Title

Risk-Aware Algorithms for Learning-Based Control With Applications to Energy and Mechatronic Systems

Permalink

https://escholarship.org/uc/item/5kw8k6j8

Author

Kandel, Aaron Isaac

Publication Date

2023

Peer reviewed|Thesis/dissertation

Risk-Aware Algorithms for Learning-Based Control With Applications to Energy and Mechatronic Systems

By

Aaron Kandel

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Francesco Borrelli, Co-chair
Professor Scott Moura, Co-chair
Professor Koushil Sreenath
Professor Alexandre Bayen

Fall 2023

Risk-Aware Algorithms for Learning-Based Control With Applications to Energy and Mechatronic Systems

Abstract

Risk-Aware Algorithms for Learning-Based Control With Applications to Energy and Mechatronic Systems

By

Aaron Kandel

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Francesco Borrelli, Co-chair

Professor Scott Moura, Co-chair

This dissertation leverages and develops the powerful out-of-sample safety certificates of Wasserstein ambiguity sets to create a suite of data-driven control algorithms that help solve safety-critical industrial problems. This work is motivated by the ongoing relevance of robustness and safety when applying data-driven decision making in the real world. For example, lithium-ion batteries are driving transitions to renewable energy sources. Optimizing their performance and longevity is of the utmost importance, but highly difficult due to their complex, nonlinear, and safety-critical electrochemical dynamics. While data-driven control can dramatically improve the performance of systems like lithium-ion batteries, certifying system safety remains an open challenge. This dissertation explores certifying learning-based controllers via distributionally robust optimization (DRO). We focus on Wasserstein ambiguity sets, DRO methods that draw worst-case realizations of random variables under relatively permissive assumptions. This makes them ideal for learning-based control, where data can be highly limited and the controller is likely encounter new experience unaccounted for in its training data.

In Chapter 2, we begin by presenting simple mathematical arguments that extend an existing reformulation of Wasserstein DRO to cases where dependence on decision variables x and random variables $\mathbf{R}$ can be nonconvex as long as x and $\mathbf{R}$ are separable. By cleverly modeling stochasticity in model uncertainty, we augment nonconvex optimal control problems with Wasserstein ambiguity sets to obtain idealized probabilistic safety certificates.

The remaining chapters extend this theoretical result across the range of model-based and model-free reinforcement learning. Chapter 2 explores offline model-based reinforcement learning within a latent state-space, with application to real-time fast-charging of li-ion batteries using electrochemical information. By leveraging the results of Chapter 2, we

can hedge against model and data errors to probabilistically guarantee safe distributional data-driven control.

Chapter 4 presents an end-to-end framework for safe learning-based control using nonlinear stochastic MPC. We focus on scenarios where the controller is applied directly to a system of which it has highly limited experience, toward safety during tabula-rasa learning-based control as a challenging case for validation. We validate findings with case studies of extreme lithium-ion battery fast charging and autonomous vehicle obstacle avoidance using a basic perception system.

Finally, in Chapter 5, we apply the same DRO architecture to value-based RL. We describe a structure for deep Q-learning within the framework of constrained Markov decision processes (CMDPs). By characterizing the uncertainty of constraint cost functions based on their temporal-difference errors, we augment relevant constraints with tightening offset variables based on DRO theory of Chapter 2.

In our concluding remarks, we discuss the broader relevance of our findings and map directions for future work.

This dissertation is dedicated to my older brother Elan, to my parents, Annette and Arie Kandel, and to my younger brother Yoni.

Contents

List of Figures

List of Tables

Acknowledgments

I wish to extend my profound gratitude to my PhD adviser, Prof. Scott Moura. His patience, generosity, and brilliance were integral to my productivity and sanity throughout my PhD. At this rate, I'd wager his support has won me at least 6 additional months with a full head of hair. The same goes for my friends and collaborators within eCal: Dylan Kato, Andrea Pozzi, Shiva Raisinghani, Chitra Dangwal, Preet Gill, Soomin Woo, Milad Memarzadeh, Saehong Park, Armando Domingos, Zach Gima, Dong Zhang, Bertrand Travacca, and Mathilde Badoual. The culture of eCal is so friendly and open that working together (even if only briefly in some cases) was a delight.

I would also like to acknowledge my adviser from my undergraduate research, Prof. Hosam Fathy. Hosam ignited my passion for research and outreach. Our collaboration during my PhD was limited, but I still felt like Hosam was one of my biggest cheerleaders.

A special thanks to my qualifying exam committee: Prof. Francesco Borrelli, Prof. Alexandre Bayen, Prof. Koushil Sreenath, and Prof. Thomas Courtade. My exam preparation coincided with a host of challenging life events, but your patience and generosity helped ensure I passed in one piece. The same goes for Prof. Borrelli, Prof. Bayen, and Prof. Sreenath's participation on my dissertation committee. Thank you for your thoughtful feedback.

I humbly thank my mentors and coworkers from my time at Google: Derek Pang, Akshay Gadde, Miaosen Wang, Melanie Hunter, and Rene Claus. Working at YouTube often took me outside of my comfort zone, and I am grateful that the skills I developed will be critical to my success throughout my career.

I would also like to thank the friends I made at NextProf Nexus 2022, especially Juan Diego Shiraishi Lombard, Xilal Rima, and Adam Caparco. Our conversations about academia, industry, graduate school, and beyond helped me iron out how my values align with different career paths. You all are among my role models.

A special thanks to my students from CE295, and my mentees from the T-PREP program at UC Berkeley.

I also want to thank my supervisors, fellow volunteers, dogs, and cats at the East Bay SPCA. I especially thank Ryan Ericksen for facilitating my onboarding, and supporting my work on the intake data team. Of all the ways I found to balance my life outside of work - my time at the SPCA was among the most rewarding and fulfilling. I'm looking forward to continuing our collaboration in the future.

Chapter 1

Introduction

This dissertation investigates techniques for learning-based control, with applications to energy and mechatronic systems. This work is motivated by the powerful capability of data-driven control to unlock new levels of performance and efficiency in real-world industrial systems. While data-driven methods can amplify impact, they still often struggle to do so while guaranteeing safe operating conditions. Current research explores guaranteeing the safe behavior of learning-based control systems, but these certificates remain largely elusive. This dissertation leverages distributionally robust optimization to create progress towards certified data-driven control by developing relevant theory and a suite of learning-based control algorithms. We validate these works on several problems in the mechatronics and energy systems domains.

1.1 Motivating Example - Energy Storage Systems

Lithium-ion batteries are a ubiquitous technology that has enabled revolutionary technological advancement for a host of applications including personal electronic devices, healthcare, vehicle electrification, and renewable energy technologies. Chances are you are reading this dissertation on a device powered by a lithium-ion battery.

Lithium-ion batteries (LIBs) are highly complex electrochemical systems. No two LIBs are precisely the same, due to even well controlled and minute variances throughout manufacturing processes, materials, storage, and usage conditions. Modern manufacturing tolerances have created greater consistency among cells, but many react with variability after prolonged usage. Even our most advanced dynamical models still fail to capture the granularity within the underlying electrochemistry that is perhaps only accessible via molecular modeling itself [32].

The relevance of li-ion batteries will continue to grow as our grid moves more towards utilization of intermittent renewable energy sources. A recent report predicts that investment into battery manufacturing infrastructure must increase by $150-$300 billion dollars in the next 30 years [105]. This will mean greater usage, as well as a growing need for our existing batteries to last longer and function more efficiently. Currently, battery management systems (BMS) utilize highly simplified battery models in their decision-making. Lithium-ion batteries are complex electrochemical systems whose full dynamics are computationally expensive to simulate. Those full-order dynamics could inform BMS that more effectively preserve the cell health, but using that information in real time and on embedded hardware is highly challenging.

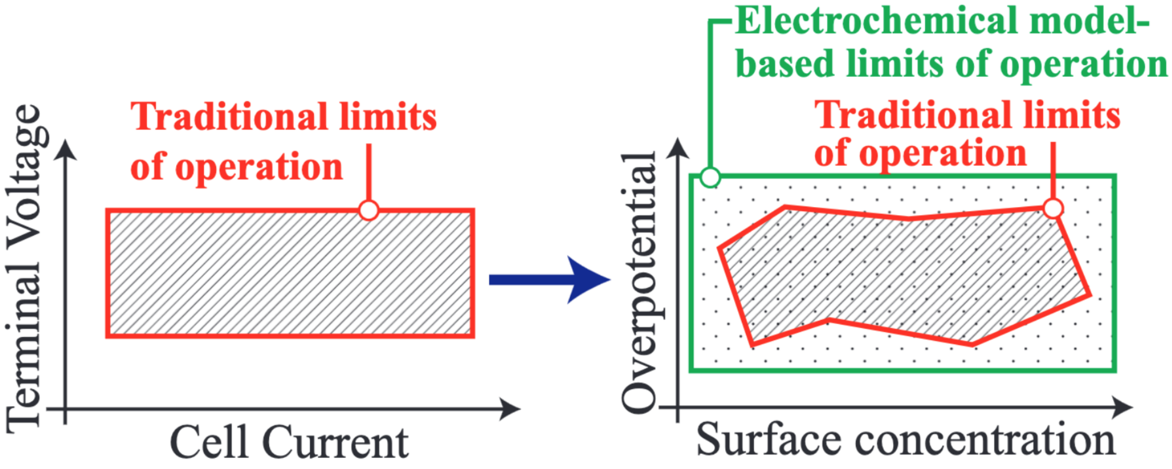
Terminal Voltage
Traditional limits of operation
Cell Current
Electrochemical model-based limits of operation
Traditional limits of operation
Overpotential
Surface concentration

Figure 1.1: Comparison of safe operating conditions defined by reduced order vs. full order model of li-ion battery dynamics. By considering granular and nonlinear electrochemical information most relevant to feasibility during control, the limits of the controller become much less restrictive compared to limits imposed by a simplified model that must approximate the truly relevant nonlinear relationships and variables [80].

Consider Figure 1.1, which visualizes this tradeoff. The left hand side visualizes the safe operating conditions for a li-ion cell as described by a reduced-order dynamical model (specifically an equivalent circuit model, or ECM). The right side shows the safe operating conditions of the cell modeled with its fine-resolution electrochemical relationships as defined by, for example, the Doyle-Fuller-Newman (DFN) model [31, 30, 32]. In the reduced-order case, we have highly limited information of the cell (e.g. input current, terminal voltage). Constraints on the reduced-order model attempt to emulate the known electrochemical relationships of the DFN that are directly tied to the safety of the battery cell. These constraints define the safe operating conditions shown on the left. When these constraints are projected into the electrochemical space on the right, we see that they create a conservative approximation of the true safe operating conditions as defined by the relevant electrochemical states themselves (two shown on the x and y axes). Thus, if we could create BMS using granular, high fidelity electrochemical information, we could take the battery closer towards the true boundary of its safe operating conditions. This would reduce charge times while also improving our ability to preserve the state-of-health (SOH) of the battery cell.

This tradeoff falls within the crux of control-oriented modeling. However, many systems are characterized beyond the scope of existing historical control-theoretic tools. Consider visuomotor control synthesis, where a control policy is learned from observations of RGB images of the environment [64, 20, 37]. Generally, the case of multimodality in the state space is becoming a significant space for LbC algorithms [67]. Modern models are being asked to process combinations of text, images, and numerical inputs, synthesizing information from all sources into appropriate predictions and decisions [106, 46]. Learning-based control is guiding development of many such models, and ensuring their alignment with safety and specifications [18]. While multimodal inputs are more immediately associated with safe control of autonomous vehicles using a fusion of classical state variables (e.g. position, velocity, etc.), camera image inputs, and lidar measurements; or guiding foundation models to avoid profanity and hallucinations in the text and images they generate - multimodal inputs from systems like batteries are also being shown to improve the performance of BMS. For example, recent work has improved estimation and control in BMS by considering fusion of classical observations (voltage, temperature, input current) with strain gauge measurements of the swelling of the battery cells [45, 36].

While LbC has seen a surge in application and study in recent years, the LbC problem space borrows many concepts from historical research on stochastic optimal control - a field which dates back decades to the original linear-quadratic Gaussian problem [7]. LbC has a powerful ability to synthesize control from high-dimensional, nonlinear, unintuitive systems, but the frequent lack of transparency of black-box learning means the robustness of the resulting controller is difficult to certify.

Exploring the question of certifying LbC systems dates back decades in the literature. The key underlying concept typically relates to uncertainty, and how we can accommodate limited or imperfect knowledge of the underlying dynamics. For instance, foundational work by Kothare et al. addresses uncertainty in linear MPC with linear matrix inequalities by allowing the state transition matrices to vary in time within a convex polytope [59]. As many high-impact modern control systems are nonlinear, multimodal, and unstructured, certification still comprises a significant ongoing investigation in the literature [102, 44, 13].

1.2 Background and Relevant Works

The objective of this dissertation is to create progress in the space of certifying learning-based controllers, and to demonstrate that progress with relevant application studies.

The systems of focus are nonlinear, high-dimensional, and potentially multimodal. Certifying the behavior of such systems is an open challenge, comprising a significant area of study in contemporary literature. Our approaches partially focus on cases with highly limited subject matter expertise. We define subject matter expertise (SME) as prior knowledge of the structure of underlying dynamics, and/or prior access to data of system trajectories and behavior. Many existing safe LbC algorithms require assumptions from either or both of these spaces. In this section, we provide relevant background and literature review to contextualize our statement of contributions.

Model-Based Control

Consider the following model-based finite-time optimal control problem statement:

$$
\begin{aligned}
\underset{u}{\text{minimize}} \quad & \sum_{k=0}^{N} J(x_k, u_k) && \text{(1.1a)} \\
\text{subject to:} \quad & x_{k+1} = f^{\theta}(x_k, u_k) && \text{(1.1b)} \\
& g(x_k, u_k) \leq 0 && \text{(1.1c)} \\
& h(x_k, u_k) = 0 && \text{(1.1d)} \\
& x_0 = x(0) && \text{(1.1e)} \\
& x_N \in \mathcal{X}_{terminal} && \text{(1.1f)}
\end{aligned}
$$

We denote $x(t)$ to be the value of the state at time t, and x_k to denote the predicted value of the state x k steps after the current time t. Here, N is the final time index during prediction; $x_k \in^n$ is the predicted state vector after k steps of prediction; $u_k \in^p$ is the control input vector; $J(x_k, u_k) :^n \times^p \rightarrow$ is the stage cost function at time k; $f(x_k, u_k) :^n \times^p \rightarrow^n$ represents the system dynamics learned with parameterization θ; $g(x_k, u_k) :^n \times^p \rightarrow^m$ represents inequality constraints; and $h(x_k, u_k) :^n \times^p \rightarrow^\ell$ represents equality constraints. The constraints are evaluated with predictions of the state trajectory, and themselves may be learned functions. The set $\mathcal{X}_{terminal}$ defines feasible terminal states for the state trajectory.

We call this problem "model-based" because it uses models of the system f, g, and h to plan the control sequence. A model-free approach might look something like:

$$
\underset{\theta}{\text{minimize}} \quad V^{\pi^{\theta}}(x(t)) \qquad \text{(1.2)}
$$

Here, π^{θ} is the control policy which we learn parameterized by θ, and V^{π} is the value function given the policy π.

In the case of model-predictive control (MPC), we solve (1.1a-1.1f) at each instant int time - planning into the future - but only applying the first control input x_0^*. Then, after transitioning with the plant in the loop, we repeat this process.

Relevant Works

Stochastic optimal control has become connected to ongoing research in the burgeoning field of LbC - often referred to as reinforcement learning (RL). Here, researchers seek guarantees on safety[1] and performance when learning and/or controlling a dynamical system. For example, offline reinforcement learning focuses on synthesizing a robust policy from a fixed set of offline data. For a review of current state of the art methods in learning-based control which utilize MPC, we direct the reader to the following thorough reviews [44, 102]. For discussion of historical RL research, Garcia provide a comprehensive review [41].

Simultaneous learning and control presents a nuanced and complex challenge for a host of reasons. Safety and feasibility pose significant barriers for proper implementation of such algorithms. Moreover, balancing the exploration-exploitation tradeoff inherent to simultaneous control and model identification has presented researchers with unique challenges which form a primary focus of research in active learning. Work by Dean et al., for instance, explores safety and persistence of excitation for a learned constrained linear-quadratic regulator [28]. They show that the complexity of the underlying dynamics plays a significant component in our ability to analytically guarantee relevant certificates on model errors and controller performance.

MPC is a highly popular use case for learning-based control problems, and provides an intuitive bridge between longstanding adaptive control theory and new developments. For instance, recent work has investigated recursive feasibility for adaptive MPC controllers based on recursive least-squares [14] and set-membership parameter identification [103], although similar papers frequently possess limitations including a dependence on linear dynamical models. Rosolia and Borrelli derive recursive feasibility and performance guarantees for a learned episodic MPC controller [98]. Koller et al. also address the safety of a learned MPC controller when imperfect model knowledge and safe control exists [57].

We note that Control Lyapunov function and control barrier function [23, 35, 24] based approaches have further strengthened the connection between classical adaptive control and more modern approaches akin to popular model-based reinforcement learning (RL) problems. Recent work by Westenbrouk et al. has even explored coupling such nonlinear control methods with a policy optimization scheme [107]. Assumptions of model structure can still be critically important in these works (e.g. control-affine nonlinear systems). Such physical constraints have, however, been shown to be strong modeling foundations for a host of diverse tasks even within the visuomotor control space [19], often leveraging neural differential equations to learn control-affine relationships in data [21]. Chow and Nachum also leverage Lyapunov stability principles to obtain improved empirical results in complex data-driven domains and applications [26]. Other methods focus on safety as a challenge relevant to transfer learning, where safe behavior can be extrapolated and expanded from simpler tasks [101]. Methods in the space of RL provide idealistic safety guarantees that generally translate into improved empirical safety properties. However, any guarantees (probabilistic or robust) or safety certificates in this space are elusive and remain an open challenge.

Guarantees in RL literature are difficult to obtain since that literature often eschews subject matter expertise (SME), or direct intuition into a specific application. Generally, when RL neglects considerations to SME it becomes applicable to a much wider body of relevant

[1] We define safety as the ability of the control policy to satisfy constraints.

decision and control problems [64] that lack permeability to our intuition and expertise. Conversely, controls literature is ubiquitous in revealing how such expertise can be leveraged to yield strong and *specific* performance and safety even in adaptive and learning contexts. As previously discussed, SME in LbC methods often takes the form of model knowledge [14, 103, 23, 35, 24] and preexisting data of safe trajectories [98, 27].

The problem with these SME assumptions is that they can very easily become optimistic. Given the overarching assumption of preexisting data of safe trajectories, we have to ask "How trustworthy is our data?" This should always be called into question, especially when safety is of the utmost importance. Many LbC methods do consider noise-corrupted data [27], and distributional shifts among data and agent experience are a subject of great importance in offline reinforcement learning research [60]. The process generating the data could be flawed in many ways, the relevance of each to existing methods varies but is persistent. An example could be sampling data locally where relevant dynamics can be effectively linearized, when the system experiences highly nonlinear behavior outside of that region. Without exploiting and trusting our SME, we cannot guarantee things like this will not happen especially in safety-critical settings. By applying a resultant controller to the underlying system, it can encounter out-of-distribution (OOD) experience and adversarial attacks that a majority of existing LbC methods simply cannot consistently accommodate. Those few LbC algorithms that do make consideration to OOD experience do so using hyperparameters that are not trivial to select and validate [27], and often assume structure of the underlying dynamics [114]. These same fundamental quandaries also apply when assuming model knowledge - which is not always possible especially for multimodal control problems.

This dissertation focuses on distributionally robust optimization (DRO) as a tool to address safety concerns during LbC. Given that MPC solves a sequence of optimization programs, methods that operate within the space of robust optimization are a powerful tool to certify the performance of learning-based MPC (LMPC).

Background on DRO and LbC

In recent practice, DRO has been gaining traction as a set of methods that provide significant value to the study and solution of the LbC problem. DRO is a field of inquiry which seeks to guarantee robust solutions to optimization programs when the distributions of relevant random variables are estimated via sampling. This uncertainty can involve the objective or the constraints of the optimization program. Uncertainty in both cases can pose significant challenges if unaccounted for, leading to suboptimal and potentially unsafe performance [84]. Given that past work in the LbC space frequently considers chance constraints [14, 55, 27], incorporating a true DRO approach possesses the potential to improve our capabilities of guaranteeing safety during learning. These methods have been recently explored to address challenges of safety and performance imposed by uncertainty. For instance, Van Parys et al. address distributional uncertainty of a random exogenous disturbance process with a moment-based framework [89]. Paulson et al. also apply polynomial chaos expansions to characterize distributional parametric uncertainty in a nonlinear model-predictive control application [90].

Within the toolbox provided by DRO, Wasserstein ambiguity sets are a foremost asset. The Wasserstein metric (or "earth mover's distance") is a symmetric distance measure in the space of probability distributions. Wasserstein ambiguity sets account for distributional uncertainty in a random variable, frequently one approximated in a data-driven application. Recent exploration of Wasserstein ambiguity sets reveals they possess powerful out-of-sample safety guarantees in data-driven stochastic optimization. Namely, formulating a DRO problem with Wasserstein ambiguity sets can robustify results subject to worst-case realizations of relevant random variables - even if those realizations are not present in available data. Likewise, Wasserstein ambiguity sets make much less restrictive assumptions on the shape of underlying probability distributions [34, 40]. Expressions exist which map the quality of the empirical distribution with size parameters for the Wasserstein ambiguity set such that desired robustness characteristics are achieved without significant sacrifices to the performance of the solution [113]. Within the control context, the Wasserstein distance metric has only recently began emerging as a valuable and widespread tool. Wasserstein ambiguity sets (1) make few if any assumptions on the shape of the underlying distribution, and (2) provide out-of-sample guarantees. Both of these features relate to subject matter expertise, and broad cases where only basic model knowledge and data are needed.

For example, work by Yang et al. explores the application of Wasserstein ambiguity sets for distributionally robust control subject to broad classes of disturbance processes [109]. Similar methods have made their way to research on model-based and model-free reinforcement learning as well [53, 51, 114]. DRO has also been applied to Markov decision processes (MDPs) in a general sense [63, 6, 2, 108]. Scalability is still an open challenge in that space. Overall, while Wasserstein ambiguity sets are seeing increased application in controls research, many of their true capabilities have yet to be fully exploited.

1.3 Statement of Contributions

The problem of synthesizing optimal control with highly limited SME is immensely challenging. The objective of this dissertation is to make progress towards that objective. This progress comes from the following contributions.

We consider stochastic LMPC problems of the following general format:

$$\begin{aligned} \text{minimize} \quad & \sum_{k=0}^{N} J(x_k, u_k) && \text{(1.3a)} \\ \text{subject to:} \quad & x_{k+1} = \hat{f}(x_k, u_k, \theta_f) && \text{(1.3b)} \\ & \hat{g}(x_k, u_k, \theta_g) \leq 0 && \text{(1.3c)} \\ & x_0 = x(0) && \text{(1.3d)} \end{aligned}$$

where uncertain models $\hat{f}$ and $\hat{g}$ are learned from data. We omit equality constraints h and terminal state constraints $\mathcal{X}_{terminal}$ for simplicity for now.

In Chapter 2, we adopt a DRO method from the literature and modify it to fit within a formulation of LMPC aligned with (2.22a-2.22d). After making this connection, we apply our method to three application cases. First, we combine our safe LbC method with data-driven sequence modeling to create a scalable finite-time optimal control architecture for high-dimensional, nonlinear, and multimodal dynamical systems. We validate this approach by safely fast charging a lithium-ion battery using electrochemical information learned in a latent state space. Our controller runs in real time and respects relevant safety constraints in the fast charging problem. By solving this problem with respect to granular electrochemical information, the fast charging protocol more effectively preserves the state of health (SOH) of the cell compared to existing industry standard charging methods.

Next, we apply our method to learning-based control problems with strong limitations on available subject matter expertise. By exploring the fundamental limitations that still allow synthesis of safe control policies, we reveal insights into the LbC problem. We apply these insights to solve two application studies: (1) vision-based autonomous vehicle obstacle avoidance, and (2) extreme fast charging of lithium-ion batteries. In both cases, we start with the least allowable amount of SME, and validate whether our LbC algorithm can safely learn to control each system from scratch. Our results demonstrate powerful capabilities that extend to general adaptive control context while accommodating multimodal and nonlinear systems.

In Chapter V we leverage the DRO formulation to robustify value-based RL algorithms. By modeling a system as a constrained Markov decision process (CMDP) and using temporal-difference (TD) errors to characterize model uncertainty, we develop a deep Q-learning algorithm that translates improved safety to real-world systems leveraging the idealistic guarantees of Wasserstein DRO.

Finally we provide broader discussion of our results in Chapter VI. This includes outlining applications outside the direct LbC problems discussed in this dissertation. For example, our theory extensions in Chapter 2 can provide robust uncertainty estimates for forecasting models. We conclude with a summary of our diverse findings, and their broader relevance to the literature.

Table 1.1 outlines the PhD contributions on which this dissertation material is based.

Table 1.1: Summary of contributions during the Ph.D. work on which this dissertation is based.

Work	**Summary**
[53]	Distributionally robust LbC using ϕ-divergence, combined with sequence modeling and dimensionality reduction, application to lithium-ion battery fast charging based on a single-particle model
[52]	Journal extension of [53], utilizing Wasserstein ambiguity sets and validating on full-order electrochemical battery model fast charging
[51]	Extends existing theory on DRO using the Wasserstein distance, and explores minimal assumptions for safe learning-based control

Chapter 2

The Interface Between Robust Optimization and Optimal Control[1]

Abstract

This chapter explores formulations of the LbC problem that are almost immediately amenable to distributionally robust optimization with the Wasserstein distance. We first provide relevant background on DRO, and identify an equivalent chance-constraint reformulation from the literature. We then discuss our formulation of LbC, and show that with some minor extensions of existing theory, we can make progress translating certificates from DRO to real systems.

2.1 Stochastic Optimization with Chance Constraints

A chance constraint is a constraint within an optimization program which is only satisfied with some probability. This is typically a necessary concession when the constraint is affected by a random variable $\mathbf{R}$:

$$\mathbb{P}[h(x_k, u_k, \mathbf{R}) \leq 0] \geq 1 - \eta \tag{2.1}$$

Here, the constraint function $h(x_k, u_k, \mathbf{R})$ outputs an m-dimensional vector. In this case, the distribution $\mathbb{P}$ relates to random variable $\mathbf{R}$ with support ξ. Here, $0 \leq \eta < 1$ is the specified risk metric or our allowed probability to violate the constraint. If $\eta = 0$, we say we have a robust optimization program which must not yield *any* probability of constraint violation. In practice, especially when approximating $\mathbb{P}$ from sampling, we admit some small probability of constraint violation leading to a value of $\eta > 0$. This is frequently necessary because it allows our probabilistically robust solution to balance conservatism with performance.

[1] This chapter is adapted from previously published work [51]. ©2023 IEEE. "Safe Learning MPC With Limited Model Knowledge and Data." IEEE Transactions on Control Systems Technology (2023).

Upon utilizing an empirical approximation of $\mathbb{P}$ derived from sampling (usually denoted $\hat{\mathbb{P}}$), we admit some distributional uncertainty which can arise from only having access to a finite group of samples. The law of large numbers states that for any number of samples $\ell \to \infty$, $\hat{\mathbb{P}} \to \mathbb{P}^*$. The discrepancy from this limited sampling creates distributional uncertainty, which can affect the quality of the solution if our approximation $\hat{\mathbb{P}}$ is inaccurate [84]. Throughout the remainder of this chapter, we discuss the application of distributionally robust optimization techniques to address this distributional uncertainty.

2.2 Wasserstein Ambiguity Sets

Wasserstein Ambiguity Sets

The Wasserstein metric is defined as follows:

Definition 1 *Given two marginal probability distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ lying within the set of feasible distributions $\mathcal{P}(\xi)$, the Wasserstein distance between them is defined by*

$$\mathcal{W}(\mathbb{P}_1, \mathbb{P}_2) = \underset{\Pi}{inf}\left\{ \int_{\xi^2} ||\boldsymbol{R}_1 - \boldsymbol{R}_2||_a \Pi(d\boldsymbol{R}_1, d\boldsymbol{R}_2) \right\} \tag{2.2}$$

where Π is a joint distribution of the random variables $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$, and a denotes any norm in $\mathbb{R}^n$.

The Wasserstein metric is colloquially referred to as the "earth-movers distance." This name is rooted in the interpretation of the Wasserstein metric as the minimum cost of redistributing mass from one distribution to another via non-uniform perturbation [109]. To show why the Wasserstein distance is a valuable tool we can leverage to robustify a data-driven optimization program, we first reference the chance constraint equation (2.1), which depends on an empirical distribution $\hat{\mathbb{P}}$. Rather than solving the optimization program with respect to an imperfect snapshot of $\mathbb{P}^*$ defined by $\hat{\mathbb{P}}$, we can optimize over any probability distribution within some ambiguity set centered around our estimate $\hat{\mathbb{P}}$. The Wasserstein distance provides a formal method to define such an ambiguity set. Namely, we can optimize against the worst-case realization of $\mathbf{R}$ sourced from a set of probability distributions within specified Wasserstein radius of our empirical estimate. We define "worst-case" as the realization which yields the lowest probability of satisfying the chance constraint. This formulation can be described mathematically with the following relation:

$$\inf_{\mathbb{P} \in \mathbb{B}_\epsilon} \mathbb{P}[h(x_k, u_k, \mathbf{R}) \leq 0] \geq 1 - \eta \tag{2.3}$$

where

$$\mathbb{B}_\epsilon := \{\mathbb{P} \in \mathcal{P}(\xi) \mid \mathcal{W}(\mathbb{P}, \hat{\mathbb{P}}) \leq \epsilon\} \tag{2.4}$$

is the ambiguity set defined for a Wasserstein ball radius ϵ. Of note is the fact that (2.3) guarantees probabilistic feasibility for any probability distribution within the ambiguity set when reformulated correctly. No assumptions must be leveled on the true distribution $\mathbb{P}^*$ for these guarantees to translate under a proper reformulation.

Reformulation is necessary because the exact constraint shown in (2.3) poses an infinite dimensional nonconvex problem. Ongoing research has pursued tractable reformulations of this constraint which facilitate its real-time solution.

This dissertation largely works with a reformulation of (2.3) detailed in [33]. This reformulation accommodates vector constraint functions and requires that the function $g(x_k, u_k, \mathbf{R})$ is linear in $\mathbf{R}$, and entails a scalar convex optimization program to derive. Our algorithm is designed to exploit the linear dependence on R such that this assumption has no affect on the applicability of our approach. Importantly, the result is a conservative *convexity-preserving* approximation of (2.3). For an m-dimensional constraint function, the exact form of the ambiguity set is $\mathcal{V} = \text{conv}(\{r^{(1)}, ..., r^{(2^m)}\})$, where the vector r is sourced from the optimization component of the overall procedure. The set of constraints we find to replace the infinite dimensional DRO chance constraint are:

$$h(x_k, u_k) + r^{(j)} \leq 0, \qquad\qquad \forall\ j = 1, ..., 2^m \tag{2.5}$$

For complete and elegant discussion of this reformulation, we highly recommend the reader reference work in [33], specifically pages 5-7 of their paper. This reformulation requires some additional information, including a tractable representation of an appropriate Wasserstein ball radius.

Finally, several expressions exist for the Wasserstein ball radius ϵ which are probabilistically guaranteed to contain the true distribution with allowed probability β. We adopt the following formulation of ϵ from [113]

$$\epsilon(\ell) = C\sqrt{\frac{2}{\ell}\log\left(\frac{1}{1-\beta}\right)} \tag{2.6}$$

where ℓ is the number of data points, β is the probability the Wasserstein ball contains the true distribution, and C relates to the diameter of the support of the distribution and is obtained by solving the following scalar optimization program:

$$C \approx 2 \inf_{\alpha>0} \left\{ \frac{1}{2\alpha} \left(1 + \ln \left(\frac{1}{\ell} \sum_{k=1}^{N} e^{\alpha ||\mathbf{R}^k - \hat{\mu}||_1^2} \right) \right) \right\}^{\frac{1}{2}} \tag{2.7}$$

where the right side bounds the value of C, and $\mathbf{R}^k$ is a sample of the random variable which comprises our empirical distribution, and $\bar{\mu}$ is the sample mean of the distribution.

2.3 Equivalent Chance-Constraint Reformulation

This chapter builds upon the equivalent reformulation of (2.3) from [33]. This reformulation leverages findings from recent work by [34]. The statement of the specific reformulation in [33] indicates a requirement that the constraint function $g(x, \mathbf{R})$ is linear in x and R, respectively.

Notably, we identify a simple extension of the reformulation in [33] that allows its application to our nonlinear MPC formulation via relaxing requirement the constraint function be linear in the decision variable x.

Restatement of the Reformulation from [33]

The reformulation from [33] is stated to require the constraint function $g(x, \mathbf{R})$ to be linear in x and $\mathbf{R}$, respectively. In this chapter, we extend the reformulation to include some broader cases of constraint functions:

$$g(x, \mathbf{R}) = g_x(x) + g_R(\mathbf{R}). \tag{2.8}$$

where the functions g_x and g_R can be nonlinear in their respective arguments. We first restate the work from [33] as a reference for our extension included in subsection III.b.

Data samples $\{R^{(1)}, R^{(2)}, ..., R^{(\ell)}\}$ corresponding to random variable $\mathbf{R} \in \mathbb{R}^m$ are drawn from the true distribution $\mathbb{P}^*$. These finite samples comprise our empirical distribution $\hat{\mathbb{P}}$. The finite-ness of our empirical distribution indicates it will not perfectly match the behavior of the true distribution $\mathbb{P}^*$. This is especially true in cases with limited samples, which are relevant to the challenging case studies explored in this dissertation.

Normalizing the data lends simplicity to the derivation:

$$\vartheta^{(i)} = \Sigma^{-\frac{1}{2}}(R^{(i)} - \mu) \tag{2.9}$$

where Σ is the sample variance of the data and μ is the sample mean. This standardization transforms the data samples such that its new mean is 0, and its new variance is $I_{m \times m}$. The support of this normalized distribution is

$$\Theta = \{\vartheta \in \mathbb{R}^m \mid -\sigma_{\max}\mathbf{1}_m \leq \vartheta \leq \sigma_{\max}\mathbf{1}_m\} \tag{2.10}$$

since we have centered the normalized variable ϑ. Note that $\mathbf{1}_m$ is a column vector of ones. Let $\mathbb{Q}^*$ and $\hat{\mathbb{Q}}$ represent the true and empirical distributions of the normalized data ϑ. We construct the ambiguity set $\hat{\mathcal{Q}}$ using the "Wasserstein ball" given by (2.4), allowing us to transform the distributionally robust chance constraint (DRCC) in (2.3) to

$$\sup_{\mathbb{Q} \in \hat{\mathcal{Q}}} \mathbb{Q}[\vartheta \notin \mathcal{V}] \leq \eta \tag{2.11}$$

which says the worst case probability that normalized random variable ϑ is outside set $\mathcal{V}$ is less than η, where the supremum is taken over all distributions $\mathcal{Q}$ in ambiguity set $\hat{\mathcal{Q}}$. We wish to obtain the least conservative (i.e. tightest) set $\mathcal{V} \subseteq \mathbb{R}^m$ in order to define the desired Wasserstein uncertainty set $\mathcal{A} = \left\{a \in \mathbb{R}^m \mid a = \Sigma^{\frac{1}{2}} v + \mu,\ v \in \mathcal{V}\right\}$ such that

$$g(x_k, u_k, \mathbf{R}) \leq 0,\ \forall\, \mathbf{R} \in \mathcal{A} \tag{2.12}$$

We restrict the overall shape of the set $\mathcal{V}$ to be a hypercube, which enables computational tractability:

$$\boldsymbol{\mathcal{V}(\sigma) = \{\vartheta \in \mathbb{R}^m | -\sigma \mathbf{1}_m < \vartheta < \sigma \mathbf{1}_m\}.} \tag{2.13}$$

Now, to compute this ambiguity set without introducing unnecessary conservatism, we need to find the minimum value of the hypercube side length $\sigma \in \mathbb{R}$. The following optimization program details this problem:

$$\min_{0 \leq \sigma \leq \hat{\sigma}_{max}} \quad \sigma \tag{2.14}$$

$$\text{subject to:} \quad \sup_{\mathbb{Q} \in \hat{\mathcal{Q}}} \mathbb{Q}[\tilde{\vartheta} \notin \mathcal{V}(\sigma)] \leq \eta \tag{2.15}$$

Here, we select $\hat{\sigma}_{max}$ using *a priori* information about the specific problem context.

The derivation in [33] provides a worst-case probability formulation, summarized by the following Lemma:

Lemma 1 (Lemma 2 of [33])

$$\sup_{\mathbb{Q} \in \hat{\mathcal{Q}}} \mathbb{Q}[\tilde{\vartheta} \notin \mathcal{V}(\sigma)] = \inf_{\lambda \geq 0} \left\{ \lambda \epsilon(\ell) + \frac{1}{\ell} \sum_{j=1}^{\ell} \left(1 - \lambda \left(\sigma - ||\vartheta^{(j)}||_\infty \right)^+ \right)^+ \right\} \tag{2.16}$$

where $(x)^+ = \max(x, 0)$.

We defer to [33] for the proof of this finding. Their result entails that (2.16) can be reformulated as

$$\min_{0 \leq \lambda, 0 \leq \sigma \leq \hat{\sigma}_{max}} \sigma \qquad \text{subject to:} \quad h(\sigma, \lambda) \leq \eta \leq \sigma_{max} \tag{2.17}$$

where

$$h(\sigma, \lambda) = \lambda \epsilon(\ell) + \frac{1}{\ell} \sum_{j=1}^{\ell} \left(1 - \lambda (\sigma - ||\vartheta^{(j)}||_\infty)^+ \right)^+ \tag{2.18}$$

The result of this optimization program is the value of σ, which is used to reformulate the chance constraints via convex approximation. For a convex approximation of the constraint function in (2.3), the hypercube $\mathcal{V}(\sigma)$ becomes the convex hull of its vertices. If for example $m = 1$ (i.e. the random variable is 1-dimensional), then $\mathcal{V}(\sigma) = (-\sigma, \sigma)$ – an open interval. The offset $r^{(j)}$ is calculated from:

$$r^{(1)} = \Sigma^{\frac{1}{2}} \mathbf{1}_m \sigma + \mu \tag{2.19}$$

$$r^{(2)} = \Sigma^{\frac{1}{2}} \mathbf{1}_m (-\sigma) + \mu \tag{2.20}$$

In the two dimensional case, this yields the ambiguity set $\mathcal{A} = \text{conv}(\{\pm\sigma, \pm\sigma\})$ where $\text{conv}(\{\cdots\})$ represents the convex hull of points $\{\cdots\}$. For an m-dimensional constraint function, the exact form of the ambiguity set is $\mathcal{V} = \text{conv}(\{r^{(1)}, ..., r^{(2^m)}\})$. In each case, the ambiguity set is a hypercube, and the change of signs is the method by which we enumerate across that hypercube's vertices with the following constraints:

$$g(x) + r^{(j)} \leq 0, \qquad\qquad \forall\ j = 1, ..., 2^m \tag{2.21}$$

Algorithm 1 details the method used to compute the offset σ.

Algorithm 1 Computation of σ

```
0: Initialize $\underline{\sigma} = 0, \bar{\sigma} = \sigma_{max}$
   while $\bar{\sigma} - \underline{\sigma} >$ tolerance do
      $\sigma = \frac{\bar{\sigma}+\underline{\sigma}}{2}$
      $[\lambda, h^*(\sigma, \lambda)] = \text{minimize}(\sigma,\ \lambda_{lb},\ \lambda_{ub},\ \epsilon,\ \theta)$
      if $h^*(\sigma, \lambda) > \eta$ then
         $\underline{\sigma} = \sigma$
      else
         $\bar{\sigma} = \sigma$
      end if
   end while
   $\sigma = \bar{\sigma}$
```

2.4 The Interface

In Chapter 1, we defined the general format of a model-predictive control program. We restate that definition here for reference:

$$\begin{aligned} \text{minimize} \quad & \sum_{k=0}^{N} J(x_k, u_k) && (2.22a) \\ \text{subject to:} \quad & x_{k+1} = \hat{f}(x_k, u_k, \theta_f) && (2.22b) \\ & \hat{g}(x_k, u_k, \theta_g) \leq -\mathcal{G} && (2.22c) \\ & x_0 = x(0) && (2.22d) \end{aligned}$$

Shuffling equation (2.22c) reveals stark similarity to (2.8) assuming the stochasticity is represented by the process noise term $\mathcal{G}$:

$$\hat{g}(x_k, u_k, \theta_g) + \mathcal{G} \leq 0 \tag{2.23}$$

This similarity raises the question of "*How can we characterize model uncertainty as a random variable that aligns with the DRO theory?*" Of note is that the function $\hat{g}$ is almost surely nonlinear, which ostensible conflicts with the setup from [33]. In the next section, we provide simple arguments to extend the applicability of the DRO reformulation to chance constraints of the form (2.8).

2.5 Extending the Reformulation

Duan et al. utilize the findings of [34] in presenting their convex reformulation. Critically, we identify that the fundamental theory presented by [34] allows applying the identical reformulation to cases where the constraint function takes the form

$$g(x, \mathbf{R}) = g_x(x) + g_R(\mathbf{R}). \tag{2.24}$$

wherein g_x and g_R may be nonlinear functions. Critically, there must not be any interdependence between x and $\mathbf{R}$.

Remark 1 *The linear separability of x and R poses a DRO program that can be thought of like solving*

$$\begin{aligned} \underset{z \in \mathbb{R}}{\mathit{minimize}} \quad & -z && \text{(2.25a)} \\ \mathit{s.\ to:} \quad & \underset{\mathbb{P} \in \mathbb{B}_\epsilon}{\mathit{inf}} \mathbb{P}\left[z - R \leq 0\right] \geq 1 - \eta && \text{(2.25b)} \end{aligned}$$

for offset z from the second stage nominal constraint boundary.

This chapter presents a modified lemma for the applicability of the previously stated reformulation first presented by [33].

Lemma 2 *If the function g satisfies*

$$g(x, \boldsymbol{R}) = g_x(x) + g_R(\boldsymbol{R}). \tag{2.26}$$

then constraints of the following form:

$$\underset{\mathbb{P} \in \mathbb{B}_\epsilon}{\mathit{inf}}\ \mathbb{P}[g(x, \boldsymbol{R}) \leq 0] \geq 1 - \eta \tag{2.27}$$

can be reformulated into the convex approximation

$$g_x(x) + r^{(j)} \leq 0, \qquad\qquad \forall\ j = 1, ..., 2^m \tag{2.28}$$

using the relations in (2.16-2.17), where $r = \Sigma^{\frac{1}{2}} \boldsymbol{1}_m \sigma + \mu$.

Proof 1 *We start by defining auxiliary variables in the constraint function. Consider that, without loss of generality, nonlinear functions of $\boldsymbol{R}$ can themselves be considered the random variable in question:*

$$\tilde{\boldsymbol{R}} = g_R(\boldsymbol{R}) \tag{2.29}$$

where $\tilde{\boldsymbol{R}}$ is the new model of the stochasticity. This gives

$$g(x, \boldsymbol{R}) = g_x(x) + \tilde{\boldsymbol{R}} \tag{2.30}$$

Now, we create a dummy auxiliary decision variable $\tilde{x}$ in the same manner:

$$\tilde{g}(\tilde{x}, \tilde{\boldsymbol{R}}) = \tilde{x} + \tilde{\boldsymbol{R}} \tag{2.31}$$

forming a function $\tilde{g}$ which is trivially linear in $\tilde{x}$ and $\tilde{\boldsymbol{R}}$, where

$$\tilde{x} = g_x(x). \tag{2.32}$$

This equality constraint (2.32) now shows up in the overall optimization program. However, the DRCC reformulation only poses conditions on the constraint function in question (namely $\tilde{g}(\tilde{x}, \tilde{\boldsymbol{R}})$*). We have transformed the distributionally robust chance constraint into*

$$\underset{\mathbb{P}\in\mathbb{B}_\epsilon}{inf}\mathbb{P}\left[\tilde{g}(\tilde{x}, \tilde{\boldsymbol{R}}) \leq 0\right] \geq 1-\eta \tag{2.33}$$

which is now linear in $\tilde{x}$ *and* $\tilde{\boldsymbol{R}}$*. Following procedure from [34], we suppress dependence on* x *(or* $\tilde{x}$*) for simplicity, leading to* $\ell(\tilde{\boldsymbol{R}}) = \tilde{g}(\tilde{x}, \tilde{\boldsymbol{R}})$ *[34, 33]:*

$$\underset{\mathbb{P}\in\mathbb{B}_\epsilon}{inf}\mathbb{P}\left[\ell(\tilde{\boldsymbol{R}}) \leq 0\right] \geq 1-\eta. \tag{2.34}$$

The remainder of the proof is identical to the Appendix in [33], leading to the convex approximation:

$$g_x(x) + r^{(j)} \leq 0, \qquad\qquad \forall\ j = 1, ..., 2^m \tag{2.35}$$

Beyond exploiting the linear presence of $\tilde{x}$ in the constraint function, suppressing dependence on decision variables is possible and helpful for the following reasons. The overall process of solving an optimization program with a DRCC is characterized by a two stage stochastic optimization problem. Here, (2.34) is the first stage problem that we solve using the equivalent reformulation. Esfahani and Kuhn show in Section 5.3 of their paper that, without loss of generality, the solution in the second stage (i.e. the overall optimization program) is unaffected by suppressing dependence of ℓ on decision variables in the first stage. Additionally, the decision-independent loss function $\ell(\tilde{\mathbf{R}})$ can trivially be expressed as a pointwise maximum of elementary measurable functions, as required by Section 4 of [34].

In practice, the dummy decision variable $\tilde{x}$ will not come into play during any stage of solution. After solving the first stage problem, we can reverse the substitution in the remaining optimization to avoid an equality constraint with poor computational tractability.

2.6 Conclusion

This chapter investigates extensions of DRO theory that unlock new applications in the LbC problem space. Our extension allows Wasserstein ambiguity sets to apply in cases where nonlinear and nonconvex functions and constraints are of interest. As long as problems of interest can be decomposed in a specific manner, we can apply Wasserstein DRO to quantify uncertainty and conduct decision making in the face of uncertainty with certificates on performance and feasibility. Critically, the decomposition format relates to the modeling of stochasticity in the problem in a manner that is widely generalizable and computationally tractable. In the following chapters, we develop algorithmic architectures across the LbC space that leverage this new DRO theory to certify the performance and safety of LbC controllers.

Chapter 3

Real-Time Electrochemical Fast Charging Using Distributionally Robust Surrogate Optimal Control[1]

Abstract

This chapter develops a data-driven surrogate modeling architecture for tractably solving optimal control and offline reinforcement learning problems for high-dimensional systems. Leveraging techniques from sequence modeling, we develop surrogate models that predict the loss function and timeseries of constraint functions in a finite-time optimal control problem from an embedding of the initial state and timeseries of control inputs. DRO theory from Chapter 2 ensures the learned surrogate models respect distributional shifts, and satisfy constraints with high probability. We validate our method by synthesizing an extreme fast-charging protocol for a lithium-ion battery directly from its electrochemistry. The resulting policy preserves the state-of-health of the battery cell more effectively than industry standards.

3.1 Introduction

This chapter presents a novel model-based data-driven method for robust optimal control and offline reinforcement learning of high-dimensional dynamical systems.

Optimal control faces unique challenges related to guaranteeing optimality and computational efficiency [56]. These challenges are generally exacerbated when the dynamical system in question is a high-dimensional system, a classification based on the cardinality of state variables n ($x \in \mathbb{R}^n$) being high (i.e. $n > 10^2$ or 10^3). Learning based methods help tackle

[1] This chapter is adapted from previously published work [52]. ©2022 IEEE. Reprinted, with permission, from Aaron Kandel, Saehong Park, and Scott J. Moura. "Distributionally Robust Surrogate Optimal Control for High-Dimensional Systems." IEEE Transactions on Control Systems Technology (2022).

dimensionality through data compression [37], but historically can struggle to guarantee feasible solutions especially in safety-critical applications.

In this chapter, we introduce a simple algorithmic framework which utilizes (i) deep sequence modeling, (ii) dimensionality reduction, and (iii) distributionally robust optimization (DRO) to obtain computationally tractable optimal control for high-dimensional nonlinear optimal control problems. This contribution is important, considering that the majority of real-life dynamical systems (i.e. heat transfer, fluid dynamics, etc...) are inherently high-dimensional. This is partially a result of their representation with partial differential equations (PDEs), which when solved numerically are frequently represented with numerous state variables [11]. Often, model-order reduction is applied to generate a "control-oriented" dynamical model when the true underlying system is complex and high-dimensional [54]. However, reductions can refute our ability to observe fundamental insights from our optimal control solution [43]. Reductions can also compromise the capability of maximizing the performance of the control policy.

Relevant literature presents a host of methods for high-dimensional optimal control. Besides use of specialized and case-specific heuristics, these generally include (i) control vector parameterization (CVP), (ii) reinforcement learning (RL) and approximate dynamic programming (ADP), (iii) pseudospectral optimal control (POC), and (iv) variational calculus and Pontryagin methods (PM) [79, 16].

CVP is a powerful tool due to its simplicity (see e.g. [75]). In CVP, the control input is represented and manipulated in reduced form. For instance, the control input can be defined using a zero-order hold over long timesteps, or as a polynomial whose coefficients we optimize. The advantage of CVP is it reduces the number of decision variables in the optimization program. For instance, CVP has been used to reduce the complexity of highly non-convex but relatively small-scale problems [99]. Nonetheless, for high-dimensional control CVP has been shown to yield useful results [75, 100]. CVP simplifies the problem, which compromises optimality. Furthermore, CVP only addresses computational cost from the cardinality of the control input. Other sources of computational expense (i.e. simulation, numerical optimization) can still prohibit tractable solution of the control problem.

ADP leverages black-box function approximations to enable policy learning beyond the spatial/memory limitations of tabular dynamic programming methods [**Bertsekas01**, 9]. The three biggest shortcomings of ADP relate to safety, optimality, and computation. In this context, safety refers to the ability of a learned control policy to satisfy relevant constraints. ADP and other model-free RL methods often require constraints to be encoded as auxiliary penalties to the objective/reward function [41]. Weighting these penalties requires tuning the objective function carefully. More importantly, however, model-free and model-based RL algorithms must learn behavior through exploration. For constrained problems, this can implicitly require *violating* constraints throughout online learning [96]. Moreover, RL can lose guarantees of converging to an optimal policy when the problem is complex (i.e. *not* linear-quadratic). Furthermore, for high-dimensional nonlinear problems, ADP and model-free RL methods can require a large number of iterations to converge to a high-performing control policy [**Bertsekas01**]. At a high level, many of these challenges are just as relevant for model-based RL methods. These challenges are exacerbated when learning policies from fixed, offline datasets. Recent research in offline reinforcement learning literature has provided modified algorithms that address these challenging questions while also proving to be amenable to high-dimensional control [83, 60]. In particular, offline RL methods address distributional shifts between the training data and data encountered from novel experience. For high-dimensional systems, these shifts become more likely, and can hamper optimality and feasibility.=

Surrogate optimization models typically map decision variables to an approximation of the true objective function. Historically, surrogate optimization has been popular in aerospace applications, where complex high-dimensional physics-based models form the basis for design and analysis [69, 94]. The surrogate functions are fit using samples from the original objective, which is typically expensive to evaluate. The most popular approach is efficient global optimization (EGO). EGO is an adaptive sampling regime which is guaranteed to yield a surrogate optimization model with bounded modeling errors under certain conditions [48]. EGO can work for simple control problems [74], however for high-dimensional problems the parameterization of the surrogate model and the required sampling depth can become intractable. Surrogate models have also been used to approximate state-transition dynamics for control [22, 82]. This application underpins modern research activity on model-based reinforcement learning [49, 61]. For high-dimensional systems, such models are ostensibly impractical again due to the expansive parameterizations which would be required to represent state-transition dynamics. Use of embeddings, latent spaces, and dimensionality reduction can ease computational demands, but add additional approximations that have yet to be addressed in a certified way [76].

Table 3.1 shows a summary of these algorithms. Existing methods possess unique strengths in solving high-dimensional optimal control problems, but there is area for further development. The objective of this chapter is to present a general, data-driven algorithmic framework applicable to high-dimensional systems which addresses the critical, unanswered question of safety and feasibility. First, we define neural network surrogates which map a reduced state representation and a finite time series of control inputs to an approximation of the objective function. Instead of constraint penalties, we develop auxiliary surrogate models which predict time series of the constraint functions using the same reduced input data. Our method is then, by definition, a model-based RL approach. For optimal control problems with a short time horizon, we obtain approximate solutions by optimizing around the models a single time. However, for optimal control problems on longer time horizons, we apply these surrogates within a receding horizon control framework. Via a sequence-modeling method, we absorb the dynamics of the state transitions into the prediction of the surrogate models, eliminating modeling drift.

By leveraging surrogate models, we introduce modeling errors. While objective uncertainty may affect optimality, uncertainty in the constraint functions can mean the difference between safe control and critically unsafe behavior. Therefore, this work accommodates uncertainty in the constraints via distributionally robust chance constraints (DRCC). These chance constraints encode distributions of modeling errors computed from testing data. We

Table 3.1: Algorithms for High-Dimensional Control. A * indicates the approach can be applied given a fixed, offline dataset with no model knowledge.

Algorithm	**Challenges**
CVP	optimality, computation, requires model knowledge
RL*	safety, optimality, computation
POC	requires model knowledge, proprietary software
PM	numerical instability, computation, requires model knowledge

apply Wasserstein ambiguity sets to strengthen robustness by optimizing with respect to worst-case modeling errors sourced from a family of distributions within some Wasserstein distance of the empirical distribution. The Wasserstein measure is distinguished from other probabilistic distances (i.e. moment-based methods of ϕ-divergence [47]) in that it is symmetric between two distributions, makes no assumptions on the shape of the distributions, and importantly provides an “out-of-sample” safety guarantee [34]. When used for DRCCs, we can probabilistically guarantee adherence to constraints even when our surrogate models experience distributional shifts relative to the training data.

To evaluate the efficacy of the algorithm, we solve the safe-fast charging problem for a high-dimensional lithium-ion battery model at low temperatures. Lithium-ion battery fast charging is currently an active research area in the energy systems and control literature. Significant challenges can arise in this problem from using reduced-order models [92]. If we leverage full order electrochemical battery models, then we benefit from more granular electrochemical information to safely operate the cell farther towards the boundary of its safe operating conditions [52]. This increases the performance of the resulting charge/discharge cycle, but requires that we strictly adhere to safety constraints. Violation of some electrochemical constraints leads to rapid aging and potential catastrophic cell failure. Consequently, the fast charging problem presents a relevant safety-critical challenge to our proposed algorithm. Historically, fast charging has been explored with reduced order models due to the nonlinearity and computational complexity of simulating the full-order dynamics [95, 17, 39]. By demonstrating that our surrogate optimal control algorithm can yield fast and feasible charge cycles based on the full-order electrochemical model in real time, we validate its use for high-dimensional nonlinear optimal control problems.

The results in this chapter comprise a significant extension of previous work in [53]. These extensions include (i) a comprehensive novel case study using a full-order electrochemical battery model, including a computational comparison to control using a reduced order model, and (ii) the use of Wasserstein ambiguity sets instead of more limited ϕ-divergence.

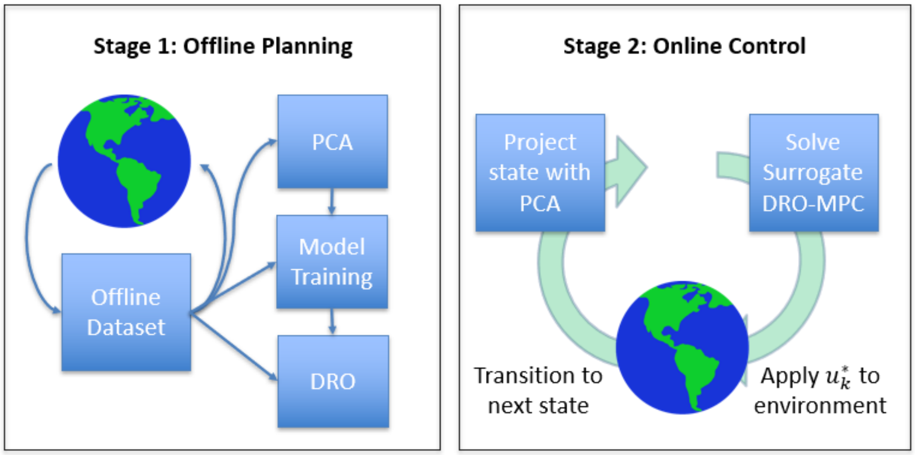

Figure 3.1: Block diagram detailing progression and flow of our proposed optimal control method. u_k^* is the optimal open-loop control input obtained from MPC with the surrogate models.

3.2 Problem Formulation

Optimal Control Problem Formulation

This work considers the following optimal control problem statement, cast in discrete time:

$$\begin{aligned} \min \quad & \sum_{k=0}^{N} J(x_k, u_k) && \text{(3.1a)} \\ \text{subject to:} \quad & x_{k+1} = f(x_k, u_k) && \text{(3.1b)} \\ & g(x_k, u_k) \leq 0 && \text{(3.1c)} \\ & h(x_k, u_k) = 0 && \text{(3.1d)} \\ & x_0 = x(0) && \text{(3.1e)} \end{aligned}$$

where k is the current time and N is the final time; $x_k \in^n$ is the state vector at time k; $u_k \in^p$ is the control input vector; $J(x_k, u_k) :^n \times^p \rightarrow$ is the stage cost function at time k; $f(x_k, u_k) :^n \times^p \rightarrow^n$ represents the system dynamics; $g(x_k, u_k) :^n \times^p \rightarrow^m$ represents inequality constraints; and $h(x_k, u_k) :^n \times^p \rightarrow^\ell$ represents equality constraints. In this chapter, we are particularly interested in problems where the cardinality of x is high, i.e. $n > 10^2, 10^3$, or more.

Our objective is to simplify the computation required to solve (3.1a)-(3.1e) when the model is high-dimensional. Figure 3.1 shows a block diagram of our method. In each of the following sections II.B and II.C, we discuss the components represented in this diagram.

Offline Dataset

Our method leverages a fixed, offline dataset composed of state trajectories matched with control input sequences. Typically, training data for surrogate optimization models is generated via a host of methods. For instance, one popular method in the literature is Latin hypercube sampling (LHS) [94]. In another method, EGO, sampling from the original objective function is organized and adaptive to the real-time evolution of modeling errors [48]. We train our surrogate models using data obtained from random, offline, parallelizable simulations of the original high-dimensional dynamical model. However, any dataset could be used to learn these surrogate models. For example, such data could come from physical experiments, an existing suboptimal controller, etc... In considering how such a dataset can be generated, the distributional shift problem becomes highly relevant. We want to minimize the degree to which real-time control data will deviate from the distribution of training data. How this question is answered is highly dependent on the specific application. Importantly, our framework is data-driven and does not require explicit model knowledge. This is differentiated from many existing methods (incl. CVP, psuedospectral optimal control).

Model Formulation and Training

Within the context of optimal control, surrogate models have been applied to represent state transition dynamics directly [22, 82]. Direct approximation of state transition dynamics is not ideal for high-dimensional dynamical systems, where the large cardinality of state variables would require function approximators with intractable parameterizations. This work proposes using a modified finite-time surrogate modeling approach which takes the following form:

$$\min \quad \mathcal{J}(x_0, U) \tag{3.2a}$$

$$\text{subject to:} \quad \mathcal{G}_i(x_0, U) \leq 0 \,\forall\, i = 1, \cdots, m \tag{3.2b}$$

The surrogate model $\mathcal{J}$ absorbs the state transition dynamics by mapping the initial state x_0 and time series of control inputs $U = [u(0), \cdots, u(N)]$ directly to an approximation of the objective function given in (3.1a). In set notation $\mathcal{J}(\cdot,\cdot) :^n \times^{p\times(N+1)} \to$. Likewise, the surrogate constraint functions $\mathcal{G}_i :^n \times^{p\times(N+1)} \to^{(N+1)}$ take the same inputs and predict as output a time series of the relevant constraint function values for each of $i = 1, ..., m$ inequality constraints. Importantly, the constraint surrogates only model the most relevant information in time series format. State variables that do not pertain to constraints in the optimization problem are disregarded by the surrogate models. Furthermore, by outputting an entire time series, we avoid the possibility of modeling drift inherent to a surrogate which predicts individual state transitions across a single time step [49].

For a model predictive control application, the optimal control problem in (3.2a)-(3.2b) becomes:

$$\min \quad \mathcal{J}(x_k, U_{k:k+N}) \tag{3.3a}$$

$$\text{subject to:} \quad \mathcal{G}_i(x_k, U_{k:k+N}) \leq 0 \,\forall\, i = 1, \cdots, m \tag{3.3b}$$

At $k = 0$, the initial state becomes the current state, and the control input time series $U_{k:k+N} = [u_k, \cdots, u(k+N)]$ starts at the current state and evolves over a horizon of N time steps into the future. Note we are re-using N here to indicate the control horizon length relative to the current time step, as opposed to the global time horizon length in (3.2a)-(3.2b). After solving this reduced optimization program, we apply the first control input to the plant, simulating one step forward and then repeating the overall process.

Perhaps the most important transformation we make relates to reducing the state with dimensionality reduction techniques. Our case study specifically uses principal component analysis (PCA) to project the state onto a reduced basis. So in fact, the optimization program becomes:

$$\begin{aligned} \min \quad & \mathcal{J}(\tilde{x}_k, U_{k:k+N}) && (3.4a) \\ \text{subject to:} \quad & \mathcal{G}_i(\tilde{x}_k, U_{k:k+N}) \leq 0 \, \forall \, i = 1, \cdots, m && (3.4b) \end{aligned}$$

where $\tilde{x}_k$ is a reduced representation of the dynamical state. Note the control is not included with state reduction, because its approximation could corrupt the input signal and negatively impact performance.

Dimensionality Reduction With Principal Component Analysis

High-dimensional, nonlinear, and multimodal state spaces are often compressed by learning an appropriate embedding space [111]. This technique falls under dimensionality reduction, which has longstanding presence in the literature. Perhaps the most basic approach relates to principal component analysis (PCA).

We can reduce the dimensionality of state vectors using PCA, provided we have some historical data samples of past states. Consider the arguments of $\mathcal{J}$ and $\mathcal{G}_i$, $(x_0, U) \in^n \times^{p\times(N+1)}$. We are interested in $n > 10^2, 10^3$ whereas $p(N+1) \sim 10^1$.

Suppose we have M training data samples for the state x, represented as matrix $X \in \mathbb{R}^{n\times M}$. Consider a so-called "principal component" which can be expressed as:

$$V = w^T X \tag{3.5}$$

where $w \in \mathbb{R}^{n\times 1}$ is a vector of weights, $V \in \mathbb{R}^{1\times M}$ is an arbitrary principal component. If we consider X as a random matrix, then we seek to choose w to maximize the variance of V

$$\text{var}(V) = w^T X X^T w \tag{3.6}$$

We then formulate the following optimization problem while constraining w to have unit length:

$$\begin{aligned} \max_{w} \quad & w^T X X^T w && (3.7) \\ \text{subject to} \quad & w^T w = 1 && (3.8) \end{aligned}$$

which yields the first principal component. This method can be extended to compute multiple principal components, and project the original data onto a reduced basis that maximizes variance [15], thus reducing $x_0 \in \mathbb{R}^n$ to a vector of dimension q, where $q << n$.

Note: Facilitating Optimization

This chapter's approach requires that we optimize around the neural network architecture. This architecture shares similar nonconvexity with the original expensive-to-evaluate objective function [48]. Past work has explored the use of convex neural architectures to facilitate this format of optimization [22]. However, input-convex neural networks can compromise the universal function approximator properties of general neural networks [5].

The use of neural function approximation allows us to exploit analytic expressions for the function input-output gradient, as done in [22]. For instance, for a single hidden layer neural network $f(x) = \sigma_{out}(W_2\sigma_{hidden}(W_1x + b_1) + b_2)$ where $\sigma_{out}(x) = x$, the Jacobian is given by:

$$\text{Jac}(f(x))_{ij} = W_1(:, i)^T W_2(j, :)\sigma'_{hidden}(W_1x + b_1) \tag{3.9}$$

Were we to solve the original optimal control problem with no surrogates, any gradients would be computed numerically via finite differences, which is highly inefficient. Numerical gradient calculations scale on the order of $\mathcal{O}[n^3]$ for a function $f :^n \rightarrow$, which would add significant computational complexity [15]. By supplying the numerical optimization solver with analytic expressions for the input-output gradients of relevant surrogate models, we avoid expensive numerical gradient calculations. Consequently, analytic gradients provide a fruitful opportunity to reduce computational complexity.

In this chapter, we evaluate and compare two optimization schemes. First, we use numerical optimization with specified analytical gradients. We compare this approach to a sample-based random search. Past work has shown for some applications that random search can provide high-performing results relative to more conventional optimization approaches [72]. In this chapter, we specifically apply a $(1 + \lambda)$ evolutionary strategy algorithm to solve the receding horizon control problem. Section 3.4 of this chapter provides more details of this comparison. Overall, however, random search outperformed the gradient-based approach.

Model Uncertainty

Surrogate models are inherently imperfect. Uncertainties are expected in approximations of both the objective and constraint functions and, if unaccounted for, these uncertainties can affect the optimality and feasibility of the final solution [15]. Likewise, nearly every process for dimensionality reduction will introduce approximation errors.

In this chapter, we leverage the DRO theory from Chapter 2 to robustify the control architecture to its sources of uncertainty. The residuals of the surrogate models, trained on low-dimensional embeddings of the initial state, and trained from a limited offline dataset, can be modeled as random variables - turning the problem into a stochastic optimization program amenable to relevant DRO tools.

$$\begin{aligned} \min \quad & \mathcal{J}(\tilde{x}_k, U_{k:k+N}) && (3.10a) \\ \text{subject to:} \quad & \mathcal{G}_i(\tilde{x}_k, U_{k:k+N}) + \mathbf{q} \leq 0 \,\forall\, i = 1, \cdots, m && (3.10b) \end{aligned}$$

where $\mathbf{q}$ is a random variable representing the random modeling residual of the function $\mathcal{G}$, and is drawn from empirical distribution $\hat{\mathbb{Q}}$ computed from the validation dataset. We describe the final DRO problem statement in the following section III.III.

3.3 Case Study

Next we present a case study to validate and characterize the performance of the proposed algorithmic architecture. Our case study is safe-fast charging of a lithium ion battery at low temperatures. Lithium-ion battery fast charging is a highly relevant safety-critical application which possesses a rich and diverse history of research. It also presents a prototypical high-dimensional optimal control problem, in that complex electrochemical battery models are described with hundreds or even thousands of state variables. While reduced-order equivalent circuit models address these dimensionality problems, the granular electrochemical information afforded by the full order models allows us to confidently take the battery closer to the safe operating envelope boundary. This grants us the ability to exploit electrochemistry to improve charging performance [53].

Low temperatures complicate the fast charging problem problem, as they sensitize many of the complex electrochemical dynamics. Specifically, the cell side-reaction overpotential constraint, which dictates the rate of lithium plating and cell degradation, can be much more readily violated at low temperatures [77]. Thus, the optimal control problem possesses many opportunities for constraint violation, which allows us to properly validate the efficacy of the proposed DRO framework.

Our case study is structured precisely as follows, where we solve a high-dimensional fast charging problem using the full-order Doyle-Fuller-Newman model (DFN) [104]. We also compare computation between the full order problem and one included in past work [53] based on a moderately reduced single particle model. We ensure comparison of our results with and without the added DRO framework, in order to validate its relative value and contributions to the safety of our algorithmic architecture.

Electrochemical Battery Model

High fidelity battery modeling provides insights on performance, without requiring one to build and experimentally test the cell. The mathematical model formulated in this dissertation's appendix is the Doyle-Fuller-Newman battery model which comes from porous electrode theory, where Li-ions intercalate/deintercalate into porous spherical particles in the negative and positive electrodes. During charging, the Li-ions in the positive electrode deintercalate, dissolve into the electrolyte, and then migate and diffuse to the negative electrode by passing through the separator. Critically, this full-order electrochemical model reveals insights into the the mechanisms within the battery cell which allow us to take the battery farther towards the limit of its safe operating conditions. By exploiting electrochemistry, we can calculate and apply faster, higher-performing charging cycles.

Table 3.2: Relevant Model Values

State Variable	**Description**	**Units**
SOC	State of Charge	-
η_S	Side-Reaction Overpotential	Volts
T	Cell Temperature	K
I	Input Current	C-Rate

While we relegate the model equations to this dissertation's appendix, we include some basic, useful information in this section in Table 3.2 for reference in discussing this chapter's problem formulation and results.

Optimal Control Problem Statement

For the DFN fast charging case study, we adopt the following optimal control problem statement within the framework of receding horizon control:

$$\min \sum_{k=t}^{t+N} (SOC_k - SOC_{targ})^2 \tag{3.11a}$$

$$\text{Subject to:} \tag{3.11b}$$

$$\text{Dynamics} \tag{3.11c}$$

$$\eta_s \geq 0 \tag{3.11d}$$

$$T \leq T_{max} \tag{3.11e}$$

$$0 \leq I \leq 2.5 \tag{3.11f}$$

The key constraints are that the side reaction overpotential stays positive, and the temperature does not exceed a maximum allowed threshold. The overpotential constraint is the most critical barrier to prevent rapid aging and potential catastrophic failure of the cell. If overpotential becomes negative, lithium metal begins to plate on the anode. This phenomena reduces the capacity of the cell and leads directly to cell failure. The temperature constraints provide indirect ways to avoid rapid aging, as the cell dynamics become more sensitive at temperature extremes.

We adapt this formulation using the distributionally robust surrogate modeling approach to yield:

$$\min \mathcal{J}(x_{u,k}) \tag{3.12a}$$

$$\text{subject to:} \tag{3.12b}$$

$$\mathcal{G}_{\eta_s}(x_{u,k}) \geq q_{\eta_s} \tag{3.12c}$$

$$\mathcal{G}_T(x_{u,k}) \leq T_{max} - q_T \tag{3.12d}$$

$$0 \leq I \leq 2.5 \tag{3.12e}$$

Since we are exploring fast charging at low temperatures, the temperature constraint is unlikely to be violated. We omit this constraint, for simplicity, but it can be added back in practice.

Results

Table 3.3 details several important hyperparameters for this case study. We consider a nickel-manganese-cobalt battery cell. The initial electrochemical states correspond to equilibrium with a voltage of $V = 3.25$ Volts. The cell is at the same uniform temperature as the ambient temperature of $T_{amb} = 281$ Kelvin. We simulate 150 random charging trajectories to generate the requisite training data to fit the surrogate models. Each trajectory was either terminated if (1) the target SOC of 0.7 was reached, or (2) the episode end time of 55 minutes was reached. The maximum allowed C-rate for these simulations is 2.5C, where the C-rate for a lithium-ion battery is the parameter describing how much input current would be needed to charge the battery from empty to full in exactly 1 hour. A typical target SOC for electric vehicle applications is 0.8 or higher. Software implementations of the DFN model lose some numerical stability when applying high C-rates at higher SOCs. To ensure we can continue utilizing a maximum C-rate of 2.5, we instead choose to set a slightly lower target SOC of 0.7 in our case study. Our algorithm can, however, be adapted to charge a battery cell to a higher SOC.

Using principal component analysis on the state trajectories, we decide to project the state vector $x \in \mathbb{R}^{2687} \to \mathbb{R}^{40}$. This decision is motivated by the explained variance of the data, plotted in Fig. 3.2. Figure 3.2 shows that the first 40 principal components of the state vector data explain 99.74% of the variance in the dataset.

The surrogate models are feed-forward neural networks each with two hidden layers, each with 10 neurons and sigmoid activation functions. The distribution of test data residuals for side reaction overpotential constraint function $\mathcal{G}_{\eta_s}$ are shown in Figure 3.3. This distribution is centered around zero with tight variance, although the tails of the distribution indicate that large residuals can occur with non-zero probability. If unaccounted for in the control algorithm, violation of the overpotential constraint by, for example 0.14 volts, would cause accelerated cell aging and could potentially sow the beginnings of a catastrophic failure. Based on the testing data from model training (using an 80/20 split), the DRO offset computed using a Wasserstein ambiguity set is $r = 0.0200066$. Given the specified chance constraint parameters, this offset is expected to yield desired safety characteristics.

We implemented our algorithm using a $(1 + \lambda)$ evolutionary strategy for optimization, depending on 25000 mutants per iteration and 12 total iterations. Cross-entropy random search also presents a useful alternative for numerical optimization [12]. As a point of comparison, we implemented a numerical optimization scheme based on Matlab's `fmincon` solver, which we supplied with analytical gradient expressions for each function approximator.

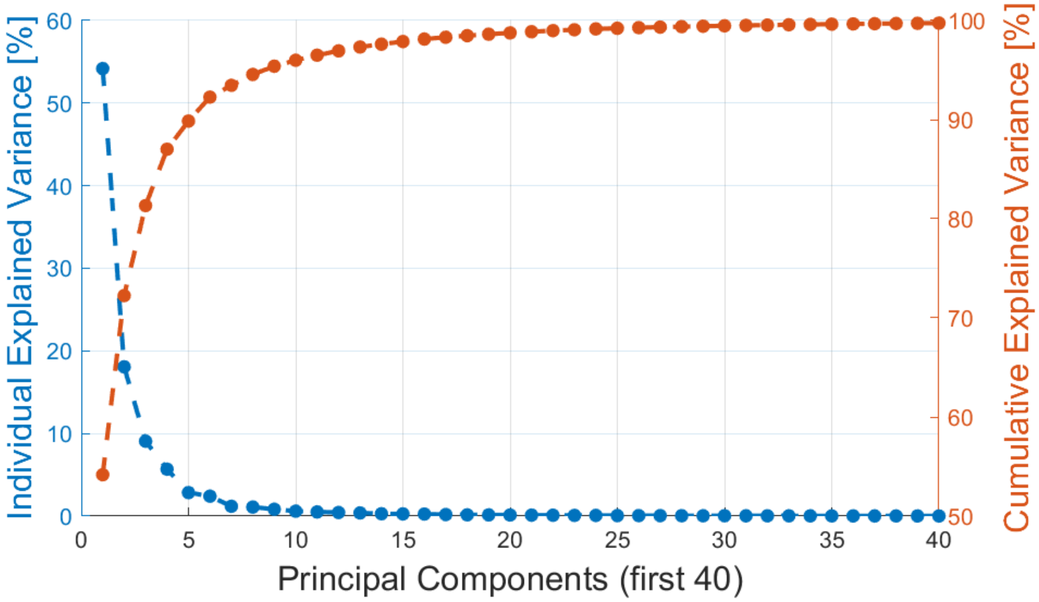

Figure 3.2: Individual and cumulative explained variance from principal component analysis of the electrochemical model state trajectories.

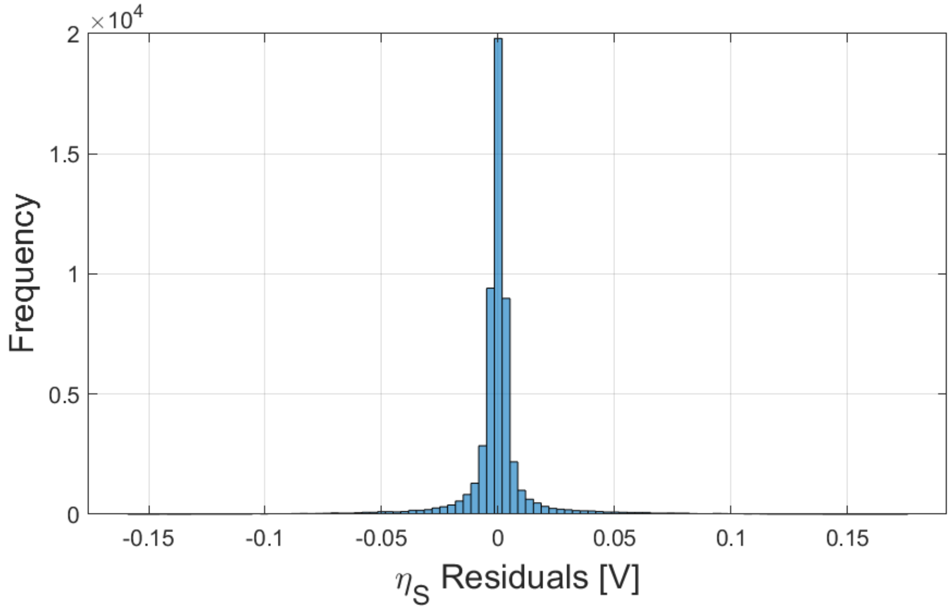

Figure 3.3: Histogram of test data residuals for $\mathcal{G}_{\eta_s}$.

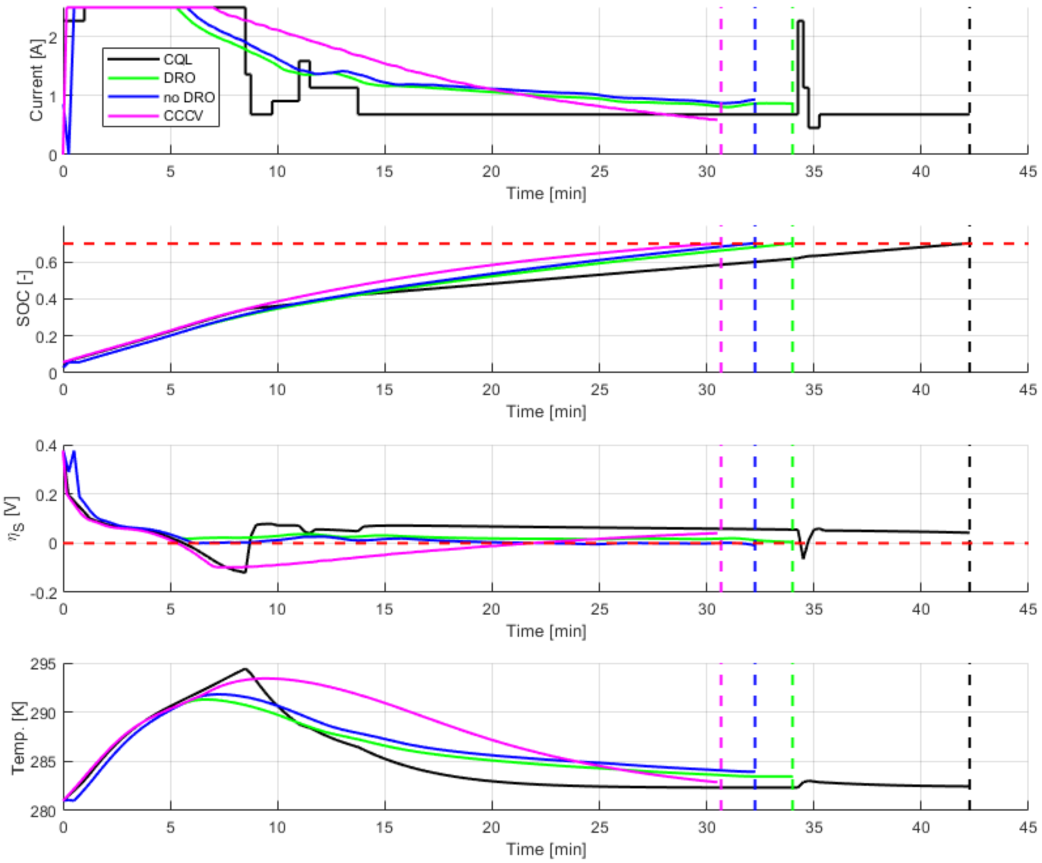

Figure 3.4: Optimal charging results for the DFN model using a nickel-manganese-cobalt (NMC) cell parameterization. Here, the maximum allowed C-Rate is 2.5C and the target SOC is 0.7. Charging is marked as complete at the vertical dotted lines for each respective trajectory.

Table 3.3: Relevant Hyperparameters

Parameter	**Description**	**Value**
Δt	Timestep	15 seconds
N	Control Horizon	4 timesteps
SOC_0	Initial state-of-charge	0.0286
SOC_{targ}	Target SOC	0.7
T_{amb}	Ambient Temperature	281 Kelvin
e	Number of Training Episodes	100
T	Length of Training Episode	3300 seconds
I_{max}	Maximum Charging Current	2.5 C
β	Ambiguity Set Confidence	0.9
ρ	Chance Constraint Risk Metric	0.1

The results from this implementation were inferior to a random search based optimization scheme. The analytic gradients made `fmincon` nearly 70% faster compared to using finite differences for gradient calculations. However, the average computation time per time step using `fmincon` was 9.1007 seconds whereas random search only required 2.0968 seconds per timestep on average. We also find that the random search approach yields results of higher relative quality in terms of the overall charging time performance compared to the `fmincon` solver. The improved performance of random search, in terms of speed and solution equality, led us to use the random search method for our final results included in this chapter.

Our first benchmark is a hyper-aggressive constant current constant voltage (CCCV) charging protocol with 2.5 C-rate maximum input current and 4.2 Volts cutoff voltage. A CCCV protocol charges the battery at the maximum allowed current until a cutoff voltage is reached. From that point on, the battery is charged at a rate that keeps the voltage at the specified threshold. Typically, CCCV profiles correspond to thresholds given in the battery cell specifications document, which tend to limit the maximum allowed input current to around 1C for most nickel-manganese-cobalt cells. For the sake of consistency, we keep the maximum allowed current the same for each method. CCCV contextualizes the relative performance of the proposed method.

As a point of comparison, we also implement conservative Q-learning (CQL), a popular offline reinforcement learning algorithm that addresses distributional shift through penalties on out-of-distribution (OOD) actions [60]. The CQL network is a feed-forward network with two hidden layers each composed of 64 neurons, and ReLU activations. The network input is the DFN state projected via the same PCA approach as our method. We discretize the input current into 11 bins between 0 and 2.5 C-rate. The network is trained in tandem with a target network iteratively with the same offline dataset used to learn the surrogate models of our approach. The reward function is given below, and is adopted with slight modification from recent work [85] successfully applying actor-critic RL methods to lithium-ion battery fast charging:

$$r = -I - 100(\mathbf{1}_{\eta_S<0}|\eta_S|) \tag{3.13}$$

We substitute an SOC-based reward with one based on input current as we find it yields CQL results that perform better with respect to charge time and constraint violation. A complementary OOD CQL loss is augmented to this reward function when training the networks [60]. CQL is a model-free method, meaning its sample efficiency isn't as high as our model-based approach. In [85], model-free actor critic methods are shown to require on the order of $3e3$ episodes of learning to achieve high-performing charging results. Given in this case we are dealing with more than an order of magnitude reduction in available data, the fidelity of these CQL results is actually quite impressive. CQL unfortunately does not provide certificates on safety and feasibility, which is reflected in the final charging profile as shown in Figure 3.4. This highlights a comparative advantage of our model-based RL methodology, namely its out-of-sample safety guarantees.

Both CCCV and CQL are relevant benchmarks for the following application and methodological reasons. Firstly, CCCV is by far the most popular fast-charging algorithm in practice. In fact, closed form optimal fast charging with reduced-order battery models collapses to the CCCV profile [86]. Speaking of methods, both algorithms share with our algorithm a lack of dependence on a-priori model knowledge. Explicit understanding of the underlying physics is not needed to achieve good control results with any of these methods, meaning they are operating on the same playing field. Second, CQL is a state-of-the-art learning-based control method that is designed to address many of the same problems our approach addresses. Principal among such problems is the "distributional shift" challenge that is prominent in offline RL. Finally, neither CQL nor any other existing state-of-the-art offline RL method (e.g. model-based offline policy optimization, or MOPO [110]) provide any explicit safety guarantees when considering the context of lacking model knowledge. This reveals the relative value of our novel methodology, insofar as our mechanism towards addressing distributional shift carries with it strong probabilistic safety guarantees.

Figure 3.4 shows the optimal fast charging results for versions of our algorithm with and without distributionally robust optimization. Overall, the CCCV protocol charges in 30.6 minutes, the non-robust predictive controller in 32.35 minutes, the full distributionally robust controller in 34.1 minutes, and the CQL controller in 42 minutes. The industry benchmark CCCV protocol yields a good performance with respect to charging time with a total time of 30.6 minutes. However, it significantly violates the safety constraint by up to 0.12 Volts, and for extended periods of the overall experiment. This would undoubtedly lead to significant degradation and potential failure of the cell. Figure 3.5 shows constraint violation for each learning-based method. Without the DRO architecture, the surrogate-based method provides a relatively high performing charging protocol which charges the battery cell in 32.35 minutes, only 5.7% slower than the CCCV approach. It also demonstrates improved safety relative to the industry CCCV benchmark. Specifically, the magnitude of the maximum constraint violation in the non-robust version of our algorithm is only 0.0082 Volts. With the added DRO framework based on Wasserstein ambiguity sets, we see that the charging protocol

Table 3.4: Comparison of relevant experiment metrics.

Algorithm	**Charge Time [min]**	**Feasible?**
CCCV	30.6	No
CQL	42	No
Surrogate Optimal Control (No DRO)	32.35	No
Robust Surrogate Optimal Control	34.1	Yes

satisfies the constraint at every instance in time, while also providing a competitive 34.1 minute charging time. These results illustrate the theoretical guarantees we expect from application of Wasserstein ambiguity sets. Relative to the non-robust version, the charging time with the DRO offset is only 5.4% slower, a tradeoff that may be worthwhile for the increased safety and mitigation of aging. CQL violates overpotential constraints and charges slowly in comparison, however we trained the CQL network with the exact same dataset as used by our method for consistency. An offline dataset with (i) more trajectories, and (ii) trajectories that more frequently violate constraints would yield higher performing CQL results, however such results would not have any guarantees of adhering to constraints. Table 3.4 shows a comparison of relevant results metrics.

Computational Effort Analysis

Comparing the computational requirements of this algorithm to those of our preliminary version in [53] reveals a host of meaningful insights. We are performing optimal control on the DFN model, which is characterized by 2687 state variables. In the past exploratory work, we tested a more rudimentary version of our algorithm on the single particle model with electrolyte and thermal dynamics (SPMeT), a model with 208 state variables. The average computation time per iteration with the DFN is 2.0968 seconds, when the algorithm is executed on a Windows desktop workstation equipped with a 9th generation Intel i5 processor. In [53], the average time per iteration was 1.7803 seconds when run on the same machine. Despite the more than 10-fold increase in the cardinality of the state vector of each model, the computational effort of the proposed algorithm only changes marginally by 17.81%. This slight difference is likely due to the more complex neural network architecture and DRO framework which we employ in our updated analysis.

Insights from Wasserstein DRO Algorithm

One unique aspect of this work from preliminary results presented in [53] is the application of Wasserstein ambiguity sets. Wasserstein ambiguity sets are differentiated from ϕ-divergence based chance constraint reformulation by their robust out-of-sample safety guarantee. We see this difference by observing that Wasserstein ambiguity sets provide a slightly more conservative result that that shown in previous work. This finding is clear from our DFN

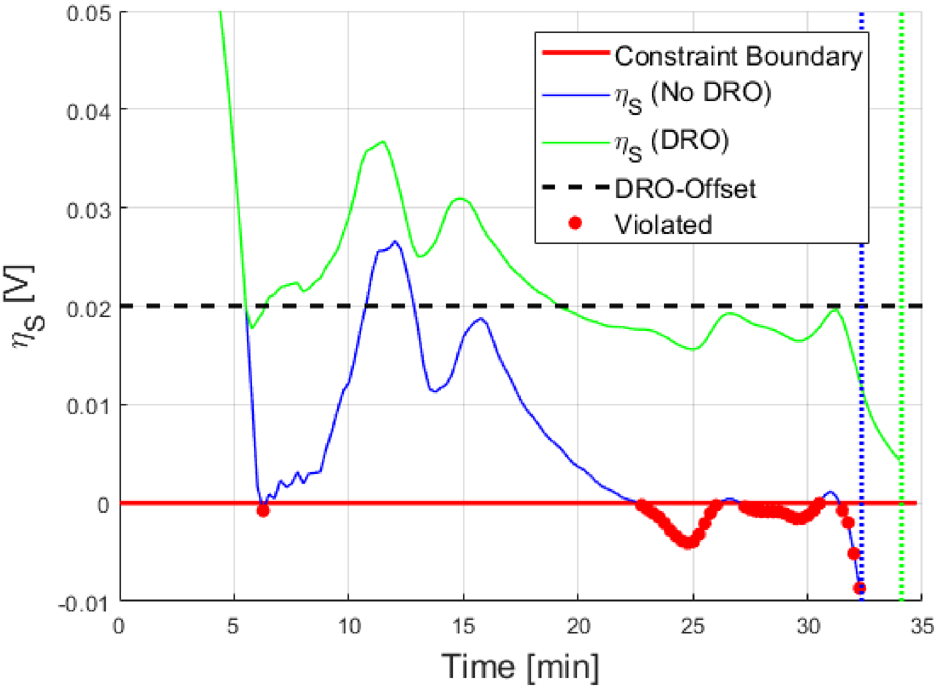

Figure 3.5: η_s evolution statistics for each respective method (omitting CCCV and CQL). While the DRO version violates the conservative constraint offset, it still yields safe charging behavior relative to the nominal constraint boundary. Conversely, the non-robust version of our algorithm violates the nominal constraint boundary in 25.38% of its timesteps.

case study. The DRO does prevent constraint violation entirely compared to the non-robust version which only attenuates its magnitude relative to CCCV. For safety critical control applications, this added safety from the out of sample safety guarantee is valuable.

To further demonstrate this added value, we refer to Figure 3.6 which shows a comparison of the cumulative distribution of $\mathcal{G}_{\eta_S}$ model residuals from test data and from the state-action pairs in the final optimal charging profile. This plot highlights the distributional shift problem which is a significant open challenge in offline RL research. Consider that when limited to a static, offline dataset for model training, applying resulting control policies to a real, dynamical system creates the opportunity for the agent to encounter states that fall out of the distribution of its training data. For high-dimensional nonlinear dynamical systems, the probability of this occurring is significant. Thus, safety must be guaranteed with respect to such OOD experience. Wasserstein ambiguity sets provide a strong means to satisfy this requirement, given their out-of-sample safety guarantee. While the final experimental distribution does not represent the true underlying distribution of residuals, it does present a significant deviation from what we observe in our test data. Besides some slight differences

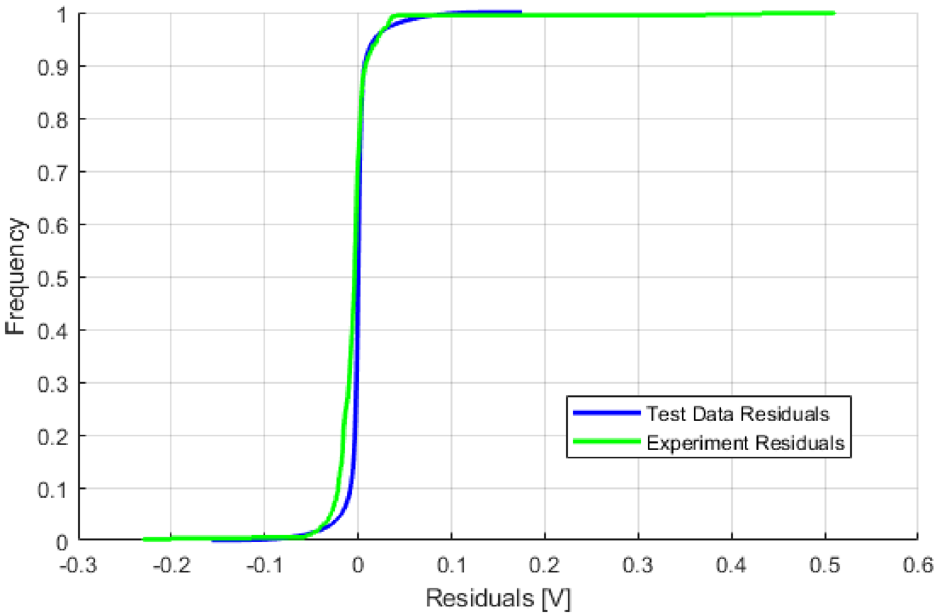

Figure 3.6: Comparison of cumulative distribution of $\mathcal{G}_{\eta_S}$ model residuals from test data and from the final optimal charging profile. These differences visualize the distributional shift problem that is a critical challenge in offline reinforcement learning.

in overall shape, the experimental residual distribution is more heavily skewed to higher magnitudes of modeling errors. Importantly in this case the maximum residual we observe is 0.5033 Volts, which is 2.908 times the magnitude of the largest residual represented in the test data set. This difference is just one way of demonstrating how distributional errors can come into play once we set out to apply an optimal charging policy

3.4 Conclusion

This chapter presents a novel framework for optimal control of high-dimensional dynamical systems. The key challenges to numerical optimal control addressed by this chapter include: (i) the "curse of dimensionality" incurred by high-dimensional systems, (ii) formulations that are not linear-quadratic, and (iii) ensuring safety/feasibility when constraint model errors occur.

We identify surrogate models that learn from limited offline datasets, and which absorb state transition dynamics to reduce compounded modeling errors. Principal component analysis applied to the training data allows us to project the high-dimensional data onto a reduced basis. This makes the modeling architecture conducive to fast identification and evaluation. Finally, we integrate these models into a receding horizon control framework.

Critically, our strategy utilizes distributionally robust optimization to robustify the solution to errors in the constraint function surrogate models. the OOD safety guarantee of Wasserstein DRO directly addresses the open challenge of distributional shift for offline RL problems. All combined, we demonstrate that the algorithmic approach yields tractable and robust control results for high-dimensional dynamical systems.

Chapter 4

Safe Learning and Adaptive MPC with Limited Model Knowledge and Data[1]

Abstract

This chapter presents an end-to-end framework for safe learning-based control using nonlinear stochastic MPC. We focus on scenarios where the controller is applied directly to a system of which it has highly limited experience, toward safety during tabula-rasa learning-based control as a challenging case for validation. We show under basic and limited assumptions that we can translate the probabilistic guarantees in Chapter 2 even with strong limitations on available data and model knowledge. We also present a coupled and intuitive formulation for the persistence of excitation (PoE) and illustrate the connection between PoE and the applicability of the proposed method. We validate these findings with case studies of extreme lithium-ion battery fast charging and autonomous vehicle obstacle avoidance using a basic perception system.

4.1 Introduction

This chapter presents a novel application of Wasserstein ambiguity sets to robustify model-based reinforcement learning (MBRL) and learning-based control (LbC) in safety-critical applications. Here, we define safety as the ability of the control policy to satisfy constraints. Translating safety to online reinforcement learning (RL) algorithms is a notoriously difficult open challenge in relevant literature. This work is motivated by unsolved shortcomings of many existing means to address this challenge, particularly a strong and often optimistic dependence on subject matter expertise. Two overarching examples include (i) assumed knowledge of underlying dynamics, and (ii) preexisting data of safe trajectories.

[1] This chapter is adapted from previously published work [51]. ©2023 IEEE. Reprinted, with permission, from Kandel, Aaron and Moura, Scott. "Safe Learning MPC With Limited Model Knowledge and Data." IEEE Transactions on Control Systems Technology (2023).

In Chapter 2, we outlined a simple extension of DRO theory that is amenable to translating results to the space of LbC with nonlinear, high-dimensional, unstructured systems. This chapter seeks to apply this method to address key open questions in the literature. Among those previously discussed, foremost is the lack of general methods that possess robustness when conducting *tabula-rasa* learning-based control, or those requiring significant assumptions on availability of prior data of safe control trajectories.

We present a novel and simple model-based LbC scheme based on MPC which provides strong probabilistic out-of-sample guarantees on safety. We validate our method using experiments that emulate *tabula-rasa* as closely as possible given our assumptions, but our algorithm is widely applicable to adaptive control scenarios especially when underlying dynamics may be poorly structured or difficult to characterize. By developing Wasserstein ambiguity sets relating to empirical distributions of modeling error, we can conduct MPC with an imperfect learned snapshot model while maintaining confidence on our ability to satisfy nominal constraints. The Wasserstein ambiguity sets allow us to optimize with respect to constraint boundaries that are shifted into the safe region. As our empirical distributions improve with more data, the offset variables tighten towards the nominal boundary in a provably safe way. Our approach yields probabilistic safety guarantees. Critically, in this work, we present this LbC algorithm along with (1) an explicit and fundamental persistence of excitation (PoE) scheme, and (2) highly limited SME assumptions. While many LbC methods are amenable to PoE schemes [28], the question of PoE is in some cases neglected despite its relevance. We actually show our explicit PoE scheme is fundamental to illustrating the applicability of our method. Our contributions combine to allow us to translate safety guarantees with highly limited model knowledge and data.

The overarching objective of this chapter is not to present the most high-performing LbC architecture, but rather to explore what kind of performance we can obtain when limiting our SME assumptions more than existing work in controls literature. Many control-theoretic methods provide stronger robust (i.e. safety w.p. 1) guarantees under much more restrictive assumptions. In our case, we label our method as “trustworthy” insofar as it relies on highly limited SME. Given the elusiveness of safety guarantees in RL literature, a probabilistic result within our context is powerful and describes improved safety we observe in our case studies.

We validate our approach with two case studies focusing on safety-critical applications. In the first, we learn a policy that safely charges a lithium-ion battery using a nonlinear equivalent circuit model. Battery fast charging presents a strong challenge for learning-based control methods, given that the optimal policy is a boundary solution which rides constraints until the terminal conditions are met. We also conduct a case study on safe autonomous driving using a nonlinear bicycle model of vehicle dynamics. We demonstrate that our algorithm provides a provably safe method for the vehicle to avoid obstacles while learning its dynamics from scratch.

We provide an open-source GitHub repository [50] for our case studies.

4.2 Distributionally Robust Model-Based Learning-Based Control

Fig. 4.2 shows a block diagram of our proposed control architecture, detailed within this chapter.

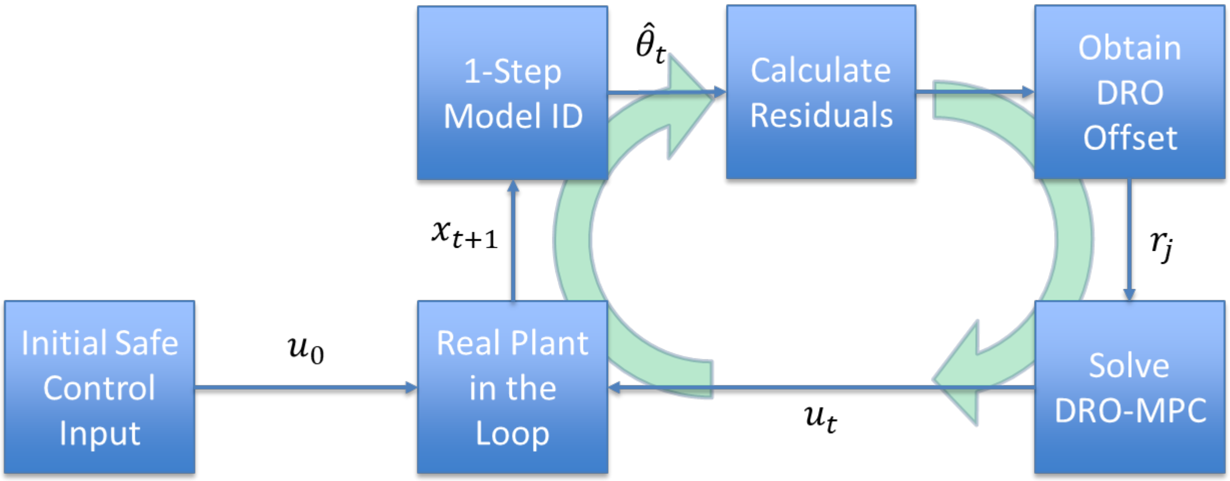

Figure 4.1: Diagram of safe Wasserstein-constrained MPC. In the most restrictive case, after initializing the controller, it immediately begins interacting with its environment. At every timestep, it observes an MDP state transition tuple, calculates model residuals, uses the residuals to calculate the DRO offset $r^{(j)}(k)$, and then solves a new MPC program at the next state. This application case serves as a purposefully extreme challenge of the robustness and behavior of our algorithm at what would otherwise be unreasonable levels of uncertainty and risk. Later in this chapter, we demonstrate that even under such extreme conditions, we manage to safely learn control policies for a host of nonlinear stochastic control problems. We do note, however, that our algorithm is much more widely applicable when prior data and SME is available.

Model Predictive Control Formulation

We apply Wasserstein ambiguity sets to robustify a learning model predictive controller, based on the following optimization program formulation. Given true plant dynamics:

$$x_{t+1} = f(x_t, u_t, W_t) \tag{4.1}$$

$$y_t = g(x_t, u_t, V_t) \tag{4.2}$$

where t is the current timestep, W_t is state noise, V_t is output measurement noise, x_t is the state variable, and y_t is the output variable. We assume access to full state and output measurements, subject to the measurement noises W_t and V_t. The capital letters represent random variables. Before considering modifications for distributional robustness to uncertainty (which also accommodate exogenous inputs), we seek to solve the following predictive control problem:

$$
\begin{aligned}
&\underset{u_{t:t+N-1}}{\text{minimize}} \sum_{k=t}^{t+N} J_k(\hat{x}_k, \hat{y}_k, u_k) && \text{(4.3a)}\\
&\text{subject to:} && \text{(4.3b)}\\
&\hat{x}_{k+1} = \hat{f}(\hat{x}_k, u_k, \theta_f) && \text{(4.3c)}\\
&\hat{y}_k = \hat{g}(\hat{x}_k, u_k, \theta_g) && \text{(4.3d)}\\
&\hat{y}_k \leq 0 && \text{(4.3e)}\\
&\hat{x}_t = x_t && \text{(4.3f)}
\end{aligned}
$$

where x_t is the known (measured) initial state at the current timestep t. The *"hat"* symbol indicates a predicted variable, and the learned models themselves are given by:

$$\hat{x}_{t+1} = \hat{f}(x_t, u_t, \theta_f) \tag{4.4}$$

$$\hat{y}_{t+1} = \hat{g}(x_t, u_t, \theta_g). \tag{4.5}$$

At a high level, these can be thought of as two separate models. However, when learning a black-box representation of the system, that single model can be trained to predict both sets of values $\hat{x}_{t+1}$ and $\hat{y}_t$. The parameters θ_f and θ_g are learned from historical data through model identification.

Model Identification

The models are used to predict state transition dynamics and constraint function outputs. We assume the true model parameters θ_f^* and θ_g^* are inaccessible to the controller. Several methods can be selected to learn the parameters online, and can depend on what type of learning model architecture is selected. In this chapter, we utilize nonlinear least-squares with neural network models for both the state transition dynamics and constraint functions:

$$\hat{f}(x_t, u_t, \theta_f) \leftarrow x_{t+1} \tag{4.6}$$

$$\hat{g}(x_t, u_t, \theta_g) \leftarrow y_t \tag{4.7}$$

where x_{k+1} and y_k are assumed to be measurable from the real system at the current timestep. When conducting MPC, the initial x_k is obtained by assuming full state observability throughout the LbC problem. From this point forward, we denote $\theta_{g;t}$ as the parameterization of the learned model of g at timestep t in the overall learning process.

Modeling Error Characterization

We characterize modeling error through comprehensive modeling residuals across varying prediction depths.

For example, consider a scalar system $x \in \mathbb{R}$, $y \in \mathbb{R}$ within three steps of model predictive control $N = 2$ with quadratic, time invariant objective function (state penalty $q = 1$, effort penalty $r = 1$, terminal state penalty $p = 1$):

$$\underset{u_t, u_{t+1}, u_{t+2}}{\text{minimize}}\; x_t^2 + \hat{x}_{t+1}^2 + u_t^2 + u_{t+1}^2 + \hat{x}_{t+2}^2 \tag{4.8a}$$

$$\text{subject to:} \tag{4.8b}$$

$$\hat{x}_t = x_t \tag{4.8c}$$

$$\hat{x}_{t+1} = \hat{f}(x_t, u_t, \theta_f) \tag{4.8d}$$

$$\hat{x}_{t+2} = \hat{f}(\hat{x}_{t+1}, u_{t+1}, \theta_f) \tag{4.8e}$$

$$\hat{x}_{t+3} = \hat{f}(\hat{x}_{t+2}, u_{t+2}, \theta_f) \tag{4.8f}$$

$$\hat{y}_t = \hat{g}(x_t, u_t, \theta_g) \tag{4.8g}$$

$$\hat{y}_{t+1} = \hat{g}(\hat{x}_{t+1}, u_{t+1}, \theta_g) \tag{4.8h}$$

$$\hat{y}_{t+2} = \hat{g}(\hat{x}_{t+2}, u_{t+2}, \theta_g) \tag{4.8i}$$

$$\hat{y}_t \leq 0 \tag{4.8j}$$

$$\hat{y}_{t+1} \leq 0 \tag{4.8k}$$

$$\hat{y}_{t+2} \leq 0 \tag{4.8l}$$

Suppose we find a sequence u_t^*, u_{t+1}^*, u_{t+2}^* from solving 3 sequential model predictive control problems with the true plant in the loop. Since we are using learned models to solve these predictive control problems, these inputs are likely not actually optimal for the system, and with added PoE they include exploratory aspects. In each case we apply the first control input to the system to obtain x_{t+1}^*, x_{t+2}^*, x_{t+2}^* We can quantify prediction error of the learned constraint function in the following manner:

$$R_1^{(t)} = g(x_t, u_t^*) - \hat{g}(x_t, u_t^*, \theta_g) \tag{4.9a}$$

$$R_1^{(t+1)} = g(x_{t+1}^*, u_{t+1}^*) - \hat{g}(\hat{x}_{t+1}, u_{t+1}^*, \theta_g) \tag{4.9b}$$

$$R_1^{(t+2)} = g(x_{t+2}^*, u_{t+2}^*) - \hat{g}(\hat{x}_{t+2}, u_{t+2}^*, \theta_g) \tag{4.9c}$$

These are 1-step residuals, as denoted by the subscript R_1, since $\hat{x}_{t+1} = f(x_t, u_t^*)$ and $\hat{x}_{t+2} = f(x_{t+1}^*, u_{t+1}^*)$. In these equations, the function g represents our observations from the real system (simple data), and the function $\hat{g}$ represents the predictions of our learned constraint model. We take the absolute value since these residuals will be introduced as variables that add conservatism relative to the existing constraint boundary. Since we conduct predictive control, we also want to quantify modeling errors after 2, 3, or more steps of prediction into the future using learned models, as errors can accumulate and become worse with successive prediction steps. This happens in the following way:

$$R_1^{(t)} = |g(x_t, u_t^*) - \hat{g}(x_t, u_t^*, \theta_g)| \tag{4.10a}$$

$$R_2^{(t)} = |g(x_{t+1}^*, u_{t+1}^*) - \hat{g}(\hat{f}(x_t, u_t^*, \theta_f), u_{t+1}^*, \theta_g)| \tag{4.10b}$$

$$R_3^{(t)} = |g(x_{t+2}^*, u_{t+2}^*) - \tag{4.10c}$$

$$\hat{g}(\hat{f}(\hat{f}(x_t, u_t^*, \theta_f), u_{t+1}^*, \theta_f), u_{t+2}^*, \theta_g)| \tag{4.10d}$$

As is shown here, modeling error accumulates from learned representation of both the constraint function $\hat{g}$ and the learned dynamics function $\hat{f}$.

Remark 2 *We choose to take the absolute value of residuals. This decision is not necessary, but makes intuitive sense given the application. Since we are intending to modify the nominal constraint boundary, signals of modeling errors that show underestimation could lead to an offset that potentially moves the constraint into the unsafe region. We seek to avoid this, and only create offsets that reduce the size of the feasible region.*

The model identification process utilizes the 1-step residuals to minimize mean-square prediction error (MSE) of the prediction of the state transition compared to past observations. The multi-step residuals are utilized by the DRO framework to adjust conservatism deeper into the future based on cumulative modeling error.

By representing modeling error this way, we lump all relevant sources of modeling error into an additive term. As previously discussed, the absolute value is taken as a precautionary measure. Omitting that transformation provides the following simple expression:

$$g(x_{t+2}^*, u_{t+2}^*) = \hat{g}(\hat{f}(\hat{f}(x_t, u_t^*, \theta_g), u_{t+1}^*, \theta_g), u_{t+2}^*, \theta_g) + R_3^{(t)} \tag{4.11}$$

By treating the residuals as random variables drawn from a true distribution $\mathbb{P}$, the constraints will by definition be additive in the random variable/modeing error.

Safety and Robustness using Wasserstein Ambiguity Sets

Now that we have outlined the distributionally robust chance constrained approach using the Wasserstein ambiguity set, we can describe how it fits within our robust control framework.

The residuals defined in the previous subsection IV.C entail a representation of the modeling error. This is only true because the constraint functions are evaluated using predicted states from the learned dynamical model, whose true representation is unknown. By considering process error/residuals as an additive noise term, we can maximize the utility of the DRO reformulation in [33] which requires this linear structure in the constraint:

$$g(x_k, u_k, \theta_{g;t}) + \mathbf{R}_1 \leq 0 \tag{4.12}$$

As previously discussed and shown in equation (4.11), by design, this linear structure will always occur. These residuals are random variables characterized by empirical distributions based on our observations. Now, we've bolded the variable $\mathbf{R}_1$ to indicate it is a random variable, whereas the previous value $R_1^{(t)}$ was a realization of this random variable at time t.

To accommodate distributional uncertainty in our estimate of $\hat{\mathbb{P}}$, we transform the constraint (4.12) for each of $1 \rightarrow N+1$ step residuals into a joint distributionally robust chance constraint via Wasserstein ambiguity set as follows:

$$\inf_{\mathbb{P}\in\mathbb{B}_\epsilon} \mathbb{P}\begin{bmatrix} \hat{g}(\hat{x}_k, u_k, \theta_{g;t}) + \mathbf{R}_1 \leq 0 \\ \hat{g}(\hat{x}_{k+1}, u_{k+1}, \theta_{g;t}) + \mathbf{R}_2 \leq 0 \\ \vdots \\ \hat{g}(\hat{x}_{k+N}, u_{k+N}, \theta_{g;t}) + \mathbf{R}_{N+1} \leq 0 \end{bmatrix} \geq 1-\eta \tag{4.13}$$

The reformulation we adopt from [33] presents a simple method to accommodate the constraint without inverting the CDF. If we operate under the assumption that the residuals for $i = 1, ..., N$ steps are uncorrelated, then we can decompose this joint chance constraint into a set of individual chance constraints. This decomposition could be useful if the optimization algorithm we select to solve the MPC problem scales unfavorably with the dimension of the constraints. Algorithm 1 provides an overview of the real-time implementation of our approach. As previously stated, the process for computing r entails a simple scalar convex optimization program.

Remark 3 *The reformulation from [33] adds cardinality of constraints that scale with order 2^m. However, our formulation of modeling error as an additive residual allows the number of constraints to remain constant. We detail this property in the Appendix of this dissertation. The simple answer is that, by taking the absolute values of the residuals, the random variable that represents modeling error is strictly non-negative. This means a negative realization is impossible to encounter, and need not be accommodated. By keeping the cardinality of constraints constant, the computational scalability of our approach is preserved for higher dimensional control problems.*

At each time step, we compute model residuals with our most recent estimate $\theta_{g;t}$ using predicted state transitions from our entire cumulative experience, compile a unique empirical distribution $\hat{\mathbb{P}}$ corresponding to each individual chance constraint, and compute the value of r in (2.5) to reformulate the distributionally robust chance constraints. We can begin the overall process with a small control horizon N, and gradually increase N as we accumulate more and more data from experience. The residuals we compute are for horizon lengths of 1 to N-steps, meaning the elements of $\mathbf{R}$ correspond to each of $i = 1, ..., N$ step residuals. Then, we assemble a joint chance constraint where the elements of the column vector of the random variable are the $1 \rightarrow N$ step residuals. In [33], authors pursue a DRO reformulation that utilizes a polytopic representation of the uncertainty set. Our formulation preserves scalability by isolating dependence on the random variable in the constraint. Our Appendix shows the logic that allows the cardinality of constraints to remain constant.

Finally, when we conduct MPC, we replace the nominal constraints with their distributionally robust counterparts:

$$\begin{aligned}
\underset{u \in \mathcal{U}}{\text{minimize}} \quad & \sum_{k=t}^{t+N} J_k(\hat{x}_k, u_k) && \text{(4.14a)} \\
\text{s. to:} \quad & \hat{x}_{k+1} = \hat{f}(\hat{x}_k, u_k, \theta_{g;t}) && \text{(4.14b)} \\
& \begin{bmatrix} \hat{g}(\hat{x}_k, u_k, \theta_{g;t}) \\ \hat{g}(x_{k+1}, u_{k+1}, \theta_{g;t}) \\ \vdots \\ \hat{g}(\hat{x}_{k+N}, u_{k+N}, \theta_{g;t}) \end{bmatrix} + r^{(j)} \leq 0 && \text{(4.14c)} \\
& \hat{x}_0 = x_t && \text{(4.14d)}
\end{aligned}$$

Algorithm 2 describes the implementation of our MPC architecture coupled with the Wasserstein distributionally robust optimization scheme:

Algorithm 2 Wasserstein Robust Learned MPC

```
Require: State space §, Action space $\mathcal{U}$
  for $t$ in range $t_{max}$ do
    if $t = 1$ then
      $u_t$ = known safe input, $N = 1$
    else
      Update the dynamical system model and constraint functions $\theta_{t-1} \rightarrow \theta_t$
      Receding horizon increment rule (i.e. $N = min\{N_{targ}, round(\frac{t}{N_{targ}}) + 1\}$)
      Obtain Wasserstein ambiguity set offset $r$:
      $u_t \leftarrow$ Solve MPC optimization program (4.15a)-(4.15i)
    end if
    $x_{t+1} = f(x_t, u_t, W_t)$ (Truth plant)
    $y_t = g(x_t, u_t, V_t)$ (Truth plant)
  end for
```

The MPC program specified in (4.15a-4.15i) details the slight modifications made to (5.13-5.34) accommodating the coupled PoE component to our LbC framework. We discuss this in more detail in part F. of this section.

One important note concerns a specific scenario of model adaptation where the true underlying system slowly changes. Our application of receding horizon control necessitates the use of a snapshot model in the prediction phase. This requires we assume the rate of change of the dynamics of the true plant is relatively small. In such conditions, however, the historical residuals we collect through measurements will slowly lose relevance. This issue can be easily reconciled with use of either a moving window of residuals, or with a proper forgetting scheme. In this chapter, we propose a simple method to accommodate such cases. Since the focus of this chapter is on *tabula-rasa* learning-based control, we relegate the discussion of this additional framework to this dissertation's appendix.

Horizon Increment Rule

MPC with well-defined dynamical structure can leverage judicious selection of the prediction horizon as a component to proving recursive feasibility. When considering a general class of systems as is the case with MBRL, the prediction horizon becomes a hyperparameter that manages the tradeoff between prediction depth and computational expense. In this work, we elect to define a simple horizon increment rule for our experiments. Typically in learning-based control, the prediction horizon is a hyperparameter whose selection can be done empirically with more nuanced methods [112, 65]. In our case studies, which we design to emulate *tabula-rasa* learning-based control as closely as is consistent with the assumptions of our algorithm, we utilize this horizon increment rule as a heuristic to simply allow the problem to be rapidly solved. By solving severely restrictive case studies, we validate the performance of our method under the most challenging context for which it is technically designed. For real-world applications, the horizon can often be selected using a combination of available subject matter expertise (which should not be ignored if it is available), and automatic tuning methods like those of [112, 65]. The increment rule is not meant as a serious method for real-world embedded control systems that often possess highly limited computational resources.

Persistence of Excitation, and Problem Assumptions

This subsection defines the set of least restrictive assumptions we identify towards achieving safe learning-based control. In this work, we consider systems with non-hybrid dynamics for simplicity. Our method leverages proved safety properties from [33], which apply to static optimization programs. We identify that these methods can apply to LbC problems under a series of assumptions made in this section. These assumptions almost entirely relate directly to situations when the dynamical, DRO, and PoE components, which are normally not considerations for static optimization programs, could create opportunities for empty feasible sets. This subsection defines a PoE scheme directly amenable to translating guarantees from [33] to our formulation. Notably, our assumptions are significantly less restrictive than those of existing LbC methods. The majority of these assumptions relate to clear necessary conditions which we detail here:

Assumption 1 *A feasible state and control trajectory exists for each prediction horizon N in the optimal control problem.*

This is the most fundamental requirement to apply safe control.

Assumption 2 *We assume we know a safe control input which we can apply at the first timestep.*

Starting with limited model knowledge, if we don't know a temporarily safe control input we can apply at the first timestep, we obviously can't translate any meaningful safety certificates. This contrasts to other work which requires knowledge of safe control trajectories throughout the time horizon, or a known safe backup policy.

Assumption 3 *Starting with an optimal control problem of the form (5.11-4.3f), suppose we have a constraint function* $g(x_k, u_k, \theta_{g;t}) : \S \times \mathcal{U} \times \theta \to \mathcal{S}$*. The sublevel set* $\mathcal{G}_{r_{DRO}} = \{(x,u) \in \S, \mathcal{U} \; : \; g(x,u) + r_{DRO} \leq 0\}$ *defines the adjusted feasible region, where feasibility is satisfied at the current timestep. This set must not be empty* $\forall r_{DRO} \in \mathcal{R}$*, where the set* $\mathcal{R} = \{r_{DRO} \in \mathbb{R} \; : \; 0 \leq r_{DRO} \leq r_{DRO;max}\}$ *describes the set of all potential values of the DRO offset.*

Since our method relies on creating an offset from the nominal constraint boundary, any potential value of the offset must lie in the image of the constraint function.

This assumption can be thought of as a generalization of a common LbC assumption that relates to "bounded modeling error," an example of which is given by Assumption 2 in [8]. In our case, using general function approximation, our method to quantify model error is empirically based on residuals. If the residuals of the learned model are too large, indicating our learned model is inaccurate, the resulting computed r_{DRO} (which is a conservative approximation of the residual, based on its distribution) will enforce a large offset from the nominal boundary. This assumption says that if the learned model is sufficiently inaccurate, the offset will be so large that the adjusted feasible region is empty, which is incompatible with the setup of [33]. The value $r_{DRO;max}$ represents any maximum residual value we can potentially infer from the problem, and can be defaulted to as an empirical approach if this case is reached in a real problem, although safety properties may not be reliable in such cases. Our experiments show such scenarios can be unlikely to occur, although the possibility of their occurrence should be considered.

The next assumption relates to a slightly stronger condition regarding persistence of excitation (PoE). The agent must be capable of exploring during LbC. In order to ensure the guarantees from [33] translate under those diverse circumstances, the same statements of 3.1-3.3 must be satisfied with respect to an additional exploration process $\mathcal{N}$ that ensures PoE.

For clarity, we define the following modified MPC program that considers an additive exploration signal from $\mathcal{N}$:

$$\underset{u, u^n \in \mathcal{U}}{\text{minimize}} \quad \sum_{k=t}^{t+N} J_k(\hat{x}_k, u_k) \tag{4.15a}$$

$$\text{s. to:} \quad \hat{x}_{k+1} = \hat{f}(\hat{x}_k, u_k, \theta_{g;t}) \tag{4.15b}$$

$$\hat{x}^n_{k+1} = \hat{f}(\hat{x}^n_k, u^n_k, \theta_{g;t}) \tag{4.15c}$$

$$\hat{g}(\hat{x}_k, u_k, \theta_{g;t}) + r_{DRO} \leq 0 \tag{4.15d}$$

$$\hat{g}(\hat{x}_k^n, u_k^n, \theta_{g;t}) + r_{DRO} \leq 0 \tag{4.15e}$$

$$u^n = u + N_{i:i+N} \tag{4.15f}$$

$$N_{i:i+N} \sim \mathcal{N} \tag{4.15g}$$

$$\hat{x}_0 = x_t \tag{4.15h}$$

$$\hat{x}_0^n = x_t \tag{4.15i}$$

where $\mathcal{N}$ is the distribution of a random exploration process which can be added to the nominal control input, and the superscript x^n and u^n denote trajectories perturbed by the exploration signal. The solution $u^n(t)^\star$ is then applied to the plant at time step t.

Remark 4 *Equations (4.15a-4.15i) guarantee feasibility from $k = t$ to $k = t+N$ for a system with parameters $\theta_{g;t}$ with a specified risk metric/probabilistic guarantee. This is formulated to guarantee feasibility over the control horizon. To assess recursive feasibility, one could utilize the methods from [114, 27] that require more significant restrictions in the form of model knowledge, mathematical structure on the feedback policy, and prior existing safe data.*

The additive noise perturbation for exploration takes inspiration from common methods with actor-critic or policy gradient learning, where noise via an Ornstein-Uhlenbeck process is added to the control input [66]. Relative to those existing methods, we make the following modifications for implementation:

Remark 5 *We must constrain both nominal and perturbed trajectories to ensure safety even with exploration. If we only add the perturbation after solving the MPC program, safety is not guaranteed.*

Remark 6 *A scalarized tradeoff between $J_k(\hat{x}_k, u_k)$ and $J_k(\hat{x}_k^n, u_k^n)$ can be formulated to balance exploration and exploitation during planning.*

Now, we define the next assumption relevant to translating safety to LbC systems under strong limitations on SME:

Assumption 4 *Given the noise process $\mathcal{N}$ defined to satisfy PoE for the model identification problem, the constraints $g(x_k, u_k, \theta_{g;t})$ and $g(x_k^n, u_k^n, \theta_{g;t})$ of the snapshot model must be satisfied for every realization from $\mathcal{N}$ throughout the overall finite-time optimal control problem.*

Given these conditions, we state the following remark detailing the properties of our method:

Remark 7 *Based on the provided safety guarantee afforded from the adopted DRO framework from [33], (5.13-5.34) admits a feasible solution that satisfies the nominal constraints w.p. $1 - \eta$ as long as the feasible set is not empty, which follows from Assumptions 3.1-3.4.*

We also state two remarks that help with implementation of our approach.

Remark 8 *These assumptions must also hold for the prediction horizons chosen at each instant in time.*

Remark 9 *If the DRO offset is so large it creates an empty feasible set, an artificial value* $r_{DRO;max}$ *can be defaulted to to facilitate implementation, although safety guarantees in such situations may be difficult to translate. If a random search is used to solve the MPC program in such cases, the evaluated trajectory that creates the least predicted constraint violation given the unmodified DRO offset can be selected.*

4.3 Case Study in Safe Online Lithium-Ion Battery Fast Charging

In this section, we validate our approach using a nonlinear lithium-ion battery fast charging problem. This problem closely emulates the performance-safety tradeoffs of common safe RL validation studies including ant-circle [1]. Specifically, the objective is to charge the battery cell as fast as possible, but the charging is limited by nonlinear voltage dynamics which must stay below critical thresholds. Violation of the voltage constraint can lead to rapid aging and potential catastrophic failure. However, higher input currents (which increase voltage) also directly charge the battery more rapidly. Thus, the optimal solution is a boundary solution where the terminal voltage rides the constraint boundary. This presents a problem with significant challenges and tradeoffs relating to safety and performance. Exploring how such algorithms accommodate these challenges can reveal insights into their overall efficacy and shortcomings.

Equivalent Circuit Model of a Lithium-Ion Battery

Lithium-ion batteries can be modeled with varying degrees of complexity. Some of the more detailed dynamical models are based on electrochemistry. For example, the Doyle-Fuller-Newman (DFN) electrochemical battery model is a high-fidelity first-principles derived physics based model of the dynamics within a lithium-ion battery [30]. Varying model-order reduction can be applied, yielding versions including the single particle model and the equivalent circuit model (ECM). For simplicity, this section's case study utilizes an ECM. The relevant state variables in this model are the state of charge SOC and capacitor voltages V_{RC} in each of two RC pairs. The relevant constraint is on the terminal voltage V. This constraint prevents the battery from overheating or aging rapidly during charging and discharging. The state evolution laws are given by:

$$SOC_{k+1} = SOC_k + \frac{1}{Q} I_k \cdot \Delta t \tag{4.16}$$

$$V_{\mathrm{RC}_1;k+1} = V_{\mathrm{RC}_1;k} - \frac{\Delta t}{R_1 C_1} V_{\mathrm{RC}_1;k} + \frac{\Delta t}{C_1} I_k \tag{4.17}$$

Table 4.1: Safety, computational, and performance comparison for DRO-MPC and MPC with battery fast charging. Activation of the DRO offset begins at minResidNum = 2.

(DRO)	% Violations [%]	Max Voltage [V]	Iteration Time [s]	Time [min]
1	0.0 %	3.5944	0.8551	7.3833
2	0.4 %	3.7004	0.8473	7.7667
3	0.2 %	3.6887	0.8529	7.3000
4	0.6 %	3.7098	0.8503	8.1833
5	0.0 %	3.5927	0.8688	7.5333
6	0.4 %	3.7344	0.8550	7.7833
7	0.4 %	3.7032	0.8643	8.1167
8	0.2 %	3.6921	0.8692	7.6667
9	0.2 %	3.6916	0.8620	7.8667
10	0.2 %	3.6985	0.8375	8.0167
Averages	0.26%	3.6806	0.8562	7.8150
(No DRO)	% Violations [%]	Max Voltage [V]	Iteration Time [s]	Time [min]
1	4.2 %	3.7795	0.8630	6.8667
2	7.4 %	3.7604	0.8345	6.8667
3	5.0 %	3.7474	0.8055	6.7833
4	13.6 %	3.7284	0.7938	6.8500
5	8.0 %	3.9072	0.8020	6.8333
6	16.2%	3.9060	0.7977	6.8667
7	8.0 %	3.9040	0.8240	6.8667
8	11.6 %	3.7651	0.7875	7.0167
9	7.2 %	3.7736	0.8237	6.8000
10	16.4 %	3.7634	0.7928	6.7500
Averages	9.76 %	3.8035	0.8125	6.8500

$$V_{\mathrm{RC}_2;k+1} = V_{\mathrm{RC}_2;k} - \frac{\Delta t}{R_2 C_2} V_{\mathrm{RC}_2;k} + \frac{\Delta t}{C_2} I_k \tag{4.18}$$

$$V_k = V_{\mathrm{ocv}}(SOC_k) + V_{\mathrm{RC}_1;k} + V_{\mathrm{RC}_2;k} + I_k R_0 \tag{4.19}$$

where $I(t)$ is the current input (which is the control variable for this problem), and V_{OCV} is the open-circuit voltage function, which is conventionally measured through experiments. The full experimental OCV curve is used to represent the true plant in the loop, and is obtained from a lithium-iron phosphate (LFP) battery cell [91]. In this case study, we learn the dynamics of the model using a simple feed-forward neural network.

Table 4.2: Relevant Parameters for Battery Case Study

Parameter	Description	Value	Units
Q	Charge Capacity	8280	$[\frac{1}{A.h}]$
R_0	Resistance	0.01	$[\Omega]$
R_1	Resistance	0.01	$[\Omega]$
R_2	Resistance	0.02	$[\Omega]$
C_1	Capacitance	2500	$[F]$
C_2	Capacitance	70000	$[F]$
Δt	Timestep	1	[s]
N_{targ}	Max Control Horizon	8	[-]
η	Risk Metric	0.025	[-]
β	Ambiguity Metric	0.99	[-]
SOC_0	Initial SOC	0.2	[-]
SOC_{targ}	Target SOC	0.8	[-]
$V_{RC_1}(0)$	Init. Cap. 1 Voltage	0	[V]
$V_{RC_2}(0)$	Init. Cap. 2 Voltage	0	[V]

Model-Predictive Control Formulation

We utilize the following formulation of fast charging:

$$\underset{I_k \in \mathcal{U}}{\text{minimize}} \sum_{k=t}^{t+N} (SOC_k - SOC_{target})^2 \tag{4.20}$$

$$\text{s. to:} \quad (5.23) - (5.25), \quad SOC(0) = SOC_0 \tag{4.21}$$

$$V_k \leq 3.6V, \quad 0A \leq I_k \leq 40A \tag{4.22}$$

The relevant parameters of the true model and DRO-MPC program are referenced in Table 4.3.

Remark 10 *In our case, we assume the controller does not have access to the form of the underlying dynamics given by (5.23-5.25). Instead, we apply our end-to-end LbC method to learn the dynamics "from scratch" as is consistent with tabula-rasa learning methods. We utilize neural network black-box models to accomplish this. The rules used to update the neural network parameters affect the convergence of the data-driven model to accurate behavior, which also effects empirical safety. We keep the neural network training consistent between our DRO algorithm and its non-robust baseline. The exact training procedure can be referenced in [50]. Updating the model more slowly at first tends to encourage more consistent behavior.*

In these case studies, we apply perturbation to the inputs that further excite the system, towards ensuring PoE. These perturbations are drawn as uniform vectors whose elements lie between $-2.5 \leq x_p \leq 2.5$ Amps. These perturbations are applied to both the distributionally robust controller, as well as the non-robust baseline controller In both cases, we seek to ensure mutual constraint satisfaction for the trajectories predicted using both the nominal and perturbed inputs.

We only allow a maximum total of 500 seconds for the battery to be charged. The timestep $\Delta t = 1$ seconds, $\eta = 0.025$, $\beta = 0.99$, and $N_{targ} = 8$ steps. Our neural network dynamical model has 1 hidden layer with 3 neurons and sigmoid activation function, with a linear output layer. To solve the MPC problem, we apply a $(1 + \lambda)$ evolutionary strategy (ES) based on a normally distributed mutation vector. In our appendix, we describe how this strategy works, why we select it, and other reasonable alternatives. The solver works with a single iteration and 250,000 mutants. The initial point of the ES is taken as the optimal point from the previous timestep. Addressing Assumption 2, we assume that at the first timestep, control inputs of $I_k \leq 25$ Amps are known to be temporarily safe. Since we constrain voltage which is a scalar, the constraint function dimension $m = 1$.

Our baseline is a learning MPC controller with no DRO framework. We adopt the same problem formulation as if we were going to add the constant r_{DRO} to the constraints, but we omit the DRO constant in the end to evaluate the impact it has on the robustness of the final control law.

Results

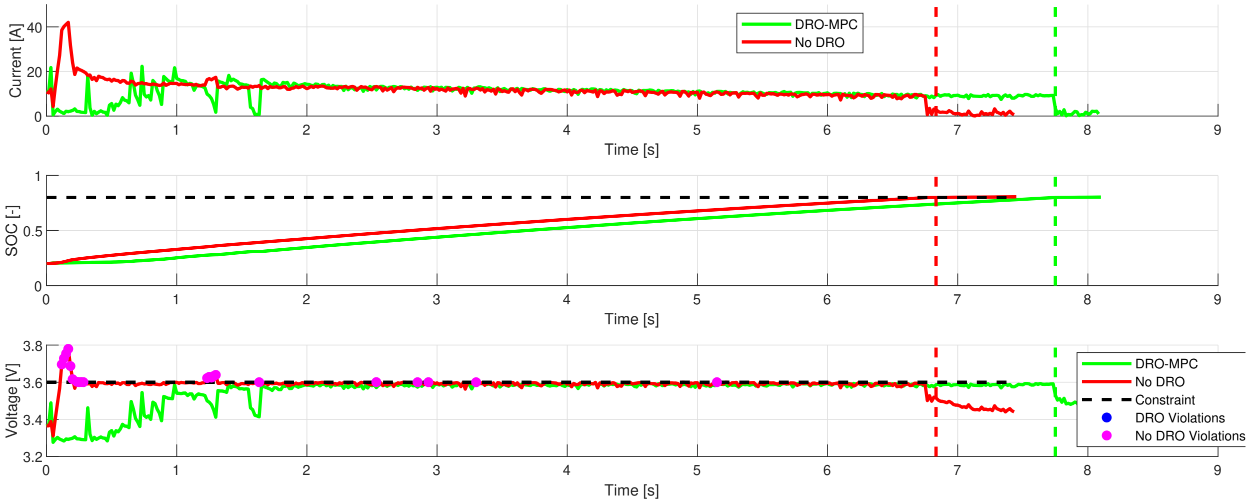

Figure 4.2: Comparison of nonlinear MPC Controller with and without DRO for lithium-ion battery fast charging. Run 1 is shown here.

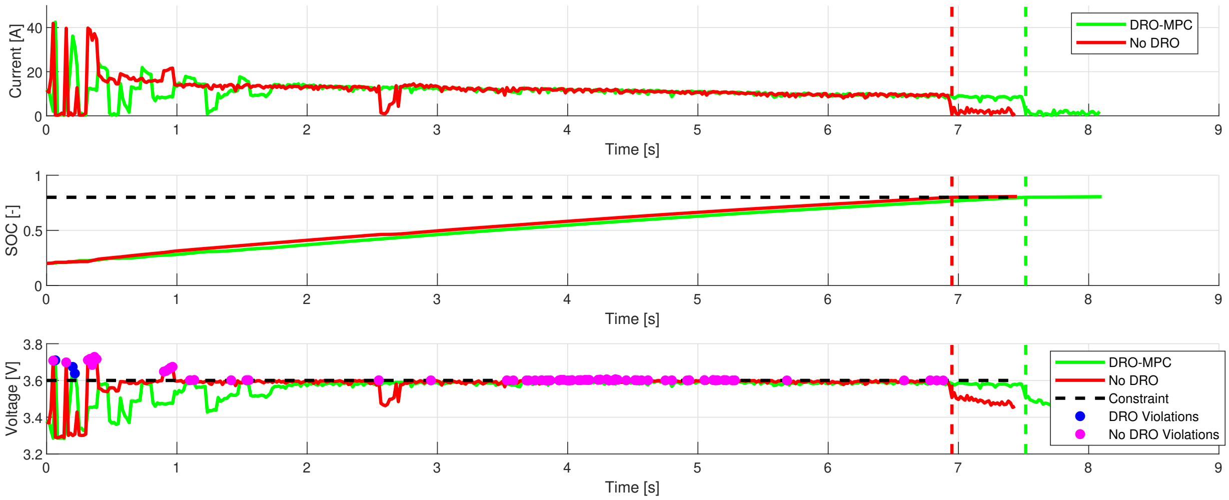

Figure 4.3: Comparison of nonlinear MPC Controller with and without DRO for lithium-ion battery fast charging. Run 4 is shown here.

In total, we conducted 10 experiments with identical designs but different initial random seeds. We run our algorithm and a non robust baseline for these 10 runs on the same battery fast charging problem detailed in the previous subsections. Table 4.3 shows the performance, computation, and safety statistics for each of these runs. For a closer look, we go to Figure 4.2 which shows one run of both the DRO algorithm and its non-robust counterpart. In the case of Figure 4.2 (run 1), the DRO-based does not violate the constraint at any point. In Figure 4.3 we see the highest incidence of constraint violation for the DRO controller (from run 4). Figure 4.4 shows the time evolution of the DRO offset from run 4.

Conversely, the non-robust versions both experiences a combination of initial, significant voltage spikes as well as minor violations which persist throughout the experiments. In total, if we focus on Figure 4.3 (run 4), the non-robust version violated constraints in 13.6 % of timesteps (68 timesteps out of 500 total). The charging time was 6.85 minutes, which was 16.29% faster than the DRO version, whose charging time was 8.1833 minutes. This makes intuitive sense, as the added DRO framework introduces additional conservatism which affects the performance of the overall control policy.

Overall across all 10 runs, our DRO version violates constraints in 0.26% of total timesteps, which is well within the chosen value of $\eta = 0.025 = 2.5\%$ over just a single optimization iteration. The non-robust version, however, violates constraints in 9.76% of total timesteps on average. Similarly, there is a stark difference in the maximum voltages seen by the robust and non-robust versions, with the DRO framework reducing the mean peak voltage by 122.9 millivolts. The DRO calculations increase the overall computation time by an average of

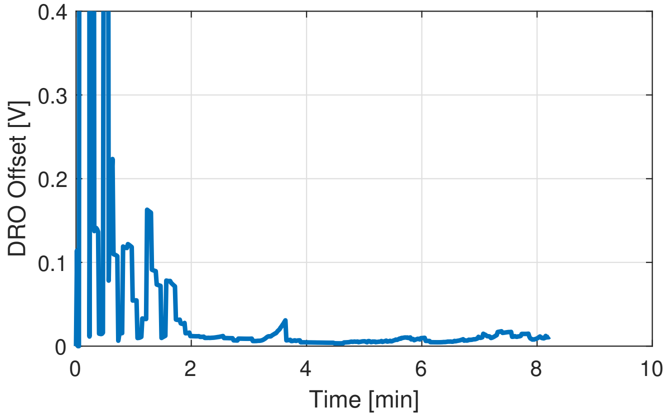

Figure 4.4: Time evolution of DRO offset from run 4.

43.7 milliseconds per timestep, and allow the algorithm in this case to run in real time. No optimizations were made to the Matlab code to expedite the runtime of either algorithm, and the only difference in code between the two algorithms is the auxiliary and separate DRO framework. Finally, across the 10 total runs the overall charging time with the DRO framework averages 7.8150 minutes, approximately 14.1% longer than that of the non-DRO version. Given the safety-critical nature of this control problem, the safety guarantees of our algorithm are likely well worth the marginal degradation to the charging performance resulting from added conservatism.

4.4 Case Study in Safe Autonomous Driving and Obstacle Avoidance

In this section, we implement our algorithmic architecture to safely learn to drive a vehicle while avoiding obstacles. This learning occurs within the same design as our battery case study, namely we begin with zero model knowledge and only a single known safe control input. We fit a data-driven model to the dynamics and conduct receding-horizon control.

This study is designed with specific decisions in mind to more effectively reveal the efficacy of our algorithm. Some of these decisions make our study somewhat unrealistic insofar as they expose the agent to greater danger than necessary. Subsections VI.A and VI.B discuss these decisions in detail.

Dynamical Simulator

In this case study, we utilize a bicycle model for the vehicle dynamics. This environment is encoded in the following equations discretized via forward Euler approximation:

$$x_{1;t+1} = x_{1;t} + \Delta t(x_{4;t} \cos(x_{3;t})) \tag{4.23}$$

$$x_{2;t+1} = x_{2;t} + \Delta t(x_{4;t} \sin(x_{3;t})) \tag{4.24}$$

$$x_{3;t+1} = x_{3;t} + \Delta t \left(x_{4;t} \frac{\tan(u_{2;t})}{L} \right) \tag{4.25}$$

$$x_{4;t+1} = x_{4;t} + \Delta t(u_{1;t}). \tag{4.26}$$

where t is the current timestep, x_1 and x_2 are the x-y position of the vehicle, x_3 is the vehicle heading angle, x_4 is the vehicle velocity, u_1 is the acceleration input (in $\frac{m}{s^2}$), and u_2 is the steering angle input in radians. These equations represent the true plant, which is unknown to our learning-based controller.

Model Predictive Control Formulation

We utilize the following formulation of simple autonomous driving with obstacle avoidance:

$$\underset{u_k \in \mathcal{U}}{\text{minimize}} - (x_1(t+N) + x_2(t+N)) \tag{4.27}$$

$$\text{s. to:} \quad (4.23) - (4.26), \quad x(0) = x(t) \tag{4.28}$$

$$Z(x_k) \leq Z_{cutoff}, \quad u_{min} \leq u_k \leq u_{max} \tag{4.29}$$

Here, $Z(x_k)$ is the obstacle function and can be thought of as a simple vision system. We limit Z to be smaller than a specified value (corresponding to the definition of the edge of the obstacle). Residuals in the DRO algorithm are with respect to this barrier using predicted values of the dynamical state, as opposed to the value of the obstacle function obtained with the true state. We create the environment defined by $Z(x_k)$ by generating and summing random Gaussians in 2 dimensions. Then, we define the obstacle boundaries by setting a threshold within the static map, below which becomes the safe region and above which the obstacles inhabit. This map is used with interpolation during the final experiment. If the constraint is violated, the agent will take actions which minimize violation until feasibility is restored. We set $u_{min} = [-1, -0.75]$, $u_{max} = -u_{min}$. The experiment ends once the vehicle leaves the 100 × 100 meter space.

With the learned neural network dynamics models, the MPC formulation in (4.27-4.29) becomes:

$$\underset{u_k \in \mathcal{U}}{\text{minimize}} - (\hat{x}_1(t+N) + \hat{x}_2(t+N)) \tag{4.30}$$

$$\text{s. to:} \quad \hat{x}_{k+1} = f^{NN}(x_k, u_k, \theta) \tag{4.31}$$

$$\hat{x}(0) = x(t) \tag{4.32}$$

$$Z(\hat{x}_k) \leq Z_{cutoff} - r_{DRO} \tag{4.33}$$

$$u_{min} \leq u_k \leq u_{max} \tag{4.34}$$

Table 4.3: Relevant Parameters For Obstacle Avoidance Case Study

Parameter	Description	Value	Units
L	Vehicle Length	0.5	[m]
Δt	Timestep	0.2	[s]
N_{targ}	Max Control Horizon	12	[-]
η	Risk Metric	0.005	[-]
β	Ambiguity Metric	0.99	[-]
$x_1(0)$	Initial x-position	5	[m]
$x_2(0)$	Initial Y-position	10	[m]
$x_3(0)$	Initial vehicle angle	$\frac{\pi}{4}$	[rad]
$x_4(0)$	Initial velocity	0.5	[m/s]

Table 4.4: Safety comparison for DRO-MPC and MPC with vehicle obstacle avoidance. Vio. stands for violations. The max violation is in terms of the Euclidean distance. The numbers in parenthesis are the total number of timesteps where constraints are violated, with the denominator being the number of timesteps before the vehicle leaves the 100 × 100 sized environment.

Run	Vio. (DRO)	Max Vio. (DRO) [m]	Vio. (no DRO)	Max Vio. (no DRO) [m]
1	0% (0/156)	0	2.05 % (3/146)	0.3877
2	0 % (0/145)	0	0.65 % (1/155)	0.0121
3	0.6% (1/174)	0.0386	3.47 % (5/144)	0.4472
4	0 % (0/184)	0	7.94 % (17/214)	0.9986
5	0 % (0/167)	0	1.12 % (2/179)	0.1897
6	0 % (0/140)	0	8.55 % (23/269)	2.6259
7	0 % (0/148)	0	6.74 % (13/193)	1.6726
8	0 % (0/143)	0	4.73 % (8/169)	0.2581
9	0 % (0/182)	0	10.27 % (23/224)	1.1720
10	0 % (0/165)	0	1.14 % (2/175)	0.1772
Avg.	0.0623%	0.00386	5.193 %	0.8041

Table 4.4 includes relevant parameters of our case study design. In this case study, we simply use 1-step residuals by relying on a basic assumption that the modeling error is uncorrelated to the depth of prediction. Based on our experiments, this assumption is reasonable.

We make a deliberate choice for this objective function for a host of reasons. While it necessarily encodes our intended behavior, it also is simple and at odds with the objective of avoiding obstacles. By allowing our simple objective function to drive the vehicle directly towards the obstacles, our control algorithm must be capable of managing the vehicle while simultaneously maintaining safety throughout the experiment. Thus, this case study is

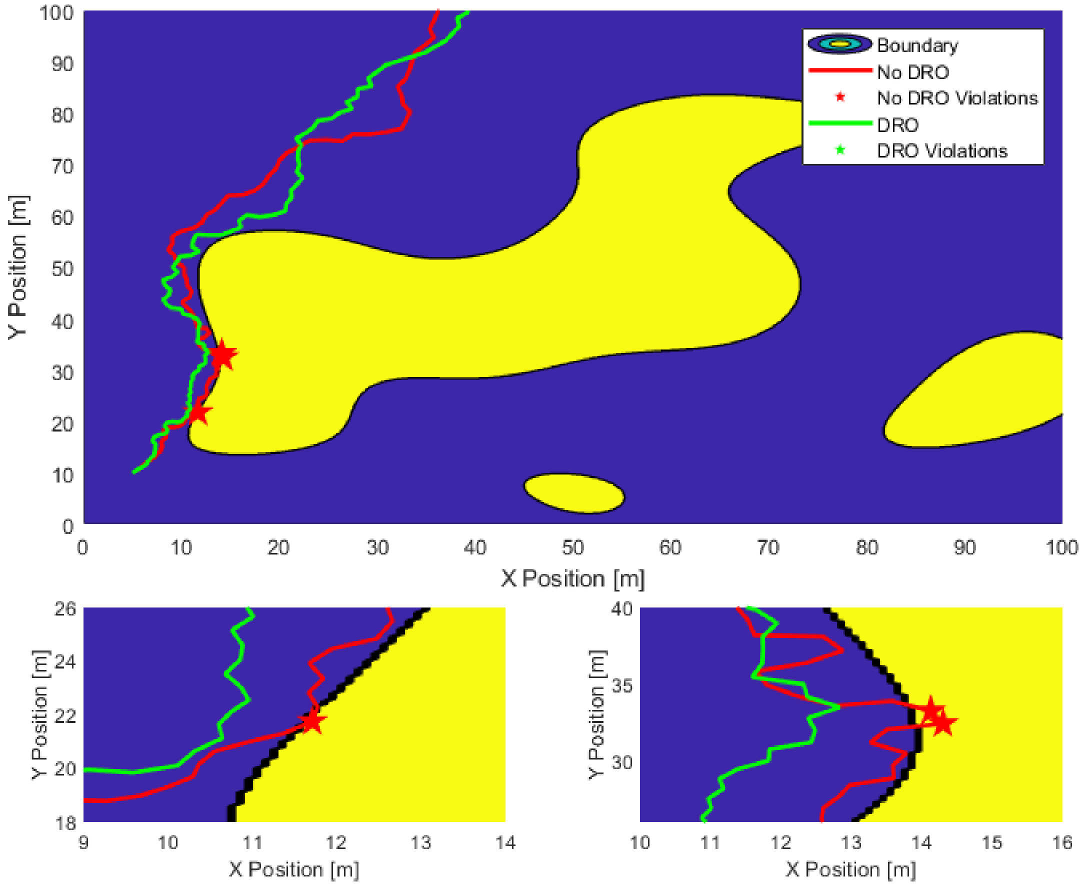

Figure 4.5: Comparison of nonlinear MPC Controller with and without DRO for vehicle obstacle avoidance. In this run, the DRO controller does not violate the constraints at all. This figure shows run 1, with the bottom plots revealing close ups of the areas with the highest constraint violation.

designed to specifically focus on the added safety contributions from the DRO framework.

For our learned model, we initialize a feed forward neural network based on a single hidden layer with 10 neurons. The hidden layer uses sigmoid activation functions, and the output layer uses linear activation. At the first timestep, we assume control inputs of a zero vector are known to be safe. To solve the MPC problem, we use the same $(1+\lambda)$ evolutionary strategy used in our battery case study. In this case, we modify the optimization algorithm such that we utilize 750,000 mutants. We also increase the maximum prediction horizon to $N_{max} = 12$ to improve the consistency of our results.

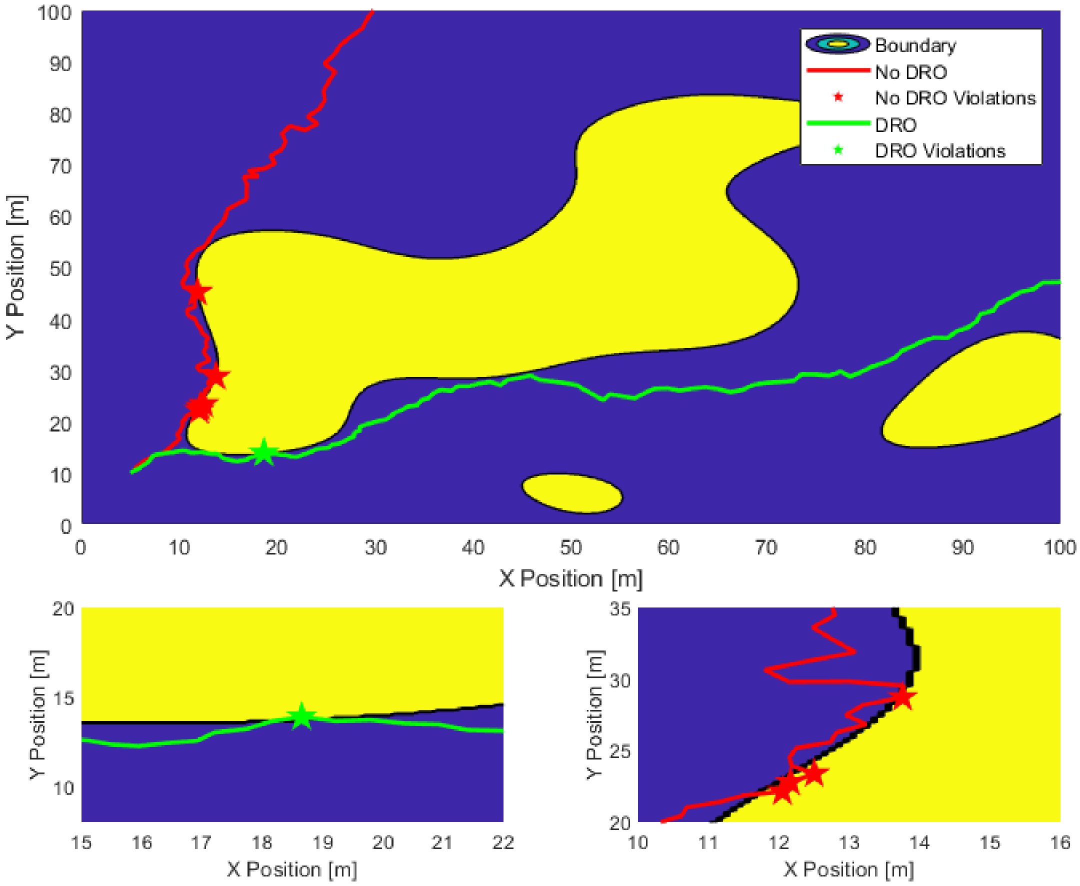

Figure 4.6: Comparison of nonlinear MPC Controller with and without DRO for vehicle obstacle avoidance. This figure shows run 3, with the bottom plots revealing close ups of the areas with the highest constraint violation.

Results

We conduct 10 individual runs with both our algorithm and a non-robust version. Figures 4.5 and 4.6 show runs 1 and 3, respectively. Table 4.4 shows safety statistics. With the DRO controller, only 1 of the 10 total runs violates constraints at all and only during a single timestep. The overall violation with the DRO controller is 0.0623% of timesteps. Moreover, the magnitude of the violation with the DRO controller is equivalent to the vehicle skimming the edge of the boundary by less than 0.0386 meters. Conversely, the non robust controller shows significant constraint violation in nearly all 10 runs. The constraint violation of the non robust controller averages 0.8041 meters of violation, which represents a complete collision with the obstacle (given our vehicle length $L = 0.5$). In one run, the non robust controller drives the vehicle nearly 3 meters into the boundary before correcting and exiting the unsafe region. To verify the model is operating in nonlinear state space, Figure 4.7 shows the range of the variable x_3 in run 1.

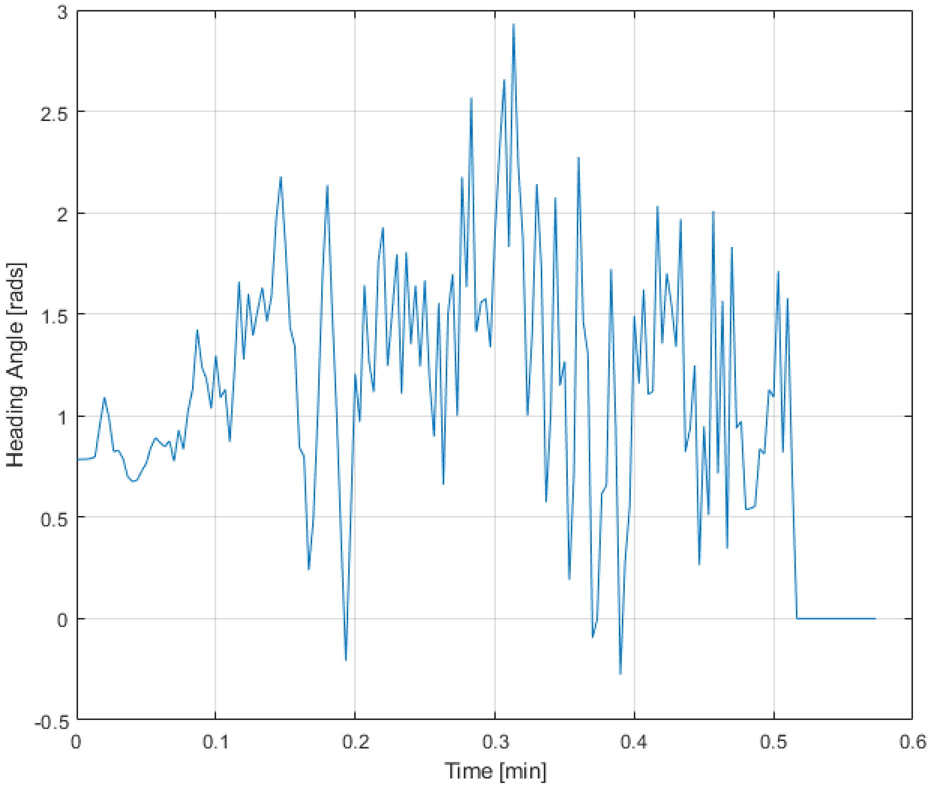

Figure 4.7: Heading angle trajectory for run 1 (same as that shown in Figure 6). The total range of heading angles is nearly π, showing exploration of highly nonlinear portions of the state space. The feasible range of steering angle input also covers a range of nonlinear behavior in the dynamics.

4.5 Discussion

Perhaps the most important available insight is that for an application, the least amount of SME needed for synthesizing safe data-driven control is tied to the minimum amount of SME that yields a DRO offset that admits a feasible solution.

We have not only explored the behavior of our algorithm at the boundary of available knowledge and data, but have validated its theoretical safety under a challenging arena of its applicability. Importantly, our approach is widely relevant in many LbC contexts (and for uncertainty quantification beyond control). For real-world applications, we are unlikely to conduct this restrictive type of *tabula-rasa* LbC. However, the same safety guarantees we have rigorously validated in these case studies are similarly applicable when more data and knowledge is available (e.g. conventional adaptive control, but with the modeling capacity of nonlinear machine-learning models). Since our approach functions as an end-to-end LbC method, it is also amenable to more unconventional applications including control synthesis from images or multimodal inputs [64]. We relegate exploration of this topic to future work.

4.6 Conclusion

This chapter presents an end-to-end distributionally robust model-based control algorithm. It addresses the problem of safety during learning-based control with strong limitations on our available knowledge and subject matter expertise. We adopt a stochastic MPC formulation where we augment constraints with random variables corresponding to empirical distributions of modeling residuals. By applying Wasserstein ambiguity sets to optimize over the worst-case modeling error, we translate an out-of-sample safety guarantee subject to new data and experience. We validate this finding through simulation experiments. Our method is applicable to nonlinear MPC, but when applying to convex MPC programs it preserves convexity.

Chapter 5

Safe Wasserstein Constrained Deep Q-Learning[1]

Abstract

This chapter presents a distributionally robust Q-Learning algorithm (DrQ) which leverages Wasserstein ambiguity sets to provide idealistic probabilistic out-of-sample safety guarantees during online learning. We illustrate that these idealistic certificates translate to imporved observations of safety in two BMS case studies. First, we follow past work by separating the constraint functions from the principal objective to create a hierarchy of machines which estimate the feasible state-action space within the constrained Markov decision process (CMDP). DrQ works within this framework by augmenting constraint costs with tightening offset variables obtained through Wasserstein distributionally robust optimization (DRO). These offset variables correspond to worst-case distributions of modeling error characterized by the TD-errors of the constraint Q-functions. This procedure allows us to safely approach the nominal constraint boundaries. Using a case study of lithium-ion battery fast charging, we explore how idealistic safety guarantees translate to generally improved safety relative to conventional methods.

5.1 Introduction

This chapter presents an algorithmic framework for improving safety with deep Q-learning.

Safe RL is the study of reinforcement learning with safety constraints. The question of safety during online learning and control - especially with respect to out of distribution (OOD) experience and data - poses perhaps the greatest open challenge to ongoing RL research. Consider that generally, the only ostensible way to learn what behavior is safe and

[1] This chapter is adapted from work that has appeared in a preprint [51]. Reprinted, with permission, from Kandel, Aaron and Moura, Scott. “Safe Wasserstein Constrained Deep Q-Learning.” (2020).

unsafe is by acting in an unsafe manner. This poses an incredible challenge for safety-critical applications where observations of unsafe behavior could involve human injuries or worse.

Recently, [41] organize safe RL research into two primary categories. The first category modifies the optimization criterion for the underlying control problem. The second category modifies the fundamental exploration process itself using either (1) external knowledge, or (2) a risk metric. Take, for example, conventional Q-learning with an ϵ - greedy exploration policy. In this case, there is no direct method to ensure constraint satisfaction when taking exploratory actions randomly. This is a problem safe RL research seeks to address when modifying the overall exploration process. A common approach is to use external information to guide the exploration of an RL agent. For instance, Mann et al. avoid random exploration by guiding exploration via transfer learning with an intertask mapping [73]. Other work addresses safe exploration using prior information about the application (e.g. a model) [71, 70, 78]. More recently, [58] use predefined safe baseline policies as an initialization for online learning. Incorporating *a priori* information into the exploration process is frequently coupled with a model-based RL approach [38].

The exploration process can also be modified using risk criteria obtained during learning. Law et al. presented early work addressing this approach, which defines a flexible risk heuristic that motivates RL agent exploration [62]. Perkins et al. address this problem by restricting the policy space based on improving identified Lyapunov functions for RL control [93]. Another risk-criterion based approach can be found in work by Gehring et al. which guides exploration via a controllability metric that represents confidence in the result of taking an action at a given state. In this work, Gehring et al. utilize the TD error given by the objective Q-function for a given state-action pair to quantify confidence in the result from that state-action pair. They show empirically that weighting this TD error in the action selection process can improve safety [42]. Our proposed safe RL algorithm similarly uses TD errors, as detailed later.

Guiding exploration can also be done based on learning safe regions. For instance, Koller et al. present an approach for learning-based model-predictive control which guarantees the existence of feasible return trajectories to a defined safe region with high-probability [57]. Other work by Richards et al. constructs a neural network Lyapunov function in order to learn safe regions for nonlinear dynamic systems [97]. Berkenkamp et al. also leverage Lyapunov stability to establish specific metrics of safety for an RL controller [8].

Recently, ideas from the literature on constrained Markov decision processes (CMDPs) have begun migrating into relevant RL research. Simply put, CMDPs are MDPs where the policy space is limited by constraints imposed on auxiliary cost functions. See work by Altman for more discussion of their specific formulation [4]. Q-learning has been applied to solve CMDPs in the past, however existing works re-frame the problem using the assumption of strong duality [29]. In most common use cases, however, Slater's constraint qualification condition rarely holds, making this approach difficult to effectively implement. The general concept of constraint costs has been applied in recent papers on the subject of safe RL [25, 1]. Chow et al. present an algorithm reminiscent of past work by [87] which defines the feasible action space for stationary deterministic MDPs with respect to such constraint cost estimates, improving adherence to constraints. They use resource constraints in the MDP formulation to act in a similar manner as a shield [3]. However, the certificates of their algorithm ostensibly depend on an assumption that the convergence of the reward and constraint Q-functions occurs on separate timescales. Their formulation is also sensitive to noisy observations, and has yet to be explored with function approximation.

While these approaches all improve safety during online learning, they still share the exact same shortcoming which remains the strongest motivating force behind this area of literature. Namely, without *a priori* information about the underlying environment it is impossible to act in a safe manner without violating constraints to some degree. In exploring this open question, this chapter takes motivation from literature on robust model-predictive control (MPC), where the concept of "constraint tightening" has become fairly popular over the past decade. We presents a novel robust approach to safely solving constrained RL problems. Our work is roughly inspired by the motivating idea of constraint tightening literature, as well as the ideas presented by [87]) regarding hierarchies of machines in RL problems. We use the methodology of [25] as a simple foundation upon which we build DrQ, a novel framework for safe RL. DrQ is an algorithmic framework for safe deep Q-learning which leverages Wasserstein ambiguity sets to enforce safety constraints. Specifically, we follow [25] by separating consideration of constraints to their own constraint cost functions. These cost functions define the feasible action space within which DrQ operates. Importantly, our DrQ algorithm leverages a novel formulation for pulling the nominal constraint boundary into the safe region, based on worst-case distributions of modeling error. These distributions are characterized by observed TD errors of the underlying constraint cost functions. By presenting a disciplined Wasserstein DRO-based method for recessing the constraint boundary into the safe region, DrQ observes and reacts to unsafe behavior before nominal constraints are violated. Our algorithm yields probabilistic safety guarantees under idealistic circumstances which arise from past theoretical work on Wasserstein ambiguity sets [34, 40, 33]. As the constraint cost models improve, the constraint offset naturally tightens towards the nominal boundary. Our case studies in safe lithium-ion battery fast charging demonstrates the strong propensity of DrQ to translate these theoretical safety certificates directly to improving safety during exploration and exploitation in more nuanced, real-world RL problems.

5.2 Distributionally Robust Q-Learning

The principal tools leveraged in DrQ are CMDPs and Wasserstein ambiguity sets. Chapter 2 discusses the relevant background and results for Wasserstein ambiguity sets. Subsection 5.2 includes additional relevant background on constrained MDPs before outlining the DrQ algorithmic structure.

Constrained MDPs

Constrained Markov decision processes are identical to MDPs except that additional cumulative costs are used to restrict the space of feasible control policies. We direct the reader to [4] for further reading on the subject. The feasible set of control policies is defined as:

$$\Pi_{feas} = \{\pi \in \Pi : \forall i, D_i^{\pi} \leq 0\} \tag{5.1}$$

where D_i are cumulative constraint cost functions (henceforth referred to as constraint Q-functions) developed subject to the policy defined by Q:

$$\pi^* = \underset{\pi \in \Pi_{feas}}{\operatorname{argmax}} \quad Q \tag{5.2}$$

where

$$Q(s_t, a_t) = r_t(s_t, a_t) + \mathbb{E}_{s_{t+1}}[\gamma \max_{a \in \mathcal{A}_{feas}(s_{t+1})} Q(s_{t+1}, a)] \tag{5.3}$$

$$D_i(s_t, a_t) = c_i(s_t, a_t) + \mathbb{E}_{s_{t+1}}[D_i(s_{t+1}, a^* = \underset{a \in \mathcal{A}_{feas}(s_{t+1})}{\operatorname{argmax}} \; Q(s_{t+1}, a))] \tag{5.4}$$

$$\mathcal{A}_{feas}(s) = \{a \in \mathcal{A} \mid D_i(s, a) \leq 0 \, \forall \, i = 1, ..., m\} \, . \tag{5.5}$$

DrQ is inspired by [87], which discusses hierarchies of machines in RL problems. More recently, a similar approach for CMDPs was presented by [25], given the title of “Two-Phase” Q-learning. In “Two-Phase” Q-learning, the objective and constraint Q-functions are learned online while limiting the feasible space based on estimates of the constraint cost functions. This approach, however, still requires we experience unsafe states in order to gradually learn safe behavior. Their algorithm is also sensitive to noise, only works for deterministic MDPs, and has yet to be explored with deep function approximation. In the following sections, we will lay groundwork for DrQ, which ameliorates the shortcomings of existing value-based approaches.

Distributionally Robust Q-Learning Algorithm

Consider that the primary problem in constrained RL is that we generally cannot plan avoidance of unsafe states without first acquiring some experience of which states are themselves unsafe. For conventional algorithms, this “*chicken and egg*” problem means the first step to learning safe control is to *violate* constraints. As a result, conventional algorithms are by nature incompatible with the principal objective of constrained RL. In order to address this challenge, we look to the control theoretic literature for potential solutions. In research on model-predictive control (MPC), the concept of “constraint tightening” is a popular method when implementing adaptive predictive controllers. This field uses heuristics or analytical methods to adapt constraints subject to the uncertainty of the nominal model. As the uncertainty of the nominal model decreases, the constraint boundaries will safely approach nominal boundaries. The idea of constraint tightening immediately radiates potential for RL algorithms. By shifting the constraint boundary into the safe region, **we can experience artificially unsafe states long before the actual safety of the underlying system comes into question**. This emulates the same logic used in Chapter 4 to learn safe behaviors using MPC. Motivated by this insight, we present an approximate constraint tightening methodology for deep Q-Learning which can idealistically provide strong probabilistic guarantees on safety, which in practice we show to generally improve overall constraint satisfaction.

From this point forward, we consider cases of deep reinforcement learning (i.e. each value function is learned via a parameterization). First, we limit our algorithm to solving optimal control problems subject to inequality constraints indexed by i, $g_i(s_t, a_t) \leq 0$. The cost functions associated with these constraints take the form:

$$c_i(s_t, a_t) = \begin{cases} 0 & \text{if } g_i(s_t, a_t) \leq 0 \\ g_i(s_t, a_t) & \text{else} \end{cases} \tag{5.6}$$

This is a key definition, showing we define constraint costs D_i as measures of **cumulative constraint violation**. Furthermore, it allows us to uncouple the updates between Q and D_i using the following D_i target:

$$D_i(s_t, a_t) = c_i(s_t, a_t) + \min_{a \in \mathcal{A}_{feas}(s_{t+1})} D_i(s_{t+1}, a) \tag{5.7}$$

By updating D_i with its own Bellman equation, we convert the constraint to its best-case counterpart. This means that D_i represents the cumulative constraint cost acquired with the safest possible policy. **Since any positive signal in D_i indicates infeasibility, we can make this change**. This uncoupling allows us to learn Q and D_i without timescale separation between their respective learning processes. The proof of convergence of the original two-phase Q-learning algorithm in [25]) ostensibly depended on this timescale separation assumption, including for more general problems which do not satisfy (5.6).

Remark 11 *We can also apply a tolerance when updating D_i to allow constraint violation to propagate backwards throughout the model.*

In order for our framework to be consistent with the ideas motivating constraint tightening approaches, we can introduce an offset variable to each constraint cost as follows:

$$c_i(s_t, a_t) = \begin{cases} 0 & \text{if } g_i(s_t, a_t) \leq -q_i \\ (g_i(s_t, a_t) + q_i) & \text{else} \end{cases} \tag{5.8}$$

This formulation begs the questions of how we set and update the value of offset variable q_i in real time. For these questions, we consider ways to characterize the modeling error of the constraint Q-functions. Ideally, we want to offset the constraint boundary proportionally to the worst-case error of our models. For this consideration, we can use TD error distribution for each constraint Q-function. Past work by [42]) has utilized TD errors in Q-learning as indicators of our confidence in the underlying model. Consider the TD error defined for the ith constraint Q-function (which unlike the objective Q-function is solving a minimization problem):

$$\delta_{D_i}(s_t, a_t) = c_i(s_t, a_t) + \min_{a \in \mathcal{A}_{feas}} \gamma D_i(s_{t+1}, a) - D_i(s_t, a_t) \tag{5.9}$$

This TD error δ_{D_i} is a modeling residual by definition, but it is not perfect. It represents how much our model of the cumulative constraint cost changes after a parameter update. The TD error provides a good indication of modeling error, but since it does not exactly represent that quantity our theoretical guarantees do not exactly translate to real applications. But we show later in this chapter that DrQ architecture still manages to dramatically improve safety during online learning. Our best indication of such model error can be obtained by using the distribution of TD errors computed after each update to describe the error inherent to our function approximator for each D_i, we

Now we can delve into the fundamental basis for our algorithmic architecture. DrQ works by defining the offset variables q_i, one for each inequality constraint, through an equivalent reformulation of the following distributionally robust chance constraint:

$$\inf_{\mathbb{P} \in \mathbb{B}_\epsilon} \mathbb{P}[D_i(s_t, a_t) + \mathbf{R}_i \leq 0] \geq 1 - \eta \tag{5.10}$$

where $\mathbb{B}_\epsilon$ defines the Wasserstein ambiguity set, $\mathbf{R}_i$ represents the realization of the TD error of the ith constraint Q-function, and η is our allowed probability of violating the constraint. We can interpret this constraint as follows: we have an empirical distribution of TD errors which we compute after each parameter update of D_i. We want to satisfy the constraint $D_i(s_t, a_t) + \mathbf{R}_i \leq 0$ for the worst-case realization of TD error $\mathbf{R}_i$ sourced from a family of probability distributions centered about our empirical distribution. This set of distributions is within ϵ distance of the empirical distribution, with the expression for ϵ given by (2.6). The reformulation we select for our algorithmic architecture comes from [33]), and yields the constant q_i that we augment to our ith constraint cost function. Thus, the greedy action selection process becomes:

$$a_t^* = \underset{a \in \mathcal{A}_{feas}(s_t)}{\operatorname{argmax}} \quad Q(s_t, a) \tag{5.11}$$

where

$$\mathcal{A}_{feas}(s) = \{a \in \mathcal{A} \mid D_i(s, a) \leq r_j \, \forall \, i = 1, ..., m\} \tag{5.12}$$

can be used to limit the feasible action space for exploration and exploitation. In order to evolve the offset q_i as our TD distributions change over time, we store the tuple $(s_t, a_t, s_{t+1}, g_i(s_t, a_t))$. Then, each time we prepare to update the D_i functions we recompute the unique values of the constraint cost function as per (5.19) based on our most recent q_i. This formulation mathematically encodes that we seek to satisfy the constraint subject to the addition of the potential worst-case modeling error. As our model improves, we can approach the constraint in a provably safe manner consistent with the theoretical guarantees afforded Wasserstein ambiguity sets.

Algorithm 3 DrQ Algorithm ($\epsilon - greedy$)

```
Require: State space $\mathcal{S}$, Action space $\mathcal{A}$, Reward $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, Constraint cost $\mathcal{R}_{C;i} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, i = 1, ..., n$, State transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, Initialize Q-functions $Q, D_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
  Conduct first episode with vanilla $\epsilon$-greedy, Store tuples of $(s_t, a_t, s_{t+1}, g_i(s_t, a_t))$
  Initialize $q_i = D_{\Xi_i}$; Fit $Q(s_t, a_t)$ and $D_i(s_t, a_t)$; store TD-errors from fitting $D_i$
  for $k$ in range episodes do
    Initialize state $s \in \mathcal{S}$
    for $j$ in range iterations do
      $\mathcal{A}_{feas}(s) = \{a \in \mathcal{A} \mid D_i(s, a) \leq 0 \; \forall i = 1, ..., n\}$
      if $|\mathcal{A}_{feas}(s)| = 0$ then
        $\mathcal{A}_{feas}(s) = \underset{a \in \mathcal{A}}{\text{argmin}} ||D(s, a)||$
      end if
      Compute $q_i$ for each $D_i$ based on TD-error distribution
      if exploring then
        Pick random action $a = a_{rand} \in \mathcal{A}_{feas}(s)$
      else
        $a \leftarrow \underset{a_m \in \mathcal{A}_{feas}(s)}{\text{argmax}} \; Q(s, a_m)$
        Store tuples $(s_t, a_t, s_{t+1}, g(s_t, a_t))$; Fit $Q(s_t, a_t)$ and $D_i(s_t, a_t)$; store TD-errors of $D_i$
      end if
    end for
  end for
```

Algorithm 3 describes the implementation of DrQ. We opt for a fitted Q-iteration method for deep Q-learning. The next section details a high level conceptual example of how DrQ works, which leads into additional discussion and finally our case studies.

- Can our algorithm accommodate nonstationary MDPs? -*No*
- Can our algorithm accommodate probabilistic MDPs? -*Yes*
- Could q_i stabilize prematurely, removing some safe states from future exploration? - *Yes, but only under very specific conditions on the measurement noise or function approximator.*

5.3 Conceptual Graphical Example of DrQ

In this appendix, we will walk through a complete episodic sequence of DrQ for the following simple optimal control problem:

$$\begin{aligned}
&\underset{\vec{a}\in A\in R^2}{max} && \sum_{k=0}^{N} r_k(s(k),a(k)) && (5.13)\\
&\text{s. to:} && s(k+1)=f(s(k),a(k))\ (unknown) && (5.14)\\
& && g_1(s(k),a(k))\le 0 && (5.15)\\
& && a_1\in\{-3,-2,-1,0,1,2,3\} && (5.16)\\
& && a_2\in\{-5,-4,-3,-2,-1,0,1,2,3,4,5\} && (5.17)\\
& && s(0)=s_0 && (5.18)
\end{aligned}$$

where the unknown state transition function $f(s(k),a(k))$ can be either deterministic or probabilistic.

Episode 1

The safety guarantees of DrQ become active after our first parameter update of the D_i functions. Therefore, for the first episode of learning, we operate similarly to conventional deep Q-learning while simultaneously recording constraint violation subject to an artificial sunken constraint boundary. For the first episode, the offset $q^{(i)}$ can be set as a hyperparameter given any understanding of the scale of the constraint functions and how close they may be to the nominal boundary. Our objective for the first episode is to observe violation of the offset while maintaining safety relative to the actual constraint boundary. A priori knowledge of the constraints can be applied to initialize the constraint offset.

For the first episode, with random initialization of the functions Q and D_1, we essentially act randomly while recording values of r and g_1. Then at the end, when we apply our first parameter update to the function approximators, the constraint costs become:

$$c_1(s_t,a_t)=\begin{cases}0 & \text{if } g_1(s_t,a_t)\le -q_1(0)\\ g_1(s_t,a_t)+q_1(0) & \text{else}\end{cases} \tag{5.19}$$

and we update Q and D_1 according to the following targets:

$$Q(s_t,a_t)=r_t(s_t,a_t)+[\gamma \underset{a\in A_{feas}(s_{t+1})}{max} Q(s_{t+1},a)] \tag{5.20}$$

$$D_1(s_t,a_t)=c_1(s_t,a_t)+D_1(s_{t+1},a^*) \tag{5.21}$$

where $a^*=\underset{a\in A_{feas}(s_{t+1})}{argmin} D_1(s_{t+1},a)$. We update D_1 first so we can superimpose the latest estimate of the feasible set $\mathcal{A}_{feas}$ on our update of Q.

Once we have updated the parameters of D_1, we compute the TD errors of our new model as follows:

$$\delta_{D_1}(s_t,a_t)=c_1(s_t,a_t)+\underset{a\in A_{feas}}{min}\gamma D_1(s_{t+1},a)-D_1(s_t,a_t) \tag{5.22}$$

We use this TD error distribution to recompute the value of q_1 for the next episode. Then, for the next episode, we observe constraint violation subject to the modified boundary given by q_1.

Episode 2

Once episode 2 begins, the real advantages of DrQ begin to manifest. To demonstrate, lets use a graphical example. Suppose we are at state s_{test} at the beginning of episode 2. After our parameter updates to D_1, we can plot in 3D the relationship between D_1 and the potential actions we can take in $\mathbb{R}^2$ for fixed state. Suppose this plot takes the form shown in Figure 5.1. Here, the action pairs within the black square correspond to the true (unknown) feasible actions. Since q_i is nonzero, our estimated feasible action space is smaller than the true space, resulting from our offset of the constraint boundary into the safe region. From this plot, we can easily deduce the feasible pairs of a_1 and a_2 as those with value of D_1 equal to zero (dark purple). We can superimpose this set onto our plot of Q to get the graphic in Figure 5.2. If we are exploring, then we can pick any action within this feasible set. If we are exploiting, we choose the pair (a_1, a_2) which maximizes Q along the restricted domain defined by $\mathcal{A}_{feas}$, which is computed from the data visualized in Figure 5.1, which in this case turns out to be $(-1, -3)$. This action is safe, but conservative. As the offset q_i progressively tightens, the action DrQ selects will slowly yield improved performance without sacrificing safety.

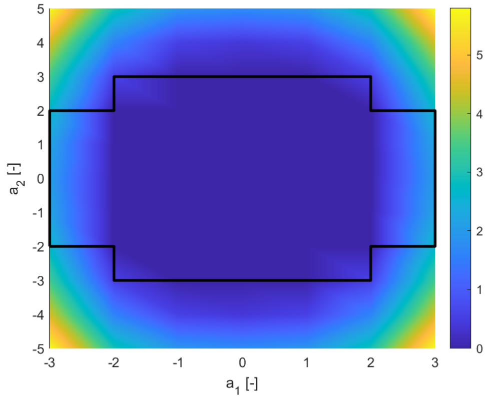

Figure 5.1: Plot of D_1 for total action space

Now, at the end of episode 2, we can use the new data to further update the parameterizations of the function approximators Q and D_1. With the new TD errors of D_1, we recompute the DRO offset q_1 and continue this process until the desired control performance is attained.

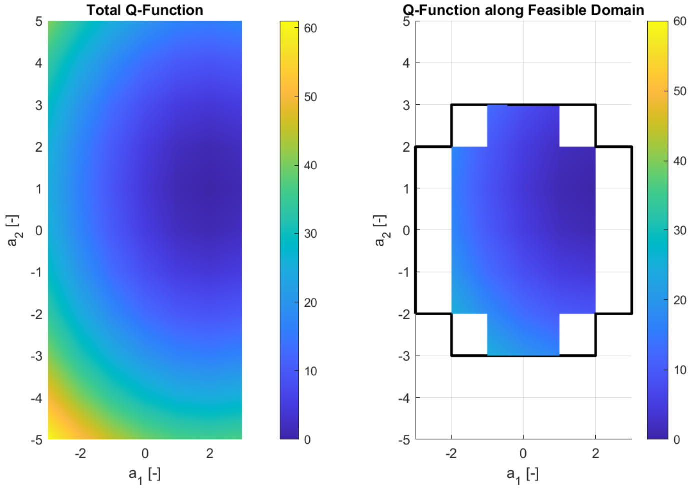

Figure 5.2: Plot of Q with superimposed feasible action space

Episode k

Suppose arbitrary number of episodes have passed, and we are now at an episode "k." After these episodes, the TD errors of D_1 have reduced in magnitude given the assumption that our model D_1 has improved. Given no measurement noise, we assume that our TD error distribution is predominantly centered about zero. Now, at the same test state s_{test} at the beginning of the episode, we evaluate the potential actions to give the following plot of D_1: Here, our estimated safe set is the same as the true safe set. Since our offset q_i is derived

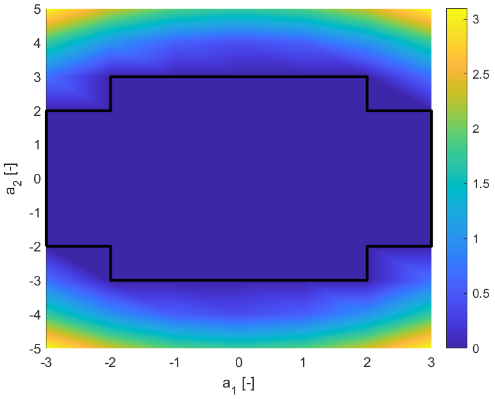

Figure 5.3: Plot of D_1 for total action space

from the TD-error distribution (which is predominantly zero given a converged function approximator), q_i=0 and we have approached the true constraint boundary. Thus, with no modeling errors in this simple conceptual example we eventually act in a truly optimal way relative to the optimal control problem statement. Here, the safe set superimposed on the objective Q-function at episode k takes the form: meaning we pick the truly optimal (and importantly nominally feasible) action $(-3, 0)$. The next section in this appendix details how to compute the DRO offset variable q_i which guides this constraint tightening procedure.

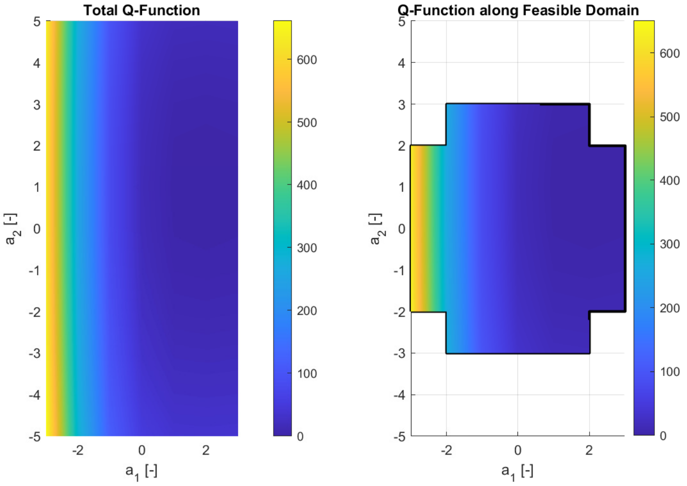

Figure 5.4: Plot of Q with superimposed feasible action space

5.4 Extended Discussion of DrQ

This appendix provides further discussion on the following questions raised in the main text:

- Can our algorithm accommodate nonstationary MDPs? -*No*
- Can our algorithm accommodate probabilistic MDPs? -*Yes*
- Could q_i stabilize prematurely, removing some safe states from future exploration? - *Yes, but only under very specific conditions on the measurement noise or function approximator*

Nonstationary MDPs

Our algorithm in its current state cannot effectively accommodate nonstationary MDPs. Consider that if the underlying dynamics change, the historical data mapping $(s_t, a_t, g_i(s_t, a_t))$ will no longer represent the actual constraint dynamics. This would cause the DRO offset

q_i to grow given growing model residuals from inconsistent historical data. Perhaps with a disciplined forgetting scheme this phenomenon could be avoided, however as of now we relegate this issue to future work.

Probabilistic MDPs

Our algorithm can accommodate probabilistic MDPs. Past work by [25]) and [88]) indicates similar algorithms cannot accommodate probabilistic MDPs without miscoordination occurring. These studies explore frameworks for multi-agent systems. When applied to single-agent systems, our algorithm avoids this shortcoming.

For the case where state transition dynamics are probabilistic, consider the definition of the functions Q and D_i. Both consider cumulative costs, which given probabilistic dynamics simply become cumulative expected costs. For D_i, any nonzero signal indicates constraint violation (subject to the offset q_i), so any state-action pair with a non-zero probabilitiy of violating constraints will eventually be pruned.

Stability of q_i

Two cases exist where the value of q_i stabilizes prematurely, removing some safe state-action pairs from future consideration. The first has to do with measurement noise. Assuming the function approximator D_i converges, if our measurements of g_i are subject to a measurement noise process, then the eventual TD error distribution of D_i will represent the underlying measurement noise process. This distribution of residuals could create a permanently nonzero q_i.

The second has to do with the properties of the function approximator. If the function approximator D_i fails to converge, then the value of q_i may not stabilize and could oscillate or diverge. Our numerical experiments have yet to show a case where this occurs, but it is a possibility for nearly any deep Q-learning algorithm. Some specific cases where this occur can be found in [68]).

5.5 Battery Fast Charging Case Study

This chapter presents two case studies on battery fast charging to validate the performance and safety of DrQ.

Equivalent Circuit Model of a Lithium-Ion Battery

This case study utilizes an equivalent circuit model of a lithium-ion battery. The relevant states in this model are the state of charge SOC and capacitor voltage V_{RC}. The relevant constraint is on the terminal voltage V. The state evolution laws are given by the following equations:

$$SOC_{t+1} = SOC_t + \frac{1}{Q} I_t \cdot \Delta t \tag{5.23}$$

$$V_{RC;t+1} = V_{RC;t} - \frac{\Delta t}{R_1 C_1} V_{RC;t} + \frac{\Delta t}{C_1} I_t \tag{5.24}$$

$$V_t = V_{OCV}(SOC_t) + V_{RC;t} + I_t R_0 \tag{5.25}$$

where I_t is the current input, and V_{OCV} is the nonlinear open-circuit voltage, which is obtained through experiments.

We utilize the following formulation of fast charging:

$$\min_{I_t \in A} \sum_{t=0}^{T} (SOC_t - SOC_{target})^2 \tag{5.26}$$

$$\text{s. to:} \quad (5.23) - (5.25), \quad SOC(0) = SOC_0 \tag{5.27}$$

$$V_t \leq 3.6V, \quad 0A \leq I_t \leq 46A \tag{5.28}$$

Table 5.1 outlines relevant parameters of the model. Figure 5.5 shows the OCV-SOC curve represented by $V_{OCV}(SOC)$.

Table 5.1: Relevant Parameters

Parameter	Description	Value	Units
Q	Charge Capacity	8280	$[\frac{1}{A.h}]$
R_1	Resistance	0.01	$[\Omega]$
C_1	Capacitance	2500	$[F]$
R_0	Resistance	0.01	$[\Omega]$
Δt	Timestep	2.5	$[s]$
γ	Discount Factor	0.5	$[-]$
α	Learning Rate	0.15	$[-]$
ϵ	Exploration Prob.	0.2	$[-]$
D_Ξ	Support Rad.	0.2	$[V]$
β	DRO Confidence	0.98	$[-]$
η	CC Confidence	0.02	$[-]$

DrQ Problem Formulation

The objective reward function for this optimal control problem takes the form:

$$r(s_t, a_t) = -(SOC_{t+1} - SOC_{target})^2 \tag{5.29}$$

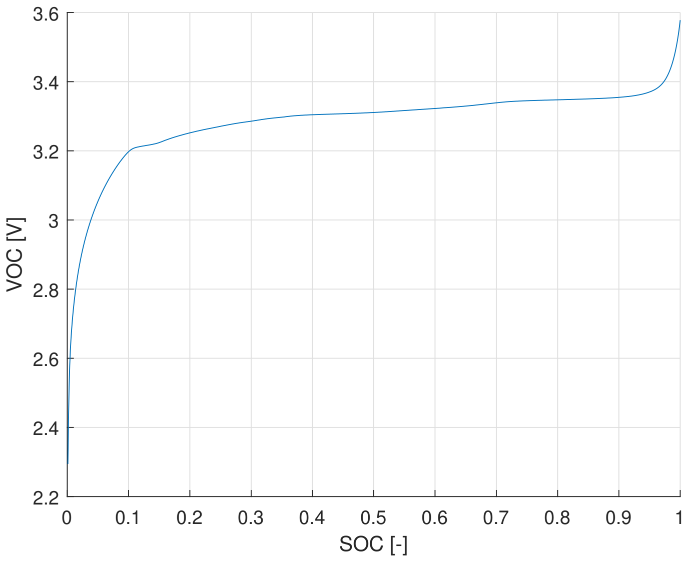

Figure 5.5: Experimental Open-Circuit Potential Function

The initial SOC in our case study is 0.2 (20% capacity), and $SOC_{target} = 0.7$ (70% capacity). The constraint penalty takes the form:

$$c = \begin{cases} 0 & (s_t, a_t) \in \mathcal{C} \\ \mid V_t - 3.6 + q \mid & (s_t, a_t) \notin \mathcal{C} \end{cases} \tag{5.30}$$

where $\mathcal{C} = \{V_t \in \mathbb{R} \mid V_t \leq 3.6 - q\}$. Our risk metric $\eta = 0.02$. For a baseline comparison, we also examine conventional deep Q-learning (DQN) with the following modified performance criterion:

$$r_{eng} = -(SOC_{t+1} - SOC_{target})^2 - \mathbb{1}(V_t > 3.6) \tag{5.31}$$

Results

We generate 10 independent runs of 25 episodes using both DQN and DrQ for this analysis. For DrQ, Q is a single hidden layer neural network with 10 neurons and sigmoid activation and D is a neural network with four hidden layers of size $(2, 5, 5, 2)$. The DQN is a neural network with two hidden layers of size $(10, 10)$. We use sigmoid activation functions for our function approximators. This demonstrates our algorithm is capable of yielding a high performing control policy which safely charges the battery. Comparatively, after 25 episodes the DQN yields a consistently unsafe control policy which overcharges the battery. In fact, analysis of our other runs indicates the DQN frequently fails to converge to any usable result entirely. Figure 1 clearly demonstrates this finding. Overall, DrQ delivers significantly more consistent and near monotonic improvements in performance, whereas the DQN shows no clear pattern of improvement after 25 episodes. DrQ also delivers tighter variance on the overall performance compared to DQN.

Figure 2, which provides our most illustrative results, displays statistics on constraint satisfaction for both DrQ and DQN throughout these 10 runs. After the first episode (where constraint satisfaction is commensurate between DrQ and DQN since we do not enforce DRO), DrQ safely learns to charge the battery by leveraging the idealized probabilistic guarantee of Wasserstein ambiguity sets. In fact, our observations of constraint violation for DrQ are entirely consistent with our chosen chance constraint risk metric $\eta = 2\%$ (see the figure caption for this analysis). Overall, only 1.25% of episodes and only 0.023% of timesteps violate constraints for DrQ. By observing the magnitude of the y-axis scale between cumulative and maximum constraint violation in Fig. 2, it is clear that DrQ also attenuates the magnitude and frequency of constraint violation in the unlikely event that violations do occur relative to DQN. In comparison, the DQN benchmark consistently violates constraints. The average computation time for each DrQ episode was 5.57 seconds, compared to 3.77 for DQN when run on a PC with a 9th generation intel i5 processor.

DQN is our comparison for several reasons. DrQ, much like DQN, can be augmented with additional and existing safe RL architecture. More importantly, our analysis is intended to quantify real-world adherence to idealized theoretical guarantees we obtain through application of Wasserstein ambiguity sets.

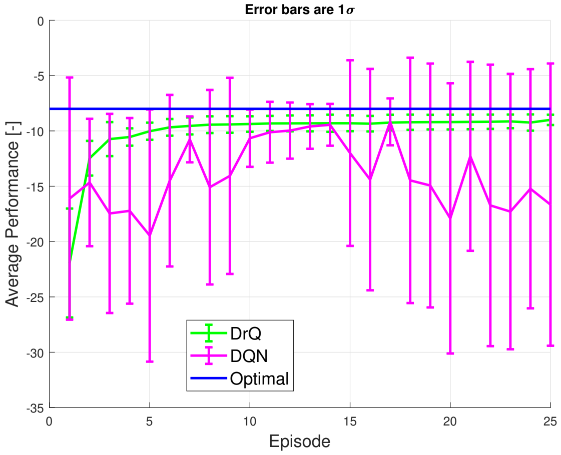

Figure 5.6: Greedy policy performance statistics over 10 runs of DrQ and DQN, based on the reward function defined by (5.40). Performance of -35 indicates no input current is applied, which occurred as the final result of 6 of the DQN runs.

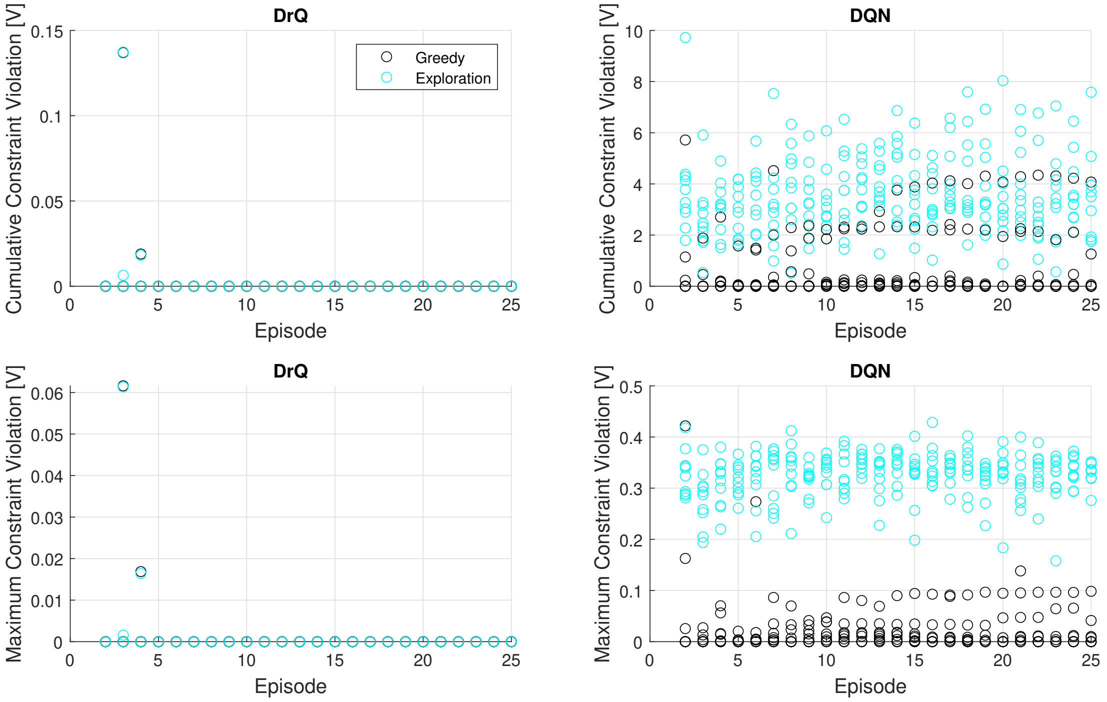

Figure 5.7: Safety statistics over 10 runs of DrQ and DQN, starting from the second episode. The black "exploration" points correspond to the data obtained from the ϵ-greedy policy. The cyan "greedy" points correspond to the greedy policy evaluated at the end of each exploratory episode. The safety observed in both exploration and exploitation with DrQ is consistent with the chance constraint risk metric $\eta = 0.02$. Out of 240 exploratory episodes (excluding the first from each run, where we do not enforce DRO), only 3 episodes exhibit constraint violation ($\frac{3}{240} = 0.0125 < \eta$).

5.6 SPMeT Case Study

In this section we detail our comprehensive case study on safety-aware learning-based fast charging control, with a large scale electrochemical battery model. The details of the single particle model with electrolyte and thermal dynamics (SPMeT) are included in the appendix of this dissertation.

Optimal Control Problem Statement

The optimal control problem statement for fast charging with SPMeT is given by:

$$\min_a J = \int_{t_0}^{t_F} \left[SOC_n(t) - SOC_{targ}\right]^2 dt \tag{5.32}$$

$$\text{s. to:} \quad (7.1) - (7.7) \tag{5.33}$$

$$s_0 = s(t) \tag{5.34}$$

$$c^k_{ss;min} \leq c^k_{ss}(t) \leq c^k_{ss;max} \quad \forall \quad k \in \{+,-\} \tag{5.35}$$

$$T_{min} \leq T(t) \leq T_{max} \tag{5.36}$$

$$c^k_{e;min} \leq c^k_e(t) \leq c^k_{e;max} \quad \forall \quad k \in \{+,-\} \tag{5.37}$$

$$\tag{5.38}$$

where SOC_n is the normalized bulk concentration in the anode, $a_t = I(t)$ is the control action, and $s_t = [c^{\pm}_s(r,t), c_e(x,t), T_{cell}(t)]$ is the state vector. We define the overall cell SOC as:

$$SOC_t = \frac{3\int_0^{R_s^-} r^2 c_s^-(r,t)dr}{(R_s^-)^3 c^-_{s,\max}|x_{100\%} - x_{0\%}|} \tag{5.39}$$

To solve this problem using reinforcemnet learning, we spatially discretize the system of PDEs to formulate a discrete-time and space model of the form $s_{t+1} = f(s_t, a_t)$. The reward function for DrQ takes the form:

$$r(s_t, a_t) = -(SOC_{t+1} - SOC_{target})^2 \tag{5.40}$$

We also compare DrQ to a conventional DQN, whose reward function takes the form:

$$r_{eng} = -(SOC_{t+1} - SOC_{target})^2 - \mathbb{1}\left[g_i(s_t, a_t) > 0\right] \tag{5.41}$$

where we apply a constant step penalty for any constraint violation.

For each relevant constraint, we define a feed forward neural network to approximate D_i with 2 hidden layers each with 10 neurons and sigmoid activation function. We approximate the objective Q-function using a network of similar architecture. Table 2 includes the hyperparameters for the overall problem. The SPMeT model we use as a simulator is parameterized for a prismatic lithium nickel manganese cobalt oxide cell, different from the lithium iron phosphate cell used in our ECM case study. The episode simulation horizon is 1400 seconds. For an optimal baseline, we use methods outlined in [92]). Figure 8 shows the baseline control result. Several meaningful insights can be taken from these baseline results. First, the anode electrolyte constraint has the potential to be violated at almost every timestep. Violation of this constraint can cause rapid aging or catastophic failure. Second, the multitude of constraints provides a stronger challenge to our algorithm.

DrQ Results

We simulate 10 independent runs of both DrQ and DQN, each with 25 episodes. Figure 5.9 shows a comparison of the performance between DrQ and DQN, averaged across the runs. We evaluate the greedy policy performance at episodes 2, 5, 10, 15, 20, and 25 for computational purposes, given the SPMeT model is numerically expensive to simulate. Relative to DrQ, the

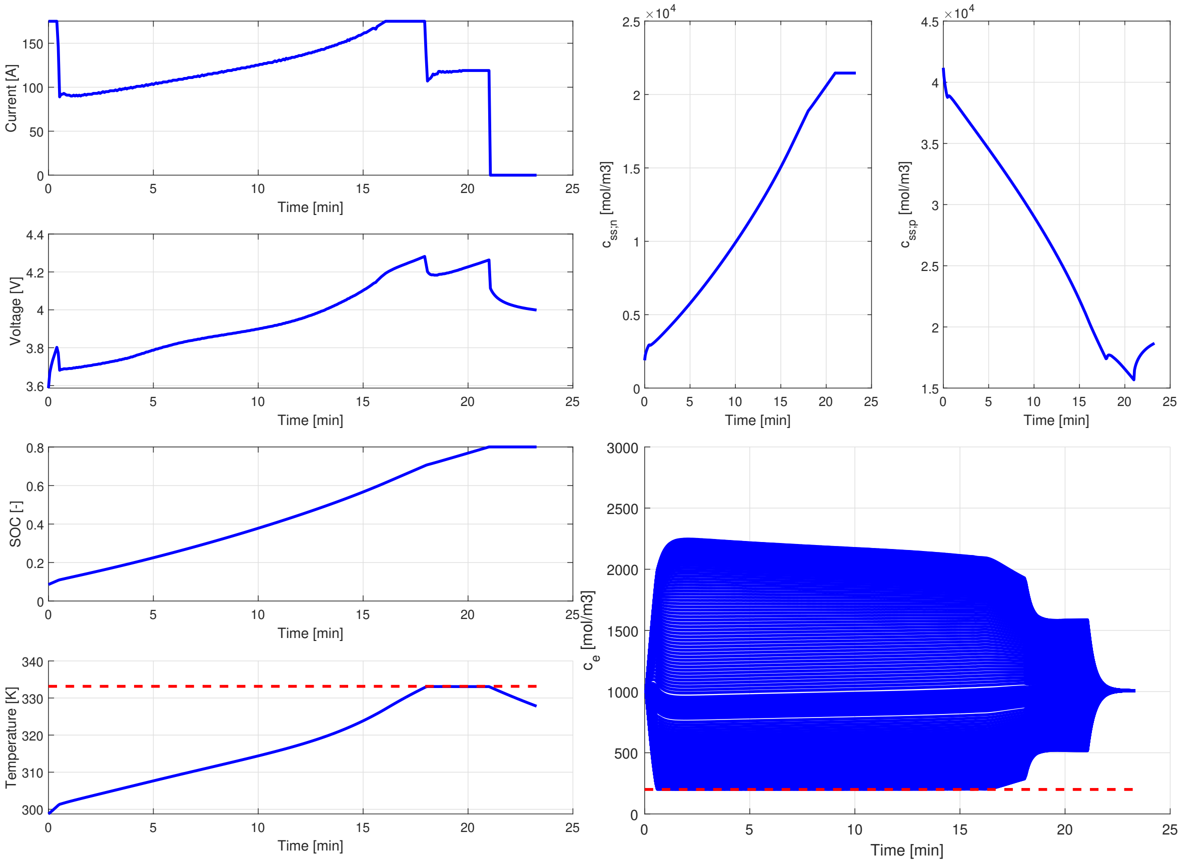

Figure 5.8: Baseline optimal control result.

Table 5.2: Relevant DrQ Parameters

Parameter	Description	Value	Units
Δt	Timestep	4	$[s]$
γ	Discount Factor	0.75	$[-]$
ϵ	Exploration Prob.	0.2	$[-]$
$D_{\Xi;i}$	Support Rad.	1	$[]$
β	DRO Confidence	0.9	$[-]$
η	CC Confidence	0.05	$[-]$

variance in the DQN performance is greater. Furthermore, the performance of DrQ is on average 31% greater compared to DQN. One important note is that, to get the DQN baseline properly running, we had to add a fail-safe which set the input current to zero if the anode electrolyte concentration c_e^- became too low. DQN tended to max out the input current in the first several episodes of each run, which would rapidly deplete the electrolyte. With a real battery cell, this would cause unsafe charging conditions and potential for catastrophic failure. In simulation, however, this would simply terminate the code with numerical errors.

The safety advantages of DrQ can be seen in plots of constraint satisfaction. Figures 5.10 and 5.11 show these results for the two active constraints, namely c_e^- and T. Figure 5.10 is particularly informative, since the electrolyte constraint is most often the dominant constraint. Here, DrQ strongly attenuates the magnitude and frequency of constraint violation. The temperature constraint is violated less for several reasons. First, the temperature constraint only becomes active after a history of high input current. DrQ shows faster convergence to an optimal policy, which aggressively charges the battery. Therefore, this constraint is violated more compared to DQN, which yields lower performing policies on average. Nevertheless, the risk metric η of the DrQ algorithm ($\eta = 5\%$) is validated within these experiments for all of the constraints, given that over all of the episodes and runs only 4.82% of timesteps exhibited any constraint violation.

Based on our data, the average DrQ episode took 361.58 seconds while the average DQN episode took 90.12 seconds. Both simulate faster than real-time, which suggests that DrQ (and DQN) could run within on-board microcontrollers, even for complex dynamical systems.

5.7 Conclusion

This chapter presents a novel algorithmic framework for deep Q-learning with probabilistic safety guarantees. Considering CMDPs, we apply a Wasserstein DRO framework to modify the constraint cost functions with offset variables that tighten towards the nominal constraint boundary as our modeling accuracy improves. We characterize the underlying modeling error of our function approximators with the TD errors of the constraint Q-functions, treated as random variables. This scheme allows us to observe constraint cost without violating nominal constraints, which provides strong information we use to define a set of feasible state-action pairs. The probabilistic guarantees of our augmented algorithm allow us to guarantee safety throughout the entire online learning process.

Our algorithm addresses critical challenges of safe RL literature. Specifically, we present a methodology for Q-learning which allows us to provide strong safety certificates during online learning. Our approach is widely applicable to a diverse set of learning-based optimal control problems. Furthermore, our approach facilitates the overall learning process with what we observe to be more consistent and dependable convergence, and more effective intermediate control results.

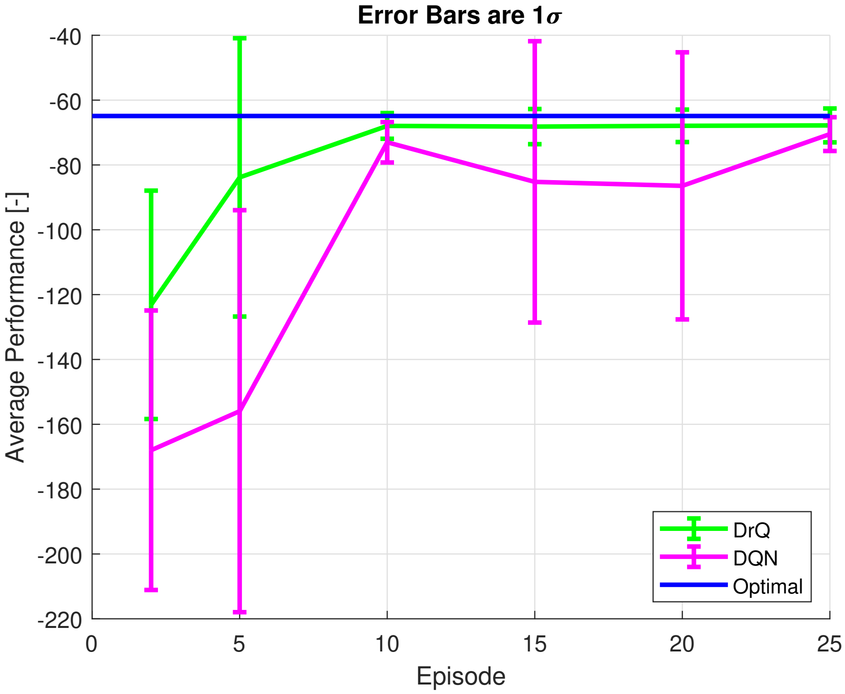

Figure 5.9: Greedy policy performance statistics over 10 runs of DrQ and DQN, based on the reward function defined by (5.40).

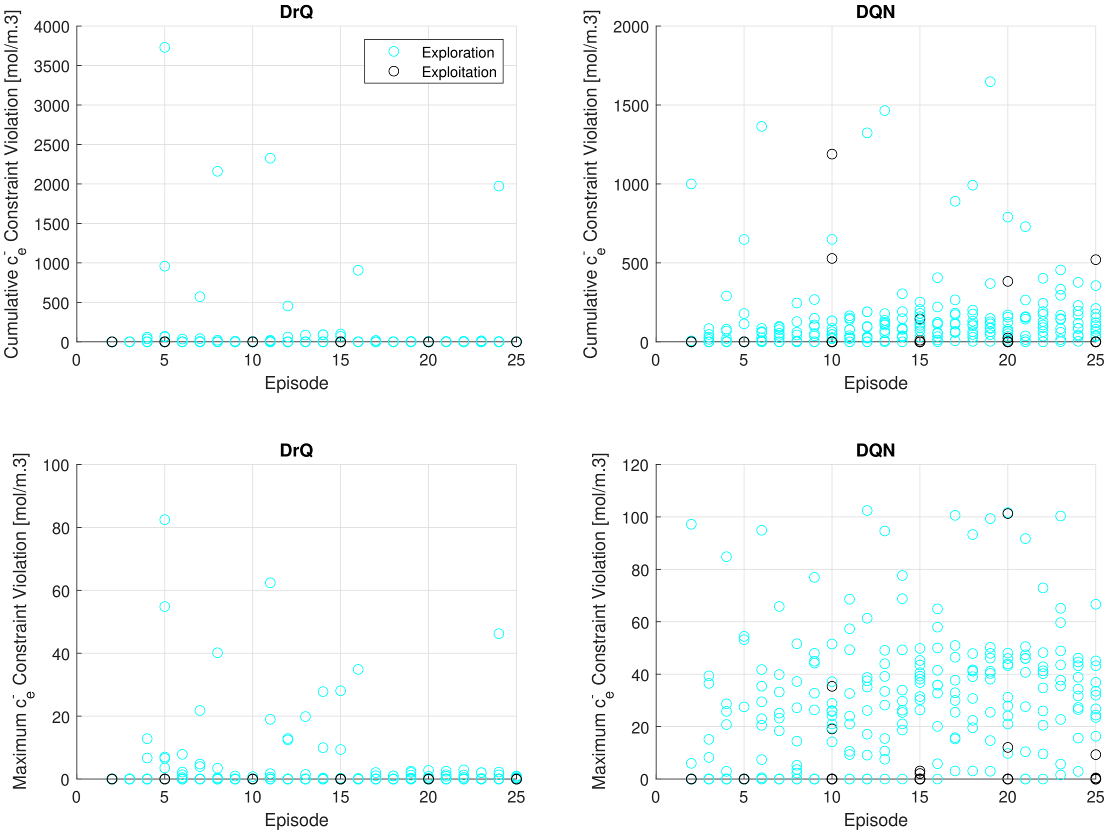

Figure 5.10: Safety statistics for the active c_e^- constraint over 10 runs of DrQ and DQN, starting from the second episode. The black "exploration" points correspond to the data obtained from the ϵ-greedy policy. The cyan "greedy" points correspond to the greedy policy evaluated incrementally across each run.

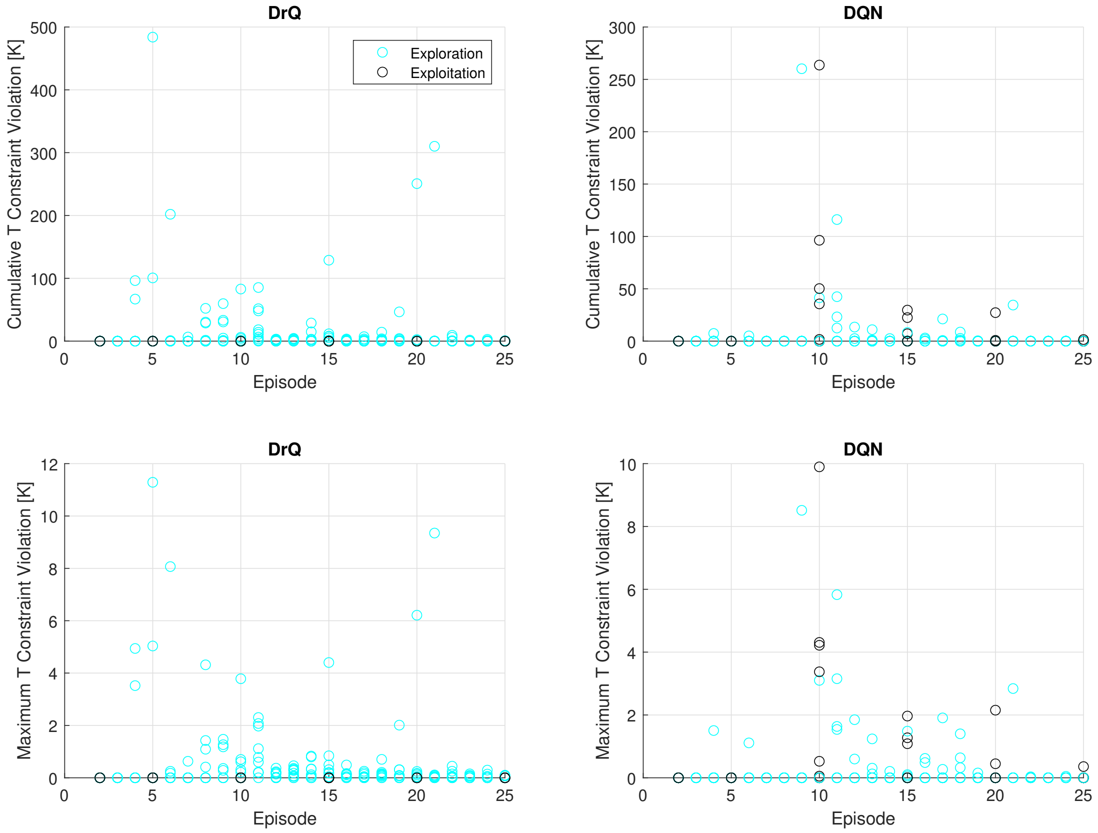

Figure 5.11: Safety statistics for the active T constraint over 10 runs of DrQ and DQN, starting from the second episode.

Chapter 6

Discussion and Conclusion

6.1 Dissertation Summary

This dissertation presents a general framework that develops tools from distributionally robust optimization to robustify learning-based controllers in three domains: (1) finite-time optimal control using data-driven black-box models; (2) online *tabula-rasa* learning-based control of nonlinear, multimodal systems; and (3) value-based reinforcement learning. This work is motivated by the ongoing relevance of robustness and safety when applying data-driven decision making to high impact industrial systems.

Many real-world industrial systems are high-dimensional, multimodal, poorly structured, and nonlinear. Data-driven control possesses a unique potential to optimize their performance, but complex model uncertainties have prevented prior algorithms from guaranteeing feasibility and safety in practice - preventing widespread adoption. For example, battery management systems utilize simple cell models. But maximizing the performance, longevity, and safety of lithium-ion batteries requires working with granular electrochemical information that is not observable and expensive to simulate. Moreover, multimodal observations of battery cells can improve the performance of control and management algorithms. The overarching contribution of this dissertation is progress it creates towards solving complex real-world problems like these. We provide simple extensions of existing DRO theory and leverage them to develop a suite of control algorithms amenable to complex, nonlinear, multimodal dynamical systems. We validate these algorithms - which span surrogate optimal control, model-based and model-free reinforcement learning - on a host of challenging case studies with emphasis on mechatronic and energy storage systems.

6.2 Summary of Contributions

The exact contributions of this dissertation can be summarized as follows:

- Chapter 2 presents simple extensions to DRO theory, focusing on optimization problems whose safety constraints are linearly separable in dependence on state variables x and modeled random variables $\mathbf{R}$.

- Chapter 3 presents an algorithm for data-driven finite-time optimal control. We leverage techniques from sequence modeling and data compression, as well as DRO theory from Chapter 2, to develop a tractable and probabilistically robust method for solving high-dimensional nonlinear optimal control problems.

- Chapter 4 of this dissertation presents an end-to-end framework for safe learning-based control using nonlinear stochastic MPC. In this Chapter, we focus on scenarios where the controller is applied directly to a system of which it has no or extremely limited direct experience, toward safety during tabula-rasa or “blank slate” model-based learning and control as a challenging case for validation. We show under basic and limited assumptions on the underlying problem that we can translate the probabilistic guarantees in Chapter 2 even with strong limitations on available data and model knowledge. We also present a coupled and intuitive formulation for the persistence of excitation (PoE) and illustrate the connection between PoE and the applicability of the proposed method.

- Chapter 5 develops a safe value-based RL algorithm based on the DRO technique in Chapter 2 within the structure of a constrained Markov decision process. A distributionally robust analysis of the distribution of constraint value function TD errors approximately characterizes the worst-case errors on our ability to predict constraint violation. We leverage this finding to add an adaptive level of conservatism to online deep Q-learning. We demonstrate that our robust RL policy more consistently respects constraint boundaries throughout the learning process.

6.3 Perspectives on Future Extensions

The proposed algorithmic architectures within this dissertation create a host of progress in translating theory to improved safety of data-driven controllers operating in real-world industrial systems. Nonetheless, there exist several routes to extend and further validate the results presented here.

Multimodality

Our case study of vehicle obstacle avoidance does synthesize physical states (e.g. position, velocity, heading angle) with input from a simple perception system. However, the bulk of this dissertation’s validation studies explore the fairly narrow application area of energy systems. Future work can apply each presented DRO control algorithm to additional nonlinear and multimodal control applications (e.g. visuo-motor control synthesis, OpenAI SafetyGym [96]).

Uncertainty Quantification

This dissertation focuses on applying results in Chapter 2 to learning-based control problems. However, the extended DRO theory can also be applied throughout applications of stochastic optimization and beyond. For example, consider the problem of training a sequence model to forecast a building's energy demand:

$$\mathcal{E}(x, t, \theta) = \vec{D} \tag{6.1}$$

where x is the current state of features utilized by the model, t is the current timestep, and θ are the model parameters. In the same sense that the black box models $\mathcal{G}$ and $f(x, u, \theta)$ forecast state transitions, $\mathcal{E}$ predicts energy demands. Likewise, the uncertainty of $\mathcal{E}$ can be characterized with the same DRO theory of Chapter 2. Since this theory allows user specified risk levels η, a sweep over η and samples of model prediction errors can provide a diverse characterization of, for example, error bars on forecast energy demand.

Bibliography

[1] Joshua Achiam et al. "Constrained Policy Optimization". In: *Proceedings of the 2017 International Conference on Machine Learning (ICML)*. Sydney, Australia: PMLR, 2017.

[2] Ibrahim Akbar. "Uncertainty Estimation in Continuous Models applied to Reinforcement Learning". PhD thesis. UC San Diego, 2019.

[3] Mohammed Alshiekh et al. "Safe Reinforcement Learning via Shielding". In: *arXiv* (2017), pp. 1–23.

[4] Eitan Altman. "Constrained Markov Decision Processes". In: (1999).

[5] Brandon Amos, Lei Xu, and J. Zico Kolter. "Input Convex Neural Networks". In: *International Conference on Machine Learning (ICML)*. Sydney, Australia, 2017.

[6] Kavosh Asadi, Dipendra Misra, and Michael L. Littman. *Lipschitz Continuity in Model-based Reinforcement Learning*. 2018. arXiv: `1804.07193 [cs.LG]`.

[7] Karl Johan Astrom. *Introduction to Stochastic Control Theory*. Courier Corporation, 1970.

[8] Felix Berkenkamp et al. "Safe Model-based Reinforcement Learning with Stability Guarantees". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[9] Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Vol. 5. Athena Scientific Belmont, MA, 1996.

[10] H.G. Beyer and HP. Schwefel. "Evolution strategies - A comprehensive introduction". In: *Natural Computing* 1.1 (Mar. 2002), pp. 3–52. DOI: `10.1023/A:1015059928466`.

[11] Lorenz T Biegler et al. "Large-Scale PDE-constrained Optimization: an introduction". In: *Lecture Notes in Computational Science and Engineering, Springer* (2003), pp. 3–13.

[12] Zdravko I. Botev et al. "Chapter 3 - The Cross-Entropy Method for Optimization". In: *Handbook of Statistics*. Ed. by C.R. Rao and Venu Govindaraju. Vol. 31. Handbook of Statistics. Elsevier, 2013, pp. 35–59. DOI: `https://doi.org/10.1016/B978-0-444-53859-8.00003-5`. URL: `https://www.sciencedirect.com/science/article/pii/B9780444538598000035`.

[13] Lukas Brunke et al. "Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning". In: *Annual Review of Control, Robotics, and Autonomous Systems* 5.1 (2022), pp. 411–444. DOI: 10.1146/annurev-control-042920-020211. eprint: https://doi.org/10.1146/annurev-control-042920-020211. URL: https://doi.org/10.1146/annurev-control-042920-020211.

[14] Monimoy Bujarbaruah, Xiaojing Zhang, and Francesco Borrelli. *Adaptive MPC with Chance Constraints for FIR Systems*. 2018. arXiv: 1804.09790 [cs.SY].

[15] Giuseppe C Calafiore and Laurent El Ghaoui. *Optimization models*. Cambridge university press, 2014.

[16] Michael Canon. *Theory of Optimal Control and Mathematical Programming*. McGraw, 1970.

[17] M Canova, K Pan, and G Fan. "A Comparison of Model Order Reduction Techniques for Electrochemical Characterization of Lithium-Ion Batteries". In: *54th IEEE Conference on Decision and Control*. Osaka, Japan, 2015.

[18] Stephen Casper et al. *Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback*. 2023. arXiv: 2307.15217 [cs.AI].

[19] Fernando Castañeda et al. *In-Distribution Barrier Functions: Self-Supervised Policy Filters that Avoid Out-of-Distribution States*. 2023. arXiv: 2301.12012 [cs.RO].

[20] Fernando Castañeda et al. "In-Distribution Barrier Functions: Self-Supervised Policy Filters that Avoid Out-of-Distribution States". In: *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*. Ed. by Nikolai Matni, Manfred Morari, and George J. Pappas. Vol. 211. Proceedings of Machine Learning Research. PMLR, 15–16 Jun 2023, pp. 286–299. URL: https://proceedings.mlr.press/v211/castaneda23a.html.

[21] Ricky T. Q. Chen et al. *Neural Ordinary Differential Equations*. 2019. arXiv: 1806.07366 [cs.LG].

[22] Yize Chen, Yuanyuan Shi, and Baosen Zhang. "Optimal Control Via Neural Networks: A Convex Approach". In: *International Conference on Learning Representations (ICLR)*. New Orleans, LA USA, 2019.

[23] Richard Cheng et al. "End-to-End Safe Reinforcement Learning through Barrier Functions for Safety-Critical Continuous Control Tasks". In: *AAAI*. 2019.

[24] Jason Choi et al. *Reinforcement Learning for Safety-Critical Control under Model Uncertainty, using Control Lyapunov Functions and Control Barrier Functions*. 2020. arXiv: 2004.07584 [eess.SY].

[25] Yinlam Chow, Jia Yuan Yu, and Marco Pavone. "Two Phase Q-learning for Bidding-based Vehicle Sharing". In: *arXiv* (2015), pp. 1–11.

[26] Yinlam Chow et al. “A Lyapunov-based Approach to Safe Reinforcement Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[27] Jeremy Coulson, John Lygeros, and Florian Dorfler. “Distributionally Robust Chance Constrained Data-enabled Predictive Control”. In: *IEEE Transactions on Automatic Control* (2021), pp. 1–1. DOI: 10.1109/TAC.2021.3097706.

[28] Sarah Dean et al. “Safely Learning to Control the Constrained Linear Quadratic Regulator”. In: *Proceedings of the 2019 American Control Conference*. Philadelphia, PA, USA: IEEE, 2019.

[29] Dejan V. Djonin and Vikram Krishnamurthy. “Q-Learning Algorithms for Constrained Markov Decision Processes with Randomized Monotone Policies: Application to MIMO Transmission Control”. In: *IEEE Transactions on Signal Processing* 55.5 (2007), pp. 2170–2181.

[30] Marc Doyle, Thomas Fuller, and John Newman. “Modeling of Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell”. In: *Journal of the Electrochemical Society* 140.6 (1993), pp. 1526–1533.

[31] Marc Doyle, Thomas F. Fuller, and John Newman. “Modeling of Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell”. In: *Journal of The Electrochemical Society* 140.6 (June 1993), p. 1526. DOI: 10.1149/1.2221597. URL: https://dx.doi.org/10.1149/1.2221597.

[32] Marc Doyle and John Newman. “The use of mathematical modeling in the design of lithium/polymer battery systems”. In: *Electrochimica Acta* 40.13-14 (1995), pp. 2191–2196.

[33] Chao Duan et al. “Distributionally Robust Chance-Constrained Approximate AC-OPF With Wasserstein Metric”. In: *IEEE Transactions on Power Systems* 33.5 (2018), pp. 4924–4936.

[34] Payman Esfahani and Daniel Kuhn. “Data-Driven Distributionally Robust Optimization Using the Wasserstein Metric: Performance Guarantees and Tractable Reformulations”. In: *Mathematical Programming* 171.1–2 (2018), pp. 115–166.

[35] D. D. Fan et al. “Bayesian Learning-Based Adaptive Control for Safety Critical Systems”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020), pp. 4093–4099.

[36] Miriam A. Figueroa-Santos, Jason B. Siegel, and Anna G. Stefanopoulou. “Leveraging Cell Expansion Sensing in State of Charge Estimation: Practical Considerations”. In: *Energies* 13.10 (2020). ISSN: 1996-1073. DOI: 10.3390/en13102653. URL: https://www.mdpi.com/1996-1073/13/10/2653.

[37] Chelsea Finn et al. “Deep spatial autoencoders for visuomotor learning”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 512–519. DOI: 10.1109/ICRA.2016.7487173.

[38] Jaime F. Fisac et al. "A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems". In: *IEEE Transactions on Automatic Control* (2018), pp. 1–1.

[39] Joel C. Forman et al. "Reduction of an Electrochemistry-Based Li-Ion Battery Model via Quasi-Linearization and Padé Approximation". In: *Journal of the Electrochemical Society* 158.2 (2011), A93–A101. URL: `http://jes.ecsdl.org/content/158/2/A93.abstract`.

[40] Rui Gao and Anton J. Kleywegt. "Distributionally Robust Stochastic Optimization with Wasserstein Distance". In: *arXiv* (2016).

[41] Javier Garcia and Fernando Fernandes. "A Comprehensive Survey on Safe Reinforcement Learning". In: *Journal of Machine Learning Research* 16 (2016), pp. 1437–1480.

[42] Clement Gehring and Doina Precup. "Smart Exploration in Reinforcement Learning using Absolute Temporal Difference Errors". In: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems.* Ed. by Takayuki Ito et al. Saint Paul, Minnesota, USA: International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1037–1043.

[43] Joao Hespanha. *Linear Systems Theory.* Princeton University Press, 2009.

[44] Lukas Hewing et al. "Learning-Based Model Predictive Control: Toward Safe Learning in Control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 3.1 (2020), pp. 269–296. DOI: `10.1146/annurev-control-090419-075625`. eprint: `https://doi.org/10.1146/annurev-control-090419-075625`. URL: `https://doi.org/10.1146/annurev-control-090419-075625`.

[45] Ryan Hickey and Thomas M. Jahns. "Measuring Individual Battery Dimensional Changes for State-of-Charge Estimation using Strain Gauge Sensors". In: *2019 IEEE Energy Conversion Congress and Exposition (ECCE).* 2019, pp. 2460–2465. DOI: `10.1109/ECCE.2019.8912578`.

[46] Joey Hong, Sergey Levine, and Anca Dragan. *Zero-Shot Goal-Directed Dialogue via RL on Imagined Conversations.* 2023. arXiv: `2311.05584 [cs.LG]`.

[47] Ruiwei Jiang and Yongpei Guan. "Data-driven chance constrained stochastic programs". In: *Mathematical Programming* 140.6 (2016), pp. 291–327.

[48] Donald R Jones, Matthias Schonlau, and William J Welch. "Efficient Global optimization of expensive black-box functions". In: *Journal of Global Optimization* 13.1 (1998), pp. 455–492.

[49] Lukasz Kaiser et al. "Model based reinforcement learning for atari". In: *arXiv* (2019).

[50] Aaron Kandel. *Wasserstein Nonlinear MPC.* Version 0.0.1. Aug. 2023. URL: `https://github.com/aaronkandel/Wasserstein-Nonlinear-MPC/tree/main`.

[51] Aaron Kandel and Scott J. Moura. “Safe Learning MPC With Limited Model Knowledge and Data”. In: *IEEE Transactions on Control Systems Technology* (2023), pp. 1–16. DOI: 10.1109/TCST.2023.3324869.

[52] Aaron Kandel, Saehong Park, and Scott J. Moura. “Distributionally Robust Surrogate Optimal Control for High-Dimensional Systems”. In: *IEEE Transactions on Control Systems Technology* (2022), pp. 1–12. DOI: 10.1109/TCST.2022.3216988.

[53] Aaron Kandel et al. “Distributionally robust surrogate optimal control for large-scale dynamical systems”. In: *Proceedings of the 2020 American Control Conference (to appear)*. Denver, CO USA: IEEE, 2020.

[54] G. Kerschen et al. “The Method of Proper Orthogonal Decomposition for Dynamical Characterization and Order Reduction of Mechanical Systems: An Overview”. In: *Nonlinear Dynamics* 41.1 (2005).

[55] Mohammad Javad Khojasteh et al. “Probabilistic Safety Constraints for Learned High Relative Degree System Dynamics”. In: *L4DC*. 2020.

[56] Donald E. Kirk. *Optimal Control Theory*. Dover, 1970.

[57] Torsten Koller et al. “Learning-based Model Predictive Control for Safe Exploration and Reinforcement Learning”. In: *arXiv* (2018), pp. 1–8.

[58] Rogier Koppejan and Shimon Whiteson. “Neuroevolutionary reinforcement learning for generalized control of simulated helicopters”. In: *Evolutionary Intelligence* 4.4 (2011), pp. 219–241.

[59] Mayuresh V. Kothare, Venkataramanan Balakrishnan, and Manfred Morari. “Robust constrained model predictive control using linear matrix inequalities”. In: *Automatica* 32.10 (1996), pp. 1361–1379.

[60] Aviral Kumar et al. *Conservative Q-Learning for Offline Reinforcement Learning*. 2020. arXiv: 2006.04779 [cs.LG].

[61] Nicholas C Landolfi, Garrett Thomas, and Tengyu Ma. “A Model-based approach for sample-efficient multitask reinforcement learning”. In: *arXiv* (2019).

[62] Edith L.M. Law et al. “Risk-directed Exploration in Reinforcement Learning”. In: *Proceedings of the IJCAI’05 Workshop on Planning and Learning in A Priori Unknown or Dynamic Domains*. Ed. by V. Bulitko and S. Koenig. Edinburgh, United Kingdom: International Joint Conferences on Artificial Intelligence, 2005, pp. 97–102.

[63] Erwan Lecarpentier and Emmanuel Rachelson. “Non-Stationary Markov Decision Processes, a Worst-Case Approach using Model-Based Reinforcement Learning”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 7216–7225.

[64] Sergey Levine et al. “End-to-End Training of Deep Visuomotor Policies”. In: *CoRR* abs/1504.00702 (2015). arXiv: 1504.00702. URL: http://arxiv.org/abs/1504.00702.

[65] Lisha Li et al. “Efficient Hyperparameter Optimization and Infinitely Many Armed Bandits”. In: *CoRR* abs/1603.06560 (2016). arXiv: `1603.06560`. URL: `http://arxiv.org/abs/1603.06560`.

[66] Timothy P. Lillicrap et al. *Continuous control with deep reinforcement learning*. 2015. arXiv: `1509.02971 [cs.LG]`.

[67] Bryan Lim et al. “Temporal Fusion Transformers for interpretable multi-horizon time series forecasting”. In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764. ISSN: 0169-2070. DOI: `https://doi.org/10.1016/j.ijforecast.2021.03.012`. URL: `https://www.sciencedirect.com/science/article/pii/S0169207021000637`.

[68] Tyler Lu, Dale Schuurmans, and Craig Boutilier. “Non-delusional Q-learning and value-iteration”. In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 9949–9959.

[69] Yolanda Mack et al. “Surrogate Model-Based Optimization Framework: A Case Study in Aerospace Design”. In: *Evolutionary Computation in Dynamic and Uncertain Systems* (2007).

[70] Richard Maclin et al. “Knowledge-based support vector regression for reinforcement learning”. In: *Proceedings of the IJCAI’05 Workshop on Reasoning, Representation, and Learning in Computer Games*. Edinburgh, Scotland: International Joint Conferences on Artificial Intelligence, 2005, pp. 61–66.

[71] Frederic Maire. “Apprenticeship learning for initial value functions in reinforcement learning”. In: *Proceedings of the IJCAI’05 Workshop on Planning and Learning in A Priori Unknown or Dynamic Domains*. Ed. by V. Bulitko and S. Koenig. Edinburgh, United Kingdom: International Joint Conferences on Artificial Intelligence, 2005, pp. 23–28.

[72] Horia Mania, Aurelia Guy, and Benjamin Recht. “Simple Random search provides a competitive approach to reinforcement learning”. In: *arXiv* (2018).

[73] Timothy A. Mann and Yoonsuck Choe. “A Comprehensive Survey on Safe Reinforcement Learning”. In: *JMLR: Workshop and Conference Proceedings 24:59–75, 2012 10th European Workshop on Reinforcement Learning*. Edinburgh, Scotland: JMLR W—&C Proceedings, 2012, pp. 1437–1480.

[74] Julien Marzat and Helene Piet-Lahanier. “Design of nonlinear MPC by Kriging-based optimization”. In: *16th IFAC Symposium on System Identification*. Brussels, Belgium: The International Federation of Automatic Control, 2012, pp. 1490–1495.

[75] Ravi Methekar et al. “Optimum charging profile for lithium-ion batteries to maximize energy storage and utilization”. In: *Transactions of the Electrochemical Society* 25.35 (2010), pp. 139–146.

[76] Thomas M. Moerland et al. *Model-based Reinforcement Learning: A Survey*. 2020. DOI: `10.48550/ARXIV.2006.16712`. URL: `https://arxiv.org/abs/2006.16712`.

[77] S. Mohan, Y. Kim, and A. G. Stefanopoulou. "Energy-Conscious Warm-Up of Li-Ion Cells From Subzero Temperatures". In: *IEEE Transactions on Industrial Electronics* 63.5 (2016), pp. 2954–2964. DOI: 10.1109/TIE.2016.2523440.

[78] David L. Moreno et al. "Using prior knowledge to improve reinforcement learning in mobile robotics". In: *Proceedings of the Conference Towards Autonomous Robotics Systems*. Bath, England: Springer, 2004.

[79] Scott Moura, Nalin Chaturvedi, and Miroslav Krstic. "Constraint Management in Li-ion Batteries: A Modified Reference Governor Approach". In: *20133 American Control Conference*. Washington, DC: IFAC, IEEE, 2013.

[80] Scott J. Moura. "Estimation and control of battery electrochemistry models: A tutorial". In: *2015 54th IEEE Conference on Decision and Control (CDC)*. 2015, pp. 3906–3912. DOI: 10.1109/CDC.2015.7402827.

[81] Scott J. Moura et al. "Battery State Estimation for a Single Particle Model with Electrolyte Dynamics". In: *IEEE Transactions on Control Systems Technology* 25.2 (Mar. 2017), pp. 453–468. DOI: 10.1109/TCST.2016.2571663. URL: http://ieeexplore.ieee.org/document/7489035/.

[82] Anusha Nagabandi et al. "Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning". In: *International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia, 2018.

[83] Ashvin Nair et al. *AWAC: Accelerating Online Reinforcement Learning with Offline Datasets*. 2021. arXiv: 2006.09359 [cs.LG].

[84] Arnab Nilim and Laurent El Ghaoui. "Robust Control of Markov Decision Processes with Uncertain Transition Matrices". In: *Operations Research* 53.5 (2005).

[85] Saehong Park et al. "A Deep Reinforcement Learning Framework for Fast Charging of Li-ion Batteries". In: *IEEE Transactions on Transportation Electrification* (2022), pp. 1–1. DOI: 10.1109/TTE.2022.3140316.

[86] Saehong Park et al. "Optimal Control of Battery Fast Charging Based-on Pontryagin's Minimum Principle". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. 2020, pp. 3506–3513. DOI: 10.1109/CDC42340.2020.9304409.

[87] Ronald Parr and Stuart Russel. "Reinforcement Learning with Hierarchies of Machines". In: *Proceedings of the 10th Conference on Neural Information Processing Systems (NIPS 1997)*. Denver, CO: Neural Information Processing Systems Foundation, Inc., 1997.

[88] Praveen Paruchuri et al. "Towards a formalization of teamwork with resource constraints". In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.* New York, NY USA: IEEE, 2004.

[89] Bart P.G. Van Parys et al. "Distributionally robust control of constrained stochastic systems". In: *IEEE Transactions on Automatic Control* 61.2 (2016), pp. 430–442.

[90] Joel Paulson, Edward Buehler, and Ali Mesbah. "Arbitrary Polynomial Chaos for Uncertainty Propagation of Correlated Random Variables in Dynamic Systems". In: *IFAC PapersOnLine* 50.1 (2017), pp. 3548–3553.

[91] HE Perez et al. "Optimal Charging of Li-Ion Batteries With Coupled Electro-Thermal-Aging Dynamics". In: *IEEE Trans. on Veh. Technology* 66.7 (2017), pp. 7761–7770.

[92] Hector Perez, Niloofar Shahmohammadhamedani, and Scott Moura. "Enhanced Performance of Li-Ion Batteries via Modified Reference Governors and Electrochemical Models". In: *IEEE/ASME Transactions on Mechatronics* 20.4 (Aug. 2015), pp. 1511–1520. DOI: https://doi.org/10.1109/TMECH.2014.2379695. URL: https://ieeexplore.ieee.org/document/7004876.

[93] T. J. Perkins and A. G. Barto. "Lyapunov design for safe reinforcement learning". In: *Journal of Machine Learning Research* 3 (2002), pp. 803–832.

[94] Nestor Queipo et al. "Surrogate-based analysis and optimization". In: *Progress in Aerospace Sciences* 41 (2005), pp. 1–28.

[95] Christopher D Rahn and Chao-Yang Wang. *Battery Systems Engineering.* John Wiley & Sons, 2012.

[96] Alex Ray, Joshua Achiam, and Dario Amodei. "Benchmarking Safe exploration in deep reinforcement learning". In: *arXiv* (2020).

[97] Spencer Richards, Felix Berkenkamp, and Andreas Krause. "The Lyapunov Neural Network: Adaptive Stability Certification for Safe Learning of Dynamic Systems". In: *arXiv* (2018), pp. 1–11.

[98] Ugo Rosolia and Francesco Borrelli. "Learning Model Predictive Control for Iterative Tasks. A Data-Driven Control Framework". In: *IEEE Transactions on Automatic Control* 63.7 (2017), pp. 1883–1896.

[99] Michael J. Rothenberger et al. "Genetic optimization and experimental validation of a test cycle that maximizes parameter identifiability for a Li-ion equivalent-circuit battery model". In: *Journal of Energy Storage* 4 (2015), pp. 156–166.

[100] Martin Schlegen et al. "Dynamic Optimization using adaptive control vector parameterization". In: *Computers and Chemical Engineering* 29.8 (2005), pp. 1731–1751.

[101] Krishnan Srinivasan et al. "Learning to be Safe: Deep RL with a Safety Critic". In: *CoRR* abs/2010.14603 (2020). arXiv: 2010.14603. URL: https://arxiv.org/abs/2010.14603.

[102] Florian Tambon et al. "How to certify machine learning based safety-critical systems? A systematic literature review". In: *Automated Software Engineering* 29.2 (Apr. 2022). DOI: 10.1007/s10515-022-00337-x. URL: https://doi.org/10.1007%2Fs10515-022-00337-x.

[103] Marko Tanaskovic et al. “Adaptive receding horizon control for constrained MIMO systems”. In: *Automatica* 50 (2014), pp. 3019–3029.

[104] K Thomas, John Newman, and R Darling. “Mathematical modeling of lithium batteries”. In: *Advances in lithium-ion batteries* (2002), pp. 345–392. DOI: 10.1007/0-306-47508-1_13. URL: http://www.springerlink.com/index/RXM87M4067U87J65.pdf.

[105] Lorenzo Usai et al. “Analysis of the Li-ion battery industry in light of the global transition to electric passenger light duty vehicles until 2050”. In: *Environmental Research: Infrastructure and Sustainability* 2.1 (Mar. 2022), p. 011002. DOI: 10.1088/2634-4505/ac49a0. URL: https://dx.doi.org/10.1088/2634-4505/ac49a0.

[106] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL].

[107] Tyler Westenbroek et al. “Combining Model-Based Design and Model-Free Policy Optimization to Learn Safe, Stabilizing Controllers”. In: *Proceedings of the 7th IFAC Conference on Analysis and Design of Hybrid Systems*. 2021.

[108] Insoon Yang. “A Convex Optimization Approach to Distributionally Robust Markov Decision Processes With Wasserstein Distance”. In: *IEEE Control Systems Letters* 1.1 (2017), pp. 164–169.

[109] Insoon Yang. “Wasserstein Distributionally Robust Stochastic Control: A Data-Driven Approach”. In: *arXiv* (2018).

[110] Tianhe Yu et al. “MOPO: Model-based Offline Policy Optimization”. In: *CoRR* abs/2005.13239 (2020). arXiv: 2005.13239. URL: https://arxiv.org/abs/2005.13239.

[111] Rowan Zellers et al. “MERLOT Reserve: Neural Script Knowledge Through Vision and Language and Sound”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 16375–16387.

[112] Baohe Zhang et al. “On the Importance of Hyperparameter Optimization for Model-based Reinforcement Learning”. In: *CoRR* abs/2102.13651 (2021). arXiv: 2102.13651. URL: https://arxiv.org/abs/2102.13651.

[113] Chaoyue Zhao and Yongpei Guan. “Data-driven risk-averse stochastic optimization with Wasserstein metric”. In: *Operations Research Letters* 46.2 (2018), pp. 262–267.

[114] Zhengang Zhong, Ehecatl Antonio del Rio-Chanona, and Panagiotis Petsagkourakis. *Data-driven distributionally robust MPC using the Wasserstein metric*. 2021. DOI: 10.48550/ARXIV.2105.08414. URL: https://arxiv.org/abs/2105.08414.

Chapter 7

Appendices

7.1 Single Particle Model with Electrolyte & Thermal Dynamics

The single particle model with electrolyte and thermal dynamics (henceforth denoted as SPMeT) is a reduced-order electrochemical lithium-ion battery model derived from the Doyle-Fuller-Newman (DFN) electrochemical battery model [81]. The DFN model employs a continuum of particles throughout the anode and cathode of the battery cell. Diffusion within this continuum is represented with partial differential equations (PDEs) and differential-algebraic equations (DAEs). The SPMeT uses a simplified representation of solid phase diffusion based on a single spherical particle in each electrode of the battery cell. Compared to the ECM model used in the main text (which is isothermal), the SPMeT incorporates thermal dynamics. Furthermore, the state variables of the SPMeT model provide direct physical intuition on the conditions occurring within the battery cell. The SPMeT model also provides much more accurate prediction at higher input current rates.

The main advantage of designing fast charging controllers with the SPMeT is that we can leverage the granular electrochemical information encoded in the dynamical state to replace the phenomenological equivalent circuit model used in the main text. Specifically, by constraining electrochemical state variables instead of terminal output voltage, we can safely expand the safe operating envelope in order to improve charging times significantly. Coincidentally, constraining electrochemical states gives us greater agency in avoiding rapid cell aging sourced from myopic charging protocols.

The governing equations for SPMeT include linear and quasiliniar PDEs and a nonlinear voltage output equation, given by:

$$\frac{\partial c_s^{\pm}}{\partial t}(r,t) = \frac{1}{r^2}\frac{\partial}{\partial r}\left[D_s^{\pm} r^2 \frac{\partial c_s^{\pm}}{\partial r}(r,t)\right], \tag{7.1}$$

$$\varepsilon_e^j \frac{\partial c_e^j}{\partial t}(x,t) = \frac{\partial}{\partial x}\left[D_e^{\mathrm{eff}}(c_e^j)\frac{\partial c_e^j}{\partial x}(x,t) + \frac{1-t_c^0}{F} i_e^j(x,t)\right], \tag{7.2}$$

where t represents time. The superscript j denotes anode, seperator and cathode, $j \in \{+, \text{sep}, -\}$, each forming essential components of the lithium ion battery cell. The terminal voltage output is governed by a combination of electric overpotential, electrode thermodyanmics, and Butler-Volmer kinetics, yielding:

$$\begin{aligned} V(t) = {} & \frac{RT_{cell}(t)}{\alpha F} \sinh^{-1}\left(\frac{I(t)}{2a^{+}AL^{+}\bar{i}_0^{+}(t)}\right) - \frac{RT_{cell}(t)}{\alpha F} \sinh^{-1}\left(\frac{-I(t)}{2a^{-}AL^{-}\bar{i}_0^{-}(t)}\right) \\ & + U^{+}(c_{ss}^{+}(t)) - U^{-}(c_{ss}^{-}(t)) + \left(\frac{R_f^{+}}{a^{+}AL^{+}} + \frac{R_f^{-}}{a^{-}AL^{-}} + \frac{R_{ce}(T_{avg}(t))}{A}\right) I(t) \\ & - \left(\frac{L^{+} + 2L^{sep} + L^{-}}{2A\bar{\kappa}^{eff}}\right) I(t) + k_{conc}(t)[ln(c_e(0^{+}, t)) - ln(c_e(0^{-}, t))], \end{aligned} \tag{7.3}$$

where c_{ss} is the solid phase surface concentration, namely $c_{ss}^{\pm}(x,t) = c_s^{\pm}(x, R_s^{\pm}, t)$, $U^{\pm}$ is the open-circuit potential, and $c_{s,\max}^{\pm}$ is the maximum possible concentration in the solid phase. The exchange current density i_0^j and solid-electrolyte surface concentration c_{ss}^j are given by:

$$i_0^j(c_{ss}^j) = k^j \sqrt{c_e^0 c_{ss}^j(t)(c_{s,\max}^j - c_{ss}^j(t))}, \tag{7.4}$$

$$c_{ss}^j(t) = c_s^j(R_s^j, t), \quad j \in \{+, -\}. \tag{7.5}$$

Note the electrolyte diffusion PDE (7.2) is quasilinear because the diffusion coefficient depends on lithium concentration, $D_e^{\text{eff}}(c_e^j)$.

The nonlinear temperature dynamics are modeled with a single lumped thermal mass subjected to heat generation from the input current:

$$\frac{dT_{cell}}{dt}(t) = \frac{\dot{Q}(t)}{mC_{p;th}} - \frac{T_{cell(t)} - T_{\infty}}{mC_{p;th}R_{th}} \tag{7.6}$$

where T_{∞} is the ambient temperature, m is the mass of the cell, $C_{p;th}$ is the thermal specific heat capacity of the cell, R_{th} is the thermal resistance of the cell, and $\dot{Q}(t)$ is the heat added from the charging, which is governed by

$$\dot{Q}(t) = I(t)\left[U^{+}(SOC_p) - U^{-}(SOC_n) - V(t)\right] \tag{7.7}$$

Here, $I(t)$ is the input current (the control input), and $V(t)$ is the voltage determined by (7.3). Both nonlinear open circuit potential functions in (7.7) are functions of the bulk state of charge (SOC) in the anode and cathode, respectively. For more details on the SPMeT equations and notation, we direct the reader to ([81]).

7.2 Doyle-Fuller-Newman Electrochemical Battery Model

We consider the Doyle-Fuller-Newman (DFN) model to predict the evolution of lithium concentration in the solid $c_s^{\pm}(x, r, t)$, lithium concentration in the electrolyte $c_e(x, t)$, solid electric potential $\phi_s^\pm(x,t)$, electrolyte electric potential $\phi_e(x,t)$, ionic current $i_e^\pm(x,t)$, molar ion fluxes $j_n^\pm(x,t)$, and battery temperature $T(t)$. The x-dimension runs across the negative electrode, separator, and positive electrode. At each x-coordinate value in the negative and positive electrodes, we consider a particle where spherical lithium intercalation occurs. The governing equations in time are given by

$$\frac{\partial c_s^\pm}{\partial t}(x,r,t) = \frac{1}{r^2}\frac{\partial}{\partial r}\left[D_s^\pm r^2 \frac{\partial c_s^\pm}{\partial r}(x,r,t)\right], \tag{7.8}$$

$$\varepsilon_e^j \frac{\partial c_e^j}{\partial t}(x,t) = \frac{\partial}{\partial x}\left[D_e^{\text{eff}}(c_e^j)\frac{\partial c_e^j}{\partial x}(x,t) + \frac{1-t_c^0}{F} i_e^j(x,t)\right], \tag{7.9}$$

$$mc_P \frac{dT}{dt}(t) = \frac{1}{R_{th}}\left[T_{\text{amb}} - T(t)\right] + \dot{Q}, \tag{7.10}$$

for $j \in \{-, \text{sep}, +\}$ and $\dot{Q}$ is the rate of heat transferred to the system [104], defined as

$$\dot{Q} = I(t)\left[U^+(t) - U^-(t) - V(t)\right] - \tag{7.11}$$

$$I(t)T(t)\frac{\partial}{\partial T}[U^+(t) - U^-(t)], \tag{7.12}$$

and differential equations in space and algebraic equations are given by

$$\sigma^{\text{eff},\pm} \cdot \frac{\partial \phi_s^\pm}{\partial x}(x,t) = i_e^\pm(x,t) - I(t), \tag{7.13}$$

$$\begin{aligned}\kappa^{\text{eff}}(c_e) \cdot \frac{\partial \phi_e}{\partial x}(x,t) = &-i_e^\pm(x,t) + \kappa^{\text{eff}}(c_e) \cdot \frac{2RT}{F}(1 - t_c^0) \\ &\times \left(1 + \frac{d\ln f_{c/a}}{d\ln c_e}(x,t)\right)\frac{\partial \ln c_e}{\partial x}(x,t),\end{aligned} \tag{7.14}$$

$$\frac{\partial i_e^\pm}{\partial x}(x,t) = a^\pm F j_n^\pm(x,t), \tag{7.15}$$

$$j_n^\pm(x,t) = \frac{1}{F} i_0^\pm(x,t)\left[e^{\frac{\alpha_a F}{RT}\eta^\pm(x,t)} - e^{-\frac{\alpha_c F}{RT}\eta^\pm(x,t)}\right], \tag{7.16}$$

$$i_0^\pm(x,t) = k^\pm\left[c_{ss}^\pm(x,t)\right]^{\alpha_c}\left[c_e(x,t)\left(c_{s,\max}^\pm - c_{ss}^\pm(x,t)\right)\right]^{\alpha_a}, \tag{7.17}$$

$$\eta^\pm(x,t) = \phi_s^\pm(x,t) - \phi_e(x,t) - U^\pm(c_{ss}^\pm(x,t)) - FR_f^\pm j_n^\pm(x,t), \tag{7.18}$$

$$c_{ss}^\pm(x,t) = c_s^\pm(x, R_s^\pm, t). \tag{7.19}$$

where $D_e^{\text{eff}} = D_e(c_e) \cdot (\varepsilon_e^j)^{\text{brug}}$, $\sigma^{\text{eff}} = \sigma \cdot (\varepsilon_s^j + \varepsilon_f^j)^{\text{brug}}$, $\kappa^{\text{eff}} = \kappa(c_e) \cdot (\varepsilon_e^j)^{\text{brug}}$ are the effective electrolyte diffusivity, effective solid conductivity, and effective electrolyte conductivity given by the Bruggeman relationship. The boundary conditions for solid-phase diffusion PDE (7.8) are

$$\frac{\partial c_s^\pm}{\partial r}(x,0,t) \quad = \quad 0, \tag{7.20}$$

$$\frac{\partial c_s^\pm}{\partial r}(x, R_s^\pm, t) \;=\; -\frac{1}{D_s^\pm} j_n^\pm(x,t). \tag{7.21}$$

The boundary conditions for the electrolyte-phase diffusion PDE (7.9) are given by

$$\begin{aligned}
\frac{\partial c_e^-}{\partial x}(0^-, t) = \frac{\partial c_e^+}{\partial x}(0^+, t) = 0, && (7.22)\\
\varepsilon_e^- D_e(L^-)\frac{\partial c_e^-}{\partial x}(L^-, t) = \varepsilon_e^{\text{sep}} D_e(0^{\text{sep}})\frac{\partial c_e^{\text{sep}}}{\partial x}(0^{\text{sep}}, t), && (7.23)\\
\varepsilon_e^{\text{sep}} D_e(L^{\text{sep}})\frac{\partial c_e^{\text{sep}}}{\partial x}(L^{\text{sep}}, t) = \varepsilon_e^+ D_e(L^+)\frac{\partial c_e^+}{\partial x}(L^+, t), && (7.24)\\
c_e(L^-, t) = c_e(0^{\text{sep}}, t), && (7.25)\\
c_e(L^{\text{sep}}, t) = c_e(L^+, t). && (7.26)
\end{aligned}$$

The boundary conditions for the electrolyte-phase potential ODE (7.14) are given by

$$\begin{aligned}
\phi_e(0^-, t) &= 0, && (7.27)\\
\phi_e(L^-, t) &= \phi_e(0^{\text{sep}}, t), && (7.28)\\
\phi_e(L^{\text{sep}}, t) &= \phi_e(L^+, t). && (7.29)
\end{aligned}$$

The boundary conditions for the ionic current ODE (7.15) are given by

$$i_e^-(0^-, t) = i_e^+(0^+, t) = 0, \tag{7.30}$$

and also note that $i_e(x,t) = I(t)$ for $x \in [0^{\text{sep}}, L^{\text{sep}}]$. In addition, the parameters, $D_s^\pm, D_e, \kappa_e, k^\pm$ vary with temperature via the Arrhenius relationship:

$$\psi = \psi_{ref} \exp\left[\frac{E_\phi}{R}\left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right] \tag{7.31}$$

where ψ represents a temperature dependent parameter, E_ψ is the activation energy and ψ_{ref} is the reference parameter value at room temperature. The model input is the applied current density $I(t)$ [A/m^2], and the output is the voltage measured across the current collectors,

$$V(t) = \phi_s^+(0^+, t) - \phi_s^-(0^-, t). \tag{7.32}$$

The level of charge in the cell is defined by the bulk state of charge (SOC) of the negative electrode, namely,

$$SOC^-(t) = \int_0^{L^-} \frac{\bar{c}_s^-(x,t)}{c_{s,max}^-(\theta_{100\%}^- - \theta_{0\%}^-)L^-} dx \tag{7.33}$$

where $\bar{c}_s^-$ represents the volume averaged of a particle in the solid phase defined as:

$$\bar{c}_s^-(x,t) = \frac{3}{(R_s^-)^3}\int_0^{R_s^-} r^2 c_s^-(r,t) dr \tag{7.34}$$

Lithium plating, which is the main battery degradation mechanism, is related to the side reaction overpotential η_s, defined as:

$$\eta_s(x,t) = \phi_s^-(x,t) - \phi_e^-(x,t) - U_{sr} \geq 0. \tag{7.35}$$

To facilitate numerical optimal control, this model is discretized in space and time. There is a rich literature on discretization methods (see e.g. [95, 17]). The discretization approached used for this dissertation involve finite difference, Padé approximation [39], and automatic differentiation methods.

7.3 Cardinality of Constraints Remains Constant

In the following lemma, we show that the number of constraints in the reformulation of the DRO problem in (4.13) need only be m (where m is the dimension of the constraint function output). When $g(\cdot)$ is non-separable, as described in [33], then the number of constraints in the reformulation scales super-linearly as 2^m.

Lemma 3 *If the modeling error residuals are defined as:*

$$\begin{aligned} R_1^{(t)} &= |g(x_t, u_t^*) - \hat{g}(x_t, u_t^*, \theta_g)| && (7.36a) \\ R_1^{(t+1)} &= |g(x_{t+1}^*, u_{t+1}^*) - \hat{g}(\hat{x}_{t+1}, u_{t+1}^*, \theta_g)| && (7.36b) \\ R_1^{(t+b)} &= |g(x_{t+b}^*, u_{t+b}^*) - \hat{g}(\hat{x}_{t+b}, u_{t+b}^*, \theta_g)| && (7.36c) \end{aligned}$$

and appear in the constraint function $g(\cdot)$ as (4.13), then the number of constraints in the reformulated problem remains identically m without jeopardizing the probabilistic guarantee.

Proof 2 *Consider the following stochastic constraint converted to a distributionally robust chance constraint:*

$$\begin{aligned} x + \boldsymbol{R} &\leq 0 && (7.37a) \\ \underset{P \in B_\epsilon}{inf} \mathbb{P}\left[x + \boldsymbol{R} \leq 0\right] &\geq 1 - \eta && (7.37b) \end{aligned}$$

representing a constraint with uncertainty. Without loss of generality, we consider a MPC program with horizon $N = 1$.

The method of [33] enumerates across the vertices of a hypercube by modulating the sign of the DRO variable σ. However, when the random variable is a separable offset from a constant constraint boundary, we only need consider perturbations that add conservatism. In the 1-dimensional case, we can see from looking at the set of constraints

$$x \leq -r \; and \, x \leq r \tag{7.38}$$

that only the first constraint $x \leq -r$ will ever be active. Therefore, $x \leq -r$ adequately defines the feasible region.

Likewise, if we consider the case where $\boldsymbol{R} \in \mathbb{R}^2$ with additive $\boldsymbol{R}$, we obtain the following set of constraints

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \leq 0 \tag{7.39}$$

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} -r_1 \\ r_2 \end{bmatrix} \leq 0 \tag{7.40}$$

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} r_1 \\ -r_2 \end{bmatrix} \leq 0 \tag{7.41}$$

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} -r_1 \\ -r_2 \end{bmatrix} \leq 0 \tag{7.42}$$

we see trivially that the feasible region defined by (7.39-7.42) is identical to that defined solely by (7.42). This pattern continues for any $m \in \mathbb{N}$ of $\boldsymbol{R} \in \mathbb{R}^m$.

7.4 Evolutionary Strategies and Random Search

In Chapters 3 and 4 of this dissertation, we utilize a $(1 + \lambda)$ evolutionary strategy to approximately solve numerical MPC optimization programs. This is a subset of what is generally referred to as a $(\frac{\mu}{\rho} + \lambda)$ evolutionary strategy, whose precise definition can be referenced in [10]. A $(\frac{\mu}{\rho} + \lambda)$ evolutionary strategy is a very simple form of a genetic algorithm, whereby at each generation/iteration of optimization, we have some number of "parents" who are mutated, and the parents are replaced by the highest performing mutated offspring. Random search has been shown to be a highly effective method for solving optimization problems in reinforcement learning literature [**RStest**]. Random search is also highly amenable to constrained optimization (without equality constraints), as infeasible mutants can be pruned from selection. If no feasible mutants are found, the mutant that least violates the constraint boundary can be selected to avoid additional computation.

7.5 Slow Model Adaptation

To accommodate potential cases where the true plant dynamics change slowly over time, we can adopt the following approach which preserves the safety guarantees of the Wasserstein DRO framework. We have system dynamics $x \in \mathbb{R}^n$ with no finite escape time. Furthermore, $g(x, u, \theta^*) \leq 0$ is our constraint function. Suppose it holds that the function g behaves in the following manner (similarly, although not identically, to a Lipschitz continuous function):

$$\max_{x \in x, u \in U, \delta\theta} |g(x, u, \theta + \delta\theta) - g(x, u, \theta)| \leq C \tag{7.43}$$

where $\delta\theta = \theta^*_{t+1} - \theta^*_t$ is any possible deviation in the model parameters over the course of a single timestep. The value $\delta\theta$ is bounded. Consider we are at time t of the experiment. Let us represent the 1-step residual at time $j = t - k$, where $k \in \{1, 2, ..., t\}$ is an integer, as:

$$R_1^{(t)} = g(x_t, u_t, \theta_t^*) - \hat{g}(x_t, u_t, \theta_t) \tag{7.44}$$

where θ_t^* is the parameterization of the true plant at time t, and θ_t is the learned model at time t. If we add a value to the residual $R_1^{(t)}$ of $C \cdot k \cdot \mathrm{sgn}(R_1^{(t)})$,

$$\tilde{R}_1^{(t)} = R_1^{(t)} + C \cdot k \cdot \mathrm{sgn}(R_1^{(t)}) \tag{7.45}$$

we accommodate for worst-case model adaptation in our algorithm. This scheme, coupled with a judiciously designed moving window of residuals, can accommodate model adaptation in the true underlying plant.