ONLINE DECISION MAKING FOR DYNAMICAL SYSTEMS: MODEL-BASED AND DATA-DRIVEN APPROACHES

by
Tianqi Zheng

A dissertation submitted to Johns Hopkins University in conformity
with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
September, 2023

Abstract

The widespread availability of data sources and their increased speed compared to the past decade have created both new opportunities and challenges for developing decision-making algorithms for data streams. The ability to process data streams and make real-time decisions that align with system dynamics is a crucial aspect in the development of online decision-making algorithms. This thesis leverages tools from control theory, optimization, and learning to address the problem of online decision-making for dynamical systems, considering streaming data and dynamically changing information.

Two online decision-making frameworks are presented in this thesis, depending on the availability of system dynamic information. In the first scenario, where the system can be represented by ordinary differential equations using a state-space model, a time-varying convex optimization framework is introduced. This framework combines motion planning and control to design control signals that lead the dynamical system to asymptotically track optimal trajectories implicitly defined through constrained time-varying optimization problems. Consequently, the nonlinear dynamical system is effectively transformed into an optimization algorithm that seeks the optimal solution to the optimization problem. Global asymptotic convergence of the optimization dynamics to the minimizer of the time-varying optimization problem is proven under sufficient regularity assumptions.

In the second scenario, when system dynamics are not available, a data-driven approach called constrained reinforcement learning is adopted. Constrained reinforcement learning deals with sequential decision-making problems where an agent aims to maximize its expected total reward while interacting with an unknown environment and receiving sequentially available information over time. The constrained reinforcement learning framework further includes safety constraints or conflicting requirements during the learning process through secondary expected cumulative rewards. To address the limitations of the learning process in constrained reinforcement learning problems, a novel first-order stochastic gradient descent-ascent (GDA) algorithm is proposed: the stochastic dissipative GDA algorithm. This algorithm almost surely converges to the optimal occupancy measure and optimal policy, overcoming the issue of policy oscillation and convergence to suboptimal policies often encountered in C-RL problems.

Thesis Committee

Dr. Enrique Mallada (Primary Advisor)
Associate Professor
Department of Electrical and Computer Engineering
Johns Hopkins University

Dr. Pablo Iglesias
Edward J. Schaefer Professor
Department of Electrical and Computer Engineering
Johns Hopkins University

Dr. Mahyar Fazlyab
Assistant Professor
Department of Electrical and Computer Engineering
Johns Hopkins University

Dr. Nicolas Loizou
Assistant Professor
Department of Applied Mathematics and Statistics
Johns Hopkins University

Dedicated to My Family

Acknowledgements

First and foremost, my deepest gratitude is to my advisor, Enrique Mallada. I consider myself incredibly fortunate to have your guidance and advice. Your role as both an academic mentor and a supportive friend in my personal life has been invaluable. Despite spending 16 years in school prior to entering the Ph.D. program, it is only now, under your tutelage, that I truly comprehend effective learning. Your words of wisdom, particularly emphasizing the importance of patience and persistence, remain etched in my memory. I am profoundly grateful for the time you invest in your distinctive advisory approach—your commitment to observing my whiteboard presentations equation by equation has been instrumental. This process not only deepens my understanding of the subject matter but also enhances my presentation and teaching abilities.

Moreover, your guidance has been even more exceptional and challenging since 2020 due to the global pandemic. Your unwavering support in navigating uncertainties, both professionally and personally, has been a cornerstone of my journey. I am genuinely appreciative of your efforts in fostering a unique and welcoming atmosphere within the NetD Lab. This environment has turned my academic pursuit into a warm and delightful experience. Thank you for all that you do.

I extend my deepest gratitude to my former advisor, Ming Guo, and the other members of the Guo lab at MIT during my undergraduate years. Meeting this group of diligent, intelligent, passionate, kind, and humble individuals has been a stroke of great fortune. Your role in introducing me to the path of my Ph.D. career and supporting my application is something I will always hold in high regard. The challenges and enjoyment I experienced while working there will forever be etched in my memory.

I would also like to express my appreciation to my committee members and fellow members of the Graduate Board Oral for their unwavering support. First and foremost, my heartfelt thanks go to Pablo Iglesias. Your contribution of exceptional courses, consistent presence as a committee member for my major milestones, and guidance in your role as department chair are immensely valued. Mahyar Fazlyab, I am grateful for your role on my dissertation committee and your motivating influence on many of my significant projects. To Nicolas Loizou, I offer my thanks for your participation on my thesis committee and your invaluable comments and suggestions during the latter part of my research. I can only wish to have had the opportunity to engage with your insights earlier. Your advice has consistently proven to be precise and invaluable. Additionally, I am appreciative of Marin Kobilarov, Donniell Fishkind, and Amitabh Basu for their roles on my graduate board oral exam committee. Your presence and input have been instrumental on my academic journey. I would like to also thank my collaborator, John Simpson-Porco.

I want to further extend my appreciation to my colleagues: Pengcheng You, Yan Jiang, Chengda Ji, Yue Shen, Hancheng Min, Rajni Kant Bansal, Eli Pivo, Mustafa Devrim Kaba, Haralampos Avraam, Gary Gao, Dhananjay Anand, Eliza Cohn, Jay Guthrie, and Roy Siegelmann from NetD Lab;

I would like to end with my special thank to my family: my parents Xukun Zheng and Xiaonan Ma, my grandparents Xinliang Ma and Fengyun Xi, my aunt's family Jianing Ma, Yu Sui, and Xin Sui. I deeply appreciate the nurturing childhood you provided, fostering an environment that encouraged my curiosity and exploration. Above all, I am truly grateful for granting me the opportunity to receive an education that fundamentally transformed my life. If I have seen further and accomplished anything, it is by standing on the shoulders of yours. Last but not least, I would like to give my greatest thanks to my girlfriend, Dr. Xinyang Liu. I'm incredibly pleased and honored to conclude this remarkable journey alongside you. The knowledge and wisdom I've gained from our shared experiences, both professionally and personally, are immeasurable. I extend my heartfelt gratitude for your unwavering love and support throughout the past decade. I can't wait to start a new chapter of my life in your company.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Online decision-making is a fundamental and rapidly evolving area of study that addresses the challenges of making effective decisions in dynamic and time-sensitive environments. In contrast to traditional decision-making scenarios, where all information is available upfront, online decision-making deals with situations where data arrives sequentially and continuously over time. This dynamic nature of data streams presents unique challenges, requiring decision-makers to adapt and respond promptly to changing conditions without the luxury of hindsight.

The ubiquity of data sources and the increasing speed of data generation in today's interconnected world have further emphasized the importance of developing robust and efficient online decision-making algorithms. Industries such as finance, healthcare, transportation, and manufacturing are increasingly reliant on real-time data to optimize processes, enhance customer experiences, and improve overall efficiency. Whether it's optimizing resource allocation, managing traffic flow, controlling autonomous vehicles, or personalizing recommendations for online shoppers, the ability to make timely and informed decisions has become a critical competitive advantage.

In this context, this thesis explores the concept of online decision-making, delving into the theoretical foundations, algorithms, and practical applications for dynamical systems. We investigate two distinct online decision-making frameworks, each catering to different scenarios based on the availability of system dynamic information. Through the exploration of case studies and empirical analyses, we aim to shed light on the opportunities and challenges associated with online decision-making, contributing to the advancement of this vital and ever-evolving field.

In the first scenario, when the system's behavior can be adequately represented by ordinary differential equations using a state-space model, a time-varying convex optimization framework is introduced. This framework ingeniously integrates motion planning and control techniques to design control signals that enable the dynamical system to asymptotically track optimal trajectories, implicitly defined through constrained time-varying optimization problems.

In the second scenario, where the system dynamics are not explicitly available, the thesis adopts a data-driven approach known as constrained reinforcement learning. This methodology revolves around sequential decision-making tasks, where an agent strives to maximize its expected cumulative reward while interacting with an unknown environment and progressively receiving new information over time. To accommodate safety constraints or conflicting requirements during the learning process, the constrained reinforcement learning framework incorporates secondary expected cumulative rewards.

1.1 Model-Based Approach: Time-varying Optimization

Autonomy refers to the capability of a robot or machine to carry out tasks without any human intervention. In domains like robotics and transportation, autonomous tasks typically entail several steps. First, the system must gather information from the environment, often dealing with noisy data (sensing). Next, it needs to determine its own precise position in the environment (localization). Then, a safe and efficient navigation strategy must be devised (planning). Finally, the system must be able to adapt and respond to unexpected changes in the environment (control) [1, 2, 3]. These components collectively enable the system to operate independently and accomplish tasks without relying on continuous human guidance. A particularly challenging step of this process is the motion planning stage [1, 2, 3, 4], wherein an agent with uncertain information about its position and environment must devise an admissible and collision-free trajectory to be followed toward a final destination. This highly complex task has received widespread research interest [5, 6, 7, 3, 8, 4, 9, 10, 11, 2], as it requires a delicate balance between computational complexity and optimality while simultaneously respecting the agent's dynamic capabilities.

Standard approaches to solving this problem can be broadly categorized into three groups: Grid-based search (GBS), Sampling-based Planning (SBP), and Optimization based (OB). GBS algorithms assign each configuration of the dynamical system to a grid point and use graph search algorithms such as Dijkstra [5], A^* [6], and D^* [7] to find a path. Although GBS algorithms are easy to implement and often provide an acceptable answer, they scale poorly with the number of degrees of freedom of the configuration space [12] and fail to ensure the dynamic feasibility of the path. SBP algorithms [3], such as rapidly-exploring random trees (RRTs) [8], probabilistic roadmap methods (PRMs) [4], and their variants scale better for high-dimensional problems. However, optimality guarantees are usually absent, and path feasibility is only achieved via sufficiently dense sampling of either the configuration or action space [12]. On the other hand, OB algorithms such as direct multiple-shooting [13] and direct collocation [14] explicitly consider the dynamic constraints in the optimization problems, providing by construction dynamically feasible trajectories which can be enforced to avoid collisions [15, 16]. However, OB algorithms suffer from high computational costs, typically requiring solving a nonlinear programming problem without convergence guarantees [11].

Two common features of the above-mentioned solutions are (a) the struggle between the computational complexity of the planning process and the need to enforce dynamic constraints and (b) the open-loop nature of the solution that does not account for unmodeled dynamics or disturbances. Thus, such methodologies are commonly complemented with a motion execution stage that implements a feedback controller that tracks the open-loop trajectory. However, such an approach requires some level of conservativeness in the planning stage to avoid collisions [17, 18], further increasing the computational complexity of the solution. This work aims to explore an alternative approach aiming at breaking the decoupling between planning and control while ensuring dynamic feasibility and accounting for changing conditions in real time.

1.1.1 Prior work

Our work also broadly aligns with and contributes to the growing literature of *online optimization with feedback loops*, *network systems*, and *algorithm design for time-varying optimization*.

Online optimization with feedback loops seeks to design online optimization algorithms to regulate the output of a dynamical system towards the optimal solution of an optimization problem. For the case of LTI systems, numerous works have design controllers that track the optimal solution of (i) a static optimization problem [19, 20, 21], and (ii) time-varying convex optimization problems [22], including also input-output constraints [23]. Nonlinear system dynamics are considered for steering a physical system to a steady state that solves a predefined constrained static optimization problem [24] and unconstrained time-varying optimization problem [25].

Online optimization of network systems considers the extension of the above framework for problems where systems and computations are distributed. The papers [26, 27] seek to design controllers to regulate the network of agents to the global minimizer of a predefined convex optimization problem. Time-varying versions of this problem are have been considered, including versions with inequality constraints [28], with double-integrator dynamics [29], and with nonlinear dynamics in a strict feedback form [30].

Time-varying optimization has been a popular subject of research for online decision-making. It provides a computationally frugal optimization framework that produces solutions in "a timely fashion and is essential when input data streams are of large-scale and decisions must be made at high frequency." [29, 31, 32, 33, 34]. Our work is a direct application of time-varying optimization formalisms in the area of feedback control and motion planning [31]. Online solvers for time-varying optimization problems have been proposed both in continuous time [29, 32] and in discrete time [33, 34].

1.1.2 Thesis contribution

This work uses time-varying optimization to combine safe motion planning and control in a unique *closed-loop task*. We seek to encode planning goals and safety constraints as a time-varying (TV) constrained optimization problem and develop a general methodology to design closed-loop feedback controllers by drawing insights from mathematical optimization. Our methodology combines tools from differential flatness and optimization theory to develop controllers which effectively transform a dynamical system into an optimization algorithm that seeks to track the optimal solution of the aforementioned optimization problem.

The contributions of our work are as follows:

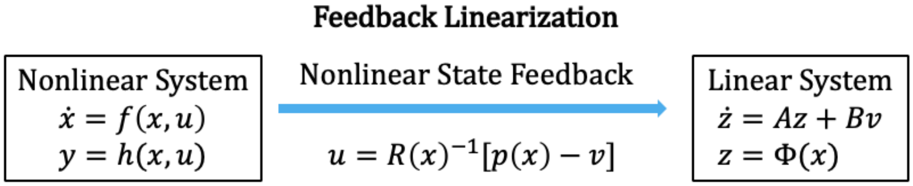

Figure 1-1. Feedback linearization transforms a nonlinear system into a linear system via nonlinear state feedback control and coordinate transformation.

- *Planning and Control as TV Optimization.* We formulate a framework to encode planning and control goals within a time-varying optimization problem, wherein planning goals are implicitly encoded as the (apriori unknown) optimal solution $y^*(t)$ of a TV-Optimization problem. This formulation allows us, in turn, to recast the control design problem as the problem of choosing an optimization algorithm.

- *Flat Systems as Optimization Algorithms.* We provide a general methodology to design control laws that steer the output $y(t)$ of a differentially flat nonlinear system towards $y^*(t)$. Inspired by feedback linearization [35, 36] (Fig. 1-1), the proposed methodology transforms any flat system of order k into a time-varying optimization algorithm that depends on the first $k-1$ time derivatives of the objective function's gradient (Fig. 1-2).

- *Theoretical Guarantees.* Our control design framework can readily provide rigorous theoretical guarantees on the asymptotic behavior of the system. Precisely, we show that under mild conditions, the output $y(t)$ of a differentially flat nonlinear system converges asymptotically to $y^*(t)$.

- *Extensions for Formation and Collision Avoidance.* We further extend our framework to allow for formation and collision avoidance specifications. We extend our time-varying feedback optimization framework to allow the asymptotic satisfaction of time-varying equality constraints (that allow for the specification of formation constraints) and inequality constraints (that can enforce

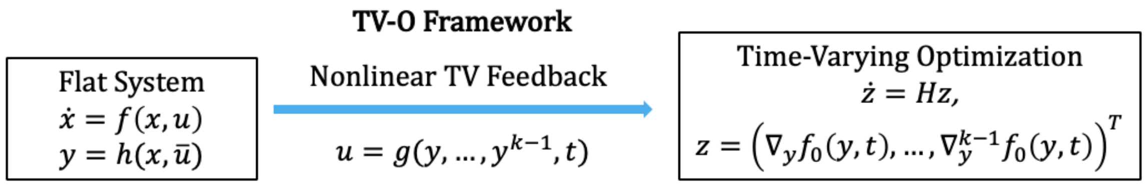
TV-O Framework
Flat System
$\dot{x} = f(x,u)$
$y = h(x,\bar{u})$
Nonlinear TV Feedback
$u = g(y,\dots,y^{k-1},t)$
Time-Varying Optimization
$\dot{z} = Hz,$
$z = \left(\nabla_y f_0(y,t),\dots,\nabla_y^{k-1} f_0(y,t)\right)^T$

Figure 1-2. Time Varying Optimization framework effectively transforms a differentially flat system into an optimization algorithm.

collision avoidance).

1.2 Data-Driven approach: Constrained Reinforcement Learning

Reinforcement learning (RL) is concerned with tackling sequential decision-making problems, where an agent seeks to maximize its expected total reward while interacting with an unknown environment over time. Nevertheless, certain real-world applications, such as electric grids and robotics, pose unique challenges. In these scenarios, the agent often encounters conflicting requirements [37] or must adhere to safety constraints during the learning process [38]. To address these complexities effectively, the constrained reinforcement learning (C-RL) framework emerges as a natural and efficient approach. C-RL allows the seamless integration of conflicting requirements and the incorporation of safety considerations, enabling the agent to navigate through such intricate environments with improved effectiveness and robustness [38, 39, 40, 41, 42, 43, 44].

In tackling constrained reinforcement learning (C-RL) problems and finding the optimal policy, there are two major approaches. The first approach involves solving the problem in the occupancy measure space using the constrained Markov Decision Process (CMDP) framework, a well-established formulation for reinforcement learning with constraints [39]. In this approach, the agent seeks to maximize the total reward function while adhering to secondary cumulative reward constraints.

The CMDP problem is transformed into an equivalent linear programming problem in the occupancy measure space, and the optimal policy is derived from the optimal occupancy measure [39]. However, this approach requires explicit knowledge of the transition kernel of the underlying dynamical system, which may not always be available in realistic applications.

An alternative approach is to tackle the C-RL problem in policy space, employing the principles of Lagrangian duality [42, 43, 44, 45, 46]. These approaches utilize sampling-based primal-dual algorithms or stochastic gradient descent-ascent (SGDA) algorithms, augmenting the Lagrangian function with possible regularization terms, like KL divergence regularization. The primal and dual variables are iteratively updated using gradient information or by solving sub-optimization problems. The outcome of these algorithms can be characterized into two cases: in the first case, the output is a mixing policy, which is a weighted average of historical outputs [42, 43, 44]. In the second case, rather than showing the output policy converges to the optimal policy, these approaches present a regret analysis for objective functions and constraints [45, 46]. A key limitation in these approaches is that the policy often oscillates and fails to converge to the optimal policy, resulting in a mismatch between the behavioral policy and the optimal one.

In this thesis, we aim to address the aforementioned limitations by introducing a novel SGDA algorithm that leverages recent results on regularized saddle flow dynamics. By leveraging the insights from regularized saddle flow dynamics, we seek to enhance the performance and reliability of decision-making algorithms in constrained reinforcement learning scenarios. The critical observation made about the sampling-based primal-dual algorithms discussed earlier is that the Lagrangian function used in the constrained reinforcement learning (C-RL) problem lacks sufficient convexity. Specifically, in occupancy measure space, the Lagrangian function is bilinear, and in policy space, it becomes non-convex-concave. As a consequence, these algorithms fail to converge reliably.

One commonly employed approach involves taking averaged iterates, which combines previous outputs with certain weights. However, theoretical guarantees for the averaged iterates are limited, especially when dealing with objective functions that are not convex-concave [47, 48]. Additionally, in the context of Reinforcement Learning (RL) and Constrained Reinforcement Learning (C-RL), relying on averaged results can be undesirable. This is because the mixture of past policies may obscure oscillating or overshooting objective/constraint functions, hindering the attainment of an optimal policy iterate [49]. Therefore, it becomes crucial to explore training algorithms that ensure the final iteration of the training process approaches the equilibrium point directly, a concept known as last-iterate convergence, rather than merely relying on an average outcome. To this end, the Extra-gradient (EG) method [50], the Optimistic gradient (OG) method [51], and their variants have gained significant attention in recent literature. These algorithms are particularly appealing due to their superior empirical performance and last-iterate convergence guarantees, especially in the convex-concave setting.

1.2.1 Prior work

Constrained Reinforcement Learning

Conventional Constrained Reinforcement Learning problems can be formulated as constrained min-max optimization problems, either in the *Policy space* [38, 52] or the *Occupancy measure space*, known as Constrained Markov Decision Process (CMDP) [39]. Since the min-max optimization problem is nonconvex in Policy space and bilinear in Occupancy measure space, vanilla GDA algorithms either fail to converge or only provide average-iterate convergence [43, 44]. Global last-iterate asymptotic convergence results in terms of occupancy measure iterates have been established by previous works [53, 54] through Saddle flow dynamics and

Optimistic Mirror Descent (OMD), respectively. [49] extends this by providing a non-asymptotic last-iterate convergence result for an infinite-horizon discounted CMDP in terms of occupancy measure and policy iterates.

Min-Max Optimization, Variational Inequalities, and Zero-Sum Games

The Extra-gradient (EG) method, the Optimistic gradient (OG) method, and their variants have been extensively studied in the context of min-max optimization problems, zero-sum games, and variational inequality problems (VIPs). In the variational inequality perspective, [55] proves linear convergence rates for the EG method in the bilinear and strongly monotone cases. More recently, in the context of machine learning, specifically Generative Adversarial Networks (GANs), several papers have explored the convergence rates of algorithms for solving saddle point problems. [56] analyzes gradient-based saddle point dynamics, including EG and OG, and shows linear convergence when the objective function is bilinear. [57] interprets GANs within the VIP framework and studies OGDA as an extrapolation from the past variant, proving a linear convergence rate for strongly monotone VIPs. [58] provides a unified convergence analysis of EG and OGDA methods as approximations of the proximal point method, deriving standard linear rates for both bilinear and strongly monotone settings ($\mu/4L$). [59] presents a unified analysis of EG and OG for both strongly monotone and bilinear games, obtaining a tighter global convergence rate through spectral analysis of the operators.

EG and OG methods are particularly well-suited for solving saddle-point problems due to their last-iterate convergence, as opposed to only average-iterate convergence to min-max solutions. [60] shows that the primal-dual gap of the averaged iterates generated by both EG and OG algorithms converges at a rate of O(1/k). [47] provides a $\mathcal{O}(1/N)$ convergence rate guarantee for the last iterate of EG, in terms of the squared norm of the operator (Hamiltonian), for monotone and Lipschitz operators with a Lipschitz Jacobian. [61] derives the same last-iterate convergence rate in terms of Hamiltonian for EG, relaxing the additional Lipschitz Jacobian assumption. In [62], they show that this $\mathcal{O}(1/N)$ last-iterate convergence can also be achieved for OG methods without any assumption on the Jacobian of the operator. [63] establishes the linear last-iterate convergence of OGDA in a constrained setting. Additionally, [64] addresses the issue of limit cycling behavior in training GANs and shows that OG exhibits final-iterate convergence to a neighborhood of the solution for bilinear games, such as WGANs.

1.2.2 Thesis contribution

To address the limitation in solving C-RL problems, our proposed method draws inspiration from the study of saddle flow dynamics. By incorporating a carefully designed augmented regularization, we introduce a dissipative saddle flow, which sets minimal requirements on convexity-concavity while ensuring asymptotic convergence to a saddle point. Building upon the tools from this dissipative saddle flow framework, we present a novel algorithm to tackle the C-RL problem in occupancy measure space. The dynamics of this algorithm converge asymptotically to the optimal occupancy measure and optimal policy. Furthermore, we extend this continuous-time algorithm to a model-free setting, where the discretized stochastic Dissipative Gradient Descent-Ascent (DGDA) emerges as the stochastic approximation of the continuous-time saddle flow dynamics. Our research establishes that the SGDA algorithm almost surely converges to the optimal solution of the C-RL problem. Notably, this work represents the first attempt to solve the C-RL problem with a guarantee of convergence to the optimal occupancy measure and policy.

Besides, we prove the proposed DGDA algorithm exhibits linear last-iterate convergence for strongly monotone (resp. bilinear) and Lipschitz VIP without any additional assumptions. Moreover, we showed that when the problem is bilinear, the proposed algorithm provides a better linear convergence rate, in terms of constant, compared with standard rates for the bilinear problem for EG and OGDA. When the problem is strongly convex-strongly concave and condition number $\kappa \geq 2$, the proposed algorithm provides a better convergence rate compared with standard rates for the bilinear problem for EG and OGDA. DGDA's effectiveness in solving bilinear and strongly convex-strongly concave problems is demonstrated through the presentation of two numerical examples. In both cases, DGDA consistently outperforms EG and OG methods.

Chapter 2

Model-Based Online Decision Making: A Time-Varying Optimization Framework

In this chapter, we introduce the model-based online decision-making framework motivated by time-varying optimization. The proposed framework encodes planning goals and safety constraints as a time-varying (TV) constrained optimization problem. In doing so, the proposed time-varying optimization framework combines safe motion planning and control in a unique closed-loop task. We combine tools from feedback linearization, differential flatness, and optimization theory, which effectively transform a dynamical system into an optimization algorithm that seeks to track the optimal solution of the optimization problem.

Specifically, consider the general nonlinear dynamical system, with the state $\mathbf{x} \in \mathbb{R}^n$, input $\mathbf{u} \in \mathbb{R}^m$ and output $\mathbf{y} \in \mathbb{R}^m$, described in state-space form:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \quad \mathbf{y} = h(\mathbf{x}, \mathbf{u}). \tag{2.1}$$

Let $t \geq 0$ be a continuous time index, and $f_0 : \mathbb{R}^m \times \mathbb{R}_+ \to \mathbb{R}$ be a time-varying objective function of flat output $\mathbf{y}$. The functions $f_i : \mathbb{R}^m \times \mathbb{R}_+ \to \mathbb{R}, \; i \in [p]$ are the time-varying inequality constraint functions. We also define the time-varying equality constraint functions $h_j : \mathbb{R}^m \times \mathbb{R}_+ \to \mathbb{R}$ taking values $h_j(\mathbf{y}, t) =$

$a_j(t)^T\mathbf{y} - b_j(t),\ j \in [q]$, where $\mathbf{A}(t) : R_+ \to \mathbb{R}^{q\times m}, q < m$ is defined as $\mathbf{A}(t) = [a_1(t)$ $, \ldots, a_q(t)]^T$, $b(t) : R_+ \to \mathbb{R}^q$ is defined as $b(t) = [b_1(t), \ldots, b_q(t)]^T$.

Consider a constrained time-varying optimization problem that has the following form:

$$\begin{aligned} \mathbf{y}^*(t) :=& \arg\min_{\mathbf{y}\in\mathbb{R}^m}\ f_0(\mathbf{y}, t) \\ s.t. \quad & f_i(\mathbf{y}, t) \leq 0, \quad i \in [p] \\ & \mathbf{A}(t)\mathbf{y} = b(t). \end{aligned} \tag{2.2}$$

where $f_0, \ldots, f_p, h_1, \ldots, h_q : \mathbb{R}^m \times \mathbb{R}_+ \to \mathbb{R}$ are assumed to be infinitely differentiable ($\mathbb{C}^\infty$) with respect to both $\mathbf{y}$ and t. Additionally, we assume the functions satisfy $\emptyset \neq \mathbf{dom} f_0 \subset (\cap_{i=1}^p \mathbf{dom} f_i) \cap (\cap_{j=1}^q \mathbf{dom} h_j)$ for all $t \geq 0$, i.e., the time-varying optimization problem is always feasible. The goal is to generate a control input $\mathbf{u}(t)$ such that $\|\mathbf{y}(t) - \mathbf{y}^*(t)\| \to 0$ as $t \to \infty$ for all initial conditions, i.e., global asymptotic convergence.

Various motion planning and control tasks can be encoded as instances of the time-varying optimization problem (2.2). For example, if the positions of a controlled robot and a moving target are denoted by $\mathbf{y}(t)$ and $\mathbf{y}^d(t)$ respectively, then minimizing the objective function $\|\mathbf{y}(t) - \mathbf{y}^d(t)\|$ represents the task of tracking a moving target. Along the same line, if $\mathbf{y}(t)$ denotes the vector of positions of a network of agents in 2 dimensions, represented by complex values $y_i \in \mathbb{C}$, one can impose formation constraints using a constraint of the form $\mathbf{L}\mathbf{y}(t) = 0$, where $\mathbf{L}(\mathbf{t})$ is the complex-valued Laplacian matrix associated with a desired formation [65]. Similarly, to ensure collision avoidance, a set of inequality constraints can be employed: $\{\mathbf{a}_i(\mathbf{y})^T\mathbf{z} - b_i(\mathbf{y}) \leq 0, i \in [m]\}$ [32], which is elaborated further in Section 2.2.6.

Chapter outline

This Chapter is organized as follows: In Section 2.1, we study the problem of regulating a *feedback linearizable system* to trajectories implicitly defined via *unconstrained* time-varying optimization problem. For feedback linearizable systems that have (non)uniform vector relative degrees, we propose a control law that will globally asymptotically converge to the optimal solution of the unconstrained time-varying optimization problem. In Section 2.2, we further extend the result, including extensions from feedback linearization to general *differentially flat systems*, as well as the inclusion of *equality and inequalities constraints*. We formally introduce the time-varying optimization framework, which generalizes the notion of feedback linearization and transforms a flat system into an optimization algorithm. Designing the nonlinear feedback controller is reduced to finding a solution to the ODE system that satisfies the optimization dynamic and system dynamic simultaneously.

Notation

Given an n-tuple $(x_1, ..., x_n)$, $\mathbf{x} \in \mathbb{R}^n$ is the associated column vector. The $n \times n$ identity matrix is denoted as $\mathbf{I}_n$. For a square symmetric matrix $\mathbf{A}$, is positive (semi-)definite, and write $\mathbf{A} \succ \mathbf{0}$ ($\mathbf{A} \succeq \mathbf{0}$), if and only if all the eigenvalues of $\mathbf{A}$ are positive (nonnegative). We further write $\mathbf{A} \succ \mathbf{B}$ ($\mathbf{A} \succeq \mathbf{B}$) whenever $\mathbf{A} - \mathbf{B} \succ \mathbf{0}$ ($\mathbf{A} - \mathbf{B} \succeq \mathbf{0}$). The Euclidean norm of a vector $\mathbf{x}$ is denoted by $\|\mathbf{x}\|_2$, and the spectral norm of a matrix $\mathbf{A}$ by $\|\mathbf{A}\|_2$. $\otimes$ denotes the Kronecker product between two matrices. We use the short-hand notation $\bar{\mathbf{x}}^{(k)} = (\mathbf{x}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)})$, where k is some finite but arbitrary integer.

Given a continuously differentiable function $f(\mathbf{x}, t)$ of state $\mathbf{x} \in \mathbb{R}^n$ and time $t \in$, the gradient with respect to $\mathbf{x}$ (resp. t) is denoted by $\nabla_{\mathbf{x}} f(\mathbf{x}, t)$ (resp. $\nabla_t f(\mathbf{x}, t)$). The total derivative of $\nabla_{\mathbf{x}} f(\mathbf{x}(t), t)$ with respect to t is denoted by $\dot{\nabla}_{\mathbf{x}} f(\mathbf{x}, t) :=$

$\frac{d}{dt}\nabla_{\mathbf{x}} f(\mathbf{x}(t), t)$, and the n-th total derivative with respect to t by $\nabla_{\mathbf{x}}^{(n)} f(\mathbf{x}, t)$. The partial derivatives of $\nabla_{\mathbf{x}} f(\mathbf{x}, t)$ with respect to $\mathbf{x}$ and t are denoted by $\nabla_{\mathbf{xx}} f(\mathbf{x}, t) := \frac{\partial}{\partial \mathbf{x}} \nabla_{\mathbf{x}} f(\mathbf{x}, t) \in \mathbb{R}^{n \times n}$ and $\nabla_{\mathbf{x}t} f(\mathbf{x}, t) := \frac{\partial}{\partial t} \nabla_{\mathbf{x}} f(\mathbf{x}, t) \in \mathbb{R}^n$, respectively. The derivative $L_f h$ of a function $h : \mathbb{R}^n \to \mathbb{R}$ along the vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ is given by $(L_f h)(\mathbf{x}) = \nabla h(\mathbf{x})^T f(\mathbf{x})$. Taking the derivative of h first along a vector field f and then along a vector field g is given by $(L_g L_f h)(\mathbf{x}) = \frac{\partial (L_f h)}{\partial x} g(\mathbf{x})$. If h is being differentiated k times along f, the notation $L_f^k h(\mathbf{x}) = \frac{\partial (L_f^{k-1} h)}{\partial x} g(\mathbf{x})$ is used.

2.1 Feedback linearizable systems

This section presents a novel optimization-based framework for joint real-time trajectory planning and feedback control of feedback-linearizable systems. To achieve this goal, we define a target trajectory as the optimal solution to a time-varying optimization problem. In general, however, such a trajectory may not be feasible due to, e.g., nonholonomic constraints. To solve this problem, we design a control law that generates feasible trajectories that asymptotically converge to the target trajectory. More precisely, for systems that are (dynamic) full-state linearizable, the proposed control law implicitly transforms the nonlinear system into an optimization algorithm of a sufficiently high order. We prove global asymptotic convergence to the target trajectory for both the optimization algorithm and the original system.

Section 2.1.1 introduces some preliminary definitions, including feedback linearization, which means a system can be transformed into a linear system by a state diffeomorphism, its dynamic feedback extension, and elementary analysis of Hurwitz linear systems. We formally state the problem and present two motivating examples with different system dynamics (integrator and wheeled mobile robot). In Section 2.1.2 and 2.1.3, for feedback linearizable systems with uniform/nonuniform vector relative degrees, we design a control law which (i) implicitly defines a target trajectory as the optimal solution of a time-varying optimization problem, and (ii) asymptotically drives the system to the target trajectory. We illustrate the effectiveness of our approach using two examples in Section 2.1.4, one where a wheeled mobile robot switches from tracking one moving object to another, and another where multiple agents must track multiple objects with internal distance constraints.

2.1.1 Preliminaries

Static Feedback Linearization

We consider a square control-affine nonlinear system with the state $\mathbf{x} \in D \subset \mathbb{R}^n$, m inputs $\mathbf{u} \in \mathbb{R}^m$ and m outputs $\mathbf{y} \in \mathbb{R}^m$, described in state-space form:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \ , \tag{2.3a}$$

$$\mathbf{y} = h(\mathbf{x}) \ , \tag{2.3b}$$

where $f : D \to \mathbb{R}^n$, $\mathbf{G} : D \to \mathbb{R}^{n\times m}$, and $h : D \to \mathbb{R}^m$ are sufficiently smooth on a domain $D \subset \mathbb{R}^n$, with $\mathbf{G}$ and h expanded as

$$\mathbf{G}(\mathbf{x}) = \left[g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})\right] \in \mathbb{R}^{n\times m},$$

$$h(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_m(\mathbf{x})) \in \mathbb{R}^m.$$

Problem 1 (State-Space Exact Linearization)**.** *Given a point $\mathbf{x_0} \in D \subset \mathbb{R}^n$, for the control-affine nonlinear system (2.3), find a feedback controller $\mathbf{u} = \alpha(\mathbf{x}) + \beta(\mathbf{x})\mathbf{v}$ defined on a neighborhood U of $\mathbf{x_0}$, a coordinate transformation $\mathbf{z} = \Phi(\mathbf{x})$ also defined on U, and a controllable pair $(\mathbf{A}, \mathbf{B})$ $(\mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{B} \in \mathbb{R}^{n\times m})$ such that:*

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{v} = \frac{\partial\Phi(\mathbf{x})}{\partial\mathbf{x}}\Big(f(\mathbf{x}) + g(\mathbf{x})(\alpha(\mathbf{x}) + \beta(\mathbf{x})\mathbf{v})\Big).$$

The key condition on (2.3) for the solvability of the State-Space Exact Linearization Problem is that the system possesses vector relative degree [36]. In other references [66], this is also called *Full State Linearization*.

Definition 1 (Vector Relative Degree [36]). *The control affine system* (2.3) *is said to have* vector relative degree $\{r_1, r_2, \ldots, r_m\}$ *at a point* $\mathbf{x_0} \in D \subset \mathbb{R}^n$ *if:*

(i) $L_{g_j} L_f^k h_i(\mathbf{x}) = 0$ *for all* $1 \leq i \leq m$, *for all* $k < r_i - 1$, *for all* $1 \leq j \leq m$, *and for all* $\mathbf{x}$ *in a neighborhood of* $\mathbf{x_0}$, *and*

(ii) the $m \times m$ *matrix,*

$$\mathbf{R}(\mathbf{x}) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(\mathbf{x}) & \ldots & L_{g_m} L_f^{r_1-1} h_1(\mathbf{x}) \\ L_{g_1} L_f^{r_2-1} h_2(\mathbf{x}) & \ldots & L_{g_m} L_f^{r_2-1} h_2(\mathbf{x}) \\ \vdots & \ldots & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(\mathbf{x}) & \ldots & L_{g_m} L_f^{r_m-1} h_m(\mathbf{x}) \end{bmatrix}, \tag{2.4}$$

is nonsingular at $\mathbf{x} = \mathbf{x_0}$.

Lemma 1 (Solution of Exact Linearization Problem with static feedback linearization [36, Lemma 5.2.1]). *Suppose the matrix* $\mathbf{G}(\mathbf{x_0})$ *has rank* m. *Then the State-Space Exact Linearization Problem is solvable if and only if there exists a neighborhood of* $\mathbf{x_0}$ *such that the system* (2.3) *has vector relative degree* $\{r_1, r_2, \ldots, r_m\}$ *at* $\mathbf{x_0}$ *and* $r_1 + r_2 + \cdots + r_m = n$. *In particular, one may choose*

(i) the feedback as

$$\mathbf{u} = -\mathbf{R}(\mathbf{x})^{-1}\mathbf{p}(\mathbf{x}) + \mathbf{R}(\mathbf{x})^{-1}\mathbf{v},$$

where $\mathbf{p}(\mathbf{x}) = \mathrm{col}(L_f^{r_1} h_1(\mathbf{x}), \ldots, L_f^{r_m} h_m(\mathbf{x})) \in \mathbb{R}^m$ *and* $\mathbf{R}(\mathbf{x})$ *is defined in* (2.4),

(ii) the coordinate transformation as

$$\Phi(\mathbf{x}) = \mathrm{col}(h_1(\mathbf{x}), \ldots, L_f^{r_1-1} h_1(\mathbf{x}), \ldots, L_f^{r_m-1} h_m(\mathbf{x})),$$

(iii) $(\mathbf{A}, \mathbf{B})$ *having the* Brunovsky Canonical Form

$$\mathbf{A} = \mathrm{diag}\mathbf{A}_1, \ldots, \mathbf{A}_m,\ \mathbf{B} = \mathrm{diag}\mathbf{b}_1, \ldots, \mathbf{b}_m,$$

where $\mathbf{A}_i \in \mathbb{R}^{r_i \times r_i}$ *and* $\mathbf{b}_i \in \mathbb{R}^{r_i}$ *are*

$$\mathbf{A}_i = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ . & . & . & \dots & . \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad \mathbf{b}_i = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Remark: The input v_i *controls only the output* y_i *throughout a chain of* r_i *integrator. When* $r_1 + r_2 + \cdots + r_m = n$*, in the closed-loop system there are no unobservable dynamics.*

Dynamic Feedback Linearization

For systems that do not have vector relative degree, one can sometimes achieve a vector relative degree by introducing auxiliary state variables $\boldsymbol{\zeta}$, e.g., for a system that is differentially flat [67], by using dynamic feedback of the form

$$\mathbf{u} = \alpha(\mathbf{x}, \boldsymbol{\zeta}) + \beta(\mathbf{x}, \boldsymbol{\zeta})\mathbf{w}, \tag{2.5a}$$

$$\dot{\boldsymbol{\zeta}} = \gamma(\mathbf{x}, \boldsymbol{\zeta}) + \delta(\mathbf{x}, \boldsymbol{\zeta})\mathbf{w}. \tag{2.5b}$$

Consider then the composite system formed by (2.3) and (2.5)

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\zeta}} \end{bmatrix} = \tilde{f}(\mathbf{x}, \boldsymbol{\zeta}) + \tilde{\mathbf{G}}(\mathbf{x}, \boldsymbol{\zeta})\mathbf{w}, \;\; \mathbf{y} = h(\mathbf{x}), \tag{2.6}$$

where

$$\tilde{f}(\mathbf{x}, \boldsymbol{\zeta}) = \begin{bmatrix} f(\mathbf{x}) + \mathbf{G}(\mathbf{x})\alpha(\mathbf{x}, \boldsymbol{\zeta}) \\ \gamma(\mathbf{x}, \boldsymbol{\zeta}) \end{bmatrix}, \tilde{\mathbf{G}}(\mathbf{x}, \boldsymbol{\zeta}) = \begin{bmatrix} g(\mathbf{x})\beta(\mathbf{x}, \boldsymbol{\zeta}) \\ \delta(\mathbf{x}, \boldsymbol{\zeta}) \end{bmatrix}.$$

The following is a direct extension of Lemma 1. Further details on this approach, known as *dynamic extension*, can be found in [36] and [66].

Lemma 2 (Solution of Exact Linearization Problem using dynamic feedback linearization [36])**.** *Suppose the matrix* $\tilde{\mathbf{G}}(\mathbf{x_0}, \boldsymbol{\zeta_0})$ *has rank* m*. Then the State-Space Exact Linearization Problem is solvable if and only if there exists a neighborhood of* $[\mathbf{x_0}, \boldsymbol{\zeta_0}]^T$ *such that the system* (2.6) *has vector relative degree* $\{r_1, r_2, \dots, r_m\}$ *at* $[\mathbf{x_0}, \boldsymbol{\zeta_0}]^T$ *and* $r_1 + r_2 + \cdots + r_m = n$*. In particular, one may choose* *(i) The dynamic feedback defined by* (2.5) *and*

$$\mathbf{w} = -\mathbf{R^{-1}}(\mathbf{x}, \boldsymbol{\zeta})\mathbf{p}(\mathbf{x}, \boldsymbol{\zeta}) + \mathbf{R^{-1}}(\mathbf{x}, \boldsymbol{\zeta})\mathbf{v}, \tag{2.7}$$

where $\mathbf{p}(\mathbf{x}, \boldsymbol{\zeta}) = \mathrm{col}(L_{\tilde{f}}^{r_1} h_1(\mathbf{x}), \ldots, L_{\tilde{f}}^{r_m} h_m(\mathbf{x})) \in \mathbb{R}^m$ *and* $\mathbf{R}(\mathbf{x}, \boldsymbol{\zeta})$ *is defined in* (2.4).

(ii) the coordinate transformation as

$$\Phi(\mathbf{x}, \boldsymbol{\zeta}) = \mathrm{col}(h_1(\mathbf{x}), \ldots, L_{\tilde{f}}^{r_1 - 1} h_1(\mathbf{x}), \ldots, L_{\tilde{f}}^{r_m - 1} h_m(\mathbf{x})),$$

(iii) $(\mathbf{A}, \mathbf{B})$ *having the* Brunovsky Canonical Form*.*

Convergence Rate of Hurwitz Matrix

A square matrix $\mathbf{H}$ is called Hurwitz if

$$\mu(\mathbf{H}) := \max_{\lambda \in \mathrm{spec}(\mathbf{H})} \Re[\lambda] < 0 \ ,$$

where $\mathrm{spec}(\mathbf{H}) := \{\lambda_i\}$ denotes the set of eigenvalues of $\mathbf{H}$. If $\mathbf{H}$ is Hurwitz, then $\lim_{t \to +\infty} e^{\mathbf{H}t} = 0$.

Theorem 3 (Exponential Convergence of Hurwitz Matrices [68, Theorem 8.1])**.** *If* $\mathbf{H}$ *is Hurwitz, then there exist constants* $c, \alpha > 0$ *such that*

$$\|e^{\mathbf{H}t}\|_2 \leq ce^{-\alpha t}, \quad \textit{for all} \ \ t \geq 0,$$

where $-\alpha := \max_{\lambda \in \mathrm{spec}(\mathbf{H})} \Re[\lambda] + \epsilon$*, for some* $\epsilon > 0$ *that are small enough.*

When $\mathbf{H}$ is diagonalizable, i.e., when all Jordan blocks of $\mathbf{H}$ have size equal to 1, one can choose $-\alpha = \max_{\lambda \in \mathrm{spec}(\mathbf{H})} \Re[\lambda]$.

2.1.2 Uniform vector relative degree

Formally, we consider a nonlinear system as described in (2.3). Let $t \geq 0$ be a continuous time index, and $f_0 : \mathbb{R}^m \times \mathbb{R}_+ \to \mathbb{R}$ be a time-varying function of the output $\mathbf{y}$. Using $f_0(\mathbf{y}, t)$ we implicitly define our target trajectory; i.e., the *minimizing path*:

$$\mathbf{y}^*(t) = \arg \min_{\mathbf{y} \in \mathbb{R}^m} \; f_0(\mathbf{y}, t). \tag{2.8}$$

The goal is to generate a control input $\mathbf{u}(t)$ such that $\|\mathbf{y}(t) - \mathbf{y}^*(t)\|_2 \to 0$ as $t \to \infty$ for all initial conditions; i.e., global asymptotic convergence. The following assumption will be used throughout this paper.

Assumption 1 (Objective Function)**.** *The objective function $f_0(\mathbf{y}, t)$ is infinitely differentiable $(\mathbb{C}^\infty)$ with respect to both $\mathbf{y}$ and t, and is uniformly strongly convex in $\mathbf{y}$; i.e., $\nabla_{\mathbf{y}\mathbf{y}} f_0(\mathbf{y}(t), t) \succeq m_f \mathbf{I}_m$ for all $t \geq 0$ and for some $m_f > 0$.*

The remainder of this section provides two examples that help motivate both our goals and our solution approach.

Example #1: Integrator

We aim to design a control law for an integrator

$$\dot{\mathbf{x}} = \mathbf{u}, \quad \mathbf{y} = \mathbf{x}, \tag{2.9}$$

such that $\mathbf{y}$ converges asymptotically to the optimal solution of time-varying optimization problem (2.8):

$$\mathbf{y}^*(t) = \arg \min_{\mathbf{y}} \; f_0(\mathbf{y}, t).$$

Notice that, even though we can instantaneously change the speed and direction of $y(t)$ in (2.9), the initial condition $y(0)$ may not match $y^*(0)$. This is illustrated in Figure 2-1.

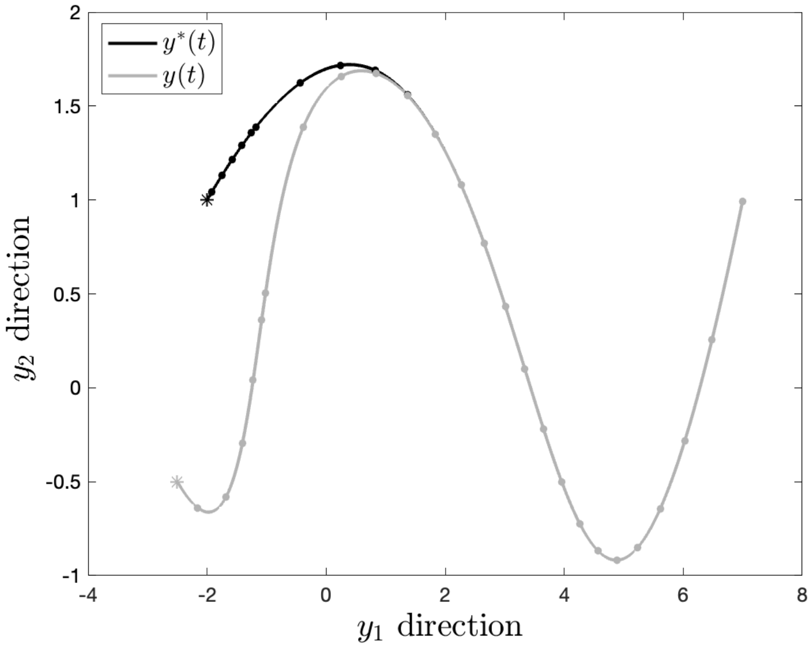

Figure 2-1. The plot of a robot tracking an object, where $\mathbf{y}^*(t)$ (2.8) is simply the object trajectory and $\mathbf{y}(t)$ represents the real trajectory of the robot. Due to mismatching initial conditions (highlighted using asterisk), we design a control law that converges asymptotically to the target trajectory $\mathbf{y}^*(t)$.

This problem can be overcome by finding a control law that transforms (2.9) into the following optimization dynamics

$$\dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) = -\mathbf{P}\nabla_{\mathbf{y}} f_0(\mathbf{y}, t), \quad \mathbf{P} \succ 0, \tag{2.10}$$

where the gradient $\nabla_{\mathbf{y}} f_0(\mathbf{y}, t)$ is driven to zero exponentially fast [32, 34]. Thus, since by convexity (see Assumption 1), the optimal trajectory $\mathbf{y}^*(t)$ is characterized by $\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t) = 0$, the controlled $\mathbf{y}$ asymptotically reaches $\mathbf{y}^*(t)$.

To achieve this transformation, we first characterize the required evolution of $\mathbf{y}$ for (2.10) to hold, and then define the proper control law. Using the chain rule to differentiate the gradient term with respect to time yields

$$\dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) = \nabla_{\mathbf{yy}} f_0(\mathbf{y}, t)\dot{\mathbf{y}} + \nabla_{\mathbf{y}t} f_0(\mathbf{y}, t).$$

Then, by combining (2.10) and the above equation, we find that $\dot{\mathbf{y}}$ is implicitly defined by

$$\dot{\mathbf{y}}_{\text{imp}} = -\nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y}, t)[\mathbf{P}\nabla_{\mathbf{y}} f_0(\mathbf{y}, t) + \nabla_{\mathbf{y}t} f_0(\mathbf{y}, t)]. \tag{2.11}$$

Finally, since by (2.9), $\mathbf{u} = \dot{\mathbf{y}}$, equation (2.11) leads to the control:

$$\mathbf{u} = -\nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y}, t)[\mathbf{P}\nabla_{\mathbf{y}} f_0(\mathbf{y}, t) + \nabla_{\mathbf{y}t} f_0(\mathbf{y}, t)].$$

The above control law implicitly transforms (2.9) into (2.10). Further, it has a nice optimization-based interpretation consisting of two terms [32, 34]:

1. a *prediction term* $-\nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y}, t)\nabla_{\mathbf{y}t} f_0(\mathbf{y}, t)$,which tracks the change of the optimal solution; i.e., target trajectory,
2. and a *correction term* $-\nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y}, t)\mathbf{P}\nabla_{\mathbf{y}} f_0(\mathbf{y}, t)$, which acts as a proportional controller that cancels the optimality error and drives the system toward the optimum.

Unfortunately, the solution approach shown in this example critically relies on the integrator structure in (2.9) that allows one to arbitrarily control $\dot{\mathbf{y}}$ by choosing $\mathbf{u}$. However, for a general nonlinear system, satisfying (2.10) may not be possible. This is shown in the next example.

Example #2: Wheeled Mobile Robot

We now show how to extend the approach described above for a more involved example where we aim to drive a nonholonomic wheeled mobile robot (WMR) [69, 66]:

$$\dot{x}_1 = \cos(x_3)u_1, \tag{2.12a}$$

$$\dot{x}_2 = \sin(x_3)u_1, \tag{2.12b}$$

$$\dot{x}_3 = u_2, \tag{2.12c}$$

$$\mathbf{y} = (x_1, x_2), \tag{2.12d}$$

such that $\mathbf{y}$ converges asymptotically to the optimal solution of time-varying optimization problem (2.8).

If we once again want (2.12) to match the dynamics (2.10), we need (2.11) to hold. However, it follows from (2.12) that $\dot{\mathbf{y}} = [\cos(x_3)u_1, \sin(x_3)u_1]^T$, which is ill-defined.

It is obvious that one cannot control every direction of $\dot{\mathbf{y}}$ with this ill-defined equation, therefore, cannot derive a control law that ensures (2.11).

This motivates the search for an alternative to (2.10) that has the equivalent effect of driving $\mathbf{y}$ towards $\mathbf{y}^*(t)$. Instead, we seek to transform (2.12) into

$$\begin{bmatrix} \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) \\ \ddot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I}_m \\ -k_{\mathrm{p}}\mathbf{I}_m & -k_{\mathrm{d}}\mathbf{I}_m \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{y}} f_0(\mathbf{y}, t) \\ \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) \end{bmatrix}, \tag{2.13}$$

where $k_{\mathrm{p}}, k_{\mathrm{d}} > 0$, which defines a Hurwitz matrix, and $\mathrm{col}(\nabla_{\mathbf{y}} f_0(\mathbf{y}, t), \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t))$ can be interpreted as the optimality error of $\mathbf{y}$, and its time derivative.

To find the control law that transforms (2.12) into (2.13), we can differentiate the gradient term with respect to time twice:

$$\ddot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) = \nabla_{\mathbf{yy}} f_0(\mathbf{y}, t)\ddot{\mathbf{y}} + \dot{\nabla}_{\mathbf{yy}} f_0(\mathbf{y}, t)\dot{\mathbf{y}} + \dot{\nabla}_{\mathbf{y}t} f_0(\mathbf{y}, t).$$

Now combining once again the second row of (2.13) and the above equation leads to the following implicit condition for the acceleration $\ddot{\mathbf{y}}$:

$$\ddot{\mathbf{y}}_{\mathrm{imp}} = -\nabla^{-1}_{\mathbf{yy}} f_0(\mathbf{y}, t)\left[\dot{\nabla}_{\mathbf{yy}} f_0(\mathbf{y}, t)\dot{\mathbf{y}} + \dot{\nabla}_{\mathbf{y}t} f_0(\mathbf{y}, t) + k_{\mathrm{p}}\nabla_{\mathbf{y}} f_0(\mathbf{y}, t) + k_{\mathrm{d}}\dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t)\right] \tag{2.14}$$

Finally, by differentiating $\mathbf{y}$ with respect to time twice we notice that the matrix on the right-hand side of

$$\ddot{\mathbf{y}} = \begin{bmatrix} \cos(x_3) & -\sin(x_3)u_1 \\ \sin(x_3) & \cos(x_3)u_1 \end{bmatrix} \begin{bmatrix} \dot{u}_1 \\ u_2 \end{bmatrix} \tag{2.15}$$

is invertible for every nonzero u_1 and thus, we can use $(\dot{u}_1, u_2)$ to control $\ddot{\mathbf{y}}$ to follow (2.14):

$$\begin{bmatrix} \dot{u}_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos(x_3) & -\sin(x_3)u_1 \\ \sin(x_3) & \cos(x_3)u_1 \end{bmatrix}^{-1} \ddot{\mathbf{y}}_{\mathrm{imp}}.$$

As long as $u_1 \neq 0$, the control law is well-defined by introducing u_1 as an auxiliary state.

We finalize this section showing a particular case of (2.14) that is familiar for most control audience. If the task is simply tracking a moving object, we can define the following time-varying problem:

$$\mathbf{y}^*(t) = \arg\min_{\mathbf{y}} \tfrac{1}{2}\|\mathbf{y} - \mathbf{y}_{\mathrm{d}}(t)\|_2^2,$$

where $\mathbf{y}_{\mathrm{d}}(t)$ represents the target trajectory. And according to (2.14), the implicitly defined trajectory takes the form:

$$\ddot{\mathbf{y}}_{\mathrm{imp}} = \ddot{\mathbf{y}}_{\mathrm{d}}(t) - k_p(\mathbf{y} - \mathbf{y}_{\mathrm{d}}(t)) - k_d(\dot{\mathbf{y}} - \dot{\mathbf{y}}_{\mathrm{d}}(t)).$$

Thus, in this case, equation $\ddot{\mathbf{y}}_{\mathrm{imp}}$ can be interpreted as a common Proportional-Derivative (PD) controller.

The above motivating example shows how to extend the algorithm from a first-order system (an integrator) to a second-order system (a unicycle). In this Section, we aim to carry this procedure over to a more general setting.

We assume now that the system under consideration has a uniform vector relative degree, which will in general need to be achieved via dynamic extension. This is a natural extension from the WMR model, where the vector relative degree is $\{2, 2\}$ and $n = 4$.

Assumption 2 (Uniform Vector Relative Degree)**.** *The multivariable nonlinear system (2.6) has vector relative degree $r_1 = \cdots = r_m = k$ and $m \times k = n$.*

Based on Lemma 2, it is straightforward that for a multivariable nonlinear system satisfying Assumption 2, the feedback function (2.7) and a state diffeomorphism $\mathbf{z} = \Phi(\mathbf{x}, \boldsymbol{\zeta})$ will transform the composite system (2.6) into $\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{v}$, with $(\mathbf{A}, \mathbf{B})$ in Brunovsky Canonical Form. By computing the higher derivatives of output channel, we can implicitly design the trajectory for $\mathbf{y}$ using $\mathrm{col}(\nabla_{\mathbf{y}} f_0(\mathbf{y},t),\ldots,\nabla_{\mathbf{y}}^{(k-1)} f_0(\mathbf{y},t))$ as a proxy for optimality error, where the goal is to construct the following dynamical system:

$$\begin{bmatrix} \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y},t) \\ \vdots \\ \nabla_{\mathbf{y}}^{(k)} f_0(\mathbf{y},t) \end{bmatrix} = \mathbf{H} \begin{bmatrix} \nabla_{\mathbf{y}} f_0(\mathbf{y},t) \\ \vdots \\ \nabla_{\mathbf{y}}^{(k-1)} f_0(\mathbf{y},t) \end{bmatrix}, \tag{2.16}$$

with

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ a_0 & a_1 & a_2 & \ldots & a_{k-1} \end{bmatrix} \otimes \mathbf{I}_m \tag{2.17}$$

being Hurwitz. The following technical lemma will be used during the calculation of new optimality error state.

Lemma 4 (Gradient Time Differentiation)**.** *Differentiating the gradient $\nabla_{\mathbf{y}} f_0(\mathbf{y},t)$ with respect to time $k-$times yields:*

$$\nabla_{\mathbf{y}}^{(k)} f_0(\mathbf{y},t) = \sum_{m=0}^{k-1} \binom{k-1}{m} \nabla_{\mathbf{yy}}^{(m)} f_0(\mathbf{y},t)\mathbf{y}^{(k-m)} + \nabla_{\mathbf{y}t}^{(k-1)} f_0(\mathbf{y},t), \tag{2.18}$$

where $\binom{k-1}{m}$ represents the binomial coefficient.

Proof: See A2.1.5.

Combining (2.16) and (2.18), we can implicitly design the trajectory for $\mathbf{y}$ by:

$$\begin{aligned} \mathbf{y}_{\text{imp}}^{(k)} = \nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y},t)[\sum_{i=0}^{k-1} a_i \nabla_{\mathbf{y}}^{(i)} f_0(\mathbf{y},t) \\ - \sum_{m=1}^{k-1} \binom{k-1}{m} \nabla_{\mathbf{yy}}^{(m)} f_0(\mathbf{y},t)\mathbf{y}^{(k-m)} - \nabla_{\mathbf{y}t}^{(k-1)} f_0(\mathbf{y},t)]. \end{aligned} \tag{2.19}$$

Now, we formally provide our solution for systems with uniform relative degrees.

Theorem 5 (Control Law for Uniform Vector Relative Degree Systems)**.** *Consider the multivariable system defined as* (2.3) *and the time-varying optimization problem defined as*

(2.8). *If both assumptions 1 and 2 are satisfied, then the system will globally asymptotically converge to the optimal solution of* (2.8)*, by using the control law:*

$$\mathbf{u} = \alpha(\mathbf{x}, \boldsymbol{\zeta}) + \beta(\mathbf{x}, \boldsymbol{\zeta})\mathbf{R}(\mathbf{x}, \boldsymbol{\zeta})^{-1}[\mathbf{y}_{\text{imp}}^{(k)} - \mathbf{p}(\mathbf{x}, \boldsymbol{\zeta})], \tag{2.20}$$

where $\mathbf{y}_{\text{imp}}^{(k)}$ *is given in* (2.19) *and the dynamic feedback function defined in* (2.7)*. Moreover, the following inequalities hold:*

$$\begin{aligned}
&\|\mathbf{y}(t) - \mathbf{y}^*(t)\|_2 \leq Ce^{-\alpha t},\\
&0 \leq f_0(\mathbf{y}(t), t) - f_0(\mathbf{y}^*(t), t) \leq m_f C^2 e^{-2\alpha t},\\
&0 < C = \left(\tfrac{c^2}{m_f^2} \sum_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} f_0(\mathbf{y}(0), 0)\_2^2)\right)^{\frac{1}{2}} < \infty,
\end{aligned}$$

for some constant $C > 0$*,* $-\alpha = \max\{\Re(\lambda_i) + \epsilon, i \in [1...n]$*, for some* $\epsilon > 0$ *small enough.*

Proof: See Section 2.1.5.

Theorem 5 makes a strong assumption on the structure of the nonlinear system, which is that the system must have equal vector degree $\{r_1 = \cdots = r_m\}$. In the next section, we relax this assumption.

2.1.3 Non-uniform vector relative degree

We now consider the less restrictive assumption.

Assumption 3 (Non-Uniform Vector Relative Degree)**.** *The multivariable nonlinear system* (2.6) *has vector relative degree* $\{r_1, \ldots, r_m\}$ *and* $r_1 + r_2 + \cdots + r_m = n$*.*

As a result of Assumption 3, the order of Lie differentiation of each channel is different (c.f.(2.4)) and we cannot directly design the trajectory as in (2.19). However, remember that according to Lemma 2, the system is transformed into $\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{v}$, with $(\mathbf{A}, \mathbf{B})$ in Brunovsky Canonical Form. As a matter of fact, the input v_i controls only the output y_i throughout a chain of r_1 integrators. If $\{r_1, ...r_m\}$ are not equal, we can always introduce $k - r_i$ auxiliary states (integrators) for each channel y_i, where $k = \max\{r_1, r_2, \ldots, r_m\}$ and define the new input s_i accordingly. Notice that this construction makes a dynamic extension of $\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{v}$ that possesses uniform order of Lie differentiation of each channel. For example, for channel y_i, we introduce the following states $\xi_1^i = v_i, \xi_2^i = \dot{\xi}_1^i, \ldots, \dot{\xi}_{k-r_i}^i = s_i$. More specifically, the auxiliary states ξ should satisfy the following dynamic:

$$\boldsymbol{v} = \tilde{\alpha}(\boldsymbol{\xi}) + \tilde{\beta}(\boldsymbol{\xi})\boldsymbol{s}, \tag{2.21a}$$

$$\dot{\boldsymbol{\xi}} = \tilde{\gamma}(\boldsymbol{\xi}) + \tilde{\delta}(\boldsymbol{\xi})\boldsymbol{s}. \tag{2.21b}$$

Then the feedback function (2.7), the auxiliary states dynamic of $\boldsymbol{\xi}$ (2.21), and a state diffeomorphism $\mathbf{z} = \Phi(\mathbf{x}, \boldsymbol{\zeta}, \boldsymbol{\xi})$ will transform the composite system (2.6) into

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{s},$$

with $\mathbf{A}, \mathbf{B}$ in Brunovsky Canonical Form.

Theorem 6 (Control Law for General Vector Relative Degree System)**.** *Consider the multivariable system defined as* (2.3) *and the time-varying optimization problem defined as* (2.8)*. Suppose that both Assumption* 4 *and Assumption* 3 *are satisfied, then the system will globally asymptotically converge to the optimal solution of* (2.8)*, by using the control law:*

$$\mathbf{u} = \alpha(\mathbf{x}, \boldsymbol{\zeta}) + \beta(\mathbf{x}, \boldsymbol{\zeta})\mathbf{R}^{-1}(\mathbf{x}, \boldsymbol{\zeta})[\tilde{\alpha}(\boldsymbol{\xi}) + \tilde{\beta}(\boldsymbol{\xi})\mathbf{y}_{\mathrm{imp}}^{(k)} - \mathbf{p}(\mathbf{x}, \boldsymbol{\zeta})] \tag{2.22}$$

where $\mathbf{y}_{\mathrm{imp}}^{(k)}$ *be the solution of* (2.19)*, the dynamic feedback function defined in* (2.7) *and the auxiliary states* ξ *satisfy* (2.21)*. Moreover, the following inequalities hold:*

$$\begin{aligned}
&\|\mathbf{y}(t) - \mathbf{y}^*(t)\|_2 \le Ce^{-\alpha t}, \\
&0 \le f_0(\mathbf{y}(t), t) - f_0(\mathbf{y}^*(t), t) \le m_f C^2 e^{-2\alpha t}, \\
&0 < C = \left(\tfrac{c^2}{m_f^2} \sum\nolimits_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} f_0(\mathbf{y}(0), 0)\|_2^2\right)^{\frac{1}{2}} < \infty,
\end{aligned}$$

for some constant $C > 0$, $-\alpha = \max \Re(\lambda_i) + \epsilon, i \in [1 \ldots n]$, *for some* $\epsilon > 0$ *small enough. Proof: Feedback function of the form* (2.22) *results in* $\mathrm{col}(y_1^{(k)}, \ldots, y_m^{(k)}) = \mathbf{y}_{\mathrm{imp}}^{(k)}$*, where* $\mathbf{y}_{\mathrm{imp}}^{(k)}$ *is the solution of* (2.19)*. The rest of the proof follows Theorem* 5*.*

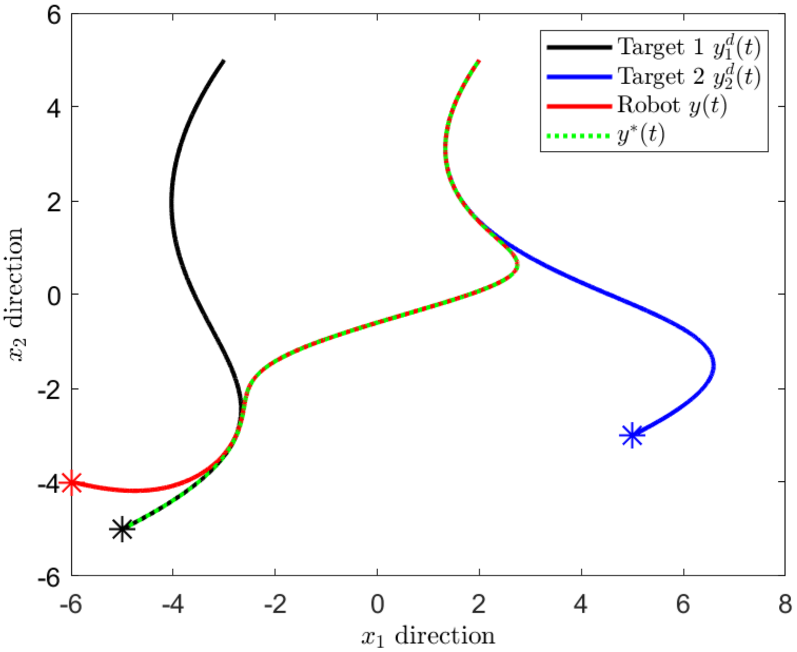

Figure 2-2. Trajectory of the optimal solution $\mathbf{y}^*(t)$ (dashed green), the robot (solid red), and the objects (black for 1 and blue for 2). The robot converges to the target trajectory, which is to track the first object from $[0s, 5s]$ and gradually switch to track the second object in $[5s, 15s]$.

2.1.4 Numerical experiments

In this section, we illustrate how to leverage the time-varying optimization algorithm to solve the following robot tracking problems.

Switching Tracking Goals

Consider a wheeled mobile robot (2.12) charged with the task of tracking two moving objects sequentially. In the first time interval $[t_0, t_s]$, the agent is required to track the first object and in the second time interval $[t_s, t_f]$ gradually switched to track the second object. The equivalent time-varying optimization problem takes the following form:

$$\min_{\mathbf{y}} S(t)\|\mathbf{y} - \mathbf{y}_1^d(t)\|_2^2 + (1 - S(t))\|\mathbf{y} - \mathbf{y}_2^d(t)\|_2^2,$$

where $\mathbf{y}(t)$ is the robot position satisfying (2.12), $\mathbf{y}_1^d(t), \mathbf{y}_2^d(t)$ represents the position of moving objects at time t respectively. The smooth switch function $S(t)$ takes the form: $S(t) = 0.5 - 0.5\tanh(\frac{t-a}{b})$, where the parameters a and b can be used to define the switch point and the smoothing level. The target trajectories are designed via time parametric representation, where we use differential flatness in this trajectory generation problem [70]. Specifically, we parametrize the components of the flat output $\boldsymbol{\phi}_1 = \mathbf{y} = [x_1, x_2], \boldsymbol{\phi}_2 = \dot{\mathbf{y}}$, by

$$\phi_i(t) = \sum_{j=0}^{n-1} A_{ij}\lambda_j(t),$$

where the $\lambda_j(t) = t^j$ are the standard polynomial basis functions and the degree of the polynomial is set to be $n = 4$. Thus, the trajectory generation problem reduces from finding a function to finding a set of parameters.

The resulting trajectories we proposed are illustrated in Figure 2-2, with $a = 10, b = 1.5$. It can be observed that the robot successfully tracks the first object up to time $t_s = 5s$, gradually switching to the second object until $t_f = 15s$, and track the second object until simulation stops. Particularly, the randomly picked starting positions (highlighted using asterisk) for the two objects are $[-5, -5]$ and $[5, -3]$ and the agent is positioned randomly near the starting position, which is $[-5, 4]$. We set $t_0 = 0s$ and the total simulation time is $20s$. For this implementation, the differential equation (2.12) is solved based on an explicit Runge-Kutta $(4, 5)$ formula, the Dormand-Prince pair.

Multi-robot Navigation

In this numerical example, two agents are required to track two moving objects respectively, but the maximum distance between two agents is limited (e.g., due to communication or formation constraints). We assume $\mathbf{y}_1(t), \mathbf{y}_2(t)$ representing the current position of each robot, whose dynamic are unicycles satisfying (2.12). We consider the following time-varying optimization problem for this task:

$$\min_{\mathbf{y}_1,\mathbf{y}_2} \|\mathbf{y}_1-\mathbf{y}_1^d(t)\|_2^2+\|\mathbf{y}_2-\mathbf{y}_2^d(t)\|_2^2+H(\|\mathbf{y}_1-\mathbf{y}_2\|_2^2),$$

where $\mathbf{y}_1^d(t), \mathbf{y}_2^d(t)$ represents the current position of the moving object. $H(x) = \alpha \tan(\frac{x\pi}{2d})^2$ is a smooth barrier function, where the parameter d determines the maximum distance allowed for the two agents and α determines the flatness of penalty gain. In this scenario, although our theory does not exactly holds since the barrier function is not defined globally, as long as the initial conditions are not violated, the numerical result suggests that our algorithm can be applied beyond the presented assumptions.

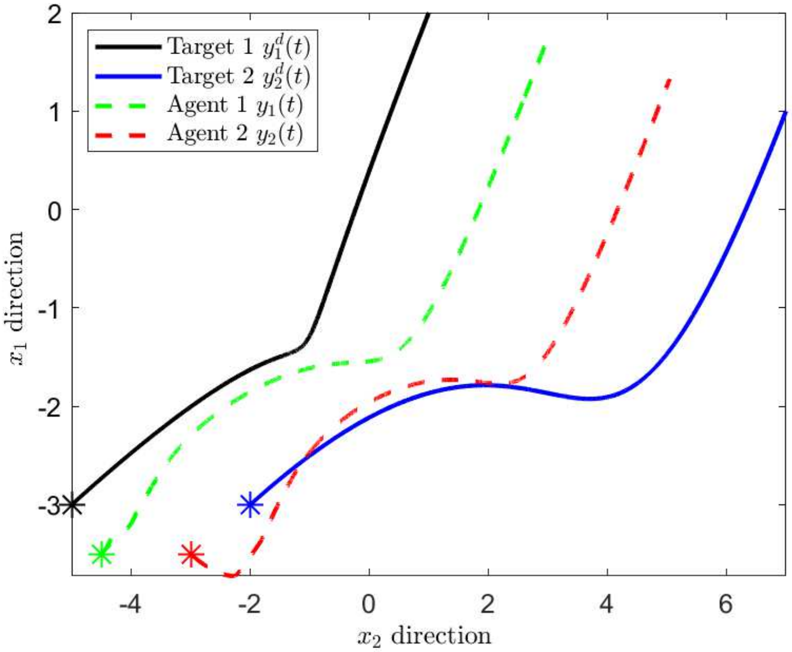

Figure 2-3. Trajectories of two objects $\mathbf{y}_1^d(t), \mathbf{y}_2^d(t)$ (solid) and two agents $\mathbf{y}_1, \mathbf{y}_2$ (dashed). Agents succeed in tracking objects while satisfying distance constraints between them.

The trajectories for the objects were also in time parametric representation, following the same computing procedure as in the previous section. Particularly, the randomly picked starting position (using asterisk) for two objects are $[-5, -3]$

and $[-2, -3]$, respectively. The maximum allowed distance is set to be $d = 2$, and the gain is $\alpha = 1e - 8$. As to the agents, they are positioned randomly near the starting position, while satisfying the distance constraint between them, which are $[-4.5, -3.5]$ and $[-3.5, -3.5]$ (using asterisk). For this implementation, the differential equation (2.12) is solved using the same procedure as in Section 2.1.4. The resulting trajectories are illustrated in Figure 2-3, where both robots, starting from arbitrary positions succeed in tracking the moving object and keep the maximum distance within limits simultaneously.

2.1.5 Appendix

Proof of Lemma 4

We prove this by mathematical induction. First, we consider when $k = 1$ and 2.

$$\begin{aligned}
\dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) &= \frac{\partial \nabla_{\mathbf{y}} f_0(\mathbf{y}, t)}{\partial \mathbf{y}} \dot{\mathbf{y}} + \frac{\partial \nabla_{\mathbf{y}} f_0(\mathbf{y}, t)}{\partial t} \\
&= \nabla_{\mathbf{yy}} f_0(\mathbf{y}, t) \dot{\mathbf{y}} + \nabla_{\mathbf{y}t} f_0(\mathbf{y}, t) \\
\ddot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) &= \frac{d}{dt} (\nabla_{\mathbf{yy}} f_0(\mathbf{y}, t) \dot{\mathbf{y}} + \nabla_{\mathbf{y}t} f_0(\mathbf{y}, t)) \\
&= \nabla_{\mathbf{yy}} f_0(\mathbf{y}, t) \ddot{\mathbf{y}} + \dot{\nabla}_{\mathbf{yy}} f_0(\mathbf{y}, t) \dot{\mathbf{y}} + \dot{\nabla}_{\mathbf{y}t} f_0(\mathbf{y}, t)
\end{aligned}$$

We want to show that for every $k \geq k_0$, $k_0 \geq 2$, if the statement holds for k, then it holds for $k + 1$.

$$\begin{aligned}
\nabla_{\mathbf{y}}^{(k)} f_0(\mathbf{y}, t) = \sum_{m=0}^{k-1} \binom{k-1}{m} \nabla_{\mathbf{yy}}^{(m)} f_0(\mathbf{y}, t) \mathbf{y}^{(k-m)} \\
+ \nabla_{\mathbf{yt}}^{(k-1)} f_0(\mathbf{y}, t)
\end{aligned}$$

Using the binomial theorem we obtain:

$$\begin{aligned}\nabla_{\mathbf{y}}^{(k+1)} f_0(\mathbf{y},t) &= \frac{d}{dt}(\sum_{m=0}^{k-1}\binom{k-1}{m}\nabla_{\mathbf{yy}}^{(m)} f_0(\mathbf{y},t)\mathbf{y}^{(k-m)}) \\ &+ \frac{d}{dt}(\nabla_{\mathbf{y}t}^{(k-1)} f_0(\mathbf{y},t)) \\ &= \sum_{m=0}^{k}\binom{k}{m}\nabla_{\mathbf{yy}}^{(m)} f_0(\mathbf{y},t)\mathbf{y}^{(k+1-m)} \\ &+ \nabla_{\mathbf{y}t}^{(k)} f_0(\mathbf{y},t),\end{aligned}$$

which completes the proof.

Proof of Theorem 5

By uniformly strong convexity of $f_0(\mathbf{y},t)$ in $\mathbf{y}$, the Hessian inverse $\nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y},t)$ is defined for all $t \geq 0$. Because the vector relative degree of the nonlinear system is $r_1 = \cdots = r_m = k$, which means $\mathbf{y}^{(k)}$ has a linear relationship with new input $\mathbf{v}$. According to Lemma 4, we have (2.18). Furthermore, as a result of Theorem 2, feedback function of the form (2.20) results in $\mathbf{y}^{(k)} = \mathbf{y}_{\text{imp}}^{(k)}$, where $\mathbf{y}_{\text{imp}}^{(k)}$ is the solution of (2.19).

Now, we are able to construct the desired dynamical system (2.16), where $\mathbf{H}$ is the designed Hurwitz matrix, and the solution of this ODE is:

$$\begin{bmatrix} \nabla_{\mathbf{y}} f_0(\mathbf{y},t) \\ \vdots \\ \nabla_{\mathbf{y}}^{(k-1)} f_0(\mathbf{y},t) \end{bmatrix} = e^{\mathbf{H}t} \begin{bmatrix} \nabla_{\mathbf{y}} f_0(\mathbf{y}(0),0) \\ \vdots \\ \nabla_{\mathbf{y}}^{(k-1)} f_0(\mathbf{y}(0),0) \end{bmatrix} \tag{2.23}$$

where $\mathbf{y}(0) \in R^m$ is the initial point. By taking the Frobenius norms of both sides and applying Theorem 3 we obtain

$$\sum_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} f_0(\mathbf{y},t)\|_2^2 \leq c^2 e^{-2\alpha t}(\sum_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} f_0(\mathbf{y}(0),0)\|_2^2) \tag{2.24}$$

for some constant $c > 0$, $-\alpha = \max \Re(\lambda_i) + \epsilon, i \in [1...n]$, for some $\epsilon > 0$ small enough.

Next, we use the mean-value theorem to expand $\nabla_{\mathbf{y}} f_0(\mathbf{y}, t)$ with respect to $\mathbf{y}$ as follows, where $\boldsymbol{\eta(t)}$ is a convex combination of $\mathbf{y}(t)$ and $\mathbf{y}^*(t)$. Additionally using the fact that $\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t) = 0$ for all $t \geq 0$, we obtain:

$$\mathbf{y}(t) - \mathbf{y}^*(t) = \nabla_{\mathbf{yy}}^{-1} f_0(\boldsymbol{\eta}(t), t) \nabla_{\mathbf{y}} f_0(\mathbf{y}(t), t). \tag{2.25}$$

It follows from Assumption 1, that $\|\nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y}, t)\|_2 \leq m_f^{-1}$. Taking the norm on both sides together with equation (2.24) we have:

$$\begin{aligned} &\|\mathbf{y}(t) - \mathbf{y}^*(t)\|_2 \leq C e^{-\alpha t}, \\ &0 \leq C = \left(\tfrac{c^2}{m_f^2} \sum_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} f_0(\mathbf{y}(0), 0)\|_2^2 \right)^{\frac{1}{2}} < \infty. \end{aligned} \tag{2.26}$$

On the other hand, convexity of $f_0(\mathbf{y}, t)$ implies that for each $t \geq 0$

$$0 \leq f_0(\mathbf{y}, t) - f_0(\mathbf{y}^*, t) \leq \nabla_{\mathbf{y}} f_0(\mathbf{y}, t)^T (\mathbf{y} - \mathbf{y}^*) \tag{2.27}$$

By applying Cauchy-Swhartz inequality on the right-hand side we obtain;

$$0 \leq f_0(\mathbf{y}(t), t) - f_0(\mathbf{y}^*(t), t) \leq m_f C^2 e^{-2\alpha t} \tag{2.28}$$

which completes the proof.

2.2 Differentially flat systems

In this section, we further extend the work in Section 2.1, including extensions from feedback linearization to general differentially flat systems, as well as the inclusion of equality and inequalities constraints. Furthermore, we generalize the notion of feedback linearization, which makes nonlinear systems behave as linear systems, and develop controllers that effectively transform a differentially flat system into an optimization algorithm that seeks to find the optimal solution of a (possibly time-varying) optimization problem.

Section 2.2.1 introduces some preliminary definitions and tools we use. We revisit the two motivating examples in Section 2.1.2, Integrator and Wheeled Mobile

Robot, with a more generalized time-varying optimization framework. Furthermore, we summarize the key features of the proposed framework, which generalize the notion of feedback linearization and transform a flat system into an optimization algorithm. Designing the nonlinear feedback controller is reduced to finding a solution to the ODE system that satisfies the optimization dynamic and system dynamic simultaneously. Our major results are illustrated in section 2.2.3,2.2.4 and 2.2.5, which deal with unconstrained and constrained time-varying optimization problems respectively. To illustrate our results, we perform two numerical evaluations in 2.2.6 on multi-object tracking problem and obstacle avoidance problem, to illustrate the effectiveness of our framework.

2.2.1 Preliminaries

Differential Flatness and Feedback Linearization

Over the past several decades, differential flatness theory has been a main direction in the area of nonlinear control for motion planning, trajectory generation, and stabilization [71]. Roughly speaking, flat systems are those systems that are equivalent to a controllable linear one, namely a system made of chains of integrators of arbitrary length [70]. A system is differentially flat if there exist outputs, called flat outputs, for which all states and inputs are determined by the outputs and a finite number of their derivatives [72]. More precisely, if the system has states $\mathbf{x} \in \mathbb{R}^n$ and inputs $\mathbf{u} \in \mathbb{R}^m$, described as a system of differential equations

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}). \tag{2.29}$$

Then the system (2.29) is differentially flat if we can find flat outputs $\mathbf{y} \in \mathbb{R}^m$ of the form

$$\mathbf{y} = h(\mathbf{x}, \mathbf{u}^{[r]}), \tag{2.30}$$

such that

$$\mathbf{x} = \varphi(\mathbf{y}^{[k]}), \quad \mathbf{u} = \alpha(\mathbf{y}^{[k]}). \tag{2.31}$$

The advantage of using flat outputs in control system design is that doing so simplifies the process of generating input trajectories that satisfy certain constraints. Instead of designing complex controllers that directly manipulate the control inputs, one can design controllers that manipulate the flat outputs, and then use the algebraic relationships between the flat outputs and the control inputs to generate optimal trajectories.

Notably, many commonly used classes of systems in nonlinear control theory are differentially flat, for example fully actuated holonomic systems, mobile robots, and classical n-trailer systems. A complete characterization of differential flatness and a catalog of finite dimensional flat systems can be found in [71, 70].

Another important concept in nonlinear control theory is *(dynamic) feedback linearization*, which means a nonlinear system can be transformed into a linear system by a state diffeomorphism and a feedback transformation [36]. Although differential flatness and feedback linearization are related concepts, they are inherently different. In fact, all state feedback linearizable systems are differentially flat. However, a differentially flat system is dynamic feedback linearizable on an open dense set, which may not include the equilibrium points. Differential flatness is an inherent geometric property of a system, independent of coordinate representation and thus we can exploit its inherent geometric structure in designing control algorithms [70].

2.2.2 Problem Statement

We formally state the problem together with some regularity assumptions needed in our derivations and introduce two motivating examples. Consider a differentially flat system described as in (2.29), along with the associated flat output

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \quad \mathbf{y} = h(\mathbf{x}, \mathbf{u}^{[r]}),$$

The goal is to generate a control input $\mathbf{u}(t)$ such that for some $C > 0$, $\|\mathbf{y}(t) - \mathbf{y}^*(t)\|_2 \leq Ce^{-\alpha t}$ for all $t \geq 0$ and all initial conditions, i.e., global asymptotic convergence, where $\mathbf{y}^*(t)$ is the solution to (2.2). Additionally, we assume the minimizer $\mathbf{y}^*(t)$ is unique for all $t \geq 0$ (see Assumption 4). The following regularity assumptions will be used throughout this section, and are commonly used in the context of time-varying optimization [31].

Assumption 4 (Uniform strong convexity). *The objective function $f_0(\mathbf{y}, t)$ is uniformly strongly convex in $\mathbf{y}$, i.e., $\nabla_{\mathbf{yy}} f_0(\mathbf{y}, t) \succeq m_f \mathbf{I}_m$ for some $m_f > 0$, for all $\mathbf{y}$ and for all $t \geq 0$. The inequality constraint functions $f_i(\mathbf{y}, t)$ are convex in $\mathbf{y}$ for all $t \geq 0$ and for all $i \in [p]$.*

Assumption 5 (Uniform Mangasarian-Fromowitz constraint qualification). *For a global minimum $\mathbf{y}^*(t)$ of (2.8)*

1. *there exists a uniformly bounded $\bar{\mathbf{d}}(t) \in \mathbb{R}^m$, i.e., $\|\bar{\mathbf{d}}(t)\|_2 \leq d$ for some constant $d > 0$, and a constant $\epsilon > 0$ such that*

$$\begin{aligned} \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)^T \bar{\mathbf{d}}(t) &\leq -\epsilon, \qquad i \in \mathbb{I}(\mathbf{y}^*(t)), \\ \mathbf{a}_j(t)^T \bar{\mathbf{d}}(t) &= 0, \qquad j = 1, \ldots q, \end{aligned}$$

   for all $t \geq 0$, where $\mathbb{I}(\mathbf{y}^*(t)) := \{i | f_i(\mathbf{y}^*(t), t) = 0\}$ denotes the index set associated with active inequality constraints.

2. *there exist constants $0 < \tau_{\min} \leq \tau_{\max} < +\infty$ such that $\sigma_{\min}(\mathbf{A}(t)) \geq \tau_{\min}$ and $\sigma_{\max}(\mathbf{A}(t)) \leq \tau_{\max}$ for all $t \geq 0$, i.e., the vectors $\{\mathbf{a}_j\}$ for $j \in [q]$ are uniformly linearly independent and uniformly bounded.*

Since the time-varying convex optimization problem has smooth objective and constraints functions, Assumption 4 and Assumption 5 imply that the Karush-Kuhn-

Tucker (KKT) conditions [73] provide necessary and sufficient conditions for optimality. Notice that these Assumptions are not written in the most familiar way. For example in Assumption 5, we are replacing the traditional $\nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)^T \bar{\mathbf{d}}(t) < 0$ with $\nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)^T \bar{\mathbf{d}}(t) \leq -\epsilon$ for some positive constants $\epsilon > 0$. Such modifications are made in order to exclude the possibility that, e.g., $\nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)^T \bar{\mathbf{d}}(t) \to 0$ as $t \to \infty$. In Section 2.2.5, we will show that these assumptions are sufficient for the time-varying optimization problem (2.8) to be well-defined for all $t \geq 0$, and exclude the possibility that the optimal dual variables escape to infinity exponentially fast, which was merely assumed to hold in prior work [32].

The remainder of this section provides two examples that help motivate both our goals and our solution approach. In Section 2.2.2, we begin with a linear system, an integrator, which is the simplest form of a flat system. We illustrate how to design a control law that steers it to the trajectory implicitly defined by an unconstrained time-varying optimization problem. We relate this case with recent research concerning *Prediction-Correction Methods* and describe a general methodology for control design wherein we match the evolution of the flat output with that of a time-varying gradient descent algorithm that converges to $\mathbf{y}^*(t)$. In Section 2.2.2, we extend the approach for a simple but representative second-order nonholonomic system, the Wheeled Mobile Robot (WMR), to illustrate how to incorporate kinematic or dynamic constraints in the system modeling by matching the flat output to a second order gradient descent algorithm. Lastly, we summarize the key features of the time-varying optimization-based framework to illustrate our solution approach, which effectively transforms a general flat system into an optimization algorithm that achieves asymptotic convergence to the optimal solution.

Example #1: Integrator

We revisit the analysis with the simplest possible flat system, the linear integrator

$$\dot{\mathbf{x}} = \mathbf{u}, \qquad \mathbf{y} = \mathbf{x}, \tag{2.32}$$

where $\mathbf{y}$ is the flat output. The inputs are determined by the flat outputs and a finite number of their derivatives, and this relationship (2.31) can be expressed using the following expression:

$$\mathbf{u} := \alpha(\mathbf{y}^{[k]}) = \dot{\mathbf{y}}. \tag{2.33}$$

For our purposes, it will be useful to represent it using the following implicit function:

$$\mathbb{F}(\dot{\mathbf{y}}, \mathbf{u}) := \dot{\mathbf{y}} - \mathbf{u} = \mathbf{0}. \tag{2.34}$$

We consider an unconstrained version of the time-varying optimization problem (2.8), in which our goal is to regulate the output $\mathbf{y}$ of the integrator (2.32) to asymptotically track the minimizer

$$\mathbf{y}^*(t) = \arg\min_{\mathbf{y}} \ f_0(\mathbf{y}, t). \tag{2.35}$$

The minimizer $\mathbf{y}^*(t)$ is characterized by $\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t) = 0$ (under convexity assumption 4). Converging to the minimizer $\mathbf{y}^*(t)$ can be solved by designing a control algorithm such that the output $\mathbf{y}$ of (2.32) satisfies the following target system :

$$\dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) = -\mathbf{P} \nabla_{\mathbf{y}} f_0(\mathbf{y}, t), \quad \mathbf{P} \succ 0, \tag{2.36}$$

where the gradient $\nabla_{\mathbf{y}} f_0(\mathbf{y}, t)$ is driven to zero exponentially fast, which by uniform strong convexity of f_0 (c.f. Assumption 4) makes $\mathbf{y}(t)$ converge to $\mathbf{y}^*(t)$ exponentially fast.

Thus, with (2.36) as our target, we first characterize the required evolution of $\mathbf{y}$ such that (2.36) holds. Using the chain rule to differentiate the gradient $\nabla_{\mathbf{y}} f_0(\mathbf{y}, t)$

with respect to time yields

$$\dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) = \nabla_{\mathbf{yy}} f_0(\mathbf{y}, t)\dot{\mathbf{y}} + \nabla_{\mathbf{y}t} f_0(\mathbf{y}, t).$$

Substituting the above into (2.36), the desired time-varying optimization dynamics can be equivalent described via the implicit function $\mathbb{G}(\dot{\mathbf{y}}, \mathbf{y}, t) = 0$, where

$$\mathbb{G}(\dot{\mathbf{y}}, \mathbf{y}, t) = \nabla_{\mathbf{yy}} f_0(\mathbf{y}, t)\dot{\mathbf{y}} + \nabla_{\mathbf{y}t} f_0(\mathbf{y}, t) + \mathbf{P}\nabla_{\mathbf{y}} f_0(\mathbf{y}, t), \tag{2.37}$$

which can be viewed as an implicit model for (2.36).

Consider now the combined implicit function $\mathbb{H}(\mathbf{y}, \dot{\mathbf{y}}, \mathbf{u}, t) := 0$ defined by simultaneously considering the previous two implicit equations

$$\mathbb{H}(\mathbf{y}, \dot{\mathbf{y}}, \mathbf{u}, t) = \begin{bmatrix} \mathbb{F}(\dot{\mathbf{y}}, \mathbf{u}) \\ \mathbb{G}(\dot{\mathbf{y}}, \mathbf{y}, t) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

We seek to resolve these implicit equations for a solution $(\dot{\mathbf{y}}, \mathbf{u}) = S(\mathbf{y}, t)$. In this simple case, by uniform strong convexity (Assumption 4), the Hessian matrix $\nabla_{\mathbf{yy}} f_0(\mathbf{y}, t)$ is uniformly positive definite. Consequently, one can uniquely solve $\mathbb{G}(\dot{\mathbf{y}}, \mathbf{y}, t) = \mathbf{0}$ for $\dot{\mathbf{y}}$ and then recover $\mathbf{u}$ from (2.34), yielding

$$\dot{\mathbf{y}} = \mathbf{u} = -\nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y}, t)[\mathbf{P}\nabla_{\mathbf{y}} f_0(\mathbf{y}, t) + \nabla_{\mathbf{y}t} f_0(\mathbf{y}, t)]. \tag{2.38}$$

This choice of control input therefore regulates the output of the integrator such that it asymptotically tracks the trajectory implicitly defined by an unconstrained time-varying optimization problem, i.e., $\mathbf{y}$ converges to the minimizer $\mathbf{y}^*(t)$ exponentially fast. Another key observation is that the proposed algorithm generalizes the notion of feedback linearization, in that it transforms the dynamical system into an optimization algorithm that seeks to find the optimizer of a time-varying optimization problem. Precisely, the nonlinear feedback control law (2.38) effectively transforms the integrator into the following linear system:

$$\dot{\mathbf{z}} = -\mathbf{P}\mathbf{z}, \quad \mathbf{P} \succ 0,$$

$$\mathbf{z} = \nabla_{\mathbf{y}} f_0(\mathbf{y}, t).$$

Such optimization algorithm exponentially converges to the optimal solution of the time-varying optimization problem.

Example #2: Wheeled Mobile Robot

We now show how the previous approach extends to a more involved example, where we aim to control a nonholonomic flat system, the wheeled mobile robot (WMR) [70]:

$$\begin{aligned} \dot{x}_1 &= \cos(x_3)u_1, & \dot{x}_2 &= \sin(x_3)u_1, \\ \dot{x}_3 &= u_2, & \mathbf{y} &= (x_1, x_2). \end{aligned} \tag{2.39}$$

The states $(x_1, x_2) \in \mathbb{R}^2$ represent the position, and x_3 is the angular position of the WMR. The control inputs (u_1, u_2) are the positional and angular velocities, respectively, and the position vector $\mathbf{y}$ is the flat output.

Consider again an unconstrained time-varying optimization problem (2.35), where our goal is to regulate the output vector $\mathbf{y}$ of a WMR to asymptotically track the time-varying minimizer.

Again, the inputs are determined by the flat outputs and a finite number of their derivatives, and this relationship (2.31) can be expressed using the following expression:

$$\mathbf{u} := \alpha(\mathbf{y}^{[k]}) = \begin{bmatrix} \sqrt{\dot{y}_1^2 + \dot{y}_2^2} \\ (\dot{y}_1\ddot{y}_2 - \ddot{y}_1\dot{y}_2)/(\dot{y}_1^2 + \dot{y}_2^2) \end{bmatrix}$$

Again, we implicitly express the input using the algebraic equation:

$$\mathbb{F}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \mathbf{u}) := \begin{cases} u_1 - \sqrt{\dot{y}_1^2 + \dot{y}_2^2} \\ u_2 - (\dot{y}_1\ddot{y}_2 - \ddot{y}_1\dot{y}_2)/(\dot{y}_1^2 + \dot{y}_2^2) \end{cases} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Accordingly, we could generalize the target system (2.36) by considering a second-order time-varying opimization algorithm:

$$\begin{bmatrix} \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) \\ \ddot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I}_m \\ -k_{\mathrm{p}}\mathbf{I}_m & -k_{\mathrm{d}}\mathbf{I}_m \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{y}} f_0(\mathbf{y}, t) \\ \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) \end{bmatrix}, \tag{2.40}$$

where $k_{\mathrm{p}}, k_{\mathrm{d}} > 0$ makes the gradient dynamics a Hurwitz linear system system with state $\mathbf{z} := (\nabla_{\mathbf{y}} f_0(\mathbf{y}, t), \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t))$.

Using the chain rule to differentiate the gradient term $\nabla_{\mathbf{y}} f_0(\mathbf{y}, t)$ with respect to time twice, we derive

$$\ddot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) = \nabla_{\mathbf{yy}} f_0(\mathbf{y}, t)\ddot{\mathbf{y}} + \dot{\nabla}_{\mathbf{yy}} f_0(\mathbf{y}, t)\dot{\mathbf{y}} + \dot{\nabla}_{\mathbf{y}t} f_0(\mathbf{y}, t).$$

Now combining once again the second row of (2.40) and the above equation leads to the following implicit function that describes the solution trajectory of the time-varying optimization problem,

$$\begin{aligned} \mathbb{G}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, t) :=& \nabla_{\mathbf{yy}} f_0(\mathbf{y}, t)\ddot{\mathbf{y}} + \dot{\nabla}_{\mathbf{yy}} f_0(\mathbf{y}, t)\dot{\mathbf{y}} \\ &+ \dot{\nabla}_{\mathbf{y}t} f_0(\mathbf{y}, t) + k_{\mathrm{p}} \nabla_{\mathbf{y}} f_0(\mathbf{y}, t) + k_{\mathrm{d}} \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) = 0, \end{aligned} \tag{2.41}$$

which can be viewed as an implicit model for (2.40). The problem of designing the control algorithm is reduced to define an implicit function of the form $(\ddot{\mathbf{y}}, \mathbf{u}) = S(\mathbf{y}, \dot{\mathbf{y}}, t)$ to the combined implicit function by simultaneously considering the previous two implicit equations

$$\mathbb{H}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \mathbf{u}, t) := \begin{cases} \mathbb{F}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \mathbf{u}) \\ \mathbb{G}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, t) \end{cases} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

where the two planning and tracking components are given by the two implicit functions. We could find the unique solution pair $(\ddot{\mathbf{y}}, \mathbf{u})$ to the ordinary differential equations:

$$\begin{aligned} \ddot{\mathbf{y}} :=& g(\mathbf{y}, \dot{\mathbf{y}}) = -\nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y}, t) \big[\dot{\nabla}_{\mathbf{yy}} f_0(\mathbf{y}, t)\dot{\mathbf{y}} + \dot{\nabla}_{\mathbf{y}t} f_0(\mathbf{y}, t) \\ &+ k_{\mathrm{p}} \nabla_{\mathbf{y}} f_0(\mathbf{y}, t) + k_{\mathrm{d}} \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t) \big] \\ u_1 =& \; \|\dot{\mathbf{y}}\|_2, \\ u_2 =& \; \frac{1}{\|\dot{\mathbf{y}}\|_2^2} \, g(\mathbf{y}, \dot{\mathbf{y}})^T \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \dot{\mathbf{y}} \end{aligned}$$

Again, we would like to emphasize that the proposed solution approach generalizes the notion of feedback linearization, where the above nonlinear feedback control law effectively transforms the WMR (2.39) into an optimization algorithm:

$$\begin{aligned}\dot{\mathbf{z}} &= \mathbf{H}\mathbf{z},\\ \mathbf{z} &= (\nabla_{\mathbf{y}} f_0(\mathbf{y}, t), \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y}, t))^T,\end{aligned}$$

where H is the Hurwitz matrix designed in (2.40). Such an optimization algorithm seeks to find the optimal solution of the unconstrained version of the time-varying optimization problem (2.8).

Key Features

We end this section by highlighting the key features that made the application of our framework possible in the two motivating examples. The framework we propose is a combination of two implicit functions

$$\mathbb{H}(\mathbf{y}^{[k]}, \mathbf{u}^{[r]}, t) := \begin{cases}\mathbb{F}(\mathbf{y}^{[k]}, \mathbf{u}^{[r]}) \\ \mathbb{G}(\mathbf{y}^{[k]}, t)\end{cases} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}$$

which comprise of:

1. a *system dynamics* or *tracking component* $\mathbb{F}(\mathbf{y}^{[r]}, \mathbf{u}^{[r]})$, which is an implicit function derived from the dynamical system characterizing the system input-output relationship;

2. an *optimization dynamic* or *planning component* $\mathbb{G}(\mathbf{y}^{[k]}, t)$, which is an implicit function derived from a set of target dynamics, guarantee asymptotic convergence to the minimizer of the time-varying optimization problem.

Thus, finding the desired controller can be reduced to the problem of finding a solution to this system of implicit functions. In the context of a flat system, the system dynamic term can be easily expressed using (2.31), resulting in $\mathbb{F}(\mathbf{y}^{[k]}, \mathbf{u}) := \mathbf{u} - \alpha(\mathbf{y}^{[k]}) = 0$. As a result, our focus in the subsequent sections shifts towards the optimization dynamic term. In the two illustrative examples, the optimization dynamics $\mathbb{G}(\mathbf{y}^{[k]}, t) = 0$ that we considered are obtained from an exponentially stable linear system. This system is characterized by a state comprising the gradient $\nabla_{\mathbf{y}} f_0(\mathbf{y}, t)$ and its higher-order time derivatives as given in (2.41).

In the rest of this paper, we seek to generalize this approach to tackle the specific problem: for an arbitrary differentially flat system (2.29) and a time-varying convex optimization problem (2.8), define an implicit function of the form $\mathbb{G}(\mathbf{y}^{[k]}, t) = 0$, such that its solutions globally asymptotically converge to the minimizer of a general time-varying constrained convex optimization problem. A key to the success of this effort is the design of general target systems of the form

$$\begin{aligned} &\dot{\mathbf{w}} = \mathbf{H}\mathbf{w}, \\ &\mathbf{w} = (\nabla_{\mathbf{z}} L(\mathbf{z}, t), ..., \nabla_{\mathbf{z}}^{k-1} L(\mathbf{z}, t))^T, \end{aligned} \tag{2.42}$$

where L, $\mathbf{z} = (\mathbf{y}, \lambda)$, and $\mathbf{H}$, are properly chosen to guarantee the asymptotic convergence of $y(t)$ to the optimal solution of a general constrained time-varying optimization problem of the form of (2.8).

2.2.3 Unconstrained time-varying optimization framework

In this section, we first consider the case where our goal is to regulate a general differentially flat system (2.29), to the minimizer $\mathbf{y}^*(t)$ of an unconstrained time-varying optimization problem (2.8), reproduced here for convenience as

$$\mathbf{y}^*(t) := \arg \min_{\mathbf{y} \in \mathbb{R}^m} \; f_0(\mathbf{y}, t). \tag{2.43}$$

Recall that for a differentially flat system (2.29), the inputs are determined by the flat outputs and a finite number of their derivatives according to (2.31). That is, the implicit function $\mathbb{F}(\mathbf{y}^{[k]}, \mathbf{u}) := \mathbf{u} - \alpha(\mathbf{y}^{[k]}) = 0$ represents the *system dynamics*, which is a function of up to k-th order derivatives of the flat output $\mathbf{y}$. Recall the time-varying optimization problem is assumed to be uniformly convex, (see Assumption 4) and infinitely differentiable. Building on the insights from previous section, we are motivated to devise a k-th order optimization dynamics that enables us to achieve convergence towards the unique minimizer $\mathbf{y}^*(t)$. Thus, we consider

$$\begin{bmatrix} \dot{\nabla}_{\mathbf{y}} f_0(\mathbf{y},t) \\ \vdots \\ \nabla_{\mathbf{y}}^{(k)} f_0(\mathbf{y},t) \end{bmatrix} = \mathbf{H} \begin{bmatrix} \nabla_{\mathbf{y}} f_0(\mathbf{y},t) \\ \vdots \\ \nabla_{\mathbf{y}}^{(k-1)} f_0(\mathbf{y},t) \end{bmatrix}, \tag{2.44}$$

with

$$\mathbf{H} = \hat{\mathbf{H}} \otimes \mathbf{I}_m, \quad \hat{\mathbf{H}} := \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ -a_0 & -a_1 & -a_2 & \dots & -a_{k-1} \end{bmatrix}$$

being Hurwitz. Equation (2.44) is a natural k-th order generalization of (2.36) and (2.40). The need to increase the order of the target system as the order of the flat system increases is evidenced by the following lemma.

Recall the result from Lemma 4, Differentiating the gradient $\nabla_{\mathbf{y}} f_0(\mathbf{y},t)$ with respect to time $k-$times yields

$$\nabla_{\mathbf{y}}^{(k)} f_0(\mathbf{y},t) = \sum_{m=0}^{k-1} \binom{k-1}{m} \nabla_{\mathbf{yy}}^{(m)} f_0(\mathbf{y},t)\mathbf{y}^{(k-m)} + \nabla_{\mathbf{y}t}^{(k-1)} f_0(\mathbf{y},t). \tag{2.45}$$

Lemma 4 shows that the k-th time derivative of the gradient is the first one where the term $\mathbf{y}^{(k)}$, which allows, in turn, to have access to the necessary information for control. Thus, now combining (2.44) and (2.45), for a general flat system and unconstrained time-varying optimization problem, we obtain the following implicit for the *optimization dynamics*:

$$\mathbb{G}_{\text{unc}}(\mathbf{y}^{[k]},t) := \sum_{m=0}^{k-1} \binom{k-1}{m} \nabla_{\mathbf{yy}}^{(m)} f_0(\mathbf{y},t)\mathbf{y}^{(k-m)} + \nabla_{\mathbf{y}t}^{(k-1)} f_0(\mathbf{y},t) + \sum_{i=0}^{k-1} a_i \nabla_{\mathbf{y}}^{(i)} f_0(\mathbf{y},t) = 0. \tag{2.46}$$

The next theorem states that if certain regularity conditions are met, the output $\mathbf{y}$, which follows the optimization dynamics described in equation (2.46), will asymptotically converge globally to the minimizer $\mathbf{y}^*(t)$ of equation (2.43).

Theorem 7 (Convergence of optimization dynamics (2.46))**.** *Let Assumption 4 hold. Then for any initial condition, the trajectory $t \mapsto \mathbf{y}(t)$ of system $\mathbb{G}_{\text{unc}}(\mathbf{y}^{[k]}, t) = 0$ defined in* (2.46) *globally asymptotically converges to the optimal solution $\mathbf{y}^*(t)$ of* (2.43)*. Moreover, the following bounds*

$$\|\mathbf{y}(t) - \mathbf{y}^*(t)\|_2 \leq Ce^{-\alpha t},$$

$$f_0(\mathbf{y}(t), t) - f_0(\mathbf{y}^*(t), t) \leq m_f C^2 e^{-2\alpha t}$$

hold, where

$$C = \left(\tfrac{c^2}{m_f^2} \sum_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} f_0(\mathbf{y}(0), 0)\|_2^2 \right)^{\frac{1}{2}} < \infty,$$

for some constant $c > 0$, and where $-\alpha := \max_{\lambda \in \text{spec}(\mathbf{H})} \Re[\lambda] + \epsilon_H$, for some $\epsilon_H > 0$ sufficiently small.

Proof: See Appendix 2.2.7.

The above theorem states that the solution trajectory of the implicit function $\mathbb{G}_{\text{unc}}(\mathbf{y}^{[k]}, t) = 0$ from (2.46) converges asymptotically to the minimizer $\mathbf{y}^*(t)$ of (2.43). It remains to show that one can indeed simultaneously solve the system of implicit equations:

$$\mathbb{H}(\mathbf{y}^{[k]}, \mathbf{u}, t) := \begin{cases} \mathbb{F}(\mathbf{y}^{[k]}, \mathbf{u}) := \mathbf{u} - \alpha(\mathbf{y}^{[k]}) \\ \mathbb{G}_{\text{unc}}(\mathbf{y}^{[k]}, t) \end{cases} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{2.47}$$

for the pair $(\mathbf{y}^{(k)}, \mathbf{u}) = S(\mathbf{y}^{[k-1]}, t)$. By uniform strong convexity (see Assumption 4), the Hessian matrix $\nabla_{\mathbf{y}\mathbf{y}} f_0(\mathbf{y}, t)$ is uniformly positive definite, which allows to solve (2.47) by solving for first for $\mathbf{y}^{(k)}$ using (2.46), and subsequently solving for $\mathbf{u}$ using (2.31). The following theorem summarizes these findings.

Theorem 8 (TVO-based control for system (2.29))**.** *Let Assumption 4 hold and consider the differentially flat system* (2.29) *with the feedback controller*

$$\mathbf{u} := \alpha(\mathbf{y}, \ldots, \mathbf{y}^{(k-1)}, g_{\text{unc}}(\mathbf{y}^{[k-1]}))$$

where

$$g_{\text{unc}}(\mathbf{y}^{[k-1]}) := -\nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y},t) \Bigg[\sum_{i=0}^{k-1} a_i \nabla_{\mathbf{y}}^{(i)} f_0(\mathbf{y},t) \\ + \nabla_{\mathbf{y}t}^{(k-1)} f_0(\mathbf{y},t) + \sum_{m=1}^{k-1} \tbinom{k-1}{m} \nabla_{\mathbf{yy}}^{(m)} f_0(\mathbf{y},t) \mathbf{y}^{(k-m)} \Bigg]$$

is a solution to (2.47) *with closed-loop dynamics given by* $\mathbf{y}^{(k)} = g_{\text{unc}}(\mathbf{y}^{[k-1]})$*. Then, for any initial condition, the flat output of* (2.29) *globally asymptotically converges to the optimal solution* $\mathbf{y}^*(t)$ *of* (2.43)*.*

Notably, the above nonlinear feedback control effectively transforms the differentially flat system into an optimization algorithm (2.42) that seeks to converge to the optimal solution of the time-varying optimization problem.

2.2.4 Equality Constrained time-varying optimization framework

In this section, we consider an equality-constrained version of the time-varying optimization problem (2.8):

$$\begin{aligned} \mathbf{y}^*(t) :=& \arg\min_{\mathbf{y}\in\mathbb{R}^m} \ f_0(\mathbf{y},t) \\ &\text{s.t.} \quad \mathbf{A}(t)\mathbf{y} = b(t). \end{aligned} \tag{2.48}$$

Define the Lagrangian $L : \mathbb{R}^m \times \mathbb{R}^q \times \mathbb{R}_+ \to \mathbb{R}$ associated with the problem as

$$L(\mathbf{y}, \boldsymbol{\nu}, t) = f_0(\mathbf{y},t) + \boldsymbol{\nu}^T(\mathbf{A}(t)\mathbf{y} - b(t)) \tag{2.49}$$

and refer $\boldsymbol{\nu}_i$ as the Lagrange multiplier associated with the ith equality constraint $a_i(t)^T\mathbf{y} = b_i(t)$. The Assumption 5 on $\mathbf{A}(t)$ means that there are fewer equality constraints than variables and that the equality constraints are independent uniformly. Additionally, the time-varying optimization problem is uniformly convex (see Assumption 4), and the KKT conditions are necessary and sufficient for the points to be primal and dual optimal. That is, the optimal trajectory

$\mathbf{z}^*(t) = \mathrm{col}(\mathbf{y}^*(t), \boldsymbol{\nu}^*(t)) \in \mathbb{R}^{m+q}$ is characterized by the following KKT conditions:

$$\begin{aligned}\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t) + \mathbf{A}(t)^T \boldsymbol{\nu}^*(t) = 0 = \nabla_{\mathbf{y}} L(\mathbf{y}^*(t), \boldsymbol{\nu}^*(t), t),\\ \mathbf{A}(t)\mathbf{y}^*(t) - b(t) = 0 = \nabla_{\boldsymbol{\nu}} L(\mathbf{y}^*(t), \boldsymbol{\nu}^*(t), t),\end{aligned}$$

which is equivalent to $\nabla_{\mathbf{z}} L(\mathbf{z}^*(t), t) = 0$.

Similarly to (2.44), one can make $z(t)$ converge to the optimal primal-dual trajectory $\mathbf{z}^*(t)$ by designing a target linear system with state represented by $\mathrm{col}(\nabla_{\mathbf{z}} L(\mathbf{z}, t), \ldots, \nabla_{\mathbf{z}}^{(k-1)} L(\mathbf{z}, t))$, i.e.,

$$\begin{bmatrix} \dot{\nabla}_{\mathbf{z}} L(\mathbf{z}, t) \\ \vdots \\ \nabla_{\mathbf{z}}^{(k)} L(\mathbf{z}, t) \end{bmatrix} = \mathbf{H} \begin{bmatrix} \nabla_{\mathbf{z}} L(\mathbf{z}, t) \\ \vdots \\ \nabla_{\mathbf{z}}^{(k-1)} L(\mathbf{z}, t) \end{bmatrix}, \tag{2.50}$$

where $\mathbf{H} = \hat{\mathbf{H}} \otimes \mathbf{I}_{m+q}$ is Hurwitz. As a result of Lemma 4, differentiating the gradient $\nabla_{\mathbf{z}} L(\mathbf{z}, t)$ with respect to time $k-$times yields

$$\nabla_{\mathbf{z}}^{(k)} L(\mathbf{z}, t) = \sum_{m=0}^{k-1} \binom{k-1}{m} \nabla_{\mathbf{zz}}^{(m)} L(\mathbf{z}, t) \mathbf{z}^{(k-m)} + \nabla_{\mathbf{z}t}^{(k-1)} L(\mathbf{z}, t). \tag{2.51}$$

Notice that the KKT matrix is defined as [73]:

$$\nabla_{\mathbf{zz}} L(\mathbf{z}, t) = \begin{bmatrix} \nabla_{\mathbf{yy}} f_0(\mathbf{y}, t) & \mathbf{A}^T(t) \\ \mathbf{A}(t) & \mathbf{0}_{q \times q} \end{bmatrix}.$$

The KKT matrix is nonsingular because $\mathrm{rank}(\nabla_{\mathbf{yy}} f_0(\mathbf{y}, t)) = m$ for all $t \geq 0$ and $\mathrm{rank}(\mathbf{A}(t)) = q$ for all $t \geq 0$ by Assumption 4 and Assumption 5. This also means the optimal primal-dual pair $(\mathbf{y}^*(t), \boldsymbol{\nu}^*(t))$ is unique at each $t \geq 0$. In classical convex optimization, the bounded inverse KKT matrix assumption $\|\nabla_{\mathbf{zz}}^{-1} L(\mathbf{z}, t)\|_2 \leq K^{-1}$ plays the role of the strong convexity assumption 4 in the unconstrained setting. In the time-varying optimization setting, the following Uniform Lipschitz Continuity assumption helps us establish the uniform boundedness of the inverse KKT matrix. To that end, we make the following assumption.

Assumption 6 (Uniform Lipschitz continuity)**.** *The objective function f_0 and the inequality constraint functions f_i, $i \in [p]$, have uniformly bounded gradients with respect to*

$\mathbf{y}$, *i.e.,*

$$\|\nabla_{\mathbf{y}} f_0(\mathbf{y}, t)\|_2 \leq L, \quad \|\nabla_{\mathbf{y}} f_i(\mathbf{y}, t)\|_2 \leq L_i,$$

for constants $L, L_i > 0$, for all $\mathbf{y}$, all $t \geq 0$, and all $i \in [p]$.

Remark 1. *The Uniform Lipchitz continuity Assumption 6 could be relaxed by the uniform Lipschitz gradient assumption whenever the feasible set is bounded. Precisely, suppose that $\mathcal{K} \subset \mathbb{R}^m$ is a bounded closed convex set. Assume that,*

$$\|\nabla_{\mathbf{y}} f_0(\mathbf{y}, t) - \nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t)\|_2 \leq L\|\mathbf{y} - \mathbf{y}^*(t)\|_2 \tag{2.52}$$

for all $\mathbf{y} \in \mathcal{K}$, all $t \geq 0$ for some constant $L \geq 0$. This implies that $\|\nabla_{\mathbf{y}} f_0(\mathbf{y}, t)\| \leq L\|\mathbf{y} - \mathbf{y}^*(t)\|_2$, which is bounded by some constant depending on $\mathcal{K}$.

The following Lemma characterizes the uniform boundedness of the eigenvalues of the KKT matrix

Lemma 9 (Eigenvalues of KKT matrix). *[74, Lemma 2.1] For all $t \geq 0$, let $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m > 0$ be the eigenvalues of $\nabla_{\mathbf{yy}} f_0(\mathbf{y}, t)$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_q \geq 0$ be the singular values of $\mathbf{A}(t)$, and denote by $\Lambda(\nabla_{\mathbf{zz}} L(\mathbf{z}, t))$ the spectrum of the KKT matrix. Then*

$$\Lambda(\nabla_{\mathbf{zz}} L(\mathbf{z}, t)) \subset I = I^- \cup I^+$$

where $I^- = [\frac{1}{2}(\mu_n - \sqrt{\mu_n^2 + 4\sigma_1^2}), \frac{1}{2}(\mu_1 - \sqrt{\mu_1^2 + 4\sigma_m^2})]$ and $I^+ = [\mu_n, \frac{1}{2}(\mu_1 + \sqrt{\mu_1^2 + 4\sigma_1^2})]]$

A direct application of the above lemma establishes the uniform boundedness of the inverse KKT matrix.

Corollary 10 (Uniform boundeness of inverse KKT matrix). *Let Assumptions 4, 5 and 6 hold, then for all $t \geq 0$, we have $\|\nabla_{\mathbf{zz}}^{-1} L(\mathbf{z}, t)\|_2 \leq K^{-1}$ for some $K > 0$.*

We are now ready to extend our framework to solve (2.48). Combining (2.50) and (2.51), we use the following implicit function to define the *optimization dynamics*, when time-varying equality constraints are included:

$$\mathbb{G}_{\text{eq}}(\mathbf{z}^{[k]},t) := \sum_{m=0}^{k-1}\binom{k-1}{m}\nabla_{\mathbf{zz}}^{(m)}L(\mathbf{z},t)\mathbf{z}^{(k-m)} \\ +\nabla_{\mathbf{z}t}^{(k-1)}L(\mathbf{z},t)+\sum_{i=0}^{k-1}a_i\nabla_{\mathbf{z}}^{(i)}L(\mathbf{z},t)=0. \tag{2.53}$$

The following theorem states that if certain regularity conditions are met, the output $\mathbf{z}$, which follows the optimization dynamicss described in equation (2.53), will globally asymptotically converge to the optimal primal solution $\mathbf{y}^*(t)$ and the optimal equality constraint dual solution $\boldsymbol{\nu}^*(t)$ of (2.48).

Theorem 11 (Convergence of equality constrained optimization dynamics (2.53))**.** *Let Assumptions 4, 5 and 6 hold. Then for any initial condition, the trajectory $t \mapsto \mathbf{z}(t)$ of system $\mathbb{G}_{\text{eq}}(\mathbf{z}^{[k]},t)=0$ defined in (2.53) globally asymptotically converges to the optimal solution of $\mathbf{z}^*(t)=\operatorname{col}(\mathbf{y}^*(t),\boldsymbol{\nu}^*(t))$ of the time-varying equality constrained optimization problem (2.48). Moreover, the following bound holds*

$$\|\mathbf{z}(t)-\mathbf{z}^*(t)\|_2 \le Ce^{-\alpha t},$$

where

$$0<C=\left(\tfrac{c^2}{K^2}\textstyle\sum_{j=0}^{k-1}\|\nabla_{\mathbf{z}}^{(j)}L(\mathbf{z}(0),0)\|_2^2\right)^{\frac{1}{2}}<\infty,$$

for some constant $c>0$,$-\alpha := \max_{\lambda\in\operatorname{spec}(\mathbf{H})}\Re[\lambda]+\epsilon_H$, for some $\epsilon_H>0$ small enough.

Proof: See Appendix 2.2.7

Lastly, it remains to show that one can simultaneously solve the system of implicit equations

$$\mathbb{H}(\mathbf{z}^{[k]},\mathbf{u},t) := \begin{cases}\mathbb{F}(\mathbf{z}^{[k]},\mathbf{u}) := \mathbf{u}-\alpha_z(\mathbf{z}^{[k]}) \\ \mathbb{G}_{\text{eq}}(\mathbf{z}^{[k]},t)\end{cases} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix} \tag{2.54}$$

for the pair $(\mathbf{z}^{(k)},\mathbf{u})=S(\mathbf{z}^{[k-1]},t)$, where $\alpha_z(\mathbf{z}^{[k]}) := \alpha(\mathbf{y}^{[k]})$. According to Corollary 10, the KKT matrix $\|\nabla_{\mathbf{zz}}^{-1}L(\mathbf{z},t)\|_2 \le K^{-1}$ is uniformly bounded, which allows to solve (2.54) by solving for first for $\mathbf{z}^{(k)}$ using (2.53) and subsequently solving for $\mathbf{u}$ using (2.31). The following theorem summarizes these findings.

Theorem 12 (Equality constrained TVO-based control for system (2.29))**.** *Let Assumption 4, 5 and 6 hold and consider the differentially flat system (2.29) with the feedback controller*

$$\mathbf{u} = \alpha_z(\mathbf{z}, \dots, \mathbf{z}^{(k-1)}, g_{\text{eq}}(\mathbf{z}^{[k-1]}))$$

where

$$\begin{aligned} g_{\text{eq}}(\mathbf{z}^{[k-1]}) := -\nabla_{\mathbf{zz}}^{-1} L(\mathbf{z}, t) \Bigg[\sum_{i=0}^{k-1} a_i \nabla_{\mathbf{z}}^{(i)} L(\mathbf{z}, t) \\ + \nabla_{\mathbf{z}t}^{(k-1)} L(\mathbf{z}, t) + \sum_{m=1}^{k-1} \binom{k-1}{m} \nabla_{\mathbf{yy}}^{(m)} L(\mathbf{z}, t) \mathbf{z}^{(k-m)} \Bigg], \end{aligned}$$

is a solution to (2.54) with closed-loop dynamics given by $\mathbf{z}^{(k)} = g_{\text{eq}}(\mathbf{z}^{[k-1]})$. Then, for any initial condition, the flat output of (2.29) globally asymptotically converges to the optimal solution $\mathbf{y}^*(t)$ of (2.48).

2.2.5 Inequality Constrained time-varying optimization framework

In Section 2.2.4, we showed how to incorporate equality constraints by Lagrangian duality in our framework. In this section, we consider a time-varying optimization problem where only inequality constraints are included:

$$\begin{aligned} \mathbf{y}^*(t) =& \arg \min_{\mathbf{y} \in \mathbb{R}^m} f_0(\mathbf{y}, t) \\ & \text{s.t.} \quad f_i(\mathbf{y}, t) \leq 0, \quad i \in [p]. \end{aligned} \tag{2.55}$$

Define the Lagrangian $L : \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}_+ \to \mathbb{R}$ associated with the problem (2.55) as

$$L(\mathbf{y}, \boldsymbol{\lambda}, t) = f_0(\mathbf{y}, t) + \sum_{i=1}^{p} \lambda_i f_i(\mathbf{y}, t)$$

and refer λ_i as the Lagrange multiplier associated with the ith inequality constraint $f_i(\mathbf{y}, t) \leq 0$. Additionally, the time-varying optimization problem is uniformly strongly convex (see Assumption 4), and the KKT conditions are necessary and sufficient for optimality [73, 75]. Precisely, for any $t \geq 0$, we have the following KKT conditions, where the primal feasibility conditions automatically hold for global minimum $\mathbf{y}^*(t)$ and are neglected:

$$\begin{aligned}
&\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t) + \sum_{i=1}^{p} \lambda_i^*(t) \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)) = 0 \\
&\lambda_i^*(t) \geq 0, \qquad\qquad i \in [p], \\
&\lambda_i^*(t) f_i(\mathbf{y}^*(t), t)) = 0, \quad i \in [p].
\end{aligned} \tag{2.56}$$

Motivated by [32, 73], in this section, we will use a particular interior-point algorithm to incorporate inequality constraints as in (2.55), the barrier method.

The goal of the barrier method is to approximately formulate the time-varying inequality-constrained problem as a time-varying unconstrained problem. Towards this goal, the first step is to rewrite the problem (2.55), making the inequality constraints implicit in the objective function:

$$\mathbf{y}^*(t) = \arg \min_{\mathbf{y} \in \mathbb{R}^m} f_0(\mathbf{y}, t) + \sum_{i=1}^{p} \mathbb{I}_-(f_i(\mathbf{y}, t)), \tag{2.57}$$

where $\mathbb{I}_- : \mathbb{R} \to \mathbb{R}$ is the indicator function for the nonpositive reals: $\mathbb{I}_-(u) = 0$ for $u \leq 0$ and $\mathbb{I}_-(u) = +\infty$ for $u > 0$. To approximate the indicator function $\mathbb{I}_-$, we use a continuously differentiable logarithmic barrier function given by: $\hat{\mathbb{I}}_-(u, t) = -\frac{1}{c(t)} \log(-u)$, where $c(t) > 0$ is a parameter that ensures the accuracy of the approximation improves as t increases. Therefore, the logarithmic barrier coefficient $c(t)$ is required to be monotonic increasing, asymptotically converging to infinity, and bounded in finite time. A convenient choice would be

$$c(t) = c_0 e^{\alpha_c t} \tag{2.58}$$

with $\alpha_c, c_0 > 0$.

Substituting $\mathbb{I}_-$ with $\hat{\mathbb{I}}_-$ gives the approximation of (2.57)

$$\Phi(\mathbf{y}, t) = f_0(\mathbf{y}, t) + \sum_{i=1}^{p} -\frac{1}{c(t)} \log(-f_i(\mathbf{y}, t)). \tag{2.59}$$

One of the limitations of (2.59), is that it requires a starting point that satisfies all the constraints. If such a point is not known a priori, a preliminary step called *phase I* is used to find a feasible point. In this phase, a time-varying slack variable denoted as $s(t)$ is introduced to ensure that the constraints are satisfied and overcome any limitations related to the initial point. We thus consider in such case the perturbed approximation of (2.59):

$$\hat{\Phi}(\mathbf{y}, t) := f_0(\mathbf{y}, t) - \frac{1}{c(t)} \sum_{i=1}^{p} \log(s(t) - f_i(\mathbf{y}, t)), \tag{2.60}$$

where a good choice of $s(t)$ is $s_0 e^{-\alpha_s t}$ and for any initial condition $\mathbf{y}(0)$, s_0 can be chosen large enough such that:

$$s_0 = \begin{cases} 0 & \textbf{if } \max_i f_i(\mathbf{y}(0), 0) \leq 0 \\ \max_i f_i(\mathbf{y}(0), 0) + \epsilon_s & \textbf{if } \max_i f_i(\mathbf{y}(0), 0) > 0 \end{cases} \tag{2.61}$$

for some $\epsilon_s > 0$. By incorporating the inequality constraints into the objective function with logarithmic barrier functions as in (2.60), the above constrained time-varying optimization problem can be approximated by:

$$\hat{\mathbf{y}}^*(t) := \arg \min_{\mathbf{y} \in \mathbb{R}^m} \hat{\Phi}(\mathbf{y}, t). \tag{2.62}$$

The following Lemma [32, Lemma 1] provides an upper bound on the duality gap associated with $\mathbf{y}^*(t)$ and the Lagrange multiplier $\boldsymbol{\lambda}^*(t)$. It also confirms that with proper choices of $s(t), c(t)$, $\hat{\mathbf{y}}^*(t)$ converges to the optimal solution $\mathbf{y}^*(t)$ as $t \to +\infty$, provided that the Lagrange multipliers $\lambda^*(t)$ remain bounded.

Lemma 13 (Approximation error[32, Lemma 1])**.** *Consider the inequality-constrained time-varying optimization problem* (2.55) *and* $\mathbf{y}^*(t)$ *be the optimal solution. Let* $\boldsymbol{\lambda}^*(t)$ *be the Lagrange multiplier associated with inequality constraints and* $\hat{\mathbf{y}}^*(t)$ *be the optimal* *solution of the perturbed approximation* (2.60). *Under Assumptions* 4 *and* 5, *the following inequality holds:*

$$|f_0(\hat{\mathbf{y}}^*(t),t) - f_0(\mathbf{y}^*(t),t)| \leq \frac{p}{c(t)} + \sum_{i=1}^{p} \lambda_i^*(t)s(t) \tag{2.63}$$

Lemma 13 not only provides a uniform bound for the optimality error, but it also suggests appropriate selections of $s(t)$, $c(t)$. In particular, choosing $c(t)$ as in (2.58) ensures that the first term in (2.63) goes to zero. Thus, if one were to further guarantee that $\sum_{i=1}^{p} \lambda_i^*(t)s(t) \to 0$ as $t \to +\infty$, then this would readily imply that the optimal solution $\hat{\mathbf{y}}^*(t)$ of (2.62) converges to the optimal solution $\mathbf{y}^*(t)$ of (2.8). Since we are interested in asymptotic convergence, roughly speaking, this requires that the optimization problem does not have exponentially unbounded optimal dual variables. We will prove next that under sufficient regularity conditions, one can provide a uniform constant bound on the value of the multipliers $\boldsymbol{\lambda}^*(t)$, thus making the approximation $\hat{\mathbf{y}}^*(t)$ of (2.62) converges asymptotically to $\mathbf{y}^*(t)$. One notable contribution of this paper is the direct establishment of regularity conditions that ensure the uniform boundedness of the Lagrange multipliers. This contributes to the literature of time-varying optimization [32, 33], wherein asymptotic boundedness of multipliers is assumed. For a static nonconvex optimization problem, *Mangasarian-Fromowitz constraint qualification* (MFCQ) is shown to be necessary and sufficient to have the set of Lagrange multipliers being nonempty and bounded [76]. For a general time-varying optimization problem (2.8), where both equality and inequality constraints are considered, the following Lemma provides a sufficient condition for the uniform boundedness of the set of Lagrange multipliers.

Lemma 14. *(Uniform boundedness of Lagrange multipliers) Let* $\mathbf{y}^*(t)$ *be the optimal solution of* (2.55), *and suppose that Assumptions* 4, 5, *and* 6 *are satisfied. For all* $t \geq 0$, *the set of Lagrange multipliers* $\boldsymbol{\lambda}^*(t) \in \mathbb{R}^p$ *satisfying the KKT conditions* (2.56) *is nonempty* *and uniformly bounded*

$$\|\boldsymbol{\lambda}^*(t)\|_1 \leq \frac{Ld}{\epsilon},$$

where $\|\cdot\|_1$ *denotes the* l_1 *vector norm.*

Proof: See Appendix 2.2.7.

A direct application of lemmas 13 and 14, shows the desired exponential decrease on the approximation error. Therefore, we consider the following target system as a natural extension of (2.44) when inequality constraints are included

$$\begin{bmatrix} \dot{\nabla}_{\mathbf{y}}\hat{\Phi}(\mathbf{y},t) \\ \vdots \\ \nabla_{\mathbf{y}}^{(k)}\hat{\Phi}(\mathbf{y},t) \end{bmatrix} = \mathbf{H} \begin{bmatrix} \nabla_{\mathbf{y}}\hat{\Phi}(\mathbf{y},t) \\ \vdots \\ \nabla_{\mathbf{y}}^{(k-1)}\hat{\Phi}(y,t) \end{bmatrix}, \tag{2.64}$$

where $\mathbf{H} = \hat{\mathbf{H}} \otimes \mathbf{I}_m$ being Hurwitz.

Analogously, combining (2.64) and Lemma 4, we can use the following implicit function to define the *optimization dynamics*, when inequality constraints are included:

$$\begin{aligned} \mathbb{G}_{\text{ineq}}(\mathbf{y}^{[k]},t) :=& \sum_{m=0}^{k-1} \binom{k-1}{m} \nabla_{\mathbf{yy}}^{(m)}\hat{\Phi}(\mathbf{y},t)\mathbf{y}^{(k-m)} \\ &+\nabla_{\mathbf{y}t}^{(k-1)}\hat{\Phi}(\mathbf{y},t) + \sum_{i=0}^{k-1} a_i \nabla_{\mathbf{y}}^{(i)}\hat{\Phi}(\mathbf{y},t) = 0. \end{aligned} \tag{2.65}$$

The following theorem shows that, under sufficient regularity conditions, the output $\mathbf{y}(t)$ satisfying the optimization dynamics (2.65) globally converges to the minimizer $\mathbf{y}^*(t)$ of the time-varying inequality constrained optimization problem (2.55).

Theorem 15 (Convergence of inequality constrained optimization dyanmics (2.65))**.** *Let Assumptions 4, 5 and 6 hold, with* $c(t)$ *given by* (2.58)*, and* $s(t) = s_0 e^{-\alpha_s t}$ *with* s_0 *as in* (2.61)*. Then for any initial condition, the trajectory* $t \to \mathbf{y}(t)$ *of system* $\mathbb{G}_{\text{ineq}}(\mathbf{y}^{[k]},t) = 0$ *defined in* (2.65) *will globally asymptotically converges to the optimal solution of* $\mathbf{y}^*(t)$

of the time-varying inequality constrained optimization problem (2.55)*. Moreover, the following bounds hold*

$$\|\mathbf{y}(t) - \hat{\mathbf{y}}^*(t)\|_2 \leq Ce^{-\alpha t},$$
$$|f_0(\mathbf{y}(t), t) - f_0(\mathbf{y}^*(t), t)| \leq LCe^{-\alpha t} + pc_0 e^{-\alpha_c t} + \frac{Lds_0}{\epsilon} e^{-\alpha_s t},$$

where

$$0 < C = \left(\tfrac{c^2}{m_f^2} \sum\nolimits_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} \hat{\Phi}(\mathbf{y}(0), 0)\|_2^2)\right)^{\frac{1}{2}} < \infty,$$

for some constant $c > 0$ *,*$-\alpha := \max_{\lambda \in \mathrm{spec}(\mathbf{H})} \Re[\lambda] + \epsilon_H$*, for some* $\epsilon_H > 0$ *small enough.*

Proof: See Appendix 2.2.7*.*

The above theorem states that the solution trajectory of the implicit function $\mathbb{G}_{\text{ineq}}(\mathbf{y}^{[k]}, t)$ from (2.65) converges to the minimizer $\mathbf{y}^*(t)$ of (2.55). It remains to show that one can simultaneously solve the system of implicit equations

$$\mathbb{H}(\mathbf{y}^{[k]}, \mathbf{u}, t) := \begin{cases} \mathbb{F}(\mathbf{y}^{[k]}, \mathbf{u}) := \mathbf{u} - \alpha(\mathbf{y}^{[k]}) \\ \mathbb{G}_{\text{ineq}}(\mathbf{y}^{[k]}, t) \end{cases} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{2.66}$$

for the pair $(\mathbf{y}^{(k)}, \mathbf{u}) = S(\mathbf{z}^{[k-1]}, t)$. Since, from the proof of Theorem 15, we have $\|\nabla_{\mathbf{yy}}^{-1} \hat{\Phi}(\mathbf{y}, t)\|_2 \leq m_f^{-1}$ (see Appendix 2.2.7), we can solve (2.66) by solving for first for $\mathbf{y}^{(k)}$ using (2.65) and subsequently solving for $\mathbf{u}$ using (2.31). The following theorem summarizes these findings.

Theorem 16 (Inequality constrained TVO-based control for system (2.29))**.** *Let Assumptions 4, 5 and 6 hold, with* $c(t)$ *given by* (2.58)*, and* $s(t) = s_0 e^{-\alpha_s t}$ *with* s_0 *as in* (2.61)*. Consider the differentially flat system* (2.29) *with the feedback controller*

$$\mathbf{u} := \alpha(\mathbf{y}, \ldots, \mathbf{y}^{(k-1)}, g_{\text{ineq}}(\mathbf{y}^{[k-1]}))$$

where

$$g_{\text{ineq}}(\mathbf{y}^{[k-1]}) := -\nabla_{\mathbf{yy}}^{-1}\hat{\Phi}(\mathbf{y},t)\left[\sum_{i=0}^{k-1} a_i \nabla_{\mathbf{y}}^{(i)}\hat{\Phi}(\mathbf{y},t)\right.$$
$$\left. + \nabla_{\mathbf{y}t}^{(k-1)}\hat{\Phi}(\mathbf{y},t) + \sum_{m=1}^{k-1}\tbinom{k-1}{m}\nabla_{\mathbf{yy}}^{(m)}\hat{\Phi}(\mathbf{y},t)\mathbf{y}^{(k-m)}\right]$$

is a solution to (2.66) *with closed-loop dynamics given by* $\mathbf{y}^{(k)} = g_{\text{ineq}}(\mathbf{y}^{[k-1]})$*. Then, for any initial condition, the flat output of* (2.29) *globally asymptotically converges to the optimal solution* $\mathbf{y}^*(t)$ *of* (2.55).

Notably, the above nonlinear feedback control effectively transforms the differentially flat system into an optimization algorithm similar to (2.42) that seeks to converge to the optimal solution of the time-varying optimization problem:

$$\begin{aligned}&\dot{\mathbf{z}} = \mathbf{H}\mathbf{z},\\ &\mathbf{z} = (\nabla_{\mathbf{y}}\hat{\Phi}(\mathbf{y},t), ..., \nabla_{\mathbf{y}}^{k-1}\hat{\Phi}(\mathbf{y},t))^T.\end{aligned} \tag{2.67}$$

2.2.6 Numerical experiments

In this section, we use two numerical examples arising in multi-robot coordination to illustrate the effectiveness of our solution approach. As our time-varying feedback optimization framework automatically guarantees asymptotic satisfaction of time-varying equality and inequality constraints, we apply the method for the specification of formation constraints (Section 2.2.6) and to enforce collision avoidance (Section 2.2.6).

Multi-robot Navigation with formation constraints

In this numerical example, two WMRs (2.39) are required to track two moving objects respectively, but the maximum distance between two agents is limited (e.g., due to communication or formation constraints). Let $\mathbf{y}_1(t), \mathbf{y}_2(t) \in \mathbb{R}^2$ denote the position of each WMR, with $\mathbf{y}_1^d(t), \mathbf{y}_2^d(t)$ denoting the positions of the two moving

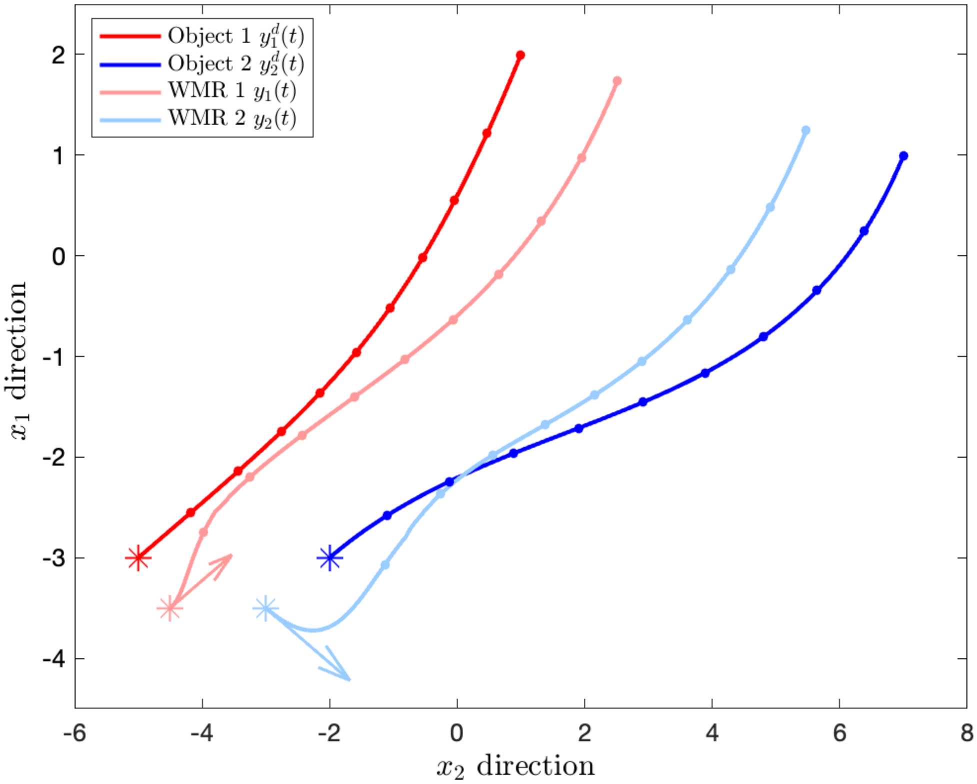

Figure 2-4. Trajectories of two moving objects $\mathbf{y}_1^d(t), \mathbf{y}_2^d(t)$ (solid) and two WMRs $\mathbf{y}_1, \mathbf{y}_2$ (dashed). WMRs succeed in tracking two moving objects while satisfying distance constraints between them.

objects. To model the above objectives, consider the time-varying optimization problem

$$\begin{aligned} &\min_{\mathbf{y}_1,\mathbf{y}_2} \|\mathbf{y}_1 - \mathbf{y}_1^d(t)\|_2^2 + \|\mathbf{y}_2 - \mathbf{y}_2^d(t)\|_2^2 \\ &s.t. \;\; \|\mathbf{y}_1 - \mathbf{y}_2\|_2 \leq d(t), \end{aligned} \tag{2.68}$$

where $d(t)$ denotes the maximum (Euclidean) separation allowed between the two robots at time t. The trajectories of the moving objects, $\mathbf{y}_1^d(t)$ and $\mathbf{y}_2^d(t)$, are designed using a time-parametric representation (Section 2.4 [70]). More specifically, we parametrize the trajectories $\mathbf{y}_1^d(t)$ and $\mathbf{y}_2^d(t)$ by

$$\mathbf{y}_i^d(t) = \sum_{j=1}^{N} \mathbf{A}_{ij} \lambda_j(t), \tag{2.69}$$

where the $\lambda_j(t) = t^j$ are the standard polynomial basis functions and $\mathbf{A}_{ij}$ can be calculated from the initialization.

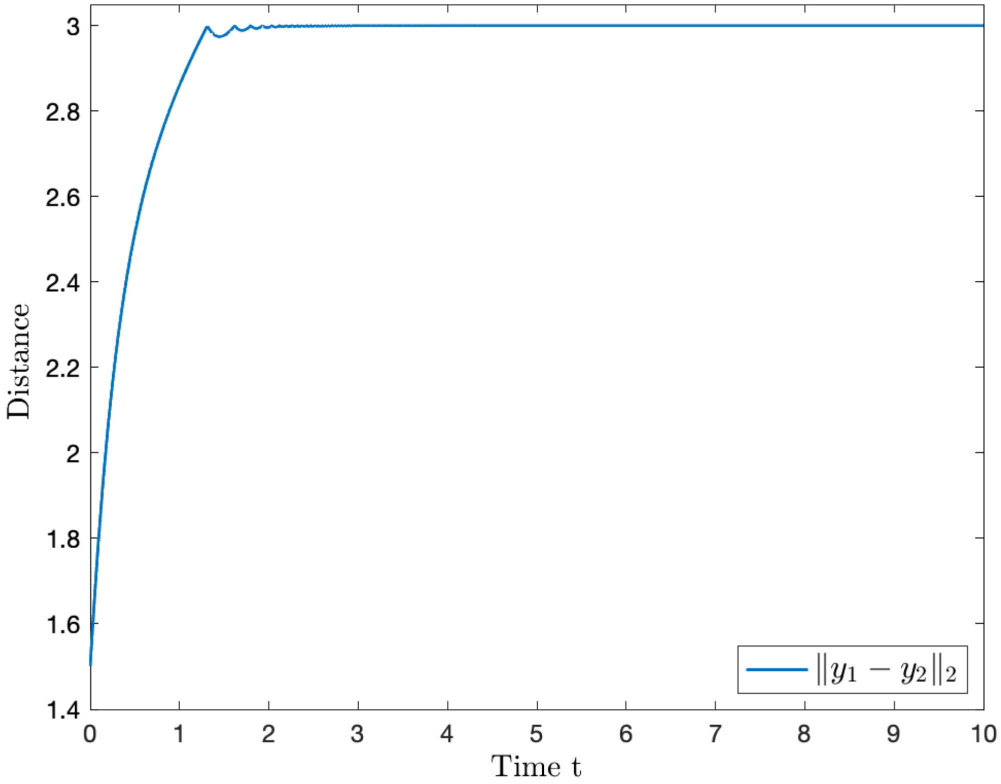

Figure 2-5. Euclidean distance between two WMRs $\|\mathbf{y}_1 - \mathbf{y}_2\|_2$. The maximum distance constraint is satisfied for all $t \leq T$.

The simulation results are illustrated in Figure 2-4 and Figure 2-5. The solid black and blue curves in Figure 2-4 represent two random moving objects trajectories which are generated using the time parametric representation (2.69). More specifically, the randomly picked starting states $\mathbf{x}$ (using asterisk) are $[-5; -3; 0.5]$ and $[-2; -3; 0.5]$. As to the robots, they are positioned around the starting position with random perturbations, which are $[-4.5, -3.5; 0.5]$ and $[-3, -3.5; -0.5]$ (using asterisk). The two WMRs' trajectories are represented using dashed green and red curves. And the arrows represent the positional velocity vector at each position $\dot{\mathbf{y}}_i$. The total simulation time $T = 10s$ and the maximum distance allowed between two robots $d(t) = 3$. For calculation simplicity, we choose the logarithmic barrier coefficient $c_0 = 1$. For this implementation, the differential equations are solved using MATLAB standard ODE solver based on an explicit Runge-Kutta (4,5) formula (ode45). In Figure 2-4, two robots starting from arbitrary positions succeed in tracking the minimizers of (2.68), i.e., two WMRs track two moving objects respectively. Furthermore, in Figure 2-5 we plot the (euclidean) distance between two WMRs $\|\mathbf{y}_1 - \mathbf{y}_2\|_2^2$ and we conclude that the time-varying inequality constraints are not violated using our solution approach, i.e., $\|\mathbf{y}_1 - \mathbf{y}_2\|_2^2 \leq 3$.

Robot tracking and obstacle avoidance

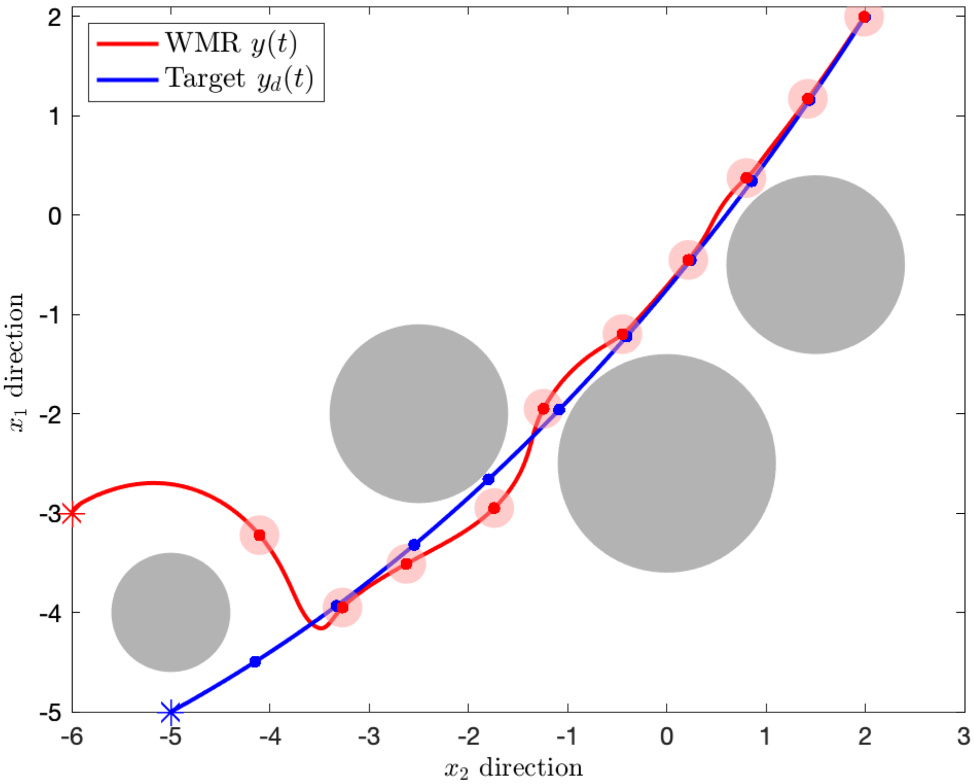

Figure 2-6. Trajectory of the WMR (red curve and red disks, starting at the red asterisk) tracking a moving target (blue curve, starting at the blue asterisk) while avoiding the obstacles (grey disks), where the dots represent their locations at different t.

In this section, we aim to solve the problem of navigating a disk-shaped wheeled mobile robot (WMR) to track a moving target without colliding with spherical obstacles in the environment. Fazlyab *et al.* [32] formulate the robot navigation problem, whose dynamic is an integrator (2.32), via a time-varying convex optimization problem using the idea of *projected goal* [77]. We first introduce some definitions and then show how to generalize the results of [32] in our framework.

Consider a closed and convex *workspace* $\mathcal{W} \subset \mathbb{R}^2$, which is populated with m non-intersecting spherical obstacles, where the center and radius of the ith obstacle are denoted by $\mathbf{y}_i \in \mathcal{W}$ and $r_i > 0$, respectively. Suppose the wheeled mobile robot (WMR) of radius $r > 0$ is defined as in (2.39), where the flat output, namely the position vector of the center of mass of the WMR, is given by $\mathbf{y}_c = (x_1, x_2)$. We define the *free space*, denoted by $\mathcal{F}$, as the set of configurations in the workspace in which the robot does not collide with any obstacle, i.e.,

$$\mathcal{F} = \{\mathbf{y} \in \mathcal{W} : \bar{B}(\mathbf{y}, r) \subseteq \mathcal{W} \setminus \cup_{i=1}^{m} B(\mathbf{y}_i, r_i)\} \tag{2.70}$$

where $B(\mathbf{y}, r)$ is the 2-dimensional open ball centered at $\mathbf{y}$ with radius r, and $\bar{B}(\mathbf{y}, r)$ denotes its closure. Given the moving target $\mathbf{y}^d(t) \in \mathcal{F}$ for all $t \geq 0$, we aim to solve for the control input $\mathbf{u}(t)$ such that $\mathbf{y}_c(t) \in \mathcal{F}$ for all $t \geq 0$ with initialization $\mathbf{y}_c(0) \in \mathcal{F}$. Moreover, we want to have $\lim_{t\to\infty} \mathbf{y}_c(t) = \mathbf{y}^d(t)$, i.e., global asymptotic convergence.

In [32], Fazlyab *et al.* reformulated this robot navigation problem in a time-varying optimization framework. The central concept *projected goal*, inspired by [77], involves the consistent calculation of the destination point's projection $\mathbf{y}^d(t)$ onto a safe area around the robot's center of mass, void of obstacles. The *collision-free local workspace* $\mathcal{LF}$ around current position $\mathbf{y}_c$ is defined as

$$\mathcal{LF}(\mathbf{y}_c) = \{\mathbf{y} \in \mathcal{W} : a_i(\mathbf{y}_c)^T \mathbf{y} - b_i(\mathbf{y}_c) \leq 0, i \in [m]\},$$

where

$$a_i(\mathbf{y}_c) = \mathbf{y}_i - \mathbf{y}_c, \qquad \theta_i(\mathbf{y}_c) = \frac{1}{2} - \frac{r_i^2 - r^2}{2\|\mathbf{y}_i - \mathbf{y}_c\|^2},$$
$$b_i(\mathbf{y}_c) = (\mathbf{y}_i - \mathbf{y}_c)^T \left(\theta_i \mathbf{y}_i + (1-\theta_i)\mathbf{y}_c + r\frac{\mathbf{y}_c - \mathbf{y}_i}{\|\mathbf{y}_c - \mathbf{y}_i\|}\right).$$

We denote $\mathbf{y}^*(t)$ as the orthogonal projection of the desired configuration $\mathbf{y}^d(t)$ onto the collision-free local workspace $\mathcal{LF}(\mathbf{y}_c)$, which can be defined as the solution of the following optimization problem:

$$\mathbf{y}^*(t) := \arg\min_{\mathbf{y}\in\mathbb{R}^2} \frac{1}{2}\|\mathbf{y} - \mathbf{y}^d(t)\|^2$$
$$\text{s.t.} \;\; a_i(\mathbf{y}_c)^T\mathbf{y} - b_i(\mathbf{y}_c) \le 0, \;\; i \in [m].$$

In this scenario, where the WMR is intended to track a moving target, this method from [77, 32] does not offer any theoretical guarantees. However, the following simulation results illustrate how our solution approach could be used to solve the navigation problem. In Figure 2-6, the red and blue curves represent the real-time trajectories of the WMR $\mathbf{y}(t)$ and moving target $\mathbf{y}_d(t)$ respectively. Likewise, the randomly moving target trajectory is generated using the time parametric representation (2.69). The randomly picked starting states (using asterisk) are $[-6; -3; 20]$ and $[-5; -5; 0.5]$ for the WMR and moving target. The black disks represent four random nonintersecting spherical obstacles and the red disks represent the robot configurations at each time instant (the WMR radius $r = 0.2$). For this implementation, the ODEs are also solved using MATLAB ode45. We observe the robot succeeds in tracking the moving target while avoiding the spherical obstacles.

2.2.7 Appendix

Proof of Theorem 7

According to Lemma 4, the trajectory $\mathbf{y}(t)$ of system $\mathbb{G}(\bar{\mathbf{y}}^{(k)}, t) = 0$ (2.46) satisfy the optimization dynamics as in (2.44), with $\mathbf{H}$ being the designed Hurwitz matrix.

And the solution to ODE system (2.44) is:

$$\begin{bmatrix} \nabla_{\mathbf{y}} f_0(\mathbf{y},t) \\ \vdots \\ \nabla_{\mathbf{y}}^{(k-1)} f_0(\mathbf{y},t) \end{bmatrix} = e^{\mathbf{H}t} \begin{bmatrix} \nabla_{\mathbf{y}} f_0(\mathbf{y}(0),0) \\ \vdots \\ \nabla_{\mathbf{y}}^{(k-1)} f_0(\mathbf{y}(0),0) \end{bmatrix}$$

where $\mathbf{y}(0) \in R^m$ is the initial point. By taking the Euclidean norms of both sides and applying Lemma 3 we obtain

$$\sum_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} f_0(\mathbf{y}(t),t)\|_2^2 \leq c^2 e^{-2\alpha t} \sum_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} f_0(\mathbf{y}(0),0)\|_2^2 \tag{2.71}$$

for some constant $c > 0$, $-\alpha := \max_{\lambda \in \text{spec}(\mathbf{H})} \Re[\lambda] + \epsilon_H$ for some $\epsilon_H > 0$ small enough.

Next, we use the mean-value theorem to expand $\nabla_{\mathbf{y}} f_0(\mathbf{y},t)$ with respect to $\mathbf{y}$ as follows, where $\boldsymbol{\eta}(\boldsymbol{t})$ is a convex combination of $\mathbf{y}(t)$ and $\mathbf{y}^*(t)$. Additionally using the fact that $\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t),t) = 0$ for all $t \geq 0$, we obtain:

$$\mathbf{y}(t) - \mathbf{y}^*(t) = \nabla_{\mathbf{yy}}^{-1} f_0(\boldsymbol{\eta}(t),t) \nabla_{\mathbf{y}} f_0(\mathbf{y}(t),t).$$

It follows from Assumption 4, that $\|\nabla_{\mathbf{yy}}^{-1} f_0(\mathbf{y},t)\|_2 \leq m_f^{-1}$. Taking the norm on both sides together with equation (2.71) we have:

$$\begin{aligned} &\|\mathbf{y}(t) - \mathbf{y}^*(t)\|_2 \leq C e^{-\alpha t}, \\ &0 \leq C = \left(\tfrac{c^2}{m_f^2} \sum_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} f_0(\mathbf{y}(0),0)\|_2^2 \right)^{\frac{1}{2}} < \infty. \end{aligned}$$

On the other hand, convexity of $f_0(\mathbf{y},t)$ implies that for each $t \geq 0$

$$0 \leq f_0(\mathbf{y}(t),t) - f_0(\mathbf{y}^*(t),t) \leq \nabla_{\mathbf{y}} f_0(\mathbf{y}(t),t)^T (\mathbf{y}(t) - \mathbf{y}^*(t))$$

By applying Cauchy-Swhartz inequality on the right-hand side we obtain;

$$0 \leq f_0(\mathbf{y}(t),t) - f_0(\mathbf{y}^*(t),t) \leq m_f C^2 e^{-2\alpha t}$$

which completes the proof.

Proof of Theorem 11

The structure of proof is similar to the proof of Theorem 7. According to Lemma 4, the trajectory $\mathbf{z}(t)$ of system (2.53) satisfy the optimization dynamics as in (2.50), with $\mathbf{H}$ being the designed Hurwitz matrix. Similarly, the solution to this ODE satisfies the following inequality:

$$\sum_{j=0}^{k-1} \|\nabla_{\mathbf{z}}^{(j)} L(\mathbf{z}(t), t)\|_2^2 \leq c^2 e^{-2\alpha t} \sum_{j=0}^{k-1} \|\nabla_{\mathbf{z}}^{(j)} L(\mathbf{z}(0), 0)\|_2^2$$

for some constant $c > 0$, $-\alpha := \max_{\lambda \in \text{spec}(\mathbf{H})} \Re[\lambda] + \epsilon_H$ for some $\epsilon_H > 0$ small enough. Next, using the mean-value theorem to expand $\nabla_{\mathbf{z}} L(\mathbf{z}(t), t)$, where $\boldsymbol{\eta}(t)$ is a convex combination of $\mathbf{z}(t)$ and $\mathbf{z}^*(t)$ yields:

$$\mathbf{z}(t) - \mathbf{z}^*(t) = \nabla_{\mathbf{zz}}^{-1} L(\boldsymbol{\eta}(t), t) \nabla_{\mathbf{z}} L(\mathbf{z}(t), t).$$

It follows from Corollary 10 $that \|\nabla_{\mathbf{zz}}^{-1} L(\mathbf{z}, t)\|_2 \leq K^{-1}$ for some $K > 0$ and therefore,

$$\begin{aligned} &\|\mathbf{z}(t) - \hat{\mathbf{z}}^*(t)\|_2 \leq C e^{-\alpha t}, \\ &0 < C = \left(\tfrac{c^2}{K^2} \sum_{j=0}^{k-1} \|\nabla_{\mathbf{z}}^{(j)} L(\mathbf{z}(0), 0)\|_2^2) \right)^{\frac{1}{2}} < \infty. \end{aligned}$$

Proof of Lemma 14

The proof follows from [76] and considers a time-varying inequality constrained optimization problem. For all $t \geq 0$, we assume that uniform MFCQ holds at $\mathbf{y}^*(t)$ (see Assumption 5). For any $\bar{\mathbf{d}}(t) \in \mathbb{R}^m$ given by uniform MFCQ, define a point $\bar{\mathbf{y}}(s, t)$ sufficiently close to $\mathbf{y}^*(t)$ by:

$$\bar{\mathbf{y}}(s, t) = \mathbf{y}^*(t) + s\bar{\mathbf{d}}(t).$$

For all active inequality constraint functions, that is $i \in \mathbb{I}(\mathbf{y}^*(t))$, we apply Taylor's theorem:

$$\begin{aligned} f_i(\bar{\mathbf{y}}(s,t),t) &= f_i(\mathbf{y}^*(t) + s\bar{\mathbf{d}}(t),t) \\ &= f_i(\mathbf{y}^*(t),t) + \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t),t)^T s\bar{\mathbf{d}}(t) \\ &\quad + \mathbf{R}(\mathbf{y}^*(t), s\bar{\mathbf{d}}(t)) \\ &= s\nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t),t)^T \bar{\mathbf{d}}(t) + \mathbf{R}(\mathbf{y}^*(t), s\bar{\mathbf{d}}(t)) \end{aligned}$$

, where $\mathbf{R}(\mathbf{y}^*(t), s\bar{\mathbf{d}}(t))$ is the remainder satisfying

$$\frac{\mathbf{R}(\mathbf{y}^*(t), s\bar{\mathbf{d}}(t))}{\|s\bar{\mathbf{d}}(t)\|} \to 0 \quad as \quad s\bar{\mathbf{d}}(t) \to 0.$$

From part 1) of uniform MFCQ it follows immediately that for s sufficiently small, $\bar{\mathbf{y}}(s,t)$ is feasible for (2.8). Thus, for s sufficiently small,

$$f_0(\mathbf{y}^*(t),t) = f_0(\bar{\mathbf{y}}(0,t),t) \leq f_0(\bar{\mathbf{y}}(s,t),t)$$

and

$$\begin{aligned} \nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t),t)^T \bar{\mathbf{d}}(t) = \nabla_{\mathbf{y}} f_0(\bar{\mathbf{y}}(0),t)^T \bar{\mathbf{d}}(t) \geq 0 \implies \\ -\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t),t)^T \bar{\mathbf{d}}(t) \leq 0. \end{aligned}$$

Next, we consider the linear program:

$$\begin{aligned} \max_{\mathbf{d}} \quad & -\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t),t)^T \mathbf{d} \\ s.t. \quad & \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t),t)^T \mathbf{d} \leq -1, \quad i \in \mathbb{I}(\mathbf{y}^*(t)) \\ & \mathbf{d} \quad \text{unrestricted.} \end{aligned}$$

Any optimization variable $\mathbf{d}$ satisfying these constraint functions also satisfy the uniform MFCQ, and the value of the objective function is upper bounded by $-\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t),t)^T \mathbf{d} \leq 0$ based on previous analysis. Besides, the feasibility of this linear program is also guaranteed by Assumption 5 (uniform MFCQ), since there exists $\|\bar{\mathbf{d}}(t)\|_2 \leq d$ for some constant $d > 0$, and a constant $\epsilon > 0$ such that,

$$
\begin{aligned}
\nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)^T \bar{\mathbf{d}}(t) \leq -\epsilon, \quad i \in \mathbb{I}(\hat{\mathbf{y}}(t)) \implies \\
\nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)^T \frac{\bar{\mathbf{d}}(t)}{\epsilon} \leq -1, \quad i \in \mathbb{I}(\hat{\mathbf{y}}(t)),
\end{aligned}
$$

which means that a feasible $\mathbf{d}$ is given by $\frac{\bar{\mathbf{d}}(t)}{\epsilon}$. Furthermore, using Assumption 6 and Cauchy-Schwarz inequality we have $-\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t)^T \mathbf{d} \geq -\frac{Ld}{\epsilon}$. Together, we showed that this linear program is feasible and bounded, with $-\frac{Ld}{\epsilon} \leq -\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t)^T \mathbf{d} \leq 0$ holds for all $t \geq 0$. Its dual problem:

$$
\begin{aligned}
&\min_{\boldsymbol{\lambda}} \quad \sum_{i \in \mathbb{I}(\mathbf{y}^*(t))} -\lambda_i \\
&s.t. \quad \lambda_i \geq 0, \quad i \in \mathbb{I}(\mathbf{y}^*(t)) \\
&\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t) + \sum_{i=1}^{p} \lambda_i \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t) = 0
\end{aligned}
$$

is also feasible and bounded since strong duality holds. That is, the set of feasible $\boldsymbol{\lambda}$ vectors is nonempty and bounded

$$
0 \leq \sum_{i \in \mathbb{I}(\mathbf{y}^*(t))} \lambda_i^*(t) \leq \frac{Ld}{\epsilon}
$$

for all $t \geq 0$, which completes the proof.

Proof of Lemma 13

The proof follows from [32, Lemma 1], where equality constraints are considered. Define $\tilde{\mathbf{y}}^*(t)$ as:

$$
\begin{aligned}
\tilde{\mathbf{y}}^*(t), t) :=& \arg\min_{\mathbf{y} \in \mathbb{R}^m} f_0(\mathbf{y}, t) + \sum_{i=1}^{p} \mathbb{I}_-(f_i(\mathbf{y}, t) - s(t)) \\
&s.t. \quad \mathbf{A}(t)\mathbf{y} = b(t),
\end{aligned}
\tag{2.72}
$$

By perturbation and sensitivity analysis [73, Sec.5.6.2], we have the following established when when $s(t) \geq 0$,

$$
0 \leq f_0(\mathbf{y}^*(t), t) - f_0(\tilde{\mathbf{y}}^*(t), t) \leq \sum_{i=1}^{p} \boldsymbol{\lambda}_i^*(t) s(t).
$$

On the other hand, replacing the indicator function $\mathbb{I}_{-}(u)$ using logarithmic barrier function $-1/c\log(-u)$, we have [73, Sec.11.2.2]

$$f_0(\hat{\mathbf{y}}^*(t),t) - f_0(\tilde{\mathbf{y}}^*(t),t) \leq \frac{p}{c(t)} \tag{2.73}$$

The result follows from combining the above two inequalities using triangular inequality.

Proof of Lemma 14

For all $t \geq 0$, if we assume that uniform MFCQ holds at $\mathbf{y}^*(t)$ (see Assumption 5), then the matrix $\mathbf{A}(t)$ has full row rank and its pseudoinverse is defined as $\mathbf{A}^{+}(t) = \mathbf{A}^T(t)[\mathbf{A}(t)\mathbf{A}^T(t)]^{-1}$. $[\mathbf{I}_m - \mathbf{A}^{+}(t)\mathbf{A}(t)]$ is the projection of $\mathbb{R}^m$ into the subspace of $\mathbb{R}^m$ which is orthogonal to $a_j(t), j = 1, \ldots, q$. Therefore, for any $\bar{\mathbf{d}}(t) \in \mathbb{R}^m$ given by uniform MFCQ, there exist a $\mathbf{z}(t) \in \mathbb{R}^m$, such that $\bar{\mathbf{d}}(t) = [\mathbf{I}_m - \mathbf{A}^{+}(t)\mathbf{A}(t)]\mathbf{z}(t)$. Define a point $\bar{\mathbf{y}}(s,t)$ sufficiently close to $\mathbf{y}^*(t)$ by:

$$\bar{\mathbf{y}}(s,t) = \mathbf{y}^*(t) + s[\mathbf{I}_m - \mathbf{A}^{+}(t)\mathbf{A}(t)]\mathbf{z}(t).$$

Notice that the point $\bar{\mathbf{y}}(s,t)$ defined above is a feasible point of (2.2) for s sufficiently small since the equality constraint functions are satisfied:

$$\begin{aligned}
\mathbf{A}(t)\bar{\mathbf{y}}(s,t) &= \mathbf{A}(t)(\mathbf{y}^*(t) + s[\mathbf{I}_m - \mathbf{A}^{+}(t)\mathbf{A}(t)]\mathbf{z}(t)) \\
&= b(t) + s[\mathbf{A}(t) - \mathbf{A}(t)\mathbf{A}^{+}(t)\mathbf{A}(t)]\mathbf{z}(t) \\
&= b(t) + s[\mathbf{A}(t) - \mathbf{A}(t)]\mathbf{z}(t) = b(t).
\end{aligned}$$

And for all active inequality constraint functions, that is $i \in \mathbb{I}(\mathbf{y}^*(t))$, we apply Taylor's theorem:

$$\begin{aligned}
f_i(\bar{\mathbf{y}}(s,t),t) &= f_i(\mathbf{y}^*(t) + s\bar{\mathbf{d}}(t),t) \\
&= f_i(\mathbf{y}^*(t),t) + \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t),t)^T s\bar{\mathbf{d}}(t) \\
&\quad + \mathbf{R}(\mathbf{y}^*(t), s\bar{\mathbf{d}}(t)) \\
&= s\nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t),t)^T \bar{\mathbf{d}}(t) + \mathbf{R}(\mathbf{y}^*(t), s\bar{\mathbf{d}}(t))
\end{aligned}$$

, where $\mathbf{R}(\mathbf{y}^*(t), s\bar{\mathbf{d}}(t))$ is the remainder satisfying

$$\frac{\mathbf{R}(\mathbf{y}^*(t), s\bar{\mathbf{d}}(t))}{\|s\bar{\mathbf{d}}(t)\|} \to 0 \quad as \quad s\bar{\mathbf{d}}(t) \to 0.$$

From part $1)$ of uniform MFCQ it follows immediately that for s sufficiently small, $\bar{\mathbf{y}}(s,t)$ is feasible for (2.2). Thus, for s sufficiently small,

$$f_0(\mathbf{y}^*(t), t) = f_0(\bar{\mathbf{y}}(0,t), t) \le f_0(\bar{\mathbf{y}}(s,t), t)$$

and

$$\begin{aligned} \nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t)^T \bar{\mathbf{d}}(t) = \nabla_{\mathbf{y}} f_0(\bar{\mathbf{y}}(0), t)^T \bar{\mathbf{d}}(t) \ge 0 \implies \\ -\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t)^T \bar{\mathbf{d}}(t) \le 0. \end{aligned}$$

Next, we consider the linear program:

$$\begin{aligned} \max_{\mathbf{d}} \quad & -\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t)^T \mathbf{d} \\ s.t. \quad & \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)^T \mathbf{d} \le -1, \quad i \in \mathbb{I}(\mathbf{y}^*(t)) \\ & a_j(t)^T \mathbf{d} = 0, \quad j = 1, \dots q \\ & \mathbf{d} \quad \text{unrestricted.} \end{aligned}$$

Any optimization variable $\mathbf{d}$ satisfying these constraint functions also satisfy the uniform MFCQ, and the value of the objective function is upper bounded by $-\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t)^T \mathbf{d} \le 0$ based on previous analysis. Besides, the feasibility of this linear program is also guaranteed by Assumption 5 (uniform MFCQ), since there exists $\|\bar{\mathbf{d}}(t)\|_2 \le d$ for some constant $d > 0$, and a constant $\epsilon > 0$ such that,

$$\begin{aligned} \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)^T \bar{\mathbf{d}}(t) \le -\epsilon, \quad i \in \mathbb{I}(\hat{\mathbf{y}}(t)) \implies \\ \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)^T \frac{\bar{\mathbf{d}}(t)}{\epsilon} \le -1, \quad i \in \mathbb{I}(\hat{\mathbf{y}}(t)), \end{aligned}$$

which means that a feasible $\mathbf{d}$ is given by $\frac{\bar{\mathbf{d}}(t)}{\epsilon}$. Furthermore, using Assumption 6 and Cauchy-Schwarz inequality we have $-\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t)^T \mathbf{d} \ge -\frac{Ld}{\epsilon}$. Together, we showed that this linear program is feasible and bounded, with $-\frac{Ld}{\epsilon} \le -\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t)^T \mathbf{d} \le 0$ holds as $t \to \infty$.

Its dual problem:

$$\begin{aligned}&\min_{\boldsymbol{\lambda},\boldsymbol{\nu}} \quad \sum_{i\in\mathbb{I}(\mathbf{y}^*(t))} -\boldsymbol{\lambda}_i\\ &s.t. \quad \boldsymbol{\lambda}_i \geq 0, \quad i \in \mathbb{I}(\mathbf{y}^*(t))\\ &\boldsymbol{\nu}_j \quad \text{unrestricted}, \quad j = 1, \dots q,\\ &\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t),t)+\sum_{i=1}^{p}\boldsymbol{\lambda}_i\nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t),t)+\mathbf{A}(t)^T\boldsymbol{\nu} = 0\end{aligned}$$

is also feasible and bounded since strong duality holds. Notice that the set of optimization variables satisfying the dual problem is $\mathbb{K}(\mathbf{y}^*(t))$. That is, the set of feasible $\boldsymbol{\lambda}$ vectors is nonempty and bounded

$$0 \leq \sum_{i\in\mathbb{I}(\mathbf{y}^*(t))} \boldsymbol{\lambda}_i^*(t) \leq \frac{Ld}{\epsilon}$$

as $t \to \infty$. From part 2) of uniform MFCQ, since $\mathbf{A}(t)^T$ has linearly independent columns, and $\mathbf{A}(t)\mathbf{A}(t)^T$ is invertible, we have:

$$\begin{aligned}\mathbf{A}(t)\mathbf{A}(t)^T\boldsymbol{\nu}^*(t) = &-\mathbf{A}(t)[\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t),t)\\ &+\sum_{i=1}^{p}\boldsymbol{\lambda}_i^*(t)\nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t),t)]\end{aligned} \tag{2.74}$$

For any invertible matrix $\mathbf{B}$, according to [78, Theorem 4.2.2(Rayleigh)] we have:

$$\lambda_{\min}(\mathbf{B}^T\mathbf{B}) = \min_{x\neq 0\in\mathbb{C}^n} \frac{\mathbf{x}^T\mathbf{B}^T\mathbf{B}\mathbf{x}}{\mathbf{x}^T\mathbf{x}},$$

by definition which implies

$$\sigma_{\min}(\mathbf{B})\|x\|_2 \leq \|\mathbf{B}\mathbf{x}\|_2.$$

Therefore, since $\mathbf{A}(t)\mathbf{A}(t)^T$ is invertible, the following inequality holds for the left side of (2.74):

$$\sigma_{\min}(\mathbf{A}(t)\mathbf{A}(t)^T)\|\boldsymbol{\nu}^*(t)\|_2 \leq \|\mathbf{A}(t)\mathbf{A}(t)^T\boldsymbol{\nu}^*(t)\|_2.$$

By definition of induced matrix norm and triangular inequality, we have:

$$\begin{aligned}
&\|\mathbf{A}(t)[\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t) + \sum_{i=1}^{p} \boldsymbol{\lambda}_i^*(t) \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)]\|_2 \\
\le &\|\mathbf{A}(t)\|_2(\|\nabla_{\mathbf{y}} f_0(\mathbf{y}^*(t), t)\|_2 + \|\sum_{i=1}^{p} \boldsymbol{\lambda}_i^*(t) \nabla_{\mathbf{y}} f_i(\mathbf{y}^*(t), t)\|_2) \\
\le &\sigma_{\max}(\mathbf{A}(t))(L + \frac{Ld}{\epsilon} \sum_{i=1}^{p} L_i)
\end{aligned}$$

Combining the above two inequalities together, we have:

$$\begin{aligned}
\|\boldsymbol{\nu}^*(t)\|_2 &\le \frac{\sigma_{\max}(\mathbf{A}(t))}{\sigma_{\min}(\mathbf{A}(t))^2}(L + \frac{Ld}{\epsilon} \sum_{i=1}^{p} L_i) \\
&\le \frac{\tau_{\max}}{\tau_{\min}^2}(L + \frac{Ld}{\epsilon} \sum_{i=1}^{p} L_i).
\end{aligned}$$

It indicates that the set of feasible $\boldsymbol{\nu}^*(t)$ vectors is also nonempty and bounded as $t \to \infty$, which completes the proof.

Proof of Theorem 15

The structure of proof is similar to the proof of Theorem 7. According to Lemma 4, the trajectory $\mathbf{y}(t)$ of system (2.65) satisfy the optimization dynamics as in (2.64), with $\mathbf{H}$ being the designed Hurwitz matrix. Similarly, the solution to this ODE satisfies the following inequality:

$$\sum_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} \hat{\Phi}(\mathbf{y}(t), t)\|_2^2 \le c^2 e^{-2\alpha t} \sum_{j=0}^{k-1} \|\nabla_{\mathbf{y}}^{(j)} \hat{\Phi}(\mathbf{y}(0), 0)\|_2^2$$

for some constant $c > 0$, $-\alpha := \max_{\lambda \in \mathrm{spec}(\mathbf{H})} \Re[\lambda] + \epsilon_H$ for some $\epsilon_H > 0$ small enough.

Next, we use the mean-value theorem to expand $\nabla_{\mathbf{y}} \hat{\Phi}(\mathbf{y}, t)$, where $\boldsymbol{\eta}(\boldsymbol{t})$ is a convex combination of $\mathbf{y}(t)$ and $\hat{\mathbf{y}}^*(t)$:

$$\mathbf{y}(t) - \hat{\mathbf{y}}^*(t) = \nabla_{\mathbf{y}\mathbf{y}}^{-1} \hat{\Phi}(\boldsymbol{\eta}(t), t) \nabla_{\mathbf{y}} \hat{\Phi}(\mathbf{y}(t), t). \tag{2.75}$$

Notice that the Hessian $\nabla_{yy} \hat{\Phi}(y, t)$ is given by:

$$\begin{aligned}
&\frac{1}{c(t)} \sum_{i=1}^{p} \frac{\nabla_{\mathbf{y}} f_i(\mathbf{y}(t), t) \nabla_{\mathbf{y}} f_i(\mathbf{y}(t), t)^T}{[s(t) - f_i(\mathbf{y}(t), t)]^2} + \frac{\nabla_{\mathbf{y}\mathbf{y}} f_i(\mathbf{y}(t), t)}{s(t) - f_i(\mathbf{y}(t), t)} \\
&+ \nabla_{\mathbf{y}\mathbf{y}} f_0(\mathbf{y}(t), t)
\end{aligned}$$

It follows from Assumption 4 and [78, Corollary 4.3.12], that $\|\nabla_{\mathbf{yy}}^{-1}\hat{\Phi}(\mathbf{y},t)\|_2 \leq \|\nabla_{\mathbf{yy}}^{-1}f_0(\mathbf{y},t)\|_2 \leq m_f^{-1}$. Taking the norm on both sides of equation (2.75) we have:

$$\begin{aligned}
&\|\mathbf{y}(t)-\hat{\mathbf{y}}^*(t)\|_2 \leq Ce^{-\alpha t},\\
&0 \leq C = \left(\tfrac{c^2}{m_f^2}\sum_{j=0}^{k-1}\|\nabla_{\mathbf{y}}^{(j)}\hat{\Phi}(\mathbf{y}(0),0)\|_2^2)\right)^{\frac{1}{2}} < \infty.
\end{aligned}$$

On the other hand, convexity of $f_0(\mathbf{y},t)$ implies that for each $t \geq 0$

$$f_0(\mathbf{y}(t),t)-f_0(\hat{\mathbf{y}}^*(t),t) \leq \nabla_{\mathbf{y}}f_0(\mathbf{y}(t),t)^T(\mathbf{y}(t)-\hat{\mathbf{y}}^*(t))$$

By applying Cauchy-Swhartz inequality on the right-hand side and using Assumption 6 we obtain:

$$|f_0(\mathbf{y}(t),t)-f_0(\hat{\mathbf{y}}^*(t),t)| \leq LCe^{-\alpha t}, \tag{2.76}$$

Lastly, a direct application of Lemma 13 and Lemma 14 yields:

$$|f_0(\hat{\mathbf{y}}^*(t),t)-f_0(\mathbf{y}^*(t),t)| \leq pc_0e^{-\alpha_c t}+\frac{Ld}{\epsilon}s_0e^{-\alpha_s t} \tag{2.77}$$

It follows from (2.76) ,(2.77) , and the triangular inequality that:

$$|f_0(\mathbf{y}(t),t)-f_0(\mathbf{y}^*(t),t)| \leq LCe^{-\alpha t}+pc_0e^{-\alpha_c t}+\frac{Ld}{\epsilon}s_0e^{-\alpha_s t}$$

which completes the proof.

2.3 Conclusion

In this chapter, we study the model-based approach for online decision-making of nonlinear dynamical systems. Particularly, we first develop an optimization-based framework for joint real-time trajectory planning and feedback control of feedback-linearizable systems. We implicitly define a target trajectory as the optimal solution of a time-varying optimization problem, which is strongly convex and smooth. For systems that are (dynamic) full-state linearizable, the proposed control law transforms the nonlinear system into an optimization algorithm of sufficiently high order. Under reasonable assumptions, our method globally asymptotically converges to the time-varying optimal solution of the original problem.

We further extend the result by considering a more general set of nonlinear dynamical systems, i.e., differentially flat systems, and considering adding time-varying equality and inequality constraints to the time-varying optimization framework. We investigate the problem of steering in real time a differentially flat system to the minimizer of a time-varying constrained optimization problem. Under reasonable assumptions, we show optimization dynamics for (un)constrained time-varying optimization problems globally asymptotically converge to the optimal solution. Lastly, the effectiveness of our method is illustrated in two numerical examples: a multi-robot navigation problem and an obstacle avoidance problem.

Chapter 3

Constrained Reinforcement Learning via Stochastic Dissipative Gradient Descent Ascent

The chapter is structured as follows: In Section 3.1, we formally present the Constrained Reinforcement Learning problem, along with two major methodologies for finding the optimal policy and their limitations. We then delve into our key insight from saddle flow dynamics, explaining the shortcomings of vanilla gradient descent ascent in convergence. Moving on to Section 3.2, we introduce the Dissipative Gradient Descent Ascent (DGDA) algorithm, designed to solve the min-max optimization problem with last iterate convergence guarantees. Moreover, we provide linear convergence rate estimates for DGDA in both the bilinear setting and strongly convex-strongly concave settings. To assess DGDA's performance, we compare it with other methods, such as Extra-Gradient (EG) and Optimistic Gradient (OG) methods, which also exhibit linear convergence in similar settings. In Section 3.3, we apply the stochastic DGDA algorithm to address the C-RL problem in occupancy measure space, showcasing global asymptotic convergence in terms of occupancy measure and the recovery of the optimal policy. Section 3.4 illustrates the effectiveness of our approach through a case study involving a discrete-time single-server queue with a limited buffer size problem. Finally, we conclude the chapter with remarks in Section 3.6.

Notation

Let $\mathcal{K} \subset \mathbb{R}^n$ be a closed convex set. Given a point $y \in \mathbb{R}^n$, $\Psi_{\mathcal{K}}[y] = \arg\min_{z\in\mathcal{K}} \|z - y\|$ denote the point-wise projection (nearest point) in $\mathcal{K}$ to y. Given $x \in \mathcal{K}$ and $v \in \mathbb{R}^n$, define the vector field projection of v at x with respect to $\mathcal{K}$ as: $\Pi_{\mathcal{K}}[x, v] = \lim_{\delta\to 0^+} \frac{\Psi_{\mathcal{K}}[x+\delta v]-x}{\delta}$

3.1 Problem Formulation

In the constrained reinforcement learning problem (C-RL), $\mathcal{S}$ denotes the finite state space, $\mathcal{A}$ denotes the finite action space, and $P : \mathcal{S} \times \mathcal{A} \to \triangle^{|\mathcal{S}|}$ gives the transition dynamics of the CMDP, where $P(\cdot|s, a)$ denotes the probability distribution of next state conditioned on the current state s and action a. $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function, $g^i : \mathcal{S} \times \mathcal{A} \to [-1, 1]$ denotes the i^{th} constraint cost function. The scalar γ denotes the discount factor, and q denotes the initial distribution of the states. A stationary policy is a map $\pi : \mathcal{S} \to \triangle^{|\mathcal{A}|}$ from states to a distribution in the action space. The value functions for both reward and constraints' cost following policy π are given by:

$$V_r^\pi(q) = (1-\gamma)\mathbf{E}_\pi[\textstyle\sum_{t=0}^\infty \gamma^t r(s_t, a_t) \,|\, s_0 \sim q],$$
$$V_{g^i}^\pi(q) = (1-\gamma)\mathbf{E}_\pi[\textstyle\sum_{t=0}^\infty \gamma^t g^i(s_t, a_t) \,|\, s_0 \sim q].$$

The standard C-RL problem aims to maximize the total reward function while satisfying requirements in secondary cumulative reward constraints:

$$\begin{aligned} \max_\pi \;\; & V_r^\pi(q) \\ s.t. \;\; & V_{g^i}^\pi(q) \geq h^i, \;\; \forall i \in [I]. \end{aligned} \tag{3.1}$$

There are two major methodologies for finding the optimal policy of a C-RL problem: The first approach is to apply Lagrangian duality to parametrize and solve the C-RL problem (3.1) in the *policy space* (π), which is equivalent to solving a min-max optimization problem [42, 43, 44, 45, 46]. These approaches solve the min-max optimization problem using a sampling-based primal-dual algorithm or stochastic gradient descent-ascent (SGDA) algorithms, where the Lagrangian function is augmented with a possible regularization term, e.g., KL divergence. The primal variables and dual variables are updated iteratively, either using gradient information or solving a sub-optimization problem. However, the Lagrangian function

$$L(\pi, \mu) = V_r^\pi + \sum_{i=1}^{I} \mu_i (V_{g^i}^\pi - h^i) \tag{3.2}$$

is nonlinear (nonconvex) in π and classical sampling-based primal-dual algorithms generally require strict convexity to converge. Therefore, proposed algorithms generally fail to converge to an equilibrium. Among the sampling-based primal-dual algorithms, several algorithms output a *mixing policy* of the form

$$\pi_T = \sum_{k=0}^{T-1} \eta_k \pi_k, \tag{3.3}$$

which is a weighted average of the history updates [42, 43, 44]. However, the output policy π_k oscillates and does not converge to the optimal policy. Therefore, the above algorithms only provide average iterate convergence instead of a last-iterate convergence guarantee, which is critical in online decision-making.

On the other hand, the well-studied constrained Markov Decision Process (CMDP) framework parametrizes and solves the C-RL in *occupancy measure space* (λ) [39]. Given a policy π, occupancy measure is defined as

$$\lambda^\pi(s, a) = (1-\gamma) \textstyle\sum_{t=0}^{\infty} \gamma^t P_q^\pi(s_t = s, a_t = a), \tag{3.4}$$

where $s_0 \sim q$. By definition, the occupancy measure belongs to the probability simplex $\lambda^\pi \in \Delta$, which can be interpreted as the total expected probability of visiting any state-action pair along the trajectory. The C-RL problem in policy space (3.1) can be equivalently written as a linear programming problem in occupancy measure space λ:

$$\begin{aligned} \max_{\lambda \in \Delta} \quad & \textstyle\sum_a \lambda_a^T r_a \\ s.t. \quad & \textstyle\sum_a \lambda_a^T g_a^i \geq h^i, \quad i \in [I] \\ & \textstyle\sum_{a \in \mathcal{A}} (I - \gamma P_a^T) \lambda_a = (1-\gamma) q, \end{aligned} \tag{3.5}$$

where $r_a = [r(1, a), \ldots, r(s, a)]^T \in \mathbb{R}^{|\mathcal{S}|}$ denotes reward function associated with action a, $\lambda_a = [\lambda(1, a), \ldots, \lambda(s, a)]^T \in \mathbb{R}^{|\mathcal{S}|}$ is the a^{th} column of λ^π, P_a denotes the transition matrix associated with action a. Besides, the optimal policy could be recovered from the optimal occupancy measure by solving the linear programming problem (3.5):

$$\pi^*(a|s) = \frac{\lambda^*(s, a)}{\sum_{a' \in \mathcal{A}} \lambda^*(s, a')}$$

However, this approach requires knowledge of the underlying transition kernel explicitly i.e., P_a, r_a, g_a^i. Also, CMDP-LPs are tabular solution methods that suffer from the curse of dimensionality, and high-dimensional solution methods are lacking.

In summary, the CMDP approach directly solves for the optimal occupancy measure and policy, but it requires explicit knowledge of the transition kernel. On the other hand, when using sampling-based gradient descent-ascent algorithms to solve the C-RL problem in policy space, the output often involves a mixture of historical policies and fails to converge to the optimal policy. These algorithms only offer average-iterate convergence without last-iterate convergence guarantees. The main limitation lies in the insufficient convexity of the Lagrangian function for the C-RL problem, which hinders standard gradient descent-ascent from converging. Specifically, the Lagrangian function is bilinear in occupancy measure space and

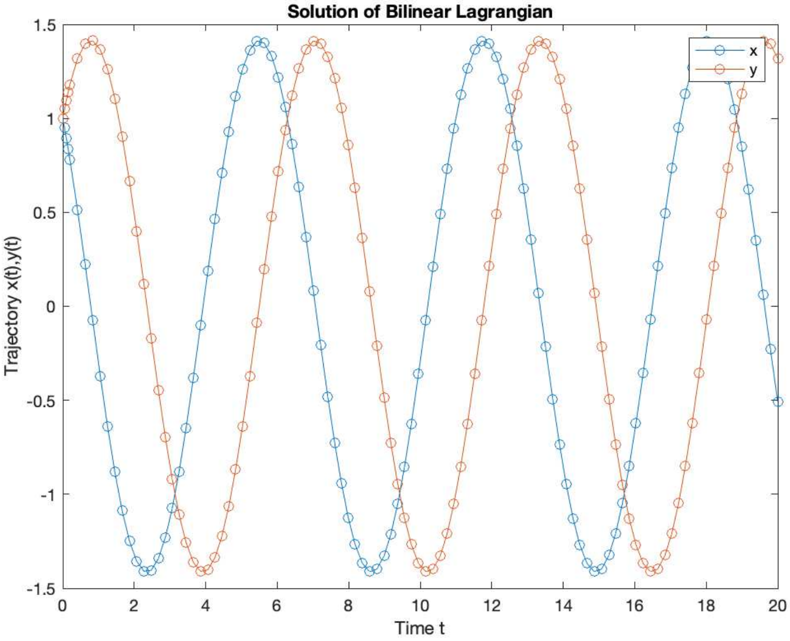

Figure 3-1. Time series trajectories of gradient descent ascent algorithm for $L(x, y) = xy$

nonconvex in policy space. In this paper, we aim to overcome these challenges by introducing a novel algorithm.

Key insight from saddle flow dynamics

Before introducing our algorithm, we would like to illustrate our key insight from saddle flow dynamics, which explains why the primal-dual algorithm oscillates and does not converge. For a min-max optimization problem, primal-dual algorithms require the Lagrangian $L(x, y)$ function to be strictly convex or concave on x or y, respectively, to converge. Consider the following motivating example with bilinear Lagrangian function:

$$\min_x \max_y L(x, y) := xy.$$

Our goal is to apply different dynamic laws that seek to converge to some saddle point $(x^*, y^*) = (0, 0)$ of $L(x, y)$, which satisfies $L(x^*, y) \leq L(x^*, y^*) \leq L(x, y^*)$. In

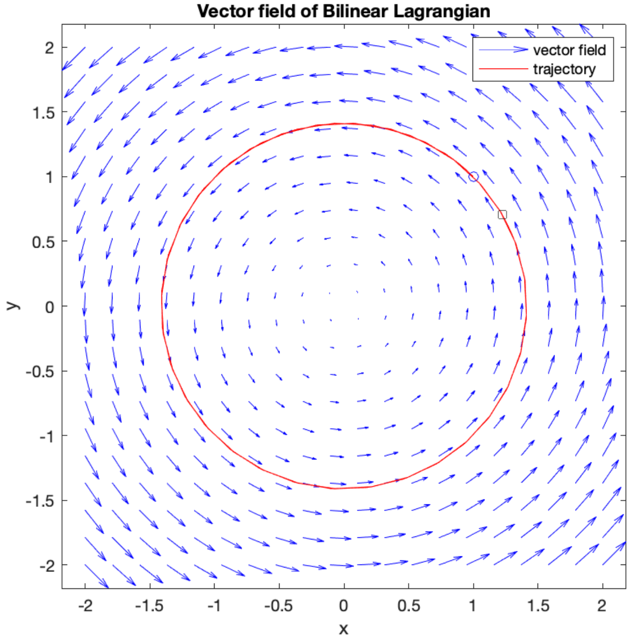

Figure 3-2. Phase portrait of gradient descent ascent algorithms for $L(x, y) = xy$

particular, consider the following classical primal-dual algorithm:

$$\dot{x} = -\nabla_x L(x, y) = -y,$$
$$\dot{y} = \nabla_y L(x, y) = x.$$

In Figure 3-1 plots the time series trajectory of states x and y, and Figure 3-2plots the vector field and corresponding phase portrait. We observe that the dynamical system oscillates and does not converge to the saddle point (0,0).

In [79], the authors introduce a regularization framework for saddle flow dynamics that guarantees asymptotic convergence to a saddle point based on mild assumptions. In this paper, we further extend the above framework to solve the C-RL problem. Specifically, consider the following constrained min-max optimization problem,

$$\min_{x \in \mathcal{K}} \max_{y \in \mathcal{V}} L(x, y)$$

where $\mathcal{K} \subset \mathbb{R}^n, \mathcal{V} \subset \mathbb{R}^m$ are bounded closed convex sets. We propose a regularized surrogate for $L(x,y)$ via the following augmentation:

$$L(x,y,z,w) := \frac{1}{2\rho}\|x-z\|^2 + L(x,y) - \frac{1}{2\rho}\|y-w\|^2$$

The following projected and regularized saddle flow dynamics aim to find the saddle points of the regularized Lagrangian, which contains the saddle point of the original Lagrangian. The regularized saddle flow dynamics still preserve the same distribution structure, which can be implemented in a fully distributed fashion, and requires the same gradient information as the classical primal-dual algorithm:

$$\begin{aligned} \dot{x} &= \Pi_{\mathcal{K}}\Big[x, -\nabla_x L(x,y) - \frac{1}{\rho}(x-z)\Big], \dot{z} = \Pi_{\mathcal{K}}\Big[z, \frac{1}{\rho}(x-z)\Big] \\ \dot{y} &= \Pi_{\mathcal{V}}\Big[y, -\nabla_y L(x,y) - \frac{1}{\rho}(y-w)\Big], \dot{w} = \Pi_{\mathcal{V}}\Big[w, \frac{1}{\rho}(y-w)\Big] \end{aligned} \tag{3.6}$$

Theorem 17. *Assume that $L(\cdot,y)$ is convex for $\forall y$ and $L(x,\cdot)$ is concave for $\forall x$, continuously differentiable, and there exists at least one saddle point $(x^* \in \mathcal{K}, y^* \in \mathcal{V})$, where $\mathcal{K} \subset \mathbb{R}^n, \mathcal{V} \subset \mathbb{R}^m$ are closed and convex. Then the projected saddle flow dynamics (3.6) asymptotically converge to some saddle point (x^*, y^*) of $L(x,y)$, while $x(t) \in \mathcal{K}, y(t) \in \mathcal{V}, \forall t$ with initialization $x(0) \in \mathcal{K}, y(0) \in \mathcal{V}$.*

Proof: See Section 3.5.

The above theorem shows the projected and regularized saddle flow dynamics will asymptotically converge to the saddle point of the Lagrangian function, which requires mild assumptions on convexity. Additionally, the following result summarizes conditions under which the solutions of the projected system exist and are unique.

Proposition 18. *[80, Prop 2.2] Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be Lipschitz on a closed convex polyhedron $\mathcal{K} \in \mathbb{R}^n$. Then, for any $x_0 \in \mathcal{K}$, there exists a unique solution $t \to x(t)$ of the projected system $\dot{x} = \Pi_{\mathcal{K}}\Big[x, f(x)\Big]$ with $x(0) = x_0$.*

We now apply the regularized saddle flow dynamics to the bilinear Lagrangian

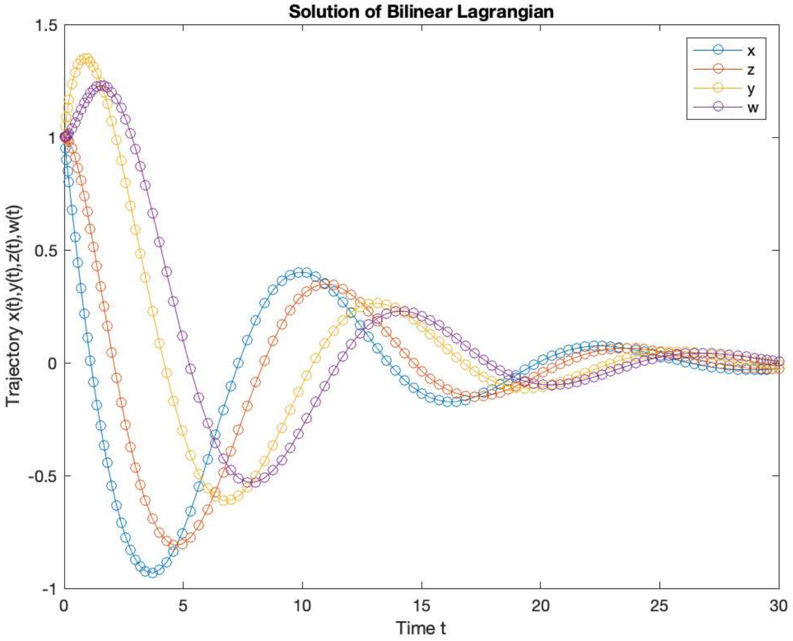

Figure 3-3. Regularized saddle flow dynamics for $L(x, y) = xy$

function $L(x, y) = xy$.

$$\dot{x} = -y - \frac{1}{\rho}(x - z), \qquad \dot{z} = \frac{1}{\rho}(x - z),$$
$$\dot{y} = x - \frac{1}{\rho}(y - w), \qquad \dot{w} = \frac{1}{\rho}(y - w).$$

According to Figure 3-3, the trajectories of the above saddle flow dynamics asymptotically converge to the saddle point $(0, 0, 0, 0)$, even when the original Lagrangian function is bilinear.

3.2 Dissipative gradient descent asecnt

In recent years, Min-max optimization and variational inequality problems (VIP) have received significant attention, particularly in domains such as Generative Adversarial Networks (GANs) [81, 57, 64], Reinforcement Learning (RL) [82], and Constrained Reinforcement Learning (C-RL) [53, 49]. However, a major challenge faced by these approaches is the instability of the training process. Specifically, when solving the min-max optimization problem using variants of the Stochastic Gradient Descent Ascent (SGDA) algorithm simultaneously, it often leads to oscillatory behavior rather than converging to an equilibrium.

A common approach to address this instability is taking averaged iterates, which combines previous outputs with certain weights. However, few theoretical guarantees exist for the averaged iterates, especially when the objective function is not convex-concave [47, 48]. Moreover, the practice of averaging the weights of neural networks proves impractical, particularly in training GANs [64]. In the context of Reinforcement Learning (RL) and Constrained Reinforcement Learning (C-RL), relying on averaging results becomes undesirable because the mixture of past policies obscures oscillating or overshooting objective/constraint functions, hindering the attainment of an optimal policy iterate [49]. As a result, it becomes crucial to explore training algorithms that can ensure the final iteration of the training process approaches the equilibrium point directly, a concept known as last-iterate convergence, rather than merely relying on an average outcome. Therefore, the Extra-gradient (EG) method [50], the Optimistic gradient (OG) method [51], and their variants have garnered significant attention in recent literature due to their superior empirical performance and last-iterate convergence guarantees, particularly in the convex-concave setting.

In this section, we introduce a novel first-order algorithm called the Dissipative Gradient Descent-Ascent (DGDA) algorithm, which demonstrates linear last-iterate convergence for strongly monotone (and bilinear) and Lipschitz variational inequality problems (VIPs) without any additional assumptions. Notably, we establish that for bilinear problems, the proposed algorithm achieves a superior linear convergence rate in terms of the constant (see Table 3-I) compared to the standard rates of EG and OGDA for bilinear problems. And for strongly convex-strongly concave games, the proposed algorithm achieves a convergence rate when the condition

	[56]	[57]	[58]	[59]	This Work
EG (Bil)	$\frac{1}{2}\frac{\gamma^2}{\kappa/\sigma_{\min}^2(A)+\gamma^2\kappa^2}$	-	$\frac{\kappa^{-1}}{20}$	$\frac{\kappa^{-1}}{64}$	-
EG (Str)	-	-	$\frac{\kappa^{-1}}{4}$	$\frac{\kappa^{-1}}{4} + \frac{\gamma^2}{64L^2}$	-
OG (Bil)	$\frac{\kappa^{-1}}{16}$	-	$\frac{\kappa^{-1}}{800}$	$\frac{\kappa^{-1}}{128}$	-
OG (Str)	-	$\frac{\kappa^{-1}}{4}$	$\frac{\kappa^{-1}}{4}$	$\frac{\kappa^{-1}}{4} + \frac{\gamma^2}{128L^2}$	-
DG (Bil)	-	-	-	-	$\frac{\kappa^{-1}}{4}$
DG (Str)	-	-	-	-	$\kappa^{-1} - \mathcal{O}(\kappa^{-2})$

Table 3-I. Comparision of global convergence rates results for bilinear and strongly-convex-strongly-concave objective functions, including our proposed Dissipative gradient descent (DGDA), Extra-gradient (EG), and Optimistic gradient descent ascent (OGDA) methods. If a result shows that the iterates converge as $\mathcal{O}((1-r)^t)$, the quantity r is reported (the larger the better).

number $\kappa \geq 2$. To validate the effectiveness of DGDA in solving bilinear and strongly convex-strongly concave problems, we present two numerical examples. In both cases, DGDA consistently outperforms EG and OG methods, showcasing its superior performance.

Preliminaries

In this paper, we study the problem of finding saddle points in the min-max optimization problem:

$$\min_{x\in\mathbb{R}^n} \max_{y\in\mathbb{R}^m} f(x,y), \tag{3.8}$$

where the function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a convex-concave function. Precisely, $f(\cdot, y)$ is convex for all $y \in \mathbb{R}^m$ and $f(x, \cdot)$ is concave for all $x \in \mathbb{R}^n$. Our goal is to study different optimization algorithms that seek to converge to some saddle point (x^*, y^*) of Problem 3.8.

Definition 2 (Saddle Point). *A point $(x^*, y^*) \in \mathbb{R}^n \times \mathbb{R}^m$ is a saddle point of convex-*

concave function (3.8) *that satisfies*

$$f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*) \tag{3.9}$$

for all $x \in \mathbb{R}^n, y \in \mathbb{R}^m$.

We present some properties and notations used in our results.

Definition 3 (L-Lipschitz)**.** *A function* $F : \mathbb{R}^n \to \mathbb{R}$ *is L-Lipschitz if it has L-Lipschitz continuous gradients on* $\mathbb{R}^n$*, i.e.,* $\forall w, w'$*, we have* $\|F(w) - F(w')\| \leq L\|w - w'\|, \forall w, w' \in \mathbb{R}$.

Definition 4 (Strongly convex)**.** *A differentiable function* $f : \mathbb{R}^n \to \mathbb{R}$ *is said to be* μ*-strongly convex if* $f(w) \geq f(w') + \nabla f(w)^T(w - w') + \frac{\mu}{2}\|w - w'\|^2$*. Further* $f(w)$ *is* μ*-strongly concave if* $-f(w)$ *is* μ*-strongly convex. If* $\mu = 0$*, then we recover the definition of convexity for a continuous differentiable function.*

And throughout this paper, we consider two specific cases of Problem 3.8, which are stated in the next set of assumptions.

Assumption 7 (Bilinear function)**.** *The function* $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ *is a bilinear function of the form* $f(x, y) = x^T Ay$*, where* $A \in \mathbb{R}^{m \times n}$ *is non-singular.*

Assumption 8 (Strongly convex-strongly concave function)**.** *The function* $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ *is continuously differentiable,* μ *strongly convex in* x*, and* μ *strongly concave in* y*. Further,* f *is L-Lipschitz. The unique saddle point of* $f(x, y)$ *is denoted by* (x^*, y^*)*.*

First-order explicit algorithms

In this section, we introduce several first-order methods for solving the min-max problem in 3.8. Precisely, we focus on Gradient Descent-Ascent (GDA), Extra-gradient (EG), and Optimistic Gradient Descent-Ascent (OGDA) methods.

Gradient descent ascent (GDA):

$$x_{k+1} = x_k - \eta \nabla_x f(x_k, y_k), \tag{3.10}$$

$$y_{k+1} = y_k + \eta \nabla_y f(x_k, y_k) \tag{3.11}$$

When the problem is strongly convex-strongly concave, the GDA method provides linear convergence. However, the GDA method is known not to converge when the game is bilinear. Therefore, EG and OGDA methods have attracted much attention in recent literature because of their superior empirical performance in solving min-max optimization problems such as training GANs and solving C-RL problems.

Extra-gradient is a classical method introduced in [50], where its linear rate of convergence for smooth and bilinear functions and strongly convex-strongly concave functions have been established in many recent literatures (See Table 3-I). The Extra-gradient method first computes an extrapolated point $(x_{k+1/2}, y_{k+1/2})$ by performing a GDA update. Then the gradients evaluated at the extrapolated point are used to compute the new iterates (x_{k+1}, y_{k+1}) by performing the following updates.

Extra-gradient (EG):

$$\begin{aligned} x_{k+1/2} &= x_k - \eta \nabla_x f(x_k, y_k), \\ y_{k+1/2} &= y_k + \eta \nabla_y f(x_k, y_k), \\ x_{k+1} &= x_k - \eta \nabla_x f(x_{k+1/2}, y_{k+1/2}), \\ y_{k+1} &= y_k + \eta \nabla_y f(x_{k+1/2}, y_{k+1/2}). \end{aligned} \tag{3.12}$$

One issue with the Extra-gradient method is that, as the name suggests, each update requires evaluation of extra gradients at the extrapolated point $(x_{k+1/2}, y_{k+1/2})$, which doubles the computational complexity of EG method compared to vanilla GDA method. On the other hand, the Optimistic gradient descent ascent (OGDA) method store and re-use the extrapolated gradient for the extrapolation, which only requires a single gradient computation per update.

Optimistic gradient descent ascent (OGDA):

$$\begin{aligned} x_{k+1} &= x_k - 2\eta\nabla_x f(x_k, y_k) + \eta\nabla_x f(x_{k-1}, y_{k-1}), \\ y_{k+1} &= y_k + 2\eta\nabla_y f(x_k, y_k) - \eta\nabla_y f(x_{k-1}, y_{k-1}). \end{aligned} \tag{3.13}$$

The convergence properties of OGDA were recently investigated in (refer to Table 3-I), demonstrating linear convergence rates with smooth and bilinear functions, as well as strongly convex-strongly concave functions.

The algorithm presented below was independently introduced by Jiawei et al. in [83]. It shares a similar structure with our own proposed algorithm. However, it was originally introduced to solve the nonconvex-concave min-max optimization problem, where the objective function is nonconvex in x and concave of y. They introduce a "smoothing" technique to the primal updates to fix the oscillation issue. Preciesly, the introduce an auxiliary sequence $\{z_k\}$ and define a function

$$K(x, z; y) = f(x, y) + \frac{p}{2}\|x - z\|^2, \tag{3.14}$$

where $p > 0$ is a constant. By alternately performing gradient descent and gradient ascent on this function and incorporating an averaging step on z, the resulting algorithm can be described as an asynchronous approach.

Smoothed gradient descent ascent (Smoothed-GDA):

$$\begin{aligned} x_{k+1} &= x_k - c\big(\nabla_x f(x_k, y_k) + p(x_k - z_k)\big) \\ y_{k+1} &= y_k + \alpha\nabla_y f(x_{k+1}, y_k) \\ z_{k+1} &= z_k + \beta(x_{k+1} - z_k) \end{aligned} \tag{3.15}$$

Bilinear Objective

In this section, we present a novel first-order method for solving the min-max optimization problem outlined in 3.2. The foundation of this method was initially introduced in [79], where the authors proposed a regularization framework for continuous saddle flow dynamics, ensuring asymptotic convergence to a saddle point under mild assumptions. Consider a regularized surrogate for $f(x, y)$ in Problem 3.8 via the following augmentation:

$$f(x, y, \hat{x}, \hat{y}) := \frac{\rho}{2}\|x - \hat{x}\|^2 + f(x, y) - \frac{\rho}{2}\|y - \hat{y}\|^2, \tag{3.16}$$

where$\hat{x} \in \mathbb{R}^n$ and $\hat{y} \in \mathbb{R}^m$ serve as two new sets of virtual variables and $\rho > 0$ is a regularization parameter. The following Lemma verifies the fixed positions of saddle points between $f(x, y)$ and $f(x, y, \hat{x}, \hat{y})$ with virtual variables aligned with original variables.

Lemma 19 (Saddle Point Invariance)**.** *[79, Lemma 6] For problem 3.8, if Assumption 7 or Assumption 8 holds, then a point (x^*, y^*) is a saddle point of $f(x, y)$ if and only if $(x^*, y^*, \hat{x}^*, \hat{y}^*)$ is a saddle point of $f(x, y, \hat{x}, \hat{y})$, with $\hat{x}^* = x^*$ and $\hat{y}^* = y^*$.*

The dissipative saddle-flow dynamic motivates the following Dissipative gradient descent ascent algorithm, where the regularized function $f(x, y, \hat{x}, \hat{y})$ is convex in $(x, \hat{x})$ and concave in $(y, \hat{y})$.

Dissipative gradient descent ascent (DGDA):

$$\begin{bmatrix} x_{k+1} \\ \hat{x}_{k+1} \\ y_{k+1} \\ \hat{y}_{k+1} \end{bmatrix} = \begin{bmatrix} x_k - \eta\nabla_x f(x_k, y_k) - \rho(x_k - \hat{x}_k) \\ \hat{x}_k - \rho(\hat{x}_k - x_k) \\ y_k + \eta\nabla_y f(x_k, y_k) - \rho(y_k - \hat{y}_k) \\ \hat{y}_k - \rho(\hat{y}_k - y_k) \end{bmatrix} \tag{3.17}$$

Significantly, the proposed DGDA algorithm has twice as many state variables as the vanilla GDA algorithm and EG methods. However, it stands out as it only necessitates a single gradient computation per update. Additionally, DGDA does not require storing and re-using the extrapolated gradient, which is a characteristic of the OGDA method. Another advantage of DGDA is that it remains a synchronized algorithm, preserving the same distributed structure that the vanilla GDA algorithm may have. As a result, it can be implemented in a fully distributed manner. In the following theorem, we elucidate the convergence rate of the DGDA method for the bilinear min-max optimization problem.

Theorem 20. *(Last-iterate convergence of DGDA, bilinear case) If Assumption 7 holds, then the updates 3.17 of DGDA with $\rho = 1/2$ and $\eta = 1/\delta_{\max}(A)$ provide linearly converging iterates:*

$$r_k \leq \left(1 - \frac{1}{4}\frac{\sigma^2_{\min}(A)}{\sigma^2_{\max}(A)}\right)^k r_0, \tag{3.18}$$

where $r_k = \|x_k - x^*\|^2 + \|y_k - y^*\|^2 + \|\hat{x}_k - \hat{x}^*\|^2 + \|\hat{y}_k - \hat{y}^*\|^2$.

Proof: *See Appendix 3.5.*

The outcome detailed in Theorem 20 underscores DGDA's linear convergence within a bilinear scenario. The total iteration count required to attain an ϵ-accurate solution is bound by $\mathcal{O}(\kappa \log(1/\epsilon))$, where $\kappa := \sigma^2_{\max}(A)/\sigma^2_{\min}(A)$ signifies the bilinear problem's condition number. It's worth noting that this convergence finding aligns with the results in Table 3-I. Notably, our proposed DGDA algorithm holds the potential for theoretically swifter convergence, courtesy of a heightened constant value of $1/4$.

Strongly Monotone Objective

In the subsequent theorem, we provide an estimation of the convergence rate for the DGDA algorithm when applied to address the broader context of a strongly convex-strongly concave min-max optimization problem as depicted in equation 3.8.

Theorem 21. *(Last-iterate convergence of DGDA, strongly convex-strongly concave case) If Assumption 8 holds, then the updates 3.17 with $\rho = 1/2$ and $\eta = 1/(L+\mu)$ of DGDA provide linearly converging iterates:*

$$r_k \leq \left(1 - \kappa^{-1} + \mathcal{O}(\kappa^{-2})\right)^k r_0 \tag{3.19}$$

where $r_k = \|x_k - x^*\|^2 + \|y_k - y^*\|^2 + \|\hat{x}_k - \hat{x}^*\|^2 + \|\hat{y}_k - \hat{y}^*\|^2$ *and* $\kappa := L/\mu \in (1, \infty)$ *denotes the condition number of the problem.*

Proof: *See Appendix 3.5.*

The upcoming Lemma provides a straightforward contrast between the estimated linear convergence rate of DGDA as given by equation (3.19), and established convergence rates for the EG and OGDA methods in the context of a strongly convex-strongly concave problem. Our demonstration reveals that our proposed method exhibits a theoretically expedited convergence outcome, particularly when $\kappa \geq 2$.

Corollary 22 (Strongly convex-strongly concave, comparison with known rates)**.** *If Assumption 8 holds and suppose that* $L \geq 2m$*, i.e.,* $\kappa \geq 2$ *, the linear convergence rate estimate of DGDA* (3.19) *is smaller (better) than the standard one* $1 - \mu/4L$ *of EG and OGDA (Theorem* 6&7 *[59] and Theorem* 4&7 *[58]).*

Again, the result in Theorem 21 demonstrates the linear convergence of DGDA when solving a general strongly convex-strongly concave min-max optimization problem. Accordingly, if we want to achieve an ϵ-accurate solution, we need to run at most $\mathcal{O}(\kappa \log(1/\epsilon))$ iterations.

Numerical Examples for DGDA

The purpose of this numerical experiment is to support our theoretical results, which form the paper's main contributions. In this section, we compare the performance of the proposed Dissipative gradient descent (DGDA) method with the Extra-gradient (EG), Gradient descent ascent method (GDA), and Optimistic gradient descent ascent (OGDA) methods.

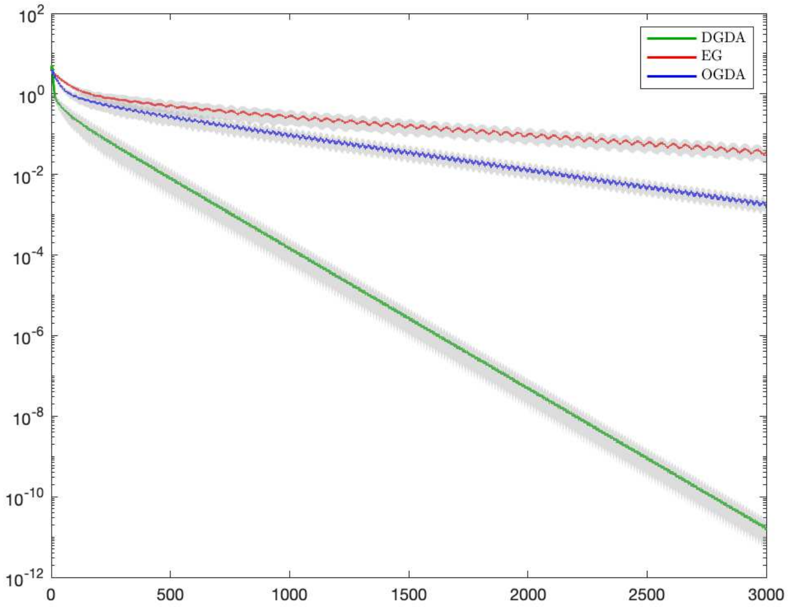

Figure 3-4. Convergence of DGDA, EG, and OGDA in terms of the number of gradient evaluations for the bilinear problem in 3.20. All algorithms converge linearly, and the DGDA method has the best performance.

Bilinear problem

We first consider the following bilinear min-max optimization problem:

$$\min_{x\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} x^T A y \tag{3.20}$$

where $A \in \mathbb{R}^{m\times n}$ is full-rank. The simulation results are illustrated in Figure 3-4 and Figure 3-5. In this experiment, we set the dimension of the problem to $m = n = 10$ and the iterates are initialized at x_0, y_o, which are randomly drawn from the standard uniform distribution on the open interval $(0, 1)$. We plot the errors (distance to saddle points) of DGDA, EG, and OGDA versus the number of gradient evaluations for this problem in Figure 3-4. The solid line and grey-shaded error bars represent the average trajectories and standard deviations of 20 trials, where in each trial the randomly generated matrix A has fixed condition number, i.e., $\kappa = \sigma^2_{\max}(A)/\sigma^2_{\min}(A) = 4$. The key motivation is that all three algorithms commit a

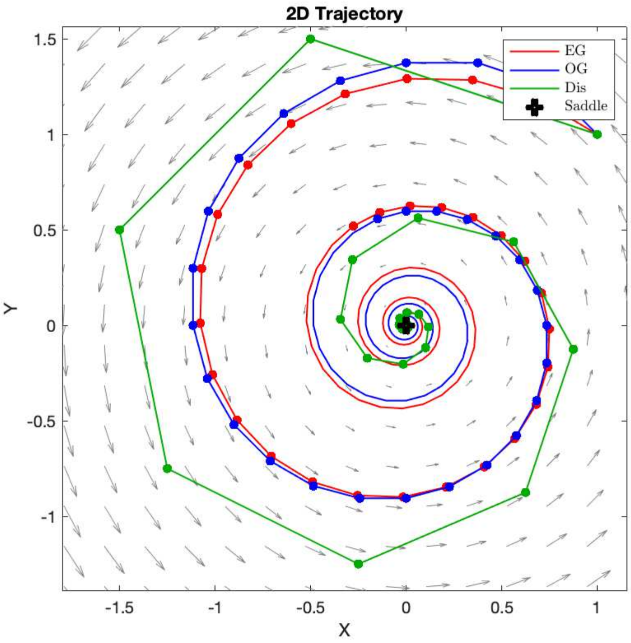

Figure 3-5. Convergence of DGDA, EG, and OGDA in terms of the number of gradient evaluations for the bilinear problem in 3.20. All algorithms converge linearly, and the DGDA method has the best performance.

linear convergence rate as a function of κ^{-1}, and by fixing the condition number we provide an explicit comparison of their convergent speed. We pick the step size for different methods according to theoretical findings. That is, we select $\rho = 1/2$ and $\eta = 1/\delta_{\max}(A)$ for DGDA (Theorem 20), $\eta = 1/4L = 1/4\delta_{\max}(A)$ for EG and OGDA (Theorem 6&7 [59] and Theorem 4&7 [58]). According to the plots, all algorithms converge linearly, and the DGDA method has the best performance.

In Figure 3-5, we plot the sample trajectories of DGDA, EG, and OGDA on a 2d bilinear game, i.e., $m = n = 1$. We can observe that all the three considered algorithms converge linearly to the saddle point $(x^*, y^*) = (\mathbf{0}, \mathbf{0})$, and our proposed algorithm (DGDA) exhibits a faster linear convergence rate.

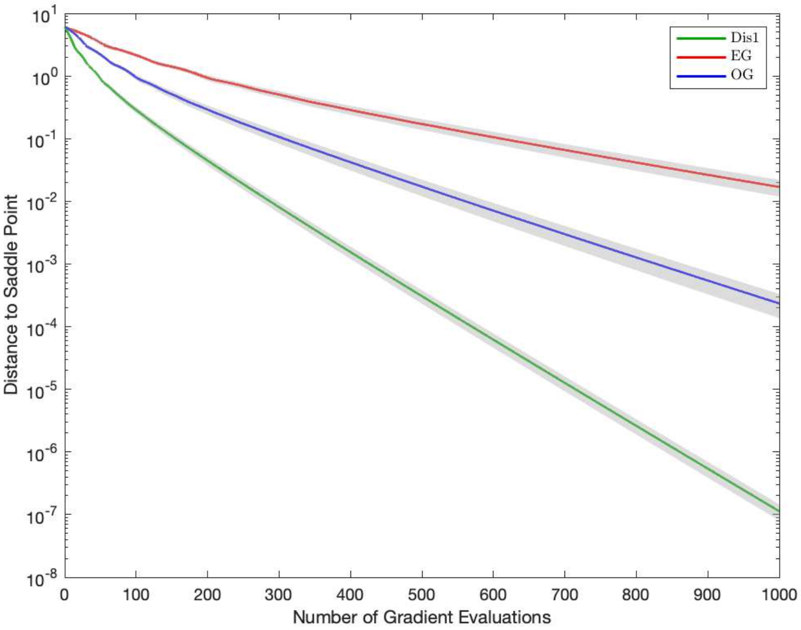

Figure 3-6. Convergence of DGDA, EG, and OGDA in terms of the number of gradient evaluations for the strongly convex-strongly concave problem in 3.21. All algorithms converge linearly, and the DGDA method has the best performance.

Strongly convex-strongly concave problem

In the second numerical example, we focus on a strongly convex-strongly concave quadratic problem of the following form:

$$\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} \frac{1}{2} x^T A x - \frac{1}{2} y^T B y + x^T C y, \tag{3.21}$$

where the matrices satisfy $\mu_A I \preceq A \preceq L_A I$, $\mu_B I \preceq B \preceq L_B I$, $\mu_c^2 I \preceq C^T C \preceq L_C I$ and that problem 3.21 satisfy Assumption 8. In this experiment, we set the dimension of the problem to $n = 50, m = 10$, and the iterates are initialized at x_0, y_o, which are randomly drawn from the standard uniform distribution on the open interval $(0, 1)$. We plot the errors (distance to saddle points) of DGDA, EG, and OGDA versus the number of gradient evaluations for this problem in Figure 3-6. Again, the solid line and grey-shaded error bars represent the average trajectories and standard deviations of 20 trials, where in each trial the randomly generated matrix

$$\begin{bmatrix} A & C \\ -C^T & B \end{bmatrix} \tag{3.22}$$

has a fixed condition number, i.e., $\kappa = L/\mu = 25$. Similarly as in the bilinear problem 3.2, we pick the step size for the DGDA method according to our theoretical finding in Theorem 21. The step sizes for EG and OGDA methods are selected as $\eta = 1/4L$ (Theorem 6&7 [59] and Theorem 4&7 [58]). According to the plots, all algorithms converge linearly, and the DGDA method has the best performance.

3.3 Stochastic approximation for constrained reinforcement learning

A direct application of the above projected and regularized saddle flow dynamic and the discretized DGDA method (3.17) is to solve the C-RL problem in occupancy measure space (3.5), where the Lagrangian function is also bilinear. Specifically, the Lagrangian function for (3.5) in occupancy measure space is:

$$L(\lambda, \mu, v) = \sum_a \lambda_a^T r_a + \sum_i \mu_i (\sum_a \lambda_a^T g_a^i - h^i) + (1-\gamma)\langle q, v\rangle - \sum_{a\in\mathcal{A}} \lambda_a^T (I - \gamma P_a) v, \tag{3.23}$$

where $\mu_i \geq 0$ is the Lagrange multiplier associated with the i^{th} inequality constraint and v is the Lagrange multiplier associated with the equality constraint. Therefore, motivated by the projected and regularized saddle flow dynamics framework, we propose a regularized surrogate for (3.23) via the following augmentation:

$$L(v, \hat{v}, \mu, \hat{\mu}, \lambda, \hat{\lambda}) := \frac{1}{2\rho}\|v - \hat{v}\|^2 + \frac{1}{2\rho}\|\mu - \hat{\mu}\|^2 + L(v, \mu, \lambda) - \frac{1}{2\rho}\|\lambda - \hat{\lambda}\|^2 \tag{3.24}$$

Slater's condition for C-RL and the following Lemma establishes the boundedness of dual decision variables, which naturally provides a closed convex set for projection.

Assumption 9 (Slater's condition for C-RL). *There exists a strictly feasible occupancy measure $\tilde{\lambda} \in \Delta$ of problem (3.5), i.e., there exist some $\psi > 0$ such that*

$$\sum_a \tilde{\lambda}_a^T g_a^i \geq h^i + \psi, \quad i \in [I]$$
$$\sum_{a \in \mathcal{A}} (I - \gamma P_a^T)\tilde{\lambda}_a = (1-\gamma)q,$$

Lemma 23. *[43, Lem.1][Bounded dual variable] Under the assumption 9, the optimal dual variables μ^*, v^* are bounded. Formally, it holds that $\|\mu^*\|_1 \leq \frac{2}{\psi}$ and $\|v^*\|_\infty \leq \frac{1}{1-\gamma} + \frac{2}{(1-\gamma)\psi}$*

Therefore, we propose the following projected saddle flow dynamics to find the saddle points of (3.24), where $\mathcal{U} := \{\mu | \mu \in \mathbb{R}^I_{\geq 0}, \|\mu\|_1 \leq \frac{2}{\psi}\}, \mathcal{V} := \{v | v \in \mathbb{R}^s, \|v^*\|_\infty \leq \frac{1}{1-\gamma} + \frac{2}{(1-\gamma)\psi}\}$ are both closed convex polyhedrons.

$$\begin{aligned}
\dot{v} \ &= \Pi_{\mathcal{V}}\Big[v, \sum_{a \in \mathcal{A}} (I - \gamma P_a^T)\lambda_a - (1-\gamma)q - \frac{1}{\rho}(v - \hat{v})\Big], \\
\dot{\hat{v}} \ &= \Pi_{\mathcal{V}}\Big[\hat{v}, \frac{1}{\rho}(v - \hat{v})\Big], \\
\dot{\mu}_i &= \Pi_{\mathcal{U}}\Big[\mu_i, h^i - \sum_a \lambda_a^T g_a^i - \frac{1}{\rho}(\mu_i - \hat{\mu}_i)\Big], \\
\dot{\hat{\mu}}_i &= \Pi_{\mathcal{U}}\Big[\hat{\mu}, \frac{1}{\rho}(\mu - \hat{\mu})\Big], \\
\dot{\lambda}_a &= \Pi_{\Delta}\Big[\lambda, r_a - (I - \gamma P_a)v + \sum_i \mu_i g_a^i - \frac{1}{\rho}(\lambda_a - \hat{\lambda}_a)\Big], \\
\dot{\hat{\lambda}}_a &= \Pi_{\Delta}\Big[\hat{\lambda}_a, \frac{1}{\rho}(\lambda - \hat{\lambda})\Big],
\end{aligned} \tag{3.25}$$

The following theorem is a direct application of Theorem 17 and Proposition 18, which guarantees (3.25) asymptotically converge to the unique (optimal) saddle point of the C-RL problem (3.5). Then we could recover the optimal policy from the optimal occupancy measure λ^*.

Theorem 24. *Let Assumption 9 hold. Then the projected saddle flow dynamics (3.25) asymptotically converge to some saddle point (λ^*, μ^*, v^*) of $L(\lambda, \mu, v)$, while satisfying $\lambda(t) \in \Delta, \mu(t) \in \mathcal{U}, \forall t$ with proper initialization.*

In the subsequent part, our goal is to expand the proposed continuous-time saddle flow algorithm (3.25) to a model-free setting. To achieve this, we introduce the stochastic-DGDA algorithm that operates without requiring knowledge of the transition kernel. We establish that the S-DGDA algorithm serves as a stochastic approximation of the continuous-time saddle flow dynamics (3.25), leading to almost sure convergence (with probability 1) to the unique saddle point of the C-RL problem.

In many optimization problems, the goal is to find some recursive numerical procedure that sequentially approximates a value of the decision variable x, which minimizes the objective function, e.g., $\dot{x} = h(x)$ or $x^{n+1} = x^n + \alpha^n h(x^n)$. Stochastic approximations attempt to solve the problem when one cannot actually observe $h(x)$, but rather $h(x)$ plus some error or noise. Consider the following projection algorithm:

$$x^{n+1} = \Psi_{\mathcal{G}}\Big[x^n + \alpha^n \Big(h(x^n) + \xi^n\Big)\Big], \tag{3.26}$$

where $\mathcal{G} := \{x : q_i(x) \leq 0, i \in [s]\}$ denotes the constraints and $\{\xi^n\}$ denotes a sequence of random variables. The goal is to generate a sequence $\{x^n\}$ estimate of the optimal value of x when the actual observation has random noise $h(x^n) + \xi^n$. In general, the projection $\Psi_{\mathcal{G}}[x]$ is easy to compute when the constraints are linear; i.e., when $\mathcal{G}$ is a polyhedron. We introduce the following list of standard assumptions for stochastic approximation

Assumption 10 (Stochastic Approximation)**.**

1.1 $h(\cdot)$ *is a continuous function.*

1.2 $\{\alpha^n\}$ *is a sequence of positive real numbers such that* $\alpha^n > 0, \sum_n \alpha^n = \infty, \sum_n (\alpha^n)^2 < \infty$, 1.3 *G is the closure of its interior and is bounded. The $q_i(\cdot), i \in [s]$ are continuously differentiable.*

1.4 *There is a $T > 0$ such that for each $\epsilon > 0$*

$$\lim_n P\{\sup_{j \geq n} \max_{t \leq T} | \sum_{i=m(jT)}^{m(jT+t)-1} \alpha^i \xi^i | \geq \epsilon\} = 0,$$

where $t^n := \sum_{i=0}^{n-1} \alpha^i$ and $m(t) := \max_n\{t^n \leq t\}$ for $t \geq 0$.

Under those standard assumptions for stochastic approximations, the sequence $\{x^n\}$ generated by the projection algorithm (3.26) will converge almost surely to a stable solution to the projected system.

Theorem 25. *[84, Theorem 5.3.1] Assume Assumption 10 hold. Consider the following ODE:*

$$\dot{x} = \Pi_{\mathcal{G}}\Big[x, h(x)\Big]. \tag{3.27}$$

Let x^* denotes an asymptotically stable point of (3.27) with domain of attraction $DA(x^*)$ and x^n generated by (3.26). If $A \in DA(x^*)$ is compact and $x^n \in A$ infinitely often, then x^n converges to x^* almost surely as $n \to \infty$.

Consider the following randomized primal-dual approach proposed in [43, 85], where we assume the presence of a generative model. For a given state action pair (s, a), the generative model provides the next state s' and the reward functions $r(s, a), g^i(s, a)$ to train the policy. Consider the following stochastic approximation for the Lagrangian function (3.23) for a distribution ξ:

$$L^{\xi}(\lambda, \mu, v) = (1 - \gamma)v(s_0) - \sum_{i \in [I]} \mu_i h^i + \tag{3.28}$$
$$\mathbf{1}_{\xi(s,a)>0} \frac{\lambda(s, a)\Big[r(s, a) - v(s) + \gamma v(s') + \sum_{i \in [I]} \mu_i g^i(s, a)\Big]}{\xi(s, a)}$$

where $s_0 \sim q$, $(s,a) \sim \xi$, and the next state $s' \sim P(\cdot|s,a)$. The stochastic approximation $L^{\xi}(\lambda,\mu,v)$ (3.28) is an unbiased estimator for the Lagrangian function (3.23), i.e., $\mathbf{E}_{\xi,P(\cdot|s,a),q}\Big[L^{\xi}(\lambda,\mu,v)\Big] = L(\lambda,\mu,v)$. Using the proposed stochastic approximation of the Lagrangian function, consider the following projection algorithm for solving the C-RL problem in a model-free setting:

$$
\begin{aligned}
v^{n+1} &= \Psi_{\mathcal{V}}\Big[v^n + \alpha^n\Big(\mathbf{1}_{\xi(s,a)>0}\frac{\lambda(s,a)[e(s)-\gamma e(s')]}{\xi(s,a)}(1-\gamma)\mathbf{e}(s_0) - \frac{1}{\rho}(v^n - \hat{v}^n)\Big)\Big],\\
\hat{v}^{n+1} &= \Psi_{\mathcal{V}}\Big[\hat{v}^n + \alpha^n\frac{1}{\rho}(v^n - \hat{v}^n)\Big],\\
\mu_i^{n+1} &= \Psi_{\mathcal{U}}\Big[\mu_i^n + \alpha^n\Big(h^i - \mathbf{1}_{\xi(s,a)>0}\frac{\lambda(s,a)g^i(s,a)}{\xi(s,a)} - \frac{1}{\rho}(\mu_i^n - \hat{\mu}_i^n)\Big)\Big],\\
\hat{\mu}_i^{n+1} &= \Psi_{\mathcal{U}}\Big[\hat{\mu}_i^n + \alpha^n\frac{1}{\rho}(\mu_i^n - \hat{\mu}_i^n)\Big],\\
\lambda_a^{n+1} &= \Psi_{\Delta}\Big[\lambda_a^n + \alpha^n\Big(-\frac{1}{\rho}(\lambda_a^n - \hat{\lambda}_a^n) + \mathbf{1}_{\xi(s,a)>0}\frac{r(s,a)-v(s)+\gamma v(s')+\sum_i \mu_i^n g^i(s,a)}{\xi(s,a)}\Big)\Big],\\
\hat{\lambda}_a^{n+1} &= \Psi_{\Delta}\Big[\hat{\lambda}_a^n + \frac{1}{\rho}(\lambda_a^n - \hat{\lambda}_a^n)\Big],
\end{aligned}
\tag{3.29}
$$

The following Theorem is a direct application of Theorem 25 and 24, which shows the sequence from (3.29) almost surely converges to the optimal solution to the C-RL problem.

Theorem 26. *Assume 9 and 10 hold, as $n \to \infty$, the sequence $\{\lambda^n, v^n, \mu^n\}$ generated by (3.29) almost surely (w.p.1) converge to the optimal solution of the C-RL problem (3.5).*

3.4 Numerical experiments

In this section, we illustrate the effectiveness of our proposed approach using a classical CMDP problem: flow and service control problem in a single-server queue [39]. Specifically, we consider a discrete-time single-server queue with a buffer of finite size L. We assume that, at most, one customer may join the system in a time slot. The state s corresponds to the number of customers in the queue at the beginning of a time slot ($|\mathcal{S}| = L+1$). The service action a is selected from a finite subset A, and the flow action b is selected from a finite subset B. Specifically, for two real numbers satisfying $0 < a_{\min} \leq a_{\max} < 1$, if the queue is non-empty and if the action of the server is $a \in A$, where A is a finite subset of $[a_{\min}, a_{\max}]$, then the service of a customer is successfully completed with probability a. Likewise, for two real numbers satisfying $0 \leq b_{\min} \leq b_{\max} < 1$, if the queue is not full and if the action of the server is $b \in B(s)$, where $B(s)$ is a finite subset of $[b_{\min}, b_{\max}]$, then the probability of having one arrival during this time slot is equal to b. We assume that $0 \in B(x)$ for all x; moreover, when the buffer is full, no arrivals are possible $(B(L) = 0)$. The transition law $P(\cdot|s, a)$ is therefore given by:

$$\begin{cases} a(1-b) & \text{if } 1 \leq x \leq L, y = x-1; \\ ab + (1-a)(1-b) & \text{if } 1 \leq x \leq L, y = x; \\ (1-a)b & \text{if } 0 \leq x < L, y = x+1; \\ 1-(1-a)b & \text{if } y = x = 0; \end{cases}$$

The reward function $r(s, a, b)$ is a real-valued decreasing function that depends only on s, which can be interpreted as a holding cost. The reward function $g^1(s, a, b)$ corresponding to the service rate is assumed to be a decreasing function that depends only on a. It can be interpreted as a higher service success rate having a higher cost. The reward function $g^2(s, a, b)$ corresponding to the flow rate b is assumed to be an increasing function that depends only on b. It can be interpreted as a higher flow rate is more desired.

Suppose we want to solve the optimal policy for C-RL problem (3.5), while satisfying constraints for service and flow. In the following numerical example, we compare the result generated by (3.29) and the ground truth result by directly solving the linear programming 3.5, where we use the transition law stated above. Specifically, we choose $L = 4, A = [0.2, 0.3, 0.5, 0.6, 0.8], B = [0.1, 0.3, 0.5, 0.9, 0]$. The initial distribution q is set as uniform distribution. The reward functions are $r(s) = -s + 5, g^1(a) = -10a + 3, g^2(b) = 10b - 3$.

We compare the cumulative reward function, constraint functions, and output

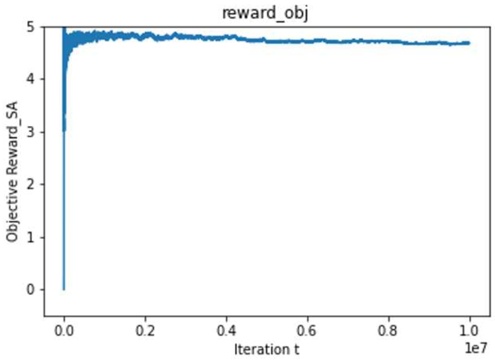

Figure 3-7. objective function space4ex

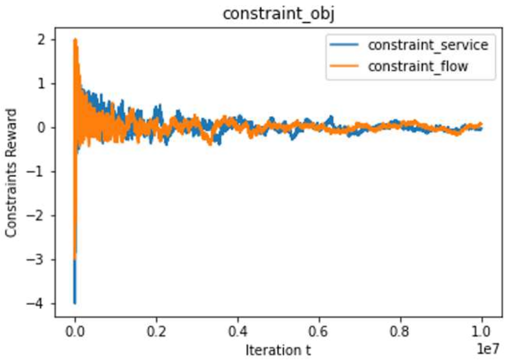

Figure 3-8. constraint functions space4ex

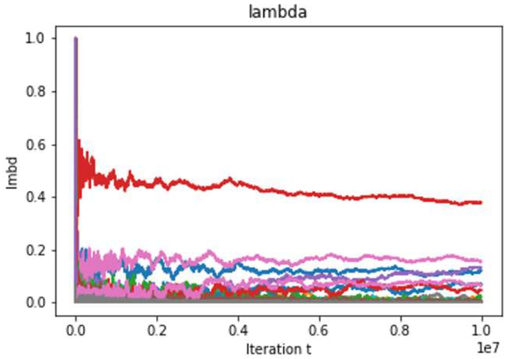

Figure 3-9. occupancy measure λ space4ex

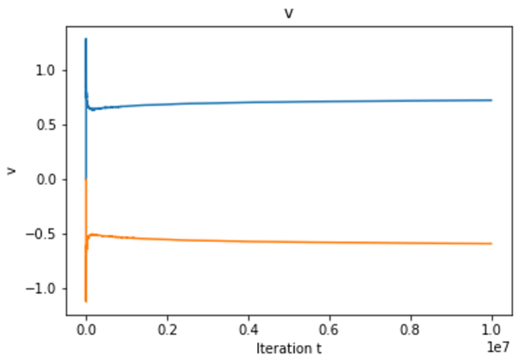

Figure 3-10. dual variable v space4ex

decision variables λ, μ, v with the ground truth result by directly solving the linear programming problem (3.5). Results show that the decision variables converge to the optimal solution while satisfying the constraints for flow and service.

3.5 Appendix

Proof of Theorem 17

We will use the following technical Lemma:

Lemma 27. *For any closed convex set $\mathcal{K} \subset \mathbb{R}^n$ and $a, b \in \mathcal{K}, v \in \mathbb{R}^n$,the inner product*

$$\langle b - a, v - \Pi_{\mathcal{K}}[a, v] \rangle \leq 0$$

Proof: According to [86, Sec.0.6, Cor.1], we have the following variational inequality holds:

$$\langle b - \Psi_{\mathcal{K}}[c], c - \Psi_{\mathcal{K}}[c] \rangle \leq 0, \forall b \in \mathcal{K}, \forall c \in \mathbb{R}^n.$$

The rest follows from [87, Lem.4]

Using this lemma, the proof of Theorem 17 essentially follows from [79, Thm.9].

Proof of Theorem 20

To begin with, let's assume $A \in \mathbb{R}^{m \times m}$ is a square matrix, which we will relax to a general non-square matrix shortly. Applying the updates 3.17 to $f(x, y) = x^T Ay$ and denote $z = [x, y]^T, \hat{z} = [\hat{x}, \hat{y}]^T$ yields:

$$\begin{bmatrix} z_{k+1} \\ \hat{z}_{k+1} \end{bmatrix} = \begin{bmatrix} z_k - \eta M z_k - \rho(z_k - \hat{z}_k) \\ \hat{z}_k - \rho(\hat{z}_k - z_k) \end{bmatrix} = \begin{bmatrix} (1-\rho)I - \eta M & \rho I \\ \rho I & (1-\rho)I \end{bmatrix} \begin{bmatrix} z_k \\ \hat{z}_k \end{bmatrix}, \tag{3.30}$$

where

$$M = \begin{bmatrix} \mathbf{0} & A \\ -A^T & \mathbf{0} \end{bmatrix}$$

According to [59, Lemma 7]

$$\mathrm{Sp}(M) = \{\pm i\delta | \delta^2 \in \mathrm{Sp}(AA^T)\}$$

Since M is a normal matrix and diagonalizable, this simplifies the spectral analysis

$$\nabla H_{\eta,\rho} := \begin{bmatrix} (1-\rho)I - \eta M & \rho I \\ \rho I & (1-\rho)I \end{bmatrix} = \begin{bmatrix} U^{-1} & \mathbf{0} \\ \mathbf{0} & U^{-1} \end{bmatrix} \begin{bmatrix} (1-\rho)I - \eta \mathbf{\Lambda} & \rho I \\ \rho I & (1-\rho)I \end{bmatrix} \begin{bmatrix} U & \mathbf{0} \\ \mathbf{0} & U \end{bmatrix}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, ..., \lambda_m)$ and $\lambda_j = \pm i\delta_j \in \mathrm{Sp}(M), j = \{1, ..., m\}$. In order to show linear convergence, we want to show that $|\mu_j|^2 < 1, \forall j \in [m]$, where μ_j are the eigenvalues of $\nabla H_{\eta,\rho}$,

$$\begin{aligned} \mu_j =& \frac{1}{2}(2 - 2\rho - \eta\lambda_j \pm \sqrt{\eta^2\lambda_j^2 + 4\rho^2}) \\ =& 1 - \rho \pm i(\frac{1}{2}\eta\delta_j) \pm \frac{1}{2}\sqrt{4\rho^2 - \eta^2\delta_j^2} \end{aligned} \tag{3.31}$$

For complex number $|c|^2 = c\bar{c}$, therefore the magnitude of eigenvalues $|\mu_j|^2$ are given by,

$$
\begin{aligned}
|\mu_j|^2 =& \left(1-\rho+i\frac{1}{2}\eta\delta_j \pm \frac{1}{2}\sqrt{4\rho^2-\eta^2\delta_j^2}\right)\overline{\left(1-\rho-i\frac{1}{2}\eta\delta_j \pm \frac{1}{2}\sqrt{4\rho^2-\eta^2\delta_j^2}\right)} \\
=& (1-\rho)^2 + \frac{1}{4}\eta^2\delta_j^2 + \frac{1}{4}(4\rho^2-\eta^2\delta_j^2) \pm (1-\rho)\Re(\sqrt{4\rho^2-\eta^2\delta_j^2}) \\
& \pm \frac{i}{4}\eta\delta_j\overline{\sqrt{4\rho^2-\eta^2\delta_j^2}} \mp \frac{i}{4}\eta\delta_j\sqrt{4\rho^2-\eta^2\delta_j^2} \\
=& \begin{cases} 2\rho^2-2\rho+1 \pm (1-\rho)\sqrt{4\rho^2-\eta^2\delta_j^2}, & \text{if } 4\rho^2-\eta^2\delta_j^2 \geq 0 \\ 2\rho^2-2\rho+1 \pm \frac{1}{2}\eta\delta_j\sqrt{\eta^2\delta_j^2-4\rho^2}, & \text{if } \eta^2\delta_j^2-4\rho^2 \geq 0 \end{cases}
\end{aligned} \tag{3.32}
$$

Suppose that we choose $\eta = \frac{2\rho}{\delta_{\max}}$ and $0 < \rho < 1$, this implies $\eta \leq \frac{2\rho}{\delta_j}$ and therefore $4\rho^2 - \eta^2\delta_j^2 \geq 0$. From (3.32), in this case we have,

$$
2\rho^2-2\rho+1-(1-\rho)\sqrt{4\rho^2-\eta^2\delta_j^2} \leq 2\rho^2-2\rho+1+(1-\rho)\sqrt{4\rho^2-\eta^2\delta_j^2}
$$

Substituting $\eta = \frac{2\rho}{\delta_{\max}}$ yields,

$$
|\mu_j|^2 \leq 2\rho^2-2\rho+1+(1-\rho)\sqrt{4\rho^2-\frac{4\rho^2}{\delta_{\max}^2}\delta_j^2} \ , \forall j \in [m]. \tag{3.33}
$$

Considering the worst-case of $|\mu_j|^2$, and its maximum is reached at $\delta_j = \delta_{\min}$,

$$
\begin{aligned}
|\mu_j|^2 &\leq 2\rho^2-2\rho+1+(1-\rho)2\rho\sqrt{1-\frac{\delta_{\min}^2}{\delta_{\max}^2}} \\
&= \rho^2\left(2-2\sqrt{1-\frac{\delta_{\min}^2}{\delta_{\max}^2}}\right) - \rho\left(2-2\sqrt{1-\frac{\delta_{\min}^2}{\delta_{\max}^2}}\right)+1 \\
&\leq 1-\frac{1}{2}(1-\sqrt{1-\frac{\delta_{\min}^2}{\delta_{\max}^2}}) = \frac{1}{2}+\frac{1}{2}\sqrt{1-\frac{\delta_{\min}^2}{\delta_{\max}^2}} \ , \forall j \in [m]
\end{aligned}
$$

where the last inequality comes from selecting optimal $\rho = \frac{1}{2}$ of a quadratic polynomial of ρ. Using the fact that $\sqrt{1-x} \leq 1 - x/2$, we have

$$
|\mu_j|^2 \leq 1 - \frac{1}{4}\frac{\delta_{\min}^2}{\delta_{\max}^2} = 1 - \frac{1}{4}\frac{\sigma_{\min}^2(A)}{\sigma_{\max}^2(A)} \ , \forall j \in [m]. \tag{3.34}
$$

Note: We could also choose $\eta = \frac{2\rho}{\delta_{\min}}$ such that $\eta^2\delta_j^2 - 4\rho^2 \geq 0$. And we could construct a similar linear convergence rate by repeating the above process. However, in practice, we found that the step sizes $\eta = \frac{2\rho}{\delta_{\max}}, \rho = 1/2$ always perform better in numerical experiments. Therefore, we choose this pair of step sizes by default.

Proof of Theorem 21

The proof relies on the application of dissipativity theory to construct Lyapunov functions and establish linear convergence. For more detailed information, refer to [88].

According to [88], a linear dynamical system of the form:

$$\xi_{k+1} = A\xi_k + Bw_k \tag{3.35}$$

Here, $\boldsymbol{\xi} \in \mathbb{R}^{n_\xi}$ is the state, $w_k \in \mathbb{R}^{n_w}$ is the input, A is the state transition matrix and B is the input matrix. Suppose that there exist a (Lyapunov) function V, satisfying $V(\boldsymbol{\xi}) \geq 0, \forall \xi \in \mathbb{R}^{n_\xi}$, some $0 \leq \alpha < 1$ and a supply rate function $S(\xi_k, w_k) \leq 0, \forall k$ such that

$$V(\boldsymbol{\xi}_{k+1}) - \alpha^2 V(\boldsymbol{\xi}_k) \leq S(\xi_k, w_k). \tag{3.36}$$

This dissipation inequality (3.36) implies that $V(\boldsymbol{\xi}_{k+1}) \leq \alpha^2 V(\boldsymbol{\xi}_k)$, and the state will approach a minimum value ate equilibrium no slower than the linear rate α^2. The flowing theorem states how to construct the dissipation inequality (3.36) by solving a semidefinite programming problem.

Theorem 28. *[88][Theorem 2] Consider the following quadratic supply rate with $X \in \mathbb{R}^{(n_\xi+n_w)\times(n_\xi+n_w)}$ and $X^T = X$*

$$S(\xi, w) := \begin{bmatrix} \xi \\ w \end{bmatrix}^T X \begin{bmatrix} \xi \\ w \end{bmatrix}. \tag{3.37}$$

If there exists matrix $P \in \mathbb{R}^{n_\xi \times n_\xi}$ with $P \succeq 0$ such that

$$\begin{bmatrix} A^T PA - \alpha^2 P & A^T PB \\ B^T PA & B^T PB \end{bmatrix} - X \leq 0, \tag{3.38}$$

then the dissipation inequality holds for all trajectories of (3.35) with $V(\boldsymbol{\xi}) = \boldsymbol{\xi}^T P\xi$.

A major benefit of the proposed constructive dissipation approach is that it replace the trouble some component of a dynamical system (e.g. the gradient term

$\nabla_x f(x_k, y_k)$) by a quadratic constraint on its inputs and outputs that is always satisfied, namely integral quadratic constraints. This leads to a two-step novel approach to the convergence analysis of optimization algorithms.

1. Choose a proper quadratic supply rate function S such that $S(\xi_k, w_k) \leq 0, \forall k$.
2. Solve the Linear Matrix Inequality (3.38) to obtain a storage function V and finding the linear convergence rate α.

Use the following concatanate notation for the DGDA update (3.17), where $z = [x, y]^T, \hat{z} = [\hat{x}, \hat{y}]^T$ and $F(z_k) = (\nabla_x f(x_k, y_k; -\nabla_y f(x_k, y_k)$, we rewrite (3.17) as in the form of (3.35):

$$\begin{bmatrix} z_{k+1} \\ \hat{z}_{k+1} \end{bmatrix} = \begin{bmatrix} z_k - \eta F(z_k) - \rho(z_k - \hat{z}_k) \\ \hat{z}_k - \rho(\hat{z}_k - z_k) \end{bmatrix} = \begin{bmatrix} 1-\rho & \rho \\ \rho & 1-\rho \end{bmatrix} \begin{bmatrix} z_k \\ \hat{z}_k \end{bmatrix} + \begin{bmatrix} -\eta \\ 0 \end{bmatrix} w_k \tag{3.39}$$

where $w_k = F(z_k)$.

According to [88] eq(7) and Assumption 8, we have

$$S(\xi_k, w_k) = \begin{bmatrix} z_k \\ \hat{z}_k \\ F(z_k) \end{bmatrix}^T \begin{bmatrix} 2\mu L I & 0 & (-\mu + L)I \\ 0 & 0 & 0 \\ (-\mu + L)I & 0 & 2I \end{bmatrix} \begin{bmatrix} z_k \\ \hat{z}_k \\ F(z_k) \end{bmatrix} \leq 0 \tag{3.40}$$

as a proper quadratic supply rate function $S(\xi_k, w_k) \leq 0$.

Then, it reduces to find a matrix $P \in \mathbf{R}^{2\times 2} \succeq 0$, $\alpha \in [0, 1)$ such that (3.38) is satisfied, where the problem parameters are given by

$$A = \begin{bmatrix} 1-\rho & \rho \\ \rho & 1-\rho \end{bmatrix}, B = \begin{bmatrix} -\eta \\ 0 \end{bmatrix}, X = \begin{bmatrix} 2\mu L I & 0 & (-\mu + L)I \\ 0 & 0 & 0 \\ (-\mu + L)I & 0 & 2I \end{bmatrix}. \tag{3.41}$$

Consequently, we are solving a LMI problem of dimension 3 by 3, with design parameters $P \in \mathbf{R}^{2\times 2}$, $\alpha^2 \in [0, 1)$, ρ and η. Motivated by the step size in the bilinear case Theorem 20, we simplify the process by choosing $\rho = 0.5$. And one set of feasible solutions is given by

$$\eta = \frac{1}{L + \mu}, \alpha^2 = \frac{3L^2 + 2L\mu + 3\mu^2 + \sqrt{(L+\mu)^4 + 16L^2\mu^2}}{4(L+\mu)^2}, P = \begin{bmatrix} (L+\mu)^2 & 0 \\ 0 & (L+\mu)^2 \end{bmatrix}, \tag{3.42}$$

where

$$\alpha^2 = 1 - \frac{\mu}{L} + \mathcal{O}\big((\frac{\mu}{L})^2\big) \tag{3.43}$$

3.6 Conclusion

In this chapter, we explore a data-driven approach for online decision-making of dynamical systems, especially when we lack precise knowledge of the system dynamics. To tackle this challenge, we adopt the framework of constrained reinforcement learning. In this setup, an agent aims not only to maximize its expected total reward but also to satisfy secondary cumulative rewards constraints while interacting with an unknown environment and receiving information over time.

The core of our investigation involves formulating the constrained reinforcement learning problem as a min-max optimization problem in the occupancy measure space. However, the presence of a bilinear Lagrangian function makes vanilla gradient descent ascent (GDA) methods ineffective, leading to convergence issues. To overcome this, we introduce a novel solution: the Dissipative GDA algorithm. This new first-order approach demonstrates strong convergence properties. We compare its performance with established methods like Extra-Gradient (EG) and Optimistic GDA (OGDA), both of which also exhibit linear convergence under similar conditions.

Our approach proves particularly effective in bilinear problems, offering a better estimate of the linear convergence rate compared to EG and OGDA. Furthermore, in scenarios that are strongly convex-strongly concave and meet the condition $\kappa \geq 2$, the DGDA algorithm provides a more accurate estimate of convergence compared to EG and OGDA.

Moving forward, we apply the stochastic DGDA algorithm to address the constrained reinforcement learning problem within the occupancy measure space.

This application showcases the algorithm's ability to achieve global asymptotic convergence concerning occupancy measure and its capacity to recover the optimal policy. To validate our method's practical utility, we employ it in a case study involving a discrete-time single-server queue with a constrained buffer size problem. This example effectively highlights the strengths of our proposed approach.

Chapter 4

Conclusions

In conclusion, this thesis explores the exciting landscape of online decision-making algorithms for dynamical systems, leveraging tools from control theory, optimization, and learning. The growing availability and speed of data sources have presented both new opportunities and challenges in this field, prompting the need for real-time decision-making capabilities that align with the dynamic nature of systems.

The research presents two distinct online decision-making frameworks based on the availability of system dynamic information. In the first scenario, where the system's behavior can be captured by ordinary differential equations using a state-space model, a time-varying convex optimization framework is introduced. This framework efficiently combines motion planning and control, enabling the design of control signals that guide the dynamical system to asymptotically track optimal trajectories defined through constrained time-varying optimization problems. By effectively transforming the nonlinear dynamical system into an optimization algorithm, the method achieves global exponential asymptotic convergence to the minimizer of the time-varying optimization problem, subject to sufficient regularity assumptions.

In the second scenario, when system dynamics remain unknown, the thesis adopts a data-driven approach known as constrained reinforcement learning. This approach tackles sequential decision-making problems, where an agent seeks to maximize its expected total reward while interacting with an unfamiliar environment and receiving sequential information over time. The constrained reinforcement learning framework incorporates safety constraints or conflicting requirements during the learning process by introducing secondary expected cumulative rewards. To overcome the limitations often faced in constrained reinforcement learning problems, the thesis proposes a novel first-order stochastic gradient descent-ascent (GDA) algorithm: the stochastic dissipative GDA algorithm. This powerful algorithm exhibits remarkable properties, almost surely converging to the optimal occupancy measure and optimal policy. By overcoming the challenges of policy oscillation and convergence to suboptimal policies frequently encountered in C-RL problems, the stochastic dissipative GDA algorithm significantly enhances the performance and efficiency of the constrained reinforcement learning process.

In essence, this thesis contributes to advancing the field of online decision-making algorithms by offering innovative methodologies for dynamical systems, with applications in various domains, such as robotics, control systems, and reinforcement learning. The proposed frameworks and algorithms pave the way for more robust and adaptive decision-making in real-time scenarios, aligning with the dynamic nature of modern data streams and dynamic environments. As data sources continue to grow in ubiquity and speed, the research findings in this thesis hold great promise in addressing the evolving challenges of decision-making in dynamic systems.

Bibliography

[1] J.-C. Latombe, “Motion Planning: A Journey of Robots, Molecules, Digital Actors, and Other Artifacts,” *The International Journal of Robotics Research*, vol. 18, no. 11, pp. 1119–1128, 1999.

[2] D. González, J. Pérez, V. Milanés, and F. Nashashibi, “A review of motion planning techniques for automated vehicles,” *IEEE Transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[3] M. Elbanhawi and M. Simic, “Sampling-based robot motion planning: A review,” *Ieee access*, vol. 2, pp. 56–77, 2014.

[4] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[5] E. W. Dijkstra, “A note on two problems in connexion with graphs,” in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, 2022, pp. 287–290.

[6] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[7] A. Stentz, “Optimal and efficient path planning for unknown and dynamic environments,” Carnegie-Mellon Univ Pittsburgh Pa Robotics Inst, Tech. Rep., 1993.

[8] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.

[9] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[10] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[11] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[12] S. R. Lindemann and S. M. LaValle, "Current issues in sampling-based motion planning," in *Robotics Research. The Eleventh International Symposium: With 303 Figures*. Springer, 2005, pp. 36–54.

[13] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[14] L. T. Biegler, "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation," *Computers & chemical engineering*, vol. 8, no. 3-4, pp. 243–247, 1984.

[15] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2020.

[16] J. Guthrie, M. Kobilarov, and E. Mallada, "Closed-form Minkowski sum approximations for efficient optimization-based collision avoidance," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 3857–3864.

[17] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone, “Robust Online Motion Planning Via Contraction Theory and Convex Optimization,” *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5883–5890, 2017.

[18] J. Guthrie, “Novel representations of semialgebraic sets arising in planning and control,” Ph.D. dissertation, Electrical and Computer Engineering, Johns Hopkins University, 10 2022.

[19] Z. E. Nelson and E. Mallada, “An integral quadratic constraint framework for real-time steady-state optimization of linear time-invariant systems,” in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 597–603.

[20] L. S. P. Lawrence, J. W. Simpson-Porco, and E. Mallada, “Linear-convex optimal steady-state control,” *IEEE Transactions on Automatic Control*, pp. 1–1, 2020.

[21] S. Menta, A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, “Stability of dynamic feedback optimization with applications to power systems,” in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2018, pp. 136–143.

[22] M. Colombino, E. Dall’Anese, and A. Bernstein, “Online optimization as a feedback controller: Stability and tracking,” *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 422–432, 2019.

[23] G. Bianchin, J. Cortés, J. I. Poveda, and E. Dall’Anese, “Time-varying optimization of lti systems via projected primal-dual gradient flows,” *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 474–486, 2021.

[24] A. Hauswirth, S. Bolognani, G. Hug, and F. Drfler, “Timescale separation in autonomous optimization,” *IEEE Transactions on Automatic Control*, 2020.

[25] T. Zheng, J. Simpson-Porco, and E. Mallada, "Implicit trajectory planning for feedback linearizable systems: A time-varying optimization approach," in *2020 American Control Conference (ACC)*, 2020, pp. 4677–4682.

[26] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2018.

[27] P. Lin, W. Ren, and J. A. Farrell, "Distributed continuous-time optimization: nonuniform gradient gains, finite-time convergence, and convex constraint set," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2239–2253, 2016.

[28] S. Sun and W. Ren, "Distributed continuous-time optimization with time-varying objective functions and inequality constraints," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 5622–5627.

[29] S. Rahili and W. Ren, "Distributed continuous-time convex optimization with time-varying cost functions," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1590–1605, 2016.

[30] B. Huang, Y. Zou, Z. Meng, and W. Ren, "Distributed time-varying convex optimization for a class of nonlinear multiagent systems," *IEEE Transactions on Automatic control*, vol. 65, no. 2, pp. 801–808, 2019.

[31] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 2032–2048, 2020.

[32] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, "Prediction-correction interior-point method for time-varying convex optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1973–1986, 2017.

[33] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, “A class of prediction-correction methods for time-varying convex optimization,” *IEEE Transactions on Signal Processing*, vol. 64, no. 17, pp. 4576–4591, 2016.

[34] A. Simonetto and E. Dall’Anese, “Prediction-correction algorithms for time-varying constrained optimization,” *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5481–5494, 2017.

[35] T. Zheng, J. W. Simpson-Porco, and E. Mallada, “Implicit trajectory planning for feedback linearizable systems: A time-varying optimization approach,” in *American Control Conference (ACC)*, 7 2020, pp. 4677–4682. [Online]. Available: http://mallada.ece.jhu.edu/pubs/2020-ACC-ZSM.pdf

[36] A. Isidori, *Nonlinear control systems*. Springer Science & Business Media, 2013.

[37] S. Mannor and N. Shimkin, “A geometric approach to multi-criterion reinforcement learning,” *The Journal of Machine Learning Research*, vol. 5, pp. 325–360, 2004.

[38] J. Achiam, D. Held, A. Tamar, and P. Abbeel, “Constrained policy optimization,” in *International conference on machine learning*. PMLR, 2017, pp. 22–31.

[39] E. Altman, *Constrained Markov decision processes: stochastic modeling*. Routledge, 1999.

[40] S. Paternain, L. Chamon, M. Calvo-Fullana, and A. Ribeiro, “Constrained reinforcement learning has zero duality gap,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[41] A. Castellano, H. Min, J. Bazerque, and E. Mallada, “Reinforcement learning with almost sure constraints,” *arXiv preprint arXiv:2112.05198*, 2021.

[42] Y. Chen, J. Dong, and Z. Wang, "A primal-dual approach to constrained markov decision processes," *arXiv preprint arXiv:2101.10895*, 2021.

[43] Q. Bai, A. S. Bedi, M. Agarwal, A. Koppel, and V. Aggarwal, "Achieving zero constraint violation for constrained reinforcement learning via primal-dual approach," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 3682–3689.

[44] M. Calvo-Fullana, S. Paternain, L. F. Chamon, and A. Ribeiro, "State augmented constrained reinforcement learning: Overcoming the limitations of learning with rewards," *arXiv preprint arXiv:2102.11941*, 2021.

[45] T. Liu, R. Zhou, D. Kalathil, P. Kumar, and C. Tian, "Learning policies with zero or bounded constraint violation for constrained mdps," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 183–17 193, 2021.

[46] D. Ding, K. Zhang, T. Basar, and M. Jovanovic, "Natural policy gradient primal-dual method for constrained markov decision processes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8378–8390, 2020.

[47] N. Golowich, S. Pattathil, C. Daskalakis, and A. Ozdaglar, "Last iterate is slower than averaged iterate in smooth convex-concave saddle point problems," in *Conference on Learning Theory*. PMLR, 2020, pp. 1758–1784.

[48] J. Abernethy, K. A. Lai, and A. Wibisono, "Last-iterate convergence rates for min-max optimization: Convergence of hamiltonian gradient descent and consensus optimization," in *Algorithmic Learning Theory*. PMLR, 2021, pp. 3–47.

[49] D. Ding, C.-Y. Wei, K. Zhang, and A. Ribeiro, "Last-iterate convergent policy gradient primal-dual methods for constrained mdps," *arXiv preprint arXiv:2306.11700*, 2023.

[50] G. M. Korpelevich, “The extragradient method for finding saddle points and other problems,” *Matecon*, vol. 12, pp. 747–756, 1976.

[51] L. D. Popov, “A modification of the arrow-hurwicz method for search of saddle points,” *Mathematical notes of the Academy of Sciences of the USSR*, vol. 28, pp. 845–848, 1980.

[52] A. Castellano, H. Min, E. Mallada, and J. A. Bazerque, “Reinforcement learning with almost sure constraints,” in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 559–570.

[53] T. Zheng, P. You, and E. Mallada, “Constrained reinforcement learning via dissipative saddle flow dynamics,” *arXiv preprint*, 2022.

[54] T. Moskovitz, B. O’Donoghue, V. Veeriah, S. Flennerhag, S. Singh, and T. Zahavy, “Reload: Reinforcement learning with optimistic ascent-descent for last-iterate convergence in constrained mdps,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 25 303–25 336.

[55] P. Tseng, “On linear convergence of iterative methods for the variational inequality problem,” *Journal of Computational and Applied Mathematics*, vol. 60, no. 1-2, pp. 237–252, 1995.

[56] T. Liang and J. Stokes, “Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks,” in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 907–915.

[57] G. Gidel, H. Berard, G. Vignoud, P. Vincent, and S. Lacoste-Julien, “A variational inequality perspective on generative adversarial networks,” *arXiv preprint arXiv:1802.10551*, 2018.

[58] A. Mokhtari, A. Ozdaglar, and S. Pattathil, “A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 1497–1507.

[59] W. Azizian, I. Mitliagkas, S. Lacoste-Julien, and G. Gidel, “A tight and unified analysis of gradient-based methods for a whole spectrum of differentiable games,” in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 2863–2873.

[60] A. Mokhtari, A. E. Ozdaglar, and S. Pattathil, “Convergence rate of o(1/k) for optimistic gradient and extragradient methods in smooth convex-concave saddle point problems,” *SIAM Journal on Optimization*, vol. 30, no. 4, pp. 3230–3251, 2020.

[61] E. Gorbunov, N. Loizou, and G. Gidel, “Extragradient method: O (1/k) last-iterate convergence for monotone variational inequalities and connections with cocoercivity,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 366–402.

[62] E. Gorbunov, A. Taylor, and G. Gidel, “Last-iterate convergence of optimistic gradient method for monotone variational inequalities,” *arXiv preprint arXiv:2205.08446*, 2022.

[63] C.-Y. Wei, C.-W. Lee, M. Zhang, and H. Luo, “Linear last-iterate convergence in constrained saddle-point optimization,” *arXiv preprint arXiv:2006.09517*, 2020.

[64] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng, “Training gans with optimism,” *arXiv preprint arXiv:1711.00141*, 2017.

[65] Z. Lin, L. Wang, Z. Han, and M. Fu, “Distributed formation control of multi-agent systems using complex laplacian,” *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1765–1777, 2014.

[66] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer Science & Business Media, 2013, vol. 10.

[67] R. M. Murray, M. Rathinam, and W. Sluis, “Differential flatness of mechanical control systems: A catalog of prototype systems,” in *ASME international mechanical engineering congress and exposition*. Citeseer, 1995.

[68] J. P. Hespanha, *Linear systems theory*. Princeton university press, 2018.

[69] G. Oriolo, A. De Luca, and M. Vendittelli, “Wmr control via dynamic feedback linearization: design, implementation, and experimental validation,” *IEEE Transactions on control systems technology*, vol. 10, no. 6, pp. 835–852, 2002.

[70] P. Martin, P. Rouchon, and R. M. Murray, “Flat systems, equivalence and trajectory generation,” Ph.D. dissertation, Optimization and Control, 2006.

[71] J. Levine, *Analysis and control of nonlinear systems: A flatness-based approach*. Springer Science & Business Media, 2009.

[72] F. Bullo and A. D. Lewis, *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*. Springer, 2019, vol. 49.

[73] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[74] T. Rusten and R. Winther, “A preconditioned iterative method for saddlepoint problems,” *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 3, pp. 887–904, 1992.

[75] J. Borwein and A. S. Lewis, *Convex analysis and nonlinear optimization: theory and examples*. Springer Science & Business Media, 2010.

[76] J. Gauvin, "A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming," *Mathematical Programming*, vol. 12, no. 1, pp. 136–138, 1977.

[77] O. Arslan and D. E. Koditschek, "Exact robot navigation using power diagrams," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1–8.

[78] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[79] P. You and E. Mallada, "Saddle flow dynamics: Observable certificates and separable regularization," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 4817–4823.

[80] A. Cherukuri, E. Mallada, and J. Cortés, "Asymptotic convergence of constrained primal–dual dynamics," *Systems & Control Letters*, vol. 87, pp. 10–15, 2016.

[81] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[82] D. Pfau and O. Vinyals, "Connecting generative adversarial networks and actor-critic methods," *arXiv preprint arXiv:1610.01945*, 2016.

[83] J. Zhang, P. Xiao, R. Sun, and Z. Luo, "A single-loop smoothed gradient descent-ascent algorithm for nonconvex-concave min-max problems," *Advances in neural information processing systems*, vol. 33, pp. 7377–7389, 2020.

[84] H. J. Kushner and D. S. Clark, *Stochastic approximation methods for constrained and unconstrained systems*. Springer Science & Business Media, 2012, vol. 26.

[85] M. Wang, "Randomized linear programming solves the markov decision problem in nearly linear (sometimes sublinear) time," *Mathematics of Operations Research*, vol. 45, no. 2, pp. 517–546, 2020.

[86] J.-P. Aubin and A. Cellina, *Differential inclusions: set-valued maps and viability theory*. Springer Science & Business Media, 2012, vol. 264.

[87] E. Mallada and F. Paganini, "Stability of node-based multipath routing and dual congestion control," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 1398–1403.

[88] B. Hu and L. Lessard, "Dissipativity theory for nesterov's accelerated method," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1549–1557.