic control with untrusted predictions. . 17

2.2 Tracking trajectories and trust parameters $(\lambda_0, \ldots, \lambda_{T-1})$ of the self-tuning control scheme. The x-axis and y-axis in the top 6 figures are locations of the robot. The y-axis in the bottom 3 figures denotes the value of the trust parameter. 30

2.3 Impact of trust parameters and performance of self-tuning control for robot tracking. 31

2.4 Adaptive battery-buffered EV charging modelled as a linear quadratic control problem. 31

2.5 An example of the daily charging demands in ACN-Data [1] on Nov 1st, 2018. 31

2.6 Impact of trust parameters and performance of self-tuning control for adaptive battery-buffered EV charging with synthetic EV charging data (top) and realistic daily EV charging data [1] (bottom). 33

2.7 The Cart-Pole model in Application 3. 33

2.8 Impact of trust parameters and performance of self-tuning control for the Cart-Pole problem. 35

3.1.1 Costs of pre-trained TRPO and ARS agents and an LQR when the initial pole angle θ (unit: radians) varies. 54

3.5.1 Competitiveness and stability of the adaptive policy. Top (*Competitiveness*): costs of pre-trained RL agents, an LQR and the adaptive policy when the initial pole angle θ (unit: radians) varies. Bottom (*Stability*): convergence of $\|x_t\|$ in t with $\theta = 0.4$ for pre-trained RL agents, a naive combination (Section 3.3) using a fixed $\lambda = 0.8$ and the adaptive policy. 67

3.5.2 Simulation results for real-world adaptive EV charging. Left: total rewards of the adaptive policy and SAC for pre-COVID-19 days and post-COVID-19 days. Right: shift of data distributions due to the work-from-home policy. 67
3.A.1 Illustration of the impact of COVID-19 on charging behaviors in terms of the total number of charging sessions and energy delivered (left) and distribution shifts (right). 72
3.A.2 Bar-plots of rewards/number of sessions corresponding to testing the SAC policy and the adaptive policy on the EV charging environment based on sessions collected from three time periods. 73
3.A.3 Supplementary results of Figure 3.5.2 with additional testing rewards for an in-COVID-19 period. 73
3.E.1 Outline of proofs of Theorem 3.4.1 and 3.4.2 with a stability analysis presented in Appendix 3.E and a competitive ratio analysis presented in Appendix 3.F. Arrows denote implications. 81
3.E.2 Telescoping sum of $x_t - x_t^*$. 89
4.3.1 System model: A feedback control approach for solving an online version of (4.2). The operator implements a control algorithm and the aggregator uses reinforcement learning to generate real-time aggregate flexibility feedback. 104
4.4.1 Feasible trajectories of power signals and the computed maximum entropy feedback in Example 3. 112
4.5.1 Learning and testing architecture for learning aggregator functions. . 115
4.5.2 Average rewards (defined in (4.11)) in the training stage with a tuning parameter $\beta = 6 \times 10^3$. Shadow region measures the variance. 117
4.7.1 Trade-offs of cost and charging performance. The dashed curve in the left figure corresponds to offline optimal cost. The tested days are selected (with no less than 30 charging sessions, i.e., $N \geq 30$) from Dec. 2, 2019 to Jan. 1, 2020. 124
4.7.2 Charging results of EVs controlled by PPC with tuning parameters $\beta = 2 \times 10^3$ (top), 4×10^3 (mid) and 6×10^3 (bot) for selected days (with no less than 30 charging sessions, i.e., $N \geq 30$) from Dec. 2, 2019 to Jan. 1, 2020. Each bar represents a charging session. 124
4.7.3 Substation charging rates generated by the PPC (orange) in the closed-loop control shown in Algorithm 5, together with the MPC generated (blue) and global optimal (dashed black) charging rates. 125

4.7.4 Cost-energy curves for the offline optimization in (4.2a)-(4.2d) (for the example in Section 4.3), MPC (defined in (4.17a)-(4.17f)) and PPC (introduced in Section 4.6). 126
5.2.1 Closed-loop interaction between a central controller and a local controller. 138
5.A.1 Graphical illustration of (5.10) in Definition 5.5.1. 152
6.3.1 The recovery of a graph matrix $\mathbf{Y}$ using the three-stage scheme in Algorithm 10. The $n - K$ columns of $\mathbf{Y}$ colored by gray are first recovered via the ℓ_1-minimization (6.11a)-(6.11c) in step (a), after they are accepted by passing the consistency check in step (b). Then, symmetry is used for recovering the entries in the matrix marked by green. Leveraging the linear measurements again, in step (c), the remaining K^2 entries in the white symmetric sub-matrix are solved using Equation (6.12). 177
6.5.1 Iterative dimension reduction of the heuristic algorithm. At step r, the s columns with the smallest scores defined in (6.20) are assumed to be "correct" and eliminated from the linear system. The dimension of variables is reduced by s and this procedure is repeated until the $\lceil n/s \rceil$ iterations are complete. 184
6.6.1 The number of samples required to accurately recover the nodal admittance matrix is shown on the vertical axis. Results are averaged over 20 independent simulations. Star and chain graphs are scaled in size between 5 and 300 nodes. IEEE test cases ranged from 5 to 200 buses. In the latter case, there are no assumptions on the random IID selection of the entries of $\mathbf{Y}$ (in contrast to the star/chain networks). Linear and logarithmic (in n) reference curves are plotted as dashed lines. 187

6.6.2 Probability of error for parameter reconstruction ε_{P} for the IEEE 30-bus test case is displayed on the vertical axis. Probability is taken over 50 independent trials. The horizontal axis shows the number of samples used to compute the estimate $\mathbf{X}$. The probability of error for independent recovery of all X_j via ℓ_1-norm minimization (double dashed line) and full rank non-sparse recovery (dot dashed line) are shown for reference. Adding the symmetry score function (second-to-left) improves over the naive column-wise scheme. Adding entry-wise positivity/negativity constraints on the entries of $\mathbf{X}$ (left-most curve) reduces sample complexity even further ($\approx 1/3$ samples needed compared to full rank recovery). 189

6.6.3 Sample complexity for accurate recovery is shown for a selection of IEEE power system test cases ranging from 5 to 57 buses. The number of samples for accurate recovery is obtained by satisfying the criterion $||\mathbf{X} - \mathbf{Y}||_{\mathrm{F}}/n^2 < 10^{-4}$. The noise $\mathbf{Z}$ is an IID Gaussian matrix with zero mean and standard deviation 0.01. The parameter γ in (6.19) is set to be 10^{-4}. As a benchmark, the number of measurements required for separately reconstructing every column of $\mathbf{Y}$ (standard compressed sensing) is also given. 189

6.6.4 A comparison between our iterative heuristic and basis pursuit. The Frobenius norm error plotted is averaged over 250 independent trials. The underlying graph is a star graph with $n = 24$. The solid and dotted gray curves are results for basis pursuit with and without a constraint emphasizing symmetry, respectively. 190

6.6.5 The impact of measurement noise on sample complexity for recovery of the IEEE 24-bus RTS test case is demonstrated. Trajectories correspond to increasing noise levels from dark (least) to light (most). From left to right, we observe—as expected—that for each variance value, the normalized Frobenius error of the recovered matrix decreases as the number of samples used for recovery increases. From bottom to top, we observe that the error increases (for every value of m) as variance of the additive noise $\mathbf{Z}$ increases. 191

7.1.1 Comparison of model distributions with actual data for Caltech during training period. 203

7.1.2 Prediction errors for Caltech (left two columns) and JPL (right two columns) for training dataset sizes ranging from 30 days to 90 days in the past. As a benchmark, we consider simply taking the mean of each user's prior behavior. For comparison, we also include the errors of user inputs. The results are measured by the mean absolute error (MAE) defined in (7.3). 203
7.3.1 Correlation between SMAPE(d) and SMAPE(e) and their marginal distributions for the JPL dataset. Kernel density estimation is used to approximate the joint distribution of the SMAPEs for predicted duration and energy which is shown as grey shading. The blue crosses represent the corresponding user input SMAPEs (for I-GMM) with respect to each charging session in the testing data set X_{Test}. 208
7.4.1 An example of a charging curve (in blue) and the corresponding pilot curve (in orange) for a charging session with *userID* 409 on Oct. 13, 2018. 210
7.4.2 Examples of charging curves where charging currents drop due to (1) scheduling, (2) battery charging state, and (3) noise, as indicated by the shaded regions. Each plot only shows a selected portion of a session. The time series are for sessions with *userID* 576 (top), 409 (mid) and 526 (bot), obtained on Nov. 07, 2018, Oct. 09, 2018 and Oct. 22, 2018, respectively. 212
7.4.3 The classification method introduced in this chapter. 215
7.5.1 An example of *extraction by matching*. The red subsequence $\mathbf{x}_1$ is a template with *userID* 409, which is extracted from the first session $\mathbf{s}_1$ of this user. The figure below visualizes the change of Euclidean distance of the second session $\mathbf{s}_2$ with respect to $\mathbf{x}_1$. The black vertical line indicates the best matching location in $\mathbf{s}_2$ for $\mathbf{x}_1$ and the tail $\mathbf{x}_2$ can be found correspondingly despite the slight difference of both tails. 217
7.6.1 The performance for different number of clusters using three different distance functions – Euclidean distance (ED), Modified Euclidean distance (MED), Dynamic Time warping distance (DTW). 219
7.6.2 Visualization of $K = 6$ clusters for MED, ED and DTW. Tails are within the same cluster if they have the same color and the tail representatives (medoids) are emphasized. 219

7.6.3 Two-dimensional visualization of our clustering results with $K = 6$ clusters. Tails for different users are colored differently. The clusters' colors are consistent with those used in Fig. 7.6.2. The marginal probabilities $p_1, \ldots, p_6$ represent the portions of charging sessions falling into the six clusters. 220

7.6.4 Examples of the training and testing data (tails) for four users. Sub-figures (a) and (b) are the tails of the two users with poor prediction performance (highlighted in blue in Table 7.6.1). The poor prediction performance is due to the fact that the tails in the training data are very different from those in the testing data. Sub-figures (c) and (d) are examples where the tail representatives achieve high-quality prediction performance. Tails in the training data and those in the testing data are similar. 221

7.6.5 Trade-offs between the number of samples m and the accuracy, sensitivity and precision. 225

LIST OF TABLES

Number		*Page*
1.1	List of learning-augmented algorithms.	4
1.2	Learning-augmented control and decision-making problems considered in this dissertation.	4
2.E.1	Hyper-parameters used in robot tracking and EV charging.	52
2.E.2	Hyper-parameters used in the Cart-Pole problem.	52
3.A.1	Hyper-parameters used in the Cart-Pole problem.	69
3.A.2	Hyper-parameters used in the real-world EV charging problem.	71
3.A.3	Average total rewards for the SAC policy and the adaptive λ-confident policy (Algorithm 4).	74
3.B.1	Symbols used in this work.	75
4.7.1	Hyper-parameters in the experiments.	122
7.1.1	Selected data fields in ACN-Data.	201
7.3.1	SMAPEs for Caltech and JPL datasets.	207
7.4.1	List of key notation used in this section.	209
7.6.1	Prediction results with user tail representative.	223
7.6.2	Prediction RMSE with cluster representative.	223
8.1.1	System models for the learning-augmented algorithms with different types of imperfect/untrusted predictions or black-box AI/ML advice. The detailed definitions of notation can be found in the corresponding chapters.	227

Chapter 1

INTRODUCTION

During the last decade, we have witnessed the development and evolution of AI decision-making techniques. With an increasing amount of data generated by real-world cyber-physical systems and major advances in AI and data science techniques, we are on the cusp of transforming black-box AI predictions and decision-making tools to more trustworthy and practical schemes for more sustainable, robust and intelligent cyber-physical systems. Towards this goal, augmenting classical methods in complex cyber-physical systems such as smart grids, internet of things (IoT), and transportation networks with black-box AI tools, forecasts and ML algorithms recently attracts growing interests. On the one hand, integrating AI techniques provides a new view and methodology to improve system performance and handle uncertainties; but on the other hand, it also creates practical issues such as *robustness*, *stability*, *reliability*, *privacy*, and *scalability*, etc. to the AI-integrated algorithms. Despite posing significant challenges, those issues have created great necessities and opportunities for developing new learning-augmented frameworks and algorithms.

The goal of this dissertation is to provide fundamental models and methods to broadly address some of the main issues in handling practical learning-augmented decision-making and control problems, with special focuses of applications in power systems. In the sequel, we introduce several remarkable problems and challenges in this area and summarize the remaining parts and chapters in this dissertation.

1.1 Key Problems and Challenges

The urgent need of applying AI techniques to real-world cyber-physical system leads to the bloom of interdisciplinary research in various areas such as learning, control and networks. Despite some remarkable milestones achieved in the recent years, we are still at the beginning of realizing trustworthy and practical AI for future cyber-physical systems, especially facing the following major challenges.

1. *Worst-case guarantees with imperfect predictions.*

   With the advances of data collection, handling, and management techniques, AI tools such as deep neural network models (DNNs) can be trained to generate predictions. However, those predictions are imperfect due to model error,

algorithm variance, or data bias. Answering the following question becomes critical:

How imperfect predictions generated by AI techniques can be used to retain worst-case guarantees of classic algorithms yet achieve optimal performance when the predictions are accurate?

Recently, developing learning-augmented algorithms with imperfect predictions has become an emerging research field at the intersection of theoretical computer science and AI, such as smoothed online convex optimization [2], online caching [3], ski-rental [4–7], online set cover [7], secretary and online matching [8], and metric task systems [9]. Extending the ideas used in existing theoretical algorithms to practical online decision-making/control models is the key to make use of AI predictions in real-world applications.

2. *Decision-making and control with untrusted AI policies*

   Going beyond AI predictions corresponding to specific parameters that are not model-agnostic, many AI decision-making policies/agents are model-free black-boxes, i.e., the policy parameters are modeled by DNNs that are neither interpretable nor adaptable. For instance, pre-trained reinforcement learning (RL) or imitation learing (IL) agents are available in certain applications, wherein interacting with the realistic environments and updating the NN parameters of those policies dynamically may be impractical. Those agents are not guaranteed to work well, as on the one hand they can sometimes be optimal or near-optimal, but on the other hand can be arbitrarily poor due to, e.g., sample inefficiency [10], reward sparsity [11], mode collapse [12], high variability of policy gradient [13, 14], or biased training data [15]. Generalizing ideas in learning-augmented algorithms with imperfect predictions to untrusted black-box policies is a prominent step to bridge the theory-practice gap between learning-augmented algorithms for theoretical computer science problems to real-world applications.

3. *Large-scale and hierarchical online control tasks*

   Practical control and online decision-making problems often involve large-scale controllable units, which are aggregated and controlled in a hierarchical scheme. For example, in a distribution electricity power network, to manage and schedule a large number of distributed energy resources (DERs), an aggregator is needed to coordinate with a system operator and the DERs.

The challenge is to design such a hierarchical system satisfying the following properties:

a) *Real-time coordination* is desirable to be consistent with real-time energy markets and handle uncertainties.

b) *Low communication and computational complexity* is preferable so that the state information of the large-scale controllable units needs to be aggregated.

c) *Privacy* of the controllable units needs to be preserved by aggregating their states.

The scale of the system and the privacy concerns of the individual units makes it impossible to directly control the controllable units and carry out computations via a central controller. Devising real-time learning-based control algorithms that can run in this novel two-controller setting is of vital importance for such applications.

4. *Learning and inference in smart grid research.* Achieving sustainability development is one of the most important and challenging goals in this century. To reduce carbon emissions, it is necessary to switch the paradigm and generate electricity using renewable and clean resources, which on the one hand, improves energy efficiency but on the other hand, brings uncertainty and creates difficulties for the traditional grids. With data generated in the next-generation electricity network, artificial intelligence (AI) can accelerate global efforts to protect the environment and conserve resources by monitoring renewable energy grids, detecting failures, identifying system information and predicting future grid conditions with a more efficient learning-based control strategy.

To tackle the challenges raised above, this dissertation focuses on designing and developing learning-augmented control and decision-making algorithms that guarantee worst-case performance with untrusted predictions [6], improve stability of black-box policies [16], facilitate coordination of controllers with large-scale controllable units [3–5] and artificial intelligence and data analysis techniques [1, 1, 2, 2] that improve the sustainability and resilience of the next-generation power grids. Below we provide overviews of the main chapters presented in this dissertation.

1.2 Related Work on Learning-Augmented Online Algorithms

The idea of augmenting robust/competitive online algorithms with machine-learned advice has attracted attention in online problems in various settings. In Table 1.1, we provide a subset of the existing results on learning-augmented online algorithms. The details of the related work will be discussed in Chapter 2 and 3.

Theoretical CS Problems	*Imperfect Predictions*	**Related Work**
Ski-rental	Number of skiing days	[5, 6]
Online secretary	Maximum price	[8]
Online bipartite matching	Adjacent edge-weights	
Online facility location	Predicted facility	[17, 18]
Online conversion	Threshold function	[19]
	AI/ML Advice	
Convex body chasing	Suggested actions	[2, 9]
Online subset sum	Decision	[20]
Online set cover	Predicted covering	[7]
Online caching	Machine-learned oracle	[3, 9]
k-server	Machine-learned predictor	[9]
Bin packing	Critical ratio approximation	[21]
Q-learning	Machine-learned Q-value functions	[22]

Table 1.1: List of learning-augmented algorithms.

The major focus of this dissertation is to address the aforementioned challenges by studying the learning-augmented control and decision-making problems listed in Table 1.2.

Control/Decision-Making	*Imperfect Predictions*	**This Dissertation**
Linear quadratic control	Perturbations	Chapter 2 ([23])
	AI/ML Advice	
Non-linear control	Black-box policy	Chapter 3 ([16])
Two-controller system	Feasibility information	Chapter 4 and 5 ([24–26])

Table 1.2: Learning-augmented control and decision-making problems considered in this dissertation.

Note that the generality of the models increases from the top to the bottom, as we will illustrate in Table 8.1.1 in Chapter 8. In the results listed in Table 1.1, additional trust parameters are introduced in classical theoretical online algorithms, but most of them do not have an approach to estimate the prediction error or if the AI/ML advice can be trusted in order to choose the best algorithm parameters. Moreover, while the previous results summarized in Table 1.1 on learning-augmented online algorithms provide significant contributions to the fundamental theory, the following question still remains:

Is it possible to improve the practicality of learning-augmented algorithms by considering control and decision-making models for real-world applications?

In this dissertation, we reply the question above in the affirmative. Unlike most of the existing results on theoretical computer science problems, we focus on more practical control models and decision-making settings and consider online learning approaches to tune the trust parameters. In the sequel, we introduce the problems considered in this dissertation.

1.3 Learning-Augmented Control

Classical control methods such as robust control and robust MPC are conservative, as they need to guarantee stability in the worst case. This usually yields poor performance in practice compared with an optimal policy. In contrast, predictions can significantly improve algorithm performance. For instance, in an electricity grid, AI-generated forecasts of electricity prices, voltage/current perturbations and battery states can help achieve near-optimal results. Deep learning methods make use of data and generate useful predictions as black-box tools that may inevitably contain errors, make mistakes and have no theoretical guarantees. Therefore, it is vital to find an adaptive approach to balance robustness/worst-case performance that is guaranteed by classical control methods and near-optimality that can be achieved with AI predictions.

In Chapter 2, we consider a linear quadratic control problem and proposes an online policy that (1) provides robustness guarantee when the predictions are inaccurate and (2) takes advantage of black box AI predictions when they are close to the ground truth. The goal is to design a controller that balances *consistency*, which measures the competitive ratio when predictions are accurate, and *robustness*, which bounds the competitive ratio when predictions are inaccurate. We propose a novel λ-confident policy and provide a competitive ratio upper bound that depends on a

trust parameter $\lambda \in [0, 1]$ set based on the confidence in the predictions and some prediction error ε. Motivated by online learning methods, we design a self-tuning policy that adaptively learns the trust parameter λ with a competitive ratio that does not scale up with the prediction error.

In many applications, instead of getting specific predictions that depend on concrete models, model-free black-box policies as functions that map the system state to a suggested action are more common. Developing safe RL with stability guarantees has attracted lots of interests recently. In [13, 27], Lyapunov analysis is applied to guarantee the stability of a model-based RL policy. Robust model predictive control (MPC) is combined with deep RL to ensure safety and stability [28]. Using regulated policy gradient, input-output stability is guaranteed for a continuous non-linear control model. Stability guarantees for (constrained) MDPs have been studied [28–30]. However, in many cases, the parameters of NNs for those black-box policies are often

1. ***Unadaptable.*** *The policy is a black-box and it is impossible or costly to access the NN parameters and make updates*.

2. ***Unwarranted.*** *The policy could make big mistakes and it is not reasonable to assume the policy behaves similarly to a stabilizing controller as in [13].*

In the previous results such as [13, 27, 28], deep RL policies are evaluated and updated during the episodic training steps. In those state-of-the-art results, the stability guarantees are proven, either considering an aforementioned episodic setting when the black-box policy can be improved or customized [27, 31], or assuming a small and bounded output distance between a black-box policy and a stabilizing policy for any input states to construct a Lyapunov equation [13], making their approaches less realistic. An distinctive feature of the result presented in Chapter 3 is that we consider stability and sub-optimality guarantee for black-box deep policies in a single trajectory such that we can neither learn from the environments nor evaluate or update the deep RL policy through extensive training steps.

To address this challenge, in Chapter 3, we consider a non-linear model and study the problem of equipping a black-box control policy with model-based advice for non-linear control on a single trajectory. We first show a general negative result that a naive convex combination of a black-box policy and a linear model-based policy can lead to instability, even if the two policies are both stabilizing. We then propose

an *adaptive λ-confident policy*, with a coefficient λ indicating the confidence in a black-box policy, and prove its stability. With bounded non-linearity, in addition, we show that the adaptive λ-confident policy achieves a bounded competitive ratio when a black-box policy is near-optimal. Finally, we propose an online learning approach to implement the adaptive λ-confident policy and verify its efficacy in case studies about the Cart-Pole problem and a real-world electric vehicle (EV) charging problem with data bias due to COVID-19.

1.4 Large-Scale Learning-Augmented Decision-Making

Aggregate information of DERs is not a brand new topic. In previous studies, convex approximations such as virtual battery models, Minkowski sum of individual polytopes or hyper-rectangles [32–36] are the most common strategies to be used for simplifying the constraint sets of large-scale DERs that may be non-convex. However, this offline design makes the control time-scales of operator-to-aggregator and aggregator-to-DERs inconsistent, prohibiting the adoption of a real-time electricity market and the ability of handling unexpected uncertainties.

Tackling problems raised by the first question, in Chapter 4 and 5, we consider large-scale control problems in power systems. In Chapter 4, a novel feedback mechanism that combines model predictive control (MPC) and deep reinforcement learning (RL) algorithms is formulated. It allows closed-loop coordination between a system operator and an aggregator of a large number of distributed energy resources, without risking to share their private information. As a new real-time aggregate flexibility design, it outperforms traditional MPC with lower electricity costs and computational/communication complexity.

Furthermore, in Chapter 5, treating the information theoretic-based feedback as a penalty term in the objective function of MPC that summarizes and simplifies constraints, the learning-based approach is showed to have a sub-linear regret under certain conditions. This set of results reveals insights that with the penetration of renewable energy resources, learning-based control has the potential to facilitate operator-aggregator coordination, and improves traditional control policies in power systems.

1.5 Learning, Inference, and Data Analysis in Smart Grids

Identifying and inferring system information of a power network such as its susceptence matrix or nodal admittance matrix from voltage/current measurements is a critical problem. We first consider a power system identification problem. Lower and

upper bounds on sample complexity are provided, together with an iterative algorithm that takes advantage of useful properties of the admittance matrix and sparsity of the graph. It outperforms classical basis pursuit in terms of sample complexity for matrix recovery. There is little exploration on the fundamental performance limits (estimation error and sample complexity) in the literature on topology and parameter reconstruction of power networks. In an attempt to shed some light on this problem, in Chapter 6, we consider a specific graph learning task: reconstructing a symmetric matrix that represents an underlying graph using linear measurements. We present a sparsity characterization for distributions of random graphs (that are allowed to contain *high-degree* nodes), based on which we study fundamental trade-offs between the number of measurements, the complexity of the graph class, and the probability of error. We first derive a necessary condition on the number of measurements. Then, by considering a three-stage recovery scheme, we give a sufficient condition for recovery. Furthermore, assuming the measurements are Gaussian IID, we prove upper and lower bounds on the (worst-case) sample complexity for both noisy and noiseless recovery. In the special cases of the uniform distribution on trees with n nodes and the Erdős-Rényi (n, p) class, the fundamental trade-offs are tight up to multiplicative factors with noiseless measurements. In addition, for practical applications, we design and implement a polynomial-time (in n) algorithm based on the three-stage recovery scheme.

The results presented in Chapter 6 close the theory-practice gap of power system identification by providing theoretical analysis on the recovery of graph Laplacian that may contain high-degree nodes.

Finally, in Chapter 7, we introduce ACN-Data that has been used frequently in EV charging research including the learning-augmented problems considered in previous chapters, together with a method that first extracts tails from a diversity of charging time series that have different lengths, contain missing data, and are distorted by scheduling algorithms and measurement noise. The charging tails are then clustered into a small number of types whose representatives are then used to improve tail extraction. This process iterates until it converges.

1.6 Dissertation Outline

The outline of the remainder of this dissertation is given in Figure 1.1.

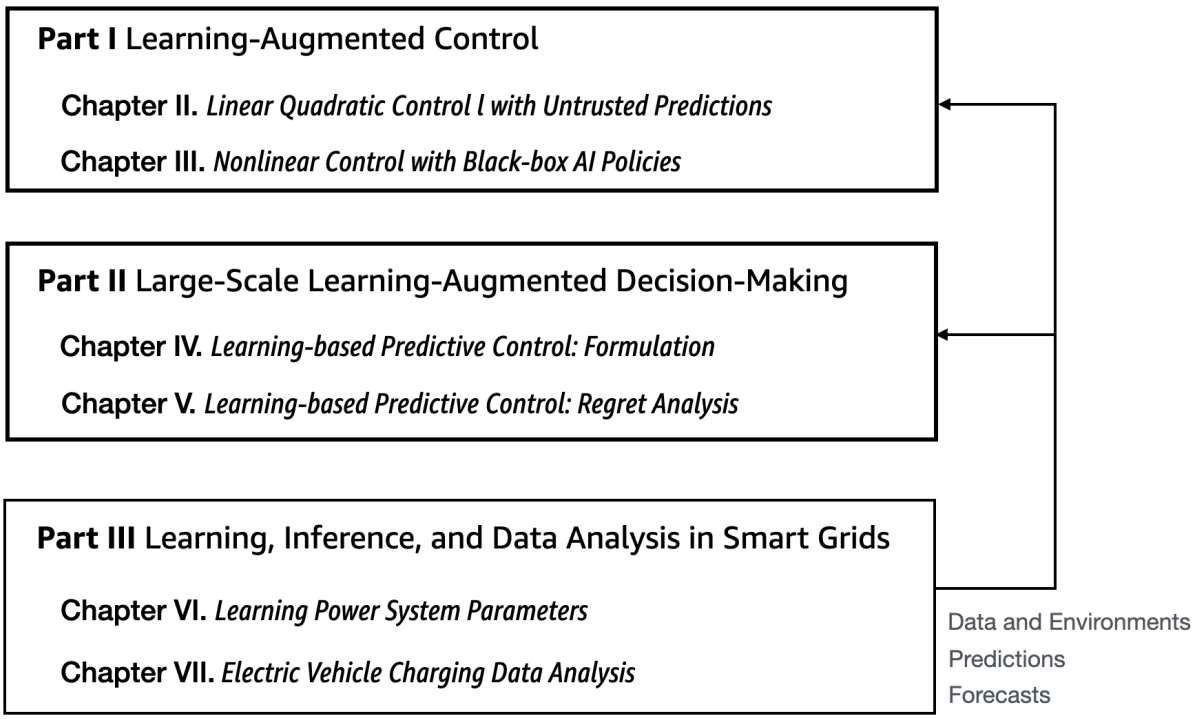


Figure 1.1: Outline of this dissertation. Part I considers learning-augmented control problems for both linear and non-linear models. Part II of this dissertation focuses on large-scale learning-augmented decision making problems. The details of data, environments, predictions, and forecasts used in the applications presented in the first two parts are described in Part III.

Part I

Learning-Augmented Control

Chapter 2

LINEAR QUADRATIC CONTROL WITH UNTRUSTED AI PREDICTIONS

[1] Tongxin Li, Ruixiao Yang, Guannan Qu, Guanya Shi, Chenkai Yu, Adam Wierman, and Steven Low. Robustness and consistency in linear quadratic control with untrusted predictions. 6(1), 2022. URL https://doi.org/10.1145/3508038.

In this chapter, we study a classical online linear quadratic control problem where the controller has access to untrusted predictions/advice during each round, potentially from a black-box AI tool.

2.1 Introduction

One consequence of the success of machine learning is that accurate predictions are available for many online decision and control problems. For example, the development of deep learning has enabled generic prediction models in various domains, e.g. weather, demand forecast, and user behaviors. Such predictions are powerful because future information plays a significant role in optimizing the current control decision. The availability of accurate future predictions can potentially lead to order-of-magnitude performance improvement in decision and control problems, where one can simply plug-in the predictions and achieve near optimal performance when compared to the best control actions in hindsight, a.k.a., *consistency*. However, an important caveat is that the predictions are helpful only when they are accurate, which is not guaranteed in many scenarios. Since many predictions are obtained from black box AI models like neural networks, there is no uncertainty quantification and it is unclear whether the predictions are accurate. In the case when the predictions are not accurate, the consequences can be catastrophic, leading to unbounded worst-case performance, e.g., an unbounded competitive ratio. The possibility of such worst-case unbounded competitive ratio prevents the use of ML predictions in safety-critical applications that are adverse to potential risks.

The use of predictions described above is a sharp contrast to the approaches developed by the online algorithm community, where the algorithms have access to no future prediction, yet can be *robust* to all future variations and achieve a finite competitive

ratio. While such algorithms miss out on the improvements possible when accurate predictions are available, their robustness properties are necessary in safety-critical settings. Therefore, a natural question arises:

Can such adversarial guarantees be provided for control policies that use black-box AI predictions?

To provide adversarial guarantees necessarily means not precisely following the black-box AI predictions. Thus, there must be a trade-off between the performance in the typical case (consistency) and the quality of the adversarial guarantee (robustness). Trade-offs between consistency and robustness have received considerable attention in recent years in the online algorithms community, starting with the work of [3], but our work represents the first work in the context of control.

Contributions. In this chapter, we answer the question above in the affirmative, in the context of linear quadratic control, providing an novel algorithm that trades off consistency and robustness to provide adversarial gaurantees on the use of untrusted predictions.

Our first result provides a novel online control algorithm, termed *λ-confident control*, that provides a competitive ratio of $1 + \min\{O(\lambda^2\varepsilon) + O(1-\lambda)^2, O(1) + O(\lambda^2)\}$, where $\lambda \in [0, 1]$ is a *trust parameter* set based on the confidence in the predictions, and ε is the prediction error (Theorem 5.5.3). When the predictions are accurate ($\varepsilon \approx 0$), setting λ close to 1 will obtain a competitive ratio close to 1, and hence the power of the predictions is fully utilized; on the other hand, when the predictions are inaccurate (ε very large), setting $\lambda \approx 0$ will still guarantee a constant competitive ratio, meaning the algorithm will still have good robustness guarantees when the predictions turn out to be bad. Therefore, our approach can get the best of both worlds, effectively using black-box predictions but still guaranteeing robustness.

The above discussion highlights that the optimal choice of λ depends on the prediction error, which may not be known a priori. Therefore, we further provide an adaptive, self-tuning learning policy (Algorithm 3) that selects λ so as to learn the optimal parameter for the actual prediction error; thus selecting the optimal balance between robustness and consistency. Our main result proves that the self-tuning policy maintains a competitive ratio bound that is always bounded regardless of the prediction error ε (Theorem 2.4.1). This result is informally presented below.

Theorem (Informal)**.** *Under our model assumptions, there is a self-tuning online control algorithm that selects some* $\lambda_t \in [0, 1]$ *for all* $t = 0, \ldots, T-1$ *and achieves a competitive ratio*

$$\mathsf{CR}(\varepsilon) \leq 1 + \frac{O(\varepsilon)}{\Theta(1) + \Theta(\varepsilon)} + O(\mu_{\mathsf{Var}})$$

as a function of the prediction error ε *where* μ_{Var} *measures the variation of perturbations and predictions.*

This result provides a worst-case performance bound for the use of untrusted predictions, e.g., the predictions from a black-box AI tool, regardless of the accuracy of the predictions. The second term in the competitive ratio upper bound indicates a nontrivial non-linear dependency of $\mathsf{CR}(\varepsilon)$ and the prediction error ε, matching our experimental results shown in Section 2.5. The third term measures the variation of perturbations and predictions. Such a term is common in regret analysis based on the "Follow The Leader" (FTL) approach [37, 38]. For example, the regret analysis of the Follow the Optimal Steady State (FOSS) method in [39] contains a similar "path length" term that captures the variation of the state trajectory.

Proving our main result is complex due to the fact that, different from classical online learning models, the cost function in our problem depends on previous actions via a linear dynamical system (see (3.9)). The time coupling can even be exponentially large if the dynamical system is unstable. To tackle this time-coupling structure, we develop a new proof technique that relates the regret and competitive ratio with the convergence rate of the trust parameter.

Finally, in Section 2.5 we demonstrate the effectiveness of our self-tuning approach using three examples: a robotic tracking problem, an adaptive battery-buffered EV charging problem and the Cart-Pole problem. For the robotic tracking and adaptive battery-buffered EV charging cases, we illustrate that the competitive ratio of the self-tuning policy performs nearly as well as the lower envelope formed by picking multiple trust parameters optimally offline. We also validate the practicality of our self-tuning policy by showing that it not only works well for linear quadratic control problems; it also performs well in the non-linear Cart-Pole problem.

Related Work. Our work contributes to the growing literature on learning-augmented online algorithm design. There has been significant interest in the goal of trading-off consistency and robustness in order to ensure worst-case performance bounds for black-box AI tools in online problems. As discussed earlier, prediction based

algorithms can achieve consistency, while online algorithms can have robustness. These two classes of algorithms can be viewed as two extremes, and a number of works attempt to develop algorithms that balance between consistency and robustness in settings like online caching [3], ski-rental [4–7], online set cover [7], secretary and online matching [8], and metric task systems [9]. For example, in the ski rental problem, [5] proposes an algorithm that achieves $1 + \lambda$ consistency and $1 + \frac{1}{\lambda}$ robustness for a tuning parameter $\lambda \in (0, 1)$. Compared to these works, our setting is fundamentally more challenging because of the existence of dynamics in the control problem couples all decision points, and a mistake at one time can be magnified and propagated to all future time steps.

Our work is also closely related to a broad literature on regret and competitive ratio analysis for Linear Quadratic Control (LQC) and Linear Quadratic Regulator (LQR) systems with predictions. In [40], LQR regret analysis for Model Predictive Control (MPC) is given, assuming accurate predictions of perturbations. Inaccurate predictions are considered in [41], with competitive results provided. It is proven in [40, 41] that the action generated by MPC can be explicitly written as the action of optimal linear control plus a linear combination of inaccurate predictions. The competitive analysis of the consistent and robust control scheme in this work makes use of this fact in the analysis of the more challenging case of untrusted predictions. Other related regret and competitive ratio results for MPC include [25, 42, 43].

While our work is the first to study learning-augmented control via the lens of robustness and consistency, there are two classical communities in control that are related to the goals of our work: robust control and adaptive control.

Robust control is a large area that concerns the design of controllers with performance guarantees that are robust against model uncertainty or adversarial disturbances [44]. Tools of robust control include H_∞ synthesis [45, 46] and robust MPC [47]. Like the robust control literature, our work also considers robustness, but our main focus is on balancing between robustness and consistency in a predictive control setting. Consistency is not a focus of the robust control literature. Further, we focus on the metrics of competitive ratio and regret, which is different from the typical performance measures in the robust control literature, which focus on measures such as system norms [46].

The design of our self-tuning control in Section 2.4 falls into the category of adaptive control. There is a rich body of literature studying Lyapunov stability and asymptotic convergence in adaptive control theory [48]. Recently, there has been increasing

interest in studying adaptive control with non-asymptotic metrics from learning theory. Typical results guarantee convergence in finite time horizons using measures such as regret [49–52], dynamic regret [39, 40, 43], and competitive ratio [41, 42]. Different from these works, this chapter deploys an adaptive policy with the goal of balancing robustness and consistency. Additionally, such results do not focus on incorporation of untrusted predictions.

2.2 Model

We consider a Linear Quadratic Control (LQC) model. Throughout the chapter, $\|\cdot\|$ denotes the ℓ_2-norm for vectors and the matrix norm induced by the ℓ_2-norm. Denote by $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ the system state and action at each time t. We consider a linear dynamic system with adversarial perturbations,

$$x_{t+1} = Ax_t + Bu_t + w_t, \text{ for } t = 0, \ldots, T-1, \tag{2.1}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, and $w_t \in \mathbb{R}^n$ denotes some unknown perturbation chosen adversarially. We make the standard assumption that the pair (A, B) is stabilizable. Without loss of generality, we also assume the system is initialized with some fixed $x_0 \in \mathbb{R}^n$. The goal of control is to minimize the following quadratic costs given matrices A, B, Q, R :

$$J := \sum_{t=0}^{T-1} (x_t^\top Q x_t + u_t^\top R u_t) + x_T P x_T,$$

where $Q, R \succ 0$ are positive definite matrices, and P is the solution of the following discrete algebraic Riccati equation (DARE), which must exist because (A, B) is stabilizable and $Q, R \succ 0$ [44].

$$P = Q + A^\top P A - A^\top P B (R + B^\top P B)^{-1} B^\top P A.$$

Given P, we can define $K := (R + B^\top P B)^{-1} B^\top P A$ as the optimal LQC controller in the case of no disturbance ($w_t = 0$). Further, let $F := A - BK$ be the closed-loop system matrix when using $u_t = -Kx_t$ as the controller. By [44], F must have a spectral radius $\rho(F)$ less than 1. Therefore, Gelfand's formula implies that there must exist a constant $C > 0$, $\rho \in (0, 1)$ s.t. $\left\|F^t\right\| \leq C\rho^t, \forall t \geq 0$.

Our model is a classical control model [46] and has broad applicability across various engineering fields. In the following, we introduce ML/AI predictions into the classical model and study the trade-off between consistency and robustness in this classical model for the first time.

Untrusted Predictions

Our focus is on predictive control and we assume that, at the beginning of the control process, a sequence of predictions of the disturbances $(\widehat{w}_0, \ldots, \widehat{w}_{T-1})$ is given to the decision maker. At time t, the decision maker observes x_t, w_{t-1} and picks a decision u_t. Then, the environment picks w_t, and the system transitions to the next step according to (2.1). We emphasize that, at time t, the decision maker has no access to $(w_t, \ldots, w_T)$ and their values may be different from the predictions $(\widehat{w}_t, \ldots, \widehat{w}_T)$. Also, note that w_t can be adversarially chosen at each time t, adaptively.

The assumption that a sequence of predictions available is $(\widehat{w}_t, \ldots, \widehat{w}_T)$ is not as strong as it may first appear, nor as strong as other similar assumptions made in literature, e.g., [39, 40], because we allow for prediction error. If there are no predictions or only a subset of predictions are available, we can simply set the unknown predictions to be zero and this does not affect our theoretical results and algorithms.

In our model, there are two types of uncertainty. The first is caused by the *perturbations* because the future perturbations $(w_t, \ldots, w_{T-1})$ are unknown to the controller at time t. The second is the *prediction error* due to the mismatch $e_t \coloneqq \widehat{w}_t - w_t$ between the perturbation w_t and the prediction $\widehat{w}_t$ at each time. Formally, we define the prediction error as

$$\varepsilon\left(F, P, e_0, \ldots, e_{T-1}\right) \coloneqq \sum_{t=0}^{T-1} \left\| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P e_t \right\|^2 . \tag{2.2}$$

Notice that the prediction error is not defined as a form of classical mean squared error for our problem. The reason is because the mismatch e_t at each time has different impact on the system. Writing the prediction error as in (2.2) simplifies our analysis. In fact, if we define u_t and $\widehat{u}_t$ as two actions given by an optimal linear controller (formally defined in Section 2.2) as if the true perturbations are $w_0, \ldots, w_{T-1}$ and $\widehat{w}_0, \ldots, \widehat{w}_{T-1}$, respectively, then it can be verified that $\varepsilon = \sum_{t=0}^{T-1} \|u_t - \widehat{u}_t\|$, which is the accumulated action mismatch for an optimal linear controllers provided with different estimates of perturbations. In Section 2.5, using experiments, we show that the competitive ratios (with a fixed "trust parameter" defined in 2.3) grow linearly in the prediction error ε defined in (2.2). Finally, we assume that the perturbations $(w_0, \ldots, w_{T-1})$ and predictions $(\widehat{w}_0, \ldots, \widehat{w}_{T-1})$ are uniformly bounded, i.e., there exist $\overline{w} > 0$ and $\widehat{w} > 0$ such that $\|w_t\| \leq \overline{w}$ and $\|\widehat{w}_t\| \leq \widehat{w}$ for all $0 \leq t \leq T-1$. In summary, Figure 2.1 demonstrates the system model considered in this chapter.

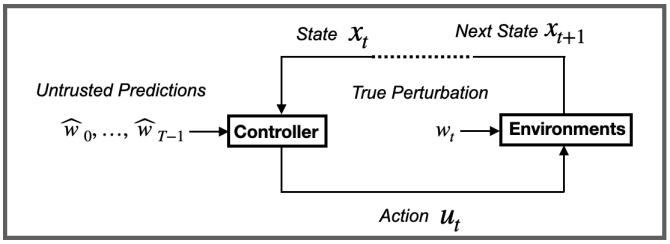


Figure 2.1: System model of linear quadratic control with untrusted predictions.

Defining Consistency and Robustness

As discussed in the in introduction, while predictions can be helpful, inaccurate predictions can lead to unbounded competitive ratio. Our goal is to utilize predictions to achieve good performance (consistency) while still providing adversarial worst-case guarantees (robustness). In this subsection, we formally define the notions of consistency and robustness we study. These notions have received increasing attention recently in the area of online algorithms with untrusted advice, e.g., [4–9].

We use the competitive ratio to measure the performance of an online control policy and quantify its robustness and consistency. Specifically, let OPT be the offline optimal cost when all the disturbances $(w_t)_{t=0}^{T}$ are known, and ALG be the cost achieved by an online algorithm. Throughout this chapter, we assume that $\mathsf{OPT} > 0$. We define the competitive ratio for a given bound on the prediction error ε, as follows.

Definition 2.2.1. *The **competitive ratio** for a given prediction error ε, $\mathsf{CR}(\varepsilon)$, is defined as the smallest constant $C \geq 1$ such that $\mathsf{ALG} \leq C \cdot \mathsf{OPT}$ for fixed A, B, Q, R and any adversarially and adaptively chosen perturbations $(w_0, \ldots, w_{T-1})$ and predictions $(\widehat{w}_0, \ldots, \widehat{w}_{T-1})$.*

Building on the definition of competitive ratio, we define robustness and consistency as follows.

Definition 2.2.2. *An online algorithm is said to be γ-**robust** if, for any prediction error $\varepsilon > 0$, the competitive ratio satisfies $\mathsf{CR}(\varepsilon) \leq \gamma$, and an algorithm is said to be β-**consistent** if the competitive ratio satisfies $\mathsf{CR}(0) \leq \beta$.*

Background: Existing Algorithms

Before proceeding to our algorithm and its analysis, we first introduce two extreme algorithm choices that have been studied previously: a myopic policy that we refer to as 1-confident control, which places full trust in the predictions, and a pure online strategy that we refer to as 0-confident control, which places no trust in the

predictions. These represent algorithms that can achieve consistency and robustness *individually*, but cannot achieve consistency and robustness *simultaneously*. The key challenge of this work is to understand how to integrate ideas such as what follows into an algorithm that achieves consistency and robustness simultaneously.

A Consistent Algorithm: 1-Confident Control

A simple way to achieve consistency is to put full faith in the untrusted predictions. In particular, if the algorithm trusts the untrusted predictions and follows them, the performance will always be optimal if the predictions are accurate. We refer to this as the 1-confident policy, which is defined by a finite-time optimal control problem that trusts that $(\widehat{w}_0, \dots, \widehat{w}_{T-1})$ are the true disturbances. Formally, at time step t, the actions $(u_t, \dots, u_T)$ are computed via

$$\underset{(u_t,\dots,u_{T-1})}{\arg\min} \left(\sum_{\tau=t}^{T-1} (x_\tau^\top Q x_\tau + u_\tau^\top R u_\tau) + x_T P x_T \right) \text{ s.t. (2.1) for all } \tau = t, \dots, T-1. \tag{2.3}$$

With the obtained solution $(u_t, \dots, u_{T-1})$, the control action u_t at time t is fixed to be u_t and the other actions $(u_{t+1}, \dots, u_{T-1})$ are discarded.

We highlight the following result (Theorem 3.2 in [40]) that provides an explicit expression of the algorithm in (2.3), which can be viewed as a form of Model Predictive Control (MPC).

Theorem 2.2.1 (Theorem 3.2 in [40])**.** *With predictions* $(\widehat{w}_0, \dots, \widehat{w}_{T-1})$ *fixed, the solution* u_t *of the algorithm in* (2.3) *can be expressed as*

$$u_t = -(R + B^\top P B)^{-1} B^\top \left(P A x_t + \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \widehat{w}_\tau \right) \tag{2.4}$$

where $F \coloneqq A - B(R + B^\top P B)^{-1} B^\top P A = A - BK$.

It is clear that this controller (2.3) (or equivalently (2.4)) achieves 1-consistency because, when the prediction errors are 0, the control action from (2.3) (and the state trajectory) will be exactly the same as the offline optimal. However, this approach is not robust, and one can show that prediction errors can lead to unbounded competitive ratios. In the next subsection, we introduce a robust (but non consistent) controller.

A Robust Algorithm: 0-Confident Control

On the other extreme, a natural way to be robust is to ignore the untrusted predictions entirely, i.e., place no confidence in the predictions. The 0-confident policy does exactly this. It places no trust in the predictions and synthesizes the controller by assuming $w_t = 0$. Formally, the policy is given by

$$u_t = -Kx_t. \tag{2.5}$$

This recovers the optimal pure online policy in classical linear control theory [53]. As shown by [40], this controller has a constant competitive ratio and therefore is $O(1)$-robust. However, this approach is not consistent as it does not utilize the predictions at all. In the next section, we discuss our proposed approach, which achieves both consistency and robustness.

2.3 Consistent and Robust Control

The goal in this chapter is to develop a controller that performs near-optimally when predictions are accurate (consistency) and meanwhile is robust when the prediction error is large. As discussed in the previous section, a myopic, 1-confident controller that puts full trust into the predictions is consistent, but not robust. On the other hand, any purely online 0-confident policy that ignores predictions is robust but not consistent.

The algorithms we present establish a trade-off between these extremes by including a "confidence/trust level" for the predictions. The algorithm design challenge is to determine the right way to balance these extremes. In the first (warmup) algorithm, the policy starts out confident in the predictions, but when a threshold of error is observed, the policy loses confidence and begins to ignore predictions. This simple threshold-based policy highlights that it is possible for a policy to be both robust and consistent. However, the result also highlights the weakness of the standard notions of robustness and consistency since the policy cannot make use of intermediate quality predictions and only performs well in the extreme cases when predictions are either perfect or poor.

Thus, we move to considering a different approach, which we term *λ-confident control*. This algorithm selects a confidence level λ that serves as a weight for a linear combination between purely myopic 1-confident control and purely online 0-confident control. Our main result shows that this policy provides a smooth trade-off between robustness and consistency and, further, in Section 2.4, we show that the

Algorithm 1: Threshold-Based Control

Initialize $\delta = 0$
for $t = 0, \ldots, T-1$ **do**
  if $\delta < \sigma$ **then**
    $u_t = -(R + B^\top P B)^{-1} B^\top \left(P A x_t + \sum_{\tau=t}^{T-1} (F^\top)^{\tau - t} P \widehat{w}_\tau \right)$
  else
    Compute u_t with the best myopic online algorithm $\mathcal{A}_{\text{Online}}$ without predictions
  end
  Update $x_{t+1} = A x_t + B u_t + w_t$ and $\delta \leftarrow \delta + \|\widehat{w}_t - w_t\|$
end

confidence level λ can be learned online adaptively so as to achieve consistency and robustness without exogenously specifying a trust level.

Warmup: Threshold-Based Control

We begin by presenting a simple threshold-based algorithm that can be both robust and consistent, though it does not perform well for predictions of intermediate quality. This distinction highlights that looking beyond the classical narrow definitions of robustness and consistency is important when evaluating algorithms.

The threshold-based algorithm is described in Algorithm 1. It works by trusting predictions (using 1-confident control update (2.4)) until a certain error threshold $\sigma > 0$ is crossed and then ignoring predictions (using an online algorithm $\mathcal{A}_{\text{Online}}$ that attains a (minimal) competitive ratio $C_{\min}$[1] for all online algorithms that do not use predictions). The following result shows that, with a small enough threshold, this algorithm is both robust and consistent because, if predictions are perfect, it trusts them entirely, but if there is an error, it immediately begins to ignore predictions and matches the 0-confident controller performance, which is optimal. A proof can be found in Appendix 2.D.

Theorem 2.3.1. *There exists a threshold parameter $\sigma > 0$ such that Algorithm 1 is 1-consistent and $(C_{\min} + o(1))$-robust, where $C_{\min}$ is the minimal competitive ratio of any pure online algorithm.*

[1]Note that $C_{\min}$ is guaranteed to exist, as setting $\lambda = 0$ in Theorem 5.5.3 gives a constant $1 + \|H\|/\lambda_{\min}(G)$ competitive ratio bound for the 0-confident control update (2.5), therefore $1 \leq C_{\min} \leq 1 + \|H\|/\lambda_{\min}(G)$.

Algorithm 2: λ-Confident Control

for $t = 0, \ldots, T-1$ **do**

$$\text{Take } u_t = -(R + B^\top P B)^{-1} B^\top \left(P A x_t + \lambda \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \widehat{w}_\tau \right)$$

$$\text{Update } x_{t+1} = A x_t + B u_t + w_t$$

end

The term $o(1)$ in Theorem 2.3.1 converges to 0 as $T \to \infty$. While Algorithm 1 is optimally robust and consistent, it is unsatisfying because it does not improve over the online algorithm unless predictions are perfect since in the proof, we set the threshold parameter $\sigma > 0$ arbitrarily small to make the algorithm robust and 1-consistent and the definition of consistency and robustness only captures the behavior of the competitive ratio $\mathsf{CR}(\varepsilon)$ for either $\varepsilon = 0$ or ε is large. As a result, in the remainder of this chapter we look beyond the extreme cases and prove results that apply for arbitrary prediction error quality. In particular, we prove competitive ratio bounds that hold for arbitrary ε, of which consistency and robustness are then special cases.

λ-Confident Control

We now present our main results, which focus on a policy that, like Algorithm 1, looks to find a balance between the two extreme cases of 1-confident and 0-confident control. However, instead of using a threshold to decide when to swap between them, the λ-confident controller considers a linear combination of the two.

Specifically, the policy presented in Algorithm 2 works as follows. Given a *trust parameter* $0 \leq \lambda \leq 1$, it implements a linear combination of (2.4) and (2.5). Intuitively, the selection of λ allows a trade-off between consistency and robustness based on the extent to which the predictions are trusted. Our main result shows a competitive ratio bound that is consistent with this intuition. A proof is given in Appendix 2.B.

Theorem 2.3.2. *Under our model assumptions, with a fixed trust parameter $\lambda > 0$, the λ-confident control in Algorithm 2 has a worst-case competitive ratio of at most*

$$\mathsf{CR}(\varepsilon) \leq 1 + 2\|H\| \min\left\{ \left(\frac{\lambda^2}{\mathsf{OPT}} \varepsilon + \frac{(1-\lambda)^2}{C} \right), \left(\frac{1}{C} + \frac{\lambda^2}{\mathsf{OPT}} \overline{W} \right) \right\} \tag{2.6}$$

where $H \coloneqq B(R + B^\top P B)^{-1} B^\top$, OPT *denotes the optimal cost,* $C > 0$ *is a constant that depends on* A, B, Q, R *and*

$$\varepsilon\left(F, P, e_0, \ldots, e_{T-1}\right) \coloneqq \sum_{t=0}^{T-1} \left\| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P\left(w_\tau - \widehat{w}_\tau\right) \right\|^2, \tag{2.7}$$

$$\overline{W}\left(F, P, \widehat{w}_0, \ldots, \widehat{w}_{T-1}\right) \coloneqq \sum_{t=0}^{T-1} \left\| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \widehat{w}_\tau \right\|^2.$$

From this result, we see that λ-confident control is guaranteed to be $\left(1 + \|H\| \frac{(1-\lambda)^2}{C}\right)$-consistent and $\left(1 + \|H\|\left(\frac{1}{C} + \frac{\lambda^2}{\mathsf{OPT}} \overline{W}\right)\right)$-robust. This highlights a trade-off between consistency and robustness such that if a large λ is used (i.e., predictions are trusted), then consistency decreases to 1, while the robustness increases unboundedly. In contrast, when a small λ is used (i.e., predictions are distrusted), the robustness of the policy converges to the optimal value, but the consistency does not improve on the robustness value. Due to the time-coupling structure in the control system, the mismatches $e_t = \widehat{w}_t - w_t$ at different times contribute unequally to the system. As a result, the prediction error ε in (2.2) and (2.7) is defined as a weighed quadratic sum of $(e_0, \ldots, e_{T-1})$. Moreover, the term OPT in (2.6) is common in the robustness and consistency analysis of online algorithms, such as [3, 5, 7, 9].

2.4 Self-Tuning λ-Confident Control

While the λ-confident control finds a balance between consistency and robustness, selecting the optimal λ parameter requires exogenous knowledge of the quality of the predictions ε, which is often not possible. For example, black-box AI tools typically do not allow uncertainty quantification. In this section, we develop a self-tuning λ-confident control approach that learns to tune λ in an online manner. We provide an upper bound on the regret of the self-tuning λ-confident control, compared with using the best possible λ in hindsight, and a competitive ratio for the complete self-tuning algorithm. These results provide the first worst-case guarantees for the integration of black-box AI tools into linear quadratic control.

Our policy is described in Algorithm 3 and is a "follow the leader" approach [38]. At each time $t = 0, \ldots, T-1$, it selects a λ_t in order to minimize the gap between ALG and OPT in the previous t rounds and chooses an action using the trust parameter λ_t. Then the state x_t is updated to x_{t+1} using the linear system dynamic in (2.1) and this process repeats. Note that the denominator of λ_t is zero if and only if $\eta\left(\widehat{w}; s, t-1\right) = 0$ for all s. To make λ_t well-defined, we set $\lambda = 1$ for this case.

Algorithm 3: Self-Tuning λ-Confident Control

for $t = 0, \ldots, T-1$ **do**
  if $t = 0$ *or* $t = 1$ **then**
    Initialize and choose λ_0
  end
  else
    Compute a trust parameter λ_t

$$\lambda_t = \frac{\sum_{s=0}^{t-1} \left(\eta(w; s, t-1)\right)^\top H \left(\eta(\widehat{w}; s, t-1)\right)}{\sum_{s=0}^{t-1} \left(\eta(\widehat{w}; s, t-1)\right)^\top H \left(\eta(\widehat{w}; s, t-1)\right)}$$

$$\text{where } \eta(w; s, t) \coloneqq \sum_{\tau=s}^{t} \left(F^\top\right)^{\tau-s} P w_\tau$$

  end
  Generate an action u_t using λ_t-confident control (Algorithm 2)
  Update $x_{t+1} = Ax_t + Bu_t + w_t$
end

The key to the algorithm is the update rule for λ_t. Given previously observed perturbations and predictions, the goal of the algorithm is to find a greedy λ_t that minimizes the gap between the algorithmic and optimal costs so that $\lambda_t \coloneqq \min_\lambda \sum_{s=0}^{t-1} \psi_s^\top H \psi_s$ where $\psi_s \coloneqq \sum_{\tau=s}^{t-1} \left(F^\top\right)^{\tau-s} P \left(w_\tau - \lambda \widehat{w}_\tau\right)$. This can be equivalently written as

$$\lambda_t = \arg\min_\lambda \sum_{s=0}^{t-1} \left[\left(\sum_{\tau=s}^{t-1} \left(F^\top\right)^{\tau-s} P(w_\tau - \lambda \widehat{w}_\tau) \right)^\top H \left(\sum_{\tau=s}^{t-1} \left(F^\top\right)^{\tau-s} P(w_\tau - \lambda \widehat{w}_\tau) \right) \right], \tag{2.8}$$

which is a quadratic function of λ. Rearranging the terms in (2.8) yields the choice of λ_t in the self-tuning control scheme.

Algorithm 3 is efficient since, in each time step, updating the η values only requires adding one more term. This means that the total computational complexity of λ_t is $O(T^2 n^\alpha)$, where $\alpha < 2.373$, which is polynomial in both the time horizon length T and state dimension n. According to the expression of λ_t in Algorithm 3, at each time t, the terms $\eta(w; s, t-2)$ and $\eta(\widehat{w}; s, t-2)$ can be pre-computed for all $s = 0, \ldots, t-1$. Therefore, the recursive formula $\eta(w; s, t) \coloneqq \sum_{\tau=s}^{t} \left(F^\top\right)^{\tau-s} P w_\tau = \eta(w; s, t-1) + \left(F^\top\right)^{t-s} P w_t$ implies the update rule of the terms $\{\eta(w; s, t-1) : s = 0, \ldots, t-1\}$ in the expression of λ_t. This gives that, at each time t, it takes no more than $O(Tn^\alpha)$ steps to compute λ_t where $\alpha < 2.373$ and $O(n^\alpha)$ is the computational complexity of matrix multiplication.

Convergence

We now move to the analysis of Algorithm 3. First, we study the convergence of λ_t, which depends on the variation of the predictions $\widehat{\mathbf{w}} \coloneqq (\widehat{w}_0, \dots, \widehat{w}_{T-1})$ and the true perturbations $\mathbf{w} \coloneqq (w_0, \dots, w_{T-1})$, where we use a boldface letter to represent a sequence of vectors. Specifically, our results are in terms of the variation of the predictions and perturbations, which we define as follows. The *self-variation* $\mu_{\mathsf{VAR}}(\mathbf{y})$ of a sequence $\mathbf{y} \coloneqq (y_0, \dots, y_{T-1})$ is defined as

$$\mu_{\mathsf{VAR}}(\mathbf{y}) \coloneqq \sum_{s=1}^{T-1} \max_{\tau=0,\dots,s-1} \|y_\tau - y_{\tau+T-s}\| .$$

The goal of the self-tuning algorithm is to converge to the optimal trust parameter λ^* for the problem instance. To specify this formally, let $\mathsf{ALG}(\lambda_0, \dots, \lambda_{T-1})$ be the algorithmic cost with adaptively chosen trust parameters $\lambda_0, \dots, \lambda_{T-1}$ and denote by $\mathsf{ALG}(\lambda)$ the cost with a fixed trust parameter λ. Then, λ^* is defined as $\lambda^* \coloneqq \min_{\lambda \in \mathbb{R}} \mathsf{ALG}(\lambda)$. Further, let $W(t) \coloneqq \sum_{s=0}^{t} \eta(\widehat{w}; s, t)^\top H \eta(\widehat{w}; s, t)$.

We can now state a bound on the convergence rate of λ_t to λ^* under Algorithm 3. The bound highlights that if the variation of the system perturbations and predictions is small, then the trust parameter λ_t converges quickly to λ^*. A proof can be found in Appendix 2.C.

Lemma 1. *Assume $W(T) = \Omega(T)$ and $\lambda_t \in [0, 1]$ for all $t = 0, \dots, T-1$. Under our model assumptions, the adaptively chosen trust parameters $(\lambda_0, \dots, \lambda_T)$ by self-tuning control satisfy that for any $1 < t \leq T$,*

$$|\lambda_t - \lambda^*| = O\left(\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)/t\right) .$$

Regret and Competitiveness

Building on the convergence analysis, we now prove bounds on the regret and competitive ratio of Algorithm 3. These are the main results presented in this chapter about the performance of an algorithm that adaptively determines the optimal trade-off between robustness and consistency.

Regret. We first study the regret as compared with the best, fixed trust parameter in hindsight, i.e., λ^*, whose corresponding worst-case competitive ratio satisfies the upper bound given in Theorem 2.3.2.

Denote by $\mathsf{Regret} \coloneqq \mathsf{ALG}(\lambda_0, \dots, \lambda_{T-1}) - \mathsf{ALG}(\lambda^*)$ the *regret* we consider where $(\lambda_0, \dots, \lambda_{T-1})$ are the trust parameters selected by the self-tuning control scheme.

Our main result is the following variation-based regret bound, which is proven in Appendix 2.C. Define the denominator of λ_t in Algorithm 2 by $W(t) \coloneqq \sum_{s=0}^{t} \eta(\widehat{w}; s, t)^\top H \eta(\widehat{w}; s, t)$.

Lemma 2. *Assume $W(t) = \Omega(T)$ and $\lambda_t \in [0, 1]$ for all $t = 0, \ldots, T-1$. Under our model assumptions, for any $\mathbf{w}$ and $\widehat{\mathbf{w}}$, the regret of Algorithm 3 is bounded by*

$$\mathsf{Regret} = O\left(\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)^2\right).$$

Note that the baseline we evaluate against in $\mathsf{Regret} = \mathsf{ALG}(\lambda_0, \ldots, \lambda_{T-1}) - \mathsf{ALG}(\lambda^*)$ is stronger than baselines in previous *static regret* analysis for LQR, such as [52, 54] where online controllers are compared with a linear control policy $u_t = -Kx_t$ with a strongly stable K. The baseline policy considered in our regret analysis is the λ-confident scheme (Algorithm 2) with

$$\begin{aligned} u_t &= -(R + B^\top P B)^{-1} B^\top \left(P A x_t + \lambda \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \widehat{w}_\tau \right) \\ &= -Kx_t - \lambda (R + B^\top P B)^{-1} B^\top \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \widehat{w}_\tau, \end{aligned}$$

which contains the class of strongly stable linear controllers as a special case. Moreover, the regret bound in Lemma 2 holds for any predictions $\widehat{w}_0, \ldots, \widehat{w}_{T-1}$. Taking $\widehat{w}_t = w_t$ for all $t = 0, \ldots, T-1$, our regret directly compares $\mathsf{ALG}(\lambda_0, \ldots, \lambda_{T-1})$ with the optimal cost OPT, and therefore, our regret also involves the *dynamic regret* considered in [39, 40, 43] for LQR as a special case. The regret bound in Lemma 2 depends on the variation of perturbations and predictions. Note that such a term commonly exists in regret analysis based on the "follow the leader" approach [37, 38]. For example, the regret analysis of the follow the optimal steady state (FOSS) method in [39] contains a similar "path length" term that captures the state variation. There is a variation budget of the predictions or prediction errors in theorem 1 of [43]. In many robotics applications (e.g., the trajectory tracking and EV charging experiments in this chapter shown in Section 2.5), each w_t is from some desired smooth trajectory to track.

To interpret this lemma, suppose the sequences of perturbations and predictions satisfy:

$$\begin{aligned} &\|\widehat{w}_\tau - \widehat{w}_{\tau+T-s}\| \le \rho(s), \\ &\|w_\tau - w_{\tau+T-s}\| \le \rho(s), \text{ for any } s \ge 0, 0 \le \tau \le s. \end{aligned}$$

These bounds correspond to an assumption of smooth variation in the disturbances and the predictions. Note that it is natural for the disturbances to vary smoothly in applications such as tracking problems where the disturbances correspond to the trajectory and in such situations one would expect the predictions to also vary smoothly. For example, machine learning algorithms are often regularized to provide smooth predictions.

Given these smoothness bounds, we have that

$$\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}}) \leq \sum_{s=0}^{T-1} 2\rho(s).$$

Note that, as long as $\left(\sum_{s=0}^{T-1} \rho(s)\right)^2 = o(T)$, the regret bound is sub-linear in T. To understand how this bound may look in particular applications, suppose we have $\rho(s) = O(1/s)$. In this case, regret is poly-logarithmic, i.e., $\mathsf{Regret} = O((\log T)^2)$. If $\rho(s)$ is exponential the regret is even smaller, i.e., if $\rho(s) = O(r^s)$ for some $0 < r < 1$, then $\mathsf{Regret} = O(1)$.

Competitive Ratio. We are now ready to present our main result, which provides an upper bound on the competitive ratio of self-tuning control (Algorithm 3). Recall that, in Lemma 2, we bound the regret $\mathsf{Regret} := \mathsf{ALG}(\lambda_0, \ldots, \lambda_{T-1}) - \mathsf{ALG}(\lambda^*)$ and, in Theorem 2.3.2, a competitive ratio bound is provided for the λ-confident control scheme, including $\mathsf{ALG}(\lambda^*)/\mathsf{OPT}$. Therefore, combining Lemma 2 and Theorem 2.3.2 leads to a novel competitive ratio bound for the self-tuning scheme (Algorithm 3). Note that compared with Theorem 2.3.2, which also provides a competitive ratio bound for λ-confident control, Theorem 2.4.1 below considers a competitive ratio bound for the self-tuning scheme in Algorithm 3 where, at each time t, a trust parameter λ_t is determined by online learning and may be time-varying.

Theorem 2.4.1. *Assume $W(T) = \Omega(T)$ and $\lambda_t \in [0, 1]$ for all $t = 0, \ldots, T-1$. Under our model assumptions, the competitive ratio of Algorithm 3 is bounded by*

$$\mathsf{CR}(\varepsilon) \leq 1 + 2\|H\| \frac{\varepsilon}{\mathsf{OPT} + C\varepsilon} + O\left(\frac{\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)^2}{\mathsf{OPT}}\right)$$

where H, C OPT and ε are defined in Theorem 2.3.2.

In contrast to the regret bound, Theorem 2.4.1 states an upper bound on the competitive ratio $\mathsf{CR}(\varepsilon)$ defined in Section 2.2, which indicates that $\mathsf{CR}(\varepsilon)$ scales as

$1 + O(\varepsilon)/(\Theta(1) + \Theta(\varepsilon))$ as a function of ε. As a comparison, the λ-confident control in Algorithm 2 has a competitive ratio upper bound that is linear in the prediction error ε (Theorem 2.3.2). This improved dependency highlights the importance of learning the trust parameter adaptively.

Our experimental results in the next section verify the implications of Theorem 2.3.2 and Theorem 2.4.1. Specifically, the simulated competitive ratio of the self-tuning control (Algorithm 3) is a non-linear envelope of the simulated competitive ratios for λ-confident control with fixed trust parameters and as a function of prediction error ε, it matches the implied competitive ratio upper bound $1 + O(\varepsilon)/(\Theta(1) + \Theta(\varepsilon))$.

Theorem 2.4.1 is proven by combining Lemma 2 with Theorem 2.3.2, which bounds the competitive ratios for fixed trust parameters.

Proof of Theorem 2.4.1. Denote by $\mathsf{ALG}(\lambda_0, \ldots, \lambda_{T-1})$ the algorithmic cost of the self-tuning control scheme. We have

$$\frac{\mathsf{ALG}(\lambda_0, \ldots, \lambda_{T-1})}{\mathsf{OPT}} \leq \frac{|\mathsf{ALG}(\lambda_0, \ldots, \lambda_{T-1}) - \mathsf{ALG}(\lambda^*)|}{\mathsf{OPT}} + \frac{\mathsf{ALG}(\lambda^*)}{\mathsf{OPT}}. \tag{2.9}$$

Using Theorem 2.3.2,

$$\begin{aligned}\frac{\mathsf{ALG}(\lambda^*)}{\mathsf{OPT}} \leq & 1 + 2\|H\| \min\left\{\min_{\lambda}\left(\frac{\lambda^2}{\mathsf{OPT}}\varepsilon + \frac{(1-\lambda)^2}{C}\right), \min_{\lambda}\left(\frac{1}{C} + \frac{\lambda^2}{\mathsf{OPT}}\overline{W}\right)\right\} \\ = & 1 + 2\|H\| \min\left\{\frac{\varepsilon}{\mathsf{OPT} + \varepsilon C}, \frac{1}{C}\right\} = 1 + 2\|H\|\frac{\varepsilon}{\mathsf{OPT} + \varepsilon C}. \end{aligned} \tag{2.10}$$

Moreover, the regret bound in Lemma 2 implies

$$\frac{|\mathsf{ALG}(\lambda_0, \ldots, \lambda_{T-1}) - \mathsf{ALG}(\lambda^*)|}{\mathsf{OPT}} = O\left(\frac{\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)^2}{\mathsf{OPT}}\right),$$

combing which with (2.10), (2.9) gives the results. □

2.5 Applications

We now illustrate our main results using numerical examples and case studies to highlight the impact of the trust parameter λ in λ-confident control and demonstrate the ability of the self-tuning control algorithm to learn the appropriate trust parameter λ. We consider three applications. The first is a robot tracking example where a robot is asked to follow locations of an unknown trajectory and the desired location is only revealed the time immediately before the robot makes a decision to modify its velocity. Predictions about the trajectory are available. However, the predictions

can be untrustworthy so that they may contain large errors. The second is an adaptive battery-buffered electric vehicle (EV) charging problem where a battery-buffered charging station adaptively supplies energy demands of arriving EVs while maintaining the state of charge of the batteries as close to a nominal level as possible. Our third application considers a non-linear control problem–the Cart-Pole problem. Our λ-confident and self-tuning control schemes use a linearized model while the algorithms are tested with the non-linear environment. We use the third application to demonstrate the practicality of our algorithms by showing that they do not only work for LQC problems, but also non-linear systems.

To illustrate the impact of randomness in prediction errors in our case studies, the three applications all use different forms of random error models. For each selected distribution of $\widehat{\mathbf{w}} - \mathbf{w}$, we repeat the experiments multiple times and report the worst case with the highest algorithmic cost; see Appendix 2.E for details.

Application 1: Robot Tracking

Problem description. The first example we consider is a two-dimensional robot tracking application [39, 41]. There is a robot controller following a fixed but unknown cloud-shaped trajectory (see Figures 2.2a and 2.2b), which is

$$y_t \coloneqq \begin{bmatrix} 2\cos(\pi t/30) + \cos(\pi t/5) \\ 2\sin(\pi t/30) + \sin(\pi t/5) \end{bmatrix}, \quad t = 0, \ldots, T-1.$$

The robot controller's location at time $t+1$, denoted by $p_{t+1} \in \mathbb{R}^2$, depends on its previous location and its velocity $v_t \in \mathbb{R}^2$ such that $p_{t+1} = p_t + 0.2v_t$ and at each time $t+1$, the controller is able to apply an adjustment u_t to modify its velocity such that $v_{t+1} = v_t + 0.2u_t$. Together, letting $x_t \coloneqq p_t - y_t$, this system can be recast in the canonical form in (2.1) as

$$\begin{bmatrix} x_{t+1} \\ v_{t+1} \end{bmatrix} = A \begin{bmatrix} x_t \\ v_t \end{bmatrix} + Bu_t + w_t, \text{ with}$$

$$A \coloneqq \begin{bmatrix} 1 & 0 & 0.2 & 0 \\ 0 & 1 & 0 & 0.2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; B \coloneqq \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}, \text{ and } w_t \coloneqq Ay_t - y_{t+1}.$$

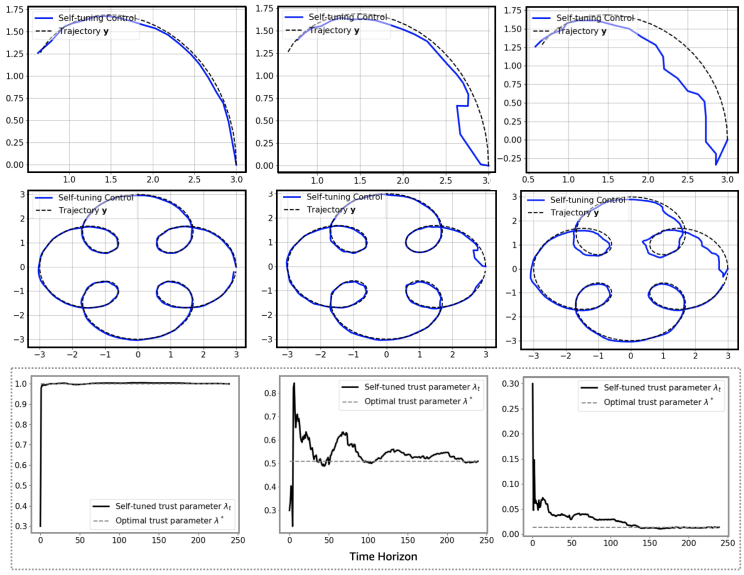
To track the trajectory, the controller sets

$$Q := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } R := \begin{bmatrix} 10^{-2} & 0 \\ 0 & 10^{-2} \end{bmatrix}.$$

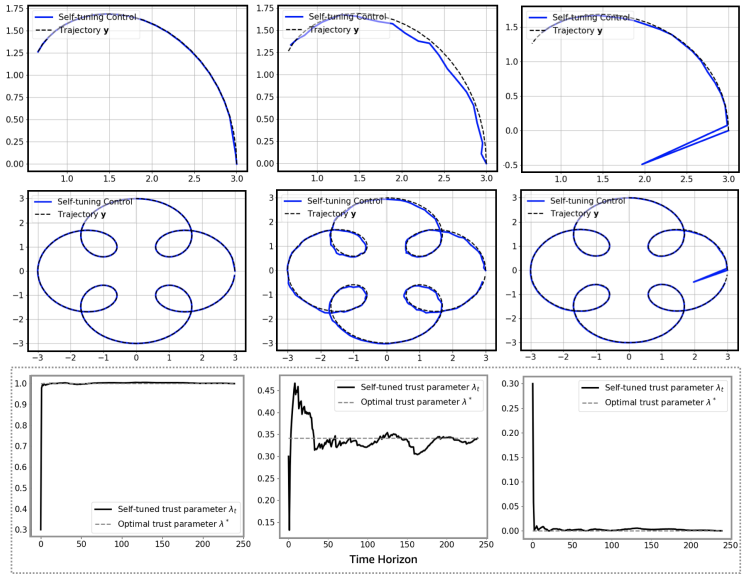
Experimental results. In our first experiment, we demonstrate the convergence of the self-tuning scheme in Algorithm 3. To mimic the worst-case error, a random prediction error $e_t = \widehat{w}_t - w_t$ at each time t is used. We then sample prediction error and implement our algorithm with several error instances and choose the one the worst competitive ratio. The details of settings can be found in Appendix 2.E. To better simulate the task of tracking a trajectory and make it easier to observe the tracking accuracy, we ignore the cost of increasing velocity by setting R as a zero matrix for Figure 2.2a and Figure 2.2b.

In Figure 2.2a, we observe that the tracking trajectory generated by the self-tuning scheme converges to the unknown trajectory $(y_1, \ldots, y_T)$, regardless of the level of prediction error. We plot the tracking trajectories every 60 time steps with a scaling parameter (defined in Appendix 2.E) $c = 10^{-2}$ (left), $c = 10^{-1}$ (mid) and $c = 1$ (right), respectively. In all cases, we observe convergence of the trust parameters. Moreover, for a wide range of prediction error levels, without knowing the prediction error level in advance, the scheme is able to automatically switch its mode and become both consistent and robust by choosing an appropriate trust parameter λ_t to accurately track the unknown trajectory. In Figure 2.2b, we observe similar behavior when the prediction error is generated from Gaussian distributions.

Next, we demonstrate the performance of self-tuning control and the impact of trust parameters. In Figure 2.3, we depict the competitive ratios of the λ-confident control algorithm described in Section 2.3 with varying trust parameters, together with the competitive ratios of the self-tuning control scheme described in Algorithm 3. The label of the x-axis is the prediction error ε (normalized by 10^3), defined in (2.7). We divide our results into two parts. The left sub-figure in Figure 2.3 considers a low-error regime where we observe that the competitive ratio of the self-tuning policy performs closely as the lower envelope formed by picking multiple trust parameters optimally offline. The right sub-figure in Figure 2.3 shows the performance of self-tuning for the case when the prediction error is high. For the high-error regime,



(a) Left: low binomial prediction error with $c = 10^{-2}$; middle: medium prediction error with $c = 10^{-1}$; right: high prediction error with $c = 1$ where c is a tuning parameter defined in Appendix 2.E.



(b) Left: low Gaussian prediction error with variance $\sigma^2 = 10^{-2}$; middle: medium Gaussian prediction error with variance $\sigma^2 = 10^{-1}$; right: high Gaussian prediction error with variance $\sigma^2 = 1$.

Figure 2.2: Tracking trajectories and trust parameters $(\lambda_0, \ldots, \lambda_{T-1})$ of the self-tuning control scheme. The x-axis and y-axis in the top 6 figures are locations of the robot. The y-axis in the bottom 3 figures denotes the value of the trust parameter.

the competitive ratio of the self-tuning control policy is close to those with the best fixed trust parameter.

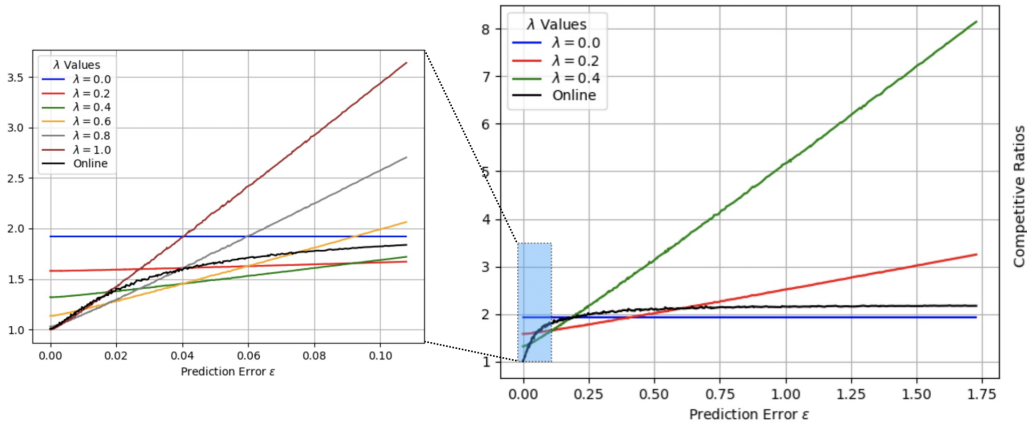


Figure 2.3: Impact of trust parameters and performance of self-tuning control for robot tracking.

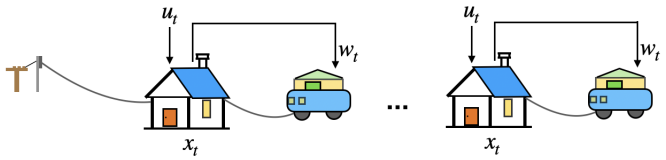


Figure 2.4: Adaptive battery-buffered EV charging modelled as a linear quadratic control problem.

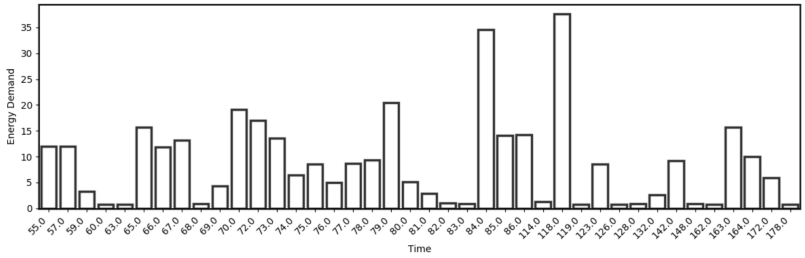


Figure 2.5: An example of the daily charging demands in ACN-Data [1] on Nov 1st, 2018.

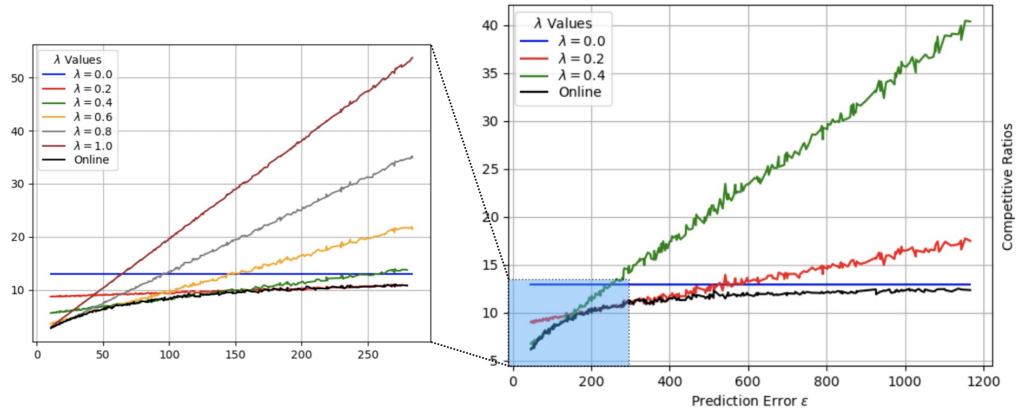
Application 2: Adaptive Battery-Buffered EV Charging

Problem description. We consider an adaptive battery-buffered Electric Vehicle (EV) charging problem. There is a charging station with N chargers, with each charger connected to a battery energy storage system. Let x_t be a vector in $\mathbb{R}_+^N$, whose entries represent the State of Charge (SoC) of the batteries at time t. The charging controller decides a charging schedule u_t in $\mathbb{R}_+^N$ where each entry in u_t is the energy to be charged to the i-th battery from external power supply at time t. The canonical form of the system can be represented by

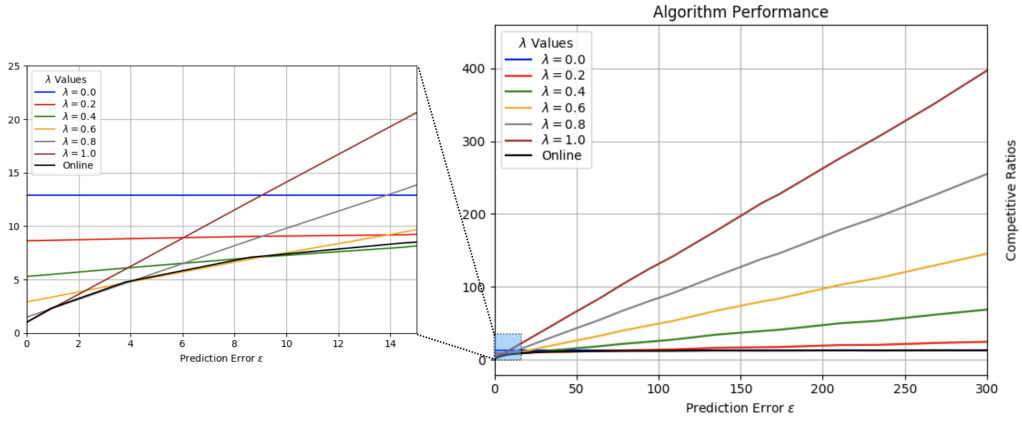
$$x_{t+1} = Ax_t + Bu_t - w_t,$$

where A is an $N \times N$ matrix denotes the *degradation* of battery charging levels and B is an $N \times N$ diagonal matrix whose diagonal entry $0 \leq B_i \leq 1$ represents the charging efficiency coefficient. In our experiments, without loss of generality, we assume A and B are identity matrices. The perturbation w_t is defined as a length-N vector, whose entry $w_t(i) = E$ when at time t an EV arrives at charger i and demands energy $E > 0$; otherwise $w_t(i) = 0$. Therefore the perturbations $(w_0, \ldots, w_{T-1})$ depend on the arrival of EVs and their energy demands. The charging controller can only make a charging decision u_t at time t before knowing w_t (as well as $w_{t+1}, \ldots, w_{T-1}$) and the EVs that arrive at time t (as well as future EV arrivals). The goal of the adaptive battery-buffered EV charging problem is to maintain the battery SoC as close to a nominal value $\overline{x}$ as possible. Therefore, the charging controller would like to minimize $\sum_{t=0}^{T-1} (x_t - \overline{x})^\top Q (x_t - \overline{x}) + u_t^\top R u_t$, equivalently, $\sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t$ where Q can be some positive-definite matrix and R encodes the costs of external power supply. In our experiments, we set Q as an identity matrix and $R = 0.1 \times Q$.

Experimental results. We show the performance of self-tuning control and the impact of trust parameters for adaptive EV charging in Figure 2.6a and 2.6b. In Figure 2.6a we consider a synthetic case when EVs with 5 kWh battery capacity arrive at a constant rate 0.2, e.g., 1 EV arrives every 5 time slots. The results are divided into two parts. In Figure 2.6b, we use daily data (ACN-Data) that contain EVs' energy demands, arrival times and departure times collected from a real-world adaptive EV charging network [1]. We select a daily charging record on on Nov 1st, 2018, depicted in Figure 2.5. The left sub-figure considers a magnified low-error regime and the right sub-figure shows the performance of self-tuning for the case when the prediction error is high. For both regimes, the competitive ratios of the self-tuning control policy perform nearly as well as the lower envelope formed by picking multiple trust parameters optimally offline. We see in both Figure 2.3 and Figure 2.6a that with fixed trust parameters the competitive ratio is linear in ε, matching what Theorem 2.3.2 indicates (in the sense of order in ε). Moreover, for the self-tuning scheme, in both Figure 2.3 and Figure 2.6a, we observe a competitive ratio $1 + O(\varepsilon)/(\Theta(1) + \Theta(\varepsilon))$, which matches the competitive ratio bound given in Theorem 2.4.1 in order sense (in ε).



(a) Experiments with Synthetic EV charging.



(b) Experiments with daily EV charging data [1].

Figure 2.6: Impact of trust parameters and performance of self-tuning control for adaptive battery-buffered EV charging with synthetic EV charging data (top) and realistic daily EV charging data [1] (bottom).

Application 3: Cart-Pole

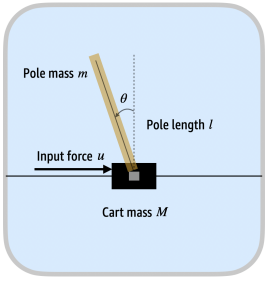


Figure 2.7: The Cart-Pole model in Application 3.

Problem description. The third set of experiments we consider is the classic Cart-Pole problem illustrated in Figure 2.7. The goal of a controller is to stabilize the pole in the upright position. This is a widely studied non-linear system. Neglecting

friction, the dynamical equations of the Cart-Pole problem are

$$\ddot{\theta} = \frac{g \sin\theta + \cos\theta \left(\frac{-u - ml\dot{\theta}^2 \sin\theta}{m+M} \right)}{l \left(\frac{4}{3} - \frac{m\cos^2\theta}{m+M} \right)}, \tag{2.11}$$

$$\ddot{y} = \frac{u + ml\left(\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta\right)}{m+M} \tag{2.12}$$

where u is the input force; θ is the angle between the pole and the vertical line; y is the location of the pole; g is the gravitational acceleration; l is the pole length; m is the pole mass; and M is the cart mass. Taking $\sin\theta \approx \theta$ and $\cos\theta \approx 1$ and ignoring higher order terms, the dynamics of the Cart-Pole problem can be linearized as

$$\frac{d}{dt}\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mlg}{\eta(m+M)} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{\eta} & 0 \end{bmatrix} \frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{(m+M)\eta + ml}{(m+M)^2\eta} \\ 0 \\ -\frac{1}{(m+M)\eta} \end{bmatrix} u_t + w_t,$$

where in the above $\eta \coloneqq l\left(\frac{4}{3} - \frac{m}{m+M}\right)$ and, in our experiments, we set the cart mass $M = 10.0kg$, pole mass $m = 1.0kg$, pole length $l = 10.0m$ and gravitational acceleration $g = 9.8m/s^2$. We set $Q = I$ and $R = 10^{-3}$ and each w_t is a fixed external force defined as

$$60 \times \left[0, \frac{(m+M)\eta + ml}{(m+M)^2\eta}, 0, -\frac{1}{(m+M)\eta}\right]^\top.$$

Experimental results. We show the performance of the self-tuning control (Algorithm 3) and the impact of trust parameters for the Cart-Pole problem in Figure 2.8, together with the λ-confident control scheme in Algorithm 2 for several fixed trust parameters λ. The algorithms are tested using the true non-linear dynamical equations in (2.11)-(2.12).

In Figure 2.8, we change the variance σ^2 of the prediction noise $e_t = \widehat{w}_t - w_t$ at each time t and plot the average episodic rewards in the OpenAI Gym environment [55]. Different from the worst-case settings in the previous two applications, we run episodes multiples times and show plot the mean rewards. The height of the shadow area in Figure 2.8 represents the standard deviation of the rewards. The detailed hyper-parameters are given in Section 2.E. Our results show that, despite the fact that the problem is *non-linear*, the self-tuning control algorithm using a linearized model is still able to automatically adjust the trust parameter λ_t and achieves both

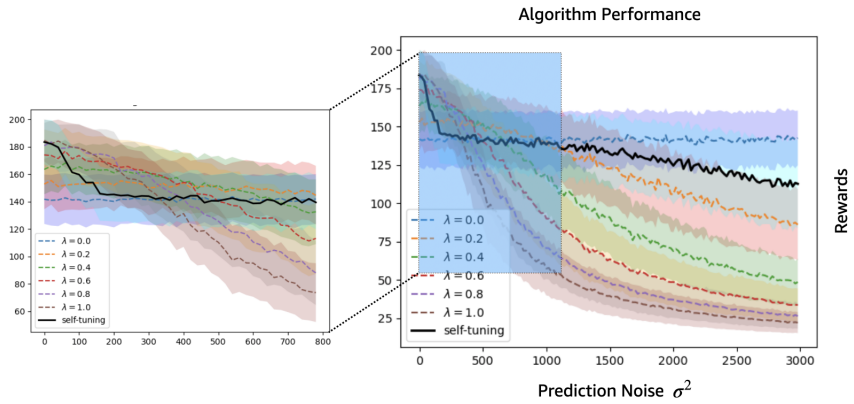


Figure 2.8: Impact of trust parameters and performance of self-tuning control for the Cart-Pole problem.

consistency and robustness, regardless of the prediction error. In particular, it is close to the best algorithms for small prediction error while also staying among the best when prediction error is large.

APPENDIX

2.A Useful Lemmas

Before proceeding to the proofs of our main results, we present some useful lemmas. We first present a lemma below from [41] that characterizes the difference between the optimal and the algorithmic costs.

Lemma 3 (Lemma 10 in [41])**.** *For any $\psi_t \in \mathbb{R}^n$, if at each time $t = 0, \dots, T-1$,*

$$u_t = -(R + B^\top P B)^{-1} B^\top \left(P A x_t + \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P w_\tau - \psi_t \right),$$

then the gap between the optimal cost OPT *and the algorithm cost* ALG *induced by selecting control actions $(u_1, \dots, u_T)$ equals to*

$$\mathsf{ALG} - \mathsf{OPT} = \sum_{t=0}^{T-1} \psi_t^\top H \psi_t$$

where $H \coloneqq B(R + B^\top P B)^{-1} B^\top$ and $F \coloneqq A - HPA$.

The next lemma describes the form of the optimal trust parameter.

Lemma 4. *The optimal trust parameter λ^* that minimizes* $\mathsf{ALG}(\lambda) - \mathsf{OPT}$ *is $\lambda^* = \lambda_T$.*

Proof of Lemma 4. The optimal trust parameter λ^* is

$$\lambda^* := \min_{\lambda} \sum_{s=0}^{T-1} \left[\left(\sum_{\tau=s}^{t-1} \left(F^\top\right)^{\tau-s} P(w_\tau - \lambda \widehat{w}_\tau) \right)^\top H \left(\sum_{\tau=s}^{T-1} \left(F^\top\right)^{\tau-s} P(w_\tau - \lambda \widehat{w}_\tau) \right) \right], \tag{2.13}$$

implying that $\lambda^* = \lambda_T$. □

Next, we note that the static regret depends on the convergence of λ_t.

Lemma 5. *The static regret satisfies*

$$\mathsf{Regret} \leq \|H\| \sum_{t=0}^{T-1} \left\| |\lambda_t - \lambda_T| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \widehat{w}_\tau \right\|^2 .$$

Proof of Lemma 5. Let $\mathsf{ALG}((\lambda_0, \ldots, \lambda_{T-1}))$ and $\mathsf{ALG}(\lambda_T)$ denote the corresponding algorithm costs for using trust parameters $(\lambda_0, \ldots, \lambda_{T-1})$ and a fixed optimal trust parameter λ_T in hindsight correspondingly. It follows that

$$\mathsf{ALG}((\lambda_0, \ldots, \lambda_{T-1})) - \mathsf{ALG}(\lambda_T) \leq \|H\| \sum_{t=0}^{T-1} \left\| |\lambda_t - \lambda_T| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P\widehat{w}_\tau \right\|^2. \tag{2.14}$$

□

Lemma 6. *Suppose two real sequences $(V_1, \ldots, V_T)$ and $(W_1, \ldots, W_T)$ with $W_t > 0$ for all $1 \leq t \leq T$, converge to V_T and $W_T > 0$ such that for any integer $1 \leq t \leq T$, $|V_t - V_T| \leq C_1/t$ and $|W_t - W_T| \leq C_2/t$ for some constants $C_1, C_2 > 0$. Then the sequence $\left(\frac{V_1}{W_1}, \ldots, \frac{V_T}{W_T}\right)$ converges to $\frac{V_T}{W_T}$ such that for any $1 \leq t \leq T$,*

$$\left|\frac{V_t}{W_t} - \frac{V_T}{W_T}\right| \leq \frac{1}{t}\left(\frac{C_1\alpha_t + C_2}{|W_T|}\right)$$

where $\alpha_t \coloneqq \max\{V_t/W_t\}$.

Proof of Lemma 6. Based on the assumption, for any $1 \leq t \leq T$, we have that

$$\begin{aligned}\left|\frac{V_t}{W_t} - \frac{V_T}{W_T}\right| = \left|\frac{V_tW_T - V_TW_t}{W_tW_T}\right| &= \left|\frac{V_tW_T - V_tW_t + V_tW_t - V_TW_t}{W_tW_T}\right| \\ &\leq \left|\frac{V_t\left(W_T - W_t\right)}{W_tW_T}\right| + \left|\frac{W_t\left(V_T - V_t\right)}{W_tW_T}\right| \\ &\leq \frac{1}{t}\left(\frac{C_1|V_t|}{|W_tW_T|} + \frac{C_2}{|W_T|}\right).\end{aligned}$$

Since $W_t \neq 0$ for all $1 \leq t \leq T$ and $W_T \neq 0$, the lemma follows. □

Lemma 7. *Suppose a sequence $(A_0, \ldots, A_{T-1})$ satisfies that for any integer $0 \leq s \leq T-1$, $|A_s - A_T| \leq \rho(s)$. Then, for any $0 \leq s \leq T$, $\left|\frac{1}{t}\left(\sum_{s=0}^{t} A_s\right) - A_T\right| \leq \frac{1}{t}\sum_{s=0}^{T-1}\rho(s)$.*

Proof of Lemma 7. Based on the assumption,

$$\left|\frac{1}{t}\sum_{s=0}^{t} A_s - A_T\right| = \frac{1}{t}\left|\sum_{s=0}^{t}\left(A_s - A_T\right)\right| \leq \frac{1}{t}\sum_{s=0}^{t}|A_s - A_t| \leq \frac{1}{t}\sum_{s=0}^{T-1}\rho(s).$$

□

2.B Competitive Analysis

Throughout, for notational convenience, we write

$$W(t) \coloneqq \sum_{s=0}^{t} \eta(\widehat{w}; s, t)^\top H \eta(\widehat{w}; s, t), \quad \text{and } V(t) \coloneqq \sum_{s=0}^{t} \eta(w; s, t)^\top H \eta(\widehat{w}; s, t)$$

where

$$\eta(w; s, t) \coloneqq \sum_{\tau=s}^{t} \left(F^\top\right)^{\tau-s} P w_\tau, \quad \text{and } \eta(\widehat{w}; s, t) \coloneqq \sum_{\tau=s}^{t} \left(F^\top\right)^{\tau-s} P \widehat{w}_\tau.$$

We first prove the following theorem.

Theorem 2.B.1. *With a fixed trust parameter $\lambda > 0$, the λ-confident control in Algorithm 2 has a worst-case competitive ratio of at most*

$$\mathsf{CR}(\varepsilon) \leq 1 + 2\|H\| \min\left\{\left(\frac{\lambda^2}{\mathsf{OPT}}\varepsilon + \frac{(1-\lambda)^2}{C}\right), \left(\frac{1}{C} + \frac{\lambda^2}{\mathsf{OPT}}\overline{W}\right)\right\}$$

where $H \coloneqq B(R + B^\top P B)^{-1} B^\top$, OPT denotes the optimal cost, $C > 0$ is a constant that depends on A, B, Q, R and

$$\varepsilon\left(F, P, e_0, \ldots, e_{T-1}\right) \coloneqq \sum_{t=0}^{T-1} \left\| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \left(w_\tau - \widehat{w}_\tau\right) \right\|^2,$$

$$\overline{W}\left(F, P, \widehat{w}_0, \ldots, \widehat{w}_{T-1}\right) \coloneqq \sum_{t=0}^{T-1} \left\| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \widehat{w}_\tau \right\|^2.$$

Proof of Theorem 2.3.2

Denote by ALG the cost induced by taking actions $(u_0, \ldots, u_{T-1})$ in Algorithm 2 and OPT the optimal total cost. Note that we assume $\mathsf{OPT} > 0$. Lemma 3 implies that

$$\mathsf{ALG} - \mathsf{OPT} = \sum_{t=0}^{T-1} \left(\sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \left(w_t - \lambda \widehat{w}_\tau\right) \right)^\top H \left(\sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \left(w_t - \lambda \widehat{w}_\tau\right) \right). \tag{2.15}$$

Therefore, with a sequence of actions $(u_1, \ldots, u_T)$ generated by the λ-confident control scheme, (2.15) leads to

$$\begin{aligned}
\mathsf{ALG} - \mathsf{OPT} \leq & \|H\| \sum_{t=0}^{T-1} \left\| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P w_\tau - \lambda \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \widehat{w}_\tau \right\|^2 \\
= & \|H\| \sum_{t=0}^{T-1} \left\| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P w_\tau - \lambda \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \left(w_\tau + e_\tau\right) \right\|^2 \\
= & \|H\| \sum_{t=0}^{T-1} \left\| (1-\lambda) \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P w_\tau - \lambda \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P e_\tau \right\|^2 \\
\leq & 2\|H\| \left((1-\lambda)^2 \sum_{t=0}^{T-1} \left\| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P w_\tau \right\|^2 + \lambda^2 \sum_{t=0}^{T-1} \left\| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P e_\tau \right\|^2 \right)
\end{aligned}$$

where $e_t \coloneqq \widehat{w}_t - w_T$ for all $t = 0, \ldots, T-1$. Moreover, denoting by x_t^* and u_t^* the offline optimal state and action at time t, the optimal cost satisfies

$$\begin{aligned}
\mathsf{OPT} = & \sum_{t=0}^{T-1} (x_t^*)^\top Q x_t^* + (u_t^*)^\top R u_t^* + (x_T^*)^\top P x_T^* \\
\geq & \sum_{t=0}^{T-1} \lambda_{\min}(Q) \left\|x_t^*\right\|^2 + \lambda_{\min}(R) \|u_t^*\|^2 + \lambda_{\min}(P) \|x_T^*\|^2 \qquad (2.16) \\
\geq & 2D_0 \sum_{t=0}^{T-1} \left(\|A x_t^*\|^2 + \|B u_t^*\|^2 \right) + \frac{1}{2} \sum_{t=0}^{T-1} \lambda_{\min}(Q) \|x_t^*\|^2 + \lambda_{\min}(P) \|x_T^*\|^2 \\
\geq & D_0 \sum_{t=0}^{T-1} \left\|A x_t^* + B u_t^*\right\|^2 + \frac{1}{2} \sum_{t=0}^{T-1} \lambda_{\min}(Q) \|x_t^*\|^2 + \lambda_{\min}(P) \|x_T^*\|^2 \\
= & D_0 \sum_{t=0}^{T-1} \left\|x_{t+1}^* - w_t\right\|^2 + \frac{1}{2} \sum_{t=0}^{T-1} \lambda_{\min}(Q) \|x_t^*\|^2 + \lambda_{\min}(P) \|x_T^*\|^2 \\
\geq & \frac{D_0}{2} \sum_{t=0}^{T-1} \|w_t\|^2 + \left(\frac{\lambda_{\min}(Q)}{2} - D_0 \right) \sum_{t=0}^{T-1} \|x_t^*\|^2 + \left(\lambda_{\min}(P) - C\right) \|x_T^*\|^2 \qquad (2.17)
\end{aligned}$$

for some constant $0 < D_0 < \min\{\lambda_{\min}(P), \lambda_{\min}(Q)/2\}$ that depends on Q, R and K where in (2.16), $\lambda_{\min}(Q)$, $\lambda_{\min}(R)$ and $\lambda_{\min}(P)$ are the smallest eigenvalues of positive definite matrices Q, R and P, respectively. Let $\psi_t \coloneqq \sum_{\tau=0}^{T-t-1} \left(F^\top\right)^\tau P w_{t+\tau}$. Note that $F = A - BK$ and we define $\rho \coloneqq \frac{1+\rho(F)}{2} < 1$ where $\rho(F)$ denotes the spectral radius of F. From Gelfand's formula, there exists a constant $D_1 \geq 0$ such

that $\|F^t\| \le D_1 \rho^t$ for all $t \ge 0$. Therefore,

$$\begin{aligned}
\sum_{t=0}^{T-1} \|\psi_t\|^2 &= \sum_{t=0}^{T-1} \left\| \sum_{\tau=0}^{T-t-1} \left(F^\top\right)^\tau P w_{t+\tau} \right\|^2 \\
&\le D_1^2 \|P\|^2 \sum_{t=0}^{T-1} \left(\sum_{\tau=0}^{T-t-1} \rho^\tau \|w_{t+\tau}\| \right)^2 \\
&= D_1^2 \|P\|^2 \sum_{t=0}^{T-1} \sum_{\tau=0}^{T-t-1} \sum_{\ell=0}^{T-t-1} \rho^\tau \rho^\ell \|w_{t+\tau}\| \|w_{t+\ell}\| \\
&\le \frac{D_1^2}{2} \|P\|^2 \sum_{t=0}^{T-1} \sum_{\tau=0}^{T-t-1} \sum_{\ell=0}^{T-t-1} \rho^\tau \rho^\ell \left(\|w_{t+\tau}\|^2 + \|w_{t+\ell}\|^2 \right). \qquad (2.18)
\end{aligned}$$

Continuing from (2.18),

$$\begin{aligned}
\sum_{t=0}^{T-1} \|\psi_t\|^2 \le& \frac{D_1^2}{2} \|P\|^2 \left(\sum_{\ell=0}^{T-t-1} \rho^\ell \right) \sum_{t=0}^{T-1} \sum_{\tau=0}^{T-t-1} \rho^\tau \|w_{t+\tau}\|^2 \\
&+ \frac{D_1^2}{2} \|P\|^2 \left(\sum_{\tau=0}^{T-t-1} \rho^\tau \right) \sum_{t=0}^{T-1} \sum_{\ell=0}^{T-t-1} \rho^\ell \|w_{t+\ell}\|^2 \qquad (2.19) \\
\le& \frac{D_1^2}{1-\rho} \|P\|^2 \sum_{t=0}^{T-1} \sum_{\tau=0}^{T-t-1} \rho^\tau \|w_{t+\tau}\|^2 \\
\le& \frac{D_1^2}{1-\rho} \|P\|^2 \sum_{t=0}^{T-1} \sum_{\tau=0}^{T-1} \rho^\tau \|w_{(t+\tau) \mod T}\|^2 \\
=& \frac{D_1^2}{1-\rho} \|P\|^2 \left(\sum_{\tau=0}^{T-1} \rho^\tau \right) \left(\sum_{t=0}^{T-1} \|w_t\|^2 \right) \\
\le& \frac{D_1^2}{(1-\rho)^2} \|P\|^2 \sum_{t=0}^{T-1} \|w_t\|^2. \qquad (2.20)
\end{aligned}$$

Putting (2.20) into (2.17), we obtain

$$\mathsf{OPT} \ge \frac{D_0 (1-\rho)^2}{D_1^2 \|P\|^2} \sum_{t=0}^{T-1} \|\psi_t\|^2,$$

which implies that

$$\frac{\mathsf{ALG} - \mathsf{OPT}}{\mathsf{OPT}} \le 2\|H\| \left(\frac{\lambda^2}{\mathsf{OPT}} \varepsilon + \frac{(1-\lambda)^2}{C} \right)$$

where $C := \frac{D_0 (1-\rho)^2}{D_1^2 \|P\|^2}$ and

$$\varepsilon := \sum_{t=0}^{T-1} \left\| \sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \left(w_\tau - \widehat{w}_\tau \right) \right\|^2.$$

To obtain the second bound, noting that

$$\begin{aligned}\mathsf{ALG}-\mathsf{OPT} \leq & \|H\| \sum_{t=0}^{T-1}\left\|\sum_{\tau=t}^{T-1}\left(F^{\top}\right)^{\tau-t} P w_{\tau}-\lambda \sum_{\tau=t}^{T-1}\left(F^{\top}\right)^{\tau-t} P \widehat{w}_{\tau}\right\|^{2} \\ \leq & 2\|H\| \sum_{t=0}^{T-1}\left(\left\|\sum_{\tau=t}^{T-1}\left(F^{\top}\right)^{\tau-t} P w_{\tau}\right\|^{2}+\lambda^{2}\left\|\sum_{\tau=t}^{T-1}\left(F^{\top}\right)^{\tau-t} P \widehat{w}_{\tau}\right\|^{2}\right).\end{aligned}$$

Noting that $W \coloneqq \sum_{t=0}^{T-1}\left\|\sum_{\tau=t}^{T-1}\left(F^{\top}\right)^{\tau-t} P \widehat{w}_{\tau}\right\|^{2}$, therefore,

$$\frac{\mathsf{ALG}-\mathsf{OPT}}{\mathsf{OPT}} \leq 2\|H\|\left(\frac{1}{C}+\frac{\lambda^{2}}{\mathsf{OPT}} W\right)$$

for some constant $C>0$ that depends on A, B, Q and R.

2.C Regret Analysis of Self-Tuning Control

Throughout, for notational convenience, we write

$$W(t) \coloneqq \sum_{s=0}^{t} \eta(\widehat{w} ; s, t)^{\top} H \eta(\widehat{w} ; s, t), \quad \text{and } V(t) \coloneqq \sum_{s=0}^{t} \eta(w ; s, t)^{\top} H \eta(\widehat{w} ; s, t)$$

where

$$\eta(w ; s, t) \coloneqq \sum_{\tau=s}^{t}\left(F^{\top}\right)^{\tau-s} P w_{\tau}, \quad \text{and } \eta(\widehat{w} ; s, t) \coloneqq \sum_{\tau=s}^{t}\left(F^{\top}\right)^{\tau-s} P \widehat{w}_{\tau}.$$

Proof of Lemma 1

In this section, we show the proof of Lemma 2 and Lemma 1. We begin with rewriting $\lambda_t - \lambda_T$ as below.

$$\lambda_t-\lambda_T=\frac{V(t-1)}{W(t-1)}-\frac{V(T-1)}{W(T-1)}=\frac{\frac{V(t-1)}{t-1}}{\frac{W(t-1)}{(t-1)}}-\frac{\frac{V(T-1)}{T-1}}{\frac{W(T-1)}{T-1}}. \tag{2.21}$$

Applying Lemma 6, it suffices to prove that for any $1 \leq t \leq T$, $\left|\frac{1}{T} V(T)-\frac{1}{t} V(t)\right| \leq \frac{C_1}{t}$ and $\left|\frac{1}{T} W(T)-\frac{1}{t} W(t)\right| \leq \frac{C_2}{t}$ for some constants $C_1>0$ and $C_2>0$. In the sequel, we show the bound on $\left|\frac{1}{T} V(T)-\frac{1}{t} V(t)\right|$ and the bound on $\left|\frac{1}{T} W(T)-\frac{1}{t} W(t)\right|$ follows

using the same argument. Continuing from (2.21),

$$\left|\frac{1}{T}V(T)-\frac{1}{t}V(t)\right| \le \underbrace{\left|\frac{1}{T}\sum_{s=0}^{T}\eta(w;s,T)^\top H\eta(\widehat{w};s,T)-\frac{1}{t}\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,T)\right|}_{=:(a)} + \underbrace{\left|\frac{1}{t}\sum_{s=0}^{t}\eta(w;s,t)^\top H\eta(\widehat{w};s,t)-\frac{1}{t}\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,T)\right|}_{=:(b)}. \tag{2.22}$$

In the following, we deal with the terms (a) and (b) separately.

Upper bound on (a). To bound the term (a) in (2.22), we notice that (a) can be regarded as a difference between two algebraic means. Rewriting the first mean in (a), we get

$$\begin{aligned}\sum_{s=0}^{T}\eta(w;s,T)^\top H\eta(\widehat{w};s,T) &= \sum_{s=0}^{T}\left(\sum_{\tau=s}^{T}\left(F^\top\right)^{\tau-s}Pw_\tau\right)^\top H\left(\sum_{\tau=s}^{T}\left(F^\top\right)^{\tau-s}P\widehat{w}_\tau\right)\\ &= \sum_{s=0}^{T}\left(\sum_{\tau=0}^{T-s}\left(F^\top\right)^{\tau}Pw_{\tau+s}\right)^\top H\left(\sum_{\tau=0}^{T-s}\left(F^\top\right)^{\tau}P\widehat{w}_{\tau+s}\right)\\ &= \sum_{s=0}^{T}\overline{\eta}(w;s,T)^\top H\overline{\eta}(\widehat{w};s,T)\end{aligned}$$

where for notational convenience, for $s \le T$ we have defined two series

$$\overline{\eta}(\widehat{w};s,T) \coloneqq \sum_{\tau=0}^{s}\left(F^\top\right)^{\tau}P\widehat{w}_{\tau+T-s}, \quad \text{and } \overline{\eta}(w;s,T) \coloneqq \sum_{\tau=0}^{s}\left(F^\top\right)^{\tau}Pw_{\tau+T-s}.$$

We state a lemma below, which states that the sequence $(\overline{\eta}(\widehat{w};0,T),\dots,\overline{\eta}(\widehat{w};T,T))$ satisfies the assumption in Lemma 7.

Lemma 8. *Given an integer s with $0 \le s \le T$, we have*

$$\begin{aligned}&\left|\overline{\eta}(w;T,T)^\top H\overline{\eta}(\widehat{w};T,T)-\overline{\eta}(w;s,T)^\top H\overline{\eta}(\widehat{w};s,T)\right|\\ \le& 2\|H\|\left(\frac{C\|P\|}{1-\rho}\right)^2\left(2\rho^{s+1}\overline{w}\widehat{w}+\max_\tau\|\widehat{w}_\tau-\widehat{w}_{\tau+T-s}\|+\max_\tau\|w_\tau-w_{\tau+T-s}\|\right).\end{aligned}$$

Proof of Lemma 8. With $s \leq T$, according to the definitions of $\overline{\eta}(w; s, T)$, $\overline{\eta}(w; T, T)$, $\overline{\eta}(\widehat{w}; s, T)$ and $\overline{\eta}(\widehat{w}; T, T)$, we obtain

$$\overline{\eta}(w; T, T) = \overline{\eta}(w; s, T) + \sum_{\tau=s+1}^{T} \left(F^\top\right)^\tau P w_\tau + \sum_{\tau=0}^{s} \left(F^\top\right)^\tau P \left(w_\tau - w_{\tau+T-s}\right),$$
$$\overline{\eta}(\widehat{w}; T, T) = \overline{\eta}(\widehat{w}; s, T) + \sum_{\tau=s+1}^{T} \left(F^\top\right)^\tau P \widehat{w}_\tau + \sum_{\tau=0}^{s} \left(F^\top\right)^\tau P \left(\widehat{w}_\tau - \widehat{w}_{\tau+T-s}\right),$$

implying that

$$\begin{aligned}&\overline{\eta}(w; T, T)^\top H \overline{\eta}(\widehat{w}; T, T) - \overline{\eta}(w; s, T)^\top H \overline{\eta}(\widehat{w}; s, T)\\ =&\overline{\eta}(w; s, T)^\top H \xi_2 + \xi_1^\top H \overline{\eta}(\widehat{w}; s, T) + \xi_1^\top H \xi_2 \end{aligned} \tag{2.23}$$

where

$$\xi_1 \coloneqq \sum_{\tau=s+1}^{T} \left(F^\top\right)^\tau P w_\tau + \sum_{\tau=0}^{s} \left(F^\top\right)^\tau P \left(w_\tau - w_{\tau+T-s}\right),$$
$$\xi_2 \coloneqq \sum_{\tau=s+1}^{T} \left(F^\top\right)^\tau P \widehat{w}_\tau + \sum_{\tau=0}^{s} \left(F^\top\right)^\tau P \left(\widehat{w}_\tau - \widehat{w}_{\tau+T-s}\right).$$

By our model assumption, $\|w_t\| \leq \omega$ and $\|\widehat{w}_t\| \leq \overline{w}$ for all $t = 0, \ldots, T-1$. Then, there exists some $e > 0$ such that the prediction error $e_t = \widehat{w}_t - w_t$ satisfies $e_t \leq e$ for all $t = 0, \ldots, T-1$. Note that $F = A - BK$ and we define $\rho \coloneqq \frac{1+\rho(F)}{2} < 1$ where $\rho(F)$ denotes the spectral radius of F. From Gelfand's formula, there exists a constant $C \geq 0$ such that $\|F^t\| \leq C\rho^t$ for all $t \geq 0$. The following holds for $\overline{\eta}(\widehat{w}; s, T)$ and $\overline{\eta}(w; s, T)$:

$$\|\overline{\eta}(\widehat{w}; s, T)\| \leq \sum_{\tau=0}^{s} \|F^\tau\| \, \|P\| \overline{w} \leq C \frac{1-\rho^{s+1}}{1-\rho} \|P\| \overline{w} \leq \frac{C}{1-\rho} \|P\| \overline{w}, \tag{2.24}$$
$$\|\overline{\eta}(w; s, T)\| \leq \sum_{\tau=0}^{s} \|F^\tau\| \, \|P\| \overline{w} = C \frac{1-\rho^{s+1}}{1-\rho} \|P\| \overline{w} \leq \frac{C}{1-\rho} \|P\| \overline{w}. \tag{2.25}$$

Moreover,

$$\|\xi_1\| \leq \sum_{\tau=s+1}^{T} \|F^\tau\| \, \|P\| \overline{w} + \sum_{\tau=0}^{s} \|F^\tau\| \, \|P\| \, \|w_\tau - w_{\tau+T-s}\| \tag{2.26}$$
$$\leq \frac{C\|P\|}{1-\rho} \left(\overline{w}\rho^{s+1} + \max_\tau \|w_\tau - w_{\tau+T-s}\|\right) \tag{2.27}$$
$$\|\xi_2\| \leq \sum_{\tau=s+1}^{T} \|F^\tau\| \, \|P\| \widehat{w} + \sum_{\tau=0}^{s} \|F^\tau\| \, \|P\| \, \|\widehat{w}_\tau - \widehat{w}_{\tau+T-s}\| \tag{2.28}$$
$$\leq \frac{C\|P\|}{1-\rho} \left(\widehat{w}\rho^{s+1} + \max_\tau \|\widehat{w}_\tau - \widehat{w}_{\tau+T-s}\|\right). \tag{2.29}$$

Combining (2.24)-(2.29) with (2.23),

$$\begin{aligned}&\left|\overline{\eta}(w;T,T)^\top H\overline{\eta}(\widehat{w};T,T)-\overline{\eta}(w;s,T)^\top H\overline{\eta}(\widehat{w};s,T)\right|\\ \leq& 2\|H\|\left(\frac{C\|P\|}{1-\rho}\right)^2\left(2\rho^{s+1}\overline{w}\widehat{w}+\max_\tau\|\widehat{w}_\tau-\widehat{w}_{\tau+T-s}\|+\max_\tau\|w_\tau-w_{\tau+T-s}\|\right).\end{aligned}$$

□

Therefore, applying Lemma 7, we conclude that

$$\begin{aligned}(a)\coloneqq&\left|\frac{1}{T}\sum_{s=0}^{T}\eta(w;s,T)^\top H\eta(\widehat{w};s,T)-\frac{1}{t}\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,T)\right|\\ \leq&\frac{4}{t}\|H\|\rho\left(\frac{C\|P\|}{(1-\rho)^{3/2}}\right)^2\overline{w}\widehat{w}+\frac{2}{t}\|H\|\left(\frac{C\|P\|}{1-\rho}\right)^2\left(\mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})+\mu_{\mathsf{VAR}}(\mathbf{w})\right)\end{aligned}\tag{2.30}$$

where $\mu_{\mathsf{VAR}}(\mathbf{x})\coloneqq\sum_{s=0}^{T}\max_\tau\|x_\tau-x_{\tau+T-s}\|$ denotes the self-variation of a sequence $\mathbf{x}$.

Upper bound on (b). Next, we provide a bound on (b) in (2.22). For (b), we have

$$\begin{aligned}(b)\coloneqq&\frac{1}{t}\left|\sum_{s=0}^{t}\eta(w;s,t)^\top H\eta(\widehat{w};s,t)-\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,T)\right|\\ \leq&\frac{1}{t}\left|\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,t)-\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,T)\right|\\ &+\frac{1}{t}\left|\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,t)-\sum_{s=0}^{t}\eta(w;s,t)^\top H\eta(\widehat{w};s,t)\right|.\end{aligned}\tag{2.31}$$

Noting that $\eta(\widehat{w};s,T)-\eta(\widehat{w};s,t)=\sum_{\tau=t+1}^{T}\left(F^\top\right)^{\tau-s}P\widehat{w}_\tau$, we obtain

$$\begin{aligned}&\left|\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,t)-\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,T)\right|\\ =&\left|\sum_{s=0}^{t}\eta(w;s,T)^\top H\left(\eta(\widehat{w};s,t)-\eta(\widehat{w};s,T)\right)\right|\\ =&\left|\sum_{s=0}^{t}\eta(w;s,T)^\top H\left(\sum_{\tau=t+1}^{T}\left(F^\top\right)^{\tau-s}P\widehat{w}_\tau\right)\right|\end{aligned}\tag{2.32}$$

and similarly,

$$\begin{aligned}
&\left|\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,t)-\sum_{s=0}^{t}\eta(w;s,t)^\top H\eta(\widehat{w};s,t)\right| \\
=&\left|\sum_{s=0}^{t}\left(\eta(w;s,T)-\eta(w;s,t)\right)^\top H\eta(\widehat{w};s,t)\right| \\
=&\left|\sum_{s=0}^{t}\left(\sum_{\tau=t+1}^{T}\left(F^\top\right)^{\tau-s}Pw_\tau\right)^\top H\eta(\widehat{w};s,t)\right|. \qquad (2.33)
\end{aligned}$$

By our assumption, $\|w_t\|\le\overline{w}$ and $\|\widehat{w}_t\|\le\widehat{w}$ for all $t=0,\dots,T-1$. Therefore, for any $s\le t$:

$$\left\|\sum_{\tau=t+1}^{T}\left(F^\top\right)^{\tau-s}P\widehat{w}_\tau\right\|\le\frac{C\rho^{t-s+1}\|P\|\widehat{w}}{1-\rho} \qquad (2.34)$$

and

$$\|\eta(w;s,T)\|=\left\|\sum_{\tau=s}^{T}\left(F^\top\right)^{\tau-s}Pw_\tau\right\|\le\frac{C\|P\|\overline{w}}{1-\rho}. \qquad (2.35)$$

Plugging (2.34) and (2.35) into (2.32),

$$\begin{aligned}
&\left|\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,t)-\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,T)\right| \\
\le&2C\|H\|\left(\frac{\|P\|}{1-\rho}\right)^2\overline{w}\widehat{w}\sum_{s=0}^{t}\left\|F^{t-s+1}\right\| \\
\le&2\|H\|\left(\frac{C\|P\|}{1-\rho}\right)^2\frac{\rho\left(1-\rho^t\right)}{1-\rho}\overline{w}\widehat{w} \\
\le&2\|H\|\left(\frac{C\|P\|}{(1-\rho)^{3/2}}\right)^2\rho\overline{w}\widehat{w}. \qquad (2.36)
\end{aligned}$$

Using the same argument, the following bound holds for (2.33):

$$\left|\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,t)-\sum_{s=0}^{t}\eta(w;s,t)^\top H\eta(\widehat{w};s,t)\right| \qquad (2.37)$$

$$\le2\|H\|\left(\frac{C\|P\|}{(1-\rho)^{3/2}}\right)^2\rho\overline{w}\widehat{w}. \qquad (2.38)$$

Combining (2.36) and (2.37) and using (2.31),

$$\begin{aligned}
(b)\coloneqq&\frac{1}{t}\left|\sum_{s=0}^{t}\eta(w;s,t)^\top H\eta(\widehat{w};s,t)-\sum_{s=0}^{t}\eta(w;s,T)^\top H\eta(\widehat{w};s,T)\right| \\
\le&\frac{4}{t}\|H\|\left(\frac{C\|P\|}{(1-\rho)^{3/2}}\right)^2\rho\overline{w}\widehat{w}. \qquad (2.39)
\end{aligned}$$

Finally, together, (2.30) and (2.39) imply the following:

$$\left|\frac{1}{T}V(T)-\frac{1}{t}V(t)\right| \leq \frac{8}{t}\|H\|\left(\frac{C\|P\|}{(1-\rho)^{3/2}}\right)^2 \rho\overline{w}\widehat{w} + \frac{2}{t}\|H\|\left(\frac{C\|P\|}{1-\rho}\right)^2 \left(\mu_{\mathsf{VAR}}(\mathbf{w})+\mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right). \tag{2.40}$$

The same argument also guarantees that

$$\left|\frac{1}{T}W(T)-\frac{1}{t}W(t)\right| \leq \frac{8}{t}\|H\|\left(\frac{C\|P\|}{(1-\rho)^{3/2}}\right)^2 \rho\widehat{w}^2 + \frac{4}{t}\|H\|\left(\frac{C\|P\|}{1-\rho}\right)^2 \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}}). \tag{2.41}$$

The following lemma together with (2.40) and (2.41) justify the conditions needed to apply Lemma 6.

Lemma 9. *For any integer* $1 \leq t \leq T$*,*

$$\frac{V(t)}{t} \leq 2\|H\|\left(\frac{C\|P\|}{(1-\rho)^{3/2}}\right)^2 \overline{w}\widehat{w}$$

where $C > 0$ *is some constant satisfying* $\|F^t\| \leq C\rho^t$ *for all* $t \geq 0$*.*

Proof of Lemma 9. We have

$$\begin{aligned}
\frac{V(t)}{t} =& \frac{1}{t}\sum_{s=0}^{t}\eta(w;s,t)^\top H\eta(\widehat{w};s,t)\\
\leq& \frac{\|H\|}{t}\sum_{s=0}^{t}\left\|\sum_{\tau=0}^{t-1-s}\left(F^\top\right)^\tau Pw_{\tau+s}\right\|\left\|\sum_{\tau=0}^{t-1-s}\left(F^\top\right)^\tau P\widehat{w}_{\tau+s}\right\|\\
\leq& \frac{\|H\|}{t}\left(\frac{C\|P\|}{1-\rho}\right)^2\sum_{s=0}^{t}\left(1-\rho^{t-s}\right)\overline{w}\widehat{w}\\
=& \frac{\|H\|}{t}\left(\frac{C\|P\|}{1-\rho}\right)^2\left(t+\frac{1-\rho^{t+1}}{1-\rho}\right)\overline{w}\widehat{w}\\
\leq& 2\|H\|\left(\frac{C\|P\|}{(1-\rho)^{3/2}}\right)^2\overline{w}\widehat{w}.
\end{aligned}$$

□

First, based on our assumption, $\lambda_t = V(t)/W(t) = V_t/W_t \leq 1$. Moreover, $W(T)/T = \Omega(1)$. Therefore, using (2.40), (2.41), Lemma 6 and Lemma 9, (2.21) implies that

for any $1 < t \leq T$,

$$\begin{aligned}|\lambda_t - \lambda_T| \leq \frac{1}{t-1}\frac{\|H\|\left(\frac{C\|P\|}{1-\rho}\right)^2}{W(T)/T}\cdot\Bigg(&\frac{8\rho\widehat{w}\overline{w}}{1-\rho} + 2\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)\\ &+\frac{2\|H\|\left(\frac{C\|P\|}{(1-\rho)^{3/2}}\right)^2\overline{w}\widehat{w}}{W(T)/T}\left(\frac{8\rho\widehat{w}^2}{1-\rho} + 4\mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)\Bigg)\\ = O\left(\frac{\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})}{t}\right).\end{aligned}$$

Proof of Lemma 2

Using Lemma 8,

$$|\lambda_t - \lambda_T| \leq \frac{C}{t}\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right), \qquad \text{where } C_1 > 0 \text{ is some constant.}$$

Applying Lemma 5, and noting that

$$\left\|\sum_{\tau=t}^{T-1}\left(F^\top\right)^{\tau-t}P\widehat{w}_\tau\right\| \leq C\frac{1-\rho^{T-t}}{1-\rho}\|P\|\widehat{w},$$

(2.14) implies

$$\begin{aligned}\mathsf{Regret} \leq& C_1^2\|H\|\sum_{t=1}^{T-1}\left\|\frac{\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})}{t}\sum_{\tau=t}^{T-1}\left(F^\top\right)^{\tau-t}P\widehat{w}_\tau\right\|^2 + C_0\\ =& C_1^2\|H\|\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)^2\sum_{t=1}^{T-1}\frac{1}{t^2}\left\|\sum_{\tau=t}^{T-1}\left(F^\top\right)^{\tau-t}P\widehat{w}_\tau\right\|^2 + C_0\\ \leq& C_1^2\|H\|\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)^2\left(\frac{C\|P\|}{1-\rho}\widehat{w}\right)^2\sum_{t=2}^{T-1}\frac{1}{t^2} + C_0\\ \leq& \frac{C_1^2\pi^2}{6}\|H\|\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)^2\left(\frac{C\|P\|}{1-\rho}\widehat{w}\right)^2 + C_0 \qquad (2.42)\end{aligned}$$

where

$$C_0 \coloneqq \|H\|\,|\lambda_T - \lambda_0|\left\|\sum_{\tau=0}^{T-1}\left(F^\top\right)^{\tau}P\widehat{w}_\tau\right\|^2 \leq \|H\|\,|\lambda_T - \lambda_0|\left(\frac{C\|P\|}{1-\rho}\widehat{w}\right)^2.$$

Moreover, for any $t = 1, \ldots, T$, $|\lambda_t| \leq 1$, whence,

$$C_0 \leq 2\|H\|\left(\frac{C\|P\|}{1-\rho}\widehat{w}\right)^2.$$

Therefore, continuing from (2.42),

$$\begin{aligned}\mathsf{Regret} \leq & \|H\| \left(\frac{C\|P\|}{1-\rho}\widehat{w}\right)^2 \left(\frac{C_1\pi^2}{6}\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)^2 + 2\right) \\ = & O\left(\left(\mu_{\mathsf{VAR}}(\mathbf{w}) + \mu_{\mathsf{VAR}}(\widehat{\mathbf{w}})\right)^2\right).\end{aligned}$$

2.D Proof of Theorem 2.3.1

First, note that the total cost is given by $J = \sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t + x_T^\top P x_T$. Since we can choose a threshold $\sigma > 0$ arbitrarily small, the error must exceed a threshold σ. Without loss of generality, we suppose that the accumulated error δ exceeds the threshold σ at time $s \geq 0$ and assume the predictions $\widehat{w}_t$, $0 < t < s-1$ are accurate.

Throughout, we define $J_1 \coloneqq \sum_{t=1}^{s-1} x_t^\top Q x_t + u_t^\top R u_t$ and $J_2 \coloneqq \sum_{t=s}^{T-1} x_t^\top Q x_t + u_t^\top R u_t$ and use diacritical letters $\widehat{J}, \widehat{x}$, and $\widehat{u}$ to denote the corresponding cost, action and state of the threshold algorithm (Algorithm 1). We consider the best online algorithm (with no predictions available) that minimizes its corresponding competitive ratio and use diacritical letters $\widetilde{J}, \widetilde{x}$, and $\widetilde{u}$ to denote the corresponding cost, action, and state. The competitive ratio of the best online algorithm is denoted by $C_{\min}$.

Upper Bound on $\widehat{J}_1$

We first provide an upper bound on $\widehat{J}_1$, the first portion of the total cost. For $1 \leq t < s$, the threshold-based algorithm gives

$$\begin{aligned}\widehat{u}_t &= -K\widehat{x}_t - (R + B^\top P B)^{-1} B^\top \left(\sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P \widehat{w}_\tau\right) \\ &= -K\widehat{x}_t - (R + B^\top P B)^{-1} B^\top \left(\sum_{\tau=t}^{T-1} \left(F^\top\right)^{\tau-t} P w_\tau - \eta_t\right).\end{aligned}$$

Lemma 10 in [41] implies

$$J_1 = \mathsf{ALG}(0:T) - \mathsf{ALG}(s:T)$$

where

$$\begin{aligned}\mathsf{ALG}(0:T) = & \sum_{t=0}^{T-1} \left(w_t^\top P w_t + 2 w_t^\top \sum_{i=1}^{T-t-1} \left(F^\top\right)^i P w_{t+i}\right) \\ & - \sum_{t=0}^{T-1} \left(\sum_{i=0}^{T-t-1} \left(F^\top\right)^i P w_{t+i}\right)^\top H \left(\sum_{i=0}^{T-t-1} \left(F^\top\right)^i P w_{t+i}\right) \\ & + \sum_{t=0}^{T-1} \eta_t^\top H \eta_t + x_0^\top P x_0 + 2 x_0^\top \sum_{i=0}^{T-1} \left(F^\top\right)^{i+1} P w_i, \end{aligned} \tag{2.43}$$

and

$$\begin{aligned}\mathsf{ALG}(s:T) \coloneqq & \sum_{t=0}^{T-s-1}\left(w_{t+s}^\top P w_{t+s} + 2w_{t+s}^\top \sum_{i=1}^{T-s-t-1}\left(F^\top\right)^i P w_{t+s+i}\right)\\ & - \sum_{t=0}^{T-s-1}\left(\sum_{i=0}^{T-s-t-1}\left(F^\top\right)^i P w_{t+s+i}\right)^\top H\left(\sum_{i=0}^{T-s-t-1}\left(F^\top\right)^i P w_{t+s+i}\right)\\ & + \sum_{t=0}^{T-s-1}\eta_{t+s}^\top H\eta_{t+s} + x_s^\top P x_s + 2x_s^\top \sum_{i=0}^{T-s-1}\left(F^\top\right)^{i+1} P w_{i+s}. \end{aligned} \tag{2.44}$$

Rewriting (2.44),

$$\begin{aligned}\mathsf{ALG}(s:T) \coloneqq & \sum_{t=s}^{T-1}\left(w_t^\top P w_t + 2w_t^\top \sum_{i=1}^{T-t-1}\left(F^\top\right)^i P w_{t+i}\right)\\ & - \sum_{t=s}^{T-1}\left(\sum_{i=0}^{T-t-1}\left(F^\top\right)^i P w_{t+i}\right)^\top H\left(\sum_{i=0}^{T-t-1}\left(F^\top\right)^i P w_{t+i}\right)\\ & + \sum_{t=s}^{T-1}\eta_t^\top H\eta_t + x_s^\top P x_s + 2x_s^\top \sum_{i=s}^{T-1}\left(F^\top\right)^{i+1-s} P w_i. \end{aligned} \tag{2.45}$$

Therefore, combining (2.43) and (2.45),

$$\begin{aligned}J_1 = & \sum_{t=0}^{s-1}\left(w_t^\top P w_t + 2w_t^\top \sum_{i=1}^{T-t-1}\left(F^\top\right)^i P w_{t+i}\right)\\ & - \sum_{t=0}^{s-1}\left(\sum_{i=0}^{T-t-1}\left(F^\top\right)^i P w_{t+i}\right)^\top H\left(\sum_{i=0}^{T-t-1}\left(F^\top\right)^i P w_{t+i}\right)\\ & + \sum_{t=0}^{s-1}\eta_t^\top H\eta_t + x_0^\top P x_0 + 2x_0^\top \sum_{i=0}^{T-1}\left(F^\top\right)^{i+1} P w_i - x_s^\top P x_s - 2x_s^\top \sum_{i=s}^{T-1}\left(F^\top\right)^{i+1-s} P w_i. \end{aligned}$$

Denote by $\Delta J_1 \coloneqq \left|J_1 - \widehat{J}_1\right|$. We obtain

$$\begin{aligned}\Delta J_1 = & \sum_{t=0}^{s-1}\eta_t^\top H\eta_t + x_s^\top P x_s - \widehat{x}_s^\top P\widehat{x}_s + 2(x_s-\widehat{x}_s)^\top \sum_{i=s}^{T-1}\left(F^\top\right)^{i+1-s} P w_i\\ = & \sum_{t=0}^{s-1}\eta_s^\top F^{s-t} H\left((F^\top)^{s-t}\eta_s\right) + x_s^\top P x_s - \widehat{x}_s^\top P\widehat{x}_s + 2(x_s-\widehat{x}_s)^\top \sum_{i=s}^{T-1}\left(F^\top\right)^{i+1-s} P w_i\\ \leq & \frac{c\|H\|}{1-\rho^2}\frac{c^2\|P\|^2R^2}{(1-\rho)^2} + 2\|P\|\|x_s\|\|x_s-\widehat{x}_s\| + \|x_s-\widehat{x}_s\|^2 + 2\|x_s-\widehat{x}_s\|\frac{c\|P\|\rho}{1-\rho}. \end{aligned}$$

Since the following is true:

$$\begin{aligned} x_s - \widehat{x}_s =& A(x_{s-1} - \widehat{x}_{s-1}) + B(u_{s-1} - \widehat{u}_{s-1}) \\ =& (A - BK)(x_{s-1} - \widehat{x}_{s-1}) + B(R + B^\top PB)^{-1} B^\top \eta_{s-1} \\ =& \sum_{t=0}^{s-1} (F^\top)^{s-t-1} B(R + B^\top PB)^{-1} B^\top \eta_t, \end{aligned}$$

we have

$$\|x_s - \widehat{x}_s\| \leq \frac{c^2 \|B(R + B^\top PB)^{-1} B^\top\| R}{(1-\rho)^2}.$$

If $\|x_s\| = O(1)$, then $\Delta J_1 = O(1)$, else

$$\frac{\Delta J_1}{J_1} \leq \frac{O(1) \cdot \|x_s\| + O(1)}{x_s^\top Q x_s} \to 0.$$

Therefore, as a conclusion, $\widehat{J}_1$ can be bounded from above by

$$\widehat{J}_1 \leq J_1 + O(1). \tag{2.46}$$

Upper Bound on $\widehat{J}_2$

For section 2.D, we know that $\|x_s - \widehat{x}_s\| = O(1)$. Let $\widetilde{J}_2$ denote the cost by running 1-confident algorithm from $\widehat{x}_s$ with correct prediction, and $\widetilde{x}_t$ denote the state we get in the procedure. Then

$$\|x_t - \widetilde{x}_t\| = \|(A - BK)(x_{t-1} - \widetilde{x}_{t-1})\| = \|F^{t-s}(x_s - \widetilde{x}_s)\| = \|F^{t-s}(x_s - \widehat{x}_s)\|.$$

Therefore,

$$\begin{aligned} |J_2 - \widetilde{J}_2| \leq & \left| \sum_{t=s}^{T-1} (\widetilde{x}_t - x_t)^\top Q x_t + x_t^\top Q (\widetilde{x}_t - x_t) + (\widetilde{x}_t - x_t)^\top Q (\widetilde{x}_t - x_t) \right| \\ & + | \sum_{t=s}^{T-1} (\widetilde{u}_t - u_t)^\top R u_t + u_t^\top R (\widetilde{u}_t - u_t) + (\widetilde{u}_t - u_t)^\top R (\widetilde{u}_t - u_t) | \\ & + \left| (\widetilde{x}_T - x_T)^\top P x_T + x_T^\top P (\widetilde{x}_T - x_T) + (\widetilde{x}_T - x_T)^\top P (\widetilde{x}_T - x_T) \right| \\ \leq & \sum_{t=s}^{T-1} \left(\|Q\| + \|K^\top R K\| \right) \left\| F^{2t-2s} \right\| \|x_s - \widehat{x}_s\|^2 \\ & + \sum_{t=s}^{T-1} 2 \left\| F^{t-s} \right\| \|x_s - \widehat{x}_s\| \left(\|Q\| \|x_t\| + \|RK\| \|u_t\| \right) \\ & + 2 \left\| F^{T-s} \right\| \|P\| \|\|x_s - \widehat{x}_s\| \|x_T\| + \left\| F^{2T-2s} \right\| \|P\| \|x_s - \widehat{x}_s\|^2 \\ = & \sum_{t=s}^{T-1} 2 \left\| F^{t-s} \right\| \|x_s - \widehat{x}_s\| \left(\|Q\| \|x_t\| + \|RK\| \|u_t\| \right) \\ & b + 2 \left\| F^{T-s} \right\| \|P\| \| \|x_s - \widehat{x}_s\| \|x_T\| + O(1). \end{aligned}$$

If $\|x_t\| = O(1)$ and $\|u_t\| = O(1)$ for all t, then $\left|J_2 - \widetilde{J}_2\right| = O(1)$. Otherwise, suppose $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ and $u_{j_1}, u_{j_2}, \ldots, u_{j_l}$ are some functions of T, then for any $1 \le m \le k$ and $1 \le n \le l$, $\|x_{i_m}\|/x_{i_m}^\top Q x_{i_m} \to 0$ and $\|u_{j_n}\|/u_{j_n}^\top R u_{j_n} \to 0$. Therefore,

$$\frac{\left|J_2 - \widetilde{J}_2\right|}{J_2} \le 2\|x_s - \widehat{x}_s\| \frac{\sum_{m=1}^{k} \left\|F^{i_m - s}\right\| \|Q\| \|x_{i_m}\| + \sum_{n=1}^{l} \left\|F^{j_n - s}\right\| \|RK\| \|u_{j_n}\|}{J_2} + \frac{O(1)}{J_2} \to 0.$$

Combining the two cases, we can conclude that

$$\left|J_2 - \widetilde{J}_2\right| \le J_2 + O(1). \tag{2.47}$$

Therefore, from (2.46) and (2.47), we conclude that

$$\begin{aligned} \widehat{J} = \widehat{J}_1 + \widehat{J}_2 &\le J_1 + O(1) + C_{\min}\widetilde{J}_2 \\ &\le J_1 + O(1) + C_{\min}(J_2 + O(1)) \\ &= C_{\min} J + O(1). \end{aligned}$$

The proof completes by noticing that when the prediction error is zero and $\widehat{w}_t = w_t$ for all $t = 0, \ldots, T-1$, the accumulated error δ will always be 0 and since the threshold σ is positive, the algorithm is always optimal and 1-consistent. As a result, Algorithm 1 is 1-consistent and $(C_{\min} + o(1))$-robust.

2.E Experimental Setup

In our three case studies, we consider i.i.d. prediction errors, i.e., $e_t = \widehat{w}_t - w_t$ is an i.i.d. additive prediction noise. To illustrate the effects of randomness for simulating the worst-case performance, we consider varying types of noise in the case studies. For the robot tracking case, we set $e_t = cX$ where $X \sim B(10, 0.5)$ is a binomial random variable with 10 trials and a success probability 0.5 and $c > 0$ is a scaling parameter. For the battery-buffered EV charging case, we set $e_t = Y$ where $X \sim N(0, \sigma^2)$ is a normal random variable with zero mean and σ^2 is a variance that can be varied to generate varying prediction error. For the Cart-Pole problem, we set $e_t = Z w_t$ where $w_t = 60 \times B$ with $\eta \coloneqq l\left(\frac{4}{3} - \frac{m}{m+M}\right)$,

$$B \coloneqq \begin{bmatrix} 0 \\ \frac{(m+M)\eta + ml}{(m+M)^2\eta} \\ 0 \\ -\frac{1}{(m+M)\eta}, \end{bmatrix}$$

and $Z \sim N(0, \sigma^2)$ is a normal random variable with zero mean and σ^2 is a variance ranging between 0 to 8×10^2. To simulate the worst-case performance of algorithms, in our experiments we run the algorithms 5 times, with a new sequence of prediction noise generated at each time and choose the one with the largest overall cost.

Finally, Table 2.E.1 and Table 2.E.2 list the detailed settings and the hyper-parameters used in the robot tracking, battery-buffered EV charging and Cart-Pole case studies.

Table 2.E.1: Hyper-parameters used in robot tracking and EV charging.

Robot Tracking	Value	EV Charging	Value
Number of Monte Carlo Tests	5	Number of Monte Carlo Tests	5
Prediction Error Type	*Binomial*	Prediction Error Type	*Gaussian*
State Dimension n	4	State Dimension n	10 (Synthetic); 52 (Realistic)
Action Dimension m	2	Action Dimension m	10 (Synthetic); 52 (Realistic)
Time Horizon Length T	Fig 2.2a: $T = 240$	Time Horizon Length T	240
	Fig 2.2b: $T = 240$		
	Fig 2.3: $T = 200$		
Initialized λ_0	0.3	Charging Efficiency	1
Scaling parameter c	Fig 2.3: $c \in [0, 1]$	Variance σ^2	$\sigma^2 \in [0, 10]$
CPU	Intel® i7-8850H	CPU	Intel® i7-8850H
		Energy Demand E (Synthetic)	5 (kWh)
		Arrival Rate (Realistic)	0.2

Table 2.E.2: Hyper-parameters used in the Cart-Pole problem.

Robot Tracking	Value
Number of Monte Carlo Tests	2000
Prediction Error Type	*Gaussian*
Action Dimension m	1
State Dimension m	4
Time Horizon Length T	200
Variance σ^2	$\sigma^2 \in [0, 800]$
Cart Mass M	$M = 10.0kg$
Pole Mass m	$m = 1.0kg$
Pole Length l	$l = 10.0m$
CPU	Intel® i7-8850H

Chapter 3

NON-LINEAR CONTROL WITH BLACK-BOX AI POLICIES

[1] Tongxin Li, Ruixiao Yang, Guannan Qu, Yiheng Lin, Steven Low, and Adam Wierman. Equipping black-box policies with model-based advice for stable nonlinear control. *Under review*.

In Chapter 2, we have considered a learning-augmented control problem for a linear control system (2.1) (Section 2.2). However, there are plenty of applications that have non-linear dynamics, wherein deep neural network (DNN)-based control/decision-making methods such as deep reinforcement learning/imitation learning have attracted great interests due to the success on a wide range of control tasks such as humanoid locomotion [56], playing Atari [57] and 3D racing games [58]. These methods are typically model-free and are capable of learning policies and value functions for complex and non-linear control tasks directly from raw data. In real-world applications such as autonomous driving, it is impractical to dynamically update the already-deployed policy. In those cases, pre-trained black-box policies are applied. Those partially-optimized solutions on the one hand can sometimes be optimal or near-optimal, but on the other hand can be arbitrarily poor in cases where there is unexpected environmental behavior due to, e.g., sample inefficiency [10], reward sparsity [11], mode collapse [12], high variability of policy gradient [13, 14], or biased training data [15]. This uncertainty raises significant concerns about applications of these tools in safety-critical settings. Meanwhile, for many real-world control problems, crude information about system models exists, e.g, linear approximations of their state transition dynamics [31, 59]. Such information can be useful in providing model-based advice to the machine-learned policies. Therefore, finding a trade-off between black-box AI/ML polices and model-based policies given crude model information is a crucial problem. In a nutshell, our goals are to

1. be aggressive and trust the pre-trained black-box polices if they are optimal or near-optimal, and *2. be conservative and only use the crude model information if the black-box polices are unstable.*

In this chapter, we generalize the learning-augmented control problem considered in Chapter 2 to a non-linear control system and achieve the goals above.

3.1 Introduction

To represent complex and non-linear control tasks, in this chapter we consider the following infinite-horizon dynamical system consisting of a *known* affine part, used for model-based advice, and an *unknown* non-linear residual function, which is (implicitly) used in developing machine-learned (DNN-based) policies:

$$x_{t+1} = \underbrace{Ax_t + Bu_t}_{\textit{Known}\text{ affine part}} + \underbrace{f_t(x_t, u_t)}_{\textit{Unknown}\text{ non-linear residual}}, \qquad \text{for } t = 0, \ldots, \infty, \tag{3.1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ are the system state and the action selected by a controller at time t; A and B are coefficient matrices in the affine part of the system dynamics. Besides the linear components, the system also has a state and action-dependent non-linear residual $f_t : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ at each time $t \geq 0$, representing the modelling error. The matrices A and B are fixed and known coefficients in the linear approximation of the true dynamics (3.1).

Given the affine part of the dynamics (3.1), it is possible to construct a model-based feedback controller $\overline{\pi}$, e.g., a linear quadratic regulator or an $\mathcal{H}_\infty$ controller. Compared with a DNN-based policy $\widehat{\pi}(x_t)$, the induced linear controller often has a worse performance on average due to the model bias, but becomes more stable in an adversarial setting. In other words, a DNN-based policy can be as good as an optimal policy in domains where accurate training data has been collected, but can perform *sub-optimally* in other situations; while a policy based on a linearized system is *stabilizing*, so that it has a guaranteed worst-case performance with bounded system perturbations, but can lose out on performance to DNN-based policies in non-adversarial situations. We illustrate this trade-off for the Cart-Pole problem (see Example 1 in Appendix 1) in Figure 3.1.1. The figure shows a pre-trained TRPO [60, 61] agent and an ARS [62] agent achieve lower costs when the initial angle of the pole is small; but become less stable when the initial angle increases. Using the affine part of the non-linear dynamics, a linear

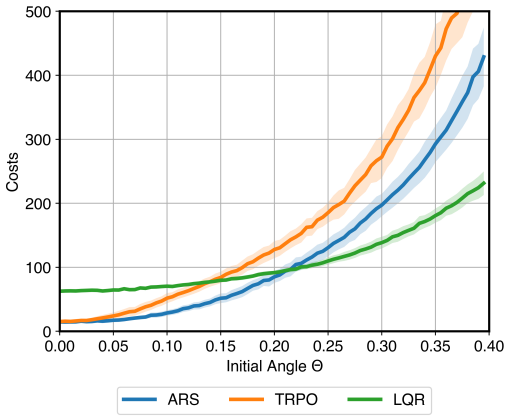


Figure 3.1.1: Costs of pre-trained TRPO and ARS agents and an LQR when the initial pole angle θ (unit: radians) varies.

quadratic regulator achieves better performance when the initial angle becomes large. Motivated by this trade-off, we ask the following question in this chapter:

Can we equip a sub-optimal machine-learned policy $\widehat{\pi}$ with stability guarantees for the non-linear system (3.1) *by utilizing model-based advice from the known, affine part?*

Traditionally, switching between different control policies has been investigated for linear systems [63, 64]. All candidate polices need to be linear and therefore can be represented by their Youla–Kucera parametrizations (or Q-parameterizations). When a policy is a black-box machine-learned policy modeled by a DNN that may be non-linear, how to combine or switch between the policies remains an open problem that is made challenging by the fact that the model-free policy typically has no theoretical guarantees associated with its performance. On the one hand, a model-free policy works well on average but, on the other hand, model-based advice stabilizes the system in extreme cases.

Contributions. In this work, we propose a novel, adaptive policy that combines model-based advice with a black-box machine-learned controller to guarantee stability while retaining the performance of the machine-learned controller when it performs well. In particular, we consider a non-linear control problem whose dynamics is given in (3.1), where we emphasize that the unknown non-linear residual function $f_t(x_t, u_t)$ is time-varying and depends not only on the state x_t but also the action u_t at each time t. Our first result is a negative result (Theorem 3.3.1) showing that a naive convex combination of a black-box model-free policy with model-based advice can lead to instability, even if both policies are stabilizing individually. This negative result highlights the challenges associated with combining model-based and model-free approaches.

Next, we present a general policy that adaptively combines a model-free black-box policy with model-based advice (Algorithm 4). We assume that the model-free policy has some consistency error ε, compared with the optimal policy and that the residual functions $(f_t : t \geq 0)$ are Lipschitz continuous with a Lipschitz constant $C_\ell > 0$. Instead of employing a hard-switching between policies, we introduce a time-varying *confidence coefficient* λ_t that only decays and switches a black-box model-free policy into a stabilizing model-based policy in a smooth way during operation as needed to ensure stabilization. The sequence of confidence coefficients converges to $\lambda \in [0, 1]$. Our main result is the following theorem, which establishes

a trade-off between competitiveness (Theorem 3.4.2) and stability (Theorem 3.4.1) of this adaptive algorithm.

Theorem (Informal)**.** *With system assumptions (Assumption 1, 2 and an upper bound on C_ℓ), the adaptive λ-confident policy (Algorithm 4) has the following properties: (a) the policy is exponentially stabilizing whose decay rate increases when λ decreases; and (b) when the consistency error ε is small, the competitive ratio of the policy satisfies*

$$\mathsf{CR}(\varepsilon) = (1-\lambda) \times \underbrace{O(\overline{\mathsf{CR}}_{\text{model}})}_{\textit{Model-based bound}} + \underbrace{O\left(1/(1-O(\varepsilon))\right)}_{\textit{Model-free error}} + \underbrace{O(C_\ell \|x_0\|_2)}_{\textit{Non-linear dynamics error}} . \tag{3.2}$$

The theorem shows that the adaptive λ-confident policy is guaranteed to be stable. Furthermore, if the black-box policy is close to an optimal control policy (in the sense that the consistency error ε is small), then the adaptive λ-confident policy has a bounded competitive ratio that consists of three components. The first one is a bound inherited from a model-based policy; the second term depends on the sub-optimality gap between a black-box policy and an optimal policy; and the last term encapsulates the loss induced by switching from a policy to another and it scales with the ℓ_2 norm of an initial state x_0 and the non-linear residuals (depending on the Lipschitz constant C_ℓ).

Our results imply an interesting trade-off between stability and sub-optimality, in the sense that if λ is smaller, it is guaranteed to stabilize with a higher rate and if λ becomes larger, it is able to have a smaller competitive ratio bound when provided with a high-quality black-box policy. Different from the linear case, where a cost characterization lemma can be directly applied to bound the difference between the policy costs and optimal costs in terms of the difference between their actions [23], for the case of non-linear dynamics (3.1), we introduce an auxiliary linear problem to derive an upper bound on the dynamic regret, whose value can be decomposed into a quadratic term and a term induced by the non-linearity. The first term can be bounded via a generalized characterization lemma and becomes the *model-based bound* and *model-free error* in (3.2). The second term becomes a *non-linear dynamics error* via a novel sensitivity analysis of an optimal non-linear policy based on its Bellman equation. Finally, we use the Cart-Pole problem to demonstrate the efficacy of the adaptive λ-confident policy.

Related work. Our work is related to a variety of classical and learning-based policies for control and reinforcement learning (RL) problems that focus on combining model-based and model-free approaches.

Combination of model-based information with model-free methods. This work adds to the recent literature seeking to combine model-free and model-based policies for online control. Some prominent recent papers with this goal include the following. First, MPC methods with penalty terms learned by model-free algorithms are considered in [65]. Second, deep neural network dynamics models are used to initialize a model-free learner to improve the sample efficiency while maintaining the high task-specific performance [66]. Third, using this idea, the authors of [59] consider a more concrete dynamical system $x_{t+1} = Ax_t + Bu_t + f(x_t)$ (similar to the dynamics in (3.1) considered in this work) where f is a state-dependent function and they show that a model-based initialization of a model-free policy is guaranteed to converge to a near-optimal linear controller. Another approach uses an $\mathcal{H}_\infty$ controller integrated into model-free RL algorithms for variance reduction [13]. Finally, the model-based value expansion is proposed in [67] as a method to incorporate learned dynamics models in model-free RL algorithms. Broadly, despite many heuristic combinations of model-free and model-based policies demonstrating empirical improvements, there are few theoretical results explaining and verifying the success of the combination of model-free and model-based methods for control tasks. Our work contributes to this goal.

Combining stabilizing linear controllers. The proposed algorithm in this work combines existing controllers and so is related to the literature of combining stabilizing linear controllers. A prominent work in this area is [63], which shows that with proper controller realizations, switching between a family of stabilizing controllers uniformly exponentially stabilizes a linear time-invariant (LTI) system. Similar results are given in [64]. The techniques applied in [63, 64] use the fact that all the stabilizing controllers can be expressed using the Youla parameterization. Different from the classical results of switching between or combining stabilizing controllers, in this work, we generalize the idea to the combination of a linear model-based policy and a model-free policy, that can be either linear or non-linear.

Learning-augmented online problems. Recently, the idea of augmenting robust/competitive online algorithms with machine-learned advice has attracted attention in online problems in settings like online caching [3], ski-rental [5, 6], smoothed online convex optimization [2] and linear quadratic control [23]. In many of these

learning-augmented online algorithms, a convex combination of machine-learned (untrusted) predictions and robust decisions is involved. For instance, in [23], competitive ratio upper bounds of a λ-confident policy are given for a linear quadratic control problem. The policy $\lambda\pi_{\text{MPC}} + (1-\lambda)\pi_{\text{LQR}}$ combines linearly a linear quadratic regulator π_{LQR} and an MPC policy π_{MPC} with machine-learned predictions where $\lambda \in [0, 1]$ measures the confidence of the machine-learned predictions. To this point, no results on learning-augmented controllers for non-linear control exist. In this work, we focus on the case of non-linear dynamics and show a general negativity result (Theorem 3.3.1) such that a simple convex combination between two policies can lead to unstable outputs and then proceed to provide a new approach that yields positive results.

	Setting	**Known**	**Unknown**	(Partial) **Assumption**(s)	**Objective**
[27]	Episodic	h	f	$h, f \in C^{0,1}$	Safe exploration
[13]	Episodic	f^{known}	$f^{unknown}$	$\|\widehat{\pi} - \overline{\pi}\|_2 \leq C_\pi$ Stabilizable f^{known}	Lyapunov stability
[31]	Episodic	A, B	$g_t(x)$	Hurwitz A	Input-ouput stability
[59]	Episodic	A, B	$f(x)$	$f \in C^{0,1}$, Stabilizable A, B	Lyapunov stability
[28–30]	Episodic		(C)MDP	Feasible baseline [29]	Safety and stability
This work	1-trajectory	A, B	$f_t(x, u)$	$f_t \in C^{0,1}$, Stabilizable A, B	Stability, CR bound

Stability-certified RL. Another highly related line of work is the recent research on developing safe RL with stability guarantees. In [27], Lyapunov analysis is applied to guarantee the stability of a model-based RL policy. If an $\mathcal{H}_\infty$ controller $\widehat{\pi}_{\mathcal{H}_\infty}$ is close enough to a model-free deep RL policy $\overline{\pi}_{\text{RL}}$, by combining the two policies linearly $\lambda\widehat{\pi}_{\mathcal{H}_\infty} + (1-\lambda)\overline{\pi}_{\text{RL}}$ at each time in each training episode, asymptotic stability and forward invariance can be guaranteed using Lyapunov analysis but the convergence rate is not provided [13]. In practice, [13] uses an empirical approach to choose a time-varying factor λ according to the temporal difference error. Robust model predictive control (MPC) is combined with deep RL to ensure safety and stability [28]. Using regulated policy gradient, input-output stability is guaranteed for a continuous non-linear control model $f_t(x(t)) = Ax(t) + Bu(t) + g_t(x(t))$ [31]. In those works, a common assumption needs to be made is the ability to access and update the deep RL policy during the episodic training steps. Moreover, in the state-of-the-art results, the stability guarantees are proven, either considering an aforementioned episodic setting when the black-box policy can be improved or customized [27, 31], or assuming a small and bounded output distance between a black-box policy and a stabilizing policy for any input states to construct a Lyapunov equation [13], which is less realistic. Stability guarantees under different model

assumptions such as (constrained) MDPs have been studied [28–30]. Different from the existing literature, the result presented in this work is unique and novel in the sense that we consider stability and sub-optimality guarantee for black-box deep policies in a single trajectory such that we can neither learn from the environments nor update the deep RL policy through extensive training steps. Denote by $C^{0,1}$ the class of Lipschiz continuous functions (with domains, ranges and norms specified according to the contexts), the related results are summarized in the table above.

3.2 Background and Model

We consider the following infinite-horizon quadratic control problem with non-linear dynamics:

$$\min_{(u_t:t\geq 0)} \sum_{t=0}^{\infty} x_t^\top Q x_t + u_t^\top R u_t, \text{ subject to (3.1)} \tag{3.3}$$

where in the problem $Q, R \succ 0$ are $n \times n$ and $m \times m$ positive definite matrices and each $f_t : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ in (3.1) is an unknown non-linear function representing state and action-dependent perturbations. An initial state x_0 is fixed. We use the following assumptions throughout this chapter. Our first assumption is the Lipschitz continuity assumption on the residual functions and it is standard [59]. Note that $\|\cdot\|$ denotes the Euclidean norm throughout the chapter.

Assumption 1 (Lipschitz continuity)**.** The function $f_t : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is Lipschitz continuous for any $t \geq 0$, i.e., there is a constant $C_\ell \geq 0$ such that $\|f_t(x) - f_t(y)\| \leq C_\ell \|x - y\|$ for any $x, y \in \mathbb{R}^n$ and $t \geq 0$. Moreover, $f(\mathbf{0}) = 0$.

Next, we make a standard assumption on the system stability and cost function [68, 69].

Assumption 2 (System stabilizability and costs)**.** The pair of matrices (A, B) is stabilizable, i.e., there exists a real matrix K such that the spectral radius $\rho(A - BK) < 1$. We assume $Q, R \geq \sigma I$. Furthermore, denote $\kappa := \max\{2, \|A\|, \|B\|\}$.

In summary, our control agent is provided with a black-box policy $\widehat{\pi}$ and system parameters A, B, Q, R. The goal is to utilize $\widehat{\pi}$ and system information to minimize the quadratic costs in (3.3), without knowing non-linear residuals $(f_t : t \geq 0)$. Next, we present our policy assumptions.

Model-based advice. In many real-world applications, linear approximations of the true non-linear system dynamics are known, i.e., the known affine part of (3.1)

is available to construct a stabilizing policy $\overline{\pi}$. To construct $\overline{\pi}$, the assumption that (A, B) is stabilizable implies that the following discrete algebraic Riccati equation (DARE) has a unique semi-positive definite solution P that stabilizes the closed-loop system [70]:

$$P = Q + A^\top PA - A^\top PB(R + B^\top PB)^{-1}B^\top PA. \tag{3.4}$$

Given P, define $K := (R + B^\top PB)^{-1}B^\top PA$. The closed-loop system matrix $F := A - BK$ must have a spectral radius $\rho(F)$ less than 1. Therefore, the Gelfand's formula implies that there must exist a constant $C_F > 0$, $\rho \in (0, 1)$ such that $\|F^t\| \leq C_F\rho^t$, for any $t \geq 0$. The model-based advice considered in this work is then defined as a sequence of actions $(u_t : t \geq 0)$ provided by a linear quadratic regulator (LQR) such that $u_t = \overline{\pi}(x_t) = -Kx_t$.

Black-box model-free policy. To solve the non-linear control problem in (3.3), we take advantage of both model-free and model-based approaches. We assume a pre-trained model-free policy, whose policy is denoted by $\widehat{\pi} : \mathbb{R}^n \to \mathbb{R}^m$, is provided beforehand. The model-free policy is regarded as a "black box," whose detail is not the major focus in this work. The only way we interact with it is to obtain a suggested action $\widehat{u}_t = \widehat{\pi}(x_t)$ when feeding into it the current system state x_t. The performance of the model-free policy is not guaranteed and it can make some error, characterized by the following definition, which compares $\widehat{\pi}$ against a clairvoyant optimal controller π_t^* knowing the non-linear residual perturbations in hindsight:

Definition 3.2.1 (ε-*consistency*)**.** A policy $\pi : \mathbb{R}^n \to \mathbb{R}^m$ is called ***ε-consistent*** if there exists $\varepsilon > 0$ such that for any $x \in \mathbb{R}^n$ and $t \geq 0$, $\left\|\pi(x) - \pi_t^*(x)\right\| \leq \varepsilon\|x\|$ where π_t^* denotes an optimal policy at time t knowing all the non-linear residual perturbations $(f_t : t \geq 0)$ in hindsight and ε is called a ***consistency error***.

The parameter ε measures the difference between the action given by the oracle policy $\widehat{\pi}$ and the optimal action given the state x. There is no guarantee that ε is small. With prior knowledge of the non-linearity of system gained from data, the sub-optimal model-free policy $\widehat{\pi}$ suffers a *consistency error* $\varepsilon > 0$, which can be either small if the black-box policy is trained by unbiased data; or high because of the high variability issue for policy gradient deep RL algorithms [13, 14] and distribution shifts of the environments. In these cases, the error $\varepsilon > 0$ can be large. In this work, we augment a black-box model-free policy with stability guarantees using the idea of adaptively switching it to a model-based stabilizing policy $\overline{\pi}$, which

often exists provided with exact or estimates of system parameters A, B, Q and R. The linear stabilizing policy is conservative and highly sub-optimal as it is neither designed based on the exact non-linear model nor interacts with the environment like the training of $\widehat{\pi}$ potentially does.

Performance metrics. Our goal is to ensure stabilization of states while also providing good performance, as measured by the competitive ratio. Formally, a policy π is (asymptotically) *stabilizing* if it induces a sequence of states $(x_t : t \geq 0)$ such that $\|x_t\| \to 0$ as $t \to \infty$. If there exist $C > 0$ and $0 \leq \gamma < 1$ such that $\|x_t\| \leq C\gamma^t \|x_0\|$ for any $t \geq 0$, the corresponding policy is said to be exponentially stabilizing. To define the competitive ratio, let OPT be the offline optimal cost of (3.3) induced by optimal control policies $(\pi_t^* : t \geq 0)$ when the non-linear residual functions $(f_t : t \geq 0)$ are known in hindsight, and ALG be the cost achieved by an online policy. Throughout this chapter we assume $\mathsf{OPT} > 0$. We formally define the competitive ratio as follows.

Definition 3.2.2. Given a policy, the corresponding **competitive ratio**, denoted by CR, is defined as the smallest constant $C \geq 1$ such that $\mathsf{ALG} \leq C \cdot \mathsf{OPT}$ for fixed A, B, Q, R satisfying Assumption 2 and any adversarially chosen residual functions $(f_t : t \geq 0)$ satisfying Assumption 1.

3.3 Warmup: A Naive Convex Combination

The main results in this work focus on augmenting a black-box policy $\widehat{\pi}$ with stability guarantees while minimizing the quadratic costs in (3.3), provided with linear system parameters A, B, Q, R of a non-linear system. Before proceeding to our policy, to highlight the challenge of combining model-based advice with model-free policies in this setting we first consider a simple strategy for combining the two via a convex combination. This is an approach that has been proposed and studied previously, e.g., [13, 23]. However, we show that it can be problematic in that it can yield an unstable policy even when the two policies are stabilizing individually. Then, in Section 3.4, we propose an approach that overcomes this challenge.

A natural approach for incorporating model-based advice is a convex combination of a model-based control policy $\overline{\pi}$ and a black-box model-free policy $\widehat{\pi}$. The combined policy generates an action $u_t = \lambda\widehat{\pi}(x_t) + (1-\lambda)\overline{\pi}(x_t)$ given a state x_t at each time, where $\lambda \in [0, 1]$. The coefficient λ determines a confidence level such that if λ is larger, we trust the black-box policy more and vice versa. In the following, however,

we highlight that, in general, the convex combination of two polices can yield an unstable policy, even if the two policies are stabilizing, with a proof in Appendix 3.D.

Theorem 3.3.1. *Assume B is an $n \times n$ full-rank matrix with $n > 1$. For any $\lambda \in (0, 1)$ and any linear controller K_1 satisfying $A - BK_1 \neq 0$, there exists a linear controller K_2 that stabilizes the system such that their convex combination $\lambda K_2 + (1 - \lambda)K_1$ is unstable, i.e., the spectral radius $\rho(A - B(\lambda K_2 + (1 - \lambda)K_1)) > 1$.*

Theorem 3.3.1 brings up an issue with the strategy of combining a stabilizing policy with a model-free policy. Even if both the model-based and model-free policies are stabilizing, the combined controller can lead to unstable state outputs. In general, the space of stabilizing linear controllers $\{K \in \mathbb{R}^{n\times m} : K \text{ is stabilizing}\}$ is nonconvex [71]. The result in Theorem 3.3.1 is a stronger statement. It implies that for any arbitrarily chosen linear policy K_1 and a coefficient $\lambda \in (0, 1)$, we can always adversarially select a second policy K_2 such that their convex combination leads to an unstable system. It is worth emphasizing that the second policy does not necessarily have to be a complicated non-linear policy. Indeed, in our proof, we construct a linear policy K_2 to derive the conclusion. In our problem, the second policy K_2 is assumed to be a black-box policy $\widehat{\pi}$ potentially parameterized by a deep neural network, yielding much more uncertainty on a similar convex combination. As a result, we must be careful when combining policies together.

Note that the idea of applying a convex combination of an RL policy and a control-theoretic policy linearly is not a new approach and similar policy combinations have been proposed in previous studies [13, 23]. However, in those results, either the model-free policy is required to satisfy specific structures [23] or to be close enough to the stabilizing policy [13] to be combined. In [23], a learning-augmented policy is combined with a linear quadratic regulator, but the learning-augmented policy has a specific form and it is not a black-box policy. In [13], a deep RL policy $\widehat{\pi}$ is combined with an $\mathcal{H}^\infty$ controller $\overline{\pi}$ and they need to satisfy that for any state $x \in \mathbb{R}^n$, $\|\widehat{\pi}(x) - \overline{\pi}(x)\| \leq C_\pi$ for some $C_\pi > 0$. However, it is possible that when the state norm $\|x\|$ becomes large, the two policies in practice behave entirely differently. Moreover, it is hard to justify the benefit of combining two policies, conditioned on the fact that they are already similar. Given that those assumptions are often not satisfied or hard to be verified in practice, we need another approach to guarantee worst-case stability when the black-box policy is biased and in addition ensure sub-optimality if the black-box policy works well.

3.4 Adaptive λ-Confident Control

Motivated by the challenge highlighted in the previous section, we now propose a general framework that adaptively selects a sequence of monotonically decreasing confidence coefficients $(\lambda_t : t \geq 0)$ in order to switch between black-box and stabilizing model-based policies. We show that it is possible to guarantee a bounded competitive ratio when the black-box policy works well, i.e., it has a small consistency error ε, and guarantee stability in cases when the black-box policy performs poorly.

Algorithm 4: Adaptive λ-Confident

Data: System parameters A, B, Q, R, α

for $t \geq 0$ **do**

  if $t = 0$ **then** Initialize $\lambda_0 \longleftarrow 1$

  if $\|x_t\| = 0$ **then** $\lambda_t \longleftarrow \lambda_t$

  else

    Obtain a coefficient λ' based on previous actions $\{u_\tau : \tau \leq t-1\}$, states $\{x_\tau : \tau \leq t\}$ and known system parameters ▷ *Online learning (Eq.* (3.6)*)*

    if $\lambda' > 0$ *and* $\lambda_{t-1} > \alpha$ **then** $\lambda_t \longleftarrow \min\{\lambda', \lambda_{t-1} - \alpha\}$

    else $\lambda_t \longleftarrow 0$

  end

  Generate an action $u_t = \lambda_t \widehat{\pi}(x_t) + (1 - \lambda_t)\overline{\pi}(x_t)$

  Update state according to (3.1)

end

The adaptive λ-confident policy introduced in Algorithm 4 involves an input coefficient λ' at each time. The value of λ_t can either be λ_{t-1} decreased by a fixed step size α, or a variable learned from known system parameters in (3.3) combined with observations of previous states and actions. In Section 3.5, we consider an online learning approach to generate a value of λ' at each time t, but it is worth emphasizing that the adaptive policy in Algorithm 4 and its theoretical guarantees in Section 3.4 do not require specifying a detailed construction of λ'.

The adaptive policy differs from the naive convex combination that has been discussed in Section 3.3 in that it adopts a sequence of time-varying and monotonically decreasing coefficients $(\lambda_t : t \geq 0)$ to combine a black-box policy and a model-based stabilizing policy, where the former policy is adaptively switched to the later one.

The coefficient λ_t converges to $\lim_{t\to\infty} \lambda_t = \lambda$, where the limit λ can be a positive value, if the state converges to a target equilibrium ($\mathbf{0}$ under our model assumptions) before λ_t decreases to zero. This helps stabilize the system under assumptions on the Lipschitz constant C_ℓ of unknown non-linear residual functions and if the black-box policy is near-optimal, a bounded competitive ratio is guaranteed, as we show in the next section.

Theoretical guarantees

The theoretical guarantees we obtain are two-fold. First, we show that the adaptive λ-confident policy in Algorithm 4 is stabilizing, as stated in Theorem 3.4.1. Second, in addition to stability, we show that the policy has a bounded competitive ratio, if the black-box policy used has a small consistency error (Theorem 3.4.2). Note that if a black-box policy has a large consistency error ε, without using model-based advice, it can lead to instability and therefore possibly an unbounded competitive ratio.

Stability. Before presenting our results, we introduce some new notation for convenience. Denote by t_0 the smallest time index when $\lambda_t = 0$ or $x_t = \mathbf{0}$ and note that $\mathbf{0}$ is an equilibrium state. Denote by $\lambda = \lim_{t\to\infty} \lambda_t$. Since $(\lambda_t : t \geq 0)$ is a monotonically decreasing sequence and λ_t has a lower bound, t_0 and λ exist and are unique. Let $H \coloneqq R + B^\top P B$. Define the parameters $\gamma \coloneqq \rho + C_F C_\ell (1 + \|K\|)$ and $\mu \coloneqq C_F \left(\varepsilon \left(C_\ell + \|B\|\right) + C_a^{\mathsf{sys}} C_\ell\right)$ where A, B, Q, R are the known system parameters in (3.3), P has been defined in the Riccati equation (3.4); $C_F > 0$ and $\rho \in (0, 1)$ are constants such that $\|F^t\| \leq C_F \rho^t$, for any $t \geq 0$ as defined in Section 3.2; C_ℓ is the Lipschitz constant in Assumption 1; $\varepsilon > 0$ is the consistency error in Definition 3.2.1; Finally, $C_a^{\mathsf{sys}}, C_b^{\mathsf{sys}}, C_c^{\mathsf{sys}} > 0$ are constants that only depend on the known system parameters in (3.3) and they are listed in Appendix 3.B.

Given the above notation, the theorem below guarantees stability of the adaptive λ-confident policy.

Theorem 3.4.1. *Suppose the Lipschitz constant C_ℓ satisfies $C_\ell < \frac{1-\rho}{C_F(1+\|K\|)}$. The adaptive λ-confident policy (Algorithm 4) is an exponentially stabilizing policy such that $\|x_t\| = O\left((\mu/\gamma)^{t_0}\gamma^t\right)\|x_0\|$.*

For Theorem 3.4.1 to hold such that $\gamma < 1$, the Lipschitz constant C_ℓ needs to have an upper bound $\frac{1-\rho}{C_F(1+\|K\|)}$ where $C_F > 0$ and $\rho \in (0, 1)$ are constants such that $\|F^t\| \leq C_F \rho^t$, for any $t \geq 0$. Since A and B are stabilizable, such C_F and ρ exist. The upper bound only depends on the known system parameters A, B, Q and R. A

small enough Lipschitz constant is required to guarantee stability. For instance, in [59], convergence exponentially to the equilibrium state is guaranteed when the Lipschitz constant satisfies $C_\ell = O\left(\frac{\sigma^2(1-\rho)^8}{\kappa^9 {C_F}^{15}}\right)$.

Competitiveness. Define a constant $\overline{\mathsf{CR}}_{\text{model}} \coloneqq 2\kappa(\frac{C_F\|P\|}{1-\rho})^2/\sigma$. The theorem below implies that when the model-free policy error ε and the Lipschitz constant C_ℓ of the residual functions are small enough, Algorithm 4 is competitive.

Theorem 3.4.2. *Suppose the Lipschitz constant satisfies* $C_\ell < \min\left\{1, C_a^{\mathsf{sys}}, C_c^{\mathsf{sys}}\right\}$. *When the consistency error satisfies* $\varepsilon < \min\left\{\frac{\sigma}{2\|H\|}, \frac{1/C_F - C_a^{\mathsf{sys}} C_\ell}{C_\ell + \|B\|}\right\}$, *the competitive ratio of the adaptive* λ*-confident policy (Algorithm 4) is bounded by*

$$\mathsf{CR}(\varepsilon) = (1-\lambda)\overline{\mathsf{CR}}_{\text{model}} + O\left(1/\left(1 - \frac{2\|H\|}{\sigma}\varepsilon\right)\right) + O(C_\ell\|x_0\|).$$

Combining Theorem 3.4.1 and 3.4.2, our main results are proved when the Lipschitz constant satisfies $C_\ell < \min\left\{1, C_a^{\mathsf{sys}}, C_c^{\mathsf{sys}}, \frac{(1-\rho)}{C_F(1+\|K\|)}\right\}$. Theorem 3.4.1 and 3.4.2 have some interesting implications. First, if the selected time-varying confidence coefficients converge to a λ that is large, then we trust the black-box policy and use a higher weight in the per-step combination. This requires a slower decaying rate of λ_t to zero so as a trade-off, t_0 can be higher and this leads to a weaker stability result and vice versa. In contrast, when the non-linear dynamics in (3.1) becomes linear with unknown constant perturbations, [23] shows a trade-off between robustness and consistency, i.e., a universal competitive ratio bound holds, regardless the error of machine-learned predictions. Different from the linear case where a competitive ratio bound always exists and can be decomposed into terms parameterized by some confidence coefficient λ, for the non-linear system dynamics (3.1), there are additional terms due to the non-linearity of the system that can only be bounded if the consistency error ε is small. This highlights a fundamental difference between linear and non-linear systems, where the latter is known to be more challenging. Proofs of Theorem 3.4.1 and 3.4.2 are provided in Appendix 3.E and 3.F.

3.5 Practical Implementation and Applications

Learning confidence coefficients online

Our main results in the previous section are stated without specifying a sequence of confidence coefficients $(\lambda_t : t \geq 0)$ for the policy; however in the following we introduce an online learning approach to generate confidence coefficients based on observations of actions, states and known system parameters. The negative result in Theorem 3.3.1 highlights that the adaptive nature of the confidence coefficients in

Algorithm 4 are crucial to ensuring stability. Naturally, learning the values of the confidence coefficients $(\lambda_t : t \geq 0)$ online can further improve performance.

In this section, we propose an online learning approach based on a linear parameterization of a black-box model-free policy $\widehat{\pi}_t(x) = -Kx - H^{-1}B^\top \sum_{\tau=t}^{\infty} (F^\top)^{t-\tau} P\widehat{f}_\tau$ where $(\widehat{f}_t : t \geq 0)$ are parameters representing estimates of the residual functions for a black-box policy. Note that when $\widehat{f}_t = f_t^* \coloneqq f_t(x_t^*, u_t^*)$ where x_t^* and u_t^* are optimal state and action at time t for an optimal policy, then the model-free policy is optimal. In general, a black-box model-free policy $\widehat{\pi}$ can be non-linear, the linear parameterization provides an example of how the time-varying confidence coefficients $(\lambda_t : t \geq 0)$ are selected and the idea can be extended to non-linear parameterizations such as kernel methods.

Under the linear parameterization assumption, for linear dynamics, [23] shows that the optimal choice of λ_{t+1} that minimizes the gap between the policy cost and optimal cost for the t time steps is

$$\lambda_{t+1} = \Big(\sum_{s=0}^{t} (\eta(f^*; s, t))^\top H \left(\eta(\widehat{f}; s, t)\right)\Big) / \Big(\sum_{s=0}^{t} \left(\eta(\widehat{f}; s, t)\right)^\top H \left(\eta(\widehat{f}; s, t)\right)\Big), \tag{3.5}$$

where $\eta(f; s, t) \coloneqq \sum_{\tau=s}^{t} (F^\top)^{\tau-s} P f_\tau$. Compared with a linear quadratic control problem, computing λ_t in (3.5) raises two problems. The first is different from a linear dynamical system where true perturbations can be observed, the optimal actions and states are unknown, making the computation of the term $\eta(f^*; s, t-1)$ impossible. The second issue is similar. Since the model-free policy is a black-box, we do not know the parameters $(\widehat{f}_t : t \geq 0)$ exactly. Therefore, we use approximations to compute the terms $\eta\left(f^*; s, t\right)$ and $\eta(\widehat{f}; s, t)$ in (3.5) and the linear parameterization and linear dynamics assumptions are used to derive the approximations, respectively, with details provided in Appendix 3.B. Let $\left(BH^{-1}\right)^\dagger$ denote the Moore–Penrose inverse of BH^{-1}. Combining (3.8) and (3.9) with (3.5) yields the following online-learning choice of a confidence coefficient $\lambda_t = \min\{\lambda', \lambda_{t-1} - \alpha\}$ where $\alpha > 0$ is a fixed step size and

$$\lambda' \coloneqq \frac{\sum_{s=1}^{t-1} \left(\sum_{\tau=s}^{t-1} (F^\top)^{\tau-s} P \left(Ax_\tau + Bu_\tau - x_{\tau+1}\right)\right)^\top B \left(\widehat{u}_s + Kx_s\right)}{\sum_{s=0}^{t-1} \left(\widehat{u}_s + Kx_s\right)^\top \left(BH^{-1}\right)^\dagger B \left(\widehat{u}_s + Kx_s\right)} \tag{3.6}$$

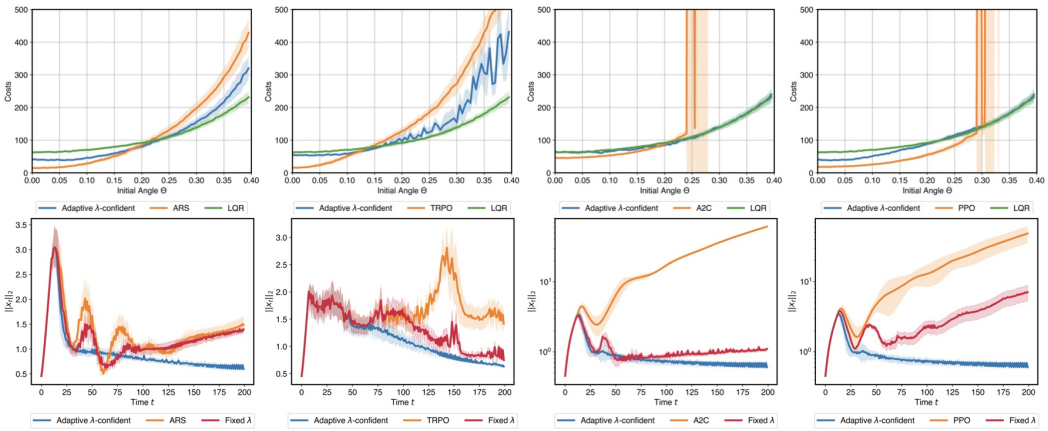


Figure 3.5.1: Competitiveness and stability of the adaptive policy. Top (*Competitiveness*): costs of pre-trained RL agents, an LQR and the adaptive policy when the initial pole angle θ (unit: radians) varies. Bottom (*Stability*): convergence of $\|x_t\|$ in t with $\theta = 0.4$ for pre-trained RL agents, a naive combination (Section 3.3) using a fixed $\lambda = 0.8$ and the adaptive policy.

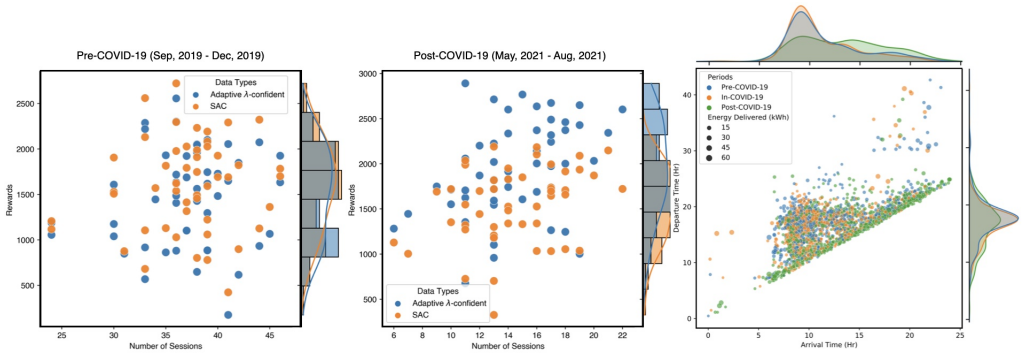


Figure 3.5.2: Simulation results for real-world adaptive EV charging. Left: total rewards of the adaptive policy and SAC for pre-COVID-19 days and post-COVID-19 days. Right: shift of data distributions due to the work-from-home policy.

based on the crude model information A, B, Q, R and previously observed states, model-free actions and policy actions. This online learning process provides a choice of the confidence coefficient in Algorithm 4. It is worth noting that other approaches for generating λ_t exist, and our theoretical guarantees apply to any approach.

Applications

To demonstrate the efficacy of the adaptive λ-confident policy (Algorithm 4), we first apply it to the Cart-Pole OpenAI gym environment (Cart-Pole-v1, Example 1 in

Appendix 3.A) [55].[1] Next, we apply it to an adaptive electric vehicle (EV) charging environment modeled by a real-world dataset [1].

The Cart-Pole problem. We use the Stable-Baselines3 pre-trained agents [61] of A2C [72], ARS [62], PPO [56] and TRPO [60] as four candidate black-box policies. In Figure 3.5.1, the adaptive policy finds a trade-off between the pre-trained black-box polices and an LQR with crude model information (i.e., about 50% estimation error in the mass and length values). In particular, when θ increases, it stabilizes the state while the A2C and PPO policies become unstable when the initial angle θ is large.

Real-world adaptive EV charging. In the EV charging application, a SAC [73] agent is trained with data collected from a pre-COVID-19 period and tested on days before and after COVID-19. Due to a policy change (the work-from-home policy), the SAC agent becomes biased in the post-COVID-19 period (see the right sub-figure in Figure 3.5.2). With crude model information, the adaptive policy has rewards matching the SAC agent in the pre-COVID-19 period and significantly outperforms the SAC agent in the post-COVID-19 period with an average total award 1951.2 versus 1540.3 for SAC. Further details on the hyper-parameters and reward function are included in Appendix 3.A.

[1] The Cart-Pole environment is modified so that quadratic costs are considered rather than discrete rewards.

APPENDIX

3.A Experimental Setup and Supplementary Results

We describe the experimental settings and choices of hyper-parameters and reward/cost functions in the two applications.

Table 3.A.1: Hyper-parameters used in the Cart-Pole problem.

Parameter	**Value**
Number of Monte Carlo Tests	10
Initial angle variation (in rad)	$\theta \pm 0.05$
Cost matrix Q	I
Cost matrix R	$\left[10^{-4}\right]$
Acceleration of gravity g (in $\mathrm{m/s^2}$)	9.8
Pole mass m (in kg)	0.2 for LQR; 0.1 for real environment
Cart mass M (in kg)	2.0 for LQR; 1.0 for real environment
Pole length l (in m)	2
Duration τ (in second)	0.02
Force magnitude F	10
CPU	Intel® i7-8850H

The Cart-Pole Problem

Problem setting. The Cart-Pole problem considered in the experiments is described by the following example.

***Example* 1** (The Cart-Pole Problem)**.** *In the Cart-Pole problem, the goal of a controller is to stabilize the pole in the upright position. Neglecting friction, the dynamical equations of the Cart-Pole problem are*

$$\ddot{\theta} = \frac{g \sin\theta + \cos\theta \left(\frac{-u - ml\dot{\theta}^2 \sin\theta}{m+M} \right)}{l \left(\frac{4}{3} - \frac{m\cos^2\theta}{m+M} \right)}, \quad \ddot{y} = \frac{u + ml\left(\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta\right)}{m+M}$$

where u is the input force; θ is the angle between the pole and the vertical line; y is the location of the pole; g is the gravitational acceleration; l is the pole length; m is the pole mass; and M is the cart mass. Taking $\sin\theta \approx \theta$ and $\cos\theta \approx 1$ and ignoring higher order terms provides a linearized system and the discretized dynamics of the

Cart-Pole problem can be represented as for any t,

$$\underbrace{\begin{bmatrix} y_{t+1} \\ \dot{y}_{t+1} \\ \theta_{t+1} \\ \dot{\theta}_{t+1} \end{bmatrix}}_{x_{t+1}} = \underbrace{\begin{bmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & -\frac{mlg\tau}{\eta(m+M)} & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & \frac{g\tau}{\eta} & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} y_t \\ \dot{y}_t \\ \theta_t \\ \dot{\theta}_t \end{bmatrix}}_{x_t} + \underbrace{\begin{bmatrix} 0 \\ \frac{(m+M)\eta+ml}{(m+M)^2\eta}\tau \\ 0 \\ -\frac{\tau}{(m+M)\eta} \end{bmatrix}}_{B} u_t + f_t\left(y_t, \dot{y}_t, \theta_t, \dot{\theta}_t, u_t\right)$$

where $(y_t, \dot{y}_t, \theta_t, \dot{\theta}_t)^\top$ *denotes the system state at time* t*;* τ *denotes the time interval between state updates;* $\eta := (4/3)l - ml/(m+M)$ *and the function* f_t *measures the difference between the linearized system and the true system dynamics. Note that* $f_t(\mathbf{0}) = 0$ *for all time steps* $t \geq 0$*.*

Policy setting. The pre-trained agents Stable-Baselines3 [61] of A2C [72], ARS [62], PPO [56] and TRPO [60] are selected as four candidate black-box policies. The Cart-Pole environment is modified so that quadratic costs are considered rather than discrete rewards to match our control problem (3.3). The choices of Q and R in the costs and other parameters are provided in Table 3.A.1. Note that we vary the values of m and M in the LQR implementation to model the case of only having crude estimates of linear dynamics. The LQR outputs an action 0 if $-Kx_t + F' < 0$ and 1 otherwise. A shifted force $F' = 15$ is used to model inaccurate linear approximations and noise. The pre-trained RL policies output a binary decision $\{0, 1\}$ representing force directions. To use our adaptive policy in this setting, given a system state x_t at each time t, we implement the following:

$$u_t = \lambda_t \left(2\pi_{\mathrm{RL}}(x_t)F - F\right) + (1 - \lambda_t)\left(2\pi_{\mathrm{LQR}}(x_t)F - F\right)$$

where F is a fixed force magnitude defined in Table 3.A.1; π_{RL} denotes an RL policy; π_{LQR} denotes an LQR policy and λ_t is a confidence coefficient generated based on (3.6). Instead of fixing a step size α, we set an upper bound $\delta = 0.2$ on the learned step size λ' to avoid converging too fast to a pure model-based policy.

Real-World Adaptive EV Charging in Tackling COVID-19

Problem setting. The second application considered is an EV charging problem modeled by real-world large-scale charging data [1]. The problem is formally described below.

***Example* 2** (Adaptive EV charging)**.** *Consider the problem of managing a fleet of electric vehicle supply equipment (EVSE). Let n be the number of EV charging*

Table 3.A.2: Hyper-parameters used in the real-world EV charging problem.

Parameter	**Value**
Problem setting	
Number of chargers n	5
Line limit γ (in kW)	6.6
Duration τ (in minute)	5
Reward coefficient ϕ_1	50
Reward coefficient ϕ_2	0.01
Reward coefficient ϕ_3	10
Policy setting	
Discount γ_{SAC}	0.9
Target smoothing coefficient τ_{SAC}	0.005
Temperature parameter α_{SAC}	0.2
Learning rate	$3 \cdot 10^{-4}$
Maximum number of steps	$10 \cdot 10^6$
Reply buffer size	$10 \cdot 10^6$
Number of hidden layers (all networks)	2
Number of hidden units per layer	256
Number of samples per minibatch	256
Non-linearity	ReLU
Training and testing data	
Pre-COVID-19 (Training)	May, 2019 - Aug, 2019
Pre-COVID-19 (Testing)	Sep, 2019 - Dec, 2019
In-COVID-19	Feb, 2020 - May, 2020
Post-COVID-19	May, 2021 - Aug, 2021
CPU	Intel® i7-8850H

stations. Denote by $x_t \in \mathbb{R}_+^n$ *the charging states of the n stations, i.e.,* $x_t^{(i)} > 0$ *if an EV is charging at station-i and* $x_t^{(i)}$ *(kWh) energy needs to be delivered; otherwise* $x_t^{(i)} = 0$*. Let* $u_t \in \mathbb{R}_+^n$ *be the allocation of energy to the n stations. There is a line limit* $\gamma > 0$ *so that* $\sum_{i=1}^n u_t^{(i)} \leq \gamma$ *for any i and t. At each time t, new EVs may arrive and EVs being charged may depart from previously occupied stations. Each new EV j induces a charging session, which can be represented by* $s_j := (a_j, d_j, e_j, i)$ *where at time* a_j*, EV j arrives at station i, with a battery capacity* $e_j > 0$*, and depart at time* d_j*. Assuming lossless charging, the system dynamics is* $x_{t+1} = x_t + u_t + f_t(x_t, u_t),\ t \geq 0$ *where the non-linear residual functions* $(f_t : t \geq 0)$ *represent uncertainty and*

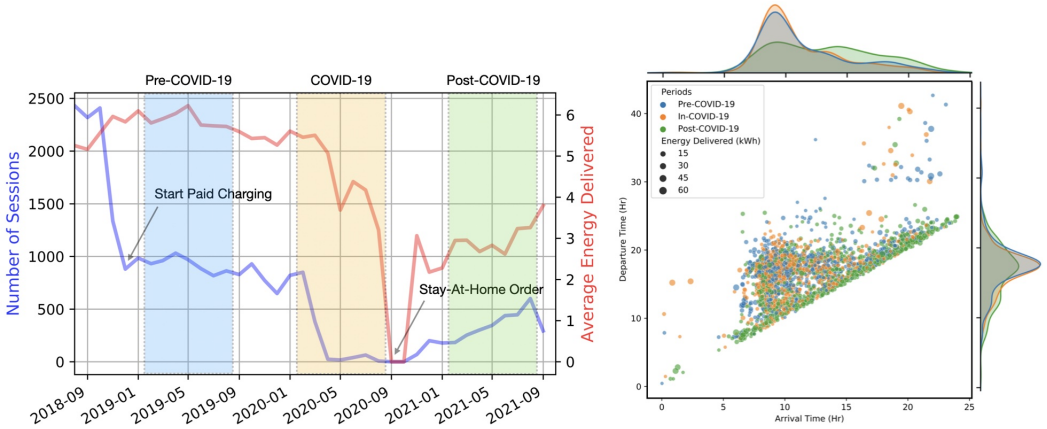


Figure 3.A.1: Illustration of the impact of COVID-19 on charging behaviors in terms of the total number of charging sessions and energy delivered (left) and distribution shifts (right).

constraint violations. Let τ be the time interval between state updates. Given fixed session information $(s_j : j > 0)$, denote by the following sets containing the sessions that are assigned to a charger i and activated (deactivated) at time t:

$$\mathcal{A}_i := \left\{(j,t) : a_j \le t \le a_j + \tau, s_j^{(4)} = i\right\},$$
$$\mathcal{D}_i := \left\{(j,t) : d_j \le t \le d_j + \tau, s_j^{(4)} = i\right\}.$$

The charging uncertainty is summarized as for any $i = 1, \ldots, n$,

$$f_t^{(i)}(x_t, u_t) := \begin{cases} s_j^{(3)} & \text{if } (j,t) \in \mathcal{A}_i \quad \textit{(New sessions are active)} \\ -x_t^{(i)} - u_t^{(i)} & \text{if } (j,t) \in \mathcal{D}_i \text{ or } x_t^{(i)} + u_t^{(i)} < 0 \quad \textit{(Sessions end)} \\ & \textit{(or battery is full)} \\ \frac{\gamma}{\|u_t\|_1} u_t^{(i)} & \text{if } \sum_{i=1}^n u_t^{(i)} > \gamma \quad \textit{(Line limit is exceeded)} \\ 0 & \textit{otherwise} \end{cases}.$$

Note that the non-linear residual functions $(f_t : t \ge 0)$ in Example 2 may not satisfy $f_t(\mathbf{0}) = 0$ for all $t \ge 0$ in Assumption 1. Our experiments further validate that the adaptive policy works well in practice even if some of the model assumptions are violated. The goal of an EV charging controller is to maximize a system-level reward function including maximizing energy delivery, avoiding a penalty due to uncharged

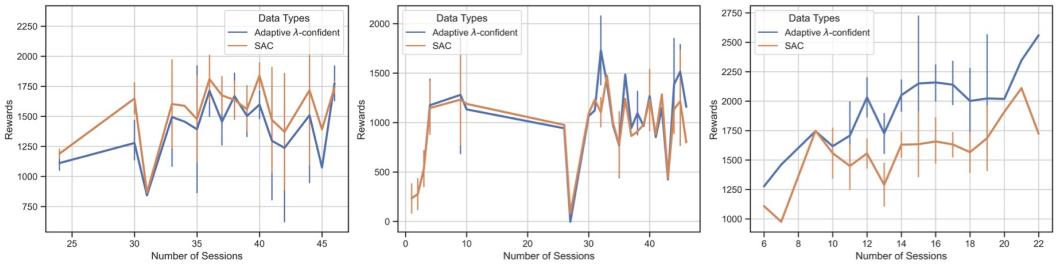


Figure 3.A.2: Bar-plots of rewards/number of sessions corresponding to testing the SAC policy and the adaptive policy on the EV charging environment based on sessions collected from three time periods.

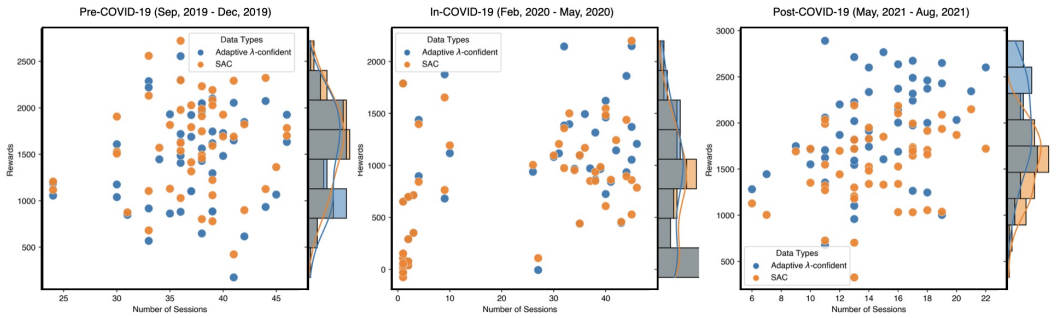


Figure 3.A.3: Supplementary results of Figure 3.5.2 with additional testing rewards for an in-COVID-19 period.

capacities and minimizing electricity costs. The reward function is

$$r(u_t, x_t) :=$$

$$\underbrace{\phi_1 \times \tau \|u_t\|_2}_{\text{Charging rewards}} - \underbrace{\phi_2 \times \|x_t\|_2}_{\text{Unfinished charging}} - \underbrace{\phi_3 \times p_t \|u_t\|_1}_{\text{Electricity cost}} - \underbrace{\phi_4 \times \sum_{i=1}^{n} \mathbf{1}((j,t) \in \mathcal{D}_i) \frac{x_t^{(i)}}{e_j}}_{\text{Penalty}}$$

with coefficients ϕ_1, ϕ_2, ϕ_3 and ϕ_4 shown in Table 3.A.2. The environment is wrapped as an OpenAI gym environment [55]. In our implementation, for convenience, the state x_t is in $\mathbb{R}_+^{2n}$ with additional n coordinates representing remaining charging duration. The electricity prices $(p_t : t \geq 0)$ are average locational marginal prices (LMPs) on the CAISO (California Independent System Operator) day-ahead market in 2016.

Policy setting. We train an SAC [73] policy π_{SAC} for EV charging with 4-month data collected from a real-world charging garage [1] before the outbreak of COVID-19. The public charging infrastructure has 54 chargers and we use the charging history to set up our charging environment with 5 chargers. Knowledge of the linear parts in the non-linear dynamics $x_{t+1} = x_t + u_t + f_t(x_t, u_t),\ t \geq 0$ is assumed to be known, based

Table 3.A.3: Average total rewards for the SAC policy and the adaptive λ-confident policy (Algorithm 4).

Policy	**Pre-COVID-19**	**In-COVID-19**	**Post-COVID-19**
SAC [73]	1601.914	765.664	1540.315
Adaptive	1489.338	839.651	1951.192

on which an LQR controller π_{LQR} is constructed. Our adaptive policy presented in Algorithm 4 learns a confidence coefficient λ_t at each time step t to combine the two policies π_{SAC} and π_{LQR}.

Impact of COVID-19. We test the policies on different periods from 2019 to 2021. The impact of COVID-19 on the charging behavior is intuitive. As COVID-19 became an outbreak in early Feb, 2020 and later a pandemic in May, 2020, limited Stay at Home Order and curfew were issued, which significantly reduce the number of active users per day. Figure 3.A.1 illustrates the dramatic fall of the total number of monthly charging sessions and total monthly energy delivered between Feb, 2020 and Sep, 2020. Moreover, despite the recovery of the two factors since Jan, 2021, COVID-19 has a long-term impact on lifestyle behaviors. For example, the right sub-figure in Figure 3.A.1 shows that the arrival times of EVs (start times of sessions) are flattened in the post-COVID-19 period, compared to a more concentrated arrival peak before COVID-19. The significant shift of distributions highly deteriorates the performance of DNN-based model-free policies, e.g., SAC that are trained on normal charging data collected before COVID-19. In this work, we demonstrate that taking advantage of model-based information, the adaptive λ-confident policy (Algorithm 4) is able to correct the mistakes made by DNN-based model-free policies trained on biased data and achieve more robust charging performance.

Additional experimental results. We provide supplementary experimental results. Besides comparing the periods of pre-COVID-19 and post-COVID-19, we include the testing rewards for an in-COVID-19 period in Figure 3.A.3, together with the corresponding bar-plots in Figure 3.A.2. In addition, the average total rewards for the SAC policy and the adaptive policy are summarized in Table 3.A.3.

3.B Notation and Supplementary Definitions

Summary of Notation

A summary of notation is provided in Table 3.B.1.

Symbol	Definition
System Model	
A, B, Q, R	Linear system parameters
P	Solution of the DARE (3.4)
H	$R + B^\top PB$
K	$H^{-1}B^\top PA$
F	$A - BK$
σ	$Q, R \succeq \sigma I$
κ	$\max\{2, \|A\|, \|B\|\}$
C_F and ρ	$\|F^t\| \le C_F \rho^t$
$f_t : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$	Non-linear residual functions
C_ℓ	Lipschitz constant of f_t
$\widehat{\pi}$	Black-box (model-free) policy
$\overline{\pi}$	Model-based policy (LQR)
ε	Consistency error of a black-box policy
Main Results	
$C_a^{\mathsf{sys}}, C_b^{\mathsf{sys}}, C_c^{\mathsf{sys}}$	Constants defined in Section 3.B
γ	$\rho + C_F C_\ell (1 + \|K\|)$
μ	$C_F \left(\varepsilon \left(C_\ell + \|B\|\right) + C_a^{\mathsf{sys}} C_\ell\right)$
ALG	Algorithm cost
OPT	Optimal cost
CR(ε)	$2\kappa(\frac{C_F\|P\|}{1-\rho})^2/\sigma$
λ	$\lim_{t\to\infty} \lambda_t$
t_0	The smallest time index when $\lambda_t = 0$ or $x_t = \mathbf{0}$

Table 3.B.1: Symbols used in this work.

Constants in Theorem 3.4.1 and 3.4.2

Let $H \coloneqq R + B^\top PB$. With $\sigma > 0$ defined in Assumption 2, the parameters $C_a^{\mathsf{sys}}, C_b^{\mathsf{sys}}, C_c^{\mathsf{sys}} > 0$ in the statements of Theorem 3.4.1 and 3.4.2 (Section 3.4) are the following constants that only depend on the known system parameters in (3.3):

$$
\begin{aligned}
C_a^{\mathsf{sys}} \coloneqq & 1/\Big(2C_F\|R + B^\top PB\|^{-1}\Big(\|PF\| + (1 + \|K\|)\,(\|PB\| + \|P\|) \\
& + \frac{C_b^{\mathsf{sys}}}{2}\|B + I\|(1 + \|F\| + \|K\|)\Big)\Big), \qquad (3.7)\\
C_b^{\mathsf{sys}} \coloneqq & \frac{2{C_F}^2\|P\|(\rho + \overline{C})\,(\rho + (1 + \|K\|))}{1 - (\rho + \overline{C})^2}\sqrt{\frac{\|Q + K^\top RK\|}{\sigma}}, \\
C_c^{\mathsf{sys}} \coloneqq & \|H\|/(4\,\|PB\| + 2\|P\| + C_\nabla(\|B\| + 1)\|B\|).
\end{aligned}
$$

Approximations of Online Learning Steps in Section 3.5

The following approximations of $\eta(\widehat{f}; s, t)$ and $\eta(f^*; s, t)$ in (3.5) are used to derive the expression of λ' in (3.6) for learning the confidence coefficients online:

$$\eta(\widehat{f}; s, t) \approx \sum_{\tau=s}^{\infty} \left(F^\top\right)^{\tau-s} P\widehat{f}_\tau = -\left(H^{-1}B^\top\right)^\dagger (\widehat{u}_s + Kx_s)\,, \tag{3.8}$$

$$\eta(f^*; s, t) = \sum_{\tau=s}^{t} \left(F^\top\right)^{\tau-s} Pf^*_\tau \approx \sum_{\tau=s}^{t} \left(F^\top\right)^{\tau-s} P(x_{\tau+1} - Ax_\tau - Bu_\tau). \tag{3.9}$$

3.C Useful Lemmas

The following lemma generalizes the results in [41].

Lemma 10 (Generalized cost characterization lemma)**.** *Consider a linear quadratic control problem below where $Q, R \succ 0$ and the pair of matrices (A, B) is stabilizable:*

$$\min_{(u_t:t\geq 0)} \sum_{t=0}^{\infty} (x_t^\top Q x_t + u_t^\top R u_t), \text{ subject to } x_{t+1} = Ax_t + Bu_t + v_t \text{ for any } t \geq 0.$$

If at each time $t \geq 0$, $u_t = -Kx_t - H^{-1}B^\top W_t + \eta_t$ where $\eta_t \in \mathbb{R}^m$, then the induced cost is

$$\begin{aligned} &x_0^\top P x_0 + 2x_0^\top F^\top V_0 + \sum_{t=0}^{\infty} \eta_t^\top H \eta_t + \sum_{t=0}^{\infty} \left(v_t^\top P v_t + 2v_t^\top F^\top V_{t+1}\right) \\ &+ \sum_{t=0}^{\infty} \left(W_t^\top B H^{-1} B^\top (W_t - 2V_t) + 2\eta_t^\top B^\top (V_t - W_t)\right) + O(1) \end{aligned}$$

where P is the unique solution of the DARE in (3.4)*, $H \coloneqq R + B^\top P B$, $F \coloneqq A - B(R + B^\top P B)^{-1} B^\top P A = A - BK$, $W_t \coloneqq \sum_{\tau=t}^{\infty} (F^\top)^\tau P w_{t+\tau}$ and $V_t \coloneqq \sum_{\tau=t}^{\infty} (F^\top)^\tau P v_{t+\tau}$.*

Proof. Denote by $\mathsf{COST}_t(x_t; \eta, w)$ the terminal cost at time t given a state x_t with fixed action perturbations $\eta \coloneqq (\eta_t : t \geq 0)$ and state perturbations $w \coloneqq (w_t : t \geq 0)$. We assume $\mathsf{COST}_t(x_t; \eta, w) \coloneqq x_t^\top P x_t + p_t^\top x_t + q_t$. Similar to the proof of Lemma 13 in [41], using the backward induction, the cost can be rewritten as

$$\begin{aligned} \mathsf{COST}_t(x_t; \eta, w) =& \mathsf{COST}_{t+1}(x_{t+1}; \eta, w) + x_t^\top Q x_t + u_t^\top R u_t \\ =& x_t^\top Q x_t + u_t^\top R u_t + (Ax_t + Bu_t + v_t)^\top P (Ax_t + Bu_t + v_t) \\ &+ p_{t+1}^\top (Ax_t + Bu_t + v_t) + q_{t+1} \\ =& x_t^\top Q x_t + (Ax_t + v_t)^\top P (Ax_t + v_t) + (Ax_t + v_t)^\top p_{t+1} + q_{t+1} \\ &+ \underbrace{u_t^\top (R + B^\top P B) u_t}_{(a)} + \underbrace{2u_t^\top B^\top (PAx_t + Pv_t + p_{t+1}/2)}_{(b)}. \end{aligned}$$

Denote by $H \coloneqq R + B^\top PB$. Noting that $u_t = -Kx_t + G_t + \eta_t$ where we denote

$$G_t \coloneqq H^{-1}B^\top W_t,\ W_t \coloneqq \sum_{\tau=t}^{\infty} \left(F^\top\right)^\tau Pw_{t+\tau} \text{ and } V_t \coloneqq \sum_{\tau=t}^{\infty} \left(F^\top\right)^\tau Pv_{t+\tau},$$

it follows that

$$\begin{aligned}
(a) =& (Kx_t + G_t - \eta_t)^\top H (Kx_t + G_t - \eta_t) - 2(Kx_t) + 2(Kx_t)^\top H(G_t - \eta_t) \\
&+ (G_t - \eta_t)^\top (R + B^\top PB)(G_t - \eta_t) \\
(b) =& -2(Kx_t)^\top H(Kx_t) - 2x_t^\top K^\top B^\top Pv_t - x_t^\top K^\top B^\top p_{t+1} - 2(Kx_t)^\top H(G_t - \eta_t) \\
&- 2(G_t - \eta_t)^\top B^\top (Pv_t + p_{t+1}/2),
\end{aligned}$$

implying

$$\begin{aligned}
\mathsf{COST}_t(x_t; \eta, w) =& x_t^\top (Q + A^\top PA - K^\top HK)x_t + x_t^\top F^\top (2Pv_t + p_{t+1}) \\
&+ (G_t - \eta_t)^\top H(G_t - \eta_t) - 2(G_t - \eta_t)^\top B^\top (Pv_t + p_{t+1}/2) \\
&+ v_t^\top Pv_t + v_t^\top p_{t+1} + q_{t+1}.
\end{aligned}$$

According to the DARE in (3.4), since $K^\top HK = A^\top PB(R + B^\top PB)^{-1}B^\top PA$, we get

$$\begin{aligned}
\mathsf{COST}_t(x_t; \eta, w) =& x_t^\top Px_t + x_t^\top F^\top (2Pv_t + p_{t+1}) + (G_t - \eta_t)^\top H(G_t - \eta_t) \\
&- 2(G_t - \eta_t)^\top B^\top (Pv_t + p_{t+1}/2) + v_t^\top Pv_t + v_t^\top p_{t+1} + q_{t+1},
\end{aligned}$$

which implies

$$p_t = 2F^\top (Pv_t + p_{t+1}) = 2\sum_{\tau=t}^{\infty} \left(F^\top\right)^{\tau+1} Pv_{t+\tau} = 2F^\top V_t, \tag{3.10}$$

$$\begin{aligned}
q_t =& q_{t+1} + v_t^\top Pv_t + 2v_t^\top F^\top V_{t+1} + (G_t - \eta_t)^\top H(G_t - \eta_t) \\
&- 2(G_t - \eta_t)^\top B^\top (Pv_t + p_{t+1}/2) \\
=& q_{t+1} + v_t^\top Pv_t + 2v_t^\top F^\top V_{t+1} + (G_t - \eta_t)^\top H(G_t - \eta_t) \\
&- 2G_t^\top B^\top V_t + 2\eta_t^\top B^\top V_t \\
=& q_{t+1} + v_t^\top Pv_t + 2v_t^\top F^\top V_{t+1} + G_t^\top B^\top (W_t - 2V_t) + 2\eta_t^\top B^\top (V_t - W_t) + \eta_t^\top H\eta_t.
\end{aligned} \tag{3.11}$$

Therefore, (3.10) and (3.11) together imply the following general cost characterization:

$$\begin{aligned}
\mathsf{COST}_t(x_t; \eta, w) =& x_0^\top Px_0 + x_0^\top p_0 + q_0 \\
=& x_0^\top Px_0 + 2x_0^\top F^\top V_0 + \sum_{t=0}^{\infty} \eta_t^\top H\eta_t + \sum_{t=0}^{\infty} \left(v_t^\top Pv_t + 2v_t^\top F^\top V_{t+1}\right) \\
&+ \sum_{t=0}^{\infty} \left(G_t^\top B^\top (W_t - 2V_t) + 2\eta_t^\top B^\top (V_t - W_t)\right).
\end{aligned}$$

Rearranging the terms above completes the proof. □

To deal with the non-linear dynamics in (3.1), we consider an auxiliary linear system, with a fixed perturbation $w_t = f_t(x_t^*, u_t^*)$ for all $t \geq 0$ where each x_t^* denotes an optimal state and u_t^* an optimal action, generated by an optimal policy π^*. We define a sequence of linear policies $\left(\pi_t' : t \geq 0\right)$ where $\pi_t' : \mathbb{R}^n \to \mathbb{R}^m$ generates an action

$$u_t' = \pi_t'(x_t) \coloneqq -(R + B^\top P B)^{-1} B^\top \left(P A x_t + \sum_{\tau=t}^{\infty} \left(F^\top\right)^{\tau - t} P f_\tau(x_\tau^*, u_\tau^*) \right), \tag{3.12}$$

which is an optimal policy for the auxiliary linear system. Utilizing Lemma 10, the gap between the optimal cost and algorithm cost for the system in (3.3) can be characterized below.

Lemma 11. *For any $\eta_t \in \mathbb{R}^m$, if at each time $t \geq 0$, a policy $\pi : \mathbb{R}^n \to \mathbb{R}^m$ takes an action $u_t = \pi(x_t) = \pi_t'(x_t) + \eta_t$, then the gap between the optimal cost* OPT *of the non-linear system* (3.3) *and the algorithm cost* ALG *induced by selecting control actions $(u_t : t \geq 0)$ equals to*

$$\begin{aligned}
\mathsf{ALG} - \mathsf{OPT} \leq & \sum_{t=0}^{\infty} \eta_t^\top H \eta_t + O(1) \\
& + 2 \sum_{t=0}^{\infty} \eta_t^\top B^\top \left(\sum_{\tau=t}^{\infty} \left(F^\top\right)^{\tau} P (f_{t+\tau} - f_{t+\tau}^*) \right) \\
& + \sum_{t=0}^{\infty} \left(f_t^\top P f_t - \left(f_t^*\right)^\top P f_t^* \right) + 2 x_0^\top \left(\sum_{t=0}^{\infty} \left(F^\top\right)^{t+1} P \left(f_t - f_t^*\right) \right) \\
& + 2 \sum_{t=0}^{\infty} \left(f_t^\top \sum_{\tau=0}^{\infty} \left(F^\top\right)^{\tau+1} P f_{t+\tau+1} - \left(f_t^*\right)^\top \sum_{\tau=0}^{\infty} \left(F^\top\right)^{\tau+1} P \left(f_{t+\tau+1}^*\right) \right) \\
& + 2 \sum_{t=0}^{\infty} \left(\sum_{\tau=t}^{\infty} \left(F^\top\right)^{\tau} P f_{t+\tau}^* \right) B H^{-1} B^\top \left(\sum_{\tau=t}^{\infty} \left(F^\top\right)^{\tau} P (f_{t+\tau}^* - f_{t+\tau}) \right)
\end{aligned} \tag{3.13}$$

where $H \coloneqq R + B^\top P B$ and $F \coloneqq A - BK$. For any $t \geq 0$, we write $f_t \coloneqq f_t(x_t, u_t)$ and $f_t^* \coloneqq f_t(x_t^*, u_t^*)$ where $(x_t : t \geq 0)$ denotes a trajectory of states generated by the policy π with actions $(u_t^* : t \geq 0)$ and $(x_t^* : t \geq 0)$ denotes an optimal trajectory of states generated by optimal actions $(u_t^* : t \geq 0)$.

Proof. Note that with optimal trajectories of states and actions fixed, the optimal controller π^* induces the same cost OPT for both the non-linear system in (3.3)

and the auxiliary linear system. Moreover, the linear controller defined in (3.12) induces a cost OPT' that is smaller than OPT when running both in the auxiliary linear system since the constructed linear policy is optimal. Therefore, according to Lemma 10, $\mathsf{ALG} - \mathsf{OPT} \leq \mathsf{ALG} - \mathsf{OPT}'$ and applying Lemma 10 with $v_t = f_t(x_t, u_t)$ and $w_t = f_t(x_t^*, u_t^*)$ for all $t \geq 0$, (3.13) is obtained.

□

3.D Proof of Theorem 3.3.1

Proof. Fix an arbitrary controller K_1 with a closed-loop system matrix $F_1 \coloneqq A - BK_1 \neq 0$. We first consider the case when F_1 is not a diagonal matrix, i.e., $A - BK_1$ has at least one non-zero off-diagonal entry. Consider the following closed-loop system matrix for the second controller K_2:

$$F_2 \coloneqq \begin{pmatrix} \beta & & \cdots & & \\ & \beta & & 0 & \\ & & \ddots & & \\ & -\frac{1-\lambda}{\lambda}\overline{L} & & \beta & \\ & & & & \beta \end{pmatrix} + S\left(F_1\right) \tag{3.14}$$

where $0 < \beta < 1$ is a value of the diagonal entry; $\overline{L}$ is the lower triangular part of the closed-loop system matrix F_1. The matrix $S(F_1)$ is a singleton matrix that depends on F_1, whose only non-zero entry $S_{i+k,i}$ corresponding to the first non-zero off-diagonal entry $(i, i+k)$ in the upper triangular part of F_1 searched according to the order $i = 1, \ldots, n$ and $k = 1, \ldots, n$ with i increases first and then k. Such a non-zero entry always exists because in this case F_1 is not a diagonal matrix. If the non-zero off-diagonal entry appears to be in the lower triangular part, we can simply transpose F_2 so without loss of generality we assume it is in the upper triangular part of F_1. Since F_1 is a lower triangular matrix, all of its eigenvalues equal to $0 < \beta < 1$, implying that the linear controller K_2 is stabilizing. Then, based on the construction of F_1 in (3.14), the linearly combined controller $K \coloneqq \lambda K_2 + (1-\lambda)K_1$ has a closed-loop system matrix F which is upper-triangular, whose determinant satisfies

$$\begin{aligned} \det(F) =& \det\left(\lambda F_2 + (1-\lambda)F_1\right) \\ =& (-1)^{2i+k} \det(F') && (3.15) \\ =& (-1)^{2i+k} \times (-1)^{k+1}(1-\lambda)\lambda^{n-1} S_{i+k,i}(F_1)_{i,i+k}\beta^{n-2} && (3.16) \\ =& -S_{i+k,i}(1-\lambda)\lambda^{n-1}\beta^{n-2} && (3.17) \end{aligned}$$

where the term $(-1)^{2i+k}$ in (3.15) comes from the computation of the determinant of F and F' is a sub-matrix of F by eliminating the $i+k$-th row and i-th column. The term $(-1)^{k+1}$ in (3.16) appears because F' is a permutation of an upper triangular matrix with $n-2$ diagonal entries being β's and one remaining entry being $(F_1)_{i,i+k}$ since the entries $S_{j,j+k}$ are zeros all $j < i$; otherwise another non-zero entry $S_{i+k,i}$ would be chosen according to our search order.

Continuing from (3.17), since $S_{i+k,i}$ can be selected arbitrarily, setting $S_{i+k,i} = \frac{-2^n\beta(\lambda\beta)^{-n+1}}{1-\lambda}$ gives

$$2^n \leq \det(F) \leq |\rho(F)|^n,$$

implying that the spectral radius $\rho(F) \geq 2$. Therefore $K = \lambda K_2 + (1-\lambda)K_1$ is an unstable controller. It remains to prove the theorem in the case when F_1 is a diagonal matrix. In the following lemma, we consider the case when $n = 2$, and extend it to the general case.

Lemma 12. *For any $\lambda \in (0,1)$ and any diagonal matrix $F_1 \in \mathbb{R}^{2\times 2}$ with a spectral radius $\rho(F_1) < 1$ $F_1 \neq \gamma I$ for any $\gamma \in \mathbb{R}$, there exists a matrix $F_2 \in \mathbb{R}^{2\times 2}$ such that $\rho(F_2) < 1$ and $\rho(\lambda F_1 + (1-\lambda)F_2) > 1$.*

Proof of Lemma 12. Suppose $n = 2$ and

$$F_1 = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

is a diagonal matrix and without loss of generality assume that $a > 0$ and $a > b$. Let

$$F_2 = \frac{4}{\lambda(1-\lambda)(a-b)} \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}. \tag{3.18}$$

Notice that $\rho(F_2) = 0 < 1$ satisfies the constraint. Then

$$\begin{aligned} \rho\left(\lambda F_1 + (1-\lambda)F_2\right) =& \rho(\lambda b I + \lambda \mathrm{Diag}(a-b,0) + (1-\lambda)F_2) \\ =& \lambda b + \rho(\lambda \mathrm{Diag}(a-b,0) + (1-\lambda)F_2) \end{aligned} \tag{3.19}$$

where we have used the assumption that $a > 0$ and $a > b$ to derive the equality in (3.19) and the notion $\mathrm{Diag}(a-b,0)$ denotes a diagonal matrix whose diagonal entries are $a-b$ and 0, respectively. The eigenvalues of $\lambda\mathrm{Diag}(a-b,0) + (1-\lambda)F_2$ are

$$\frac{\lambda(a-b) \pm \sqrt{(\lambda(a-b) + \frac{8}{\lambda(a-b)})^2 - 4(\frac{4}{\lambda(a-b)})^2}}{2} = \frac{\lambda(a-b) \pm \sqrt{(\lambda(a-b))^2 + 16}}{2}.$$

Since $a - b > 0$, the spectral radius of $\lambda F_1 + (1-\lambda)F_2$ satisfies

$$\rho\left(\lambda F_1 + (1-\lambda)F_2\right) = \lambda b + \frac{\lambda(a-b) + \sqrt{(\lambda(a-b))^2 + 16}}{2} > -1 + \frac{\sqrt{16}}{2} = 1.$$

□

Applying Lemma 12, when F_1 is an $n \times n$ matrix with $n > 2$, we can always create an $n \times n$ matrix F_2 whose first two columns and rows form a sub-matrix that is the same as (3.18) and the remaining entries are zeros. Therefore the spectral radius of the convex combination $\lambda F_1 + (1-\lambda)F_2$ is greater than one. This completes the proof. □

3.E Stability Analysis

Proof Outline

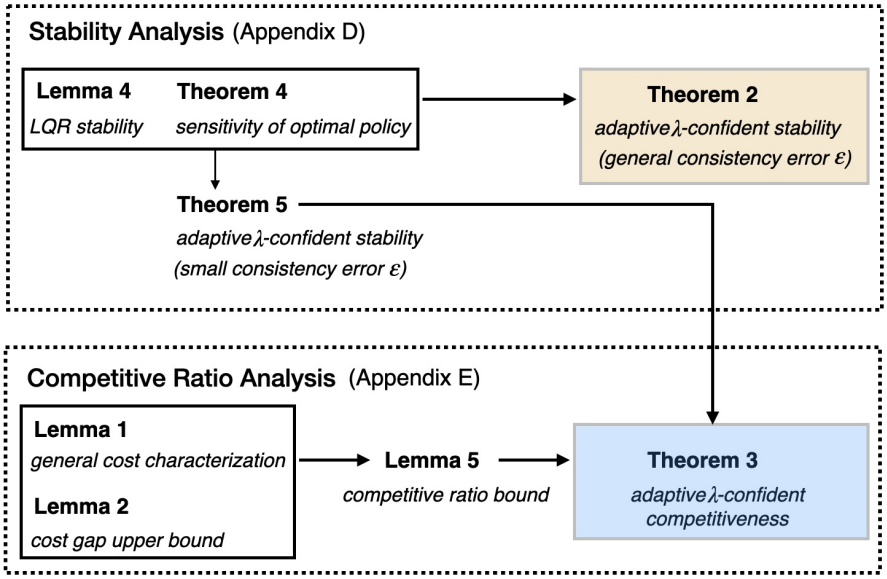


Figure 3.E.1: Outline of proofs of Theorem 3.4.1 and 3.4.2 with a stability analysis presented in Appendix 3.E and a competitive ratio analysis presented in Appendix 3.F. Arrows denote implications.

In the sequel, we present the proofs of Theorem 3.4.1 and 3.4.2. An outline of the proof structure is provided in Figure 3.E.1. The proof of our main results contains two parts–the *stability analysis* and *competitive ratio analysis*. First, we prove that Algorithm 4 guarantees a stabilizing policy, regardless of the prediction error ε (Theorem 3.4.1). Second, in our competitive ratio analysis, we provide a competitive ratio bound in terms of ε and λ. We show in Lemma 14 that the competitive ratio bound is bounded if the adaptive policy is exponentially stabilizing and has

a decay ratio that scales up with C_ℓ, which holds assuming the prediction error ε for a black-box model-free policy is small enough, as shown in Theorem 3.E.2. Theorem 3.E.2 is proven based on a sensitivity analysis of an optimal policy π^* in Theorem 13.

We first analyze the model-based policy $\widehat{\pi}(x) = -Kx$ where $K \coloneqq (R+B^\top PB)^{-1}B^\top PA$ and P is a unique solution of the Riccati equation in (3.4).

Lemma 13. *Suppose the Lipschitz constant C_ℓ, K and the closed-loop matrix $F \coloneqq A - BK$ satisfy $\rho + C_F C_\ell(1 + \|K\|) < 1$ where $\left\|F^t\right\| \le C_F\rho^t$ for any $t \ge 0$. Then the model-based policy $\widehat{\pi}(x) = -Kx$ exponentially stabilizes the system such that $\|x_t\| \le C_F\left(\rho + C_F\overline{C}\right)^t \|x_0\|$ for any $t \ge 0$.*

Proof. Let $u_t = \widehat{\pi}(x_t) = -Kx_t$ for all $t \ge 0$ and let $F \coloneqq A - BK$. It follows that

$$x_{t+1} = Ax_t + Bu_t + f_t(x_t, u_t) = (A - BK)x_t + f_t(x_t, -Kx_t) = Fx_t + f_t(x_t, -Kx_t). \tag{3.20}$$

Rewriting (3.20) recursively, for any $t \ge 0$,

$$x_t = F^t x_0 + \sum_{\tau=0}^{t-1} F^{t-1-\tau} f_\tau(x_\tau, -Kx_\tau). \tag{3.21}$$

Since $(f_t : t \ge 0)$ are Lipschitz continuous with a constant C_ℓ (Assumption 1), we have

$$\begin{aligned}
\|x_t\| &\le \left\|F^t x_0\right\| + \left\|\sum_{\tau=0}^{t-1} F^{t-1-\tau} f_\tau(x_\tau, -Kx_\tau)\right\| \\
&\le \left\|F^t x_0\right\| + \sum_{\tau=0}^{t-1} \left\|F^{t-1-\tau} f_\tau(x_\tau, -Kx_\tau)\right\| \\
&\le C_F\rho^t \underbrace{\left(\|x_0\| + C_\ell(1 + \|K\|)\sum_{\tau=0}^{t-1} \rho^{-1-\tau}\|x_\tau\|\right)}_{\coloneqq S_t}
\end{aligned} \tag{3.22}$$

where $C_F > 1$ is a constant such that $\left\|F^t\right\| \le C_F\rho^t$ for any $t \ge 0$. Denote by $\overline{C} \coloneqq C_\ell(1 + \|K\|)$. Then, using (3.22),

$$S_t = S_{t-1} + \overline{C}\rho^{-t}\|x_{t-1}\| \le S_{t-1} + \frac{C_F\overline{C}}{\rho} S_{t-1} = \left(1 + \frac{C_F\overline{C}}{\rho}\right) S_{t-1}.$$

Therefore, noting that $S_1 = \left(1 + \overline{C}/\rho\right) \|x_0\|$, recursively we obtain

$$S_t \leq \left(1 + \frac{C_F \overline{C}}{\rho}\right)^{t-1} \left(1 + \frac{\overline{C}}{\rho}\right) \|x_0\|,$$

which implies

$$\begin{aligned}\|x_t\| \leq C_F \rho^t S_t \leq & C_F \rho^t \left(1 + \frac{C_F \overline{C}}{\rho}\right)^{t-1} \left(1 + \frac{\overline{C}}{\rho}\right) \|x_0\| \\ = & C_F \left(\rho + C_F \overline{C}\right)^{t-1} \left(\rho + \overline{C}\right) \|x_0\| \\ \leq & C_F \left(\rho + C_F \overline{C}\right)^{t} \|x_0\|.\end{aligned}$$

□

Next, based on Lemma 13, we consider the stability of the convex-combined policy $\pi = \lambda\widehat{\pi} + (1-\lambda)\overline{\pi}$ where $\overline{\pi}$ is a model-free policy satisfying the *ε-consistency* in Definition 3.2.1 and $\overline{\pi}$ is a model-based policy.

Theorem 3.E.1. *Let π^* be an optimal policy and $\overline{\pi}(x) = -Kx$ be a linear model-based policy. It follows that for any $t \geq 0$, $\|\pi_t^*(x) - \overline{\pi}(x)\| \leq C_a^{\mathsf{sys}} C_\ell \|x\|$ for some constant $C_a^{\mathsf{sys}} > 0$ where C_ℓ is the Lipschitz constant defined in Assumption 1 and*

$$\begin{aligned}C_a^{\mathsf{sys}} := & 2\|R + B^\top P B\|^{-1} \Big(\|PF\| + (1 + \|K\|)\left(\|PB\| + \|P\|\right) \\ & + \frac{C_b^{\mathsf{sys}}}{2} \|B + I\| (1 + \|F\| + \|K\|) \Big), \\ C_b^{\mathsf{sys}} := & \frac{2{C_F}^2 \|P\| (\rho + \overline{C})\left(\rho + (1 + \|K\|)\right)}{1 - (\rho + \overline{C})^2} \sqrt{\frac{\|Q + K^\top R K\|}{\sigma}}.\end{aligned}$$

Proof. We use $\pi_t^*(x) = -Kx + h_t(x)$ to characterize an optimal policy at each time $t \geq 0$. Consider the following Bellman optimality equation:

$$V_t(x) = \min_u \{x^\top Q x + u_t^\top R u_t + V_{t+1}(Ax + Bu + f_t(x,u))\} \tag{3.23}$$

where $V_t : \mathbb{R}^n \to \mathbb{R}_+$ denotes the optimal value function. Using lemma 13,

$$\begin{aligned}V_t(x) \leq & x^\top (Q + K^\top R K) x + V_{t+1}(F^\top x + f_t(x,u)) \\ \leq & V_\infty(0) + \|Q + K^\top R K\| \sum_{t=0}^{\infty} {C_F}^2 (\rho + C_F \overline{C})^{2t} \|x\|^2 \\ = & V_\infty(0) + \frac{{C_F}^2 \|Q + K^\top R K\|}{1 - (\rho + C_F \overline{C})^2} \|x\|^2.\end{aligned} \tag{3.24}$$

Let $x_t^{(\tau)}$ be the $(t+\tau)$-th state when applying optimal control with $x_t = x$ and write $u_t^{(\tau)} = \pi_{t+\tau}^*(x_t^{(\tau)})$ for all $t \geq 0$. Rearranging the terms in (3.24), it follows that

$$\begin{aligned}\frac{{C_F}^2\|Q + K^\top RK\|}{1-(\rho + C_F\overline{C})^2}\|x\|^2 \geq & V_t(x) - V_{(\infty)}(0) \\ = & \sum_{\tau=t}^{\infty}\left(x_t^{(\tau-t)}\right)^\top Qx_t^{(\tau-t)} + \left(u_t^{(\tau-t)}\right)^\top Ru_t^{(\tau-t)} \\ \geq & \lambda_{\min}(Q)\sum_{\tau=0}^{\infty}\|x_t^{(\tau)}\|^2 + \lambda_{\min}(R)\sum_{\tau=0}^{\infty}\|u_t^{(\tau)}\|^2.\end{aligned}$$

Write $V_t(x) = x^\top Px + g_t(x)$ with P denoting the solution of the Riccatti equation in (3.4). Then

$$\begin{aligned}& x^\top Px + g_t(x) \\ & = \min_u \Big[x^\top Qx + u^\top Ru + (Ax+Bu)^\top P(Ax+Bu) \\ & \quad + 2(Ax+Bu)^\top Pf_t(x,u) + f_t(x,u)^\top Pf_t(x,u) + g_{t+1}(Ax+Bu+f_t(x,u))\Big] \\ & = \min_u \Big[x^\top(Q+A^\top PA)x + u^\top(R+B^\top PB)u + 2u^\top B^\top PAx + 2(Ax+Bu)^\top Pf_t(x,u) \\ & \quad + f_t(x,u)^\top Pf_t 2(x,u) + g_{t+1}(Ax+Bu+f_t(x,u))\Big] \\ & = \min_u \Big[x^\top(Q+A^\top PA - A^\top PB(R+B^\top PB)^{-1}B^\top PA)x \\ & \quad + (u+(R+B^\top PB))^{-1}B^\top PAx)^\top(R+B^\top PB)(u+(R+B^\top PB)^{-1}B^\top PAx) \\ & \quad + 2(Ax+Bu)^\top Pf_t(x,u) + f_t(x,u)^\top Pf_t(x,u) + g_{t+1}(Ax+Bu+f_t(x,u))\Big].\end{aligned}$$

Since P is the solution of the DARE in (3.4), letting $u = -Kx + v$ and $F \coloneqq A - BK$,

$$\begin{aligned}g_t(x) = \min_v \Big[& v^\top(R+B^\top PB)v + 2x^\top F^\top Pf_t(x,-Kx+v) + 2v^\top B^\top Pf_t(x,-Kx+v) \\ & + f_t(x,-Kx+v)^\top Pf_t(x,-Kx+v) + g_{t+1}(Fx+Bv+f_t(x,-Kx+v))\Big].\end{aligned}$$

Denoting by v_t^* an optimal solution,

$$\begin{aligned}g_t(x) = & (v_t^*)^\top(R+B^\top PB)v_t^* + 2x^\top F^\top Pf_t(x,-Kx+v_t^*) \\ & + 2(v_t^*)^\top B^\top Pf_t(x,-Kx+v_t^*) \\ & + f_t(x,-Kx+v_t^*)^\top Pf_t(x,-Kx+v_t^*) + g_{t+1}\left(Fx+Bv_t^*+f_t(x,-Kx+v_t^*)\right).\end{aligned}$$

Denote by ∇g_t the Jacobian of g_t. We obtain

$$\begin{aligned}
\nabla g_t(x) = {} & 2(R + B^\top P B) v_t^* \nabla v_t^* \\
& + 2F^\top P f_t(x, -Kx + v_t^*) \\
& + 2x^\top F^\top P \nabla f_t(x, -Kx + v_t^*)[I, -K + \nabla v_t^*] \\
& + 2\nabla (v_t^*)^\top B^\top P f_t(x, -Kx + v_t^*) \\
& + 2(v_t^*)^\top B^\top P \nabla f_t(x, -Kx + v_t^*)[I, -K + \nabla v_t^*] \\
& + 2P f_t(x, -Kx + v_t^*) \nabla f_t(x, -Kx + v_t^*)[I, -K + \nabla v_t^*] \\
& + \nabla g_{t+1}(Fx + Bv_t^* + f_t(x, -Kx + v_t^*)) \\
& \quad \left(F + B\nabla v_t^* + \nabla f_t(x, -Kx + v_t^*)[I, -K + \nabla v_t^*]\right)
\end{aligned}$$

Noting that v_t^* is a minimizer, the Jacobian of g_t with respect to v takes zero at $v = v_t^*$:

$$\begin{aligned}
& 2(R + B^\top P B) v + 2x^\top F^\top P \nabla f_t(x, -Kx + v)[0, I] + 2B^\top P f_t(x, -Kx + v) \\
& + 2v^\top B^\top P \nabla f_t(x, -Kx + v)[0, I] + 2P f_t(x, -Kx + v) \nabla f_t(x, -Kx + v)[0, I] \\
& + (B + \nabla f_t(x, -Kx + v)[0, I])^\top \nabla g_{t+1}(Fx + Bv + f_t(x, -Kx + v))|_{v = v_t^*} = 0
\end{aligned} \tag{3.25}$$

Substituting above into the Jacobian of g_t, we get

$$\begin{aligned}
\nabla g_t(x) = {} & 2F^\top P f_t(x, -Kx + v_t^*) \\
& + 2x^\top F^\top P \nabla f_t(x, -Kx + v_t^*)[I, -K] \\
& + 2(v_t^*)^\top B^\top P \nabla f_t(x, -Kx + v_t^*)[I, -K] \\
& + 2P f_t(x, -Kx + v_t^*) \nabla f_t(x, -Kx + v_t^*)[I, -K] \\
& + \nabla g_{t+1}(Fx + Bv_t^* + f_t(x, -Kx + v_t^*))(F + \nabla f_t(x, -Kx + v_t^*)[I, -K]) \\
= {} & 2F^\top P f_t(x, -Kx + v_t^*) + 2(\nabla f_t(x, -Kx + v_t^*)[I, K])^\top P x \\
& + (F + \nabla f_t(x, -Kx + v_t^*)[I, -K])^\top \nabla g_{t+1}(x_t^{(1)}),
\end{aligned}$$

which implies that

$$\begin{aligned}
\nabla g_t(x) = {} & \prod_{\tau=t}^{\infty} (F + \nabla f_\tau(x_t^{(\tau-t)}, u_t^{(\tau-t)})[I, -K])^\top g_\infty(0) \\
& + \sum_{\tau=t}^{\infty} \prod_{k=t}^{\tau} (F + \nabla f_k(x_t^{(k-t)}, u_t^{(k-t)})[I, -K])^\top 2F^\top P f_\tau(x_t^{(\tau-t)}, u_t^{(\tau-t)}) \\
& + \sum_{\tau=t}^{\infty} \prod_{k=t}^{\tau} (F + \nabla f_k(x_t^{(k-t)}, u_t^{(k-t)})[I, -K])^\top \\
& \quad 2(\nabla f_\tau(x_\tau^{(\tau-t)}, u_\tau^{(\tau-t)})[I, -K])^\top P_{\tau+1}^*.
\end{aligned} \tag{3.26}$$

Note that for any sequence of pairs $\left((x_t^{(\tau)}, u_t^{(\tau)}) : \tau \geq 0\right)$,

$$\begin{aligned}
&\left\| \prod_{k=t}^{\tau} F + \nabla f_k(x_t^{(k-t)}, u_t^{(k-t)}[I, -K] \right\| \\
&\leq \sum_{k=t}^{\tau+1} \left\| F^{\tau+1-k} \sum_{\mathsf{S} \subseteq \{t,\ldots,\tau\}, |\mathsf{S}|=k-t} \prod_{s \in \mathsf{S}} \nabla f_s(x_t^{(s-t)}, u_t^{(s-t)})[I, -K] \right\| \\
&\leq \sum_{k=t}^{\tau+1} C_F \rho^{\tau+1-k} \sum_{\mathsf{S} \subseteq \{t,\ldots,\tau\}, |\mathsf{S}|=k-t} \prod_{s \in \mathsf{S}} \left\| \nabla f_s(x_t^{(s-t)}, u_t^{(s-t)})[I, -K] \right\|.
\end{aligned}$$

Since the residual functions are Lipschitz continuous as in Assumption 1, their Jacobians satisfy $\|\nabla f_s(x_t^{(s-t)}, u_t^{(s-t)})\| \leq C_\ell$ for any $x_t^{(s-t)}$ and $u_t^{(s-t)}$. Letting $\overline{C} := C_\ell(1 + \|K\|)$, we get

$$\begin{aligned}
&\left\| \prod_{k=t}^{\tau} F + \nabla f_k(x_t^{(k-t)}, u_t^{(k-t)})[I, -K] \right\| \\
&\leq \sum_{k=t}^{\tau+1} C_F \rho^{\tau+1-k} \sum_{\mathsf{S} \subseteq \{t,\ldots,\tau\}, |\mathsf{S}|=k-t} \prod_{s \in \mathsf{S}} \overline{C} = C_F(\rho + \overline{C})^{\tau+1-t}. \qquad (3.27)
\end{aligned}$$

Therefore, using (3.27),

$$\begin{aligned}
&\left\| \nabla f_\tau(x_t^{(\tau+1t-)}, u_t^{(\tau+1-t)})[I, -K] \prod_{k=t}^{\tau} F + \nabla f_k(x_t^{(k-t)}, u_t^{(k-t)})[I, -K] \right\| \\
&\leq \left\| \nabla f_\tau(x_t^{(\tau+1t-)}, u_t^{(\tau+1-t)})[I, -K] \right\| \left\| \prod_{k=t}^{\tau} F + \nabla f_k(x_t^{(k-t)}, u_t^{(k-t)})[I, -K] \right\| \\
&\leq C_F \overline{C}(\rho + \overline{C})^{\tau+1-t}. \qquad (3.28)
\end{aligned}$$

Combing (3.27) and (3.28) with (3.26),

$$\begin{aligned}
\|\nabla g_t(x)\| \leq &\left\| \prod_{\tau=t}^{\infty} (F + \nabla f_\tau(x_t^{(\tau-t)}, u_t^{(\tau-t)})[I, -K])^\top g_\infty(0) \right\| \\
&+ \underbrace{\left\| \sum_{\tau=t}^{\infty} \prod_{k=t}^{\tau} (F + \nabla f_k(x_t^{(k-t)}, u_t^{(k-t)})[I, -K])^\top 2F^\top P f_\tau(x_t^{(\tau-t)}, u_t^{(\tau-t)}) \right\|}_{(a)} \\
&+ \underbrace{\left\| \sum_{\tau=t}^{\infty} \prod_{k=t}^{\tau} (F + \nabla f_k(x_t^{(k-t)}, u_t^{(k-t)})[I, -K])^\top 2(\nabla f_\tau(x_t^{(\tau-t)}, u_t^{(\tau-t)})[I, -K])^\top P x_t^{(\tau+1-t)} \right\|}_{(b)}.
\end{aligned}$$

Since $g_\infty(0) = 0$, the first term in the inequality above is 0. The second term satisfies

$$
\begin{aligned}
(a) \leq & 2C_F C_\ell \|P\| \rho \sum_{\tau=t}^{\infty} (\rho + \overline{C})^{\tau+1-t} \|(x_t^{(\tau-t)}, u_t^{(\tau-t)})\| \\
= & 2C_F C_\ell \|P\| \rho \sum_{\tau=0}^{\infty} (\rho + \overline{C})^{\tau+1} \|x_t^{(\tau)}, u_t^{(\tau)}\| \\
\leq & 2C_F C_\ell \|P\| \rho \sqrt{\sum_{\tau=1}^{\infty} (\rho + \overline{C})^{2\tau}} \sqrt{\sum_{\tau=0}^{\infty} \left\|x_t^{(\tau)}\right\|^2 + \left\|u_t^{(\tau)}\right\|^2} \qquad (3.29) \\
\leq & 2C_F C_\ell \|P\| \rho \sqrt{\frac{(\rho + \overline{C})^2}{1 - (\rho + \overline{C})^2}} \sqrt{\frac{{C_F}^2 \|Q + K^\top R K\|}{\min\{\lambda_{\min}(Q), \lambda_{\min}(R)\} (1 - (\rho + \overline{C})^2)} \|x\|^2} \\
= & \frac{2{C_F}^2 \|P\| \rho C_\ell (\rho + \overline{C})}{1 - (\rho + \overline{C})^2} \sqrt{\frac{\|Q + K^\top R K\|}{\sigma}} \|x\| \qquad (3.30)
\end{aligned}
$$

where we have used the Cauchy to derive (3.29); $\lambda_{\min}(Q)$ and $\lambda_{\min}(R)$ are smallest eigenvalues of the matrices Q and R, respectively, and (3.30) follows from Assumption 2. Similarly, the third term satisfies

$$
\begin{aligned}
(b) \leq & 2C_F \overline{C} \|P\| \sum_{\tau=t}^{\infty} (\rho + \overline{C})^{\tau+1-t} \left\|x_t^{(\tau+1-t)}\right\| \\
\leq & 2C_F \overline{C} \|P\| \sqrt{\sum_{\tau=1}^{\infty} (\rho + \overline{C})^{2\tau}} \sqrt{\sum_{\tau=1}^{\infty} \left\|x_t^{(\tau)}\right\|^2} \\
\leq & 2C_F \overline{C} \|P\| \sqrt{\frac{(\rho + \overline{C})^2}{1 - (\rho + \overline{C})^2}} \sqrt{\frac{{C_F}^2 \|Q + K^\top R K\|}{\lambda_{\min}(Q)(1 - (\rho + \overline{C})^2} \|x\|^2} \\
\leq & \frac{2{C_F}^2 \|P\| \overline{C} (\rho + \overline{C})}{1 - (\rho + \overline{C})^2} \sqrt{\frac{\|Q + K^\top R K\|}{\sigma}} \|x\|. \qquad (3.31)
\end{aligned}
$$

Putting (3.30) and (3.31) together, we conclude that

$$
\|\nabla g_t(x)\| \leq \frac{2{C_F}^2 \|P\| (\rho + \overline{C}) \left(\rho C_\ell + \overline{C}\right)}{1 - (\rho + \overline{C})^2} \sqrt{\frac{\|Q + K^\top R K\|}{\sigma}} \|x\| =: C_\nabla C_\ell \|x\|. \qquad (3.32)
$$

Rewriting (3.25) as

$$
\begin{aligned}
-v_t^* = & (R + B^\top P B)^{-1} x^\top F^\top P \nabla f_t(x, -Kx + v_t^*)[0, I] \\
& + (R + B^\top P B)^{-1} B^\top P f_t(x, -Kx + v_t^*) \\
& + (R + B^\top P B)^{-1} (v_t^*)^\top B^\top P \nabla f_t(x, -Kx + v_t^*)[0, I] \\
& + (R + B^\top P B)^{-1} P f_t(x, -Kx + v_t^*) \nabla f_t(x, -Kx + v_t^*)[0, I] \\
& + \frac{1}{2} (R + B^\top P B)^{-1} (B + \nabla f_t(x, -Kx + v_t^*)[0, I])^\top \nabla g_t(Fx + Bv + f_t(x, -Kx + v_t^*))
\end{aligned}
$$

and taking the Euclidean norm on both sides, we obtain

$$\begin{aligned}
\|v_t^*\| \le & \left\|(R+B^\top PB)^{-1}x^\top F^\top P\nabla f_t(x,-Kx+v_t^*)[0,I]\right\| \\
& + \left\|(R+B^\top PB)^{-1}B^\top P f_t(x,-Kx+v_t^*)\right\| \\
& + \left\|(R+B^\top PB)^{-1}(v_t^*)^\top B^\top P\nabla f_t(x,-Kx+v_t^*)[0,I]\right\| \\
& + \left\|(R+B^\top PB)^{-1}P f_t(x,-Kx+v_t^*)\nabla f_t(x,-Kx+v_t^*)[0,I]\right\| \\
& + \left\|\frac{1}{2}(R+B^\top PB)^{-1}(B+\nabla f_t(x,-Kx+v_t^*)[0,I])^\top \right. \\
& \qquad \left. \nabla g_t\left(Fx+Bv+f_t(x,-Kx+v_t^*)\right)\right\| \\
\le & \left\|R+B^\top PB\right\|^{-1}\Big(C_\ell\|PF\|\|x\| + \overline{C}\left\|B^\top P\right\|\|x\| \\
& + 2C_\ell\left\|B^\top P\right\|\|v_t^*\| + \overline{C}\|P\|\|x\| + C_\ell\|P\|\|v_t^*\| \\
& + \frac{1}{2}C_\nabla C_\ell\|B+C_\ell I\|(\|F\|\|x\| + \|B\|\|v_t^*\| + \overline{C}\|x\| + C_\ell\|x\|)\Big). \qquad (3.33)
\end{aligned}$$

Finally, assuming $C_\ell \le \min\left\{1, \frac{\|R+B^\top PB\|}{4\|B^\top P\|+2\|P\|+C_\nabla(\|B\|+1)\|B\|}\right\}$, (3.33) yields

$$\begin{aligned}
&\left\|\pi_t^*(x)-\overline{\pi}(x)\right\| \le C_\ell\|x\|\times \\
&\underbrace{\left(2\|R+B^\top PB\|^{-1}\left((\|PF\|+(1+\|K\|)\left(\|PB\|+\|P\|\right)+\frac{C_\nabla}{2}\|B+I\|(2+\|F\|+\|K\|)\right)\right)}_{=:C_a^{\mathsf{sys}}}
\end{aligned}$$

where the constant C_∇ is defined as

$$C_\nabla := \frac{2{C_F}^2\|P\|(\rho+\overline{C})\left(\rho+(1+\|K\|)\right)}{1-(\rho+\overline{C})^2}\sqrt{\frac{\|Q+K^\top RK\|}{\sigma}}.$$

□

Theorem 3.E.2. *Let $\gamma := (\rho + C_F C_\ell(1+\|K\|))$. Suppose the black-box policy $\widehat{\pi}$ is ε-consistent with the consistency constant ε satisfying $\varepsilon < \frac{1/C_F - C_a^{\mathsf{sys}}C_\ell}{C_\ell+\|B\|}$. Suppose the Lipschitz constant C_ℓ satisfies $C_\ell < \min\left\{1, 1/(C_F C_a^{\mathsf{sys}}), C_c^{\mathsf{sys}}, (1-\rho)/(C_F(1+\|K\|))\right\}$. Let $(x_t : t \ge 0)$ denote a trajectory of states generated by the adaptive λ-confident policy $\pi_t = \lambda_t\widehat{\pi} + (1-\lambda_t)\overline{\pi}$ (Algorithm 4). Then it follows that π_t is an exponentially stabilizing policy such that $\|x_t\| \le \frac{\gamma^t-\mu^t}{1-\mu\gamma^{-1}}\left(C_F+\mu\gamma^{-1}\right)\|x_0\|$ for any $t \ge 0$ where $\mu := C_F\left(\varepsilon\left(C_\ell+\|B\|\right)+C_a^{\mathsf{sys}}C_\ell\right)$.*

Proof. We first introduce a new symbol $x_t^{(\tau)}$, which is the t-th state of a trajectory generated by the combined policy $\pi_t(x) = \lambda_t\widehat{\pi}(x) + (1-\lambda_t)\overline{\pi}(x)$ for the first τ steps and then switch to the model-based policy $\overline{\pi}$ for the remaining steps. Let $(\overline{x}_t : t \ge 0)$

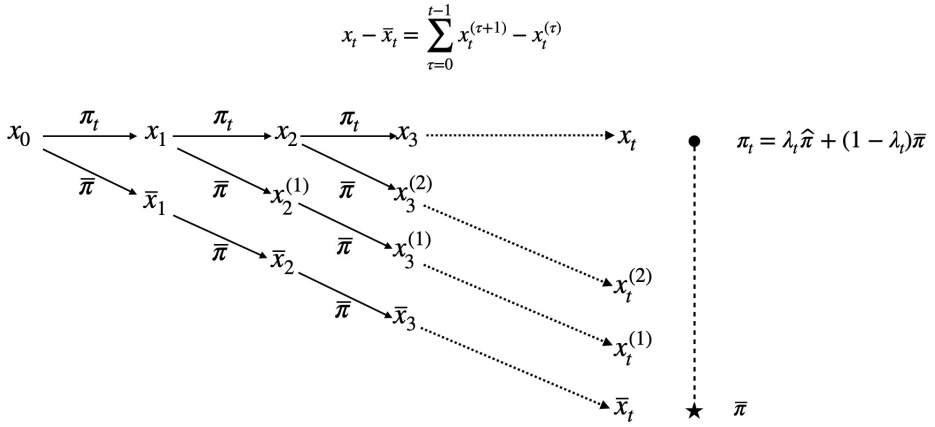


Figure 3.E.2: Telescoping sum of $x_t - x_t^*$.

be a trajectory of states generated by a model-based policy $\overline{\pi}$. As illustrated in Figure 3.1.1, for any $t \geq 0$,

$$x_t - \overline{x}_t = \sum_{\tau=0}^{t-1} x_t^{(\tau+1)} - x_t^{(\tau)}. \tag{3.34}$$

Let $(x_t : t \geq 0)$ and $(x'_t : t \geq 0)$ denote the trajectories of states at time t generated by the model-based policy $\widehat{\pi}(x) = -Kx$ when the initial states are x_0 and x'_0, respectively. Then (3.21) leads to

$$x_t - x'_t = F^t(x_0 - x'_0) + \sum_{\tau=0}^{t-1} F^{t-1-\tau} \left(f_\tau(x_\tau, -Kx_\tau) - f_\tau(x'_\tau, -Kx'_\tau) \right),$$

yielding

$$\|x_t - x'_t\| \leq C_F \rho^t \left(\left\|x_0 - x'_0\right\| + C_\ell (1 + \|K\|) \sum_{\tau=0}^{t-1} \rho^{-1-\tau} \left\|x_\tau - x'_\tau\right\| \right)$$

where we have used the Lipschitz continuity of $(f_t : t \geq 0)$ and Assumption 2 so that $\left\|F^t\right\| \leq C_F \rho^t$ for any $t \geq 0$. The same argument as in Lemma 13 gives that for any $t \geq 0$,

$$\|x_t - x'_t\| \leq C_F \left(\rho + C_F \overline{C} \right)^t \|x_0 - x'_0\|$$

where $\overline{C} \coloneqq C_\ell(1 + \|K\|)$. Continuing from (3.34), $x_t - x_t^*$ can be represented by a telescoping sum (illustrated in Figure 3.E.2):

$$\|x_t - \overline{x}_t\| = \left\| \sum_{\tau=0}^{t-1} x_t^{(\tau+1)} - x_t^{(\tau)} \right\| \leq \sum_{\tau=0}^{t-1} \left\| x_t^{(\tau+1)} - x_t^{(\tau)} \right\| \leq \sum_{\tau=0}^{t-1} C_F \gamma^{t-\tau-1} \left\| x_{\tau+1}^{(\tau+1)} - x_{\tau+1}^{(\tau)} \right\|.$$

For any $t \geq 0$,

$$\begin{aligned}
&x_t^{(t)} - x_t^{(t-1)} = x_t - x_t^{(t-1)} \\
=&(x_t - \lambda_t x_t') + \left(\lambda_t x_t' - x_t^{(t-1)}\right) \\
=&\lambda_t \left(\widehat{x}_t - x_t'\right) + (1-\lambda_t)\left(\overline{x}_t - x_t^{(t-1)}\right) + \lambda_t \left(x_t' - x_t^{(t-1)}\right) \\
=&\lambda_t \left(\widehat{x}_t - x_t'\right) + \lambda_t \left(x_t' - x_t^{(t-1)}\right)
\end{aligned}$$

where x_t', $\widehat{x}_t$ and $\overline{x}_t$ are the states generated by running an optimal policy π^*, a model-free policy $\widehat{\pi}$ and a model-based policy $\overline{\pi}$, respectively, for one step with the same initial state x_{t-1}. Note that $x_t = \lambda_t \widehat{x}_t + (1-\lambda_t)\overline{x}_t$ and $\overline{x}_t = x_t^{(t-1)}$. Therefore,

$$\lambda_t \left(\widehat{x}_t - x_t'\right) = \lambda_t \big(\underbrace{B(\widehat{\pi}(x_{t-1}) - \pi^*(x_{t-1}))}_{:=(a)} + \underbrace{f_{t-1}(x_{t-1}, \widehat{\pi}(x_{t-1})) - f_{t-1}(x_{t-1}, \pi^*(x_{t-1}))}_{:=(b)} \big).$$

For (a), we obtain the following bound:

$$\|(a)\| \leq \|B\| \|\widehat{\pi}(x_{t-1}) - \pi^*(x_{t-1})\| \leq \varepsilon \|B\| \|x_{t-1}\| \tag{3.35}$$

and (3.35) holds since the model-free policy $\widehat{\pi}$ is ε-consistent (Definition 3.2.1). Similarly, for (b), since the functions $(f_t : t \geq 0)$ are Lipschitz continuous (Assumption 1),

$$\|(b)\| \leq \varepsilon C_\ell \|x_{t-1}\|. \tag{3.36}$$

Applying Theorem 3.E.1, $\left\|x_t' - x_t^{(t-1)}\right\| \leq C_a^{\mathsf{sys}} C_\ell \|x_{t-1}\|$. Combining (3.35) and (3.36) and applying Lemma 13, $\|x_t - \overline{x}_t\| \leq \mu \sum_{\tau=0}^{t-1} \gamma^{t-\tau-1} \|x_\tau\|$ where

$$\mu \coloneqq C_F \left(\varepsilon \left(C_\ell + \|B\|\right) + C_a^{\mathsf{sys}} C_\ell\right),$$

therefore,

$$\frac{\|x_t\|}{\gamma^t} \leq \frac{1 - (\mu\gamma^{-1})^t}{1 - \mu\gamma^{-1}} \left(C_F + \mu\gamma^{-1}\right) \|x_0\|.$$

Hence, if $\mu < 1$, $\|x_t\| \leq \frac{\gamma^t - \mu^t}{1-\mu\gamma^{-1}} \left(C_F + \mu\gamma^{-1}\right) \|x_0\|$, the linearly combined policy π_t is an exponentially stabilizing policy. □

Proof of Theorem 3.4.1

To show the stability results, noting that the policy is switched to the model-based policy after $t \geq t_0$. Since the Lipschitz constant C_ℓ satisfies $C_\ell < \frac{1-\rho}{C_F(1+\|K\|)}$

where $\left\|F^t\right\| \le C_F\rho^t$ for any $t \ge 0$, then the model-based policy $\overline{\pi}$ is exponentially stable as shown in Lemma 13. Let $\mu := C_F\left(\varepsilon\left(C_\ell + \|B\|\right) + C_a^{\mathsf{sys}} C_\ell\right)$ and $\gamma := \rho + C_F C_\ell(1 + \|K\|) < 1$. Applying Theorem 3.E.2 and Lemma 13, for any $t \ge 0$,

$$\|x_t\| \le \frac{\mu^{t_0}}{\mu\gamma^{-1} - 1}\left(C_F + \mu\gamma^{-1}\right)\left(\rho + C_F C_\ell(1 + \|K\|)\right)^{t-t_0}\|x_0\|.$$

Since $\lambda_{t+1} < \lambda_t - \alpha$ for all $t \ge 0$ with some $\alpha > 0$, t_0 is finite and the adaptive λ-confident policy is an exponentially stabilizing policy.

3.F Competitive Ratio Analysis

Proof of Theorem 3.4.2

Before proceeding to the proof of Theorem 3.4.2, we prove the following lemma that will be useful.

Lemma 14. *Suppose $\varepsilon \le \lambda_{\min}(Q)/(2\|H\|)$. If the adaptive λ-confident policy $\pi_t = \lambda_t\widehat{\pi} + (1-\lambda_t)\overline{\pi}$ (Algorithm 4) is an exponentially stabilizing policy, then the competitive ratio of the linearly combined policy is $\mathsf{CR}(\varepsilon) = O((1-\lambda)\overline{\mathsf{CR}}_{\text{model}}) + O\left(1/\left(1 - \frac{2\|H\|}{\sigma}\varepsilon\right)\right) + O(C_\ell\|x_0\|)$.*

Proof of Lemma 14. Let $H := R + B^\top P B$. Fix any sequence of residual functions $(f_t : t \ge 0)$. Denote by $f_t := f_t(x_t, u_t)$, $f_t^* := f_t(x_t^*, u_t^*)$ and x_t^* and u_t^* the offline optimal state and action at time t. With $u_t = \lambda_t\widehat{u}_t + (1-\lambda_t)\overline{u}_t$ at each time t, Lemma 11 implies that the *dynamic regret* can be bounded by

$$\begin{aligned}
\mathsf{DynamicRegret} := \mathsf{ALG} - \mathsf{OPT} \le{}& \sum_{t=0}^{\infty}\eta_t^\top H\eta_t + O(1)\\
&+ 2\sum_{t=0}^{\infty}\eta_t^\top B^\top\left(\sum_{\tau=t}^{\infty}\left(F^\top\right)^\tau P(f_{t+\tau} - f_{t+\tau}^*)\right)\\
&+ \sum_{t=0}^{\infty}\left(f_t^\top P f_t - \left(f_t^*\right)^\top P f_t^*\right) + 2x_0^\top\left(\sum_{t=0}^{\infty}\left(F^\top\right)^{t+1} P\left(f_t - f_t^*\right)\right)\\
&+ 2\sum_{t=0}^{\infty}\left(f_t^\top\sum_{\tau=0}^{\infty}\left(F^\top\right)^{\tau+1} P f_{t+\tau+1} - \left(f_t^*\right)^\top\sum_{\tau=0}^{\infty}\left(F^\top\right)^{\tau+1} P\left(f_{t+\tau+1}^*\right)\right)\\
&+ 2\sum_{t=0}^{\infty}\left(\sum_{\tau=t}^{\infty}\left(F^\top\right)^\tau P f_{t+\tau}^*\right) B H^{-1} B^\top\left(\sum_{\tau=t}^{\infty}\left(F^\top\right)^\tau P(f_{t+\tau}^* - f_{t+\tau})\right).
\end{aligned} \tag{3.37}$$

Consider the auxiliary linear policy defined in (3.12). Provided with any state $x \in \mathbb{R}^n$, the linearly combined policy π_t is given by

$$\pi_t(x) = \lambda_t\widehat{\pi}(x) + (1-\lambda_t)\overline{\pi}(x) = \pi'(x) + \lambda_t(\widehat{\pi}(x) - \pi'(x)) + (1-\lambda_t)\left(\overline{\pi}(x) - \pi'(x)\right),$$

implying $\eta_t = \lambda_t(\widehat{\pi}(x_t) - \pi'(x_t)) + (1-\lambda_t)\,(\overline{\pi}(x_t) - \pi'(x_t))$ in (3.37). Moreover, $\sum_{t=0}^{\infty} \eta_t^\top H \eta_t \leq \sum_{t=0}^{\infty} \|H\| \|\eta_t\|^2$, therefore, denoting by $\lambda \coloneqq \lim_{t\to\infty} \lambda_t$,

$$\sum_{t=0}^{\infty} \eta_t^\top H \eta_t \leq 2\|H\| \left(\sum_{t=0}^{\infty} \|\widehat{\pi}(x_t) - \pi'(x_t)\|^2 + (1-\lambda) \sum_{t=0}^{\infty} \|\overline{\pi}(x_t) - \pi'(x_t)\|^2 \right), \tag{3.38}$$

where in (3.38) we have used the the Cauchy–Schwarz inequality. Since the model-free policy $\widehat{\pi}$ is ε-consistent, it follows that

$$\begin{aligned} \|\widehat{\pi}(x_t) - \pi'(x_t)\|^2 &\leq 2 \left\|\pi_t^*(x_t) - \pi'(x_t)\right\|^2 + 2 \left\|\widehat{\pi}(x_t) - \pi_t^*(x_t)\right\|^2 \\ &\leq 2 \left\|\pi_t^*(x_t) - \pi'(x_t)\right\|^2 + 2\varepsilon \|x_t\|^2. \end{aligned}$$

Furthermore, since the cost OPT' induced by the auxiliary linear policy (3.12) is smaller than OPT,

$$\sum_{t=0}^{\infty} \left\|\pi_t^*(x_t) - \pi'(x_t)\right\|^2 \leq \frac{\mathsf{OPT} + \mathsf{OPT}'}{\lambda_{\min}(R)} \leq \frac{2\mathsf{OPT}}{\lambda_{\min}(R)} \tag{3.39}$$

where $\lambda_{\min}(R) > 0$ denotes the smallest eigenvalue of $R > 0$.

The linear quadratic regulator is $\overline{\pi}(x) = -Kx = -(R + B^\top P B)^{-1} B^\top P A x$ for a given state $x \in \mathbb{R}^n$, we have for all $t \geq 0$,

$$\|\overline{\pi}(x_t) - \pi'(x_t)\|^2 = \left\| \sum_{\tau=t}^{\infty} \left(F^\top\right)^{\tau-t} P f_\tau^* \right\|^2. \tag{3.40}$$

Plugging (3.39) and (3.40) into (3.38),

$$\sum_{t=0}^{\infty} \eta_t^\top H \eta_t \leq 2\|H\| \left(\left(\frac{2\mathsf{OPT}}{\lambda_{\min}(R)} + \varepsilon \sum_{t=0}^{\infty} \|x_t\|^2 \right) + \sum_{t=0}^{\infty} \left\| \sum_{\tau=t}^{\infty} \left(F^\top\right)^{\tau-t} P f_\tau^* \right\|^2 \right). \tag{3.41}$$

The algorithm cost ALG can be bounded by $\mathsf{ALG} \geq \sum_{t=0}^{\infty} x_t^\top Q x_t + u_t^\top R u_t \geq \sum_{t=0}^{\infty} \lambda_{\min}(Q) \|x_t\|^2$, therefore, (3.41) leads to

$$\sum_{t=0}^{\infty} \eta_t^\top H \eta_t \leq 2\|H\| \left(\left(\frac{2\mathsf{OPT}}{\lambda_{\min}(R)} + \varepsilon \frac{\mathsf{ALG}}{\lambda_{\min}(Q)} \right) + \sum_{t=0}^{\infty} \left\| \sum_{\tau=t}^{\infty} \left(F^\top\right)^{\tau-t} P f_\tau^* \right\|^2 \right). \tag{3.42}$$

Moreover, we have the following lemma holds.

Lemma 15. *The optimal cost* OPT *can be bounded from below by*

$$\mathsf{OPT} \geq \frac{D_0 (1-\rho)^2}{{C_F}^2 \|P\|^2} \sum_{t=0}^{\infty} \left\| \sum_{\tau=t}^{\infty} \left(F^\top\right)^{\tau-t} P f_\tau^* \right\|^2.$$

Proof of Lemma 15. the optimal cost can be bounded from below by

$$\begin{aligned}
\mathsf{OPT} &= \sum_{t=0}^{\infty} (x_t^*)^\top Q x_t^* + (u_t^*)^\top R u_t^* \\
&\geq \sum_{t=0}^{\infty} \lambda_{\min}(Q) \left\|x_t^*\right\|^2 + \lambda_{\min}(R) \|u_t^*\|^2 \qquad (3.43) \\
&\geq 2D_0 \sum_{t=0}^{\infty} \left(\|Ax_t^*\|^2 + \|Bu_t^*\|^2 \right) + \frac{1}{2} \sum_{t=0}^{\infty} \lambda_{\min}(Q) \|x_t^*\|^2 \\
&\geq D_0 \sum_{t=0}^{\infty} \left\|Ax_t^* + Bu_t^*\right\|^2 + \frac{1}{2} \sum_{t=0}^{\infty} \lambda_{\min}(Q) \|x_t^*\|^2.
\end{aligned}$$

Since $x_{t+1}^* = Ax_t^* + Bu_t^* + f_t^*$ for all $t \geq 0$,

$$\begin{aligned}
\mathsf{OPT} &\geq D_0 \sum_{t=0}^{\infty} \left\|x_{t+1}^* - f_t^*\right\|^2 + \frac{1}{2} \sum_{t=0}^{\infty} \lambda_{\min}(Q) \left\|x_t^*\right\|^2 \\
&\geq \frac{D_0}{2} \sum_{t=0}^{\infty} \left\|f_t^*\right\|^2 + \left(\frac{\lambda_{\min}(Q)}{2} - D_0 \right) \sum_{t=0}^{\infty} \left\|x_t^*\right\|^2 \qquad (3.44)
\end{aligned}$$

for some constant $D_0 \coloneqq \min\{\frac{\lambda_{\min}(R)}{\|B\|}, \frac{\lambda_{\min}(Q)}{2\|A\|}, \frac{\lambda_{\min}(Q)}{2}\} \geq \frac{\sigma}{\max\{2, \|A\|, \|B\|\}}$ (Assumption 2) that depends on known system parameters A, B, Q and R where in (3.43), $\lambda_{\min}(Q)$, $\lambda_{\min}(R)$ are the smallest eigenvalues of positive definite matrices Q, R, respectively. Let $\psi_t \coloneqq \sum_{\tau=t}^{\infty} (F^\top)^\tau P f_{t+\tau}^*$ for all $t \geq 0$. Note that $F = A - BK$ and we define $\rho \coloneqq (1 + \rho(F))/2 < 1$ where $\rho(F)$ denotes the spectral radius of F. From the Gelfand's formula, there exists a constant $C_F \geq 0$ such that $\|F^t\| \leq C_F \rho^t$ for all $t \geq 0$. Therefore,

$$\begin{aligned}
\sum_{t=0}^{\infty} \|\psi_t\|^2 \coloneqq \sum_{t=0}^{\infty} \left\| \sum_{\tau=t}^{\infty} \left(F^\top\right)^\tau P f_{t+\tau}^* \right\|^2 &\leq {C_F}^2 \|P\|^2 \sum_{t=0}^{\infty} \left(\sum_{\tau=t}^{\infty} \rho^\tau \left\|f_{t+\tau}^*\right\| \right)^2 \\
&= {C_F}^2 \|P\|^2 \sum_{t=0}^{\infty} \sum_{\tau=t}^{\infty} \sum_{\ell=0}^{\infty} \rho^\tau \rho^\ell \left\|f_{t+\tau}^*\right\| \left\|f_{t+\ell}^*\right\| \\
&\leq \frac{{C_F}^2}{2} \|P\|^2 \sum_{t=0}^{\infty} \sum_{\tau=t}^{\infty} \sum_{\ell=0}^{\infty} \rho^\tau \rho^\ell \left(\left\|f_{t+\tau}^*\right\|^2 + \left\|f_{t+\ell}^*\right\|^2 \right). \qquad (3.45)
\end{aligned}$$

Continuing from (3.45),

$$\begin{aligned}\sum_{t=0}^{\infty}\|\psi_t\|^2 \leq& \frac{{C_F}^2}{2}\|P\|^2\left(\sum_{\ell=0}^{\infty}\rho^\ell\right)\sum_{t=0}^{\infty}\sum_{\tau=t}^{\infty}\rho^\tau\left\|f^*_{t+\tau}\right\|^2\\ &+\frac{{C_F}^2}{2}\|P\|^2\left(\sum_{\tau=t}^{\infty}\rho^\tau\right)\sum_{t=0}^{\infty}\sum_{\ell=0}^{\infty}\rho^\ell\left\|f^*_{t+\ell}\right\|^2\\ \leq& \frac{{C_F}^2}{1-\rho}\|P\|^2\sum_{t=0}^{\infty}\sum_{\tau=t}^{\infty}\rho^\tau\left\|f^*_{t+\tau}\right\|^2 \leq \frac{{C_F}^2}{1-\rho}\|P\|^2\sum_{t=0}^{\infty}\sum_{\tau=0}^{\infty}\rho^\tau\left\|f^*_{t+\tau}\right\|^2\\ =& \frac{{C_F}^2}{1-\rho}\|P\|^2\left(\sum_{\tau=0}^{\infty}\rho^\tau\right)\left(\sum_{t=0}^{\infty}\left\|f^*_t\right\|^2\right)\\ \leq& \frac{{C_F}^2}{(1-\rho)^2}\|P\|^2\sum_{t=0}^{\infty}\left\|f^*_t\right\|^2. \end{aligned} \tag{3.46}$$

Putting (3.46) into (3.44), we obtain $\mathsf{OPT} \geq \frac{D_0(1-\rho)^2}{{C_F}^2\|P\|^2}\sum_{t=0}^{\infty}\|\psi_t\|^2$. □

Combining Lemma 15 with (3.42),

$$\sum_{t=0}^{\infty}\eta_t^\top H\eta_t \leq 2\|H\|\left(\frac{2\mathsf{OPT}}{\lambda_{\min}(R)}+\varepsilon\frac{\mathsf{ALG}}{\lambda_{\min}(Q)}+(1-\lambda)\frac{{C_F}^2\|P\|^2\mathsf{OPT}}{D_0(1-\rho)^2}\right). \tag{3.47}$$

Furthermore, since P is symmetric, $f_t^\top P f_t - (f_t^*)^\top P f_t^* = \left(f_t+f_t^*\right)^\top P\left(f_t-f_t^*\right)$, the RHS of the inequality (3.37) can be bounded by

$$\begin{aligned}&\sum_{t=0}^{\infty}\eta_t^\top H\eta_t + 2\sum_{t=0}^{\infty}\left(\|B\eta\|\left\|\sum_{\tau=t}^{\infty}\left(F^\top\right)^\tau P(f_{t+\tau}-f^*_{t+\tau})\right\| + \|P\left(f_t+f_t^*\right)\|\left\|f_t-f_t^*\right\|\right.\\ &\quad+\|x_0\|\left\|\sum_{t=0}^{\infty}\left(F^\top\right)^{t+1}P(f_t-f_t^*)\right\| + \|f_t\|\left\|\sum_{\tau=0}^{\infty}\left(F^\top\right)^{\tau+1}P(f_{t+\tau+1}-f^*_{t+\tau+1})\right\|\\ &\quad+\left\|\sum_{\tau=0}^{\infty}\left(F^\top\right)^{\tau+1}Pf^*_{t+\tau+1}\right\|\left\|f_t^*-f_t\right\|\\ &\quad\left.+\left\|BH^{-1}B^\top\right\|\left\|\sum_{\tau=t}^{\infty}\left(F^\top\right)^\tau Pf^*_{t+\tau}\right\|\left\|\sum_{\tau=t}^{\infty}\left(F^\top\right)^\tau P(f^*_{t+\tau}-f_{t+\tau})\right\|\right).\end{aligned}$$

Furthermore, by our assumption, the linearly combined policy π is an exponentially stabilizing policy and since f_t is Lipschitz continuous with a Lipschitz constant C_ℓ,

using (3.47) and noting that $\mathsf{OPT} > 0$,

$$\begin{aligned}&\mathsf{ALG} - \mathsf{OPT}\\ \leq& 2\|H\|\left(\frac{1}{\sigma}\left(2\mathsf{OPT} + \varepsilon\mathsf{ALG}\right) + (1-\lambda)\frac{{C_F}^2\|P\|^2\max\{2, \|A\|, \|B\|\}\mathsf{OPT}}{\sigma(1-\rho)^2}\right)\\ &+ O(C_\ell\|x_0\|).\end{aligned}$$

Rearranging the terms gives the competitive ratio bound in Lemma 14. □

Now, applying Lemma 14, we complete our competitive analysis by proving Theorem 3.4.2.

Proof of Theorem 3.4.2. Continuing from Theorem 3.E.2, it shows that if the Lipschitz constant C_ℓ satisfies $C_\ell < \frac{1-\rho}{C_F(1+\|K\|)}$ where $\left\|F^t\right\| \leq C_F\rho^t$ for any $t \geq 0$ and the consistency error ε satisfies $\varepsilon < \min\left\{\frac{\sigma}{2\|H\|}, \frac{1/C_F - C_a^{\mathsf{sys}}C_\ell}{C_\ell + \|B\|}\right\}$ where C_a^{sys} is defined in (3.7), then

$$\|x_t\| \leq \frac{\gamma^t - \mu^t}{1 - \mu\gamma^{-1}}\left(C_F + \mu\gamma^{-1}\right)\|x_0\|,$$

for all $t \geq 0$ with some $\mu < 1$. Therefore, Lemma 14 implies Theorem 3.4.2. □

Part II

Learning-Augmented Decision-Making

Chapter 4

LEARNING-BASED PREDICTIVE CONTROL: FORMULATION

[1] Tongxin Li, Steven H. Low, and Adam Wierman. Real-time flexibility feedback for closed-loop aggregator and system operator coordination. New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380096. URL https://doi.org/10.1145/3396851.3397725.

[2] Tongxin Li, Bo Sun, Yue Chen, Zixin Ye, Steven H. Low, and Adam Wierman. Learning-based predictive control via real-time aggregate flexibility. *IEEE Transactions on Smart Grid*, 12(6):4897–4913, 2021. URL https://doi.org/10.1109/TSG.2021.3094719.

The uncertainty and volatility of renewable sources such as wind and solar power has created a need to exploit the flexibility of distributed energy resources (DERs) and aggregators have emerged as dominate players for coordinating these loads [74, 75]. An aggregator can coordinate a large pool of DERs and be a single point of contact for independent system operators (ISOs) to call on for flexibility. This enables ISOs to minimize cost, respond to unexpected fluctuations of renewables, and even mitigate failures quickly and reliably. Typically, an ISO communicates a time-varying signal to an aggregator, e.g., a desired power profile, that optimizes ISO objectives and the aggregator coordinates with the DERs to collectively respond to the time-varying signal as faithfully as possible, e.g., by shaping their aggregate power consumption to follow ISO's power profile, while satisfying DER constraints. These constraints are often private to the loads, e.g., satisfying energy demands of electric vehicles before their deadlines. They limit the flexibility available to the aggregator so the aggregator must also communicate with the ISO by providing a feedback signal that quantifies its available flexibility. This feedback provides ISO with crucial information for determining the signal it sends to the aggregator. Thus the aggregator and the ISO form a closed-loop control system to manage the aggregate flexibility of DERs. This chapter focuses on the design of this closed-loop system and, in particular, the design of real-time feedback signals from the aggregator to the ISO quantifying the available flexibility and learning-augmented decision-making algorithms for the ISO.

4.1 Introduction

The design of *the aggregate flexibility feedback* signal is complex and has been the subject of significant research over the last decade, e.g., [32–36, 76–79]. Any feedback design must balance a variety of conflicting goals. Given the scale, complexity and privacy of the load constraints, it may neither be possible nor desirable to communicate precise information about every load. Instead, aggregate flexibility feedback must be a concise summary of a system's constraints and it must limit the leakage about specific load constraints. On the other hand, the feedback sent by an aggregator needs to be informative enough that it allows the ISO to achieve operational objectives, e.g., minimize cost, and, most importantly, containing feasibility information of the whole system with respect to the private load constraints. Moreover, a design for a flexibility feedback signal must be general enough to be applicable for a wide variety of controllable loads, e.g., electric vehicles (EVs), heating, ventilation, and air conditioning (HVAC) systems, energy storage units, thermostatically controlled loads, residential loads, and pool pumps. It is impractical to design different feedback signals for each load, so the same design must work for all DERs.

The challenge and importance of the design of flexibility feedback signals has led to the emergence of a rich literature. In many cases, the literature focuses on specific classes of controllable loads, such as EVs [80], heating, ventilation, and air conditioning (HVAC) systems [81, 82], energy storage units [79], thermostatically controlled loads [33] or residential loads and pool pumps [76, 83]. In the context of these applications, a variety of approaches have been suggested, e.g., convex geometric approximations via virtual battery models [33, 35], hyper-rectangles [36] and graphical interpretations [79]; scheduling based aggregation [84, 85]; linear combination of demand bit curves [82]; and probability-based characterization [76, 83]. These approaches have all yielded some success, especially in terms of quantifying available aggregate flexibility (see Section 4.2 for more detail on related work). However, nearly all prior work only focused on slower-timescale estimations and does not meet the goal of providing *real-time* aggregate flexibility feedback. The fast-changing environment and the uncertainties of the DERs, however, demand real-time flexibility feedback. For example, in an EV charging facility, it is notoriously challenging to predict future EV arrivals and their battery capacities. With on-site solar generation, the aggregator's dynamical system can be time-varying and non-stationary, so it is crucial that real-time feedback be defined and approximated for it to be used in online feedback-based applications.

Furthermore, most of the existing frameworks are designated for specific tasks, such as managing HVAC systems [81, 82], and therefore may not be applicable to other applications. Reinforcement learning (RL), especially, deep RL, has been used widely as approximation tools in smart grid applications. Joint pricing and EV charging scheduling for a single EV charger is considered in [86] using state–action–reward–state–action (SARSA). But it is unclear how the proposed method in [86] can be extended to allow multiple chargers. Q-learning is used to estimate the residual energy in an energy storage system at the end of each day in [87] and determine the aggregate action for thermostatically controlled loads (TCLs) [88]. The authors in [89] combine evolution strategies and model predictive control (MPC) to coordinate heterogeneous TCLs. Most existing studies, including the aforementioned works typically use RL for a "central controller" (which is an operator in our context). Instead we use it for the aggregator to learn flexibility representations.

To the best of our knowledge, no existing study has focused on the design of real-time coordination between an aggregator and a system operator that achieves the goals laid out above, except for some preliminary results in [24, 25]. Those results rely on a novel design of a real-time feedback signal that can be used to quantify the aggregate flexibility and coordinate real-time control. In this chapter, we extend the design of the feedback signal to a more general dynamic system with time-varying and non-stationary constraints, and we mainly focus on how to apply the real-time feedback to practical applications (e.g., EV charging) in power systems. Towards this goal, we propose a reinforcement learning based approach to approximate this feedback and further incorporate the feedback into a penalized predictive control (PPC) scheme. On the theory side, we prove the optimality of the proposed PPC scheme, and through extensive numerical tests, we validate the superior empirical performance of PPC over classic benchmarks, such as MPC.

Contributions. In summary, to complement previous research, in this chapter we consider a closed-loop control model formed by a system operator (central controller) and an aggregator (local controller) and propose a novel design of *real-time aggregate flexibility* feedback, called the *maximum entropy feedback* (MEF) that quantifies the flexibility available to an aggregator. Based on the definition of MEF, we design a reward function, which allows MEF to be efficiently learned by model-free RL algorithms. Our main contributions are:

1. We introduce a model of the real-time closed-loop control system formed by a system operator and an aggregator. This work is the first to close the loop and both define a concise measure of aggregate flexibility and show how it can be used by the system operator in an online manner to optimize system objectives while respecting the constraints of the aggregator's loads.

2. Within this model we define the "optimal" real-time flexibility feedback as the solution to an optimization problem that maximizes the entropy of the feedback vector. The use of entropy in this context is novel and to the best of our knowledge, this article is among the first to rigorously define a notion for *real-time* aggregate flexibility with provable properties. In particular we show that the exact MEF allows the system operator to maintain feasibility and enhance flexibility.

3. Furthermore, we propose a novel combination of control and learning by integrating model predictive control (MPC) and the defined MEF. Using the MEF as a penalty term, we introduce an algorithm called the *penalized predictive control* (PPC), which only requires the system operator to receive the MEF at each time, *without* knowing the states and dynamics of the aggregator. We also prove that, under certain regularity conditions, the actions given by PPC are optimal.

4. Finally, we demonstrate the efficacy of the proposed scheme using real EV charging data from Caltech's ACN-Data [1]. Our experiments show that by sending simple action signals generated by the PPC, a system operator is able to coordinate with an EV charging aggregator to satisfy almost all EV charging demands, while only knowing the MEF learned by a model-free off-policy RL algorithm. The PPC is also showed to achieve lower cost than MPC, which in addition needs to have access to the complete state of the loads.

4.2 Related Literature.

The growing importance of aggregators for the integration of controllable loads and the challenge of defining and quantifying the flexibility provided by aggregators has led to the emergence of a rich literature. Broadly, this work can be separated into three approaches.

Convex geometric approximation. The idea of representing the set of aggregate loads as a virtual battery model dates back to [32, 33]. In [35], flexibility of an aggregation

of thermostatically controlled loads (TCLs) was defined as the Minkowski sum of individual polytopes, which is approximated by the homothets of a virtual battery model using linear programming. The recent paper [36] takes a different approach and defines the aggregate flexibility as upper and lower bounds so that each trajectory to be tracked between the bounds is disaggregatable and thus feasible. However, convex geometric approaches cannot be extended to generate real-time flexibility signals because the approximated sets cannot be decomposed along the time axis. In [34], a belief function of setpoints is introduced for real-time control. However, feasibility can only be guaranteed when each setpoint is in the belief set and this may not be the case for systems with memory.

Scheduling algorithm-driven analysis. Scheduling algorithms that enable the aggregation of loads have been studied in depth over the past decade. The authors of [90, 91] introduced a decentralized algorithm with a real-time implementation for EV charging to track a given load profile. The authors of [84] considered the feasibility of matching a given power trajectory and show that causal optimal policies do not exist. In this work, aggregate flexibility was implicitly considered as the set of all feasible power trajectories. Three heuristic causal scheduling policies were compared and the results were extended to aggregation of deferrable loads and storage in [84]. Furthermore, decentralized participation of flexible demand from heat pumps and EVs was addressed in [85]. Notably, the flexibility signals that have emerged from this literature generally are applicable only to specific policies and DERs.

Probability-based characterization. There is much less work on probabilistic methods. The aggregate flexibility of residential loads was defined based on positive and negative pattern variations by analyzing collective behaviour of aggregate users [76]. A randomized and decentralized control architecture for systems of deferrable loads was proposed in [83], with a linear time-invariant system approximation of the derived aggregate non-linear model. Flexibility in this work was defined as an estimate of the proportion of loads that are operating. Our work falls into this category, but differs from previous papers in that entropy maximization for a closed-loop control system yields an interpretable signal that can be informative for operator objectives in real-time, as well as guarantee feasibility of the private constraints of loads (if the signal is accurate). In our previous work [25], we study the problem of real-time coordination of an aggregator and a system operator under the paradigm of a control framework and provide regret analysis assuming feasibility predictions are available.

Other approaches. Beyond the works described above, there are many other suggestions for metrics of aggregate flexibility, e.g., graphical-based measures [92] and data-driven approaches [92]. Most of these, and the approaches described above, are evaluated on the aggregator side only, and much less attention has been paid to the question of real-time coordination between an ISO and an aggregator that controls decentralized loads.

The assessment and enhancement of aggregate flexibility are often considered independent of the operational objectives. For instance, in a reserve market, an aggregator will report to the ISO a day in advance an offline notion of aggregated flexibility based on forecast for the ISO to compute a energy and reserve schedule for the following day, e.g., [32, 36, 77, 93], with notable exceptions, such as [80], which considered charging and discharging of EV fleets batteries for tracking a sequence of automatic generation control (AGC) signals. However, this approach has several limitations. First, in large-scale systems, knowing the exact states of each load is not realistic. Second, classical flexibility representations often rely on a precise state-transition model on the aggregator's side. Third, traditional ISO market designs, such as a day-ahead energy market, often make use of ex ante estimates of future system states. The forecasts of the future states can sometime be far from reality, because of either an inaccurate model is used, or an uncertain event occurs. In contrast, a real-time energy market [94, 95] provides more robust system control when facing uncertainty in the environment, e.g., from fast-changing renewable resources or human behavioral parameters. This further highlights the need for real-time flexibility feedback, and serves to differentiate the approach in our chapter. Below we present the notation frequently used in the remainder of this chapter.

Notation and Conventions. We use $\mathbb{P}(\cdot)$ and $\mathbb{E}(\cdot)$ to denote the probability distribution and expectation of random variables. The (differential) entropy function is denoted by $\mathbb{H}(\cdot)$. To distinguish random variables and their realizations, we follow the convention to denote the former by capital letters (e.g., U) and the latter by lower case letters (e.g., u). Furthermore, we denote the length-t prefix of a vector u by $u_{\leq t} := (u_1, \ldots, u_t)$. Similarly, $u_{<t} := (u_1, \ldots, u_{t-1})$ and $u_{a \to b} := (u_a, \ldots, u_b)$. The concatenation of two vectors u and v is denoted by (u, v). Given two vectors $u, v \in \mathbb{R}^n$, we write $u \preceq v$ if $u_i \leq v_i$ for all $i = 1, \ldots, n$. For $x \in \mathbb{R}$, denote $[x]_+ := \max\{0, x\}$. The set of non-negative real numbers is denoted by $\mathbb{R}_+$.

The rest of the chapter is organized as follows. We present our closed-loop control model in Section 4.3. We define real-time aggregate flexibility, called the MEF, and

prove its properties in Section 4.4. An RL-based approach for estimating the MEF is provided in Section 4.5. Combining MEF and model MPC, we propose an algorithm, termed the PPC in Section 4.6. Numerical results are given in Section 4.7.

4.3 Problem Formulation

In this chapter, we consider a real-time control problem involving two parties – a *load aggregator* and an independent system operator (*ISO*), or simply called an *operator* that interact over a discrete time horizon $[T] := \{1, \dots, T\}$.

Load Aggregator

A *load aggregator* is a device, often considered as a local controller that controls a fleet of controllable loads. In this part, we formally state the model of an aggregator and its objective. Let x_t denote the *aggregator state* at time t that takes value in a certain set $\mathsf{X} \subseteq \mathbb{R}^m$. To this end, the aggregator receives an *action* $u_t \in \mathsf{U}$ where $\mathsf{U} \subseteq \mathbb{R}$ denotes a closed and bounded set of actions at each time t from a *system operator*, which will be formally defined in Section 4.3. The action space U and state space X are prefixed and known as common knowledge to both the aggregator and the system operator. The goal of the aggregator is to accomplish a certain task over the horizon $[T]$, e.g., delivering energy to a set of EVs by their deadlines while minimizing the costs, subject to system constraints. Mathematically, the constraints are represented by two collections of *time-varying* and *time-coupling* sets $\{\mathsf{X}_t(x_{<t}, u_{<t}) \subseteq \mathsf{X} : t \in [T]\}$ and $\{\mathsf{U}_t(x_{<t}, u_{<t})_t \subseteq \mathsf{U} : t \in [T]\}$. For notational simplicity, we denote $\mathsf{X}_t(x_{<t}, u_{<t})$ by X_t and $\mathsf{U}_t(x_{<t}, u_{<t})$ by U_t in the remaining contexts. The states and actions must satisfy $x_t \in \mathsf{X}_t$ and $u_t \in \mathsf{U}_t$ for all $t \in [T]$. The decision changes the aggregator state x_t according to a *state transition function* f_t:

$$x_{t+1} = f_t(x_t, u_t),\ x_t \in \mathsf{X}_t,\ u_t \in \mathsf{U}_t, \tag{4.1}$$

where f_t represents the transition of the state x_t. The initial state x_1 is assumed to be the origin without loss of generality. The aggregator state x_t and decision u_t need to be chosen from two time-varying sets X_t and U_t. We make the following model assumptions:

Assumption 3. The dynamic $f_t(\cdot, \cdot) : \mathsf{X}_t \times \mathsf{U}_t \to \mathsf{X}_{t+1}$ is a Borel measurable function for $t \in [T]$. The time-varying and time-coupling sets $\{\mathsf{U}_t : t \in [T]\}$ and $\{\mathsf{X}_t : t \in [T]\}$ are Borel sets in $\mathbb{R}$ and $\mathbb{R}^m$.

The aggregator has flexibility in its actions u_t for accomplishing its task and, we assume for this chapter, is indifferent to these decisions as long as the task is

accomplished by time T. At each time t, based on its current state x_t, the aggregator needs to send *flexibility feedback*, p_t, a probability density function, from a collection of feedback signals P, to the system operator, which describes the flexibility of the aggregator for accepting different actions u_t. We formally define p_t and P in Section 4.4. Designing p_t is one of the central problems considered in this chapter (see Section 4.4 for more details). Below we state the aggregator's goal in the real-time control system.

Aggregator's Objective. *The goal of the aggregator is two-fold: (1). Maintain the feasibility of the system and guarantee that $x_t \in \mathsf{X}_t$ and $u_t \in \mathsf{U}_t$ for all $t \in [T]$. (2). Generate flexibility feedback p_t and send it to the operator at time $t \in [T]$.*

Remark 1. *We assume that the action space U is a continuous set in $\mathbb{R}$ only for simplicity of presentation. The results and definitions in the chapter can be extended to discrete setting by changing the integrals to summations, and replacing the differential entropy functions by discrete entropy functions, e.g., see the definition of maximum entropy feedback (Definition 4.4.1) and Lemma 17. In practical systems e.g., an electric system consisting of an EV aggregator and an operator, U often represents the set of power levels and when the gap between power levels is small, U can be modeled as a continuous set.*

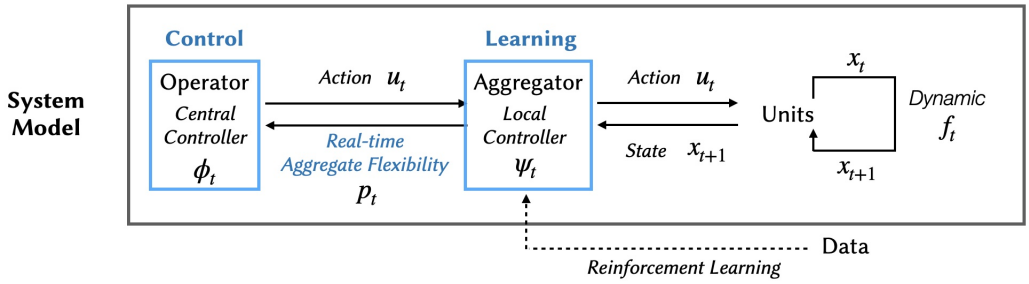


Figure 4.3.1: System model: A feedback control approach for solving an online version of (4.2). The operator implements a control algorithm and the aggregator uses reinforcement learning to generate real-time aggregate flexibility feedback.

System Operator

A system operator is a central controller that operates the power grid. Knowing the flexibility feedback p_t from the aggregator, the operator sends an action u_t, chosen from U to the aggregator at each time $t \in [T]$. Each action is associated with a cost function $c_t(\cdot) : \mathsf{U} \to \mathbb{R}_+$, e.g., the aggregate EV charging rate increases load on the electricity grid. The system's objective is stated as follows.

Operator's Objective. *The goal of the system operator is to provide an action $u_t \in \mathsf{U}$ at time $t \in [T]$ to the aggregator so as to minimize the cumulative system costs given by $C_T(u_1, \dots, u_T) := \sum_{t=1}^{T} c_t(u_t)$.*

Real-Time Operator-Aggregator Coordination

Overall, considering the aggregator and operator's objectives, the goal of the closed-loop system is to solve the following problem in *real-time*, by coordinating the operator and aggregator via $\{p_t : t \in [T]\}$ and $\{u_t : t \in [T]\}$:

$$\min_{u_1,\dots,u_T} C_T(u_1, \dots, u_T) \tag{4.2a}$$

$$\text{subject to } \forall t = 1, \dots, T :$$

$$x_{t+1} = f_t(x_t, u_t) \tag{4.2b}$$

$$x_t \in \mathsf{X}_t, \tag{4.2c}$$

$$u_t \in \mathsf{U}_t \tag{4.2d}$$

i.e., the operator aims to minimize its cost C_T in (4.2a) while the load aggregator needs to fulfill its obligations in the form of constraints (4.2b)-(4.2d). This is an offline problem that involves global information at all times $t \in [T]$.

Remark 2. *For simplicity, we describe our model in an offline setting where the cost and the constraints in the optimization problem* (4.2) *are expressed in terms of the entire trajectories of states and actions. The goal of the closed-loop control system is, however, to solve an online optimization via operator-aggregator coordination.*

The challenges are: (i) the aggregator and operator need to solve the online version of (4.2) jointly, and (ii) the cost function C_T is private to the operator and the constraints (4.2b)-(4.2d) are private to the operator. It is impractical for the aggregator to communicate the constraints to the operator because of privacy concerns or computational effort. Moreover, in an online setting, even the aggregator will not know the constraints that involve future information, e.g., future EV arrivals in an EV charging station. Formally, at each time $t \in [T]$, we assume that the operator and aggregator have access to the following information, respectively:

1. *An operator knows the costs $(c_1, \dots, c_t)$ and feedback $(p_1, \dots, p_t)$, but not the future costs $(c_{t+1}, \dots, c_T)$ and feedback $(p_{t+1}, \dots, p_T)$.*
2. *An aggregator knows the state transition functions $(f_1, \dots, f_T)$, the initial state x_1 and actions $(u_1, \dots, u_t)$.*

System's Goal. *Overall, the goal of a aggregator-operator system is to jointly solve the online version of* (4.2a)-(4.2d) *whose partial information is known to an aggregator and an operator, respectively.*

Necessities of Combining Learning and Control

With the assumptions above, on the one hand the aggregator cannot solve the problem independently because it does not have cost information (since the costs are often sensitive and only of the operator's interests) from the operator and even if the aggregator could, it may not have enough power to solve an optimization to obtain an action. On the other hand, the operator has to receive flexibility information from the aggregator in order to act. Well-known methods in pure learning or control cannot be used for this problem directly. From a learning perspective, the aggregator cannot simply use reinforcement learning and transmit parameters of a learned Q-function or an actor-critic model to the operator because the aggregator does not know the costs. From a control perspective, although model predictive control (MPC) is widely used for EV charging scheduling in practical charging systems [1, 96], it requires precise state information of electric vehicle supply equipment (EVSE). Thus, to solve the induced MPC problem, the system operator or aggregator needs to solve an online optimization at each time step that involves hundreds or even thousands of variables. This not just a complex problem, but the state information of the controllable units is potentially sensitive. This combination makes controlling sub-systems using precise information impractical for a future smart grid [24, 25] In this work, we explore a solution where the system operator and the aggregator jointly solve an online version of (4.2) in a closed loop, as illustrated in Figure 4.3.1.

The real-time operator-aggregator coordination illustrated in Figure 4.3.1 combines learning and control approaches. It does not require the aggregator to know the system operator's objective in (4.2a), but only the action u_t at each time $t \in [T]$ from the operator. In addition, it does not require the system operator to know the aggregator constraints in (4.2b), but only a feedback signal p_t (to be designed) from the aggregator. After receiving flexibility feedback p_t, which could be generated by machine learning algorithms, the system operator outputs an action u_t using a causal *operator function* $\phi_t(\cdot) : \mathsf{P} \to \mathsf{U}$. Knowing the state x_t, the aggregator generates its feedback p_t using a causal *aggregator function* $\psi_t(\cdot) : \mathsf{X} \to \mathsf{P}$ where P denotes the domain of flexibility feedback that will be formally defined in Section 4.4. By an "online feedback" solution, we mean that these functions (ϕ_t, ψ_t) use only information available locally at time $t \in [T]$.

Algorithm 5: Closed-Loop Feedback Control Framework *(*for a system operator (central controller) and an aggregator (local controller)).

for $t \in [T]$ **do**

Operator (Central Controller)

Generate actions using the PPC:

$$u_t = \phi_t\,(p_t)$$
$$C_t = C_{t-1} + c_t(u_t)$$

Aggregator (Local Controller)

Update system state:

$$x_{t+1} = f_t(x_t, u_t)$$

Compute estimated MEF:

$$p_{t+1} = \psi_t(x_{t+1})$$

end

Return total cost C_T;

In summary, the closed-loop control system in our model proceeds as follows. At each time t, the aggregator learns or computes a length-$|\mathsf{U}|$ vector p_t based on previously received action trajectory $u_{<t} = (u_1, \ldots, u_{t-1})$, and sends it to the system operator.[1] The system operator thencomputes a (possibly random) action $u_t = \phi_t(p_t)$ based on the flexibility feedback p_t and sends it to the aggregator. The operator chooses its signal u_t in order to solve the time-t problem in an online version of (4.2), so the function ϕ_t denotes the mapping from the flexibility feedback p_t to an optimal solution of the time-t problem. See 4.6 for examples. The aggregator then computes the next feedback p_{t+1} and the cycle repeats; see Algorithm 5. The goal of this chapter is to provide concrete constructions of an aggregator function ψ (as an MEF generator; see Section 4.4) and an operator function ϕ (via the PPC scheme; see Section 4.6).

In the sequel, we demonstrate our system model using an EV charging application, as an example of the problem stated in (4.2).

[1] We will omit $u_{<t}$ in the notation when it is not essential to our discussion and simplify the probability vector as p_t. Note that in (4.7c) we slightly abuse the notation and use p_t to denote a conditional distribution. This is only for computational purposes and the information sent from an aggregator to an operator at time $t \in [T]$ is still a length-$|\mathsf{U}|$ probability vector, conditioned on a fixed $u_{<t}$.

An EV Charging Example

Consider an aggregator that is an EV charging facility with n accepted users. Each user j has a private vector $(a(j), d(j), e(j), r(j)) \in \mathbb{R}^4$ where $a(j)$ denotes its arrival (connecting) time; $d(j)$ denotes its departure (disconnecting) time, normalized according to the time indices in $[T]$; $e(j)$ denotes the total energy to be delivered, and $r(j)$ is its peak charging rate. Fix a set of n users with their private vectors $(a(j), d(j), e(j), r(j))$, the *aggregator state* x_t at time $t \in [T]$ is a collection of length-2 vectors $(d_t(j), e_t(j) : a(j) \leq t \leq d(j))$ for each EV that has arrived and has not departed by time t. Here $e_t(j)$ is the remaining energy demand of user j at time t and $d_t(j)$ is the remaining charging time. The decision $s_t(j)$ is the energy delivered to each user j at time t, determined by a *scheduling policy* π_t such as the well-known earliest-deadline-first, least-laxity-first, etc. Let $s_t := (s_t(1), \ldots, s_t(n))$ and we have $s_t = \pi_t(u_t)$ where u_t in this example is the aggregate substation power level, chosen from a small discrete set U. The aggregator decision $s_t(j) \in \mathbb{R}_+$ at each time t updates the state, in particular $e_t(j)$ such that

$$e_t(j) = e_{t-1}(j) - s_t(j) \tag{4.3a}$$

$$d_t(j) = d_{t-1}(j) - \Delta \tag{4.3b}$$

where Δ denotes the time unit and we assume that there is no energy loss. The laws (4.3a)-(4.3b) are examples of the generic transition functions $f_1, \ldots, f_T$ in (4.1).

Suppose, in the context of demand response, the system operator (a local utility company, or a building management) sends a signal u_t that is the aggregate energy that can be allocated to EV charging. The aggregator makes charging decisions $s_t(j)$ to track the signal u_t received from the system operator as long as they will meet the energy demands of all users before their deadlines. Then the constraints in (4.2b)-(4.2d) are the following constraints on the charging decisions s_t, as a function of u_t:

$$s_t(j) = 0\,,\ t < a(j),\ j = 1, \ldots, n, \tag{4.4a}$$

$$s_t(j) = 0\,,\ t > d(j),\ j = 1, \ldots, n, \tag{4.4b}$$

$$\sum_{j=1}^{n} s_t(j) = u_t,\ t = 1, \ldots, T, \tag{4.4c}$$

$$\sum_{t=1}^{T} s_t(j) = e(j),\ j = 1, \ldots, n, \tag{4.4d}$$

$$0 \leq s_t(j) \leq r(j),\ t = 1, \ldots, T. \tag{4.4e}$$

In above, constraint (4.4c) ensures that the aggregator decision s_t tracks the signal u_t at each time $t \in [T]$, the constraint (4.4d) guarantees that EV j's energy demand is satisfied, and the other constraints say that the aggregator cannot charge an EV before its arrival, after its departure, or at a rate that exceeds its limit. Inequalities (4.4a)-(4.4e) above are examples of the constraints in (4.1). Together, for this EV charging application, (4.3a)-(4.3b) and (4.4a)-(4.4e) exemplify the dynamic system in (3.3).

The system operator's objective to minimize the cumulative costs $C_T(u) := \sum_{t=1}^{T} c_t u_t$ where $u = (u_1, \ldots, u_T)$ are substation power levels, as outlined in Section 4.3. The cost c_t depends on multiple factors such as the electricity prices and injections from an installed rooftop solar panel. Overall, the EV charging problem is formulated below, as a specific example of the generic optimization (4.2a)-(4.2d):

$$\min_{u_1,\ldots,u_T} \sum_{t=1}^{T} c_t u_t \tag{4.5a}$$

$$(4.3a) - (4.3b) \text{ and } (4.4a) - (4.4e). \tag{4.5b}$$

4.4 Definitions of Real-Time Aggregate Flexibility: Maximum Entropy Feedback

In this section, we propose a specific function ψ_t in the class defined by (4.6) for computing flexibility feedback to quantify its future flexibility. We will justify our proposal by showing that the proposed ψ_t has several desirable properties for solving an online version of (4.2) using the real-time feedback-based approach described in Section 4.3.

Definition of Flexibility Feedback p_t

A major challenge in our problem is that the operator has access to neither the feasible set nor the dynamics directly. Therefore, a notion termed *aggregate flexibility* has to be designed. It is often a "simplified" summary of the constraints in (4.2b)-(4.2d), as we reviewed in Section 4.2. Notably, existing aggregate flexibility definitions (for instance, in [32, 33, 35, 36, 76–79]) all focus on the offline version of (4.2). It remains unclear that first, *what is the right notion of real-time aggregate flexibility? i.e., what is the right form of the flexibility feedback p_t?* Second, *how can this p_t be used by an operator?*

In the following, we present a design of the flexibility feedback p_t, which is first proposed in our previous work [24] for discrete U and [25] for continuous U. It quantifies future flexibility that will be enabled by an operator action u_t. The feedback p_t therefore is a surrogate for the aggregator constraints (4.2b) to guide the operator's decision. Let $u := (u_1, \dots, u_T)$. Specifically, define the set of all *feasible action trajectories* for the aggregator as:

$$\mathsf{S} := \left\{ u \in \mathsf{U}^T : u \text{ satisfies (4.2b) – (4.2d)} \right\}.$$

The following property of the set S is useful, whose proof can be found in Appendix 4.A.

Lemma 16. *The set of feasible action trajectories* S *is Borel measurable.*

Existing aggregate flexibility definitions focus on approximating S such as finding its convex approximation (see Section 4.2 for more details). Our problem formulation needs a *real-time* approximation of this set S, i.e., decompose S along the time axis $t = 1, \dots, T$. Throughout, we assume that S is non-empty. Next, we define the space of flexibility feedback p_t. Formally, we let P denote a set of density functions $p_t(\cdot) : \mathsf{U} \to [0, 1]$ that maps an action to a value in $[0, 1]$ and satisfies

$$\int_{u \in \mathsf{U}} p(u) \mathrm{d}u = 1.$$

Fix x_t at time $t \in [T]$. The aggregator function $\psi_t(\cdot) : \mathsf{X} \to \mathsf{P}$ at each time t outputs:

$$\psi_t(x_t) = p_t(\cdot | u_{<t}) \tag{4.6}$$

such that $p_t(\cdot|u_{<t}) : \mathsf{U} \to [0, 1]$ is a conditional density function in P. We refer to p_t as *flexibility feedback* sent at time $t \in [T]$ from the aggregator to the system operator. In this sense, (4.6) does not specify a specific aggregator function ψ_t, but a class of possible functions ψ_t. Every function in this collection is *causal* in that it depends only on information available to the aggregator at time t. In contrast to most aggregate flexibility notions in the literature [32, 33, 35, 36, 76–79], the flexibility feedback here is specifically designed for an online feedback control setting.

Maximum Entropy Feedback

The intuition behind our proposal is using the conditional probability $p_t(u_t|u_{<t})$ to measure the resulting future flexibility of the aggregator if the system operator

chooses u_t as the signal at time t, given the action trajectory up to time $t-1$. The sum of the conditional entropy of p_t thus is a measure of how informative the overall feedback is. This suggests choosing a conditional distribution p_t that maximizes its conditional entropy. Consider the optimization problem:

$$F := \max_{p_1,\ldots,p_T} \sum_{t=1}^{T} \mathbb{H}(U_t|U_{<t}) \text{ subject to } U \in \mathsf{S} \tag{4.7a}$$

where the variables are conditional density functions:

$$p_t := p_t(\cdot|\cdot) := \mathbb{P}_{U_t|U_{<t}}(\cdot|\cdot), \qquad t \in [T], \tag{4.7b}$$

$U \in \mathsf{U}$ is a random variable distributed according to the joint distribution $\prod_{t=1}^{T} p_t$ and $\mathbb{H}(U_t|U_{<t})$ is the differential conditional entropy of p_t defined as:

$$\mathbb{H}(U_t|U_{<t}) := \int_{u_{\le t}\in\mathsf{U}^t} \Big(-\prod_{\ell=1}^{t} p_\ell(u_\ell|u_{<\ell}) \Big) \log p_t(u_t|u_{<t}) \mathrm{d}u_{\le t}. \tag{4.7c}$$

By definition, a quantity conditioned on "$u_{<1}$" means an unconditional quantity, so in the above, $\mathbb{H}(U_1|U_{<1}) := \mathbb{H}(U_1) := \mathbb{H}(p_1)$.

The chain rule shows that $\sum_{t=1}^{T} \mathbb{H}(U_t|U_{<t}) = \mathbb{H}(U)$. Hence (4.7) can be interpreted as maximizing the entropy $\mathbb{H}(U)$ of a random trajectory U sampled according to the joint distribution $\prod_{t=1}^{T} p_t$, conditioned on U satisfying $U \in \mathsf{S}$, where the maximization is over the collection of conditional distributions $(p_1, \ldots, p_T)$.

Definition 4.4.1 (Maximum entropy feedback)**.** The flexibility feedback $p_t^* = \psi_t^*(u_{<t})$ for $t \in [T]$ is called the maximum entropy feedback (MEF) if $(p_1^*, \ldots, p_T^*)$ is the unique optimal solution of (4.7).

Remark 3. *Even though the optimization problem* (4.7) *involves variables p_t for the entire time horizon $[T]$, the individual variables p_t in* (4.7c) *are conditional probabilities that depend only on information available to the aggregator at times t. Therefore the maximum entropy feedback p_t^* in Definition 4.4.1 is indeed causal and in the class of p_t^* defined in* (4.6)*. The existence of p_t^* is guaranteed by Lemma 17 below, which also implies that p_t^* is unique.*

We demonstrate Definition 4.4.1 using a toy example.

***Example* 3** (Maximum entropy feedback p^*)**.** *Consider the following instance of the EV charging example in Section 4.3. Suppose the number of charging time slots is* $T = 3$ *and there is one customer, whose private vector is* $(1, 3, 1, 1)$ *and possible energy levels are* 0 *(kWh) and* 1 *(kWh), i.e.,* $\mathsf{U} \equiv \{0, 1\}$*. Since there is only one EV, the scheduling algorithm* u *(disaggregation policy) assigns all power to this single EV. For this particular choices of* x *and* u*, the set of feasible trajectories is* $\mathsf{S} = \{(0, 0, 1), (0, 1, 0), (1, 0, 0)\}$*, shown in Figure 4.4.1 with the corresponding optimal conditional distributions given by* (4.7)*.*

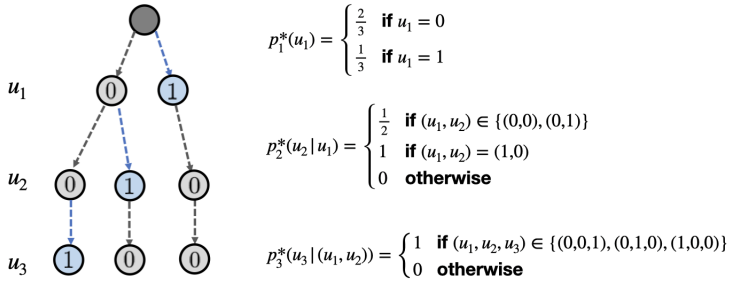


Figure 4.4.1: Feasible trajectories of power signals and the computed maximum entropy feedback in Example 3.

Properties of p_t^*

We now show that the proposed maximum entropy feedback p_t^* has several desirable properties. We start by computing p_t^* explicitly. Given any action trajectory $u_{\leq t}$, define the set of *subsequent* feasible trajectories as:

$$\mathsf{S}(u_{\leq t}) := \left\{ v_{>t} \in \mathsf{U}^{T-t} : v \text{ satisfies (4.2b) – (4.2d)}, v_{\leq t} = u_{\leq t} \right\}.$$

As a corollary, the size $|\mathsf{S}(u_{\leq t})|$ of the set of subsequent feasible trajectories is a measure of future flexibility, conditioned on $u_{\leq t}$. Our first result justifies our calling p_t^* the optimal flexibility feedback: p_t^* is a measure of the future flexibility that will be enabled by the operator's action u_t and it attains a measure of system capacity for flexibility. By definition, $p_1^*(u_1|u_{<1}) := p_1^*(u_1)$.

Lemma 17. *Let* $\mu(\cdot)$ *denote the Lebesgue measure. The MEF as optimal solutions of the maximization in (4.7a)-(4.7c) are given by*

$$p_t^*(u|u_{<t}) \equiv \frac{\mu\left(\mathsf{S}((u_{<t}, u))\right)}{\mu\left(\mathsf{S}(u_{<t})\right)}, \quad \forall (u_{<t}, u_t) \in \mathsf{U}^t. \tag{4.8}$$

Moreover, the optimal value of (4.7a)-(4.7c) *is equal to* $\log \mu(\mathsf{S})$.

Remark 4. *When the denominator $\mu(\mathsf{S}(u_{<t}))$ is zero, the numerator $\mu(\mathsf{S}((u_{<t}, u)))$ has also to be zero. For this case, we set $p_t^*(u|u_{<t}) = 0$ and this does not affect the optimality of* (4.7a)-(4.7b)*.*

The proof can be found in Appendix 4.B. The volume $\mu(\mathsf{S})$ is a measure of flexibility inherent in the aggregator. We will hence call $\log \mu(\mathsf{S})$ the *system capacity*. Lemma 17 then says that the optimal value of (4.7) is the system capacity, $F = \log \mu(\mathsf{S})$. Moreover the maximum entropy feedback $(p_1^*, \ldots, p_T^*)$ is the unique collection of conditional distributions that attains the system capacity in (4.7). This is intuitive since the entropy of a random trajectory x in S is maximized by the uniform distribution q^* in (4.12) induced by the conditional distributions $(p_1^*, \ldots, p_T^*)$.

Lemma 17 implies the following important properties of the maximum entropy feedback.

***Corollary* 4.4.1** (Feasibility and flexibility)**.** *Let $p_t^* = p_t^*(\cdot|u_{<t})$ be the maximum entropy feedback at each time $t \in [T]$.*

1. *For any action trajectory $u = (u_1, \ldots, u_T)$, if*

$$p_t^*(u_t|u_{<t}) \;>\; 0 \quad \textit{for all } t \in [T],$$

   then $u \in \mathsf{S}$.

2. *For all $u_t, u_t' \in \mathsf{U}$ at each time $t \in [T]$, if*

$$p_t^*(u_t|u_{<t}) \;\geq\; p_t^*(u_t'|u_{<t}),$$

   then $\mu(|\mathsf{S}((u_{<t}, u_t))) \geq \mu\left(\mathsf{S}((u_{<t}, u_t'))\right)$.

The proof is provided in Appendix 4.C. We elaborate the implication of Corollary 4.4.1 for our online feedback-based solution approach.

Remark 5 (Feasibility and flexibility)**.** *Corollary 4.4.1 says that the proposed optimal flexibility feedback p_t^* provides the right information for the system operator to choose its action u_t at time t.*

1. *(Feasibility) Specifically, the first statement of the corollary says that if the operator always chooses an action u_t with positive conditional probability $p_t^*(u_t) > 0$ for each time t, then the resulting action trajectory is guaranteed to be feasible, $u \in \mathsf{S}$, i.e., the system will remain feasible at* every *time $t \in [T]$ along the way.*

2. *(Flexibility) Moreover, according to the second statement of the corollary, if the system operator chooses an action u_t with a larger $p_t^*(u_t)$ value at time t, then the system will be more flexible going forward than if it had chosen another signal u_t' with a smaller $p_t^*(u_t')$ value, in the sense that there are more feasible trajectories in $\mathsf{S}((u_{<t}, u_t))$ going forward.*

As noted in Remark 2, despite characterizations that involve the whole action trajectory u, such as $u \in \mathsf{S}$, these are *online* properties. This guarantees the feasibility of the online closed-loop control system depicted in Figure 4.3.1, and confirms the suitability of p_t^* for online applications.

4.5 Approximating Maximum Entropy Feedback via Reinforcement Learning

For real-world applications, computing the maximum entropy feedback (MEF) could be computationally intensive. Thus, instead of computing it precisely, it is desirable to approximate it. In this section, we discuss the use of model-free reinforcement learning (RL) to generate an *aggregator function* ψ. For practical implementation, we switch to the case when U is a discrete set and reuse the notation P to denote a probability simplex that contains all possible discrete MEF:

$$\mathsf{P} := \left\{ p \in \mathbb{R}^{|\mathsf{U}|} : p(u) \geq 0, u \in \mathsf{U}; \sum_{u \in \mathsf{U}} p(u) = 1 \right\}. \tag{4.9}$$

We demonstrate that RL can be used to train a generator that outputs approximate MEF, given the state of the system. To be more precise, the learned aggregator function $\psi : \mathsf{X} \to \mathsf{P}$ outputs an estimate of the MEF given the state x_t at each time $t \in [T]$, where X is the state space and P is the set of all possible MEF. Note that the aggregator does not know the cost functions, so it cannot directly use an RL algorithm and transmit the learned Q-function or actor-critic model to the operator. Moreover, even if the aggregator knows the cost functions, generating actions using RL needs to solve two contradicting tasks of both optimizing rewards and penalizing feasibility violations, which makes the design of reward function and reward clipping a challenging goal. In our approach, we separate the tasks of enforcing feasibility and minimizing costs. We generate MEF as feasibility signals via reinforcement learning methods, and optimize the operator's objective via a MPC-based method (introduced in Section 4.6). It is also worth noting that a number of effective heuristics may be available such as a greedy approximation in [24] and other gradient-based or density estimation [97] methods. We leave to future work the question of finding an optimal approximation algorithm.

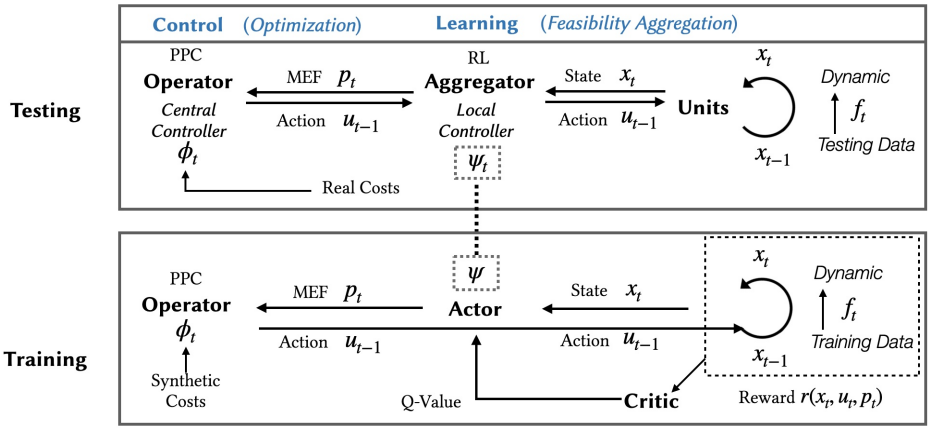


Figure 4.5.1: Learning and testing architecture for learning aggregator functions.

Offline Learning of Aggregator Functions

To learn an aggregator function ψ for estimating MEF, we use an actor-critic architecture [98] with separate policy and value function networks to enable the learning of policies on continuous action and state spaces. The actor-critic architecture is presented in Figure 4.5.1, which shows the information update between actor and critic networks. Note that in practical actor-critic algorithms, typically the policy, Q-function(s) and value function(s) are modeled using deep neural networks and the parameters are updated using policy iteration via stochastic gradient descent. We omit those details in Figure 4.5.1.

Training Process

During the training process, the data used for defining training dynamics are the episodes $(\mathsf{U}_t, \mathsf{X}_t, f_t)_{t=1}^T$. For example, for the EV charging application in Section 4.3, the training data of each episode (day) consist of historical private vectors $(a(j), d(j), e(j), r(j))$ specified by the users visited the charging station on the corresponding day. Among actor-critic-based RL algorithms, off-policy actor-critic methods, such as deep deterministic policy gradient (DDPG) [99] and soft actor-critic (SAC) [73] are known to attain better data efficiency in many applications. Below we take SAC, a maximum entropy deep RL algorithm, as an example to demonstrate the offline learning of an aggregator function ψ. In particular, for learning ψ, the objective of SAC is to maximize both the expected return and the expected entropy of the policy:

$$J(\psi) = \sum_{t=1}^{T} \mathbb{E}_{(\bar{x}_t, p_t)\sim\rho_\psi} \left[r(\bar{x}_t, p_t) + \alpha \mathbb{H}(\psi(\cdot|x_t)) \right] \tag{4.10}$$

where $\overline{x}_t := (x_t, u_t)$; ρ_ψ denotes the state-action marginals of the trajectory distribution induced by a policy ψ and $r(\overline{x}_t, p_t)$ is a customized reward function. To estimate MEF, we need to determine a reward function $r(\overline{x}_t, p_t)$ in (4.10). We adopt the following reward function that incorporates the constraints and the definition of MEF:

$$r(\overline{x}_t, p_t) = \mathbb{H}(p_t) + \sigma g(\overline{x}_t; \mathsf{X}_t, \mathsf{U}_t) \tag{4.11}$$

where the first term is critical and it maximizes the entropy of the probability distribution p_t, based the definition of the MEF in Definition 4.4.1; $g(\overline{x}_t) = g(x_t, u_t)$ is a function that rewards the state and action if they satisfy the constraints $x_t \in \mathsf{X}_t$ and $u_t \in \mathsf{U}_t$. The reward function is independent of the cost functions, which are synthetic costs in the training stage. A concrete example of $g(\overline{x}_t)$ is given in Section 4.7. We clip the output MEF given by the policy to make sure it is a probability vector in the probability defined in simplex (4.9). In Figure 4.5.2, a training curve is given and it displays the changes of rewards regarding to the number of training episodes.

Testing Process

With a trained aggregator function ψ that tries to optimize $J(\psi)$ in (4.10), we test the closed-loop system on new episodes defined by testing data, as shown in Figure 4.5.1. The trained aggregator function (parameterized by a deep neural network) is used as a "black box" function that maps each state x_t to feedback p_t.[2] Note that the real costs used in the testing process may not be same as the synthetic costs used in the training process, because the aggregator has no access to the costs as assumed in Section 4.3.

In the sequel, with the learned MEF, we introduce a closed-loop framework that combines model predictive control (MPC) and RL to coordinate a system operator and an aggregator in real-time. It is worth noting that the learned MEF may be different from the exact MEF provided in Definition 4.4.1. However, later we show in Section 4.7 that with the learned MEF, the constraints on the aggregator's side can almost be satisfied with a reasonable tuning parameter. In the EV charging example described in Section 4.3, this means the EV's batteries are fully charged; see Figure 4.7.2 for details.

[2] In our model, in general the aggregator functions $\psi_1, \ldots, \psi_T$ can be time-dependent. In the offline learning process presented in this section, we use a single function to generate feedback.

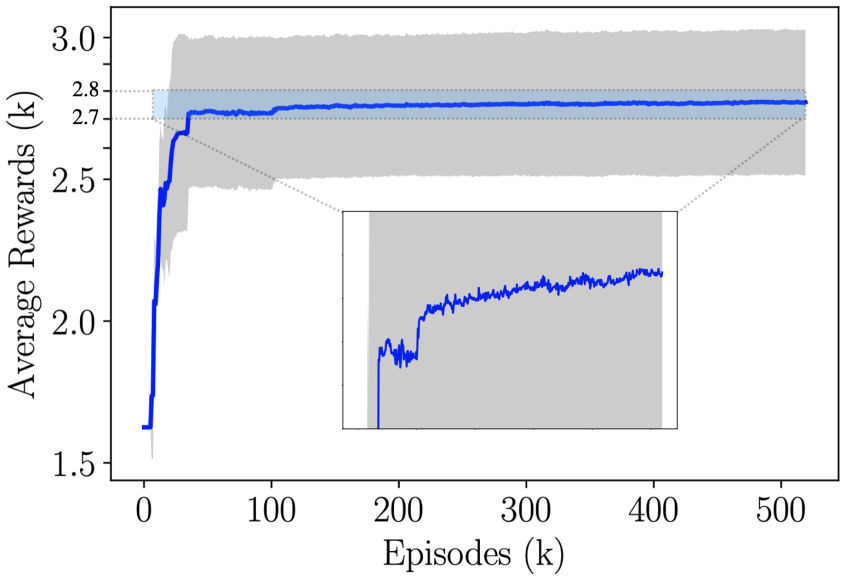


Figure 4.5.2: Average rewards (defined in (4.11)) in the training stage with a tuning parameter $\beta = 6 \times 10^3$. Shadow region measures the variance.

4.6 Penalized Predictive Control

Consider the system model in Section 4.3. In this setting, the operator seeks to minimize the cost in an online manner, i.e., at time $t \in [T]$ the operator only knows the objective functions $c_1, \ldots, c_t$ and the flexibility feedback $p_1, \ldots, p_t$. The task of the operator is to, given the maximum entropy feedback, design a sequence of *operator functions* $\phi_1, \ldots, \phi_T$ to generate actions $u_1, \ldots, u_T$ that are always feasible with respect to the constraints *and* that minimize the cumulative cost.

Key Idea: Maximum Entropy Feedback as a Penalty Term

There is in general a trade-off between ensuring future flexibility and minimizing the current system cost in predictive control. The action u_t guaranteeing the maximal future flexibility, i.e., having the largest $p_t^*(u_t|u_{<t})$ may not be the one that minimizes the current cost function c_t and vice versa. Therefore, the online algorithm for the central controller must balance future flexibility and current cost. The key idea is to use MEF as a penalty term in the offline optimization problem. Note that Corollary 4.4.1 guarantees that the online agent can always find a feasible action $u \in \mathsf{S}$. Indeed, knowing the MEF p_t^* for every $t \in [T]$ is equivalent to knowing the set of all admissible sequences of actions S. To see this, consider the unique maximum entropy feedback $(p_1^*, \ldots, p_T^*)$ guaranteed by Lemma 17 and let $q(u) = \prod_{t=1}^{T} p_t^*(u_t|u_{<t})$ denote the joint distribution of the action trajectory u. Then (4.8) implies that the joint distribution q is the uniform distribution over the set S of

all feasible trajectories:

$$q(u) := \begin{cases} 1/\mu(\mathsf{S}) & \text{if } u \in \mathsf{S} \\ 0 & \text{otherwise} \end{cases}. \tag{4.12}$$

Using this observation, the constraints (4.2b)-(4.2d) in the offline optimization can be rewritten as a penalty in the objective of (4.2a). We present a useful lemma that both motivates our online control algorithm and builds up the optimality analysis in Section 4.6.

Lemma 18. *The offline optimization* (4.2a)-(4.2d) *is equivalent to the following unconstrained minimization for any* $\beta > 0$*:*

$$\inf_{u \in \mathsf{U}^T} \sum_{t=1}^{T} \left(c_t(u_t) - \beta \log p_t^*(u_t | u_{<t}) \right).$$

The proof of Lemma 18 can be found in Appendix 4.E. It draws a clear connection between MEF and the offline optimal, which we exploit in the design of an online system operator in the next section.

Algorithm: Penalized Predictive Control via MEF

Our proposed design, termed penalized predictive control (PPC), is a combination of model predictive control (MPC) (*c.f.* [100]), which is a competitive policy for online optimization with predictions, and the idea of using MEF as a penalty term. This design makes a connection between the MEF and the well-known MPC scheme. The MEF as a feedback function, only contains limited information about the dynamical system in the local controller's side. (It contains only the feasibility information of the current and future time slots, as explained in Section 4.4). The PPC scheme therefore is itself a novel contribution since it shows that, even if *only* feasibility information is available, it is still possible to incorporate the limited information to MPC as a *penalty term*.

We present PPC in Algorithm 6, where we use the following notation. Let $\beta_t > 0$ be a *tuning parameter* in predictive control to trade-off the flexibility in the future and minimization of the current system cost at each time $t \in [T]$. The next corollary follows whose proof is in Appendix 4.C.

***Corollary* 4.6.1** (Feasibility of PPC)**.** *When* $p_t = p_t^*$ *for all* $t \in [T]$*, the MEF defined in Definition 4.4.1, the sequence of actions* $u = (u_1, \ldots, u_T)$ *generated by the PPC in* (4.13) *always satisfies* $u \in \mathsf{S}$ *for any sequence of tuning parameters* $(\beta_1, \ldots, \beta_T)$*.*

Algorithm 6: Penalized Predictive Control (PPC).

Data: Sequentially arrived cost functions and MEF
Result: Actions $u = (u_1, \ldots, u_T)$
for $t = 1, \ldots, T$ **do**
  Choose an action u_t by minimizing:

$$u_t = \phi_t(p_t) := \arg\inf_{u_t \in \mathsf{U}} \left(c_t(u_t) - \beta_t \log p_t(u_t|u_{<t})\right) \tag{4.13}$$

end
Return u;

Framework: Closed-Loop Control between Local and Central Controllers

Given the PPC scheme described above, we can now formally present our online control framework for the distant central controller and local controller (defined in Section 4.3). Recall that an overview of the closed-loop control framework has been given in Algorithm 5, where ϕ denotes an operator function and ψ is an aggregator function. To the best of our knowledge, this chapter is the first to consider such a closed-loop control framework with limited information communicated in real-time between two geographically separate controllers seeking to solve an online control problem. We present the framework below.

At each time $t \in [T]$, the local controller first efficiently generates estimated MEF $p_t \in \mathsf{P}$ using an aggregator function ψ_t trained by a reinforcement learning algorithm. After receiving the current MEF p_t and cost function c_t (future w MEF and costs if predictions are available), the central controller uses the PPC scheme in Algorithm 6 to generate an action $u_t \in \mathsf{U}$ and sends it back to the local controller. The local controller then updates its state $x_t \in \mathsf{X}$ to a new state x_{t+1} based on the system dynamic in (3.3) and repeats this procedure again. In the next Section, we use an EV charging example to verify the efficacy of the proposed method.

Optimality Analysis

To end our discussion of PPC we focus on optimality. For the ease of analysis, we assume that the action space U is the set of real numbers $\mathbb{R}$; however, as noted in Remark 1, our system and the definition of MEF can also be made consistent with a discrete action space.

To understand the optimality of PPC we focus on standard regularity assumptions for the cost functions and the time-varying constraints. We assume cost functions

are strictly convex and differentiable, which is common in practice. Further, let $\mu(\cdot)$ denote the Lebesgue measure. Note that the set of subsequent feasible action trajectories $\mathsf{S}(u_{\leq t})$ is Borel-measurable for all $t \in [T]$, implied by the proof of Corollary 16. We also assume that the measure of the set of feasible actions $\mu(\mathsf{S}(u_{\leq t}))$ is differentiable and strictly logarithmically-concave with respect to the subsequence of actions $u_t = (u_1, \dots, u_t)$ for all $t \in [T]$, which is also common in practice, e.g., it holds in the case of inventory constraints $\sum_{t=1}^{T} \|u_t\|_2 \leq B$ with a budget $B > 0$. Finally, recall the definition of the set of subsequent feasible action trajectories:

$$\mathsf{S}(u_{\leq t}) := \left\{ v_{>t} \in \mathsf{U}^{T-t} : v \text{ satisfies (4.2b) – (4.2d)}, v_{\leq t} = u_{\leq t} \right\}.$$

Putting the above together, we can state our assumption formally as follows.

Assumption 4. The cost functions $c_t(u) : \mathbb{R} \to \mathbb{R}_+$ are differentiable and strictly convex. The mappings $\mu(\mathsf{S}(u_{\leq t})) : \mathbb{R}^t \to \mathbb{R}_+$ are differentiable and strictly logarithmically-concave.

Given regularity of the cost functions and time-varying constraints, we can prove optimality of PPC.

Theorem 4.6.1 (Existence of optimal actions)**.** *Let* $\mathsf{U} = \mathbb{R}$*. Under Assumption 3 and 4, there exists a sequence* $b_1, \dots, b_T$ *such that implementing* (4.13) *with* $\beta_t = b_t$ *and* $p_t = p_t^*$ *at each time* $t \in [T]$ *ensures* $u = (u_1^*, \dots, u_T^*)$*, i.e., the generated actions are optimal.*

Crucially, Theorem 4.6.1 shows that there exists a sequence of "good" tuning parameters so that the PPC scheme is able to generate optimal actions under reasonable assumptions. However, note that the assumption of $\mathsf{U} = \mathbb{R}$ is fundamental. When the action space U is discrete or U is a high-dimensional space, it is impossible to generate the optimal actions because, in general, fixing t, the differential equations in the proof of Theorem 4.6.1 (see Appendix 4.F) do not have the same solution for all $\beta_t > 0$. Therefore a detailed regret analysis is necessary in such cases, which is a challenging task for future work.

4.7 Application

In this section, we present experimental results for the case of online EV charging, introduced in Section 4.3 as an example of our system model (see Section 4.3). The notation used in this section, if not defined, can be found in Section 4.3.

Experimental Setups

In the following, we present settings of parameters and useful metrics in our experiments.

Dataset and hardware. We use real EV charging data from ACN-Data [1], which is a dataset collected from adaptive EV charging networks (ACNs) at Caltech and JPL. The detailed hardware setup for that EV charging network structure can be found in [101].

Error metrics. Recall the EV charging example in optimization (4.5a)-(4.5b). We first introduce two error metrics to measure the EV charging constraint violations. Note that the constraints (4.4a), (4.4b) and (4.4e) are hard constraints depending only on the scheduling policy, but not the actions and energy demands. Therefore they can be automatically satisfied in our experiments by fixing a scheduling policy satisfying them such as least laxity first. Violations may happen on constraint (4.4c) and (4.4d). To measure the violation of (4.4c), we use the (normalized) mean squared error (MSE) as the tracking error:

$$\mathsf{MSE} := \sum_{k=1}^{L}\sum_{t=1}^{T}\Big|\sum_{j=1}^{N} s_t^{(k)}(j) - u_t^{(k)}\Big|^2/(L\times T\times \xi)\,, \tag{4.14}$$

where $u_t^{(k)}$ is the t-th power signal for the k-th test and $s_t^{(k)}(j)$ is the energy scheduled to the j-th charging session at time t for the k-th test. To better approximate real-world cases, we consider an additional *operational constraints* for the operator (central controller) and require that $u_t \le \xi$ (kWh) for every $t \in [T]$. The total number of tests is L and the total number of charging sessions is n. Additionally, define the mean percentage error with respect to the undelivered energy corresponding to (4.4d) as

$$\mathsf{MPE} := 1 - \sum_{k=1}^{L}\sum_{t=1}^{T}\sum_{j=1}^{n} s_t^{(k)}(j)\Big/\Big((L\times T)\cdot\sum_{j=1}^{n} e_j\Big), \tag{4.15}$$

where e_j is the energy request for each charging session $j \in [n]$; $s_t^{(k)}(j)$ is the energy scheduled to the j-th charging session at time t for the k-th test.

Hyper-parameters.

The detailed parameters used in our experiments are shown in Table 4.7.1.

Control spaces. For the experimental results presented in this section, the control state space is $\mathsf{X} = \mathbb{R}_+^{2\times W}$ where W is the total number of charging stations and a state vector for each charging station is $(e_t, [d(j)-t]^+)$, i.e., the remaining energy to be charged

Table 4.7.1: Hyper-parameters in the experiments.

Parameter	Value
System Operator	
Number of power levels $\lvert \mathsf{U} \rvert$	10
Cost functions $c_1, \ldots, c_T$	Average LMPs
Operator function ϕ	Penalized Predictive Control
Tuning parameter β	1×10^3 - 1×10^6
EV Charging Aggregator	
Number of Chargers W	54
State space X	$\mathbb{R}_+^{108}$
Action space	$[0, 1]^{10}$
Time interval Δ	12 minutes
Private vector $(a(j), d(j), e(j), r(j))$	ACN-Data [1]
Power rating	150 kW
Scheduling algorithm π	Least Laxity First (LLF)
Laxity	$d_t(j) - e_t(j)/r(j)$
RL algorithm	Soft Actor-Critic (SAC) [73]
Optimizer	Adam [102]
Learning rate	$3 \cdot 10^{-4}$
Discount factor	0.5
Relay buffer size	10^6
Number of hidden layers	2
Number of hidden units per layer	256
Number of samples per minibatch	256
Non-linearity	ReLU
Reward function	$\sigma_1 = 0.1$, $\sigma_2 = 0.2$, $\sigma_3 = 2$
Temperature parameter	0.5

and the remaining charging time if it is being used (see Section 4.3); otherwise the vector is an all-zero vector. The control action space is $\mathsf{U} = \{0, 15, 30, \ldots, 150\}$ (unit: kW) with $\lvert \mathsf{U} \rvert = 10$, unless explicitly stated. The scheduling policy π is fixed to be least-laxity-first (LLF).

RL spaces. The RL action space[3] of the Markov decision process used in the RL algorithm is $[0, 1]^{10}$. The outputs of the neural networks are clipped into the probability simplex (space of MEF) P afterwards.

[3]Note that the RL action space (consisting of p_t's) and state space (consisting of x_t's) referred here are the standard definitions in the context of RL and they are different from the "control action space" U and "control state space" X defined in Section 4.3.

RL rewards. We use the following specific reward function for our EV charging scenario, as a concrete example of (4.11):

$$\begin{aligned} r_{\mathrm{EV}}(\overline{x}_t, p_t) =& \mathbb{H}(p_t) \\ &+\sigma_1 \sum_{i=1}^{n'} \|u_t(i)\|_2 \\ &-\sigma_2 \sum_{i=1}^{n'} \left(\mathbf{I}(a(j_i) \le t \le a(j_i) + \Delta) \Big[e(i) - \sum_{t=1}^{T} u_t(i) \Big]_+ \right) \\ &-\sigma_3 \left| \phi_t(p_t) - \sum_{i=1}^{n'} u_t(j) \right| \end{aligned} \tag{4.16}$$

where σ_1, σ_2 and σ_3 are positive constants; n' is the number of EVs being charged; ϕ_t is the operator function, which is specified by (4.13); $\mathbf{I}(\cdot)$ denotes an indicator function and $a(j_i)$ is the arrival time of the $i-th$ EV in the charging station with j_i being the index of this EV in the total accepted charging sessions $[n]$. The entropy function $\mathbb{H}(p_t)$ in the first term is a greedy approximation of the definition of MEF (see Definition 4.4.1). The second term is to further enhance charging performance and the last two terms are realizations of the last term in (4.11) for constraints (4.4c) and (4.4d). Note that The other constraints in the example shown in Section 4.3 can automatically be satisfied by enforcing the constraints in the fixed scheduling algorithm π. With the settings described above, in Figure 4.5.2 we show a typical training curve of the reward function in (4.16). We observe policy convergence with respect to a wide range of choices of the hyper-parameters σ_1, σ_2 and σ_3. In our experiments, we do not optimize them but fix the constants in (4.16) as $\sigma_1 = 0.1$, $\sigma_2 = 0.2$ and $\sigma_3 = 2$.

Cost functions. We consider the specific form of costs in (4.5a). In the RL training process, we train an aggregator function ψ using linear price functions $c_t = 1 - t/24$ where $t \in [0, 24]$ (unit: Hrs) is the time index and we test the trained system with real price functions $c_1, \ldots, c_T$ being the average locational marginal prices (LMPs) on the CAISO (California Independent System Operator) day-ahead market in 2016 (depicted at the bottom of Figure 4.7.3).

Tuning parameters. In PPC defined in Algorithm 6, there is a sequence of tuning parameters $(\beta_1, \ldots, \beta_T)$. In our experiments, we fix $\beta_t = \beta$ for all $t \in [T]$ where $\beta > 0$ is a universal tuning parameter that can be varied in our experiments.

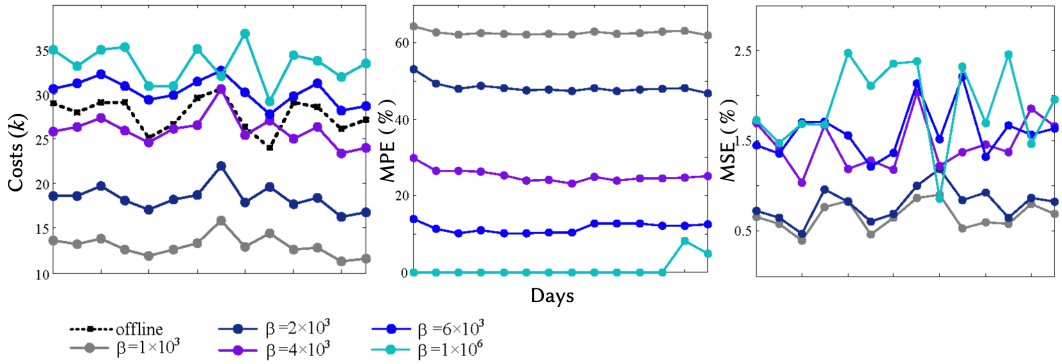


Figure 4.7.1: Trade-offs of cost and charging performance. The dashed curve in the left figure corresponds to offline optimal cost. The tested days are selected (with no less than 30 charging sessions, i.e., $N \geq 30$) from Dec. 2, 2019 to Jan. 1, 2020.

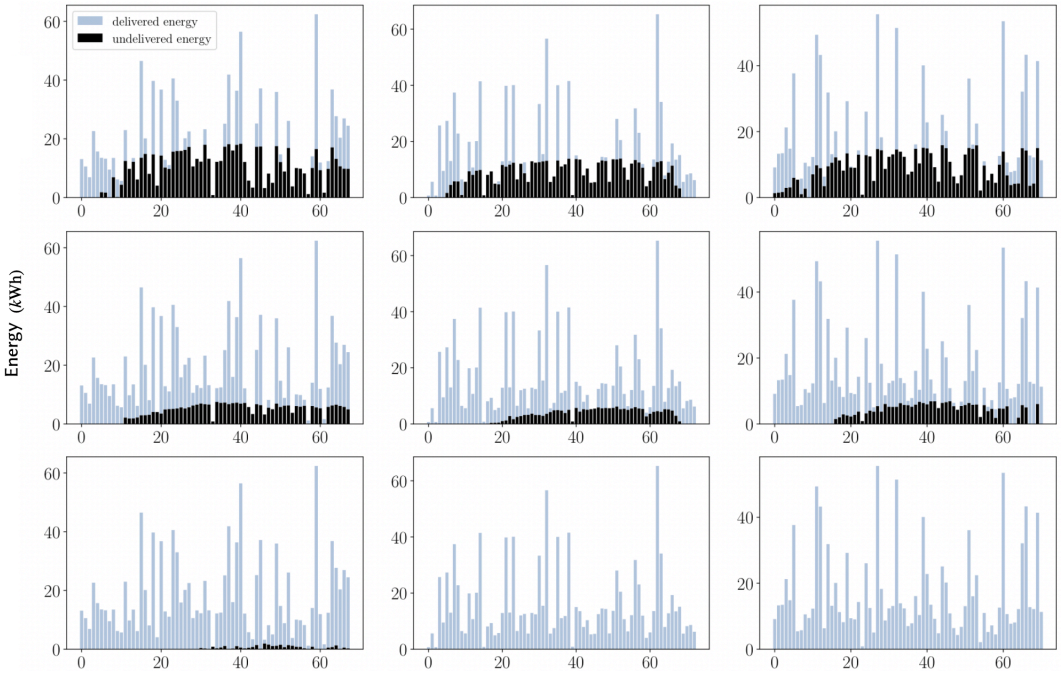


Figure 4.7.2: Charging results of EVs controlled by PPC with tuning parameters $\beta = 2 \times 10^3$ (top), 4×10^3 (mid) and 6×10^3 (bot) for selected days (with no less than 30 charging sessions, i.e., $N \geq 30$) from Dec. 2, 2019 to Jan. 1, 2020. Each bar represents a charging session.

Experimental Results.

Sensitivity of β. We first show how the changes of the tuning parameter β affect the total cost and feasibility. Figure 4.7.1 compares the results by varying β. The agents are trained on data collected from Nov. 1, 2018 to Dec. 1, 2019 and the tests are performed on data from Dec. 2, 2019 to Jan. 1, 2020 . Weekends and days with less than 30 charging sessions are removed from both training and testing data. For

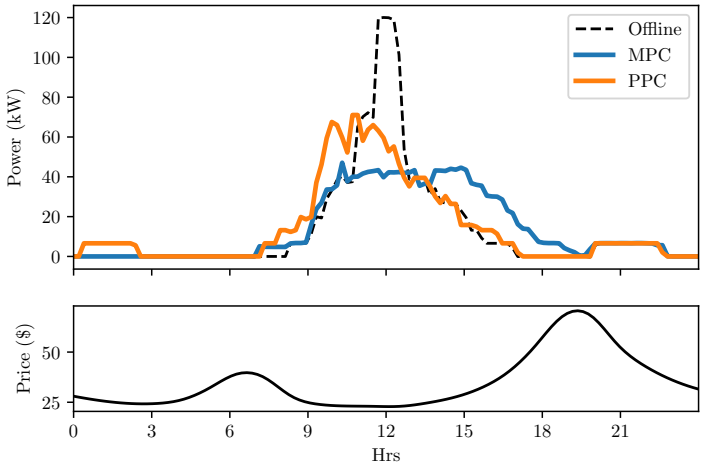


Figure 4.7.3: Substation charging rates generated by the PPC (orange) in the closed-loop control shown in Algorithm 5, together with the MPC generated (blue) and global optimal (dashed black) charging rates.

charging performance, we show in Figure 4.7.2 the battery states of each session after the charging cycle ends, tested with tuning parameters $\beta = 2 \times 10^3, 4 \times 10^3$ and 6×10^3, respectively. The results indicate that with a sufficiently large tuning parameter, the charging actions given by the PPC is able to satisfy EVs' charging demands and in practice, there is a trade-off between costs and feasibility depending on the choice of tuning parameters.

Charging curves. In Figure 4.7.3, substation charging rates (in kW) are shown. The charging rates generated by the PPC correspond to a trajectory

$$\left(\sum_j s_1(j)/\Delta, \ldots, \sum_j s_T(j))\Delta \right),$$

which is the aggregate charging power given by the PPC for all EVs at each time $t = 1, \ldots, T$. The agent is trained on data collected at Caltech from Nov. 1, 2018 to Dec. 1, 2019 and tested on Dec. 16, 2019 at Caltech using real LMPs on the CAISO day-ahead market in 2016. We use a tuning parameter $\beta = 4 \times 10^3$ for both training and testing. The figure highlights that, with a suitable choice of tuning parameter, the operator is able to schedule charging at time slots where prices are lower and avoid charging at the peak of prices, as desired. In particular, it achieves a lower cost compared with the commonly used MPC scheme described in (2.3)-(4.17f).The offline optimal charging rates are also provided.

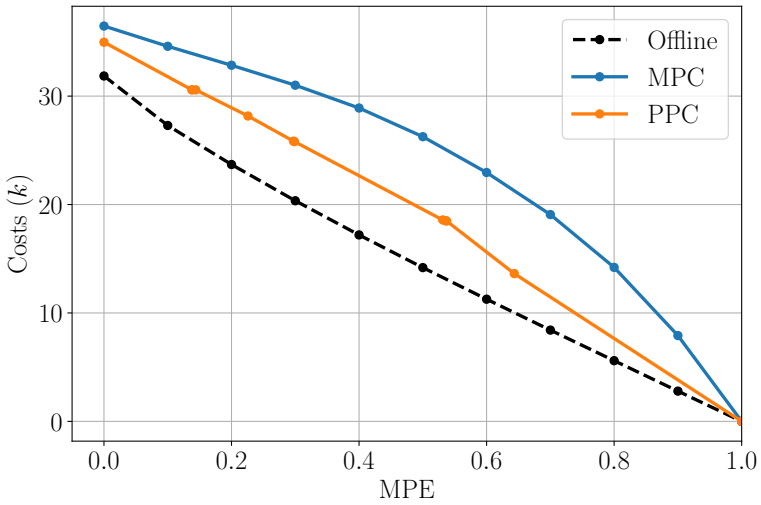


Figure 4.7.4: Cost-energy curves for the offline optimization in (4.2a)-(4.2d) (for the example in Section 4.3), MPC (defined in (4.17a)-(4.17f)) and PPC (introduced in Section 4.6).

Comparison of PPC and MPC. In Figure 4.7.4, we show the changes of the cumulative costs by varying the mean percentage error (MPE) with respect to the undelivered energy defined in (4.15). There are in total $K = 14$ episodes tested for days selected from Dec. 2, 2019 to Jan. 1, 2020 (days with less than 30 charging sessions are removed, i.e. we require, $N \geq 30$). Note that $0 \leq \mathsf{MPE} \leq 1$ and the larger MPE is, the higher level of constraint violations we observe. We allow constraint violations and modify parameters in the MPC and PPC to obtain varying MPE values. For the PPC, we vary the tuning parameter β to obtain the corresponding costs and MPE. For the MPC in our tests, we solve the following optimization at each time for obtaining the charging decisions $s_t = (s_t(1), \ldots, s_t(n'))$:

$$s_t = \arg\min_{s_t} \sum_{\tau=t}^{t'} c_\tau\Big(\sum_{i=1}^{n'} s_\tau(i)\Big) \text{ subject to :} \tag{4.17a}$$

$$s_\tau(i) = 0\,,\ \tau < a(i), i = 1, \ldots, n', \tag{4.17b}$$

$$s_\tau(i) = 0\,,\ \tau > d(i), i = 1, \ldots, n', \tag{4.17c}$$

$$\sum_{i=1}^{n'} s_\tau(i) = u_t, \tau = t, \ldots, t', \tag{4.17d}$$

$$\sum_{\tau=1}^{T} s_\tau(i) = \gamma \cdot e(i), i = 1, \ldots, n', \tag{4.17e}$$

$$0 \leq s_\tau(i) \leq r(i), \tau = t, \ldots, t' \tag{4.17f}$$

where at time t, the integer n' denotes the number of EVs being charged at the charging station and the time horizon of the online optimization is from $\tau = t$ to t', which is the latest departure time of the present charging sessions; $a(i)$ and $d(i)$ are the arrival time and departure time of the i-th session; $\gamma > 0$ relaxes the energy demand constraints and therefore changes the MPE region for MPC. The offline cost-energy curve is obtained by varying the energy demand constraints in (4.4d) in a similar way. We assume there is no admission control and an arriving EV will take a charger whenever it is idle for both MPC and PPC. Note that this MPC framework is widely studied [103] and used in EV charging applications [96]. It requires the precise knowledge of a 108-dimensional state vector of 54 chargers at each time step. We observe that with *only* feasibility information, PPC outperforms MPC for all $0 \leq \mathsf{MPE} \leq 1$. The main reason that PPC outperforms vanilla MPC is that PPC utilizes MEF as its input, which is generated by a pre-trained aggregator function. Therefore the MEF may contain useful future feasibility information that vanilla MPC does not know, despite that it is trained and tested on separate datasets.

APPENDIX

4.A Proof of Lemma 16

Proof. We first define a set $f^{-1}(\mathsf{X}_t)$ denoting the inverse image of the set X_t for actions: $f^{-1}(\mathsf{X}_t)(u_{<t}) := \{u \in \mathsf{U} : f(x_{t-1}, u) \in \mathsf{X}_t\}$. The inverse image $f^{-1}(\mathsf{X}_t)$ depends only on the past actions $u_{<t}$ since the states $x_{<t}$ are determined by $u_{<t}$ and a pre-fixed initial state x_1 via the dynamics in (3.3). Note that X_t and the dynamic f are Borel measurable. Therefore the inverse image $f^{-1}(\mathsf{X}_t)$ is also a Borel set, implying that the intersection $\mathsf{U}_t \bigcap f^{-1}(\mathsf{X}_t)$ is also Borel measurable. The set of feasible action trajectories S can be reprised as

$$\mathsf{S} := \left\{u \in \mathsf{U}^T : u_t \in \mathsf{U}_t \bigcap f^{-1}(\mathsf{X}_t)(u_{<t}), \forall t \in [T]\right\},$$

which is a Borel measurable set of all *feasible* sequences of actions. □

4.B Proof of Lemma 17

Proof of Lemma 17. We prove the statement by induction. It is straightforward to verify the results hold when $T = 1$. We suppose the lemma is true when $T = m$. Suppose $T = m + 1$. Let

$$F(u) := \max_{p_2,\ldots,p_T} \sum_{t=2}^{T} \mathbb{H}(U_t|\mathbf{U}_{2:t-1}; U_1 = u)$$

denote the optimal value corresponding to the time horizon $t \in [T]$, given the first action $U_1 = u$. By the definition of conditional entropy, we have

$$F = \max_{p_1} \int_{u\in\mathsf{U}} p_1(u)F(u)\mathsf{d}u + \mathbb{H}(p_1).$$

By the induction hypothesis, $F(u) = \mu(\mathsf{S}(u))$. Therefore,

$$\begin{aligned} F &= \max_{p_1} \int_{u\in\mathsf{U}} p_1(u) \log \mu(\mathsf{S}(u))\, \mathsf{d}u + \mathbb{H}(p_1) \\ &= \max_{p_1} \int_{u\in\mathsf{U}} p_1(u) \log\left(\frac{\mu(\mathsf{S}(u))}{p_1(u)}\right) \mathsf{d}u \end{aligned}$$

whose optimizer p_1^* satisfies (4.8) and we get $F = \mu(\mathsf{S})$. The lemma follows by finding the optimal conditional distributions $p_1^*, \ldots, p_T^*$ inductively. □

4.C Proof of Corollary 4.4.1

Proof of Corollary 4.4.1. Lemma 17 shows that the value of the density function corresponding to choosing $u_t = u$ in the MEF is proportional to the measure of

$\mathsf{S}((u_{<t}, u))$, completing the proof of interpretability. According to the explicit expression in (4.8) of the MEF, the selected action u always ensures that $\mu((\mathsf{S}((u_{<t}, u))) > 0$ and therefore the set $\mathsf{S}((u_{<t}, u)$ is non-empty. This guarantees that the generated sequence u is always in S. □

4.D Proof of Corollary 4.6.1

Proof of Corollary 4.6.1. The explicit expression in Lemma 17 ensures that whenever $p_t^*(u_t|u_{<t}) > 0$, then there is always a feasible sequence of actions in $\mathsf{S}(u_{<t})$. Now, if the tuning parameter $\beta_t > 0$, then the optimization (4.13) guarantees that $p_t^*(u_t|u_{<t}) > 0$ for all $t \in [T]$; otherwise, the objective value in (4.13) is unbounded. Corollary 4.4.1 guarantees that for any sequence of actions $u = (u_1, \ldots, u_T)$, if $p_t^*(u_t|u_{<t}) > 0$ for all $t \in [T]$, then $u \in \mathsf{S}$. Therefore, the sequence of actions u given by the PPC is always feasible. □

4.E Proof of Theorem 18

Proof of Lemma 18. We note that the offline optimization (4.2a)–(4.2d) is equivalent to

$$\inf_{u \in \mathsf{U}^T} \sum_{t=1}^{T} c_t(u_t) - \beta \log q(u) \tag{4.18}$$

for any $\beta > 0$ and $q(u)$ is a uniform distribution on S:

$$q(u) := \begin{cases} 1/\mu(\mathsf{S}) & \text{if } u \in \mathsf{S} \\ 0 & \text{otherwise} \end{cases}$$

where $\mu(\cdot)$ is the Lebesgue measure. Further, decomposing the joint distribution $q(u) = \prod_{t=1}^{T} p_t^*(u_t|u_{<t})$ into the conditional distributions given by (4.7a)-(4.7c), the objective function (4.18) becomes

$$\begin{aligned} &\sum_{t=1}^{T} c_t(u_t) - \beta \log \left(\prod_{t=1}^{T} p_t^*(u_t|u_{<t}) \right) \\ &= \sum_{t=1}^{T} \left(c_t(u_t) - \beta \log p_t^*(u_t|u_{<t}) \right), \end{aligned}$$

which implies the lemma. □

4.F Proof of Theorem 4.6.1

Proof. Define the following optimal *cost-to-go function*, which computes the minimal cost given a subsequence of actions:

$$
\begin{aligned}
V_t^{\mathsf{OPT}}(u_{<t}) &:= \min_{u_{t:T}\in \mathsf{U}^{T-t+1}} \left(\sum_{\tau=t}^{T} c(u_\tau) - \beta \log p_t^*(u_{t:T}|u^*_{t-k:t-1}) \right) \\
&= \min_{u_t \in \mathsf{U}} \Bigg(c(u_t) - \log p_t^*(u_t|u^*_{t-k:t-1}) + \\
&\qquad \min_{u_{t+1:T}\in \mathsf{U}^{T-t}} \Big(\sum_{\tau=t+1}^{T} c(u_\tau) - \log p_{t+1}^*(u_{t+1:T}|u^*_{t-k:t}) \Big) \Bigg) \\
&= \min_{u_t \in \mathsf{U}} \left(c(u_t) - \log p_t^*(u_t|u^*_{t-k:t-1}) + V_{t+1}^{\mathsf{OPT}}(u_{\le t}) \right).
\end{aligned}
$$

Let $\mu(\cdot)$ denote the Lebesgue measure. Based on the definition of the optimal cost-to-go functions defined above and applying Lemma 18, we obtain the following expression of the optimal action u_t^* at each time $t \in [T]$:

$$
\begin{aligned}
u_t^* &= \arg\min_{u_t \in \mathsf{U}} \left(c_t(u_t) - \beta \log p_t^*(u_t|u^*_{t-k:t-1}) + V_{t+1}^{\mathsf{OPT}}\left(u_{t-k+1:t}\right) \right) \\
&= \arg\min_{u_t \in \mathsf{U}} \Bigg(c_t(u_t) - \beta \log \frac{\mu\left(\mathsf{S}(u^*_{t-k:t-1}, u_t)\right)}{\mu\left(\mathsf{S}(u^*_{t-k:t-1})\right)} \\
&\qquad + \min_{u_{t+1:T}} \Big(\sum_{\tau=t+1}^{T} \big(c(u_\tau) - \beta \log \frac{\mu\left(\mathsf{S}(u^*_{t-k:t-1}, u_{t:\tau})\right)}{\mu\left(\mathsf{S}(u^*_{t-k:t-1}, u_{t:\tau-1})\right)} \big) \Big) \Bigg) \\
&= \arg\min_{u_t \in \mathsf{U}} \Bigg(c_t(u_t) - \beta \log \frac{\mu\left(\mathsf{S}(u^*_{t-k:t-1}, u_t)\right)}{\mu\left(\mathsf{S}(u^*_{t-k:t-1})\right)} \\
&\qquad + \min_{u_{t+1:T}} \Big(\sum_{\tau=t+1}^{T} c(u_\tau) + \beta \log \frac{\mu\left(\mathsf{S}(u^*_{t-k:t-1}, u_t)\right)}{\mu\left(\mathsf{S}(u^*_{t-k:t-1}, u_{t:T})\right)} \Big) \Bigg),
\end{aligned}
$$

which implies (4.19) below

$$
u_t^* = \arg\min_{u_t \in \mathsf{U}} \Bigg(c_t(u_t) + \underbrace{\min_{u_{t+1:T}} \left(\sum_{\tau=t+1}^{T} c(u_\tau) - \log \mu\left(\mathsf{S}(u^*_{t-k:t-1}, u_{t:T})\right) \right)}_{f(u_t)} \Bigg) \tag{4.19}
$$

and when $u_{<t} = u^*_{<t}$, the solution of the PPC in Algorithm 6 satisfies

$$\begin{aligned} u_t &= \arg\min_{u_t \in \mathsf{U}} \left(c_t(u_t) - \beta \log p_t(u_t | u^*_{t-k:t-1}) \right) \\ &= \arg\min_{u_t \in \mathsf{U}} \left(c_t(u_t) - \beta \log \frac{\mu\left(\mathsf{S}(u^*_{t-k:t-1}, u_t)\right)}{\mu\left(\mathsf{S}(u^*_{t-k:t-1})\right)} \right) \\ &= \arg\min_{u_t \in \mathsf{U}} \left(c_t(u_t) + \beta \underbrace{\log\left(1/\mu\left(\mathsf{S}(u^*_{t-k:t-1}, u_t)\right)\right)}_{g(u_t)} \right). \end{aligned} \tag{4.20}$$

Since the cost functions $c_t(u)$ and the measure $\log(1/\mu(\mathsf{S}(u)))$ are strictly convex, the inner minimization in (4.19) is a convex minimization and hence $f(u_t)$ is convex. Therefore, u^*_t in (4.19) is unique. Denoting by c' and f' the corresponding derivatives of a given cost function c and the function f defined in (4.19), we have

$$c'_t(u^*_t) + f'(u^*_t) = 0.$$

Furthermore, the unique solution of the PPC scheme satisfies

$$c'_t(u_t) + \beta g'(u_t) = 0$$

where g' is the derivative of the function g defined in (4.20). Choosing $\beta = b_t = f'(u^*_t)/g'(u^*_t)$ implies that $u_t = u^*_t$ for all $t \in [T]$. □

Chapter 5

LEARNING-BASED PREDICTIVE CONTROL: REGRET ANALYSIS

[1] Tongxin Li, Yue Chen, Bo Sun, Adam Wierman, and Steven H. Low. Information aggregation for constrained online control. 5(2), 2021. URL https://doi.org/10.1145/3460085.

Continuing from the formulation provided in Chapter 4, in this chapter, we consider the case when predictive MEF (see Definition 4.4.1) is available and present the corresponding regret analysis of the penalized predictive control as a learning-augmented decision-making algorithm.

5.1 Introduction

The use of online learning methods for controlling dynamical systems has captured increasing attention from both the learning and control communities. Significant effort has been made to design online optimal controllers using tools from machine learning in a variety of contexts in recent years [104–110]. One general dynamic model of particular interest for many applications is the following, which has *time-varying* and *time-coupling* constraints:

$$x_{t+1} = f_t(x_t, u_t), \quad x_t \in \mathcal{X}_t\left(\mathbf{x}_{<t}, \mathbf{u}_{<t}\right), \; u_t \in \mathcal{U}_t\left(\mathbf{x}_{<t}, \mathbf{u}_{<t}\right), \tag{5.1}$$

where the deterministic function f_t represents the transition of the state x_t, and u_t is the control or action determined by the controller. Crucially, the constraints $\mathcal{X}_t\left(\mathbf{x}_{<t}, \mathbf{u}_{<t}\right)$ and $\mathcal{U}_t\left(\mathbf{x}_{<t}, \mathbf{u}_{<t}\right)$ may change over time and they depend on the past history of states $\mathbf{x}_{<t} = (x_1, \ldots, x_{t-1})$ and actions $\mathbf{u}_{<t} = (u_1, \ldots, u_{t-1})$. At each time t, the action u_t chosen by the online controller incurs a cost $c_t(u_t)$ and the goal is to minimize the cumulative costs without violating the dynamical constraints.

Designing controllers for the general constrained dynamics in (5.1) is challenging and, as a result, traditional online optimization models often adopt simplified versions of (5.1), such as unconstrained optimization [111–113] or time-invariant constraints [114, 115] that are known a priori. More specifically, previous studies in online control and online optimization mostly focus on specific forms of constraints or costs

depending on different applications, such as switching costs [116–119], ramping constraints [117, 120], polytopic constraints [121], time-varying memoryless cumulative constraints [104, 107, 109, 122, 123], convex loss functions with memory [52, 124–126] or inventory constraints [118, 127]. Within this literature, the goal is to derive policies with either small regret [104, 107, 116, 118, 128, 129] or competitive ratio [117, 119, 120, 126, 130]. However, there has been little progress deriving policies with small regret/competitive ratio under general forms of constraints, like in (5.1).

In particular, our work is motivated by settings where transmitting the constraints precisely is hard, either due to complexity or privacy concerns. These issues often arise in a two-controller system where a *local controller* governs large-scale infrastructure consisting of many controllable devices and a *central controller* operates remotely. Therefore, the control system often contains two controllers: a central controller that wishes to optimize costs but is far away from a large fleet of controllable devices and a *local controller* that has direct access to the controllable devices, whose dynamics can be modeled by (5.1). In many situations, full information about the local controllers' dynamics and constraints is not available to the central controller and the local controller cannot access the system's costs. In such settings, the central and local controllers each have part of the information needed to control the whole dynamical system online. The task of designing controllers is therefore made even more challenging than the single controller case.

A motivating example for our work is the coordination between a system operator (central controller) and an aggregator (local controller) in large-scale electric vehicle (EV) charging systems, as discussed in [1, 24, 131, 132]. In a smart grid, it is desirable to increase the ability of the system to provide flexibility via distributed energy resources (such as electric vehicles). Aggregators have emerged as dominate players for coordinating these resources and they are able to provide coordination among large pools of DERs and then give a single point of contact for independent system operators (ISOs) to call on for obtaining the aggregate flexibility of DERs [24]. This enables ISOs to minimize cost and respond to unexpected fluctuations of renewables. An example of a system operator is the California ISO, that takes charge of managing and balancing various controllable loads and providing auxiliary services, e.g., demand response, energy storage and flexibility reserves to enhance the system stability and quality. In the case of managing an EV charging garage, on the one hand the aggregator cannot solve the problem independently because it does not have cost

information (since the costs are often sensitive and only of the operator's interests) from the operator and even if the aggregator could, it may not have enough power to solve an optimization to obtain an action, such as deciding the substation power level of the EV charging garage. On the other hand, the operator has to receive flexibility information from the aggregator in order to act. Well-known methods in learning or control cannot be used for this problem directly. From a learning perspective, the aggregator cannot simply use reinforcement learning and transmit a learned Q-function or an actor-critic model to the operator because the aggregator does not know the costs. From a control perspective, although model predictive control (MPC) is widely used for EV charging scheduling in practical charging systems [1, 96], it requires precise state information of electric vehicle supply equipment (EVSE). Thus, to solve the induced MPC problem, the system operator or aggregator needs to solve an online optimization at each time step that involves hundreds or even thousands of variables. This not just a complex problem, but the state information of the EVSE is potentially sensitive. This combination makes controlling subsystems using precise information impractical for a future smart grid [24]. Note that there are a wide variety of other situations that face similar challenges, including data center scheduling [107] and fog computing [105].

The complexities illustrated by the examples above highlight the necessity for system operators to use aggregate information about the constraints of the agents being controlled. However, controlling a dynamical system such as (5.1) using only aggregate information adds to the difficult of the design of the controller. Specifically, the two-controller system is now faced with two design tasks: (i) *summarize*–design a form of the aggregate signal summarizing the constraints and dynamics of the controllable devices for a local controller, and (ii) *optimize*–design an central controller that uses the aggregate signal to minimize the cumulative cost while satisfying the constraints of the local controller.

Contributions. In this work, we consider a system formed by a central controller and a local controller and propose a design of aggregate feedback that integrates into a novel predictive control policy for online control of a system with time-varying and time-coupling constraints. Importantly, the aggregate feedback can be efficiently approximated via a data-driven approach. Our main results bound the dynamic regret of our policy and show that it achieves near-optimal performance. In more detail, our main contributions are four-fold.

First, we design an approach for information aggregation, termed maximum entropy feedback (MEF). MEF, introduced in Section 5.3, is defined as a density function on the action space. We introduce a control policy, *penalized predictive control* (PPC) (see Section 4.6), that incorporates MEF as a penalty term into an MPC-like policy. This technique for incorporating aggregate feedback is a novel design approach that we expect to have applicability beyond the context of this work. We also introduce a method to approximate the MEF using model-free reinforcement learning.

Second, denoting by d the diameter of the action space U, T the number of total time steps and w the number of predictions available, we show that the dynamic regret of any deterministic policy must satisfy a lower bound on $\mathsf{Regret}(\mathbf{u}) = \Omega\left(d(T-w)\right)$ for any feasible sequence of actions $\mathbf{u}$ generated by the deterministic policy, even if it has full information of the constraints. Note that it is well-known that, in the worst case, a sub-linear dynamic regret without the use of predictions is impossible (cf. [133]). Therefore, conditions on the constraints and predictions are necessary to obtain a sub-linear dynamic regret.

Third, we introduce a new assumption, termed *causally invariance* (see Definition 5.5.1) on the set of feasible actions. The condition holds quite generally, including in applications with inventory constraints (Example 4) and tracking problems (Example 5). We show that when the constraints are causally invariant (see Definition 5.5.1), the *dynamic regret* of PPC is bounded from above by

$$\mathsf{Regret}(\mathbf{u}) = O\left(dT\left(\frac{\delta \log \lambda}{\sqrt{w}} + \frac{\sqrt{\delta}}{w^{1/4}}\right)\right)$$

where $\mathbf{u}$ is the sequence of actions generated by PPC, $\lambda, \delta > 0$ are parameters of the causal invariance assumption, w is the prediction window size, and d is the diameter of the action space. In particular, when $w = \omega(1)$, even with aggregate feedback information, PPC establishes a sub-linear (in T) bound on the dynamic regret. To the best of our knowledge, this is the first bound on dynamic regret for a policy that uses only aggregate feedback information in the context of time-varying and time-coupling constraints.

Related work. This model falls into the growing literature seeking to understand the role of information and feedback in online decision-making. In this line of work, the online controller does not know (at least) one of the objective function, dynamics or constraints in advance, and therefore, feedback containing information about the objective function, dynamics and constraints becomes necessary.

When the cost functions are unknown, using a semi-definite relaxation for an online controller, [54] shows a $O(\sqrt{T})$ regret where T is the length of the time horizon for quadratic costs and stochastic noise. It is well known that without the curvature assumptions, $O(\sqrt{T})$ regret is tight [134]. Similarly, in [52], $O(\sqrt{T})$ regret is proved for convex costs with adversarial noise. Recently, with the help of online learning, [125] considers a linear dynamical system under adversarially changing strongly convex cost functions, which includes the Kalman filter and the linear quadratic regulator. Assuming the transition dynamics are known, the authors in [125] obtain a logarithmic upper bound $O(\mathsf{poly}(\log T))$ on the regret. When the dynamics are not known, non-linear control with safety constraints have been considered in [27], wherein an algorithm is introduced to safely learn about the dynamics modeled by a Gaussian process.

When the constraints and costs are not known a priori, results are less general and often hold for specific forms of constraints. To help the online controller be aware of the constraint information, feedback is necessary. Two widely adopted feedback signals are bandit feedback [104, 105], which reveals the values of the objective and constraint functions with certain chosen actions, and gradient feedback [106–109], which further offers the gradient of the unknown functions. To briefly mention some of the results, in [105], with bandit feedback, the dynamic regret is bounded by $o(T)$. In [104], the dynamic regret is bounded by $O(\sqrt{T\Delta(T)})$ and the constraint violations are bounded by $O(T^{3/4}\Delta(T)^{1/4})$ where $\Delta(T)$ is the "drift" of the sequence of optimal actions. In [110], $O(\sqrt{T})$ regret is proven, and the results are extended to incorporate future information (predictions). In the case of stochastic long-term constraints, the authors in [107] achieve $O(\sqrt{T}\log T)$ regret and constraint violations with high probability. However, both bandit and gradient feedback are not designed to deal with time-coupling constraints and there are no results providing guaranteed performance for the general setting in (5.1). Indeed, for the case when the offline constraints on the actions $\mathbf{u} = (u_1, \ldots, u_T)$ are of the form $\sum_{t=1}^{T} g_t(u_t) \leq 0$, [135] shows that if the feedback at time t contains only information on the function g_t and a convex cost c_t, it is impossible to achieve both sub-linear regret and constraint violations even if the functions g_t and c_t are linear.

Finally, note that the literature described above typically compares algorithms with the best *fixed* decision in hindsight, with notable exceptions such as [105, 109], which shows sub-linear dynamic regret and constraint violations. This is in contrast to the considered work which compares to dynamic optimal decisions in hindsight.

Notation and Conventions. The (differential) entropy function is denoted by $\mathbb{H}(\cdot)$. Sequences of vectors are written as boldface letters, such as $\mathbf{u} = (u_1, \ldots, u_T)$ and u_t is a vector in an Euclidean space. To distinguish random variables and their realizations, we follow the convention to denote the former by capital letters (e.g., U and $\mathbf{U}$) and the latter by lower case letters (e.g., u and $\mathbf{u}$). We fix the base of the logarithms to be the natural number e, unless otherwise stated. The concatenation of two sequences $\mathbf{x}$ and $\mathbf{y}$ is denoted by $(\mathbf{x}, \mathbf{y})$. The ℓ_2 norm of a vector x is written as $||x||_2$. Let $\mu(\mathsf{S})$ be the Lebesgue measure of a measurable set S. The Hausdorff distance between two sets S_1 and S_2 in an Euclidean space with respect to the ℓ_2 norm is denoted as

$$d_{\mathsf{H}}(\mathsf{S}_1, \mathsf{S}_2) := \max \left\{ \sup_{x \in \mathsf{S}_1} \inf_{y \in \mathsf{S}_2} ||x - y||_2, \sup_{y \in \mathsf{S}_2} \inf_{x \in \mathsf{S}_1} ||x - y||_2 \right\}.$$

5.2 Model

We consider the deterministic dynamical system in (5.1) over a discrete time horizon $[T] := \{1, \ldots, T\}$ with time-varying and time-coupling constraints.

The dynamical system is governed by a *local controller* ψ, which manages a large fleet of controllable units. The collection of the states of the units is represented by x_t in a state space $\mathsf{X} \subseteq \mathbb{R}^n$.

There is a distant *central controller* π that communicates with the local controller. The central controller selects an action u_t at each time $t \in [T]$. The actions must be selected from a closed and bounded domain $\mathsf{U} \subseteq \mathbb{R}^m$. The initial point u_0 is assumed to be the origin without loss of generality.

Both the state and action at each time are confined by *safety constraints* that maybe time-varying and time-coupling, i.e., $x_t \in \mathcal{X}_t(\mathbf{x}_{<t}, \mathbf{u}_{<t})$ and $u_t \in \mathcal{U}_t(\mathbf{x}_{<t}, \mathbf{u}_{<t})$ for $t \in [T]$. For simplicity, we denote the *safety sets* $\mathcal{U}_t(\mathbf{x}_{<t}, \mathbf{u}_{<t})$ and $\mathcal{X}_t(\mathbf{x}_{<t}, \mathbf{u}_{<t})$ by $\mathcal{U}_t$ and $\mathcal{X}_t$ in future contexts. The central controller receives time-varying cost functions online from an external environment and each $c_t(\cdot) : \mathsf{U} \to \mathbb{R}_+$ only depends on the action u_t chosen by the central controller. We assume that the local controller does not know the costs and has to choose the action given by the central controller and the central controller cannot access the constraints directly, but some information about the constraints, summarized as some feedback p_t, can be transmitted during the control (more details of p_t will be given in Section 5.3). In settings with constraints, predictions are crucial to maintaining feasibility and reducing costs. Thus, in this work, we suppose the online controller has (perfect) predictions of the cost functions

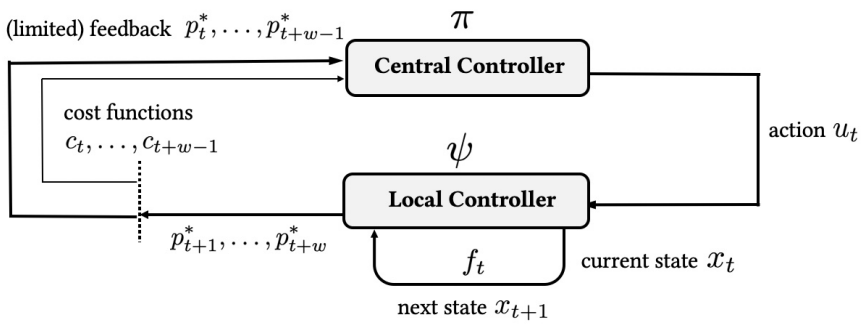


Figure 5.2.1: Closed-loop interaction between a central controller and a local controller.

and feedback functions of the current and the next w time slots. We formally define the predictions available in Section 5.3. The goal of an online control policy in this setting is to make the local and central controllers jointly minimize a cumulative cost $C_T(\mathbf{u}) := \sum_{t=1}^{T} c_t(u_t)$ while satisfying (5.1). Our system model is shown in Figure 5.2.1.

Throughout this chapter, we make the following assumptions on the model.

Assumption 5. *The dynamic $f_t(\cdot,\cdot) : \mathcal{X}_t \times \mathcal{U}_t \to \mathcal{X}_{t+1}$ is a Borel measurable function for $t \in [T]$.*

Assumption 6. *The action space $\mathsf{U} \subseteq \mathbb{R}^m$ is closed and bounded.*

Assumption 7. *The safety sets $\{\mathcal{U}_t : t \in [T]\}$ and $\{\mathcal{X}_t : t \in [T]\}$ are Borel sets in $\mathbb{R}^m$ and $\mathbb{R}^n$. Furthermore, the safety sets are atoms, i.e., $\mu(\mathcal{X}_t) > 0$ and $\mu(\mathcal{U}_t) > 0$ for all $t \in [T]$ if $\mathcal{X}_t, \mathcal{U}_t \neq \emptyset$.*

Additionally, we adopt the following smoothness condition for cost functions.

Assumption 8. *For each $t \in [T]$, the cost function $c_t(\cdot) : \mathsf{U} \to \mathbb{R}_+$ is Lipschitz continuous. We assume that there exists a Lipschitz constant $L_{\mathsf{c}} > 0$ such that*

$$|c_t(u) - c_t(v)| \leq L_{\mathsf{c}} \, ||u - v||_2 \quad \textit{for all } u, v \in \mathsf{U} \textit{ and } t \in [T].$$

Dynamic Regret

The focus of this work is the analysis of worst-case bounds on the *dynamic regret* [136, 137], which is the difference between the cost of the algorithm and that of the offline optimal decision. Formally, the offline optimal cost with full information of the

functions and the safety sets is defined as:

$$C_T^* := \inf_{\mathbf{u}} C_T(\mathbf{u}) \tag{5.2a}$$

$$\text{subject to (5.1)}, \forall t \in [T], \tag{5.2b}$$

and the dynamic regret, $\mathsf{Regret}(\mathbf{u})$, is:

$$\mathsf{Regret}(\mathbf{u}) := \sup_{\mathbf{c}\in\mathsf{C}} \sup_{\mathbf{f}\in\mathsf{F}} \sup_{(\mathsf{U},\mathsf{X})\in\mathcal{I}} C_T(\mathbf{u}) - C_T^* \tag{5.3}$$

where $\mathbf{u}$ is the sequence of actions generated by the online policy π, $\mathbf{f} := (f_1, \ldots, f_T)$ denotes a sequence of dynamics chosen from a set of Borel measurable functions F satisfying Assumption 5, $\mathbf{c} := (c_1, \ldots, c_T)$ denotes a sequence of cost functions chosen from the set of all Lipschitz continuous functions C; $\mathsf{U} := (\mathcal{U}_1, \ldots, \mathcal{U}_T)$ and $\mathsf{X} := (\mathcal{X}_1, \ldots, \mathcal{X}_T)$ are the collections of safety constraints. It is important to note that without any restrictions on U and X, $\mathsf{Regret}(\mathbf{u})$ can be no better than $\Omega(T)$ for any deterministic online policy π, even with predictions (see Theorem 5.5.1 for more details). Therefore, the focus of this work is to find conditions on (U, X) so that given enough predictions, the regret can be bounded by a sub-linear (in T) function. We denote by $\mathcal{I}$ the domain of safety constraints (U, X) satisfying certain conditions depending on the contexts and will formally state the conditions (such as the *causal invariance criterion* used in Theorem 5.5.3) in the theorem statements. In the remainder of the chapter, we sometimes write $\mathsf{Regret}(\pi)$, in replacement of $\mathsf{Regret}(\mathbf{u})$ defined in (5.3), with $u_t = \pi(\mathbf{c}_{t:t+w-1}, p_{t:t+w-1}(\cdot|\mathbf{u}_{<t}))$. It is worth mentioning that, to the best of our knowledge, there is no existing bound on the dynamic regret defined in (5.3) in the current setting, nor are there existing results for a similar worst-case metric called *competitive ratio* [113, 117, 118, 130, 138].

5.3 Information Aggregation

We first present the design of feedback in the two-controller system.

Limited Feedback Information

A distinctive feature of the work presented in this chapter is the question of how to design online controllers when limited information is available. Limitations on the information available may occur because (1). the central controller is distant from the local controller and sending the full information about the dynamical system renders communication issues, (2). the central controller may have other tasks running in parallel and its computational power is limited and (3). the size n of the state space is much larger than the size m of the action space, preventing the central controller from

accessing the full state $\{x_t \in \mathbb{R}^n : t \in [T]\}$ and the safety sets $\{\mathcal{U}_t : t \in [T]\}$ and $\{\mathcal{X}_t : t \in [T]\}$ from the local controller. For example, consider an electric vehicle charging station described earlier in Section 4.3, there may be hundreds or even thousands of electric vehicle chargers. Each of the chargers has a state and the overall state vector has a high dimension. However, there are often only a few choices of the sub-station power levels to be chosen by a remote system operator (see Section 4.7 for more details about a realistic EV charging setting). (4). Alternatively, limitations may result because of privacy concerns, e.g., EV owners do not want to directly share their charging session information with a third-party, making transmission of the exact constraints to the central controller undesirable.

To capture such limitations, we consider a setting where the central controller receives a simplified feedback signal summarizing the state and safety sets. These signals have domains over the action space U. The regime of interest is when the action space is much smaller than the state space, so sending the density functions requires much less communication resources than sending the states and constraints. While this definition is abstract, recall that we give a concrete example in the case of power systems (see Section 4.3). In the following, we formally state the form of the feedback and make the following model assumption on the feedback sent from the local controller to the central controller.

Assumption 9. *At each time $t \in [T]$, the local controller is allowed to send a feedback (density) function whose domain is the action space U, denoted by $p_t(\cdot) : \mathsf{U} \to [0, 1]$ to the central controller.*

Remark 6. At first glance transmitting a density function from a local controller to a central controller seems challenging. In practice, given the regime that we are interested in when the action space dimension $|\mathsf{U}|$ is much smaller than the state space dimension X, sending a quantized density function simplifies the communication compared to sending the whole state. Further, in certain applications the action space is discrete, e.g., the EV charging application described in Section 4.3, so p_t becomes a probability vector whose length is much smaller than the state vector. More discussion about learning the feedback function can be found in Section 4.5. Note that the online control cannot be directly implemented in a local controller, because the central controller may not want the local controller to know the exact process of how the actions are generated. When generalizing to a multi-controller system, the central controller needs to take into account of information from other

aggregators as well. The discussion of this setup is beyond the scope of this article and would be an interesting extension.

Since the dynamic system in (5.1) has memory, the feedback function also depends on previous actions. To be more precise, at each time $t \in [T]$, a (conditional) density function denoted by $p_t(\cdot|\mathbf{u}_{<t}) : \mathsf{U} \to [0, 1]$, given past actions $\mathbf{u}_{<t}$ is sent from the local controller to the central controller. The density function contains only information about the constraints. A natural question is how to design such feedback, and we discuss this question in Section 5.3.

Predictions

For general dynamics such as (5.1), it is well-known that, in the worst case, a sub-linear dynamic regret without the use of predictions is impossible (cf. [133]). In many applications, while the cost functions are not known a priori, predictions of future cost functions are available. The question of how to make use of such predictions for online control and optimization has received considerable attention in recent years, e.g., [113, 116, 130, 138]. Formally, at time $t \in [T]$, the cost functions $c_t, \ldots, c_{t+w-1}$ are given to the central controller from an oracle and the joint feedback function $p_{t:t+w-1}(\cdot|\mathbf{u}_{<t}) : \mathsf{U}^w \to [0, 1]$, as a density function on the subsequence of actions $\mathbf{u}_{t:t+w-1}$, are sent to the central controller from the local controller where $w > 0$ is an integer denoting the *prediction window size*. Let $\mathsf{C}_{<t+w}$ and $\mathsf{P}_{<t+w}$ be the sets of cost functions and feedback functions received by the central controller at time $t \in [T]$. The goal of the central controller is to design an *online policy* $\pi : \mathsf{C}_{<t+w} \times \mathsf{P}_{<t+w} \to \mathsf{U}$ that generates actions. This model for predictions that we consider is standard, and has been adopted in [113, 116, 138]. While the assumption that predictions are perfect is overly optimistic, the insights derived typically extend to settings with inexact predictions, albeit with considerable technical effort, e.g., [139–141]. We denote by $\psi : \mathsf{X} \to \mathsf{P}_{t:t+w-1}$ an *information aggregation function* that outputs a sequence of feedback predictions and $\mathsf{P}_{t:t+w-1}$ is the set of feedback functions $p_t, \ldots, p_{t+w-1}$.

Aggregation Feedback: Maximum Entropy Feedback (MEF)

A key feature of our model is the aggregate feedback signal p_t provided to the controller, motivated by the problem formulated in Chapter 4. Such an aggregate signal p_t is needed in many situations for a variety of reasons. One prominent motivation is that it can be hard to transmit complicated constraints precisely from large-scale controllable units to the agent as a result of communication limitations.

Another common motivation is that the information structure among the agent and units is asymmetric, and the dynamical operating parameters of each controllable unit are private. As a result, it is undesirable and impractical to ask for exact constraints and system states. To ensure the feasibility of the chosen action profile while avoiding the leakage of private parameters, one compromise is to aggregate the necessary information via a feedback signal p_t, defined earlier in Assumption 9. In the following we consider a special design of p_t.

Note that the design of an aggregate feedback signal must balance possibly competing goals. First, it must contain the information about *what actions are feasible*. Second, it must contain information about the *future impact* of the actions to be selected on the system's feasibility. Finally, it must be compact, leaking as little individually identifiable information as possible.

Before we proceed to the details of our proposed design, we first introduce some useful notation.

Set of all feasible actions S. To unify notation, we begin with defining a set $f^{-1}(\mathcal{X}_t)$ denoting the inverse image of the safety set $\mathcal{X}_t$ for actions: $f^{-1}(\mathcal{X}_t)(\mathbf{u}_{<t}) := \{u \in \mathbb{R}^m : f(x_{t-1}, u) \in \mathcal{X}_t\}$. The inverse image $f^{-1}(\mathcal{X}_t)$ depends only on the past actions $\mathbf{u}_{<t}$ since the states $\mathbf{x}_{<t}$ are determined by $\mathbf{u}_{<t}$ via the dynamics in (5.1). Note that $\mathcal{X}_t$ and the dynamic f are Borel measurable by Assumption 5 and 7. Therefore the inverse image $f^{-1}(\mathcal{X}_t)$ is also a Borel set, implying that the intersection $\mathcal{U}_t \bigcap f^{-1}(\mathcal{X}_t)$ is also Borel measurable. Denote by

$$\mathsf{S} := \left\{\mathbf{u} \in \mathsf{U}^T : u_t \in \mathcal{U}_t \bigcap f^{-1}(\mathcal{X}_t), \forall t \in [T]\right\}$$

the non-empty Borel measurable set of all *feasible* sequences of actions satisfying the constraints in (4.7b). Since the actions space U is bounded, the set $\mathsf{S} \subseteq \mathsf{U}^T$ is also bounded.

Set of all subsequent feasible actions S_k. We denote the set of subsequent feasible trajectories by:

$$\mathsf{S}_k(\mathbf{u}_{\leq t}) := \left\{\mathbf{v}_{t+1:t+k} \in \mathsf{U}^k : \mathbf{v}_{\leq t} \equiv \mathbf{u}_{\leq t}; \mathbf{v} \in \mathsf{S}\right\}, \tag{5.4}$$

which consists of all feasible k actions at time $t+1, \ldots, \min\{t+k, T\} \in [T]$, given the past actions $\mathbf{u}_{\leq t}$. For the case when all future actions are considered, we simplify $\mathsf{S}_k(\mathbf{u}_{\leq t})$ as $\mathsf{S}(\mathbf{u}_{\leq t})$.

The core of our design is the *maximum entropy feedback* (MEF) presented in Chapter 4 (Definition 4.4.1), which provides a way for summarizing the time-varying

and coupling safety sets $\{\mathcal{U}_t \bigcap f^{-1}(\mathcal{X}_t) \subseteq \mathbb{R}^m : t \in [T]\}$. The intuition behind our definition is that the conditional density function $p_t(x_t|\mathbf{x}_{<t})$ encapsulates the resulting future flexibility of the constraints if the agent chooses x_t as the action at time t, given the previous actions up to time $t-1$. The sum of the conditional entropy of p_t is thus a measure of the information in p_t. This suggests choosing a conditional density function p_t that maximizes its conditional entropy. The MEF possesses some nice properties as shown in Corollary 4.4.1.

It is remarkable that the MEF only depends on the constraints, but not the costs. Further, the theoretical definition we present in (4.7a)-(4.7b) involves the offline information of S, the set of all *feasible* sequences of actions. This leads to computational difficulties; however learning techniques can be used to generate the MEF, as we have described in Section 4.5 in Chapter 4.

5.4 Penalized Predictive Control via Predicted MEF

In this section, with the two-controller system described in Section 5.2 and the feedback and predictions defined in Section 5.3, we revisit the penalized predictive control (PPC) scheme in Section 4.6 and formally present it in a closed-loop control framework with predicted maximum entropy feedback.

Key Idea: Maximum Entropy Feedback is a Penalty Term for the Offline Optimal

When first seeing the definition of MEF it is not immediately clear why it is useful in the context of predictive control. To make that clear, in this section we highlight the key idea in the design of our controller–MEF can act as an effective penalty term in the offline optimization problem. More specifically, there is in general a trade-off between ensuring future flexibility and minimizing the current system cost in predictive control. The action u_t guaranteeing the maximal future flexibility, i.e., having the largest $p_t^*(u_t|\mathbf{u}_{<t})$ may not be the one that minimizes the current cost function c_t and vice versa. Therefore, we need to design an online algorithm for the central controller that balances the MEF and the cost functions.

To further illustrate this point, note that Corollary 4.4.1 guarantees that the online agent can always find a feasible action $\mathbf{u} \in \mathsf{S}$. Indeed, knowing the MEF p_t^* for every $t \in [T]$ is equivalent to knowing the set of all admissible sequences of actions S. Using this observation, the constraints (5.2b) in the offline optimization can be rewritten as a penalty in the objective of (5.2a). Formally, the offline optimization can be recast as the following.

Lemma 19. *The offline optimization* (5.2a)-(5.2b) *is equivalent to the following unconstrained minimization for any* $\beta > 0$*:*

$$\inf_{\mathbf{u}\in\mathsf{U}^T} \sum_{t=1}^{T} \left(c_t(u_t) - \beta \log p_t^*(u_t|\mathbf{u}_{<t})\right). \tag{5.5}$$

This draws a clear connection between MEF and the offline optimal, which we exploit in the design of an online controller in the next section.

Proof. Recall that as shown in Lemma 18, the offline optimization (5.2a)–(5.2b) is equivalent to

$$\inf_{\mathbf{u}\in\mathsf{U}^T} \sum_{t=1}^{T} c_t(u_t) - \beta \log q(\mathbf{u}) \tag{5.6}$$

for any $\beta > 0$ and $q(\mathbf{u})$ is a uniform distribution on S:

$$q(\mathbf{u}) := \begin{cases} 1/\mu(\mathsf{S}) & \text{if } \mathbf{u} \in \mathsf{S} \\ 0 & \text{otherwise} \end{cases}.$$

Further, decomposing the joint distribution $q(\mathbf{u}) = \prod_{t=1}^{T} p_t^*(u_t|\mathbf{u}_{<t})$ into the conditional distributions given by (4.7a)-(4.7b), the objective function (5.6) becomes

$$\sum_{t=1}^{T} c_t(u_t) - \beta \log\left(\prod_{t=1}^{T} p_t^*(u_t|\mathbf{u}_{<t})\right) = \sum_{t=1}^{T} \left(c_t(u_t) - \beta \log p_t^*(u_t|\mathbf{u}_{<t})\right). \tag{5.7}$$

□

Algorithm: Penalized Predictive Control via Maximum Entropy Feedback

Our proposed design, termed Penalized Predictive Control (PPC), is a combination of Model Predictive Control (MPC), which is a competitive policy for online optimization with predictions, and the idea of using MEF as a penalty term. This design makes a connection between the MEF and the well-known MPC scheme. The MEF as a feedback function, only contains limited information about the dynamical system in the local controller's side. (It contains only the feasibility information of the current and future time slots, as explained in Section 5.3). The PPC scheme therefore is itself a novel contribution since it shows that, even if *only* feasibility information is available, it is still possible to incorporate the limited information to MPC as a *penalty term*. Moreover, this innovation allows PPC to achieve nearly optimal dynamic regret despite having only aggregate feasibility information about

constraints and dynamics, a setting where no prior algorithms have any provable guarantees.

Algorithm 7: Penalized Predictive Control (PPC)

Input : Sequential cost functions and MEF with a prediction window size w

Output Actions $\mathbf{u} = (u_1, \ldots, u_T)$

:

for $t = 1, \ldots, T$ **do**

  if $t \in \mathcal{I}$ **then**

    Choose an action u_t by minimizing over the next w time slots using (5.8)-(5.9)

  end

end

Return $\mathbf{u}$;

We present PPC in Algorithm 7, where we use the following notation. Let $t' := \min\{t + w - 1, T\}$. Define a set of time indices $\mathcal{I} := \{t \in [T] : t \equiv 1 \mod w\}$. Consider for any $t \in \mathcal{I}$:

$$\mathbf{u}_{t:t'} = \arg\inf_{\mathbf{u}_{t:t'}} \sum_{\tau=t}^{t'} \left(c_\tau(u_\tau) - \beta \log p^*_\tau(u_\tau|\mathbf{u}_{<\tau})\right) \tag{5.8}$$

$$\text{subject to } \mathbf{u}_{t:t'} \in \mathsf{U}^{t'-t+1} \tag{5.9}$$

where in above $\beta > 0$ is a *tuning parameter* in predictive control to trade-off the flexibility in the future and minimization of the current system cost. In the remainder of the chapter, we denote by π_{PPC} the online policy that uses the PPC scheme. Note that the algorithm presented in Algorithm 7 is a generalized version of the PPC scheme in Algorithm 6 (Chapter 4). It is reformulated using the two-controller model described in Section 5.2 and generalized to take feasibility predictions $\left(p^*_t, \ldots, p^*_{t'}\right)$.

Framework: Closed-Loop Control between Local and Central Controllers

Given the PPC scheme described in Section 5.4, we can now formally present our online control framework for the distant central controller and local controller (defined in Section 5.2). An overview is given in Algorithm 8, where π_{PPC} denotes a *PPC online policy* and ψ_{IA} a specific *information aggregation function* (see Section 4.5 for an example of learning-based ψ_{IA}). To the best of our knowledge, the work presented in this chapter is the first to consider such a closed-loop control framework with

limited information communicated in real-time between two geographically separate controllers seeking to solve an online control problem. We present the framework below.

At each time $t \in [T]$, the local controller first efficiently generates an estimated MEF sequence $\mathbf{p}_{t:t+w-1} := (p_t, \ldots, p_{t+w-1}) \in \mathsf{P}$ using an information aggregation function (for example, in Section 4.7 we use a reinforcement learning algorithm to train an information aggregation function). After receiving the MEF $\mathbf{p}_{t:t+w-1}$ and cost functions with predictions $\mathbf{c}_{t:t+w-1} = (c_t, \ldots, c_{t+w-1})$, the central controller uses the PPC scheme in Algorithm 7 to generate an action $u_t \in \mathsf{U}$ and sends it back to the local controller. The local controller then updates its state $x_t \in \mathsf{U}$ to a new state x_{t+1} based on the system dynamic in (5.1) and repeats this procedure again. Later in Section 5.5, we show that if the generated MEF is exact, i.e., $p_t = p_t^*$, for every $t \in [T]$, then feasibility can be ensured, i.e., $x_t \in \mathcal{X}_t$ and $u_t \in \mathcal{U}_t$ for every $t \in [T]$ and with assumptions, a sub-linear (in T) bound on the dynamic regret is possible with $w = \omega(1)$ using our online control scheme presented above.

Algorithm 8: Closed-loop online control framework

for $t = 1, \ldots, T$ **do**

Central Controller

Generate actions using the PPC:

$$u_t = \pi_{\text{PPC}}\left(\mathbf{c}_{t:t+w-1}, \mathbf{p}_{t:t+w-1}\right)$$
$$C_t = C_{t-1} + c_t(u_t)$$

Local Controller

Update system state:

$$x_{t+1} = f_t(x_t, u_t)$$

Compute estimated MEF:

$$\mathbf{p}_{t+1:t+w} = \psi_{\mathsf{IA}}(x_{t+1})$$

end

Return Total cost C_T;

5.5 Results

In this section, we state our main results, which guarantee feasibility and bound the dynamic regret of the PPC controller. Additionally, we present a lower bound for any online deterministic policy that have full information about safety constraints. This lower bound highlights the near optimality of PPC.

Feasibility

To begin, we highlight that PPC always yields a feasible trajectory of actions. To see this, recall that Corollary 4.4.1 shows that the MEF can guarantee feasibility if it is used appropriately. It implies that the feedback measures the volume of the set consisting of all sequences of actions that are feasible, conditional on the past actions taken by the agent. Therefore, as long as it is non-zero, the feasible set remains non-empty and there is always a feasible sequence of actions the agent can choose. The feedback is used as a penalty term in (5.8), therefore for any tuning parameter $\beta > 0$, the action $u_t \in \mathbb{R}^m$ selected by PPC always satisfies that $p_t^*(u_t|\mathbf{u}_{<t}) > 0$ and hence feasibility is guaranteed. Otherwise, suppose $p_t^*(u_t|\mathbf{u}_{<t}) = 0$ for the chosen action u_t, then the objective in (5.8) would blow up. Similar to what have been shown in Corollary 4.6.1, this is summarized in the following corollary, which shows that the actions determined by PPC are feasible, subject to the constraints in (5.1).

***Corollary* 5.5.1.** *For any predication window size $w \geq 1$, the sequence of actions $\mathbf{u} = (u_1, \ldots, u_T)$ generated by the PPC in* (5.8) *always satisfies $\mathbf{u} \in \mathsf{S}$.*

A Fundamental Limit

Before proceeding to the analysis of the dynamic regret of the PPC, we first consider a lower bound on $\mathsf{Regret}(\mathbf{u})$, for any sequence of actions $\mathbf{u}$ generated by a deterministic online policy.

Theorem 5.5.1 (Fundamental limit)**.** *For any sequence of actions $\mathbf{u} \in \mathsf{S}$ generated by a deterministic online policy that can access the safety sets $\{\mathcal{U}_t : t \in [T]\}$ and $\{\mathcal{X}_t : t \in [T]\}$ satisfying Assumption 6, for any $w \geq 1$, $\mathsf{Regret}(\mathbf{u}) = \Omega\left(d(T-w)\right)$ where $d := \mathsf{diam}(\mathsf{U}) := \sup\{||u - v||_2 : u, v \in \mathsf{U}\}$ is the diameter of the action space U, w is the prediction window size and T is the total number of time slots.*

The proof can be found in Appendix This result highlights that additional assumptions are needed if one hopes to obtain a positive result. The next section considers such an assumption. Note that the proof shows that it suffices to have a memory size of one, i.e., the safety sets $\mathcal{U}_t$ and $\mathcal{X}_t$ only depend on the action u_{t-1} and state x_{t-1}.

Causally Invariant Safety Constraints

Motivated by the lower bound in the previous section, we now introduce a particular class of safety constraints where it is possible to have better performance. The class is defined by a form of causal invariance that is intuitive and general. Specifically, we state a condition under which the sets of subsequent feasible actions do not change too much if the measures of the sets are close.

We define the following specific sequences of actions. Let $\overline{\mathbf{u}}_{\leq t} = (\overline{u}_1, \ldots, \overline{u}_t)$ be a subsequence of optimal actions that maximizes the volume of the set of feasible actions, defined as $\overline{\mathbf{u}}_{\leq t} := \arg\sup_{\mathbf{u} \in \mathsf{U}^t} \mu\left(\mathsf{S}(\mathbf{u})\right)$. With a slight abuse of notation, given $\mathbf{u}_{\leq t}$, define the length-k maximizing subsequence of actions as $\overline{\mathbf{u}}_{t+1:t+k} := \arg\sup_{\mathbf{u} \in \mathsf{U}^k} \mu\left(\mathsf{S}_k\left(\mathbf{u}_{\leq t}, \mathbf{u}\right)\right)$.

Definition 5.5.1 ((k, δ, λ)-causal invariance)**.** *The safety sets are (k, δ, λ)-causally invariant if there exist constants $\delta, \lambda > 0$ such that the following holds:*

1. *For all $t \in [T]$ and sequences of actions $\mathbf{u}_{\leq t}$ and $\mathbf{v}_{\leq t}$,*

$$d_{\mathsf{H}}\left(\mathsf{S}_k(\mathbf{u}_{\leq t}), \mathsf{S}_k(\mathbf{v}_{\leq t})\right) \leq \delta \left(\frac{|\mu(\mathsf{S}(\mathbf{u}_{\leq t})) - \mu(\mathsf{S}(\mathbf{v}_{\leq t}))|}{\mu\left(\mathcal{B}\right)} \right)^{1/((T-t)m)} \tag{5.10}$$

   where $\mathcal{B}$ denotes the unit ball in $\mathbb{R}^{m \times (T-t)}$.

2. *For all $t \in [T]$ and sequences of actions $\mathbf{u}_{\leq t}$,*

$$\frac{\mu\left(\mathsf{S}\left(\mathbf{u}_{\leq t}\right)\right)}{\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{\leq t}\right)\right)} \leq \lambda \left(\frac{\mu\left(\mathsf{S}\left(\left(\mathbf{u}_{\leq t}, \overline{\mathbf{u}}_{t+1:t+k}\right)\right)\right)}{\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{\leq t+k}\right)\right)} \right)^{\frac{T-t}{T-t-k}}. \tag{5.11}$$

Note that the definition of causal invariance is independent of the costs. Condition (1) is an inverse of the isodiametric inequality. It says that if the two subsequences of actions $\mathbf{u}_{\leq t}$ and $\mathbf{v}_{\leq t}$ do not affect the measure of the space of feasible sequences of actions too much, then the Hausdorff distance between the two sets $\mathsf{S}_k(\mathbf{u}_{\leq t})$ and $\mathsf{S}_k(\mathbf{v}_{\leq t})$ is also small. Condition (2) states that the ratio of the measure of a space of feasible sequences of actions, given any previous actions $\mathbf{u}_{\leq t}$, does not differ too much from the ratio of the measure of another space by fixing the next k actions to be those preserving the most future flexibility.

Definition 5.5.1 is general and practically applicable, as the following examples show.

***Example* 4** (Inventory constraints)**.** *Consider the following set of inventory constraints, which is a common form of constraints in energy storage and demand response problems [118, 127, 142]. The summation of the squares of the ℓ_2 norms of the actions is bounded from above by $\gamma > 0$, representing the limited resources of the system: $\sum_{t=1}^{T} ||u_t||_2^2 \leq \gamma$ where $u_t \in \mathbb{R}^m$. These inventory constraints are $(k, 1, 1)$-causally invariant for all $1 \leq k \leq T$.*

***Example* 5** (Tracking constraints)**.** *The following form of tracking constraints are common in situations where the system has a target "optimal" resource configuration that varies depending on the environment [116, 137, 143]. The agent seeks to track a fixed sequence of nominal actions $\mathbf{y} := (y_1, \ldots, y_T)$. The constraints are $\sum_{t=1}^{T} |u_t - y_t|^p \leq \sigma$ with $p \geq 2$ where $u_t, y_t \in \mathbb{R}$ and $\sigma > 0$ represents the system's adjusting ability. These tracking constraints are $(k, \frac{2}{\sqrt{\pi}}, 1)$-causally invariant for all $1 \leq k \leq T$.*

Additionally, to highlight the structures that are *not* causally invariant, we return to the construction of constraints used in the proof of the lower bound in Theorem 5.5.1.

***Example* 6** (Constraints in the lower bound (Theorem 5.5.1))**.** *Consider the following set of constraints defined in the proof of Theorem 5.5.1 for obtaining a lower bound $\Omega(d(T-w))$ on the dynamic regret: $\mathsf{S} := \{\mathbf{u} \in \mathsf{U}^T : u_t \in \mathcal{U}_t, \forall t \in [T]\}$ where*

$$\mathcal{U}_t := \begin{cases} \mathcal{B}(a/2), & \text{if } ||u_{t-1} - v_{t-1}||_2 \leq a \\ \mathsf{U} \backslash \mathcal{B}(a), & \text{if } ||u_{t-1} - v_{t-1}||_2 > a \end{cases} \tag{5.12}$$

for some $a > 0$. Suppose the action space $\mathsf{U} \subseteq \mathbb{R}^m$ is closed and bounded. These constraints are not causally invariant for all $k = \Omega(T)$. The reason why the causal invariance criterion is violated for the safety constraints in (5.12) *is that, in order to satisfy Condition (1) in* (5.10)*, it is necessary to have $\delta = \Omega(k) = \Omega(T)$, which is not a constant.*

Motivated by the example above, and mimicking the adversarial construction in Theorem 5.5.1, we can derive the following theorem and corollary, whose proofs are in Appendix 5.B.

Theorem 5.5.2 (Lower bound on Regret)**.** *If there exist a sequence of actions $\mathbf{v}_{\leq s} \in \mathsf{U}^s$, $1 \leq s \leq T-k$, $k \geq w$ with $k = \Omega(T)$ and some constant $\alpha > 0$ such that $d_{\mathsf{H}}(\mathsf{S}_k(\mathbf{u}_{\leq s}), \mathsf{S}_k(\mathbf{v}_{\leq s})) \geq \alpha dk$ for any $\mathbf{u}_{\leq s} \in \mathsf{U}^s$, then for any sequence of actions $\mathbf{u} \in \mathsf{S}$ generated by a deterministic online policy that can access the safety sets*

$\{\mathcal{U}_t : t \in [T]\}$ and $\{\mathcal{X}_t : t \in [T]\}$, $\mathsf{Regret}(\mathbf{u}) = \Omega\left(d(T-w)\right)$ where $\lambda(T)$ is a parameter that may depend on T; d is the diameter of the action space U; w is the prediction window size and T is the total number of time slots.

The theorem above states that, for any online policy knowing the safety sets $\{\mathcal{U}_t : t \in [T]\}$ and $\{\mathcal{X}_t : t \in [T]\}$ in advance, if there are two sets of length-k subsequences of actions $\mathsf{S}_k(\mathbf{u}_{\leq t})$ and $\mathsf{S}_k(\mathbf{v}_{\leq t})$ that are far from each other in terms of the Hausdorff distance, then a sub-linear regret is impossible. This highlights the necessity of the causal invariance condition. We make this explicit by further restricting the power of the online policy and assuming that it can only access the MEF $p_t, \ldots, p_{t+k-1}$ at time t, which yields the following impossibility result as a corollary of Theorem 5.5.2.

***Corollary* 5.5.2** (Lower bound on Regret)**.** *If there exist a sequence of actions $\mathbf{v}_{\leq t} \in \mathsf{U}^t$, $1 \leq s \leq T-k$, $k \geq w$ with $k = \Omega(T)$, a constant $\xi > 0$ and $\alpha = \Omega(T)$ such that*

$$\begin{aligned} \mu(\mathsf{S}(\mathbf{u}_{\leq s})) >& \xi \mu(\mathsf{S}(\overline{\mathbf{u}}_{\leq s})), \\ d_{\mathsf{H}}\left(\mathsf{S}_k(\mathbf{u}_{\leq s}), \mathsf{S}_k(\mathbf{v}_{\leq s})\right) \geq& \alpha \left(\frac{|\mu(\mathsf{S}(\mathbf{u}_{\leq s})) - \xi \mu(\mathsf{S}(\mathbf{v}_{\leq s}))|}{\mu\left(\mathcal{B}\right)} \right)^{1/((T-s)m)} \end{aligned}$$

for any $\mathbf{u}_{\leq s} \in \mathsf{U}^s$, then for any sequence of actions $\mathbf{u} \in \mathsf{S}$ generated by a deterministic online policy that can access the MEF $p_t, \ldots, p_{t+w-1}$ at each time $t \in [T]$, $\mathsf{Regret}(\mathbf{u}) = \Omega\left(d(T-w)\right)$ where $\lambda(T)$ is a parameter that may depend on T; d is the diameter of the action space U; w is the prediction window size and T is the total number of time slots.

The corollary above indicates that it is necessary to have an upper bound on the Hausdorff distance between $\mathsf{S}_k(\mathbf{u}_{\leq t})$ and $\mathsf{S}_k(\mathbf{v}_{\leq t})$ (such as Condition (1) and (2) in the causal invariance criterion) in order to make the dynamic regret sub-linear. In the next section, we show that, however, if the causal invariance criterion holds such that both δ and λ are constants, then the dynamic regret can be made sub-linear with sufficiently many predictions, even for arbitrary Lipschitz continuous cost functions.

Bounding the Dynamic Regret of PPC

We are now ready to present our main result, which bounds the dynamic regret by a decreasing function of the prediction window size under the assumption that the safety sets are causally invariant.

Theorem 5.5.3 (Upper bound on Regret)**.** *Suppose the safety sets are* (w, δ, λ)*-causally invariant. The dynamic regret for the sequence of actions* $\mathbf{u}$ *given by PPC is bounded from above by*

$$\mathsf{Regret}(\mathbf{u}) = O\left(dT\left(\frac{\delta \log \lambda}{\sqrt{w}} + \frac{\sqrt{\delta}}{w^{1/4}}\right)\right)$$

where d *denotes the diameter of the action space* U*,* w *is the prediction window size and* T *is the total number of time slots.*

This theorem implies that, with additional assumptions on the safety constraints, a *sub-linear* (in T) dynamic regret is achievable, given a sufficiently large prediction window size $w = \omega(1)$ (in T). Notably, Theorem 5.5.1 implies that for the worst-case costs and constraints, a deterministic online controller that has full information of the constraints suffers from a linear regret. As a comparison, Theorem 5.5.3 shows that, under additional assumptions, even if only aggregated information is available, PPC achieves a sub-linear regret. This does not contradict to the lower bound on the dynamic regret in Theorem 5.5.1, since if there is no regulation assumptions on the safety sets, in the worst-case, the Hausdorff distance $d_{\mathsf{H}}\left(\mathsf{S}_w(\mathbf{u}_{\leq t}), \mathsf{S}_w(\mathbf{v}_{\leq t})\right) = O(dw)$. This implies that a trivial upper bound of $O(dT)$ holds. Additionally, note that there is a trade-off between *flexibility* and *optimality* when selecting the tuning parameter $\beta > 0$. On the one hand, if the tuning parameter is too small, the algorithm is greedy and may suffer losses in the future; on the other hand, if the tuning parameter is too large, the algorithm is penalized by the feedback and therefore is far from being optimal.

A proof is presented in Appendix 5.B. Briefly, Theorem 5.5.1 is proven by optimizing β in an upper bound $\mathsf{Regret}(\mathbf{u}) = O\left(Td\left(\delta \log \lambda / w + d\delta\sqrt{w}/\beta + \beta/dw\right)\right)$, which holds for any sequence of actions $\mathbf{u}$ given by the PCC with any tuning parameter $\beta > 0$.

APPENDIX

5.A Explanation of Definition 5.5.1 and Two Examples

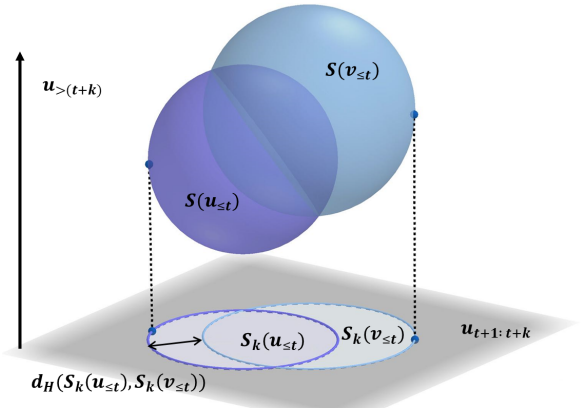


Figure 5.A.1: Graphical illustration of (5.10) in Definition 5.5.1.

Example 1: *Inventory constraints.*

Recall that the feasible set of actions for the inventory constraints is

$$\mathsf{S} = \left\{ \mathbf{u} \in \mathbb{R}^{T \times m} : \sum_{t=1}^{T} ||u_t||_2^2 \leq \gamma \right\}.$$

The sequence of actions $\overline{\mathbf{u}}_{\leq t}$ maximizing the size of the set of admissible actions is the all-zero vector. Hence,

$$\left(\frac{\mu\left(\mathsf{S}\left(\mathbf{u}_{\leq t}\right)\right)}{\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{\leq t}\right)\right)} \right)^{\frac{1}{T-t}} = \left(1 - \frac{\sum_{\tau=1}^{t} ||u_\tau||_2^2}{\gamma} \right)^{\frac{m}{2}} = \left(\frac{\mu\left(\mathsf{S}\left((\mathbf{u}_{\leq t}, \overline{\mathbf{u}}_{t+1:t+k})\right)\right)}{\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{<t+k}\right)\right)} \right)^{\frac{1}{T-t-k}}.$$

Therefore setting $\lambda = 1$, (5.11) is satisfied.

It remains to check the bound on the Hausdorff distance. Figure 5.A.1 shows the idea behind the definition. If the two sets $\mathsf{S}(\mathbf{v}_{\leq t})$ and $\mathsf{S}(\mathbf{u}_{\leq t})$ are close to each other, the Hausdorff distance of the projected sub-spaces $\mathsf{S}_k(\mathbf{u}_{\leq t})$ and $\mathsf{S}_k(\mathbf{v}_{\leq t})$ can also be bounded. For inventory constraints, this is indeed the case. For all $1 \leq k \leq T$, sequences of actions $\mathbf{u}_{\leq t}$ and $\mathbf{v}_{\leq t}$,

$$d_{\mathsf{H}}\left(\mathsf{S}_k(\mathbf{u}_{\leq t}), \mathsf{S}_k(\mathbf{v}_{\leq t})\right) = \left(\left| \sum_{\tau=1}^{t} ||u_\tau||_2^2 - \sum_{\tau=1}^{t} ||v_\tau||_2^2 \right| \right)^{1/2} \tag{5.13}$$

and

$$\mu(\mathcal{B})^{\frac{2}{(T-t)m}}\left|\sum_{\tau=1}^{t}||u_\tau||_2^2-\sum_{\tau=1}^{t}||v_\tau||_2^2\right|=\left|\mu(\mathsf{S}(\mathbf{v}_{\le t}))^{\frac{2}{(T-t)m}}-\mu(\mathsf{S}(\mathbf{u}_{\le t}))^{\frac{2}{(T-t)m}}\right|$$
$$\le|\mu(\mathsf{S}(\mathbf{v}_{\le t}))-\mu(\mathsf{S}(\mathbf{u}_{\le t}))|^{\frac{2}{(T-t)m}}. \qquad (5.14)$$

Therefore setting $\delta=1$, (5.13) and (5.14) imply (5.10). We validate that the inventory constraints in Example 4 are $(k,1,1)$-*causally invariant* for all $1\le k\le T$.

Example 2: *Tracking constraints.*

Recall that the feasible set of actions for the tracking constraints is (with $p\ge 2$):

$$\mathsf{S}=\left\{\mathbf{u}\in\mathbb{R}^T:||\mathbf{u}-\mathbf{y}||_p^p\le\sigma\right\}.$$

The sequence of actions maximizing the size of the set of admissible actions is $\overline{\mathbf{u}}_{\le t}=\mathbf{y}_{\le t}$. Similar to Example 4, according to the formula of the volume of an ℓ_p-ball [144], we have

$$\left(\frac{\mu\left(\mathsf{S}\left(\mathbf{u}_{\le t}\right)\right)}{\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{\le t}\right)\right)}\right)^{\frac{1}{T-t}}=\left(1-\frac{||\mathbf{u}_{\le t}-\mathbf{y}_{\le t}||_p^p}{\sigma}\right)^{1/p}=\left(\frac{\mu\left(\mathsf{S}\left((\mathbf{u}_{\le t},\overline{\mathbf{u}}_{t+1:t+k})\right)\right)}{\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{<t+k}\right)\right)}\right)^{\frac{1}{T-t-k}}.$$

Therefore setting $\lambda=1$, (5.11) is satisfied. Next,we give a bound on the Hausdorff distance. For all $1\le k\le T$, sequences of actions $\mathbf{u}_{\le t}$ and $\mathbf{v}_{\le t}$,

$$\begin{aligned}
&\left(\frac{(2\Gamma\left(1+1/p\right))^{T-t}}{\Gamma\left((T-t)/p+1\right)}\right)^{\frac{1}{T-t}}d_{\mathsf{H}}\left(\mathsf{S}_k(\mathbf{u}_{\le t}),\mathsf{S}_k(\mathbf{v}_{\le t})\right)\\
\le&\frac{2\Gamma(1+1/p)}{\sqrt{\pi}}\left(\frac{\pi^{(T-t)/2}}{\Gamma\left((T-t)/2+1\right)}\right)^{\frac{1}{T-t}}\left|\sum_{\tau=1}^{t}||u_\tau-y_\tau||_p-\sum_{\tau=1}^{t}||v_\tau-y_\tau||_p\right|\\
=&\frac{2\Gamma(1+1/p)}{\sqrt{\pi}}\mu(\mathcal{B})^{\frac{1}{T-t}}\left|\sum_{\tau=1}^{t}||u_\tau-y_\tau||_p-\sum_{\tau=1}^{t}||v_\tau-y_\tau||_p\right|\\
\le&\frac{2\Gamma(1+1/p)}{\sqrt{\pi}}|\mu(\mathsf{S}(\mathbf{v}_{\le t}))-\mu(\mathsf{S}(\mathbf{u}_{\le t}))|^{\frac{1}{T-t}}
\end{aligned}$$

where $\Gamma(\cdot)$ is Euler's gamma function. Therefore setting $\delta=\frac{2\Gamma(1+1/p)}{\sqrt{\pi}}\le\frac{2}{\sqrt{\pi}}$ for all $p\ge 2$, (5.10) holds. The tracking constraints in Example 5 are $(k,2/\sqrt{\pi},1)$-*causally invariant* for all $1\le k\le T$.

Example 3: *Constraints in the Proof of Theorem 5.5.1.*

Denote by $\mathsf{A}_1:=\mathcal{B}(a/2)$ and $\mathsf{A}_2:=\mathsf{U}\backslash\mathcal{B}(a/2)$. The feasible length-$k$ subsequences of actions are either in the Cartesian product of sets $\mathsf{A}_1^k:=\mathsf{A}_1\times\cdots\times\mathsf{A}_1$

or $\mathsf{A}_2^k := \mathsf{A}_2 \times \cdots \times \mathsf{A}_2$. If the two sequences $\mathbf{u}_{<t}$ and $\mathbf{v}_{<t}$ are in the same set, then $\delta = \lambda = 1$; otherwise, the RHS of (5.11) becomes a constant term. Assuming $t + k \leq T$, the Hausdorff distance between $\mathsf{S}_k(\mathbf{u}_{<t}) = \mathsf{A}_1^k$ and $\mathsf{S}_k(\mathbf{v}_{<t}) = \mathsf{A}_2^k$ is $\Omega(k)$. Therefore, a non-scalar parameter $\delta = \Omega(k)$ is necessary for (5.11) to hold.

5.B Proofs

Proofs of Corollary 5.5.1

Proof of Corollary 5.5.1. The explicit expression in Lemma 17 ensures that whenever $p_t^*(u_t|\mathbf{u}_{<t}) > 0$, then there is always a feasible sequence of actions in $\mathsf{S}(\mathbf{u}_{<t})$. Now, if the tuning parameter $\beta > 0$, then the optimization (5.8)-(5.9) guarantees that $p_t^*(u_t|\mathbf{u}_{<t}) > 0$ for all $t \in [T]$; otherwise, the objective value in (5.8)-(5.9) is unbounded. Corollary 4.4.1 guarantees that for any sequence of actions $\mathbf{u} = (u_1, \ldots, u_T)$, if $p_t^*(u_t|\mathbf{u}_{<t}) > 0$ for all $t \in [T]$, then $\mathbf{u} \in \mathsf{S}$. Therefore, the sequence of actions $\mathbf{u}$ given by the PPC is always feasible. □

Proof of Theorem 5.5.1

The actions space U is closed and bounded by Assumption 5. Therefore, there exists a closed ball $\mathcal{B}(a)$ of radius $a > 0$ centered at $\mathbf{v} = (v_1, \ldots, v_T) \in \mathsf{U}$ such that $\mathcal{B}(a) \subseteq \mathsf{U}$. Consider the following safety constraints for all $t \in [T]\backslash\{1\}$ with memory size one:

$$\mathcal{U}_t := \begin{cases} \mathcal{B}(a/2), & \text{if } ||u_{t-1} - v_{t-1}||_2 \leq a \\ \mathsf{U}\backslash\mathcal{B}(a), & \text{if } ||u_{t-1} - v_{t-1}||_2 > a \end{cases} \tag{5.15}$$

where $\mathcal{B}(a/2)$ is a closed balls around the same center as $\mathcal{B}(a)$ with radius $a/2$. Let the state safety set $\mathcal{X}_t = \mathbb{R}^n$ for all $t \in [T]$ (i.e., no constraints on states). Whenever the first action u_1 at time $t = 1$ is taken, the future actions have to stay in the ball $\mathcal{B}(a/2)$, or stay outside $\mathcal{B}(a)$. Any deterministic policy at time $t \in [T]\backslash\{1\}$ has to take either $||u_t - v_t||_2 \leq a/2$ or $||u_t - v_t||_2 > a$.

Consider the following Lipschitz continuous cost functions that can be chosen adversarially:

$$c_t(u_t) = \begin{cases} 0, & \text{if } t \leq w \\ ||u_t - v_t||_2, & \text{if } ||u_{t-1} - v_{t-1}||_2 > a, t > w \\ (M - ||u_t - v_t||_2), & \text{if } ||u_{t-1} - v_{t-1}||_2 \leq a/2, t > w \end{cases}$$

where $M := \sup_{u \in \mathsf{U}} ||u||_2$. The construction of $\mathbf{c} = (c_1, \ldots, c_T)$ guarantees that the dynamic regret for any sequence of actions $\mathbf{u}$ given by a deterministic online policy

is bounded from below by

$$\mathsf{Regret}(\mathbf{u}) \geq (T-w)\min\left\{a, \left(M - \frac{a}{2}\right)\right\}.$$

Take $a = M/2$ and note that $M = \Omega(d)$ where $d := \mathsf{diam}(\mathsf{U}) := \sup\{||u-v||_2 : u, v \in \mathsf{U}\}$ is the diameter of the action space U. Therefore the dynamic regret is bounded from below as

$$\mathsf{Regret}(\mathbf{u}) = \Omega\left(d(T-w)\right)$$

for any sequence of actions $\mathbf{u}$ given by a deterministic online policy.

Proofs of Theorem 5.5.2 and Corollary 5.5.2

Proof of Theorem 5.5.2. Suppose there exists a sequence of actions $\mathbf{v}_{\leq s} \in \mathsf{U}^s$, $1 \leq s \leq T-k$, $k \geq w$ with $k = \Omega(T)$ and some constant $\alpha > 0$ such that

$$d_{\mathsf{H}}\left(\mathsf{S}_k(\mathbf{u}_{\leq s}), \mathsf{S}_k(\mathbf{v}_{\leq s})\right) \geq \alpha d k \tag{5.16}$$

for any $\mathbf{u}_{\leq s} \in \mathsf{U}^s$. For any safety sets satisfying (5.16), whenever the first s actions $\mathbf{u}_{\leq s}$ are taken, the future actions have to stay in $\mathsf{S}_k(\mathbf{u}_{\leq s})$. The same argument also holds for choosing $\mathbf{v}_{\leq s}$. This means that there exist two subsequences $\mathbf{u}_{s+1:s+k}$ and $\mathbf{v}_{s+1:s+k}$ such that

$$\begin{aligned}
||\mathbf{u}_{s+w:s+k} - \mathbf{v}_{s+w:s+k}||_2 &\geq ||\mathbf{u}_{s+1:s+k} - \mathbf{v}_{s+1:s+k}||_2 - ||\mathbf{u}_{s+1:s+w-1} - \mathbf{v}_{s+1:s+k}||_2 \\
&\geq ||\mathbf{u}_{s+1:s+k}||_2 - \sum_{t=s+1}^{s+w-1} ||u_t - v_t||_2 \\
&\geq \alpha d k - d(w-1) \\
&= d(\alpha k - w + 1).
\end{aligned}$$

Then the adversary can construct Lipschitz continuous cost functions similar to the one we used in the proof of Theorem 5.5.1 so that the costs $c_{s+w}, \ldots, c_{s+k}$ satisfy $\sum_{t=s+w}^{s+k} c(a_t) = \Omega((k-w+1)d) = \Omega(d(T-w))$ (since $k = \Omega(T)$) for the case when $a_t = v_t$ for $s < t \leq s+k$ and $a_t = v_t$ for $t \leq s$; but $c_{s+w} = \cdots = c_{s+k} = 0$ for the case when $a_t = u_t$ for $s < t \leq s+k$ and $a_t = v_t$ for $t \leq s$ and vice versa. This "switching pattern" attack is possible, because the online agent knows nothing about the costs $c_{s+w}, \ldots, c_{s+k}$ at time s and the adversary can design costs freely as long as they satisfy Assumption 5. The bound on the dynamic regret follows since for the optimal actions $\mathbf{u}^* = (u_1, \ldots, u_T)$, $\sum_{t=s+w}^{s+k} c(u_t^*) = 0$ but $\sum_{t=s+w}^{s+k} c(a_t) = \Omega(d(T-w))$ for any actions $\mathbf{a}$ generated by a deterministic online policy. □

Proof of Corollary 5.5.2. Applying Theorem 5.5.2 and noting that $\alpha = \Omega(T)$ and at each time t the received MEF functions $p_t^*, \ldots, p_{t+w-1}^*$ equivalently form a joint density function on $\mathbf{u}_{t+1:t+w}$, which has less information than an online policy that knows the offline safety sets. Since $\overline{\mathbf{u}}$ maximizes $\mu(\mathsf{S}(\cdot))$,

$$\frac{\mu(\mathsf{S}(\mathbf{u}_{\leq s}))}{\mu(\mathsf{S}(\mathbf{v}_{\leq s}))} \geq \frac{\mu(\mathsf{S}(\mathbf{u}_{\leq s}))}{\mu(\mathsf{S}(\overline{\mathbf{u}}_{\leq s}))} > \xi,$$

it follows that

$$\left(\frac{|\mu(\mathsf{S}(\mathbf{u}_{\leq s})) - \xi\mu(\mathsf{S}(\mathbf{v}_{\leq s}))|}{\mu(\mathcal{B})}\right)^{1/((T-s)m)} = \Omega(1).$$

Hence, the same argument in the proof of Theorem 5.5.2 follows and we obtain the desired regret lower bound.

□

Proof of Theorem 5.5.3

In this appendix, we prove Theorem 5.5.3 in four steps. First, in Lemma 20, we bound the deviation of the density function (feedback) evaluated at the actions selected by the PPC from the largest density function value, given the previous actions selected by the PPC. The deviation decreases with the tuning parameter $\beta > 0$. Next, using the bound obtained from the first step, in Lemma 21 we show that the ratio of the volume of the set of feasible actions, given the previous actions selected by the PPC and the volume of the largest feasible set is bounded from above by an exponential function, that is decreasing with β as well. Next, we bound the partial cost difference, of the cost induced by a subsequence of feasible actions and the offline optimal cost. Finally, combing the partial costs, we bound the total cost and this leads to an upper bound on the dynamic regret.

Step 1. *Bound the feedback deviation*

Recall that $t' := \min\{t + w - 1, T\}$. For every $t \in \mathcal{I}$, let $\overline{\mathbf{u}}_{t:t'} = \overline{u}_t, \ldots, \overline{u}_{t'}$ be a subsequence of optimal actions that maximizes the penalty term:

$$\overline{\mathbf{u}}_{t:t'} := \operatorname*{arg\,sup}_{\mathbf{u} \in \mathsf{U}^{t'-t+1}} p_{t:t'}^*(\mathbf{u}|\mathbf{u}_{<t}) = \operatorname*{arg\,sup}_{\mathbf{u} \in \mathsf{U}^{t'-t+1}} \mu\left(\mathsf{S}\left((\mathbf{u}_{<t}, \mathbf{u})\right)\right).$$

Before proceeding to the proof of Theorem 5.5.3, we first show the following lemma, which gives a lower bound on the feedback given the sequence of actions selected by the PPC. Note that when $t - 1 < 0$, the density functions become unconditional.

Lemma 20. *For any $t \in \mathcal{I}$, the sequence of actions $\mathbf{u} = (\mathbf{u}_{\leq t'})$ selected by the PPC satisfies*

$$\frac{p^*_{t:t'}(\mathbf{u}_{t:t'}|\mathbf{u}_{<t})}{p^*_{t:t'}(\overline{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t})} \geq \exp\left(-L_{\mathsf{c}} w d/\beta\right)$$

where L_{c} is the Lipschitz constant, $d := \mathsf{diam}(\mathsf{U}) = \sup\{||u - v||_2 : u, v \in \mathsf{U}\}$ is the diameter of the action space U.

Proof of Lemma 20. First, we note that the actions $\mathbf{u}_{t:t'}$ chosen by the PPC must satisfy

$$\sum_{\tau=t}^{t'} \beta \left(\log p^*_\tau(\overline{u}_\tau|\mathbf{u}_{<\tau}) - \log p^*_\tau(u_\tau|\mathbf{u}_{<\tau})\right) \leq \left|\sum_{\tau=t}^{t'} \left(c_\tau(\overline{u}_\tau) - c_\tau(u_\tau)\right)\right|. \tag{5.17}$$

To see this, suppose (5.17) does not hold. Then choosing $\overline{\mathbf{u}}_{t:t'}$ gives a smaller objective value in (5.8)-(5.9), which violates the definition of PPC-generated actions $\mathbf{u}_{t:t'}$. Using the chain rule, (5.17) becomes

$$\log p^*_{t:t'}(\overline{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t}) \leq \log p^*_{t:t'}(\mathbf{u}_{t:t'}|\mathbf{u}_{<t}) + \frac{1}{\beta}\left|\sum_{\tau=t}^{t'} \left(c_\tau(\overline{u}_\tau) - c_\tau(u_\tau)\right)\right|.$$

Therefore, since the cost functions are Lipschitz continuous,

$$\begin{aligned} p^*_{t:t'}(\overline{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t}) &\leq \exp\left(\frac{1}{\beta}\left|\sum_{\tau=t}^{t'} \left(c_\tau(\overline{u}_\tau) - c_\tau(u_\tau)\right)\right|\right) p^*_{t:t'}(\mathbf{u}_{t:t'}|\mathbf{u}_{<t}) \\ &\leq \exp\left(L_{\mathsf{c}} \frac{w d}{\beta}\right) p^*_{t:t'}(\mathbf{u}_{t:t'}|\mathbf{u}_{<t}). \end{aligned}$$

□

Step 2. *Bound the Lebesgue measure deviation*

Based on Lemma 20, the following lemma holds. For every $t \in \mathcal{I}$, let $\overline{\mathbf{u}}_{<t'} = u_1, \ldots, u_{t'}$ be a subsequence of optimal actions maximizing the volume of the set of feasible actions:

$$\overline{\mathbf{u}}_{<t'} := \arg\sup_{\mathbf{u} \in \mathsf{U}^{t'-1}} \mu\left(\mathsf{S}(\mathbf{u})\right).$$

Lemma 21. *Suppose the safety sets are (w, δ, λ)-causally invariant. For any $t \in \mathcal{I}$, the actions selected by the PPC satisfy that*

$$\frac{\mu\left(\mathsf{S}\left(\mathbf{u}_{\leq t'}\right)\right)}{\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{<t'}\right)\right)} \geq \exp\left(-L_{\mathsf{c}} t d/\beta\right) \lambda^{-\lceil t/w \rceil}$$

where $L_c > 0$ is the Lipschitz constant, $d := \sup\{||u - v||_2 : u, v \in \mathsf{U}\}$ is the diameter of the action space U.

Proof. For any $t \in \mathcal{I}$, since the safety sets are (w, δ, λ)-causally invariant and $\mu\left(\mathsf{S}\left(\mathbf{u}_{<t'}\right)\right) = \mu\left(\mathsf{S}\left(\mathbf{u}_{\leq t'}\right)\right)$,

$$\frac{\mu\left(\mathsf{S}\left(\mathbf{u}_{\leq t'}\right)\right)}{\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{<t'}\right)\right)} \geq \frac{1}{\lambda}\left(\frac{\mu\left(\mathsf{S}\left(\mathbf{u}_{<t}\right)\right)}{\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{<t}\right)\right)}\right)^{\frac{T-t'+1}{T-t+1}} \cdot \frac{\mu\left(\mathsf{S}\left(\mathbf{u}_{\leq t'}\right)\right)}{\mu\left(\mathsf{S}\left(\left(\mathbf{u}_{<t}, \overline{\mathbf{u}}_{t:t'}\right)\right)\right)}. \tag{5.18}$$

Applying the explicit expression in Lemma 17, Lemma 20 implies that

$$\frac{\mu\left(\mathsf{S}\left(\mathbf{u}_{\leq t'}\right)\right)}{\mu\left(\mathsf{S}\left(\left(\mathbf{u}_{<t}, \overline{\mathbf{u}}_{t:t'}\right)\right)\right)} = \frac{p^*_{t:t'}(\mathbf{u}_{t:t'}|\mathbf{u}_{<t})}{p^*_{t:t'}(\overline{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t})} \geq \exp\left(-wL_{\mathsf{c}}d/\beta\right).$$

Therefore, applying (5.18) recursively leads to that for any $t \in \mathcal{I}$,

$$\frac{\mu\left(\mathsf{S}\left(\mathbf{u}_{\leq t'}\right)\right)}{\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{<t'}\right)\right)} \geq \exp\left(-L_{\mathsf{c}}td/\beta\right)\lambda^{-\lceil t/w\rceil}.$$

□

Step 3. *Bound the partial cost deviation*

Lemma 21 implies that for the offline optimal solution $\mathbf{u}^*_{t':t'+w} := (u^*_{t'}, \ldots, u^*_{t'+w})$,

$$\begin{aligned}\mu\left(\mathsf{S}\left(\mathbf{u}_{\leq t'}\right)\right) &\geq \exp\left(-L_{\mathsf{c}}td/\beta\right)\lambda^{-2t/w}\mu\left(\mathsf{S}\left(\overline{\mathbf{u}}_{<t'}\right)\right)\\ &\geq \exp\left(-L_{\mathsf{c}}td/\beta\right)\lambda^{-2t/w}\mu\left(\mathsf{S}\left(\mathbf{u}^*_{<t'}\right)\right),\end{aligned}$$

which leads to

$$\begin{aligned}\mu\left(\mathsf{S}\left(\mathbf{u}^*_{<t'}\right)\right) - \mu\left(\mathsf{S}\left(\mathbf{u}_{\leq t'}\right)\right) &\leq \left(1 + \exp\left(\frac{2t}{w}\log\lambda - L_{\mathsf{c}}\frac{td}{\beta}\right)\right)\mu\left(\mathsf{S}\left(\mathbf{u}^*_{<t'}\right)\right)\\ &\leq \left(L_{\mathsf{c}}\frac{td}{\beta} + \frac{2t}{w}\log\lambda\right)\mu\left(\mathsf{S}\left(\mathbf{u}^*{}_{<t'}\right)\right)\\ &\leq \left(L_{\mathsf{c}}\frac{td}{\beta} + \frac{2t}{w}\log\lambda\right)\mu\left(\mathcal{B}\right)\left(\frac{d}{2}\right)^{(T-t'+1)m}\end{aligned}$$

where we have used the inequality $e^x \geq 1 + x$ for all $x \in \mathbb{R}$. The last inequality follows from the isodiametric inequality, with $\mathcal{B}$ denoting the unit ball in $\mathbb{R}^{(T-t'+1)m}$.

Since the safety sets are (w, δ, λ)-causally invariant, it follows that for any $t \in \mathcal{I}$,

$$||\mathbf{u}^*_{t:t'} - \widehat{\mathbf{u}}_{t:t'}||_2 \leq d_{\mathsf{H}}\left(\mathsf{S}_w\left(\mathbf{u}^*_{<t}\right), \mathsf{S}_w\left(\mathbf{u}_{<t}\right)\right) \leq \delta d\left(L_{\mathsf{c}}\frac{td}{\beta} + \frac{2t}{w}\log\lambda\right)^{\frac{1}{(T-t+1)m}}$$

for some $\widehat{\mathbf{u}}_{t:t'} \in \mathsf{S}_w\left(\mathbf{u}_{<t}\right)$ satisfying $p^*_{t:t'}(\widehat{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t}) > 0$.

Next, we consider the objective for the PPC. For any $t \in \mathcal{I}$, denote by $C(\mathbf{u}_{t:t'}) := \sum_{\tau=t}^{t'} c_\tau(u_\tau)$ the cost for the times slots between t and $t' := \min\{t+w, T\}$. Since the costs are Lipschitz continuous,

$$\begin{aligned}
\left|C(\widehat{\mathbf{u}}_{t:t'}) - C(\mathbf{u}^*_{t:t'})\right| \leq& \sum_{\tau=t}^{t'} \left|c_\tau(u_\tau) - c_\tau(u^*_\tau)\right| \\
\leq& \sum_{\tau=t}^{t'} L_{\mathsf{c}} ||u^*_\tau - \widehat{u}_\tau||_2 \\
\leq& \sqrt{w} L_{\mathsf{c}} ||\mathbf{u}^*_{t:t'} - \widehat{\mathbf{u}}_{t:t'}||_2 \\
\leq& d\delta L_{\mathsf{c}} \left(L_{\mathsf{c}} \frac{td\sqrt{w}}{\beta} + \frac{2t}{\sqrt{w}} \log \lambda\right)^{\frac{1}{(T-t+1)m}}. \qquad (5.19)
\end{aligned}$$

Since $\mathbf{u}_{t:t'}$ is a minimizer of (5.8)-(5.9), we obtain

$$C(\mathbf{u}_{t:t'}) - \sum_{\tau=t}^{t'} \beta \log p_\tau(u_\tau|\mathbf{u}_{<\tau}) \leq C(\widehat{\mathbf{u}}_{t:t'}) - \sum_{\tau=t}^{t'} \beta \log p_\tau(\widehat{u}_\tau|\mathbf{u}_{<\tau}),$$

which implies

$$C(\mathbf{u}_{t:t'}) \leq C(\widehat{\mathbf{u}}_{t:t'}) + \beta \log \frac{p^*_{t:t'}(\mathbf{u}_{t:t'}|\mathbf{u}_{<t})}{p^*_{t:t'}(\widehat{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t})}. \qquad (5.20)$$

Step 4. *Bound the dynamic regret* $\mathsf{Regret}(\mathbf{u})$

Proof of Theorem 5.5.3. For the total cost, it follows that

$$\begin{aligned}
C_T(\mathbf{u}) = \sum_{t\in\mathcal{I}} c_t(\mathbf{u}_{t:t'}) \leq& \sum_{t\in\mathcal{I}} \left[C(\widehat{\mathbf{u}}_{t:t'}) + \beta \log \frac{p^*_{t:t'}(\mathbf{u}_{t:t'}|\mathbf{u}_{<t})}{p^*_{t:t'}(\widehat{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t})}\right] \\
\leq& \underbrace{\sum_{t\in\mathcal{I}} C(\widehat{\mathbf{u}}_{t:t'})}_{:=(a)} + \underbrace{\beta \sum_{t\in\mathcal{I}} \log \frac{\mu\left(\mathsf{S}(\mathbf{u}_{<t}, \mathbf{u}_{t:t'})\right)}{\mu\left(\mathsf{S}(\mathbf{u}_{<t}, \widehat{\mathbf{u}}_{t:t'})\right)}}_{:=(b)}.
\end{aligned}$$

For (a), plugging in (5.19), the total cost for the sequence $\widehat{\mathbf{u}}$ is bounded from above by

$$\begin{aligned}
\sum_{t\in\mathcal{I}} C(\widehat{\mathbf{u}}_{t:t'}) \leq& C^*_T + \sum_{t\in\mathcal{I}} \left|C(\widehat{\mathbf{u}}_{t:t'}) - C(\mathbf{u}^*_{t:t'})\right| \\
\leq& C^*_T + \sum_{k=1}^{2T/w} d\delta L_{\mathsf{c}} \left(L_{\mathsf{c}} \frac{kd\sqrt{w}}{\beta} + \frac{k}{\sqrt{w}} \log \lambda\right)^{\frac{1}{(T-kw+1)m}} \\
=& C^*_T + O\left(\frac{T}{\sqrt{w}} d\delta \log \lambda + \frac{d^2\delta}{\beta} T\sqrt{w}\right)
\end{aligned}$$

where $C_T^* := \sum_{t=1}^{T} c_t^*(u_t^*)$ is the optimal total cost. Finally, (b) becomes $O(T\beta/w)$ since the safety sets are atomic and U is bounded by Assumption 6 and 7. Rearranging the terms,

$$\mathsf{Regret}(\mathbf{u}) = O\left(Td\left(\frac{\delta \log \lambda}{\sqrt{w}} + \frac{d\delta\sqrt{w}}{\beta} + \frac{\beta}{dw}\right)\right).$$

Setting $\beta = d\sqrt{\delta}w^{3/4}$ immediately implies

$$\mathsf{Regret}(\mathbf{u}) = O\left(Td\left((\delta \log \lambda)/\sqrt{w} + \sqrt{\delta}/w^{1/4}\right)\right).$$

□

Part III

Learning, Inference, and Data Analysis in Smart Grids

Chapter 6

LEARNING POWER SYSTEM PARAMETERS FROM LINEAR MEASUREMENTS

[1] Tongxin Li, Lucien Werner, and Steven H. Low. Learning graph parameters from linear measurements: Fundamental trade-offs and application to electric grids. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6554–6559, 2019. URL https://doi.org/10.1109/CDC40024.2019.9029949.

[2] Tongxin Li, Lucien Werner, and Steven H. Low. Learning graphs from linear measurements: Fundamental trade-offs and applications. *IEEE Transactions on Signal and Information Processing over Networks*, 6:163–178, 2020. URL https://doi.org/10.1109/TSIPN.2020.2975368.

In the results on learning-augmented control and decision-making presented in Chapter 2 and 3, system matrices A and B or their crude estimates are assumed to be available. In this chapter, we consider system identification and inference problems of those system matrices for power systems. We provide a characterization of fundamental trade-offs for system identification between the number of samples, the complexity of the graph class, and the probability of error.

6.1 Introduction

Symmetric matrices are ubiquitous in graphical models with examples such as the $(0, 1)$ adjacency matrix and the (generalized) Laplacian of an undirected graph. A major challenge in graph learning is inferring graph parameters embedded in those graph-based matrices from historical data or real-time measurements. In contrast to traditional statistical inference methods [145–147], model-based graph learning, such as physically-motivated models and graph signal processing (GSP) [148], takes advantage of additional data structures offered freely by nature. Among different measurement models for graph learning, linear models have been used and analyzed widely for different tasks, *e.g.,* linear structural equation models (SEMs) [149, 150], linear graph measurements [151], generalized linear cascade models [152], *etc*.

Despite extra efforts required on data collection, processing and storage, model-based graph learning often guarantees provable sample complexity, which is often significantly lower than the empirical number of measurements needed with traditional

inference methods. In many problem settings, having computationally efficient algorithms with low sample complexity is important. One reason for this is that the graph parameters may change in a short time-scale, making sample complexity a vital metric to guarantee that the learning can be accomplished with limited measurements. Indeed many applications, such as real-time optimal power flow [153–155], real-time contingency analysis [156] and frequency control [157] in power systems *etc*., require data about the network that are time-varying. For example, the generations or net loads may change rapidly due to the proliferation of distributed energy resources. The topology and line parameters of the grid may be reconfigured to mitigate cascading failure [158]. Line switching has changed the traditional idea of a power network with a fixed topology, enabling power flow control by switching lines [159], *etc*. Hence analyzing fundamental limits of parameter reconstruction and designing graph algorithms that are efficient in both computational and sample complexity are important.

The number of measurements needed for reconstructing a graph Laplacian can be affected by various system parameters, such as data quality (distribution), physical laws, and graph structures. In particular, existing recovery algorithms often assume the graph to be recovered is in a specific class, *e.g.,* trees [145], sparse graphs [160], graphs with no high-degree nodes [161], with notable exceptions such as [162], which considers an empirical algorithm for topology identification. However, there is still a lack of understanding of sample complexity for learning general undirected graphs that may contain high-degree nodes, especially with measurements constrained naturally by a linear system.

In this work, we consider a general graph learning problem where the measurements and underlying matrix to be recovered can be represented as or approximated by a linear system. A *graph matrix* $\mathbf{Y}(G)$ with respect to an underlying graph G, which may have *high-degree* nodes (see Definition 6.2.1) is defined as an $n \times n$ symmetric matrix with each nonzero (i, j)-th entry corresponding to an edge connecting node i and node j where $n \in \mathbb{N}_+$ is the number of nodes of the underlying *undirected* graph. The diagonal entries can be arbitrary. The measurements are summarized as two $m \times n$ ($1 \leq m \leq n$) real or complex matrices $\mathbf{A}$ and $\mathbf{B}$ satisfying

$$\mathbf{A} = \mathbf{B}\mathbf{Y}(G) + \mathbf{Z} \tag{6.1}$$

where $\mathbf{Z}$ denotes additive noise.

We focus on the following problems:

- *Fundamental Trade-offs*. What is the *minimum number m* of linear measurements required for reconstructing the *symmetric* matrix $\mathbf{Y}(G)$? Is there an algorithm *asymptotically achieving* recovery with the minimum number of measurements? As a special case, can we characterize the sample complexity when the measurements are Gaussian IID[1]?

- *Applications to Electrical Grids*. Do the theoretical guarantees on sample complexity result in a practical algorithm (in terms of both sample and computational complexity) for recovering electric grid topology and parameters?

Some comments about the above model and the results in this work are as follows.

Remark 7. It has been noted that vectorization and standard compressed sensing techniques do not lead to straightforward results (see [161] for detailed arguments about a similar linear system). This issue is discussed extensively in Section 6.1.

Remark 8. The results in this work do not assume low-degree nodes as most of existing results do, with notable exceptions such as [162] which gives empirical and data-based subroutines for topology identification.

Related Work

Graph Learning. Algorithms for learning sparse graphical model structures have a rich tradition in the literature. For general Markov random fields (MRFs), learning the underlying graph structures is known to be NP-hard [163]. However, in the case when the underlying graph is a tree, the classical Chow-Liu algorithm [145] offers an efficient approach to structure estimation. Recent results contribute to an extensive understanding of the Chow-Liu algorithm. The authors in [147] analyzed the error exponent and showed experimental results for chain graphs and star graphs. For pairwise binary MRFs with bounded maximum degree, [164] provides sufficient conditions for correct graph selection. Model-based graph learning has been emerging recently and assuming the measurements form linear SEMs, the authors in [149, 150] showed theoretical guarantees of the sample complexity for learning a directed acyclic graph (DAG) structure, under mild conditions on the class of graphs.

For converse, information-theoretic tools have been widely applied to derive fundamental limits for learning graph structures. For a Markov random field with bounded

[1]This means the entries of the matrix $\mathbf{B}$ are IID normally distributed.

maximum degree, necessary conditions on the number of samples for estimating the underlying graph structure were derived in [164] using Fano's inequality (see [165]). For Ising models, [166] combines Fano's inequality with the idea of *typicality* to derive weak and strong converse. Similar techniques have also been applied to Bayesian networks [167]. Fundamental limits for noisy compressed sensing have been extensively studied in [168] under an information-theoretic framework.

System Identification in Power Systems. Graph learning has been widely used in electric grids applications, such as state estimation [169, 170] and topology identification [171, 172]. Most of the literature focuses on topology identification or change detection, but there is less work on joint topology and parameter reconstruction, with notable exceptions of [173–176]. However, the linear system proposed in [174] does not leverage the sparsity of the graph[2]. Thus, in the worst case, the matrix $\mathbf{B}$ needs to have full column rank, implying that $m = \Omega(n)$ measurements are necessary for recovery.

Moreover, there is little exploration on the fundamental performance limits (estimation error and sample complexity) on topology and parameter reconstruction of power networks, with the exception of [177] where a sparsity condition was given for exact recovery of outage lines. Based on single-type measurements (either current or voltage), correlation analysis has been applied for topology identification [178–180]. Approximating the measurements as normal distributed random variables, the authors of [171] proposed an approach for topology identification with limited measurements. A graphical learning-based approach can be found in [181]. Recently, data-driven methods were studied for parameter estimation [175]. In [174], a similar linear system as (6.6) was used combined with regression to recover the symmetric graph parameters (which is the admittance matrix in the power network).

Compressed Sensing and Sketching. It is well known that compressed sensing ([182, 183]) techniques allow for recovery of a sparse matrix with a limited number of measurements in various applications such as medical imaging [184], wireless communication [185], channel estimation [186] and circuit design [187], *etc*. For electricity grids, in [188], based on these techniques, experimental results have been given for topology recovery. However, nodal admittance matrices (generalized Laplacians) for power systems have two properties for which there are gaps in

[2]With respect to sparsity, we consider not only graphs with bounded degrees, but a broader class of graphs which may contain high-degree nodes. Definition 6.3.1 gives a comprehensive characterization of sparsity.

the sparse recovery literature: 1) the presence of high-degree nodes in a graph (corresponding to dense columns in its Laplacian) and 2) symmetry.

Consider a vectorization of system (6.1) using tensor product notation, with $\mathbf{a} := \text{vec}(\mathbf{A})$ and $\mathbf{y}(G) := \text{vec}(\mathbf{Y}(G))$. Then linear system (6.1) is equivalent to $\mathbf{a} = (\mathbf{I} \otimes \mathbf{B})\mathbf{y}(G)$ where $\text{vec}(\cdot)$ produces a column vector by stacking the columns of the input matrix and $\mathbf{I} \otimes \mathbf{B}$ is the Kronecker product of an identity matrix $\mathbf{I} \in \mathbb{R}^{n \times n}$ and $\mathbf{B}$. With the sensing matrix being a Kronecker product of two matrices, traditional compressed sensing analysis works for the case when $\mathbf{y}$ contains only $\mu = \Theta(1)$ non-zeros [189]. For instance, the authors of [190] showed that the restricted isometry constant (see Section 6.3 for the definition), $\delta_\mu(\mathbf{I} \otimes \mathbf{B})$ is bounded from above by $\delta_\mu(\mathbf{B})$, the restricted isometry constant of $\mathbf{B}$. However, if a column (or row) of $\mathbf{Y}(\mathbf{G})$ is dense, classical restricted isometry-based approach cannot be applied straightforwardly.

Another way of viewing it is that vectorizing $\mathbf{A}$ and $\mathbf{Y}(G)$ and constructing a sensing matrix $\mathbf{I} \otimes \mathbf{B}$ is equivalent to recovering each of the column (or row) of $\mathbf{Y}(G)$ separately from $A_j = \mathbf{B}Y_j(G)$ for $j = 1, \ldots, n$ where A_j's and $Y_j(G)$'s are columns of $\mathbf{A}$ and $\mathbf{Y}(G)$. For a general "sparse" graph G, such as a star graph, some of the columns (or rows) of the graph matrix $\mathbf{Y}(G)$ may be dense vectors consisting of many non-zeros. The results in [189, 190] give no guarantee for the recovery of the dense columns of $\mathbf{Y}(G)$ (correspondingly, the high-degree nodes in G), and thus they cannot be applied directly to the analysis of sample complexity. This statement is further validated in our experimental results shown in Figure 6.6.2 and Figure 6.6.3.

The authors of [161] considered the recovery of an unknown sparse matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ (not necessarily symmetric) from an $m \times m$ matrix $\overline{\mathbf{A}} = \overline{\mathbf{B}}\mathbf{M}\overline{\mathbf{C}}^T$ where $\overline{\mathbf{B}} \in \mathbb{R}^{m \times n}$ and $\overline{\mathbf{C}} \in \mathbb{R}^{m \times n}$ with $m \ll n$. By adding a symmetry constraint to their recovery formulation, we obtain the following modified basis pursuit as a convex optimization:

$$\text{minimize } ||\mathbf{Y}(G)||_1 \tag{6.2}$$

$$\text{subject to } \mathbf{B}\mathbf{Y}(G) = \mathbf{A}, \tag{6.3}$$

$$\mathbf{Y}(G) \in \mathbb{S}^{n \times n} \tag{6.4}$$

where $||\mathbf{Y}(G)||_1 = ||\text{vec}(\mathbf{Y}(G))||_1$ is the entry-wise ℓ_1-norm of $\mathbf{Y}(G)$ and $\mathbb{S}^{n \times n}$ denotes the set of all symmetric matrices in $\mathbb{R}^{n \times n}$. However, the approach in [161] does not carry through to our setting for two reasons. First, the analysis of such an optimization often requires stronger assumptions, *e.g.,* the non-zeros are not concentrated in any single column (or row) of $\mathbf{Y}(G)$, as in [161]. Second, having

the symmetry property of $\mathbf{Y}$ as a constraint does not explicitly make use of the fact that many columns in $\mathbf{Y}$ are indeed sparse and can be recovered correctly. As a consequence, basis pursuit may produce poor results in certain scenarios where our approach performs well, as demonstration in our experimental results on star graphs in Section 6.6.

Although the columns of $\mathbf{Y}(G)$ are correlated because of the symmetry, in general there are no constraints on the support sets of the columns. Thus distributed compressed sensing schemes (for instance, [191] requires the columns to share the same support set) are not directly applicable in this situation.

The previous studies and aforementioned issues together motivate us to propose a novel three-stage recovery scheme for the derivation of a sufficient recovery condition, which leads to a practical algorithm that is sample and computationally efficient as well as robust to noise.

Our Contributions

We demonstrate that the linear system in (6.1) can be used to learn the topology and parameters of a graph. Our framework can be applied to perform system identification in electrical grids by leveraging synchronous nodal current and voltage measurements obtained from phasor measurement units (PMUs).

Compared to existing methods and analysis, the main results of this work are three-fold:

1. *Fundamental Trade-offs*: In Theorem 6.3.1, we derive a general lower bound on the *probability of error* for topology identification (defined in (6.7)). In Section 6.3, we describe a simple three-stage recovery scheme combining ℓ_1-norm minimization with an additional step called *consistency-checking*, rendering which allows us to bound the number of measurements for exact recovery from above as in Theorem 6.3.2.

2. *(Worst-case) Sample Complexity*: We provide sample complexity results for recovering a random graph that may contain *high-degree* nodes. The unknown distribution that the graph is sampled from is characterized based on the definition of "(μ, K, ρ)-*sparsity*" (see Definition 6.3.1). Under the assumption that the matrix $\mathbf{B}$ has Gaussian IID entries, in Section 6.4, we provide upper and lower bounds on the worst-case sample complexity in Theorem 6.4.1. We

show two applications of Theorem 6.4.1 for the uniform sampling of trees and the Erdős-Rényi (n, p) model in Corollary 6.4.1 and 6.4.2, respectively.

3. *(Heuristic) Algorithm*: Motivated by the three-stage recovery scheme, a heuristic algorithm with polynomial (in n) running-time is reported in Section 6.5, together with simulation results for power system test cases validating its performance in Section 6.6.

Some comments about the above results are as follows:

Outline of This Chapter

The remaining content is organized as follows. In Section 6.2, we specify our models. In Section 6.3, we present the converse result as fundamental limits for recovery. The achievability is provided in 6.3. We present our main result as the worst-case sample complexity for Gaussian IID measurements in Section 6.4. A heuristic algorithm together with simulation results are reported in Sections 6.5 and 6.6.

6.2 Model and Definitions

Notation. Let $\mathbb{F}$ denote a field that can either be the set of real numbers $\mathbb{R}$, or the set of complex numbers $\mathbb{C}$. The set of all symmetric $n \times n$ matrices whose entries are in $\mathbb{F}$ is denoted by $\mathbb{S}^{n\times n}$. The imaginary unit is denoted by j. Throughout the work, let $\log(\cdot)$ denote the binary logarithm with base 2 and let $\ln(\cdot)$ denote the natural logarithm with base e. We use $\mathbb{E}[\cdot]$ to denote the expectation of random variables. The mutual information is denoted by $\mathbb{I}(\cdot)$. The entropy function (either differential or discrete) is denoted by $\mathbb{H}(\cdot)$ and in particular, we reserve $h(\cdot)$ for the binary entropy function. To distinguish random variables and their realizations, we follow the convention and denote the former by capital letters (*e.g.,* A) and the latter by lower case letters (*e.g.,* a). The symbol C is used to designate a constant.

Matrices are denoted in boldface (*e.g.,* $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{Y}$). The i-th row, the j-th column and the (i, j)-th entry of a matrix $\mathbf{A}$ are denoted by $A^{(i)}$, A_j and $A_{i,j}$, respectively. For notational convenience, let $\mathcal{S}$ be a subset of $\mathcal{V}$. Denote by $\overline{\mathcal{S}} := \mathcal{V}\backslash\mathcal{S}$ the complement of $\mathcal{S}$ and by $\mathbf{A}_{\mathcal{S}}$ a sub-matrix consisting of $|\mathcal{S}|$ columns of the matrix $\mathbf{A}$ whose indices are chosen from $\mathcal{S}$. The notation $\top$ denotes the transpose of a matrix, $\det(\cdot)$ calculates its determinant. For the sake of notational simplicity, we use big O notation (o,ω,O,Ω,Θ) to quantify asymptotic behavior.

Graphical Model

Denote by $\mathcal{V} = \{1, \ldots, n\}$ a set of n nodes and consider an *undirected* graph $G = (\mathcal{V}, \mathcal{E})$ (with no self-loops) whose edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ contains the desired topology information. The degree of each node j is denoted by d_j. The connectivity between the nodes is unknown and our goal is to determine it by learning the associated *graph matrix* using linear measurements.

Definition 6.2.1 (Graph matrix)**.** Provided with an underlying graph $G = (\mathcal{V}, \mathcal{E})$, a *symmetric* matrix $\mathbf{Y}(G) \in \mathbb{S}^{n\times n}$ is called a *graph matrix* if the following conditions hold:

$$Y_{i,j}(G) = \begin{cases} \neq 0 & \text{if } i \neq j \text{ and } (i,j) \in \mathcal{E} \\ 0 & \text{if } i \neq j \text{ and } (i,j) \notin \mathcal{E} \\ \text{arbitrary} & \text{otherwise} \end{cases}.$$

Remark 9. Our theorems can be generalized to recover a broader class of symmetric matrices, as long as the matrix to be recovered satisfies (1) Knowing $\mathbf{Y}(G) \in \mathbb{F}^{n\times n}$ gives the full knowledge of the topology of G; (2) The number of non-zero entries in a column of $\mathbf{Y}(G)$ has the same order as the degree of the corresponding node, *i.e.,* $|\text{supp}(Y_j)| = O(d_j)$. for all $j \in \mathcal{V}$. To have a clear presentation, we consider specifically the case $|\text{supp}(Y_j)| = d_j$.

In this work, we employ a probabilistic model and assume that the graph G is chosen randomly from a *candidacy set* $\mathsf{C}(n)$ (with n nodes), according to some distribution $\mathcal{G}_n$. Both the candidacy set $\mathsf{C}(n)$ and distribution $\mathcal{G}_n$ are not known to the estimator. For simplicity, we often omit the subscripts of $\mathsf{C}(n)$ and $\mathcal{G}_n$.

***Example* 7.** *We exemplify some possible choices of the candidacy set and distribution:*

(a) (Mesh Network) When G represents a transmission (mesh) power network and no prior information is available, the corresponding candidacy set $\mathsf{G}(n)$ *consisting of all graphs with n nodes and G is selected uniformly at random from* $\mathsf{G}(n)$*. Moreover,* $|\mathsf{G}(n)| = 2^{\binom{n}{2}}$ *in this case.*

(b) (Radial Network) When G represents a distribution (radial) power network and no other prior information is available, then the corresponding candidacy set $\mathsf{T}(n)$ *is a set containing all spanning trees of the complete graph with n buses (nodes) and G is selected uniformly at random from* $\mathsf{T}(n)$*; the cardinality is* $|\mathsf{T}(n)| = n^{n-2}$ *by Cayley's formula.*

(c) *(Radial Network with Prior Information) When $G = (\mathcal{V}, \mathcal{E})$ represents a distribution (radial) power network, and we further know that some of the buses cannot be connected (which may be inferred from locational/geographical information), then the corresponding candidacy set $\mathsf{T}_H(n)$ is a set of spanning trees of a sub-graph $H = (\mathcal{V}, \mathcal{E}_H)$ with n buses. An edge $e \notin \mathcal{E}_H$ if and only if we know $e \notin \mathcal{E}$. The size of $\mathsf{T}_H(n)$ is given by Kirchhoff's matrix tree theorem (c.f. [192]).*

(d) *(Erdős-Rényi (n, p) model) In a more general setting, G can be a random graph chosen from an ensemble of graphs according to a certain distribution. When a graph G is sampled according to the Erdős-Rényi (n, p) model, each edge of G is connected IID with probability p. We denote the corresponding graph distribution for this case by $\mathcal{G}_{\mathrm{ER}}(n, p)$.*

The next section is devoted to describing available measurements.

Linear System of Measurements

Suppose the measurements are sampled discretely and indexed by the elements of the set $\{1, \ldots, m\}$. As a general framework, the measurements are collected in two matrices $\mathbf{A}$ and $\mathbf{B}$ and defined as follows.

Definition 6.2.2 (Generator and measurement matrices)**.** Let m be an integer with $1 \leq m \leq n$. The *generator matrix* $\mathbf{B}$ is an $m \times n$ *random* matrix and the *measurement matrix* $\mathbf{A}$ is an $m \times n$ matrix with entries selected from $\mathbb{F}$ that satisfy the linear system (6.1):

$$\mathbf{A} = \mathbf{B}\mathbf{Y}(G) + \mathbf{Z}$$

where $\mathbf{Y}(G) \in \mathbb{S}^{n \times n}$ is a graph matrix to be recovered, with an underlying graph G and $\mathbf{Z} \in \mathbb{F}^{m \times n}$ denotes the random *additive noise*. We call the recovery *noiseless* if $\mathbf{Z} = \mathbf{0}$. Our goal is to resolve the matrix $\mathbf{Y}(G)$ based on given matrices $\mathbf{A}$ and $\mathbf{B}$.

In the remaining contexts, we sometime simplify the matrix $\mathbf{Y}(G)$ as $\mathbf{Y}$ if there is no confusion.

Applications to Electrical Grids

Various applications fall into the framework in (6.1). Here we present two examples of the graph identification problem in power systems. The measurements are modeled as time series data obtained via nodal sensors at each node, *e.g.,* PMUs, smart switches, or smart meters.

Example 1: Nodal Current and Voltage Measurements

We assume data is obtained from a short time interval over which the unknown parameters in the network are *time-invariant*. $\mathbf{Y} \in \mathbb{C}^{n\times n}$ denotes the *nodal admittance matrix* of the network and is defined

$$Y_{i,j} := \begin{cases} -y_{i,j} & \text{if } i \neq j \\ y_i + \sum_{k\neq i} y_{i,k} & \text{if } i = j \end{cases} \tag{6.5}$$

where $y_{i,j} \in \mathbb{C}$ is the admittance of line $(i, j) \in \mathcal{E}$ and y_i is the self-admittance of bus i. Note that if two buses are not connected then $Y_{i,j} = 0$.

The corresponding generator and measurement matrices are formed by simultaneously measuring both current (or equivalently, power injection) and voltage at each node and at each time step. For each $t = 1, \ldots, m$, the nodal current injection is collected in an n-dimensional random vector $I_t = (I_{t,1}, \ldots, I_{t,n})$. Concatenating the I_t into a matrix we get $\mathbf{I} := [I_1, I_2, \ldots, I_m]^\top \in \mathbb{C}^{m\times n}$. The generator matrix $\mathbf{V} := [V_1, V_2, \ldots, V_m]^\top \in \mathbb{C}^{m\times n}$ is constructed analogously. Each pair of measurement vectors (I_t, V_t) from $\mathbf{I}$ and $\mathbf{V}$ must satisfy Kirchhoff's and Ohm's laws,

$$I_t = \mathbf{Y}V_t, \quad t = 1, \ldots, m. \tag{6.6}$$

In matrix notation, (6.6) is equivalent to $\mathbf{I} = \mathbf{VY}$, which is a noiseless version of the linear system defined in (6.1).

Compared with only obtaining one of the current, power injection or voltage measurements (for example, as in [147, 178, 179]), collecting simultaneous current-voltage pairs doubles the amount of data to be acquired and stored. There are benefits however. First, exploiting the physical law relating voltage and current not only enables us to identify the topology of a power network but also recover the parameters of the admittance matrix. Furthermore, dual-type measurements significantly reduce the sample complexity for learning the graph, compared with the results for single-type measurements.

Example 2: Nodal Power Injection and Phase Angles

Similar to the previous example, at each time $t = 1, \ldots, m$, denote by $P_{t,j}$ and $\theta_{t,j}$ the active nodal power injection and the phase of voltage at node j, respectively. The matrices $\mathbf{P} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{\theta} \in \mathbb{R}^{m\times n}$ are constructed in a similar way by concatenating the vectors $P_t = (P_{t,1}, \ldots, P_{t,n})$ and $\theta_t = (\theta_{t,1}, \ldots, \theta_{t,n})$. The matrix representation

of the DC power flow model can be expressed as a linear system $\mathbf{P} = \boldsymbol{\theta}\mathbf{C}\mathbf{S}\mathbf{C}^\top$, which belongs to the general class represented in (6.1). Here, the diagonal matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is the susceptence matrix whose e-th diagonal entry represents the susceptence on the e-th edge in $\mathcal{E}$ and $\mathbf{C} \in \{-1, 0, 1\}^{n\times|\mathcal{E}|}$ is the node-to-link incidence matrix of the graph. The vertex-edge incidence matrix[3] $\mathbf{C} \in \{-1, 0, 1\}^{n\times|\mathcal{E}|}$ is defined as

$$C_{j,e} := \begin{cases} 1, & \text{if bus } j \text{ is the source of } e \\ -1, & \text{if bus } j \text{ is the target of } e \\ 0, & \text{otherwise} \end{cases}.$$

Note that $\mathbf{C}\mathbf{S}\mathbf{C}^\top$ specifies both the network topology and the susceptences of power lines.

Probability of Error as the Recovery Metric

We define the error criteria considered in this chapter. We refer to finding the edge set $\mathcal{E}$ of G via matrices $\mathbf{A}$ and $\mathbf{B}$ as the *topology identification problem* and recovering the graph matrix $\mathbf{Y}$ via matrices $\mathbf{A}$ and $\mathbf{B}$ as the *parameter reconstruction problem*.

Definition 6.2.3. Let f be a function or algorithm that returns an estimated graph matrix $\mathbf{X} = f(\mathbf{A}, \mathbf{B})$ given inputs $\mathbf{A}$ and $\mathbf{B}$. The *probability of error for topology identification* ε_{T} is defined to be the probability that the estimated edge set is not equal to the correct edge set:

$$\varepsilon_{\mathrm{T}} := \mathbb{P}\left(\exists\, i \neq j \;\middle|\; \operatorname{sign}(X_{i,j}) \neq \operatorname{sign}\left(Y_{i,j}(G)\right)\right) \tag{6.7}$$

where the probability is taken over the randomness in $G, \mathbf{B}$ and $\mathbf{Z}$. The *probability of error for parameter reconstruction* $\varepsilon_{\mathrm{P}}(\eta)$ is defined to be the probability that the Frobenius norm of the difference between the estimate $\mathbf{X}$ and the original graph matrix $\mathbf{Y}(G)$ is larger than $\eta > 0$:

$$\varepsilon_{\mathrm{P}}(\eta) := \sup_{\mathbf{Y}\in\mathsf{Y}(G)} \mathbb{P}\left(||\mathbf{X} - \mathbf{Y}(G)||_{\mathrm{F}} > \eta\right) \tag{6.8}$$

where $||\cdot||_{\mathrm{F}}$ denotes the Frobenius norm, $\eta > 0$ and $\mathsf{Y}(G)$ is the set of all graph matrices $Y(G)$ that satisfy Definition 6.2.1 for the underlying graph G, and the probability is taken over the randomness in G, $\mathbf{B}$ and $\mathbf{Z}$. Note that for noiseless parameter reconstruction, *i.e.,* $\mathbf{Z} = 0$, we always consider exact recovery and set $\eta = 0$ and abbreviate the probability of error as ε_{P}.

[3] Although the underlying network is a directed graph, when considering the fundamental limit for topology identification, we still refer to the recovery of an undirected graph G.

6.3 Fundamental Trade-offs

We discuss fundamental trade-offs of the parameter reconstruction problem defined in Section 6.2 and 6.2. The converse result is summarized in Theorem 6.3.1 as an inequality involving the probability of error, the distributions of the underlying graph, generator matrix and noise. Next, in Section 6.3, we focus on a particular three-stage scheme, and show in Theorem 6.3.2 that under certain conditions, the probability of error is asymptotically zero (in n).

Necessary Conditions

The following theorem states the fundamental limit.

Theorem 6.3.1 (Converse)**.** *The probability of error for topology identification ε_{T} is bounded from below as*

$$\varepsilon_{\mathrm{T}} \geq 1 - \frac{\mathbb{H}(\mathbf{A}) - \mathbb{H}(\mathbf{Z}) + \ln 2}{\mathbb{H}(\mathcal{G}_n)} \tag{6.9}$$

where $\mathbb{H}(\mathbf{A})$, $\mathbb{H}(\mathbf{Z})$ are differential entropy (in base e) functions of the random variables $\mathbf{A}$, $\mathbf{Z}$, respectively, and $\mathbb{H}(\mathcal{G}_n)$ is the entropy (in base e) of the probability distribution $\mathcal{G}_n$.

Remark 10. It can be inferred from the theorem that $\varepsilon_{\mathrm{T}} = 1 - O(mn/\mathbb{H}(\mathcal{G}_n))$, given that the generator matrix $\mathbf{B}$ has Gaussian IID entries and the noise $\mathbf{Z}$ is additive white Gaussian (see Lemma 24). Therefore, the structure of the graphs reflected in the corresponding entropy of the graph distribution determines the number of samples needed. Consider the four cases listed in Example 7. The number of samples must be at least linear in n (size of the graph) to ensure a small probability of error, given that the graph, as a mesh network, is chosen uniformly at random from $\mathsf{C}(n)$ (see Example 7 (a)) since $\mathbb{H}(\mathcal{U}_{\mathsf{G}(n)}) = \binom{n}{2}$. On the other hand, as corollaries, under the assumptions of Gaussian IID measurements, $m = \Omega(\log n)$ is *necessary* for making the probability of error less or equal to $1/2$, if the graph is chosen uniformly at random from $\mathsf{T}(n)$; $m = \Omega(nh(p))$ is *necessary* if the graph is sampled according to $\mathcal{G}_{\mathrm{ER}}(n, p)$, as in Examples 7 (b) and (c), respectively. The theorem can be generalized to complex measurements by adding additional multiplicative constants.

Note that $\varepsilon_{\mathrm{P}} \geq \varepsilon_{\mathrm{T}}$ for any fixed noiseless parameter reconstruction algorithm, the necessary conditions work for both topology and (noiseless) parameter reconstruction. The proof is postponed to Appendix 6.A and the key steps are first applying the generalized Fano's inequality (see [165, 168]) and then bounding the mutual

information $\mathbb{I}(G; \mathbf{A}|\mathbf{B})$ from above by $\mathbb{H}(\mathbf{A}) - \mathbb{H}(\mathbf{Z})$. The general converse stated in Theorem 6.3.1 is used in asserting the results on worst-case sample complexity in Theorem 6.4.1. Next, we analyze the sufficient condition for recovering a graph matrix $\mathbf{Y}(G)$. Before proceeding to the results, we introduce a novel characterization of the distribution $\mathcal{G}_n$, from which a graph G is sampled. In particular, the graph G is allowed to have high-degree nodes.

Characterization of Graph Distributions

Let $d_j(G)$ denote the degree of node $j \in \mathcal{V}$ in G. Denote by $\mathcal{V}_{\text{Large}}(\mu) := \{j \in \mathcal{V} \,\big|\, d_j(G) > \mu\}$ the set of nodes having degrees greater than the *threshold parameter* $0 \leq \mu \leq n-2$ and $\mathcal{V}_{\text{Small}}(\mu) := \mathcal{V} \backslash \mathcal{V}_{\text{Large}}(\mu)$ the set of nodes for all μ-sparse column vectors of $\mathbf{Y}$. With a *counting parameter* $0 \leq K \leq n$, we define a set of graphs wherein each graph consists of no more than K nodes with degree larger than μ, denoted by $\mathsf{C}(n, \mu, K) := \{G \in \mathsf{C}(n) \,|\, \left|\mathcal{V}_{\text{Large}}(\mu)\right| \leq K\}$. The following definition characterizes graph distributions.

Definition 6.3.1 ((μ, K, ρ)-sparse distribution)**.** A graph distribution $\mathcal{G}_n$ is said to be (μ, K, ρ)-*sparse* if assuming that G is distributed according to $\mathcal{G}_n$, then the probability that G belongs to $\mathsf{C}(n, \mu, K)$ is larger than $1 - \rho$, *i.e.,*

$$\mathbb{P}_{\mathcal{G}_n}\left(G \notin \mathsf{C}(n, \mu, K)\right) \leq \rho. \tag{6.10}$$

1) Uniform Sampling of Trees:

Based on the definition above, for particular graph distributions, we can find the associated parameters. We exemplify by considering two graph distributions introduced in Example 7. Denote by $\mathcal{U}_{\mathsf{T}(n)}$ the uniform distribution on the set $\mathsf{T}(n)$ of all trees with n nodes.

Lemma 22. *For any $\mu \geq 1$ and $K > 0$, the distribution $\mathcal{U}_{\mathsf{T}(n)}$ is $(\mu, K, 1/K)$-sparse.*

2) Erdős-Rényi (n, p) model:

Denote by $\mathcal{G}_{\text{ER}}(n, p)$ the graph distribution for the Erdős-Rényi (n, p) model. Similarly, the lemma below classifies $\mathcal{G}_{\text{ER}}(n, p)$ into a (μ, K, ρ)-sparse distribution with appropriate parameters.

Lemma 23. *For any $\mu(n, p)$ that satisfies $\mu(n, p) \geq 2nh(p)/(\ln 1/p)$ and $K > 0$, the distribution $\mathcal{G}_{\text{ER}}(n, p)$ is $(\mu, K, n\exp(-nh(p))/K)$-sparse.*

The proofs of Lemmas 22 and 23 are in Appendix 6.D.

Remark 11. It is worth noting that the (μ, K, ρ)-*sparsity* is capable of characterizing *any* arbitrarily chosen distribution. The interesting part is that for some of the well-known distributions, such as $\mathcal{G}_{\text{ER}}(n, p)$, this sparsity characterization offers a method that can be used in the analysis and moreover, it leads to an *exact characterization* of sample complexity for the noiseless case. Therefore, for the particular examples presented in Lemma 22 and Lemma 23, the selected threshold and counting parameters for both of them are "tight" (up to multiplicative factors), in the sense that the corresponding sample complexity matches (up to multiplicative factors) the lower bounds derived from Theorem 6.3.1. This can be seen in Corollary 6.4.1 and 6.4.2.

Algorithm 9: A Three-stage Recovery Scheme. The first stage focuses on solving each column of the matrix $\mathbf{Y}$ independently using ℓ_1-minimization. In the second stage, the recovery correctness of the first stage is further verified via *consistency-checking*, which utilizes the fact that the matrix to be recovered $\mathbf{Y}$ is *symmetric*. The parameter γ is set to zero for the analysis of noiseless parameter reconstruction.

Data: Matrices of measurements $\mathbf{A}$ and $\mathbf{B}$

Result: Estimated graph matrix $\mathbf{X}$

Step (a): Recovering columns independently:

for $j \in \mathcal{V}$ **do**

Solve the following ℓ_1-minimization and obtain an optimal $\mathbf{X}$:

$$\begin{aligned} \text{minimize} \quad & \left\|X_j\right\|_1 \\ \text{subject to} \quad & ||\mathbf{B}X_j - A_j||_2 \leq \gamma, \\ & X_j \in \mathbb{F}^n. \end{aligned}$$

end

Step (b): Consistency-checking:

for $\mathcal{S} \subseteq \mathcal{V}$ *with* $|\mathcal{S}| = n - K$ **do**

if $|X_{i,j} - X_{j,i}| > 2\gamma$ *for some* $i, j \in \mathcal{S}$ **then**

continue;

end

else

for $j \in \overline{\mathcal{S}}$ **do**

Step (c): Resolving unknown entries:

Update $X_j^{\overline{\mathcal{S}}}$ by solving the linear system:

$$\mathbf{B}_{\overline{\mathcal{S}}} X_j^{\overline{\mathcal{S}}} = A_j - \mathbf{B}_{\mathcal{S}} X_j^{\mathcal{S}}.$$

end

end

break;

end

return $\mathbf{X} = (X_1, \ldots, X_n)$;

Sufficient Conditions

In this subsection, we consider the sufficient conditions (achievability) for parameter reconstruction. The proofs rely on constructing a three-stage recovery scheme (Algorithm 9), which contains three steps – *column-retrieving*, *consistency-checking* and *solving unknown entries*. The worst-case running time of this scheme depends on the underlying distribution $\mathcal{G}_n$[4]. The scheme is presented as follows.

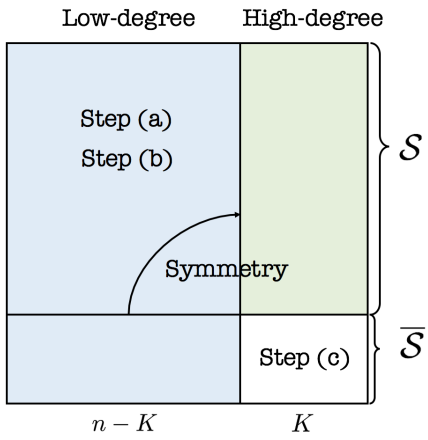


Figure 6.3.1: The recovery of a graph matrix $\mathbf{Y}$ using the three-stage scheme in Algorithm 10. The $n-K$ columns of $\mathbf{Y}$ colored by gray are first recovered via the ℓ_1-minimization (6.11a)-(6.11c) in step (a), after they are accepted by passing the consistency check in step (b). Then, symmetry is used for recovering the entries in the matrix marked by green. Leveraging the linear measurements again, in step (c), the remaining K^2 entries in the white symmetric sub-matrix are solved using Equation (6.12).

1) Three-stage Recovery Scheme:

Step (a): Retrieving columns. In the first stage, using ℓ_1-norm minimization, we recover each column of $\mathbf{Y}$ based on (6.1):

$$\begin{aligned} \text{minimize} \quad & \left\|X_j\right\|_1 && \text{(6.11a)} \\ \text{subject to} \quad & ||\mathbf{B}X_j - A_j||_2 \leq \gamma, && \text{(6.11b)} \\ & X_j \in \mathbb{F}^n. && \text{(6.11c)} \end{aligned}$$

Let $X_j^{\mathcal{S}} := (X_{i,j})_{i\in\mathcal{S}}$ be a length-$|\mathcal{S}|$ column vector consisting of $|\mathcal{S}|$ coordinates in X_j, the j-th retrieved column. We do not restrict the methods for solving the ℓ_1-norm

[4]Although for certain distributions, the computational complexity is not polynomial in n, the scheme still provides insights on the fundamental trade-offs between the number of samples and the probability of error for recovering graph matrices. Furthermore, motivated by the scheme, a polynomial-time heuristic algorithm is provided in Section 6.5 and experimental results are reported in Section 6.6.

minimization in (6.11a)-(6.11c), as long as there is a unique solution for sparse columns with fewer than μ non-zeros (provided enough number of measurements and the parameter $\mu > 0$ is defined in Definition 6.3.1).

Step (b): Checking consistency.

In the second stage, we check for error in the decoded columns $X_1, \ldots, X_n$ using the symmetry property (perturbed by noise) of the graph matrix $\mathbf{Y}$. Specifically, we fix a subset $\mathcal{S} \subseteq \mathcal{V}$ with a given size $|\mathcal{S}| = n - K$ for some integer[5] $0 \leq K \leq n$. Then we check if $|X_{i,j} - X_{j,i}| \leq 2\gamma$ for all $i, j \in \mathcal{S}$. If not, we choose a different set $\mathcal{S}$ of the same size. This procedure stops until either we find such a subset $\mathcal{S}$ of columns, or we go through all possible subsets without finding one. In the latter case, an error is declared and the recovery is unsuccessful. It remains to recover the vectors X_j for $j \in \overline{\mathcal{S}}$.

Step (c): Resolving unknown entries. In the former case, for each vector $X_j, j \in \mathcal{S}$, we accept its entries $X_{i,j}, i \in \overline{\mathcal{S}}$, as correct and therefore, according to the symmetry assumption, we know the entries $X_{i,j}, i \in \mathcal{S}, j \in \overline{\mathcal{S}}$ (equivalently $\{X_j^{\mathcal{S}} : j \in \overline{\mathcal{S}}\}$), which are used together with the sub-matrices $\mathbf{B}_{\mathcal{S}}$ and $\mathbf{B}_{\overline{\mathcal{S}}}$ to compute the other entries $X_{i,j}, i \in \overline{\mathcal{S}}$, of X_j using (6.11b):

$$\mathbf{B}_{\overline{\mathcal{S}}} X_j^{\overline{\mathcal{S}}} = A_j - \mathbf{B}_{\mathcal{S}} X_j^{\mathcal{S}}, \quad j \in \overline{\mathcal{S}}. \tag{6.12}$$

Note that to avoid being over-determined, in practice, we solve

$$\mathbf{B}_{\overline{\mathcal{S}}}^{\mathcal{K}} X_j^{\overline{\mathcal{S}}} = A_j^{\mathcal{K}} - \mathbf{B}_{\mathcal{S}}^{\mathcal{K}} X_j^{\mathcal{S}}, \quad j \in \overline{\mathcal{S}}$$

where $\mathbf{B}_{\overline{\mathcal{S}}}^{\mathcal{K}}$ is a $K \times K$ matrix whose rows are selected from $\mathbf{B}_{\overline{\mathcal{S}}}$ corresponding to $\mathcal{K} \subseteq \mathcal{V}$ with $|\mathcal{K}| = K$ and $\mathbf{B}_{\mathcal{S}}^{\mathcal{K}}$ selects the rows of $\mathbf{B}_{\mathcal{S}}$ in the same way. We combine $X_j^{\mathcal{S}}$ and $X_j^{\overline{\mathcal{S}}}$ to obtain a new estimate X_j for each $j \in \overline{\mathcal{S}}$. Together with the columns X_j, $j \in \mathcal{S}$, that we have accepted, they form the estimated graph matrix $\mathbf{X}$. We illustrate the three-stage scheme in Figure 6.3.1. In the sequel, we analyze the sample complexity of the three-stage scheme based on the (μ, K, ρ)-sparse distributions defined in Definition 6.3.1.

2) Analysis of the Scheme:

[5]The choice of K depends on the structure of the graph to be recovered and more specifically, K is the counting parameter in Definition 6.3.1. In Theorem 6.3.2 and Corollary 6.3.1, we analyze the sample complexity of this three-stage recovery scheme by characterizing an arbitrary graph into the classes specified by Definition 6.3.1 with a fixed K.

Let $\mathbb{F} \equiv \mathbb{R}$ for the simplicity of representation and analysis. We now present another of our main theorems. Consider the models defined in Section 6.2 and 6.2. The Γ-*probability of error* is defined to be the maximal probability that the ℓ_2-norm of the difference between the estimated vector $X \in \mathbb{R}^n$ and the original vector $Y \in \mathbb{R}^n$ (satisfying $A = \mathbf{B}Y + Z$ and both A and $\mathbf{B}$ are known to the estimator) is larger than $\Gamma > 0$:

$$\overline{\varepsilon}_{\mathrm{P}}(\Gamma) := \sup_{Y \in \mathsf{Y}(\mu)} \mathbb{P}\left(||X - Y||_2 > \Gamma\right)$$

where $\mathsf{Y}(\mu)$ is the set of all μ-sparse vectors in $\mathbb{R}^n$ and the probability is taken over the randomness in the generator matrix $\mathbf{B}$ and the additive noise Z. Given a generator matrix $\mathbf{B}$, the corresponding *restricted isometry constant* denoted by δ_μ is the smallest positive number with

$$\left(1 - \delta_\mu\right) ||\mathbf{x}||_2^2 \leq ||\mathbf{B}_{\mathcal{S}}\mathbf{x}||_2^2 \leq \left(1 + \delta_\mu\right) ||\mathbf{x}||_2^2 \tag{6.13}$$

for all subsets $\mathcal{S} \subseteq \mathcal{V}$ of size $|\mathcal{S}| \leq \mu$ and all $\mathbf{x} \in \mathbb{R}^{|\mathcal{S}|}$. Below we state a sufficient condition[6] derived form the three-stage scheme for parameter reconstruction.

Theorem 6.3.2 (Achievability)**.** *Suppose the generator matrix satisfies that* $\mathbf{B}_{\overline{\mathcal{S}}}^{\mathcal{K}} \in \mathbb{R}^{K \times K}$ *is invertible for all* $\overline{\mathcal{S}} \subseteq \mathcal{V}$ *and* $\mathcal{K} \subseteq \mathcal{V}$ *with* $|\overline{\mathcal{S}}| = |\mathcal{K}| = K$*. Let the distribution* $\mathcal{G}_n$ *be* (μ, K, ρ)*-sparse. If the three-stage scheme in Algorithm 10 is used for recovering a graph matrix* $\mathbf{Y}(G_n)$ *of* G_n *that is sampled according to* $\mathcal{G}_n$*, then the probability of error satisfies* $\varepsilon_{\mathrm{P}}(\eta) \leq \rho + (n - K)\overline{\varepsilon}_{\mathrm{P}}(\Gamma)$ *with* η *greater or equal to*

$$2\left(n\Gamma + \frac{\Gamma||\mathbf{B}||_2 + \gamma}{1 - \delta_{2K}}\right)\left(2(n - K) + K\xi(\mathbf{B})\right)$$

where δ_{2K} *is the corresponding restricted isometry constant of* $\mathbf{B}$ *with* $\mu = 2K$ *defined in* (6.13) *and*

$$\xi(\mathbf{B}) := \max_{\mathcal{S}, \mathcal{K} \subseteq \mathcal{V}, |\overline{\mathcal{S}}| = |\mathcal{K}| = K} \left\|\mathbf{B}_{\mathcal{S}}\right\|_2 \left\| \left(\mathbf{B}_{\overline{\mathcal{S}}}^{\mathcal{K}}\right)^{-1} \right\|_2.$$

The proof is in Appendix 6.B. The theory of classical compressed sensing (see [182, 183, 193]) implies that for noiseless parameter reconstruction, if the generator matrix $\mathbf{B}$ has restricted isometry constants $\delta_{2\mu}$ and $\delta_{3\mu}$ satisfying $\delta_{2\mu} + \delta_{3\mu} < 1$, then

[6]Note that γ cannot be chosen arbitrarily and Γ depends on γ; otherwise the probability of error $\overline{\varepsilon}_{\mathrm{P}}(\Gamma)$ will blow up. Theorem 6.4.2 indicates that for Gaussian ensembles setting $\Gamma = O(\gamma) = O(\sqrt{n}\sigma_{\mathrm{N}})$ is a valid choice where σ_{N} is the standard deviation of each independent $Z_{i,j}$ in $\mathbf{Z}$.

all columns Y_j with $j \in \mathcal{V}_{\text{Small}}$ are correctly recovered using the minimization in (6.11a)-(6.11c). Denote by spark($\mathbf{B}$) the smallest number of columns in the matrix $\mathbf{B}$ that are linearly dependent (see [194] for the requirements on the spark of the generator matrix to guarantee desired recovery criteria). The following corollary is an improvement of Theorem 6.3.2 for the noiseless case. The proof is postponed to Appendix 6.C.

***Corollary* 6.3.1.** *Let* $\mathbf{Z} = 0$ *and suppose the generator matrix* $\mathbf{B}$ *has restricted isometry constants* $\delta_{2\mu}$ *and* $\delta_{3\mu}$ *satisfying* $\delta_{2\mu} + \delta_{3\mu} < 1$ *and furthermore,* spark($\mathbf{B}$) $> 2K$. *If the distribution* $\mathcal{G}_n$ *is* (μ, K, ρ)*-sparse, then the probability of error for the three-stage scheme to recover the parameters of a graph matrix* $\mathbf{Y}(G_n)$ *of* G_n *that is sampled according to* $\mathcal{G}_n$ *satisfies* $\varepsilon_{\text{P}} \leq \rho$.

6.4 Gaussian IID Measurements

In this section, we consider a special regime when the measurements in the matrix $\mathbf{B}$ are Gaussian IID random variables. Utilizing the converse in Theorem 6.3.1 and the achievability in Theorem 6.3.2, the Gaussian IID assumption allows the derivation of explicit expressions of sample complexity as upper and lower bounds on the number of measurements m. Combining with the results in Lemma 22 and 23, we are able to show that for the corresponding lower and upper bounds match each other for graphs distributions $\mathcal{U}_{\mathsf{T}(n)}$ and $\mathcal{G}_{\text{ER}}(n, p)$ (with certain conditions on p and n).

For the convenience of presentation, in the remainder of the chapter, we restrict that the measurements are chosen from $\mathbb{R}$, although the theorems can be generalized to the complex measurements. In realistic scenarios, for instance, a power network, besides the measurements collected from the nodes, nominal state values, *e.g.,* operating current and voltage measurements are known to the system designer a priori. Representing the nominal values at the nodes by $\overline{A} \in \mathbb{R}^n$ and $\overline{B} \in \mathbb{R}^n$, respectively, the measurements in $\mathbf{A}$ and $\mathbf{B}$ are centered around $m \times n$ matrices $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ defined as

$$\overline{\mathbf{A}} := \begin{bmatrix} \cdots & \overline{A} & \cdots \\ \cdots & \overline{A} & \cdots \\ & \vdots & \\ \cdots & \overline{A} & \cdots \end{bmatrix}, \quad \overline{\mathbf{B}} := \begin{bmatrix} \cdots & \overline{B} & \cdots \\ \cdots & \overline{B} & \cdots \\ & \vdots & \\ \cdots & \overline{B} & \cdots \end{bmatrix}.$$

The rows in $\mathbf{A}$ and $\mathbf{B}$ are the same, because the graph parameters are time-invariant, so are the nominal values. Without system fluctuations and noise, the nominal values

satisfy the linear system in (6.1), *i.e.,*

$$\overline{\mathbf{A}} = \overline{\mathbf{B}}\mathbf{Y}. \tag{6.14}$$

Knowing $\overline{A}$ and $\overline{B}$ is not sufficient to infer the network parameters (the entries in the graph matrix $\mathbf{Y}$), since the rank of the matrix $\overline{B}$ is one. However, measurement fluctuations can be used to facilitate the recovery of $\mathbf{Y}$. The deviations from the nominal values are denoted by additive perturbation matrices $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$ such that $\mathbf{A} = \overline{\mathbf{A}} + \widetilde{\mathbf{A}}$. Similarly, $\mathbf{B} = \overline{\mathbf{B}} + \widetilde{\mathbf{B}}$ where $\widetilde{\mathbf{B}}$ is an $m \times n$ matrix consisting of additive perturbations. Therefore, considering the original linear system in (6.1), the equations above imply that $\overline{\mathbf{A}} + \widetilde{\mathbf{A}} = \mathbf{BY} + \mathbf{Z} = \overline{\mathbf{B}}\mathbf{Y} + \widetilde{\mathbf{B}}\mathbf{Y} + \mathbf{Z}$ leading to $\widetilde{\mathbf{A}} = \widetilde{\mathbf{B}}\mathbf{Y} + \mathbf{Z}$ where we have made use of (6.14) and extracted the perturbation matrices $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$. We specifically consider the case when the additive perturbations $\widetilde{\mathbf{B}}$ is a matrix with Gaussian IID entries. Without loss of generality, we suppose the mean of the Gaussian random variable is zero and the standard deviation is σ_{S}. We consider additive white Gaussian noise (AWGN) with mean zero and standard deviation σ_{N}. For simplicity, in the remainder of this chapter, we slightly abuse the notation and replace the perturbation matrices $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$ by $\mathbf{A}$ and $\mathbf{B}$ (we assume that $\mathbf{B}$ is Gaussian IID), if the context is clear. Under the assumptions above, the following lemma can be inferred from Theorem 6.3.1 and the proof is in Appendix 6.F.

Lemma 24. *Consider the linear model* $\mathbf{A} = \mathbf{BY} + \mathbf{Z}$. *Suppose* $B_{i,j} \sim \mathcal{N}(0, \sigma_{\mathrm{S}}^2)$ *and* $Z_{i,j} \sim \mathcal{N}(0, \sigma_{\mathrm{N}}^2)$ *are mutually independent Gaussian random variables for all* $i, j \in \mathcal{V}$. *The probability of error for topology identification* ε_{T} *is bounded from below as*

$$\varepsilon_{\mathrm{T}} \geq 1 - \frac{nm \ln\left(1 + \frac{\sigma_{\mathrm{S}}^2}{\sigma_{\mathrm{N}}^2}\overline{Y}\right)}{2\mathbb{H}\left(\mathcal{G}_n\right)} \tag{6.15}$$

where $\overline{Y} := \max_{i,j} \left|Y_{i,j}\right|$ *denotes the maximal absolute value of the entries in the graph matrix* $\mathbf{Y}$. *In particular, if* $\mathbf{Z} = 0$, *then for parameter reconstruction,*

$$\varepsilon_{\mathrm{P}} \geq 1 - \frac{nm \ln\left(2\pi e \overline{Y} \sigma_{\mathrm{S}}^2\right)}{2\mathbb{H}\left(\mathcal{G}_n\right)}. \tag{6.16}$$

Sample Complexity for Sparse Distributions

We consider the worst-case sample complexity for recovering graphs generated according to a sequence of sparse distributions, defined similarly as Definition 6.3.1 to characterize asymptotic behavior of graph distributions.

Definition 6.4.1 (Sequence of sparse distributions)**.** A sequence $\{\mathcal{G}_n\}$ of graph distributions is said to be (μ, K)-*sparse* if assuming a sequence of graphs $\{G_n\}$ is generated according to $\{\mathcal{G}_n\}$, the sequences $\{\mu(n)\}$ and $\{K(n)\}$ guarantee that

$$\lim_{n\to\infty} \mathbb{P}_{\mathcal{G}_n} \left(G_n \notin \mathsf{C}(n)(\mu(n), K(n))\right) = 0. \tag{6.17}$$

In the remaining contexts, we write $\mu(n)$ and $K(n)$ as μ and K for simplicity if there is no confusion. Based on the sequence of sparse distributions we defined above, we show the following theorem, which provides upper and lower bounds on the worst-case sample complexity, with Gaussian IID measurements.

Theorem 6.4.1 (Noiseless worst-case sample complexity)**.** *Let* $\mathbf{Z} = 0$*. Suppose that the generator matrix* $\mathbf{B}$ *has Gaussian IID entries with mean zero and variance one and assume* $\mu < n^{-3/\mu}(n - K)$ *and* $K = o(n)$*. For any sequence of distributions that is* (μ, K)*-sparse, the three-stage scheme guarantees that* $\lim_{n\to\infty} \varepsilon_{\mathrm{P}} = 0$ *using* $m = O\left(\mu \log(n/\mu) + K\right)$ *measurements. Conversely, there exists a* (μ, K)*-sparse sequence of distributions such that the number of measurements must satisfy* $m = \Omega\left(\mu \log(n/\mu) + K/n^{3/\mu}\right)$ *to make the probability of error* ε_{P} *less than* $1/2$ *for all* n*.*

The proof is postponed to Appendix 6.G.

Remark 12. The upper bound on m that we are able to show differs from the lower bound by a sub-linear term $n^{3/\mu}$. In particular, when the term $\mu \log(n/\mu)$ dominates K, the lower and upper bounds become tight up to a multiplicative factor.

Applications of Theorem 6.4.1

1) Uniform Sampling of Trees:

As one of the applications of Theorem 6.4.1, we characterize the sample complexity of the uniform sampling of trees.

***Corollary* 6.4.1.** *Let* $\mathbf{Z} = 0$*. Suppose that the generator matrix* $\mathbf{B}$ *has Gaussian IID entries with mean zero and variance one and assume* G_n *is distributed according to* $\mathcal{U}_{\mathsf{T}(n)}$*. There exists an algorithm that guarantees* $\lim_{n\to\infty} \varepsilon_{\mathrm{P}} = 0$ *using* $m = O\left(\log n\right)$ *measurements. Conversely, the number of measurements must satisfy* $m = \Omega\left(\log n\right)$ *to make the probability of error* ε_{P} *less than* $1/2$*.*

Proof. The achievability follows from combining Theorem 6.4.1 and Lemma 22, by setting $K(n) = \log n$. Substituting $\mathbb{H}(\mathcal{U}_{\mathsf{T}(n)}) = \Omega\left(n \log n\right)$ into (6.16) yields the desired result for converse. □

2) Erdős-Rényi (n, p) model:

Similarly, recalling Lemma 23, the sample complexity for recovering a random graph generated according to the Erdős-Rényi (n, p) model is obtained.

***Corollary* 6.4.2.** *Let $\mathbf{Z} = 0$. Assume G_n is a random graph sampled according to $\mathcal{G}_{\mathrm{ER}}(n, p)$ with $1/n \leq p \leq 1 - 1/n$. Under the same conditions in Corollary 6.4.1, there exists an algorithm that guarantees $\lim_{n\to\infty} \varepsilon_{\mathrm{P}} = 0$ using $m = O\left(nh(p)\right)$ measurements. Conversely, the number of measurements must satisfy $m = \Omega\left(nh(p)\right)$ to make the probability of error ε_{P} less than $1/2$.*

Proof. Taking $K = nh(p)/\log n$ and $\mu = 2nh(p)/(\ln 1/p)$, we check that $\mu < n^{-3/\mu}(n-K)$ and $K = o(n)$. The assumptions on $h(p)$ guarantee that $h(p) \geq \log n/n$, whence $nh(p) = \omega\left(\log(n/K)\right)$. The choices of $\{\mu(n)\}$ and $\{K(n)\}$ make sure that the sequence of distributions is $(\mu(n), K(n))$-sparse. Theorem 6.4.1 implies that $m = O(nh(p))$ is sufficient for achieving a vanishing probability of error. For the second part of the corollary, substituting $\mathbb{H}(\mathcal{G}_{\mathrm{ER}}(n, p)) = h(p)\binom{n}{2} = \Omega\left(n^2 h(p)\right)$ into (6.16) yields the desired result. □

Measurements Corrupted by Additive White Gaussian Noise (AWGN)

The results on sample complexity can be extended to the case with noisy measurements. The following theorem is proved by combining Theorem 6.3.2 and Lemma 24. The details can be found in Appendix 6.H.

Theorem 6.4.2 (Noisy worst-case sample complexity)**.** *Suppose that $\mathbf{B}$ and $\mathbf{Z}$ are defined as in Lemma 24. Let $\mu < n^{-3/\mu}(n-K)$ and $K = o(n)$. Conversely, there exists a (μ, K)-sparse sequence of distributions such that the number of measurements must satisfy*

$$m = \Omega\left(\frac{\mu \log(n/\mu) + K/n^{3/\mu}}{\log(1 + \sigma_{\mathrm{S}}^2/\sigma_{\mathrm{N}}^2)}\right)$$

to make the probability of error ε_{T} less than $1/2$ for all n. Moreover, if $\sigma_{\mathrm{N}} = o(1/n^{5/2})$, $\sigma_{\mathrm{S}} = 1/\sqrt{m}$ and $K \leq \mu$, then for any sequence of distributions that is (μ, K)-sparse, the three-stage scheme guarantees that $\lim_{n\to\infty} \varepsilon_{\mathrm{T}} = 0$ using $m = O\left(\mu \log(n/\mu)\right)$ measurements. Moreover, $\lim_{n\to\infty} \varepsilon_{\mathrm{P}}(\eta) = 0$ with $\eta = o(1)$.

Remark 13. The proof of Theorem 6.4.2 implies that $\eta = O(n^{5/2}\sigma_{\mathrm{N}})$. Therefore, if we consider the normalized Frobenius norm of $(1/n^2)||\mathbf{Y} - \mathbf{X}||_{\mathrm{F}}$ where $\mathbf{X}$ and $\mathbf{Y}$ are the recovered and original graph matrices, respectively, then $\sigma_{\mathrm{N}} = o(1/\sqrt{n})$

guarantees that the normalized Frobenius norm vanishes. For topology identification, we need to consider the Frobenius norm bound, η, to rule out the worst-case situation and the sufficient condition becomes $\sigma_{\mathrm{N}} = o(1/n^{5/2})$. Another implication is that the choice of γ in (6.11b) satisfying $\gamma = O(\sqrt{n}\sigma_{\mathrm{N}})$ (used in the proof) guarantees the reconstruction criteria and its effectiveness is also validated in our experiments in Section 6.6.

6.5 Heuristic Algorithm

We present in this section an algorithm motivated by the consistency-checking step in the proof of achievability (see Section 6.3). Instead of checking the consistency of each subset of $\mathcal{V}$ consisting of $n - K$ nodes, as the three-stage scheme does and which requires $O(n^K)$ operations, we compute an estimate X_j for each column of the graph matrix independently and then assign a score to each column based on its symemtric consistency with respect to the other columns in the matrix. The lower the score, the closer the estimate of the matrix column X_j is to the ground truth Y_j. Using a scoring function we rank the columns, select a subset of them to be "correct", and then eliminate this subset from the system. The size of the subset determines the number of iterations. Heuristically, this procedure results in a polynomial-time algorithm to compute an estimate $\mathbf{X}$ of the graph matrix $\mathbf{Y}$.

The algorithm proceeds in four steps.

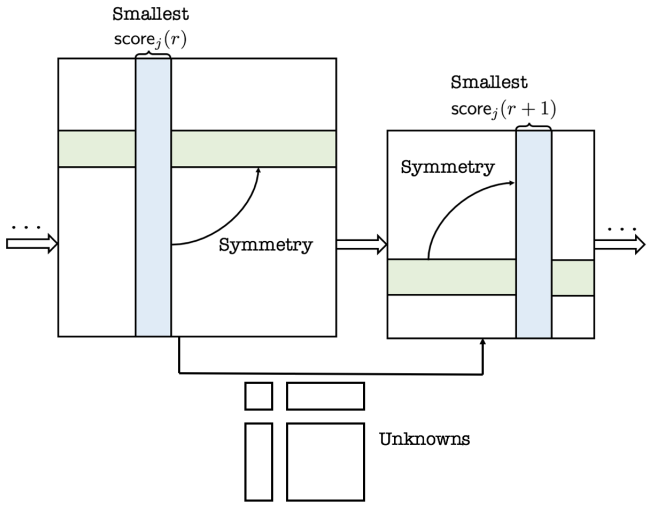


Figure 6.5.1: Iterative dimension reduction of the heuristic algorithm. At step r, the s columns with the smallest scores defined in (6.20) are assumed to be "correct" and eliminated from the linear system. The dimension of variables is reduced by s and this procedure is repeated until the $\lceil n/s \rceil$ iterations are complete.

Step 1**. Initialization**

Let matrices $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times n}$ be given and set the number of columns fixed in each iteration to be an integer s such that $1 \leq s \leq n$. For the first iteration, set $\mathcal{S}(0) \leftarrow \mathcal{V}$, $\mathbf{A}(0) \leftarrow \mathbf{A}$, and $\mathbf{B}(0) \leftarrow \mathbf{B}$.

For each iteration $r = 0, \ldots, \lceil n/s \rceil - 1$, we perform the remaining three stages. The system dimension is reduced by s after each iteration.

Step 2**. Independent ℓ_1-minimization**

For all $j \in \mathcal{S}(r)$, we solve the following ℓ_1-minimization:

$$X_j(r) = \underset{x\in\mathbb{R}^{n-sr}}{\arg\min} \quad \left\|x\right\|_1 \tag{6.18}$$

$$\text{subject to} \quad ||\mathbf{B}(r)x - A_j(r)||_2 \leq \gamma, \tag{6.19}$$

$$x \in \mathcal{X}_j(r).$$

Constraint (6.18) is optional; the set $\mathcal{X}_j(r)$ may encode additional constraints on the form of x such as entry-wise positivity or negativity (*e.g.,* Section 6.6). The forms of reduced matrix $\mathbf{B}(r)$ and reduced vector $A_j(r)$ are specified in Step 4.

Step 3**. Column scoring**

We rank the *symmetric consistency* of the independently solved columns. For all $j \in \mathcal{S}(r)$, let

$$\mathsf{score}_j(r) := \sum_{i=1}^{n-sr} \left|X_{i,j}(r) - X_{j,i}(r)\right|. \tag{6.20}$$

Note that if $\mathsf{score}_j(r) = 0$ then $X_j(r)$ and its partner symmetric row in $\mathbf{X}(r)$ are identical. Otherwise there will be some discrepancies between the entries and the sum will be positive. The subset of the $X_j(r)$ corresponding to the s smallest values of $\mathsf{score}_j(r)$ is deemed "correct." Call this subset of correct indices $\mathcal{S}'(r)$.

Step 4**. System dimension reduction**

Based on the assumption that s of the previously computed columns $X_j(r)$ are correct, the dimension of the linear system is reduced by s. We set $\mathcal{S}(r+1) \leftarrow \mathcal{S}(r)\backslash\mathcal{S}'(r)$. For all $i, j \in \mathcal{S}'(r)$, we fix

$$X_{i,j} = X_{i,j}(r),\ X_{j,i} = X_{i,j}(r). \tag{6.21}$$

The measurement matrices are reduced to

$$\mathbf{B}(r+1) \leftarrow \underline{\mathbf{B}}_{\mathcal{S}(r+1)},$$
$$A_j(r+1) \leftarrow \underline{A}_j(r) - \sum_{i \in \mathcal{S}'(r)} \underline{B}_i X_{i,j}.$$

When $r \leq n-m$, $\underline{\mathbf{B}}_{\mathcal{S}(r+1)} = \mathbf{B}_{\mathcal{S}(r+1)}$, $\underline{A}_j(r) = A_j(r)$ and $\underline{B}_i = B_i$. When $r > n-m$, to avoid making the reduced matrix $\mathbf{B}(r+1)$ over-determined, we set $\mathbf{B}(r+1)$ to be an $(n-r) \times (n-r)$ sub-matrix of $\mathbf{B}_{\mathcal{S}(r+1)}$ by selecting $n-r$ rows of $\mathbf{B}_{\mathcal{S}(r+1)}$ uniformly at random. A new length-$(n-r)$ vector $\underline{A}_j(r)$ is formed by selecting the corresponding entries from $A_j(r)$. Once the $\lceil n/s \rceil$ iterations complete, an estimate $\mathbf{X}$ is returned using (6.21). The algorithm requires at most $\lceil n/s \rceil$ iterations and in each iteration, the algorithm solves an ℓ_1-minimization and updates a linear system. Solving an ℓ_1-minimization can be done in polynomial time (*c.f.* [195]). Thus, the heuristic algorithm is a polynomial-time algorithm.

6.6 Applications in Electric Grids

Experimental results for the heuristic algorithm are given here for both synthetic data and IEEE standard power system test cases. The algorithm was implemented in Matlab; simulated power flow data was generated using Matpower 7.0 [196] and CVX 2.1 [197] with the Gurobi solver [198] was used to solve the sparse optimization subroutine.

Scalable Topologies and Error Criteria

We first demonstrate our results using synthetic data and two typical graph ensembles – stars and chains. For both topologies, we increment the graph size from $n = 5$ to $n = 300$ and record the number of samples required for accurate recovery of parameters and topology. For each simulation, we generate a complex-valued random admittance matrix $\mathbf{Y}$ as the ground truth. Both the real and imaginary parts of the line impedances of the network are selected uniformly and IID from $[-100, 100]$. A valid electrical admittance matrix is then constructed using these impedances. The real components of the entries of $\mathbf{B}$ are distributed IID according to $\mathcal{V}(1,1)$ and the imaginary components according to $\mathcal{V}(0,1)$. $\mathbf{A} = \mathbf{YB}$ gives the corresponding complex-valued measurement matrix. The parameter γ in (6.19) is 0 since we consider noiseless reconstruction here.

Given data matrices $\mathbf{A}, \mathbf{B}$ the algorithm returns an estimate $\mathbf{X}$ of the ground truth $\mathbf{Y}$. We set $s = \lceil n/2 \rceil$ for each graph. If an entry of $\mathbf{X}$ has magnitude $|X_{i,j}| < 10^{-5}$,

then we fix it to be 0. Following this, if $\text{supp}(\mathbf{X}) = \text{supp}(\mathbf{Y})$ then the topology identification is deemed exact. The criterion for accurate parameter reconstruction is $||\mathbf{Y} - \mathbf{X}||_{\text{F}}/n^2 < 10^{-6}$. The number of samples m (averaged over repeated trials) required to meet both of these criteria is designated as the sample complexity for accurate recovery. The sample complexity trade-off displayed in Figure 6.6.1 shows approximately logarithmic dependence on graph size n for both ensembles.

IEEE Test Cases

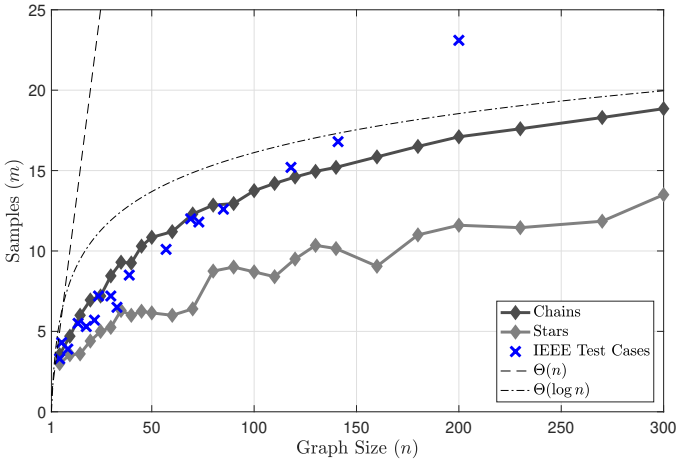


Figure 6.6.1: The number of samples required to accurately recover the nodal admittance matrix is shown on the vertical axis. Results are averaged over 20 independent simulations. Star and chain graphs are scaled in size between 5 and 300 nodes. IEEE test cases ranged from 5 to 200 buses. In the latter case, there are no assumptions on the random IID selection of the entries of $\mathbf{Y}$ (in contrast to the star/chain networks). Linear and logarithmic (in n) reference curves are plotted as dashed lines.

We also validate the heuristic algorithm on 17 IEEE standard power system test cases ranging from 5 to 200 buses. The procedure for determining sample complexity for accurate recovery is the same as above, but the data generation is more involved.

Power flow data generation

A sequence of time-varying loads is created by scaling the nominal load values in the test cases by a times series of Bonneville Power Administration's aggregate load on 02/08/2016, 6am to 12pm [199]. For each test case network, we perform the following steps to generate a set of measurements:

a) Interpolate the aggregate load profile to 6-second intervals, extract a length-m random consecutive subsequence, and then scale the real parts of bus power injections by the load factors in the subsequence.

b) Compute optimal power flow in Matpower for the network at each time step to determine bus voltage phasors.

c) Add a small amount of Gaussian random noise ($\sigma^2 = 0.001$) to the voltage measurements and generate corresponding current phasor measurements using the known admittance matrix.

Sample complexity for recovery of IEEE test cases

Figure 6.6.1 shows the sample complexity for accurate recovery of the IEEE test cases. The procedure and criteria for determining the necessary number of samples for accurate recovery of the admittance matrix are the same as for the synthetic data case. Unlike the previous setting, here we have no prior assumptions about the structure of the IEEE networks: networks have both mesh and radial topologies. However, because power system topologies are typically highly sparse, the heuristic algorithm was able to achieve accurate recovery with a comparable (logarithmic) dependence on graph size.

Influence of structure constraints on recovery

There are structural properties of the nodal admittance matrix for power systems—symmetry, sparsity, and entry-wise positivity/negativity—that we exploit in the heuristic algorithm to improve sample complexity for accurate recovery. The score function $\mathsf{score}_j(r)$ rewards symmetric consistency between columns in $\mathbf{X}$; the use of ℓ_1-minimization promotes sparsity in the recovered columns; and the constraint set $\mathcal{X}_j$ in (6.18) forces $\mathrm{Re}(X_{i,j}) \leq 0,\ \mathrm{Im}(X_{i,j}) \geq 0$ for $i \neq j$ and $\mathrm{Re}(X_{i,j}) \geq 0$ for $i = j$. These entry-wise properties are commonly found in power system admittance matrices. In Figure 6.6.2 we show the results of an experiment on the IEEE 30-bus test case that quantify the effects of the structure constraints on the probability of error. In Figure 6.6.3 we show that the score function and the constraints are effective across a range of IEEE test cases, compared with the standard compressed sensing recovery discussed in Section 6.1. Furthermore, this demonstrates the heuristic algorithm is robust to noise for a broad range of real-world graph structures with respect to Frobenius norm error.

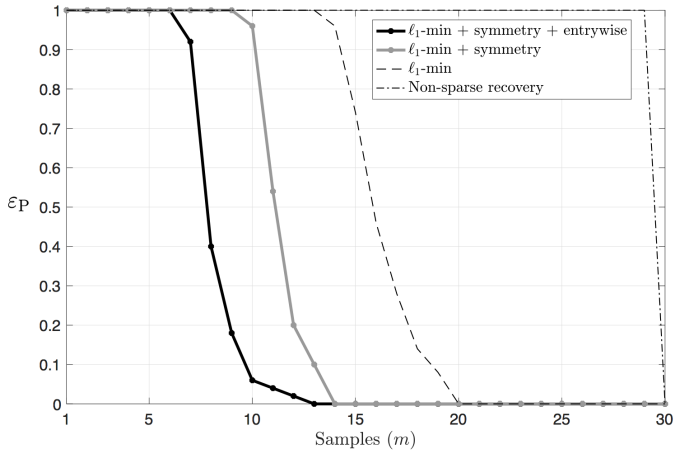


Figure 6.6.2: Probability of error for parameter reconstruction ε_{P} for the IEEE 30-bus test case is displayed on the vertical axis. Probability is taken over 50 independent trials. The horizontal axis shows the number of samples used to compute the estimate $\mathbf{X}$. The probability of error for independent recovery of all X_j via ℓ_1-norm minimization (double dashed line) and full rank non-sparse recovery (dot dashed line) are shown for reference. Adding the symmetry score function (second-to-left) improves over the naive column-wise scheme. Adding entry-wise positivity/negativity constraints on the entries of $\mathbf{X}$ (left-most curve) reduces sample complexity even further ($\approx 1/3$ samples needed compared to full rank recovery).

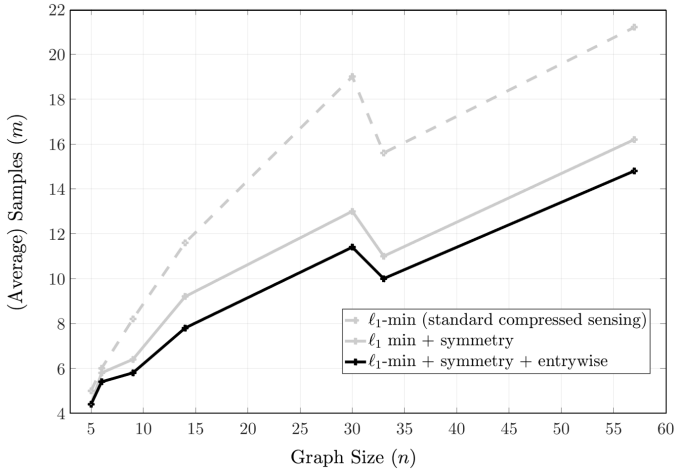


Figure 6.6.3: Sample complexity for accurate recovery is shown for a selection of IEEE power system test cases ranging from 5 to 57 buses. The number of samples for accurate recovery is obtained by satisfying the criterion $||\mathbf{X} - \mathbf{Y}||_{\mathrm{F}}/n^2 < 10^{-4}$. The noise $\mathbf{Z}$ is an IID Gaussian matrix with zero mean and standard deviation 0.01. The parameter γ in (6.19) is set to be 10^{-4}. As a benchmark, the number of measurements required for separately reconstructing every column of $\mathbf{Y}$ (standard compressed sensing) is also given.

Comparison with basis pursuit on star graphs

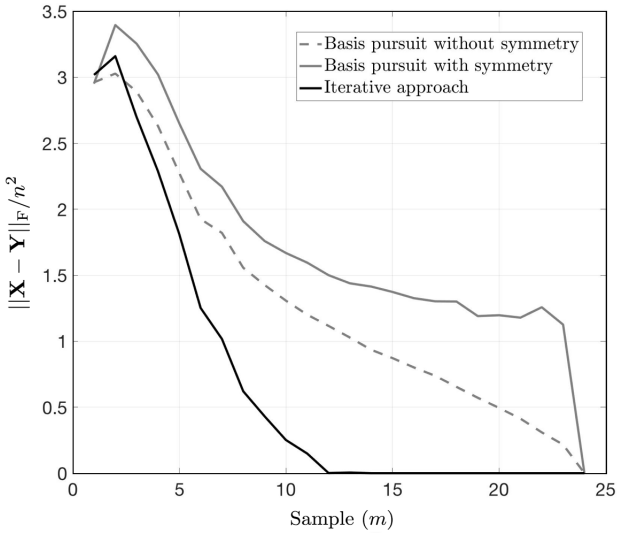


Figure 6.6.4: A comparison between our iterative heuristic and basis pursuit. The Frobenius norm error plotted is averaged over 250 independent trials. The underlying graph is a star graph with $n = 24$. The solid and dotted gray curves are results for basis pursuit with and without a constraint emphasizing symmetry, respectively.

In Figure 6.6.4, we consider star graphs and compare our heuristic algorithm with the modified basis pursuit subroutine in (6.2)-(6.4) with noiseless measurements. For a star graph with $n = 24$ nodes, the iterative recovery scheme with $s = 12$ outperforms the basis pursuit, with or without a symmetry constraint. The solid and dotted gray curves show the normalized Frobenius error for cases where $\mathbf{Y}(G)$ is constrained to be symmetric and where it is not, respectively. Our experiments show that convex optimization-based approach breaks down if there are highly dense columns in $\mathbf{Y}$. The star graph contains a high-degree node (degree $n - 1$), hindering the standard compressed sensing (basis pursuit without the symmetry constraint) from recovering the whole matrix until the number of measurements reaches n. Surprisingly, adding the symmetry constraint suggests basis pursuit performs less well than basis pursuit without the symmetry condition. This is evidence to support the assumption made in [161]. There, the non-zeros in the matrix to be recovered should not be concentrated in any single column (or row) of $\mathbf{Y}(G)$.

Effects of noise and selection of γ

In this section, we consider noisy measurements and fix the additive noise $\mathbf{Z}$ be IID Gaussian with mean zero and variance $\sigma_N^2 \in [10^{-9}, 10^{-2}]$. We set $\gamma = \sqrt{n}\sigma_N$ in

(6.19), as indicated in Remark 13. Due to the presence of noise, there is error in the recovered matrix $\mathbf{X}$. However, the mean absolute percentage error is small.

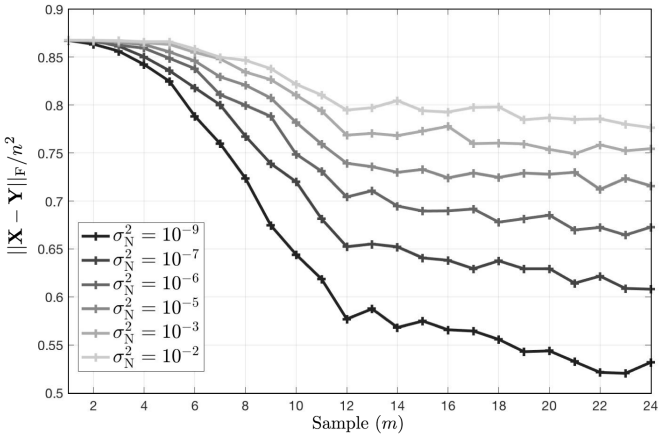


Figure 6.6.5: The impact of measurement noise on sample complexity for recovery of the IEEE 24-bus RTS test case is demonstrated. Trajectories correspond to increasing noise levels from dark (least) to light (most). From left to right, we observe—as expected—that for each variance value, the normalized Frobenius error of the recovered matrix decreases as the number of samples used for recovery increases. From bottom to top, we observe that the error increases (for every value of m) as variance of the additive noise $\mathbf{Z}$ increases.

APPENDIX

6.A Proof of Theorem 6.3.1

Proof. The graph G is chosen from a discrete set $\mathsf{C}(n)$ according to some probability distribution $\mathcal{G}_n$. Fano's inequality [165] plays an important role in deriving fundamental limits. We especially focus on its extended version. Similar generalizations appear in many places, *e.g.,* [164, 168] and [167]. We repeat the lemma here for the sake of completion:

Lemma 25 (Generalized Fano's inequality)**.** *Let G be a random graph and let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices defined in Section 6.2 and 6.2. Suppose the original graph G is selected from a nonempty candidacy set* $\mathsf{C}(n)$ *according to a probability distribution $\mathcal{G}_n$. Let $\hat{G}$ denote the estimated graph. Then the conditional probability of error for estimating G from* $\mathbf{A}$ *given* $\mathbf{B}$ *is always bounded from below as*

$$\mathbb{P}\left(\hat{G} \neq G \middle| \mathbf{B}\right) \geq 1 - \frac{\mathbb{I}\left(G; \mathbf{A} \middle| \mathbf{B}\right) + \ln 2}{\mathbb{H}\left(\mathcal{G}_n\right)} \tag{6.22}$$

where the randomness is over the selections of the original graph G and the estimated graph $\hat{G}$.

In (6.22), the term $\mathbb{I}\left(G; \mathbf{B} \middle| \mathbf{A}\right)$ denotes the conditional mutual information (base e) between G and $\mathbf{B}$ conditioned on $\mathbf{A}$. Furthermore, the conditional mutual information $\mathbb{I}\left(G; \mathbf{A} | \mathbf{B}\right)$ is bounded from above by the differential entropies of $\mathbf{A}$ and $\mathbf{B}$. It follows that

$$\begin{aligned}
\mathbb{I}\left(G; \mathbf{A} | \mathbf{B}\right) &= \mathbb{H}\left(\mathbf{A} | \mathbf{B}\right) - \mathbb{H}\left(\mathbf{A} | G, \mathbf{B}\right) && (6.23)\\
&\leq \mathbb{H}\left(\mathbf{A} | \mathbf{B}\right) - \mathbb{H}\left(\mathbf{A} | \mathbf{Y}, \mathbf{B}\right) && (6.24)\\
&= \mathbb{H}\left(\mathbf{A} | \mathbf{B}\right) - \mathbb{H}\left(\mathbf{Z}\right) && (6.25)\\
&\leq \mathbb{H}\left(\mathbf{A}\right) - \mathbb{H}\left(\mathbf{Z}\right). && (6.26)
\end{aligned}$$

Here, Eq. (6.23) follows from the definitions of mutual information and differential entropy. Moreover, knowing $\mathbf{Y}$, the graph G can be inferred. Thus, $\mathbb{H}\left(\mathbf{A} | G, \mathbf{B}\right) \geq \mathbb{H}\left(\mathbf{A} | \mathbf{Y}, \mathbf{B}\right)$ yields (6.24). Recalling the linear system in (6.1), we obtain (6.25). Furthermore, (6.26) holds since $\mathbb{H}\left(\mathbf{A}\right) \geq \mathbb{H}\left(\mathbf{A} | \mathbf{B}\right)$.

Plugging (6.26) into (6.22),

$$\begin{aligned}
\varepsilon_{\mathrm{T}} =& \mathbb{E}_{\mathbf{B}}\left[\mathbb{P}\left(\hat{G} \neq G \middle| \mathbf{B}\right)\right]\\
\geq& 1 - \frac{\mathbb{H}\left(\mathbf{A}\right) - \mathbb{H}\left(\mathbf{Z}\right) + \ln 2}{\mathbb{H}\left(\mathcal{G}_n\right)},
\end{aligned}$$

which yields the desired (6.9). □

6.B Proof of Theorem 6.3.2

Conditioning on that no less than $n - K$ many columns are recovered with respect to the Γ-probability of error, *i,e.,* for each entry, the absolute value of the difference between the recovered one and the original one is bounded from above by γ, the union bound ensures the desired bound on the probability of error for noisy parameter reconstruction. It remains to show that the consistency-check in our scheme gives the expression for η. First, if no less than $n - K$ many columns are recovered, there must be a subset $\mathcal{S} \subseteq \mathcal{V}$ passing through the consistency-check. Let us consider the vectors that are not μ-sparse. For any such vector $Y^* \in \mathbb{R}^n$, denote by $e = Y^* - Y'$ the difference of Y^* and the original vector Y'. It follows that e can be decomposed as a summation of a $2K$-sparse vector $\overline{e} \in \mathbb{R}^n$ and a vector $f \in \mathbb{R}^n$ that satisfies $|f_i| \leq 2\Gamma$ for all $i \in \mathcal{V}$. Therefore, the definition of restricted isometry constants ensures the following:

$$\begin{aligned} ||e||_2 \leq & ||\overline{e}||_2 + ||f||_2 \\ \leq & \frac{1}{1-\delta_{2K}} ||\mathbf{B}\overline{e}||_2 + 2n\Gamma \\ \leq & \frac{1}{1-\delta_{2K}} ||\mathbf{B}e||_2 + \left(2n + \frac{2||\mathbf{B}||_2}{1-\delta_{2K}}\right)\Gamma \end{aligned}$$

which can be further bounded by noting that

$$||\mathbf{B}e||_2 = ||(\mathbf{B}Y^* - A) - (\mathbf{B}Y' - A)||_2 \leq 2\gamma$$

since both Y' and Y^* satisfy (6.11b) where A is a column of $\mathbf{A}$. Thus, the consistency-check guarantees that for each j in the set $\mathcal{S} \subseteq \mathcal{V}$ that passes the check,

$$||X_j - Y_j||_2 \leq 2\left(n + \frac{||\mathbf{B}||_2}{1-\delta_{2K}}\right)\Gamma + \frac{2\gamma}{1-\delta_{2K}}.$$

Consider the reduced linear system in (6.12). For each j in the set $\overline{\mathcal{S}} \subseteq \mathcal{V}$,

$$\begin{aligned} ||X_j^{\overline{\mathcal{S}}} - Y_j^{\overline{\mathcal{S}}}||_2 \leq & \left\| \left(\mathbf{B}_{\overline{\mathcal{S}}}^{\mathcal{K}}\right)^{-1} \right\|_2 \left\| \mathbf{B}_{\mathcal{S}}(X_j^{\mathcal{S}} - Y_j^{\mathcal{S}}) \right\|_2 \\ \leq & \left\| \left(\mathbf{B}_{\overline{\mathcal{S}}}^{\mathcal{K}}\right)^{-1} \right\|_2 ||\mathbf{B}_{\mathcal{S}}||_2 \left\| X_j^{\mathcal{S}} - Y_j^{\mathcal{S}} \right\|_2 . \end{aligned}$$

Summing up the bounds on the ℓ_2 norms for each column and considering the worst case of the invertible matrix $\mathbf{B}_{\overline{\mathcal{S}}}^{\mathcal{K}}$, the bound η on the Frobenius norm follows by arranging the terms.

6.C Proof of Corollary 6.3.1

Proof. Conditioned on $G \in \mathsf{C}(n)(\mu, K)$ and the assumption $\delta_{3\mu} + 3\delta_{4\mu} < 2$, there are no less than $n - K$ many columns correctly recovered. Therefore, any such set $\mathcal{S}$ with $|\mathcal{S}| = n - K$ must contain at least $n - 2K$ many corresponding indexes of the correctly recovered columns. The consistency-checking verifies that if the collection of an arbitrary set of nodes $\mathcal{S}$ of cardinality $n - K$ satisfies the symmetry property as the true graph $\mathbf{Y}$ must obey. If the consistency-checking fails, it is necessary that there exist two distinct length-n vectors Y' and Y^* in $\mathbb{F}^n$ such that Y^* is the minimizer of the ℓ_1-minimization (6.11a)-(6.11c) that differs from the correct answer Y', *i.e.,* $Y' \neq Y^*$ where $A = \mathbf{B}Y'$ and

$$\begin{aligned} Y^* = \arg\min_{Y} \ & ||Y||_1 \\ \text{subject to } & A = \mathbf{B}Y \\ & Y \in \mathbb{F}^n \end{aligned}$$

for some $A \in \mathbb{F}^m$ and furthermore, the vectors Y' and Y^* can have at most $2K$ distinct coordinates,

$$|\text{supp}\,(Y' - Y^*)| \leq 2K.$$

However, the constraints $\mathbf{B}Y' = A$ and $\mathbf{B}Y^* = A$ imply that $\mathbf{B}\,(Y' - Y^*) = 0$, contradicting to $\text{spark}(\mathbf{B}) > 2K$. Therefore, $n - K$ many columns can be successfully recovered if the decoded solution passes the consistency-checking. Moreover, since $\text{spark}(\mathbf{B}) > 2K$ and number of unknown coordinates in each length-K vector $X_j^{\overline{\mathcal{S}}}$ (for $j = 1, \ldots, |\overline{\mathcal{S}}|$) to be recovered is K, the solution of the system (6.12) is guaranteed to be unique. Thus, Algorithm 10 always recovers the correct columns $Y_1, \ldots, Y_N$ conditioned on $\text{spark}(\mathbf{B}) > 2K$. It follows that $\varepsilon_{\text{P}} \leq 1 - \mathbb{P}_{\mathcal{G}}(G \in \mathsf{C}(n, \mu, K))$ provided $\text{spark}(\mathbf{B}) > 2K$. In agreement with the assumption that the distribution $\mathcal{G}$ is (μ, K, ρ)-sparse, (6.10) must be satisfied. Therefore, the probability of error must be less than ρ. □

6.D Proof of Lemma 22

Proof. Consider the following function

$$F(\mathcal{E}) = \sum_{j=1}^{n} f(d_j(G))$$

where $d_j(G)$ denotes the degree of the j-th node and consider the following indicator function:

$$f(d_j(G)) := \begin{cases} 1 & \text{if } d_j(G) > \mu \\ 0 & \text{otherwise} \end{cases}.$$

Applying the Markov's inequality,

$$\begin{aligned} \mathbb{P}\left(G \notin \mathsf{T}(n)(\mu, K)\right) &= \mathbb{P}_{\mathcal{U}_{\mathsf{T}(n)}}\left(F(\mathcal{E}) \geq K\right) \\ &\leq \frac{\mathbb{E}_{\mathcal{U}_{\mathsf{T}(n)}}\left[F(\mathcal{E})\right]}{K}. \end{aligned} \tag{6.27}$$

Continuing from (6.27), the expectation $\mathbb{E}_{\mathcal{U}_{\mathsf{T}(n)}}\left[F(\mathcal{E})\right]$ can be further expressed and bounded as

$$\begin{aligned} \mathbb{E}_{\mathcal{U}_{\mathsf{T}(n)}}\left[F(\mathcal{E})\right] &= \sum_{j=1}^{n} \mathbb{E}_{\mathcal{U}_{\mathsf{T}(n)}}\left[f(d_j(G))\right] \\ &= \sum_{j=1}^{n} \mathbb{P}_{\mathcal{U}_{\mathsf{T}(n)}}\left(d_j(G) > \mu\right). \end{aligned} \tag{6.28}$$

Since G is chosen uniformly at random from $\mathsf{T}(n)$, it is equivalent to selecting its corresponding Prüfer sequence (by choosing $n-2$ integers independently and uniformly from the set $\mathcal{V}$, *c.f.* [200]) and the number of appearances of each $j \in \mathcal{V}$ equals to $d_j(G) - 1$. Therefore, for any fixed node $j \in \mathcal{V}$, the Chernoff bound implies that

$$\mathbb{P}_{\mathcal{U}_{\mathsf{T}(n)}}\left(d_j(G) > \mu\right) \leq \exp\left(-(n-2)\mathbb{D}_{\mathrm{KL}}\left(\frac{\mu}{n-2}\middle\|\frac{1}{n}\right)\right) \tag{6.29}$$

where $\mathbb{D}_{\mathrm{KL}}(\cdot||\cdot)$ is the Kullback-Leibler divergence and

$$\mathbb{D}_{\mathrm{KL}}\left(\frac{\mu}{n-2}\middle\|\frac{1}{n}\right) \geq \frac{\mu}{n-2}\ln n. \tag{6.30}$$

Therefore, substituting (6.30) back into (6.29) and combining (6.27) and (6.28), setting $\mu \geq 1$ leads to

$$\mathbb{P}\left(G \notin \mathsf{T}(n)(\mu, K)\right) \leq \frac{n\exp(-\mu\ln n)}{K} \leq \frac{1}{K}.$$

□

6.E Proof of Lemma 23

Proof. For any fixed node $j \in \mathcal{V}$, applying the Chernoff bound,

$$\mathbb{P}_{\mathcal{G}_{\mathrm{ER}}(n,p)}\left(d_j(G) > \mu\right) \leq \exp\left(-n\mathbb{D}_{\mathrm{KL}}\left(\frac{\mu}{n}\middle\|p\right)\right).$$

Continuing from (6.27), the expectation $\mathbb{E}_{\mathcal{G}_{\mathrm{ER}}(n,p}\left[F(\mathcal{E})\right]$ can be further expressed and bounded as

$$\mathbb{E}_{\mathcal{G}_{\mathrm{ER}}(n,p)}\left[F(\mathcal{E})\right] \leq n \cdot \exp\left(-n\mathbb{D}_{\mathrm{KL}}\left(\frac{\mu}{n}\middle\|p\right)\right) \tag{6.31}$$

where the probability p satisfies $0 < p \leq \mu/n < 1$. Note that

$$\mathbb{D}_{\mathrm{KL}}\left(\frac{\mu}{n}\middle\|p\right) = \frac{\mu}{n}\ln\frac{1}{p} + \left(1-\frac{\mu}{n}\right)\ln\frac{1}{1-p} - h(p) \tag{6.32}$$

where the binary entropy $h(p)$ is in base e. Taking $\mu \geq 2nh(p)/(\ln 1/p) \geq 2np$, substituting (6.32) into (6.31) leads to

$$\mathbb{E}_{\mathcal{G}_{\mathrm{ER}}(n,p)}\left[F(\mathcal{E})\right] \leq n\exp\left(-nh(p)\right).$$

Therefore, (6.27) gives

$$\mathbb{P}\left(G \notin \mathsf{C}(n)(\mu, K)\right) \leq \frac{n\exp\left(-nh(p)\right)}{K}.$$

□

6.F Proof of Lemma 24

Proof. Continuing from Theorem 6.3.1,

$$\begin{aligned}
&\mathbb{H}(\mathbf{A}) - \mathbb{H}(\mathbf{Z}) \\
&= \sum_{i=1}^{m}\left[\mathbb{H}\left(A^{(i)}\right) - \mathbb{H}\left(Z^{(i)}\right)\right] \\
&\overset{\text{(a)}}{\leq} \sum_{i=1}^{m}\frac{n}{2}\left[\ln\left(2\pi e\frac{\mathrm{Tr}\left(\Sigma_{\mathbf{A}^{(\mathbf{i})}}\right)}{n}\right) - \ln(2\pi e\sigma_{\mathrm{N}}^2)\right]
\end{aligned} \tag{6.33}$$

where $\mathrm{Tr}\left(\Sigma_{\mathbf{A}^{(\mathbf{i})}}\right)$ is the trace of the covariance matrix of $\mathbf{A}^{(\mathbf{i})}$ and we have used the fact that normal distributions maximize entropy and the inequality $\det(\Sigma_{\mathbf{A}^{(\mathbf{i})}}) \leq (\mathrm{Tr}\left(\Sigma_{\mathbf{A}^{(\mathbf{i})}}\right)/n)^n$ to obtain (a). Note that because of the assumption of independence, the trace is bounded from above by $n\sigma_{\mathrm{S}}^2\overline{Y} + n\sigma_{\mathrm{N}}^2$ where $\overline{Y} := \max_{i,j}|Y_{i,j}|$. Substituting this into (6.33) completes the proof. The special case when $\mathbf{Z} = 0$ follows similarly.

□

6.G Proof of Theorem 6.4.1

Proof. The first part is based on Corollary 6.3.1. Under the assumption of the generator matrix $\mathbf{B}$, using Gordon's escape-through-the-mesh theorem, Theorem 4.3 in [183] implies that for any columns Y_j with $j \in \mathcal{V}_{\text{Small}}$ are correctly recovered using the minimization in (6.11a)-(6.11c) with probability at least $1 - 2.5 \exp\left(-(4/9)\mu \log(n/\mu)\right)$, as long as the number of measurements satisfies $m \geq 48\mu\,(3 + 2\log(n/\mu))$, and $n/\mu > 2, \mu \geq 4$ (if $\mu \leq 3$, the multiplicative constant increases but our theorem still holds). Similar results were first proved by Candes, *et al.* in [182] (see their Theorem 1.3). Therefore, applying the union bound, the probability that all the μ-sparse columns can be recovered simultaneously is at least $1 - 2.5n \exp\left(-(4/9)\mu \log(n/\mu)\right)$. On the other hand, conditioned on that all the μ-sparse columns are recovered, Corollary 6.3.1 indicates that $\text{spark}(\mathbf{B}) > 2K$ is sufficient for the three-stage scheme to succeed. Since each entry in $\mathbf{B}$ is an IID Gaussian random variable with zero mean and variance one, if $m \geq 48\mu\,(3 + 2\log(n/\mu)) + 2K$, with probability one that the spark of $\mathbf{B}$ is greater than $2K$, verifying the statement.

The converse follows by applying Lemma 24 with $\mathbf{Z} = 0$. Consider the uniform distribution $\mathcal{U}_{\mathsf{C}(n)(\mu,K)}$ on $\mathsf{C}(n)(\mu, K)$. Then $\mathbb{H}\left(\mathcal{U}_{\mathsf{C}(n)(\mu,K)}\right) = \ln|\mathsf{C}(n)(\mu, K)|$. Let $0 \leq \alpha, \beta \leq 1$ be parameters such that $\mu < \beta(n - \alpha K)$. To bound the size of $\mathsf{C}(n)(\mu, K)$, we partition $\mathcal{V}$ into $\mathcal{V}_1$ and $\mathcal{V}_2$ with $|\mathcal{V}_1| = n - \alpha K$ and $|\mathcal{V}_2| = \alpha K$. First, we assume that the nodes in $\mathcal{V}_1$ form a $\mu/2$-regular graph. For each node in $\mathcal{V}_2$, construct $\beta(n - \alpha K) \in \mathbb{N}_+$ edges and connect them to the other nodes in $\mathcal{V}$ with uniform probability. A graph constructed in this way always belongs to $\mathsf{C}(n)(\mu, K)$, unless the added edges create more than K nodes with degrees larger than μ. Therefore, as $n \to \infty$,

$$|\mathsf{C}(n)(\mu, K)| \geq \rho \cdot \frac{e^{1/4}\binom{N-1}{\phi}^N \binom{\binom{N}{2}}{\phi N/2}}{\binom{N(N-1)}{\phi N}} \cdot \binom{n-1}{M}^{\alpha K} \tag{6.34}$$

where $N := n - \alpha K$, $M := \beta(n - \alpha K)$ and $\phi := \mu/2$. The first term ρ denotes the fraction of the constructed graphs that are in $\mathsf{C}(n)(\mu, K)$. The second term in (6.34) counts the total number of ϕ-regular graphs, and the last term is the total number of graphs created by adding new edges for the nodes in $\mathcal{V}_2$. If $K = O(\mu)$, there exists a constant $\alpha > 0$ small enough such that $\rho = 1$. If $\mu = o(K)$, for any fixed node in $\mathcal{V}_1$,

the probability that its degree is larger than μ is

$$\sum_{i=\phi+1}^{\alpha K} \binom{\alpha K}{i} \beta^i (1-\beta)^{\alpha K - i}$$
$$\leq \sum_{i=\phi+1}^{\alpha K} \alpha K h\left(\frac{i}{\alpha K}\right) \beta^i \leq (\alpha K)^2 \beta^{\phi+1}$$

where $h(i/\alpha K)$ is in base e. Take $\beta = n^{-3/\mu}$ and $\alpha = 1/2$. The condition $\mu < n^{-3/\mu}(n-K)$ guarantees that $\mu < \beta(n - \alpha K)$. Letting $F(n) := 1/n$ be the assignment function for each node in $\mathcal{V}_1$, we check that

$$(\alpha K)^2 \beta^{\phi+1} \leq \frac{1}{4n} \leq F(n) \cdot \left(1 - \frac{1}{F(n)}\right)^N \leq \frac{1}{en}.$$

Therefore, applying the Lovász local lemma, the probability that all the nodes in $\mathcal{V}_1$ have degree less than or equal to μ can be bounded from below by $(1 - F(n))^N \geq 1/4$ if $n \geq 2$, which furthermore is a lower bound on ρ. Therefore, taking the logarithm,

$$\begin{aligned}
\mathbb{H}\left(\mathcal{U}_{\mathsf{C}(n)(\mu,K)}\right) \geq & \frac{(N-1)^2}{2} h(\varepsilon) - O(N \ln \mu) \\
& + \frac{K}{2}\left((n-1) h\left(\frac{M}{n-1}\right) - O(\ln n)\right) - O(1) \qquad (6.35) \\
= & \Omega\left(n^2 h(\varepsilon) + n^{1-3/\mu} K\right) \qquad (6.36)
\end{aligned}$$

where $\varepsilon := \phi/(N-1) \leq 1/2$. In (6.35), we have used Stirling's approximation and the assumption that $K = o(n)$. Continuing from (6.36), since $2nh(\varepsilon) \geq \mu \ln(n/\mu)$, for sufficiently large n,

$$\mathbb{H}\left(\mathcal{U}_{\mathsf{C}(n)(\mu,K)}\right) = \Omega\left(n\mu \log \frac{n}{\mu} + n^{1-3/\mu} K\right). \qquad (6.37)$$

Substituting (6.37) into (6.16), when $n \to \infty$, it must hold that

$$m = \Omega\left(\mu \log(n/\mu) + K/n^{3/\mu}\right)$$

to ensure that ε_{P} is smaller than $1/2$. □

6.H Proof of Theorem 6.4.2

The structure of the proof is the same as Theorem 6.4.1. The converse follows directly by putting the bounds in (6.37) and (6.15) together. For proving the achievability, it is sufficient to show that with high probability (in n), $|Y_{i,j} - X_{i,j}| = o(1)$ for all $i, j \in \mathcal{V}$ where $X_{i,j}$ and $Y_{i,j}$ are the recovered and original (i,j)-th entry of the

graph matrix. For the Gaussian IID ensemble considered, the ℓ_2-norm of the inverse matrix $(\mathbf{B}^{\mathcal{K}}_{\mathcal{S}})^{-1}$, equivalently, the minimal singular value of $\mathbf{B}^{\mathcal{K}}_{\mathcal{S}}$ is strictly positive with probability $o(1)$ (see the proof of Lemma III-9 in [168]). Using the Chernoff bound, with high probability,

$$||\mathbf{B}||_2^2 \leq ||\mathbf{B}||_{\mathrm{F}}^2 \leq C_1 nm\sigma_{\mathrm{S}}^2, \tag{6.38}$$

$$||Z_j||_2^2 \leq C_2 n\sigma_{\mathrm{N}}^2, \text{ for all } j \in \mathcal{V} \tag{6.39}$$

for some positive constants C_1 and C_2. Noting that if $K \leq \mu$, then $\delta_{2K} < 1$ with high probability, the bound in (6.38) and the bound on the ℓ_2-norm of the inverse matrix $(\mathbf{B}^{\mathcal{K}}_{\mathcal{S}})^{-1}$ imply $\eta = O(n^2\gamma)$, by applying our Theorem 6.3.2. Moreover, with Gaussian measurements, for each μ-sparse vector Y_j in $\mathbb{R}^n$, $||X_j - Y_j||_2 \leq C_3||Z_j||_2$ for some constant $C_3 > 0$ (*cf.* Theorem 1 in [201]) where Y_j satisfies $\mathbf{B}Y_j + Z_j = A_j$ and X_j is the optimal solution of (6.11a)-(6.11c) (with $\mathbb{F} \equiv \mathbb{R}$). Therefore, $\Gamma = O(\gamma)$ and $\gamma = O(\sqrt{n}\sigma_{\mathrm{N}})$ using (6.39). Since $\eta = O(n^2\gamma)$, the condition $\sigma_{\mathrm{N}} = o(1/n^{5/2})$ guarantees that $\eta = o(1)$, whence $|Y_{i,j} - X_{i,j}| = o(1)$ for all $i, j \in \mathcal{V}$ and the proof is complete.

Chapter 7

ELECTRIC VEHICLE CHARGING DATA ANALYSIS

[1] Zachary J. Lee, Tongxin Li, and Steven H. Low. Acn-data: Analysis and applications of an open ev charging dataset. New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366717. URL https://doi.org/10.1145/3307772.3328313.

[2] Classification of electric vehicle charging time series with selective clustering. *Electric Power Systems Research*, 189:106695, 2020. ISSN 0378-7796. URL https://doi.org/10.1016/j.epsr.2020.106695.

Electric vehicles (EVs) have the potential to drastically reduce the carbon-footprint of the transportation sector. According to the International Energy Agency (IEA)[202], the global electric vehicle (EV) stock will exceed 130 million vehicles by 2030. This trend has motivated a large body of EV research in the last decade, from pilot studies to testbeds and data analytics, from charging algorithms to user behavior to optimal investments, from impact on electric grid [203–207] to energy services such as reducing demand variability [208, 209], minimizing costs when subject to time-varying prices [208–210], taking advantage of intermittent renewable resources [211–214], or meeting charging demands using limited infrastructure capacity [101, 215]. While some of these studies, for example [101, 204, 206, 209, 215, 216], have had access to real EV data to analyze their proposed algorithms, many others have had to rely on distributions derived from data collected from internal combustion engine (ICE) vehicles [207, 208, 210–212] or assumed behaviors [203, 205, 213]. In addition, since all of these studies utilize different data sources, it can be difficult to compare one algorithm or approach against another. In this chapter we first introduce basic attributes of a public EV charging dataset, then present basic EV charging data analysis methods to predict user behavior and learn battery behavior based on fine-grained charging data from the field. The dataset is used to test and validate various algorithms, methods, and practical implementations, including the results described in the first two parts of this dissertation.

7.1 The ACN-DATA Dataset

In this section, we describe the dataset. More details on the charging facility and adaptive algorithm can be found in [101].

Table 7.1.1: Selected data fields in ACN-Data.

Field	Description
connectionTime	Time when the user plugs in.
doneChargingTime	Time of the last non-zero charging rate.
disconnectTime	Time when the user unplugs.
kWhDelivered	Measured Energy Delivered
siteID	Identifier of the site where the session took place.
stationID	Unique identifier of the EVSE.
sessionID	Unique identifier for the session.
timezone	Timezone for the site.
pilotSignal	Time series of pilot signals during the session.
chargingCurrent	Time series of actual charging current of the EV.
userID*	Unique identifier of the user.
requestedDeparture*	Estimated time of departure.
kWhRequested*	Estimated energy demand.

*Field not available for every session.

Adaptive Charging Network (ACN)

ACN-Data was collected from two Adaptive Charging Networks located in California. The ACN on the Caltech campus is in a parking garage and has 54 EVSEs (Electric Vehicle Supply Equipment or charging stations) along with a 50 kW dc fast charger. The Caltech ACN is open to the public and is often used by non-Caltech drivers. Since the parking garage is near the campus gym, many drivers charge their EVs while working out in the morning or evening. JPL's ACN includes 52 EVSEs in a parking garage. In contrast with Caltech, access to the JPL campus is restricted and only employees are able to use the charging system. The JPL site is representative of workplace charging while Caltech is a hybrid between workplace and public use charging. EV penetration is also quite high at JPL. This leads to high utilization of the EVSEs as well as an ad-hoc program where drivers move their EVs after they have finished charging to free up plugs for other drivers. In both cases, to reduce capital costs, infrastructure elements such as transformers have been oversubscribed. The current architecture of the ACN for Caltech is described in [101] though both systems have a similar structure.

The ACN framework allows us to collect detailed data about each charging session which occurs in the system. Table 7.1.1 describes some of the relevant data fields we collect. To obtain data directly from users, we use a mobile application. The driver first scans a QR code on the EVSE which allows us to associate the driver with a particular charging session. The driver is then able to input their estimated departure time and requested energy. We refer to this as user input data. When a user does

not use the mobile application, default values for energy requested and duration are assumed and no user identifier is attached to the session. We refer to sessions with an associated user input as claimed and those without as unclaimed.

An ACN typically consists of tens of level-2 chargers controlled by a local controller that communicates wirelessly with these chargers and servers in the cloud. An ACN is capable of real-time measurement, communication, computing and control. It adapts EV charging currents to driver needs as well as capacity limits of the electric system. A typical charging session starts when a driver plugs in her EV and informs ACN through a mobile app the amount of energy required (in terms of miles) and her estimated departure time. The EV will be charged until either the requested energy is delivered, or the battery is fully charged, or the EV is unplugged, whichever occurs first. The charging currents of all EVs that have not finished charging are jointly optimized and updated every minute. Every 5 to 10 seconds, a control (pilot) signal is sent to the EV and the actual charging current drawn by the vehicle is measured. ACN-Data contains both session data (user's ID, arrival time, departure time, requested energy, and actual energy delivered) and fine-grained charging data at seconds resolution (time series of control signals and charging currents). Unfortunately, the current EV charging standard does not collect batteries' states of charge nor EV specifications. Table 7.4.1 summarizes some of the available features of ACN-Data used in this work. Note that not all sessions contain user inputs (*i.e.*, the last three fields of Table 7.1.1.) In this chapter, we shall focus on the claimed sessions that are associated with user inputs.

7.2 Learning User Behavior

In this section, we illustrate how to learn the underlying joint distribution of arrival time, session duration, and energy delivered using Gaussian mixture models (GMMs) (e.g., [217, 218]). We then use these GMMs to predict user behavior in Section 7.3.

Problem Formulation

We utilize the GMM as a second-order approximation to the underlying distribution. Our dataset can be modeled as follows to fit a GMM. Consider a dataset X consisting of N charging sessions. The data for each session $i = 1, \ldots, N$, is represented by a triple $x_i = (a_i, d_i, e_i)$ in $\mathbb{R}^3$ where a_i denotes the arrival time, d_i denotes the duration and e_i is the total energy (in kWh) delivered. The data point X_i (we use capital letters for random variables) are independently and identically distributed (i.i.d.) according to some unknown distribution. In practice, each driver in a workplace

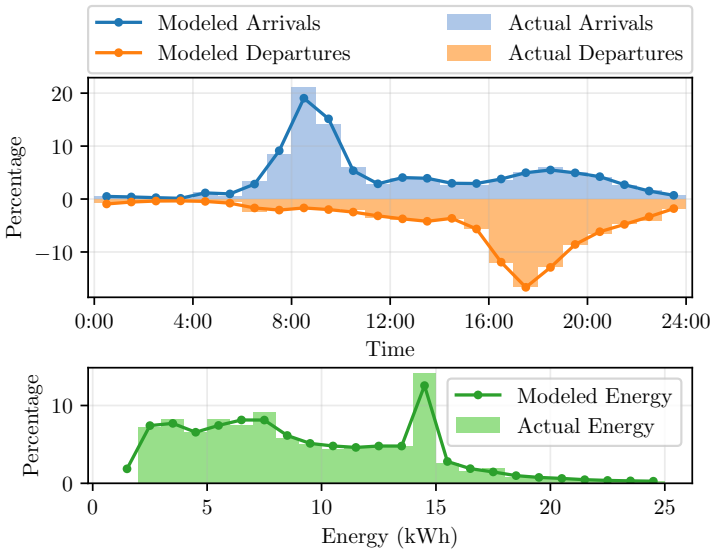


Figure 7.1.1: Comparison of model distributions with actual data for Caltech during training period.

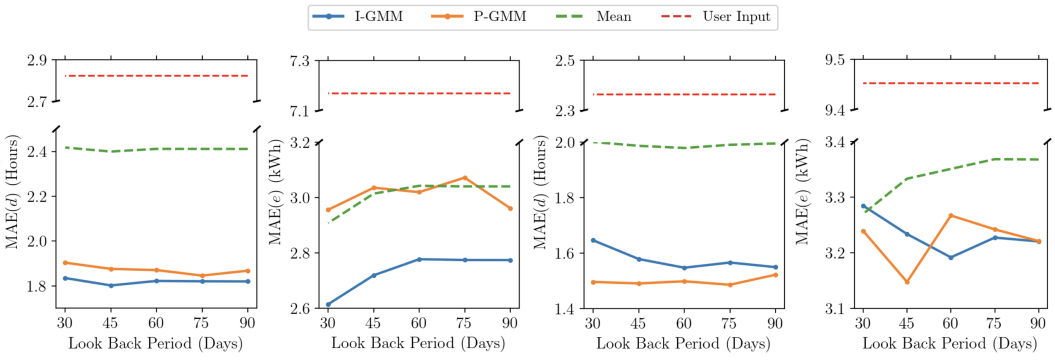


Figure 7.1.2: Prediction errors for Caltech (left two columns) and JPL (right two columns) for training dataset sizes ranging from 30 days to 90 days in the past. As a benchmark, we consider simply taking the mean of each user's prior behavior. For comparison, we also include the errors of user inputs. The results are measured by the mean absolute error (MAE) defined in (7.3).

environment exhibits only a few regular patterns. For example, on weekdays, a driver may typically arrive at 8 am and leave around 6 pm, though her actual arrival and departure times may be randomly perturbed around their typical values. On weekends, driver behavior may change such that the same driver may come around noon. We hence assume that drivers have finitely many behavior profiles. Therefore, let K be the number of typical profiles denoted by $\mu_1, \ldots, \mu_K$.[1] Each data point X_i

[1]We assume the number K of components is known. In our experiments in Section 7.2, grid search [219] and cross-validation is used to find the best number of components.

can be regarded a corrupted version of a typical profile with a certain probability. Define a latent variable $Y_i \equiv k$ if and only if X_i is corrupted from μ_k. Moreover, by the i.i.d. assumption, each incoming EV has an identical probability π_k taking μ_k, *i.e.,* $\pi_k := \mathbb{P}(Y_i = k)$ for $i = 1, \ldots, N,\ k = 1, \ldots, K$. Conditioned on $Y_i = k$, the difference $X_i - \mu_k$ that the profile X_i deviates from the typical profile μ_k can be regarded as Gaussian noise. In this manner, assuming $Y_i = k$, we let $X_i \sim \mathcal{N}(\mu_k, \Sigma_k)$ be a Gaussian random variable with mean μ_k and covariance matrix Σ_k. To estimate the underlying distribution and approximate it as a mixture of Gaussians, it suffices to estimate the parameters $\theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K$. The probability density of observing a data point x can then be approximated using the learned GMM as

$$p(x|\theta) = \sum_{k=1}^{K} \pi_k \frac{\exp\left(-||x - \mu_k||^2_{\Sigma_k^{-1}}/2\right)}{\sqrt{(2\pi)^3 \det(\Sigma_k)}}.$$

Population and Individual-level GMMs

We train GMMs based on a training dataset $\mathsf{X}_{\text{Train}}$ and predict the charging duration and energy delivered for drivers in a set $\mathcal{U}$. The results are tested on a corresponding testing dataset X_{Test}. The training data collected at both Caltech and JPL can be divided into two parts: user-claimed data X_{C} and unclaimed data X_{U}.

This motivates us to study two different approaches. The first approach generates a population-level GMM (P-GMM) based on the overall training data $\mathsf{X}_{\text{Train}} = \mathsf{X}_{\text{C}} \bigcup \mathsf{X}_{\text{U}}$. However, users can have distinctive charging behaviors. To achieve better prediction accuracy, we take advantage of the user-claimed data and predict the charging duration and energy delivered for each individual user. In the second approach, the claimed data can be partitioned into a collection of smaller datasets consisting of the charging information of each user in $\mathcal{U}$. We write $\mathsf{X}_{\text{C}} = \bigcup_{j \in \mathcal{U}} \mathsf{X}_j$. We can then train individual-level GMMs (I-GMM) for each user $j \in \mathcal{U}$ by fine tuning the weights of the components of the P-GMM with data from each of the users to arrive at a final model for each of them.

Distribution Learned by P-GMM

To evaluate how well our learned population-level GMM fits the underlying distribution, we gather $100,000$ samples from a P-GMM trained on data from Caltech collected prior to Sep. 1, 2018. We then plot in Figure 7.1.1 the distribution of these samples along with the empirical distribution from our training set. We choose to plot departure time instead of duration directly as this demonstrates that our model has learned not only the distribution of session duration but also the correlation

between arrival time and duration. We see that in all cases, our learned distribution matches the empirical distribution well. We next present the applications of the ACN-Data dataset and the learned distribution in Sections 7.3.

7.3 Predicting User Behavior

In this section, we use the GMM that we have learned from the ACN-Data dataset to predict a user's departure time and the associated energy consumption based on their known arrival time. Despite recent advances in arrival time based prediction via kernel density estimation [216, 220, 221], simple empirical predictions are commonly used in practical EV charging systems. For example, the ACNs from which this data was collected use user inputs directly in the scheduling problem [101], while other charging systems simply take the average of the past behavior as a prediction. Our data, however, shows that user input can be quite unreliable, partially because of a lack of incentives for users to provide accurate predictions. We demonstrate that the predictions can be more precise using simple probabilistic models.

Calculating Arrival Time-Based Predictions

Let $\mathcal{U}$ denote the set of users. Suppose a convergent solution $\theta^{(j)} = (\pi_k^{(j)}, \mu_k^{(j)}, \Sigma_k^{(j)})_{k=1}^K$ is obtained for user $j \in \mathcal{U}$ where $\mu_k^{(j)} := (a_k^{(j)}, d_k^{(j)}, e_k^{(j)})$ and the user's arrival time is known *a priori* as $\alpha^{(j)}$. For the sake of completeness, we present the following formulas used for predicting the duration $\delta^{(j)}$ and energy to be delivered $\varepsilon^{(j)}$ as conditional Gaussians of the user $j \in \mathcal{U}$:

$$\delta^{(j)} = \sum_{k=1}^{K} \overline{\pi}_k^{(j)} \left(d_k^{(j)} + (\alpha^{(j)} - a_k^{(j)}) \frac{\Sigma_k^{(j)}(1,2)}{\Sigma_k^{(j)}(1,1)} \right), \tag{7.1}$$

$$\varepsilon^{(j)} = \sum_{k=1}^{K} \overline{\pi}_k^{(j)} \left(e_k^{(j)} + (\alpha^{(j)} - a_k^{(j)}) \frac{\Sigma_k^{(j)}(1,3)}{\Sigma_k^{(j)}(1,1)} \right), \tag{7.2}$$

where $\Sigma_k^{(j)}(1,1)$, $\Sigma_k^{(j)}(1,2)$ and $\Sigma_k^{(j)}(1,3)$ are the first, second and third entries in the first column (or row) of the covariance matrix $\Sigma_k^{(j)}$, respectively. Denoting by $p(\cdot|\mu, \sigma^2)$ the probability density for a normal distribution with mean μ and variance σ^2, the modified weights conditioned on arrival time in (7.1) and (7.2) above are

$$\overline{\pi}_k := \frac{p\left(\alpha^{(j)} | a_k^{(j)}, \Sigma_k^{(j)}(1,1)\right)}{\sum_{k=1}^{K} p\left(\alpha^{(j)} | a_k^{(j)}, \Sigma_k^{(j)}(1,1)\right)}.$$

Error Metrics

We consider both absolute error and percentage error when evaluating duration and energy predictions.

Mean absolute error

Recall that $\mathcal{U}$ is the set of all users in a testing dataset X_{Test}. Let $\mathcal{A}_j$ denote the set of charging sessions for user $j \in \mathcal{U}$. The Mean Absolute Error (MAE) is defined in (7.3) to assess the overall deviation of the duration and energy consumption. For a testing dataset $\mathsf{X}_{\text{Test}} = \{(a_{i,j}, d_{i,j}, e_{i,j})\}_{j\in\mathcal{U}, i\in\mathcal{A}_j}$, the corresponding MAEs for duration and energy are represented by MAE(d) and MAE(e) with

$$\text{MAE}(x) := \sum_{j\in\mathcal{U}} \frac{1}{|\mathcal{U}|} \sum_{i\in\mathcal{A}_j} \frac{1}{|\mathcal{A}|_j} \left| x_{i,j} - \widehat{x}_{i,j} \right| \tag{7.3}$$

where $\widehat{x}_{i,j}$ is the estimate of $x_{i,j}$ and $x = d$ or e.

Symmetric mean absolute percentage error

The Symmetric Mean Absolute Percentage Error (SMAPE) in (7.4) is commonly used (for example, see [220]) to avoid skewing the overall error by the data points wherein the duration and energy consumption take small values. The corresponding SMAPEs for duration and energy are represented by SMAPE(d) and SMAPE(e) with

$$\text{SMAPE}(x) := \sum_{j\in\mathcal{U}} \frac{1}{|\mathcal{U}|} \sum_{i\in\mathcal{A}_j} \frac{1}{|\mathcal{A}|_j} \left| \frac{x_{i,j} - \widehat{x}_{i,j}}{x_{i,j} + \widehat{x}_{i,j}} \right| \times 100\%. \tag{7.4}$$

Results and Discussion

Experimental setup

In Figure 7.1.2, we report MAE(d) and MAE(e) for I-GMM and P-GMM on Caltech dataset as a function of the look back period which defines the length of the training set. Users with larger than 20 sessions during Nov. 1, 2018 and Jan. 1, 2019 are included in $\mathcal{U}$ and tested. Note that the size of the training data may not be proportional to the length of periods since in general there is less claimed session data early in the dataset. The 30-day testing data is collected from Dec. 1, 2018 to Jan. 1, 2019. We study the behavior of prediction accuracy with different training data sizes by training the GMMs with data collected from five time intervals ending on Nov. 30, 2018 and starting on Sep. 1, 2018, Sep. 15, 2018, Oct. 1, 2018, Oct. 15, 2018

Caltech	I-GMM	P-GMM	Mean	User Input
SMAPE(d)%	15.8543	16.6313	20.4432	25.8093
SMAPE(e)%	14.4273	17.2927	15.9275	27.5523

JPL	I-GMM	P-GMM	Mean	User Input
SMAPE(d)%	12.2500	12.5079	15.8985	18.5994
SMAPE(e)%	12.7318	13.6863	13.3014	26.8769

Table 7.3.1: SMAPEs for Caltech and JPL datasets.

and Nov. 1, 2018, respectively. The GMM components are initialized using k-means clustering as implemented by the Scikit Learn GMM package [219]. Since it is not deterministic, we repeat this initialization 25 times and keep the model with the highest log-likelihood on the training dataset. Grid search and cross validation [219] are used to find the best number of components for each GMM.

Observations

As observed from Figure 7.1.2, for the JPL dataset with testing data obtained from Dec. 1, 2018 to Jan. 1, 2019, the 60-day training data gives the best overall performance. This coincides with our intuition that user behavior changes over time and there is a trade-off between data quality and size. The Caltech dataset also displays this trade-off; however, the best performance was found for only a 30-day training set. This is likely because there was a transition from free to paid charging on Nov. 1, which meant that data prior to that date had very different properties.

Hence, for the JPL dataset, we fix the training data as the one collected from Oct. 1, 2018 to Dec. 1, 2018 and show the scatterings of SMAPEs for each session in the testing data (from Dec. 1, 2018 to Jan. 1, 2019) in Figure 7.3.1. The SMAPEs are concentrated on small values with a few outliers and high-quality duration prediction has a positive correlation with high-quality energy prediction. As a comparison, user input SMAPEs, shown as Xs, are much worse.

Table 7.3.1 shows the average SMAPEs for the various methods tested. For Caltech and JPL, we display the results using the 30 and 60-day training data, respectively. For reference we also calculate the error of two additional ways to predict user parameters: 1) we use the mean of the training data X_j as our prediction for each user, 2) we treat the user input data directly as the prediction. Note that to account for stochasticity in the GMM training process, the results in Figure 7.1.2 and Table 7.3.1 are obtained via 50 Monte Carlo simulations.

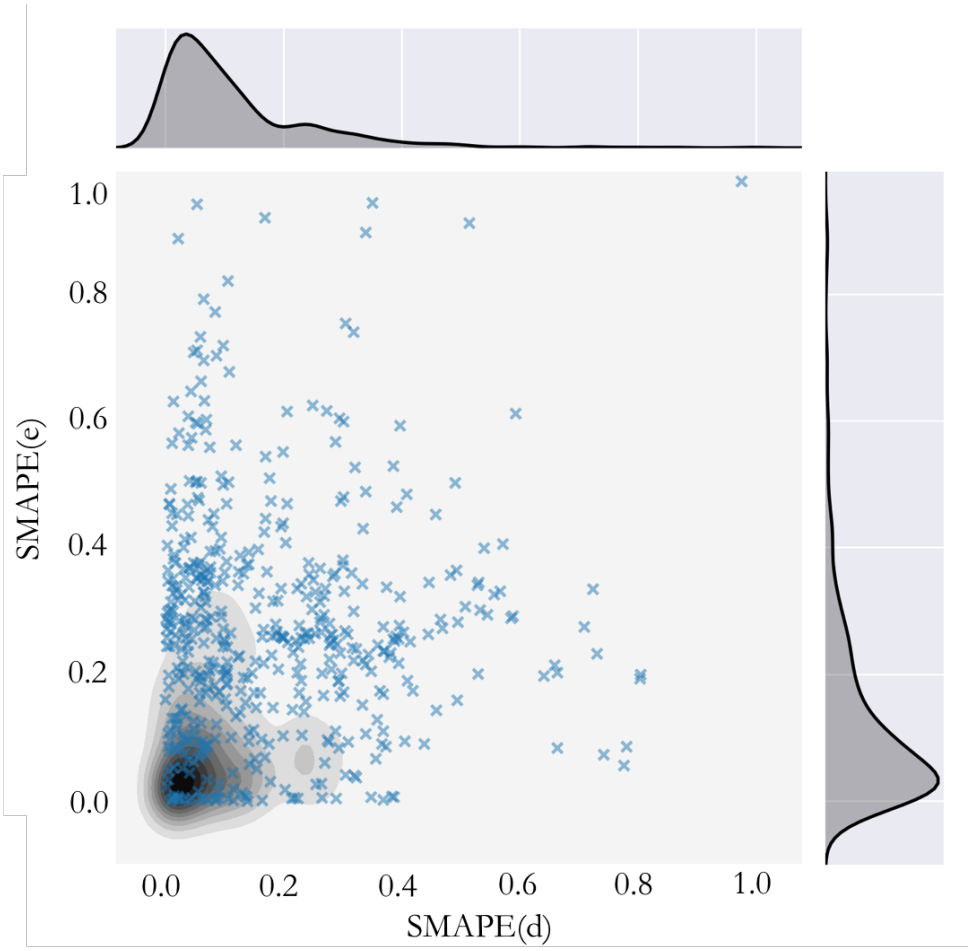


Figure 7.3.1: Correlation between SMAPE(d) and SMAPE(e) and their marginal distributions for the JPL dataset. Kernel density estimation is used to approximate the joint distribution of the SMAPEs for predicted duration and energy which is shown as grey shading. The blue crosses represent the corresponding user input SMAPEs (for I-GMM) with respect to each charging session in the testing data set X_{Test}.

Implications

EV users need incentives to provide more accurate predictions. As shown in Figures 7.1.2, Figure 7.3.1 and Table 7.3.1, user input data conspicuously gives the worst overall prediction. However, in some commercial EV charging companies, *e.g.,* PowerFlex, user input data is used as the direct input for the scheduling and pricing algorithms. Therefore, significant improvements can be made in the future by leveraging tools from statistics and machine learning to better predict user behaviors, *e.g.,* using GMMs. In addition, we find that when predicting user behavior there is

Table 7.4.1: List of key notation used in this section.

Clustering Parameters	
$\mathcal{N}$	Set of n charging sessions
$\mathcal{T}$	Set of T charging time slots
$\mathcal{S}$	Set of n charging curves
C_k	Cluster indexed by k ($k = 1, \ldots, K$)
Sequences	
$\mathbf{p}_i$	Pilot curve for session $i \in \mathcal{N}$
$\mathbf{s}_i$	Charging curve for session $i \in \mathcal{N}$
$\mathbf{x}_i$	Charging tail for session $i \in \mathcal{N}$
$\mathbf{c}_k$	Tail representative for cluster C_k

a trade-off between between training data quantity and quality caused by changing user behavior over time which must be considered.

7.4 ACN-Data Charging Curves Analysis

Charging curves

With the terminology introduced in Table 7.4.1, denote by $\mathcal{N} := \{1, \ldots, n\}$ the set of charging sessions. Each charging session refers to the charging duration from *connectionTime* to *disconnectTime* (see Table 7.4.1). Without loss of generality, we assume the times series of charging currents have the same length T and time granularity (If not, we preprocess the time series as explained in Section 6.2 and pad the shorter ones with zeros). Let $\mathcal{T} := \{1, \ldots, T\}$ be the set of time slots from *connectionTime* to *disconnectTime*. In the remaining contexts, we refer to "time series" as the raw data and "charging curves" the sequences with equally sampled points after preprocessing (introduced in Section 6.2), unless otherwise stated. We first define a charging curve and its associated pilot curve. For any session $i \in \mathcal{N}$, a *charging curve* $\mathbf{s}_i \in \mathbb{R}^T$ is the sequence of actual charging currents during the session i, *i.e.*, $\mathbf{s}_i := (s_i(1), \ldots, s_i(T))$. For any session $i \in \mathcal{N}$, a *pilot curve* $\mathbf{p}_i \in \mathbb{R}^T$ is the sequence of control signals during the session i, *i.e.*, $\mathbf{p}_i := (p_i(1), \ldots, p_i(T))$. At each time $t \in \mathcal{T}$, a charger sends a pilot signal $p_i(t)$ to the vehicle which then draws a current $s_i(t)$ that is no higher than $p_i(t)$ (both $s_i(t)$ and $p_i(t)$ are in units of Amp). Given a set of n charging curves $\mathcal{S} := \{\mathbf{s}_i \in \mathbb{R}^T : i \in \mathcal{N}\}$ and the associated pilot curves $\mathcal{P} := \{\mathbf{p}_i \in \mathbb{R}^T : i \in \mathcal{N}\}$, the key issue considered in this chapter is: how to classify the elements of $\mathcal{S}$ into different groups and implement the classification efficiently?

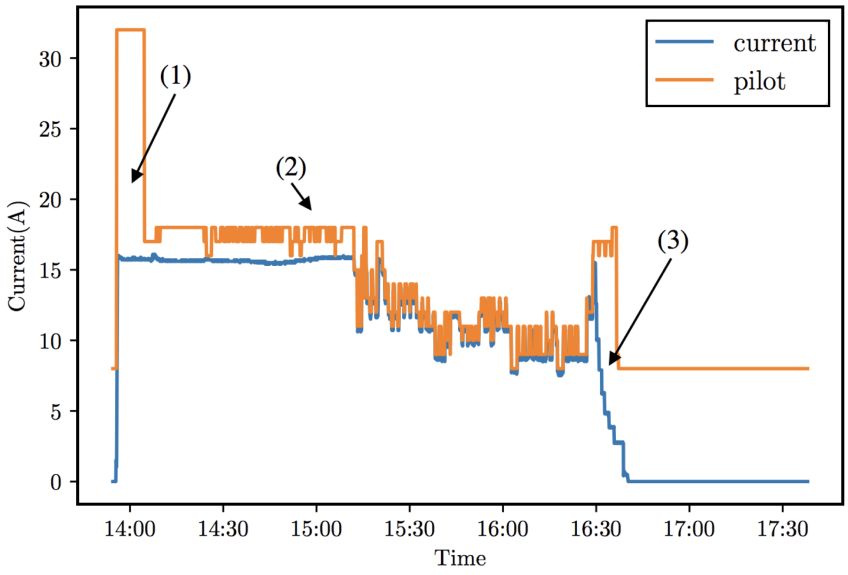


Figure 7.4.1: An example of a charging curve (in blue) and the corresponding pilot curve (in orange) for a charging session with *userID* 409 on Oct. 13, 2018.

Typically, a charging curve from a charging session consists of two stages – the *bulk charging stage* and the *absorption stage*. In the bulk stage which usually occurs before the state of charge (SoC) reaches 80% full, the charging current is usually equal approximately to the pilot signal and the charging voltage steadily increases. In the absorption stage, the voltage stays approximately at its peak level and the charging currents decreases as the battery reaches full charge. In cases when the available time for charging is sufficiently long, a charging session may contain an additional stage, namely the *idle stage* where the charging current is closed to zero (neglecting noise). An example of a charging curve and its associated pilot curve is shown in Fig. 7.4.1. It can be observed that the measured charging current does not follow the pilot signal exactly. The gap between the pilot signal and charging current fluctuates due to the following reasons: (1) the maximum charging current that the vehicle can draw being smaller than the control signal; (2) random noise; (3) entry into the absorption or idle stage.

Charging tails

The charging current in the bulk charging stage is controlled by the scheduling algorithm and therefore it exhibits little information of the battery. For classification purposes, we are mainly interested in the second stage, during which the charging current might exhibit distinct patterns because of different types of batteries. Let t_s^i and t_e^i denote the start time and end time of the absorption stage for session $i \in \mathcal{N}$.

We refer to the subsequence of the charging curve in this stage as charging tail, defined as follows.

Definition 7.4.1 (Charging Tail)**.** For session $i \in \mathcal{N}$, a *charging tail* $\mathbf{x}_i := \left(s_i(t), t = t^i_{\mathrm{s}}, \ldots, t^i_{\mathrm{e}}\right)$ is the subsequence of the charging curve $\mathbf{s}_i$ in the absorption stage $\left\{t^i_{\mathrm{s}}, \ldots, t^i_{\mathrm{e}}\right\}$.

Since charging tails display distinctive characteristics of their corresponding charging curves, we will classify charging curves based on their tails. A common battery model assumes that a charging curve starts and stays at some maximum charging current $C^i_{\max}$ until the battery enters the absorption stage when the charging current steadily decreases to zero. In this model, the start time t^i_s of the charging tail is easily identifiable to be the last time the charging current stays at the maximum rate $C^i_{\max}$ and the end time t^i_e is the first time the charging current drops to zero, *i.e.*, an (ideal) charging tail $\mathbf{x}_i$ is a decreasing sequence defined by: $C^i_{\max} = s(t^i_s) > s(t^i_s + 1) \geq \cdots \geq s(t^i_e - 1) > s(t^i_e) = 0$. In practice, however, extracting the charging tail $\mathbf{x}_i$ from a real charging curve $\mathbf{s}_i$, *i.e.*, identifying the start time t^i_s and end time t^i_e of the absorption stage, can be difficult. A charging curve $\mathbf{s}_i$ is rarely a decreasing sequence as the simple model above assumes. The charging current fluctuates for multiple reasons, not only the *internal* charging state of a battery, but also *external* factors such as pilot signal control (scheduling) or noise. In Fig. 7.4.2, we display examples of charging curves where the rates drop due to these reasons.

The confusion caused by scheduling can be cleared up using the first tail extraction method in Section 7.5. The confusion caused by noise is trickier to deal with since, in particular, the noise can be large and fluctuate frequently as shown in Fig. 7.4.1 and Fig. 7.4.2. Thus, it is nontrivial to differentiate the changes due to noise from the other scenarios. In addition, it is possible that more than one scenarios occur simultaneously, *e.g.,*, scheduling within the tail stage. In this case, the charging tails may not be decreasing sequences. Therefore, determining the exact starting point (and ending point) of the absorption stage is difficult. Moreover, for a given length-T charging curve $\mathbf{s}_i \in \mathbb{R}^T$, different tail extraction methods (as introduced in Section 7.5) may give distinct tails. Therefore, we consider the set of all candidates of charging tails for session $i \in \mathcal{N}$, denoted by $\mathcal{X}_i$. As subsequences of $\mathbf{s}_i$, the tails in $\mathcal{X}_i$ may not have the same dimension. This motivates a novel *selective clustering* problem with a *new* objective: *How to cluster n candidates (of charging tails)* $\{\mathbf{x}_i \in \mathcal{X}_i : i \in \mathcal{N}\}$ *with the ability of choosing a candidate* $\mathbf{x}_i \in \mathcal{X}_i$ *for each charging curve* $\mathbf{s}_i$*?* In the sequel, we formalize our clustering problem.

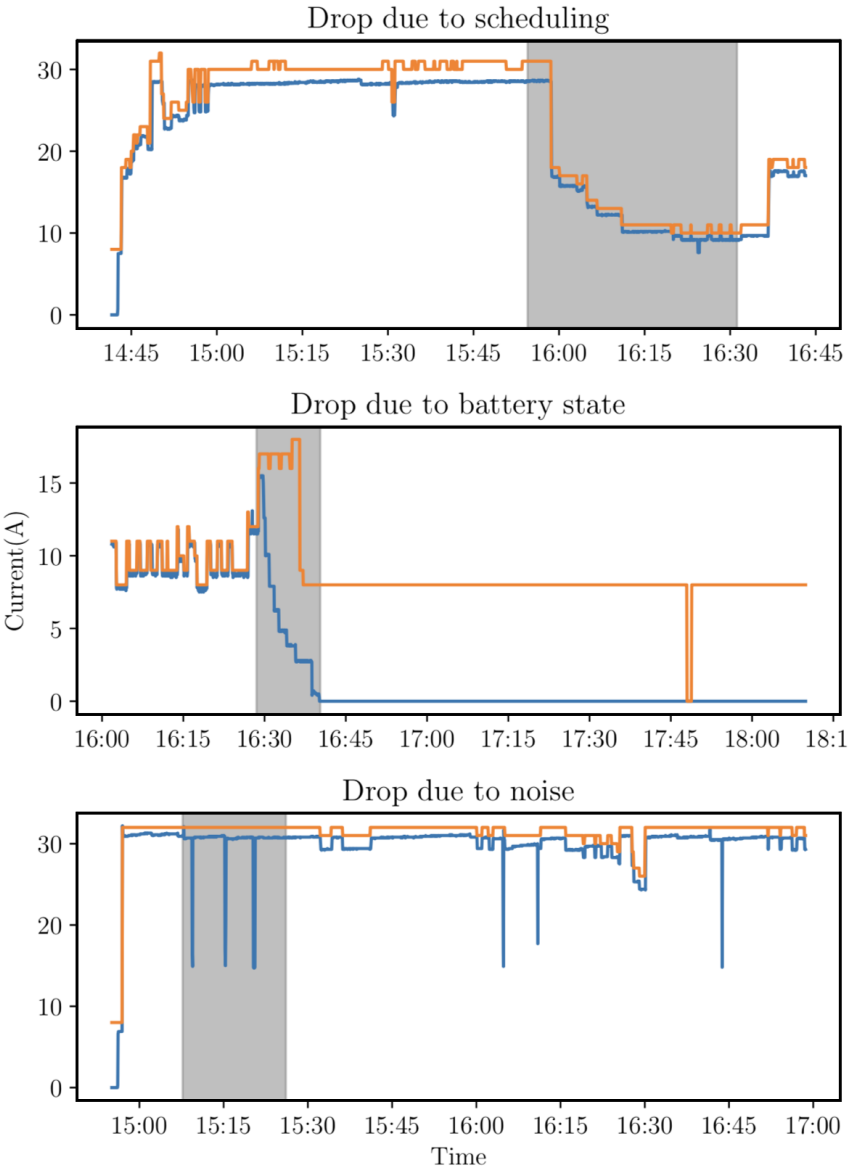


Figure 7.4.2: Examples of charging curves where charging currents drop due to (1) scheduling, (2) battery charging state, and (3) noise, as indicated by the shaded regions. Each plot only shows a selected portion of a session. The time series are for sessions with *userID* 576 (top), 409 (mid) and 526 (bot), obtained on Nov. 07, 2018, Oct. 09, 2018 and Oct. 22, 2018, respectively.

Selective clustering

With the above definitions, the charging tail classification problem can be defined as the following optimization:

$$\min_{\mathcal{X}} \min_{C} \sum_{k=1}^{K} \sum_{i \in C_k} d(\mathbf{c}_k, \mathbf{x}_i) \tag{7.5}$$

where $\mathcal{X} := \{\mathbf{x}_i \in \mathcal{X}_i : i \in \mathcal{N}\}$ is a set of n *candidates*, constructed by selecting exactly one tail from $\mathcal{X}_1, \ldots, \mathcal{X}_n$. We assume the number of clusters K is known and searching the best K is beyond the scope of this chapter. Let $\mathcal{K} := \{1, \ldots, K\}$. The set $C := \{C_k : k \in \mathcal{K}\}$ specifies a partition $\mathcal{N} = \bigcup_{k=1}^{K} C_k$ of the charging sessions $\mathcal{N}$ with each C_k representing a distinctive cluster. Moreover, $\mathbf{c}_k$ is a *tail representative* for the k-th cluster, defined as its *medoid*. The *distance function(s)* is denoted by $d(\cdot, \cdot)$, which will be specified in Section 7.5.

To solve the minimization in (7.5), we use the idea of alternating minimization (AM) and refine the representative of each cluster by iteratively implementing the following until convergence. With suitable initialization, the iterations (the $(\ell+1)$-step) consist of two main steps.

- *Tail Extraction (TE)*: Given n fixed tail representatives, we find new candidates that minimize the following:

$$\mathbf{x}_i^{(\ell+1)} := \arg\min_{\mathbf{x} \in \mathcal{X}_i} \min_{k \in \mathcal{K}} d\left(\mathbf{c}_k^{(\ell)}, \mathbf{x}\right), \ i \in \mathcal{N}. \tag{7.6}$$

- *Tail Clustering (TC)*: We cluster the new tails obtained via TE and find new representatives $\mathbf{c}_1^{(\ell+1)}, \ldots, \mathbf{c}_K^{(\ell+1)}$ by solving the following minimization:

$$\min_{C} \sum_{k \in \mathcal{K}} \sum_{i \in C_k} d\left(\mathbf{c}_k^{(\ell+1)}, \mathbf{x}_i^{(\ell+1)}\right). \tag{7.7}$$

In Algorithm 10, we summarize the iterative process. The details of the initialization step is described in Section 7.5. Note that conducting TC and TE repeatedly cannot increase the objective function in (7.5). Therefore, the AM we established is guaranteed to have local convergence.

Theorem 7.4.1. *With arbitrary initialization $\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_n^{(1)}$, by iteratively performing (7.6) and (7.7), Algorithm 10 converges to a local optimum consisting of representative tails $\widehat{\mathbf{c}}_1, \ldots, \widehat{\mathbf{c}}_K$.*

Algorithm 10: AM for Selective Clustering

Input: Charging curves $\mathcal{S}$ and pilot curves $\mathcal{P}$;
Output: Clustering C and representatives $\widehat{\mathbf{c}}_1, \ldots, \widehat{\mathbf{c}}_K$;
$\ell \longleftarrow 1$;
Initialization $\longrightarrow \mathbf{x}_1^{(\ell)}, \ldots, \mathbf{x}_n^{(\ell)}$;
while *not converge* **do**
  Tail clustering (TC) $\longrightarrow \mathbf{c}_1^{(\ell)}, \ldots, \mathbf{c}_n^{(\ell)}$;
  Tail extraction (TE) $\longrightarrow \mathbf{x}_1^{(\ell+1)}, \ldots, \mathbf{x}_n^{(\ell+1)}$;
  $\ell \longleftarrow \ell + 1$
end

Proof. First, TE cannot decrease the objective function in (7.5). For any session $i \in \mathcal{N}$ and pair of $\mathbf{x}_i^{(\ell)} \in C_{k(i)}$ and the corresponding tail representative $\mathbf{c}_{k(i)}^{(\ell)}$, the minimization in (7.6) guarantees that there exists a tail representative $\mathbf{c}_{k'}$ such that

$$d\left(\mathbf{c}_{k'}, \mathbf{x}_i^{(\ell+1)}\right) \leq d\left(\mathbf{c}_{k(i)}^{(\ell)}, \mathbf{x}_i^{(\ell)}\right).$$

Therefore, this specifies a clustering with the objective function less than or equal to the previous clustering. Similarly, TC cannot decrease the objective function, since we just show that there exists a better clustering for the new tails, and the minimization in (7.7) can only result in an objective value that is equal to or smaller than the original one. □

The computational complexity for solving (7.6) in TE is $O(nK\gamma(d))$ where $\gamma(d)$ is the complexity for computing the distance function with fixed input sequences. Our experiments use an approximation in (7.8) for a more efficient implementation. In practice, for efficiently implementing TC, heuristics are used for finding a local optimum of (7.7). Moreover, as the AM procedure also leads to a local minimum, an initialization that is close to a global minimum is important. In the next section, we introduce tail extraction methods for initialization and heuristic algorithms for clustering.

7.5 Classification Method

In this section, we present our framework for charging curve clustering. It consists of three main stages depicted in Fig. 7.4.3.

Preprocessing

In general, the charging curve $\mathbf{s}_i$ and the pilot curve $\mathbf{p}_i$ for session $i \in \mathcal{N}$ are neither sampled at a fixed rate nor perfectly aligned. Most analysis techniques, however,

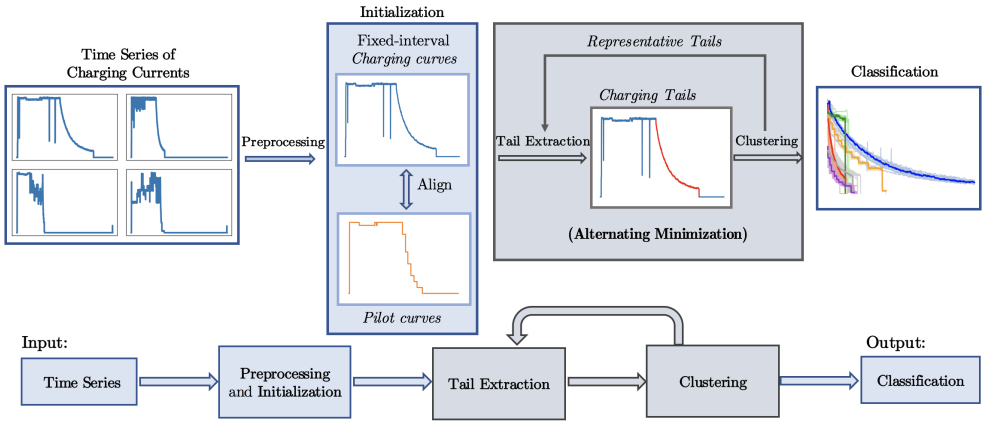


Figure 7.4.3: The classification method introduced in this chapter.

require that the time series be unevenly spaced. We therefore re-sample the time series as the mean over a fixed interval δ (if there is at least one sample) and fill in the missing points by linear interpolation. This preprocessing step ensures the alignment of signals in the time domain $\mathcal{T} = \{1, \ldots, T\}$ so that the distance metric $d(\mathbf{c}, \mathbf{x})$ is well defined.

Tail extraction

Not all charging curves contain tails, for two reasons. First, certain batteries do not exhibit a smooth absorption stage and the current just drops from C to 0 directly. Second, EVs may be unplugged before they are fully charged. This can happen when the EV leaves earlier than the input departure time, or when the requested energy is lower than the battery's remaining capacity. We consider three rules of thumb for tail extraction.

Extraction by pilot signals

As mentioned in Section 7.4, a tail is typically a decreasing sequence. Therefore, we declare that a battery has entered the absorption stage if the charging current $s(t)$ falls below a certain value $C > 0$. The end of the stage is the time when the charging current first reaches approximately zero. This simple rule of thumb is straightforward to implement. A drawback however is that it is hard to determine a suitable threshold $C > 0$. Scheduling, system congestion, or noise may cause the charging current to drop below the threshold C even before entering the absorption stage. To mitigate the confusion due to scheduling, we utilize the pilot curves and call a subsequence $\mathbf{s}'$ of a charging curve $\mathbf{s}$ *piloted at time* t if $p(t) - p(t-1) \geq s(t) - s(t-1)$. We accept

a tail if it is not piloted everywhere and $s(t) \leq C$ for a given *threshold parameter* $C > 0$.

Extraction by duration

The end of the absorption stage can be found by locating the first (approximately) zero value of the charging currents. If we have an estimate of the duration of the adsorption stage, we will be able to extract the tail. This approach requires the knowledge of the tail duration. Moreover, even for the same battery, the duration of the adsorption stage varies across different sessions because of noise and our re-sampling.

The first two extraction methods can be combined to extract tails. In our experiments reported in Section 7.6, for each distinct user, we regard the first two methods as a two-layer filter and extract a tail representative for each session if the tail passes the filtering criteria. In particular, for session i, we employ grid search for the selection of threshold parameter $C > 0$ by decreasing it from the maximal charging current $C^i_{\max}$.

Extraction by matching

Our third method assumes that all charging tails from the same EV have similar properties such as duration and shape. Before the iterative steps, suppose that for a fixed user, we are able to obtain an initial charging tail $\mathbf{x}^{(1)}$, *e.g.*, using the two methods above. This $\mathbf{x}^{(1)}$ is used as a "template" to extract the tails of all other charging curves of the same user. Then, we go through the subsequences of the charging curve that have the same length as the template, and find a charging tail with improved noise robustness. Suppose we obtain a tail representative $\mathbf{x}$ for a fixed user. For the remaining sessions i of the same user, we minimize the Euclidean distance $d_{\mathrm{ED}}\big(\mathbf{x}, \mathbf{x}_i^{(1)}\big)$ over all consecutive subsequences $\mathbf{x}_i^{(1)}$ of the charging curve $\mathbf{s}_i$ that have the same length as $\mathbf{x}$. In this way, we use the three extraction rules jointly to compute the initial tails $\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_n^{(1)}$ in Algorithm 10. Fig. 7.5.1 illustrates the idea and effectiveness of this approach.

Besides speeding up the initialization, the third approach is also used as the TE step as an approximation of the optimization in (7.6). At the ℓ-th iteration, by setting the medoid (tail representative) $\mathbf{c}_k^{(\ell)}$ of the k-th cluster that the charging curve $\mathbf{x}_i$

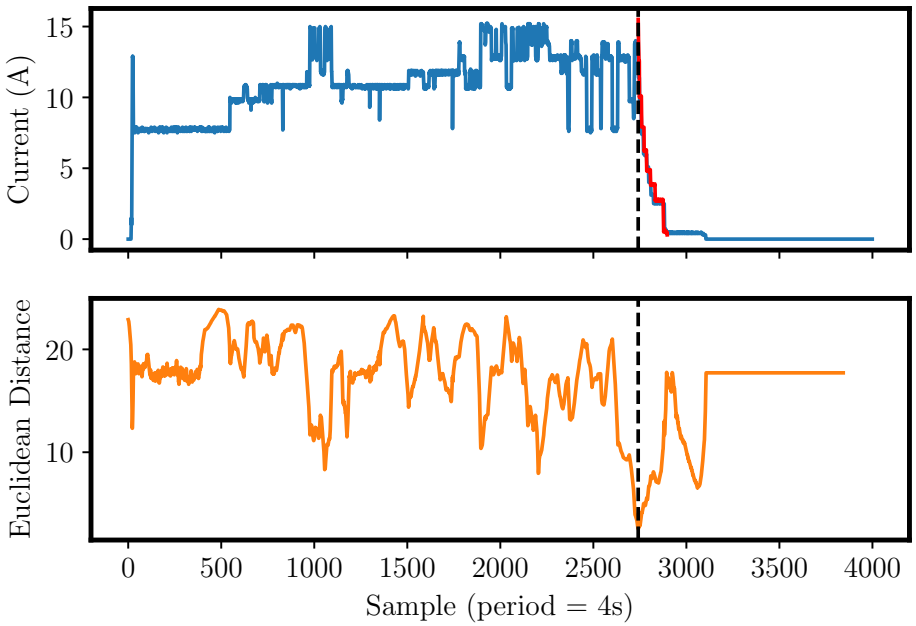


Figure 7.5.1: An example of *extraction by matching*. The red subsequence $\mathbf{x}_1$ is a template with *userID* 409, which is extracted from the first session $\mathbf{s}_1$ of this user. The figure below visualizes the change of Euclidean distance of the second session $\mathbf{s}_2$ with respect to $\mathbf{x}_1$. The black vertical line indicates the best matching location in $\mathbf{s}_2$ for $\mathbf{x}_1$ and the tail $\mathbf{x}_2$ can be found correspondingly despite the slight difference of both tails.

is classified into as the template[2] and using the Euclidean distance as the distance function, we approximate the optimization in (7.6) for the ℓ-th iteration:

$$\widehat{\mathbf{x}}_i^{(\ell+1)} = \arg\min_{\mathbf{x}} d_{\mathrm{ED}}\left(\mathbf{c}_k^{(\ell)}, \mathbf{x}\right) \tag{7.8}$$

where the minimization is over all $\mathbf{x} \in \mathcal{X}_i\left(\mathbf{c}_k^{\ell}\right)$ and $\mathcal{X}_i\left(\mathbf{c}_k^{\ell}\right)$ is the set containing all consecutive subsequences of the charging curve $\mathbf{s}_i$ that have the same length as $\mathbf{c}_k^{(\ell)}$.

Tail clustering

Time series clustering is a well-studied problem; see [222] for a review and [223] for a detailed experimental comparison. One of the main problems considered in the literature is determining the distance/similarity between time series. Based on

[2]In our experiments (elaborated in Section 7.6), for improving efficiency, we implement a simplified TE, wherein we focus on the medoid of the cluster that the charging curve $\mathbf{s}_i$ for session i belongs to and remove the minimization over k in (7.6). This modification does not affect the local convergence property stated in Theorem 7.4.1.

their own applications, a variety of similarity distance metrics have been proposed, including the Euclidean distance[224] for stock price movements clustering, the edit distance [225] for trajectory clustering and the cross correlation [226] for electrocardiogram time series clustering, *etc*. However, most of the existing metrics require that the two sequences have the same length. As an exception, dynamic time warping (DTW)[227] is able to calculate the distance between two sequences with different lengths. However, it is computationally expensive. For clustering tails, we introduce a penalty term to the Euclidean distance and our experiments show that the new distance function (defined as the MED in (7.9)) surpasses the others for charging time series clustering. We use similarity based clustering techniques for solving the minimization in (7.7). Tails of varying lengths are clustered in two steps: (a) similarity matrix construction (b) similarity based clustering.

Similarity matrix construction

The lengths of tails extracted from the charging curves of different EVs are generally different. This creates difficulty in comparing two tails as the standard Euclidean distance is defined for two vectors of the same length. We compare three different distance definitions for tails of different lengths and more results can be found in Section 7.6. The first method simply pads the shorter tail with zeros to make two tails the same length so their distance is the standard Euclidean distance (ED). The second method uses a distance function defined as follows. Suppose $\mathbf{x} \in \mathbb{R}^s$ and $\mathbf{y} \in \mathbb{R}^l$ with $s \leq l$. Their corresponding *modified Euclidean distance* (MED) is

$$d_{\text{MED}}(\mathbf{x}, \mathbf{y}) := \min\left\{d_{\text{ED}}(\mathbf{x}, \mathbf{y}(\leq s)), d_{\text{ED}}(\mathbf{x}, \mathbf{y}(\geq l - s + 1))\right\} + \lambda |l - s| \quad (7.9)$$

where $d_{\text{ED}}(\cdot, \cdot)$ is the Euclidean distance and $\mathbf{y}(\leq s)$ and $\mathbf{y}(\geq l - s + 1)$ represent the first s and last s coordinates of $\mathbf{y}$, respectively. The *penalty parameter* $\lambda > 0$ can be tuned. By default we set it to 1. Note that the distance function MED in (7.9) may not satisfy the triangle inequality. The third method uses the dynamic time warping (DTW) defined in [227, 228]. The clustering results obtained via the ED with zero padding technique, the MED defined above and the DTW are compared in Fig. 7.6.1, with more details in Section 7.6.

Similarity based clustering

For similarity based clustering, we apply the spectral clustering [229–231] as the heuristic for approximating the minimization in (7.7).

7.6 Clustering, Applications, and Discussions

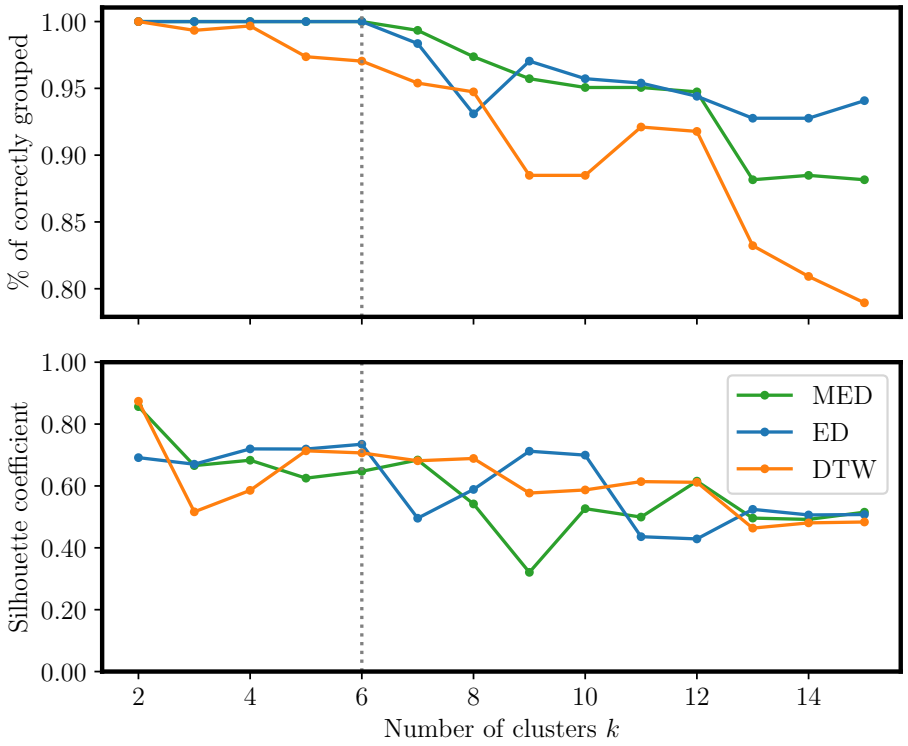


Figure 7.6.1: The performance for different number of clusters using three different distance functions – Euclidean distance (ED), Modified Euclidean distance (MED), Dynamic Time warping distance (DTW).

Clustering evaluation

In this section, we evaluate the proposed method (shown in Fig. 7.4.3) on ACN-Data [1]. We use the dataset from JPL from Sep. 2018 to Dec. 2018 as the training

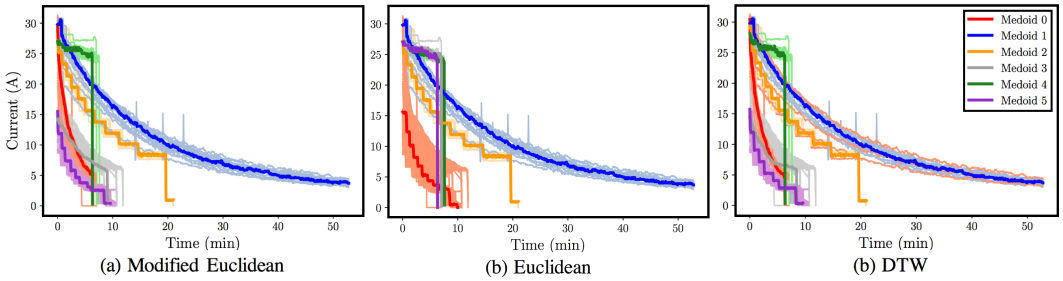


Figure 7.6.2: Visualization of $K = 6$ clusters for MED, ED and DTW. Tails are within the same cluster if they have the same color and the tail representatives (medoids) are emphasized.

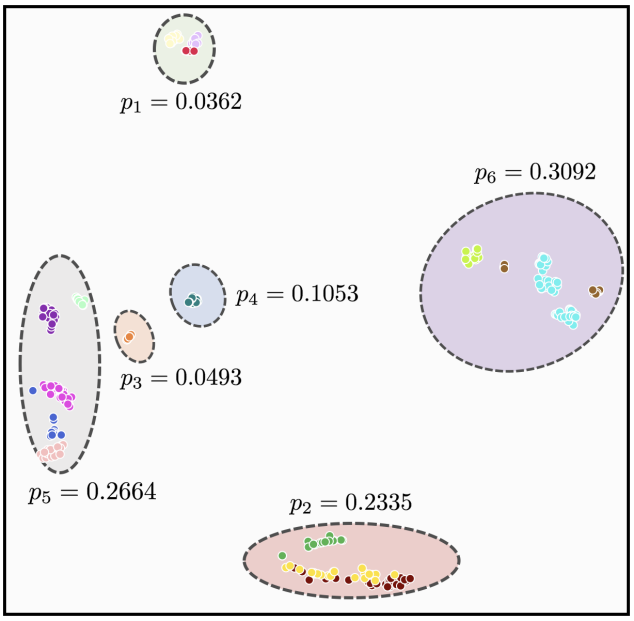


Figure 7.6.3: Two-dimensional visualization of our clustering results with $K = 6$ clusters. Tails for different users are colored differently. The clusters' colors are consistent with those used in Fig. 7.6.2. The marginal probabilities $p_1, \ldots, p_6$ represent the portions of charging sessions falling into the six clusters.

data, which contains 2933 claimed sessions from 195 users.[3] In preprocessing (see Section 7.5), we resample the data at a time resolution of $\delta = 4$ seconds.

We use two evaluation metrics to find the number of clusters K. The first is the *silhouette coefficient* [232], which takes a value in $[-1, 1]$. A higher silhouette coefficient indicates better clustering performance. The second is the *correctly classified percentage*. Recall that each tail is associated with a *userID* (see Table 7.4.1). We evaluate the clustering quality by checking if the tails with the same *userID* are consistently grouped into the same cluster.[4] A tail is considered *correctly classified* if it is clustered into a group wherein the majority of the tails have the same *userID* as the considered tail.

The evaluation results for three different distance functions – the modified Euclidean distance (MED), Euclidean distance (ED), and dynamic time wrapping (DTW) are shown in Fig. 7.6.1. We use $\lambda = 1$ for MED. For distance to similarity conversion,

[3]More than a half of the users have less than 12 charging sessions during the period. In the clustering experiment, we only consider the 35 users with more than 30 sessions. Out of the 35 users, 16 of them have sufficient number of charging curves with tail-like features. Our experiments used the 304 charging curves from these 16 users.

[4]It is possible that the same *userID* exhibits different charging patterns. This may occur if the user changes her EV or owns more than one EV. But as shown in Table 7.6.2, such scenario is rare.

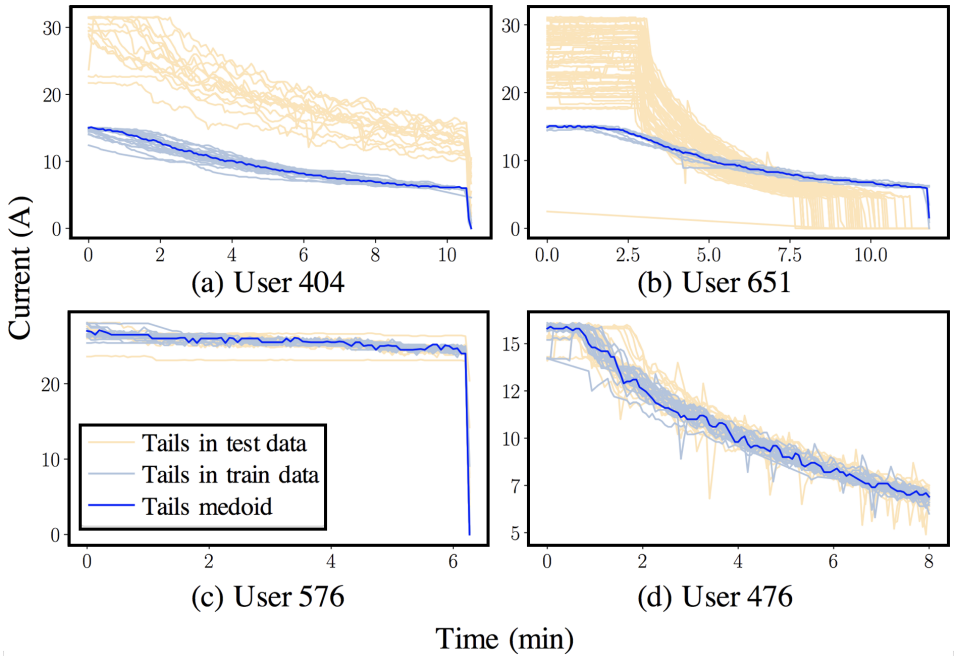


Figure 7.6.4: Examples of the training and testing data (tails) for four users. Sub-figures (a) and (b) are the tails of the two users with poor prediction performance (highlighted in blue in Table 7.6.1). The poor prediction performance is due to the fact that the tails in the training data are very different from those in the testing data. Sub-figures (c) and (d) are examples where the tail representatives achieve high-quality prediction performance. Tails in the training data and those in the testing data are similar.

we use the Gaussian kernel $\kappa(d) = \exp\left(-d^2/\theta\right) \in (0, 1]$ where d is the pairwise distance between two sequences and $\theta > 0$ is a tuning parameter. In particular, we choose $\theta = 20$ for MED and ED and $\theta = 110$ for DTW. It can be seen that $K = 6$ is a good choice of number of clusters for this dataset, as it is the largest value at which all the tails are correctly classified for both ED and MED. In addition, the silhouette coefficient is relatively high for $K = 6$. Fixing $K = 6$, the clustering results for different distance functions are visualized in Fig. 7.6.2. It can be seen that using MED, the six clusters are well-separated and the corresponding medoids provide informative patterns for charging tails. In Fig. 7.6.3, we project the tails to a two-dimensional space using the t-distributed stochastic neighbor embedding (t-SNE) and MED. It demonstrates the hierarchical relationship between the groups of users and the clusters for our training data.

Charging behavior prediction

The ability to classify charging behavior can enable both offline and online applications in the future (see Section 8.2). One of the building blocks for these applications will be the use of cluster representatives for prediction. In this subsection, we illustrate its accuracy.

The training data is the same as in Section 7.6 and the testing data contains 731 tails for 1441 sessions collected from Jan. 2019 to Aug. 2019. We use the tail representatives of the training data obtained using our framework in Fig. 7.4.3 to predict the behavior of the charging tails of the testing data. Denote by $\mathbf{s}$ a real charging curve in the testing data and $\widehat{\mathbf{x}}$ the estimated tail. We consider two situations–with and without the knowledge of *userID*, and the results are shown in Table 7.6.1 and Table 7.6.2, respectively. We evaluate the prediction quality using the following three metrics. The first metric is the *coefficient of determination* (R^2) (generalized in our case for comparing two sequences of different lengths) defined as:

$$R^2_{\text{Predict}}(\mathbf{s}, \widehat{\mathbf{x}}) := \min_{\mathbf{x}} \left\{ 1 - \frac{\sum_{t=1}^{m} (x_t - \widehat{x}_t)^2}{\sum_{t=1}^{m} (x_t - \bar{x})^2} \right\} \tag{7.10}$$

where the minimization is over all consecutive subsequences $\mathbf{x}$ of the charging curve $\mathbf{s}$ that have the same length as $\widehat{\mathbf{x}}$ and $\bar{x} = \sum_{t=1}^{m} x_t/n$ and m is the length of $\mathbf{x}$ and $\widehat{\mathbf{x}}$. It ranges from $(-\infty, 1]$ and the larger the better. A negative value indicates that performance is worse than the arithmetic mean mean. Our second metric is the *root mean square error (RMSE)* that is useful for measuring scale-dependent prediction error. The last metric is the *mean absolute error (MAE)*. Similar to (7.10), the last two metrics are also generalized with an additional minimization over consecutive subsequences of charging curves in the testing data.

Table 7.6.1 shows the *userID*-based prediction results. Each tail representative (medoid) corresponds to each group of users. As can be observed from the results, except for user 404 and user 651, the tail representatives of the other 14 users can well predict the charging tail behavior in incoming sessions for the same user. Fig. 7.6.4 visualizes the training tails, testing tails and tail representatives of 4 users, including the two users with high prediction error. Note that the charging tails of user 404 exhibit two distinct groups, one is from Sep. 2018 to Dec. 2018 (tails colored in light blue) and the other is from Jan. 2019 onward (tails colored in light orange); similarly for user 476. Unlike for the other users, the tails in the training data are very different from those in the testing data for user 404 and 651. Ignoring the user labels, Table 7.6.2 compares the prediction RMSE using the most similar cluster

Table 7.6.1: Prediction results with user tail representative.

userID	R^2	RMSE	MAE
322	0.6464± 1.2509	0.9306 ± 1.1596	0.7947 ± 1.1773
334	0.8783± 0.4765	1.3840 ± 1.3656	0.9954 ± 1.2012
368	0.6812 ± 0.6259	2.4878 ± 1.9991	2.0742 ± 1.7777
371	0.8615 ± 0.2040	0.8735 ± 0.6297	0.6946 ± 0.5655
374	0.9329 ± 0.0571	1.3872 ± 0.5800	0.9128 ± 0.4418
404	-11.906 ± 4.9304	10.522 ± 2.1957	10.230 ± 2.1357
405	0.8335 ± 0.1253	1.2936 ± 0.4793	0.9011 ± 0.3733
406	0.8917 ± 0.1315	0.5243 ± 0.2889	0.4082 ± 0.2305
409	0.9078 ± 0.0464	0.9412 ± 0.2310	0.6423 ± 0.1853
476	0.9509 ± 0.0757	0.5369 ± 0.3062	0.4077 ± 0.2536
551	0.9199 ± 0.1256	1.4938 ± 1.0355	1.1755 ± 0.7750
576	0.9209 ± 0.0249	0.6258 ± 0.1136	0.4802 ± 0.1178
577	0.9150 ± 0.0749	1.0301 ± 0.4422	0.6683 ± 0.2736
592	0.8699 ± 0.2607	0.7686 ± 0.6002	0.5394 ± 0.4607
607	0.9506 ± 0.0936	1.0141 ± 0.7940	0.8195 ± 0.6497
651	-3.5447 ± 1.9932	6.7702 ± 1.5415	5.4010 ± 1.0258

Table 7.6.2: Prediction RMSE with cluster representative.

UserID	MED	ED	DTW
322	2.7170 ± 0.8737	**0.9306 ± 1.1596**	2.7363 ± 0.9032
334	**1.1880 ±1.4109**	4.3331 ± 1.0066	2.8291 ± 1.7102
368	**2.6745 ± 2.0675**	2.7885 ± 1.2932	2.8450 ± 1.5884
371	**0.8266 ± 0.5495**	3.1160 ± 0.7986	**0.8266 ± 0.5495**
374	**1.3872 ± 0.5800**	**1.3872 ± 0.5800**	3.8939 ± 2.4183
404	**9.4865 ± 2.0709**	13.4698 ± 2.1212	**9.4865 ± 2.0709**
405	**1.2805 ± 0.5074**	1.4343 ± 0.5525	1.3289 ± 0.4918
406	**1.4506 ± 0.4911**	3.0573 ± 0.1223	**1.4506 ± 0.4911**
409	1.0244 ± 0.1878	1.6821 ± 0.5381	**0.9960 ± 0.1940**
476	**1.5438 ± 0.3304**	4.2103 ± 0.3307	**1.5438 ± 0.3304**
551	**1.5861 ± 1.0745**	**1.5861 ± 1.0745**	4.8002 ± 3.3426
576	2.5593 ± 0.0552	**0.7033 ± 0.1582**	2.6073 ± 0.0357
577	0.8972 ± 0.2218	1.4100 ± 0.5203	**0.8832 ± 0.2140**
592	**0.7682 ± 0.5871**	0.7686 ± 0.6002	0.8690 ± 0.5895
607	**1.0393 ± 0.6984**	4.2828 ± 0.9654	2.7691 ± 1.4346
651	**4.7786 ± 0.5789**	5.4719 ± 1.8010	**4.7786 ± 0.5789**

representative from the 6 clusters obtained for three different distance functions – MED, ED and DTW. In this case, the estimate is the tail representative of the cluster to which the charging curve in the testing data belongs. The best distance function for each user is highlighted in bold. MED is the best for most of the cases. In addition, Tables 7.6.1 and 7.6.2 show that the cluster representatives with MED achieves comparable and even better prediction than user representatives, indicating the existence of charging tail patterns.

Charging stage decision

In the remainder of our experimental results, we consider a real-time binary decision problem on whether an EV is in the absorption stage (AS) (see Section 7.4 for more details of the AS) or not. Our training data remains the same. In particular, for the testing data, we choose the user with ID 476 as an example, and manually label the start time t_{s} and end time t_{e} of the AS for each of the charging sessions since Jan. 2019. There are $n = 38$ out of 46 sessions in total that contain tails. The MAE is used for deciding the charging stage. Let $\varepsilon_{\mathrm{MAE}}$ be the *error threshold*. At time $t \in \mathcal{T}$, denote by m the number of samples that can be used in our decision. Equivalently, m is the time delay that are allowed for deciding if at time t the EV enters the AS. The decision rule in our experiments is that if $d_{\mathrm{ED}}\left(\mathbf{s}\left(t : t+m\right), \mathbf{c}\left(\leq m\right)\right) \leq \varepsilon_{\mathrm{MAE}}$, then we claim that the EV is in the AS; otherwise the EV is not in the AS where $\mathbf{s}\left(t : t+m\right) := s(t), \ldots, s(t+m)$ and $\mathbf{c}\left(\leq m\right) := c(1), \ldots, c(m)$. We set $\varepsilon_{\mathrm{MAE}} = 0.7$ in the tests.

Fig. 7.6.5 shows the trade-offs between the decision accuracy and the number of samples. In particular, in Fig. 7.6.5, the *average accuracy* is defined as

$$\sum_{i=1}^{n} \frac{\mathrm{TP}_m(\mathbf{s}_i) + \mathrm{TN}_m(\mathbf{s}_i)}{\mathrm{TP}_m(\mathbf{s}_i) + \mathrm{TN}_m(\mathbf{s}_i) + \mathrm{FN}_m(\mathbf{s}_i) + \mathrm{FP}_m(\mathbf{s}_i)}$$

where $\mathrm{TP}_m(\mathbf{s}_i)$, $\mathrm{FP}_m(\mathbf{s}_i)$, $\mathrm{TN}_m(\mathbf{s}_i)$ and $\mathrm{FN}_m(\mathbf{s}_i)$ are the numbers of true positive, false positive, true negative and false negative decisions for the charging stage decision of a charging curve $\mathbf{s}_i$ with m samples. The *average sensitivity* and *average precision* are defined similarly as

$$\sum_{i=1}^{n} \frac{\mathrm{TP}_m(\mathbf{s}_i)}{\mathrm{TP}_m(\mathbf{s}_i) + \mathrm{FN}_m(\mathbf{s}_i)} \text{ and } \sum_{i=1}^{n} \frac{\mathrm{TP}_m(\mathbf{s}_i)}{\mathrm{TP}_m(\mathbf{s}_i) + \mathrm{FP}_m(\mathbf{s}_i)},$$

respectively. Both the average precision and the average sensitivity grow with the number of samples m.

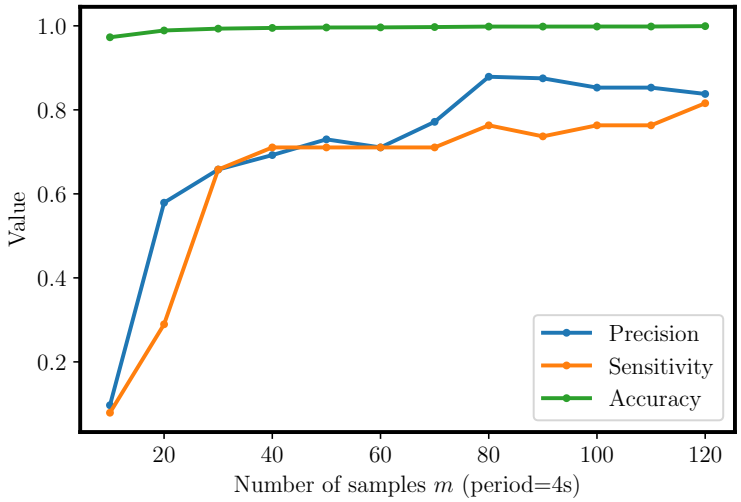


Figure 7.6.5: Trade-offs between the number of samples m and the accuracy, sensitivity and precision.

Part IV

Impact and Future Directions

Chapter 8

CONCLUSIONS

Achieving the net zero carbon goal requires a combination of advanced AI/ML techniques with classical methods in power grids. Utilizing the data generated in smart grids in a robust and resilient way is still a challenging task. In the previous chapters, we have presented several control and decision-making models that help unify the use of data subject to untrusted forecasts and predictions, black-box AI tools and distribution shifts.

The goal of this dissertation has been to study learning-augmented control and decision-making problems and their applications in smart grids.

8.1 Summary of Learning-Augmented Control Models

As a summary, the models for the learning-augmented online algorithms presented in Chapter 2, 3, 4 and 5 are shown below. The generality of the control models increases from the top row (Chapter 2) to the bottom (Chapter 4 and 5).

Dynamics	**Cost Functions**	*Imperfect Predictions/Advice*
$x_{t+1} = Ax_t + Bu_t + w_t$	$\sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t$ $+x_T^\top Q_f x_T$	$\widehat{w}_0, \ldots, \widehat{w}_{T-1}$ Chapter 2 ([23])
$x_{t+1} = Ax_t + Bu_t + f_t(x_t, u_t)$	$\sum_{t=0}^{\infty} x_t^\top Q x_t + u_t^\top R u_t$	$\widehat{\pi} : \mathbb{R}^n \to \mathbb{R}^m$ Chapter 3 ([16])
$x_{t+1} = f_t(x_t, u_t)$ $x_t \in \mathsf{X}_t\,(\mathbf{x}_{<t}, \mathbf{u}_{<t}) \subseteq \mathbb{X}$ $u_t \in \mathsf{U}_t\,(\mathbf{x}_{<t}, \mathbf{u}_{<t}) \subseteq \mathbb{U}$	$\sum_{t=1}^{T} c_t(u_t)$	$p_t^* : \mathbb{U} \to \mathbb{X}$ Chapter 4 and 5 ([24–26])

Table 8.1.1: System models for the learning-augmented algorithms with different types of imperfect/untrusted predictions or black-box AI/ML advice. The detailed definitions of notation can be found in the corresponding chapters.

Next, we conclude the previous chapters.

8.2 Summary of Chapters

In Chapter 2, we have detailed an approach that allows the use of black-box AI tools in a way that ensures worst-case performance bounds for linear quadratic

control. Further, we demonstrate the effectiveness of our approach in multiple applications. The results highlight a trade-off between robustness and consistency in linear quadratic control problems wherein the system perturbations are adversarial and the predictions of perturbations are untrusted.

Chapter 3 considers a novel combination of pre-trained black-box policies with model-based advice from crude model information. A general adaptive policy is proposed, with theoretical guarantees on both stability and sub-optimality. The effectiveness of the adaptive policy is validated empirically. The results presented lead to an important first step towards improving the practicality of existing DNN-based algorithms when using them as black-boxes in non-linear real-world control problems.

Chapter 4 formalizes and studies the closed-loop control framework created by the interaction between a system operator and an aggregator in a smart grid. Our focus is on the feedback signal provided by the aggregator to the operator that summarizes the real-time availability of flexibility among the loads controlled by the aggregator. We present the design of an maximum entropy feedback (MEF) signal based on entropic maximization. We prove a close connection between the MEF signal and the system capacity, and show that when the signal is used the system operator can perform online cost minimization while provably respecting the private constraints of the loads controlled by the aggregator and satisfying optimality under certain regularity assumptions. Further, we illustrate the effectiveness of these designs using simulation experiments of an EV charging facility.

In Chapter 5, we have studied and analyzed a closed-loop control framework created by the interaction between a central controller and a local controller. Our analysis shows that it is possible to combine model predictive control with limited aggregate feedback about feasibility to, under certain model assumptions, achieve a sub-linear dynamic regret. Focusing on the analytic side, this work presents the first analysis of a two-controller system where a local controller governs a complicated time-varying and coupling dynamical system and a central controller minimizes costs. To achieve our analytic guarantees, we make several modeling assumptions and the relaxation of these assumptions is an important task for future work. First, our penalized predictive control scheme is designed for 1-1 coordination between two controllers. It would be interesting to extend the control scheme to allow for the interaction between a central controller and multiple local participants. Second, our analysis assumes perfect maximum entropy feedback (MEF) but we see empirically that

approximations of the MEF can still guarantee feasibility. Generalizing our current results to approximations of MEF is an challenging and important direction. Third, we have used reinforcement learning to approximate the MEF in this work, but the approach is general and it is important to explore additional deep learning techniques for the estimation of the MEF.

In Chapter 6, a power system identification problem is considered. A sparsity characterization for distributions of random graphs (that are allowed to contain *high-degree* nodes) is provided, based on which we study fundamental trade-offs between the number of measurements, the complexity of the graph class, and the probability of error. Necessary and sufficient conditions on the number of measurements are given for both noisy and noiseless cases. For practical applications in power systems, a polynomial-time (in n) algorithm is designed and implemented.

Finally, Chapter 7 presents a publicly accessible EV charging dataset–ACN-Data, and the first analysis of the fine-grained charging data in ACN-Data, which develops a systematic method to learn battery behavior from the data. The results are used in the first two parts of this dissertation to build an EV charging environment that helps validate the proposed algorithms and schemes. The analysis shows that, even though the number of charging curves is large, they can be accurately classified into a small number of types. Moreover, the cluster representatives can be used for effective prediction. The analysis opens up potentials for future applications. For instance, a natural statistical EV model consists of $(\mathbf{c}_k, p_k, k = 1, \ldots, K)$ where $\mathbf{c}_k$ is the tail representative and p_k is the marginal probability that an EV arrival is of type $k = 1, \ldots, K$ as exemplified in Fig. 7.6.3. This model can be useful for planning purposes and for simulations, *e.g.*, to determine the capacity of electric infrastructure supplying a large-scale EV charging facility. For another instance, online optimization of EV charging can be implemented as a model predictive control (MPC) where a forward optimization problem is solved in each control interval (*c.f.*, [96, 101]). The representative for each individual user can be used as prediction to improve the performance of MPC. Moreover, the ability to decide the charging stage in real time as illustrated in Section 7.6 can be helpful to online scheduling. Yet another application is to use the representative tail $\mathbf{c}_k$ to detect abnormal battery behavior in real-time and alert the drivers, or charging facilities, or EV manufacturers.

8.3 Impact on Smart Grid Applications

Making use of predictions, forecasts, and simulation environments built upon ACN-Data presented in Chapter 7, we have demonstrated the efficacy of the learning-augmented control and decision-making policies developed in Chapter 2, 3, 4 ,and 5. By illustrating various smart grid applications using these results, we reply to the question raised at the beginning of this dissertation (Section 1.2) in the affirmative. In fact, besides the applications of large-scale adaptive EV charging and system operator-aggregator coordination presented in the previous chapters, there are many similar applications in smart grids that can be enabled and improved using novel learning-augmented approaches. The transition from a traditional power grid where coal power plants dominate to a smart grid where distributed energy resources are major components induces a transition from classical control and decision-making methods to data-driven robust learning-augmented policies. One of the key steps in the future is to explore novel learning-augmented algorithms that can impact more of these applications, as we will discuss in more detail in the next section.

8.4 Future Directions

Learning-Augmented Control

There are many potential future directions that build on the results in Chapter 2 and 3. First, in Chapter 2, we have considered a linear quadratic control problem, and an important extension will be to analyze the robustness and consistency of non-linear control systems. Second, our regret bound (Lemma 2) and competitive results (Theorem 2.4.1) are not tight when the variation of perturbations or predictions is high, therefore it is interesting to explore the idea of "follow-the-regularized-leader" [233] and understand if adding an extra regularizer in the update rule of λ for self-tuning control can improve the convergence and/or the regret. Third, characterizing a tight trade-off between robustness and consistency for linear quadratic control is of particular interest. For example, the results in [5, 6] together imply a tight robustness and consistency trade-off for the ski-rental problem. It would be interesting to explore if it is possible to do the same for linear quadratic control. Exploring other forms of model-based advice theoretically, and verifying practically other implementations to learn the confidence coefficients online are interesting future directions. Finally, the learning-augmented online policies considered in Chapter 2 and 3 involve linear combinations between classical policies and black-box policies equipped with predictions. Going beyond this aspect and developing non-linear approaches that make AI/ML policies more robust is another important future direction.

Learning-Augmented Decision-Making

There is much left to explore about the MEF signal presented in Chapter 4 and 5. In particular, computing it is computationally intensive and we use reinforcement learning for approximating the MEF. Improving the learning design and developing other approximations are of particular interest. Further, exploring the use of flexibility feedback for operational objectives beyond cost minimization and capacity estimation is an important goal. Finally, exploring the application of the defined real-time aggregate flexibility in other settings, such as multi-aggregator systems, frequency regulation, and real-time pricing, is exciting. We have illustrated the application of the proposed scheme to an EV charging application. Though we focus on the application of EV charging, the results in this work are applicable well beyond EV charging. For example, a similar scheme may be used in cloud computing and data center scheduling [105, 107] to make the network more sustainable. Further, the same design applies to the networks formed by other distributed energy resources (DERs) such as HVAC (heating, ventilation, and air conditioning) systems, rooftop solar PV units, energy storage systems and inverters. It will be interesting for future work to explore these applications.

EV Charging Data Analysis

We have presented some fundamental EV charging data analysis in Chapter 7. The charging tail analysis has several limitations that motivate extensions. First, our current method works well only with charging curves that exhibit relatively clean tail behavior. Additional techniques are needed to extract useful information from other charging curves. Second, our current method is offline. It would be useful to extend it to an online setting, for continuous improvement of classification performance and adaptation to changing EV behavior. Such an online method will be useful as the building block for many online applications. Here theories and algorithms in statistical detection and signal processing will prove to be helpful. Third, we model battery behavior by the representative tail $\mathbf{c}_k$ as functions of time. More detailed battery models can be developed using $\mathbf{c}_k$ and other information such as the energy capacities of the batteries and the voltage time series, *e.g.*, their current and voltage behavior in the absorption stage as functions of their states of charge. Finally, it would be interesting to develop a tractable mathematical model of the classification framework shown in Figure 7.4.3 and formally prove its convergence and optimality properties.

Bibliography

[1] Zachary J. Lee, Tongxin Li, and Steven H. Low. Acn-data: Analysis and applications of an open ev charging dataset. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, pages 139–149, 2019.

[2] Nicolas Christianson, Tinashe Handina, and Adam Wierman. Chasing convex bodies and functions with black-box advice. In *Conference on Learning Theory*, pages 867–908. PMLR, 2022.

[3] Thodoris Lykouris and Sergei Vassilvtiskii. Competitive caching with machine learned advice. In *International Conference on Machine Learning*, pages 3296–3305. PMLR, 2018.

[4] Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, and Marc Renault. Online computation with untrusted advice. *arXiv preprint arXiv:1905.05655*, 2019.

[5] Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ml predictions. In *Advances in Neural Information Processing Systems*, pages 9661–9670, 2018.

[6] Alexander Wei and Fred Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. *Advances in Neural Information Processing Systems*, 33:8042–8053, 2020.

[7] Étienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. *Advances in Neural Information Processing Systems*, 33:20083–20094, 2020.

[8] Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. Secretary and online matching problems with machine learned advice. *Advances in Neural Information Processing Systems*, 33:7933–7944, 2020.

[9] Antonios Antoniadis, Christian Coester, Marek Elias, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. In *International Conference on Machine Learning*, pages 345–355. PMLR, 2020.

[10] Matthew Botvinick, Sam Ritter, Jane X Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement learning, fast and slow. *Trends in cognitive sciences*, 23(5):408–422, 2019.

[11] Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing solving sparse reward tasks from scratch. In *International Conference on Machine Learning*, pages 4344–4353. PMLR, 2018.

[12] Allan Jabri, Kyle Hsu, Abhishek Gupta, Ben Eysenbach, Sergey Levine, and Chelsea Finn. Unsupervised curricula for visual meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[13] Richard Cheng, Abhinav Verma, Gabor Orosz, Swarat Chaudhuri, Yisong Yue, and Joel Burdick. Control regularization for reduced variance reinforcement learning. In *International Conference on Machine Learning*, pages 1141–1150. PMLR, 2019.

[14] Benjamin Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:253–279, 2019.

[15] Xueying Bai, Jian Guan, and Hongning Wang. A model-based reinforcement learning with adversarial training for online recommendation. *Advances in Neural Information Processing Systems*, 32, 2019.

[16] Tongxin Li, Ruixiao Yang, Guannan Qu, Yiheng Lin, Steven Low, and Adam Wierman. Equipping black-box policies with model-based advice for stable nonlinear control. *Under review*.

[17] Keerti Anand, Rong Ge, Amit Kumar, and Debmalya Panigrahi. Online algorithms with multiple predictions. *arXiv preprint arXiv:2205.03921*, 2022.

[18] Shaofeng H-C Jiang, Erzhi Liu, You Lyu, Zhihao Gavin Tang, and Yubo Zhang. Online facility location with predictions. In *International Conference on Learning Representations*, 2021.

[19] Bo Sun, Russell Lee, Mohammad Hajiesmaili, Adam Wierman, and Danny Tsang. Pareto-optimal learning-augmented algorithms for online conversion problems. *Advances in Neural Information Processing Systems*, 34:10339–10350, 2021.

[20] Chenyang Xu and Guochuan Zhang. Learning-augmented algorithms for online subset sum. *Journal of Global Optimization*, pages 1–20, 2022.

[21] Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, and Marc Renault. Online computation with untrusted advice. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[22] Noah Golowich and Ankur Moitra. Can q-learning be improved with advice? In *Conference on Learning Theory*, pages 4548–4619. PMLR, 2022.

[23] Tongxin Li, Ruixiao Yang, Guannan Qu, Guanya Shi, Chenkai Yu, Adam Wierman, and Steven Low. Robustness and consistency in linear quadratic control with untrusted predictions. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(1):1–35, 2022.

[24] Tongxin Li, Steven H. Low, and Adam Wierman. Real-time flexibility feedback for closed-loop aggregator and system operator coordination. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, pages 279–292, 2020.

[25] Tongxin Li, Yue Chen, Bo Sun, Adam Wierman, and Steven H Low. Information aggregation for constrained online control. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(2):1–35, 2021.

[26] Tongxin Li, Bo Sun, Yue Chen, Zixin Ye, Steven H Low, and Adam Wierman. Learning-based predictive control via real-time aggregate flexibility. *IEEE Transactions on Smart Grid*, 12(6):4897–4913, 2021.

[27] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in Neural Information Processing Systems*, 30, 2017.

[28] Mario Zanon and Sébastien Gros. Safe reinforcement learning using robust mpc. *IEEE Transactions on Automatic Control*, 66(8):3638–3652, 2020.

[29] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[30] Theodore J Perkins and Andrew G Barto. Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3(Dec):803–832, 2002.

[31] Ming Jin and Javad Lavaei. Stability-certified reinforcement learning: A control-theoretic perspective. *IEEE Access*, 8:229086–229100, 2020.

[32] He Hao and Wei Chen. Characterizing flexibility of an aggregation of deferrable loads. In *53rd IEEE Conference on Decision and Control*, pages 4059–4064. IEEE, 2014.

[33] He Hao, Borhan M Sanandaji, Kameshwar Poolla, and Tyrone L Vincent. Aggregate flexibility of thermostatically controlled loads. *IEEE Transactions on Power Systems*, 30(1):189–198, 2014.

[34] Andrey Bernstein, Jean-Yves Le Boudec, Mario Paolone, Lorenzo Reyes-Chamorro, and Wajeb Saab. Aggregation of power capabilities of heterogeneous resources for real-time control of power grids. In *2016 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE, 2016.

[35] Lin Zhao, Wei Zhang, He Hao, and Karanjit Kalsi. A geometric approach to aggregate flexibility modeling of thermostatically controlled loads. *IEEE Transactions on Power Systems*, 32(6):4721–4731, 2017.

[36] Tianyi Chen, Na Li, and Georgios B Giannakis. Aggregating flexibility of heterogeneous energy resources in distribution networks. In *2018 Annual American Control Conference (ACC)*, pages 4604–4609. IEEE, 2018.

[37] James Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3(2):97–139, 1957.

[38] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[39] Yingying Li, Xin Chen, and Na Li. Online optimal control with linear dynamics and predictions: Algorithms and regret analysis. *Advances in Neural Information Processing Systems*, 32, 2019.

[40] Chenkai Yu, Guanya Shi, Soon-Jo Chung, Yisong Yue, and Adam Wierman. The power of predictions in online control. *arXiv preprint arXiv:2006.07569*, 2020.

[41] Chenkai Yu, Guanya Shi, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Competitive control with delayed imperfect information. *arXiv preprint arXiv:2010.11637*, 2020.

[42] Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Online optimization with memory and competitive control. In *Advances in Neural Information Processing Systems*, volume 33, pages 20636–20647. Curran Associates, Inc., 2020.

[43] Runyu Zhang, Yingying Li, and Na Li. On the regret analysis of online lqr control with predictions. *arXiv preprint arXiv:2102.01309*, 2021.

[44] Geir E Dullerud and Fernando Paganini. *A course in robust control theory: A convex approach*, volume 36. Springer Science & Business Media, 2013.

[45] John Doyle, Keith Glover, Pramod Khargonekar, and Bruce Francis. State-space solutions to standard h_2 and h_∞ control problems. In *1988 American Control Conference*, pages 1691–1696. IEEE, 1988.

[46] Kemin Zhou and John Comstock Doyle. *Essentials of robust control*, volume 104. Prentice Hall Upper Saddle River, NJ, 1998.

[47] Alberto Bemporad and Manfred Morari. Robust model predictive control: A survey. In *Robustness in Identification and Control*, pages 207–226. Springer, 1999.

[48] Jean-Jacques E Slotine and Weiping Li. *Applied nonlinear control*, volume 199. Prentice Hall Englewood Cliffs, NJ, 1991.

[49] Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 1–26. JMLR Workshop and Conference Proceedings, 2011.

[50] Max Simchowitz and Dylan Foster. Naive exploration is optimal for online lqr. In *International Conference on Machine Learning*, pages 8937–8948. PMLR, 2020.

[51] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, pages 1–47, 2019.

[52] Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. Online control with adversarial disturbances. In *International Conference on Machine Learning*, pages 111–119. PMLR, 2019.

[53] Brian D. O. Anderson and John B. Moore. *Optimal filtering*. Courier Corporation, 2012.

[54] Alon Cohen, Avinatan Hasidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. In *International Conference on Machine Learning*, pages 1029–1038. PMLR, 2018.

[55] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[56] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[57] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[58] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[59] Guannan Qu, Chenkai Yu, Steven Low, and Adam Wierman. Exploiting linear models for model-free nonlinear control: A provably convergent policy gradient approach. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6539–6546. IEEE, 2021.

[60] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

[61] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[62] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[63] Joao P. Hespanha and A. Stephen Morse. Switching between stabilizing controllers. *Automatica*, 38(11):1905–1917, 2002.

[64] Henrik Niemann, Jakob Stoustrup, and Rune B. Abrahamsen. Switching between multivariable controllers. *Optimal Control Applications and Methods*, 25(2):51–66, 2004.

[65] Ugo Rosolia and Francesco Borrelli. Learning model predictive control for iterative tasks: A data-driven control framework. *IEEE Transactions on Automatic Control*, 63(7):1883–1896, 2017.

[66] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE, 2018.

[67] Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value expansion for efficient model-free reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, 2018.

[68] Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalence is efficient for linear quadratic control. *Advances in Neural Information Processing Systems*, 32, 2019.

[69] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4):633–679, 2020.

[70] William M Wonham. On a matrix riccati equation of stochastic control. *SIAM Journal on Control*, 6(4):681–697, 1968.

[71] Yang Zheng, Luca Furieri, Antonis Papachristodoulou, Na Li, and Maryam Kamgarpour. On the equivalence of youla, system-level, and input–output parameterizations. *IEEE Transactions on Automatic Control*, 66(1):413–420, 2020.

[72] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937. PMLR, 2016.

[73] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[74] Duncan S. Callaway and Ian A. Hiskens. Achieving controllability of electric loads. *Proceedings of the IEEE*, 99(1):184–199, 2010.

[75] Scott Burger, Jose Pablo Chaves-Ávila, Carlos Batlle, and Ignacio J. Pérez-Arriaga. A review of the value of aggregators in electricity systems. *Renewable and Sustainable Energy Reviews*, 77:395–405, 2017.

[76] Intisar Ali Sajjad, Gianfranco Chicco, and Roberto Napoli. Definitions of demand flexibility for aggregate residential loads. *IEEE Transactions on Smart Grid*, 7(6):2633–2643, 2016.

[77] Daria Madjidian, Mardavij Roozbehani, and Munther A Dahleh. Energy storage from aggregate deferrable demand: Fundamental trade-offs and scheduling policies. *IEEE Transactions on Power Systems*, 33(4):3573–3586, 2018.

[78] Nasrin Sadeghianpourhamami, Nazir Refa, Matthias Strobbe, and Chris Develder. Quantitive analysis of electric vehicle flexibility: A data-driven approach. *International Journal of Electrical Power & Energy Systems*, 95:451–462, 2018.

[79] Michael P. Evans, Simon H. Tindemans, and David Angeli. A graphical measure of aggregate flexibility for energy-constrained distributed resources. *IEEE Transactions on Smart Grid*, 2019.

[80] George Wenzel, Matias Negrete-Pincetic, Daniel E Olivares, Jason MacDonald, and Duncan S Callaway. Real-time charging strategies for an electric vehicle aggregator to provide ancillary services. *IEEE Transactions on Smart Grid*, 9(5):5141–5151, 2017.

[81] He Hao, Yashen Lin, Anupama S. Kowli, Prabir Barooah, and Sean Meyn. Ancillary service to the grid through control of fans in commercial building hvac systems. *IEEE Transactions on Smart Grid*, 5(4):2066–2074, 2014.

[82] Tianshu Wei, Qi Zhu, and Nanpeng Yu. Proactive demand participation of smart buildings in smart grid. *IEEE Transactions on Computers*, 65(5):1392–1406, 2015.

[83] Sean P Meyn, Prabir Barooah, Ana Bušić, Yue Chen, and Jordan Ehren. Ancillary service to the grid using intelligent deferrable loads. *IEEE Transactions on Automatic Control*, 60(11):2847–2862, 2015.

[84] Anand Subramanian, Manuel J Garcia, Duncan S Callaway, Kameshwar Poolla, and Pravin Varaiya. Real-time scheduling of distributed resources. *IEEE Transactions on Smart Grid*, 4(4):2122–2130, 2013.

[85] Dimitrios Papadaskalopoulos, Goran Strbac, Pierluigi Mancarella, Marko Aunedi, and Vladimir Stanojevic. Decentralized participation of flexible demand in electricity markets–Part II: Application with electric vehicles and heat pump systems. *IEEE Transactions on Power Systems*, 28(4):3667–3674, 2013.

[86] Shuoyao Wang, Suzhi Bi, and Ying Jun Angela Zhang. Reinforcement learning for real-time pricing and scheduling control in ev charging stations. *IEEE Transactions on Industrial Informatics*, 2019.

[87] Yanzhi Wang, Xue Lin, and Massoud Pedram. A near-optimal model-based control algorithm for households equipped with residential photovoltaic power generation and energy storage systems. *IEEE Transactions on Sustainable Energy*, 7(1):77–86, 2015.

[88] Bert J. Claessens, Dirk Vanhoudt, Johan Desmedt, and Frederik Ruelens. Model-free control of thermostatically controlled loads connected to a district heating network. *Energy and Buildings*, 159:1–10, 2018.

[89] Bingqing Chen, Weiran Yao, Jonathan Francis, and Mario Bergés. Learning a distributed control scheme for demand flexibility in thermostatically controlled loads. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–7. IEEE, 2020.

[90] Zhongjing Ma, Duncan S. Callaway, and Ian A. Hiskens. Decentralized charging control of large populations of plug-in electric vehicles. *IEEE Transactions on Control Systems Technology*, 21(1):67–78, 2011.

[91] Lingwen Gan, Ufuk Topcu, and Steven H. Low. Optimal decentralized protocol for electric vehicle charging. *IEEE Transactions on Power Systems*, 28(2):940–951, 2012.

[92] Emre C. Kara, Jason S. Macdonald, Douglas Black, Mario Bérges, Gabriela Hug, and Sila Kiliccote. Estimating the benefits of electric vehicle smart charging at non-residential locations: A data-driven approach. *Applied Energy*, 155:515–525, 2015.

[93] Xin Chen, Emiliano Dall'Anese, Changhong Zhao, and Na Li. Aggregate power flexibility in unbalanced distribution systems. *IEEE Transactions on Smart Grid*, 11(1):258–269, 2019.

[94] Mousa Marzband, Andreas Sumper, José Luis Domínguez-García, and Ramon Gumara-Ferret. Experimental validation of a real time energy management system for microgrids in islanded mode using a local day-ahead electricity market and minlp. *Energy Conversion and Management*, 76:314–322, 2013.

[95] Pierluigi Siano and Debora Sarno. Assessing the benefits of residential demand response in a real time distribution energy market. *Applied Energy*, 161:533–551, 2016.

[96] Zachary J. Lee, Daniel Chang, Cheng Jin, George S. Lee, Rand Lee, Ted Lee, and Steven H. Low. Large-scale adaptive electric vehicle charging. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–7. IEEE, 2018.

[97] Shashank Singh, Ananya Uppal, Boyue Li, Chun-Liang Li, Manzil Zaheer, and Barnabás Póczos. Nonparametric density estimation under adversarial losses. In *Advances in Neural Information Processing Systems*, pages 10225–10236, 2018.

[98] Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on Systems, Man, and Cybernetics*, (5):834–846, 1983.

[99] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[100] Eduardo F. Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[101] Zachary J Lee, George Lee, Ted Lee, Cheng Jin, Rand Lee, Zhi Low, Daniel Chang, Christine Ortega, and Steven H Low. Adaptive charging networks: A framework for smart electric vehicle charging. *IEEE Transactions on Smart Grid*, 12(5):4339–4350, 2021.

[102] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[103] Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli. Data-driven predictive control for autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:259–286, 2018.

[104] Xuanyu Cao and K. J. Ray Liu. Online convex optimization with time-varying constraints and bandit feedback. *IEEE Transactions on Automatic Control*, 64(7):2665–2680, 2018.

[105] Tianyi Chen and Georgios B. Giannakis. Bandit convex optimization for scalable and dynamic iot management. *IEEE Internet of Things Journal*, 6(1):1276–1286, 2018.

[106] Tianyi Chen, Qing Ling, and Georgios B Giannakis. An online convex optimization approach to proactive network resource allocation. *IEEE Transactions on Signal Processing*, 65(24):6350–6364, 2017.

[107] Hao Yu, Michael Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. In *Advances in Neural Information Processing Systems*, pages 1428–1438, 2017.

[108] Andrey Bernstein, Emiliano Dall'Anese, and Andrea Simonetto. Online primal-dual methods with measurement feedback for time-varying convex optimization. *IEEE Transactions on Signal Processing*, 67(8):1978–1991, 2019.

[109] Xinlei Yi, Xiuxian Li, Lihua Xie, and Karl H. Johansson. Distributed online convex optimization with time-varying coupled inequality constraints. *IEEE Transactions on Signal Processing*, 68:731–746, 2020.

[110] Antoine Lesage-Landry, Iman Shames, and Joshua A Taylor. Predictive online convex optimization. *Automatica*, 113:108771, 2020.

[111] Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7195–7201. IEEE, 2016.

[112] Eric C. Hall and Rebecca M. Willett. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, 2015.

[113] Niangjun Chen, Joshua Comden, Zhenhua Liu, Anshul Gandhi, and Adam Wierman. Using predictions in online optimization: Looking forward with an eye on the past. *ACM SIGMETRICS Performance Evaluation Review*, 44(1):193–206, 2016.

[114] Alec Koppel, Felicia Y. Jakubiec, and Alejandro Ribeiro. A saddle point algorithm for networked online convex optimization. *IEEE Transactions on Signal Processing*, 63(19):5149–5164, 2015.

[115] Alec Koppel, Brian M. Sadler, and Alejandro Ribeiro. Proximity without consensus in online multiagent optimization. *IEEE Transactions on Signal Processing*, 65(12):3062–3077, 2017.

[116] Yingying Li, Guannan Qu, and Na Li. Using predictions in online optimization with switching costs: A fast algorithm and a fundamental limit. In *2018 Annual American Control Conference (ACC)*, pages 3008–3013. IEEE, 2018.

[117] Ming Shi, Xiaojun Lin, Sonia Fahmy, and Dong-Hoon Shin. Competitive online convex optimization with switching costs and ramp constraints. In *IEEE*

INFOCOM 2018-IEEE Conference on Computer Communications, pages 1835–1843. IEEE, 2018.

[118] Qiulin Lin, Hanling Yi, John Pang, Minghua Chen, Adam Wierman, Michael Honig, and Yuanzhang Xiao. Competitive online optimization under inventory constraints. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(1):1–28, 2019.

[119] Gautam Goel and Adam Wierman. An online algorithm for smoothed regression and lqr control. *Proceedings of Machine Learning Research*, 89:2504–2513, 2019.

[120] Masoud Badiei, Na Li, and Adam Wierman. Online convex optimization with ramp constraints. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 6730–6736. IEEE, 2015.

[121] Sarah Dean, Stephen Tu, Nikolai Matni, and Benjamin Recht. Safely learning to control the constrained linear quadratic regulator. In *2019 American Control Conference (ACC)*, pages 5582–5588. IEEE, 2019.

[122] Wen Sun, Debadeepta Dey, and Ashish Kapoor. Safety-aware algorithms for adversarial contextual bandit. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3280–3288. JMLR. org, 2017.

[123] Jianjun Yuan and Andrew Lamperski. Online convex optimization for cumulative constraints. In *Advances in Neural Information Processing Systems*, pages 6137–6146, 2018.

[124] Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: Price of past mistakes. In *Advances in Neural Information Processing Systems*, pages 784–792, 2015.

[125] Naman Agarwal, Elad Hazan, and Karan Singh. Logarithmic regret for online control. In *Advances in Neural Information Processing Systems*, pages 10175–10184, 2019.

[126] Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Beyond no-regret: Competitive control via online optimization with memory. *arXiv preprint arXiv:2002.05318*, 2020.

[127] Yanzhe Murray Lei, Stefanus Jasin, and Amitabh Sinha. Near-optimal bisection search for nonparametric dynamic pricing with inventory constraint. *Ross School of Business Paper*, (1252), 2014.

[128] Lijun Zhang, Tianbao Yang, Zhi-Hua Zhou, et al. Dynamic regret of strongly adaptive methods. In *International Conference on Machine Learning*, pages 5882–5891, 2018.

[129] Alon Cohen, Tomer Koren, and Yishay Mansour. Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret. In *International Conference on Machine Learning*, pages 1300–1309, 2019.

[130] Minghong Lin, Zhenhua Liu, Adam Wierman, and Lachlan LH Andrew. Online algorithms for geographical load balancing. In *2012 International Green Computing Conference (IGCC)*, pages 1–10. IEEE, 2012.

[131] Miguel A. Ortega-Vazquez, François Bouffard, and Vera Silva. Electric vehicle aggregator/system operator coordination for charging scheduling and services procurement. *IEEE Transactions on Power Systems*, 28(2):1806–1815, 2012.

[132] Linqi Guo, Karl F. Erliksson, and Steven H. Low. Optimal online adaptive electric vehicle charging. In *2017 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2017.

[133] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Artificial Intelligence and Statistics*, pages 398–406, 2015.

[134] Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

[135] Shie Mannor, John N Tsitsiklis, and Jia Yuan Yu. Online learning with sample path constraints. *Journal of Machine Learning Research*, 10(Mar):569–590, 2009.

[136] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.

[137] Eric Hall and Rebecca Willett. Dynamical models and tracking regret in online convex programming. In *International Conference on Machine Learning*, pages 579–587, 2013.

[138] Yiheng Lin, Gautam Goel, and Adam Wierman. Online optimization with predictions and non-convex losses. *arXiv preprint arXiv:1911.03827*, 2019.

[139] Niangjun Chen, Anish Agarwal, Adam Wierman, Siddharth Barman, and Lachlan L. H. Andrew. Online convex optimization using predictions. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 191–204, 2015.

[140] Lars Grüne and Simon Pirkelmann. Economic model predictive control for time-varying system: Performance and stability results. *Optimal Control Applications and Methods*, 41(1):42–64, 2020.

[141] Aditya Bhaskara, Ashok Cutkosky, Ravi Kumar, and Manish Purohit. Online learning with imperfect hints. *arXiv preprint arXiv:2002.04726*, 2020.

[142] Bo Sun, Ali Zeynali, Tongxin Li, Mohammad Hajiesmaili, Adam Wierman, and Danny H. K. Tsang. Competitive algorithms for the online multiple knapsack problem with application to electric vehicle charging. *arXiv preprint arXiv:2010.00412*, 2020.

[143] Romer Rosales and Stan Sclaroff. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. Technical report, Boston University Computer Science Department, 1998.

[144] Xianfu Wang. Volumes of generalized unit balls. *Mathematics Magazine*, 78(5):390–395, 2005.

[145] C. K. Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

[146] C. K. Chow and T. Wagner. Consistency of an estimate of tree-dependent probability distributions (corresp.). *IEEE Transactions on Information Theory*, 19(3):369–371, 1973.

[147] Vincent Y. F. Tan, Animashree Anandkumar, and Alan S. Willsky. Learning gaussian tree models: Analysis of error exponents and extremal structures. *IEEE Transactions on Signal Processing*, 58(5):2701–2714, 2010.

[148] Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard. Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, 36(3):44–63, 2019.

[149] Asish Ghoshal and Jean Honorio. Learning linear structural equation models in polynomial time and sample complexity. In *International Conference on Artificial Intelligence and Statistics*, pages 1466–1475, 2018.

[150] Asish Ghoshal and Jean Honorio. Learning identifiable gaussian bayesian networks in polynomial time and sample complexity. In *Advances in Neural Information Processing Systems*, pages 6457–6466, 2017.

[151] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Analyzing graph structure via linear measurements. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 459–467. SIAM, 2012.

[152] Jean Pouget-Abadie and Thibaut Horel. Inferring graphs from cascades: A sparse recovery framework. In *International Conference on Machine Learning*, pages 977–986. PMLR, 2015.

[153] James A. Momoh, Rambabu Adapa, and M. E. El-Hawary. A review of selected optimal power flow literature to 1993. I. nonlinear and quadratic programming approaches. *IEEE transactions on Power Systems*, 14(1):96–104, 1999.

[154] Steven H. Low. Convex relaxation of optimal power flow—Part I: formulations and equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27, 2014.

[155] Yujie Tang, Krishnamurthy Dvijotham, and Steven Low. Real-time optimal power flow. *IEEE Transactions on Smart Grid*, 8(6):2963–2973, 2017.

[156] Anshul Mittal, Jagabondhu Hazra, Nikhil Jain, Vivek Goyal, Deva P. Seetharam, and Yogish Sabharwal. Real-time contingency analysis for power grids. In *European Conference on Parallel Processing*, pages 303–315. Springer, 2011.

[157] Ricardo Horta, Jairo Espinosa, and Julián Patiño. Frequency and voltage control of a power system with information about grid topology. In *Automatic Control (CCAC), 2015 IEEE 2nd Colombian Conference on*, pages 1–6. IEEE, 2015.

[158] Linqi Guo, Chen Liang, Alessandro Zocca, Steven H. Low, and Adam Wierman. Failure localization in power systems via tree partitions. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6832–6839. IEEE, 2018.

[159] Long Zhao and Bo Zeng. Vulnerability analysis of power grids with line switching. *IEEE Transactions on Power Systems*, 28(3):2727–2736, 2013.

[160] Sundeep Prabhakar Chepuri, Sijia Liu, Geert Leus, and Alfred O Hero. Learning sparse graphs under smoothness prior. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6508–6512. IEEE, 2017.

[161] Gautam Dasarathy, Parikshit Shah, Badri Narayan Bhaskar, and Robert D Nowak. Sketching sparse matrices, covariances, and graphs via tensor products. *IEEE Transactions on Information Theory*, 61(3):1373–1388, 2015.

[162] Eugene Belilovsky, Kyle Kastner, Gaël Varoquaux, and Matthew B Blaschko. Learning to discover sparse graphical models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 440–448. JMLR. org, 2017.

[163] Andrej Bogdanov, Elchanan Mossel, and Salil Vadhan. The complexity of distinguishing markov random fields. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 331–342. Springer, 2008.

[164] Narayana P. Santhanam and Martin J. Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Transactions on Information Theory*, 58(7):4117–4134, 2012.

[165] Robert M. Fano and David Hawkins. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29:793–794, 1961.

[166] Animashree Anandkumar, Vincent Y. F. Tan, Furong Huang, and Alan S. Willsky. High-dimensional structure estimation in ising models: Local separation criterion. *The Annals of Statistics*, 40(3):1346–1375, 2012.

[167] Asish Ghoshal and Jean Honorio. Information-theoretic limits of Bayesian network structure learning. In *Artificial Intelligence and Statistics*, pages 767–775. PMLR, 2017.

[168] Shuchin Aeron, Venkatesh Saligrama, and Manqi Zhao. Information theoretic bounds for compressed sensing. *IEEE Transactions on Information Theory*, 56(10):5111–5130, 2010.

[169] Guido Cavraro, Vassilis Kekatos, and Sriharsha Veeramachaneni. Voltage analytics for power distribution network topology verification. *IEEE Transactions on Smart Grid*, 10(1):1058–1067, 2017.

[170] Yu Christine Chen, Taposh Banerjee, Alejandro D. Dominguez-Garcia, and Venugopal V. Veeravalli. Quickest line outage detection and identification. *IEEE Transactions on Power Systems*, 31(1):749–758, 2015.

[171] Yoav Sharon, Anuradha M. Annaswamy, Alexis L. Motto, and Amit Chakraborty. Topology identification in distribution network with limited measurements. In *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pages 1–6. IEEE, 2012.

[172] Deepjyoti Deka, Scott Backhaus, and Michael Chertkov. Structure learning in power distribution networks. *IEEE Transactions on Control of Network Systems*, 5(3):1061–1074, 2017.

[173] Xiao Li, H. Vincent Poor, and Anna Scaglione. Blind topology identification for power systems. In *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 91–96. IEEE, 2013.

[174] Ye Yuan, Omid Ardakanian, Steven Low, and Claire Tomlin. On the inverse power flow problem. *arXiv preprint arXiv:1610.06631*, 2016.

[175] Jiafan Yu, Yang Weng, and Ram Rajagopal. Patopa: A data-driven parameter and topology joint estimation framework in distribution grids. *IEEE Transactions on Power Systems*, 33(4):4335–4347, 2017.

[176] Seiun Park, Deepjyoti Deka, and Michael Chcrtkov. Exact topology and parameter estimation in distribution grids with minimal observability. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–6. IEEE, 2018.

[177] Hao Zhu and Georgios B. Giannakis. Sparse overcomplete representations for efficient identification of power line outages. *IEEE Transactions on Power Systems*, 27(4):2215–2224, 2012.

[178] Vincent Y. F. Tan and Alan S. Willsky. Sample complexity for topology estimation in networks of lti systems. *IFAC Proceedings Volumes*, 44(1):9079–9084, 2011.

[179] Yizheng Liao, Yang Weng, Meng Wu, and Ram Rajagopal. Distribution grid topology reconstruction: An information theoretic approach. In *North American Power Symposium (NAPS), 2015*, pages 1–6. IEEE, 2015.

[180] Saverio Bolognani, Nicoletta Bof, Davide Michelotti, Riccardo Muraro, and Luca Schenato. Identification of power distribution network topology via voltage correlation analysis. In *52nd IEEE Conference on Decision and Control*, pages 1659–1664. IEEE, 2013.

[181] Deepjyoti Deka, Scott Backhaus, and Michael Chertkov. Estimating distribution grid topologies: A graphical learning based approach. In *Power Systems Computation Conference (PSCC), 2016*, pages 1–7. IEEE, 2016.

[182] Emmanuel Candes, Mark Rudelson, Terence Tao, and Roman Vershynin. Error correction via linear programming. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 668–681. IEEE, 2005.

[183] Mark Rudelson and Roman Vershynin. On sparse reconstruction from fourier and gaussian measurements. *Communications on Pure and Applied Mathematics*, 61(8):1025–1045, 2008.

[184] Michael Lustig, David L. Donoho, Juan M. Santos, and John M. Pauly. Compressed sensing MRI. *IEEE Signal Processing Magazine*, 25(2):72, 2008.

[185] Shancang Li, Li Da Xu, and Xinheng Wang. Compressed sensing signal and data acquisition in wireless sensor networks and internet of things. *IEEE Transactions on Industrial Informatics*, 9(4):2177–2186, 2012.

[186] Christian R. Berger, Zhaohui Wang, Jianzhong Huang, and Shengli Zhou. Application of compressive sensing to sparse channel estimation. *IEEE Communications Magazine*, 48(11):164–174, 2010.

[187] Tongxin Li, Mayank Bakshi, and Pulkit Grover. Fundamental limits and achievable strategies for low energy compressed sensing with applications in wireless communication. In *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–6. IEEE, 2016.

[188] Mohammad Babakmehr, Marcelo G. Simões, Michael B. Wakin, and Farnaz Harirchi. Compressive sensing-based topology identification for smart grids. *IEEE Transactions on Industrial Informatics*, 12(2):532–543, 2016.

[189] Marco F. Duarte and Richard G. Baraniuk. Kronecker compressive sensing. *IEEE Transactions on Image Processing*, 21(2):494–504, 2011.

[190] Sadegh Jokar and Volker Mehrmann. Sparse solutions to underdetermined kronecker product systems. *Linear Algebra and its Applications*, 431(12):2437–2447, 2009.

[191] Shriram Sarvotham, Dror Baron, Michael Wakin, Marco F. Duarte, and Richard G. Baraniuk. Distributed compressed sensing of jointly sparse signals. In *Asilomar Conference on Signals, Systems, and Computers*, pages 1537–1541, 2005.

[192] Douglas Brent West. *Introduction to graph theory*, volume 2. Prentice Hall Upper Saddle River, 2001.

[193] Emmanuel J. Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[194] David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[195] Dongdong Ge, Xiaoye Jiang, and Yinyu Ye. A note on the complexity of lp minimization. *Mathematical Programming*, 129(2):285–299, 2011.

[196] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011.

[197] CVX Research, Inc. CVX: Matlab software for disciplined convex programming, version 2.0, August 2012.

[198] Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2018.

[199] *Bonneville Power Administration*. Accessed on Oct. 2016.

[200] Hiroshi Kajimoto. An extension of the prüfer code and assembly of connected graphs from their blocks. *Graphs and Combinatorics*, 19(2):231–239, 2003.

[201] Emmanuel J. Candes, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

[202] IEA 2019. Global EV outlook 2019. Available at `www.iea.org/publications/reports/globalevoutlook2019/`.

[203] G. A. Putrus, Pasist Suwanapingkarl, David Johnston, E. C. Bentley, and Mahinsasa Narayana. Impact of electric vehicles on power distribution networks. In *2009 IEEE Vehicle Power and Propulsion Conference*, pages 827–831. IEEE, 2009.

[204] J. D. Cross and R. Hartshorn. My electric avenue: Integrating electric vehicles into the electrical networks. 2016.

[205] Kristien Clement, Edwin Haesen, and Johan Driesen. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Transactions on Power Systems*, 25(1):371–380, 2009.

[206] Qiuming Gong, Shawn Midlam-Mohler, Vincenzo Marano, and Giorgio Rizzoni. Study of PEV Charging on Residential Distribution Transformer Life. *IEEE Transactions on Smart Grid*, 3(1):404–412, March 2012.

[207] Jonathan Coignard, Samveg Saxena, Jeffery Greenblatt, and Dai Wang. Clean vehicles as an enabler for a clean electricity grid. *Environmental Research Letters*, 13(5):054031, 2018.

[208] Jose Rivera, Christoph Goebel, and Hans-Arno Jacobsen. Distributed convex optimization for electric vehicle aggregators. *IEEE Transactions on Smart Grid*, 8(4):1852–1863, July 2017.

[209] Behnam Khaki, Chicheng Chu, and Rajit Gadh. A hierarchical ADMM-based framework for EV charging scheduling. In *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, pages 1–9, Denver, CO, USA, April 2018. IEEE.

[210] Julian de Hoog, Tansu Alpcan, Marcus Brazil, Doreen Anne Thomas, and Iven Mareels. Optimal charging of electric vehicles taking distribution network constraints into account. *IEEE Transactions on Power Systems*, 30(1):365–375, January 2015.

[211] Alexander Schuller, Christoph M. Flath, and Sebastian Gottwalt. Quantifying load flexibility of electric vehicles for renewable energy integration. *Applied Energy*, 151:335–344, August 2015.

[212] Paul Denholm, Michael Kuss, and Robert M. Margolis. Co-benefits of large scale plug-in hybrid electric vehicle and solar PV deployment. *Journal of Power Sources*, 236:350–356, August 2013.

[213] Di Wu, Haibo Zeng, Chao Lu, and Benoit Boulet. Two-stage energy management for office buildings with workplace EV charging and renewable energy. *IEEE Transactions on Transportation Electrification*, 3(1):225–237, March 2017.

[214] Stephen Lee, Srinivasan Iyengar, David Irwin, and Prashant Shenoy. Shared solar-powered EV charging stations: Feasibility and benefits. In *2016 Seventh International Green and Sustainable Computing Conference (IGSC)*, pages 1–8, Hangzhou, China, 2016. IEEE.

[215] Yorie Nakahira, Niangjun Chen, Lijun Chen, and Steven H. Low. Smoothed least-laxity-first algorithm for EV charging. pages 242–251. ACM Press, 2017.

[216] Bin Wang, Yubo Wang, Hamidreza Nazaripouya, Charlie Qiu, Chi-cheng Chu, and Rajit Gadh. Predictive scheduling framework for electric vehicles considering uncertainties of user behaviors. *IEEE Internet of Things Journal*, pages 1–1, 2016.

[217] Bruce G. Lindsay. Mixture models: Theory, geometry and applications. *NSF-CBMS Regional Conference Series in Probability and Statistics*, 5:i–163, 1995.

[218] Emil Eirola and Amaury Lendasse. Gaussian mixture models for time series modelling, forecasting, and interpolation. In *International Symposium on Intelligent Data Analysis*, pages 162–173. Springer, 2013.

[219] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[220] Yu-Wei Chung, Behnam Khaki, Chicheng Chu, and Rajit Gadh. Electric vehicle user behavior prediction using hybrid kernel density estimator. In *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, pages 1–6. IEEE, 2018.

[221] Zhong Chen, Ziqi Zhang, Jiaqing Zhao, Bowen Wu, and Xueliang Huang. An Analysis of the charging characteristics of electric vehicles based on measured data and its application. *IEEE Access*, 6:24475–24487, 2018.

[222] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review. *Information Systems*, 53:16–38, 2015.

[223] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, 2013.

[224] Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. *ACM Sigmod Record*, 23(2):419–429, 1994.

[225] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pages 593–604. ACM, 2007.

[226] John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1855–1870. ACM, 2015.

[227] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

[228] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[229] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002.

[230] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000.

[231] Chenxi Sun, Tongxin Li, and Victor OK Li. Robust and consistent clustering recovery via sdp approaches. In *2018 IEEE Data Science Workshop (DSW)*, pages 46–50. IEEE, 2018.

[232] Peter J Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[233] Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 525–533. JMLR Workshop and Conference Proceedings, 2011.