UNIVERSITY OF CALIFORNIA SAN DIEGO

Data-Driven Online Optimization and Control with Performance Guarantees

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

in

Engineering Sciences
(Mechanical Engineering)

by

Dan Li

Committee in charge:

Professor Sonia Martínez, Chair
Professor Philip Gill
Professor Boris Krämer
Professor Miroslav Krstić
Professor Behrouz Touri

2021

The Dissertation of Dan Li is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

DEDICATION

To my loving parents and to the memory of my grandpa.

EPIGRAPH

Decisions must often be taken in the face of the unknown. Actions decided upon in the present will have consequences that can't fully be determined until a later stage.
But there may be openings for corrective action later or even multiple opportunities for recourse as more and more becomes known…

– Ralph Tyrrell Rockafellar

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

It brings me great pleasure for an opportunity to work on data-driven control and optimization, which is the main facet of my research at UC San Diego. For this I deeply indebted to my adviser, Sonia Martínez. I cannot make this thesis work possible without her invaluable mindset, sharp vision and outstanding guidance. In addition, let me express my sincere gratitude and appreciation for her kindness, expertise, enormous support and perpetual patience. Without these I would not have the freedom to explore new ideas, shape research goals and eventually develop an excellent mindset as a professional researcher.

Next, I would like to thank all my committee members, Prof. Philip Gill, Prof. Boris Krämer, Prof. Miroslav Krstić and Prof. Behrouz Touri, for their valuable time and comments on the development of this work. In particular, I am immensely grateful to Prof. Gill for his rigorous, easy-to-follow courses and *Practical Optimization* textbook, which built up a solid foundation to problems I have worked on; I am very thankful to Prof. Krämer for his responsiveness and support whenever possible; I am indebted to Prof. Krstić, who helped me in many ways, from insightful nonlinear control-system course to the many constructive comments on my work; and finally I would like to thank Prof. Touri for his broad research interests towards my work, various from distributed optimization to learning and control. In addition, I would like to extend my thanks to Prof. Jorge Cortés for his very insightful views on a variety of topics during our group meetings over the past a few years, which truly solidified my understanding on distributed control, optimization, multi-agent systems, etc. Special thanks go to one of my collaborators, Dr. Dariush Fooladivanda, for his constant availability and help, which contributed to lots of fruitful discussions and exciting outcomes.

Further, I would like to thank all the current and former members of our group as well as members of other groups at UC San Diego, including, but not limited to, Aamodh, Aaron, Ahmed, Ashish, Beth, Chin-Yao, Dariush, Dimitris, Drew, Eduardo, Erfan, Jason, Javier, Huan, Li, Masih, Miguel, Parth, Paul, Pengcheng, Pio, Priyank, Sai, Scott, Shenyu, Sven, Tor, Xuan, Yangsheng, Yifu, Yunhai, Zhichao. Their friendship and company made my journey an endless

joy.

Finally, I am very very much thankful to my family for their love, care, understanding, encouragement and support over these many years. My parents have been great friends of mine who guide me through hard time and challenges in my life and constantly inspire me to interpret things with fresh and special views. Without their love and enormous sacrifice I would not be possible to get to this point. Further, I would be remiss not to mention my grandparents, uncles, aunts and cousins who always stand behind me and give me unconditional support.

This research was developed with funding from (1) Defense Advanced Research Projects Agency (DARPA) Lagrange grant contract N660011824027, (2) Office of Naval Research (ONR) contract N00014-19-1-2471, and (3) Air Force Office of Scientific Research (AFOSR) grant contract FA9550-19-1-0235 and FA9550-18-1-0158.

Chapter 2, in full, is a reprint of *Data Assimilation and Online Optimization With Performance Guarantees*, D. Li and S. Martínez, IEEE Transactions on Automatic Control, (66)5:2115-2129, 2021. A preliminary version appeared in the proceddings of IEEE International Conference on Decision and Control, pp. 1961-1966, Miami, FL, USA, 2018, as *Online data assimilation in distributionally robust optimization*, D. Li and S. Martínez. The dissertation author was the primary investigator and author of these papers.

Chapter 3, in full, is under review for publication in International Journal on Robust and Nonlinear Control, entitled as *Data-driven predictive control for a class of uncertain control-affine systems*, D. Li, D. Fooladivanda, and S. Martínez. A motivating work appeared in the proceddings of European Control Conference, pp. 1055-1061, Napoli, Italy, 2019, as *Data-driven variable speed limit design for highways via distributionally robust optimization*, D. Li, D. Fooladivanda, and S. Martínez. The dissertation author was the primary investigator and author of these papers.

Chapter 4, in full, is a reprint of *Online learning of parameterized uncertain dynamical environments with finite-sample guarantees*, D. Li, D. Fooladivanda, and S. Martínez, IEEE

Control Systems Letters, 5(4):1351-1356, 2021, which was presented at American Control Conference, New Orleans, LA, US, 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 5, in full, is under revision for publication in Automatica, as *Online optimization and learning in uncertain dynamical environments with performance guarantees*, D. Li, D. Fooladivanda, and S. Martínez. The dissertation author was the primary investigator and author of this paper.

Chapter 6, in full, is a reprint of *High-confidence attack detection via Wasserstein-metric computations*, D. Li and S. Martínez, IEEE Control Systems Letters, 5(2):379-384, 2020, which was presented at IEEE International Conference on Decision and Control, Jeju Island, Korea, 2020. The dissertation author was the primary investigator and author of this paper.

VITA

2021 Ph.D., University of California San Diego, USA

2016 M.S., Queen's University, Canada

2013 B.S., Zhejiang University, China

PUBLICATIONS

1. *Distributionally-robust Feedback Controllers for Piecewise Linear Systems in Multiple Tasks*, D. Li and S. Martínez, In preparation.

2. *Data-driven predictive control for a class of uncertain control-affine systems*, D. Li, D. Fooladivanda, and S. Martínez, International Journal on Robust and Nonlinear Control, 2021. Under review.

3. *Online optimization and learning in uncertain dynamical environments with performance guarantees*, D. Li, D. Fooladivanda, and S. Martínez, Automatica, 2021. Under review.

4. *Data Assimilation and Online Optimization With Performance Guarantees*, D. Li and S. Martínez, IEEE Transactions on Automatic Control, (66)5:2115-2129, 2021.

5. *Online learning of parameterized uncertain dynamical environments with finite-sample guarantees*, D. Li, D. Fooladivanda, and S. Martínez, IEEE Control Systems Letters, 5(4):1351-1356, 2021.

6. *High-confidence attack detection via Wasserstein-metric computations*, D. Li and S. Martínez, IEEE Control Systems Letters, 5(2):379-384, 2020.

7. *Data-driven variable speed limit design for highways via distributionally robust optimization*, D. Li, D. Fooladivanda, and S. Martínez, European Control Conference, pages 1055-1061, Napoli, Italy, June 2019.

8. *Online data assimilation in distributionally robust optimization*, D. Li and S. Martínez, IEEE Int. Conf. on Decision and Control, pages 1961-1966, Miami, FL, USA, December 2018.

ABSTRACT OF THE DISSERTATION

Data-Driven Online Optimization and Control with Performance Guarantees

by

Dan Li

Doctor of Philosophy in Engineering Sciences
(Mechanical Engineering)

University of California San Diego, 2021

Professor Sonia Martínez, Chair

This thesis considers the analysis and design of algorithms for the management and control of uncertain intelligent systems which are observable through (limited) online-accessible data. Examples include online equity trading systems under extreme price fluctuations, robotic systems moving in unknown environments, and transportation systems subject to uncertain drivers' actions and other (accident) events.

To ensure safe, reliable, and resilient system behaviors, this thesis studies various theoretical problem scenarios, which focus on reducing uncertainty with performance guarantees via the assimilation of streaming data, the data-driven design of control, and online learning of

system models, resilient operations in uncertain environments, and anomaly detection.

These formulations are largely rooted in two mechanisms: online optimization and distributionally robust optimization, where the first enables online-tractable formulations of the problem, and the latter accounts for systemic uncertainty with high confidence. Both approaches are applicable beyond the particular systems of study, to virtually any type of dynamic system where sensitive data is progressively available and may be exploited to the advantage of management and control. This work is unique in that it brings together current tools in optimization, control of dynamical systems, data-based modeling and probability theory, significantly advancing the state of the art.

Introduction

Uncertainty is ubiquitous in real-world complex systems, including financial markets [1], human-robot mixed systems [31, 78, 125] or transportation networks [44, 67, 102]. Thanks to new advances in computation, communication, and innovative infrastructure, large amounts of data have become widely accessible, which can help reduce uncertainty in system design, control and optimization [125]. Consequently, the state-of-art, robust and intelligent complex systems, which enable data collection, uncertainty mitigation and as well as decision making, become possible online. Examples include safe autonomous driving, multi-agent cooperative control, large-scale transportation and smart energy systems. However, system performance guarantees typically require large amounts of data processing which makes challenging the practical implementation of such controllers.

This thesis concerns the analysis and design of algorithms for the robust intelligent-system operations with high confidence. While designing such algorithms, we aim at contributing to the theoretical foundations that can make a reality the safe deployment of intelligent system via data-driven control, and enable non-asymptotic performance guarantees with finite amount of data. This may be the case when data collection is costly or decisions need to be made before a very large amount of data can be collected, because an adversary purportedly hides it. The collected data are assumed to be realizations or samples of some random-variable distribution which, in real-world applications, is unknown. Of special interest are online optimization and control frameworks where the challenges are the finite, incremental availability of new information in the scenario where large amounts of data were needed for safety. A main approach we leverage is Distributionally robust optimization (DRO), which has attracted recent attention

due to its capability to deal with uncertainty and provide *out-of-sample* performance guarantees with a finite number of data. We apply DRO and recent measure-of-concentration results in online optimization, learning for control and anomaly detection.

To achieve those objectives via DRO, the very first framework we consider involves the minimization of time-varying convex loss functions, resulting into Online Convex Programming (OCP). Typically, loss objectives in OCP are functions of non-stationary stochastic processes [49, 141]. Regret minimization aims to deal with nonstationarity by reducing the difference between an optimal decision made with information in hindsight, and one made as information is increasingly revealed. Thus, several online algorithms and techniques are aimed at minimizing various types of regret functions [54, 94]. More recently, and with the aim of further reducing the cost, regret-based OCP has integrated prediction models of loss functions [23, 46, 73, 109]. However, exact models of evolving loss functions may not be available, while alternative data-based approximate models may require large amounts of data that are hard to obtain. This motivates the need of developing new frameworks and algorithms for loss functions that can employ finite data sets, while guaranteeing a precise performance of the corresponding optimization, which is the core of the responsive and intelligent systems.

In this regard, the thesis examines largely three problem scenarios:

(1): Loss functions are random, but they are determined explicitly by an uncertainty distribution $\mathbb{P}$. Even if $\mathbb{P}$ is unknown, samples of $\mathbb{P}$ are revealed incrementally over time. Thus, Chapter 2 integrates streaming data sets into an online optimization framework, scrutinizes data-assimilation capabilities, and meanwhile guarantees online-decision performance via DRO. An Online Data Assimilation Algorithm is developed as a general framework, which enables data-driven on-the-fly optimization with guarantees to minimize inaccessible loss functions.

(2): Loss functions are random and implicit, depending on the uncertainty behavior which is governed by a given, uncertain dynamical system via control. Such a scenario has natural connections with optimal control problems, including stochastic model predictive control [22, 83, 130] and Kalman filtering [48]. Chapter 3 then aims to achieve tractable,

data-driven predictive control for a class of control-affine systems which is subject to uncertainty. An innovative solution approach is established and applied to a class of traffic control problems as an example.

(3): Built upon the last scenario, the uncertain dynamical systems which govern uncertainty behaviors are unknown as well. In such scenario, several techniques, from first-principles system identification to, more recently, (deep) neural networks, have been successfully used in various domains to capture uncertainty behaviors. However, as we mentioned, safe performance usually depends upon the assimilation of vast amounts of data, which is mostly done offline and prevents its application in real-time scenarios. Motivated by this, Chapter 4 investigates the integration of recently-developed probabilistically-guaranteed system descriptions with online, predictor-based learning algorithms. And further, in Chapter 5, a tractable, performance-guaranteed online optimization framework is developed, leveraging DRO and the system learning methodology in Chapter 4.

In these scenarios, there is a fundamental assumption on data-collection process: the adversaries are absent. Otherwise, when there are attacks or corruptions in the procedure, quantification of system performance remain elusive. In Chapter 6, a novel data-driven detection mechanism is provided, which identifies attacks in real-time and, as the result, it serves as a complement of frameworks developed.

Chapter 1

Notation and Preliminaries

This chapter introduces standard notations whichever are not specified in the rest of the thesis, and as well as the basic distributionally robust optimization framework.

1.1 Notation

Let $\mathbb{R}^m$, $\mathbb{R}^m_{\geq 0}$, $\mathbb{Z}^m_{\geq 0}$ and $\mathbb{R}^{m\times n}$ denote the m-dimensional real space, nonnegative orthant, nonnegative integer space, and the space of $m \times n$ matrices, respectively. In particular, we denote $\mathbb{N} := \mathbb{Z}_{\geq 0}$. By $\boldsymbol{x} \in \mathbb{R}^m$ we denote a column vector of dimension m, while $\boldsymbol{x}^\top$ represents its transpose. We use the shorthand symbol $\mathbf{0}_m$ for the column vector $(0, \cdots, 0)^\top \in \mathbb{R}^m$, the symbol $\mathbf{1}_m$ for the column vector $(1, \cdots, 1)^\top \in \mathbb{R}^m$, and $\boldsymbol{I}_m \in \mathbb{R}^{m\times m}$ for the identity matrix. For any vector $\boldsymbol{x} \in \mathbb{R}^m$, let us denote $\boldsymbol{x} \geq \mathbf{0}_m$ if all the entries are nonnegative. We use either subscripts or parentheses superscripts to index vectors, i.e., $\boldsymbol{x}_k \in \mathbb{R}^m$ or $\boldsymbol{x}^{(k)} \in \mathbb{R}^m$, for $k \in \mathbb{N}$, and we use x_i to denote the i^{th} component of $\boldsymbol{x} \in \mathbb{R}^m$. We use $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{m+d}$ to indicate the concatenated column vector from $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^d$. Let us denote by $\|\boldsymbol{x}\|_1$, $\|\boldsymbol{x}\| := \|\boldsymbol{x}\|_2$ and $\|\boldsymbol{x}\|_\infty$ the 1-norm, 2-norm and ∞-norm, respectively. We define the m-dimensional norm ball with center $\boldsymbol{x} \in \mathbb{R}^m$ and radius $\epsilon \in \mathbb{R}_{\geq 0}$ as the set $B_\epsilon(\boldsymbol{x}) := \{\boldsymbol{y} \in \mathbb{R}^m \mid \|\boldsymbol{y} - \boldsymbol{x}\| \leq \epsilon\}$. We denote by $\langle \cdot, \cdot \rangle$ an inner product in the space of interest. Consider the space $\mathbb{R}^m$, we define $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y}$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$. In particular, $\|\boldsymbol{x}\| := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$. Consider Finsler manifold $\mathbb{R}^2 \times [-\pi, \pi) \cong \mathbb{R} \times \mathbb{S}^1$ where $\mathbb{S}^1$ stands for the unit circle. For $(\boldsymbol{x}, \theta_1), (\boldsymbol{y}, \theta_2) \in \mathbb{R}^2 \times [-\pi, \pi)$,

we define $\langle(\boldsymbol{x},\theta_1),(\boldsymbol{y},\theta_2)\rangle := \boldsymbol{x}^\top\boldsymbol{y} + \cos(\min\{|\theta_1-\theta_2|,\, 2\pi - |\theta_1-\theta_2|\})$. In particular, we use $\|(\boldsymbol{x},\theta)\| := \sqrt{\langle \boldsymbol{x},\boldsymbol{x}\rangle + 1}$. Given a norm $\|\boldsymbol{x}\|$, $\boldsymbol{x}\in\mathbb{R}^m$, we denote by $\|\boldsymbol{x}\|_\star := \sup_{\|z\|\le 1}\langle \boldsymbol{z},\, \boldsymbol{x}\rangle$ the corresponding dual norm. Notice that $\|\boldsymbol{x}\|_{**} = \|\boldsymbol{x}\|$ and $\|\boldsymbol{x}\|_\infty \equiv (\|\boldsymbol{x}\|_1)_\star$. Given a set of points $I\subset\mathbb{R}^m$, we let $\mathrm{conv}(I)$ indicate its convex hull. Let $\boldsymbol{x}\circ\boldsymbol{y}$ denote the component-wise product of vectors $\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^m$. The component-wise square of vector $\boldsymbol{x}\in\mathbb{R}^m$ is denoted by $\boldsymbol{x}^2 := \boldsymbol{x}\circ\boldsymbol{x}$. In addition, let $\boldsymbol{x}\otimes\boldsymbol{y}$ denote the Kronecker product of vectors $\boldsymbol{x},\boldsymbol{y}$ with the arbitrary dimension. For matrices $A_1\in\mathbb{R}^{m\times d}$ and $A_2\in\mathbb{R}^{p\times q}$, we let $A_1\oplus A_2$ denote their direct sum. The shorthand notation $\oplus_{i=1}^m A_i$ represents $A_1\oplus\cdots\oplus A_m$.

Notation Compression Rule: Any variable or letter x may have appended the following indices and arguments: it may have the subscript x_e with $e\in\mathbb{N}$, the argument $x_e(t)$ with $t\in\mathbb{N}$, and finally a superscript $l\in\mathbb{N}$ as in $x_e^{(l)}(t)$. Given a finite number of elements $x_e^{(l)}(t)\in\mathbb{R}$ where $e,t,l\in\mathbb{N}$, we define vectors $x^{(l)}(t) := (x_1^{(l)}(t), x_2^{(l)}(t),\ldots)$, $x^{(l)} := (x^{(l)}(1), x^{(l)}(2),\ldots)$, and $x := (x^{(1)}, x^{(2)},\ldots)$.

Convexity and Projection: The gradient of a real-valued function $f:\mathbb{R}^m\to\mathbb{R}$ is written as $\nabla f(\boldsymbol{x})$ or $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$. The i^{th} component of the gradient vector is denoted by $\nabla_i f(\boldsymbol{x})$ or $\nabla_{x_i} f(\boldsymbol{x})$. We use $\mathrm{dom}(f)$ to denote the domain of the function f, i.e., $\mathrm{dom}(f) := \{\boldsymbol{x}\in\mathbb{R}^m \mid -\infty < f(\boldsymbol{x}) < +\infty\}$. We call the function f *proper* if $\mathrm{dom}(f)\neq\varnothing$. A function $\ell:\mathrm{dom}(\ell)\to\mathbb{R}$ is M-strongly convex, if for any $\boldsymbol{y},\boldsymbol{z}\in\mathrm{dom}(\ell)$ there exists $\boldsymbol{g}$ such that $\ell(\boldsymbol{y})\ge\ell(\boldsymbol{z})+\boldsymbol{g}^\top(\boldsymbol{y}-\boldsymbol{z})+M\|\boldsymbol{y}-\boldsymbol{z}\|^2/2$, for some $M\ge 0$. The function ℓ is convex if $M=0$. We call the vector $\boldsymbol{g}$ a subgradient of ℓ at $\boldsymbol{z}$ and denote by $\partial\ell(\boldsymbol{z})$ the set of subgradients. Note that, if ℓ is differentiable at $\boldsymbol{z}$, then $\partial\ell(\boldsymbol{z}) = \{\nabla\ell(\boldsymbol{z})\}$, i.e., $\partial\ell(\boldsymbol{z})$ contains only the gradient of ℓ at $\boldsymbol{z}$. A function ℓ is *concave* if $-\ell$ is convex. We say a function $F:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}$ is *convex-concave* on $\mathcal{X}\times\mathcal{Y}$ if, for any point $(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\in\mathcal{X}\times\mathcal{Y}$, $\boldsymbol{x}\mapsto F(\boldsymbol{x},\tilde{\boldsymbol{y}})$ is convex and $\boldsymbol{y}\mapsto F(\tilde{\boldsymbol{x}},\boldsymbol{y})$ is concave. We use the notation $\mathrm{sgn}:\mathbb{R}\to\mathbb{R}$, $x\mapsto\{-1,0,1\}$ denote the sign function. Finally, the projection operation $\mathrm{proj}_{\mathcal{U}}(\mathcal{X})$ or $\Pi_{\mathcal{U}}(\mathcal{X}):\mathcal{X}\to\mathcal{U}$ projects the set $\mathcal{X}$ onto $\mathcal{U}$ under the Euclidean norm. In particular, we write $\Pi_{\mathcal{U}}(\boldsymbol{x}) := \mathrm{argmin}_{\boldsymbol{z}}\,\|\boldsymbol{x}-\boldsymbol{z}\|^2/2 + \chi_{\mathcal{U}}(\boldsymbol{z})$, where $\boldsymbol{x}\in\mathcal{X}$, and $\chi_{\mathcal{U}}(\boldsymbol{z}) = 0$ if $\boldsymbol{z}\in\mathcal{U}$, otherwise $+\infty$.

Singular Value Decomposition (SVD): Given an $A\in\mathbb{R}^{m\times m}$, we write its SVD as

$A = U\Sigma V^\top$, where U, $V \in \mathbb{R}^m$ are orthonormal and Σ is diagonal with non-negative entries. These entries are called singular values of A, and we denoted by $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ the maximal and non-zero minimal singular value of A, respectively. We denote by $A^\dagger := V\Sigma^\dagger U^\top$ the Moore–Penrose inverse of A, where $\Sigma^\dagger$ is the same as Σ except the replacement of each positive entry by its inverse.

Convex Conjugate Functions: Consider a bounded function $f : X \to \mathbb{R}$ where $X \subseteq \mathbb{R}^n$. The function f is lower semi-continuous on X if $f(x) \leq \liminf_{y \to x} f(y)$ for all $x \in X$. Similarly, the function f is lower semi-continuous on X if and only if its sublevel sets $\{x \in X \mid f(x) \leq \gamma\}$ are closed for each $\gamma \in \mathbb{R}$. We let $f^\star : X \to \mathbb{R} \cup \{+\infty\}$ denote the convex conjugate of f, which is defined as $f^\star(x) := \sup_{y \in X} \langle x,\ y\rangle - f(y)$. Further, the infimal convolution of two functions f and g on X is defined as $(f \square g)(x) := \inf_{y \in X} f(x - y) + g(y)$. If f and g are bounded, convex, and lower semi-continuous functions on X, we will have $(f + g)^\star = (f^\star \square g^\star)$. Consider a subset $A \subset X$, and let $\chi_A : X \to \mathbb{R} \cup \{+\infty\}$ denote the characteristic function of A, i.e., $\chi_A(x)$ is equal to 0 if and only if $x \in A$ and $+\infty$ otherwise. In addition, let $\sigma_A : X \to \mathbb{R}$ denote the support function of A, which is defined as $\sigma_A(x) := \sup_{y \in A} \langle x,\ y\rangle$. Notice that χ_A is lower semi-continuous if and only if A is closed, and that $\sigma_A(x) = [\chi_A]^\star(x)$ for all $x \in X$.

1.2 Distributionally Robust Optimization

This section introduces basic Probability Theory to describe the Distributionally Robust Optimization (DRO) framework.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, with Ω the sample space, $\mathcal{F}$ a σ-algebra on Ω, and $\mathbb{P}$ the associated probability distribution. Let $\boldsymbol{\xi} : \Omega \to \mathbb{R}^m$ be an induced $\mathbb{R}^m$-valued random variable. We denote by $\mathcal{Z} \subseteq \mathbb{R}^m$ the support of the random variable $\boldsymbol{\xi}$ and denote by $\mathcal{M}(\mathcal{Z})$ the space of all probability distributions supported on $\mathcal{Z}$ with finite mean or first moment. In particular, we assume $\mathbb{P} \in \mathcal{M}(\mathcal{Z})$. We define in the following a class of distributions considered.

Definition 1 (q-light-tailed distributions). *For a random vector $\boldsymbol{\xi}$ such that $\boldsymbol{\xi} \sim \mathbb{P}$, we say $\mathbb{P}$ is*

q-light-tailed with a $q = 1, 2, \ldots$, *if* $c := \mathbb{E}_{\mathbb{P}}[\exp(b\|\boldsymbol{\xi}\|^a)] < \infty$ *for some* $a > q$ *and* $b > 0$.

Remark 1 (Class of distributions satisfying Definition 1). Intuitively, Definition 1 is a refinement of a class of distributions with a finite moment generating function. Any distribution with an exponentially decaying tail satisfies this assumption, such as Gaussian, subGaussian, exponential, and geometric distributions. Any distribution with a compact support $\mathcal{Z}$ will trivially satisfy the definition. In engineering problems, the values of random variables are usually truncated to a compact set and hence Definition 1 is automatically satisfied.

To measure the distance between distributions, we use in this thesis the Wasserstein metric. Let $\mathcal{M}_q(\mathcal{Z}) \subseteq \mathcal{M}(\mathcal{Z})$ denote the space of all q-light-tailed probability distributions supported on $\mathcal{Z} \subseteq \mathbb{R}^m$. Then for any two distributions $\mathbb{Q}_1$, $\mathbb{Q}_2 \in \mathcal{M}_q(\mathcal{Z})$, the q-Wasserstein metric [115] $d_{W,q} : \mathcal{M}_q(\mathcal{Z}) \times \mathcal{M}_q(\mathcal{Z}) \to \mathbb{R}_{\geq 0}$ is defined by

$$d_{W,q}(\mathbb{Q}_1, \mathbb{Q}_2) := \left(\min_{\Pi} \int_{\mathcal{Z}\times\mathcal{Z}} \ell^q(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) \Pi(d\boldsymbol{\xi}_1, d\boldsymbol{\xi}_2) \right)^{1/q},$$

where Π is in a set of all the probability distributions on $\mathcal{Z} \times \mathcal{Z}$ with marginals $\mathbb{Q}_1$ and $\mathbb{Q}_2$. The cost $\ell(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) := \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|$ is a norm on $\mathcal{Z}$. The Wasserstein metric can be interpreted as an *optimal mass transport problem* in the literature [115]. In this thesis, wherever q is not specified, we consider $q = 1$ and denote directly $\mathcal{M}_{\text{lt}}$ and d_W for $\mathcal{M}_1$ and $d_{W,1}$, respectively. Or we denote $\mathcal{M}_1$ by $\mathcal{M}$ in chapters where these definitions are customized and clarified for the particular problem of interest. In addition, for some part of the thesis, we use the equivalent, dual characterization of 1-Wasserstein metric [59, Kantorovich-Rubinstein Theorem] $d_W : \mathcal{M}_{\text{lt}}(\mathcal{Z}) \times \mathcal{M}_{\text{lt}}(\mathcal{Z}) \to \mathbb{R}_{\geq 0}$, defined by

$$d_W(\mathbb{Q}_1, \mathbb{Q}_2) := \sup_{f \in \mathcal{L}} \int_{\mathcal{Z}} f(\boldsymbol{\xi}) \mathbb{Q}_1(d\boldsymbol{\xi}) - \int_{\mathcal{Z}} f(\boldsymbol{\xi}) \mathbb{Q}_2(d\boldsymbol{\xi}),$$

where $\mathcal{L}$ is the space of all Lipschitz functions defined on $\mathcal{Z}$ with Lipschitz constant 1. A closed Wasserstein ball of radius $\omega \in \mathbb{R}_{\geq 0}$ centered at a distribution $\mathbb{P} \in \mathcal{M}(\mathcal{Z})$ is denoted by $\mathbb{B}_\omega(\mathbb{P}) := \{\mathbb{Q} \in \mathcal{M}(\mathcal{Z}) \mid d_W(\mathbb{P}, \mathbb{Q}) \leq \omega\}$. We denote the Dirac measure at $x_0 \in \Omega$ as $\delta_{\{x_0\}} : \Omega \to \{0, 1\}$. For

any set $A \in \mathcal{F}$, we let $\delta_{\{x_0\}}(A) = 1$, if $x_0 \in A$, otherwise 0. For an $\boldsymbol{x} \in \Omega$, we denote $\mathbb{P} \equiv \mathbb{Q} + \boldsymbol{x}$, if $\mathbb{P}$ is a translation of $\mathbb{Q}$ by $\boldsymbol{x}$.

Distributionally Robust Optimization (DRO) framework concerns a decision problem under uncertainty, where each decision leads to a measurable, uncertain and $\mathbb{R}$-valued loss function $\ell(\boldsymbol{\xi})$, with $\boldsymbol{\xi} \in \mathbb{R}^m$ an uncertainty-related random vector governed by a distribution $\mathbb{P}$. We denote by $\mathcal{L}$ the feasible set of all available loss functions and consider the decision problem

$$\inf_{\ell \in \mathcal{L}} \mathbb{E}_{\mathbb{P}}[\ell(\boldsymbol{\xi})], \tag{1.1}$$

where the distribution $\mathbb{P}$, in most real life scenarios, is fundamentally unknown. However, $\mathbb{P}$ can be observable through *a-priori* knowledge, e.g., the support of $\mathbb{P}$ from structural properties on the problem of interest, or accessible and independently collected data, denoted by $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n$. These knowledge contribute to the interpretation of $\mathbb{P}$ empirically and, in practice, it is convenient to construct an *empirical distribution* of $\mathbb{P}$ as $\hat{\mathbb{P}}_n := \frac{1}{n}\sum_{k=1}^{n} \delta_{\boldsymbol{\xi}_k}$. In addition, $\mathbb{P}$ can be characterized by a high-probabilistic set of distributions or *ambiguity set*, defined as a Wasserstein ball centered at $\hat{\mathbb{P}}_n$ with a radius ϵ determined by measure concentration results [40, Theorem 2]. We denote the ambiguity set by $\mathcal{P}_n := \mathbb{B}_\epsilon(\hat{\mathbb{P}}_n)$. In particular, for a given scalar $\beta \in (0, 1)$, the radius ϵ can be selected such that $\text{Prob}\,(\mathbb{P} \in \mathcal{P}_n) \geq 1 - \beta$, where $\text{Prob} := \mathbb{P}^n$ is the probability defined on the space of samples $\{\boldsymbol{\xi}_k\}_{k=1}^n$. With such characterization, we consider the DRO formulation of (1.1) as follows

$$\inf_{\ell \in \mathcal{L}} \sup_{\mathbb{Q} \in \mathcal{P}_n} \mathbb{E}_{\mathbb{Q}}[\ell(\boldsymbol{\xi})], \tag{1.2}$$

where, by construction, we have

$$\text{Prob}\left(\mathbb{E}_{\mathbb{P}}[\ell(\boldsymbol{\xi})] \leq \sup_{\mathbb{Q} \in \mathcal{P}_n} \mathbb{E}_{\mathbb{Q}}[\ell(\boldsymbol{\xi})]\right) \geq 1 - \beta, \quad \forall \ell \in \mathcal{L}.$$

Such a framework enables us to work on accessible Problem (1.2), which, in high probability $1 - \beta$, guarantees performance of (1.1) with a properly selected ℓ. In following chapters, various problem scenarios are tackled, and extensions of this framework will be addressed accordingly.

Chapter 2

Data Assimilation and Online Optimization

This chapter considers a class of real-time stochastic optimization problems dependent on an unknown probability distribution. In the considered scenario, data is streaming frequently while trying to reach a decision. Thus, we aim to devise a procedure that incorporates samples (data) of the distribution sequentially and adjusts decisions accordingly. We approach this problem in a distributionally robust optimization framework and propose a novel Online Data Assimilation Algorithm (OnDA Algorithm) for this purpose. This algorithm guarantees out-of-sample performance of decisions with high probability, and gradually improves the quality of the decisions by incorporating the streaming data. We show that the OnDA Algorithm converges under a sufficiently slow data streaming rate, and provide a criteria for its termination after certain number of data have been collected.

2.1 Related Works

Optimization under uncertainty is a vast research area, and as such, available methods include stochastic optimization [120] and robust optimization [10]. Recently, data-driven distributionally robust optimization has regained popularity thanks to its out-of-sample performance guarantees, see e.g. [36, 41] and [24, 25], for a distributed algorithm counterpart, and references therein. In this setup, one defines a set of distributions or *ambiguity set*, which contains the

true distribution of the data-generating system with high probability. Then, the out-of-sample performance of the data-driven decision is obtained as the worst-case optimization over the ambiguity set. An attractive way of designing these sets is to consider a ball in the space of probability distributions centered at a reference or most-likely distribution constructed from the available data. In the space of distributions, the popular distance metric is the Prokhorov metric [35], ϕ-divergence [57] and the Wasserstein distance [36]. In particular, the work [36] presents a tractable reformulation of DRO via Wasserstein balls, and is extended in [24] to a distributed setting. However, the available problem reformulation in [36] and the distributed algorithm in [24] do not consider the update of the decision over time and streaming data, which is the focus of this chapter. To design a tractable algorithm incorporating streaming data, this chapter connects to various convex optimization methods [12, 17] such as the Frank-Wolfe (FW) Algorithm (e.g., conditional gradient algorithm), the Subgradient Algorithm, and their variants, see e.g. [52, 58, 132] and references therein. Our emphasis on the convergence of the data-driven decision obtained through a sequence of optimization problems contrasts with typical algorithms developed for single (non-updated) problems.

Statement of Contributions

In this chapter, we propose a new Online Data Assimilation Algorithm (OnDA Algorithm) to solve decision-making problems subject to uncertainty. The distribution of the uncertainty is unknown and the algorithm adjusts decisions based on realizations of the stochastic variable sequentially revealed over time. The new algorithm addresses four challenges: 1) the evaluation of the out-of-sample performance of every possible online decision; 2) the adaptation to online, increasingly-larger data sets to reach a decision with performance guarantees with increasingly higher probabilities; 3) the availability of an online decision vector with performance guarantees at any time; 4) the capability of handling sufficiently large streaming data sets.

To address 1), we start from a DRO problem setting. This leads to a worst-case

optimization over an ambiguity set or neighborhood of the empirical distribution constructed from a data set. To solve this intractable problem, we reformulate it into an equivalent convex optimization over a simplex. This enables us to explore the simplex vertex set to find a certificate (a value bounding the cost) of a given decision with certain confidence. When the data is streaming, we consider a sequence of DRO problems and their equivalent convex reformulations employing increasingly larger data sets. Thus, as the data streams, the associated problems are defined over simplices of increasingly larger dimension. The similarities of the feasible sets allow us to assimilate the data via specialized Frank-Wolfe Algorithm variants, thus solving 2) via a Certificate Generation Algorithm (C-Gen Algorithm) described in Section 2.4. Further, to seek for decisions that approach to the minimizers of the optimization problem, the OnDA Algorithm adapts its iterations online via a Subgradient Algorithm as described in Section 2.5. We show in Section 2.6 that the resulting OnDA Algorithm is finitely convergent in the sense that the confidence of the out-of-sample performance guarantee for the generated data-driven decision converges to 1 as the number of data samples increases to a sufficiently large but finite value. Under this scheme, a data-driven decision with certain performance guarantee is also available any time as soon as the algorithm finishes generating the first certificate for the initial decision, which resolves the challenge 3). To expedite the algorithm and deal with challenge 4), we develop in Section 2.7 an Incremental Covering Algorithm (I-Cover Algorithm) to obtain low-dimensional ambiguity sets. These new sets are based on a weighted version of the empirical distribution and thus close to the full empirical distribution of the data. We finally illustrate the performance of the proposed OnDA Algorithm in Section 2.8, with and without the I-Cover Algorithm.

2.2 Problem Description

Consider a decision-making problem under of the form

$$\inf_{\boldsymbol{x}\in\mathbb{R}^d} \mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x},\xi)], \tag{P}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the decision variable, the uncertainty random variable $\xi : \Omega \to \mathbb{R}^m$ is induced by the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and the expectation of f is taken w.r.t. the unknown distribution $\mathbb{P} \in \mathcal{M}(\mathcal{Z})$. We aim to develop an Online Data Assimilation Algorithm (OnDA Algorithm) that efficiently adapts iterations on decisions $\boldsymbol{x}$ of (P) with streaming data. The streaming data are sequentially available iid realizations of the random variable ξ under $\mathbb{P}$, denoted by ξ_n, $n = 1, 2, \ldots$. This defines a sequence of streaming data sets, $\Xi_n \subseteq \Xi_{n+1}$, for each n. W.l.o.g. assume that each Ξ_{n+1} consists of just one more new data point, i.e., $\Xi_{n+1} = \Xi_n \cup \{\xi_{n+1}\}$ and $\Xi_1 = \{\xi_1\}$. In the following, we refer to the time slot between the updates Ξ_n and Ξ_{n+1} as the n^{th}*-time period* and to its rate of change as the *data-streaming rate*.

In practice, we cannot evaluate the objective function of (P) because $\mathbb{P}$ is unknown. We call a decision $\boldsymbol{x} \in \mathbb{R}^d$ *a proper data-driven decision* of (P), if its *out-of-sample performance*, defined by $\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x},\xi)]$, satisfies the *performance guarantee*

$$\mathbf{P}^n(\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x},\xi)] \le J_n(\boldsymbol{x})) \ge 1 - \beta_n, \tag{2.1}$$

where the expected cost upper bound or *certificate* $J_n(\boldsymbol{x})$ is a function that indicates the goodness of $\boldsymbol{x}$ under the data set Ξ_n. If $\boldsymbol{x}$ is adopted during the n^{th} time period, then $\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x},\xi)] \le J_n(\boldsymbol{x})$ is an event that depends on the n samples in Ξ_n, and $\mathbf{P}^n$ denotes the probability with respect to these. The *confidence* $1 - \beta_n \in (0, 1) \subset \mathbb{R}$ governs the choice of $\boldsymbol{x}$ and the resulting certificate $J_n(\boldsymbol{x})$. In words, the inequality (2.1) establishes that, given finite data n, the performance of the decision under the unknown distribution will not surpass the upper-bound certificate $J_n(\boldsymbol{x})$ with high probability. In the following section, we will determine the values J_n via the solution of a

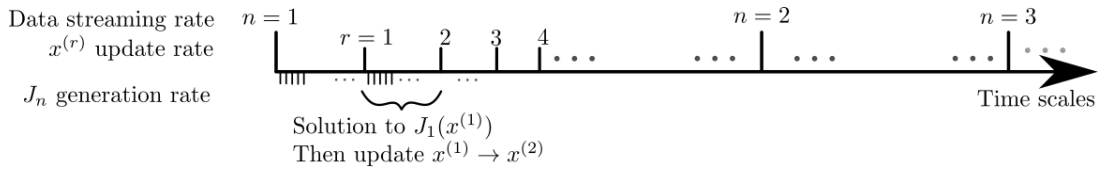


Figure 2.1. Time scales of Online Data Assimilation Algorithm.

parameterized maximization problem over $\boldsymbol{x}$. Therefore, finding an approximate certificate will be much easier than finding the exact one. Based on this, we call $\boldsymbol{x}$ ϵ_1-*proper*, if it satisfies (2.1) with $J_n^{\epsilon_1}(\boldsymbol{x})$ such that $J_n(\boldsymbol{x}) \leq J_n^{\epsilon_1}(\boldsymbol{x}) + \epsilon_1$. Thus, the approximates $J_n^{\epsilon_1}(\boldsymbol{x})$ also provide upper bounds to the optimal value of (P) with high confidence $1 - \beta_n$.

To sum up, for any n, given a confidence level $1 - \beta_n$, our goal is to approach to an ϵ_1-proper data-driven decision with a low certificate. Later in Section 2.5, we will show, under assumptions on f, that both Problem (P) and these certificates J_n are convex. To find a decision with a low certificate, we will call any proper data-driven decision ϵ_2-*optimal*, labeled as $\boldsymbol{x}_n^{\epsilon_2}$, if $J_n(\boldsymbol{x}_n^{\epsilon_2}) - J_n(\boldsymbol{x}) \leq \epsilon_2$ for all $\boldsymbol{x} \in \mathbb{R}^d$. Then, for any ϵ_2-optimal and ϵ_1-proper data-driven decision $\boldsymbol{x}_n^{\epsilon_2}$ with certificate $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})$ and $\epsilon_1 \ll \epsilon_2$, we have the guarantee

$$\mathbf{P}^n(\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}_n^{\epsilon_2}, \xi)] \leq J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) + \epsilon_1) \geq 1 - \beta_n. \tag{2.2}$$

Then, any decision $\boldsymbol{x}_n^{\epsilon_2}$ ensures a high-confidence, potentially-low objective value of (P), upper bounded by $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) + \epsilon_1$.

2.2.1 Motivating Example in Portfolio Optimization

Consider an agent who does short-term trading, i.e., she is to select a minute-based portfolio weight $\boldsymbol{x} := (x, 1 - x)$, $0 \leq x \leq 1$, for two risky assets that give random returns, with return rates $\xi := (\xi_1, \xi_2)$ following some unknown distribution $\mathbb{P}$. The agent aims to select $\boldsymbol{x}$ such that the expected profit is maximized, or equivalently, she seeks to solve

$$\min_{0 \leq x \leq 1} \mathbb{E}_{\mathbb{P}}[-\xi_1 x - \xi_2(1 - x)]. \tag{P0}$$

Assume that $\mathbb{P}$ is unknown and independent from the selection of the portfolio and that, at every minute, the agent has access to return rates, which are iid samples of $\mathbb{P}$. Due to the independence of $\mathbb{P}$ and $\boldsymbol{x}$, Problem (P0) is equivalent to $\min_{0\le x\le 1} \mathbb{E}_{\mathbb{P}}[-\xi_1 x - \xi_2(1-x) - \xi^\top \xi]$. To adapt this problem to our unconstrained setting, we consider an approximation

$$\min_x \mathbb{E}_{\mathbb{P}}[-\xi_1 x - \xi_2(1-x) - \xi^\top \xi] - \rho\,(\log(x) + \log(1-x)),$$

where $\rho > 0$ is some penalty for the constraint terms. This problem is in form (P), fitting in our problem setting with

$$f(\boldsymbol{x}, \xi) := -\xi_1 x - \xi_2(1-x) - \rho\,(\log(x) + \log(1-x)) - \xi^\top \xi.$$

Our proposed algorithm allows the agent to make online decisions $\boldsymbol{x}_n^{\epsilon_2}$ that minimize the objective with high confidence.

2.2.2 High-level Goal and Procedure

We describe now the goal of the OnDA Algorithm that handles a streaming sequence of data sets with $n \in \{1, \ldots, N\}$. Let tolerance parameters ϵ_1 and ϵ_2 be given and let us choose strictly increasing confidence levels $\{1-\beta_n\}_{n=1}^N$ such that $\sum_{n=1}^\infty \beta_n < \infty$ whenever $N \to \infty$. The algorithm aims to find a sequence of ϵ_2-optimal and ϵ_1-proper decisions $\{\boldsymbol{x}_n^{\epsilon_2}\}_{n=1}^N$ associated with the sequence of the certificates $\{J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})\}_{n=1}^N$ so that (2.2) holds for all $n \in \{1, \ldots, N\}$. Additionally, as the data set streams to infinite cardinality, i.e., $N \to \infty$, there exists a large enough but finite n_0 such that the algorithm returns a final $\boldsymbol{x}_{n_0}^{\epsilon_2}$ after processing the data set Ξ_{n_0}. The final decision $\boldsymbol{x}_{n_0}^{\epsilon_2}$ guarantees performance almost surely, that is, $\mathbf{P}^{n_0}(\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}_{n_0}^{\epsilon_2}, \xi)] \le J_{n_0}^{\epsilon_1}(\boldsymbol{x}_{n_0}^{\epsilon_2}) + \epsilon_1) = 1$, with a certificate $J_{n_0}^{\epsilon_1}(\boldsymbol{x}_{n_0}^{\epsilon_2})$ close to the optimal objective value of Problem (P).

To achieve this goal, the algorithm will output a sequence of decisions, $\{\boldsymbol{x}^{(r)}\}_{r=1}^\infty$, on a time scale that is faster than the data-streaming rate. We refer to this as the *decision-update rate*

and we use a parenthetical superscript (r) to denote its iterations. New data arrival will reset the OnDA Algorithm's subroutines to update the sub-sequences of decisions in each n^{th} time period efficiently. We denote these sub-sequences as $\{\boldsymbol{x}^{(r)}\}_{r=r_n}^{r_{n+1}}$, where $\boldsymbol{x}^{(r_n)}$ is the initial decision adapted from the the time period $n-1$. These updates will require the computation of certificates (cost upper bounds) and the progressive reduction of these bounds.

The computation of certificates is carried out by the Certificate Generation Algorithm (or C-Gen Algorithm, for short). Given a current decision value, $\boldsymbol{x}^{(r)}$, and data set Ξ_n, the C-Gen Algorithm finds an ϵ_1 certificate $J_n^{\epsilon_1}(\boldsymbol{x}^{(r)})$ and a worst-case distribution associated with the data set. It operates on a faster time scale than the decision-update rate, the so-called *certificate-generation rate*. Upon the receipt of new data, this algorithm will reset as described in Section 2.4.

The second process of the OnDA Algorithm relies on iterating decisions to reduce the values of the functions $J_n^{\epsilon_1}(\boldsymbol{x})$. This employs the Subgradient Algorithm and is described in Section 2.5. A more thorough description of how new data triggers a reset in the algorithm is described in the following sections. A summary of the OnDA Algorithm can be found in Section 2.6 as well as a descriptive table.

2.3 Certificate Design

In this section, we present a tractable formulation of certificates $J_n(\boldsymbol{x})$ and its approximation $J_n^{\epsilon_1}(\boldsymbol{x})$ for a fixed $\boldsymbol{x} = \boldsymbol{x}^{(r)}$, as described in (2.1) and (2.2), respectively. To achieve this, we first follow [24, 25, 36] on DRO to find certificates J_n. This defines a parameterized maximization problem for J_n, called Problem (P1$_n$). Then we reformulate (P1$_n$) as Problem (P2$_n$), a convex optimization problem over a simplex, for efficient solutions of approximated certificates $J_n^{\epsilon_1}$ in the next section.

To design J_n, a reasonable attempt is to use the data Ξ_n to estimate an empirical distribution, $\hat{\mathbb{P}}^n$, and let $\mathbb{E}_{\hat{\mathbb{P}}^n}[f(\boldsymbol{x}, \xi)]$ be the candidate certificate for the performance guarantee (2.1). More precisely, assume that the data set Ξ_n are uniformly sampled from $\mathbb{P}$. The discrete empirical

probability measure associated with Ξ_n is the following $\hat{\mathbb{P}}^n := \frac{1}{n}\sum_{k=1}^n \delta_{\{\xi_k\}}$, where $\delta_{\{\xi_k\}}$ is the Dirac measure at ξ_k. The candidate certificate is

$$J_n^{\mathrm{sae}}(\boldsymbol{x}) := \mathbb{E}_{\hat{\mathbb{P}}^n}[f(\boldsymbol{x},\xi)] = \frac{1}{n}\sum_{k=1}^n f(\boldsymbol{x},\xi_k).$$

The above approximation $\hat{\mathbb{P}}^n$ of $\mathbb{P}$, also known as the *sample-average estimate*, makes J_n^{sae} easy to compute. However, such value only results in an approximation of the unknown out-of-sample performance $\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x},\xi)]$. Following [24, 36], we are to determine an *ambiguity set* $\mathcal{P}_n$ containing all the possible probability distributions supported on $\mathcal{Z} \subseteq \mathbb{R}^m$ that can generate Ξ_n with high confidence. Then, one can consider the worst-case expectation of $f(\boldsymbol{x},\xi)$ with respect to all distributions contained in $\mathcal{P}_n$. The solution to such problem offers an upper bound for the out-of-sample performance with high probability in the form of (2.1), and we refer to this upper bound as the certificate of the decision $\boldsymbol{x}$.

In order to quantify the ambiguity set and certificate for an ϵ_1-proper data-driven decision, we denote by $\mathcal{M}_{\mathrm{lt}}(\mathcal{Z}) \subset \mathcal{M}(\mathcal{Z})$ the set of light-tailed probability measures in $\mathcal{M}(\mathcal{Z})$, and introduce the following assumption for $\mathbb{P}$

Assumption 1 (Light tailed unknown distributions). *It holds that $\mathbb{P} \in \mathcal{M}_{\mathrm{lt}}(\mathcal{Z})$, i.e., there exists an exponent $a > 1$ such that $b := \mathbb{E}_{\mathbb{P}}[\exp(\|\xi\|^a)] < \infty$.*

Assumption 1 validates the following modern measure concentration result, which provides an intuition for considering the Wasserstein ball $\mathbb{B}_\epsilon(\hat{\mathbb{P}}^n)$ of center $\hat{\mathbb{P}}^n$ and radius ϵ as the ambiguity set $\mathcal{P}_n$.

Theorem 1 (Measure concentration [40, Theorem 2]). *If $\mathbb{P} \in \mathcal{M}_{\mathrm{lt}}(\mathcal{Z})$, then*

$$\mathbf{P}^n\{d_W(\mathbb{P},\hat{\mathbb{P}}^n) \geq \epsilon\} \leq \begin{cases} c_1 e^{-c_2 n \epsilon^{\max\{2,m\}}}, & \text{if } \epsilon \leq 1, \\ c_1 e^{-c_2 n \epsilon^a}, & \text{if } \epsilon > 1, \end{cases} \tag{2.3}$$

for all $n \geq 1$, $m \neq 2$, and $\epsilon > 0$, where c_1, c_2 are positive constants that depend on m, a, and b. □

Equipped with this result, we are able to provide the certificate that ensures the performance guarantee in (2.1), for any decision $\boldsymbol{x} \in \mathbb{R}^d$.

Lemma 1 (Certificate in performance guarantee (2.1)**).** *Given* $\Xi_n := \{\xi_k\}_{k=1}^n$, $\beta_n \in (0,1)$ *and* $\boldsymbol{x} \in \mathbb{R}^d$, *let*

$$\epsilon(\beta_n) := \begin{cases} \left(\frac{\log(c_1\beta_n^{-1})}{c_2 n}\right)^{1/\max\{2,m\}}, & \text{if } n \geq \frac{\log(c_1\beta_n^{-1})}{c_2}, \\ \left(\frac{\log(c_1\beta_n^{-1})}{c_2 n}\right)^{1/a}, & \text{if } n < \frac{\log(c_1\beta_n^{-1})}{c_2}, \end{cases} \tag{2.4}$$

and $\mathcal{P}_n := \mathbb{B}_{\epsilon(\beta_n)}(\hat{\mathbb{P}}^n)$. *Then the following certificate satisfies the performance guarantee in* (2.1) *for all* $\boldsymbol{x} \in \mathbb{R}^d$

$$J_n(\boldsymbol{x}) := \sup_{\mathbb{Q}\in\mathcal{P}_n} \mathbb{E}_{\mathbb{Q}}[f(\boldsymbol{x},\xi)]. \tag{2.5}$$

Proof. Following [24, 36] and from Theorem 1, we prove that $J_n(\boldsymbol{x})$ is a valid certificate for (2.1). Knowing that (2.4) is obtained by letting the right-hand side of (2.3) to be equal to a given β_n, for each n we substitute (2.4) into the right-hand side of (2.3), yielding $\mathbf{P}^n\{d_W(\mathbb{P},\hat{\mathbb{P}}^n) \geq \epsilon(\beta_n)\} \leq \beta_n$ for each n. This means that a data set Ξ_n we can construct an empirical probability measure $\hat{\mathbb{P}}^n$ such that $d_W(\mathbb{P},\hat{\mathbb{P}}^n) \leq \epsilon(\beta_n)$ with probability at least $1-\beta_n$. Namely, $\mathbf{P}^n\{\mathbb{P} \in \mathbb{B}_{\epsilon(\beta_n)}(\hat{\mathbb{P}}^n)\} \geq 1-\beta_n$. Thus, for all $\boldsymbol{x} \in \mathbb{R}^d$, we have $\mathbf{P}^n\{\mathbb{P} \in \mathbb{B}_{\epsilon(\beta_n)}(\hat{\mathbb{P}}^n)\} = \mathbf{P}^n\{\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x},\xi)] \leq \sup_{\mathbb{Q}\in\mathcal{P}_n} \mathbb{E}_{\mathbb{Q}}[f(\boldsymbol{x},\xi)]\} = \mathbf{P}^n\{\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x},\xi)] \leq J_n(\boldsymbol{x})\} \geq 1-\beta_n$. □

To get J_n in (2.5), one needs to solve an infinite-dimensional optimization problem. Luckily, Problem (2.5) can be reformulated into a finite-dimensional convex problem as follows.

Theorem 2 (Convex reduction of (2.5) **[36, Application of Theorem 4.4]).** *Under Assumption 1, on* $\mathbb{P}$ *being light-tailed, for all* $\beta_n \in (0,1)$ *the value of the certificate in* (2.5) *for a given decision*

$\boldsymbol{x}$ under a data set Ξ_n is equal to the optimal value of the following optimization problem

$$\begin{aligned} J_n(\boldsymbol{x}) := \sup_{\boldsymbol{y}} \ & \frac{1}{n}\sum_{k=1}^{n} f(\boldsymbol{x}, \xi_k - \boldsymbol{y}_k), \\ \text{s.t.} \ & \frac{1}{n}\sum_{k=1}^{n} \|\boldsymbol{y}_k\|_1 \leq \epsilon(\beta_n), \end{aligned} \tag{P1$_n$}$$

where each component of the concatenated variable $\boldsymbol{y} := (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$ is in $\mathbb{R}^m$, and the parameter $\epsilon(\beta_n)$ is the radius of $\mathbb{B}_{\epsilon(\beta_n)}$ calculated from (2.4)*. Moreover, given any feasible point $\boldsymbol{y}^{(l)} := (\boldsymbol{y}_1^{(l)}, \ldots, \boldsymbol{y}_n^{(l)})$ of* (P1$_n$)*, indexed by l, define a finite atomic probability measure at $\boldsymbol{x}$ in the Wasserstein ball $\mathbb{B}_{\epsilon(\beta_n)}$ of the form*

$$\mathbb{Q}_n^{(l)}(\boldsymbol{x}) := \frac{1}{n}\sum_{k=1}^{n} \delta_{\{\xi_k - \boldsymbol{y}_k^{(l)}\}}. \tag{2.6}$$

Now, denote by $\mathbb{Q}_n^\star(\boldsymbol{x})$ the distribution in (2.6) *constructed by an optimizer $\boldsymbol{y}^\star := (\boldsymbol{y}_1^\star, \ldots, \boldsymbol{y}_n^\star)$ of* (P1$_n$) *and evaluated over $\boldsymbol{x}$. Then, $\mathbb{Q}_n^\star$ is a worst-case distribution that can generate the data set Ξ_n with (high) probability no less than $1 - \beta_n$.* □

Remark 2. Theorem 2 provides a way of computing certificates of (2.1) as the solution to a parameterized optimization problem for a decision $\boldsymbol{x}$. In addition, it constructs a worst-case distribution that achieves the worst-case bound.

To enable efficient online solutions of approximated certificates $J_n^{\epsilon_1}$, let us define parameterized functions $h_k : \mathbb{R}^m \to \mathbb{R}$

$$h_k(\boldsymbol{y}) := f(\boldsymbol{x}, \xi_k - \boldsymbol{y}), \quad k \in \{1, \ldots, n\},$$

and consider the following convex optimization problem over a simplex

$$\begin{aligned} J_n(\boldsymbol{x}) := \max_{\boldsymbol{u},\boldsymbol{v}} \ & \frac{1}{n}\sum_{k=1}^{n} h_k(\boldsymbol{u}_k - \boldsymbol{v}_k), \\ \text{s.t.} \ & (\boldsymbol{u}, \boldsymbol{v}) \in n\epsilon(\beta_n)\Delta_{2mn}, \end{aligned} \tag{P2$_n$}$$

where the concatenated variable $(\boldsymbol{u}, \boldsymbol{v})$ is composed of $\boldsymbol{u} := (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n)$ and $\boldsymbol{v} := (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ with $\boldsymbol{u}_k, \boldsymbol{v}_k \in \mathbb{R}^m$ for all $k \in \{1, \ldots, n\}$; and the scalar $n\epsilon(\beta_n)$ regulates the size of the feasible set via scaling of the unit simplex $\Delta_{2mn} := \{(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^{2mn} \mid \mathbf{1}_{2mn}{}^\top(\boldsymbol{u}, \boldsymbol{v}) = 1,\ \boldsymbol{u} \geq 0,\ \boldsymbol{v} \geq 0\}$. We denote by Λ_{2mn} the set of all the extreme points for the simplex $n\epsilon(\beta_n)\Delta_{2mn}$.

The following lemma shows that Problem ($\mathrm{P1}_n$) and Problem ($\mathrm{P2}_n$) are equivalent. Thus, we can approximately solve ($\mathrm{P2}_n$) to find $J_n^{\epsilon_1}(\boldsymbol{x})$ and $\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x})$.

Lemma 2 (Equivalence of the problem formulation). *Solving* ($\mathrm{P1}_n$) *is equivalent to solving* ($\mathrm{P2}_n$) *in the sense that*

1 *For any feasible solution* $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}})$ *of* ($\mathrm{P2}_n$)*, let* $\tilde{\boldsymbol{y}} := \tilde{\boldsymbol{u}} - \tilde{\boldsymbol{v}}$*. Then* $\tilde{\boldsymbol{y}}$ *is feasible for* ($\mathrm{P1}_n$)*.*

2 *For any feasible solution* $\tilde{\boldsymbol{y}}$ *of* ($\mathrm{P1}_n$)*, there exists a feasible point* $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}})$ *of* ($\mathrm{P2}_n$)*.*

3 *Assume that the point* $(\tilde{\boldsymbol{u}}^\star, \tilde{\boldsymbol{v}}^\star)$ *is an optimizer of* ($\mathrm{P2}_n$)*. Then by letting* $\tilde{\boldsymbol{y}}^\star := \tilde{\boldsymbol{u}}^\star - \tilde{\boldsymbol{v}}^\star$*, the point* $\tilde{\boldsymbol{y}}^\star$ *is also an optimizer of* ($\mathrm{P1}_n$)*, with the same optimal value.*

Proof. To prove 1, for any feasible solution $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}})$ of ($\mathrm{P2}_n$), we compute

$$\begin{aligned}\frac{1}{n}\sum_{k=1}^{n}\|\tilde{\boldsymbol{y}}_k\|_1 &= \frac{1}{n}\sum_{k=1}^{n}\|\tilde{\boldsymbol{u}}_k - \tilde{\boldsymbol{v}}_k\|_1 \\ &\leq \frac{1}{n}\sum_{k=1}^{n}\|\tilde{\boldsymbol{u}}_k\|_1 + \frac{1}{n}\sum_{k=1}^{n}\|\tilde{\boldsymbol{v}}_k\|_1 = \frac{1}{n}\mathbf{1}_{mn}{}^\top\tilde{\boldsymbol{u}} + \frac{1}{n}\mathbf{1}_{mn}{}^\top\tilde{\boldsymbol{v}} = \frac{1}{n}\mathbf{1}_{2mn}{}^\top(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) = \epsilon(\beta_n).\end{aligned}$$

Therefore $(\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_n)$ is feasible for ($\mathrm{P1}_n$).

For 2, we exploit that any feasible solution $\tilde{\boldsymbol{y}}$ of ($\mathrm{P1}_n$) is a linear combination of the extreme points of the constraint set in ($\mathrm{P1}_n$). Let us denote the matrix $A_n := [\oplus_{i=1}^n I_m, -\oplus_{i=1}^n I_m] \in \mathbb{R}^{mn \times 2mn}$. By construction of Problem ($\mathrm{P2}_n$), we see that each column vector of the matrix $n\epsilon(\beta_n)A_n$ is a concatenated vector of an extreme point of Problem ($\mathrm{P1}_n$), and that all the extreme points of ($\mathrm{P1}_n$) are included. Then, any feasible solution of ($\mathrm{P1}_n$) can be written as $\tilde{\boldsymbol{y}} = n\epsilon(\beta_n)A_n(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$ where $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$ is a vector of the convex combination coefficients of the extreme points of the

constraint set in (P1$_n$). Clearly, we have $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}) \in \Delta_{2mn}$, i.e., $n\epsilon(\beta_n)(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$ is in the feasible set of the Problem (P2$_n$). Then, by construction $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) := n\epsilon(\beta_n)(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$ is feasible for (P2$_n$).

For 3, since (P1$_n$) and (P2$_n$) are the same in the sense of (1) and (2), then if $(\tilde{\boldsymbol{u}}^\star, \tilde{\boldsymbol{v}}^\star)$ is an optimizer of (P2$_n$), by letting $\tilde{\boldsymbol{y}}_k^\star := \tilde{\boldsymbol{u}}_k^\star - \tilde{\boldsymbol{v}}_k^\star$ for each $k \in \{1, \ldots, n\}$ we know the objective values of the two problems coincide. We claim that the optimum of (P1$_n$) is achieved via the optimizer $\tilde{\boldsymbol{y}}^\star$. If not, then there exists $\hat{\boldsymbol{y}}^\star \neq \tilde{\boldsymbol{y}}^\star$ such that the optimum is achieved with higher value. Then, from the construction in (2) we can find a feasible solution $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$ of (P2$_n$) that results in a higher objective value. This contradicts the assumption that $(\tilde{\boldsymbol{u}}^\star, \tilde{\boldsymbol{v}}^\star)$ is an optimizer of (P2$_n$). □

2.4 Certificate Generation Algorithm

Given a tolerance ϵ_1, sequentially available data sets $\{\Xi_n\}_{n=1}^N$ and decisions $\{\boldsymbol{x}^{(r)}\}_{r=1}^\infty$, we present in this section the Certificate Generation Algorithm (C-Gen Algorithm) to obtain approximated certificates $\{J_n^{\epsilon_1}(\boldsymbol{x}^{(r)})\}_{n,r}$ and associated ϵ_1-worst-case distributions $\{\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})\}_{n,r}$. To achieve this, we first design for each fixed $\boldsymbol{x} = \boldsymbol{x}^{(r)}$ the C-Gen Algorithm to solve (P2$_n$) to $J_n^{\epsilon_1}(\boldsymbol{x})$ efficiently. This is developed via Frank-Wolfe Algorithm variants, e.g., the Simplicial Algorithm [52] and the AFWA as described in the Appendix. Then we analyze the convergence of the C-Gen Algorithm under $\{\Xi_n\}_{n=1}^N$.

2.4.1 The C-Gen Algorithm

For each fixed $\boldsymbol{x} = \boldsymbol{x}^{(r)} \in \mathbb{R}^d$ the algorithm is run at a fast time scale (the certificate-update rate), over iterations $l = 0, 1, 2, \ldots$. The algorithm is then employed inside the OnDA Algorithm, so its execution rate is the fastest within this algorithm. At each iteration l, the C-Gen Algorithm generates $(\boldsymbol{u}^{(l)}, \boldsymbol{v}^{(l)})$, the candidate optimizer of (P2$_n$). Let the objective value of (P2$_n$) at $(\boldsymbol{u}^{(l)}, \boldsymbol{v}^{(l)})$ be $J_n^{(l)}(\boldsymbol{x})$, and, equivalently, write the candidate optimizer in form of $\boldsymbol{y}^{(l)} := \boldsymbol{u}^{(l)} - \boldsymbol{v}^{(l)}$ (exploiting the equivalence in Lemma 2). Each candidate $\boldsymbol{y}^{(l)}$ is associated with a set of search points denoted by $I_n^{(l)} := \{\tilde{\boldsymbol{y}}^{[i]} := \tilde{\boldsymbol{u}}^{[i]} - \tilde{\boldsymbol{v}}^{[i]},\ i \in \{1, \ldots, T\}\}$, where we use bracket superscript $[i]$ to index

C-Gen Algorithm 1. $\mathrm{CG}(\boldsymbol{x}, \{\Xi_n\}_{n=1}^N, \boldsymbol{y}^{(0)}, I_n^{(0)})$

Require: Goes to Step 1 upon data arrival, i.e. $\Xi_n \leftarrow \Xi_{n+1}$.

1: $l \leftarrow 0$; ▷ Procedure for Ξ_n
2: Update $\boldsymbol{y}^{(l)}$, $I_n^{(l)}$, T and γ^{ϵ_1}; ▷ Adapted from Ξ_{n-1}
3: **repeat**
4: $l \leftarrow l + 1$;
5: $(\Omega^{(l)}, \eta^{(l)}) \leftarrow \mathrm{LP}(\boldsymbol{x}, \Xi_n, \boldsymbol{y}^{(l-1)})$;
6: $I_n^{(l)} \leftarrow I_n^{(l-1)} \cup \Omega^{(l)}$, $T \leftarrow |I_n^{(l)}|$;
7: $(\gamma^{\epsilon_1}, J_n^{(l)}(\boldsymbol{x})) \leftarrow \mathrm{AFWA}\ (\mathrm{CP}_n^{(l)})$;
8: $\boldsymbol{y}^{(l)} \leftarrow \sum_{i=0}^T \gamma_i^{\epsilon_1} \tilde{\boldsymbol{y}}^{[i]}$, $\tilde{\boldsymbol{y}}^{[i]} \in I_n^{(l)}$ for each i;
9: **until** $\eta^{(l)} \leq \epsilon_1$
10: **return** $J_n^{\epsilon_1}(\boldsymbol{x}) = J_n^{(l)}(\boldsymbol{x})$, $\boldsymbol{y}^{\epsilon_1} = \boldsymbol{y}^{(l)}$, $\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}) = \frac{1}{n}\sum_{k=1}^n \delta_{\{\xi_k - \boldsymbol{y}_k^{\epsilon_1}\}}$.

its elements. As we will see later, the set $I_n^{(l)}$ plays a key role in generating the certificate when assimilating data, and is called the *candidate vertex set*.

Given a data set Ξ_n, and until new data arrives, the C-Gen Algorithm solves the following problems alternatively

$$\begin{aligned} \max_{\boldsymbol{u},\boldsymbol{v}}\ & \frac{1}{n}\sum_{k=1}^n \left\langle \nabla h_k(\boldsymbol{y}_k^{(l-1)}),\ \boldsymbol{u}_k - \boldsymbol{v}_k - \boldsymbol{y}_k^{(l-1)} \right\rangle, \\ \text{s.t.}\quad & (\boldsymbol{u}, \boldsymbol{v}) \in n\epsilon(\beta_n)\Delta_{2mn}, \end{aligned} \tag{$\mathrm{LP}_n^{(l)}$}$$

$$\begin{aligned} \max_{\gamma \in \mathbb{R}^T}\ & \frac{1}{n}\sum_{k=1}^n h_k\Big(\sum_{i=0}^T \gamma_i \tilde{\boldsymbol{y}}_k^{[i]}\Big), \\ \text{s.t.}\quad & \gamma \in \Delta_T. \end{aligned} \tag{$\mathrm{CP}_n^{(l)}$}$$

(Note how the solution to one problem parameterizes the other.) In this way, $\boldsymbol{y}^{(l-1)}$ (the solution to the $\mathrm{CP}_n^{(l-1)}$) parameterizes the linear problem ($\mathrm{LP}_n^{(l)}$). The solution to ($\mathrm{LP}_n^{(l)}$) is then used to refine the set point $I_n^{(l)} = \{\tilde{\boldsymbol{y}}^{[i]}\}_i$, which spans the constraint set $\Delta_T \equiv \mathrm{conv}(I_n^{(l)})$ in problem ($\mathrm{CP}_n^{(l)}$). A solution to ($\mathrm{CP}_n^{(l)}$) then determines the new $\boldsymbol{y}^{(l)}$ of the next LP problem. This process corresponds to lines 3: to 9: in the following C-Gen Algorithm table.

More precisely, at each iteration $l = 1, 2, \ldots$, the C-Gen Algorithm first solves ($\mathrm{LP}_n^{(l)}$)

Point Search Algorithm 2. LP($\boldsymbol{x}, \Xi_n, \boldsymbol{y}^{(l-1)}$)

Set $\Omega^{(l)} := \varnothing$;
Let $H := \{(j,k) \mid j \in \{1,\ldots,m\},\ k \in \{1,\ldots,n\}\}$;
Let $S := \operatorname{argmax}_{(j,k)\in H}\{\pm\nabla_j h_k(\boldsymbol{y}_k^{(l-1)})\}$;
while $S \neq \varnothing$, **do**
    Pick $(\hbar,\ell) \in S$ and let $\tilde{\boldsymbol{y}} = \boldsymbol{0}_{mn}$;
    Update scalar $\tilde{\boldsymbol{y}}_{\hbar\ell} \leftarrow n\epsilon(\beta_n)\operatorname{sgn}(\nabla_\hbar h_\ell(\boldsymbol{y}_\ell^{(l-1)}))$;
    Update $\Omega^{(l)} \leftarrow \Omega^{(l)} \cup \{\tilde{\boldsymbol{y}}\}$;
    Update $S \leftarrow S \setminus \{(\hbar,\ell)\}$;
Pick any $\tilde{\boldsymbol{y}} \in \Omega^{(l)}$ and,
set $\eta^{(l)} = \frac{1}{n}\sum_{k=1}^n \left\langle \nabla h_k(\boldsymbol{y}_k^{(l-1)}), \tilde{\boldsymbol{y}}_k - \boldsymbol{y}_k^{(l-1)} \right\rangle$;
return the set $\Omega^{(l)}$ and the optimality gap $\eta^{(l)}$.

using the Point Search Algorithm, which returns the optimal objective value $\eta^{(l)}$ and the set of maximizers $\Omega^{(l)}$ such that $\eta^{(l)} \geq J_n(\boldsymbol{x}) - J_n^{(l)}(\boldsymbol{x})$ and $\Omega^{(l)} \subset \Lambda_{2mn}$. The value $\eta^{(l)}$ is then used to determine the ϵ_1-suboptimality condition to the optimal objective of Problem (P2$_n$) (see below). Meanwhile, the set $\Omega^{(l)}$ is used to update candidate vertex set to $I_n^{(l)} := I_n^{(l-1)} \cup \Omega^{(l)}$, which is used in problem (CP$_n^{(l)}$). In particular, the Point Search Algorithm computes all optimizers by iteratively choosing a sparse vector with only a positive entry. That is, an extreme point of the feasible set of (LP$_n^{(l)}$), such that the nonzero component of $(\tilde{\boldsymbol{u}}^{(l)}, \tilde{\boldsymbol{v}}^{(l)})$ has the largest absolute gradient component in the linear cost function of (LP$_n^{(l)}$). Using the obtained $I_n^{(l)}$, the algorithm solves the Problem (CP$_n^{(l)}$) over the simplex $\Delta_T := \{\gamma \in \mathbb{R}^T \mid \boldsymbol{1}_T{}^\top\gamma = 1,\ \gamma \geq 0\}$, where T is the cardinality of $I_n^{(l)}$ and each component γ_i of $\gamma \in \Delta_T$ represents the convex combination coefficient of a candidate vertex $\tilde{\boldsymbol{y}}^{[i]} \in I_n^{(l)}$. After solving (CP$_n^{(l)}$) to ϵ_1-optimality via the AFWA (see Appendix), an ϵ_1-optimal weighting $\gamma^{\epsilon_1} \in \Delta_T$ with the objective value $J_n^{(l)}(\boldsymbol{x})$ is obtained. A new candidate optimizer $\boldsymbol{y}^{(l)}$ is then calculated by $\boldsymbol{y}^{(l)} = \sum_{i=0}^T \gamma_i^{\epsilon_1}\tilde{\boldsymbol{y}}^{[i]}$. The algorithm repeats the process and increments l if the optimality gap $\eta^{(l)}$ is greater than ϵ_1, otherwise it returns the certificate $J_n^{\epsilon_1}(\boldsymbol{x}) := J_n^{(l)}(\boldsymbol{x})$, an ϵ_1-optimal solution $\boldsymbol{y}^{\epsilon_1} := \boldsymbol{y}^{(l)}$ and an ϵ_1-worst-case distribution $\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}) := \frac{1}{n}\sum_{k=1}^n \delta_{\{\xi_k - \boldsymbol{y}_k^{\epsilon_1}\}}$.

When new data arrives, the algorithm will reset by adapting the Problem (P2$_{n+1}$) from

Problem (P2$_n$) (line 2: in the table of the C-Gen Algorithm (update from Ξ_n to Ξ_{n+1}). Note that adapting the C-Gen Algorithm to online data sets $\{\Xi_n\}_{n=1}^N$ is inherently difficult due to the changes in the Problems (P2$_n$). As the size of Ξ_n grows by 1, the dimension of the Problem (P2$_n$) increases by $2m$. To obtain $J_n^{\epsilon_1}(\boldsymbol{x})$ and $\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x})$ sufficiently fast, we exploit the relationship among Problems (P2$_n$), for different n, by adapting the candidate vertex sets $I_n^{(l)}$. Specifically, we initialize the set $I_{n+1}^{(0)}$ for the new Problem (P2$_{n+1}$) by $I_n^{(l)}$, constructed from the previous (P2$_n$). Suppose that the C-Gen Algorithm receives a new data set $\Xi_{n+1} \supset \Xi_n$ at some intermediate iteration l with candidate vertex set $I_n^{(l)}$. At this stage, the subset conv($I_n^{(l)}$) has been explored by the previous optimization problem, and the gradient information of the objective function based on the data set Ξ_n has been partially integrated. Then, by projecting the set $I_n^{(l)}$ onto the set of extreme points of the new Problem (P2$_n$), i.e., $I_{n+1}^{(0)} := \text{proj}_{\Lambda_{2m(n+1)}}(\{(\tilde{\boldsymbol{y}}^{[i]}, \mathbf{0}_m) \mid \tilde{\boldsymbol{y}}^{[i]} \in I_n^{(l)}\})$, the subset conv($I_{n+1}^{(0)}$) of the feasible set of (P2$_n$) is already explored. Such integration contributes to the reduction of the number of iterations in the C-Gen Algorithm for Problems (P2$_n$). This insight gives us a sense of the worst-case efficiency to update a certificate under streaming data.

2.4.2 Convergence Analysis

We make the following assumptions on the local strong concavity of the function f and the computation of its gradient

Assumption 2 (Local strong concavity). *For any $\boldsymbol{x} \in \mathbb{R}^d$ and $\xi \in \mathbb{R}^m$, the function $h : \mathbb{R}^m \to \mathbb{R}$, $\boldsymbol{y} \mapsto f(\boldsymbol{x}, \xi - \boldsymbol{y})$ is differentiable, concave with a curvature constant C_h, and with a positive geometric strong concavity constant μ_h on Δ_{2mN} [1].*

Assumption 3 (Accessible gradients). *For any decision $\boldsymbol{x} \in \mathbb{R}^d$, we denote by $\nabla h(\boldsymbol{y})$ the gradient of the function $h : \mathbb{R}^m \to \mathbb{R}$, $\boldsymbol{y} \mapsto f(\boldsymbol{x}, \boldsymbol{y})$ and assume it is accessible.*

[1] For a concave function $h : \mathbb{R}^m \to \mathbb{R}$ on Δ, we define $C_h := \sup -\frac{2}{\gamma^2}\left(h(\boldsymbol{y}^\star) - h(\boldsymbol{y}) - \left\langle \nabla h(\boldsymbol{y}), \boldsymbol{y}^\star - \boldsymbol{y} \right\rangle\right)$, s.t. $\boldsymbol{y}^\star = \boldsymbol{y} + \gamma(\boldsymbol{s} - \boldsymbol{r})$, $\gamma \in [0,1]$, $\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{r} \in \Delta$. and $\mu_h := \inf_{\boldsymbol{y} \in \Delta} \inf_{\boldsymbol{y}^\star \in \Delta} -\frac{2}{\Gamma(\boldsymbol{y}, \boldsymbol{y}^\star)^2} \times \left(h(\boldsymbol{y}^\star) - h(\boldsymbol{y}) - \left\langle \nabla h(\boldsymbol{y}), \boldsymbol{y}^\star - \boldsymbol{y} \right\rangle\right)$, s.t. $\left\langle \nabla h(\boldsymbol{y}), \boldsymbol{y}^\star - \boldsymbol{y} \right\rangle > 0$, where $\Gamma(\boldsymbol{y}, \boldsymbol{y}^\star)$ is a step-size measure in AFWA. See, e.g., [58] for details. We say h is locally strongly concave, if $\mu_h > 0$.

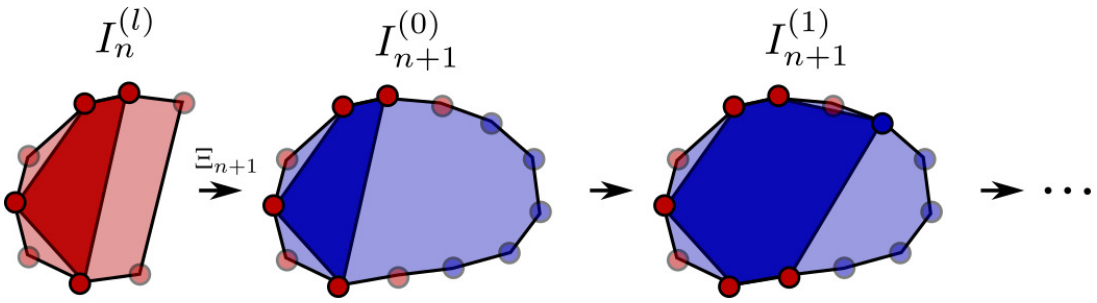


Figure 2.2. C-GEN Algorithm Procedure on a projected plane. At each particular time period (n or $n+1$) and iteration l, the dots, shaded region and solid region represent the projection of vertices of Δ, Δ and $\text{conv}(I)$, respectively. The solid region $\text{conv}(I)$ implicitly expands for solutions to various (P2$_n$).

Under Assumptions 2 and 3, we show the convergence properties of the C-GEN Algorithm.

Theorem 3 (Convergence of the C-GEN Algorithm). *Let a tolerance ϵ_1 and a decision $\boldsymbol{x}$ be given. Let us choose $\boldsymbol{y}^{(0)} = \mathbf{0}_m$ and $I_1^{(0)} = \varnothing$ as the initial candidate optimizer and candidate vertex set for the C-GEN Algorithm, respectively. Consider the online data sets $\{\Xi_n\}_{n=1}^N$ and the set of parameterized functions $\{h_n\}_{n=1}^N$. Under Assumption 2 and Assumption 3, we have that for all data set Ξ_n, there exists a parameter $\kappa \in (0,1) \subset \mathbb{R}$ such that the worst-case computational bound $\phi(n)$ of the C-GEN Algorithm, depending on n, is*

$$\phi(n) \leq (2mn)\log_\kappa\left(\frac{\epsilon_1}{J_n(\boldsymbol{x}) - J_n^{(0)}(\boldsymbol{x})}\right).$$

Moreover, consider that data sets $\{\Xi_n\}_{n=1}^N$ are streaming and consider function $J_N^{\text{sae}}(\boldsymbol{x})$ defined as in Section 2.3. Then there exists a parameter $\bar{\kappa} \in (0,1) \subset \mathbb{R}$ and a computational bound

$$\bar{\phi}(n) := (2mn)\log_{\bar{\kappa}}\left(\frac{\epsilon_1}{J_N(\boldsymbol{x}) - J_N^{\text{sae}}(\boldsymbol{x})}\right)$$

such that, if the data-streaming rate is slower or equal than $(\bar{\phi}(1))^{-1}$, then the C-GEN Algorithm is guaranteed to obtain the certificates $\{J_n^{\epsilon_1}(\boldsymbol{x})\}_{n=1}^N$ and $\{\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x})\}_{n=1}^N$.

Proof. Given tolerance ϵ_1, decision $\boldsymbol{x}$ and any data set Ξ_n with $n \in \{1,\ldots,N\}$, let $H_n : \mathbb{R}^{mn} \to \mathbb{R}$,

$H_n := \frac{1}{n}\sum_{k=1}^{n} h_k$ denote the objective function of (P2$_n$) and let $\mathcal{S}_n$ denote the family of subsets of Λ_{2mn}. In the procedure of C-Gen Algorithm, let us consider a sequence of generated candidate vertex sets: $I_n^{(l)} \subset I_n^{(l+1)}$, $l = 0, 1, 2, \ldots$ with $I_n^{(l)} \in \mathcal{S}_n$. We show the convergence of C-Gen Algorithm for any data set Ξ_n, by two steps.

Step 1: (Finite algorithm iterations) Here we show that the sequence $\{I_n^{(l)}\}_l$ is finite and the number of iterations is at most $2mn$. For each l and candidate optimizer $\boldsymbol{y}^{(l-1)}$, we generate a nonempty set of search points $\Omega^{(l)}$ with suboptimality gap $\eta^{(l)}$ via (LP$_n^{(l)}$). If $\eta^{(l)} \leq \epsilon_1$, then we solved (P2$_n$) to ϵ_1-optimality and l is therefore finite, otherwise we update $I_n^{(l)} := I_n^{(l-1)} \cup \Omega^{(l)}$. Given that the maximal cardinality of each $I_n^{(l)} \in \mathcal{S}_n$ is bounded by $2mn$, then it is sufficient to show $\Omega^{(l)} \cap I_n^{(l-1)} = \varnothing$. Because $\boldsymbol{y}^{(l-1)}$ is an ϵ_1-optimal of (CP$_n^{(l)}$) under conv($I_n^{(l-1)}$), then for any $\boldsymbol{y} \in \text{conv}(I_n^{(l-1)})$, it holds that $\frac{1}{n}\sum_{k=1}^{n}\langle \nabla h_k(\boldsymbol{y}_k^{(l-1)}), \boldsymbol{y}_k - \boldsymbol{y}_k^{(l-1)}\rangle \leq \epsilon_1$. Since any element in $\Omega^{(l)}$ is such that $\eta^{(l)} > \epsilon_1$, then for any $\boldsymbol{y} \in \text{conv}(I_n^{(l-1)})$, we have $\boldsymbol{y} \notin \Omega^{(l)}$, which concludes $\Omega^{(l)} \cap I_n^{(l-1)} = \varnothing$. Further, the cardinality of $\Omega^{(l)}$ is at least one for every iteration l, then after at most $2mn$ steps the cardinality of $I_n^{(l)}$ becomes $2mn$, which implies the ϵ_1-optimality of (P2$_n$) by the ϵ_1-optimality of (CP$_n^{(l)}$).

Step 2: (Quantification of the computational bound of C-Gen Algorithm) To see this, consider the problems $\{(\text{LP}_n^{(l)})\}_l$ and $\{(\text{CP}_n^{(l)})\}_l$. By Assumption 3 on the cheap access of the gradients, the computation of (LP$_n^{(l)}$) is negligible. Thus, the computational bound is given by the sum of the steps to solve the $\{(\text{CP}_n^{(l)})\}_l$, where the number of iterations l is $2mn$ in the worst case.

For each (CP$_n^{(l)}$) solved by AFWA, index the AFWA iterations by $i = 0, 1, 2, \ldots$, let $\text{obj}_i^{(l)}$ be the objective value at each iteration, and assume the optimal objective value is $\text{obj}_\star^{(l)}$. As in Theorem 14, let $\kappa_{n,l} \in (0, 1) \subset \mathbb{R}$ be the decay parameter related to local strong concavity of H_n over conv($I_n^{(l)}$). Then using the linear convergence rate of the AFWA, each (CP$_n^{(l)}$) achieves the following computational bound

$$\text{obj}_\star^{(l)} - \text{obj}_i^{(l)} \leq \kappa_{n,l}^{i}(\text{obj}_\star^{(l)} - \text{obj}_0^{(l)}),$$

where the initial condition $\text{obj}_0^{(l)}$ results from an ϵ_1-optimal optimizer of CP at iteration $l-1$, i.e., we can equivalently denote $\text{obj}_0^{(l)}$ by $J_n^{(l-1)}(\boldsymbol{x})$, for all $l \in \{1,\ldots,2mn\}$.

Let us consider sequence $\{(\text{CP}_n^{(l)})\}_l$ with feasible sets $\{\text{conv}(I_n^{(l)})\}_l$. Then we have

$$\text{conv}(I_n^{(0)}) \subset \text{conv}(I_n^{(1)}) \subset \cdots \subset \text{conv}(I_n^{(2mn)}).$$

This results into monotonically decaying parameters and (ϵ_1-)optimal objective values, as given in the following

$$\begin{aligned} 0 < \kappa_{n,1} \leq \kappa_{n,2} \leq \cdots \leq \kappa_{n,2mn} < 1, \\ J_n^{(0)}(\boldsymbol{x}) \leq J_n^{(1)}(\boldsymbol{x}) \leq \cdots \leq J_n^{(2mn)}(\boldsymbol{x}), \\ \text{obj}_\star^{(0)} \leq \text{obj}_\star^{(1)} \leq \cdots \leq \text{obj}_\star^{(2mn)}. \end{aligned}$$

Using the previous notation, we can identify $J_n^{(2mn)}(\boldsymbol{x}) \equiv J_n^{\epsilon_1}(\boldsymbol{x})$, $\text{obj}_\star^{(0)} \equiv J_n^{(0)}(\boldsymbol{x})$, and $\text{obj}_\star^{(2mn)} \equiv J_n(\boldsymbol{x})$. Let us denote $\kappa := \max_{n,l}\{\kappa_{n,l}\}$. Then, by solving each $(\text{CP}_n^{(l)})$ to ϵ_1-optimality, it leads to the accumulated computational steps $\phi(n) := \sum_l i_l$, where each i_l is the computation step for ϵ_1-optimal $(\text{CP}_n^{(l)})$ that satisfies the following inequality

$$\kappa^{i_l}(J_n(\boldsymbol{x}) - J_n^{(0)}(\boldsymbol{x})) \leq \epsilon_1, \quad l \in \{1,\ldots,2mn\}.$$

Finally, in the worst-case scenario, the computational bound of the C-Gen Algorithm is

$$\phi(n) \leq (2mn)\log_\kappa\left(\frac{\epsilon_1}{J_n(\boldsymbol{x}) - J_n^{(0)}(\boldsymbol{x})}\right).$$

Next, we show the convergence of the C-Gen Algorithm under online data sets $\{\Xi_n\}_{n=1}^N$. Similarly to the proof for the computational bound for a given n, we can compute the worst-case bound under $\{\Xi_n\}_{n=1}^N$, by summing over the steps required to solve the $\{(\text{CP}_n^{(l)})\}_{n,l}$. This leads to the stated bound $\bar{\phi}(n)$, where the empirical cost $J_N^{\text{sae}}(\boldsymbol{x})$ serves as the cost of initial condition $\boldsymbol{y}^{(0)} := \boldsymbol{0}_{2mN}$. In this way, when the data-streaming rate is slower or equal than $(\bar{\phi}(1))^{-1}$, we claim that C-Gen Algorithm can always find the certificate for each data set Ξ_n. This is because in each

time period n, we only have $2mn$ extreme points, and $2m(n-1)$ has been explored due to the adaptation of the candidate vertex set $I_n^{(0)}$. □

Theorem 3 relates the worst-case computational bound of the C-Gen Algorithm, executed on the certificate-update rate (the fastest of the time scales considered), to the data-streaming rate. Note that, as ϵ_1 decreases, the bound $\bar{\phi}(1)$ increases and therefore the smaller the data-streaming rate has to be so that the certificates can be generated by the algorithm. In practice, the C-Gen Algorithm tends to find the smallest implicit feasible set that contains an optimal solution of (P2$_n$). This means that the computation of the C-Gen Algorithm generally performs better than its worst-case bound as in Theorem 3 and so it can handle data-streaming rates faster than $(\bar{\phi}(1))^{-1}$. In the sequel, we assume that the C-Gen Algorithm converges with a rate that is faster than the worst-case bound in Theorem 3.

Remark 3 (Effects of Assumption 2 and 3). The essential ingredients for convergence of the Certificate Generation Algorithm are 1) the concavity of h, which ensures that (P2$_n$) is a convex problem, and 2) accessible gradients of h, which allows for computations to a solution of (P2$_n$). To obtain a fast, linear convergence rate as in Theorem 3, we assume that h is strongly concave on the simplex Δ_{2mN} located at each data point $\xi \in \Xi_n$. Intuitively, as Ξ_n comes from $\mathbb{P}$, Assumption 2 eventually requires h to be strongly concave on a subset of the support $\mathcal{Z}$ of $\mathbb{P}$ where the high-probability outcomes are concentrated onto. Otherwise, if h is concave but not locally strongly concave, or if the gradients of h are inaccessible (e.g., when only non-biased gradient estimate of h are available), the convergence of AFWA, as described in Theorem 14, reduces to a sublinear rate. This, in turn, reduces the computational bound $\bar{\phi}(n)$ in Theorem 3 to a bound of order $O(1/\epsilon_1)$.

Remark 4 (Example in portfolio optimization). The portfolio problem in Section 2.2 results in a strongly concave h which implies the local strong concavity as required by Assumption 2.

Let $\boldsymbol{y} := (y_1, y_2)$ and, for any given data point $\xi_k := (\xi_{k,1}, \xi_{k,2})$, ∇h is accessible and computed by

$$\nabla h(\boldsymbol{y}) := \begin{pmatrix} x + 2(\xi_{k,1} + \xi_{k,2} - y_1 - y_2) \\ 1 - x + 2(\xi_{k,1} + \xi_{k,2} - y_1 - y_2) \end{pmatrix}.$$

2.5 Sub-optimal Decisions with Guarantees

In this section, we aim to construct a sub-sequence of ϵ_2-optimal data-driven decisions $\{\boldsymbol{x}_n^{\epsilon_2}\}_{n=1}^N$, associated with the ϵ_2-lowest certificates $\{J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})\}_{n=1}^N$ over time. We achieve this by means of the Subgradient Algorithm to derive an ϵ_1-proper decision sequence $\{\boldsymbol{x}^{(r)}\}_{r=r_n}^{r_{n+1}}$; and the concatenation of $\{\boldsymbol{x}^{(r)}\}_{r=r_n}^{r_{n+1}}$ for different n to obtain $\{\boldsymbol{x}_n^{\epsilon_2}\}_{n=1}^N$. To construct an ϵ_1-proper decision sub-sequence $\{\boldsymbol{x}^{(r)}\}_{r=r_n}^{r_{n+1}}$ let us consider the following problem

$$J_n^\star := \inf_{\boldsymbol{x} \in \mathbb{R}^d} J_n(\boldsymbol{x}),$$

where the function $J_n(\boldsymbol{x})$ is defined as in either (2.5) or (P1$_n$), and we assume the approximation of $J_n(\boldsymbol{x})$, $J_n^{\epsilon_1}(\boldsymbol{x})$, can be evaluated as in Section 2.4. To solve this Problem to $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})$, we have the following assumption on the convexity of f

Assumption 4 (Convexity in $\boldsymbol{x}$). *The function $f_\xi : \mathbb{R}^d \to \mathbb{R}$ $\boldsymbol{x} \mapsto f(\boldsymbol{x}, \xi)$ is convex for all $\xi \in \mathbb{R}^m$.*

Assumption 4 results in convexity of $J_n(\boldsymbol{x})$ as follows.

Lemma 3 (Convexity of $J_n(\boldsymbol{x})$). *If Assumption 4 (convexity in $\boldsymbol{x}$) holds, then for each $n \in \{1, \ldots, N\}$ the certificate $J_n(\boldsymbol{x})$ defined by* (2.5) *is convex in $\boldsymbol{x}$.*

Proof. For any $\boldsymbol{x}$, $\boldsymbol{y} \in \mathbb{R}^d$ and $t \in [0, 1] \subset \mathbb{R}$, we have $\boldsymbol{z} = t\boldsymbol{x} + (1-t)\boldsymbol{y} \in \mathbb{R}^d$ and an optimizer

of (2.5), $\mathbb{Q}_n^\star(z)$, such that

$$\begin{aligned} J_n(z) \leq & \mathbb{E}_{\mathbb{Q}_n^\star(z)}[t f(\boldsymbol{x},\xi) + (1-t) f(\boldsymbol{y},\xi)] \\ = & t\mathbb{E}_{\mathbb{Q}_n^\star(z)}[f(\boldsymbol{x},\xi)] + (1-t)\mathbb{E}_{\mathbb{Q}_n^\star(z)}[f(\boldsymbol{y},\xi)] \\ \leq & t J_n(\boldsymbol{x}) + (1-t) J_n(\boldsymbol{y}). \end{aligned}$$

□

Lemma 3 allows us to apply the Subgradient Algorithm [13, 112, 123] to obtain $\boldsymbol{x}_n^{\epsilon_2}$ via $\{\boldsymbol{x}^{(r)}\}_{r=r_n}^{r_{n+1}}$ and the following lemma.

Lemma 4 (Easy estimate of the ϵ-subgradients of $J_n(\boldsymbol{x})$). *Let the tolerance ϵ_1 and time period n be given. For any decision $\boldsymbol{x}^{(r)}$, we denote an ϵ_1-optimal solution and ϵ_1-worst-case distribution of* (P1$_n$) *by $\boldsymbol{y}^{\epsilon_1}$ and $\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})$, respectively. Let us consider the function $g_n^r : \mathbb{R}^d \to \mathbb{R}^d$, defined as*

$$g_n^r(\boldsymbol{x}) := \frac{d}{d\boldsymbol{x}} \mathbb{E}_{\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})}[f(\boldsymbol{x},\xi)].$$

Denote an ϵ-subdifferential of $J_n(\boldsymbol{x})$ at $\boldsymbol{x}$, by $\partial_\epsilon J_n(\boldsymbol{x})$. Then, for all $\epsilon \geq \epsilon_1$ we have the following

$$g_n^r(\boldsymbol{x}^{(r)}) \in \partial_\epsilon J_n(\boldsymbol{x}^{(r)}),$$

or equivalently, for every $z \in \mathrm{dom}(J_n)$ and $\epsilon \geq \epsilon_1$, we have

$$J_n(z) \geq J_n(\boldsymbol{x}^{(r)}) + g_n^r(\boldsymbol{x}^{(r)})^\top (z - \boldsymbol{x}^{(r)}) - \epsilon.$$

Moreover, for any $\tilde{\boldsymbol{x}} \in \mathbb{R}^d$, there exists $\eta > 0$ such that for all $\epsilon \geq \eta$ the following relation holds

$$g_n^r(\tilde{\boldsymbol{x}}) \in \partial_\epsilon J_n(\tilde{\boldsymbol{x}}).$$

Proof. Let us consider the function $\mathbb{E}_{\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})}[f(\boldsymbol{x},\xi)]$. Using Assumption 4 on convexity of f in $\boldsymbol{x}$, we have for any $z \in \mathrm{dom}(J_n)$ the following relation

$$\mathbb{E}_{\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})}[f(z,\xi)] \geq J_n^{\epsilon_1}(\boldsymbol{x}^{(r)}) + g_n^r(\boldsymbol{x}^{(r)})^\top(z - \boldsymbol{x}^{(r)}).$$

Knowing that $J_n^{\epsilon_1}(\boldsymbol{x}^{(r)}) \geq J_n(\boldsymbol{x}^{(r)}) - \epsilon_1$ and $J_n(z) \geq \mathbb{E}_{\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})}[f(z,\xi)]$, this concludes the first part of the proof.

To show the second part, similarly, we have for any $\tilde{\boldsymbol{x}}$, $z \in \mathrm{dom}(J_n)$ the following relation

$$\mathbb{E}_{\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})}[f(z,\xi)] \geq \mathbb{E}_{\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})}[f(\tilde{\boldsymbol{x}},\xi)] + g_n^r(\tilde{\boldsymbol{x}})^\top(z - \tilde{\boldsymbol{x}}).$$

Using Point Search Algorithm, we achieve an $\eta > 0$ such that $\mathbb{E}_{\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})}[f(\tilde{\boldsymbol{x}},\xi)] \geq J_n(\tilde{\boldsymbol{x}}) - \eta$. Finally, by similar statement as in the first part, we claim $g_n^r(\tilde{\boldsymbol{x}}) \in \partial_\epsilon J_n(\tilde{\boldsymbol{x}})$. □

Note how Lemma 4 employs the discrete distribution $\mathbb{Q}_n^{\epsilon_1}$ generated from the C-Gen Algorithm in the computation of an ϵ-subgradient function of J_n. Thus, the Subgradient Algorithm can be employed to reach an ϵ_1-proper data-driven decision with a lower certificate.

To do this, we make use of the scaled ϵ-subgradient direction for the update of decisions $\{\boldsymbol{x}^{(r)}\}_{r=r_n}^{r_{n+1}}$, as follows

$$\boldsymbol{x}^{(r+1)} = \boldsymbol{x}^{(r)} - \alpha^{(r)} \frac{g_n^r(\boldsymbol{x}^{(r)})}{\max\{\|g_n^r(\boldsymbol{x}^{(r)})\|\,,\ 1\}}, \tag{2.7}$$

where the nonnegative step size rule $\{\alpha^{(r)}\}_r$ is determined in advance. Later in the next subsection we will see how the choice of a step size rule affects the convergence of the Subgradient Algorithm to an $\boldsymbol{x}_n^{\epsilon_2}$.

The Subgradient Algorithm requires access of $\{g_n^r\}_{r=r_n}^{r_{n+1}}$, which are obtained from C-Gen Algorithm. To reduce the number of computations, we estimate the candidate subgradient functions $\{g_n^r\}_{r=r_n}^{r_{n+1}}$ as follows. Let $\epsilon_{\mathrm{SA}} \geq \epsilon_1$ be a specified tolerance. At some iteration $r \geq r_n$, assume that an ϵ_1-optimizer $\boldsymbol{y}^{\epsilon_1}$ and ϵ_1-worst-case distribution $\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})$ are obtained from

the C-Gen Algorithm. Using $\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})$, we calculate the function g_n^r at $\boldsymbol{x}^{(r)}$ and perform the subgradient iteration (2.7). At iteration $r+1$ with $\boldsymbol{x}^{(r+1)}$, we firstly check for the suboptimality of Problem ($\mathrm{P1}_n^{(r+1)}$) using the initial candidate optimizer $\boldsymbol{y}^{(0)} := \boldsymbol{y}^{\epsilon_1}$ in the Point Search Algorithm. If the optimality gap $\eta^{(1)}$ is less than ϵ_{SA}, we estimate the candidate subgradient function g_n^{r+1} using g_n^r and proceed with the subgradient iteration. Otherwise, we obtain g_n^{r+1} from the C-Gen Algorithm, which is again an ϵ_1-subgradient function at $\boldsymbol{x}^{(r+1)}$. Thus, we construct a sequence of ϵ_{SA}-subgradient functions $\{g_n^r\}_{r=r_n}^{r_{n+1}}$ that achieve an $\boldsymbol{x}_n^{\epsilon_2}$ efficiently.

Remark 5 (Effect of tolerance ϵ_{SA}). The tolerance ϵ_{SA} quantifies whether the function g_n^r generated by the current worst-case distribution can also provide a good estimate of the ϵ-subgradient at the next iteration point. If the function g_n^r is an ϵ-subgradient for ϵ small enough, there is no need of employing the C-Gen Algorithm to obtain a new subgradient function, which will be again an ϵ_1-subgradient. In practice, we suggest to choose $\epsilon_{\mathrm{SA}} \gg \epsilon_1$ as it reduces the number of computations from the C-Gen Algorithm.

2.5.1 Convergence Analysis

The following lemma follows from the convergence of the Subgradient Algorithm applied to our problem scenario.

Lemma 5 (Convergence of ϵ_{SA}-subgradient algorithm). *For each time period n with an initial data-driven decision $\boldsymbol{x}^{(r_n)}$, assume that the subgradients defined in Lemma 4 are uniformly bounded, i.e., there exists a constant $L > 0$ such that $\|g_n^r\| \leq L$ for all $r \geq r_n$.*

Given a predefined $\epsilon_2 > 0$, let the certificate tolerance ϵ_1 and the subgradient tolerance ϵ_{SA} be such that $0 < \epsilon_1 \leq \epsilon_{\mathrm{SA}} < \epsilon_2/\mu$ with $\mu := \max\{L, 1\}$. Let $\boldsymbol{x}_n^\star \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} J_n(\boldsymbol{x})$. Then, there exists a large enough number $\bar{r}$, depending on ϵ_{SA} and the step size rule $\{\alpha^{(r)}\}_r$, such that the above designed Subgradient Algorithm in (2.7) *has the following performance bounds*

$$\min_{k\in\{r_n,\ldots,r+r_n\}} \{J_n(\boldsymbol{x}^{(k)})\} - J_n(\boldsymbol{x}_n^\star) \leq \epsilon_2, \quad \forall\, r \geq \bar{r},$$

and terminates at the iteration $r_{n+1} := \bar{r} + r_n$ *with an* ϵ_2*-optimal decision by choosing* $\boldsymbol{x}_n^{\epsilon_2} \in \operatorname{argmin}_{k \in \{r_n, \ldots, \bar{r}+r_n\}} \{J_n(\boldsymbol{x}^{(k)})\}$. *In particular, there exists a large enough parameter* M *such that we can select 1) a constant step-size rule given by*

$$\alpha^{(i)} := \frac{M}{\sqrt{\bar{r}+1}}, \ \forall i \in \{r_n, \ldots, \bar{r}+r_n\},$$

where $\bar{r} := M^2 \left(\frac{\epsilon_2}{\mu} - \epsilon_{\mathrm{SA}}\right)^{-2}$*; or 2) a divergent, but square-summable, step-size rule given by*

$$\alpha^{(i)} := \frac{M}{i - r_n + 1}, \ \forall i \in \{r_n, \ldots, \bar{r}+r_n\},$$

where $\bar{r} = \min\{r \in \mathbb{N} \mid M(3 - \frac{1}{r+1}) \leq 2(\frac{\epsilon_2}{\mu} - \epsilon_{\mathrm{SA}}) \ln(r+1)\}$.

Proof. In the n^{th} time period, let us consider subgradient iterates i for all $r_n \leq i \leq r$

$$\begin{aligned}
\|\boldsymbol{x}^{(i+1)} - \boldsymbol{x}_n^\star\|^2 &= \|\boldsymbol{x}^{(i)} - \boldsymbol{x}_n^\star - \alpha^{(i)} \frac{g_n^i(\boldsymbol{x}^{(i)})}{\max\{\|g_n^i(\boldsymbol{x}^{(i)})\|\,,\ 1\}}\|^2 \\
&= \|\boldsymbol{x}^{(i)} - \boldsymbol{x}_n^\star\|^2 + (\alpha^{(i)})^2 \min\{\|g_n^i(\boldsymbol{x}^{(i)})\|^2, 1\} - 2\alpha^{(i)} \frac{g_n^i(\boldsymbol{x}^{(i)})^\top (\boldsymbol{x}^{(i)} - \boldsymbol{x}_n^\star)}{\max\{\|g_n^i(\boldsymbol{x}^{(i)})\|\,,\ 1\}}.
\end{aligned}$$

From Lemma 4, we know that $J_n(\boldsymbol{x}_n^\star) \geq J_n(\boldsymbol{x}^{(i)}) + g_n^i(\boldsymbol{x}^{(i)})^\top (\boldsymbol{x}_n^\star - \boldsymbol{x}^{(i)}) - \epsilon_{\mathrm{SA}}$ for all $\boldsymbol{x}^{(i)}$. Then,

$$\|\boldsymbol{x}^{(i+1)} - \boldsymbol{x}_n^\star\|^2 \leq (\alpha^{(i)})^2 \min\{\|g_n^i(\boldsymbol{x}^{(i)})\|^2, 1\} + \|\boldsymbol{x}^{(i)} - \boldsymbol{x}_n^\star\|^2 + \frac{2\alpha^{(i)}(J_n(\boldsymbol{x}_n^\star) - J_n(\boldsymbol{x}^{(i)}) + \epsilon_{\mathrm{SA}})}{\max\{\|g_n^i(\boldsymbol{x}^{(i)})\|\,,\ 1\}}.$$

Combining the inequalities over iterations from r_n to r gives

$$\begin{aligned}
0 &\leq \|\boldsymbol{x}^{(r_n)} - \boldsymbol{x}_n^\star\|^2 + \sum_{i=r_n}^{r} (\alpha^{(i)})^2 \min\{\|g_n^r(\boldsymbol{x}^{(i)})\|^2\,,\ 1\} + \sum_{i=r_n}^{r} \frac{2\alpha^{(i)}(J_n(\boldsymbol{x}_n^\star) - J_n(\boldsymbol{x}^{(i)}) + \epsilon_{\mathrm{SA}})}{\max\{\|g_n^r(\boldsymbol{x}^{(i)})\|\,,\ 1\}} \\
&\leq \|\boldsymbol{x}^{(r_n)} - \boldsymbol{x}_n^\star\|^2 + 2\epsilon_{\mathrm{SA}} \sum_{i=r_n}^{r} \alpha^{(i)} + \sum_{i=r_n}^{r} (\alpha^{(i)})^2 + \sum_{i=r_n}^{r} \frac{2\alpha^{(i)}(J_n(\boldsymbol{x}_n^\star) - J_n(\boldsymbol{x}^{(i)}))}{\max\{\|g_n^r(\boldsymbol{x}^{(i)})\|\,,\ 1\}}.
\end{aligned}$$

Then, using the fact that

$$\sum_{i=r_n}^{r} \frac{2\alpha^{(i)}(J_n(\boldsymbol{x}_n^\star) - J_n(\boldsymbol{x}^{(i)}))}{\max\{\|g_n^r(\boldsymbol{x}^{(i)})\|\,,\,1\}} \leq \sum_{i=r_n}^{r} \frac{-2\alpha^{(i)} \min\limits_{k\in\{r_n,\ldots,r\}}\{J_n(\boldsymbol{x}^{(k)}) - J_n(\boldsymbol{x}_n^\star)\}}{\max\{\|g_n^r(\boldsymbol{x}^{(i)})\|\,,\,1\}}$$
$$\leq -2(\sum_{i=r_n}^{r}\alpha^{(i)})\frac{\min\limits_{k\in\{r_n,\ldots,r\}}\{J_n(\boldsymbol{x}^{(k)})\} - J_n(\boldsymbol{x}_n^\star)}{\mu},$$

and the previous iteration, we have

$$\min_{k\in\{r_n,\ldots,r\}}\{J_n(\boldsymbol{x}^{(k)})\} - J_n(\boldsymbol{x}_n^\star) \leq \frac{\mu\|\boldsymbol{x}^{(r_n)} - \boldsymbol{x}_n^\star\|^2 + \mu\sum_{i=r_n}^{r}(\alpha^{(i)})^2}{2(\sum_{i=r_n}^{r}\alpha^{(i)})} + \mu\epsilon_{\mathrm{SA}}.$$

Next, it remains to select a step-size rule $\{\alpha^{(i)}\}_{i=r_n}^{r}$ such that 1) the above right hand side term is upper bounded by ϵ_2, and 2) the number of subgradient iterations $\bar{r}$ as described in the lemma is bounded. Note that the selection procedure is not unique, so we propose two step-size rules to obtain an explicit expression of $\bar{r}$.

For any data set Ξ_n, let us select a sufficiently large value M to be the diameter of the decision domain of interest, i.e.,

$$\|\boldsymbol{x}^{(r_n)} - \boldsymbol{x}_n^\star\| \leq M,\ \forall n \in \{1,\ldots,N\}.$$

Then the step size rule and $\bar{r}$ has to satisfy the following

$$\frac{M^2 + \sum_{i=r_n}^{\bar{r}+r_n}(\alpha^{(i)})^2}{2(\sum_{i=r_n}^{\bar{r}+r_n}\alpha^{(i)})} + \epsilon_{\mathrm{SA}} < \frac{\epsilon_2}{\mu}. \tag{2.8}$$

We first consider a constant step-size rule, and select the step size as follows

$$\alpha^{(i)} := \frac{M}{\sqrt{\bar{r}+1}},\ \forall i \in \{r_n,\ldots,\bar{r}+r_n\}.$$

Then, to satisfy (2.8), we determine

$$\bar{r} := M^2 \left(\frac{\epsilon_2}{\mu} - \epsilon_{\text{SA}} \right)^{-2}.$$

Alternatively, consider the divergent but square-summable step-size rule, i.e. $\sum_{i=r_n}^{\infty} \alpha^{(i)} = \infty$, $\sum_{i=r_n}^{\infty} (\alpha^{(i)})^2 < \infty$. For this class of step-size rules, as $\bar{r}$ increases to ∞, we have the left-hand side term of (2.8) goes to $\mu\epsilon_{\text{SA}} < \epsilon_2$, then there exists a large enough but finite number $\bar{r}$, such that (2.8) holds. To see this explicitly, we select the step-size rule to be harmonic sequences as follows

$$\alpha^{(i)} := \frac{M}{i - r_n + 1}, \ \forall i \in \{r_n, \ldots, \bar{r} + r_n\}.$$

Now we upper bound the numerator and lower bound the denominator of (2.8) using the fact

$$\sum_{i=1}^{\bar{r}+1} \frac{1}{i^2} \leq 2 - \frac{1}{\bar{r}+1}, \quad \sum_{i=1}^{\bar{r}+1} \frac{1}{i} \geq \ln(\bar{r}+1).$$

Then we determine $\bar{r}$ to be the following

$$\bar{r} = \min\{r \in \mathbb{N} \mid M(3 - \frac{1}{r+1}) \leq 2(\frac{\epsilon_2}{\mu} - \epsilon_{\text{SA}}) \ln(r+1)\}.$$

This concludes the proof. □

In other words, Lemma 5 specifies that there is a finite, large enough iteration step at which the ϵ_{SA}-Subgradient Algorithm terminates using the estimated ϵ_{SA}-subgradient functions.

To quantify the effect of the subgradient estimation on the convergence rate under Ξ_n, we have the following theorem.

Theorem 4 (Worst-case computational bound for an $x_n^{\epsilon_2}$). *For each time period n with an initial $\boldsymbol{x}^{(r_n)}$, let us consider the algorithm setting as in Lemma 5. Then, there exist parameters*

$\kappa \in (0,1)$, *and* $t > \epsilon_1$ *such that the computational steps* $\varphi(n,\bar{r})$ *to reach* $\boldsymbol{x}_n^{\epsilon_2}$ *are bounded by*

$$\varphi(n,\bar{r}) \leq \phi(n) + \bar{r}\left(\log_\kappa(\frac{\epsilon_1}{t}) + 1\right),$$

where $\bar{r}$ *are the subgradient steps of Lemma 5. The value* $\phi(n)$ *is the worst-case computational bound as in Theorem 3 and one should use* $\bar{\phi}(1)$ *in the bound in place of* $\phi(n)$ *if considering a data-streaming scenario.*

Proof. The computational bound to achieve an $\boldsymbol{x}_n^{\epsilon_2}$ strongly depends on the subgradient iterations $\bar{r} := r_{n+1} - r_n$ in Lemma 5 and the number of subgradient functions $\{g_n^r\}_{r=r_n}^{r_{n+1}}$ constructed via the C-Gen Algorithm. To characterize this bound, we quantify computational steps for $\{g_n^r\}_{r=r_n}^{r_{n+1}}$ next.

For each time period n, let us assume the C-Gen Algorithm has explored the feasible set of (P2$_n$) when obtaining the initial certificate $J_n^{\epsilon_1}(\boldsymbol{x}^{(r_n)})$. This procedure consumes a worst-case computational time $\phi(n)$, (or $\bar{\phi}(1)$ if a data-streaming scenario), as stated in Theorem 3. After this initial step, every time the Subgradient Algorithm needs to execute C-Gen Algorithm at some $r \leq r_{n+1}$, C-Gen Algorithm will solve a unique (CP$_n^{(l)}$) and return $\mathbb{Q}_n^{\epsilon_1}$ for an ϵ_1-subgradient function g_n^r at $\boldsymbol{x}^{(r)}$. Let CP$_r$ denote the unique (CP$_n^{(l)}$) solved at $\boldsymbol{x}^{(r)}$. Then, to quantify the computational steps for $\{g_n^r\}_{r=r_n}^{r_{n+1}}$, we compute the sum of the steps to solve $\{\text{CP}_r\}_r$.

Let us denote the number of steps solving CP$_r$ by i_r, for all $r \in \{r_n, \ldots, r_{n+1}\}$. Then, we aim to quantify i_{r+1} for g_n^{r+1}. To achieve this, let us assume a subgradient function g_n^r is computed at an iteration r. Then we perform a subgradient iteration (2.7) and obtain an $\boldsymbol{x}^{(r+1)}$. By using a subgradient estimation strategy, we obtain the optimality gap $\eta^{(1)}$ via Point Search Algorithm, denoted by $\bar{\eta}_{r+1} := \eta^{(1)}$. This gap $\bar{\eta}_{r+1}$ enables us to quantify the distance between the initial objective value and the optimal objective value of CP$_{r+1}$. When $\bar{\eta}_{r+1} \leq \epsilon_{\text{SA}}$, the algorithm uses the estimated subgradient function and $i_{r+1} = 0$. Otherwise, the computational steps can be calculated via convergence of AFWA for CP$_{r+1}$, by $\kappa^{i_{r+1}}\bar{\eta}_{r+1} \leq \epsilon_1$, where κ, or using $\bar{\kappa}$ for the

data-streaming case, is determined as in Theorem 3. Let us consider a threshold value t_r

$$t_r := \begin{cases} \epsilon_1, & \text{if } \bar{\eta}_r \leq \epsilon_{\text{SA}}, \\ \bar{\eta}_r, & \text{o.w.} \end{cases}$$

Then we can represent each value i_r by $i_r = \log_\kappa(\frac{\epsilon_1}{t_r})$, $r \in \{r_n, \ldots, r_{n+1}\}$. Let us denote $t := \max_r\{t_r\}$. Then, the computational steps for $\{g_n^r\}_{r=r_n}^{r_{n+1}}$, $\sum_r i_r$, are bounded by $\sum_r i_r \leq \bar{r}\log_\kappa(\frac{\epsilon_1}{t})$. Finally, the computational steps to achieve an $\boldsymbol{x}_n^{\epsilon_2}$, denoted by $\varphi(n, \bar{r}) := \phi(n) + \sum_r i_r + \bar{r}$, are bounded as $\varphi(n, \bar{r}) \leq \phi(n) + \bar{r}\left(\log_\kappa(\frac{\epsilon_1}{t}) + 1\right)$. Again, one should use $\bar{\phi}(1)$ in the bound in place of $\phi(n)$ if considering a data-streaming scenario. □

Theorem 4 integrates together the obtained bounds for the Subgradient Algorithm as well as the C-Gen Algorithm. As the result, a worst-case computational bound for the OnDA Algorithm, executed on the decision-update rate (the second time scale in (r)), is related to the data-streaming rate. Whenever the data-streaming rate is greater than the worst-case bound, the OnDA Algorithm provides an $\boldsymbol{x}_n^{\epsilon_2}$ decision together with its estimated certificate $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})$, $\forall n$.

From the Subgradient Algorithm, we provide, for each Ξ_n, a sequence $\{\boldsymbol{x}^{(r)}\}_{r=r_n}^{r_{n+1}}$ that approaches an $\boldsymbol{x}_n^{\epsilon_2}$. If the new data set Ξ_{n+1} is received before reaching $\boldsymbol{x}_n^{\epsilon_2}$, we initialize the next sub-sequence obtained by applying the Subgradient Algorithm, using the best decision at current iteration r, i.e., $\boldsymbol{x}^{(r_{n+1})} := \boldsymbol{x}_n^{\text{best}} \in \text{argmin}_{k\in\{r_n,\ldots,r\}}\{J_n^{\epsilon_1}(\boldsymbol{x}^{(k)})\}$. Then by connecting these sequences over n, our goal is achieved.

2.6 Data Assimilation via OnDA Algorithm

This section summarizes and analyzes our Online Data Assimilation Algorithm (OnDA Algorithm) for online data sets $\{\Xi_n\}_{n=1}^N$. Specifically, we present the algorithm procedure, its transient behavior and the convergence result.

The OnDA Algorithm starts from some random initial decision $\boldsymbol{x}^{(r)} \in \mathbb{R}^d$ and a data set Ξ_n,

OnDA Algorithm 3

Require: Goes to Step 3 upon data arrival, i.e. $\Xi_n \leftarrow \Xi_{n+1}$.

Set ϵ_1, ϵ_2, ϵ_{SA}, Ξ_1, $\boldsymbol{x}^{(0)} \in \mathbb{R}^d$, $\boldsymbol{y}^{(0)} = \mathbf{0}_m$ and $I_1^{(0)} = \varnothing$;
$n \leftarrow 1$, $r \leftarrow 1$;
$r_n \leftarrow r$;
$(J_n^{\epsilon_1}(\boldsymbol{x}^{(r)}), \boldsymbol{y}^{\epsilon_1}, \mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{(r)})) \leftarrow$ C-Gen Algorithm;
repeat
    $\boldsymbol{x}^{(r+1)} \leftarrow (\boldsymbol{x}^{(r)}, g_n^r)$ as in (2.7), $r \leftarrow r + 1$;
    $\eta \leftarrow$ Point Search Algorithm;
    if $\eta > \epsilon_{\text{SA}}$, **then**
        Goes to Step 4;
    else
        Update $g_n^r \leftarrow g_n^{r-1}$;
        if $J_n^{\epsilon_1}(\boldsymbol{x}^{(r)}) < J_n^{\epsilon_1}(\boldsymbol{x}_n^{\text{best}})$, **then**
            Update and post $(\boldsymbol{x}_n^{\text{best}}, J_n^{\epsilon_1}(\boldsymbol{x}_n^{\text{best}}))$;
until $\|\boldsymbol{x}^{(r)} - \boldsymbol{x}^{(r-1)}\| < \epsilon_2$;
$r_{n+1} \leftarrow r$;
Post $\boldsymbol{x}_n^{\epsilon_2} := \boldsymbol{x}_n^{\text{best}}$, $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) := J_n^{\epsilon_1}(\boldsymbol{x}_n^{\text{best}})$;
Wait for Ξ_{n+1}, or Termination if $n = n_0$.

with $r = 1$ and $n = 1$. Then, it first generates the certificate $J_n^{\epsilon_1}(\boldsymbol{x}^{(r)})$ via C-Gen Algorithm, then it executes the Subgradient Algorithm to obtain the decisions $\{\boldsymbol{x}^{(r+1)}, \boldsymbol{x}^{(r+2)}, \ldots\}$ with lower and lower certificates $\{J_n^{\epsilon_1}(\boldsymbol{x}^{(r+1)}), J_n^{\epsilon_1}(\boldsymbol{x}^{(r+2)}), \ldots\}$. This algorithm has the anytime property, meaning that the performance guarantee is provided anytime, as soon as the first ϵ_1-proper data-driven decision with certificate $J_n^{\epsilon_1}(\boldsymbol{x}^{(r)})$ is found. If no new data set Ξ_{n+1} comes in, the algorithm terminates as soon as the Subgradient Algorithm terminates at iteration r_{n+1}. Otherwise, the algorithm resets the C-Gen Algorithm and the Subgradient Algorithm to update the decision using more data. This achieves lower certificates with higher confidence until we obtain the lowest possible certificate and guarantee the performance almost surely. The details of the whole algorithm procedure are summarized in the table of OnDA Algorithm.

The transient behavior of the OnDA Algorithm is affected by the data-streaming rate and the rate of convergence of the intermediate algorithms (decision-update rate and certificate-update rate). To further describe these effects in each time period n, we say that the data-streaming rate is *slow with respect to the decision-update rate*, if we can find an $\boldsymbol{x}_n^{\epsilon_2}$ via the OnDA Algorithm,

where the worst-case scenario is described in Theorem 4. Further, we call it *slow with respect to the certificate-update rate*, if we can find at least one certificate during this time period, where the worst-case scenario is described in Theorem 3. When the data-streaming rate is slow w.r.t. the decision-update rate for all time periods, the OnDA Algorithm guarantees to find $\{\boldsymbol{x}_n^{\epsilon_2}\}_{n=1}^N$. When the data-streaming rate is slow w.r.t. the decision-update rate for at least one time period n_0, it guarantees to find an $\boldsymbol{x}_{n_0}^{\epsilon_2}$. When the data-streaming rate is slow w.r.t. the certificate-update rate for at least one time period n_0, the OnDA Algorithm guarantees to find a $J_{n_0}^{\epsilon_1}$ for an $\boldsymbol{x}^{(r)}$ and $r \geq r_{n_0}$. When the data-streaming rate is *not slow w.r.t. the certificate-update rate* for any time period, the OnDA Algorithm will hold on the newly streamed data set, to make the data-streaming rate *slow w.r.t. the decision-update rate* and achieve a better data-driven decision efficiently.

Next, we state the convergence result of the OnDA Algorithm when the data streams are slow w.r.t. the decision-update rate for all time periods.

Theorem 5 (Finite convergence of the OnDA Algorithm). *Consider tolerances ϵ_1, $\epsilon_2 > 0$ and streaming data sets $\{\Xi_n\}_{n=1}^N$ with $N < \infty$ for a decision making problem* (P)*. Assume that the data streams are slow w.r.t. the to decision-update rate for all n, i.e., assume the length of each time period n is no shorter than $\varphi(1, \bar{r})$, where φ and $\bar{r}$ are described as in Theorem 4 and Lemma 5, respectively. Then, the OnDA Algorithm guarantees to find a sequence of ϵ_2-optimal ϵ_1-proper data-driven decisions $\{\boldsymbol{x}_n^{\epsilon_2}\}_{n=1}^N$ associated with the sequence of the certificates $\{J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})\}_{n=1}^N$ so that the performance guarantee* (2.2) *holds for all n. Furthermore, the values of these certificates are guaranteed to be low in high probability. That is, for each n*

$$\mathbf{P}^n(J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) \leq J^\star + \epsilon_1 + \epsilon_2 + 2\hat{L}\epsilon(\beta_n)) \geq 1 - \beta_n \tag{2.9}$$

holds, where $J^\star := \inf_{\boldsymbol{x}\in\mathbb{R}^d} \mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}, \xi)]$ is the optimal objective value for the original problem (P)*, the parameter $\hat{L}$ depends on steepness of the function f, and the parameter $\epsilon(\beta_n)$ is determined as in Lemma 2.5.*

In addition, given any tolerance ϵ_3, data stream that is slow w.r.t. the decision-update

rate for all $n \in \{1, \ldots, N\}$ *with* $N \to \infty$*, and* $\sum_{n=1}^{\infty} \beta_n < \infty$*, there exists a large enough number* $n_0(\epsilon_3) > 0$*, such that the algorithm terminates in finite time with a guaranteed* ϵ_2*-optimal and* ϵ_1*-proper data-driven decision* $\boldsymbol{x}_{n_0}^{\epsilon_2}$ *and a certificate* $J_{n_0}^{\epsilon_1}(\boldsymbol{x}_{n_0}^{\epsilon_2})$ *such that the performance guarantee holds almost surely. That is,*

$$\mathbf{P}^{n_0}(\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}_{n_0}^{\epsilon_2}, \xi)] \leq J_{n_0}^{\epsilon_1}(\boldsymbol{x}_{n_0}^{\epsilon_2}) + \epsilon_1) = 1, \tag{2.10}$$

and meanwhile the quality of the designed certificate $J_{n_0}^{\epsilon_1}(\boldsymbol{x}_{n_0}^{\epsilon_2})$ *is guaranteed. In other words, for all the rest of the data sets* $\{\Xi_n\}_{n=n_0}^{\infty}$*, any element in the desired certificate sequence* $\{J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})\}_{n=n_0}^{\infty}$ *satisfies*

$$\sup_{n \geq n_0} J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) \leq J^{\star} + \epsilon_1 + \epsilon_2 + \epsilon_3. \tag{2.11}$$

Proof. The first part of the proof is an application of Theorem 3 and Theorem 4. For any data set Ξ_n and the initial data-driven decision $\boldsymbol{x}^{(r_n)}$, by Theorem 3 we can show $\boldsymbol{x}^{(r_n)}$ to be ϵ_1-proper, via finding $J_n^{\epsilon_1}(\boldsymbol{x}^{(r_n)})$ such that $\mathbf{P}^n(\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}^{(r_n)}, \xi)] \leq J_n^{\epsilon_1}(\boldsymbol{x}^{(r_n)}) + \epsilon_1) \geq 1 - \beta_n$. Then using Theorem 4, an ϵ_2-optimal ϵ_1-proper data-driven decision $\boldsymbol{x}_n^{\epsilon_2}$ with certificate $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})$ can be achieved. Therefore the performance guarantee (2.2) holds for $\boldsymbol{x}_n^{\epsilon_2}$, i.e., $\mathbf{P}^n(\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}_n^{\epsilon_2}, \xi)] \leq J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) + \epsilon_1) \geq 1 - \beta_n$.

In the following, we show the certificate $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})$ can be upper bounded in high probability, for each n.

First, let $\boldsymbol{x}^{\delta}$ denote the δ-optimal solution of (P), i.e., $\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}^{\delta}, \xi)] \leq J^{\star} + \delta$. By construction of the certificate in the algorithm we have $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) \leq J_n(\boldsymbol{x}_n^{\epsilon_2}) \leq J_n(\boldsymbol{x}_n^{\star}) + \epsilon_2 \leq J_n(\boldsymbol{x}^{\delta}) + \epsilon_2 \leq J_n^{\epsilon_1}(\boldsymbol{x}^{\delta}) + \epsilon_1 + \epsilon_2$ for all n, where the first inequality holds because J_n is the function that achieves the supreme of Problem (2.5) while $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})$ is the objective value for a feasible distribution $\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^{\epsilon_2})$, the second inequality holds because $\boldsymbol{x}_n^{\epsilon_2}$ is ϵ_2-optimal, the third inequality holds because $\boldsymbol{x}_n^{\star}$ is a minimizer of the certificate function J_n, the last inequality holds because the C-Gen Algorithm for certificate generation guarantees the existence of $J_n^{\epsilon_1}(\boldsymbol{x}^{\delta})$ such that

$J_n(\boldsymbol{x}^\delta) \le J_n^{\epsilon_1}(\boldsymbol{x}^\delta) + \epsilon_1$, with an distribution $\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^\delta)$ satisfying $d_W(\hat{\mathbb{P}}^n, \mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^\delta)) \le \epsilon(\beta_n)$.

Next, we exploit the connection between $J_n^{\epsilon_1}(\boldsymbol{x}^\delta)$ and $J^\star$. By Assumption 2 on the concavity of f in ξ, there exists a constant $\hat{L} > 0$ such that $f(x,\xi) \le \hat{L}(1 + \|\xi\|_1)$ holds for all $x \in \mathbb{R}^d$ and $\xi \in \mathcal{Z}$. Then by the dual representation of the Wasserstein metric from Kantorovich and Rubinstein [36, 59] we have $J_n^{\epsilon_1}(\boldsymbol{x}^\delta) := \mathbb{E}_{\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^\delta)}[f(\boldsymbol{x}^\delta,\xi)] \le \mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}^\delta,\xi)] + \hat{L} d_W(\mathbb{P}, \mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^\delta))$. In order to quantify the last term, we apply the triangle inequality, which gives us $d_W(\mathbb{P}, \mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^\delta)) \le d_W(\mathbb{P}, \hat{\mathbb{P}}^n) + d_W(\hat{\mathbb{P}}^n, \mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^\delta))$. Then by the performance guarantee we have $\mathbf{P}^n\{d_W(\mathbb{P}, \hat{\mathbb{P}}^n) \le \epsilon(\beta_n)\} \ge 1 - \beta_n$, and by the the way of constructing $\mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^\delta)$ we have $d_W(\hat{\mathbb{P}}^n, \mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^\delta)) \le \epsilon(\beta_n)$. These inequalities result in $\mathbf{P}^n\{d_W(\mathbb{P}, \mathbb{Q}_n^{\epsilon_1}(\boldsymbol{x}^\delta)) \le 2\epsilon(\beta_n)\} \ge 1 - \beta_n$. We use now this bound to deal with the last term in the upper bound of $J_n^{\epsilon_1}(\boldsymbol{x}^\delta)$. In particular, we have $\mathbf{P}^n\{J_n^{\epsilon_1}(\boldsymbol{x}^\delta) \le \mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}^\delta,\xi)] + 2\hat{L}\epsilon(\beta_n)\} \ge 1 - \beta_n$ for all n. Using the obtained inequality $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) \le J_n^{\epsilon_1}(\boldsymbol{x}^\delta) + \epsilon_1 + \epsilon_2$ and knowing δ can be arbitrary small, we achieved the goal as in (2.9).

Now, it remains to find an n_0, associated with an ϵ_2-optimal and ϵ_1-proper data-driven decision $\boldsymbol{x}_{n_0}^{\epsilon_2}$, such that the almost sure guarantee (2.10) and bound (2.11) of the certificate $J_{n_0}^{\epsilon_1}(\boldsymbol{x}_{n_0}^{\epsilon_2})$ can be guaranteed for the termination of the OnDA Algorithm as $N \to \infty$. We achieve this by two steps.

First, we show the almost sure performance guarantee when the data set is sufficiently large. For any time period n, the algorithm finds $\boldsymbol{x}_n^{\epsilon_2}$ with the performance guarantee (2.2), which can be equivalently written as $\mathbf{P}^n(\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}_n^{\epsilon_2},\xi)] \ge J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) + \epsilon_1) \le \beta_n$. As $\sum_{n=1}^{\infty} \beta_n < \infty$, from the 1st Borel-Cantelli Lemma we have that $\mathbf{P}^\infty\{\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}_n^{\epsilon_2},\xi)] \ge J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) + \epsilon_1 \text{ occurs infinitely many often}\} = 0$. That is, almost surely we have that $\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}_n^{\epsilon_2},\xi)] \ge J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) + \epsilon_1$ occurs at most for finite number of n. Thus, there exists a sufficiently large n_1, such that for all $n \ge n_1$, we have $\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}_n^{\epsilon_2},\xi)] \le J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) + \epsilon_1$ occurs almost surely, i.e., $\mathbf{P}^n(\mathbb{E}_{\mathbb{P}}[f(\boldsymbol{x}_n^{\epsilon_2},\xi)] \le J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) + \epsilon_1) = 1$ for all $n \ge n_1$. Later if we pick $n_0 \ge n_1$, then the almost sure performance guarantee holds for such $\boldsymbol{x}_{n_0}^{\epsilon_2}$ and $J_{n_0}^{\epsilon_1}(\boldsymbol{x}_{n_0}^{\epsilon_2})$.

Second, we show a tight certificate bound can be achieved almost surely. Consider performance bound (2.9). As $\epsilon(\beta_n)$ decreases and goes to 0 as $n \to \infty$, there exists n_2 such that $2\hat{L}\epsilon(\beta_n) \le \epsilon_3$ holds for all $n \ge n_2$. Therefore, we have $\mathbf{P}^n\{J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) \le J^\star + \epsilon_1 + \epsilon_2 + \epsilon_3\} \ge 1 - \beta_n$

for all $n \geq n_2$, or equivalently, $\mathbf{P}^n\{J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) \geq J^\star + \epsilon_1 + \epsilon_2 + \epsilon_3\} \leq \beta_n$. As $\sum_{n=1}^{\infty} \beta_n < \infty$, then the 1st Borel-Cantelli Lemma applies to this situation. Thus we claim that there exists a sufficiently large n_3 such that for all $n \geq \max\{n_2, n_3\}$ we have almost surely, $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) \leq J^\star + \epsilon_1 + \epsilon_2 + \epsilon_3$.

Then, by letting $n_0 := \max\{n_1, n_2, n_3\}$ we have almost sure performance guarantee (2.10) and almost surely, the bound (2.11). □

Theorem 5 quantifies the goodness of the certificates that are achievable via the OnDA Algorithm, under the condition that the data-streaming rate is slow w.r.t. the decision-update rate. Intuitively, the smaller the tolerances ϵ_i are, the lower the certificates become. Further, as $N \to \infty$, the smaller the parameter $\epsilon(\beta_N)$ is and the higher the confidence $1 - \beta_N \to 1$. When infinitely many data are streamed in, the theorem implies that we can get arbitrarily close to the optimal decision with probability one.

Remark 6 (Selection of tolerances ϵ_2, ϵ_1 and ϵ_{SA}). In practice, the tolerance ϵ_2 determines the performance bound of J_n, which governs the whole algorithm. With a given ϵ_2, tolerance ϵ_1 and ϵ_{SA} can be chosen following the rule in Lemma 5. Intuitively, ϵ_1 can be chosen to be two orders of magnitude smaller than ϵ_2, while ϵ_{SA} can be an order of magnitude smaller than ϵ_2. These tolerances can also be chosen in a data-driven fashion, to achieve asymptotic convergence, or a better transient behavior of the algorithm.

Remark 7 (Asymptotic behavior of the OnDA Algorithm). Theorem 5 claims the convergence of the OnDA Algorithm to a decision with a desired certificate using large but a finite data set. The smaller ϵ_3 is, the larger data set is needed to achieve the desired certificate. Because tolerances ϵ_1,ϵ_2 and ϵ_3 can be chosen arbitrarily small, the certificate can indeed approach to $J^\star$. However, ϵ_1,ϵ_2 may affect the transient behavior of the algorithm and the data-streaming rate. In practice, to reach $J^\star$, these tolerances can be chosen in a data-driven fashion, for example diminishing sequences.

2.7 Data Incremental Covering

In this section, we aim to handle large streaming data sets for efficient Online Data Assimilation Algorithm (OnDA Algorithm). To achieve this, we firstly propose an Incremental Covering Algorithm (I-Cover Algorithm). This algorithm leverages the pattern of the data points to obtain a new ambiguity set, denoted by $\tilde{\mathcal{P}}_n$. Then, we adapt $\tilde{\mathcal{P}}_n$ for a variant of the OnDA Algorithm. The resulting algorithm enables us to construct subproblems which have a lower dimension than those generated without it, and we verify its capability of handling large data sets in simulation.

2.7.1 The I-Cover Algorithm

Let ζ and ω denote the center and radius of the Euclidean ball $B_\omega(\zeta)$, respectively. For each data set Ξ_n and a given ω, let $C_n \subset \Xi_n$ denote the set of points such that $\Xi_n \subset \cup_{\zeta \in C_n} B_\omega(\zeta)$. Let $p := |C_n|$ denote the number of these Euclidean balls. To account for the number of data points that are covered by a specific ball, we associate each ball $B_\omega(\zeta_k)$ a weighting parameter θ_k. We denote by $Q_n := \{\theta_k\}_{k=1}^p$ the set of these parameters. Then, as data sets $\{\Xi_n\}_{n=1}^N$ are sequentially accessible, we are to incrementally cover data sets by adapting C_n and Q_n.

Formally, the I-Cover Algorithm works as follows. Let $C_0 = \varnothing$ and $Q_0 = \varnothing$. For the n^{th} time period with set Ξ_n, we initialize sets as $C_n := C_{n-1}$ and $Q_n := Q_{n-1}$. To generate a random cover for Ξ_n, we randomly and sequentially evaluate each newly streamed data point. Let $\varsigma \in \Xi_n \setminus \Xi_{n-1}$ denote the data point under consideration. If $\varsigma \notin B_\omega(\zeta_k)$ for all $\zeta_k \in C_n$, we update $C_n \leftarrow C_n \cup \{\zeta_{p+1} := \varsigma\}$, $Q_n \leftarrow Q_n \cup \{\theta_{p+1} := 1\}$ and $p \leftarrow |C_n|$. If ς is covered by some (at least one) Euclidean balls, i.e., $\varsigma \in B_\omega(\zeta_k)$ for some k with $\zeta_k \in C_n$, we only update Q_n. Let ℓ_ς denote the number of the balls that cover ς and let $I_\varsigma \subset \{1, \ldots, p\}$ denote the index set of these balls. Then we update elements of Q_n via $\theta_k \leftarrow \theta_k + \ell_\varsigma^{-1}$ for all $k \in I_\varsigma$. After all the new data points have been evaluated in this way, we achieve a cover of Ξ_n. Then, as the data set streams over time, the algorithm incrementally updates the cover and weights. By construction, we see that $|C_n| \leq n$.

Next, we use C_n and Q_n to construct a new ambiguity set that results in potentially low dimensional subproblems in the OnDA Algorithm.

2.7.2 Integration of I-Cover Algorithm

Following the I-Cover Algorithm, we consider a distribution $\tilde{\mathbb{P}}^n$ associated with Ξ_n, as follows

$$\tilde{\mathbb{P}}^n := \frac{1}{n}\sum_{k=1}^{p} \theta_k \delta_{\{\zeta_k\}}, \tag{2.12}$$

where $\delta_{\{\zeta_k\}}$ is a Dirac measure at the center of the covering ball $B_\omega(\zeta_k)$ and θ_k is the associated weight of $B_\omega(\zeta_k)$. We claim the distribution $\tilde{\mathbb{P}}^n$ is close to the empirical distribution $\hat{\mathbb{P}}^n$ under the Wasserstein metric, using the following lemma.

Lemma 6 (Distribution $\tilde{\mathbb{P}}^n$ is a good estimate of $\hat{\mathbb{P}}^n$). *Let the radius ω of the Euclidean ball be chosen. Then the distribution $\tilde{\mathbb{P}}^n$ constructed by the I-Cover Algorithm on Ξ_n is close to $\hat{\mathbb{P}}^n$ under the Wasserstein metric, i.e., $d_W(\hat{\mathbb{P}}^n, \tilde{\mathbb{P}}^n) \leq \omega$.*

Proof. The proof is an application of the dual characterization of the Wasserstein distance. Let us consider

$$\begin{aligned} d_W(\hat{\mathbb{P}}^n, \tilde{\mathbb{P}}^n) &= \sup_{f\in\mathcal{L}}\{\int_{\mathcal{Z}} f(\xi)\hat{\mathbb{P}}^n(d\xi) - \int_{\mathcal{Z}} f(\xi)\tilde{\mathbb{P}}^n(d\xi)\}, \\ &= \frac{1}{n}\sup_{f\in\mathcal{L}}\{\sum_{k=1}^{n} f(\xi_k) - \sum_{k=1}^{p}\theta_k f(\zeta_k)\}. \end{aligned}$$

By partitioning the data set Ξ_n into C_n and $\Xi_n \setminus C_n$ for each summation term, we have

$$\begin{aligned} \sum_{k=1}^{n} f(\xi_k) &= \sum_{\varsigma\in C_n} f(\varsigma) + \sum_{\varsigma\in\Xi_n\setminus C_n} f(\varsigma), \\ \sum_{k=1}^{p}\theta_k f(\zeta_k) &= \sum_{k=1}^{p} f(\zeta_k) + \sum_{\varsigma\in\Xi_n\setminus C_n} \ell_\varsigma^{-1} \sum_{k\in I_\varsigma} f(\zeta_k). \end{aligned}$$

Canceling the first summation term gives us the following

$$\begin{aligned} d_W(\hat{\mathbb{P}}^n, \tilde{\mathbb{P}}^n) &= \frac{1}{n} \sup_{f\in\mathcal{L}} \{ \sum_{\varsigma\in\Xi_n\setminus C_n} \ell_\varsigma^{-1} \sum_{k\in I_\varsigma} f(\varsigma) - f(\zeta_k)\}, \\ &\le \frac{1}{n} \sup_{f\in\mathcal{L}} \{ \sum_{\varsigma\in\Xi_n\setminus C_n} \ell_\varsigma^{-1} \sum_{k\in I_\varsigma} |f(\varsigma) - f(\zeta_k)|\}, \\ &\le \frac{1}{n} \sum_{\varsigma\in\Xi_n\setminus C_n} \ell_\varsigma^{-1} \sum_{k\in I_\varsigma} \|\varsigma - \zeta_k\|_1, \\ &\le \frac{1}{n} \sum_{\varsigma\in\Xi_n\setminus C_n} \ell_\varsigma^{-1} \sum_{k\in I_\varsigma} \omega = \frac{n-p}{n}\omega \le \omega, \end{aligned}$$

where the first inequality is derived taking component-wise absolute values; the second inequality is due to f being in the space of Lipschitz functions defined on $\mathcal{Z}$ with Lipschitz constant 1; and the third inequality is due to $\varsigma \in B_\omega(\zeta_k)$. □

Then equipped with Lemma 6 and Theorem 1 on the measure of concentration result, we can provide the certificate that ensures the performance guarantee in (2.1).

Lemma 7 (Tractable certificate generation for x with performance guarantee (2.1) **using $\tilde{\mathbb{P}}^n$).** *Given $\Xi_n := \{\xi_k\}_{k=1}^n$, $\beta_n \in (0,1)$, $\boldsymbol{x} \in \mathbb{R}^d$, and the radius ω of the covering balls. Define the new ambiguity set $\tilde{\mathcal{P}}_n := \mathbb{B}_{\tilde{\epsilon}(\beta_n)}(\tilde{\mathbb{P}}^n)$ where the center of the Wasserstein ball $\tilde{\mathbb{P}}^n$ is defined in* (2.12) *and the radius $\tilde{\epsilon}(\beta_n) := \epsilon(\beta_n) + \omega$. Then the following certificate satisfies* (2.1) *for all $\boldsymbol{x} \in \mathbb{R}^d$*

$$J_n(\boldsymbol{x}) := \sup_{\mathbb{Q}\in\tilde{\mathcal{P}}_n} \mathbb{E}_{\mathbb{Q}}[f(\boldsymbol{x}, \xi)]. \tag{2.13}$$

Further, under the same assumptions required in Theorem 2 we have the new version of ($\mathrm{P1}_n$) *as follows*

$$\begin{aligned} J_n(\boldsymbol{x}) := \sup_{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_p\in\mathbb{R}^m} & \frac{1}{n}\sum_{k=1}^p \theta_k f(\boldsymbol{x}, \zeta_k - \boldsymbol{y}_k), \\ \text{s.t.} \quad & \frac{1}{n}\sum_{k=1}^p \theta_k \|\boldsymbol{y}_k\|_1 \le \tilde{\epsilon}(\beta_n), \end{aligned} \tag{$\tilde{\mathrm{P1}}_n$}$$

and the associated worst-case distribution $\tilde{\mathbb{Q}}_n^\star(\boldsymbol{x})$ is a weighted version of $\mathbb{Q}_n^\star(\boldsymbol{x})$ in Theorem 2,

i.e.,

$$\tilde{\mathbb{Q}}_n^\star(\boldsymbol{x}) := \frac{1}{n}\sum_{k=1}^{p} \theta_k \delta_{\{\zeta_k - \boldsymbol{y}_k^\star\}},$$

where $\boldsymbol{y}^\star := (\boldsymbol{y}_1^\star, \ldots, \boldsymbol{y}_p^\star)$ *is an optimizer of* $(\tilde{\mathrm{P}1}_n)$.

Proof. From Lemma 1 we have $\mathbf{P}^n\{d_W(\mathbb{P}, \hat{\mathbb{P}}^n) \leq \epsilon(\beta_n)\} \geq 1 - \beta_n$ for each n. Then using the result from Lemma 6 we have $\mathbf{P}^n\{d_W(\mathbb{P}, \tilde{\mathbb{P}}^n) \leq d_W(\tilde{\mathbb{P}}^n, \hat{\mathbb{P}}^n) + d_W(\mathbb{P}, \hat{\mathbb{P}}^n) \leq \epsilon(\beta_n) + \omega\} \geq 1 - \beta_n$, i.e., $\mathbf{P}^n\{d_W(\mathbb{P}, \tilde{\mathbb{P}}^n) \leq \tilde{\epsilon}(\beta_n)\} \geq 1 - \beta_n$ for each n. The rest of the proof follows directly from that in Lemma 1 and Theorem 2. □

Remark 8 (New version of $(\mathrm{P}2_n)$**).** The equivalent formulation of Problem $(\tilde{\mathrm{P}1}_n)$ is a new version of $(\mathrm{P}2_n)$, defined as follows

$$\begin{aligned} J_n(\boldsymbol{x}) := \max_{\substack{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_p \in \mathbb{R}^m \\ \boldsymbol{v}_1,\ldots,\boldsymbol{v}_p \in \mathbb{R}^m}} & \frac{1}{n}\sum_{k=1}^{p} \tilde{h}_k\left(\frac{\boldsymbol{u}_k - \boldsymbol{v}_k}{\theta_k}\right), \\ \text{s.t.} \quad & (\boldsymbol{u}, \boldsymbol{v}) \in n\tilde{\epsilon}(\beta_n)\Delta_{2mp}, \end{aligned} \tag{$\tilde{\mathrm{P}2}_{n,p}$}$$

where for each $k \in \{1, \ldots, p\}$, $\zeta_k \in C_n$ and $\boldsymbol{x} \in \mathbb{R}^d$, we define $\tilde{h}_k : \mathbb{R}^m \to \mathbb{R}$ as

$$\tilde{h}_k(\boldsymbol{y}) := \theta_k f(\boldsymbol{x}, \zeta_k - \boldsymbol{y}).$$

With the constructed ambiguity set $\tilde{\mathcal{P}}_n$ and certificate function J_n, the the developed algorithms in Section 2.4 and Section 2.5 are valid to solve Problem $(\tilde{\mathrm{P}2}_{n,p})$. And the main Theorem 5 on the finite convergence of the OnDA Algorithm is valid for the certificate function J_n where the only difference is that the quality of the certificate for $\boldsymbol{x}_n^{\epsilon_2}$ in (2.11) is replaced by

$$\sup_{n \geq n_0} J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2}) \leq J^\star + \epsilon_1 + \epsilon_2 + \epsilon_3 + 2(1 - \frac{p_{n_0}}{n_0})\hat{L}\omega,$$

where n_0 is the number of the data set in Ξ_{n_0} and p_{n_0} indicates the number of Euclidean balls that cover Ξ_{n_0}.

2.8 Case Studies

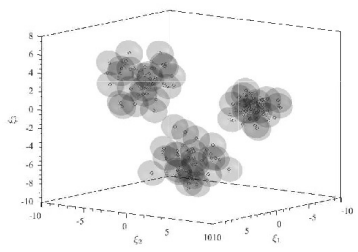
In this section, we demonstrate the application of the proposed algorithms on two case studies, with a potentially large streaming data set.

2.8.1 Study 1: The Effect of the I-Cover Algorithm

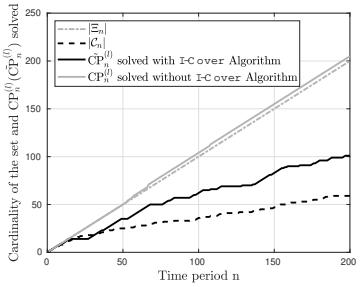
In order to visualize the effect of the I-Cover Algorithm, here we solve a toy problem in form of (P) using OnDA Algorithm, with and without the I-Cover Algorithm respectively. Let $x \in \mathbb{R}$ be the variable for Problem (P). Assume there are $N = 200$ data points $\{\xi_k\}_{k=1}^N$ streaming into the algorithm. Assume each time period is one second, and for each second k we only stream in one data point $\xi_k \in \mathbb{R}^3$, where ξ_k is a realization of the unknown distribution $\mathbb{P}$. The $\mathbb{P}$ we use for simulation is a multivariate weighted Gaussian mixture distribution with three centers, where each center has mean $\mu_1 = (2, -4, 3)$, $\mu_2 = (-3, 5, 0)$, $\mu_3 = (0, 0, -6)$, variance $\sum_1 = \text{diag}(1, 3, 2)$, $\sum_2 = 2 \cdot \boldsymbol{I}_3$, $\sum_3 = \boldsymbol{I}_3$, and weights 0.25, 0.5, 0.25, respectively. Let the cost function to be $f(x, \xi) := x^2 - \xi^\top \xi$, the confidence be $1 - \beta_n := 1 - 0.95e^{1-\sqrt{n}}$ and we use the parameter $c_1 = 2$, $c_1 = 1$ to design the radius $\epsilon(\beta_n)$ of the Wasserstein ball in (2.4). The radius of the Euclidean ball for the I-Cover Algorithm is $w = 1.5$. We sample the initial decision $x^{(0)}$ from the uniform distribution $[-10, 10]$. The tolerance for the algorithm is $\epsilon_1 = 10^{-5}$, $\epsilon_2 = 10^{-4}$.

Figure 2.3a and Figure 2.3b demonstrate the effect of the I-Cover Algorithm in the OnDA Algorithm. Specifically, Figure 2.3a shows the incremental data covering at the end of the 200^{th} time period in the (ξ_1, ξ_2, ξ_3) coordinates. The large shaded area are 59 Euclidean balls B_ω with their centers $\{\zeta_k\}_{k=1}^{59}$ denoted by some of the small circles, where all these small circles constitute the streamed data set $\Xi_{200} := \{\xi_k\}_{k=1}^{200}$. In Figure 2.3b, the gray dashed line represents the number of the data points used as centers of the empirical distribution $\hat{\mathbb{P}}^n$ over time and the black dashed line is that for distribution $\tilde{\mathbb{P}}^n$. Clearly as the data streams over time, the number $p := |C_n|$ is significantly smaller than $n := |\Xi_n|$, which results in the size of Problem ($\tilde{\text{P}}2_{n,p}$) being much smaller than that of ($\text{P}2_n$). Further, the gray solid line counts the total number of

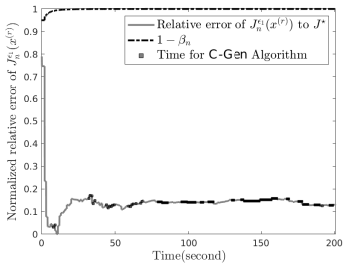
subproblems ($\mathrm{CP}_n^{(l)}$) solved to generate certificates over time and the black solid line represents that for subproblems ($\tilde{\mathrm{CP}}_n^{(l)}$) in solution to ($\tilde{\mathrm{P}}2_{n,p}$). These subproblems search the explicit solution for the ϵ_1-worst-case distribution and consume the major computing resources in the OnDA Algorithm. It can be seen that the number of ($\tilde{\mathrm{CP}}_n^{(l)}$) solved over time is on average only half of the ($\mathrm{CP}_n^{(l)}$) in each time period. Together, the dimension and total number of subproblems ($\tilde{\mathrm{CP}}_n^{(l)}$) solved with the I-Cover Algorithm is significantly smaller than that without it.



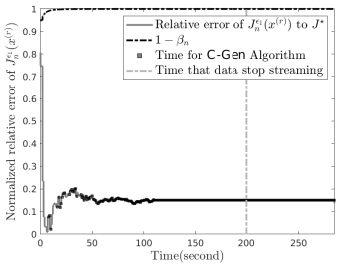
(a) Data set cover at 200th time period.



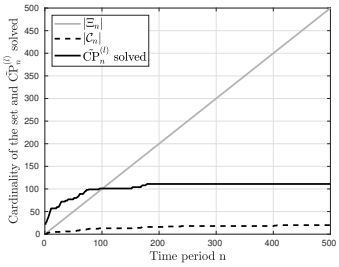
(b) Size of $\tilde{\mathbb{P}}^n$ ($\hat{\mathbb{P}}^n$) and count of $\tilde{\mathrm{CP}}_n^{(l)}$ ($\mathrm{CP}_n^{(l)}$) solved with(out) I-Cover Algorithm.



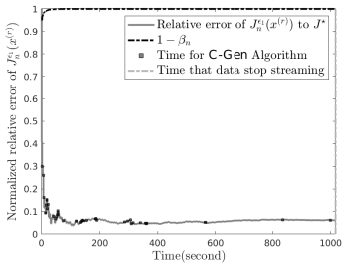
(c) Relative error of $J_n^{\epsilon_1}(\boldsymbol{x}^{(r)})$ over time, with the I-Cover Algorithm.



(d) Relative error of $J_n^{\epsilon_1}(\boldsymbol{x}^{(r)})$ over time, without the I-Cover Algorithm.



(e) Size of $\tilde{\mathbb{P}}^n$ ($\hat{\mathbb{P}}^n$) indicated by $|\Xi_n|$ ($|\mathcal{C}_n|$) and count of $\tilde{\mathrm{CP}}_n^{(l)}$ solved.



(f) Relative error of $J_n^{\epsilon_1}(\boldsymbol{x}^{(r)})$ over time, with the I-Cover Algorithm.

Figure 2.3. Simulation results of the Online Data Assimilation Algorithm, with and without the Incremental Covering Algorithm.

To evaluate the quality of the obtained ϵ_1-proper data-driven decision with the streaming data, we estimate the optimizer of (P), $\boldsymbol{x}^\star$, by minimizing the average value of the cost function f for a validation data set of $N_{\mathrm{val}} = 10^4$ data points randomly generated from the distribution $\mathbb{P}$ (in the simulation case $\mathbb{P}$ is known). We take the resulting objective value as the estimated optimal objective value for Problem (P), i.e., $J^\star := J^\star(\boldsymbol{x}^\star)$. We calculate $J^\star(\boldsymbol{x}^\star)$ using the underline distribution $\mathbb{P}$, serving as the true but unknown scale to evaluate the goodness of the certificate

obtained throughout the algorithm.

Figure 2.3c and Figure 2.3d show the evolution of the certificate sequence $\{J_n^{\epsilon_1}(\boldsymbol{x}^{(r)})\}_{n=1,r=1}^{N,\infty}$ with the I-Cover Algorithm and that without the I-Cover Algorithm, respectively. Here, the optimal decision of (P) is trivially $\boldsymbol{x}^\star = 0$, and for both algorithms the subgradient counterpart of the OnDA Algorithm returns the optimal decision after the first data point ξ_1 is used. Therefore, after a very short period within the first second, both figures start reflecting the certificate evolution under the decision sequence $\{\boldsymbol{x}^{(r)} \approx 0\}_{r=r_2}^{\infty}$. The gray solid line in both Figure 2.3c and Figure 2.3d show the relative goodness of the certificates for the currently used ϵ_1-proper data-driven decision $\boldsymbol{x}^{(r)} \approx 0$ calibrated by the estimated optimal value $J^\star$ over time. The black segments on the gray solid line indicate that the C-Gen Algorithm is executing for certificates update, while at these time intervals the old certificate $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})$, associated with the ϵ_2-optimal and ϵ_1-proper data-driven decision $\boldsymbol{x}_n^{\epsilon_2}$, is still valid to guarantee the performance under the old confidence $1-\beta_n$. This situation commonly happens when a new data set Ξ_{n+1} is streamed in and a new certificate $J_{n+1}^{\epsilon_1}(\boldsymbol{x}^{(r)})$ is yet to be obtained. It can be seen that after a few samples streamed, both the obtained certificate becomes close (within 10%) to the estimated true optimal value $J^\star$. In Figure 2.3d however, as the data streams over 50 seconds, the computing cost for updating certificates becomes significant for the algorithm without the I-Cover Algorithm. After 100^{th} data point has been assimilated, the certificate $J_n^{\epsilon_1}(\boldsymbol{x}^{(r)})$ stops updating for all $n \geq 100$. And, further, after all the data points streamed (in 200 seconds), the algorithm took about 70 seconds to terminate the algorithm with certificate $J_{200}^{\epsilon_1}(\boldsymbol{x}^{(r)})$. This is a clear disadvantage compared to the algorithm with the I-Cover Algorithm, which terminates as soon as all the data points were taken in.

2.8.2 Study 2: OnDA Algorithm with Large Streaming Data Sets

Here, we are to find an ϵ_2-optimal, ϵ_1-proper decision $\boldsymbol{x} \in \mathbb{R}^{30}$ for Problem (P). We consider $N = 500$ iid sample points $\{\xi_k\}_{k=1}^N$ streaming randomly in between every 1 to 3 seconds with each data point $\xi_k \in \mathbb{R}^{10}$ a realization of $\mathbb{P}$. We assume that the unknown distribution is a

multivariate Gaussian mixture distribution with three centers where the components of the mean of each center are uniformly chosen between $[-10, 10]$, and the variance matrix is $\boldsymbol{I}_m$ for each center. We assume the cost function $f : \mathbb{R}^{30} \times \mathbb{R}^{10} \to \mathbb{R}$ to be $f(\boldsymbol{x}, \xi) := \boldsymbol{x}^\top A\boldsymbol{x} + \boldsymbol{x}^\top B\xi + \xi^\top C\xi$ with random values for the positive semi-definite matrix $A \in \mathbb{R}^{30\times30}$, $B \in \mathbb{R}^{30\times10}$ and negative definite matrix $C \in \mathbb{R}^{10\times10}$. The radius of the Euclidean ball for the I-Cover Algorithm is $w = 5$.

Similarly to Figure 2.3b, Figure 2.3e demonstrates the incremental construction of the distribution $\tilde{\mathbb{P}}^n$ and the accumulated number of Problem ($\tilde{\text{P2}}_{n,p}$) solved over time. Clearly, after certain amount of data have been assimilated, the structure of the data set was inferred by the I-Cover Algorithm and the number of Euclidean balls used to cover the data set is about 20. Also, after the 100^{th} time period (from 100 to 200 seconds in this case), the algorithm can validate new certificate without solving any Problem ($\tilde{\text{P2}}_{n,p}$). This feature dramatically improves the performance of the OnDA Algorithm and makes the algorithm flexible for online settings.

Similarly to Figure 2.3c, Figure 2.3f shows the evolution of the certificate sequence $\{J_n^{\epsilon_1}(\boldsymbol{x}^{(r)})\}_{n=1,r=1}^{N,\infty}$ for the decision sequence $\{\boldsymbol{x}^{(r)}\}_{r=1}^{\infty}$. In the same way as in the last case study, the obtained certificate becomes close to the estimated true optimal value $J^\star$ (within 10%) after about 25 seconds with the assimilation of 10 data sets. Also, as more data sets are assimilated, the update of the certificate $J_n^{\epsilon_1}(\boldsymbol{x}_n^{\epsilon_2})$ remains fast and the algorithm terminates within a second after the last data set was streamed in.

Chapter 2, in full, is a reprint of *Data Assimilation and Online Optimization With Performance Guarantees*, D. Li and S. Martínez, IEEE Transactions on Automatic Control, (66)5:2115-2129, 2021. A preliminary version appeared in the proceddings of IEEE International Conference on Decision and Control, pp. 1961-1966, Miami, FL, USA, 2018, as *Online data assimilation in distributionally robust optimization*, D. Li and S. Martínez. The dissertation author was the primary investigator and author of these papers.

Chapter 3

Data-driven Predictive Control

This chapter studies a data-driven predictive control for a class of control-affine systems which is subject to uncertainty. With the accessibility to finite sample measurements of the uncertain variables, we aim to find controls which are feasible and provide superior performance guarantees with high probability. This results into the formulation of a stochastic optimization problem, which is intractable due to the unknown distribution of the uncertainty variables. By developing a distributionally robust optimization framework, we present an equivalent and yet tractable reformulation. Further, we propose an efficient algorithm that provides online suboptimal data-driven solutions and guarantees performance with high probability. To illustrate the effectiveness of the proposed approach, we consider a highway speed-limit control problem. We then develop a set of data-driven speed controls that allow us to prevent traffic congestion with high probability. Finally, we employ the resulting control method on a traffic simulator to illustrate the effectiveness of this approach numerically.

3.1 Related Works

Motivated by the need of developing finite-data-driven predictive control methods, we consider the application of a *Distributionally Robust Optimization* (DRO) framework to a class of *Model Predictive Control* (MPC) problems. Stochastic MPC is a general framework that can handle broad types of system uncertainty in a tractable manner [39, 84, 86, 95, 110]. In

general, the effectiveness of MPC depends on the particular problem of interest, roughly classified according to three criterion: 1) the class of systems to be controlled, e.g., linear [37, 108] or nonlinear systems [80,87,106,118]; 2) the way uncertainty is handled, e.g., stochastic [19,82,117] or bounded uncertainty [20]; and 3) the solution method, e.g., quadratic programming [110], stochastic programming [62, 87], or nonconvex optimization [79, 86]. In practice, system performance guarantees typically require large amounts of data processing which, for online settings such as MPC, may become specially challenging. DRO has attracted recent attention due to its finite-sample performance guarantees [36, 41], problem tractability via Wasserstein ambiguity sets [16, 45, 136], its distributed formulation [24, 25] and online implementation as in Chapter 2. These characteristics provide a novel mechanism to deal with uncertainty in MPC, while allowing for the tractability of the associated nonconvex optimization problems. In practice, the efficacy of the DRO framework depends on the explicit system structure. Here, our proposed approach applies to a problem class whose dynamics can be nonlinear, but affine both in control and states, and constraints can be linear or bi-linear.

Highway Speed-Limit Control: For illustration purposes, we apply our solution method to toy examples related to highway speed-limit control, focusing our discussion on its performance with respect to uncertainty. As the highway congestion significantly affects system operations [60], a variety of congestion mitigation strategies have been proposed, including those based on optimization [55, 56, 135], logic-based control [27], extremum seeking control [139], and autonomous-vehicle scheduling [134]. More recently, *Speed-Limit Control* has been proposed as an effective mechanism in transportation [121,137]. In particular, optimization-based speed-limit control has successfully demonstrated the containment of traffic congestion. For example, [50] proposed a discrete macroscopic second-order model, METANET, and used it in optimization problems for speed limit. However, at that time, the robustness analysis with respect to the system uncertainty was absent. Later in [47], a linear model predictive control of speed limit was developed for congestion mitigation. This chapter exploited the celebrated *Cell Transmission Model* (CTM) [33] or its extension for inhomogeneous traffic, the *Link Transmission Model*

(LTM), to capture the deterministic distribution of traffic densities along a highway, and the dynamic properties of highways are characterized for congestion reduction, with relative low online computational costs. In addition, [75] proposed a scenario-based optimization to account for the bounded uncertainty of transportation systems. However, an explicit treatment of the system uncertainty, such as unknown drivers actions, vehicle arrival and departures, and as well as random events that happen on highways, are yet to be fully explored.

With increasing accessibility of real-time traffic data [51, 133] for uncertainty reduction, congestion mitigation and traffic control under uncertainty can become practical. As a first step into developing a novel data-driven traffic control methods, we consider a problem related to highway transportation, formulated by way of an extended CTM and controlled by speed limit, and customize our proposed framework with the following question in mind: *Can we find an efficient approach for the computation of data-driven controls with guarantees on congestion elimination?* We would like to note that, while the particular problem we look at is inspired by traffic-congestion mitigation, the main emphasis of this chapter is on the solution approach.

Statement of Contributions

This chapter presents the following contributions: 1) We first provide a general framework for data-driven predictive control under uncertainty. To demonstrate our approach explicitly, we consider a problem inspired by highway speed-limit control which extends the CTM to account for random events (due to drivers actions) as well as vehicle arrival and departures. This provides an analytical framework and a stochastic optimization problem formulation for the computation of speed limit using the available flow measurements (in Section 3.2). 2) We propose an optimization-based data-driven control approach that extends DRO to account for system dynamics. The resulting approach guarantees congestion elimination with high probability, using only finite flow measurements (in Section 3.3). 3) As the proposed data-driven optimization problem is infinite dimensional and intractable, we propose an equivalent reformulation that

reduces the proposed problem into a finite-dimensional optimization problem (in Section 3.3). 4) Yet this problem is non-convex and difficult to solve, so we find an equivalent reformulation via a binary representation technique, and propose a computationally efficient algorithm to provide online sub-optimal speed limits that ensure congestion elimination and guarantee of highway throughput with high probability (in Section 3.5). 5) We propose in Section 3.6 an optimization tool to analyze the performance of the proposed algorithm offline. This tool is developed via a second-order cone relaxation technique and we show that, under mild conditions, the resulting Mixed-Integer Second-Order Cone Problem (MISOCP) is exact, and can be handled by commercial solvers. 6) We numerically demonstrate the effectiveness of our data-driven approach with performance guarantees, in Section 3.7 and 3.8. We claim that the proposed approach is suitable for problems which are subject to control-affine constraints.

3.2 Problem Formulation

We first present our data-driven predictive control approach in a general setting with the main goal of control design employing a finite data set. We then focus on an one-way highway speed-limit control as an application. This problem leverages a traffic model based on the Lighthill-Whitham-Richards (LWR) discretization [32, 61, 127, 138]. Finally, we adapt the proposed predictive control approach for our control problem, resulting into a stochastic optimization problem that can be used to find speed limits with performance guarantees.

3.2.1 General Framework

The goal of the proposed framework is to address system uncertainty explicitly and find a control law which satisfies system constraints and optimizes the expected objective in uncertainty. To achieve this, let us denote by $t \in \mathbb{N}$, $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ the time, system state and control at time t, respectively. We consider the control law u to be solutions of the following T-long

receding-horizon stochastic predictive control problem:

$$\begin{aligned} &\sup_{u} \ \mathbb{E}_{\mathbb{P}(u)}[\ell(u,x)], && \textbf{(P)}\\ &\text{s.t. } x \sim \mathbb{P}(u) \text{ characterized by} \\ &\quad x(t+1) = F(x(t),u(t),\xi(t)),\ t = 0,1,\ldots,T-1, \\ &\quad x \in \mathcal{Z}(u),\ u \in \mathcal{U},\ \xi \sim \mathbb{P}_{\xi}, \end{aligned}$$

where u is a concatenated variable of $u(0), u(1), \ldots, u(T-1)$ and x is that of those $x(t)$. Notice that x is a stochastic process and we denote by $\mathbb{P}(u)$ the distribution of x given u. The objective is to maximize the expectation of a given reward function $\ell : \mathbb{R}^{mT} \times \mathbb{R}^{nT} \to \mathbb{R}$ taken under $\mathbb{P}(u)$. We denote by $F : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^n$ the given system dynamics where ξ represents the uncertainty. We assume that the process x is constrained in a given set $\mathcal{Z}(u)$ and so does u in $\mathcal{U}$. We denote by $\mathbb{P}_{\xi}$ the unknown distribution of the uncertainty process.

Due to the unknown $\mathbb{P}_{\xi}$, Problem (**P**) cannot be solved exactly. To find a control law that solves (**P**), we propose a distributionally robust optimization (DRO) framework. By means of this, we will employ a finite set of realizations or samples of the random variables ξ to approximate the unknown distribution and compute a set of feasible u. The main appeal of this robust method is that it can provide *out-of-sample* probabilistic guarantees of performance [36]. In particular, at each t, let us assume that N samples of $x(0)$ and $\xi := (\xi(0), \ldots, \xi(T-1))$ are accessible. Under some mild conditions on these samples, we will show in Section 3.3 that a tractable optimization-based function $J(u)$ can be constructed using those state and uncertainty samples. The function $J(u)$ is a surrogate objective of (**P**) which accounts for the system dynamics as well as constraints on the states. In addition, given a confidence value $\beta \in (0,1)$, we provide the *out-of-sample* performance guarantee in the sense that the probability of the true objective

Algorithm 4. Data-driven predictive control with guarantees.

Initialize $t = 0$
while True **do**
    Take N measurements $x^{(l)}(t)$ and $\xi^{(l)}$, $l = 1, \ldots, N$
    Adapt an approach to optimize $J(u)$ over u
    Apply performance-guaranteed u to the system
    $t \leftarrow t + 1$

function in (**P**) being greater than $J(u)$ is greater than $1 - \beta$. In other words, we have

$$\text{Prob}^N \left(\mathbb{E}_{\mathbb{P}(u)} [\ell(u, x)] \geq J(u) \right) \geq 1 - \beta,$$

where $\text{Prob}^N := \mathbb{P}^N$ is the product probability over N sample trajectories of the system. Thus, with high probability, the choice of samples to approximate the problem will provide a minimum lower value for the original problem. As Problem (**P**) needs to be solved in a moving horizon fashion, the surrogate functions $J(u)$ is optimized similarly, as in Algorithm 4. Notice that, the functions $J(u)$ depend on samples as wells as the system structure, and the solution to $J(u)$ is nontrivial.

To enable the proposed tractable optimization method for $J(u)$ as in Section 3.5, we assume the following system structures.

Assumption 5 (Control-and-state-affine systems with bi-linear constraints). *The system dynamics $F(x, u, \xi)$ is continuous, affine in x and affine in u. In addition, the set $\mathcal{Z}(u)$ is composed of constraints which are linear and bi-linear in (x, u). And the set $\mathcal{U}$ is a finite set.*

Assumption 5 covers a wide class of dynamical systems, including linear systems and bi-linear systems. Furthermore, with a minor modification of the proposed reformulation techniques in Section 3.4, the proposed solution method can be extended to control-affine systems. We leave the extension as the future work. Next, we consider a traffic speed-limit control problem and design a set of speed controls using the proposed framework.

3.2.2 Highway Traffic Model

In order to introduce our discrete traffic model, we start by describing our time and space discretization, respectively. Let δ be a sufficiently small time-discretization step, then we let $\mathcal{T} = \{1, \ldots, T-1\}$ denote the set of time slots, where we identify time by index $t \equiv t\delta$. Further, let us consider a one-way highway of length L, and divide the highway into n segments. The topology of the highway can be described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{0, 1, \ldots, n\}$ and each node $v \in \mathcal{V}$ corresponds to the junction between two consecutive road segments. The fictitious node 0 represents the mainstream inflow into the highway. The graph edge set is $\mathcal{E} = \{(0,1), \cdots, (n-1, n)\}$ with each element $e = (v, v+1) \in \mathcal{E}$ corresponding to the road segment between nodes v and $v+1$ for all $v \in \{0, 1, \ldots, n-1\}$. To illustrate this, the reader is referred to Fig. 3.1. Nodes 0 and n are the source and sink nodes of the graph G, respectively. Further, node $v \in \mathcal{V} \setminus \{0\}$ is called an arrival node if there exists an on-ramp at node v. Similarly, node $v \in \mathcal{V} \setminus \{n\}$ is called a departure node if there exists an off-ramp at node v. Let $\mathcal{V}_A$ and $\mathcal{V}_D$ denote the set of arrival and departure nodes, respectively. By convention, we set node $0 \notin \mathcal{V}_A$ and node $n \notin \mathcal{V}_D$. For each edge $e = (v, v+1) \in \mathcal{E}$ with starting node v, let e' denote the on-ramp of edge e if node $v \in \mathcal{V}_A$ and let e'' denote the off-ramp of edge e if node $v+1 \in \mathcal{V}_D$. For each $e \in \mathcal{E}$, we denote by len_e and lane_e the segment length and the number of lanes on e, respectively. In particular, the highway length $L = \sum_{e \in \mathcal{E}} \text{len}_e$.

At each time $t \in \mathcal{T}$, macroscopic models of traffic use aggregated variables to describe the behavior of traffic. We start presenting constraints that relate the state variable $\rho_e(t)$, the average traffic density in segment e and period t, with the average traffic flow $f_e(t)$ on segment e, as well as the average traffic velocity $s_e(t)$ on segment e.

A first relationship is given by the traffic-flow definition

$$f_e(t) = s_e(t)\rho_e(t),$$

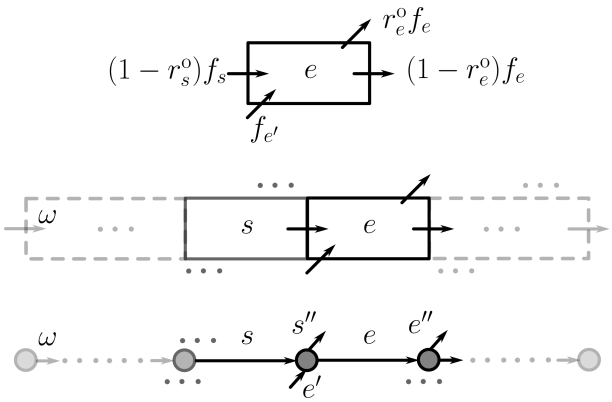


Figure 3.1. Segment of highways and its graph representation. The symbol e indicates a highway segment, and s is its preceding segment. The on-ramp and off-ramp of e are denoted by e' and e'', respectively. The variable ω, f and r_e^o represents the mainstream inflow, edge flows and fraction of outflows, respectively.

where $s_e(t)$ is taken to be a function of speed limit as follows. Let $u_e(t)$ denote the speed limit set on e and assume that the majority of drivers comply with it. Then,

$$s_e(t) = \min\{\overline{s}_e(\rho_e(t)), u_e(t)\},$$

where $\overline{s}_e(\rho_e(t))$ is the maximal admissible speed on e when the traffic density is $\rho_e(t)$. In practice, the non-negative values of $\overline{s}_e$ monotonically decrease as $\rho_e(t)$ increases, reflecting the safe-driving behavior.

Further, the fundamental diagram is a plot of the nonlinear relationship between traffic flow and traffic density at a location of a highway. In this plot, the density at which the traffic flow attains its maximum value, $\overline{f}_e$, is called the *critical density*, ρ_e^c. The density at which the traffic flow is zero is the traffic jam density $\overline{\rho}_e$. The traffic flow is an increasing function of ρ_e on $(0, \rho_e^c)$, and a strictly decreasing function of ρ_e on $(\rho_e^c, \overline{\rho}_e)$.

How speed limits affect the fundamental diagram has been a subject of debate [121, 137]. Following [121], we will assume that in the presence of speed limits, the flow rate and traffic density still hold a similar relationship, however the critical density will be a function of the velocity limit $u_e(t)$. Let $\overline{u}_e$ be the free flow velocity corresponding to a maximum value of flow

under no speed limits.[1] A comparison of two fundamental diagrams with constant speed limits $\overline{u}_e$ and u_e are shown in Fig. 3.2. Notice that the reduction of the speed limit from $\overline{u}_e$ to u_e increases the critical density and decreases the maximal flow rate on edge e.

To model the fundamental diagram in the presence of speed limits, consider edge $e \in \mathcal{E}$, and let $\rho_e^c(u_e(t))$ denote the critical density of edge e at speed limit $u_e(t)$. That is, the critical density $\rho_e^c(u_e(t))$ determines the traffic density at which the maximum edge flow $f_e(t)$ is achievable. Given speed limit $u_e(t)$, we will work with an approximation of the fundamental diagram of edge $e \in \mathcal{E}$ given as follows

$$f_e(t) = \begin{cases} u_e(t)\rho_e(t), & \text{if } \rho_e(t) \leq \rho_e^c(u_e(t)), \\ \tau_e \overline{u}_e \left(\overline{\rho}_e - \rho_e(t)\right), & \text{otherwise,} \end{cases} \tag{3.1}$$

with[2]

$$\rho_e^c(u_e(t)) := \left(\tau_e \overline{\rho}_e \overline{u}_e\right) / (\tau_e \overline{u}_e + u_e(t)),$$

where the parameter $\tau_e := \overline{f}_e / \left(\overline{u}_e \overline{\rho}_e - \overline{f}_e\right)$. To illustrate this, we refer the reader to Fig. 3.2.

Each segment e is congested when its density is higher than its critical density (see, e.g., [43, 137]). Since critical densities are determined by speed limits, we select the speed limits such that the following constraint is satisfied at all time slots $t \in \mathcal{T}$

$$0 \leq \rho_e(t) \leq \rho_e^c(u_e(t)), \ \forall e \in \mathcal{E}. \tag{3.2}$$

The constraint above ensures that the highway is not congested regardless of the flow rates. Using the constraint above and the fundamental diagram approximation in (3.1), we obtain

$$f_e(t) = u_e(t)\rho_e(t), \ \forall e \in \mathcal{E}, \ t \in \mathcal{T}. \tag{3.3}$$

[1] Given drivers behavior $\overline{s}_e$, the value $\overline{u}_e = \max_{\rho_e} \overline{s}_e(\rho_e)$, $\overline{\rho}_e$ admits $\overline{s}_e(\overline{\rho}_e) = 0$ and $\overline{f}_e$ maximizes f_e over any u_e and $\rho_e \in (0, \overline{\rho}_e)$. On the other hand, the function $\overline{s}_e$ is highly dependent on the physical structure of the segment e as well as random events, such as accidents and temporary lane closures (see, e.g., [34, 61]).

[2] The parameter $\tau_e \overline{u}_e$ is known as the backward wave speed (see, e.g., [74]).

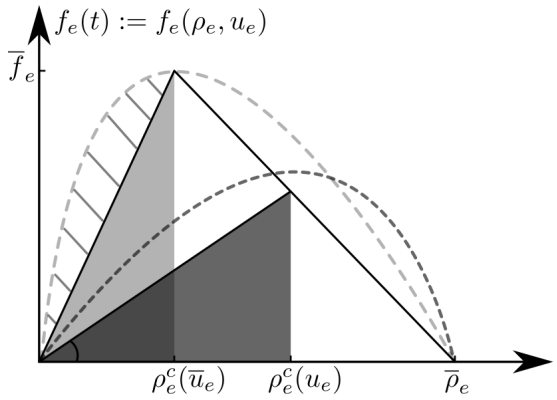


Figure 3.2. Flow rate as a function of traffic density of edge e for two speed limits $\overline{u}_e$ and u_e such that $\overline{u}_e \geq u_e$. The two nonlinear curves are the fundamental diagrams corresponding to each speed limit, while the straight lines are piece-wise linear approximations of these. The slope of linear approximations at the origin represents the speed limits $\overline{u}_e$ and u_e, respectively. The light and dark shaded region guarantees no congestion of edge e under speed limit $\overline{u}_e$ and u_e, respectively. (That is, under speed limit u_e, a density such that $\rho_e(t) \leq \rho_e^c(u_e)$, for all t, guarantees no congestion.) The slashed area reflects the compliance of drivers with speed limit $\overline{u}_e$. A higher compliance of drivers results in a smaller area.

Finally, based on the physical principle of conservation of mass, the discretized LWR model provides a set of difference equations for each road segment e which enable us to analyze the dynamics of traffic flows on highways [26, 138]. For each segment $e = (v, v+1) \in \mathcal{E}$, the dynamics of ρ_e are determined by the *demand* flow $f_e^{\mathrm{D}}(t)$ from v and the *supply* flow $f_e^{\mathrm{S}}(t)$ to $v+1$ as follows:

$$\rho_e(t+1) = \rho_e(t) + h_e(f_e^{\mathrm{D}}(t) - f_e^{\mathrm{S}}(t)), \quad \forall t \in \mathcal{T},$$

where $h_e := \delta/\mathrm{len}_e$. For numerical stability, δ must be selected such that $h_e \leq 1/\max_{t\in\mathcal{T}}\{u_e(t)\}$, $\forall e \in \mathcal{E}$ [55].

The supply flow $f_e^{\mathrm{S}}(t)$ denotes the maximal flow that can be transferred through edge e, and is given by

$$f_e^{\mathrm{S}}(t) = \begin{cases} u_e(t)\rho_e(t), & \text{if } \rho_e(t) \leq \rho_e^c(u_e(t)), \\ u_e(t)\rho_e^c(u_e(t)), & \text{otherwise,} \end{cases}$$

Notice that $f_e^{\mathrm{S}}(t) = f_e(t)$ when the highway segment e is not congested, i.e., the constraint in (3.2)

is satisfied. Consider junction $v \in \mathcal{V}_D$, i.e., the junction v is a departure node of the preceding edge of e or that of some edge s. Since $v \in \mathcal{V}_D$, a fraction of the supply $f_s^{\mathrm{S}}(t)$ will depart the highway through an off-ramp edge s'' and the rest will enter into succeeding segment e. Let $r_s^{\mathrm{o}}(t) \in [0, 1)$ denote the fraction of the supply $f_s^{\mathrm{S}}(t)$ that departs the system. Hence, the flow through the off-ramp edge s'' is $f_{s''}(t) := r_s^{\mathrm{o}}(t) f_s^{\mathrm{S}}(t)$. Notice that the fraction $r_s^{\mathrm{o}}(t)$ is determined by the drivers' actions. Therefore, r_s^{o} is random, and its value is unknown to the system operator in advance. Each random variable $r_s^{\mathrm{o}}(t)$ will have a nonempty support set denoted by $\mathcal{Z}_{r_s^{\mathrm{o}}(t)} \subset \mathbb{R}_{\geq 0}$.

The traffic demand $f_e^{\mathrm{D}}(t)$ depends on the supply of its preceding edge $s \in \mathcal{E}$ as well as the ramp flows on their connected junction v. At each $v \in \mathcal{V}_A$, a fraction of the traffic demand $f_e^{\mathrm{D}}(t)$ is originated from the on-ramp edge e'. Let $r_e^{\mathrm{in}}(t) \in [0, 1)$ denote the fraction of the traffic demand $f_e^{\mathrm{D}}(t)$ originated from the on-ramp edge e'. Hence, the on-ramp traffic flow is given by $f_{e'}(t) := r_e^{\mathrm{in}}(t) f_e^{\mathrm{D}}(t)$. Notice that the ratio $r_e^{\mathrm{in}}(t)$ is an exogenous parameter that depends on the traffic flow at the on-ramp edge e'. Hence, each ratio $r_e^{\mathrm{in}}(t)$ can be modeled as a random variable with nonempty support $\mathcal{Z}_{r_e^{\mathrm{in}}(t)} \subset \mathbb{R}_{\geq 0}$[3]. Then, by the conservation of flows, at each time slot t, the traffic demand $f_e^{\mathrm{D}}(t)$ must satisfy the following constraint:

$$f_e^{\mathrm{D}}(t) = f_s^{\mathrm{S}}(t) - f_{s''}(t) + f_{e'}(t), \quad \forall\, t \in \mathcal{T}.$$

At edge $e = (0, 1)$, we have $f_e^{\mathrm{D}}(t) := \omega(t)$ which is a random mainstream with support $\mathcal{Z}_{\omega(t)} \subset \mathbb{R}_{\geq 0}$. At each edge e, the demand $f_e^{\mathrm{D}}(t)$ must be admissible to edge e, i.e.,

$$f_e^{\mathrm{D}}(t) \leq \min\{\overline{f}_e,\ \tau_e \overline{u}_e \left(\overline{\rho}_e - \rho_e(t)\right)\}. \tag{3.4}$$

Notice that, the constraints (3.4) allows for transient speed of $f_e^{\mathrm{D}}(t)$ higher than speed limit $u_e(t)$, as long as the mean speed $s_e(t)$ complies with $u_e(t)$.

Let $\rho(0) = (\rho_1(0), \ldots, \rho_n(0))$ denote the traffic density of the highway with support

[3]For $v \notin \mathcal{V}_D$ or $v \notin \mathcal{V}_A$, the value of $r_s^{\mathrm{o}}(t)$ or $r_e^{\mathrm{in}}(t)$ is zero, respectively.

$\mathcal{Z}_{\rho(0)} \subset \mathbb{R}^n_{\geq 0}$. Using the constraints above, the traffic density dynamics at each time slot $t \in \mathcal{T}$ are given by

$$\begin{aligned} \rho_e(t+1) &= \rho_e(t) + h_e \frac{1 - r_s^{\mathrm{o}}(t)}{1 - r_e^{\mathrm{in}}(t)} f_s(t) - h_e f_e(t), \quad \forall e \in \mathcal{E} \setminus \{(0,1)\}, \\ \rho_e(t+1) &= \rho_e(t) + h_e(\omega(t) - f_e(t)), \quad e = (0,1). \end{aligned} \tag{3.5}$$

Recall that $f_e(t) = u_e(t)\rho_e(t)$. For each $e \in \mathcal{E} \setminus \{(0,1)\}$ and $t \in \mathcal{T}$, the constraint in (3.4) can be written as

$$\frac{1 - r_s^{\mathrm{o}}(t)}{1 - r_e^{\mathrm{in}}(t)} f_s(t) \leq \min\{\overline{f}_e,\ \tau_e \overline{u}_e \left(\overline{\rho}_e - \rho_e(t)\right)\}, \tag{3.6}$$

where s is the preceding edge of edge e.

Our goal is to design a set of speed limits for drivers. To achieve this goal, we consider a finite set of speed limits, and then approximate the fundamental diagram of each segment e with a finite set of piece-wise linear functions, as shown in Fig. 3.2. Let Γ be a finite set of feasible speed limits for the highway segments. More precisely, the speed limit of each edge $e \in \mathcal{E}$ must satisfy

$$u_e(t) \in \Gamma := \{\gamma^{(1)}, \ldots, \gamma^{(m)}\}, \quad t \in \mathcal{T}. \tag{3.7}$$

The set of real values Γ is determined by the physical structure of the highway as well as its maximal free flow speed and traffic jam density. As mentioned earlier, random events, such as traffic incidents or lane closure, can change these values.

3.2.3 Traffic-Control-Problem Formulation

We aim at maximizing the expected flow rate of highway segments while reducing congestion via speed limits. To compute the average flow, let $\mathbb{P}_{\varpi}$ denote the distribution of the concatenated random variable $\varpi := (\omega, \rho(0), r^{\mathrm{in}}, r^{\mathrm{o}})$. Given the parameters $\{\overline{f}_e\}_{e\in\mathcal{E}}$, $\{\overline{\rho}_e\}_{e\in\mathcal{E}}$, and Γ, the problem of computing speed limits which maximize the expected flow, can be formulated

as follows:

$$\begin{aligned} \max_{u,\rho} \quad & \mathbb{E}_{\mathbb{P}_{\varpi}}\left[\frac{1}{T}\sum_{e\in\mathcal{E},t\in\mathcal{T}}\rho_e(t)u_e(t)\right], \\ \text{s.t.} \quad & (3.2),\ (3.3),\ (3.5),\ (3.6),\ (3.7), \end{aligned} \tag{\textbf{P}}$$

where ρ and u are the concatenated variables of $\{\rho_e(t)\}_{e\in\mathcal{E},t\in\mathcal{T}}$[4] and $\{u_e(t)\}_{e\in\mathcal{E},t\in\mathcal{T}}$, respectively.

Problem (**P**) is nonconvex. In addition, the probability distribution $\mathbb{P}_{\varpi}$ is unknown, i.e., it is impossible to compute the expected flow (i.e., the objective function) exactly. Our goal is to compute a set of speed limits that are feasible to Problem (**P**) and guarantee a minimum achievable expected flow in the presence of uncertainty on $\mathbb{P}_{\varpi}$. To achieve this goal, we adapt the proposed framework to compute the desired speed limits. In this way, given an optimal speed limit u and a set of N samples, we obtain $J(u)$, an achievable average flow rate—let us call it certificate. In particular, $J(u)$ is a function of the N random samples. This certificate is a minimum with confidence $\beta \in (0,1)$ in the sense that the probability of the true objective function being greater than $J(u)$ is greater than $1-\beta$. In other words, let Prob^N denote the product probability distribution over N samples. Under certain conditions on $\mathbb{P}_{\varpi}$, the proposed approach guarantees that the following out-of-sample performance constraint is satisfied:

$$\text{Prob}^N\left(\mathbb{E}_{\mathbb{P}_{\varpi}}\left[\frac{1}{T}\sum_{e\in\mathcal{E},t\in\mathcal{T}}\rho_e(t)u_e(t)\right] \geq J(u)\right) \geq 1-\beta. \tag{3.8}$$

The probabilistic guarantee (3.8) enables us to evaluate the performance of a feasible solution u to Problem (**P**) via only finite samples of ϖ and a lower bound $J(u)$. We call a solution u with the probabilistic guarantee (3.8) *a data-driven Speed-Limit control*.

Remark 9 (Implementation of data-driven control). With online-accessible samples of ϖ and parameters $(\overline{f}, \overline{\rho}, \overline{u})$ which are related to real-time highway random events, the data-driven control u can be achieved via online solutions to (**P**) in a moving horizon fashion. Fig. 3.3 demonstrates

[4] $\rho := (\rho_1(0), \rho_2(0), \ldots, \rho_n(0), \rho_1(1), \ldots, \rho_n(T-1))$.

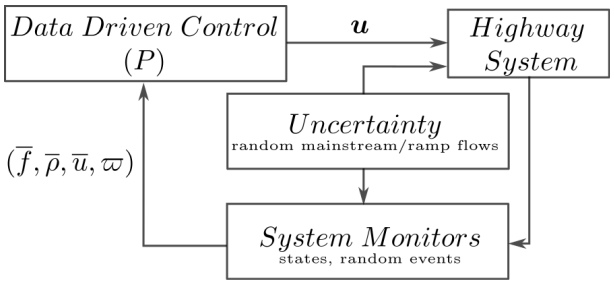


Figure 3.3. Data-driven speed-limit control on highways under random ramp flows and system events.

how to implement the proposed approach in real-world applications.

Remark 10 (Admissible operation zone). In this chapter, we propose a set of speed-limit controls that prevents congestion with probabilistic guarantees. Note that the existence of such controls is highly dependent on the feasibility of Problem (P). When traffic demands are higher than the highway capacity in a sufficiently long time, congestion is inevitable. Therefore, there is no hope to prevent congestion via the speed-limit control in the presence of high traffic demands. In such scenarios, we set speed limits to a set of predefined values.

3.3 Performance Guaranteed Speed-Limit Design

Our goal is to compute a set of speed limits with certain out-of-sample performance guarantees. To achieve this goal, we follow a four-step procedure. First, we reformulate Problem (**P**) into an equivalent problem (call it Problem (**P1**)). Second, we propagate the admissible sample trajectories using the N measurements of ϖ. Third, we adopt a distributionally robust optimization approach to (**P1**). The first three steps enable us to obtain a distributionally robust optimization framework for computing speed limits with guarantees equivalent to (3.8). Finally, we obtain a tractable problem reformulation.

Step 1: (Equivalent reformulation of (P)) Traffic densities $\rho_e(t)$, for all $e \in \mathcal{E}$ and $t \in \mathcal{T}$, are random since traffic inflows and outflows are random in each segment e. Using this observation, we now consider the variable ρ in Problem (**P**) as a random variable, and derive an

equivalent Problem (**P1**) via a reformulation of the constraints in (**P**).

Given a speed limit u that satisfies the constraint (3.7), let $\mathbb{P}(u)$ and $\mathcal{Z}(u)$ denote the probability distribution of variable ρ and the support of ρ, respectively.[5] Recall that in Problem (**P**), constraints (3.2) ensure no congestion on the highway. Hence, the given u should be such that $\mathcal{Z}(u) \subseteq \{\rho \in \mathbb{R}^{nT} \mid (3.2)\}$. Without loss of generality, we consider the largest possible support $\mathcal{Z}(u) := \{\rho \in \mathbb{R}^{nT} \mid (3.2)\}$. To fully characterize random variable ρ, the distribution of $\mathbb{P}(u)$ needs to be determined. Using the traffic density dynamics in (3.5), flow representation in (3.3) and sustainability constraints in (3.6), the probability distribution $\mathbb{P}(u)$ can be represented as a convolution of the distribution $\mathbb{P}_{\varpi}$.

Let $\mathcal{M}(\mathcal{Z}(u))$ denote the space of all probability distributions supported on $\mathcal{Z}(u)$. Let us write the objective function of Problem (**P**) compactly as

$$H(u;\rho) := \frac{1}{T} \sum_{e\in\mathcal{E},t\in\mathcal{T}} \rho_e(t)u_e(t),$$

and reformulate (**P**) as follows

$$\begin{aligned} \max_{u} \quad & \mathbb{E}_{\mathbb{P}(u)}[H(u;\rho)], && \textbf{(P1)} \\ \text{s.t.} \quad & \mathbb{P}(u) \text{ characterized by (3.3), (3.5), (3.6) and } \mathbb{P}_{\varpi}, \\ & \mathbb{P}(u) \in \mathcal{M}(\mathcal{Z}(u)), \text{ (3.7)}. \end{aligned}$$

Notice that Problems (**P**) and (**P1**) are equivalent. Hence, we obtain the performance guarantee of (**P1**) by considering the induced out-of-sample performance on $\mathbb{P}(u)$, written as $\mathbb{E}_{\mathbb{P}(u)}[H(u;\rho)]$. For all problems derived later, we use the performance guarantees equivalent to (3.8), as follows

$$\text{Prob}^N\left(\mathbb{E}_{\mathbb{P}(u)}[H(u;\rho)] \geq J(u)\right) \geq 1-\beta. \tag{3.9}$$

[5]The support $\mathcal{Z}(u)$ is the smallest closed set such that $P(\rho \in \mathcal{Z}(u)) = 1$.

Recall that Prob^N denotes the probability that the event $\mathbb{E}_{\mathbb{P}(u)}[H(u;\rho)] \geq J(u)$ happens on the N product of the sample space that defines ρ, the value $J(u)$ is the certificate to be determined, and $\beta \in (0,1)$ is the desired confidence value. Next, we characterize the probability distribution $\mathbb{P}(u)$ using the N sample measurements of ϖ.

Step 2: (Admissible sample trajectory propagation) Let $\mathcal{L} = \{1, \ldots, N\}$ denote the index set for the N realizations of the random variable ϖ, and let $\{\varpi^{(l)}\}_{l\in\mathcal{L}}$, where $\varpi^{(l)} := (\omega^{(l)}, \rho^{(l)}(0), r^{\text{in},(l)}, r^{\text{o},(l)})$, denote the set of independent and identically distributed (i.i.d.) realizations of ϖ. Given a speed limit u and measurement $\varpi^{(l)}$, a unique traffic density trajectory $\rho^{(l)}$ can be computed by using (3.3), (3.5). Notice that this trajectory is unique since density dynamics (3.5) is linear (in ρ) for a given speed limit u and measurement $\varpi^{(l)}$. Further, the resulting $\rho^{(l)}$ is an *admissible* traffic trajectory if the given u achieves the flow sustainability constraints (3.6). Given these realizations $\{\varpi^{(l)}\}_{l\in\mathcal{L}}$, the admissible sample trajectories $\{\rho^{(l)}\}_{l\in\mathcal{L}}$ of the random traffic flow densities for each edge $e \in \mathcal{E}$ with its precedent $s \in \mathcal{E} \cup \varnothing$, are given by

$$
\begin{aligned}
&\rho_e^{(l)}(t+1) = h_e \frac{1 - r_s^{\text{o},(l)}(t)}{1 - r_e^{\text{in},(l)}(t)} u_s(t)\rho_s^{(l)}(t) - h_e u_e(t)\rho_e^{(l)}(t) + \rho_e^{(l)}(t), \quad \forall e \in \mathcal{E} \setminus \{(0,1)\}, \\
&\rho_e^{(l)}(t+1) = \rho_e^{(l)}(t) + h_e(\omega^{(l)}(t) - u_e(t)\rho_e^{(l)}(t)), \quad e = (0,1), \\
&\frac{1 - r_s^{\text{o},(l)}(t)}{1 - r_e^{\text{in},(l)}(t)} u_s(t)\rho_s^{(l)}(t) \leq \min\left\{\overline{f}_e,\ \tau_e \overline{u}_e \left(\overline{\rho}_e - \rho_e^{(l)}(t)\right)\right\}, \quad \forall e \in \mathcal{E} \setminus \{(0,1)\},
\end{aligned}
\tag{3.10}
$$

for all $t \in \mathcal{T}$, $l \in \mathcal{L}$. The following lemma establishes that $\{\rho^{(l)}\}_{l\in\mathcal{L}}$ are i.i.d. samples from $\mathbb{P}(u)$.

Lemma 8 (Independent and identically distributed sample generators of ρ). *Given a speed limit u and a set of i.i.d. realizations of ϖ, the system dynamics* (3.10) *generate i.i.d. admissible sample trajectories $\{\rho^{(l)}\}_{l\in\mathcal{L}}$ of $\mathbb{P}(u)$.*

Proof. Continuous functions of i.i.d. random variables generate i.i.d. random variables. Thus, the admissible sample trajectories $\{\rho^{(l)}\}_{l\in\mathcal{L}}$ generated by (3.10) are i.i.d. realizations of $\mathbb{P}(u)$. □

Let $\mathcal{M}_{\text{lt}}(\mathcal{Z}_\varpi) \subset \mathcal{M}(\mathcal{Z}_\varpi)$ denote the space of all light-tailed probability distributions

supported on $\mathcal{Z}_\varpi$[6]. We make the following assumption on the probability distribution $\mathbb{P}_\varpi$:

Assumption 6 (Light-tailed unknown distributions). *The distribution $\mathbb{P}_\varpi$ satisfies $\mathbb{P}_\varpi \in \mathcal{M}_{\text{lt}}(\mathcal{Z}_\varpi)$, i.e., there exists an exponent $a > 1$ such that $b := \mathbb{E}_{\mathbb{P}_\varpi}\left[\exp(\|\varpi\|_1^a)\right] < \infty$.*

Remark 11 (Accessible light-tailed i.i.d. samples of ϖ). The random variable ϖ essentially represents the random flows and densities on the highway. This results into an unknown compact support of $\mathbb{P}_\varpi$, indicating that $\mathbb{P}_\varpi$ is also light-tailed. Further, online samples of ϖ can come from various independent system monitors, e.g., Traffic Performance Measurement Systems (T-PeMS), or real-time GPS systems, which provide online i.i.d. samples of ϖ.

The following lemma establishes that the probability distribution of traffic densities is light-tailed when $\mathbb{P}_\varpi$ is light-tailed.

Lemma 9 (Light-tailed distribution of ρ). *Let Assumption 6 hold, then $\mathbb{P}(u) \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u))$.*

Proof. The proof consists of two steps. First, we explore the boundedness property of $\rho_e(t)$ w.r.t. ϖ, for all $e \in \mathcal{E}$ and $t \in \mathcal{T}$. Second, we claim that there exists an $a > 1$ such that $\mathbb{E}_{\mathbb{P}(u)}[\exp(\|\rho\|_1^a)] < \infty$.

Step 1: (Boundedness of ρ) Consider for each speed limit u and time $t \in \mathcal{T} \setminus \{0\}$. Let $A(u,t)$ denote the matrix that is consistent with the highway system $\mathcal{G}$, let $B(t)$ denote the column vector that encodes the mainstream flow ω, and write the density $\rho(t)$ in the following compact form

$$\rho(t) = \Phi(t,0)\rho(0) + \sum_{\tau=0}^{t-1} \Phi(t,\tau+1)B(\tau),$$

where

$$\Phi(t,\tau) := \begin{cases} I_n, & \text{if } t = \tau, \\ A(u,t-1)A(u,t-2)\cdots A(u,\tau), & \text{if } t > \tau, \end{cases}$$

[6]For any set $\mathcal{Z}$, we use $\mathcal{M}_{\text{lt}}(\mathcal{Z})$ to denote the space of all light-tailed probability distributions supported on $\mathcal{Z}$.

$$A(u,t) = \begin{bmatrix} m_1(t) & & & \\ n_1(t) & \ddots & & \\ & \ddots & \ddots & \\ & & & \\ & & & \\ & & n_{n-1}(t) & m_n(t) \end{bmatrix}, B(t) = \begin{bmatrix} h_{(0,1)}\omega(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

with

$$m_i(t) = 1 - h_e u_e(t), \quad \forall i \in \{1,\dots,n\}, e = (i-1,i),$$
$$n_j(t) = \min\left\{ \frac{1 - r_s^{\mathrm{o}}(t)}{1 - r_e^{\mathrm{in}}(t)} h_e u_s(t), \frac{\overline{f}_e}{\rho_s(t)}, \tau_e \overline{u}_e \frac{\overline{\rho}_e - \rho_e(t)}{\rho_s(t)} \right\},$$
$$\forall j \in \{1,\dots,n-1\}, s = (i-1,i), e = (i,i+1).$$

Given that each component of $A(u,t)$ is bounded for fixed u and t, the induced norm of $A(u,t)$ is also bounded and we denote their universal bound by A_0. Similarly, we denote by $\overline{h}$ the upper bound on h_e, $e \in \mathcal{E}$. Then for every t we can bound the infinity norm of $\rho(t)$ as follows

$$\begin{aligned} \|\rho(t)\|_\infty &\le \|\Phi(t,0)\|_\infty \|\rho(0)\|_\infty + \overline{h} \sum_{\tau=0}^{t-1} \|\Phi(t,\tau+1)\|_\infty \|\omega(\tau)\|_\infty, \\ &\le A_0^t \|\rho(0)\|_\infty + \overline{h} \sum_{\tau=0}^{t-1} A_0^{t-\tau-1} \|\omega(\tau)\|_\infty, \\ &\le M_1(t) \left(\|\rho(0)\|_\infty + \sum_{\tau=0}^{t-1} \|\omega(\tau)\|_\infty \right), \\ &\le (t+1) M_1(t) \|\varpi\|_\infty, \le (t+1) M_1(t) \|\varpi\|_1, \end{aligned}$$

where $M_1(t) := \max\{A_0^t,\ \overline{h},\ \overline{h} A_0^t\}$.

Step 2: (Light-tailed distribution) Given an u, consider a $t^\star \in \operatorname{argmax}_{t \in \mathcal{T} \setminus \{0\}} \{\|\rho(t)\|_\infty\}$.

Then by norm equivalence we have

$$\begin{aligned}\|\rho\|_1 \leq n\|\rho\|_\infty &= n \max_{t\in\mathcal{T}\setminus\{0\}} \{\|\rho(t)\|_\infty\},\\ &= n\|\rho(t^\star)\|_\infty, \leq n(t^\star+1)M_1(t^\star)\|\varpi\|_1.\end{aligned}$$

Then for any $a > 1$ with $\mathbb{E}_{\mathbb{P}_\varpi}[\exp(\|\varpi\|_1^a)] < \infty$, let $M_2 = [n(t^\star+1)M_1(t^\star)]^a < \infty$ and we have

$$\begin{aligned}\mathbb{E}_{\mathbb{P}(u)}[\exp(\|\rho\|_1^a)] &\leq \mathbb{E}_{\mathbb{P}_\varpi}[\exp(M_2\|\varpi\|_1^a)],\\ &= \exp(M_2)^a \cdot \mathbb{E}_{\mathbb{P}_\varpi}[\exp(\|\varpi\|_1^a)] < \infty.\end{aligned}$$

That is, $\mathbb{P}(u)$ is light tailed. □

The above Lemmas 8 and 9 enable the application of the proposed framework to (**P1**) in the next step.

Step 3: (Performance-guarantee certificate) Given a speed limit u, we design a certificate $J(u)$ that satisfies the performance guarantee condition (3.9). To achieve this goal, the proposed approach consists of solving a robust version of the problem over a set of distributions. In particular, we will consider a set of distributions $\mathcal{P}(u)$ that is small, tractable and yet rich enough to contain $\mathbb{P}(u)$ with high probability. Then by evaluating (**P1**) under the worst-case distribution in $\mathcal{P}(u)$, the performance of (**P1**) in the form of (3.9) can be guaranteed.

Consider the Wasserstein ball[7] $\mathbb{B}_\epsilon(\hat{\mathbb{P}}(u))$ of center $\hat{\mathbb{P}}(u) := (1/N)\sum_{l\in\mathcal{L}} \delta_{\{\rho^{(l)}\}}$ and radius ϵ. Notice that the center $\hat{\mathbb{P}}(u)$ is obtained using the point mass operator δ under i.i.d. admissible sample trajectories $\{\rho^{(l)}\}_{l\in\mathcal{L}}$. These trajectories are distributed according to $\mathbb{P}(u)$ which is a function of the controlled system dynamics (3.10) and samples of ϖ. Then we propose certificates

[7]Let $\mathcal{M}(\mathcal{Z})$ denote the space of all probability distributions supported on $\mathcal{Z}$. Then for any two distributions $\mathbb{Q}_1$, $\mathbb{Q}_2 \in \mathcal{M}(\mathcal{Z})$, the Wasserstein metric [59] $d_W : \mathcal{M}(\mathcal{Z})\times\mathcal{M}(\mathcal{Z}) \to \mathbb{R}_{\geq 0}$ is defined by

$$d_W(\mathbb{Q}_1,\mathbb{Q}_2) := \min_{\Pi} \int_{\mathcal{Z}\times\mathcal{Z}} \|\xi_1 - \xi_2\|_1 \Pi(d\xi_1, d\xi_2),$$

where Π is in a set of all distributions on $\mathcal{Z}\times\mathcal{Z}$ with marginals $\mathbb{Q}_1$ and $\mathbb{Q}_2$. A closed Wasserstein ball of radius ϵ centered at a distribution $\mathbb{P} \in \mathcal{M}(\mathcal{Z})$ is denoted by $\mathbb{B}_\epsilon(\mathbb{P}) := \{\mathbb{Q} \in \mathcal{M}(\mathcal{Z}) \mid d_W(\mathbb{P},\mathbb{Q}) \leq \epsilon\}$.

$J(u)$ of (**P1**) in the following theorem.

Theorem 6 (Performance guarantees of (P1)). *Let us assume that N i.i.d. samples of ϖ are given, together with a speed limit u, and confidence value β, and that Assumption 6 on light-tailed distributions of ϖ holds. Further, let us define the set of distributions $\mathcal{P}(u)$ as follows:*

$$\mathcal{P}(u) := \mathbb{B}_{\epsilon(\beta)}(\hat{\mathbb{P}}(u)) \cap \mathcal{M}_{\text{lt}}(\mathcal{Z}(u)).$$

Then, there exists a Wasserstein radius $\epsilon := \epsilon(\beta)$, depending on the confidence value β and Assumption 6, such that $\mathbb{P}(u)$ is in the set of distributions $\mathcal{P}(u)$ as described in (**P1**) *with probability at least $1-\beta$, i.e.,*

$$\text{Prob}^N\left(\mathbb{P}(u) \in \mathcal{P}(u)\right) \geq 1-\beta.$$

Further, the following proposed certificate $J(u)$ of (**P1**) *satisfies guarantee* (3.9)

$$J(u) := \inf_{\mathbb{Q} \in \mathcal{P}(u)} \mathbb{E}_{\mathbb{Q}}\left[H(u;\rho)\right].$$

Proof. We prove the result in two steps. First, we show the proposed set $\mathcal{P}(u)$ contains the unknown distribution $\mathbb{P}(u)$ with high probability. Then, we show that the proposed certificate $J(u)$ provides performance guarantees as in (3.9).

Step 1: (Tractable set containing $\mathbb{P}(u)$) Under Assumption 6 and using N i.i.d. samples of ϖ, we obtain i.i.d. samples $\{\rho^{(l)}\}_{l\in\mathcal{L}}$ of $\mathbb{P}(u)$ via Lemma 8. Then, with the distribution

$$\hat{\mathbb{P}}(u) := \frac{1}{N}\sum_{l\in\mathcal{L}} \delta_{\{\rho^{(l)}\}},$$

we quantify the relation between $\hat{\mathbb{P}}(u)$ and $\mathbb{P}(u)$ via Lemma 9 and the following theorem:

Theorem 7 (Measure concentration [40, Theorem 2]). *If* $\mathbb{P}(u) \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u))$*, then*

$$\text{Prob}^N\{d_W(\mathbb{P}(u), \hat{\mathbb{P}}(u)) > \epsilon\} < \begin{cases} c_1 e^{-c_2 N \epsilon^{\max\{2,\ell\}}}, & \text{if } \epsilon \leq 1, \\ c_1 e^{-c_2 N \epsilon^a}, & \text{if } \epsilon > 1, \end{cases}$$

for all $N \geq 1$*,* $\ell \neq 2$*, and* $\epsilon > 0$*, where the parameter* ℓ *is the dimension of* ρ*, and parameters* c_1*,* c_2 *are positive constants that only depend on* ℓ*,* a *and* b *as in Assumption 6.* □

Let us select $\epsilon := \epsilon(\beta)$ to be the following

$$\epsilon(\beta) := \begin{cases} \left(\frac{\log(c_1 \beta^{-1})}{c_2 N}\right)^{1/\max\{2,\ell\}}, & \text{if } N \geq \frac{\log(c_1 \beta^{-1})}{c_2}, \\ \left(\frac{\log(c_1 \beta^{-1})}{c_2 N}\right)^{1/a}, & \text{if } N < \frac{\log(c_1 \beta^{-1})}{c_2}, \end{cases}$$

then Theorem 7 leads to

$$\text{Prob}^N\left(d_W(\mathbb{P}(u), \hat{\mathbb{P}}(u)) > \epsilon(\beta)\right) < \beta,$$

or, equivalently,

$$\text{Prob}^N\left(d_W(\mathbb{P}(u), \hat{\mathbb{P}}(u)) \leq \epsilon(\beta)\right) \geq 1 - \beta.$$

From the definition of the Wasserstein ball $\mathbb{B}_{\epsilon(\beta)}(\hat{\mathbb{P}}(u))$ and the fact that $\mathbb{P}(u) \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u))$, we have

$$\text{Prob}^N\left(\mathbb{P}(u) \in \mathbb{B}_{\epsilon(\beta)}(\hat{\mathbb{P}}(u))\right) \geq 1 - \beta,$$

$$\text{Prob}^N\left(\mathbb{P}(u) \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u))\right) = 1,$$

which results in

$$\text{Prob}^N\left(\mathbb{P}(u) \in \mathcal{P}(u)\right) \geq 1 - \beta,$$

with

$$\mathcal{P}(u) := \mathbb{B}_{\epsilon(\beta)}(\hat{\mathbb{P}}(u)) \cap \mathcal{M}_{\text{lt}}(\mathcal{Z}(u)).$$

Step 2: (Certificate of (P1)) From the previous reasoning, for a given u we have

$\mathbb{P}(u) \in \mathcal{P}(u)$ with probability at least $1-\beta$. Then, with probability at least $1-\beta$ the objective value of (**P1**) satisfies the following

$$\mathbb{E}_{\mathbb{P}(u)}[H(u;\rho)] \geq \inf_{\mathbb{Q}\in\mathcal{P}(u)} \mathbb{E}_{\mathbb{Q}}[H(u;\rho)].$$

Let $J(u)$ be

$$J(u) := \inf_{\mathbb{Q}\in\mathcal{P}(u)} \mathbb{E}_{\mathbb{Q}}[H(u;\rho)],$$

which results in

$$\text{Prob}^N\left(\mathbb{E}_{\mathbb{P}(u)}[H(u;\rho)] \geq J(u)\right) \geq 1-\beta.$$

Then the proposed certificate $J(u)$ of (**P1**) satisfies guarantee (3.9), which completes the proof. □

Remark 12 (Effect of Assumption 6 on $J(u)$). The certificate $J(u)$ highly depends on the set $\mathcal{P}(u)$ and the Wasserstein radius $\epsilon(\beta)$. In Theorem 6, the value $\epsilon(\beta)$ is calculated via the parameters a and b in Assumption 6. As these parameters may not be known, one can determine $\epsilon(\beta)$ in a data driven fashion via Monte-Carlo simulations. That is, we start by setting $\epsilon(\beta) = 0$ and gradually increase it until the performance guarantees (3.9) hold with that given β for a sufficiently large number of simulation runs.

Theorem 6 provides each feasible solution u of (**P1**) with a certificate $J(u)$ that guarantees performance as in (3.9). This motivates a tractable reformulation of (**P1**) as follows.

Step 4: (Tractable reformulation of (P1)) Our goal is to obtain a speed limit u that maximizes the average flow through the highway as in (**P1**) while ensuring that the performance guarantee condition (3.9) is satisfied. Given a set of inflow-and-outflow-related samples $\{\varpi^{(l)}\}_{l\in\mathcal{L}}$, a speed limit u that provides the highest $J(u)$ (i.e., the best objective lower bound), can be computed by solving the following optimization problem:

$$\sup_{u \text{ s.t. } (3.7)} \quad J(u). \tag{**P2**}$$

Problem (**P2**) is an infinite-dimensional optimization problem, and, hence, it is hard to solve. The following theorem, provides a finite-dimensional reformulation of Problem (**P2**), called (**P3**), and shows that problems (**P2**) and (**P3**) are equivalent for (u, J).

Theorem 8 (Tractable reformulation of (P2)). *Consider*

$$
\begin{aligned}
\max_{u,\rho,\lambda,\mu,\nu,\eta} \quad & -\lambda\epsilon(\beta) - \frac{1}{N}\sum_{e\in\mathcal{E},t\in\mathcal{T},l\in\mathcal{L}} \overline{f}_e\overline{\rho}_e\eta_e^{(l)}(t) + \frac{1}{N}\sum_{l\in\mathcal{L}}\left\langle \nu^{(l)}, \rho^{(l)}\right\rangle, && \textbf{(P3)}\\
\text{s.t.} \quad & [\overline{f}\otimes\mathbf{1}_T + (\overline{\rho} - \rho^c(\overline{u}))\otimes\mathbf{1}_T \circ u]\circ\eta^{(l)} - \mu^{(l)} \geq \mathbf{0}_{nT},\ \forall\, l\in\mathcal{L},\\
& \nu^{(l)} = \mu^{(l)} + \frac{1}{T}u,\ \forall\, l\in\mathcal{L},\\
& \|\nu^{(l)}\|_\star \leq \lambda,\ \forall\, l\in\mathcal{L},\\
& \eta^{(l)} \geq \mathbf{0}_{nT},\ \forall\, l\in\mathcal{L},\\
& (3.7),\ (3.10),
\end{aligned}
$$

where decision variables $(u, \rho, \lambda, \mu, \nu, \eta)$ are concatenated versions of $u_e(t)$, $\rho_e^{(l)}(t)$, λ, $\mu_e^{(l)}(t)$, $\nu_e^{(l)}(t)$, $\eta_e^{(l)}(t) \in \mathbb{R}$, for all $l \in \mathcal{L}$, $t \in \mathcal{T}$, and $e \in \mathcal{E}$. The value $\rho^c(\overline{u}) := \overline{f}/\overline{u}$ is the vector of critical densities under the free flow, and $\overline{\rho}$ is the jam density vector. The norm $\|\cdot\|_\star := \|\cdot\|_\infty$.

Problem (**P2**) *is equivalent to* (**P3**) *in the sense that their optimal objective values coincide and the set of optimizers of* (**P2**) *are the projection of that of* (**P3**)*. Further, for any feasible point $(u, \rho, \lambda, \mu, \nu, \eta)$ of* (**P3**)*, let $\hat{J}(u)$ denote the value of its objective function. Then the pair $(u, \hat{J}(u))$ gives a data-driven solution u with an estimate of its certificate $J(u)$ by $\hat{J}(u)$, such that the performance guarantee* (3.9) *holds for $(u, \hat{J}(u))$.*

Proof. We achieve it by three steps. First, we show that Problem (**P2**) can be equivalently reduced to a finite-dimensional problem. Then we equivalently reformulate the resulting problem into a maximization problem. Finally, we show the performance guarantees of (**P3**).

Step 1: (Finite-dimensional reduction of (P2)) We express $J(u)$ as follows

$$
\begin{aligned}
J(u) = &\begin{cases} \inf\limits_{\mathbb{Q}} & \int_{\mathcal{Z}(u)} H(u;\rho)\mathbb{Q}(d\rho), \\ \text{s.t.} & \mathbb{Q} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u)),\ d_W(\mathbb{Q}, \hat{\mathbb{P}}(u)) \le \epsilon(\beta), \end{cases} \\
= &\begin{cases} \inf\limits_{\mathbb{Q},\Pi} & \int_{\mathcal{Z}(u)} H(u;\rho)\mathbb{Q}(d\rho), \\ \text{s.t.} & \int_{\mathcal{Z}(u)\times\mathcal{Z}(u)} \|\rho-\rho'\|_1 \Pi(d\rho, d\rho') \le \epsilon(\beta), \\ & \Pi \text{ is a distribution of } \rho \text{ and } \rho' \\ & \qquad \text{with marginals } \mathbb{Q} \text{ and } \hat{\mathbb{P}}(u) \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u)), \end{cases} \\
= &\begin{cases} \inf\limits_{\mathbb{Q}^{(l)},\, l\in\mathcal{L}} & \frac{1}{N}\sum\limits_{l\in\mathcal{L}} \int_{\mathcal{Z}(u)} H(u;\rho)\mathbb{Q}^{(l)}(d\rho), \\ \text{s.t.} & \frac{1}{N}\sum\limits_{l\in\mathcal{L}} \int_{\mathcal{Z}(u)} \|\rho-\rho^{(l)}\|_1 \mathbb{Q}^{(l)}(d\rho) \le \epsilon(\beta), \\ & (3.10),\ \mathbb{Q}^{(l)} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u)),\ \forall l \in \mathcal{L}, \end{cases} \\
= &\begin{cases} \inf\limits_{\mathbb{Q}^{(l)},\, l\in\mathcal{L}} \sup\limits_{\lambda\ge 0} & \frac{1}{N}\sum\limits_{l\in\mathcal{L}} \int_{\mathcal{Z}(u)} H(u;\rho)\mathbb{Q}^{(l)}(d\rho) + \lambda\left(\frac{1}{N}\sum\limits_{l\in\mathcal{L}} \int_{\mathcal{Z}(u)} \|\rho-\rho^{(l)}\|_1 \mathbb{Q}^{(l)}(d\rho) - \epsilon(\beta)\right), \\ \text{s.t.} & (3.10),\ \mathbb{Q}^{(l)} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u)),\ \forall l \in \mathcal{L}, \end{cases}
\end{aligned}
$$

where the first equality applies the definition of the expectation operation; the second equality uses the definition of Wasserstein metric; the third equality exploits the fact that the joint distribution Π can be characterized by the marginal distribution $\hat{\mathbb{P}}(u)$ of ρ' and the conditional distributions $\mathbb{Q}^{(l)}$ of ρ given $\rho' = \rho^{(l)}$, $l \in \mathcal{L}$, written as

$$
\Pi := \frac{1}{N}\sum_{l\in\mathcal{L}} \delta_{\{\rho^{(l)}\}} \otimes \mathbb{Q}^{(l)},
$$

where admissible sample trajectories $\{\rho^{(l)}\}_{l\in\mathcal{L}}$ come from (3.10) and each conditional distribution $\mathbb{Q}^{(l)}$ is supported on $\mathcal{M}_{\text{lt}}(\mathcal{Z}(u))$; and, on the other hand, the fourth equality applies the Lagrangian representation of the problem.

Then, with an extended version of the strong duality results for the moment problem [119, Lemma 3.4], the order of the inf-sup operator in the resulting representation of $J(u)$ can be switched, resulting in the following expression

$$J(u) = \begin{cases} \sup\limits_{\lambda \geq 0} \inf\limits_{\mathbb{Q}^{(l)},\, l \in \mathcal{L}} -\lambda\epsilon(\beta) + \frac{1}{N} \sum\limits_{l \in \mathcal{L}} \int_{\mathcal{Z}(u)} \Big(\lambda \|\rho - \rho^{(l)}\|_1 + H(u;\rho) \Big) \mathbb{Q}^{(l)}(d\rho), \\ \text{s.t.} \quad (3.10),\ \mathbb{Q}^{(l)} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u)),\ \forall l \in \mathcal{L}, \end{cases}$$

Move the inf operator into the sum operator, we have

$$J(u) = \begin{cases} \sup\limits_{\lambda \geq 0} -\lambda\epsilon(\beta) + \frac{1}{N} \sum\limits_{l \in \mathcal{L}} \left\{ \inf\limits_{\mathbb{Q}^{(l)} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u))} \mathbb{E}_{\mathbb{Q}^{(l)}} \Big[\lambda \|\rho - \rho^{(l)}\|_1 + H(u;\rho) \Big] \right\}, \\ \text{s.t.} \quad (3.10). \end{cases}$$

For each $l \in \mathcal{L}$, we claim that

$$\inf_{\mathbb{Q}^{(l)} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u))} \mathbb{E}_{\mathbb{Q}^{(l)}} \Big[\lambda \|\rho - \rho^{(l)}\|_1 + H(u;\rho) \Big] = \inf_{\rho \in \mathcal{Z}(u)} \Big(\lambda \|\rho - \rho^{(l)}\|_1 + H(u;\rho) \Big).$$

The above claim can be clarified as the following

(a) Let $p^\star$ denote the value of the second term. Then, for any $\rho \in \mathcal{Z}(u)$, we have

$$p^\star \leq \lambda \|\rho - \rho^{(l)}\|_1 + H(u;\rho),$$

implying

$$p^\star \leq \mathbb{E}_{\mathbb{Q}^{(l)}} \Big[\lambda \|\rho - \rho^{(l)}\|_1 + H(u;\rho) \Big],$$

holds for any probability distribution $\mathbb{Q}^{(l)} \in \mathcal{M}(\mathcal{Z}(u))$, so does for $\mathbb{Q}^{(l)} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u))$. Therefore, we have

$$p^\star \leq \inf_{\mathbb{Q}^{(l)} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u))} \mathbb{E}_{\mathbb{Q}^{(l)}} \Big[\lambda \|\rho - \rho^{(l)}\|_1 + H(u;\rho) \Big].$$

(b) Next, we claim that the set $\mathcal{M}_{\text{lt}}(\mathcal{Z}(u))$ contains all the Dirac distributions supported on $\mathcal{Z}(u)$. Then, for any $\rho' \in \mathcal{Z}(u)$, we have

$$\delta_{\{\rho'\}} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u)),$$

resulting in

$$\inf_{\mathbb{Q}^{(l)} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u))} \mathbb{E}_{\mathbb{Q}^{(l)}} \left[\lambda \|\rho - \rho^{(l)}\|_1 + H(u;\rho) \right] \leq \lambda \|\rho' - \rho^{(l)}\|_1 + H(u;\rho'), \ \forall \rho' \in \mathcal{Z}(u),$$

which implies

$$\inf_{\mathbb{Q}^{(l)} \in \mathcal{M}_{\text{lt}}(\mathcal{Z}(u))} \mathbb{E}_{\mathbb{Q}^{(l)}} \left[\lambda \|\rho - \rho^{(l)}\| + H(u;\rho) \right] \leq p^\star.$$

Therefore, we equivalently write $J(u)$ as

$$J(u) = \sup_{\lambda \geq 0} -\lambda \epsilon(\beta) + \frac{1}{N} \sum_{l \in \mathcal{L}} \inf_{\rho \in \mathcal{Z}(u)} \left(\lambda \|\rho - \rho^{(l)}\|_1 + H(u;\rho) \right),$$

$$\text{s.t. (3.10).}$$

Finally, we write Problem (**P2**) as follows

$$\sup_{u, \lambda \geq 0} \ -\lambda \epsilon(\beta) + \frac{1}{N} \sum_{l \in \mathcal{L}} \inf_{\rho \in \mathcal{Z}(u)} \left(\lambda \|\rho - \rho^{(l)}\|_1 + H(u;\rho) \right), \tag{**P2′**}$$

$$\text{s.t. (3.7), (3.10).}$$

Step 2: (Equivalent reformulation of (P2′)) Using the definition of the dual norm and

moving its sup operator we can write Problem (**P2′**) as

$$\sup_{u,\lambda\geq 0} -\lambda\epsilon(\beta)+\frac{1}{N}\sum_{l\in\mathcal{L}}\inf_{\rho\in\mathcal{Z}(u)}\sup_{\|\nu^{(l)}\|_\star\leq\lambda}\Big(\Big\langle\nu^{(l)},\rho-\rho^{(l)}\Big\rangle+H(u;\rho)\Big),$$

$$\text{s.t. (3.7), (3.10).}$$

Given $\lambda\geq 0$, the sets $\{\nu^{(l)}\in\mathbb{R}^{nT}\mid\|\nu^{(l)}\|_\star\leq\lambda\}$ are compact for all $l\in\mathcal{L}$. We then apply the minmax theorem between inf and the second sup operators. This results in the switch of the operators, and by combining the two sup operators we have

$$\sup_{u,\lambda,\nu} -\lambda\epsilon(\beta)+\frac{1}{N}\sum_{l\in\mathcal{L}}\inf_{\rho\in\mathcal{Z}(u)}\Big(\Big\langle\nu^{(l)},\rho-\rho^{(l)}\Big\rangle+H(u;\rho)\Big),$$

$$\text{s.t. (3.7), (3.10), } \lambda\geq 0,$$

$$\|\nu^{(l)}\|_\star\leq\lambda,\ \forall l\in\mathcal{L}.$$

The objective function can be simplified as follows

$$-\lambda\epsilon(\beta)+\frac{1}{N}\sum_{l\in\mathcal{L}}\Big\langle-\nu^{(l)},\rho^{(l)}\Big\rangle+\frac{1}{N}\sum_{l\in\mathcal{L}}h^{(l)}(u),$$

where

$$h^{(l)}(u):=\inf_{\rho\in\mathcal{Z}(u)}\Big(\Big\langle\nu^{(l)},\rho\Big\rangle+H(u;\rho)\Big),\ \forall l\in\mathcal{L}.$$

For each $l\in\mathcal{L}$, we rewrite $h^{(l)}(u)$ by firstly taking a minus sign out of the inf operator, then exploiting the equivalent representation of sup operation, and finally using the definition of conjugate functions. The function $h^{(l)}(u)$ results in the following form

$$\begin{aligned}h^{(l)}(u)&=-\sup_{\rho\in\mathcal{Z}(u)}\Big(\Big\langle-\nu^{(l)},\rho\Big\rangle-H(u;\rho)\Big),\\&=-\sup_{\rho}\Big(\Big\langle-\nu^{(l)},\rho\Big\rangle-H(u;\rho)-\chi_{\mathcal{Z}(u)}(\rho)\Big),\\&=-\big[H(u;\cdot)+\chi_{\mathcal{Z}(u)}(\cdot)\big]^\star(-\nu^{(l)}).\end{aligned}$$

Further, we apply the property of the inf-convolution operation and push the minus sign back into the inf operator, for each $h^{(l)}(u)$, $l \in \mathcal{L}$. The representation of $h^{(l)}(u)$ results in the following relation

$$\begin{aligned} h^{(l)}(u) &= -\inf_{\mu} \left([H(u;\cdot)]^\star(-\mu^{(l)} - \nu^{(l)}) + \left[\chi_{\mathcal{Z}(u)}(\cdot)\right]^\star(\mu^{(l)}) \right), \\ &= \sup_{\mu} \left(-[H(u;\cdot)]^\star(-\mu^{(l)} - \nu^{(l)}) - \left[\chi_{\mathcal{Z}(u)}(\cdot)\right]^\star(\mu^{(l)}) \right). \end{aligned}$$

By substituting $-\nu^{(l)}$ by $\nu^{(l)}$, the resulting optimization problem has the following form

$$\begin{aligned} \sup_{u,\lambda,\mu,\nu} \; & -\lambda\epsilon(\beta) - \frac{1}{N}\sum_{l\in\mathcal{L}} \left([H(u;\cdot)]^\star(-\mu^{(l)} + \nu^{(l)}) + \left[\chi_{\mathcal{Z}(u)}(\cdot)\right]^\star(\mu^{(l)}) - \left\langle \nu^{(l)}, \rho^{(l)} \right\rangle \right), \\ & \text{s.t. (3.7), (3.10), } \lambda \geq 0, \\ & \|\nu^{(l)}\|_\star \leq \lambda, \; \forall l \in \mathcal{L}. \end{aligned}$$

Given u, the strong duality of linear programs is applicable for the conjugate of the function $H(u;\cdot)$ and the support function $\sigma_{\mathcal{Z}(u)}(\mu^{(l)})$. Using the strong duality and the definition of the support function, we compute, for each $l \in \mathcal{L}$, the following

$$[H(u;\cdot)]^\star(\nu^{(l)} - \mu^{(l)}) := \begin{cases} 0, & \nu^{(l)} = \mu^{(l)} + \frac{1}{T}u, \\ \infty, & \text{o.w.}, \end{cases}$$

and

$$\begin{aligned} \left[\chi_{\mathcal{Z}(u)}(\cdot)\right]^\star(\mu^{(l)}) &= \sigma_{\mathcal{Z}(u)}(\mu^{(l)}) \\ &= \begin{cases} \sup\limits_{\xi^{(l)}} & \left\langle \mu^{(l)}, \xi^{(l)} \right\rangle, \\ \text{s.t.} & 0 \leq \xi_e^{(l)}(t) \leq \rho_e^c(u_e(t)), \; \forall e \in \mathcal{E}, \; \forall t \in \mathcal{T}, \end{cases} \\ &= \begin{cases} \inf\limits_{\eta^{(l)}} & \sum\limits_{e\in\mathcal{E}, t\in\mathcal{T}} \overline{f}_e \overline{\rho}_e \eta_e^{(l)}(t), \\ \text{s.t.} & [\overline{f} \otimes \mathbf{1}_T + (\overline{\rho} - \rho^c(\overline{u})) \otimes \mathbf{1}_T \circ u] \circ \eta^{(l)} - \mu^{(l)} \geq \mathbf{0}_{nT}, \\ & \eta^{(l)} \geq \mathbf{0}_{nT}, \end{cases} \end{aligned}$$

We substitute these parts for that in the objective function, take the above inf operator out of a minus sign, and obtain (**P3**).

Given that all the reformulations in this step hold with equalities, we therefore claim that the above problem is equivalent to (**P2**). Finally, we claim that the sup operation is indeed achievable, because 1) each component of the variable u is in a finite set Γ and 2), for any u that is feasible to the above problem, the above problem with that fixed u satisfies the Slater's condition, which implies that the above problem is achievable. We therefore claim (**P2**) is equivalent to (**P3**).

Step 3: (Performance guarantees of (P3)) Given any feasible point $(u, \rho, \lambda, \mu, \nu, \eta)$ of (**P3**), we denote its objective value by $\hat{J}(u)$. The value $\hat{J}(u)$ is a lower bound of (**P3**) and therefore a lower bound for (**P2**), i.e., $\hat{J}(u) \leq J(u)$. Thus $\hat{J}(u)$ is an estimate of the certificate for the performance guarantee (3.9). Therefore, $(u,\ \hat{J}(u))$ is a data-driven solution and certificate pair for (**P1**). □

Remark 13 (Formulation (P3) depends on data). Problem (**P3**) is parameterized by the variables $(\overline{f}, \overline{\rho}, \overline{u})$, which are related to the highway infrastructure and random events, and the data $\{\varpi^{(l)}\}_{l\in\mathcal{L}}$, which are related to the traffic system initial state, $\{\rho^{(l)}(0)\}_{l\in\mathcal{L}}$, on-ramp, and off-ramp flows. The decision variables include future states of the density $\rho^{(l)}(t)$, $t > 0$, the speed limits, and other multipliers to make the constraints hold. In particular, note that the ambiguity-ball constraint is enforced via the λ multiplier and maximization in ρ. In practice, highway random events such as accidents are monitored in real time, which provides particular information about $(\overline{f}, \overline{\rho}, \overline{u})$. On the other hand, data $\{\varpi^{(l)}\}_{l\in\mathcal{L}}$ is accessible from various independent monitors as mentioned in the previous remark on data-driven control implementation.

Problem (**P3**) is inherently difficult to solve due to the discrete decision variables u, bi-linear terms $u \circ \eta^{(l)}$ in the first group of constraints, and the nonlinear admissible sample trajectories $\{\rho^{(l)}\}_{l\in\mathcal{L}}$, which motivates our next section.

3.4 Equivalent Reformulation

Our goal is to compute exact solutions to Problem (**P3**). To achieve this goal, we focus on the feasibility set of Problem (**P3**), and transform a group of its non-convex quadratic terms, which are comprised of a continuous variable and a binary variable, into a set of mixed-integer linear constraints. We call the new equivalent formulation, Problem (**P4**).

Binary representation of speed limits: Let $O := \{1, \ldots, m\}$ be the index set of the speed limit set Γ. For each edge $e \in \mathcal{E}$, time slot $t \in \mathcal{T}$ and speed limit value $\gamma^{(i)} \in \Gamma$, let us define the binary variable $x_{e,i}(t)$ to be equal to one if $u_e(t) = \gamma^{(i)}$; otherwise $x_{e,i}(t) = 0$. We will then have $u_e(t) = \sum_{i \in O} \gamma^{(i)} x_{e,i}(t)$ for each edge $e \in \mathcal{E}$. Using this representation, we can reformulate the speed limit constraints (3.7) as follows

$$\begin{aligned}
&\gamma^{(1)} \leq \sum_{i \in O} \gamma^{(i)} x_{e,i}(t) \leq \gamma^{(m)}, \ \forall e \in \mathcal{E}, \ \forall t \in \mathcal{T}, \\
&\sum_{i \in O} x_{e,i}(t) = 1, \forall e \in \mathcal{E}, \ \forall t \in \mathcal{T}, \\
&x_{e,i}(t) \in \{0, \ 1\}, \ \forall e \in \mathcal{E}, \ i \in O, \ t \in \mathcal{T},
\end{aligned} \tag{3.11}$$

and we update the admissible sample trajectories formula (3.10) for all $t \in \mathcal{T}$ and $l \in \mathcal{L}$ as follows

$$\begin{aligned}
&\rho_e^{(l)}(t+1) = h_e \frac{1 - r_s^{\mathrm{o},(l)}(t)}{1 - r_e^{\mathrm{in},(l)}(t)} \sum_{i \in O} \gamma^{(i)} x_{s,i}(t) \rho_s^{(l)}(t) + \rho_e^{(l)}(t) - h_e \sum_{i \in O} \gamma^{(i)} x_{e,i}(t) \rho_e^{(l)}(t), \ \forall e \in \mathcal{E} \setminus \{(0,1)\}, \\
&\rho_e^{(l)}(t+1) = \rho_e^{(l)}(t) + h_e \omega^{(l)}(t) - h_e \sum_{i \in O} \gamma^{(i)} x_{e,i}(t) \rho_e^{(l)}(t), \quad e = (0,1). \\
&\frac{1 - r_s^{\mathrm{o},(l)}(t)}{1 - r_e^{\mathrm{in},(l)}(t)} \sum_{i \in O} \gamma^{(i)} x_{s,i}(t) \rho_s^{(l)}(t) \leq \min \left\{ \overline{f}_e, \ \tau_e \overline{u}_e \left(\overline{\rho}_e - \rho_e^{(l)}(t) \right) \right\}, \quad \forall e \in \mathcal{E} \setminus \{(0,1)\},
\end{aligned} \tag{3.12}$$

We are particularly interested in two groups of bi-linear terms: 1) the bi-linear terms $x_{e,i}(t)\rho_e^{(l)}(t)$ in the admissible sample trajectories formula (3.12) and 2) the bi-linear terms $\sum_{i \in O} \gamma^{(i)} x_{e,i}(t) \eta_e^{(l)}(t)$ which appear in the first set of constraints, e.g., $u \circ \eta^{(l)}$. Each of these bi-linear terms is comprised of a continuous variable and a binary variable. We represent each of

these bi-linear terms with a set of linear constraints using the following linearization technique.

Let us introduce variables $y_{e,i}^{(l)}(t)$ and $z_{e,i}^{(l)}(t)$ for all $e \in \mathcal{E}$, $i \in \mathcal{O}$, $t \in \mathcal{T}$ and $l \in \mathcal{L}$ as follows

$$\begin{aligned} y_{e,i}^{(l)}(t) &= x_{e,i}(t)\rho_e^{(l)}(t), \\ z_{e,i}^{(l)}(t) &= x_{e,i}(t)\eta_e^{(l)}(t). \end{aligned} \tag{3.13}$$

We further make the following assumption

Assumption 7 (Bounded dual variable η). *There exists a positive constant $\overline{\eta}$ such that for any optimizers of* (**P3**)*, the components $\eta_e^{(l)}(t) \leq \overline{\eta}$ for all $e \in \mathcal{E}$, $t \in \mathcal{T}$ and $l \in \mathcal{L}$.*

We achieve Assumption 7 by selecting $\overline{\eta}$ large enough. This enables the following lemma to represent the non-convex equality constraint in (3.13) with a set of linear constraints.

Lemma 10 (Linearization technique). *Let Assumption 7 hold. Then for all $e \in \mathcal{E}$, $i \in \mathcal{O}$, $t \in \mathcal{T}$ and $l \in \mathcal{L}$, the non-convex equality constraint in* (3.13) *can be equivalently represented with the following set of linear constraints*

$$\begin{aligned} &0 \leq z_{e,i}^{(l)}(t) \leq \overline{\eta} x_{e,i}(t), \\ &\eta_e^{(l)}(t) - \overline{\eta}(1 - x_{e,i}(t)) \leq z_{e,i}^{(l)}(t) \leq \eta_e^{(l)}(t), \end{aligned} \tag{3.14}$$

$$\begin{aligned} &0 \leq y_{e,i}^{(l)}(t) \leq \overline{\rho}_e x_{e,i}(t), \\ &\rho_e^{(l)}(t) - \overline{\rho}_e(1 - x_{e,i}(t)) \leq y_{e,i}^{(l)}(t) \leq \rho_e^{(l)}(t). \end{aligned} \tag{3.15}$$

Proof. The proof follows by the application of the following proposition on each bi-linear term in (3.13):

Proposition 1 (Equivalent reformulation of bi-linear terms [42, Section 2]). *Let $\mathcal{Y} \subset \mathbb{R}$ be a compact set. Given a binary variable x and a linear function $g(y)$ in a continuous variable*

$y \in \mathcal{Y}$, z equals the quadratic function $xg(y)$ if and only if

$$\begin{aligned}
&\underline{g}x \le z \le \overline{g}x,\\
&g(y) - \overline{g}\cdot(1-x) \le z \le g(y) - \underline{g}\cdot(1-x),
\end{aligned}$$

where $\underline{g} = \min_{y\in\mathcal{Y}}\{g(y)\}$ *and* $\overline{g} = \max_{y\in\mathcal{Y}}\{g(y)\}$. □

Remark 14 (Regularization technique). In a later program, we add the following extra constraints to speed up the internal computation of solvers

$$\begin{aligned}
&\textstyle\sum_{i\in\mathcal{O}} z_{e,i}^{(l)}(t) = \eta_e^{(l)}(t), \quad \forall e \in \mathcal{E},\ t \in \mathcal{T}\setminus\{0\},\ l \in \mathcal{L},\\
&\textstyle\sum_{i\in\mathcal{O}} y_{e,i}^{(l)}(t) = \rho_e^{(l)}(t), \quad \forall e \in \mathcal{E},\ t \in \mathcal{T}\setminus\{0\},\ l \in \mathcal{L}.
\end{aligned}$$

These are adapted from the binary representation (3.11) and the definition of $y_{e,i}^{(l)}(t)$ and $z_{e,i}^{(l)}(t)$.

□

In particular, we let $y_{e,i}^{(l)}(0) = x_{e,i}(0)\rho_e^{(l)}(0)$ for each $e \in \mathcal{E}$, $i \in \mathcal{O}$ and $l \in \mathcal{L}$. Then using the new variables $y_{e,i}^{(l)}(t)$ and $z_{e,i}^{(l)}(t)$, we can now reformulate the admissible sample trajectories formula (3.12) as follows

$$\begin{aligned}
&\rho_e^{(l)}(t+1) = h_e\frac{1-r_s^{\mathrm{o},(l)}(t)}{1-r_e^{\mathrm{in},(l)}(t)}\sum_{i\in\mathcal{O}}\gamma^{(i)}y_{s,i}^{(l)}(t) + \rho_e^{(l)}(t) - h_e\sum_{i\in\mathcal{O}}\gamma^{(i)}y_{e,i}^{(l)}(t), \quad \forall e \in \mathcal{E}\setminus\{(0,1)\},\\
&\rho_e^{(l)}(t+1) = \rho_e^{(l)}(t) + h_e\omega^{(l)}(t) - h_e\sum_{i\in\mathcal{O}}\gamma^{(i)}y_{e,i}^{(l)}(t), \quad e = (0,1).\\
&\frac{1-r_s^{\mathrm{o},(l)}(t)}{1-r_e^{\mathrm{in},(l)}(t)}\sum_{i\in\mathcal{O}}\gamma^{(i)}y_{s,i}^{(l)}(t) \le \min\left\{\overline{f}_e,\ \tau_e\overline{u}_e\left(\overline{\rho}_e - \rho_e^{(l)}(t)\right)\right\},\ \forall e \in \mathcal{E}\setminus\{(0,1)\},
\end{aligned} \tag{3.16}$$

Problem (**P3**) can now be equivalently reformulated as the following optimization problem

$$\max_{\substack{x,y,z,\rho,\\ \lambda,\mu,\nu,\eta}} \quad -\lambda\epsilon(\beta) - \frac{1}{N}\sum_{e,t,l} \overline{f}_e \overline{\rho}_e \eta_e^{(l)}(t) + \frac{1}{N}\sum_{e,t,l} \nu_e^{(l)}(t)\rho_e^{(l)}(t), \tag{**P4**}$$

$$\text{s.t.} \quad \sum_{i\in O} \gamma^{(i)}(\overline{\rho} - \rho^c(\overline{u})) \otimes \mathbf{1}_T \circ z_i^{(l)} - \mu^{(l)} + \overline{f} \otimes \mathbf{1}_T \circ \eta^{(l)} \geq \mathbf{0}_{nT},\ \forall\ l \in \mathcal{L}, \tag{3.17}$$

$$\nu^{(l)} = \mu^{(l)} + \frac{1}{T}\sum_{i\in O} \gamma^{(i)} x_i,\ \forall\ l \in \mathcal{L}, \tag{3.18}$$

$$\|\nu^{(l)}\|_\star \leq \lambda,\ \forall\ l \in \mathcal{L}, \tag{3.19}$$

$$\mathbf{0}_{nT} \leq \eta^{(l)} \leq \overline{\eta},\ \forall\ l \in \mathcal{L}, \tag{3.20}$$

speed limits (3.11), **dual variable** (3.14),

sample trajectories{(3.15), (3.16)}.

Remark 15 (Performance guarantee (3.9) **in the setting of Problem** (**P4**)**).** Let $\hat{J}(u)$ denote the value of the objective function of (**P4**) at a computed feasible solution $(x, y, z, \rho, \lambda, \mu, \nu, \eta)$. Then, the resulting speed limits $u := \sum_{i\in O} \gamma^{(i)} x_i$ provide a data-driven solution such that $(u,\ \hat{J}(u))$ satisfies the performance guarantee (3.9) of (**P1**). This result exploits the fact that Problem (**P4**) is equivalent to (**P3**) and (**P3**) is equivalent to (**P2**) as in Theorem 8.

3.5 Computationally Efficient Algorithms

We propose a decomposition-based, integer-solution search algorithm which computes online-tractable, high-quality feasible solutions to (**P4**) with performance guarantees. Similar algorithms have been proposed in the literature [69, 72]. Such methods allow us to compute sub-optimal solutions to mix-integer nonlinear programs efficiently. The proposed integer-solution search algorithm is shown in Algorithm 5. This algorithm iteratively computes sub-optimal

Algorithm 5. Integer solution search algorithm

Initialize $k = 0$
repeat
  $k \leftarrow k + 1$
  Solve Problem (UBP_k), **return** $x^{(k)}$ and UB_k
  Generate admissible sample trajectories $\{\rho^{(l,k)}\}_{l\in\mathcal{L}}$
  Solve Problem (LBP_k), **return** obj_k and LB_k
until $\text{UB}_k - \text{LB}_k \leq \epsilon$, or ($\text{UBP}_k$) is infeasible, or a satisfactory sub-optimal solution is found after certain running time T_{run}
return data driven solution $u_{\text{best}} := u^{(q)}$ with certificate $\hat{J}(u^{(q)})$ such that $q \in \text{argmax}_{p=1,\ldots,k}\{\text{obj}_p\}$

solutions to (**P4**) until a stopping criteria is met. At each iteration, the algorithm solves an upper-bounding problem to (**P4**), and then solves a lower-bounding problem to (**P4**). The upper-bounding Problem (UBP_k) is obtained through McCormick relaxations of the bi-linear terms $\{v^{(l)} \circ \rho^{(l)}\}_{l\in\mathcal{L}}$. This upper bounding problem is a mixed-integer linear program and its solution provides us with an upper bound on Problem (**P4**) and a candidate speed limits $x^{(k)}$. Notice that $x^{(k)}$ respects the sustainability constraints in (3.16). We then use the computed speed limit $x^{(k)}$ to construct a set of admissible sample trajectories $\{\rho^{(l,k)}\}_{l\in\mathcal{L}}$ and equivalently reduce Problem (**P4**) to a linear lower-bounding Problem (LBP_k) for potential feasible solutions of (**P4**). If (LBP_k) is feasible, then the candidate speed limits together with the objective value of (LBP_k) provide guarantee (3.9) for (**P4**). Next, we present the upper-bounding and lower-bounding problems in detail.

3.5.1 Upper-bounding Problem

Problem (UBP_k) is constructed in two stages:

Stage 1: We use a standard McCormick relaxation to handle the non-convex quadratic terms $\{v_e^{(l)}(t)\rho_e^{(l)}(t)\}_{e\in\mathcal{E},t\in\mathcal{T},l\in\mathcal{L}}$ in the objective function of (**P4**). Notice that the McCormick envelope [85] provides relaxations of bi-linear terms, which is stated in the following remark.

Remark 16 (McCormick envelope). Consider two variables $x,\ y\ \in \mathbb{R}$ with upper and lower bounds, $\underline{x} \leq x \leq \overline{x}$, $\underline{y} \leq y \leq \overline{y}$. The McCormick envelope of the variable $s := xy \in \mathbb{R}$ is

characterized by the following constraints

$$s \geq \overline{x}y + x\overline{y} - \overline{x}\overline{y}, \quad s \geq \underline{x}y + x\underline{y} - \underline{x}\underline{y},$$
$$s \leq \overline{x}y + x\underline{y} - \overline{x}\underline{y}, \quad s \leq \underline{x}y + x\overline{y} - \underline{x}\overline{y}.$$

To construct a McCormick envelope for (UBP$_k$), let us denote $\overline{v}_e := \overline{u}_e \left(T^{-1} + \overline{\rho}_e \overline{\eta}\right)$ for each edge $e \in \mathcal{E}$. We have $0 \leq v_e^{(l)}(t) \leq \overline{v}_e$, $0 \leq \rho_e^{(l)}(t) \leq \overline{\rho}_e$ for all $e \in \mathcal{E}$, $t \in \mathcal{T}$, and $l \in \mathcal{L}$. Therefore, the McCormick envelope of $s_e^{(l)}(t) := v_e^{(l)}(t)\rho_e^{(l)}(t)$ is given by

$$\begin{aligned}
&s_e^{(l)}(t) \geq \overline{v}_e \rho_e^{(l)}(t) + v_e^{(l)}(t)\overline{\rho}_e - \overline{v}_e\overline{\rho}_e, \\
&s_e^{(l)}(t) \geq 0, \\
&s_e^{(l)}(t) \leq \overline{v}_e \rho_e^{(l)}(t), \\
&s_e^{(l)}(t) \leq v_e^{(l)}(t)\overline{\rho}_e.
\end{aligned} \tag{3.21}$$

Stage 2: We identify appropriate canonical integer cuts to prevent (UBP$_k$) from choosing examined candidate variable speed limits $\{x^{(p)}\}_{p=1}^{k-1}$. Let $\Omega^{(p)} := \{(e,i,t) \in \mathcal{E} \times O \times \mathcal{T} \mid x_{e,i}^{(p)}(t) = 1\}$ denote the index set of x for which the value $x_{e,i}^{(p)}(t)$ is 1 at the previous iteration p. In addition, let $c^{(p)} := |\Omega^{(p)}|$ and $\overline{\Omega}^{(p)} := (\mathcal{E} \times O \times \mathcal{T}) \setminus \Omega^{(p)}$ denote the cardinality of the set $\Omega^{(p)}$ and the complement of $\Omega^{(p)}$, respectively. Therefore, the canonical integer cuts of Problem (UBP$_k$) at iteration k are given by

$$\sum_{(e,i,t)\in\Omega^{(p)}} x_{e,i}(t) - \sum_{(e,i,t)\in\overline{\Omega}^{(p)}} x_{e,i}(t) \leq c^{(p)} - 1, \quad \forall p \in \{1,\ldots,k-1\}. \tag{3.22}$$

Upper-bounding Problem (UBP_k) can be formulated as follows

$$\max_{\substack{x,y,z,s,\rho,\\ \lambda,\mu,\nu,\eta}} -\lambda\epsilon(\beta) - \frac{1}{N}\sum_{e,t,l}\left(\overline{f}_e\overline{\rho}_e\eta_e^{(l)}(t) - s_e^{(l)}(t)\right), \tag{UBP_k}$$

$$\text{s.t.} \quad \textbf{speed limits } (3.11),\ \textbf{sample trajectories } \{(3.15),\ (3.16)\},$$

$$\textbf{no congestion } \{(3.14),\ (3.17),\ (3.18),\ (3.19),\ (3.20)\},$$

$$\textbf{McCormick envelope } (3.21),\ \textbf{integer cuts } (3.22).$$

Let UB_k denote the optimal objective value of (UBP_k), and let $x^{(k)}$ denote the integer part of the optimizers of (UBP_k). Then UB_k is an upper bound of the original non-convex Problem (**P4**). We use $x^{(k)}$ as a candidate speed limit in the lower-bounding problem LBP_k.

3.5.2 Lower-bounding Problem

Problem (**P4**) can be equivalently written as

$$\max_{\substack{x,y,z,\rho,\\ \lambda,\mu,\nu,\eta}} -\lambda\epsilon(\beta) - \frac{1}{N}\sum_{e,t,l}\left(\overline{f}_e\overline{\rho}_e\eta_e^{(l)}(t) - \nu_e^{(l)}(t)\rho_e^{(l)}(t)\right),$$

$$\text{s.t.} \quad (z,\lambda,\mu,\nu,\eta) \in \Phi(x),\ (y,\rho) \in \Psi(x),\ x \in X.$$

where

$$\Phi(x) := \{(z,\lambda,\mu,\nu,\eta) \mid \textbf{no congestion}\},$$

$$\Psi(x) := \{(y,\rho) \mid \textbf{sample trajectories}\},$$

$$X := \{x \mid \textbf{speed limits}\}.$$

The solution $x^{(k)}$ to (UBP_k) at iteration k provides us with a candidate speed limit $u^{(k)} := \sum_{i\in O}\gamma^{(i)}x_i^{(k)}$ which respect the sustainability constraints. For each $l \in \mathcal{L}$ with the given $u^{(k)}$, the admissible sample trajectory $\rho^{(l)}$ is uniquely determined by $(\omega^{(l)}, \rho^{(l)}(0), r^{\text{in},(l)}, r^{\text{out},(l)})$ using the

uniqueness solution of the linear time-invariant systems. Therefore, the element $(y, \rho) \in \Psi(x^{(k)})$ is unique. Using the constraints set $\Psi(x^{(k)})$, we then construct the unique admissible sample trajectories $\{\rho^{(l,k)}\}_{l\in\mathcal{L}}$. The unique admissible sample trajectories allow us to reduce (**P4**) to the following lower bounding problem[8]

$$\max_{z,\lambda,\mu,\nu,\eta} -\lambda\epsilon(\beta) - \frac{1}{N}\sum_{e,t,l}\left(\overline{f}_e\overline{\rho}_e\eta_e^{(l)}(t) - \nu_e^{(l)}(t)\rho_e^{(l,k)}(t)\right), \tag{LBP$_k$}$$

$$\text{s.t.} \quad (z, \lambda, \mu, \nu, \eta) \in \Phi(x^{(k)}).$$

Note that (LBP$_k$) is a linear program and much easier to solve than the non-convex Problem (**P4**). Let obj$_k$ denote the optimal objective value of (LBP$_k$). If Problem (LBP$_k$) is solved to optimum with a finite obj$_k$, we will then obtain a feasible solution of (**P4**) with speed limit $u^{(k)} := \sum_{i\in\mathcal{O}}\gamma^{(i)}x_i^{(k)}$ and certificate $\hat{J}(u^{(k)}) := \text{obj}_k$; otherwise, Problem (LBP$_k$) is either infeasible or unbounded, i.e., obj$_k = -\infty$. The lower bound of (**P4**) can be calculated by LB$_k = \max_{p=1,\ldots,k}\{\text{obj}_p\}$. The stopping criterion of the algorithm can be determined by one of the following criteria

1. UB$_k$ − LB$_k \le \epsilon$,
2. (UBP$_k$) is infeasible,
3. A satisfactory sub-optimal solution is found after certain running time T_{run}.

In [72], it is shown that such algorithms convergence to a global ϵ-optimal solution after finite number of iterations when we use the first and second stopping criteria. The third solution

[8]Given $u^{(k)}$ and $\{\rho^{(l,k)}\}_{l\in\mathcal{L}}$, the equivalent dual of (LBP$_k$) is

$$\begin{aligned}\min_{\rho^{(l)},l\in\mathcal{L}} \ & \frac{1}{NT}\sum_{e,t,l}u_e^{(k)}(t)\rho_e^{(l)}(t)\\ \text{s.t.} \ & 0 \le \rho^{(l)} \le \rho^c(u^{(k)}),\ \forall l \in \mathcal{L},\\ & \sum_{l\in\mathcal{L}}\|\rho^{(l)} - \rho^{(l,k)}\|_1 \le \epsilon(\beta),\end{aligned}$$

which results in more efficient online solutions.

criterion allow us to find a potentially good performance-guaranteed feasible solution within certain running time T_{run}.

Remark 17 (Online tractable solutions to (P4)). The data-driven control requires to solve a sequence of (**P4**) online. We achieve this by an online "warm start" of Algorithm 5, which employs an assimilation set $\mathcal{I}_t := \{u^{(s)}\}_s$ that contains the historically-generated speed-limit candidates, where s indexes these candidates. In particular, at each time solving a problem (**P4**), the candidates in $\mathcal{I}_t$ can be explored by the solution to (LBP$_k$) of each $u^{(s)} \in \mathcal{I}_t$. Notice that these (LBP$_k$) can be executed in parallel. Then, these examined candidates contribute to integer cuts in (UBP$_k$) when executing Algorithm 5. At the termination of the current (**P4**), a new set of candidates are updated to $\mathcal{I}_{t+1}$ for later evaluation of (**P4**).

3.6 Analysis via Second-order Cone Problems

This section provides a tool to analyze the efficacy of the proposed algorithm for the nonconvex Problem (**P4**). In particular, we propose a second-order cone relaxation for the non-convex quadratic terms $\{v^{(l)} \circ \rho^{(l)}\}_{l\in\mathcal{L}}$ in (**P4**), and a second-order cone relaxation for it. We then present the conditions under which this convex relaxation is exact. To enable the tool for analysis, we assume the following:

Assumption 8 (Highway densities are nontrivial). *For all $e \in \mathcal{E}$, $t \in \mathcal{T}$ and $l \in \mathcal{L}$, we assume $\rho_e^{(l)}(t) \geq \epsilon(\beta)$, where the parameter $\epsilon(\beta)$ is the radius of the Wasserstein ball.*

Remark 18 (On nontrivial highway densities). Assumption 8 depends on the radius of the Wasserstein ball, which is selected as in Remark 12. In reality, we could select the value $\epsilon(\beta)$ to be sufficiently small, even if it potentially sacrifices confidence on performance guarantees. In any case, there are three cases to consider: 1) the density on each segment of highway is just zero, 2) there are zero density values $\rho_e^{(l)}(t)$, for some (e,t,l), while the rest of $\rho_e^{(l)}(t)$ are upper bounded by a value that is smaller than the maximal critical density $\max_{u_e(t)\in\Gamma} \rho_e^c(u_e(t))$, and 3) there are some

values $\rho_e^{(l)}(t)$ that go beyond the maximal critical density, e.g., $\rho_e^{(l)}(t) > \epsilon(\beta) + \text{argmax}_{u\in\Gamma}\, \rho_e^c(u)$. In the first case, no congestion would happen and there is no need for speed-limit control. The second case can be handled by tuning $\epsilon(\beta)$ to be small enough. In the third case, with a given small $\epsilon(\beta)$, there is already congestion on some segment of the highway and no feasible speed limit would eliminate that congestion.

Assumption 8 enables us to explore properties of the optimizers of (**P4**) as in the following

Proposition 2 (Optimizers in a cone). *Let* $\text{sol}^\star := (x^\star, y^\star, z^\star, \rho^\star, \lambda^\star, \mu^\star, \nu^\star, \eta^\star)$ *be any optimizer of Problem* (**P4**)*. If Assumption 8 holds, we have* $\nu^\star \circ \rho^\star \geq \mathbf{0}_{nTN}$.

Proof. Knowing that $\rho^\star \geq \mathbf{0}_{nTN}$ by constraints (3.15), we only need to show $\nu^\star \geq \mathbf{0}_{nTN}$. To prove this, let us assume there exists an optimizer $\text{sol}^\star$ such that, for at least one $\varepsilon \in \mathcal{E}$, $\tau \in \mathcal{T}$ and $\ell \in \mathcal{L}$, it holds $\nu_\varepsilon^{(\ell),\star}(\tau) < 0$. Then, using constraint (3.18), we claim $\mu_\varepsilon^{(\ell),\star}(\tau) < 0$. Next, we show the contradiction to an optimizer by constructing a feasible solution that gives us higher objective value than that resulted from $\text{sol}^\star$. To achieve this, we perturb variables $\lambda^\star$, $\mu_\varepsilon^{(\ell),\star}(\tau)$ and $\nu_\varepsilon^{(\ell),\star}(\tau)$, and leave other components the same as that in $\text{sol}^\star$. With such perturbation, only constraints (3.17), (3.18), and (3.19) are varied.

Let $\text{sol} := (x^\star, y^\star, z^\star, \rho^\star, \hat{\lambda}, \hat{\mu}, \hat{\nu}, \eta^\star)$ denote the feasible solution we are to construct. We denote by $\hbar^\star := \sum_{i\in O} \gamma^{(i)}(\overline{\rho} - \rho^c(\overline{u})) \otimes \mathbf{1}_T \circ z_i^{(l),\star} + \overline{f} \otimes \mathbf{1}_T \circ \eta^{(l),\star}$ the unperturbed part in constraint (3.17) and construct $\hat{\mu}$ as follows

$$\hat{\mu}_e^{(l)}(t) = \begin{cases} \mu_e^{(l),\star}(t), & \text{if } (e,t,l) \neq (\varepsilon,\tau,\ell), \\ \min\{\hbar_\varepsilon^{(\ell),\star}(\tau),\ -\mu_\varepsilon^{(\ell),\star}(\tau)\}, & \text{o.w.} \end{cases}$$

The above construction ensures the feasibility of constraints (3.17) and furthermore, because $\hbar_\varepsilon^{(\ell),\star}(\tau) \geq 0$ and $\mu_\varepsilon^{(\ell),\star}(\tau) < 0$, we claim $\hat{\mu}_\varepsilon^{(\ell)}(\tau) \geq 0$. Then let us denote by $g^\star := \frac{1}{T}\sum_{i\in O} \gamma^{(i)} x_i^\star$

the unperturbed part of constraints (3.18) and construct variable $\hat{v}$ as follows

$$\hat{v}_e^{(l)}(t) = \begin{cases} v_e^{(l),\star}(t), & \text{if } (e,t,l) \neq (\varepsilon,\tau,\ell), \\ \hat{\mu}_\varepsilon^{(\ell)}(\tau) + g_\varepsilon^\star(\tau), & \text{o.w.} \end{cases}$$

Again, we have $\hat{v}_\varepsilon^{(\ell)}(\tau) \geq 0$. Then by letting $\hat{\lambda} := \max\{\lambda^\star,\ \hat{v}_\varepsilon^{(\ell)}(\tau)\}$, constraints (3.19) are satisfied. In this way, a feasible solution sol is constructed.

Next, we evaluate the difference of the objective values of (**P4**) resulting from sol and sol* in the following

$$\begin{aligned} &\text{objective(sol)} - \text{objective(sol}^\star\text{)} \\ &= \left(-\hat{\lambda} + \lambda^\star\right)\epsilon(\beta) + \left(\hat{v}_\varepsilon^{(\ell)}(\tau) - v_\varepsilon^{(\ell),\star}(\tau)\right)\rho_\varepsilon^{(\ell),\star}(\tau), \\ &\geq \left(-\hat{\lambda} + \lambda^\star + \hat{v}_\varepsilon^{(\ell)}(\tau) - v_\varepsilon^{(\ell),\star}(\tau)\right)\epsilon(\beta), \\ &= \left(\min\{-\lambda^\star,\ -\hat{v}_\varepsilon^{(\ell)}(\tau)\} + \lambda^\star + \hat{v}_\varepsilon^{(\ell)}(\tau)\right)\epsilon(\beta) - v_\varepsilon^{(\ell),\star}(\tau)\epsilon(\beta), \\ &> 0, \end{aligned}$$

where the first equality cancels out unperturbed terms; the second inequality applies Assumption 8 and the fact that $\hat{v}_\varepsilon^{(\ell)}(\tau) \geq 0$ and $v_\varepsilon^{(\ell),\star}(\tau) < 0$; the third equality applies construction of $\hat{\lambda}$; and the last one is achieved by summing the nonnegative first term and the strict positive second term.

By the above computation, we constructed a feasible solution sol with a higher objective value than that of sol*, contradicting the assumption that sol* is an optimizer. □

Proposition 2 allows us to explore structure of the bi-linear terms $\{v^{(l)} \circ \rho^{(l)}\}_{l\in\mathcal{L}}$ via second-order cone constraints. This is achieved by introducing variables $\vartheta_e^{(l)}(t)$ and writing Problem (**P4**) as follows

$$\max_{\substack{x,y,z,\rho,\\ \lambda,\mu,\nu,\eta,\vartheta}} -\lambda\epsilon(\beta) - \frac{1}{N}\sum_{e,t,l}\overline{f}_e\overline{\rho}_e\eta_e^{(l)}(t) + \frac{1}{N}\sum_{e,t,l}\left(\vartheta_e^{(l)}(t)\right)^2, \qquad (\mathbf{P4'})$$

$$\text{s.t.} \quad \vartheta^2 \leq \nu \circ \rho, \tag{3.23}$$

speed limits (3.11), **sample trajectories** {(3.15), (3.16)},

no congestion {(3.14), (3.17), (3.18), (3.19), (3.20)}.

For each $e \in \mathcal{E}$, $t \in \mathcal{T}$ and $l \in \mathcal{L}$, the constraint (3.23) can be equivalently written as the following second-order cone:

$$\sqrt{\left(\nu_e^{(l)}(t)\right)^2 + \left(\rho_e^{(l)}(t)\right)^2 + 2\left(\vartheta_e^{(l)}(t)\right)^2} \leq \nu_e^{(l)}(t) + \rho_e^{(l)}(t). \tag{3.24}$$

Problem (**P4**) and (**P4′**) are equivalent as the following:

Lemma 11 (Equivalent optimizer sets of (P4) and (P4′)). *Problem* (**P4′**) *is equivalent to* (**P4**) *in the sense that their optimal objective values are the same and the set of optimizer of* (**P4**) *is the projection of that of* (**P4′**). *Further, any feasible solution of* (**P4′**) *can give us a valid performance guarantee* (3.9) *with certificate to be objective function of* (**P4′**) *evaluated at that feasible solution.*

Proof. Let us denote by (P4″) the Problem (**P4**) with an extra set of constraints $\nu \geq \mathbf{0}_{nTN}$. We prove the lemma in two steps.

Step 1: (Equivalence of optimizers sets) First, we use Proposition 2 to claim that the set of optimizers of (**P4**) is the same as that of (P4″). Second, we claim that for any optimizer of (**P4′**), all the constraints in (3.23) are active. This means that the set of optimizers of (P4″) are the same as the projection of that of (**P4′**). Therefore, the optimizers set of (**P4**) and (**P4′**) are equivalent.

Step 2: (Performance guarantees) First, any feasible solution of (**P4′**) correspond to a feasible solution of (**P4**). This holds because any feasible solution of (**P4′**) satisfies all the constraints of (**P4**). Next, the objective value of (**P4′**) gives a lower bounds of that of (**P4**). This

can be verified using constraints (3.23). Finally, the performance guarantees (3.9) of feasible solution of (**P4′**) can be derived from that of (**P4**) as in Remark 15. □

Problem (**P4′**) is still non-convex. To approximate the quadratic terms in the objective, we will first show that variables $\vartheta_e^{(l)}(t)$ are bounded.

Lemma 12 (Bounded variable ϑ). *Let Assumption 7 on bounded η hold, then there exists large enough scalar $\overline{\vartheta}$ such that $|\vartheta_e^{(l)}(t)| \leq \overline{\vartheta}$ for all $e \in \mathcal{E}$, $t \in \mathcal{T}$ and $l \in \mathcal{L}$.*

Proof. We construct $\overline{\vartheta}$ by showing boundedness of $\nu \circ \rho$. It's known for each $e \in \mathcal{E}$, $t \in \mathcal{T}$ and $l \in \mathcal{L}$, the density $\rho_e^{(l)}(t)$ is nonnegative and bounded above by $\max_{e\in\mathcal{E}}\{\overline{\rho}_e\}$. Then we only need to find the upper bound of $\nu_e^{(l)}(t)$. By Assumption 7, the variable $\eta_e^{(l)}(t)$ is bounded. Then computations via constraints (3.17) and (3.18) result in upper bound of $\nu_e^{(l)}(t)$ as the following

$$\begin{aligned}\nu_e^{(l)}(t) &\leq \max_{e\in\mathcal{E},u_e(t)\in\Gamma}\left\{\left(\overline{f}_e + u_e(t)(\overline{\rho}_e - \rho_e^c(\overline{u}_e))\right)\overline{\eta} + \frac{1}{T}u_e(t)\right\},\\ &= \max_{e\in\mathcal{E}}\left\{\overline{u}_e\overline{\rho}_e\overline{\eta} + \frac{1}{T}\overline{u}_e\right\}.\end{aligned}$$

By letting $\overline{\vartheta} = \sqrt{\max_{e\in\mathcal{E}}\left\{\overline{u}_e\overline{\rho}_e^2\overline{\eta} + \frac{1}{T}\overline{u}_e\overline{\rho}_e\right\}}$, we complete the proof. □

Lemma 12 enables us to approximate each component of ϑ by a finite set of points within its range. Let (sufficiently large) K denote the number of points and let us denote the set of these points by $Q := \{\pi_1, \ldots, \pi_K\} \subset \mathbb{R}$. We use the set $\mathcal{Q} := \{1, \ldots, K\}$ to index these points. For each edge $e \in \mathcal{E}$, time $t \in \mathcal{T}$ and sample $l \in \mathcal{L}$, let us define the binary variable $q_{e,k}^{(l)}(t)$ to be equal to one if $\vartheta_e^{(l)}(t)$ is approximated by π_k; otherwise $q_{e,k}^{(l)}(t) = 0$. Then for each $e \in \mathcal{E}$, $t \in \mathcal{T}$ and $l \in \mathcal{L}$, we will then represent $\vartheta_e^{(l)}(t)$ by the following constraints

$$\begin{aligned}&\vartheta_e^{(l)}(t) = \sum_{k\in\mathcal{Q}} \pi_k q_{e,k}^{(l)}(t), \quad \sum_{k\in\mathcal{Q}} q_{e,k}^{(l)}(t) = 1, \quad \forall e \in \mathcal{E},\ t \in \mathcal{T},\ l \in \mathcal{L},\\ &q_{e,k}^{(l)}(t) \in \{0,\ 1\},\ \forall e \in \mathcal{E},\ t \in \mathcal{T},\ l \in \mathcal{L},\ k \in \mathcal{Q}.\end{aligned} \tag{3.25}$$

Using this representation, we find approximated solutions of (**P4′**) by solving the following

$$\max_{\substack{x,q,y,z,\rho,\\ \lambda,\mu,\nu,\eta,\vartheta}} -\lambda\epsilon(\beta) - \frac{1}{N}\sum_{e,t,l} \overline{f}_e \overline{\rho}_e \eta_e^{(l)}(t) + \frac{1}{N}\sum_{e,t,l,k} (\pi_k)^2 q_{e,k}^{(l)}(t), \tag{**P5**}$$

$$\text{s.t.} \quad \textbf{level approximation } (3.25),\ \textbf{second-order cone } (3.24),$$

$$\textbf{speed limits } (3.11),\ \textbf{sample trajectories } \{(3.15),\ (3.16)\},$$

$$\textbf{no congestion } \{(3.14),\ (3.17),\ (3.18),\ (3.19),\ (3.20)\}.$$

Problem (**P5**) is an SOCMIP and can be solved to optimum by commercial solvers, such as GUROBI and MOSEK. Note that for any feasible solution of (**P5**), it is feasible for (**P4′**) and its objective value is a valid lower bound for that of (**P4′**). Thus, any feasible solution of (**P5**) together with its objective value provide performance guarantees (3.9) via Lemma 11. Further, as the number of partition points $K \to \infty$, Problem (**P5**) is computationally equivalent to Problem (**P4′**), and therefore the same as Problem (**P4**). Notice that online solutions to Problem (**P5**) are inaccessible and (**P5**) serves as a tool for the performance analysis of the proposed algorithm. In the following section, we leverage this tool to analyze the performance of the algorithm in an academic example.

3.7 Case Study 1: Effectiveness of the Approach

In this section, we demonstrate in an example how to efficiently find a solution to (**P4**) that results in a robust data-driven variable-speed limit u with performance guarantee (3.9). The analysis of the results are twofold. First, we verify the effectiveness of the proposed integer solution search algorithm by comparing it with the monolith approach that solves an approximation to (**P4**), Problem (**P5**). Second, to verify the robustness of the solution and performance guarantees in probability, we compare the resulting distributionally robust data-driven control with the control obtained from the sample-averaged optimization problem. Both controls are applied on a naive highway simulator developed via the standard cell transmission model [33].

Simulation setting: We consider an 8-lane highway with length $L = 10$km and we divide it into $n = 5$ segments with equal length. Let the unit size of each time slot $\delta = 30$ sec and consider $T = 20$ time slots for a 10min planning horizon. For each edge $e \in \mathcal{E}$, we propose a traffic jam density of $\overline{\rho}_e = 1050$vec/km, a capacity of $\overline{f}_e = 3.1 \times 10^4$vec/h[9] and a maximal speed limit of $\overline{u}_e = 140$km/h. Let us consider $m = 5$ different candidate speed limits $\Gamma = \{40\text{km/h}, 60\text{km/h}, 80\text{km/h}, 100\text{km/h}, 120\text{km/h}\}$. On the 4^{th} edge $e := (3,4) \in \mathcal{E}$, we assume an accident happens during $\mathcal{T}$ with parameters $\overline{f}_e = 2.7 \times 10^4$vec/h. To evaluate the effect of the proposed algorithm, samples of the random variables $\rho(0)$, w, r^{in} and r^{o} are needed. In real-case studies, samples $\{\rho^{(l)}(0)\}_{l\in\mathcal{L}}$ can be obtained from highway sensors (loop detectors), while samples of the uncertain mainstream flows $\{\omega^{(l)}\}_{l\in\mathcal{L}}$ and flow fractions $\{r^{\text{in},(l)}, r^{\text{o},(l)}\}_{l\in\mathcal{L}}$ can be constructed either from a database of flow data on the highway, or from current measurements of ramp flows with the assumption that the process $\{\omega(t), r^{\text{in}}(t), r^{\text{o}}(t)\}_{t\in\mathcal{T}}$ is trend stationary.

Fictitious datasets: In this simulation example, the index set of accessible samples is given by $\mathcal{L} = \{1,2,3\}$. For each $l \in \mathcal{L}$, let us assume that each segment $e \in \mathcal{E}$ initially operates under a free flow condition with an initial density $\rho_e^{(l)}(0) = 260$vec/km. To ensure significant inflows of the system, we let the samples $\{\omega^{(l)}(t)\}_{l\in\mathcal{L},t\in\mathcal{T}}$ of the mainstream inflow to be chosen from the uniform distribution within interval $[2\times10^4, 2.4\times10^4]$vec/h. For each edge $e \in \mathcal{E}$ and time $t \in \mathcal{T}$, we further assume that samples $\{r^{\text{in},(l)}(t)\}_{l\in\mathcal{L}}$ and $\{r^{\text{o},(l)}(t)\}_{l\in\mathcal{L}}$ are generated from uniform distributions within interval $[0,5\%]$ and $[0,3\%]$, respectively. We also let the confidence value be $\beta = 0.05$ and the radius of the Wasserstein Ball $\epsilon(\beta) = 0.985$ as calculated in [70].

Effectiveness of the algorithm: To generate feasible solutions that can be carried out for a real time transportation system, we allocate $T_{\text{run}} = 1$min execution time for control design and run algorithms on a machine with two core 1.8GHz CPU and 8G RAM. In this allocated 1 minute, we consider the speed limit design in 2 approaches: 1) we run the proposed Algorithm 5 to solutions of the Problem (**P4**), and 2) we run optimization solver MOSEK to solutions of the

[9]The unit "vec" stands for "vehicles". Notice the proposed capacity is about 50% higher than the actually highway capacity in order to leverage the actual fundamental diagram for control.

Table 3.1. The efficiency of the proposed Algorithm 5.

	Algorithm 5[a]	Monolith
# of feasible cand.[b]	2	1
# of infeasible cand.	17	NA
LB (vec/h)	1.17×10^5	3.93×10^4
UB (vec/h)	1.55×10^5	1.55×10^5

[a]: Subproblems (UBP_k) and (LBP_k) are solved via MOSEK.
[b]: Candidate speed limit.

monolith Problem (**P5**). The partition number $K = 5$ is selected for the monolith approach.

We present in Table 3.1 the comparison of the mentioned two approaches. In 1 minute, the Algorithm 5 executed 19 candidate speed limits where 2 feasible speed limits were found at time 6.7 sec and 28.7 sec. We verified that $\hat{J}(u^{(2)}) = 1.17 \times 10^5 \text{vec/h}$ is the highest certificate obtained, i.e., $\hat{J}(u^{(2)}) \in \text{argmax}_{p=1,2}\{\hat{J}(u^{(p)}) \mid u^{(p)} \text{ is feasible}\}$, and the desired speed limits are

$$u^{(2)} = [120, 100, 80, 80, 100]\text{km/h}.$$

Compared with the proposed algorithm, the monolith approach returned a feasible solution with the speed limits $u^{\text{mon}} = [120, 120, 120, 40, 40]\text{km/h}$ and an approximated throughput $3.93 \times 10^4 \text{vec/h}$. It can be seen that 1) the gap (difference between UB and LB) obtained from the Algorithm 5 is tighter than that obtained from the monolith approach, and 2) the implementable speed limits proposed by the Algorithm 5 results in higher throughput than that achieved by the monolith.

In the following subsection, we use the speed limits $u^{(2)}$ to verify the guarantees on congestion elimination with high probability.

Distributionally robust decisions: To demonstrate the distributional robustness of our approach, we compare the performance of our speed limits design $u^{(2)}$ with the performance of the speed limits developed from a sample average optimization problem, which is also known as the dual version of the scenario-based approach, such as in [75]. In particular, the sample averaged version of (**P**) (equivalently, (**P1**)) is the one substitutes the unknown distribution $\mathbb{P}(u)$

with its empirical distribution $\hat{\mathbb{P}}(u)$. The resulting tractable formulation of the sample average problem, analogous to (**P4**), is the following

$$\begin{aligned}
\max_{\substack{x,y,z,\rho,\\ \mu,\nu,\eta}} \; & -\frac{1}{N}\sum_{e,t,l}\overline{f}_e\overline{\rho}_e\eta_e^{(l)}(t)+\frac{1}{N}\sum_{e,t,l}\nu_e^{(l)}(t)\rho_e^{(l)}(t),\\
\text{s.t.} \quad & \sum_{i\in O}\gamma^{(i)}(\overline{\rho}-\rho^c(\overline{u}))\otimes\mathbf{1}_T\circ z_i^{(l)}-\mu^{(l)}\\
& \qquad\qquad +\overline{f}\otimes\mathbf{1}_T\circ\eta^{(l)}\geq\mathbf{0}_{nT},\ \forall\, l\in\mathcal{L},\\
& \nu^{(l)}=\mu^{(l)}+\frac{1}{T}\sum_{i\in O}\gamma^{(i)}x_i,\ \forall\, l\in\mathcal{L},\\
& \mathbf{0}_{nT}\leq\eta^{(l)}\leq\overline{\eta},\ \forall\, l\in\mathcal{L},\\
& \textbf{speed limits}\ (3.11),\ \textbf{dual variable}\ (3.14),\\
& \textbf{sample trajectories}\{(3.15),\ (3.16)\}.
\end{aligned}$$

Note that the difference between the previous sample average problem and (**P4**) is that the former has a Wasserstein Ball radius $\epsilon(\beta)=0$ and, thus, unlike (**P4**), it does not provide a performance guarantee on congestion. We solve the above sample average problem to a suboptimal solution via the Algorithm 5 with the same setting as in solution to $u^{(2)}$. The resulting speed limit design is the following

$$u^{\text{sav}}=[60,60,80,60,100]\text{km/h}.$$

To verify the performance of $u^{(2)}$ and u^{sav}, we generated $N_{\text{val}}=10^3$ validation samples of random variables $\rho(0)$, w, r^{in} and r^{o} that are from the distributions described in the Fictitious datasets paragraph. The speed limit design $u^{(2)}$, u^{sav} as well as the validation dataset are integrated into a highway simulator with the highway parameter settings described in the Simulation setting paragraph.

Due to space limitations we cannot showcase all admissible sample trajectories for 10^3 scenarios, therefore in Fig. 3.4 we show an average of the admissible sample trajectories, i.e., the

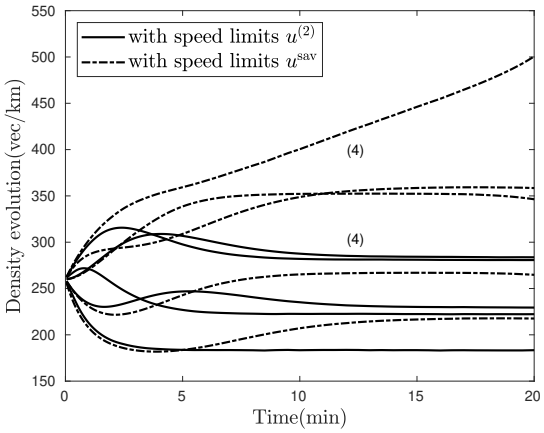


Figure 3.4. Density evolution of each segment e, with speed limits $u^{(2)}$ and u^{sav}. Each trajectory corresponds to a segment $e \in \{(1), (2), \ldots, (5)\}$. For simplicity we only label segment (4), which happens to have an accident during the planning horizon.

function $\frac{1}{N_{\text{val}}} \sum_{l \in \{1,\ldots,N_{\text{val}}\}} \rho_e^{(l)}(t)$ for each segment e, with speed limits $u^{(2)}$ and u^{sav}, and present an arbitrarily chosen scenario 638 in Fig. 3.5. We select the simulation time horizon to be twice of that the planning horizon's in order to see the effect of the design clearly. We verified that the density evolution under speed limits $u^{(2)}$, and, in particular, the density trajectory of accident edge (4), did not exceed the critical density values ($\rho_4^c(80\text{km/h}) = 335\text{vec/km}$) for each sample. Thus the highway $\mathcal{G}$ is kept free of congestion in this planning horizon $\mathcal{T}$ with high probability. However, the same robust behavior can not be guaranteed under speed limits u^{sav}, as vehicles accumulate significantly on edge (4) for too many samples (contrast to its critical density $\rho_4^c(60\text{km/h}) = 403\text{vec/km}$), see Fig. 3.4. We claim the robustness of our design compared to the design from sample average problem, as the latter does not ensure such out-of-sample performance guarantees.

3.8 Case study 2: Speed-Limit Control on San Diego Highway

In this section, we illustrate the proposed data-driven speed-limit control on a highway system located in San Diego, California, USA. The purpose of this simulation is to show the good

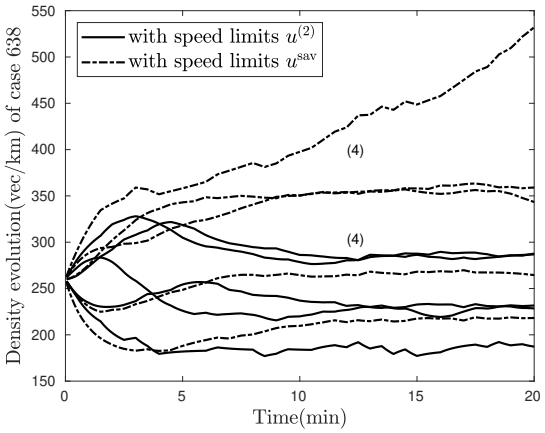


Figure 3.5. A representative density evolution of segments, with speed limits $u^{(2)}$ and u^{sav}. The sample 638 was arbitrarily chosen for demonstration purpose.

behavior of the proposed method under uncertainty as compared with that of the current traffic speed limits on the highway. While this is a first good indication, more work would be required to assess the behavior of the method over different traffic indices. However, our main focus is the theoretical development of the algorithm itself and so these questions are left for future work.

Highway system: We selected a highway section from Encinitas to Del Mar on the I-5 San Diego Freeway with length $L \approx 11$ km, as shown in Fig. 3.6. The highway was divided into $n = 26$ segments with various lengths $\{\text{len}_e\}_{e\in\mathcal{E}}$ ranging from 200 m to 2 km. These segments have a number of lanes $\{\text{lane}_e\}_{e\in\mathcal{E}}$ ranging from 4 to 8, and there are 7 on-ramps and 5 off-ramps distributed on the highway. We obtained real-time traffic data with 30 seconds precision from the California Highway Performance Measurement System (PeMS), and used it to reproduce the actual traffic flow features via the software Simulation of Urban MObility (SUMO) [77], which is a microscopic and continuous traffic simulation package. We calibrated the simulator using PeMS data which were collected between 12pm and 2pm on a particular day, and the speed limit on San Diego Freeway was 105 km/h or 65 mph.

Data-driven control: The data-driven control to solve (**P**) considered $T = 80$ time slots dividing 4-minutes planning horizon, with a unit size of each time slot $\delta = 3$ seconds. The infrastructure-related parameters $(n, L, \text{len}, \text{lane})$ were selected to be the same as that of the

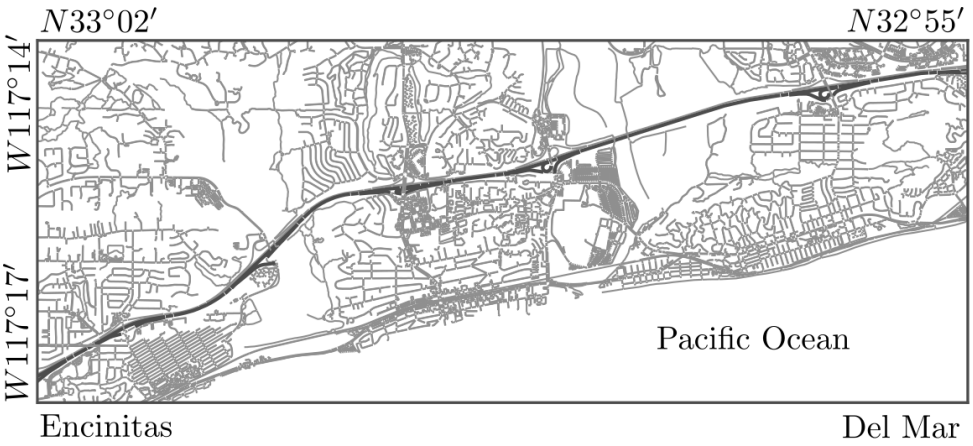


Figure 3.6. Highway section from Encinitas to Del Mar, San Diego, US.

highway system, only that ramp-related ultra-short lanes were excluded. We considered $m = 6$ different candidate speed limits $\Gamma = \{30, 50, 70, 90, 110, 130\}$ (km/h). The control u was obtained by implementing Algorithm 5 that solves the equivalent (**P4**), taking a $T_{\text{run}} = 1$ minute and executing Algorithm 5 every 2 minutes. To achieve realistic driving instructions, we added extra speed limit constraints to ensure constant u over each 1 minute interval.

System monitor: We assume the existence of a system monitor which provides information of the real-time traffic flow data $\{\varpi^{(l)}\}_{l\in\mathcal{L}}$ as well as the random events parameters $(\overline{f}, \overline{\rho}, \overline{u})$. Practically, these parameters can be calibrated in advance from PeMS historical data. In particular, at each time when Algorithm 5 is to be executed, we consider $N = 2$ accessible samples with the set $\mathcal{L} = \{1, 2\}$. Precisely, values of $\varpi^{(1)}$ were constructed and propagated using real-time highway sensor measurements (loop detectors data in PeMS) and values of $\varpi^{(2)}$ were obtained as the 7-day average data corresponding to the same time period (12pm to 2pm). This results in a radius for the Wasserstein Ball $\epsilon(\beta) \approx 5$, given the confidence value $\beta = 0.05$. Notice that more samples can be added to reduce the radius if various source of measurements are accessible, e.g., such as real-time GPS data. On the other hand, the infrastructure and event-related data $\{(\overline{f}_e, \overline{\rho}_e, \overline{u}_e)\}_{e\in\mathcal{E}}$ are determined theoretically, where values $\{\overline{f}_e\}_{e\in\mathcal{E}}$ have range $[1.1, 2.3]\times 10^4$ (vec/h), $\{\overline{\rho}_e\}_{e\in\mathcal{E}}$ with values in $[0.5, 1]\times 10^3$ (vec/km/edge), and $\{\overline{u}_e\}_{e\in\mathcal{E}}$ are assumed 200 (km/h).

Benchmarks: We consider a 2-hour scenario, from 12pm to 2pm, and assume a temporary

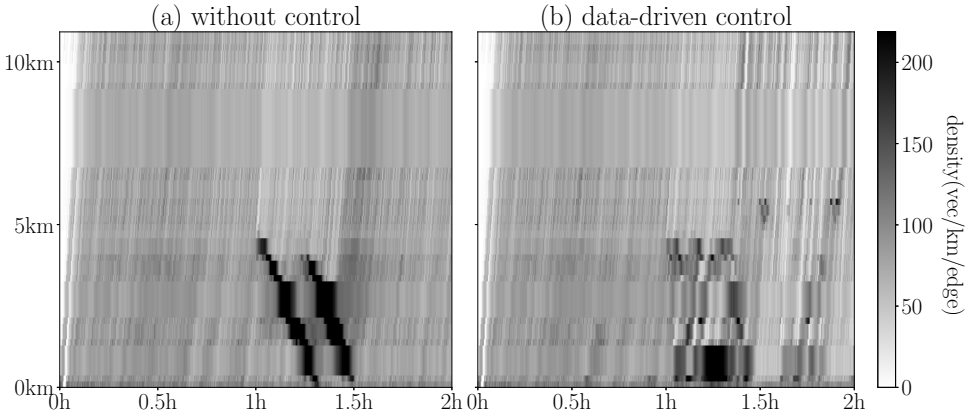


Figure 3.7. Time-space profile of traffic average density.

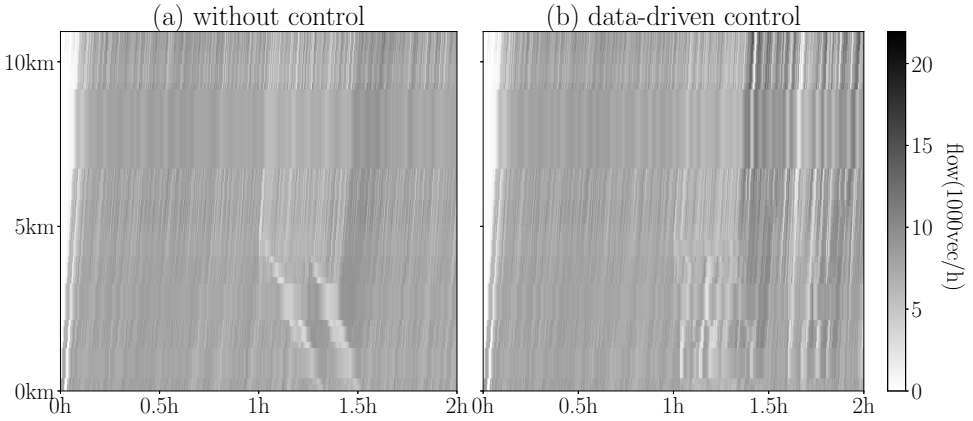


Figure 3.8. Time-space profile of traffic average flow.

lane closure on the 15^{th} segment between 1pm to 1 : 20pm, which introduces a capacity and jam density drop by 35% on that particular segment, located at 4.5 km from the entry. Further, we implement the proposed data-driven control between 12 : 30pm to 2pm, and compare the resulting performance with that of the highway system without control, i.e., with constant speed limit 105 km/h. Notice that, if congestion is inevitable, namely, the data-driven control problem (**P**) is infeasible under the admissible regime, we implement the default speed limit (105 km/h) instead.

Performance analysis: Fig. 3.7 shows the traffic density profile in time and space, where the origin indicates the entry of the highway at the initial time, i.e., Encinitas at time 12pm. The x-axis indicates the time (number of hours) passed from 12pm and the y-axis is the distance

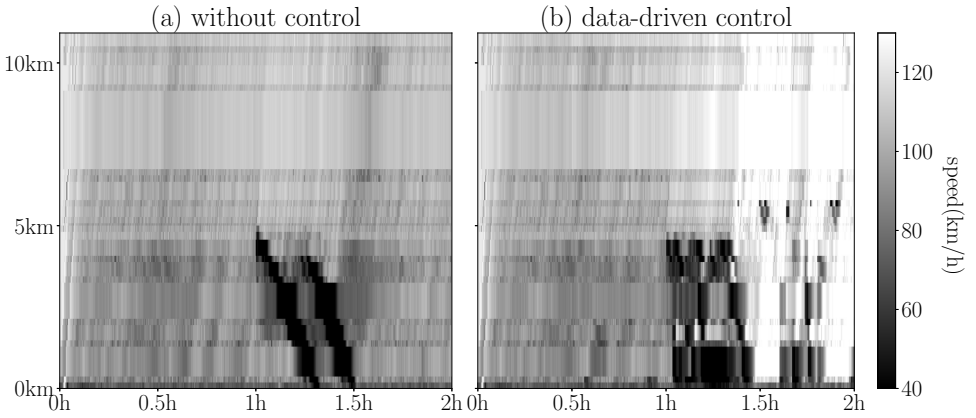


Figure 3.9. Time-space profile of traffic average speed.

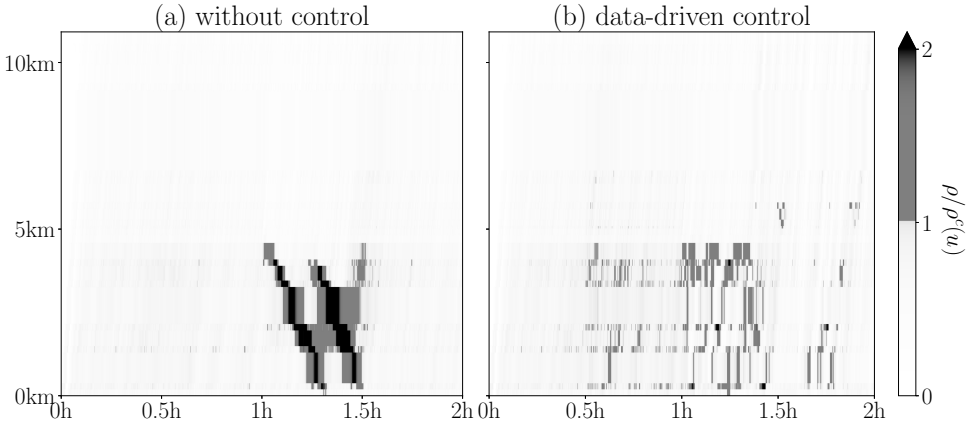


Figure 3.10. Time-space profile of congestion ratio $\rho(t)/\rho^c(u(t))$.

to the highway entry. Similarly, Fig. 3.8 and Fig. 3.9 demonstrate the flow and average speed profiles of those, respectively. For comparison, subfigures (a) are profiles without control and subfigures (b) are those with the data-driven control. It can be observed that, during the period 0h to 0.5h (12pm to 12 : 30pm), the two profiles were statistically identical. During period 0.5h to 1h (12 : 30pm to 1pm), as the mainstream flows were moderate, the speed-limit control assigned a higher speed limit (130 km/h) than the default 105 km/h on the majority of the segments. See Fig. 3.11 for the speed limit profile. After 1h (1pm), a significant capacity drop occurs on the middle of the highway due to a temporary lane closure. This event leads to congestion on the preceding segment and backward congestion waves start to transmit on the highway, see, e.g., subfigures (a). The darker parts on the profiles indicate congestion and notice how the congestion was transmitted to the entry of the highway over time. In addition, during the lane closure period (1pm to 1 : 20pm), the data-driven control then took effect to cancel/reducing the congestion transmission by dynamically assigning low speed limits on the upper stream of the highway. See, e.g., Fig. 3.11 on the speed limit assignment during 1h to 1.4h, which regulates the highway average speed as in Fig. 3.9(b). These actions eliminated the congestion waves and reduced the effect of the random events. See, e.g., Fig. 3.7 and 3.8 for comparison of the effect of the congestion elimination, Fig. 3.10 for the significant reduction of the congestion ratio $\rho(t)/\rho^c(t)$, and Fig. 3.11 for the assignment of the variable speed limit. When the random event ended, the speed-limit control then resumed to normal operation, which, during 1.4h to 2h, dynamically assigned speed limits to account for uncertainties on random ramp flows. See, e.g., the scattered speed restrictions in Fig. 3.11. Furthermore, notice in Fig. 3.10(b) how the actual highway density $\rho(t)$ violated the prediction-and-assignment critical density $\rho^c(u)$ via speed limits u. These are the original driving forces to update speed limits. Notice that, during this whole scenario, the data-driven control problem (**P**) is feasible. Otherwise, the speed limit would be set to the default 105 km/h at some time later than 0.5h in Fig. 3.11. This indicates a successful containment of the congestion. When the congestion is too heavy, (**P**) could be infeasible for some time period due to the extreme-high density on some of the segments. In

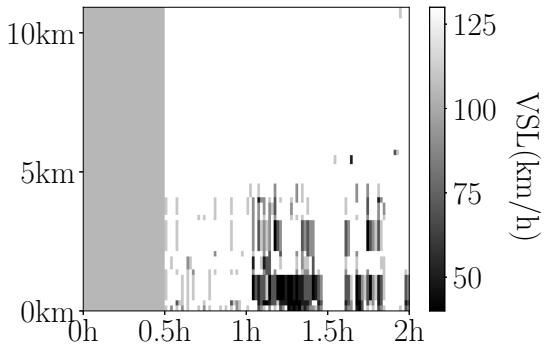


Figure 3.11. Time-space profile of speed limits $u(t)$.

those scenarios, to handle the congestion, small enough candidates can be added to the candidate speed limit set Γ in order for larger, admissible operation zone of (**P**). Otherwise, the congestion is inevitable as (**P**) is infeasible, and we simply select the default, pre-selected speed limits. At last, notice that the control performance relies heavily on the selection of the objective function of (**P**) as well as on the available information on the flow and random events data. We leave the questions regarding other traffic performance metrics and the improvement of the controller employing more accurate traffic models for the future work.

Chapter 3, in full, is under review for publication in International Journal on Robust and Nonlinear Control, entitled as *Data-driven predictive control for a class of uncertain control-affine systems*, D. Li, D. Fooladivanda, and S. Martínez. A motivating work appeared in the proceddings of European Control Conference, pp. 1055-1061, Napoli, Italy, 2019, as *Data-driven variable speed limit design for highways via distributionally robust optimization*, D. Li, D. Fooladivanda, and S. Martínez. The dissertation author was the primary investigator and author of these papers.

Chapter 4

Online Learning of Uncertain System Dynamics

This chapter presents a novel online learning algorithm for a class of unknown and uncertain dynamical systems or environments that are fully observable. First, we obtain a novel probabilistic characterization of systems whose mean behavior is known but which are subject to additive, unknown subGaussian disturbances. This characterization relies on concentration of measure results and is given in terms of ambiguity sets. Second, we extend the results to systems whose mean behavior is also unknown but described by a parameterized class of possible mean behaviors. The proposed algorithm adapts the ambiguity set dynamically by learning the parametric dependence online, and retaining similar probabilistic guarantees with respect to the additive, unknown disturbance. We illustrate the results on a differential-drive robot subject to environmental uncertainty.

4.1 Related Works

Fundamentally, the online learning of uncertain dynamical systems exploits input-output data to identify the representation of the system that best captures its behavior. In this way, first-principles system identification has been a success. The system identification literature broadly encompasses linear [76, 128] and non-linear systems [89, 103], with asymptotic performance guarantees. More recently, finite-sample analysis of identification methods have been proposed

for linear systems [38, 104, 116, 126]. These methods leverage modern measure-of-concentration results [40, 129] for non-asymptotic guarantees of the identification error bounds. Measure-of-concentration results are also used in [14, 16]. However, the goal of [14, 16] is to learn an unknown initial distribution evolving under a known dynamical system while assimilating data via a linear observer. This characterization is given in terms of *ambiguity sets*, which are constructed via multiple system trajectories or realizations. In contrast, here we employ Wasserstein metrics to develop an online learning algorithm for uncertain dynamical systems with similar-in-spirit probabilistic guarantees.

Statement of Contributions

This chapter proposes an online learning algorithm that characterizes a class of unknown and uncertain dynamical systems with probabilistic guarantees using a finite amount of online data. To achieve this, we first assume that the mean behavior of the stochastic system is known but the system states are subject to an additive, unknown subGaussian distribution, characterized by a set of distributions or *ambiguity set*. Then, we extend the results to systems whose mean behavior is unknown but belongs to a parameterized class of behaviors. In this regard, we propose a time-varying parameterized ambiguity set and a learning methodology to capture the behavior of the environment. We show how the proposed online learning algorithm retains desirable probabilistic guarantees with high confidence. A differential-drive robot subject to environmental uncertainty is provided for an illustration.

4.2 Problem Statement

This section presents the description of the uncertain dynamical environment which we aim to learn, with a problem definition. Let $t \in \mathbb{Z}_{\geq 0}$ denote time discretization. For each t, the uncertain system is characterized by a random variable $\boldsymbol{x} \in \mathbb{R}^n$ which evolves according to an

unknown, discrete-time, stochastic and, potentially, time-varying system

$$\boldsymbol{x}_{t+1} = f(t, \boldsymbol{x}_t, \boldsymbol{d}_t) + \boldsymbol{w}_t, \text{ with some } \boldsymbol{x}_0 \sim \mathbb{P}_0. \tag{4.1}$$

The distribution $\mathbb{P}_{t+1}$ characterizing $\boldsymbol{x}_{t+1}$ is determined by the current state's distribution, the unknown mapping $f : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, and random vectors $\boldsymbol{w}_t$ that cannot be captured by f. We further assume that $\boldsymbol{d}_t$ is an exogenous signal that is selected in advance or revealed online, which can play the role of an external reference or control. Let us denote by $\mathbb{W}_t$ the distribution of the random vector $\boldsymbol{w}_t \in \mathbb{R}^n$.

Assumption 9 (Independent and stationary subGaussian distributions). Consider random vectors $\boldsymbol{w}_t \in \mathbb{R}^n$, $t \in \mathbb{Z}_{\geq 0}$. It is assumed that: **(1)** The random vectors $\boldsymbol{w}_t$ are component-wise and time-wise independent, i.e., $w_{t,i}$ and $w_{k,j}$ are independent, for all $t \neq k$, $i \neq j$, $(t,k) \in \mathbb{Z}^2_{\geq 0}$ and $(i,j) \in \{1,\ldots,n\}$. **(2)** For each t, $\boldsymbol{w}_t$ is a zero-mean σ-subGaussian, i.e., for any $a \in \mathbb{R}^n$ we have $\mathbb{E}\left[\exp(a^\top \boldsymbol{w}_t)\right] \leq \exp(\|a\|^2\sigma^2/2)$.

Example 1 (σ-subGaussian distributions). A trivial example is a $\mathbb{W} \equiv \mathcal{N}(\boldsymbol{0}, \Sigma)$ with $\sigma_{\max}(\Sigma) \leq \sigma^2$. As any random vector supported on a compact set belongs to the subGaussian class, in particular, the following are σ-subGaussian distributions: **(1)** any zero-mean uniform distribution $\boldsymbol{w} \sim \mathcal{U}(\Omega)$ supported over $\Omega \subset B_\sigma(\boldsymbol{0})$; **(2)** any zero-mean discrete distribution with support $\Omega \subset B_\sigma(\boldsymbol{0})$.

This paper aims to obtain a tractable characterization of the unknown distribution $\mathbb{P}_{t+1}$ of the immediate-future environment state $\boldsymbol{x}_{t+1}$ online, $\forall t$. This is to be done by employing historical measurements, $\hat{\boldsymbol{x}}_k$, $k \leq t$, and data $\hat{\boldsymbol{d}}_k$, $k \leq t$.

Remark 19 (On finite-horizon learning of (4.1)**).** Our learning problem can be extended over finite horizons as follows. Let N be a learning horizon, then for each t the goal is to characterize the dynamical environment over the next N time slots, $\{t+1,\ldots,t+N\}$, with the previous knowledge of $\boldsymbol{d} := (\boldsymbol{d}^{(1)},\ldots,\boldsymbol{d}^{(N)})$. In other words, the objective is to characterize the joint distribution

$\mathbb{Q} := \mathbb{P}_{t+1} \otimes \cdots \otimes \mathbb{P}_{t+N}$ of the stochastic process $\boldsymbol{x} := (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)})$, where $\mathbb{P}_{t+i}$, $i \in \{1, \ldots, N\}$, are marginal distributions of those random state variables.

4.3 System Characterization with Perfect Information

We aim to provide a description the random dynamical system (4.1) via ambiguity sets. More precisely, given knowledge $\boldsymbol{d}$, and system data $\hat{\boldsymbol{x}}$, we look for a set of distributions $\mathcal{P}_{t+1} := \mathcal{P}_{t+1}(\boldsymbol{d}, \hat{\boldsymbol{x}})$ characterizing $\mathbb{P}_{t+1}$ via

$$\text{Prob}\,(\mathbb{P}_{t+1} \in \mathcal{P}_{t+1}) \geq 1 - \beta, \tag{4.2}$$

for some $\beta \in (0, 1)$. Observe that the probability Prob is taken wrt the historical random data outcomes. To do this, let $T_0 \in \mathbb{Z}_{>0}$ and $T := \min\{t, T_0\} \geq 1$, and consider the historical data, $\hat{\boldsymbol{x}}_k$ and $\hat{\boldsymbol{d}}_k$, for $k \in \mathcal{T} := \{t - T, \ldots, t - 1\}$. Particularly, Prob $:= \mathbb{P}_{t+1}^T$. Assuming a perfect knowledge of f, we show first how to use the data set $\mathcal{I} := \{\hat{\boldsymbol{x}}_t,\ \hat{\boldsymbol{x}}_k,\ \hat{\boldsymbol{d}}_k,\ k \in \mathcal{T}\}$ to construct $\mathcal{P}_{t+1}$, $\forall\, t \geq 0$.

Let us denote by $\mathbb{Q}_{t+1} \equiv \mathbb{Q}_{t+1}(\boldsymbol{d})$ the empirical distribution of $\boldsymbol{x}_{t+1}$ and define it as follows

$$\mathbb{Q}_{t+1} := \frac{1}{T} \sum_{k \in \mathcal{T}} \delta_{\{\xi_k(\boldsymbol{d})\}},$$

where $\xi_k(\boldsymbol{d}) := f(t, \hat{\boldsymbol{x}}_t, \boldsymbol{d}) + \hat{\boldsymbol{x}}_{k+1} - f(k, \hat{\boldsymbol{x}}_k, \hat{\boldsymbol{d}}_k),\ \forall k \in \mathcal{T}$. The following result enables us to construct the ambiguity set $\mathcal{P}_{t+1}$ that satisfies (4.2).

Lemma 13 (Asymptotic dynamic ambiguity set). *Let us assume that the system f is known at each time t. Given a confidence level $\beta \in (0, 1)$, parameter $T_0 \in \mathbb{Z}_{>0}$, and horizon $T = \min\{t, T_0\}$, let us assume $\boldsymbol{w}_k$ is i.d. for $k \in \mathcal{T}$. Then, there exists a positive scalar $\epsilon := \epsilon(T, \beta)$ such that* (4.2) *holds by selecting*

$$\mathcal{P}_{t+1} := \mathbb{B}_\epsilon(\mathbb{Q}_{t+1}) = \{\mathbb{P} \mid d_W(\mathbb{P}, \mathbb{Q}_{t+1}) \leq \epsilon\},$$

a Wasserstein ball centered at $\mathbb{Q}_{t+1}$ with radius

$$\epsilon := \sqrt{\frac{2n\sigma^2}{T}\ln(\frac{1}{\beta})} + O(T^{-1/\max\{n,2\}}),$$

where n is the dimension of $\boldsymbol{x}$ and σ is as in Assumption 9. Further, if $T_0 = \infty$, then as $t \to \infty$, $\epsilon \to 0$, i.e., the set $\mathcal{P}_{t+1}$ shrinks to the singleton $\mathbb{P}_{t+1}$ at a rate $O(1/T^{-1/\max\{n,2\}})$.

Proof. We prove this in two steps. First, we exploit the properties of $\boldsymbol{w}_t$. Then, we show the probabilistic guarantees of the dynamic ambiguity set.

Step 1: (SubGaussian Wasserstein distances): Given Assumption 9, on the subGaussian $\mathbb{W}_t$, the following holds:

(a) Following [101, Lemma 1], the distribution $\mathbb{W}_t$ satisfies

$$d_W(\hat{\mathbb{W}}_t, \mathbb{W}_t) \leq \sqrt{2n\sigma^2 \mathcal{D}(\hat{\mathbb{W}}_t|\mathbb{W}_t)}, \quad \forall\, \hat{\mathbb{W}}_t \in \mathcal{M}(\mathbb{R}^n), \tag{4.3}$$

where d_W and $\mathcal{D}$ denote the 1-Wasserstein metric and the KL divergence of two distributions $\hat{\mathbb{W}}_t$ and $\mathbb{W}_t$, respectively; and the set $\mathcal{M}(\mathbb{R}^n)$ is the space of all probability distributions supported on $\mathbb{R}^n$ with a finite first moment.

(b) Let us denote $\hat{\mathbb{W}}_t := \frac{1}{T}\sum_{k\in\mathcal{T}} \delta_{\{\hat{\boldsymbol{x}}_{k+1} - f(k,\hat{\boldsymbol{x}}_k,\hat{\boldsymbol{d}}_k)\}}$. Note that, by the assumption that $\boldsymbol{w}_k$ is i.d. for $k \in \mathcal{T}$, $\hat{\mathbb{W}}_t$ is the empirical distribution of $\boldsymbol{w}_t$. Then, following [101, Theorem 6], we claim that the equation (4.3) holds if and only if the random variable $d_W(\hat{\mathbb{W}}_t, \mathbb{W}_t)$ is $\sqrt{n}\sigma/\sqrt{T}$-subGaussian for all t. Equivalently, for all t and any $\lambda \in \mathbb{R}$, we have

$$\mathbb{E}\left[\exp\left(\lambda \cdot (d_W(\hat{\mathbb{W}}_t, \mathbb{W}_t) - \mathbb{E}[d_W(\hat{\mathbb{W}}_t, \mathbb{W}_t)])\right)\right] \leq \exp(\frac{n\lambda^2\sigma^2}{2T}). \tag{4.4}$$

(c) At each t, let us consider $C_t := \mathbb{E}[d_W(\hat{\mathbb{W}}_t, \mathbb{W}_t)]$. Following [40, Theorem 1] and [64, Theorem 3.1], we claim that, for $n \neq 2$, there exists a constant c, depending on Assumption 9,

such that

$$C_t \leq c \cdot T^{-1/\max\{n,2\}}.$$

In particular, we have $c = 3^{3.5} \times 2^{10} \times \sigma^3$ when $n = 1$. When $n > 3$, the parameter c is calculated by[1]

$$c := (1+\sqrt{2})(1+\sqrt{3}) \times 3^{3.5-1/n} \times 2^7 \times \sigma^3 \times n^{1.5}.$$

Step 2: (Probabilistically-guaranteed dynamic ambiguity sets): Knowing that the distributions $\mathbb{P}_{t+1}$ and $\mathbb{W}_t$ obey the environment dynamics (4.1), in other words, $\mathbb{P}_{t+1} \equiv f(t, \boldsymbol{x}_t, \boldsymbol{d}_t) + \mathbb{W}_t$, holds for any deterministic f. Similarly, we have $\mathbb{Q}_{t+1} \equiv f(t, \boldsymbol{x}_t, \boldsymbol{d}_t) + \hat{\mathbb{W}}_t$. Therefore, by the definition of the Wasserstein metric, we claim that

$$d_W(\mathbb{Q}_{t+1}, \mathbb{P}_{t+1}) \equiv d_W(\hat{\mathbb{W}}_t, \mathbb{W}_t), \quad \forall \boldsymbol{d},\ \forall t,$$

where the empirical distribution $\mathbb{Q}_{t+1}$ is described as in the statement of the lemma. Then by the Markov inequality, for any $\gamma \geq 0$ and $\lambda \geq 0$, we have

$$\begin{aligned}
\text{Prob}\left(d_W(\mathbb{Q}_{t+1}, \mathbb{P}_{t+1}) \geq \gamma\right) &= \text{Prob}\left(d_W(\hat{\mathbb{W}}_t, \mathbb{W}_t) \geq \gamma\right) \\
&= \text{Prob}\left(\exp(\lambda \cdot d_W(\hat{\mathbb{W}}_t, \mathbb{W}_t)) \geq \exp(\gamma\lambda)\right), \\
&\leq \exp(-\gamma\lambda)\,\mathbb{E}\left[\exp\left(\lambda \cdot d_W(\hat{\mathbb{W}}_t, \mathbb{W}_t)\right)\right].
\end{aligned}$$

Then by the property (4.4), we have

$$\text{Prob}\left(d_W(\mathbb{Q}_{t+1}, \mathbb{P}_{t+1}) \geq \gamma\right) \leq \exp\left(-(\gamma - C_t)\lambda + \frac{n\lambda^2\sigma^2}{2T}\right),$$

[1] These parameters are obtained based on 1−Wasserstein metric and the third moment of $\mathbb{W}$; the bound is slightly different when using the moment information with different order. For the case $n = 2$, the bound introduces logarithm term of T,e.g., $C_t \leq c(T/\log(T))^{-1/\max\{n,2\}}$, we refer reader to [40] and [64] for details.

where $C_t := \mathbb{E}[d_W(\hat{\mathbb{W}}_t, \mathbb{W}_t)]$. The optimal $\lambda \in \mathbb{R}_{\geq 0}$ that results in the tightest bound is taken to be

$$\lambda := \begin{cases} \frac{(\gamma - C_t)T}{n\sigma^2}, & \text{if } \gamma > C_t, \\ 0, & \text{if } \gamma \leq C_t, \end{cases}$$

resulting in, when $\gamma > C_t$,

$$\text{Prob}\left(d_W(\mathbb{Q}_{t+1}, \mathbb{P}_{t+1}) \geq \gamma\right) \leq \exp\left(-\frac{(\gamma - C_t)^2 T}{2n\sigma^2}\right).$$

Finally, let $\gamma > C_t$ be

$$\gamma = \epsilon := \epsilon(T, \beta) = \sqrt{\frac{2n\sigma^2}{T} \ln(\frac{1}{\beta})} + c \cdot T^{-1/\max\{n,2\}},$$

where c is determined as in step 1. This results in

$$\text{Prob}\left(d_W(\mathbb{Q}_{t+1}, \mathbb{P}_{t+1}) \geq \epsilon\right) \leq \beta,$$

and further, we have

$$\text{Prob}\left(d_W(\mathbb{Q}_{t+1}, \mathbb{P}_{t+1}) \leq \epsilon\right) \geq 1 - \beta.$$

Equivalently, we have (4.2) by selecting $\mathcal{P}_{t+1} := \mathbb{B}_\epsilon(\mathbb{Q}_{t+1})$. If we take $T_0 = \infty$, we have $T = t$ with $t \to \infty$. Then, it obviously follows that $\mathcal{P}_{t+1}$ shrinks to $\mathbb{P}_{t+1}$ as $t \to \infty$. □

In practice, T_0, and β need to be selected empirically, in order to efficiently address the particular problem that leverages the characterization of (4.1).

4.4 Characterization in a Parameterized Family

The construction of the empirical distribution $\mathbb{Q}_{t+1}$ of the previous section relies on the knowledge of f. When f is unknown, one may represent f as belonging to a parameterized class

of functions. Such as the approach adopted in the neural networks field and Koopman operator theory. Here, we focus on the case that f is approximated by a linear combination of a class of functions or "predictors" as follows.

Assumption 10 (Environment predictor class). There exists a set of predictors $f^{(i)} : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $(t, \boldsymbol{x}, \boldsymbol{d}) \mapsto f^{(i)}(t, \boldsymbol{x}, \boldsymbol{d})$, $i \in \{1, \ldots, p\}$, such that: (1) The vector fields $f^{(1)}, f^{(2)}, \ldots, f^{(p)}$ are linearly independent almost everywhere. (2) There exists potentially time-varying coefficients $\boldsymbol{\alpha}^\star := (\alpha_1^\star, \ldots, \alpha_p^\star) \in \mathbb{R}^p$ such that

$$f(t, \boldsymbol{x}, \boldsymbol{d}) = \sum_{i=1}^{p} \alpha_i^\star f^{(i)}(t, \boldsymbol{x}, \boldsymbol{d}).$$

As the selection of the predictors is not the subject of this study, we assume that the predictors are found in advance, and hence they are known to the learning algorithm.

The construction of an effective ambiguity set now depends on learning the dynamical environment mapping. Let us denote by $\boldsymbol{\alpha} \equiv \boldsymbol{\alpha}_t$ the estimated value of the parameter $\boldsymbol{\alpha}^\star$ at time t. To construct $\mathcal{P}_{t+1}$, consider T predictions of $\boldsymbol{x}_{t+1}$ using $f^{(i)}$, denoted by $\xi_k^{(i)}(\boldsymbol{\alpha}, \boldsymbol{d})$. For each $k \in \mathcal{T}$, $i \in \{1, \cdots, p\}$, and given $\boldsymbol{d} := \boldsymbol{d}_t$, we define

$$\xi_k^{(i)}(\boldsymbol{\alpha}, \boldsymbol{d}) := f^{(i)}(t, \hat{\boldsymbol{x}}_t, \boldsymbol{d}) + \frac{\hat{\boldsymbol{x}}_{k+1}}{\boldsymbol{\alpha}^\top \mathbf{1}_p} - f^{(i)}(k, \hat{\boldsymbol{x}}_k, \hat{\boldsymbol{d}}_k).$$

Now, we select the empirical $\hat{\mathbb{P}}_{t+1} \equiv \hat{\mathbb{P}}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{d})$, as follows:

$$\hat{\mathbb{P}}_{t+1} := \frac{1}{T} \sum_{k \in \mathcal{T}} \delta_{\{\sum_{i=1}^{p} \alpha_i \xi_k^{(i)}(\boldsymbol{\alpha}, \boldsymbol{d})\}}. \tag{4.5}$$

The following result enables the construction of the ambiguity set $\mathcal{P}_{t+1}$, relying on both $\boldsymbol{d}$ and $\boldsymbol{\alpha}$, which satisfies (4.2).

Theorem 9 (Adaptive dynamic ambiguity set). *Assume that the data set $\mathcal{I}$ is accessible, $\forall\, t$. Further, let Assumption 10, on the environment predictor class, hold for some $\boldsymbol{\alpha}^\star$ at time $t \in \mathcal{T}$.*

Then, given a confidence level $\beta \in (0,1)$, horizon parameter T_0, and a learning parameter $\boldsymbol{\alpha} \equiv \boldsymbol{\alpha}_t \in \mathbb{R}^p$, there exists a scalar $\hat{\epsilon} := \hat{\epsilon}(t, T, \beta, \boldsymbol{\alpha}, \boldsymbol{d})$ such that (4.2) *holds by selecting*

$$\mathcal{P}_{t+1} := \mathbb{B}_{\hat{\epsilon}}(\hat{\mathbb{P}}_{t+1}) = \{\mathbb{P} \mid d_W(\mathbb{P}, \hat{\mathbb{P}}_{t+1}) \leq \hat{\epsilon}\},$$

where $\hat{\epsilon} = \epsilon + \|\boldsymbol{\alpha}^\star - \boldsymbol{\alpha}\|_\infty H(t, T, \boldsymbol{d})$, with

$$H(t, T, \boldsymbol{d}) := \frac{1}{T} \sum_{i=1}^{p} \sum_{k \in \mathcal{T}} \|f^{(i)}(k, \hat{\boldsymbol{x}}_k, \hat{\boldsymbol{d}}_k) - f^{(i)}(t, \hat{\boldsymbol{x}}_t, \boldsymbol{d})\|,$$

and the radius ϵ is selected as in Lemma 13.

Proof. From the triangular inequality,

$$d_{\mathrm{W}}(\mathbb{P}_{t+1}, \hat{\mathbb{P}}_{t+1}) \leq d_{\mathrm{W}}(\mathbb{P}_{t+1}, \mathbb{Q}_{t+1}) + d_{\mathrm{W}}(\mathbb{Q}_{t+1}, \hat{\mathbb{P}}_{t+1}),$$

where by Lemma 13, we have

$$\mathrm{Prob}\,(d_{\mathrm{W}}(\mathbb{P}_{t+1}, \mathbb{Q}_{t+1}) \leq \epsilon) \geq 1 - \beta.$$

To evaluate the second term above, we apply the definition of Wasserstein metric, given as

$$d_{\mathrm{W}}(\mathbb{Q}_{t+1}, \hat{\mathbb{P}}_{t+1}) = \sup_{h \in \mathcal{L}} \int_{\mathcal{Z}} h(\xi) \mathbb{Q}_{t+1}(d\xi) - \int_{\mathcal{Z}} h(\xi) \hat{\mathbb{P}}_{t+1}(d\xi),$$

where the set $\mathcal{Z}$ is the support of the random variable $\boldsymbol{x}_{t+1}$ and the set $\mathcal{L}$ is the space of all

Lipschitz functions defined on $\mathcal{Z}$ with Lipschitz constant 1. Then, we equivalently write

$$
\begin{aligned}
d_{\mathrm{W}}(\mathbb{Q}_{t+1}, \hat{\mathbb{P}}_{t+1}) &= \sup_{h \in \mathcal{L}} \left\{ \frac{1}{T} \sum_{k \in \mathcal{T}} \left(h(\xi_k) - h(\sum_{i=1}^{p} \alpha_i \xi_k^{(i)}) \right) \right\}, \\
&\leq \sup_{h \in \mathcal{L}} \left\{ \frac{1}{T} \sum_{k \in \mathcal{T}} |h(\xi_k) - h(\sum_{i=1}^{p} \alpha_i \xi_k^{(i)})| \right\} \\
&\leq \frac{1}{T} \sum_{k \in \mathcal{T}} \|\xi_k - \sum_{i=1}^{p} \alpha_i \xi_k^{(i)}\| \\
&= \frac{1}{T} \sum_{k \in \mathcal{T}} \| \sum_{i=1}^{p} (\alpha_i - \alpha_i^\star)(f^{(i)}(k, \hat{\boldsymbol{x}}_k, \hat{\boldsymbol{d}}_k) - f^{(i)}(t, \hat{\boldsymbol{x}}_t, \boldsymbol{d}))\|, \\
&\leq \sum_{i=1}^{p} |\alpha_i^\star - \alpha_i| \left[\frac{1}{T} \sum_{k \in \mathcal{T}} \|f^{(i)}(k, \hat{\boldsymbol{x}}_k, \hat{\boldsymbol{d}}_k) - f^{(i)}(t, \hat{\boldsymbol{x}}_t, \boldsymbol{d})\| \right], \\
&\leq \|\boldsymbol{\alpha}^\star - \boldsymbol{\alpha}\|_\infty H(t, T, \boldsymbol{d}),
\end{aligned}
$$

where the first line comes from the Wasserstein distance between discrete distributions $\mathbb{Q}_{t+1}$ and $\hat{\mathbb{P}}_{t+1}$; the second line is followed by adding absolute operation and applying triangular inequality; the third line comes from the definition of the set $\mathcal{L}$; the fourth line is from the definition of the environment predictions and Assumption 10, on the representation of unknown environment f; the fifth one applies triangular inequality, and the last line uses the Hölder's inequality. Note that the derived bound of $d_{\mathrm{W}}(\mathbb{Q}_{t+1}, \hat{\mathbb{P}}_{t+1})$ holds true with probability one. Then, by summing the probability bounds of the two terms, we obtain

$$
\mathrm{Prob}\left(d_{\mathrm{W}}(\mathbb{P}_{t+1}, \hat{\mathbb{P}}_{t+1}) \leq \hat{\epsilon} \right) \geq 1 - \beta,
$$

which can be written as (4.2) with $\mathcal{P}_{t+1} := \mathbb{B}_{\hat{\epsilon}}(\hat{\mathbb{P}}_{t+1})$. □

Theorem 9 indicates that, if we select $\boldsymbol{\alpha}$ wisely, i.e., $\boldsymbol{\alpha} \equiv \boldsymbol{\alpha}^\star$, then the adaptive dynamic ambiguity set is identical to that of Lemma 13.

To estimate an unknown $\boldsymbol{\alpha}^\star$ while preserving the probabilistic guarantees, we propose an online learning algorithm that attempts to bring $\boldsymbol{\alpha}$ close to $\boldsymbol{\alpha}^\star$ with high probability. Intuitively,

our approach is based on the comparison of new obtained data with updates given by a predictor combination.

Theorem 10 (Learning of $\boldsymbol{\alpha}^\star$). *Let the data set $\mathcal{I}$ and predictors $\{f^{(i)}\}_i$ be given. For each $k \in \mathcal{T}$ and $i \in \{1,\dots,p\}$, let us denote $f_k^{(i)} := f^{(i)}(k, \hat{\boldsymbol{x}}_k, \hat{\boldsymbol{d}}_k)$. Consider the data matrix $A \equiv A_t \in \mathbb{R}^{p\times p}$ with*

$$A(i,j) := \frac{1}{T}\sum_{k\in\mathcal{T}} \langle f_k^{(j)}, P_k f_k^{(i)}\rangle,\ \ i,\ j \in \{1,\dots,p\},$$

where P_k is an online regularization matrix at time k, and let us consider the data vector $\boldsymbol{b} \equiv \boldsymbol{b}_t \in \mathbb{R}^p$, with components

$$\boldsymbol{b}(i) := \frac{1}{T}\sum_{k\in\mathcal{T}} \langle \hat{\boldsymbol{x}}_{k+1}, P_k f_k^{(i)}\rangle,\ \ i \in \{1,\dots,p\}.$$

Given $\eta > 0$, we select P_k such that $\|P_k f_k^{(i)}\| \le \eta$ for all $i \in \{1,\dots,p\}$, $k \in \mathcal{T}$, and select $\boldsymbol{\alpha} \equiv \boldsymbol{\alpha}_t$ to be

$$\boldsymbol{\alpha} = A^\dagger \boldsymbol{b}, \tag{4.6}$$

where $A^\dagger$ denotes the Moore–Penrose inverse of A. Let Assumption 9 and Assumption 10 hold, and take

$$c := \sigma e \eta \sqrt{np}\,\sigma_{\min}^{-1}(A),$$

where σ is that in Assumption 9, the constant $e \approx 2.718$, and $\sigma_{\min}(A)$ is the minimal non-zero principal singular value of A. Then by selecting $\gamma \ge nc$, the parameter $\boldsymbol{\alpha}$ is ensured to be close to $\boldsymbol{\alpha}^\star$ with high probability in the following sense:

$$\mathrm{Prob}\left(\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_\infty \le \gamma\right) \ge 1 - \exp\left(-\frac{(nc-\gamma)^2 T^2}{2\left[(2T-1)c\gamma + nc^2\right]}\right).$$

In particular, selecting $\gamma \ge nc/e$, we obtain a non-trivial bound with a slow confidence growth

rate as follows

$$\text{Prob}\left(\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\star}\|_{\infty}\leq\gamma\right)\geq 1-\frac{1}{\gamma}n\sigma\eta\sqrt{np}\sigma_{\min}^{-1}(A).$$

Proof. To see this, first, we will bound $\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\star}\|_{\infty}$ by samples of $\boldsymbol{w}_k$, $k\in\mathcal{T}$, then we apply concentration results for probabilistic bounds on $\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\star}\|_{\infty}$.

Step 1: (Bound on $\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\star}\|_{\infty}$): At each $k\in\mathcal{T}$, let us denote by $\hat{\boldsymbol{w}}_k$ a sample of $\boldsymbol{w}_k$ represented by

$$\hat{\boldsymbol{w}}_k:=\hat{\boldsymbol{x}}_{k+1}-\sum_{j=1}^{p}\alpha_j^{\star}f_k^{(j)}.$$

Then, we project the data $\hat{\boldsymbol{x}}_{k+1}$ on the direction of each regularized predictor $i\in\{1,\ldots,p\}$,

$$\langle\hat{\boldsymbol{x}}_{k+1},P_kf_k^{(i)}\rangle=\langle\sum_{j=1}^{p}\alpha_j^{\star}f_k^{(j)}+\hat{\boldsymbol{w}}_k,P_kf_k^{(i)}\rangle,$$

where, given a scalar $\eta>0$, the time-dependent regularization matrix is selected so that $\|P_kf_k^{(i)}\|\leq\eta$, for all $i\in\{1,\ldots,p\}$. Averaging the above equalities over $k\in\mathcal{T}$, we have for each component i:

$$\boldsymbol{b}(i)=\sum_{j=1}^{p}\alpha_j^{\star}A(i,j)+\frac{1}{T}\sum_{k\in\mathcal{T}}\langle\hat{\boldsymbol{w}}_k,P_kf_k^{(i)}\rangle,$$

where

$$\boldsymbol{b}(i):=\frac{1}{T}\sum_{k\in\mathcal{T}}\langle\hat{\boldsymbol{x}}_{k+1},P_kf_k^{(i)}\rangle,\ \ i\in\{1,\ldots,p\},$$

$$A(i,j):=\frac{1}{T}\sum_{k\in\mathcal{T}}\langle f_k^{(j)},P_kf_k^{(i)}\rangle,\ \ i,\ j\in\{1,\ldots,p\}.$$

By selecting $\boldsymbol{\alpha}$ as in (4.6), the relation $\boldsymbol{b}(i)=\sum_{j=1}^{p}\alpha_jA(i,j)$ holds, for each i. By subtracting the above equation from the one related to $\boldsymbol{\alpha}^{\star}$, we have

$$\sum_{j=1}^{p}(\alpha_j-\alpha^{\star})A(i,j)=\frac{1}{T}\sum_{k\in\mathcal{T}}\langle\hat{\boldsymbol{w}}_k,P_kf_k^{(i)}\rangle.$$

By taking the Moore–Penrose inverse of A, we obtain

$$\boldsymbol{\alpha} - \boldsymbol{\alpha}^{\star} = A^{\dagger}\boldsymbol{c},$$

where the vector $\boldsymbol{c}$ is

$$\frac{1}{T}\sum_{k\in\mathcal{T}}\left(\langle\hat{\boldsymbol{w}}_k, P_k f_k^{(1)}\rangle, \ldots, \langle\hat{\boldsymbol{w}}_k, P_k f_k^{(p)}\rangle\right)^{\top}.$$

Take the ∞-norm operation on both sides, we have

$$\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^{\star}\|_{\infty} \leq \|A^{\dagger}\|_{\infty}\|\boldsymbol{c}\|_{\infty},$$

where we can write $\|\boldsymbol{c}\|_{\infty}$ as the following

$$\begin{aligned}
\|\boldsymbol{c}\|_{\infty} :&= \frac{1}{T}\max_{i\in\{1,\ldots,p\}}\left\{\left|\sum_{k\in\mathcal{T}}\langle\hat{\boldsymbol{w}}_k, P_k f_k^{(i)}\rangle\right|\right\},\\
&\leq \frac{1}{T}\max_{i\in\{1,\ldots,p\}}\left\{\sum_{k\in\mathcal{T}}|\langle\hat{\boldsymbol{w}}_k, P_k f_k^{(i)}\rangle|\right\},\\
&\leq \frac{1}{T}\max_{i\in\{1,\ldots,p\}}\left\{\sum_{k\in\mathcal{T}}\left(\|\hat{\boldsymbol{w}}_k\|\cdot\|P_k f_k^{(i)}\|\right)\right\},\\
&\leq \frac{1}{T}\sum_{k\in\mathcal{T}}\left(\|\hat{\boldsymbol{w}}_k\|\cdot\max_{i\in\{1,\ldots,p\}}\left\{\|P_k f_k^{(i)}\|\right\}\right),\\
&\leq \frac{\eta\sqrt{n}}{T}\sum_{k\in\mathcal{T}}\|\hat{\boldsymbol{w}}_k\|_{\infty},
\end{aligned}$$

where we achieve the first inequality by moving the absolute operation into the sum operation; the second inequality uses Hölder's inequality; the third inequality is achieved by moving max operation into sum operation; the forth one is achieved by the norm equivalence and the fact that $\|P_k f_k^{(i)}\| \leq \eta$ for all $i \in \{1,\ldots,p\}$. Then, we achieve the following bound

$$\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^{\star}\|_{\infty} \leq \eta\sqrt{n}\|A^{\dagger}\|_{\infty}\left[\frac{1}{T}\sum_{k\in\mathcal{T}}(\|\hat{\boldsymbol{w}}_k\|_{\infty})\right]. \tag{4.7}$$

Note that, by the equivalence of the matrix norm, we have

$$\|A^\dagger\|_\infty \le \sqrt{p}\|A^\dagger\|_2 = \sqrt{p}\,\sigma_{\max}(A^\dagger) \le \sqrt{p}\,\sigma_{\min}^{-1}(A),$$

where $\sigma_{\max}(A^\dagger)$ and $\sigma_{\min}(A)$ denote the maximal singular value of $A^\dagger$ and the minimal principal non-zero singular value of A, respectively.

Step 2: (Measure concentration on $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_\infty$): In this step, we find the probabilistic bound of $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_\infty$ by developing that of $\|\boldsymbol{w}_k\|_\infty$. Equivalently, given any $\gamma > 0$, we compute the following term

$$\text{Prob}\left(\frac{1}{T}\sum_{k\in\mathcal{T}}(\|\boldsymbol{w}_k\|_\infty) \ge \gamma\right). \tag{4.8}$$

There are two options to obtain the bound.

(1) (A naive bound via Markov inequality): By the Markov inequality, we obtain a bound (4.8) as

$$\text{Prob}\left(\frac{1}{T}\sum_{k\in\mathcal{T}}(\|\boldsymbol{w}_k\|_\infty) \ge \gamma\right) \le \frac{1}{\gamma T}\sum_{k\in\mathcal{T}}\mathbb{E}[\|\boldsymbol{w}_k\|_\infty].$$

By Lemma 23, we have $\mathbb{E}[\|\boldsymbol{w}_k\|_\infty] \le n\sigma$, resulting in

$$\text{Prob}\left(\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_\infty \le \gamma\right) \ge 1 - \frac{1}{\gamma}n\sigma\eta\sqrt{np}\sigma_{\min}^{-1}(A),$$

with non-trivial bound if we take $\gamma > n\sigma\eta\sqrt{np}\,\sigma_{\min}^{-1}(A)$.

(2) (A bound with exponential decay over T): For any $\lambda \ge 0$, the probability (4.8) is equivalent to

$$\text{Prob}\left(\exp\left(\sum_{k\in\mathcal{T}}\left(\frac{\lambda}{T}\|\boldsymbol{w}_k\|_\infty\right)\right) \ge \exp(\gamma\lambda)\right).$$

By the Markov inequality to the above probablity, we have

$$\text{Prob}\left(\frac{1}{T}\sum_{k\in\mathcal{T}}(\|\boldsymbol{w}_k\|_\infty) \ge \gamma\right) \le \exp(-\gamma\lambda)\mathbb{E}\left[\prod_{k\in\mathcal{T}}\exp\left(\frac{\lambda}{T}\|\boldsymbol{w}_k\|_\infty\right)\right].$$

By Assumption 9 on independence of $\boldsymbol{w}_k$, we have

$$\mathbb{E}\left[\prod_{k\in\mathcal{T}}\exp\left(\frac{\lambda}{T}\|\boldsymbol{w}_k\|_\infty\right)\right]=\prod_{k\in\mathcal{T}}\mathbb{E}\left[\exp\left(\frac{\lambda}{T}\|\boldsymbol{w}_k\|_\infty\right)\right].$$

For each $k\in\mathcal{T}$, we write each exp operation in its power series form as the following

$$\begin{aligned}\mathbb{E}\left[\exp\left(\frac{\lambda}{T}\|\boldsymbol{w}_k\|_\infty\right)\right]&=\mathbb{E}\left[1+\sum_{l=1}^{\infty}\frac{\left(\frac{\lambda}{T}\right)^l\|\boldsymbol{w}_k\|_\infty^l}{l!}\right],\\&=1+\sum_{l=1}^{\infty}\frac{\left(\frac{\lambda}{T}\right)^l\mathbb{E}\left[\|\boldsymbol{w}_k\|_\infty^l\right]}{l!}.\end{aligned}$$

By Lemma 23, we have

$$\mathbb{E}\left[\|\boldsymbol{w}_k\|_\infty^l\right]\leq n\sigma^l l^{\frac{l}{2}+1},\ \forall l=1,2,\ldots.$$

This gives[2]

$$\mathbb{E}\left[\exp\left(\frac{\lambda}{T}\|\boldsymbol{w}_k\|_\infty\right)\right]\leq 1+n\sum_{l=1}^{\infty}\left(\frac{\lambda\sigma e}{T}\right)^l.$$

To tighten the previous upper bound, consider any λ such that $\lambda\in[0,\frac{T}{\sigma e})$. Then the following bound holds[3]

$$\mathbb{E}\left[\exp\left(\frac{\lambda}{T}\|\boldsymbol{w}_k\|_\infty\right)\right]\leq 1+\frac{\lambda\sigma ne}{T-\lambda\sigma e}\leq\exp\left(\frac{\lambda\sigma ne}{T-\lambda\sigma e}\right).$$

Finally, we achieve

$$\text{Prob}\left(\frac{1}{T}\sum_{k\in\mathcal{T}}(\|\boldsymbol{w}_k\|_\infty)\geq\gamma\right)\leq\exp\left(-\gamma\lambda+\sum_{k\in\mathcal{T}}\frac{\lambda\sigma ne}{T-\lambda\sigma e}\right).$$

Finding an optimal bound is hard, and therefore we find a sub-optimal bound by selecting λ to be

$$\lambda=\begin{cases}\frac{T}{2\sigma e}-\frac{nT}{2\gamma}, & \text{if } \gamma\geq\sigma ne,\\ 0, & \text{if } \gamma<\sigma ne.\end{cases}$$

[2]We use two facts: 1) $l!\geq(l/e)^l$ and 2) $l^{1-\frac{l}{2}}\leq 1$, for all $l\in\mathbb{Z}_{\geq 0}$, where the constant $e=2.71828...$.

[3]We use the fact: $1+x\leq\exp(x)$ for $x\in\mathbb{R}$.

Then, we have the following

$$\begin{aligned}
&\text{Prob}\left(\frac{1}{T}\sum_{k\in\mathcal{T}}(\|\boldsymbol{w}_k\|_\infty)\geq\gamma\right)\\
&\leq\begin{cases}\exp\left(-\frac{(\sigma ne-\gamma)^2T^2}{2\left[(2T-1)\gamma\sigma e+n(\sigma e)^2\right]}\right), & \text{if } \gamma\geq\sigma ne,\\ 1, & \text{if } \gamma<\sigma ne.\end{cases}
\end{aligned}$$

In words, the probability bounds on the quality of $\boldsymbol{\alpha}$ is

$$\text{Prob}\left(\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^\star\|_\infty\leq\gamma\right)\geq 1-\exp\left(-\frac{(nc-\gamma)^2T^2}{2\left[(2T-1)c\gamma+nc^2\right]}\right),$$

with any $\gamma\geq nc$, where $c:=\sigma e\eta\sqrt{np}\,\sigma_{\min}^{-1}(A)$. □

Theorem 10 provides an online computation of a real-time $\boldsymbol{\alpha}$ that is close to $\boldsymbol{\alpha}^\star$ within a time varying distance γ with arbitrary high probability, where this distance γ depends only on the environment predictors as well as on the data sets. Note that, the confidence of selecting $\gamma>nc$ as a bound of $\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^\star\|_\infty$ increases exponentially as we increase the length T of the data sets. This motivates us to propose a computable dynamic ambiguity set, described as in Theorem 9, by selecting its dynamic radius as

$$\hat{\epsilon}=\epsilon+\gamma H(t,T,\boldsymbol{d}), \tag{4.9}$$

where ϵ, $\gamma>nc$ and H are chosen as in Lemma 13, Theorem 10, and Theorem 9, respectively. Such selection results in modified guarantees of (4.2) as follows

$$\text{Prob}\left(\mathbb{P}_{t+1}\in\mathcal{P}_{t+1}\right)\geq(1-\beta)\left(1-\exp\left(-\frac{(nc-\gamma)^2T^2}{2\left[(2T-1)c\gamma+nc^2\right]}\right)\right), \tag{4.10}$$

where as time t increases with a selection of $T_0=\infty$ (or $T=t$), the confidence value on the right hand side increases to $1-\beta$ exponentially fast. Fig. 4.1 compares the adaptation of the ambiguity

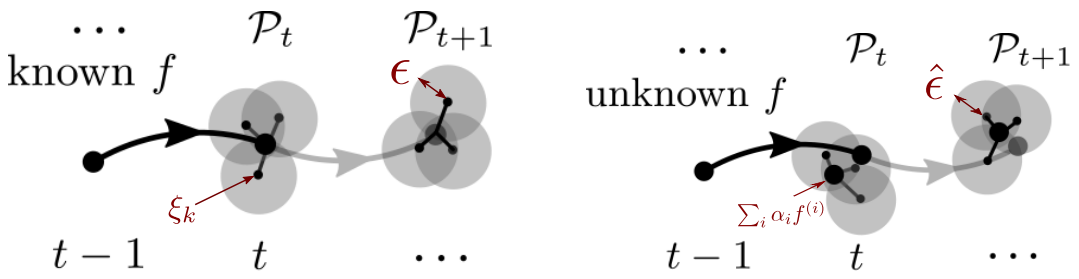


Figure 4.1. Online characterization of $\mathcal{P}_{t+1}$, with (without) f. The dark line is the trajectory of $\boldsymbol{x}$ and the gray part is yet to be revealed. At t, we obtain $\mathcal{P}_{t+1}$ with its elements supported on $T_0 = 3$ shaded regions with high probability. Each region k has center ξ_k ($\sum_i \alpha_i \xi_k^{(i)}$) and radius proportional to ϵ ($\hat{\epsilon}$). Note the centers of these regions are related to a known (learned) point $f(t, \hat{\boldsymbol{x}}_t, \boldsymbol{d})$ ($\sum_i \alpha_i f^{(i)}(t, \hat{\boldsymbol{x}}_t, \boldsymbol{d})$), they are close if the learning is effective.

set with and without knowing f.

Remark 20 (Data-driven selection of the radius). The radius of the adaptive ambiguity set (4.9) depends on the unknown, noise-related parameter σ, the regularization constant η, and on the online parameters $\sigma_{\min}(A)$. In many engineering problems, an upper bound σ of the noise-related parameter can be determined *a-priori* or empirically. The parameter η, together with the regularization matrices P, are introduced to ensure that (4.6) is well posed. In particular, P can be a diagonal matrix with each diagonal term scaling its corresponding components. At each t, the computation (4.6) needs an additional online regularization matrix, denoted by P_{t-1}. For example, P_{t-1} can be a diagonal matrix with the j^{th} diagonal term equal to $1/(\sqrt{p}\max_{i\in\{1,\ldots,p\}} |f_{t-1}^{(i)}(j)|)$, where $f_{t-1}^{(i)}(j)$ is the j^{th} component of $f_{t-1}^{(i)}$, which results in $\eta = 1$. Finally, $\sigma_{\min}(A)$ relies on the selection of the model set $\{f^{(i)}\}_i$ as well as the other two parameters η and σ. In practice, all the zero singular values of A is perturbed by the noise with a factor of σ. One could select the minimal non-zero principal singular value to be $\sigma_{\min}(A) = \min\{\sigma_i(A) \mid \sigma_i(A) > \sigma, i \in \{1, \ldots, p\}\}$.

Online procedure: To summarize, our online learning methodology is given in Algorithm table 6. Our approach leverages the adaptation of a dynamic ambiguity set, together with *a-priori* knowledge of $\boldsymbol{d}$, and learns model parameter $\boldsymbol{\alpha}$, and characterizes the unknown f online via $\mathcal{P}$.

$\mathbb{P}$-Learning 6. Learn($\mathcal{I}, \boldsymbol{d}$)

Require: $\{f^{(i)}\}_i$, β, T_0, σ, θ and $t = 1$;
Ensure: Online $\boldsymbol{\alpha}$, $\hat{\mathbb{P}}$, $\hat{\epsilon}$;
1: **repeat**
2: Update data set $\mathcal{I} := \mathcal{I}_t$ and knowledge $\boldsymbol{d} := \boldsymbol{d}_t$;
3: Compute $\boldsymbol{\alpha} := \boldsymbol{\alpha}_t$ as in (4.6);
4: Select $\hat{\mathbb{P}}_{t+1}$ as in (4.5) and $\hat{\epsilon} := \hat{\epsilon}_t$ as in (4.9);
5: Leverage $(\hat{\mathbb{P}}_{t+1}, \hat{\epsilon})$ as characterization of f;
6: $t \leftarrow t + 1$;
7: **until** time t stops.

4.5 Case Study: Vehicles in Unknown Road Conditions

In this section, we illustrate the previous results on a simple vehicle example. Consider a vehicle driving under various road conditions, where its control signal is derived in advance, according to a path-planner in an ideal environment.

Our goal is to learn the real-time environment and estimate the system states via our adaptive $\mathbb{P}$-Learning algorithm. Our vehicle is modeled as a differential-drive robot subject to uncertainty, see [63]:

$$\begin{aligned} x_1^+ =& x_1 + h\cos(x_3)u_1 + hw_1, \\ x_2^+ =& x_2 + h\sin(x_3)u_1 + hw_2, \\ x_3^+ =& x_3 - hu_2 + hw_3, \\ u_1 =& \frac{r}{2}(v_l + v_r + e_1), \\ u_2 =& \frac{r}{2R}(v_l - v_r + e_2), \end{aligned} \tag{4.11}$$

where $\boldsymbol{x} := (x_1, x_2, x_3) \in \mathbb{R}^2 \times [-\pi, \pi) \cong \mathbb{R} \times \mathbb{S}^1$ stands for vehicle position and orientation on the 2-D plane. We denote by $\boldsymbol{x}^+$ the state at the next time step and $\boldsymbol{w} := (w_1, w_2, w_3)$ a zero-mean, mixture of Gaussian and Uniform distributions, which are subGaussian uncertainties with $\sigma = 0.5$. We assume $\boldsymbol{x}_0 = (0, 0, 0)$ and $h = 10^{-3}$. The velocity $\boldsymbol{u} := (u_1, u_2)$ is determined by a wheel radius $r = 0.15$ m, the distance between wheels $R = 0.4$ m, the given wheel speed $\boldsymbol{d} := (v_l, v_r)$ and an unknown parameter $\boldsymbol{e} := (e_1, e_2)$, which depends on the wheel and road conditions. For

simulation purposes, we assume that the vehicle may move over three road zones, a slippery zone with $\boldsymbol{e}^{(1)} = (4, 0)$, a sandy zone with $\boldsymbol{e}^{(2)} = (-6, 0)$, and a smooth, regular zone with $\boldsymbol{e}^{(3)} = (0, 0)$, as described in Fig. 4.2. The vehicle executes the following left and right wheel speed plan (rad/s):

$$v_l = 10 - 0.5\sin(20h\pi t),$$
$$v_r = 10 + 0.5\sin(20h\pi t).$$

Now we employ our adaptive learning algorithm for the characterization of the uncertain vehicle states and learning of the unknown road-condition parameter $\boldsymbol{e}$ in real time. To do this, we take $p = 3$ predictors as in (4.11) with $\boldsymbol{w} \equiv 0$, and

$$\begin{aligned} i = 1, \quad & e_1 = 0, \qquad & e_2 = 0, \\ i = 2, \quad & e_1 = 10, \qquad & e_2 = 0, \\ i = 3, \quad & e_1 = 0, \qquad & e_2 = 10. \end{aligned}$$

Note that Assumption 10 holds with $\boldsymbol{\alpha}^\star := (0.6, 0.4, 0)$ in the slippery zone, $\boldsymbol{\alpha}^\star := (1.6, -0.6, 0)$ in the sandy zone and $\boldsymbol{\alpha}^\star := (1, 0, 0)$ in the smooth zone. We select $T_0 = 300$, and, at each time t, we have access to model sets $\{f^{(i)}\}_i$ as well as the real-time data set $\mathcal{I}_t$ and $\boldsymbol{d}$. Note that the notions of inner product and norm are those defined on the vector space $T(\mathbb{R}^2 \times \mathbb{S}) \equiv \mathbb{R}^3$. Recall that $h = 10^{-3}$, so a $T_0 = 300$ corresponds to a time window of order 0.3sec. We select online diagonal regularization matrices P with diagonal $(1/(\sqrt{3}\max_{i=1,2,3}|f^{(i)}(j)|)$ for $j = 1, 2$ and 1 for $j = 3$, resulting in $\eta = \max_{i,k\in\mathcal{T}} \|P_k f_k^{(i)}\|$.

Fig. 4.3 demonstrates the real-time parameter learning of α_1 and α_2. It can be seen that these unknown parameters are effectively learned and tracked over time. Fig. 4.4 shows the quality of the learned parameter $\boldsymbol{\alpha}$ and its effect on the determination of the radius of the adaptive ambiguity set. We note that, for a particular noise realization sequence, the estimated value $\gamma = n\sigma e\eta\sqrt{np}\sigma_{\min}^{-1}(A) + \theta$, with $\theta = 0.01$, upper bounds $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_\infty$ in high probability. The large spikes in the figure are due to the change of zone, resulting in a large error. This is

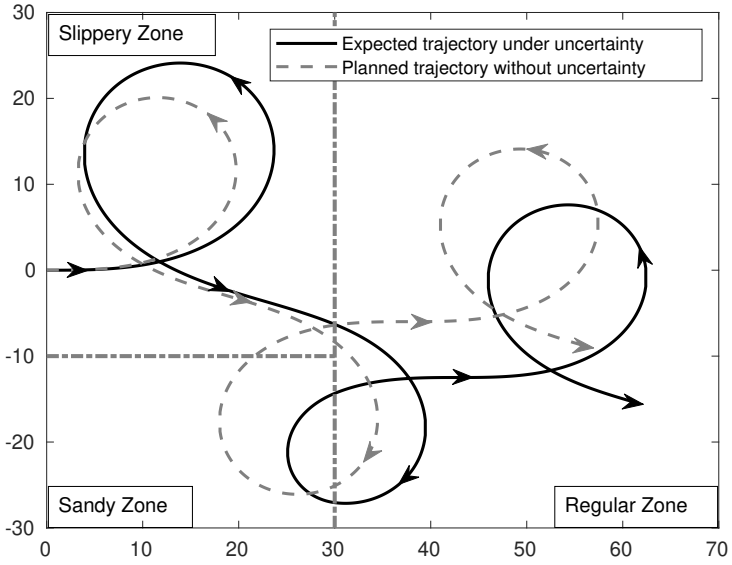


Figure 4.2. Path plan and actual trajectory in various $\mathbb{R}^2$ road zones.

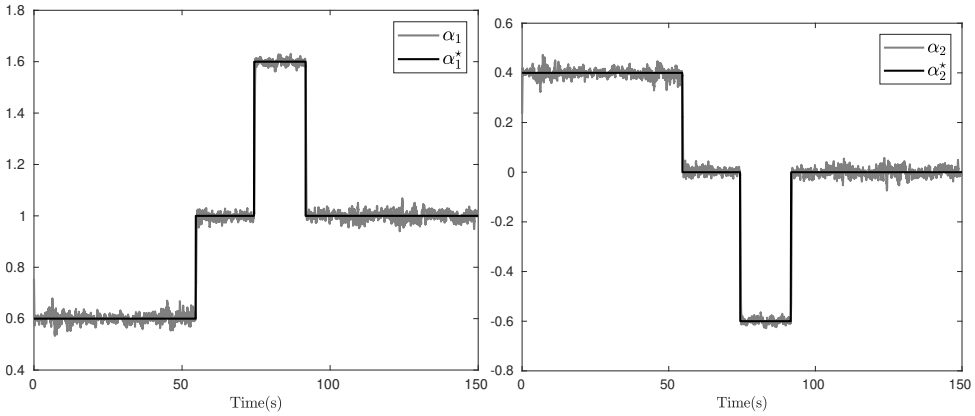


Figure 4.3. Real-time learning parameter α_1 and α_2.

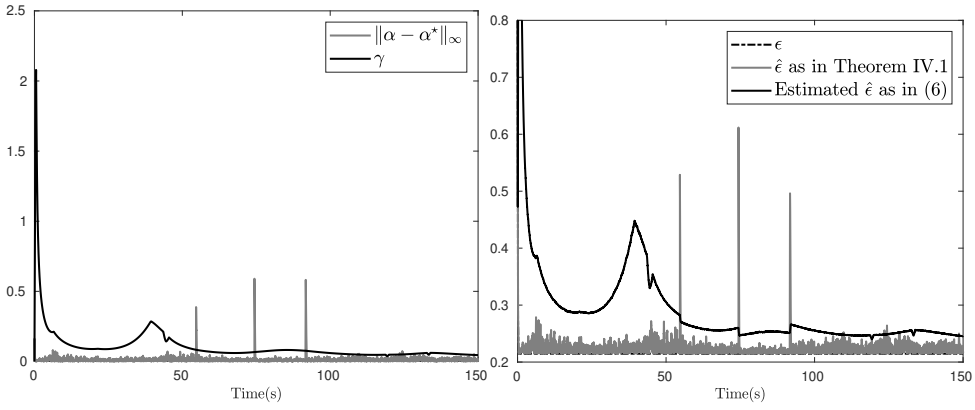


Figure 4.4. Quality of $\boldsymbol{\alpha}$ and the estimated radius $\hat{\epsilon}$.

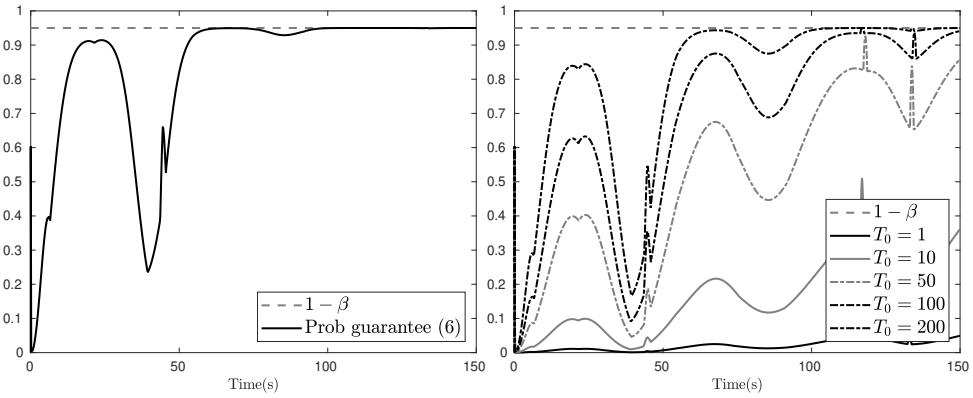


Figure 4.5. Online guarantee (4.10) and samples of (4.10) with various T_0.

expected, as the true $\alpha^\star$ changed discontinuously. Meanwhile, the estimated radius $\hat{\epsilon}$ of the adaptive ambiguity set, calculated as in (4.9), is a conservative estimate of the unknown *a-priori* $\hat{\epsilon}$ as in Theorem 9. The true $\hat{\epsilon}$ captures exactly the ambiguity set over the time sequence $\mathcal{T}$, for a $\beta = 0.05$. Over time, we empirically see the difference between the approximated $\hat{\epsilon}$ via γ and the true one become close. In practice, the radius $\hat{\epsilon}$ can be selected in a data-driven fashion, e.g., as in Remark 20, to serve as a way for less conservative estimation of the radius in probability. We show in Fig. 4.5 the online guarantee (4.10) of this particular case study, and various samples of (4.10), obtained by taking different time horizon T_0.

Chapter 4, in full, is a reprint of *Online learning of parameterized uncertain dynamical environments with finite-sample guarantees*, D. Li, D. Fooladivanda, and S. Martínez, IEEE Control Systems Letters, 5(4):1351-1356, 2021, which was presented at American Control Conference, New Orleans, LA, US, 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Online Optimization with Learned Systems

This chapter presents a new framework to solve online optimization and learning problems with unknown and uncertain dynamical systems or environments. This framework enables us to simultaneously learn the uncertain dynamical system while making online decisions in a quantifiably robust manner. The main technical approach relies on the theory of distributional robust optimization that leverages adaptive probabilistic ambiguity sets. However, as defined, the ambiguity set usually leads to online intractable problems, and the first part of this chapter is directed to find reformulations in the form of online convex problems for two subclasses of objective functions. To solve the resulting problems in the proposed framework, we further introduce an online version of the Nesterov's accelerated-gradient algorithm. We determine how the proposed solution system achieves a probabilistic regret bound under certain conditions. Two applications illustrate the applicability of the proposed framework.

5.1 Related Works

This chapter aims to refine online optimization techniques, e.g., Online Convex Programming (OCP), with control-theoretic techniques, including Model Predictive Control [22, 83] and Kalman Filtering [48]. Intuitively, control-theoretic approaches implicitly aim to learn models of loss functions of OCP in connection with an underlying and uncertain dynamical system of

interest. However, the effective characterization of loss functions requires strong assumptions on the type of systems as well as uncertainty. Clearly, approximate system models result in inaccurate loss-function predictions, which further leads to performance degradation of OCPs.

To deal with such challenges, data-driven approaches have regained attention. These methods are originated from system identification literature [76, 103, 128], which learn models from sufficient amount of properly collected data. More recently, Willem's fundamental lemma in Behavioral System Theory has been leveraged for system learning [81, 131], as well as estimation [30, 81] and predictive control [2, 11, 29, 107]. These approaches approximate models well using only data, however the size of the data can be very large. In the light of recent developments on the measure-of-concentration results [40] and its applications to distributionally robust optimization [36, 114], linear systems [38, 104, 116, 126] and stochastic processes [15, 16], new online-tractable and performance-guaranteed solutions to OCPs under unknown systems or environments are possible. In Chapter 4, we proposed a method for the online learning of unknown system dynamics, which provided a guaranteed, probabilistic characterization of the system behavior using tight sets of distributions, or *online ambiguity sets*. Here, we further consider an associated online optimization problem, where the objective functions explicitly depend on an unknown and uncertain dynamical system or environment. Further, we assume that imperfect models are accessible for online learning using ambiguity sets. The integration of such learning procedure achieves accurate predictions of the loss functions and enables the solution to OCPs a similar-in-spirit, worst-case guarantees in high probability.

Statement of Contributions

The contributions of this chapter are the following: 1) We formulate a class of online optimization problems in which the dynamical environment is uncertain, as an optimization with respect to worst-case-environment characterization using ambiguity sets. We show that such formulations provide guarantees in performance while maintaining online-problem tractability

under some conditions. 2) To derive the tractable formulation explicitly, two application scenarios are considered. The first formulation refers to an optimal control problem for an uncertain dynamical system. The second formulation is an online resource allocation under uncertainty, where the proposed formulation results in online and non-smooth convex optimization problems. 3) We propose an online solution technique which extends Nesterov's accelerated-gradient method achieving an optimal first-order convergence rate for smooth and offline convex problems. These algorithms allow us to solve the derived online non-smooth problems. 4) We analytically quantify the dynamic regret of online decisions subject to unknown environments with a probabilistic regret bound. In particular, we characterize an interplay between the derived probabilistic regret bound and the learning parameters. 5) We numerically quantify our analytical results using two examples to demonstrate the effectiveness of the proposed framework.

5.2 Problem Statement

Here, we introduce a class of online optimization problems, where the objective function is time-varying according to an unknown dynamical environment. At every time instant $t \in \mathbb{Z}_{\geq 0}$, we characterize the dynamical environment via an unknown random variable $\boldsymbol{x}_t \in \mathbb{R}^n$ which is subject to a distribution $\mathbb{P}_t$. Let $\boldsymbol{u}_t \in \mathcal{U} \subset \mathbb{R}^m$ be the online decision at time t and denote by $\ell : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, $(\boldsymbol{u}, \boldsymbol{x}) \mapsto \ell(\boldsymbol{u}, \boldsymbol{x})$ *a priori* selected, measurable loss function. We assume that the set $\mathcal{U}$ is compact, and we are interested in making an online decision or control $\boldsymbol{u}_t \in \mathcal{U}$ that minimizes the following expected loss function

$$\min_{\boldsymbol{u}_t \in \mathcal{U}} \quad \left\{ \mathbb{E}_{\mathbb{P}_t} \left[\ell(\boldsymbol{u}_t, \boldsymbol{x}_t) \right] := \int_{\mathbb{R}^n} \ell(\boldsymbol{u}_t, \boldsymbol{x}_t) \, \mathbb{P}_t(d\boldsymbol{x}_t) \right\}.$$

Note that the objective value is inaccessible since $\mathbb{P}_t$ is unknown, and its evolution is highly dependent on the environment dynamics as well as on the decisions taken. We assume that the

environment is described as an unknown stochastic system

$$\boldsymbol{x}_{t+1} = f(t, \boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{w}_t, \text{ from a given } \boldsymbol{x}_0, \tag{5.1}$$

where the distribution $\mathbb{P}_{t+1}$ of $\boldsymbol{x}_{t+1}$ is determined by the online decision $\boldsymbol{u}_t$, the unknown but measurable environment evolution $f : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, and additive disturbance vectors $\boldsymbol{w}_t \in \mathbb{R}^n$. In particular, let us denote by $\mathbb{W}_t$ the distribution of $\boldsymbol{w}_t$ and make the following assumption.

Assumption 11 (Independent and stationary subGaussian distributions). (1) Random vectors $\boldsymbol{w}_t := (w_{t,1}^\top, \ldots, w_{t,n}^\top)^\top$, $t \in \mathbb{Z}_{\geq 0}$, are time-and-component-wise independent, i.e., $w_{t,i}$ and $w_{k,j}$ are independent, for all $t \neq k$, $i \neq j$, $(t,k) \in \mathbb{Z}_{\geq 0}^2$ and $(i,j) \in \{1, \ldots, n\}$. **(2)** The process $\{\boldsymbol{w}_t\}_t$ is stationary and, for each t, the vector $\boldsymbol{w}_t$ is zero-mean and σ-subGaussian, i.e., for any $a \in \mathbb{R}^n$ we have $\mathbb{E}\left[\exp(a^\top \boldsymbol{w}_t)\right] \leq \exp(\|a\|^2 \sigma^2/2)$.

In this chapter, we aim to propose an effective online optimization and learning algorithm which tracks minimizers of the time-varying, environment-dependent objective function with low regret in high probability. In particular, at each time t, we aim to find an online decision $\boldsymbol{u} := \boldsymbol{u}_t$ that minimizes the loss in the immediate future environment at $t+1$, as follows

$$\begin{aligned} &\min_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}_{\mathbb{P}_{t+1}} [\ell(\boldsymbol{u}, \boldsymbol{x}_{t+1})], \\ &\text{s.t. } \mathbb{P}_{t+1} \text{ is characterized by (5.1).} \end{aligned} \tag{P}$$

Remark 21 (Finite-time horizon version of (P)**).** Problem (P) can be extended to a N-finite-time horizon problem. In this way, at each time t, we aim to find an online decision $\boldsymbol{u} := (\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(N)})$ that minimizes the loss over the next N time steps $\{t+1, \ldots, t+N\}$. In this scenario, we characterize the unknown dynamical environment via a stochastic process $\boldsymbol{x} := (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)})$ under the unknown product distribution $\mathbb{Q} := \mathbb{P}_{t+1} \otimes \cdots \otimes \mathbb{P}_{t+N}$. Similarly, we consider the loss

function $\ell : \mathbb{R}^{mN} \times \mathbb{R}^{nN} \to \mathbb{R}$, $(\boldsymbol{u}, \boldsymbol{x}) \mapsto \ell(\boldsymbol{u}, \boldsymbol{x})$, and solve the following problem online

$$\begin{aligned} &\min_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}_{\mathbb{Q}} [\ell(\boldsymbol{u}, \boldsymbol{x})], \\ &\text{s.t. } \mathbb{Q} \text{ is characterized by (5.1).} \end{aligned}$$

Several types of problems fall under this finite-time horizon formulation, such as data-driven control, stochastic model predictive control, and stochastic state estimation problems. In Chapter 3, we have leveraged the structure of this finite-time horizon formulation (P) for a traffic control problem in which the system dynamics were known. In contrast, in this chapter, we seek to learn the dynamical environment f and make online decisions altogether. We leave the consideration of finite-time horizon problems for future work.

5.3 Online Learning of Unknown System Dynamics

To obtain online solutions of Problem (P), we employ the dynamic ambiguity set $\mathcal{P}_{t+1}$ proposed in Chapter 4. The set $\mathcal{P}_{t+1}$ contains a class of distributions, which is, in high probability, large enough to include the unknown $\mathbb{P}_{t+1}$ under certain conditions. Thus, we can use it to formulate a robust version of the problem at each time instant. In the following section, we leverage the probabilistic characterization $\mathcal{P}_{t+1} := \mathcal{P}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})$ of the distribution $\mathbb{P}_{t+1}$ for online-tractable solutions to (P).

5.4 Tractable Reformulation

This section presents an online-tractable formulation of Problem (P) by leveraging the adaptive ambiguity set $\mathcal{P}_{t+1}$. To achieve this, we first consider the expectation of the loss over the worst-case distribution in $\mathcal{P}_{t+1}$ and reformulate (P) to a problem called (P1). The solution of (P1) provides guarantees on performance of (P). Then, we propose a tractable reformulation (P2) which is equivalent to (P1) under certain conditions.

Formally, let us consider

$$\inf_{\boldsymbol{u}\in\mathcal{U}} \sup_{\mathbb{Q}\in\mathcal{P}_{t+1}(\boldsymbol{\alpha},\boldsymbol{u})} \mathbb{E}_{\mathbb{Q}}[\ell(\boldsymbol{u},\boldsymbol{x})], \tag{P1}$$

where, for a fixed $\boldsymbol{\alpha} := \boldsymbol{\alpha}_t$ and $\boldsymbol{u} := \boldsymbol{u}_t \in \mathcal{U}$, it holds that $\mathbb{P}_{t+1} \in \mathcal{P}_{t+1}(\boldsymbol{\alpha},\boldsymbol{u})$ with high probability. This results in

$$\text{Prob}\left(\mathbb{E}_{\mathbb{P}_{t+1}}[\ell(\boldsymbol{u},\boldsymbol{x})] \leq \sup_{\mathbb{Q}\in\mathcal{P}_{t+1}} \mathbb{E}_{\mathbb{Q}}[\ell(\boldsymbol{u},\boldsymbol{x})]\right) \geq \rho,$$

where the confidence bound $\rho = (1-\beta)\left(1-\exp\left(-\frac{\theta^2 T^2}{2[(2T-1)c\gamma+nc^2]}\right)\right)$, with parameters the same as in (4.10). Notice that, the value ρ increases exponentially to the given $(1-\beta)$ as we increase θ or the data-set sizes T. In practice, these parameters need to be selected empirically, in order to efficiently address the particular problem of interest. We refer reader to Chapter 4, for a guide on how to select these parameters.

The solution $\boldsymbol{u}$ and the objective value of (P1) ensure that, when we select $\boldsymbol{u}$ to be the decision for (P), the expected loss of (P) is no worse than that from (P1) with high probability. The formulation (P1) still requires expensive online computations due to its semi-infinite inner optimization problem. Thus, we propose an equivalent reformulation of (P1) for a class of loss functions as in the following assumption.

Assumption 12 (Lipschitz loss functions). Consider the loss function $\ell : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, $(\boldsymbol{u},\boldsymbol{x}) \mapsto \ell(\boldsymbol{u},\boldsymbol{x})$. There exists a Lipschitz function $L : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ such that for each $\boldsymbol{u} \in \mathbb{R}^m$, it holds that $\|\ell(\boldsymbol{u},\boldsymbol{x}) - \ell(\boldsymbol{u},\boldsymbol{y})\| \leq L(\boldsymbol{u})\|\boldsymbol{x}-\boldsymbol{y}\|$ for any $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n$.

With this, we obtain the following upper bound:

Lemma 14 (An upper bound of (P1)**).** *Let Assumption 12 hold. Then, for each $\boldsymbol{u}$, $\boldsymbol{\alpha}$, β, T and t, we have*

$$\sup_{\mathbb{Q}\in\mathcal{P}_{t+1}(\boldsymbol{\alpha},\boldsymbol{u})} \mathbb{E}_{\mathbb{Q}}[\ell(\boldsymbol{u},\boldsymbol{x})] \leq \mathbb{E}_{\hat{\mathbb{P}}_{t+1}(\boldsymbol{\alpha},\boldsymbol{u})}[\ell(\boldsymbol{u},\boldsymbol{x})] + \hat{\epsilon}(t,T,\beta,\boldsymbol{\alpha},\boldsymbol{u})L(\boldsymbol{u}),$$

where the empirical distribution $\hat{\mathbb{P}}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})$ *and scalar* $\hat{\epsilon}(t, T, \beta, \boldsymbol{\alpha}, \boldsymbol{u})$ *are described as in Chapter 4.*

Proof. By the definition of the ambiguity set, we have that, for any distribution $\mathbb{Q} \in \mathcal{P}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})$

$$d_{\mathrm{W}}(\mathbb{Q}, \hat{\mathbb{P}}_{t+1}) \leq \hat{\epsilon},$$

which is equivalent to

$$\int_{\mathcal{Z}} h(\boldsymbol{x}) \mathbb{Q}(d\boldsymbol{x}) - \int_{\mathcal{Z}} h(\boldsymbol{x}) \hat{\mathbb{P}}_{t+1}(d\boldsymbol{x}) \leq \hat{\epsilon}, \quad \forall h \in \mathcal{L},$$

where $\mathcal{L}$ is the set of functions with Lipschitz constant 1. For a given $\boldsymbol{u}$, let us select h to be

$$h(\boldsymbol{x}) := \frac{\ell(\boldsymbol{u}, \boldsymbol{x})}{L(\boldsymbol{u})},$$

where L is the positive Lipschitz function as in Assumption 12. Substituting h to the above inequality, we have

$$\int_{\mathcal{Z}} \ell(\boldsymbol{u}, \boldsymbol{x}) \mathbb{Q}(d\boldsymbol{x}) - \int_{\mathcal{Z}} \ell(\boldsymbol{u}, \boldsymbol{x}) \hat{\mathbb{P}}_{t+1}(d\boldsymbol{x}) \leq \hat{\epsilon} L(\boldsymbol{u}),$$

or equivalently

$$\mathbb{E}_{\mathbb{Q}}[\ell(\boldsymbol{u}, \boldsymbol{x})] \leq \mathbb{E}_{\hat{\mathbb{P}}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})}[\ell(\boldsymbol{u}, \boldsymbol{x})] + \hat{\epsilon} L(\boldsymbol{u}).$$

As the inequality holds for every $\mathbb{Q} \in \mathcal{P}_{t+1}$, therefore

$$\sup_{\mathbb{Q} \in \mathcal{P}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})} \mathbb{E}_{\mathbb{Q}}[\ell(\boldsymbol{u}, \boldsymbol{x})] \leq \mathbb{E}_{\hat{\mathbb{P}}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})}[\ell(\boldsymbol{u}, \boldsymbol{x})] + \hat{\epsilon}(t, T, \beta, \boldsymbol{\alpha}, \boldsymbol{u}) L(\boldsymbol{u}).$$

□

Next, we claim that the upper bound in Lemma 14 is tight if the following assumption

holds.

Assumption 13 (Convex and gradient-accessible functions). The loss function ℓ is convex in $\boldsymbol{x}$ for each $\boldsymbol{u}$. Further, for each time t with given $\boldsymbol{u} \in \mathcal{U}$ and $\boldsymbol{\alpha} \in \mathbb{R}^p$, there is an environment prediction $\sum_{i=1}^p \alpha_i \xi_k^{(i)}(\boldsymbol{\alpha}, \boldsymbol{u})$ for some $k \in \mathcal{T}$ such that $\nabla_{\boldsymbol{x}} \ell$ exists and $L(\boldsymbol{u}) = \|\nabla_{\boldsymbol{x}} \ell\|$ at $(\boldsymbol{u}, \sum_{i=1}^p \alpha_i \xi_k^{(i)}(\boldsymbol{\alpha}, \boldsymbol{u}))$.

The above statement enables the following theorem.

Theorem 11 (Equivalent reformulation of (P1)**).** *Let Assumptions 12 and 13 hold. Let* Ξ_{t+1} *denote the support of the distribution* $\mathbb{P}_{t+1}$*. Then, if* $\Xi_{t+1} = \mathbb{R}^n$*,* (P1) *is equivalent to the following problem*

$$\min_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}_{\hat{\mathbb{P}}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})} [\ell(\boldsymbol{u}, \boldsymbol{x})] + \hat{\epsilon}(t, T, \beta, \boldsymbol{\alpha}, \boldsymbol{u}) L(\boldsymbol{u}). \tag{P2}$$

Proof. We show this by constructing a distribution in the ambiguity set. By Assumption 13 on convex and gradient-accessible functions, there exist an index $j \in \mathcal{T}$ such that the derivative $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{u}, \boldsymbol{x})$ at $(\boldsymbol{u}, \bar{\boldsymbol{x}}^{(j)})$, $\bar{\boldsymbol{x}}^{(j)} := \sum_{i=1}^p \alpha_i \xi_j^{(i)}(\boldsymbol{\alpha}, \boldsymbol{u})$, satisfies

$$\|\nabla_{\boldsymbol{x}} \ell(\boldsymbol{u}, \bar{\boldsymbol{x}}^{(j)})\| = L(\boldsymbol{u}).$$

Now using this index j, we construct a parameterized distribution as follows

$$\mathbb{Q}(\Delta \boldsymbol{x}) = \frac{1}{T} \sum_{k \in \mathcal{T}, k \neq j} \delta_{\{\sum_{i=1}^p \alpha_i \xi_k^{(i)}(\boldsymbol{\alpha}, \boldsymbol{u})\}} + \frac{1}{T} \delta_{\{\bar{\boldsymbol{x}}^{(j)} + \Delta \boldsymbol{x}\}},$$

where $\Delta \boldsymbol{x} \in \mathbb{R}^n$ with $\|\Delta \boldsymbol{x}\| \leq T\hat{\epsilon}$. By the definition of the ambiguity set and, since the support of the distribution $\mathbb{P}$ is $\Xi_{t+1} = \mathbb{R}^n$, we have $\mathbb{Q}(\Delta \boldsymbol{x}) \in \mathcal{P}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})$.

Next, we quantify the lower bound of the following term

$$\mathbb{E}_{\mathbb{Q}(\Delta \boldsymbol{x})} [\ell(\boldsymbol{u}, \boldsymbol{x})] - \mathbb{E}_{\hat{\mathbb{P}}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})} [\ell(\boldsymbol{u}, \boldsymbol{x})] = \frac{1}{T} \left(\ell(\boldsymbol{u}, \bar{\boldsymbol{x}}^{(j)} + \Delta \boldsymbol{x}) - \ell(\boldsymbol{u}, \bar{\boldsymbol{x}}^{(j)}) \right).$$

By Assumption 13 on the convexity of ℓ on $\boldsymbol{x}$, we have

$$\ell(\boldsymbol{u}, \bar{\boldsymbol{x}}^{(j)} + \Delta\boldsymbol{x}) - \ell(\boldsymbol{u}, \bar{\boldsymbol{x}}^{(j)}) \geq \nabla_{\boldsymbol{x}} \ell(\boldsymbol{u}, \bar{\boldsymbol{x}}^{(j)})^\top \Delta\boldsymbol{x}.$$

Then, by selecting

$$\Delta\boldsymbol{x} := \frac{T\hat{\epsilon}\nabla_{\boldsymbol{x}} \ell(\boldsymbol{u}, \bar{\boldsymbol{x}}^{(j)})}{\|\nabla_{\boldsymbol{x}} \ell(\boldsymbol{u}, \bar{\boldsymbol{x}}^{(j)})\|},$$

we have

$$\nabla_{\boldsymbol{x}} \ell(\boldsymbol{u}, \bar{\boldsymbol{x}}^{(j)})^\top \Delta\boldsymbol{x} = T\hat{\epsilon} L(\boldsymbol{u}).$$

These bounds result in

$$\mathbb{E}_{\mathbb{Q}(\Delta\boldsymbol{x})}[\ell(\boldsymbol{u}, \boldsymbol{x})] - \mathbb{E}_{\hat{\mathbb{P}}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})}[\ell(\boldsymbol{u}, \boldsymbol{x})] \geq \hat{\epsilon} L(\boldsymbol{u}).$$

As $\mathbb{Q}(\Delta\boldsymbol{x}) \in \mathcal{P}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})$, therefore

$$\sup_{\mathbb{Q} \in \mathcal{P}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})} \mathbb{E}_{\mathbb{Q}}[\ell(\boldsymbol{u}, \boldsymbol{x})] \geq \mathbb{E}_{\hat{\mathbb{P}}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})}[\ell(\boldsymbol{u}, \boldsymbol{x})] + \hat{\epsilon} L(\boldsymbol{u}).$$

Finally, with Assumption 12 on Lipschitz loss functions and Lemma 14 on an upper bound of (P1), we equivalently write Problem (P1) as

$$\inf_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}_{\hat{\mathbb{P}}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u})}[\ell(\boldsymbol{u}, \boldsymbol{x})] + \hat{\epsilon}(t, T, \beta, \boldsymbol{\alpha}, \boldsymbol{u}) L(\boldsymbol{u}),$$

which is the Problem (P2). □

Notice that the tractability of solutions to (P2) now depend on: 1) the choice of the loss function ℓ and the associated Lipschitz function L, and 2) the decision space $\mathcal{U}$. To be able to further analyze (P2) and further evaluate Assumption 13 on gradient-accessible functions, we will impose further structure on the system as follows:

Assumption 14 (Locally Lipschitz, control-affine environment and predictors). *The environ-*

ment f is locally Lipschitz in $(t, \boldsymbol{x}, \boldsymbol{u})$ and affine in $\boldsymbol{u}$, i.e.,

$$f(t, \boldsymbol{x}, \boldsymbol{u}) := f_1(t, \boldsymbol{x}) + f_2(t, \boldsymbol{x})\boldsymbol{u},$$

for some unknown $f_1 : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^n$, $f_2 : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^{n \times m}$, $\boldsymbol{u} \in \mathcal{U}$ and $t \in \mathbb{Z}_{\geq 0}$. Similarly, for each $i \in \{1, \ldots, p\}$, the predictor $f^{(i)}$ is selected to be

$$f^{(i)}(t, \boldsymbol{x}, \boldsymbol{u}) := f_1^{(i)}(t, \boldsymbol{x}) + f_2^{(i)}(t, \boldsymbol{x})\boldsymbol{u},$$

for some given locally Lipschitz functions $f_1^{(i)}$ and $f_2^{(i)}$.

Assumption 15 (Convex decision oracle). *The set $\mathcal{U}$ is convex and compact, and, in addition, the projection operation of any $\boldsymbol{u} \in \mathbb{R}^m$ onto $\mathcal{U}$, $\Pi_{\mathcal{U}}(\boldsymbol{u})$, admits $O(1)$ computation complexity.*

Remark 22 ($\mathcal{U}$ examples). *Examples of $\mathcal{U}$ include the following: 1) the non-negative orthant as $\{\boldsymbol{u} \mid \boldsymbol{u} \geq \boldsymbol{0}_m\}$, 2) an m-cell as $\{\boldsymbol{u} \mid \underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}}\}$, for some constant vectors $\underline{\boldsymbol{u}}$ and $\overline{\boldsymbol{u}}$, 3) a unit simplex as $\{\boldsymbol{u} \mid \boldsymbol{u}^\top \boldsymbol{1}_m = 1,\ \boldsymbol{u} \geq \boldsymbol{0}_m\}$, or 4) a ball $\{\boldsymbol{u} \mid \|\boldsymbol{u}\| \leq 1\}$.*

For simplicity of the discussion, we rewrite (P2) as

$$\min_{\boldsymbol{u} \in \mathcal{U}} G(t, \boldsymbol{u}),$$

where G represents the objective function of (P2), depending on ℓ, L and $\mathcal{P}_{t+1}$. Then, Assumption 14 allows an explicit expression of G w.r.t. $\boldsymbol{u} := \boldsymbol{u}_t$ and Assumption 15 characterizes the convex feasible set of (P2). Note that $G(t, \boldsymbol{u})$ is locally Lipschitz in t.[1]

5.5 Two Application Scenarios

We consider two scenarios in form of (P2): 1) an optimal control problem under the uncertainty; 2) an online resource allocation problem with a switch. These problems leverage

[1]This can be verified by the local Lipschitz condition on $f^{(i)}$, ℓ, and finite composition of local Lipschitz functions are locally Lipschitz.

the probabilistic characterization of the environment and common loss functions ℓ. Then, we propose an online algorithm to achieve tractable solutions with a probabilistic regret bound in the next section.

Problem 1: (Optimal control under uncertainty) We consider a problem in form (P), where the environment is a system to be optimally controlled. In particular, we employ the following separable loss function

$$\ell(\boldsymbol{u}, \boldsymbol{x}) := \ell_1(\boldsymbol{u}) + \ell_2(\boldsymbol{x}), \quad \ell_1 : \mathbb{R}^m \to \mathbb{R},\ \ell_2 : \mathbb{R}^n \to \mathbb{R},$$

with ℓ_1 the cost for the immediate control and ℓ_2 the optimal cost-to-go function. We assume that both ℓ_1 and ℓ_2 are convex, and in addition, ℓ_2 is Lipschitz continuous with a constant $\mathrm{Lip}(\ell_2)$, resulting in $L(\boldsymbol{u}) \equiv \mathrm{Lip}(\ell_2)$. Then, by selecting the ambiguity radius of $\mathcal{P}_{t+1}$ as in (4.9), the objective function of (P2) is the following

$$G(t, \boldsymbol{u}) = \ell_1(\boldsymbol{u}) + \frac{1}{T} \sum_{k \in \mathcal{T}} \ell_2(\boldsymbol{p}_{k,t}) + \mathrm{Lip}(\ell_2)\epsilon + \frac{\gamma\, \mathrm{Lip}(\ell_2)}{T} \sum_{i=1}^{p} \sum_{k \in \mathcal{T}} \|H_k^{(i)}\|,$$

where $\boldsymbol{p}_{k,t}$ and $H_k^{(i)}$ are affine in $\boldsymbol{u}$, for each i, k, as

$$\boldsymbol{p}_{k,t} := \sum_{i=1}^{p} \alpha_i \left(f_1^{(i)}(t, \hat{\boldsymbol{x}}_t) - f^{(i)}(k, \hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) \right) + \hat{\boldsymbol{x}}_{k+1} + \left(\sum_{i=1}^{p} \alpha_i f_2^{(i)}(t, \hat{\boldsymbol{x}}_t) \right) \boldsymbol{u},$$

$$H_k^{(i)}(\boldsymbol{u}) := f^{(i)}(k, \hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) - f_1^{(i)}(t, \hat{\boldsymbol{x}}_t) - f_2^{(i)}(t, \hat{\boldsymbol{x}}_t)\boldsymbol{u},$$

and parameters $\boldsymbol{\alpha}$, ϵ and γ are as in Theorem 9. Notice that the objective function G is convex in $\boldsymbol{u}$ and therefore online problems (P2) are tractable. In addition, if ℓ_2 has a constant gradient almost everywhere, then Assumption 13 on accessible gradients holds and (P2) is equivalent to (P1).

Problem 2: (Online resource allocation) We consider an online resource allocation problem with a switch, where a decision maker aims to make online resource allocation decisions

in an uncertain environment. This problem is in form (P) and its objective is

$$\ell(\boldsymbol{u}, \boldsymbol{x}) = \max\{0, 1 - \langle \boldsymbol{u}, \phi(\boldsymbol{x})\rangle\}, \quad \phi : \mathbb{R}^n \to \mathbb{R}^m,$$

where ϕ is an affine feature map selected in advance. The decision maker updates the decision $\boldsymbol{u}$ online when $\langle \boldsymbol{u}, \phi(\boldsymbol{x})\rangle < 1$, otherwise switches off. Notice that this type of objective functions appears in many classification problems. In particular, we assume that the environment f is independent from the allocation variable, i.e., $f_2 \equiv 0$. See Example 5.7.2 for a more explicit problem formulation involving online resource allocation with an assignment switch.

Then, problem (P2) has the objective function

$$G(t, \boldsymbol{u}) = \frac{1}{T} \sum_{k \in \mathcal{T}} \max\{0, 1 - \langle \boldsymbol{u}, \phi(\boldsymbol{p}_{k,t})\rangle\} + q_t L(\boldsymbol{u}),$$

with the time-dependent parameters

$$\boldsymbol{p}_{k,t} = \hat{\boldsymbol{x}}_{k+1} + \sum_{i=1}^{p} \alpha_i \left(f_1^{(i)}(t, \hat{\boldsymbol{x}}_t) - f_1^{(i)}(k, \hat{\boldsymbol{x}}_k) \right), \ \forall\ k,\ t,$$
$$q_t = \epsilon + \frac{\gamma}{T} \sum_{i=1}^{p} \sum_{k \in \mathcal{T}} \| f_1^{(i)}(k, \hat{\boldsymbol{x}}_k) - f_1^{(i)}(t, \hat{\boldsymbol{x}}_t) \|, \ \forall\ t,$$

where $\boldsymbol{\alpha}$, ϵ and γ are as in Theorem 9. We characterize the function $L(\boldsymbol{u})$ by subgradients of the loss function ℓ.

Lemma 15 (Quantification of L). *Consider $\ell(\boldsymbol{u}, \boldsymbol{x}) := \max\{0, 1 - \langle \boldsymbol{u}, \phi(\boldsymbol{x})\rangle\}$, where $\phi(\boldsymbol{x})$ is differentiable in $\boldsymbol{x}$. Then, the function $L(\boldsymbol{u})$ is*

$$L(u) = \sup_{\boldsymbol{g} \in \partial_{\boldsymbol{x}} \ell(\boldsymbol{u}, \boldsymbol{x}),\ \boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{g}\|,$$

where the set $\partial_{\boldsymbol{x}}\ell(\boldsymbol{u},\boldsymbol{x})$ contains all the subgradients of ℓ at $\boldsymbol{x}$, given any $\boldsymbol{u}$ in advance, i.e.,

$$\partial_{\boldsymbol{x}}\ell(\boldsymbol{u},\boldsymbol{x}) := h(\boldsymbol{x},\boldsymbol{u}) \cdot \frac{\partial \phi}{\partial \boldsymbol{x}}(\boldsymbol{x})\boldsymbol{u},$$

where

$$h(\boldsymbol{x},\boldsymbol{u}) = \begin{cases} -1, & \text{if } \langle \boldsymbol{u}, \phi(\boldsymbol{x}) \rangle < 1 \\ 0, & \text{if } \langle \boldsymbol{u}, \phi(\boldsymbol{x}) \rangle > 1 \cdot \\ [-1,0], & \text{o.w.} \end{cases}$$

In particular, if $\phi(\boldsymbol{x}) := J\boldsymbol{x}$ for some matrix J, then $L(u) = \|J^\top \boldsymbol{u}\|$. If $\boldsymbol{x}$ is contained in a compact set X, then $L(u) = \mathrm{Lip}(\phi)\|\boldsymbol{u}\|$,where $\mathrm{Lip}(\phi)$ is the Lipschitz constant of ϕ on X.

Proof. This is the direct application of the definition of the local Lipschitz condition. □

Lemma 15 indicates that, given a properly selected feature mapping ϕ, the objective G is convex in $\boldsymbol{u}$ and therefore online problems (P2) are convex and tractable. In addition, if ϕ is a linear map almost everywhere, then Assumption 13 on accessible gradients holds and (P2) is equivalent to (P1).

5.6 Online Algorithms

The online convex problems (P2) are non-smooth due to the normed regularization terms in G. To achieve fast, online solutions, we propose a two-step procedure. First, we adapt an approach in [9,99] to obtain a smooth version of (P2), called (P2′). Then, we extend the *Nesterov's accelerated-gradient* method [100]—known to achieve an optimal first-order convergence rate for smooth and offline convex problems—to solve the problem (P2′). Finally, we quantify the dynamic regret [141] of online decisions w.r.t. solutions of (P1) in probability.

Step 1: (Smooth approximation of (P2)**)** To simplify the discussion, let us use the generic notation $F : \mathcal{U} \to \mathbb{R}$ for a convex and potentially non-smooth function, which can represent any particular component of the objective function $G(t,\boldsymbol{u})$ of (P2) at time t.

Definition 2 (Smoothable function). *We call a convex function $F(\boldsymbol{u})$ smoothable on $\mathcal{U}$ if there exists $a > 0$ such that, for every $\mu > 0$, there is a continuously differentiable convex function $F_\mu : \mathcal{U} \to \mathbb{R}$ satisfying*

(1) $F_\mu(\boldsymbol{u}) \le F(\boldsymbol{u}) \le F_\mu(\boldsymbol{u}) + a\mu$, for all $\boldsymbol{u} \in \mathcal{U}$.

(2) There exists $b > 0$ such that F_μ has a Lipschitz gradient over $\mathcal{U}$ with Lipschitz constant b/μ, i.e.,

$$\|\nabla F_\mu(\boldsymbol{u}_1) - \nabla F_\mu(\boldsymbol{u}_2)\| \le \frac{b}{\mu}\|\boldsymbol{u}_1 - \boldsymbol{u}_2\|, \ \forall \ \boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathcal{U}.$$

To obtain a smooth approximation F_μ of F, we follow the *Moreau proximal approximation* technique [9], described as in the following lemma.

Lemma 16 (Moreau-Yosida approximation). *Given a convex function $F : \mathcal{U} \to \mathbb{R}$ and any $\mu > 0$, let us denote by $\partial F(\boldsymbol{u})$ the set of subgradients of F at $\boldsymbol{u}$, respectively. Let $D := \sup_{g \in \partial F(\boldsymbol{u}), \boldsymbol{u} \in \mathcal{U}} \|g\|^2 < +\infty$. Then, F is smoothable with parameters $(a, b) := (D/2, 1)$, where the smoothed version $F_\mu : \mathcal{U} \to \mathbb{R}$ is the Moreau approximation:*

$$F_\mu(\boldsymbol{u}) := \inf_{z \in \mathcal{U}} \left\{ F(z) + \frac{1}{2\mu}\|z - \boldsymbol{u}\|^2 \right\}, \ \boldsymbol{u} \in \mathcal{U}.$$

In addition, if F is M-strongly convex with some $M > 0$, then F_μ is $M/(1 + \mu M)$-strongly convex. And further, the minimization of $F(\boldsymbol{u})$ over $\boldsymbol{u} \in \mathcal{U}$ is equivalent to that of $F_\mu(\boldsymbol{u})$ over $\boldsymbol{u} \in \mathcal{U}$ in the sense that the set of minimizers of two problems are the same.

Proof. First, we have

$$F_\mu(\boldsymbol{u}) \le F(\boldsymbol{u}) + \frac{1}{2\mu}\|\boldsymbol{u} - \boldsymbol{u}\|^2 = F(\boldsymbol{u}), \ \forall \ \boldsymbol{u} \in \mathcal{U}.$$

Then, we compute

$$\begin{aligned} F(\boldsymbol{u}) - F_\mu(\boldsymbol{u}) &= \sup_{z \in \mathcal{U}} \left\{ F(\boldsymbol{u}) - F(z) - \frac{1}{2\mu}\|z - \boldsymbol{u}\|^2 \right\}, \\ &\leq \sup_{z \in \mathcal{U}} \left\{ \boldsymbol{g}(\boldsymbol{u})^\top (\boldsymbol{u} - z) - \frac{1}{2\mu}\|z - \boldsymbol{u}\|^2 \right\}, \\ &\leq \frac{\mu}{2} \boldsymbol{g}(\boldsymbol{u})^\top \boldsymbol{g}(\boldsymbol{u}) \leq \frac{D}{2}\mu, \end{aligned}$$

where the equality comes from the definition of $F_\mu(\boldsymbol{u})$, the first inequality leverages the convexity of F, the second one applies the achieved optimizer $z^\star = \boldsymbol{u} - \mu \boldsymbol{g}(\boldsymbol{u})$, and the last one is from the boundedness of subgradients.

Further, given F as described, it is well-known (see, e.g., [7, Proposition 12.15] for details) that F_μ is convex and continuously differentiable where its gradient ∇F_μ is Lipschitz continuous with constant $1/\mu$. In addition, the minimizer $z^\star(\boldsymbol{u})$ of F_μ is achievable and unique, resulting in an explicit gradient expression of F_μ as follows

$$\nabla F_\mu(\boldsymbol{u}) = \frac{1}{\mu}(\boldsymbol{u} - z^\star(\boldsymbol{u})).$$

In addition, we claim that, if F is M-stronly convex, F_μ is $M/(1+\mu M)$-strongly convex, following [65, Theorem 2.2]. Finally, we equivalently write the minimization problem as follows

$$\begin{aligned} \min_{\boldsymbol{u} \in \mathcal{U}} F_\mu(\boldsymbol{u}) &= \min_{\boldsymbol{u} \in \mathcal{U}} \min_{z \in \mathcal{U}} \left\{ F(z) + \frac{1}{2\mu}\|z - \boldsymbol{u}\|^2 \right\} \\ &= \min_{z \in \mathcal{U}} \min_{\boldsymbol{u} \in \mathcal{U}} \left\{ F(z) + \frac{1}{2\mu}\|z - \boldsymbol{u}\|^2 \right\} \\ &= \min_{z \in \mathcal{U}} F(z), \end{aligned}$$

where the first line applies the achievability of the minimizer of the problem that defines F_μ, the second line switches the minimization operators, the third line applies the fact that $\boldsymbol{u} = z$ solves the inner minimization problem. This concludes that any $\boldsymbol{u}$ that minimizes F_μ also minimizes F, and vice versa. □

From the definition of the smoothable function, we know that: 1) a positive linear combination of smoothable functions is smoothable[2], and 2) the composition of a smoothable function with a linear transformation is smoothable[3]. These properties enable us to smooth each component of G, i.e., h, ℓ_1, ℓ_2 and $\|\cdot\|$, which results in a smooth approximation of (P2) via the corresponding G_μ as follows

$$\min_{\boldsymbol{u}\in\mathcal{U}} G_\mu(t,\boldsymbol{u}). \tag{P2$'$}$$

Note that G_μ is locally Lipschitz and minimizers of (P2$'$) are that of (P2). We provide in the following lemma explicit expressions of (P2$'$) for the two application scenarios.

Lemma 17 (Examples of (P2$'$)**).**

Problem 1: *Consider the following loss function*

$$\ell(\boldsymbol{u},\boldsymbol{x}) := \frac{1}{2}\|\boldsymbol{u}\|^2 + F_\mu(\boldsymbol{x}), \;\; \textit{given some } \mu > 0,$$

where $F_\mu : \mathbb{R}^n \to \mathbb{R}$ is the smoothed ℓ_2-norm function defined as in Appendix C.1, with $\mathrm{Lip}(F_\mu) = 1$. *Then, the objective function $G_\mu(t,\boldsymbol{u})$ is*

$$\frac{1}{2}\|\boldsymbol{u}\|^2 + \frac{1}{T}\sum_{k\in\mathcal{T}} F_\mu(\boldsymbol{p}_{k,t}) + \epsilon + \frac{\gamma}{T}\sum_{i=1}^{p}\sum_{k\in\mathcal{T}} F_\mu(H_k^{(i)}),$$

where $\boldsymbol{p}$, H are affine in $\boldsymbol{u}$, defined as in Section 5.5. In addition, we have the smoothing parameter of $G_\mu(t,\boldsymbol{u})$, $(a,b) := ((1+p\gamma)/2, \mu + s_0 + \gamma\sum_i s_i)$, where

$$s_0 = \sigma_{\max}\left(\left(\sum_{i=1}^{p}\alpha_i f_2^{(i)}(t,\hat{\boldsymbol{x}}_t)\right)^\top\left(\sum_{i=1}^{p}\alpha_i f_2^{(i)}(t,\hat{\boldsymbol{x}}_t)\right)\right),$$

[2]If F_1 is smoothable with parameter (a_1,b_1) and F_2 with parameter (a_2,b_2), then $c_1F_1 + c_2F_2$ is smoothable with the parameter $(c_1a_1 + c_2a_2, c_1b_1 + c_2b_2)$, for any $c_1, c_2 \geq 0$.

[3]Let $A : \mathcal{U} \to \mathcal{X}$ be a linear transformation and let $\boldsymbol{b} \in \mathcal{X}$. Let $\ell : \mathcal{X} \to \mathbb{R}$ be a smoothable function with the parameter (a,b). Then, the function $F : \mathcal{U} \to \mathbb{R}$, $\boldsymbol{u} \mapsto \ell(A\boldsymbol{u} + \boldsymbol{b})$ is smoothable with the parameter $(a, b\|A\|^2)$, where $\|A\| := \max_{\|\boldsymbol{u}\|=1}\|A\boldsymbol{u}\|$. If $\mathcal{X} = \mathbb{R}$, the norm $\|A\|$ becomes ℓ_∞ norm.

with $\sigma_{\max}$ denoting the maximum singular value of the matrix, and

$$s_i = \sigma_{\max}\left(f_2^{(i)}(t,\hat{\boldsymbol{x}}_t)^\top f_2^{(i)}(t,\hat{\boldsymbol{x}}_t)\right),\ i \in \{1,\dots,p\}.$$

Problem 2: *Let us select the feature map ϕ to be the identity map with the dimension $m = n$, and consider*

$$\ell(\boldsymbol{u},\boldsymbol{x}) := \max\{0, 1 - \langle \boldsymbol{u},\boldsymbol{x}\rangle\}, \quad \textit{with } L(u) = \|\boldsymbol{u}\|,$$

resulting in

$$G_\mu(t,\boldsymbol{u}) = \frac{1}{T}\sum_{k\in\mathcal{T}} F_\mu^{\mathrm{S}}(\langle \boldsymbol{u}, \boldsymbol{p}_{k,t}\rangle) + q_t F_\mu(\boldsymbol{u}),$$

where $\mu > 0$, parameters $\boldsymbol{p}$, q are as in Section 5.5, and functions F_μ^{S} and F_μ are the smoothed switch function and ℓ_2-norm function as in Appendix C.1, respectively. Note that G_μ has the smoothing parameter $(a,b) := ((1+q_t)/2, q_t + 1/T\sum_{k\in\mathcal{T}}\|\boldsymbol{p}_{k,t}\|_\infty^2)$.

Step 2: (Solution to (P2′) **as a dynamical system)** To solve (P2′) online, we propose a dynamical system extending the Nesterov's accelerated-gradient method by adapting gradients of the time-varying objective function. In particular, let $\boldsymbol{u}_t$, $t \in \mathbb{Z}_{\geq 0}$, be solutions of (P2′) and let us consider the solution system with some $\boldsymbol{u}_0 \in \mathcal{U}$ and $\boldsymbol{y}_0 = \boldsymbol{u}_0$, as

$$\begin{aligned} \boldsymbol{u}_{t+1} &= \Pi_{\mathcal{U}}(\boldsymbol{y}_t - \varepsilon_t \nabla G_\mu(t,\boldsymbol{y}_t)), \\ \boldsymbol{y}_{t+1} &= \boldsymbol{u}_{t+1} + \eta_t(\boldsymbol{u}_{t+1} - \boldsymbol{u}_t), \end{aligned} \tag{5.2}$$

where $\varepsilon_t \leq \mu/b_t$ with positive parameters μ and $b_t := b$ being those define $G_\mu(t,\boldsymbol{u})$. We denote by ∇G_μ the derivative of G_μ w.r.t. its second argument and denote by $\Pi_{\mathcal{U}}(\boldsymbol{y})$ the projection of $\boldsymbol{y}$ onto $\mathcal{U}$ as in Assumption 15 on convex decision oracle. We derive the gradient function ∇G_μ as in Appendix C.2 and select the moment coefficient η_t as in Appendix C.3. In the following, we leverage Appendix C.3 on the stability analysis of the solution system (5.2) for a regret bound between online decisions and optimal solutions of (P1).

Theorem 12 (Probabilistic regret bound of (P1)**).** *Given any $t \geq 2$, let us denote by $\boldsymbol{u}_t$ and $\boldsymbol{u}_t^\star$ the decision generated by* (5.2) *and an optimal solution which solves the online Problem* (P1)*, respectively. Consider the dynamic regret to be the difference of the cost expected to incur if we implement $\boldsymbol{u}_t$ instead of $\boldsymbol{u}_t^\star$, defined as*

$$R_t := \mathbb{E}_{\mathbb{P}_{t+1}}\left[\ell(\boldsymbol{u}_t, \boldsymbol{x})\right] - \mathbb{E}_{\mathbb{P}_{t+1}}\left[\ell(\boldsymbol{u}_t^\star, \boldsymbol{x})\right].$$

Then, the regret R_t is bounded in probability as follows

$$\text{Prob}\left(R_t \leq \frac{4W_t}{(t+2)^2} + TF_t + a\mu + L(\boldsymbol{u}_t^\star)\hat{\epsilon}\right) \geq \rho,$$

where W_t depends on the system state at time $t-T$, and F_t depends on the variation of the optimal objective values in $\mathcal{T}$, i.e.,

$$F_t = \max_{k \in \mathcal{T}} \left\{|G_{k+1}^\star - G_k^\star|\right\} + \bar{L},$$

where $G_k^\star := G(k, \boldsymbol{u}_k^\star)$ is the optimal objective value of (P2)*, or equivalently that of* (P1)*. Further, $\bar{L}$ is the Lipschitz constant of G w.r.t. time, and the rest of the parameters are the same as before.*

Proof. Let us consider the solution system (5.2). At each time t, let us select $\varepsilon := \varepsilon_t = 1/\text{Lip}(G_\mu)$, or equivalently, μ/b with $b = \max_{k \in \mathcal{T}} b_k$. Let η_t satisfy

$$\delta_{-1} = 1,\ \delta_{t+1} := \frac{1 + \sqrt{1 + 4\delta_t^2}}{2},\ \eta_t := \frac{\delta_{t-1} - 1}{\delta_t}.$$

Then, by Theorem 15 with $t \geq 2$, the following holds

$$G_\mu(t, \boldsymbol{u}_t) - G_\mu(t, \boldsymbol{u}_t^\star) \leq \frac{4W_t}{(t+2)^2} + TF_t, \tag{5.3}$$

where $\boldsymbol{u}_t^\star$ is a solution to (P2′), $T = \min\{t-1, T_0\}$ with some horizon parameter $T_0 \in \mathbb{Z}_{>0}$, the time-varying parameter W_t depends on the initial objective discrepancy and the accumulated

energy storage in the considered time horizon $\mathcal{T}$, and F_t is the variation bound of the optimal objective values in $\mathcal{T}$. Specifically, we have

$$F_t = \max_{k \in \mathcal{T}} \left\{ |G_\mu(k+1, \boldsymbol{u}_{k+1}^\star) - G_\mu(k, \boldsymbol{u}_k^\star)| \right\} + \bar{L},$$

with $\bar{L}$ the Lipschitz constant of $G_\mu(t, \boldsymbol{u}_t)$ w.r.t. time t. Let us consider the storage function $V_t(z_t) := {z_t}^\top H_t z_t$, where $z_t := (\boldsymbol{u}_t - \boldsymbol{u}_t^\star,\ \boldsymbol{u}_{t-1} - \boldsymbol{u}_{t-1}^\star,\ \boldsymbol{u}_t^\star - \boldsymbol{u}_{t-1}^\star)$ and

$$H_t := \frac{1}{2\varepsilon_{t-1}} \begin{bmatrix} \delta_{t-1} \\ 1 - \delta_{t-1} \\ \delta_{t-1} \end{bmatrix} \begin{bmatrix} \delta_{t-1} & 1 - \delta_{t-1} & \delta_{t-1} \end{bmatrix} \succeq 0.$$

Then we have

$$\begin{aligned} W_t = V_{t-T}(z_{t-T}) - V_t(z_t) - \sum_{k \in \mathcal{T}} (1 - \frac{\varepsilon_{k-1}}{\varepsilon_k}) V_k(z_k) \\ + (t - T - 1 + \delta_0)^2 (f_{t-T}(\boldsymbol{x}_{t-T}) - f_{t-T}(\boldsymbol{x}_{t-T}^\star)), \end{aligned}$$

where the first two term is the energy decrease in the horizon $\mathcal{T}$; the third sum term indicates the instantaneous energy change, which depends on the online, estimated Lipschitz constant; the last term depends on the goodness of the initial decision at the beginning of the current $\mathcal{T}$. Note how the selection of ε_t and T affects G_t. In the most conservative scenario, we select $\varepsilon_t := \min\{\varepsilon_{t-1}, \mu/b_t\}$ and $T_0 = \infty$, which results in a constant upper bound of W_t as follows

$$W_t \leq V_1(z_1) + \delta_0^2 (f_1(\boldsymbol{x}_1) - f_1(\boldsymbol{x}_1^\star)),$$

therefore, in this case, the bound (5.3) essentially depends on the growing term $(t-1)F_t$. A less conservative way is to use moving horizon strategy, with $\varepsilon_t := \min\{\varepsilon_{t-1}, \mu/b_t\}$ but a finite T_0.

Then, as t is sufficiently large, we have

$$W_t \leq V_{t-T}(\boldsymbol{z}_{t-T}) + t^2(f_{t-T}(\boldsymbol{x}_{t-T}) - f_{t-T}(\boldsymbol{x}^\star_{t-T})),$$

where, in this case, the bound (5.3) essentially depends on F_t and $f_{t-T}(\boldsymbol{x}_{t-T}) - f_{t-T}(\boldsymbol{x}^\star_{t-T})$.

Now, we consider for any $t \geq 2$. By Definition 2, there exists a constant $a > 0$ such that

$$G(t, \boldsymbol{u}_t) - a\mu \leq G_\mu(t, \boldsymbol{u}_t),$$

and by Lemma 16, we have that $\boldsymbol{u}^\star_t$ is a minimizer of (P2′) if and only if it is that of (P2), and

$$G_\mu(t, \boldsymbol{u}^\star_t) \equiv G(t, \boldsymbol{u}^\star_t).$$

This results in

$$G(t, \boldsymbol{u}_t) - G(t, \boldsymbol{u}^\star_t) \leq \frac{4W_t}{(t+2)^2} + TF_t + a\mu, \tag{5.4}$$

with an equivalent expression of F_t as

$$F_t = \max_{k \in \mathcal{T}} \left\{ |G^\star_{k+1} - G^\star_k| \right\} + \bar{L},$$

where $G^\star_k := G(k, \boldsymbol{u}^\star_k)$ is the optimal objective value of (P2) or, later we see, equivalent to that of (P1).

Next, by Theorem 11 on the equivalence of (P1) and (P2), $\boldsymbol{u}^\star_t$ is a minimizer of (P2) if and only if it is also that of (P1), and

$$G(t, \boldsymbol{u}^\star_t) \equiv \sup_{\mathbb{Q} \in \mathcal{P}_{t+1}(\boldsymbol{\alpha}, \boldsymbol{u}^\star_t)} \mathbb{E}_{\mathbb{Q}} \left[\ell(\boldsymbol{u}^\star_t, \boldsymbol{x}) \right].$$

By Assumption 12 on local Lipschitz of ℓ and the Wasserstein metric presentation from

Kantorovich and Rubinstein [59, 71], for every $\boldsymbol{u}$, we have

$$\sup_{\mathbb{Q}\in\mathcal{P}_{t+1}(\boldsymbol{\alpha},\boldsymbol{u})} \mathbb{E}_{\mathbb{Q}}\left[\ell(\boldsymbol{u},\boldsymbol{x})\right] \leq \mathbb{E}_{\mathbb{P}_{t+1}}\left[\ell(\boldsymbol{u},\boldsymbol{x})\right] + L(\boldsymbol{u})\hat{\epsilon},$$

where $\hat{\epsilon}$ is selected as in (4.9). Take $\boldsymbol{u} := \boldsymbol{u}_t^\star$, we have

$$G(t,\boldsymbol{u}_t^\star) \leq \mathbb{E}_{\mathbb{P}_{t+1}}\left[\ell(\boldsymbol{u}_t^\star,\boldsymbol{x})\right] + L(\boldsymbol{u}_t^\star)\hat{\epsilon}.$$

Further, as in Section 5.4, we claim that, Problem (P1) provides a high probabilistic bound for the objective of (P), resulting in

$$\text{Prob}\left(\mathbb{E}_{\mathbb{P}_{t+1}}\left[\ell(\boldsymbol{u}_t,\boldsymbol{x})\right] \leq G(t,\boldsymbol{u}_t)\right) \geq \rho,$$

with

$$\rho = (1-\beta)\left(1-\exp\left(-\frac{(nc-\gamma)^2T^2}{2\left[(2T-1)c\gamma+nc^2\right]}\right)\right).$$

We summarize the above two inequalities and substitute them into (5.4), resulting in

$$\mathbb{E}_{\mathbb{P}_{t+1}}\left[\ell(\boldsymbol{u}_t,\boldsymbol{x})\right] - \mathbb{E}_{\mathbb{P}_{t+1}}\left[\ell(\boldsymbol{u}_t^\star,\boldsymbol{x})\right] \leq \frac{4W_t}{(t+2)^2} + TF_t + a\mu + L(\boldsymbol{u}_t^\star)\hat{\epsilon},$$

with the probability at least ρ, holds for any $t \geq 2$. □

Theorem 12 quantifies the dynamic regret of online decisions $\boldsymbol{u}$ w.r.t. solutions to (P1) in high probability. Notice that, the regret bound is dominated by terms: TF_t, $a\mu$ and $L(\boldsymbol{u}_t^\star)\hat{\epsilon}$, which mainly depend on three factors: the data-driven parameters ε, η and μ of the solution system (5.2), the variation F_t over optimal objective values, and the parameters T, β, γ and $\hat{\epsilon}$ that are related to the environment learning. In practice, a small regret bound is determined by 1) an effective learning procedure which contributes to small $\hat{\epsilon}$; 2) a proper selection of the loss function ℓ which results in smoothing procedure with a small parameter $a\mu$; and 3) the problem

structure leading to small variations F_t of the optimal objectives values.

Online Procedure: Our online algorithm is summarized in the Algorithmic table 7.

Online Optimization and Learning 7. Opal($\mathcal{I}$)

1: Select $\{f^{(i)}\}_i$, ℓ, β, $\mathcal{U}$, $\boldsymbol{u}_0$, μ, and $t = 1$;
2: **repeat**
3: Update data set $\mathcal{I} := \mathcal{I}_t$;
4: Compute $\boldsymbol{\alpha} := \boldsymbol{\alpha}_t$ as in (4.6);
5: Select $\hat{\mathbb{P}}_{t+1}$ as in (4.5) and $\hat{\epsilon} := \hat{\epsilon}_t$ as in (4.9);
6: Run dynamical system (5.2) for $\boldsymbol{u} := \boldsymbol{u}_t$;
7: Apply $\boldsymbol{u}$ to (P) with the regret guarantee;
8: $t \leftarrow t + 1$;
9: **until** time t stops.

5.7 Case Studies

In this section, we apply our algorithm to an optimal control problem which stabilizes an uncertain periodic system to certain desired period. Then, we consider a resource allocation problem in which a decision maker aims to make online decisions while learning its dynamical environment.

5.7.1 Study 1: Optimal Control of an Uncertain Periodic System

We consider a periodic system which is subject to the uncertainty, and our goal is to make an one-step prediction of the system state using the ambiguity set $\mathcal{P}_{t+1}$. In particular, at time t, we consider a periodic system with the state $\boldsymbol{x} \in \mathbb{R}^2$, control $\boldsymbol{u} \in \mathbb{R}^2$ and uncertainty $\boldsymbol{w} \in \mathbb{R}^2$, characterized by

$$\boldsymbol{x}^+ = A(\boldsymbol{x})\boldsymbol{x} + h\boldsymbol{u} + h\boldsymbol{w}, \text{ with } \boldsymbol{x}_0 \in \mathbb{R}^2, \tag{5.5}$$

$$A(\boldsymbol{x}) := \begin{pmatrix} 1 + a_0 h(1 - \boldsymbol{x}^\top \boldsymbol{x}) & b_0 h \\ -b_0 h & 1 + a_0 h(1 - \boldsymbol{x}^\top \boldsymbol{x}) \end{pmatrix},$$

where the state at the next time step is denoted by $\boldsymbol{x}^+$, the parameter $a_0 > 0$ governs the rate of convergence to the limit cycle of the system and $b_0 > 0$ determines the period of the system. Note

that $h = 10^{-3}$ and the period of the system is $2\pi/b_0$. In particular, we assume the true periodic system has the parameter $a_0 = 0.1$, $b_0 = 0.5\pi$ and the random vector $\boldsymbol{w}$ is zero-mean subGaussian with $\sigma = 1$. Further, we assume the parameter a_0 is known while our goal is to learn the period of the system in real time (the parameter b_0). Therefore, we propose the following predictors

$$f^{(i)}(\boldsymbol{x}, \boldsymbol{u}) = A^{(i)}(\boldsymbol{x})\boldsymbol{x} + h\boldsymbol{u}, \quad i \in \{1, 2\},$$

$$A^{(i)}(\boldsymbol{x}) := \begin{pmatrix} 1 + a_0 h(1 - \boldsymbol{x}^\top \boldsymbol{x}) & b_i h \\ -b_i h & 1 + a_0 h(1 - \boldsymbol{x}^\top \boldsymbol{x}) \end{pmatrix},$$

where $b_1 = 0$ and $b_2 = 1$. We denote by $\alpha^\star$ the true underline parameter which represents f. Note that $\boldsymbol{\alpha}^\star := (1 - 0.5\pi, 0.5\pi)$. At each time $t > 0$, we assume that the model set $\{f^{(i)}\}_{i=1,2}$ and 0.5 second online data set $\mathcal{I}_t = \{\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_\tau, \boldsymbol{u}_\tau, \tau \in \mathcal{T}, |\mathcal{T}| = 500\}$ are available for the computation of adaptive ambiguity sets.

Let us consider the learning and control problem (P)

$$\begin{aligned} &\min_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}_{\mathbb{P}_{t+1}} \left[\frac{1}{2} \|\boldsymbol{u}\|^2 + F_\mu(\boldsymbol{x}_{t+1} - \bar{\boldsymbol{x}}_{t+1}) \right], \\ &\text{s.t. } \mathbb{P}_{t+1} \text{ of } \boldsymbol{x}_{t+1} \text{ is characterized by (5.5)}, \end{aligned}$$

where $\mu = 0.1$, $\mathcal{U} := [-0.6, 0.6]^2$, and the reference signal $\bar{\boldsymbol{x}}$ is generated by the system

$$\bar{\boldsymbol{x}}^+ = \begin{pmatrix} 1 + a_0 h(1 - \bar{\boldsymbol{x}}^\top \bar{\boldsymbol{x}}) & h \\ -h & 1 + a_0 h(1 - \bar{\boldsymbol{x}}^\top \bar{\boldsymbol{x}}) \end{pmatrix} \bar{\boldsymbol{x}},$$

with the period 2π. Using the proposed technique, we reformulate the above problem into form (P2′), where its objective function at each t is

$$\frac{1}{2}\|\boldsymbol{u}\|^2 + \frac{1}{T} \sum_{k \in \mathcal{T}} F_\mu(\boldsymbol{p}_{k,t}) + \epsilon + \frac{\gamma}{T} \sum_{i=1}^{p} \sum_{k \in \mathcal{T}} F_\mu(H_k^{(i)}),$$

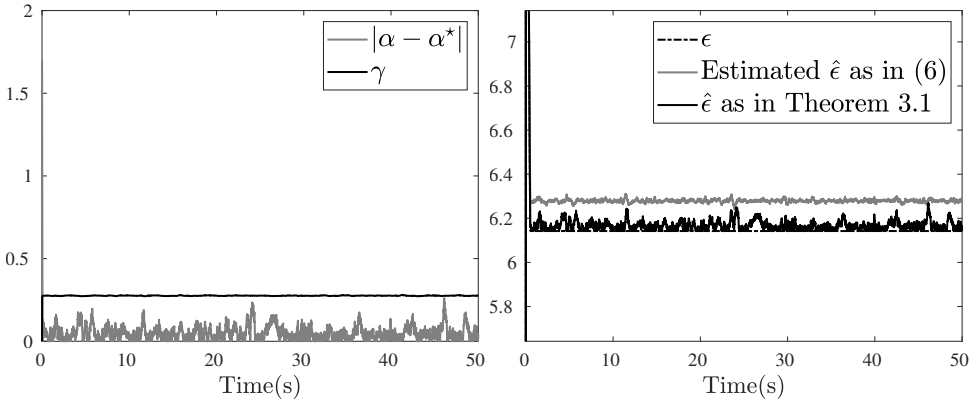


Figure 5.1. The estimated bound γ and radius $\hat{\epsilon}$ in probability, where γ is an online-accessible indicator for the quality of learned $\boldsymbol{\alpha}$ in dynamic environments, and $\hat{\epsilon}$ is a time-varying radius of adaptive ambiguity sets constructed online.

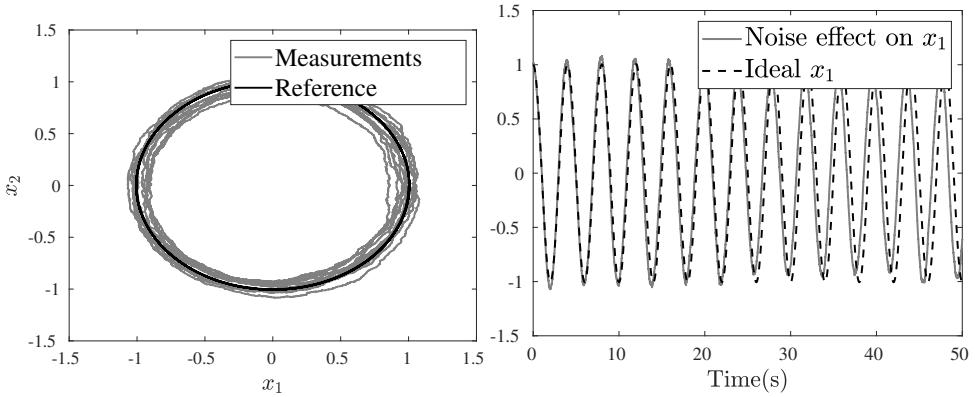


Figure 5.2. System trajectory and evolution of x_1 without control.

with

$$\boldsymbol{p}_{k,t} := \sum_{i=1}^{p} \alpha_i \left(A^{(i)}(\hat{\boldsymbol{x}}_t)\hat{\boldsymbol{x}}_t - A^{(i)}(\hat{\boldsymbol{x}}_k)\hat{\boldsymbol{x}}_k - h\boldsymbol{u}_k \right) - \bar{\boldsymbol{x}}_{t+1} + \hat{\boldsymbol{x}}_{k+1} + \left(\sum_{i=1}^{p} \alpha_i \right) h\boldsymbol{u},$$

$$H_k^{(i)}(\boldsymbol{u}) := A^{(i)}(\hat{\boldsymbol{x}}_k)\hat{\boldsymbol{x}}_k + h\boldsymbol{u}_k - A^{(i)}(\hat{\boldsymbol{x}}_t)\hat{\boldsymbol{x}}_t - h\boldsymbol{u},$$

where its Lipschitz gradient constant, at each t, is

$$\mathrm{Lip}(G_\mu) = 1 + (\sum_{i=1}^{p} \alpha_i)^2 h^2/\mu + \gamma p h/\mu,$$

and the parameter ϵ, γ and $\boldsymbol{\alpha}$ are determined as in Theorem 9. We compute the online solution $\boldsymbol{u}_t$ using the solution system (5.2), with the time-dependent step-size $\varepsilon := 1/\mathrm{Lip}(G_\mu)$.

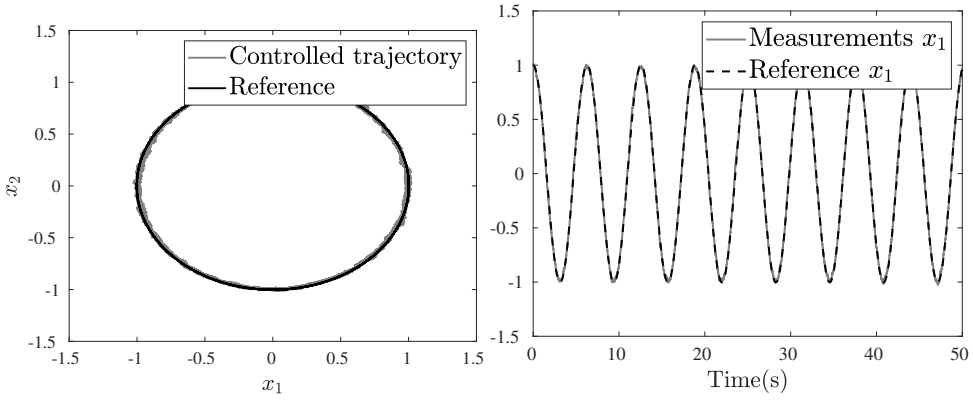


Figure 5.3. Controlled system trajectory and evolution of x_1.

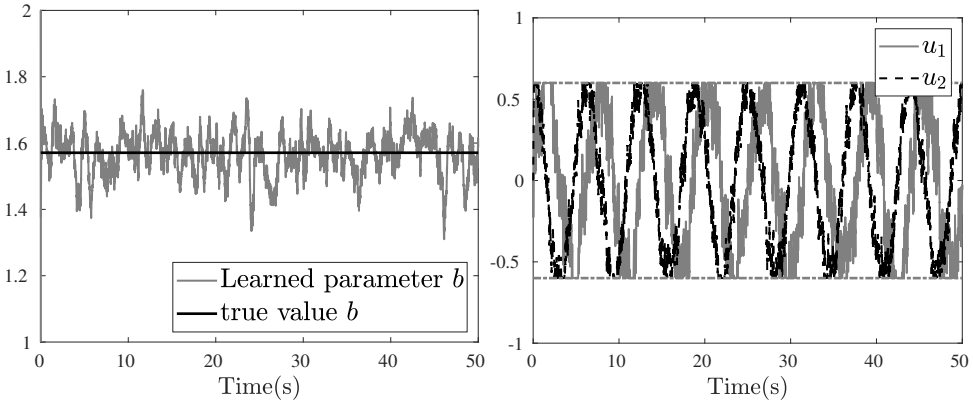


Figure 5.4. Estimated period b and control $\boldsymbol{u}$.

To demonstrate the learning effect of the algorithm, let us first characterize adaptive ambiguity sets. Fig. 5.1(a) shows the quality of the learned parameter $\boldsymbol{\alpha}$, computed as in (4.6), and its estimated bound γ, which is used for the estimation of the radius of ambiguity sets [68]. In reality, the vector $\boldsymbol{\alpha}^\star$ is unknown and the value γ verifies the quality of its estimate $\boldsymbol{\alpha}$ in probability, as described in Chapter 4. When the learning procedure (4.6) is effective, i.e., the value $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_\infty$ is close to zero, then the parameter γ is small with high probability. In this case study, the selected $\boldsymbol{\alpha}$ is indeed a reasonable estimate of $\boldsymbol{\alpha}^\star$, even if the estimation introduces ambiguity and enlarges the radius of ambiguity sets as illustrated in Fig. 5.1(b).

Fig. 5.2 and Fig. 5.3 demonstrate the system trajectory under uncertainty, with and without control from (P). It can be seen that, the system is stabilized to the desired frequency, and the trajectory follows the reference signal in high probability. When there is no control, the system is dominated by the uncertainty. Fig. 5.4(a) shows the estimated parameter b_0, which is calculated from $\boldsymbol{\alpha}$. The estimation of b_0 is unbiased from its true value, which helps solution system (5.2) to obtain control with performance guarantees. We show in Fig. 5.4(b) the online control obtained from (5.2). The proposed algorithm accounts for the imposed constraints on control and tracks the optimal solution of (P) in high probability.

5.7.2 Study 2: Online Resource Allocation Problem

We consider an online resource allocation problem where an agent or decision maker aims to 1) achieve at least target profit under uncertainty, and 2) allocate resources as uniformly as possible. To do this, the agent distributes available resources, e.g., wealth, time, energy or human resources, to various projects or assets. In particular, let us consider that the agent tries to make an online allocation $\boldsymbol{u} \in \mathcal{U}$ of a unit wealth to three assets. At each time t, the agent receives random return rates $\boldsymbol{x} \in \mathbb{R}^3_{\geq 0}$ of assets, following some unknown and uncertain dynamics

$$\boldsymbol{x}^+ = \boldsymbol{x} + hA(t) + h\boldsymbol{w}, \text{ with some } \boldsymbol{x}_0 \in \mathbb{R}^2, \tag{5.6}$$

where $h = 10^{-3}$ is a stepsize, the vector $A(t)$ is randomly generated, unknown and piecewise constant, and the uncertainty vector $\boldsymbol{w}$ is assumed to be subGaussian with $\sigma = 0.1$. Note that this model can serve to characterize a wide class of dynamic (linear and nonlinear) systems. In addition, we assume that the third asset is value preserved, i.e., the third component of $A(t)$ and $\boldsymbol{w}$ are zero and $x_3 \equiv 1$. An example of the resulting unit return rates $\boldsymbol{x}$ is demonstrated in Fig. 5.5. Then, we denote by $r_0 = 1.3$ and $\langle \boldsymbol{u}, \boldsymbol{x} \rangle$ the target profit and the instantaneous profit, respectively. Note that the decision maker aims to obtain at least a 30% profit and allocate resources online for this purpose. In particular, the decision maker implements an allocation online if $\langle \boldsymbol{u}, \boldsymbol{x} \rangle \leq r_0$, otherwise does nothing. This results in (P) with the loss function

$$\ell(\boldsymbol{u}, \boldsymbol{x}) = \max\{0, 1 - \frac{1}{r_0}\langle \boldsymbol{u}, \boldsymbol{x} \rangle\},$$

and set $\mathcal{U}$ a unit simplex. We propose $p = 3$ predictors

$$f^{(1)} = \boldsymbol{x},\ f^{(2)} = \boldsymbol{x} + 0.1h\boldsymbol{e}_1,\ f^{(3)} = \boldsymbol{x} + 0.1h\boldsymbol{e}_2,$$

where $\boldsymbol{e}_1 = (1, 0, 0)^\top$ and $\boldsymbol{e}_2 = (0, 1, 0)^\top$. At each t, we assume that only historical data are available for online resource allocations. Applying the proposed probabilistic characterization of $\boldsymbol{x}_{t+1}$ as in (P1), we equivalently write it as in form (P2′), where

$$G_\mu(t, \boldsymbol{u}) = \frac{1}{T}\sum_{k \in \mathcal{T}} F_\mu^{\mathrm{S}}(\langle \boldsymbol{u}, \frac{\boldsymbol{p}_{k,t}}{r_0} \rangle) + \frac{q_t}{r_0} F_\mu(\boldsymbol{u}),\ \mu = 0.01,$$

with real-time data $\boldsymbol{p}_{k,t}$ and q_t determined as in Problem 2. We claim that $G_\mu(t, \boldsymbol{u})$ has a time-dependent Lipschitz gradient constant in $\boldsymbol{u}$ given by $\mathrm{Lip}(G_\mu) = q_t/r_0 + 1/(r_0^2 T)\sum_{k \in \mathcal{T}} \|\boldsymbol{p}_{k,t}\|_\infty^2$, and we use $\varepsilon := 1/\mathrm{Lip}(G_\mu)$ in the solution system (5.2) to compute the online decisions.

Fig. 5.6 shows the real-time evolution α_1 of the parameter $\boldsymbol{\alpha} := (\alpha_1, \alpha_2, \alpha_3)$, while the behavior of α_2 and α_3 can be similarly characterized. In this figure, black line $\alpha_1^\star$ is determined by the unknown signal $A(t)$ while gray line α_1 is that computed as in (4.6). Note that $\boldsymbol{\alpha}^\star$

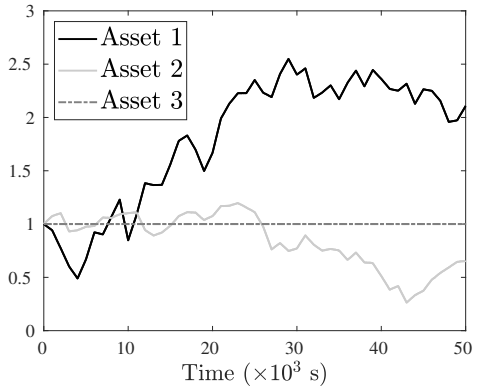


Figure 5.5. An example of random returns $\boldsymbol{x} = (x_1, x_2, x_3)$, where returns of the first two assets $x_1, x_2 \in [0, +\infty)$ are highly fluctuated and the third is value-preserved with return $x_3 \equiv 1$.

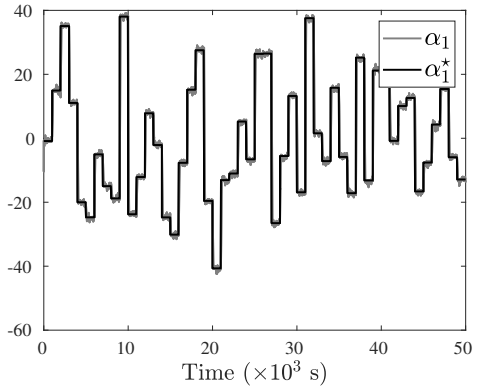


Figure 5.6. The component α_1 of the real-time parameter $\boldsymbol{\alpha} := (\alpha_1, \alpha_2, \alpha_3)$ in learning, where the value $\alpha_1^\star$ is the online-inaccessible ground truth. Notice the responsive behavior of the proposed learning algorithm.

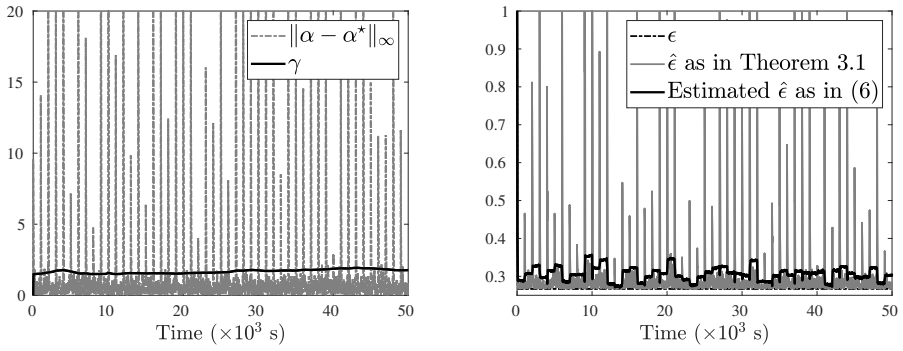


Figure 5.7. The estimated bound γ and radius $\hat{\epsilon}$, where the online-accessible parameter γ upper bounds the quality degradation of $\boldsymbol{\alpha}$ w.r.t. $\boldsymbol{\alpha}^\star$ in high probability. The bound γ contributes to tight radius $\hat{\epsilon}$ of ambiguity sets.

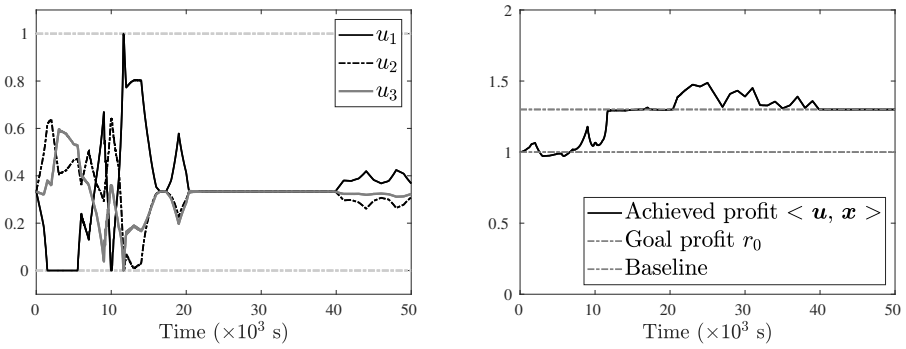


Figure 5.8. Real-time resource allocation $\boldsymbol{u}$ and profit $\langle \boldsymbol{u}, \boldsymbol{x} \rangle$. Notice how the decision $\boldsymbol{u} = (u_1, u_2, u_3)$ respects constraints and how the allocation is balanced when the goal profit r_0 is achieved.

represents the unknown dynamics f and they are not accessible in reality. It can be seen that the proposed method effectively learns $\boldsymbol{\alpha}^\star$. More precisely, we verify the quality of the estimated $\boldsymbol{\alpha}$ in probability by an upper bound γ of $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_\infty$, as shown in Fig. 5.7(a). Fig 5.7(b) shows the estimated radius of adaptive ambiguity sets, calculated as in (4.9), and we compare it with its true value, which is calculated using $\boldsymbol{\alpha}^\star$ as in Theorem 9. Note that, in these figures, a spike appears when the unknown $\boldsymbol{\alpha}^\star$ changes, and the proposed learning method immediately reduces it back.

Fig. 5.8 demonstrates the online resource allocation obtained by implementing (5.2) and the achieved real-time profit $\langle \boldsymbol{u}, \boldsymbol{x} \rangle$. The decision $\boldsymbol{u}$ starts from the uniform allocation $\boldsymbol{u}_0 = (1/3, 1/3, 1/3)$ and is then adjusted to approach the target profit $r_0 = 1.3$. Once the target is achieved, the agent then maintains the profit while trying to balance the allocation if possible. When the return rate $\boldsymbol{x}$ is low/unbalanced, the agent tries to improve and achieve the target profit by allocating resources more aggressively. In case that the return rate is high and the target profit value is achieved, the agent focuses on balancing the allocation while maintaining the profit. If both requirements are achieved, then the agent stops re-allocating resources and monitors the return rate $\boldsymbol{x}$ until switch turns on, e.g., when the future profit prediction drops below r_0 again.

Chapter 5, in full, is under revision for publication in Automatica, as *Online optimization and learning in uncertain dynamical environments with performance guarantees*, D. Li,

D. Fooladivanda, and S. Martínez. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Data-driven High-Confidence Attack Detection

This chapter considers a sensor attack and fault detection problem for linear cyber-physical systems, which are subject to system noise that can obey an unknown light-tailed distribution. In this chapter, we propose a new threshold-based detection mechanism that employs the Wasserstein metric, and which guarantees system performance with high confidence employing a finite number of measurements. The proposed detector may generate false alarms with a rate Δ in normal operation, where Δ can be tuned to be arbitrarily small by means of a *benchmark distribution* which is part of our mechanism. Thus, the proposed detector is sensitive to sensor attacks and faults which have a statistical behavior that is different from that of the system noise. We quantify the impact of *stealthy* attacks—which aim to perturb the system operation while producing false alarms that are consistent with the natural system noise—via a *probabilistic* reachable set. To enable tractable implementation of our methods, we propose a linear optimization problem that computes the proposed detection measure and a semidefinite program that produces the proposed reachable set.

6.1 Related Works

Cyber-Physical Systems (CPS) are physical processes that are tightly integrated with computation and communication systems for monitoring and control. These systems are usually

complex, large-scale and insufficiently supervised, making them vulnerable to attacks [21,105]. A significant literature has studied various *denial of service* [3], *false data-injection* [6,88], *replay* [92, 140], *sensor, and integrity* attacks [90,91,93,96] in a control-theoretic framework, by comparing estimation and measurements w.r.t. predefined metrics. However, attacks could be *stealthy*, and exploit knowledge of the system structure, uncertainty and noise information to inflict significant damage on the physical system while avoiding detection. This motivates the characterization of the impact of stealthy attacks via e.g. reachability set analysis [5,93,98]. To ensure computational tractability, these works assume either Gaussian or bounded system noise. However, these assumptions fall short in modeling all natural disturbances that can affect a system. When designing detectors, an added difficulty is in obtaining tractable computations that can handle these more general distributions. More recently, novel measure of concentration has opened the way for online tractable and robust attack detection with probability guarantees under uncertainty. A first attempt in this direction is [111], which exploits the Chebyshev's inequality to design a detector, and characterizes stealthy attacks on stable systems affected by bounded system noises. With the aim of obtaining a less conservative detection mechanism, we leverage an alternative measure-concentration result via Wasserstein metric. This metric is built from data gathered on the system, and can provide significantly sharper results than those stemming from the Chebyshev inequality. In particular, we address the following question for linear CPSs: *How to design an online attack-detection mechanism that is robust to light-tailed distributions of system noise while remaining sensitive to attacks and limiting the impact of the stealthy attack?*

Statement of Contributions

To address the previous question in this chapter: 1) We propose a novel detection measure, which employs the Wasserstein distance between the benchmark and a distribution of the residual sequence obtained online. 2) We propose a novel threshold-detection mechanism, which exploits measure-of-concentration results to guarantee the robust detection of an attack

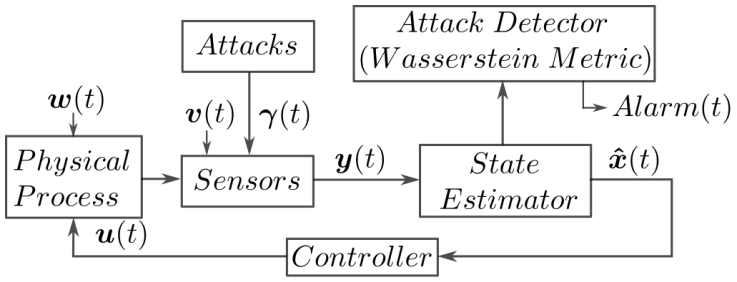


Figure 6.1. Cyber-Physical System Diagram.

with high confidence using a finite set of data, and which further enables the robust tuning of the false alarm rate. The proposed detector can effectively identify real-time attacks when its behavior differs from that of the system noise. In addition, the detector can handle systems noises that are not necessarily distributed as Gaussian. 3) We propose a quantifiable, probabilistic state-reachable set, which reveals the impact of the stealthy attacker and system noise on open loop stable systems with high probability. 4) To implement the proposed mechanism, we formulate a linear optimization problem and a semidefinite problem for the computation of the detection measure as well as the reachable set, respectively. We illustrate our methods in a two-dimensional linear system with irregular noise distributions and stealthy sensor attacks.

6.2 Cyber-Physical Systems

A remotely-observed, cyber-physical system subject to sensor-measurement attacks, as in Fig. 6.1, is described as a discrete-time, stochastic, linear, and time-invariant system

$$\begin{aligned} \boldsymbol{x}(t+1) &= A\boldsymbol{x}(t) + B\boldsymbol{u}(t) + \boldsymbol{w}(t), \\ \boldsymbol{y}(t) &= C\boldsymbol{x}(t) + \boldsymbol{v}(t) + \boldsymbol{\gamma}(t), \end{aligned} \tag{6.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathbb{R}^m$ and $\boldsymbol{y}(t) \in \mathbb{R}^p$ denote the system state, input and output at time $t \in \mathbb{N}$, respectively. The state matrix A, input matrix B and output matrix C are assumed to be known in advance. In particular, we assume that the pair (A, B) is stabilizable, and (A, C) is detectable. The process noise $\boldsymbol{w}(t) \in \mathbb{R}^n$ and output noise $\boldsymbol{v}(t) \in \mathbb{R}^p$ are independent zero-mean random vectors.

We assume that each $\boldsymbol{w}(t)$ and $\boldsymbol{v}(t)$ are independent and identically distributed (i.i.d.) over time. We denote their (unknown, not-necessarily equal) distributions by $\mathbb{P}_{\boldsymbol{w}}$ and $\mathbb{P}_{\boldsymbol{v}}$, respectively. In addition, we assume that $\mathbb{P}_{\boldsymbol{w}}$ and $\mathbb{P}_{\boldsymbol{v}}$ are light-tailed[1], excluding scenarios of systems operating under extreme events, or subject to large delays. In fact, Gaussian, Sub-Gaussian, Exponential distributions, and any distribution with a compact support set are admissible. This distribution class is sufficient to characterize the uncertainty or noise of many practical problems.

An additive sensor-measurement attack is implemented via $\boldsymbol{\gamma}(t) \in \mathbb{R}^p$ in (6.1), on which we assume the following

Assumption 16 (Attack model). *It holds that 1)* $\boldsymbol{\gamma}(t) = \mathbf{0}$ *whenever there is no attack; 2) the attacker can modulate any component of* $\boldsymbol{\gamma}(t)$ *at any time; 3) the attacker has unlimited computational resources and access to system information, e.g., A, B, C,* $\boldsymbol{u}$*,* $\mathbb{P}_{\boldsymbol{w}}$ *and* $\mathbb{P}_{\boldsymbol{v}}$ *to decide on* $\boldsymbol{\gamma}(t)$*,* $t \in \mathbb{N}$*.*

6.2.1 Normal System Operation

Here, we introduce the state observer that enables prediction in the absence of attacks ($\boldsymbol{\gamma}(t) = \mathbf{0}$). Since the distribution of system noise is unknown, we identify a benchmark distribution to capture this unknown distribution with high confidence.

To predict the system behavior, we employ a Kalman filter

$$\begin{aligned} \hat{\boldsymbol{x}}(t+1) &= A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t) + L(t)(\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)), \\ \hat{\boldsymbol{y}}(t) &= C\hat{\boldsymbol{x}}(t), \end{aligned}$$

where $\hat{\boldsymbol{x}}(t)$ is the state estimate and $L(t) \equiv L$ is the steady-state Kalman gain matrix. As the pair (A, C) is detectable, the gain L is selected to bring the eigenvalues of $A - LC$ inside the unit circle.

[1] See Definition 1 for details. All examples listed have a moment generating function, so their exponential moment can be constructed for at least $q = 1$.

This ensures that the estimation error $\boldsymbol{e}(t) := \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)$ satisfies

$$\mathbb{E}[\boldsymbol{e}(t)] \to 0 \text{ as } t \to \infty, \text{ for any } \boldsymbol{x}(0),\ \hat{\boldsymbol{x}}(0).$$

We additionally consider the estimated state feedback $\boldsymbol{u}(t) = K\hat{\boldsymbol{x}}(t)$, where K is selected to make the next system stable[2]

$$\boldsymbol{\xi}(t+1) = F\boldsymbol{\xi}(t) + G\boldsymbol{\sigma}(t), \tag{6.2}$$

where $\boldsymbol{\xi}(t) := (\boldsymbol{x}(t), \boldsymbol{e}(t))^\top$, $\boldsymbol{\sigma}(t) := (\boldsymbol{w}(t), \boldsymbol{v}(t) + \boldsymbol{\gamma}(t))^\top$,

$$F = \begin{bmatrix} A + BK & -BK \\ 0 & A - LC \end{bmatrix}, G = \begin{bmatrix} I & 0 \\ I & -L \end{bmatrix} \text{ and some } \boldsymbol{\xi}(0).$$

Remark 23 (Selection of L and K). In general, the selection of the matrices L and K for the system (6.1) is a nontrivial task, especially when certain performance criteria are to be satified, such as fast system response, energy conservation, or noise minimization. However, there are a few scenarios in which the *Separation Principle* can be invoked for a tractable design of L and K. For example, 1) when there is no system noise, matrices L and K can be designed separately, such that each $A + BK$ and $A - LC$ have all eigenvalues contained inside the unit circle, respectively. 2) when noise are Gaussian, the gain matrices L and K can be designed to minimize the steady-state covariance matrix and control effort, via a separated design of a Kalman filter (as an observer) and a linear-quadratic regulator (as a controller). The resulting design is referred to as a Linear-Quadratic-Gaussian (LQG) control [4].

Consider the system initially operates normally after selecting L and K, and assume that the augmented system (6.2) is in steady state, i.e., $\mathbb{E}[\boldsymbol{\xi}(t)] = \mathbf{0}$. In order to design our attack detector, we need a characterization of the distribution of the *residue* $\boldsymbol{r}(t) \in \mathbb{R}^p$, evaluating the

[2]System (6.2) is input-to-state stable in probability (ISSp) relative to any compact set $\mathcal{A}$ which contains the origin, if we select K such that eigenvalues of the matrix $A + BK$ are inside the unit circle, see e.g. [124].

difference between what we measure and what we expect to receive:

$$\boldsymbol{r}(t) := \boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t) = C\boldsymbol{e}(t) + \boldsymbol{v}(t) + \boldsymbol{\gamma}(t).$$

When there is no attack, it can be verified that $\boldsymbol{r}(t)$ is zero-mean, and light-tailed[3]. Let us denote its unknown distribution by $\mathbb{P}_{\boldsymbol{r}}$. We assume that a finite, but large number N of i.i.d. samples of $\mathbb{P}_{\boldsymbol{r}}$, are accessible, and acquired by collecting $\boldsymbol{r}(t)$ for a sufficiently large time. We call these i.i.d. samples a *benchmark data set*, $\Xi_{\mathrm{B}} := \{\boldsymbol{r}^{(i)} = \boldsymbol{y}^{(i)} - \hat{\boldsymbol{y}}^{(i)}\}_{i=1}^{N}$, and construct the resulting empirical distribution $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ by

$$\mathbb{P}_{\boldsymbol{r},\mathrm{B}} := \frac{1}{N}\sum_{i=1}^{N} \delta_{\{\boldsymbol{r}^{(i)}\}},$$

where the operator δ is the mass function, and the subscript B indicates that $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ is the benchmark distribution of the data. We claim that $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ provides a characterization of the effect of the noise on (6.2) via the following result:

Theorem 13 (Measure of concentration [40, Application of Theorem 2]). *If $\mathbb{P}_{\boldsymbol{r}}$ is a q-light-tailed distribution for some $q \geq 1$, then for a given $\beta \in (0,1)$, the following holds*

$$\mathrm{Prob}\left(d_{W,q}(\mathbb{P}_{\boldsymbol{r},\mathrm{B}}, \mathbb{P}_{\boldsymbol{r}}) \leq \epsilon_{\mathrm{B}}\right) \geq 1 - \beta,$$

where Prob *denotes the Probability of the samples in $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$, $d_{W,q}$ denotes the q-Wasserstein metric[4], and the parameter ϵ_{B} is selected as*

$$\epsilon_{\mathrm{B}} := \begin{cases} \left(\frac{\log(c_1\beta^{-1})}{c_2 N}\right)^{q/a}, & \text{if } N < \frac{\log(c_1\beta^{-1})}{c_2}, \\ \bar{\epsilon}, & \text{if } N \geq \frac{\log(c_1\beta^{-1})}{c_2}, \end{cases} \tag{6.3}$$

[3]This can be checked from the definition in footnote 1, and follows from $\boldsymbol{r}(t)$ being a linear combination of zero-mean q-light-tailed distributions.

[4] See Section 1.2 for the definition.

for some constant[5] $a > q$, c_1, $c_2 > 0$, *and* $\bar{\epsilon}$ *is such that* $c_2 N(\bar{\epsilon})^{\max\{2,p/q\}} = \log(c_1\beta^{-1})$, *if* $p \neq 2q$, *or* $\frac{\bar{\epsilon}}{\log(2+1/\bar{\epsilon})} = \left(\frac{\log(c_1\beta^{-1})}{c_2 N}\right)^{1/2}$, *if* $p = 2q$, *where* p *is the dimension of* $\boldsymbol{r}$. □

Theorem 13 provides a probabilistic bound ϵ_{B} on the q-Wasserstein distance between $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ and $\mathbb{P}_{\boldsymbol{r}}$, with a confidence at least $1-\beta$. It indicates how to tune the parameter β and the number of benchmark samples N that are needed to find a sufficiently good approximation of $\mathbb{P}_{\boldsymbol{r}}$, by means of $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$. In this way, given an ϵ, we can increase our confidence $(1-\beta)$ on whether $\mathbb{P}_{\boldsymbol{r}}$ and $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ are within distance ϵ, by increasing the number of samples. We assume that $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ has been determined in advance, selecting a very large (unique) N to ensure various very small bounds ϵ_{B} associated with various β. Later, we discuss how the parameter β can be interpreted as a *false alarm rate* in the proposed attack detector. The resulting $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$, with a tunnable false alarm rate (depending on β), will allow us to design a detection procedure which is robust to the system noise.

6.3 Threshold-based Robust Detection of Attacks, and Stealthiness

This section presents our online detection procedure, and a threshold-based detector with high-confidence performance guarantees. Then, we propose a tractable computation of the detection measure used for online detection. We finish the section by introducing a class of stealthy attacks.

Online Detection Procedure (ODP): At each time $t \geq T$, we construct a T-step detector distribution

$$\mathbb{P}_{\boldsymbol{r},\mathrm{D}} := \frac{1}{T}\sum_{j=0}^{T-1} \delta_{\{\boldsymbol{r}(t-j)\}},$$

where $\boldsymbol{r}(t-j)$ is the residue data collected independently at time $t-j$, for $j \in \{0,\ldots,T-1\}$. Then

[5]The parameter a is determined as in the definition of $\mathbb{P}_{\boldsymbol{r}}$ and the constants c_1, c_2 depend on q, m, and $\mathbb{P}_{\boldsymbol{r}}$ (via a, b, c). When information on $\mathbb{P}_{\boldsymbol{r}}$ is absent, the parameters a, c_1 and c_2 can be determined in a data-driven fashion using sufficiently many samples of $\mathbb{P}_{\boldsymbol{r}}$. See [40] for details.

with a given q and a threshold $\alpha > 0$, we consider the *detection measure*

$$z(t) := d_{W,q}(\mathbb{P}_{\boldsymbol{r},\mathrm{B}}, \mathbb{P}_{\boldsymbol{r},\mathrm{D}}), \tag{6.4}$$

and the *attack detector*

$$\begin{cases} z(t) \leq \alpha, & \text{no alarm at } t : \mathrm{Alarm}(t) = 0, \\ z(t) > \alpha, & \text{alarm at } t : \mathrm{Alarm}(t) = 1, \end{cases} \tag{6.5}$$

with $\mathrm{Alarm}(t)$ the sequence of alarms generated online based on the previous threshold. The distribution $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$ uses a small number T of samples to ensure the online computational tractability of $z(t)$, so $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$ is highly dependent on instantaneous samples. Thus, $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$ may significantly deviate from the true $\mathbb{P}_{\boldsymbol{r}}$, and from $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$. Therefore, even if there is no attack, the attack detector is expected to generate false alarms due to the system noise as well as an improper selection of the threshold α. In the following, we discuss how to select an α that is robust to the system noise and which results in a desired false alarm rate. Note that the value α should be small to be able to distinguish attacks from noise, as discussed later.

Lemma 18 (Selection of α for robust detectors). *Given parameters N, T, q, β, and a desired false alarm rate $\Delta > \beta$ at time t, if we select the threshold α as*

$$\alpha := \epsilon_{\mathrm{B}} + \epsilon_{\mathrm{D}},$$

where ϵ_{B} is chosen as in (6.3) *and ϵ_{D} is selected following the ϵ_{B}-formula* (6.3)*, but with T and $\frac{\Delta-\beta}{1-\beta}$ in place of N and β, respectively. Then, the detection measure* (6.4) *satisfies*

$$\mathrm{Prob}\left(z(t) \leq \alpha\right) \geq 1 - \Delta,$$

for any zero-mean q-light-tailed underlying distribution $\mathbb{P}_{\boldsymbol{r}}$.

Proof. To prove this, $z(t) \leq d_{W,q}(\mathbb{P}_{\boldsymbol{r},\mathrm{B}}, \mathbb{P}_{\boldsymbol{r}}) + d_{W,q}(\mathbb{P}_{\boldsymbol{r},\mathrm{D}}, \mathbb{P}_{\boldsymbol{r}})$ follows from the triangular inequality. Then we apply Theorem 13 for each $d_{W,q}$ term, and the fact that $\mathrm{Prob}\left(d_{W,q}(\mathbb{P}_{\boldsymbol{r},\mathrm{D}}, \mathbb{P}_{\boldsymbol{r}}) \leq \epsilon_{\mathrm{D}}\right) \geq 1 - \frac{\Delta-\beta}{1-\beta}$. Note also that samples of $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ and $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$ are collected independently. □

Lemma 18 ensures that the false alarm rate is no higher than Δ when there is no attack, i.e.,

$$\mathrm{Prob}(\mathrm{Alarm}(t) = 1 \mid \text{no attack}) \leq \Delta, \quad \forall\, t.$$

Note that the rate Δ can be selected by properly choosing the threshold α. Intuitively, if we fix all the other parameters, then the smaller the rate Δ, the larger the threshold α. Also, large values of N, T, $1-\beta$ contribute to small α.

Remark 24 (Comparison with χ^2-detector). Consider an alternative detection measure

$$z_\chi(t) := \boldsymbol{r}(t)^\top \Sigma^{-1} \boldsymbol{r}(t),$$

where Σ is the constant covariance matrix of the residue $\boldsymbol{r}(t)$ under normal system operation. In particular, if $\boldsymbol{r}$ is Gaussian, the detection measure $z_\chi(t)$ is χ^2-distributed and referred to as χ^2 detection measure with p degree of freedom. The detector threshold α is selected via look-up tables of χ^2 distribution, given the desired false alarm rate Δ. To compare $z(t)$ with $z_\chi(t)$, we leverage the assumption that $\boldsymbol{r}$ is Gaussian with the given covariance Σ. This gives explicitly the expression of $z(t)$ the following

$$z(t) := \left(\mathbb{E}_{\xi \sim \mathcal{N}(\boldsymbol{r}(t),\Sigma)}[\|\xi\|^q]\right)^{-1/q}.$$

By selecting $q = 2$, we have

$$z(t) := \left(\boldsymbol{r}(t)^\top \boldsymbol{r}(t) + Tr(\Sigma)\right)^{-1/2}.$$

Note that, the measure-of-concentration result in Theorem 13 is sharp when $\boldsymbol{r}$ is Gaussian, which

in fact results in the threshold α as tight as that derived for χ^2-detector.

Computation of detection measure: To achieve a tractable computation of the detection measure $z(t)$, we leverage the definition of the Wasserstein distance (see footnote 4) and the fact that both $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ and $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$ are discrete. The solution is given as a linear program.

The Wasserstein distance $d_{W,q}(\mathbb{P}_{\boldsymbol{r},\mathrm{B}}, \mathbb{P}_{\boldsymbol{r},\mathrm{D}})$, originally solving the *Kantorovich optimal transport problem* [115], can be interpreted as the minimal work needed to move a mass of points described via a probability distribution $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}(\boldsymbol{r})$, on the space $\mathcal{Z} \subset \mathbb{R}^p$, to a mass of points described by the probability distribution $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}(\boldsymbol{r})$ on the same space, with some transportation cost ℓ. The minimization that defines $d_{W,q}$ is taken over the space of all the joint distributions Π on $\mathcal{Z} \times \mathcal{Z}$ whose marginals are $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ and $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$, respectively.

Assuming that both $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ and $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$ are discrete, we can equivalently characterize the joint distribution Π as a discrete mass *optimal transportation plan* [115]. To do this, let us consider two sets $\mathcal{N} := \{1, \ldots, N\}$ and $\mathcal{T} := \{0, \ldots, T-1\}$. Then, Π can be parameterized (by λ) as follows

$$\begin{aligned} \Pi_\lambda(\xi_1, \xi_2) &:= \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{T}} \lambda_{ij} \delta_{\{\boldsymbol{r}^{(i)}\}}(\xi_1) \delta_{\{\boldsymbol{r}(t-j)\}}(\xi_2), \\ \text{s.t.} \sum_{i \in \mathcal{N}} \lambda_{ij} &= \frac{1}{T}, \ \forall\ j \in \mathcal{T}, \quad \sum_{j \in \mathcal{T}} \lambda_{ij} = \frac{1}{N}, \ \forall\ i \in \mathcal{N}, \qquad (6.6) \\ \lambda_{ij} &\geq 0, \ \forall\ i \in \mathcal{N},\ j \in \mathcal{T}. \qquad (6.7) \end{aligned}$$

Note that this characterizes all the joint distributions with marginals $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ and $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$, where λ is the allocation of the mass from $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ to $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$. Then, the proposed detection measure $z(t)$ in (6.4) reduces to the following

$$\begin{aligned} (z(t))^q := \min_{\lambda} & \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{T}} \lambda_{ij} \|\boldsymbol{r}^{(i)} - \boldsymbol{r}(t-j)\|^q, \\ & \text{s.t. (6.6), (6.7),} \end{aligned} \qquad \text{(P)}$$

which is a linear program over a compact polyhedral set. Therefore, the solution exists and (P)

can be solved to global optimal in polynomial time by e.g., a CPLEX solver.

6.3.1 Detection and Stealthiness of Attacks

Following from the previous discussion, we now introduce an Erroneous Detection Quantification Problem, then specialize it to the Attack Detection Problem considered in this chapter. In particular, we analyze the sensitivity of the proposed attack detector method for the cyber-physical system under attacks.

Problem 1: (Erroneous detection quantification problem) Given the augmented system (6.2), the online detection procedure in Section 6.3, and the attacker type described in Assumption 16, compute the erroneous detection probability

$$\begin{aligned}&\text{Prob(erroneous detection at } t) :=\\ &\quad \text{Prob}(\text{Alarm}(t) = 1 \mid \text{no attack})\,\text{Prob}(\text{no attack}) + \text{Prob}(\text{Alarm}(t) = 0 \mid \text{attack})\,\text{Prob}(\text{attack}).\end{aligned}$$

Problem 1, on the computation of the erroneous detection probability, requires prior information of the attack probability Prob(attack). In this chapter, we are interested in *stealthy attacks*, i.e., attacks that can avoid detection by (6.5). These attacks, in the worst case, can induce significant system damage before notice. We are led to the following problem.

Problem 2: (Attack detection problem) Given the setting of Problem 1, provide conditions that characterize stealthy attacks, i.e., attacks that contribute to $\text{Prob}(\text{Alarm}(t) = 0 \mid \text{attack})$, and quantify their potential impact on the system.

To remain undetected, the attacker must select $\boldsymbol{\gamma}(t)$ such that $z(t)$ is limited to within the threshold α. To quantify the effects of these attacks, let us consider an attacker sequence backward in time with length T. At time t, denote the arbitrary injected attacker sequence by $\boldsymbol{\gamma}(t-j) \in \mathbb{R}^p$, $j \in \{0, \ldots, T-1\}$ (if $t - j < 0$, assume $\boldsymbol{\gamma}(t-j) = 0$). This sequence, together with (6.2), results in a detection sequence $\{\boldsymbol{r}(t-j)\}_j$ that can be used to construct detector distribution $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$ and detection measure $z(t)$. We characterize the scenarios that can occur, providing a first, partial

answer to Problem 2. Then, we will come back to analyzing the impact of stealthy attacks in Section 6.4.

Definition 3 (Attack detection characterization). *Assume* (6.2) *is subject to attack, i.e.,* $\boldsymbol{\gamma}(t) \neq \mathbf{0}$ *for some* $t \geq 0$.

- *If* $z(t) \leq \alpha$, $\forall\, t \geq 0$, *then the attack is stealthy with probability one, i.e.,* $\text{Prob}(\text{Alarm}(t) = 0 \,|\, \text{attack}) = 1$.
- *If* $z(t) \leq \alpha$, $\forall t \leq M$, *then the attack is* M*-step stealthy.*
- *If* $z(t) > \alpha$, $\forall t \geq 0$, *then the attack is active with probability one, i.e.,* $\text{Prob}(\text{Alarm}(t) = 0 \,|\, \text{attack}) = 0$.

For simplicity and w.l.o.g., let us assume that $\boldsymbol{\gamma}(t)$ is in form

$$\boldsymbol{\gamma}(t) := \hat{\boldsymbol{y}}^{\text{o}}(t) - \boldsymbol{y}^{\text{o}}(t) + \bar{\gamma}(t) = -C\boldsymbol{e}(t) - \boldsymbol{v}(t) + \bar{\gamma}(t), \tag{6.8}$$

where $\hat{\boldsymbol{y}}^{\text{o}}(t)$, $\boldsymbol{y}^{\text{o}}(t)$ are online noisy measurements of $\hat{\boldsymbol{y}}(t)$, $\boldsymbol{y}(t)$, and $\bar{\gamma}(t) \in \mathbb{R}^p$ is any vector selected by the attacker.[6]

Lemma 19 (Stealthy attacks leveraging system noise). *Assume* (6.2) *is subject to attack that leverages measurements* $\hat{\boldsymbol{y}}^{\text{o}}(t)$ *and* $\boldsymbol{y}^{\text{o}}(t)$ *as in form* (6.8)*, where* $\bar{\gamma}(t)$ *is stochastic and distributed as* $\mathbb{P}_{\bar{\gamma}}$. *If* $\mathbb{P}_{\bar{\gamma}}$ *is selected such that* $d_{W,q}(\mathbb{P}_{\bar{\gamma}}, \mathbb{P}_{\boldsymbol{r},\text{B}}) \leq \epsilon_{\text{B}}$*, then the attacker is stealthy with (high) probability at least* $\frac{1-\Delta}{1-\beta}$*, i.e.,* $\text{Prob}(\text{Alarm}(t) = 0 \mid \text{attack}) \geq \frac{1-\Delta}{1-\beta}$. [7]

Proof. Assume (6.2) is under attack. Leveraging the measure concentration result,

$$\text{Prob}\left(d_{W,q}(\mathbb{P}_{\bar{\gamma}}, \mathbb{P}_{\boldsymbol{r},\text{D}}) \leq \epsilon_{\text{D}}\right) \geq 1 - \frac{\Delta - \beta}{1 - \beta},$$

[6]Note that, when there is no attack at t, we have $\boldsymbol{\gamma}(t) = \mathbf{0}$, resulting in selection $\bar{\gamma}(t) = \boldsymbol{y}^{\text{o}}(t) - \hat{\boldsymbol{y}}^{\text{o}}(t)$. Similar techniques are in, e.g., [93, 97].

[7]Note that $\alpha > \epsilon_{\text{B}}$, which allows the attacker to select $\mathbb{P}_{\bar{\gamma}}$ with $\epsilon_{\text{B}} < d_{W,q}(\mathbb{P}_{\bar{\gamma}}, \mathbb{P}_{\boldsymbol{r},\text{B}}) \leq \alpha$. However, the probability of being stealthy can be indefinitely low, with the range $[0, \frac{1-\Delta}{1-\beta}]$.

which holds as $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$ is constructed using samples of $\mathbb{P}_{\bar{\gamma}}$. This together with the triangular inequality $z(t) \leq d_{W,q}(\mathbb{P}_{\boldsymbol{r},\mathrm{B}}, \mathbb{P}_{\bar{\gamma}}) + d_{W,q}(\mathbb{P}_{\boldsymbol{r},\mathrm{D}}, \mathbb{P}_{\bar{\gamma}})$, results into $z(t) \leq \alpha$ with probability at least $\frac{1-\Delta}{1-\beta}$. □

6.4 Stealthy Attack Analysis

In this section, we address the second question in Problem 2 via reachable-set analysis. In particular, note that the CPS (6.1) is resilient to stealthy attacks only when (6.1) is open-loop stable. Under this assumption, we achieve an outer-approximation of the finite-step probabilistic reachable set, quantifying the effect of the stealthy attacks and the system noise in probability.

Consider an attack sequence $\boldsymbol{\gamma}(t)$ as in (6.8), where $\bar{\gamma}(t) \in \mathbb{R}^p$ is any vector such that the attack remains stealthy. That is, $\bar{\gamma}(t)$ results in the detected distribution $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$, which is close to $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ as prescribed by α. This results in the representation of (6.2) as

$$\boldsymbol{\xi}(t+1) = \underbrace{\begin{bmatrix} A+BK & -BK \\ 0 & A \end{bmatrix}}_{H} \boldsymbol{\xi}(t) + \underbrace{\begin{bmatrix} I & 0 \\ I & -L \end{bmatrix}}_{G} \begin{bmatrix} \boldsymbol{w}(t) \\ \bar{\gamma}(t) \end{bmatrix}. \tag{6.9}$$

We provide in the following remark an intuition of how restrictive the stealthy attacker's action $\bar{\gamma}(t)$ has to be.

Remark 25 (Constant attacks). Consider a constant offset attack $\bar{\gamma}(t) := \gamma_0$ for some $\gamma_0 \in \mathbb{R}^p$, $\forall t$. Then by (P),

$$z(t) = N^{1-1/q} \|\gamma_0 - \mathrm{C}(\Xi_{\mathrm{B}})\|, \quad \mathrm{C}(\Xi_{\mathrm{B}}) := \frac{1}{N} \sum_{i \in \mathcal{N}} \boldsymbol{r}^{(i)}.$$

To ensure stealth, we require $z(t) \leq \alpha$, this then limits the selection of γ_0 in a ball centered at $\mathrm{C}(\Xi_{\mathrm{B}})$ with radius $\alpha / N^{1-1/q}$. Note that the radius can be arbitrarily small by choosing the benchmark size N large.

To quantify the reachable set of the system under attacks, prior information on the process noise $\boldsymbol{w}(t)$ is needed. To characterize $\boldsymbol{w}(t)$, let us assume that, similarly to the benchmark $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$, we are able to construct a noise benchmark distribution, denoted by $\mathbb{P}_{\boldsymbol{w},\mathrm{B}}$. As before,

$$\mathrm{Prob}\left(d_{W,q}(\mathbb{P}_{\boldsymbol{w},\mathrm{B}}, \mathbb{P}_{\boldsymbol{w}}) \leq \epsilon_{\boldsymbol{w},\mathrm{B}}\right) \geq 1-\beta,$$

for some $\epsilon_{\boldsymbol{w},\mathrm{B}}$. Given certain time, we are interested in where, with high probability, the state of the system can evolve from some $\boldsymbol{\xi}_0$. To do this, we consider the *M-step probabilistic reachable set* of stealthy attacks, defined as follows

$$\mathcal{R}_{\boldsymbol{x},M} := \left\{ \boldsymbol{x}(M) \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \text{system (6.9) with } \boldsymbol{\xi}(0) = \boldsymbol{\xi}_0, \\ \exists\, \mathbb{P}_{\boldsymbol{w}} \ni d_{W,q}(\mathbb{P}_{\boldsymbol{w}}, \mathbb{P}_{\boldsymbol{w},\mathrm{B}}) \leq \epsilon_{\boldsymbol{w},\mathrm{B}}, \\ \exists\, \mathbb{P}_{\tilde{\gamma}} \ni d_{W,q}(\mathbb{P}_{\tilde{\gamma}}, \mathbb{P}_{\boldsymbol{r},\mathrm{B}}) \leq \alpha, \end{array} \right\},$$

then the true system state $\boldsymbol{x}(t)$ at time M, $\boldsymbol{x}(M)$, satisfies

$$\mathrm{Prob}\left(\boldsymbol{x}(M) \in \mathcal{R}_{\boldsymbol{x},M}\right) \geq 1-\beta,$$

where $1-\beta$ accounts for the independent restriction of the unknown distributions $\mathbb{P}_{\boldsymbol{w}}$ to be "close" to its benchmark.

The exact computation of $\mathcal{R}_{\boldsymbol{x},M}$ is intractable due to the unbounded support of the unknown distributions $\mathbb{P}_{\boldsymbol{w}}$ and $\mathbb{P}_{\tilde{\gamma}}$, even if they are close to their benchmark. To ensure a tractable approximation, we follow a two-step procedure. First, we equivalently characterize the constraints on $\mathbb{P}$ by its *probabilistic support set*. Then, we outer-approximate the probabilistic support by ellipsoids, and then the reachable set by an ellipsoidal bound.

Step 1: (Probabilistic support of $\mathbb{P}_{\tilde{\gamma}} \ni d_{W,q}(\mathbb{P}_{\tilde{\gamma}}, \mathbb{P}_{\boldsymbol{r},\mathrm{B}}) \leq \alpha$) We achieve this by leveraging 1) the *Monge formulation* [115] of optimal transport, 2) the fact that $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ is discrete, and 3) results

from coverage control [18, 28]. W.l.o.g., let us assume $\mathbb{P}_{\bar{\gamma}}$ is non-atomic (or continuous) and, consider the distribution $\mathbb{P}_{\bar{\gamma}}$ and $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ supported on $\mathcal{Z} \subset \mathbb{R}^p$. Let us denote by $f : \mathbb{P}_{\bar{\gamma}} \mapsto \mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ the *transport map* that assigns mass over $\mathcal{Z}$ from $\mathbb{P}_{\bar{\gamma}}$ to $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$. The Monge formulation aims to find an optimal transport map that minimizes the transportation cost ℓ as follows

$$d_{M,q}(\mathbb{P}_{\bar{\gamma}}, \mathbb{P}_{\boldsymbol{r},\mathrm{B}}) := \left(\min_{f} \int_{\mathcal{Z}} \ell^q(\xi, f(\xi)) \mathbb{P}_{\bar{\gamma}}(\xi) d\xi \right)^{1/q}.$$

It is known that if an optimal transport map $f^\star$ exists, then the optimal transportation plan $\Pi^\star$ exists and the two problems $d_{M,q}$ and $d_{W,q}$ coincide[8]. In our setting, for strongly convex ℓ^p, and by the fact that $\mathbb{P}_{\bar{\gamma}}$ is absolutely continuous, a unique optimal transport map can indeed be guaranteed[9], and therefore, $d_{M,q} = d_{W,q}$. Let us now consider any transport map f of $d_{M,q}$, and define a partition of the support of $\mathbb{P}_{\bar{\gamma}}$ by

$$W_i := \{\boldsymbol{r} \in \mathcal{Z} \mid f(\boldsymbol{r}) = \boldsymbol{r}^{(i)}\},\ i \in \mathcal{N},$$

where $\boldsymbol{r}^{(i)}$ are samples in Ξ_{B}, which generate $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$. Then, we equivalently rewrite the objective function defined in $d_{M,q}$, as

$$\begin{aligned} \mathcal{H}(\mathbb{P}_{\bar{\gamma}}, W_1, \ldots, W_N) := & \sum_{i=1}^{N} \int_{W_i} \ell^q(\xi, \boldsymbol{r}^{(i)}) \mathbb{P}_{\bar{\gamma}}(\xi) d\xi, \\ \text{s.t.} & \int_{W_i} \mathbb{P}_{\bar{\gamma}}(\xi) d\xi = \frac{1}{N},\ \forall\ i \in \mathcal{N}, \end{aligned} \tag{6.10}$$

where the i^{th} constraints come from the fact that a transport map f should lead to consistent calculation of set volumes under $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ and $\mathbb{P}_{\bar{\gamma}}$, respectively. This results in the following equivalent

[8]This is because the Kantorovich transport problem is the tightest relaxation of the Monge transport problem. See e.g., [115] for details.

[9]The Monge formulation is not always well-posed, i.e., there exists optimal transportation plans $\Pi^\star$ while transport map does not exist [115].

problem to $d_{M,q}$ as

$$(d_{M,q}(\mathbb{P}_{\bar{\gamma}}, \mathbb{P}_{\boldsymbol{r},\mathrm{B}}))^q := \min_{W_i, i\in\mathcal{N}} \mathcal{H}(\mathbb{P}_{\bar{\gamma}}, W_1, \ldots, W_N), \quad \text{s.t. (6.10).} \tag{P1}$$

Given the distribution $\mathbb{P}_{\bar{\gamma}}$, Problem (P1) reduces to a load-balancing problem as in [28]. Let us define the Generalized Voronoi Partition (GVP) of $\mathcal{Z}$ associated to the sample set Ξ_{B} and weight $\omega \in \mathbb{R}^N$, for all $i \in \mathcal{N}$, as

$$\mathcal{V}_i(\Xi_{\mathrm{B}}, \omega) := \{\xi \in \mathcal{Z} \mid \|\xi - \boldsymbol{r}^{(i)}\|^q - \omega_i \leq \|\xi - \boldsymbol{r}^{(j)}\|^q - \omega_j,\ \forall j \in \mathcal{N}\}.$$

It has been established that the optimal Partition of (P1) is the GVP [28, Proposition V.1]. Further, the standard Voronoi partition, i.e., the GVP with equal weights $\bar{\omega} := \mathbf{0}$, results in a lower boundof (P1), when constraints (6.10) are removed [18], and therefore that of $d_{M,q}$. We denote this lower bound as $L(\mathbb{P}_{\bar{\gamma}}, \mathcal{V}(\Xi_{\mathrm{B}}))$, and use this to quantify a probabilistic support of $\mathbb{P}_{\bar{\gamma}}$. Let us consider the support set

$$\Omega(\Xi_{\mathrm{B}}, \epsilon) := \cup_{i\in\mathcal{N}} \left(\mathcal{V}_i(\Xi_{\mathrm{B}}) \cap \mathbb{B}_\epsilon(\boldsymbol{r}^{(i)})\right),$$

where $\mathbb{B}_\epsilon(\boldsymbol{r}^{(i)}) := \{\boldsymbol{r} \in \mathbb{R}^p \mid \|\boldsymbol{r} - \boldsymbol{r}^{(i)}\| \leq \epsilon\}$.

Lemma 20 (Probabilistic support). *Let $\epsilon > 0$ and let $\mathbb{P}_{\bar{\gamma}}$ be such that $L(\mathbb{P}_{\bar{\gamma}}, \mathcal{V}(\Xi_{\mathrm{B}})) \leq \epsilon^q$. Then, for any given $s > 1$, at least $1 - 1/s^q$ portion of the mass of $\mathbb{P}_{\bar{\gamma}}$ is supported on $\Omega(\Xi_{\mathrm{B}}, s\epsilon)$, i.e., $\int_{\Omega(\Xi_{\mathrm{B}}, s\epsilon)} \mathbb{P}_{\bar{\gamma}}(\xi) d\xi \geq 1 - 1/s^q$.*

Proof. Suppose otherwise, i.e., $\int_{\mathbb{R}^p \setminus \Omega(\Xi_{\mathrm{B}}, s\epsilon)} \mathbb{P}_{\bar{\gamma}}(\xi) d\xi = 1 - \int_{\Omega(\Xi_{\mathrm{B}}, s\epsilon)} \mathbb{P}_{\bar{\gamma}}(\xi) d\xi > 1/s^q$. Then,

$$\begin{aligned} L(\mathbb{P}_{\bar{\gamma}}, \mathcal{V}(\Xi_{\mathrm{B}})) &\geq \int_{\mathbb{R}^p \setminus \Omega(\Xi_{\mathrm{B}}, s\epsilon)} \|\xi - \boldsymbol{r}^{(i)}\|^q \mathbb{P}_{\bar{\gamma}}(\xi) d\xi, \\ &\geq s^q \epsilon^q \int_{\mathbb{R}^p \setminus \Omega(\Xi_{\mathrm{B}}, s\epsilon)} \mathbb{P}_{\bar{\gamma}}(\xi) d\xi > \epsilon^q, \text{ contradiction.} \end{aligned}$$

□

In this way, the support $\Omega(\Xi_{\mathrm{B}}, 2\alpha)$ contains at least $1 - 1/2^q$ of the mass of all the

distributions $\mathbb{P}_{\bar{\gamma}}$ such that $d_{W,q}(\mathbb{P}_{\bar{\gamma}}, \mathbb{P}_{\boldsymbol{r},\mathrm{B}}) \leq \alpha$. Equivalently, we have $\mathrm{Prob}\,(\bar{\gamma} \in \Omega(\Xi_{\mathrm{B}}, 2\alpha)) \geq 1 - 1/2^q$, where the random variable $\bar{\gamma}$ has a distribution $\mathbb{P}_{\bar{\gamma}}$ such that $d_{W,q}(\mathbb{P}_{\bar{\gamma}}, \mathbb{P}_{\boldsymbol{r},\mathrm{B}}) \leq \alpha$. We characterize $\mathbb{P}_{\boldsymbol{w}}$ similarly. Note that in practice, one can choose ball radius factor s large in order to generate support which contains higher portion of the mass of unknown $\mathbb{P}$. However, it comes at a cost on the approximation of the reachable set.

Step 2: (Outer-approximation of $\mathcal{R}_{\boldsymbol{x},M}$) Making use of the probabilistic support, we can now obtain a finite-dimensional characterization of the probabilistic reachable set, as follows

$$\mathcal{R}_{\boldsymbol{x},M} := \left\{ \boldsymbol{x}(M) \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \text{system (6.9), } \boldsymbol{\xi}(0) = \boldsymbol{\xi}_0, \\ \boldsymbol{w} \in \Omega(\Xi_{\boldsymbol{w},\mathrm{B}}, 2\epsilon_{\boldsymbol{w},\mathrm{B}}), \\ \bar{\gamma} \in \Omega(\Xi_{\mathrm{B}}, 2\alpha) \end{array} \right\},$$

and the true system state $\boldsymbol{x}(t)$ at time M, $\boldsymbol{x}(M)$, satisfies $\mathrm{Prob}\left(\boldsymbol{x}(M) \in \mathcal{R}_{\boldsymbol{x},M}\right) \geq (1-\beta)(1-1/2^q)^2$. Note that, if the CPS (6.1) is open-loop unstable, so does (6.9). This leads to vulnerable CPS to stealthy sensor attacks. That is, almost surely any stealthy attack $\boldsymbol{\gamma}(t)$ in form (6.8) inflicts significant damage of the system with an unbounded reachable set, i.e., $\exists\, \boldsymbol{x}(M) \in \mathcal{R}_{\boldsymbol{x},M}$ such that $\boldsymbol{x}(M) \to \infty$ as $M \to \infty$. Many works focus on the tractable evolution of geometric shapes when (6.1) is stable and resilient to stealthy attacks[10], e.g. [93, 98]. Here, we follow [98] and propose outer ellipsoidal bounds for $\mathcal{R}_{\boldsymbol{x},M}$. Let $Q_{\boldsymbol{w}}$ be the positive-definite shape matrix such that $\Omega(\Xi_{\boldsymbol{w},\mathrm{B}}, \epsilon_{\boldsymbol{w},\mathrm{B}}) \subset \mathcal{E}_{\boldsymbol{w}} := \{\boldsymbol{w} \mid \boldsymbol{w}^\top Q_{\boldsymbol{w}} \boldsymbol{w} \leq 1\}$. Similarly, we denote $Q_{\bar{\gamma}}$ and $\mathcal{E}_{\bar{\gamma}}$ for that of $\bar{\gamma}$. We now state the lemma, that applies [98, Proposition 1] for our case.

Lemma 21 (Outer bounds of $\mathcal{R}_{\boldsymbol{x},M}$). *Given any $a_0 \in (0,1)$, we claim $\mathcal{R}_{\boldsymbol{x},M} \subset \mathcal{E}(Q) := \{\boldsymbol{x} \in$*

[10] If (6.1) is unstable, we either need extra protected sensors or benchmark data of the state estimate $\hat{\boldsymbol{x}}$, $\mathbb{P}_{\hat{\boldsymbol{x}},\mathrm{B}}$, to ensure effective stealthy attack detection.

$\mathbb{R}^n \mid \boldsymbol{\xi}^\top Q \boldsymbol{\xi} \le a_0^M \boldsymbol{\xi}_0{}^\top Q \boldsymbol{\xi}_0 + \frac{(2-a_0)(1-a_0^M)}{1-a_0}\}$, *with* Q *satisfying*

$$Q \succ 0, \begin{bmatrix} a_0 Q & H^\top Q & \mathbf{0} \\ QH & Q & QG \\ \mathbf{0} & G^\top Q & W \end{bmatrix} \succeq 0, \tag{6.11}$$

where H, G *are that in* (6.9) *and*

$$W = \begin{bmatrix} (1-a_1)Q_{\boldsymbol{w}} & \mathbf{0} \\ \mathbf{0} & (1-a_2)Q_{\bar{\gamma}} \end{bmatrix}, \tag{6.12}$$

$$\text{for some } a_1 + a_2 \ge a_0,\ a_1, a_2 \in (0,1).$$

A tight reachable set bound can be now derived by solving

$$\begin{aligned} \min_{Q, a_1, a_2} \quad & -\log\det(Q), \\ \text{s.t.} \quad & (6.11),\ (6.12), \end{aligned} \tag{P2}$$

which is a convex semidefinite program, solvable via e.g., SeDuMi [122]. Note that the probabilistic reachable set is

$$\mathcal{R}_{\boldsymbol{x}} := \cup_{M=1}^{\infty} \mathcal{R}_{\boldsymbol{x},M},$$

which again can be approximated via $Q^\star$ solving (P2) for[11]

$$\mathcal{R}_{\boldsymbol{x}} \subset \mathcal{E}(Q^\star) = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{\xi}^\top Q^\star \boldsymbol{\xi} \le \frac{(2-a_0)}{1-a_0}\}.$$

[11]The set $\mathcal{R}_{\boldsymbol{x}}$ is in fact contained in the projection of $\mathcal{E}(Q^\star)$ onto the state subspace, i.e., $\mathcal{R}_{\boldsymbol{x}} \subset \{\boldsymbol{x} \mid \boldsymbol{x}^\top (Q_{xx} - Q_{xe} Q_{ee}^{-1} Q_{xe}{}^\top)\boldsymbol{x} \le \frac{(2-a_0)}{1-a_0}\}$ with $Q^\star := \begin{bmatrix} Q_{xx} & Q_{xe} \\ Q_{xe}{}^\top & Q_{ee} \end{bmatrix}$. See, e.g., [98] for details.

6.5 Simulations

In this section, we demonstrate the performance of the proposed attack detector, illustrating its distributional robustness w.r.t. the system noise. Then, we consider stealthy attacks as in (6.8) and analyze their impact by quantifying the probabilistic reachable set and outer-approximation bound.

Consider the stochastic system (6.2), given as

$$
\begin{aligned}
&A = \begin{bmatrix} 1.00 & 0.10 \\ -0.20 & 0.75 \end{bmatrix},\ B = \begin{bmatrix} 0.10 \\ 0.20 \end{bmatrix},\ L = \begin{bmatrix} 0.23 \\ -0.20 \end{bmatrix},\\
&C = \begin{bmatrix} 1 & 0 \end{bmatrix},\ K = \begin{bmatrix} -0.13 & 0.01 \end{bmatrix},\ n = 2,\ m = p = 1,\\
&\boldsymbol{w}_1 \sim \mathcal{N}(-0.25, 0.02) + \mathcal{U}(0, 0.5),\ \boldsymbol{v} \sim \mathcal{U}(-0.3, 0.3),\\
&\boldsymbol{w}_2 \sim \mathcal{N}(0, 0.04) + \mathcal{U}(-0.2, 0.2),
\end{aligned}
$$

where $\mathcal{N}$ and $\mathcal{U}$ represent the normal and uniform distributions, respectively. We consider $N = 10^3$ benchmark samples for $\mathbb{P}_{\boldsymbol{r},\mathrm{B}}$ and $T = 10^2$ real-time samples for $\mathbb{P}_{\boldsymbol{r},\mathrm{D}}$. We select $q = 1$, $\beta = 0.01$ and false alarm rate $\Delta = 0.05$. We select the prior information of the system noise via $a = 1.5$, $c_1 = 1.84 \times 10^6$ and $c_2 = 12.5$. Using the measure-of-concentration results, we determine the detector threshold to be $\alpha = 0.158$. In the normal system operation (no attack), we run the online detection procedure for 10^4 time steps and draw the distribution of the computed detection measure $z(t)$ as in Fig. 6.2. We verify that the false alarm rate is 3.68%, within the required rate $\Delta = 5\%$. When the system is subject to stealthy attacks, we assume $\boldsymbol{\xi}_0 = \mathbf{0}$ and visualize the Voronoi partition $\mathcal{V}(\Xi_{\boldsymbol{w},\mathrm{B}})$ (convex sets with blue boundaries) of the probabilistic support $\Omega(\Xi_{\boldsymbol{w},\mathrm{B}}, \epsilon_{\boldsymbol{w},\mathrm{B}})$ and its estimated ellipsoidal bound (red line) as in Fig. 6.3. Further, we demonstrate the impact of the stealthy attacks (6.8) with $a_0 = 0.85$, as in Fig. 6.4. We used 10^4 empirical points of $\mathcal{R}_{\boldsymbol{x}}$ as its estimate and provided an ellipsoidal bound of $\mathcal{R}_{\boldsymbol{x}}$ computed by solution of (P2). It can be seen that the proposed probabilistic reachable set effectively captures the reachable set

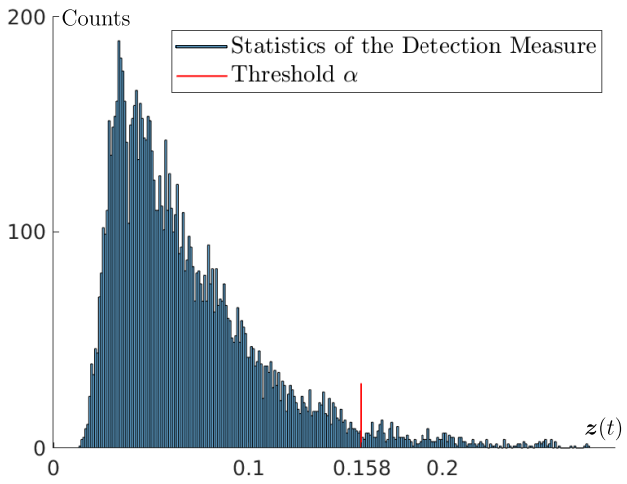


Figure 6.2. Statistics of z.

in probability. Due to the space limits, we omit the comparison of our approach to the existing ones, such as the classical χ^2 detector in [93] and the CUMSUM procedure [90]. However, the difference should be clear: our proposed approach is robust w.r.t. noise distributions while others leverage the moment information up to the second order, which only capture sufficient information about certain noise distributions, e.g., Gaussian.

Chapter 6, in full, is a reprint of *High-confidence attack detection via Wasserstein-metric computations*, D. Li and S. Martínez, IEEE Control Systems Letters, 5(2):379-384, 2020, which was presented at IEEE International Conference on Decision and Control, Jeju Island, Korea, 2020. The dissertation author was the primary investigator and author of this paper.

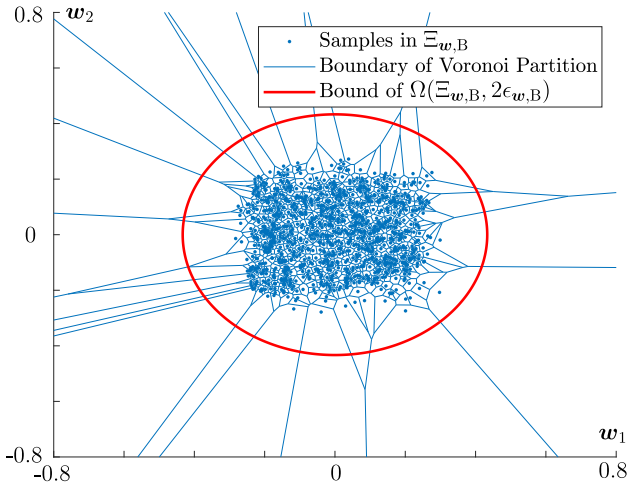


Figure 6.3. Probabilistic Support of $\mathbb{P}_{\boldsymbol{w}}$.

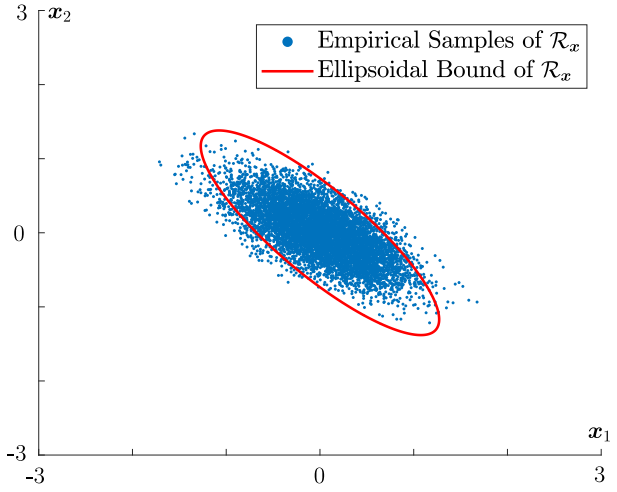


Figure 6.4. Empirical and Bound of $\mathcal{R}_{\boldsymbol{x}}$.

Chapter 7

Conclusion

In this thesis, we proposed theoretical and algorithmic foundations to support a class of uncertain systems which integrate optimization and control where rigorous performance guarantees are made available. In particular, we developed five finite-online-data-driven frameworks, focusing on online data-assimilation capabilities, data-driven control design, online learning of system models, resilient operations in uncertain environments, and anomaly detection. Each framework was approached with theoretical analysis, explainable proofs, and as well as numerical validations on efficacy via various case studies. More specific, chapter-by-chapter conclusions and future works are as the following:

In Chapter 2, we have proposed the Online Data Assimilation Algorithm (the OnDA Algorithm) to solve the problem in the form of (P), where the realizations of the unknown distribution (i.e., the streaming data) are collected over time in order for the real-time data-driven decision of (P) to have guaranteed out-of-sample performance. The data-driven decision with the certificate that guarantees out-of-sample performance are available any time during the execution of the algorithm, and the optimal data-driven decision are approached with a (sub)linear convergence rate. The algorithm terminates after collecting a sufficient amount of data to make good decision. To facilitate the decision making, an enhanced version of the proposed algorithm is further constructed, by using an Incremental Covering Algorithm (the I-Cover Algorithm) to estimate new ambiguity sets over time. We provided sample problems and showed the actual

performance of the proposed OnDA Algorithm with the I-Cover Algorithm over time. Future work may generalize the results for weaker assumptions of the problem and potentially extend the algorithm to scenarios that include system dynamics.

Chapter 3 formulates a data-driven predictive control problem with probabilistic performance guarantees as a distributionally optimization problem. We equivalently reformulate this intractable Problem (**P**) into a non-convex but tractable Problem (**P4**), or Mixed Integer Second Order Cone Problem (**P5**). This is achieved by: 1) extending Distributionally Robust Optimization theory to account for system dynamics; 2) reformulation and relaxation techniques. Finally, we adapt the idea of decomposition and propose an integer-solution search algorithm for efficient solutions of Problem (**P4**), or commercial solvers for that of Problem (**P5**). The proposed approaches provide performance guarantee (3.8) for Problem (**P**). To explicitly demonstrate the proposed approach, we consider a highway speed-limit control problem that accounts for random inflows, outflows and events. Finally, we demonstrate the theoretical effectiveness of this work via simulations and we simulate traffic flows on a highway in San Diego, showing the effectiveness of the designed speed limits numerically. Future work is on considering more complex traffic networks and the use of the framework for other complex systems.

In Chapter 4, we proposed an approach for online learning of unknown and uncertain dynamical systems or environments in a parameterized class. The proposed method allows us to learn the system, while providing an online characterization of the approximation via online-quantifiable probabilistic guarantees. The approach opens a way for the robust integration of the online learning with control design.

Chapter 5 extends results of Chapter 4, which proposes a unified solution framework for online learning and optimization problems in form of (P). The proposed method allows us to learn an unknown and uncertain dynamic environment, while providing a characterization of the environment with online-quantifiable probabilistic guarantees that certify the performance of online decisions. The approach provides tractable, online convex version of (P), via a series of equivalent reformulation techniques. We explicitly demonstrate the framework via two problem

scenarios conforming to (P): an optimal control problem under uncertainty and an online resource allocation problem. These two scenarios result in explicit, online and non-smooth convex optimization problems. We extend Nesterov's accelerated-gradient method to an online fashion and provide a solution system for online decision generation of (P). The quality of the online decisions are analytically certified via a probabilistic regret bound, which reveals its relation to the learning parameters and ambiguity sets.

Finaly, in Chapter 6, a novel detection measure is proposed to enable distributionally robust detection of attacks w.r.t. unknown, and light-tailed system noise. The proposed detection measure restricts the behavior of the stealthy attacks, whose impact is quantified via reachable-set analysis. Future work may focus on more extensive comparision of the proposed approach with other methods under various types of attacks.

Appendix A

Numerical Methods used in Chapter 2

There are mainly two types of Numerical methods that serve as the main ingredients of our OnDA Algorithm. One type is given by Frank-Wolfe Algorithm (FWA) variants and another is the Subgradient Algorithm. In this Section, we describe FWA and the Away-step Frank-Wolfe Algorithm (AFWA) for the sake of completeness. We combine AFWA with another variant, the Simplicial Algorithm, in Section 2.4. For the Subgradient Algorithm, please refer to [100, 112, 123].

A.1 Frank-Wolfe Algorithm over a Unit Simplex

To solve convex programs over a unit simplex, we introduce the FWA and AFWA following [52, 58]. Let us denote the m-dimensional unit simplex by $\Delta_m := \{\lambda \in \mathbb{R}^m \mid \mathbf{1}_m{}^\top \lambda = 1,\ \lambda \geq 0\}$. Let Λ_m be the set of all extreme points for the simplex Δ_m. Consider the maximization of a concave function $f(\boldsymbol{x})$ subject to $\boldsymbol{x} \in \Delta_m$; we refer to this problem by $(\star)$ and denote by $\boldsymbol{x}^\star$ an optimizer of $(\star)$. We call $\boldsymbol{x}^\epsilon$ an ϵ-optimal solution of $(\star)$, if $\boldsymbol{x}^\epsilon \in \Delta_m$ and $f(\boldsymbol{x}^\star) - f(\boldsymbol{x}^\epsilon) \leq \epsilon$. The classical FWA solves problem $(\star)$ to an $\boldsymbol{x}^\epsilon$ via the iterative process as follows. Let $\boldsymbol{x}^{(0)} \in \Delta_m$ denote a random initial point for FWA. For each iteration k with an $\boldsymbol{x}^{(k)} \in \Delta_m$, the concavity of f enables $f(\boldsymbol{x}^\star) \leq f(\boldsymbol{x}^{(k)}) + \nabla f(\boldsymbol{x}^{(k)})^\top (\boldsymbol{x}^\star - \boldsymbol{x}^{(k)})$, which implies $f(\boldsymbol{x}^\star) \leq f(\boldsymbol{x}^{(k)}) + \max_{\boldsymbol{x} \in \Delta_m} \nabla f(\boldsymbol{x}^{(k)})^\top (\boldsymbol{x} - \boldsymbol{x}^{(k)})$. Using this property, we define a FW search point $\boldsymbol{s}^{(k)}$ by an extreme point such that $\boldsymbol{s}^{(k)} \in \operatorname{argmax}_{\boldsymbol{x} \in \Delta_m} \nabla f(\boldsymbol{x}^{(k)})^\top (\boldsymbol{x} - \boldsymbol{x}^{(k)})$. With this search point we

define the FW direction at $\boldsymbol{x}^{(k)}$ by $d_{\mathrm{FW}}^{(k)} := \boldsymbol{s}^{(k)} - \boldsymbol{x}^{(k)}$. The classical FWA then iteratively finds a FW direction and solves a line search problem over this direction until an ϵ-optimal solution $\boldsymbol{x}^{\epsilon} := \boldsymbol{x}^{(k)}$ is found, certified by $\eta^{(k)} := \nabla f(\boldsymbol{x}^{(k)})^{\top} d_{\mathrm{FW}}^{(k)} \leq \epsilon$.

It is known that the classical FWA has linear convergence rate if the cost function f is μ-strongly concave and the optimum is achieved in the relative interior of the feasible set Δ_m. If the optimal solution lies on the boundary of Δ_m, then this algorithm only has a sublinear convergence rate, due to a zig-zagging phenomenon [58]. AFWA is an extension of the FWA that guarantees the linear convergence rate of the problem ($\star$) under some conditions related to the local strong concavity. The main difference between AFWA and the classical FWA is that the latter solves the line-search problem after obtaining a ascent direction by considering all extreme points, while the AFWA chooses a ascend direction that prevents zig-zagging. We summarize the convergence properties of the AFWA here. For complete descriptions of the AFWA, we refer the reader to [58]. The detailed FWA and AFWA are shown in Algorithm tables.

Algorithm 8. Classical FWA for ($\star$): FW($\boldsymbol{x}^{(0)}, \Delta_m, \epsilon$).

Ensure: ϵ-optimal $\boldsymbol{x}^{\epsilon}$;
1: Set $k \leftarrow 0$, $\eta^{(k)} \leftarrow +\infty$;
2: **repeat**
3: Pick $\boldsymbol{s}^{(k)} \in \underset{\boldsymbol{x} \in \Lambda_m}{\operatorname{argmax}} \nabla f(\boldsymbol{x}^{(k)})^{\top} (\boldsymbol{x} - \boldsymbol{x}^{(k)})$;
4: $d_{\mathrm{FW}}^{(k)} \leftarrow \boldsymbol{s}^{(k)} - \boldsymbol{x}^{(k)}$;
5: $\eta^{(k)} \leftarrow \nabla f(\boldsymbol{x}^{(k)})^{\top} d_{\mathrm{FW}}^{(k)}$;
6: Pick $\gamma^{(k)} \in \underset{\gamma \in [0,1]}{\operatorname{argmax}} f(\boldsymbol{x}^{(k)} + \gamma d_{\mathrm{FW}}^{(k)})$;
7: $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^{(k)} + \gamma^{(k)} d_{\mathrm{FW}}^{(k)}$;
8: $k \leftarrow k + 1$;
9: **until** $\eta^{(k)} \leq \epsilon$;
10: Return $\boldsymbol{x}^{(k)}$.

Theorem 14 (Linear convergence of AFWA [58, Theorem 8]). *Suppose the function f has a curvature constant C_f and a geometric strong concavity constant μ_f on Δ_m, as defined in footnote 1 Let us define the decay rate $\kappa := 1 - \mu_f/(4C_f) \in (0,1) \subset \mathbb{R}$. Then the suboptimality*

Algorithm 9. AFWA for $(\star)$: $(\boldsymbol{x}^{\epsilon}, \text{obj}^{\epsilon}) \leftarrow \text{AFW}(f(\boldsymbol{x}), \Delta_m, \epsilon)$.

Ensure: ϵ-optimal $\boldsymbol{x}^{\epsilon}$ with objective obj^{ϵ};
Set $k \leftarrow 0$, $\eta^{(k)} \leftarrow +\infty$;
Pick $\boldsymbol{x}^{(k)} \in \Lambda_m$, $I_{\text{Act}}^{(k)} := \{\boldsymbol{x}^{(k)}\}$, $p = |I_{\text{Act}}^{(k)}|$;
Let $\alpha_v^{(k)} = \begin{cases} 1/p, \text{ if } \boldsymbol{v} \in I_{\text{Act}}^{(k)}, \\ 0, \text{ if } \boldsymbol{v} \in \Lambda_m - I_{\text{Act}}^{(k)}, \end{cases}$
repeat
  Pick $\boldsymbol{s}^{(k)} \in \underset{\boldsymbol{x} \in \Lambda_m}{\text{argmax}} \nabla f(\boldsymbol{x}^{(k)})^T (\boldsymbol{x} - \boldsymbol{x}^{(k)})$;
  $d_{\text{FW}}^{(k)} \leftarrow \boldsymbol{s}^{(k)} - \boldsymbol{x}^{(k)}$;
  Pick $\boldsymbol{v}^{(k)} \in \underset{\boldsymbol{x} \in I_{\text{Act}}^{(k)}}{\text{argmin}} \nabla f(\boldsymbol{x}^{(k)})^T (\boldsymbol{x} - \boldsymbol{x}^{(k)})$;
  $d_{\text{A}}^{(k)} \leftarrow \boldsymbol{x}^{(k)} - \boldsymbol{v}^{(k)}$; ▷ Away-step direction
  if $\left\langle \nabla f(\boldsymbol{x}^{(k)}), d_{\text{FW}}^{(k)} \right\rangle \geq \left\langle \nabla f(\boldsymbol{x}^{(k)}), d_{\text{A}}^{(k)} \right\rangle$, **then**
    $d^{(k)} \leftarrow d_{\text{FW}}^{(k)}$;
    flag $\leftarrow$ True, $\gamma_{\max} \leftarrow 1$;
  else ▷ AFW direction has larger potential ascent
    $d^{(k)} \leftarrow d_{\text{A}}^{(k)}$;
    $\gamma_{\max} \leftarrow \alpha_{\boldsymbol{v}^{(k)}}^{(k)} / (1 - \alpha_{\boldsymbol{v}^{(k)}}^{(k)})$;
  Pick $\gamma^{(k)} \in \underset{\gamma \in [0, \gamma_{\max}]}{\text{argmax}} f(\boldsymbol{x}^{(k)} + \gamma d^{(k)})$;
  if flag is True, **then**
    if $\gamma^{(k)} = 1$, **then** ▷ Hit extreme point
      $I_{\text{Act}}^{(k+1)} \leftarrow \{\boldsymbol{s}^{(k)}\}$;
    else
      $I_{\text{Act}}^{(k+1)} \leftarrow I_{\text{Act}}^{(k)} \cup \{\boldsymbol{s}^{(k)}\}$;
    $\alpha_{\boldsymbol{s}^{(k)}}^{(k+1)} \leftarrow (1 - \gamma^{(k)}) \alpha_{\boldsymbol{s}^{(k)}}^{(k)} + \gamma^{(k)}$;
    $\alpha_{\boldsymbol{v}}^{(k+1)} \leftarrow (1 - \gamma^{(k)}) \alpha_{\boldsymbol{v}}^{(k)}$, $\forall \boldsymbol{v} \in I_{\text{Act}}^{(k)} - \{\boldsymbol{s}^{(k)}\}$;
  else
    if $\gamma^{(k)} = \gamma_{\max}$, **then** ▷ Hit Δ_m boundary
      $I_{\text{Act}}^{(k+1)} \leftarrow I_{\text{Act}}^{(k)} - \{\boldsymbol{v}^{(k)}\}$;
    else
      $I_{\text{Act}}^{(k+1)} \leftarrow I_{\text{Act}}^{(k)}$;
    $\alpha_{\boldsymbol{v}^{(k)}}^{(k+1)} = (1 + \gamma^{(k)}) \alpha_{\boldsymbol{v}^{(k)}}^{(k)} - \gamma^{(k)}$;
    $\alpha_{\boldsymbol{v}}^{(k+1)} = (1 + \gamma^{(k)}) \alpha_{\boldsymbol{v}}^{(k)}$, $\forall \boldsymbol{v} \in I_{\text{Act}}^{(k)} - \{\boldsymbol{v}^{(k)}\}$;
  $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^{(k)} + \gamma^{(k)} d^{(k)}$;
  $k \leftarrow k + 1$;
until $\eta^{(k)} \leq \epsilon$;
Return $\boldsymbol{x}^{\epsilon} \leftarrow \boldsymbol{x}^{(k)}$ and $\text{obj}^{\epsilon} \leftarrow f(\boldsymbol{x}^{\epsilon})$.

bound at the iteration point $\boldsymbol{x}^{(k)}$ of the AFWA decreases geometrically as $f(\boldsymbol{x}^{\star}) - f(\boldsymbol{x}^{(k+1)}) \leq \kappa(f(\boldsymbol{x}^{\star}) - f(\boldsymbol{x}^{(k)}))$. □

Appendix B

SubGaussian Properties used in Chapter 4

We adapt these two subGaussian properties for proofs in Chapter 4.

Lemma 22 ($\infty-$norm of subGaussian vectors have subGaussian tails [129]). *If Assumption 9 holds, then for each $k = 0, 1, 2, \ldots$ and any $\eta \geq 0$, we have*

$$\text{Prob}\left(\|\boldsymbol{w}_k\|_\infty \geq \eta\right) \leq 2n \exp\left(-\frac{\eta^2}{2\sigma^2}\right).$$

Proof. Lemma 22 Let $w_{k,i}$ denote the i^{th} component of $\boldsymbol{w}_k$ where $i \in \{1, \ldots, n\}$. We apply the definition of $\infty-$norm as the following

$$\text{Prob}\left(\|\boldsymbol{w}_k\|_\infty \geq \eta\right) = \text{Prob}\left(\max_{i\in\{1,\ldots,n\}} |w_{k,i}| \geq \eta\right) = 1 - \text{Prob}\left(|w_{k,i}| \leq \eta,\ \forall i \in \{1, \ldots, n\}\right).$$

By the independence of $w_{k,i}$ as in Assumption 9, we have

$$\text{Prob}\left(|w_{k,i}| \leq \eta,\ \forall i \in \{1, \ldots, n\}\right) = \text{Prob}\left(|w_{k,1}| \leq \eta\right) \cdots \text{Prob}\left(|w_{k,n}| \leq \eta\right).$$

Then for each $i \in \{1, \ldots, n\}$, we have[1]

$$\text{Prob}\left(|w_{k,i}| \leq \eta\right) = 1 - \text{Prob}\left(|w_{k,i}| \geq \eta\right) \geq 1 - 2\exp\left(-\frac{\eta^2}{2\sigma^2}\right),$$

[1] An equivalent representation of Assumption 9: For any $\eta \geq 0$, $\text{Prob}\left(|a^\top w_t| \geq \eta\right) \leq 2\exp\left(-\frac{\eta^2}{2\|a\|^2\sigma^2}\right)$.

which results in

$$\text{Prob}\left(|w_{k,i}| \leq \eta,\ \forall i \in \{1,\ldots,n\}\right) \geq \left[1 - 2\exp\left(-\frac{\eta^2}{2\sigma^2}\right)\right]^n.$$

Finally, we have[2]

$$\text{Prob}\left(\|\boldsymbol{w}_k\|_\infty \geq \eta\right) \leq 1 - \left[1 - 2\exp\left(-\frac{\eta^2}{2\sigma^2}\right)\right]^n \leq 2n\exp\left(-\frac{\eta^2}{2\sigma^2}\right).$$

□

Lemma 23 (Bounded moments of normed-subGaussian vectors [129]). *If Assumption 9 holds,*

$$\mathbb{E}\left[\|\boldsymbol{w}_k\|_\infty^l\right] \leq n\sigma^l l^{\frac{l}{2}+1},\ \ \forall l \in \mathbb{Z}_{\geq 0}.$$

Proof. Lemma 23 The moments can be equivalently computed by

$$\mathbb{E}\left[\|\boldsymbol{w}_k\|_\infty^l\right] = \int_0^\infty \text{Prob}\left(\|\boldsymbol{w}_k\|_\infty \geq \eta\right) l\eta^{l-1} d\eta.$$

Applying the result of Lemma 22, we have

$$\mathbb{E}\left[\|\boldsymbol{w}_k\|_\infty^l\right] \leq 2nl \int_0^\infty \exp\left(-\frac{\eta^2}{2\sigma^2}\right)\eta^{l-1} d\eta.$$

By the variable substitute $\bar{\eta} := \frac{\eta^2}{2\sigma^2}$, the above bound becomes

$$\mathbb{E}\left[\|\boldsymbol{w}_k\|_\infty^l\right] \leq nl(2\sigma^2)^{\frac{l}{2}} \int_0^\infty \exp\left(-\bar{\eta}\right)\bar{\eta}^{\frac{l}{2}-1} d\bar{\eta}.$$

[2]Bernoulli's inequality: $(1+x)^n \geq 1 + nx$ for $\forall n \in \mathbb{Z}_{>0}$, $\forall x \geq -2$.

By the definition of Γ function and its property[3], we have

$$\mathbb{E}\left[\|\boldsymbol{w}_k\|_\infty^l\right] \leq n\sigma^l l^{\frac{l}{2}+1}.$$

□

[3]The property of the Γ function: $\Gamma(\frac{l}{2}) := \int_0^\infty \exp(-\bar{\eta})\bar{\eta}^{\frac{l}{2}-1}d\bar{\eta} \leq (\frac{l}{2})^{\frac{l}{2}}$.

Appendix C

Solution Analysis used in Chapter 5

C.1 Smooth Approximation of Standard Functions

Example 2 (ℓ_2-norm function). *(1) Consider $\boldsymbol{x} \in \mathbb{R}^n$, $F : \boldsymbol{x} \mapsto \|\boldsymbol{x}\|$, and $\mu > 0$. Clearly, F is differentiable almost everywhere, except at the origin. Then,*

$$
\begin{aligned}
F_\mu(\boldsymbol{x}) &:= \min_{z \in \mathbb{R}^n} \left\{ \|z\| + \frac{1}{2\mu}\|z - \boldsymbol{x}\|^2 \right\}, \\
&= \min_{r \geq 0} \min_{\|z\| = r} \left\{ r + \frac{1}{2\mu}\left(r^2 - 2z^\top \boldsymbol{x} + \|\boldsymbol{x}\|^2 \right) \right\}, \\
&= \min_{r \geq 0} \left\{ r + \frac{1}{2\mu}\left(r^2 - 2r\|\boldsymbol{x}\| + \|\boldsymbol{x}\|^2 \right) \right\}, \\
&= \begin{cases} \frac{\|\boldsymbol{x}\|^2}{2\mu}, & \text{if } \|\boldsymbol{x}\| \leq \mu, \\ \|\boldsymbol{x}\| - \frac{\mu}{2}, & \text{o.w.}, \end{cases}
\end{aligned}
$$

with the smoothing parameter $(1/2, 1)$.

Example 3 (ℓ_1-norm function). *(2) Consider $\boldsymbol{u} \in \mathbb{R}^m$, $G : \boldsymbol{u} \mapsto \|\boldsymbol{u}\|_1$, and $\mu > 0$. Then, using the fact that $\|\boldsymbol{u}\|_1 := \sum_i |u_i|$, we have $G_\mu(\boldsymbol{u}) := \sum_{i=1}^{m} F_\mu(u_i)$, with the parameter* $(m/2, 1)$.

Example 4 (Switch function). *Consider $u \in \mathbb{R}$, $F^{\mathrm{S}} : u \mapsto \max\{0, 1-u\}$, which is differentiable*

almost everywhere. For a given $\mu > 0$, we compute

$$\begin{aligned} F_\mu^{\mathrm{S}}(u) &:= \min_{z\in\mathbb{R}} \left\{ \max\{0, 1-z\} + \frac{1}{2\mu}\|z-u\|^2 \right\}, \\ &= \min\left\{ \min_{z\le 1} 1 - z + \frac{1}{2\mu}\|z-u\|^2, \min_{z\ge 1} \frac{1}{2\mu}\|z-u\|^2 \right\}. \end{aligned}$$

Given that

$$\min_{z\le 1} 1 - z + \frac{1}{2\mu}\|z-u\|^2 = \begin{cases} \frac{1}{2\mu}\|1-u\|^2, & \text{if } u > 1-\mu, \\ 1 - u - \frac{\mu}{2}, & \text{if } u \le 1-\mu, \end{cases}$$

and

$$\min_{z\ge 1} \frac{1}{2\mu}\|z-u\|^2 = \begin{cases} \frac{1}{2\mu}\|1-u\|^2, & \text{if } u < 1, \\ 0, & \text{if } u \ge 1, \end{cases}$$

resulting in

$$F_\mu^{\mathrm{S}}(u) := \begin{cases} 1 - u - \frac{\mu}{2}, & \text{if } u \le 1-\mu, \\ \frac{1}{2\mu}\|1-u\|^2, & \text{if } 1-\mu \le u < 1, \\ 0, & \text{if } u \ge 1, \end{cases}$$

with the smoothing parameter $(1/2, 1)$.

C.2 Computation of the Objective Gradients

Let ℓ, G and G_μ be those in Lemma 17 on examples of (P2′). We now derive $\nabla G_\mu := \nabla_{\boldsymbol{u}} G_\mu(t, \boldsymbol{u})$ as follows.

Problem 1: (Optimal control under uncertainty)

$$\nabla_{\boldsymbol{u}} G_\mu(t, \boldsymbol{u}) = \frac{1}{\mu}\boldsymbol{u} + \frac{1}{T}\sum_{k\in\mathcal{T}} \nabla_{\boldsymbol{u}} F_\mu(\boldsymbol{p}_{k,t}) + \frac{\gamma}{T}\sum_{i=1}^{p}\sum_{k\in\mathcal{T}} \nabla_{\boldsymbol{u}} F_\mu(H_k^{(i)}),$$

where, for each $k \in \mathcal{T}$, the term $\nabla_{\boldsymbol{u}} F_\mu(\boldsymbol{p}_{k,t})$ is

$$\begin{cases} \frac{1}{\mu}\left(\sum_{i=1}^{p} \alpha_i f_2^{(i)}(t, \hat{\boldsymbol{x}}_t)\right)^\top \boldsymbol{p}_{k,t}, & \text{if } \|\boldsymbol{p}_{k,t}\| \le \mu, \\ \frac{1}{\|\boldsymbol{p}_{k,t}\|}\left(\sum_{i=1}^{p} \alpha_i f_2^{(i)}(t, \hat{\boldsymbol{x}}_t)\right)^\top \boldsymbol{p}_{k,t}, & \text{o.w.}, \end{cases}$$

and, for $k \in \mathcal{T}$, $i \in \{1, \ldots, p\}$, the term $\nabla_{\boldsymbol{u}} F_\mu(H_k^{(i)})$ is

$$\begin{cases} -\frac{1}{\mu}(f_2^{(i)}(t, \hat{\boldsymbol{x}}_t))^\top H_k^{(i)}, & \text{if } \|H_k^{(i)}\| \le \mu, \\ -\frac{1}{\|H_k^{(i)}\|}(f_2^{(i)}(t, \hat{\boldsymbol{x}}_t))^\top H_k^{(i)}, & \text{o.w.}. \end{cases}$$

Problem 2: (Online resource allocation)

$$\nabla_{\boldsymbol{u}} G_\mu(t, \boldsymbol{u}) = \frac{1}{T} \sum_{k \in \mathcal{T}} \nabla_{\boldsymbol{u}} F_\mu^{\mathrm{S}}(\langle \boldsymbol{u}, \boldsymbol{p}_{k,t} \rangle) + q_t \nabla_{\boldsymbol{u}} F_\mu(\boldsymbol{u}),$$

where

$$\nabla_{\boldsymbol{u}} F_\mu(\boldsymbol{u}) := \begin{cases} \frac{1}{\mu}\boldsymbol{u}, & \text{if } \|\boldsymbol{u}\| \le \mu, \\ \frac{1}{\|\boldsymbol{u}\|}\boldsymbol{u}, & \text{o.w.}, \end{cases}$$

and, for each $k \in \mathcal{T}$, the gradient $\nabla_{\boldsymbol{u}} F_\mu^{\mathrm{S}}(\langle \boldsymbol{u}, \boldsymbol{p}_{k,t} \rangle)$ is

$$\begin{cases} -\boldsymbol{p}_{k,t}, & \text{if } \langle \boldsymbol{u}, \boldsymbol{p}_{k,t} \rangle \le 1 - \mu, \\ -\frac{1 - \langle \boldsymbol{u}, \boldsymbol{p}_{k,t} \rangle}{\mu} \boldsymbol{p}_{k,t}, & \text{if } 1 - \mu \le \langle \boldsymbol{u}, \boldsymbol{p}_{k,t} \rangle < 1, \\ 0, & \text{if } \langle \boldsymbol{u}, \boldsymbol{p}_{k,t} \rangle \ge 1. \end{cases}$$

These explicit expressions provide ingredients for the solution system. With different selections of the norm, the expression varies accordingly.

C.3 Stability Analysis of the Solution System

Here, we adapt dissipativity theory to address the performance of the online solution system (5.2). This part of the work is an online-algorithmic extension of the existing Nesterov's accelerated-gradient method and its convergence analysis in [8, 53, 66]. Our extension (5.2) inherits from the work in [66], where the difference is that gradient computations in (5.2) are from time-varying objective functions in (P2′). To simplify the discussion, the notation we used in this section is different from that in the main body of Chapter 5. Consider the online problem, analogous to (P2′), defined as follows

$$\min_{\boldsymbol{x}\in\mathcal{X}} f_t(\boldsymbol{x}), \quad t = 0, 1, 2, \ldots \tag{C.1}$$

where $f_t(\boldsymbol{x})$ is locally Lipschitz in t with the parameter $h(\boldsymbol{x})$ and, at each time t, the objective function f_t are m_t-strongly convex and L_t-smooth, with $m_t \geq 0$ and $L_t > 0$. The convex set $\mathcal{X} \subset \mathbb{R}^n$ is analogous to that in Assumption 15 on convex decision oracle. The solution system to (C.1), analogous to (5.2), is

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \Pi(\boldsymbol{y}_t - \alpha_t \nabla f_t(\boldsymbol{y}_t)), \\ \boldsymbol{y}_{t+1} &= \boldsymbol{x}_{t+1} + \beta_t\,(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t), \\ &\text{with some } \boldsymbol{y}_0 = \boldsymbol{x}_0 \in \mathcal{X}, \end{aligned} \tag{C.2}$$

where $\alpha_t \leq 1/L_t$ and β_t is selected iteratively, following

$$\delta_{-1} = 1,\ \delta_{t+1} := \frac{1 + \sqrt{1 + 4\delta_t^2}}{2},\ \beta_t := \frac{\delta_{t-1} - 1}{\delta_t}.$$

Note that $\delta_t^2 - \delta_t = \delta_{t-1}^2$, $t = 0, 1, 2, \ldots$. The projection $\Pi(\boldsymbol{x})$ at each time t is equivalently written as

$$\Pi(\boldsymbol{x}) = \operatorname*{argmin}_{z\in\mathbb{R}^n} \frac{1}{2}\|z - \boldsymbol{x}\|^2 + \alpha_t \ell(z),$$

with $\ell(\boldsymbol{z}) = 0$ if $\boldsymbol{z} \in \mathcal{X}$, otherwise $+\infty$. Note that the projection operation is a convex problem with the objective function being strongly convex. Thus, $\Pi(\boldsymbol{x})$ is a singleton (the unique minimizer) and satisfies the optimality condition [113]

$$\boldsymbol{x} - \Pi(\boldsymbol{x}) \in \alpha_t \partial \ell(\Pi(\boldsymbol{x})),$$

where the r.h.s. is the sub-differential set of ℓ at $\Pi(\boldsymbol{x})$. Equivalently, we write the above condition as

$$\Pi(\boldsymbol{x}) = \boldsymbol{x} - \alpha_t \partial \ell(\Pi(\boldsymbol{x})).$$

We apply this equivalent representation to the solution system (C.2), resulting in

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \boldsymbol{y}_t - \alpha_t \nabla f_t(\boldsymbol{y}_t) - \alpha_t \partial \ell(\boldsymbol{w}_t), \\ \boldsymbol{y}_{t+1} &= \boldsymbol{x}_{t+1} + \beta_t \, (\boldsymbol{x}_{t+1} - \boldsymbol{x}_t), \\ \boldsymbol{w}_t &= \boldsymbol{x}_{t+1}. \end{aligned} \tag{C.3}$$

Note that (C.3) is not an explicit online algorithm, as the state $\boldsymbol{x}_{t+1}$ is determined implicitly. However, we leverage this equivalent reformulation for the convergence analysis of solutions to (C.2) to a sequence of optimizers of (C.1), denoted by $\{\boldsymbol{x}_t^\star\}$. To do this, let $\boldsymbol{z}_t := (\boldsymbol{x}_t - \boldsymbol{x}_t^\star, \, \boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^\star)$ denote the tracking error vector and represent (C.3) as the error dynamical system

$$\begin{aligned} \boldsymbol{z}_{t+1} &= A_t \boldsymbol{z}_t + B_t^{\boldsymbol{u}} \boldsymbol{u}_t + B_t^{\boldsymbol{v}} \boldsymbol{v}_t, \\ &\text{with } \boldsymbol{z}_1 = (\boldsymbol{x}_1 - \boldsymbol{x}_1^\star, \, \boldsymbol{x}_0 - \boldsymbol{x}_0^\star), \end{aligned} \tag{C.4}$$

with the gradient input $\boldsymbol{u}_t := \nabla f_t(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t)$, the reference signal $\boldsymbol{v}_t := (\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star, \, \boldsymbol{x}_{t+1}^\star - \boldsymbol{x}_t^\star)$, the matrices

$$A_t = \begin{bmatrix} 1+\beta_t & -\beta_t \\ 1 & 0 \end{bmatrix}, \; B_t^{\boldsymbol{u}} = \begin{bmatrix} \alpha_t \\ 0 \end{bmatrix}, \; B_t^{\boldsymbol{v}} = \begin{bmatrix} \beta_t & -1 \\ 0 & 0 \end{bmatrix},$$

and the auxiliary variables

$$\boldsymbol{y}_t - \boldsymbol{x}_t^\star = \begin{bmatrix} 1+\beta_t & -\beta_t \end{bmatrix} \boldsymbol{z}_t + \begin{bmatrix} \beta_t & 0 \end{bmatrix} \boldsymbol{v}_t,$$
$$\boldsymbol{w}_t - \boldsymbol{x}_t^\star = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{z}_{t+1} + \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{v}_t.$$

We provide the following stability analysis of the system.

Theorem 15 (Stability of (C.2)**).** *Consider the solution algorithm* (C.2)*, or equivalently* (C.3)*.*

(1) For each $t \geq 1$, we have the following

$$f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_{t+1}) \geq \boldsymbol{\xi}_t{}^\top X_{1,t}\, \boldsymbol{\xi}_t,$$

$$f_t(\boldsymbol{x}_t^\star) - f_t(\boldsymbol{x}_{t+1}) \geq \boldsymbol{\xi}_t{}^\top X_{2,t}\, \boldsymbol{\xi}_t.$$

Here, $\boldsymbol{\xi}_t := (\boldsymbol{z}_t,\ \boldsymbol{u}_t,\ \boldsymbol{v}_t)$, and

$$X_{1,t} := \frac{1}{2} \begin{pmatrix} m\beta^2 & -m\beta^2 & -\beta & m\beta^2 & 0 \\ -m\beta^2 & m\beta^2 & \beta & -m\beta^2 & 0 \\ -\beta & \beta & \alpha(2-L\alpha) & -\beta & 0 \\ m\beta^2 & -m\beta^2 & -\beta & m\beta^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$X_{2,t} := \frac{1}{2} \begin{pmatrix} m(1+\beta)^2 & -\eta & -(1+\beta) & \eta & 0 \\ -\eta & m\beta^2 & \beta & -m\beta^2 & 0 \\ -(1+\beta) & \beta & \alpha(2-L\alpha) & -\beta & 0 \\ \eta & -m\beta^2 & -\beta & m\beta^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

with $\eta = m(1+\beta)\beta$ *and the parameters* (m, L, α, β) *are a short-hand notation for* $(m_t, L_t, \alpha_t, \beta_t)$.

(2) *Given the horizon parameter* $T_0 \in \mathbb{Z}_{>0}$ *with* $T = \min\{t-1, T_0\}$. *Then, for any* $t \geq 2$, *the solution* $\boldsymbol{x}_t$ *from* (C.2) *achieves*

$$f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_t^\star) \leq \frac{4G_t}{(t+2)^2} + TF_t + TK_t + \frac{4(t-T-1+\delta_0)^2}{(t+2)^2}(f_{t-T}(\boldsymbol{x}_{t-T}) - f_{t-T}(\boldsymbol{x}_{t-T}^\star)).$$

where the time-dependent parameters G_t, F_t *and* K_t *are determined by* f_t, α_t *and* β_t.

Proof. (1) By the m-strong convexity and L-smoothness of f, we have

$$f(\boldsymbol{x}) - f(\boldsymbol{y}) \geq \nabla f(\boldsymbol{y})^\top(\boldsymbol{x}-\boldsymbol{y}) + \frac{m}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2, \tag{C.5}$$

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \geq \nabla f(\boldsymbol{y})^\top(\boldsymbol{y}-\boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2. \tag{C.6}$$

(1a) Consider (C.5) with $(\boldsymbol{x}, \boldsymbol{y}) \equiv (\boldsymbol{x}_t, \boldsymbol{y}_t)$. We leverage $\boldsymbol{y}_t = \boldsymbol{x}_t + \beta(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})$ and the distributive law[1] for

$$\begin{aligned}
f(\boldsymbol{x}_t) - f(\boldsymbol{y}_t) &\geq \beta\nabla f(\boldsymbol{y}_t)^\top(\boldsymbol{x}_{t-1}-\boldsymbol{x}_t) + \frac{m\beta^2}{2}\|\boldsymbol{x}_{t-1}-\boldsymbol{x}_t\|^2, \\
&= \beta(\nabla f(\boldsymbol{y}_t) + \partial\ell(\boldsymbol{w}_t))^\top(\boldsymbol{x}_{t-1}-\boldsymbol{x}_t-\boldsymbol{x}_{t-1}^\star+\boldsymbol{x}_t^\star) + \frac{m\beta^2}{2}\|\boldsymbol{x}_{t-1}-\boldsymbol{x}_t-\boldsymbol{x}_{t-1}^\star+\boldsymbol{x}_t^\star\|^2 \\
&\quad + \beta(\nabla f(\boldsymbol{y}_t) + \partial\ell(\boldsymbol{w}_t))^\top(\boldsymbol{x}_{t-1}^\star-\boldsymbol{x}_t^\star) - \beta\partial\ell(\boldsymbol{w}_t)^\top(\boldsymbol{x}_{t-1}-\boldsymbol{x}_t) \\
&\quad + m\beta^2(\boldsymbol{x}_{t-1}-\boldsymbol{x}_t-\boldsymbol{x}_{t-1}^\star+\boldsymbol{x}_t^\star)^\top(\boldsymbol{x}_{t-1}^\star-\boldsymbol{x}_t^\star) + \frac{m\beta^2}{2}\|\boldsymbol{x}_{t-1}^\star-\boldsymbol{x}_t^\star\|^2, \\
&= \frac{1}{2}\boldsymbol{\delta}_t{}^\top\begin{pmatrix} m\beta^2 & -m\beta^2 & -\beta & m\beta^2 \\ -m\beta^2 & m\beta^2 & \beta & -m\beta^2 \\ -\beta & \beta & 0 & -\beta \\ m\beta^2 & -m\beta^2 & -\beta & m\beta^2 \end{pmatrix}\boldsymbol{\delta}_t - \beta\partial\ell(\boldsymbol{w}_t)^\top(\boldsymbol{x}_{t-1}-\boldsymbol{x}_t),
\end{aligned}$$

[1] Apply 1) $a^\top c = (a+b)^\top(c-d) + (a+b)^\top d - b^\top c$ and 2) $c^\top c = (c-d)^\top(c-d) + 2(c-d)^\top d + d^\top d$, with $a = \nabla f(\boldsymbol{y}_t)$, $b = \partial\ell(\boldsymbol{w}_t)$, $c = \boldsymbol{x}_{t-1} - \boldsymbol{x}_t$, $d = \boldsymbol{x}_{t-1}^\star - \boldsymbol{x}_t^\star$,

with $\boldsymbol{\delta}_t{}^\top := (\boldsymbol{x}_t - \boldsymbol{x}_t^\star, \boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^\star, \nabla f(\boldsymbol{y}_t) + \partial\ell(\boldsymbol{w}_t), \boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star)$.

(1b) Consider (C.6) with $(\boldsymbol{x}, \boldsymbol{y}) \equiv (\boldsymbol{x}_{t+1}, \boldsymbol{y}_t)$. We leverage $\boldsymbol{x}_{t+1} = \boldsymbol{y}_t - \alpha\nabla f_t(\boldsymbol{y}_t) - \alpha\partial\ell(\boldsymbol{w}_t)$ and the distribution law, resulting in

$$\begin{aligned} f(\boldsymbol{y}_t) - f(\boldsymbol{x}_{t+1}) &\geq \alpha\nabla f(\boldsymbol{y}_t)^\top(\nabla f(\boldsymbol{y}_t) + \partial\ell(\boldsymbol{w}_t)) - \frac{L\alpha^2}{2}\|\nabla f(\boldsymbol{y}_t) + \partial\ell(\boldsymbol{w}_t)\|^2, \\ &= \frac{\alpha(2 - L\alpha)}{2}\|\nabla f(\boldsymbol{y}_t) + \partial\ell(\boldsymbol{w}_t)\|^2 - \alpha\partial\ell(\boldsymbol{w}_t)^\top(\nabla f(\boldsymbol{y}_t) + \partial\ell(\boldsymbol{w}_t)). \end{aligned}$$

Now, we sum the terms involving $\partial\ell(\boldsymbol{w}_t)$ in the r.h.s. of inequalities in (1a) and (1b), leverage (C.3), and then apply the convexity of ℓ, $\boldsymbol{x}_t \in \mathcal{X}$ and $\boldsymbol{w}_t = \boldsymbol{x}_{t+1} \in \mathcal{X}$, to obtain the following

$$\begin{aligned} -\beta\partial\ell(\boldsymbol{w}_t)^\top(\boldsymbol{x}_{t-1} - \boldsymbol{x}_t) - \alpha\partial\ell(\boldsymbol{w}_t)^\top(\nabla f(\boldsymbol{y}_t) + \partial\ell(\boldsymbol{w}_t)) &= -\partial\ell(\boldsymbol{w}_t)^\top(\boldsymbol{x}_t - \boldsymbol{w}_t), \\ &\geq \ell(\boldsymbol{w}_t) - \ell(\boldsymbol{x}_t) = 0, \end{aligned}$$

which results in

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}) \geq \boldsymbol{\xi}_t{}^\top X_{1,t}\, \boldsymbol{\xi}_t.$$

Note that we have identified (f, m, L, α, β) with $(f_t, m_t, L_t, \alpha_t, \beta_t)$, and notice $\nabla f_t(\boldsymbol{x}_t^\star) + \partial\ell(\boldsymbol{x}_t^\star) = 0$.

(1c) Similarly, consider (C.5) with $(\boldsymbol{x}, \boldsymbol{y}) \equiv (\boldsymbol{x}_t^\star, \boldsymbol{y}_t)$. From $\boldsymbol{y}_t = \boldsymbol{x}_t + \beta(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})$ and the

distributive law,

$$
\begin{aligned}
f(\boldsymbol{x}_t^\star) - f(\boldsymbol{y}_t) &\geq \nabla f(\boldsymbol{y}_t)^\top (\boldsymbol{x}_t^\star - \boldsymbol{y}_t) + \frac{m}{2}\|\boldsymbol{x}_t^\star - \boldsymbol{y}_t\|^2, \\
&= (\nabla f(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t))^\top (\boldsymbol{x}_t^\star - \boldsymbol{y}_t + \beta \boldsymbol{x}_t^\star - \beta \boldsymbol{x}_{t-1}^\star) - \partial \ell(\boldsymbol{w}_t)^\top (\boldsymbol{x}_t^\star - \boldsymbol{y}_t) \\
&\quad - \beta (\nabla f(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t))^\top (\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star) + \frac{m}{2}\| - (1+\beta)(\boldsymbol{x}_t - \boldsymbol{x}_t^\star) + \beta(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^\star)\|^2 \\
&\quad - m\beta[-(1+\beta)(\boldsymbol{x}_t - \boldsymbol{x}_t^\star) + \beta(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^\star)]^\top (\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star) + \frac{m\beta^2}{2}\|\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star\|^2, \\
&= \frac{1}{2}{\boldsymbol{\delta}_t}^\top \begin{pmatrix} m(1+\beta)^2 & -\eta & -(1+\beta) & \eta \\ -\eta & m\beta^2 & \beta & -m\beta^2 \\ -(1+\beta) & \beta & 0 & -\beta \\ \eta & -m\beta^2 & -\beta & m\beta^2 \end{pmatrix} \boldsymbol{\delta}_t - \partial \ell(\boldsymbol{w}_t)^\top (\boldsymbol{x}_t^\star - \boldsymbol{y}_t),
\end{aligned}
$$

with $\eta = m(1+\beta)\beta$. We add this inequality to that in (1b) and leverage

$$
\begin{aligned}
-\partial \ell(\boldsymbol{w}_t)^\top (\boldsymbol{x}_t^\star - \boldsymbol{y}_t) - \alpha \partial \ell(\boldsymbol{w}_t)^\top (\nabla f(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t)) &= -\partial \ell(\boldsymbol{w}_t)^\top (\boldsymbol{x}_t^\star - \boldsymbol{w}_t), \\
&\geq \ell(\boldsymbol{w}_t) - \ell(\boldsymbol{x}_t^\star) = 0,
\end{aligned}
$$

resulting in $f(\boldsymbol{x}_t^\star) - f(\boldsymbol{x}_{t+1}) \geq {\boldsymbol{\xi}_t}^\top X_{2,t}\, \boldsymbol{\xi}_t$. By the definition of the m-strong convexity and L-smoothness of f_t, $\forall t$, we have

$$
f_t(\boldsymbol{x}) - f_t(\boldsymbol{y}) \geq \nabla f_t(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) + \frac{m}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2, \tag{C.7}
$$

$$
f_t(\boldsymbol{y}) - f_t(\boldsymbol{x}) \geq \nabla f_t(\boldsymbol{y})^\top (\boldsymbol{y} - \boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2, \tag{C.8}
$$

Consider (C.7) with $(\boldsymbol{x}, \boldsymbol{y}) \mapsto (\boldsymbol{x}_t, \boldsymbol{y}_t)$, we leverage $\boldsymbol{y}_t = \boldsymbol{x}_t + \beta\,(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})$ and the distribution

law[2] for

$$
\begin{aligned}
f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{y}_t) &\geq \beta \nabla f_t(\boldsymbol{y}_t)^\top (\boldsymbol{x}_{t-1} - \boldsymbol{x}_t) + \frac{m\beta^2}{2} \|\boldsymbol{x}_{t-1} - \boldsymbol{x}_t\|^2, \\
&= \frac{1}{2} {\boldsymbol{\xi}_t}^\top \left[\begin{pmatrix} m\beta^2 & -m\beta^2 & -\beta \\ -m\beta^2 & m\beta^2 & \beta \\ -\beta & \beta & 0 \end{pmatrix} \otimes I_n \right] \boldsymbol{\xi}_t + \beta (\nabla f_t(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t))^\top (\boldsymbol{x}^\star_{t-1} - \boldsymbol{x}^\star_t) \\
&\quad - \beta \partial \ell(\boldsymbol{w}_t)^\top (\boldsymbol{x}_{t-1} - \boldsymbol{x}_t) + m\beta^2 (\boldsymbol{x}_{t-1} - \boldsymbol{x}_t)^\top (\boldsymbol{x}^\star_{t-1} - \boldsymbol{x}^\star_t) - \frac{m\beta^2}{2} \|\boldsymbol{x}^\star_{t-1} - \boldsymbol{x}^\star_t\|^2,
\end{aligned}
$$

where ${\boldsymbol{\xi}_t}^\top := \left(\boldsymbol{x}_t - \boldsymbol{x}^\star_t, \quad \boldsymbol{x}_{t-1} - \boldsymbol{x}^\star_{t-1}, \quad \nabla f_t(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t) \right)$.

Now, we consider (C.8) with $(\boldsymbol{x}, \boldsymbol{y}) \mapsto (\boldsymbol{x}_{t+1}, \boldsymbol{y}_t)$, leverage $\boldsymbol{x}_{t+1} = \boldsymbol{y}_t - \alpha \nabla f_t(\boldsymbol{y}_t) - \alpha \partial \ell(\boldsymbol{w}_t)$ and add-subtract terms that are related to $\partial \ell(\boldsymbol{w}_t)$, resulting in

$$
\begin{aligned}
f_t(\boldsymbol{y}_t) - f_t(\boldsymbol{x}_{t+1}) &\geq \alpha \nabla f_t(\boldsymbol{y}_t)^\top (\nabla f_t(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t)) - \frac{L\alpha^2}{2} \|\nabla f_t(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t)\|^2, \\
&= \frac{\alpha(2 - L\alpha)}{2} \|\nabla f_t(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t)\|^2 - \alpha \partial \ell(\boldsymbol{w}_t)^\top (\nabla f_t(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t)).
\end{aligned}
$$

We add the above two inequalities and leverage (C.3) for the following fact

$$
\beta (\nabla f_t(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t)) = \frac{\beta}{\alpha} (\boldsymbol{x}_t - \boldsymbol{x}_{t+1}) + \frac{\beta^2}{\alpha} (\boldsymbol{x}_t - \boldsymbol{x}_{t-1}),
$$

and by convexity of ℓ, $\boldsymbol{x}_t \in \mathcal{X}$ and $\boldsymbol{w}_t = \boldsymbol{x}_{t+1} \in \mathcal{X}$,

$$
- \beta \partial \ell(\boldsymbol{w}_t)^\top (\boldsymbol{x}_{t-1} - \boldsymbol{x}_t) - \alpha \partial \ell(\boldsymbol{w}_t)^\top (\nabla f_t(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t)) = - \partial \ell(\boldsymbol{w}_t)^\top (\boldsymbol{x}_t - \boldsymbol{w}_t) \geq 0,
$$

[2] Apply 1) $a^\top c = (a+b)^\top (c-d) + (a+b)^\top d - b^\top c$ and 2) $c^\top c = (c-d)^\top (c-d) + 2c^\top d - d^\top d$, with $a = \nabla f_t(\boldsymbol{y})$, $b = \partial \ell(\boldsymbol{w}_t)$, $c = \boldsymbol{x}_{t-1} - \boldsymbol{x}_t$, $d = \boldsymbol{x}^\star_{t-1} - \boldsymbol{x}^\star_t$,

resulting in

$$f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_{t+1}) \geq \frac{1}{2}{\boldsymbol{\xi}_t}^\top \left[\begin{pmatrix} m\beta^2 & -m\beta^2 & -\beta \\ -m\beta^2 & m\beta^2 & \beta \\ -\beta & \beta & \alpha(2-L\alpha) \end{pmatrix} \otimes I_n \right] \boldsymbol{\xi}_t$$
$$+ \frac{1}{2}{\boldsymbol{\eta}_t}^\top \left[\begin{pmatrix} 0 & 0 & -(\frac{1}{\alpha}-m)\beta^2 \\ 0 & 0 & \frac{\beta}{\alpha} \\ -(\frac{1}{\alpha}-m)\beta^2 & \frac{\beta}{\alpha} & -m\beta^2 \end{pmatrix} \otimes I_n \right] \boldsymbol{\eta}_t,$$

where ${\boldsymbol{\eta}_t}^\top := \begin{pmatrix} \boldsymbol{x}_t - \boldsymbol{x}_{t-1}, & \boldsymbol{x}_{t+1} - \boldsymbol{x}_t, & \boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star \end{pmatrix}$.

Similarly, we consider (C.7) with $(\boldsymbol{x}, \boldsymbol{y}) \mapsto (\boldsymbol{x}_t^\star, \boldsymbol{y}_t)$, leverage $\boldsymbol{y}_t = \boldsymbol{x}_t + \beta\,(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})$ and the distribution law, resulting in

$$f_t(\boldsymbol{x}_t^\star) - f_t(\boldsymbol{y}_t) \geq \nabla f_t(\boldsymbol{y}_t)^\top (\boldsymbol{x}_t^\star - \boldsymbol{y}_t) + \frac{m}{2}\|\boldsymbol{x}_t^\star - \boldsymbol{y}_t\|^2,$$
$$= \frac{1}{2}{\boldsymbol{\xi}_t}^\top \left[\begin{pmatrix} m(1+\beta)^2 & -m(1+\beta)\beta & -(1+\beta) \\ -m(1+\beta)\beta & m\beta^2 & \beta \\ -(1+\beta) & \beta & 0 \end{pmatrix} \otimes I_n \right] \boldsymbol{\xi}_t$$
$$- \beta(\nabla f_t(\boldsymbol{y}_t) + \partial\ell(\boldsymbol{w}_t))^\top (\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star) - \partial\ell(\boldsymbol{w}_t)^\top (\boldsymbol{x}_t^\star - \boldsymbol{y}_t)$$
$$- m\beta(\boldsymbol{x}_t^\star - \boldsymbol{y}_t)^\top (\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star) - \frac{m\beta^2}{2}\|\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star\|^2,$$

and we add this inequality to the one related to $f_t(\boldsymbol{y}_t) - f_t(\boldsymbol{x}_{t+1})$, resulting in

$$
\begin{aligned}
f_t(\boldsymbol{x}_t^\star) - f_t(\boldsymbol{x}_{t+1}) \geq \frac{1}{2}\boldsymbol{\xi}_t{}^\top & \left[\begin{pmatrix} m(1+\beta)^2 & -m(1+\beta)\beta & -(1+\beta) \\ -m(1+\beta)\beta & m\beta^2 & \beta \\ -(1+\beta) & \beta & \alpha(2-L\alpha) \end{pmatrix} \otimes I_n \right] \boldsymbol{\xi}_t \\
& + \frac{1}{2}\boldsymbol{\eta}_t{}^\top \left[\begin{pmatrix} 0 & 0 & -(\frac{1}{\alpha}-m)\beta^2 \\ 0 & 0 & \frac{\beta}{\alpha} \\ -(\frac{1}{\alpha}-m)\beta^2 & \frac{\beta}{\alpha} & -m\beta^2 \end{pmatrix} \otimes I_n \right] \boldsymbol{\eta}_t \\
& + m\beta(\boldsymbol{x}_t - \boldsymbol{x}_t^\star)^\top(\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star).
\end{aligned}
$$

(2) Let us define the time varying function

$$
V_t(z_t) := \begin{bmatrix} z_t \\ \boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star \end{bmatrix}^\top H_t \begin{bmatrix} z_t \\ \boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star \end{bmatrix},
$$

where we take

$$
H_t := \frac{1}{2\alpha_{t-1}} \begin{bmatrix} \delta_{t-1} \\ 1-\delta_{t-1} \\ \delta_{t-1} \end{bmatrix} \begin{bmatrix} \delta_{t-1}, & 1-\delta_{t-1}, & \delta_{t-1} \end{bmatrix},
$$

with $\{\alpha_t\}_t$ those in the solution system (C.2) and $\{\delta_t\}_t$ the sequence of scalars which defines $\{\beta_t\}_t$. Now, verify

$$
V_{t+1}(z_{t+1}) - \frac{\alpha_{t-1}}{\alpha_t} V_t(z_t) = \boldsymbol{\xi}_t{}^\top J_t \boldsymbol{\xi}_t,
$$

where $\boldsymbol{\xi}_t := (z_t,\ \boldsymbol{u}_t,\ \boldsymbol{v}_t)$, which are those define (C.4), resulting in $\boldsymbol{\xi}_t := (\boldsymbol{x}_t - \boldsymbol{x}_t^\star,\ \boldsymbol{x}_{t-1} -$

$\boldsymbol{x}_{t-1}^\star, \nabla f_t(\boldsymbol{y}_t) + \partial \ell(\boldsymbol{w}_t), \boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star,\ \boldsymbol{x}_{t+1}^\star - \boldsymbol{x}_t^\star)$ and

$$J_t = \frac{1}{2\alpha_t}\begin{pmatrix} 0 & 0 & -\alpha_t\delta_t\delta_{t-1} & -\delta_{t-1} & 0 \\ 0 & 0 & \alpha_t\beta_t\delta_t^2 & \beta_t\delta_t & 0 \\ -\alpha_t\delta_t\delta_{t-1} & \alpha_t\beta_t\delta_t^2 & \alpha_t^2\delta_t^2 & -\alpha_t\beta_t\delta_t^2 & 0 \\ -\delta_{t-1} & \beta_t\delta_t & -\alpha_t\beta_t\delta_t^2 & 1-2\delta_{t-1} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Let us compute

$$\begin{aligned} M_t :=& \delta_{t-1}^2 X_{1,t} + \delta_t X_{2,t} \\ =& \frac{1}{2}\begin{pmatrix} m_t(\delta_t^2-1) & -m_t\beta_t\delta_t\delta_{t-1} & -\delta_t\delta_{t-1} & m_t\beta_t\delta_t\delta_{t-1} & 0 \\ -m_t\beta_t\delta_t\delta_{t-1} & m\beta_t^2\delta_t^2 & \beta_t\delta_t^2 & -m_t\beta_t^2\delta_t^2 & 0 \\ -\delta_t\delta_{t-1} & \beta_t\delta_t^2 & \alpha_t(2-L_t\alpha_t)\delta_t^2 & -\beta_t\delta_t^2 & 0 \\ m_t\beta_t\delta_t\delta_{t-1} & -m_t\beta_t^2\delta_t^2 & -\beta_t\delta_t^2 & m_t\beta_t^2\delta_t^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \end{aligned}$$

and then achieve

$$\begin{aligned} \boldsymbol{\xi}_t{}^\top (J_t - M_t)\boldsymbol{\xi}_t =& \begin{bmatrix} \boldsymbol{z}_t \\ \boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star \end{bmatrix}^\top N_{1,t} \begin{bmatrix} \boldsymbol{z}_t \\ \boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star \end{bmatrix} \\ &+ \begin{bmatrix} \boldsymbol{z}_t \\ \boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star \end{bmatrix}^\top N_{2,t} \begin{bmatrix} \boldsymbol{z}_t \\ \boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star \end{bmatrix} - \alpha_t(1-L_t\alpha_t)\boldsymbol{u}_t{}^\top \boldsymbol{u}_t, \end{aligned}$$

with, for each $t \geq 1$,

$$N_{1,t} := \frac{1}{2}\begin{pmatrix} -m_t(\delta_t^2 - 1) & m_t\beta_t\delta_t\delta_{t-1} & -m_t\beta_t\delta_t\delta_{t-1} \\ m_t\beta_t\delta_t\delta_{t-1} & -m_t\beta_t^2\delta_t^2 & m_t\beta_t^2\delta_t^2 \\ -m_t\beta_t\delta_t\delta_{t-1} & m_t\beta_t^2\delta_t^2 & -m_t\beta_t^2\delta_t^2 \end{pmatrix},$$
$$\cong \frac{m_t}{2}\begin{pmatrix} -(\delta_t^2 - 1) & \beta_t\delta_t\delta_{t-1} & 0 \\ \beta_t\delta_t\delta_{t-1} & -\beta_t^2\delta_t^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \leq 0,$$

and, using the fact that $\delta_t > (t+1)/2$, $\forall\, t \geq 0$, we have

$$N_{2,t} := \frac{1}{2}\begin{pmatrix} 0 & 0 & -\delta_{t-1} \\ 0 & 0 & \beta_t\delta_t \\ -\delta_{t-1} & \beta_t\delta_t & 1 - 2\delta_{t-1} \end{pmatrix} \leq 0.$$

Then, if we select $\alpha_t \leq 1/L_t$, it results in

$$\boldsymbol{\xi}_t{}^{\top}(J_t - M_t)\boldsymbol{\xi}_t \leq 0.$$

We rewrite it as

$$\begin{aligned} V_{t+1}(\boldsymbol{z}_{t+1}) - \frac{\alpha_{t-1}}{\alpha_t}V_t(z_t) &\leq \boldsymbol{\xi}_t{}^{\top} M_t \boldsymbol{\xi}_t, \\ &\leq \delta_{t-1}^2(f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_{t+1})) + \delta_t(f_t(\boldsymbol{x}_t^{\star}) - f_t(\boldsymbol{x}_{t+1})), \\ &= -\delta_t^2(f_t(\boldsymbol{x}_{t+1}) - f_t(\boldsymbol{x}_t^{\star})) + \delta_{t-1}^2(f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_t^{\star})). \end{aligned}$$

As f_t being locally Lipschitz in t, there exists a non-negative function $h(\boldsymbol{x})$ such that

$$f_{t+1}(\boldsymbol{x}_{t+1}) - f_t(\boldsymbol{x}_{t+1}) \leq h(\boldsymbol{x}_{t+1}),$$

resulting in

$$\begin{aligned} V_{t+1}(\boldsymbol{z}_{t+1}) - \frac{\alpha_{t-1}}{\alpha_t} V_t(\boldsymbol{z}_t) \leq & -\delta_t^2 (f_{t+1}(\boldsymbol{x}_{t+1}) - f_{t+1}(\boldsymbol{x}_{t+1}^\star)) + \delta_{t-1}^2 (f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_t^\star)) \\ & - \delta_t^2 (f_{t+1}(\boldsymbol{x}_{t+1}^\star) - f_t(\boldsymbol{x}_t^\star)) + \delta_t^2 h(\boldsymbol{x}_{t+1}), \ \forall\, t. \end{aligned}$$

Summing up above set of inequalities over the moving horizon window $t \in \mathcal{T} = \{t-1, \ldots, t-T\}$, where $T = \min\{t-1, T_0\}$ with some $T_0 \in \mathbb{Z}_{>0}$, we obtain

$$\begin{aligned} V_t(\boldsymbol{z}_t) + \sum_{k \in \mathcal{T}} (1 - \frac{\alpha_{k-1}}{\alpha_k}) V_k(\boldsymbol{z}_k) - V_{t-T}(\boldsymbol{z}_{t-T}) \leq & -\delta_{t-1}^2 (f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_t^\star)) \\ & + \delta_{t-T-1}^2 (f_{t-T}(\boldsymbol{x}_{t-T}) - f_{t-T}(\boldsymbol{x}_{t-T}^\star)) \\ & - \sum_{k \in \mathcal{T}} \delta_k^2 (f_{k+1}(\boldsymbol{x}_{k+1}^\star) - f_k(\boldsymbol{x}_k^\star)) + \sum_{k \in \mathcal{T}} \delta_k^2 h(\boldsymbol{x}_{k+1}). \end{aligned}$$

Let us denote by G_t, K_t, and F_t, respectively, the horizon accumulated potential, the bound of the locally Lipschitz function h, and the variation bound of the optimal objective values. That is,

$$\begin{aligned} G_t :=& V_{t-T}(\boldsymbol{z}_{t-T}) - V_t(\boldsymbol{z}_t) - \sum_{k \in \mathcal{T}} (1 - \frac{\alpha_{k-1}}{\alpha_k}) V_k(\boldsymbol{z}_k), \\ K_t :=& \max_{k \in \mathcal{T}} \{h(\boldsymbol{x}_{k+1})\}, \\ F_t :=& \max_{k \in \in \mathcal{T}} \left\{ |f_{k+1}(\boldsymbol{x}_{k+1}^\star) - f_k(\boldsymbol{x}_k^\star)| \right\}. \end{aligned}$$

Then, using the fact that **(1)** $\delta_{t-1} \geq (t+2)/2$, for all $t \geq 0$; **(2)** $\delta_{t-T-1} \leq t - T - 1 + \delta_0$ with $\delta_0 = (1+\sqrt{5})/2$, and **(3)** δ_t is monotonically increasing, we have

$$f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_t^\star) \leq \frac{4G_t}{(t+2)^2} + TF_t + TK_t + \frac{4(t-T-1+\delta_0)^2}{(t+2)^2} (f_{t-T}(\boldsymbol{x}_{t-T}) - f_{t-T}(\boldsymbol{x}_{t-T}^\star)).$$

Note that, when $t \leq T_0 + 1$, we have $T = t - 1$. This gives

$$f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_t^\star) \leq \frac{4G_t}{(t+2)^2} + (t-1)F_t + (t-1)K_t + \frac{4\delta_0^2}{(t+2)^2}(f_1(\boldsymbol{x}_1) - f_1(\boldsymbol{x}_1^\star)).$$

□

Bibliography

[1] David Aikman, Mirta Galesic, Gerd Gigerenzer, Sujit Kapadia, Konstantinos V Katsikopoulos, Amit Kothiyal, Emma Murphy, and Tobias Neumann. Taking uncertainty seriously: simplicity versus complexity in financial regulation. *Bank of England Financial Stability Paper*, (28), 2014.

[2] A. Allibhoy and J. Cortés. Data-based receding horizon control of linear network systems. *IEEE Control Systems Letters*, 5(4):1207–1212, 2020.

[3] S. Amin, A. Cardenas, and S. S. Sastry. Safe and secure networked control systems under denial-of-service attacks. In *Hybrid systems: Computation and Control*, page 31–45, 2009.

[4] M. Athans. The role and use of the stochastic linear-quadratic-Gaussian problem in control system design. *IEEE Transactions on Automatic Control*, 16(6):529–552, 1971.

[5] C. Bai, F. Pasqualetti, and V. Gupta. Security in stochastic control systems: Fundamental limitations and performance bounds. In *American Control Conference*, pages 195–200, 2015.

[6] C. Bai, F. Pasqualetti, and V. Gupta. Data-injection attacks in stochastic control systems: Detectability and performance tradeoffs. *Automatica*, 82:251–260, 2017.

[7] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

[8] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[9] A. Beck and M. Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

[10] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[11] J. Berberich, J. Köhler, M. Muller, and F. Allgower. Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, 2020.

[12] D. P. Bertsekas. *Convex optimization algorithms*. Athena Scientific Belmont, 2015.

[13] D. P. Bertsekas, A. Nedić, and A. Ozdaglar. *Convex analysis and optimization*. Athena Scientific, 2003.

[14] D. Boskos, J. Cortés, and S. Martínez. Dynamic evolution of distributional ambiguity sets and precision tradeoffs in data assimilation. In *European Control Conference*, pages 2252–2257, Naples, Italy, June 2019.

[15] D. Boskos, J. Cortés, and S. Martinez. Data-driven ambiguity sets for linear systems under disturbances and noisy observations. In *American Control Conference*, pages 4491–4496, Denver, CO, July 2020.

[16] D. Boskos, J. Cortés, and S. Martínez. Data-driven ambiguity sets with probabilistic guarantees for dynamic processes. *IEEE Transactions on Automatic Control*, 2020. To appear.

[17] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[18] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009.

[19] Giuseppe C Calafiore and Lorenzo Fagiano. Robust model predictive control via scenario optimization. *IEEE Transactions on Automatic Control*, 58(1):219–224, 2012.

[20] Mark Cannon, Basil Kouvaritakis, Saša V Raković, and Qifeng Cheng. Stochastic tubes in model predictive control with probabilistic constraints. *IEEE Transactions on Automatic Control*, 56(1):194–200, 2010.

[21] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry. Challenges for securing cyber physical systems. In *Workshop on Future Directions of Cyber-Physical Systems*, volume 5, 2009.

[22] M. Morari C.E. Garcia, D.M. Prett. Model predictive control: theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[23] N. Chen, A. Agarwal, A. Wierman, S. Barman, and L Andrew. Online convex optimization using predictions. In *ACM Int. Conf. on Measurement and Modeling of Computer Systems*, pages 191–204, 2015.

[24] A. Cherukuri and J. Cortés. Data-driven distributed optimization using Wasserstein ambiguity sets. In *Allerton Conf. on Communications, Control and Computing*, pages 38–44, Monticello, IL, 2017.

[25] A. Cherukuri and J. Cortés. Cooperative data-driven distributionally robust optimization. *IEEE Transactions on Automatic Control*, 65(10):4400–4407, 2020.

[26] S. Coogan and M. Arcak. A compartmental model for traffic networks and its dynamical behavior. *IEEE Transactions on Automatic Control*, 60(10):2698–2703, 2015.

[27] Samuel Coogan, Ebru Aydin Gol, Murat Arcak, and Calin Belta. Traffic network control from temporal logic specifications. *IEEE Transactions on Control of Network Systems*, 3(2):162–172, 2016.

[28] J. Cortés. Coverage optimization and spatial load balancing by robotic sensor networks. *IEEE Transactions on Automatic Control*, 55(3):749–754, 2010.

[29] J. Coulson, J. Lygeros, and F. Dörfler. Data-enabled predictive control: In the shallows of the DeePC. In *European Control Conference*, pages 307–312, 2019.

[30] J. Coulson, J. Lygeros, and F. Dörfler. Distributionally robust chance constrained data-enabled predictive control. *ArXiv:2006.01702*, 2020.

[31] N. Dadkhah and B. Mettler. Survey of motion planning literature in the presence of uncertainty: considerations for UAV guidance. *Journal of Intelligent and Robotic Systems*, 65(1-4):233–246, 2012.

[32] C. Daganzo. *Fundamentals of transportation and traffic operations*, volume 30. Oxford, UK, 1997.

[33] C. F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994.

[34] K.C. Dey, L. Yan, X. Wang, Y. Wang, H. Shen, M. Chowdhury, L. Yu, C. Qiu, and V. Soundararaj. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC). *IEEE Transactions on Intelligent Transportation Systems*, 17(2):491–509, 2015.

[35] E. Erdoğan and G. Iyengar. Ambiguous chance constrained problems and robust optimization. *Mathematical Programming*, 107(1-2):37–61, 2006.

[36] P. M. Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1-2):115–166, 2018.

[37] Marcello Farina, Luca Giulioni, and Riccardo Scattolini. Stochastic linear model predictive control with chance constraints–a review. *Journal of Process Control*, 44:53–67, 2016.

[38] S. Fattahi, N. Matni, and S. Sojoudi. Learning sparse dynamical systems from a single sample trajectory. In *IEEE Int. Conf. on Decision and Control*, pages 2682–2689, 2019.

[39] H. J. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit mpc. *International Journal on Robust and Nonlinear Control*, 18(8):816–830, 2008.

[40] N. Fournier and A. Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3-4):707–738, 2015.

[41] R. Gao and A.J. Kleywegt. Distributionally robust stochastic optimization with Wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016.

[42] Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.

[43] G. Gomes, R. Horowitz, A. A. Kurzhanskiy, P. Varaiya, and J. Kwon. Behavior of the cell transmission model and effectiveness of ramp metering. *Transportation Research Part C: Emerging Technologies*, 16(4):485–513, 2008.

[44] Andrew Gray, Yiqi Gao, Theresa Lin, Karl J Hedrick, and Francesco Borrelli. Stochastic predictive control for semi-autonomous vehicles with an uncertain driver model. In *IEEE Int. Conf. on Intelligent Transportation Systems*, pages 2329–2334, 2013.

[45] A. Hakobyan and I. Yang. Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk. *arXiv preprint arXiv:2001.04727*, 2020. Available at https://arxiv.org/abs/2001.04727.

[46] E.C. Hall and R.M. Willett. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, 2015.

[47] Yu Han, Andreas Hegyi, Yufei Yuan, Serge Hoogendoorn, Markos Papageorgiou, and Claudio Roncoli. Resolving freeway jam waves by discrete first-order model-based predictive control of variable speed limits. *Transportation Research Part C: Emerging Technologies*, 77:405–420, 2017.

[48] A. C. Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

[49] E. Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

[50] A. Hegyi, B. D. Schutter, and J. Heelendoorn. MPC-based optimal coordination of variable speed limits to suppress shock waves in freeway traffic. In *American Control Conference*, volume 5, pages 4083–4088, 2003.

[51] J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen. Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transportation Research Part C: Emerging Technologies*, 18(4):568–583, 2010.

[52] C. Holloway. An extension of the Frank and Wolfe method of feasible directions. *Mathematical Programming*, 6(1):14–27, 1974.

[53] B. Hu and L. Lessard. Dissipativity theory for Nesterov's accelerated method. *arXiv preprint arXiv:1706.04381*, 2017.

[54] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan. Online optimization: Competing with dynamic comparators. In *Artificial Intelligence and Statistics*, pages 398–406, 2015.

[55] S. Jafari and K. Savla. On structural properties of feedback optimal control of traffic flow under the cell transmission model. *arXiv preprint arXiv:1805.11271*, 2018.

[56] A. Jamshidnejad, I. Papamichail, M. Papageorgiou, and B.D. Schutter. Sustainable model-predictive control in urban traffic networks: efficient solution based on general smoothening methods. *IEEE Transactions on Control Systems Technology*, 2017.

[57] R. Jiang and Y. Guan. Data-driven chance constrained stochastic program. *Mathematical Programming*, 158(1-2):291–327, 2016.

[58] S.L. Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *Advances in Neural Information Processing Systems*, page 496–504, 2015.

[59] Leonid Vasilevich Kantorovich and Gennady S Rubinstein. On a space of completely additive functions. *Vestnik Leningrad. Univ*, 13(7):52–59, 1958.

[60] A. A. Kurzhanskiy and P. Varaiya. Active traffic management on road networks: a macroscopic approach. *Philosophical Transactions of the Royal Society A*, 368(1928):4607–4626, 2010.

[61] A. A. Kurzhanskiy and P. Varaiya. Guaranteed prediction and estimation of the state of a road network. *Transportation Research Part C: Emerging Technologies*, 21(1):163–180, 2012.

[62] DM De la Penad, Alberto Bemporad, and Teodoro Alamo. Stochastic programming applied to model predictive control. In *IEEE Int. Conf. on Decision and Control*, pages 1361–1366, 2005.

[63] S. M. LaValle. *Planning algorithms*. Cambridge University Press, 2006.

[64] J. Lei. Convergence and concentration of empirical measures under Wasserstein distance in unbounded functional spaces. *arXiv preprint arXiv:1804.10556*, 2018.

[65] C. Lemaréchal and C. Sagastizábal. Practical aspects of the Moreau-Yosida regularization: Theoretical preliminaries. *SIAM Journal on Optimization*, 7(2):367–385, 1997.

[66] L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

[67] D. Li, D. Fooladivanda, and S. Martínez. Data-driven variable speed limit design for highways via distributionally robust optimization. In *European Control Conference*, pages 1055–1061, Napoli, Italy, June 2019.

[68] D. Li, D. Fooladivanda, and S. Martínez. Online learning of parameterized uncertain dynamical environments with finite-sample guarantees. *IEEE Control Systems Letters*, 2021. To appear.

[69] D. Li and X. Li. Domain reduction for Benders decomposition based global optimization. *Computers & Chemical Engineering*, 93:248–265, 2016.

[70] D. Li and S. Martínez. Online data assimilation in distributionally robust optimization. In *IEEE Int. Conf. on Decision and Control*, pages 1961–1966, Miami, FL, USA, December 2018.

[71] D. Li and S. Martínez. Online optimization and data assimilation with performance guarantees. *IEEE Transactions on Automatic Control*, 2021. To appear 2021. Extended version at arXiv:1901.07377.

[72] Xiang Li, Asgeir Tomasgard, and Paul I Barton. Nonconvex generalized Benders decomposition for stochastic separable mixed-integer nonlinear programs. *Journal of Optimization Theory & Applications*, 151(3):425, 2011.

[73] Y. Li, G. Qu, and N. Li. Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit. *arXiv preprint arXiv:1801.07780*, 2018.

[74] M.J. Lighthill and G.B. Whitham. On kinematic waves. II. a theory of traffic flow on long crowded roads. *Royal Society of London. Proceedings Series A: Mathematical, Physical and Engineering Sciences*, 229(1178):317–345, 1955.

[75] S. Liu, A. Sadowska, J. Frejo, A. Núñez, E. Camacho, H. Hellendoorn, and B. De Schutter. Robust receding horizon parameterized control for multi-class freeway networks: A tractable scenario-based approach. *International Journal on Robust and Nonlinear Control*, 26(6):1211–1245, 2016.

[76] L. Ljung. *System identification*. Prentice Hall, 1999.

[77] P.A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. WieBner. Microscopic traffic simulation using SUMO. In *IEEE Int. Conf. on Intelligent Transportation Systems*, pages 2575–2582, 2018.

[78] B. D. Luders, M. Kothari, and J. P. How. Chance constrained RRT for probabilistic robustness to environmental uncertainty. *AIAA Conf. on Guidance, Navigation and Control*, 2010.

[79] Jan M Maciejowski, Andrea Lecchini Visintini, and John Lygeros. Nmpc for complex stochastic systems using a markov chain monte carlo approach. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, pages 269–281. 2007.

[80] L. Magni, G. De Nicolao, R. Scattolini, and F. Allgöwer. Robust model predictive control for nonlinear discrete-time systems. *International Journal on Robust and Nonlinear Control*, 13(3-4):229–246, 2003.

[81] T. Maupong and P. Rapisarda. Data-driven control: A behavioral approach. *Systems and Control Letters*, 101:37–43, 2017.

[82] D. Q. Mayne, E. C. Kerrigan, W. E. Van, and P. Falugi. Tube-based robust nonlinear model predictive control. *International Journal on Robust and Nonlinear Control*, 21(11):1341–1353, 2011.

[83] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36(6):789–814, 2000.

[84] David Mayne. Robust and stochastic model predictive control: Are we going in the right direction? *Annual Reviews in Control*, 41:184–192, 2016.

[85] G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I—convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

[86] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016.

[87] Ali Mesbah, Stefan Streif, Rolf Findeisen, and Richard D Braatz. Stochastic nonlinear model predictive control with probabilistic constraints. In *American Control Conference*, pages 2413–2419, 2014.

[88] F. Miao, Q. Zhu, M. Pajic, and G.J. Pappas. Coding schemes for securing cyber-physical systems against stealthy data injection attacks. *IEEE Transactions on Control of Network Systems*, 4(1):106–117, 2016.

[89] M. Milanese and C. Novara. Unified set membership theory for identification, prediction and filtering of nonlinear systems. *Automatica*, 47(10):2141–2151, 2011.

[90] J. Milošević, D. Umsonst, H. Sandberg, and K. Johansson. Quantifying the impact of cyber-attack strategies for control systems equipped with an anomaly detector. In *European Control Conference*, pages 331–337, 2018.

[91] S. Mishra, Y. Shoukry, N. Karamchandani, S. Diggavi, and P. Tabuada. Secure state estimation against sensor attacks in the presence of noise. *IEEE Transactions on Control of Network Systems*, 4(1):49–59, 2016.

[92] Y. Mo and B. Sinopoli. Secure control against replay attacks. In *Allerton Conf. on Communications, Control and Computing*, pages 911–918, Illinois, USA, September 2009.

[93] Y. Mo and B. Sinopoli. On the performance degradation of cyber-physical systems under stealthy integrity attacks. *IEEE Transactions on Automatic Control*, 61(9):2618–2624, 2015.

[94] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *IEEE Int. Conf. on Decision and Control*, pages 7195–7201, 2016.

[95] M. Morari and Jay H Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682, 1999.

[96] C. Murguia, N. Van de Wouw, and J. Ruths. Reachable sets of hidden CPS sensor attacks: Analysis and synthesis tools. *IFAC World Congress*, 50(1):2088–2094, 2017.

[97] C. Murguia and J. Ruths. CUSUM and Chi-squared attack detection of compromised sensors. In *IEEE Conf. on Control Applications*, pages 474–480, 2016.

[98] C. Murguia, I. Shames, J. Ruths, and D. Nesic. Security metrics of networked control systems under sensor attacks. *arXiv preprint arXiv:1809.01808*, 2018.

[99] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[100] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[101] J. Niles-Weed and P. Rigollet. Estimation of Wasserstein distances in the spiked transport model. *arXiv preprint arXiv:1909.07513*, 2019.

[102] A. Nilim, L. El Ghaoui, and V. Duong. Robust dynamic routing of aircraft under uncertainty. In *Digital Avionics Systems Conference*, volume 1, page 1A5–1 – 1A5–13, 2002.

[103] C. Novara, A. Nicolì, and G. C. Calafiore. Nonlinear system identification in Sobolev spaces. *preprint arXiv:1911.02930*, 2019.

[104] S. Oymak and N. Ozay. Non-asymptotic identification of LTI systems from a single trajectory. In *American Control Conference*, pages 5655–5661, 2019.

[105] F. Pasqualetti, F. Dorfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, 2013.

[106] J. A. Paulson and A. Mesbah. An efficient method for stochastic optimal control with joint chance constraints for nonlinear systems. *International Journal on Robust and Nonlinear Control*, 29(15):5017–5037, 2019.

[107] C. De Persis and P. Tesi. Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924, 2019.

[108] Maria Prandini, Simone Garatti, and John Lygeros. A randomized approach to stochastic model predictive control. In *IEEE Int. Conf. on Decision and Control*, pages 7315–7320, 2012.

[109] A. Rakhlin and K. Sridharan. Online learning with predictable sequences. *Journal of Machine Learning Research*, 2013.

[110] J. Rawlings, D. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation and Design*. Nob Hill Publishing, 2017.

[111] V. Renganathan, N. Hashemi, J. Ruths, and T. Summers. Distributionally robust tuning of anomaly detectors in Cyber-Physical systems with stealthy attacks. *arXiv preprint arXiv:1909.12506*, 2019.

[112] S. M. Robinson. Linear convergence of epsilon-subgradient descent methods for a class of convex functions. *Mathematical Programming*, 86(1):41–50, 1999.

[113] R. T. Rockafellar and R. J.-B Wets. *Variational analysis*. Springer, 1998.

[114] P.M. Esfahani S. Shafieezadeh-Abadeh, D. Kuhn. Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103):1–68, 2019.

[115] F. Santambrogio. *Optimal transport for applied mathematicians*. Springer, 2015.

[116] T. Sarkar and A. Rakhlin. Near optimal finite time identification of arbitrary linear dynamical systems. In *Int. Conf. on Machine Learning*, pages 5610–5618, 2019.

[117] Georg Schildbach, Lorenzo Fagiano, Christoph Frei, and Manfred Morari. The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica*, 50(12):3009–3018, 2014.

[118] Martin A Sehr and Robert R Bitmead. Particle model predictive control: Tractable stochastic nonlinear output-feedback mpc. *IFAC Papers Online*, 50(1):15361–15366, 2017.

[119] A. Shapiro. *On duality theory of conic linear problems*, page 135–165. Springer, 2001.

[120] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*, volume 16. SIAM, Philadelphia, PA, 2014.

[121] F. Soriguera, I. Martínez, M. Sala, and M. Menéndez. Effects of low speed limits on freeway traffic flow. *Transportation Research Part C: Emerging Technologies*, 77:257–274, 2017.

[122] J. Sturm. Using Sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625–653, 1999.

[123] M. Patriksson T. Larsson and A. Strömberg. On the convergence of conditional ε-subgradient methods for convex programs and convex–concave saddle-point problems. *European Journal of Operational Research*, 151(3):461–473, 2003.

[124] A. R. Teel, J. P. Hespanha, and A. Subbaraman. Equivalent characterizations of input-to-state stability for stochastic discrete-time systems. *IEEE Transactions on Automatic Control*, 59(2):516–522, 2014.

[125] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. The MIT Press, 2005.

[126] A. Tsiamis and G. J. Pappas. Finite-sample analysis of stochastic system identification. In *IEEE Int. Conf. on Decision and Control*, pages 3648–3654, 2019.

[127] M. Van den Berg, A. Hegyi, B. D. Schutter, and J. Hellendoorn. A macroscopic traffic flow model for integrated control of freeway and urban traffic networks. In *IEEE Int. Conf. on Decision and Control*, volume 3, pages 2774–2779, 2003.

[128] M. Verhaegen and V. Verdult. *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.

[129] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

[130] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2009.

[131] J. C. Willems, P. Rapisarda, I. Markovsky, and B. De Moor. A note on persistency of excitation. *Systems and Control Letters*, 54(4):325–329, 2005.

[132] Philip Wolfe. Convergence theory in nonlinear programming. In J. Abadie, editor, *Integer and Nonlinear Programming*, page 1–36. North-Holland, Amsterdam, 1970.

[133] D.B. Work, S. Blandin, O.P. Tossavainen, B. Piccoli, and A.M. Bayen. A traffic model for velocity data assimilation. *Applied Mathematics Research eXpress (AMRX)*, 2010(1):1–35, 2010.

[134] C. Wu, A.M. Bayen, and A. Mehta. Stabilizing traffic with autonomous vehicles. In *IEEE Int. Conf. on Robotics and Automation*, pages 1–7, 2018.

[135] J. Yan and R. R. Bitmead. Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica*, 41(4):595–604, 2005.

[136] I. Yang. Wasserstein distributionally robust stochastic control: A data-driven approach. *IEEE Transactions on Automatic Control*, pages 1–8, 2020.

[137] A. Y. Yazıcıoğlu, M. Roozbehani, and M. A. Dahleh. Resilient operation of transportation networks via variable speed limits. In *American Control Conference*, page 5623–5628, 2017.

[138] I. Yperman. *The link transmission model for dynamic network loading*. PhD thesis, Katholieke Universiteit Leuven, 2007.

[139] H. Yu, S. Koga, T. R. Oliveira, and M. Krstic. Extremum seeking for traffic congestion control with a downstream bottleneck. *arXiv preprint arXiv:1904.04315*, 2019.

[140] M. Zhu and S. Martínez. On the performance analysis of resilient networked control systems under replay attacks. *IEEE Transactions on Automatic Control*, 59(3):804–808, 2014.

[141] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Int. Conf. on Machine Learning*, pages 928–936, 2003.