Geometry and Dynamical Systems in Machine Learning and Control

Thesis by

Victor D. Dorobantu

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

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2023
Defended May 25, 2023

Victor D. Dorobantu
ORCID: 0000-0002-2797-7802

ACKNOWLEDGEMENTS

Like some instances of describing a set via a description of its complement, it might be easier for me to describe all the people who *haven't* helped make this thesis a reality than to describe all the people who *have*. Nevertheless, I'll attempt the latter.

Thank you to the Computing and Mathematical Sciences department at Caltech, for trusting me educating me, connecting me with all sorts of brilliant people, and for pointing me in the direction of applied math. I can't believe all the things I know now, especially all the things that 2017 me never thought I would learn. Thank you to **Venkat Chandrasekaran** and **Andrew Stuart** for spearheading my dive into applied math. Thank you to **Walter Kortschak**, **Adam Wierman**, and **Chris Umans** for the generous support and flexibility you afforded me to pursue what inspired me. Thank you to **Riley Murray** and **Natalie Bernat** for your advocacy and for making me (and my whole class) feel welcome.

A very broad thank you to the **Yue Crew**[1] and my unofficial, adoptive **AMBER Lab**. It was a joy to get to know all of you as researchers and as people. Your voracious appetites to learn, thought provoking conversations, and affinity to laughs will be the most memorable parts of my experience here. Thank you to **Hoang Le** for being my mentor early on, and thank you to **Charlotte Borcherds** for putting up with my half-baked ideas and pushing our work forward nonetheless. Thank you to **Ryan Cosner** for jumping head-first into the weird world of sampled-data control and helping us all find our ways. Thank you to **Sarah Dean** for such smooth remote collaboration, and to **Jason Choi**, **Fernando Castañeda**, and **Bike Zhang** for the years of chatting and learning.

Thank you to **Yousuf Soliman** and **Peter Schröder** for the truly absurd amount of geometric insights you offered. Research can have a very high activation energy and you two were so often the catalysts. Thank you to **Ivan Dario Jimenez Rodriguez** and **Albert Li** for all the fantastic ideas and the nerd snipes. My perspectives on geometry, generative modeling, and robotic manipulation would be woefully insufficient without all of our conversations. I wouldn't understand geometry the way that I do without the many, many whiteboard sessions with **Noel Csomay-Shanklin**; thank you for always agreeing to study everything just for the fun of it, but also for knowing how to turn what we learned into actual research. Thank you

[1] I am still a little disappointed that that is not the official name.

to **Kamyar Azizzadenesheli** for helping me find the math I love in my work and for always being willing to dive into the details. Thank you to **John Dabiri** for agreeing to come along my whirlwind of a research journey at some of its most critical stages.

Thank you to **Keenan Crane** and **Steve Brunton** for teaching me almost everything I know about differential geometry and partial differential equations, respectively. Perhaps most impressively, you taught me all of this without even knowing you were teaching me; thank you for all of the time and effort you put into your videos.

Thank you to **Kristján Eldjárn Hjörleifsson** for all the Kitchen Companionship and all the Turtle Time. All the extra time I had with you here was a delight. You changed the way I see and understand the world for the better. I hope we both continue deep living and deep learning. Also, while I'm at it, thank you to the **Caltech Turtles**.

Thank you to **Andrew Taylor** for just about everything in my experience here. Thank you for understanding analysis with me, teaching me most of what I know about nonlinear control, and bringing me along a wild arc of research. Thank you for the music, the food, the drinks, the games, the national parks, the commiseration, the laughs, and of course, the traditions with **Noosha Razavian**.

I cannot thank my advisor **Yisong Yue** enough for making my PhD as smooth as I could ask for. Your former student Jialin Song was absolutely right; "trust Yisong" is great advice. The dedication you showed not just to the research successes of your students but to their senses of belonging had (and continues to have) a real impact. My experience in your group will inspire and shape how I hope to work with future students of my own. Thank you for helping me find my research vision, for giving me autonomy and giving me guidance, and for helping me make sense of the future.

Finally, thank you to my family, **Mariana**, **Mihai**, **Andrei**, and **Adam**. Thank you for all the love, support, inspiration, understanding, and patience as I navigated every single step of this complicated process. I am fortunate to have gone through this time with you all so close geographically. Thank you for being there through every phase: Animal Crossing, Mario Kart, and Ticket to Ride, among others.

ABSTRACT

For many problems of interest in machine learning and control, we have access to rich information about underlying geometry and dynamics; we can leverage this information to build robust and performant solutions in new algorithms, optimizations, and designs. In this thesis we study four problem settings to stress this central assumption. First, we study *conformal generative modeling*, using computational geometry techniques to simplify and register complex 2D surfaces and enabling the use of a variety of flow-based generative models as plug-and-play subroutines. Second, we study *data-driven robust optimization problems* in control, modeling the precise impact of dynamics uncertainty in several control frameworks using *convex geometry*. Third, we study *compactly-restrictable policy optimization*, constraining the available states and actions in reinforcement learning and optimal control problems to be consistent with the inherent dynamics of the systems to be controlled. Finally, we study *nonlinear model predictive control on Lie groups* as applied to a 3D hopping robot platform, developing a control methodology compatible with nontrivial state space geometry and hybrid system dynamics.

PUBLISHED CONTENT AND CONTRIBUTIONS

Dorobantu, Victor D., Kamyar Azizzadenesheli, and Yisong Yue (2023). "Compactly restrictable metric policy optimization problems." In: *IEEE Transactions on Automatic Control* 68.5, pp. 3115–3122. DOI: 10.1109/TAC.2022.3217269.
Victor Dorobantu primarily developed the theory and writing.

Dorobantu, Victor D., Charlotte Borcherds, and Yisong Yue (2023). "Conformal generative modeling on triangulated surfaces." In: *arXiv preprint arXiv:2303.10251*.
Victor Dorobantu primarily developed the theory, code, experiments, and writing. Charlotte Borcherds assisted with the code and experiments.

Csomay-Shanklin, Noel, Victor D. Dorobantu, and Aaron D. Ames (2022). "Nonlinear model predictive control of a 3d hopping robot: Leveraging Lie group integrators for dynamically stable behaviors." In: *arXiv preprint arXiv:2209.11808*.
Victor Dorobantu supported the work, primarily lead by Noel Csomay-Shanklin. In particular, theory and writing were developed together.

Taylor, Andrew J., Victor D. Dorobantu, Ryan K. Cosner, Yisong Yue, and Aaron D. Ames (2022). "Safety of sampled-data systems with control barrier functions via approximate discrete time models." In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, pp. 7127–7134.
Victor Dorobantu contributed equally with Andrew Taylor and Ryan Cosner. Theory, simulation, and writing were all developed together.

Taylor, Andrew J., Victor D. Dorobantu, Sarah Dean, Benjamin Recht, Yisong Yue, and Aaron D. Ames (2021). "Towards robust data-driven control synthesis for nonlinear systems with actuation uncertainty." In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 6469–6476. DOI: 10.1109/CDC45484.2021.9683511.
Victor Dorobantu contributed equally with Andrew Taylor and Sarah Dean. In particular, Victor Dorobantu conceived of and formulated the robust optimization problem and implemented much of the controller code.

Taylor, Andrew J., Victor D. Dorobantu, Yisong Yue, Paulo Tabuada, and Aaron D. Ames (2021). "Sampled-data stabilization with control Lyapunov functions via quadratically constrained quadratic programs." In: *IEEE Control Systems Letters* 6, pp. 680–685.
Victor Dorobantu contributed equally with Andrew Taylor. Theory, simulation, and writing were all developed together.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Number *Page*

LIST OF TABLES

Chapter 1

INTRODUCTION

In this thesis, we study several engineering problems in generative modeling, uncertainty modeling, optimal control, and reinforcement learning, and robotics. Mathematically, we study these problems from the lenses of *geometry* and *dynamical systems*. Our central assumption is the following:

> For many problems of interest in machine learning and control, we have access to rich information about underlying geometry and dynamics; we can leverage this information to build robust and performant solutions in new algorithms, optimizations, and designs.

What observations support this assumption? For problems with physical domains, solutions that do not respect the problem geometry are often nonphysical. We can, for example, streamline the training of a robotic manipulation pipeline by ensuring that our contact models are constrained to object surfaces by construction. Building off of a strong geometric foundation can also help us build rich priors and regularizers to guide learning and inverse problem solving, reason about symmetries, invariances, and equivariances to build flexible models with low complexity and augment data when we only have a partial picture, and generalize to unseen problem domains via deformations or object registration.

Moreover, many real world problems are governed by parsimonious dynamical systems, with simple building blocks generating complex evolution. Throughout physics, robotics, and the control of physical systems, accurate dynamics models are a critical piece of many theories and designs. Complicated timeseries modeling requires a deep understanding of dynamics. Starting from partially accurate models that encode the correct evolutionary structure, we can develop precise and useful uncertainty models for improved prediction and control. Reasoning about reachability, invariance, and infinitesimal changes can help us better explore complex behaviors in reinforcement learning. Moreover, at the time of this writing, many state-of-the-art techniques in generative modeling employ dynamics models, from differentiating through ODE solvers to evolve probability distributions (Chen,

Rubanova, et al., 2018), to building reversible SDEs compatible with Markov Chain Monte Carlo simulation (Song, Sohl-Dickstein, et al., 2020).

There are also intimate links between geometry and dynamical systems that can be easily leveraged in many problem settings. Symmetries and connections are naturally described in a geometric language, and correspond to powerful computational tools that can be used for simulation, scientific discovery, and statistics. Harmonic analysis can be employed for both data processing and dynamics modeling. Diffusion and mass conservation are key elements of flexible probabilistic modeling and manifold learning, especially for high-dimensional problems with low-dimensional structures.

We study the connections between geometry and dynamics, and illustrate their uses in four problem settings. In Chapter 2, we discuss *conformal generative modeling*, the subject of (Dorobantu, Borcherds, and Yue, 2023). Here we use *conformal geometry* to enable generative modeling on complicated 2D surfaces. Specifically, we make use of *discrete* conformal equivalence to conformally *uniformize* data domains in manner that transforms probability densities in understandable ways and allows us to aggregate data from multiple sources through surface registration. The generative models we deploy are *flow-based* models, transporting probability densities via dynamical systems with trainable parameters.

In Chapter 3, we discuss data-driven robust control, the subject of (Taylor, Dorobantu, Dean, et al., 2021). Here we model uncertainties in dynamical systems with *convex geometry*, and study its interplay with robust optimization and convex outer approximation. Specifically, we consider all unmodeled uncertainty that is consistent with data, observed from trajectories of the system in action. We build such uncertainty sets in a way that is compatible with *convex optimization*. Rather than collect data to mitigate the entire gap between a nominal and a true model, we consider the precise effect of uncertainty on key metrics for stability and safety of dynamical systems and robustly optimize *only* over these effects. We also explore the need for *convex relaxations* as the forms of our robust constraints get more complex.

In Chapter 4, we discuss *compactly-restrictable policy optimization problems*, the subject of (Dorobantu, Azizzadenesheli, and Yue, 2023). Here, we describe sufficient conditions for the well-posedness of *value iteration* in a context appropriate for the control of robotic systems. We provide an overview of value iteration (a central tool in classical reinforcement learning and optimal control) and the difficulties involved in transfering the method from the tabular setting to the continuous

state space and action space setting. We show how nonlinear control techniques to achieve *forward invariance* of safe sets can be used to constrain optimal Bellman operators to produce global maximizers. We illustrate the connection between safe set selection and the inherent dynamics of the systems to be controlled.

In Chapter 5, we discuss *nonlinear model predictive control* with *Lie group integrators* and its application to a 3D hopping robot, the subject of (Csomay-Shanklin, Dorobantu, and Ames, 2022). In this work, we formulate a control framework based on *sequential quadratic programming*, requiring a careful geometric treatment of trajectories during optimization. Specifically, we incorporate the Lie group structure of unit quaternions (representing orientation) into our linearizations. A further difficulty comes in the *hybrid system* nature of the robot, switching between classes of dynamics at discrete impacts and necessitating further geometric considerations. This methodology is demonstrated in simulation as well as a hardware platform.

Lastly, we detail our recent approaches to *sampled data control* in Appendix A. This is the subject of (Taylor, Dorobantu, Yue, et al., 2021; Taylor, Dorobantu, Cosner, et al., 2022), and is used in several appropriate contexts throughout the thesis.

1.1 Common Notation

Here, we specify some notation common to all works in this thesis. $\mathbb{N}$ denotes the natural numbers (not including 0), $\mathbb{Z}$ denotes the integers, $\mathbb{Z}_+$ denotes the nonnegative integers (including 0), $\mathbb{R}$ denotes the reals, $\mathbb{R}_+$ denotes the nonnegative reals (including zero), $\mathbb{R}_{++}$ denotes the positive reals (not including zero), $\mathbb{C}$ denotes the complex numbers, and $\mathbb{H}$ denotes the quaternions. For dimension d, $\mathbb{R}^d$ denotes d-dimensional Euclidean space, and $\mathbb{R}^d_+$ denotes the corresponding nonnegative orthant (consisting of vectors with nonnegative entries). $\mathbb{S}^d$ denotes the set of $d \times d$ symmetric matrices; $\mathbb{S}^d_+$ denotes the corresponding set of positive semi-definite matrices (including those with zero eigenvalues) and $\mathbb{S}^d_{++}$ denotes the set of positive definite matrices (excluding those with zero eigenvalues). Both $\mathbb{R}^d_+$ and $\mathbb{S}^d_+$ are self-dual convex cones. S^d denotes the unit d-sphere, canonically represented as the set of unit vectors in $\mathbb{R}^{d+1}$.

We will always use $|\cdot|$ to denote only absolute values of real numbers or magnitudes of complex numbers or quaternions; we will use $\|\cdot\|$ to denote norms of vectors (including functions). When $\|\cdot\|$ is not decorated, it can represent any kind of norm; we use $\|\cdot\|_2$ explicitly when the norm we use must be the 2-norm in Euclidean space.

We reserve the notation $\subset$ for *proper* subsets only; when set equality is possible, we always use $\subseteq$. For a function $f : X \to Y$, we call X the domain, Y the *codomain*, and the image $f(X) \subseteq Y$ the *range*. When we write $f : X \to Y$, we mean f is defined *for all* $x \in X$; if we need to exclude inputs from our definition, we explicitly make note of this with a modified domain.

A continuous function $\alpha : [0, a) \to \mathbb{R}_+$, with $a > 0$ is *class* $\mathcal{K}$ ($\alpha \in \mathcal{K}$) if $\alpha(0) = 0$ and α is strictly monotonically increasing. If $a = \infty$ and $\lim_{r\to\infty} \alpha(r) = \infty$, then α is *class* $\mathcal{K}_\infty$ ($\alpha \in \mathcal{K}_\infty$). A continuous function $\alpha : (-b, a) \to \mathbb{R}$, with $a, b > 0$, is *extended class* $\mathcal{K}$ ($\alpha \in \mathcal{K}_e$) if $\alpha(0) = 0$ and α is strictly monotonically increasing. If $a, b = \infty$, $\lim_{r\to\infty} \alpha(r) = \infty$, and $\lim_{r\to-\infty} \alpha(r) = -\infty$, then α is *extended class* $\mathcal{K}_\infty$ ($\alpha \in \mathcal{K}_{\infty,e}$). A continuous function $\beta : [0, a) \times \mathbb{R}_+ \to \mathbb{R}_+$ with $a > 0$ is *class* $\mathcal{KL}$ ($\beta \in \mathcal{KL}$) if for each $r \in [0, a)$, the map $s \mapsto \beta(r, s)$ is strictly monotonically decreasing with 0 limit and for each $s \in \mathbb{R}_+$, the map $r \mapsto \beta(r, s)$ is class $\mathcal{K}$. If $a = \infty$ and for each $s \in \mathbb{R}_+$, the map $r \mapsto \beta(r, s)$ is class $\mathcal{K}_\infty$, then β is *class* $\mathcal{KL}_\infty$ ($\beta \in \mathcal{KL}_\infty$). Comparison functions in $\mathcal{K}$ ($\mathcal{K}_\infty$) and $\mathcal{K}_e$ ($\mathcal{K}_{\infty,e}$) are invertible and composable (the composition of class $\mathcal{K}$ functions is also class $\mathcal{K}$). Class $\mathcal{KL}$ functions often arise from differential inequalities. For example, if $\alpha \in \mathcal{K}$ and $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ is a trajectory with $\gamma(0)$ in the domain of α and:

$$\dot{\gamma}(t) \leq -\alpha(\gamma(t)), \tag{1.1}$$

for all times $t \in \mathbb{R}_+$, then there is some $\beta \in \mathcal{KL}$ such that:

$$\gamma(t) \leq \beta(\gamma(0), t), \tag{1.2}$$

for all times $t \in \mathbb{R}_+$.

Chapter 2

CONFORMAL GENERATIVE MODELING

2.1 Introduction

The study of learning expressive generative models has seen increasing interest in recent years, owing to the emergence of approaches such as continuous normalizing flows (Rezende and Mohamed, 2015; Dinh, Sohl-Dickstein, and Bengio, 2016; Papamakarios et al., 2021) and diffusion models (Sohl-Dickstein et al., 2015; Song, Sohl-Dickstein, et al., 2020). Such models have the ability to both efficiently draw samples from the learned distribution, as well as compute exact probabilities on the input space, making them useful for many applications in science and engineering from experiment design (Song, Yu, et al., 2022) to inverse problems (Gao et al., 2021; Song, Shen, et al., 2021).

A key challenge in many domains is that the distribution to be estimated lies on a complicated manifold, rather than in Euclidean space. Examples include modeling molecular activity (Chen, Tu, and Lu, 2012; Shapovalov and Dunbrack Jr., 2011),

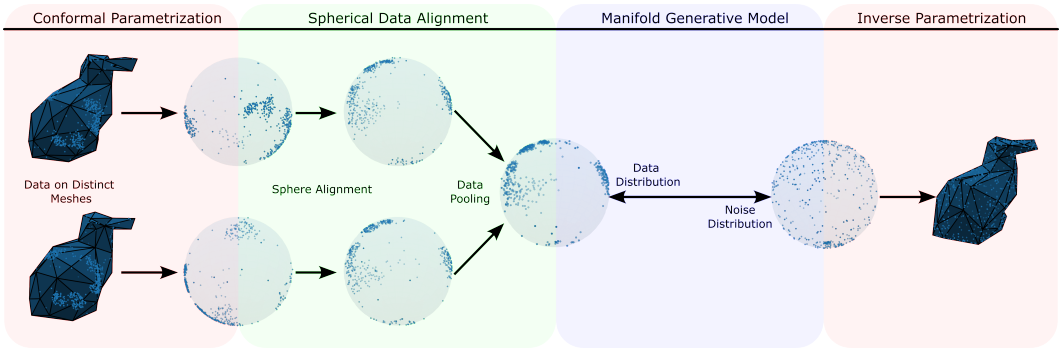


Figure 2.1: Overview of our approach. From left to right, original data samples are gathered from multiple meshes and mapped to unit sphere via discrete conformal transformations (red). The spheres are aligned with a reference mesh using rotations maximizing the correlation between log conformal factors, geometric signatures of the original meshes. Data is then aggregated on one sphere (green). A generative model transforms the data distribution on the sphere to a noise distribution on the sphere in the forward direction (blue). The resulting noise distribution is mapped to the reference mesh, again via a discrete conformal transformation. Here, the noise distribution on the reference mesh is uniform. Meshes are decimated to more clearly distinguish distinct meshes; for full meshes see Figure 2.6.

robotic motion (Feiten, Lang, and Hirche, 2013), high energy physics (Brehmer and Cranmer, 2020), and brain activity (Gerber et al., 2010). Moreover, many domains use discrete approximations of the true manifold (e.g., triangle meshes).

Our goal is to develop generative modeling approaches for complicated 2D surfaces in the case where the training data is represented using discrete mesh approximations. Existing work on Riemannian generative modeling typically operate on canonical manifolds such as spheres, tori, hyperbolic spaces, and matrix Lie groups (De Bortoli et al., 2022; Mathieu and Nickel, 2020; Lou et al., 2020; Katsman et al., 2021) or else require uncontrolled approximations via learned implicit surface representations (Rozen et al., 2021). The challenge of learning generative models directly on complicated meshes remains open.

We propose *conformal generative modeling*, a framework for generative modeling on Riemannian manifolds approximated by discrete triangle meshes. Our approach is based on establishing a conformal transformation between the source mesh and a target mesh that approximates a simple manifold such as a sphere (Springborn, Schröder, and Pinkall, 2008). Such a transformation amounts to a diffeomorphism between the two manifolds, subject to accounting for approximation error from the mesh discretization. Afterwards, one can use any generative model for simple manifolds as a plug-and-play subroutine, which we demonstrate empirically on eight complicated manifolds (which is significantly more than has been demonstrated by other Riemannian generative modeling approaches).

A further benefit of conformal generative modeling is that it can learn simultaneously using data from multiple distinct meshes of the same underlying manifold. This benefit comes from the ability to establish alignments between the conformal transformations of the meshes on the unit sphere (e.g., Baden, Crane, and Kazhdan (2018) and Wang et al. (2006)), as shown in Figure 2.1. We demonstrate this ability empirically, using data from multiple distinct meshes to train a single generative model capable of generating data on unseen meshes.

2.2 Preliminaries & Problem Statement

Our approach uses topological equivalences to simple manifolds (e.g., closed 2D Riemannian manifolds with no "holes" are topological spheres). We first describe notation to describe such concepts and then the problem statement.

2.2.1 Notation, Definitions, and Conventions

Basics. We denote the unit two-sphere by S^2. For manifold $\mathcal{M}$, we denote its tangent space at any $p \in \mathcal{M}$ by $T_p\mathcal{M}$. For manifolds $\mathcal{M}$ and $\mathcal{N}$ and a smooth map $f : \mathcal{M} \to \mathcal{N}$, we denote the differential of f at $p \in \mathcal{M}$ by $df_p : T_p\mathcal{M} \to T_{f(p)}\mathcal{N}$; this is a linear map between tangent spaces.

Triangle Meshes & Vertex Embeddings. A (Euclidean) triangle mesh is described by a triple $\mathsf{T} = (\mathsf{V}, \mathsf{E}, \mathsf{F})$ and a vertex embedding $\mathsf{f} : \mathsf{V} \to \mathbb{R}^d$, where V is the set of vertices, E is a set of ordered vertex pairs comprising edges, F is a set of ordered vertex triples comprising faces, and $\mathsf{f}_i \in \mathbb{R}^d$ denotes the position in space of vertex $i \in \mathsf{V}$. As a simplicial complex, for each face in F, each pair of vertices must belong to the edge set E, potentially with the vertex order reversed. We will exclusively consider *manifold* triangle meshes; in this case, each edge belongs to exactly two faces, with the exception of boundary edges (if they exist), which each belong to exactly one face. Such a triangle mesh is equipped with a *discrete metric* (the discrete analogue of a Riemannian metric) $\ell : \mathsf{E} \to \mathbb{R}_{++}$:

$$\ell_{ij} = \|\mathsf{f}_i - \mathsf{f}_j\|_2. \tag{2.1}$$

This discrete metric must satisfy triangle inequality:

$$\ell_{ij} + \ell_{jk} \leq \ell_{ki}, \qquad \ell_{jk} + \ell_{ki} \leq \ell_{ij}, \qquad \ell_{ki} + \ell_{ij} \leq \ell_{jk}, \tag{2.2}$$

for each face $(i, j, k) \in \mathsf{F}$.

Piecewise Linear Surfaces. Often we will need to consider the piecewise linear surface generated from a vertex embedding by placing triangle vertices at the vertex positions of each face. We will typically denote such a surface as M; we will assume available training data belongs to such surfaces.

Topological Equivalences. The mesh T is a topological sphere if it has no boundary and has an Euler characteristic $|\mathsf{F}| - |\mathsf{E}| + |\mathsf{V}| = 2$ (i.e, has no "holes"). This implies only that T has the connectivity of a sphere (topological information), though not necessarily the geometry of one. If T is a topological sphere *and* $\mathsf{f}_i \in S^2$ for each vertex $i \in \mathsf{V}$, we will also consider *spherical* triangle meshes, which partition the sphere into spherical triangles.

A Riemannian manifold is a topological sphere if it is homeomorphic to a sphere; that is, if it admits a continuous bijection to the sphere with a continuous inverse.

2.2.2 Problem Statement

Consider a 2D Riemannian manifold $\mathcal{M}$ that is a topological sphere. Our goal is to estimate a probability density function $\rho : \mathcal{M} \to \mathbb{R}_+$ using training samples generated i.i.d. from some underlying distribution on $\mathcal{M}$. We are generally interested in complicated manifolds (see Figures 2.1 and 2.5 for examples). Moreover, we assume that the training samples are represented on triangulated mesh approximations of $\mathcal{M}$.

Concretely, suppose we have access to n triangle meshes $\mathsf{T}^{(1)}, \ldots, \mathsf{T}^{(n)}$ that approximate $\mathcal{M}$, with respective vertex embeddings $\mathsf{f}^{(1)}, \ldots, \mathsf{f}^{(n)}$ and piecewise linear surfaces $\mathsf{M}^{(1)}, \ldots, \mathsf{M}^{(n)}$. We assume the vertex embeddings place the vertices on $\mathcal{M}$. We also have access to n datasets, $\mathcal{D}^{(1)} \subset \mathsf{M}^{(1)}, \ldots, \mathcal{D}^{(n)} \subset \mathsf{M}^{(n)}$, with each $\mathcal{D}^{(i)}$ comprising samples drawn i.i.d. from the underlying distribution on $\mathcal{M}$ but represented spatially on $\mathsf{M}^{(1)}, \ldots, \mathsf{M}^{(n)}$. Figure 2.1 (Left/Red) gives a depiction of the training data collected on two distinct meshes of the Stanford Bunny surface.

If we train a model $\rho_\theta : S^2 \to \mathbb{R}_+$ on the sphere via likelihood maximization (with parameters θ), then our learning objective is:

$$\max_\theta \sum_{l=1}^{n} \sum_{x \in \tilde{D}^{(l)}} \left(\log \rho_\theta(x) - \log \Delta^{(l)}(x) \right), \tag{2.3}$$

where $\Delta^{(l)}$ encodes the differential change of area from the sphere to $\mathsf{M}^{(l)}$. We will further comment on this objective in Section 2.3.2.

2.3 Conformal Generative Modeling

Our *conformal generative modeling* framework is predicated on the idea of identifying an invertible transformation from each source mesh to the unit sphere.[1] Given such a transformation, training samples from each mesh are effectively transformed to form a single pooled set of (weighted) training samples on the unit sphere. Afterwards, one can use any existing Riemannian generative modeling approach estimate a probability density function on the sphere (Lou et al., 2020; Mathieu and Nickel, 2020; Rozen et al., 2021; De Bortoli et al., 2022). One can then invert the transformation to produce outputs on any of the meshes. Figure 2.1 presents a high-level depiction of our framework.

[1] One could in principle compute transformations to other simple manifolds with higher Euler numbers, such as tori. However, certain algorithmic steps in the framework become more challenging, such as aligning the different meshes.

We first summarize the key steps, and then develop the necessary technical tools on conformal geometry, spherical parametrizations, and Möbius registration (Sections 2.3.1 & 2.3.2) to describe the full technical details of the approach (Algorithm 1).

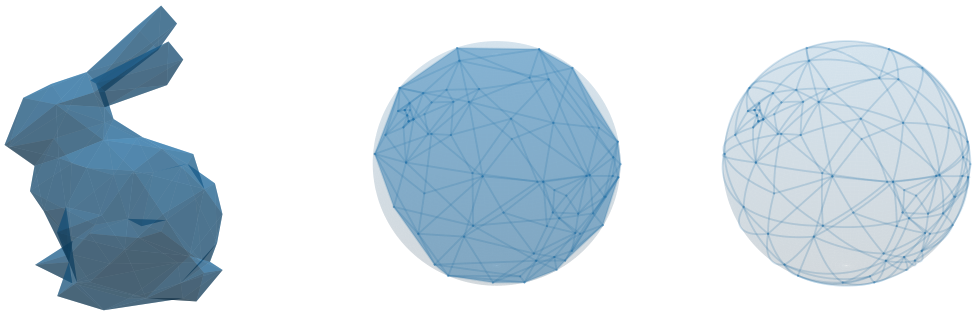
Conformal Transformations & Conformal Factors. The specific type of transformations we will use are *discrete conformal transformations* (Springborn, Schröder, and Pinkall, 2008). Informally, conformal transformations are maximally angle-preserving[2] maps between two meshes (e.g., the source manifold and the sphere). In other words, for any two edges that share a vertex in common, the angle between those two edges are preserved as much as possible. A direct consequence is that the shape of each face on the mesh is also approximately preserved in the transformation. Importantly, however, conformal transformations may distort differential area. In doing so, a conformal transformation rescales edge lengths via a *conformal factor* at each vertex.

Aligning Multiple Meshes. In order to pool training data from multiple meshes, it is necessary to align their transformed meshes on the sphere, which amounts to finding a rotation of each mesh (Section 2.3.1.3). This step is skipped in the special case where we learn with only one mesh.

Creating Training Samples on the Sphere. In order to generate proper training examples on the sphere, two additional steps are needed. First, the conformal transformation has stretched or shrunk various parts of the source mesh, and so one must re-weight training samples in order to transport the original measure to the sphere. Intuitively, data points in stretched portions should receive higher weight, and those in shrunk portions should receive lower weight.

The second step is to transform the linear triangle surfaces into spherical triangle surfaces. Figure 2.2 depicts this step, where the middle is a piecewise linear mesh that is circumscribed by the unit sphere (all the edges lie inside the sphere), and the right is a conversion of each triangle surface to a spherical one. It is straightforward to compose this transformation with the conformal transformation. In Section 2.3.1.2, we describe spherical parameterization, and in Section 2.3.2 we derive the additional differentiable change in area to compute an additional weighting factor for the training samples when transforming to a spherical mesh.

[2]Complete angle preservation overly restricts the class of admissible transformations, yielding only rigid isometries. See Crane (2020) for further discussion.



Original Mesh

Conformally Equivalent Sphere-Inscribed Mesh

Spherical Triangle Mesh (Partition)

Figure 2.2: Correspondence between piecewise linear surfaces and unit sphere, with the original mesh on the left, the mesh after spherical parametrization in the center (shown inscribed within the unit sphere), and the associated partition of the sphere into spherical triangles on the right.

2.3.1 Background on Conformal Geometry

2.3.1.1 Conformal Equivalence

Conformal maps locally preserve angles and orientations. Let $\mathcal{M}$ and $\tilde{\mathcal{M}}$ be Riemannian manifolds with corresponding Riemannian metrics g and $\tilde{g}$, respectively. A smooth local embedding $f : \mathcal{M} \to \tilde{\mathcal{M}}$ is *conformal* if it preserves orientations and there exists a $u : \mathcal{M} \to \mathbb{R}$ satisfying:

$$\tilde{g}_{f(p)}(df_p(v), df_p(w)) = e^{2u(p)} \cdot g_p(v, w), \tag{2.4}$$

for all points $p \in \mathcal{M}$ and tangent vectors $v, w \in T_p\mathcal{M}$. The function u is called the *log conformal factor*. Intuitively, we can equivalently push forward v and w via the differential df_p and compute their inner product in the tangent space at $f(p)$, or we can compute the inner product of v and w in the tangent space at p and scale the product by $e^{2u(p)}$.

Analogously, consider a Euclidean triangle mesh $\mathsf{T} = (\mathsf{V}, \mathsf{E}, \mathsf{F})$ with two embeddings f and $\tilde{\mathsf{f}}$ (e.g., one for the source manifold and one for the sphere), corresponding to discrete metrics ℓ and $\tilde{\ell}$, respectively. The metrics are *(discretely) conformally equivalent* (Springborn, Schröder, and Pinkall, 2008) if there is a function $u : \mathsf{V} \to \mathbb{R}$ satisfying:

$$\tilde{\ell}_{ij} = e^{(u_i + u_j)/2} \cdot \ell_{ij}, \tag{2.5}$$

for all edges $(i, j) \in \mathsf{E}$. As in the continuous case, the function u is also called the log conformal factor. Intuitively, $e^{2u_i} \in \mathbb{R}_{++}$ captures how differential area

changes (multiplicatively) as the vertex $i \in \mathsf{V}$ is sent from f_i to $\tilde{\mathsf{f}}_i$. Two important properties of discrete conformal equivalence are transitivity and invariance under Möbius transformations. First, suppose ℓ_1, ℓ_2, and ℓ_3 are discrete metrics on the same triangle mesh. If ℓ_1 and ℓ_2 are discretely conformally equivalent with log conformal factor u_1 and ℓ_2 and ℓ_3 are discretely conformally equivalent with log conformal factor u_2, then ℓ_1 and ℓ_3 are discretely conformally equivalent with log conformal factor $u_1 + u_2$. Second, if two vertex embeddings corresponding to the same mesh are related via a Möbius transformation (which includes Euclidean transformations, sphere inversions, and stereographic projections), then the corresponding discrete metrics are discretely conformally equivalent (Springborn, Schröder, and Pinkall, 2008).

2.3.1.2 Spherical Parametrizations

Consider a triangle mesh $\mathsf{T} = (\mathsf{V}, \mathsf{E}, \mathsf{F})$ that approximates a sphere, with vertex embedding f and discrete metric ℓ. Typically, such a mesh is circumscribed by the sphere, meaning the vertices lie on the sphere and the edges lie within. Our goal here is to transform each linear triangle surface in T into a spherical triangle (Figure 2.3). We use the following procedure (Springborn, Schröder, and Pinkall, 2008; Bobenko, Sechelmann, and Springborn, 2016), which returns a new embedding $\tilde{\mathsf{f}}$ of T that maps the vertices to the sphere S^2 and corresponds to a discrete metric that is discretely conformally equivalent to ℓ:

1. Select an arbitrary vertex $i^* \in \mathsf{V}$ to be removed.
2. Apply a change of discrete metric, making each neighbor of i^* equally distant from i^* (this metric is discretely conformally equivalent to ℓ).
3. Remove i^* from V, and all incident edges from E and incident faces from F (yielding a topological disk).
4. Compute a *flat* discretely conformally equivalent metric via convex optimization which leaves the distances between boundary vertices unchanged.
5. Embed the vertices in the plane $\mathbb{R}^2$ with edge lengths determined by the new flat metric.
6. Stereographically project the vertices onto the unit sphere S^2 (through the north pole), maintaining discrete conformal equivalence of metrics.

7. Reinsert i^* at the north pole, along with removed edges and faces.

8. Apply Möbius transformations that map the sphere to itself (Lorentz transformations) to move the center of the vertex positions to the sphere center, again maintaining discrete equivalence of metrics.

To compute the flat metric, permitting a planar vertex embedding, we minimize the convex energy in Equation 7 of (Springborn, Schröder, and Pinkall, 2008), a function of the corresponding log conformal factor. To describe this energy, first note that from log conformal factors $u : \mathsf{V} \to \mathbb{R}$, we can compute new lengths $\tilde{\ell} : \mathsf{E} \to \mathbb{R}$. For face $(i, j, k) \in \mathsf{F}$, let $\tilde{\alpha}^i_{jk}$, $\tilde{\alpha}^j_{ki}$, and $\tilde{\alpha}^k_{ij}$ denote the interior angles at vertices i, j, and k, respectively, which can be computed from $\tilde{\ell}_{ij}$, $\tilde{\ell}_{jk}$, and $\tilde{\ell}_{ki}$ using the law of cosines. Let $\widehat{\Theta} : \mathsf{V} \to \mathbb{R}$ denote the desired sums of the interior angles at each vertex (2π for interior vertices and π for boundary vertices). The energy is then:

$$E(u) = \sum_{(i,j,k)\in F} \left(\tilde{\alpha}^i_{jk} \log \tilde{\ell}_{jk} + \tilde{\alpha}^j_{ki} \log \ell_{ki} + \tilde{\alpha}^k_{ij} \log \tilde{\ell}_{ij} - \frac{\pi}{2}(u_i + u_j + u_k) \right.$$
$$\left. - \int_0^{\tilde{\alpha}^i_{jk}} \log |2 \sin t| \, \mathrm{d}t - \int_0^{\tilde{\alpha}^j_{ki}} \log |2 \sin t| \, \mathrm{d}t - \int_0^{\tilde{\alpha}^k_{ij}} \log |2 \sin t| \, \mathrm{d}t \right), \quad (2.6)$$

where the integral results from applying Milnor's Lobachevsky function to the interior angles. Though this energy is complicated to evaluate, we can still minimize the energy with second-order optimization (Newton's method), where the gradient of the energy simply measures the defect of the sum of angles around each vertex from 2π (or π for boundary vertices) and the Hessian is the (cotangent) Laplacian, computed from the transformed angles. When the gradient is 0, the sum of the angles around each interior vertex is exactly 2π, which is the required curvature of a flat surface. The boundary edge lengths are unchanged by imposing 0 boundary conditions during optimization.

The domain over which the convex energy is optimized includes log conformal factors corresponding to discrete metrics that violate the triangle inequality, meaning it is possible to find a solution which cannot be used to embed vertices in the plane. (Springborn, Schröder, and Pinkall, 2008) propose flipping edges when such violations are detected, and many follow-up works have investigated this problem further, see (Gillespie, Springborn, and Crane, 2021) for a complete discussion and solution. For the meshes in our experiments, we did not encounter such problems.

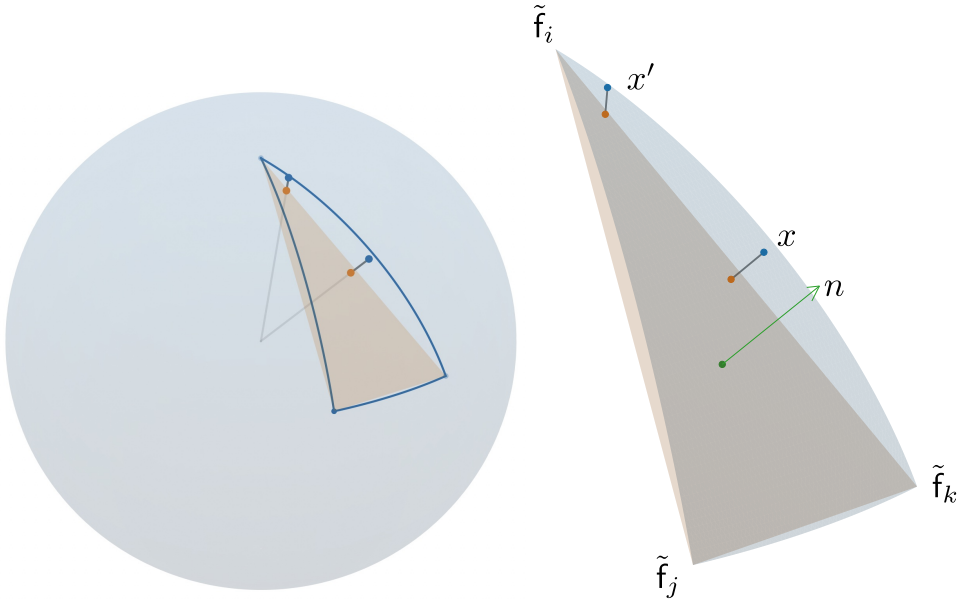


Figure 2.3: Corrsepondence between an inscribed Euclidean triangle and its associated spherical triangle. Points on the sphere are in blue, projected points (onto the inscribed triangles) are in orange, and the inscribed triangle normal vector is in green.

After computing new vertex positions on the sphere, Möbius transformations moving the center of the vertex positions to the center of the unit sphere can be computed using either (Bobenko, Sechelmann, and Springborn, 2016) or (Baden, Crane, and Kazhdan, 2018); we used the latter, which adds weights to the vertex positions based on corresponding triangle areas from the original mesh.

We denote the resulting vertex positions by $\tilde{\mathsf{f}}$. Denote the discrete metric corresponding to $\tilde{\mathsf{f}}$ by $\tilde{\ell}$. As a byproduct of this procedure, we also obtain the log conformal factor u which establishes discrete conformal equivalence between ℓ and $\tilde{\ell}$. We refer to $\tilde{\mathsf{f}}$ as a spherical parametrization of T.

Through spherical parametrization, we establish correspondences between three surfaces: a piecewise linear surface described by f, a piecewise linear surface inscribed within the unit sphere and described by $\tilde{\mathsf{f}}$, and the unit sphere itself. These correspondences are illustrated in Figure 2.2. We move between the piecewise linear surfaces via barycentric coordinates within each face (this transformation is also piecewise linear). To move from the surface inscribed within the sphere to the sphere itself, we simply normalize points to have unit norm. To move in the opposite direction (from the sphere to the surface inscribed within the sphere), we compute the intersection of the ray from the origin in the direction of a query point

and the inscribed surface. Specifically, if $x \in S^2$ is a query point located in the spherical triangle characterized by $(i, j, k) \in \mathsf{F}$, then the intersection $\alpha x \in \mathbb{R}^3$ (for some scalar $\alpha \in [0, 1]$) with the inscribed surface satisfies:

$$((\tilde{\mathsf{f}}_j - \tilde{\mathsf{f}}_i) \times (\tilde{\mathsf{f}}_k - \tilde{\mathsf{f}}_i))^\top (\alpha x - \tilde{\mathsf{f}}_i) = 0, \tag{2.7}$$

where the cross product $(\tilde{\mathsf{f}}_j - \tilde{\mathsf{f}}_i) \times (\tilde{\mathsf{f}}_k - \tilde{\mathsf{f}}_i)$ is perpendicular to the inscribed Euclidean triangle characterized by (i, j, k). Rearranging terms, we have:

$$\alpha = \frac{((\tilde{\mathsf{f}}_j - \tilde{\mathsf{f}}_i) \times (\tilde{\mathsf{f}}_k - \tilde{\mathsf{f}}_i))^\top \tilde{\mathsf{f}}_i}{((\tilde{\mathsf{f}}_j - \tilde{\mathsf{f}}_i) \times (\tilde{\mathsf{f}}_k - \tilde{\mathsf{f}}_i))^\top x} = \frac{(\tilde{\mathsf{f}}_j \times \tilde{\mathsf{f}}_k)^\top \tilde{\mathsf{f}}_i}{((\tilde{\mathsf{f}}_j - \tilde{\mathsf{f}}_i) \times (\tilde{\mathsf{f}}_k - \tilde{\mathsf{f}}_i))^\top x}. \tag{2.8}$$

This latter correspondence is illustrated in Figure 2.3.

With these correspondences established, we denote the two piecewise linear surfaces as $\mathsf{M} \subset \mathbb{R}^d$ and $\mathsf{M}_{\text{inscr}} \subset \mathbb{R}^3$, respectively. We also generate an interpolation of the log conformal factor u as a function on the sphere, $\bar{u} : S^2 \to \mathbb{R}$. To compute the interpolated value $\bar{u}(x)$ at a query point $x \in S^2$, we first project x onto $\mathsf{M}_{\text{inscr}}$ using Equation (2.8); we then interpolate the values of u at the vertices of the intersected triangle using the corresponding barycentric coordinates as interpolation weights.

2.3.1.3 Möbius Registration

In order to completely relate two meshes of the same manifold (e.g., to pool their training data), we must align their representations. With the result of Section 2.3.1.2, two spherical parametrizations of the same mesh will differ only by a rotation (Baden, Crane, and Kazhdan, 2018). The set of rotations is substantially easier to work with than the set of Lorentz transformations. This rotation can be found via harmonic analysis.

For two complex-valued functions on the sphere $f, g : S^2 \to \mathbb{C}$, we can define a correlation function $C : SO(3) \to \mathbb{C}$ on the rotation group $SO(3)$ as:

$$C(R) = \int_{S^2} f(x)\overline{g(R^\top x)}\, \mathrm{d}x, \tag{2.9}$$

for all rotation matrices $R \in SO(3)$. If f and g are real-valued, then so is C, and we can seek its global maximizer. The spherical harmonic transform (Kostelec and Rockmore, 2008) yields coefficients such that:

$$f(x) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \hat{f}_{lm} Y_l^m(x), \qquad g(x) = \sum_{l=0}^{\infty} \sum_{m=-l}^{\ell} \hat{g}_{lm} Y_l^m(x), \tag{2.10}$$

Algorithm 1 Conformal Generative Modeling

input Triangle meshes $\mathsf{T}^{(1)}, \ldots, \mathsf{T}^{(n)}$ and correpsonding piecewise linear surfaces $\mathsf{M}^{(1)}, \ldots, \mathsf{M}^{(n)}$

input Training sets $\mathcal{D}^{(1)} \subset \mathsf{M}^{(1)}, \ldots, \mathcal{D}^{(n)} \subset \mathsf{M}^{(n)}$

1: Compute conformal transformation from each mesh to the unit sphere (Section 2.3.1.2), obtaining $\mathsf{M}^{(1)}_{\text{inscr}}, \ldots, \mathsf{M}^{(n)}_{\text{inscr}}$
2: Align $\mathsf{M}^{(2)}_{\text{inscr}}, \ldots, \mathsf{M}^{(n)}_{\text{inscr}}$ to $\mathsf{M}^{(1)}_{\text{inscr}}$ with maximum correlation rotations (Equation (2.9))
3: Map data to aligned spheres, obtaining $\tilde{\mathcal{D}}^{(1)}, \ldots, \tilde{\mathcal{D}}^{(n)}$
4: Compute log changes of area from projection (Equation (2.16)) and triangle area ratios for each data point in each of $\tilde{\mathcal{D}}^{(1)}, \ldots, \tilde{\mathcal{D}}^{(n)}$
5: Train likelihood-based generative model on sphere, subtracting log change of area from log probability densities in loss function (Equation (2.21))
6: Return sphere generative model and inverse conformal transformations mapping to each of $\mathsf{M}^{(1)}, \ldots, \mathsf{M}^{(n)}$

for all $x \in S^2$, where $Y_l^m : S^2 \to \mathbb{C}$ denotes the spherical harmonic of l and order m. The $SO(3)$ Fourier transform of C can then be written in terms of the spherical harmonic transforms of f and g, as:

$$C(R) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \sum_{m'=-l}^{l} \hat{f}_{lm} \overline{\hat{g}_{lm'}} \cdot \overline{D^l_{mm'}(R)}, \tag{2.11}$$

for all rotations $R \in SO(3)$, where $D^l_{mm'}$ denotes a Wigner D-function (Kostelec and Rockmore, 2008). Intuitively, the coefficients of Equation (2.11) are degree-wise outer products of the coefficients comprising each degree of the spherical harmonic transforms of f and g.

While evaluating C at any rotation is computationally difficult, (Kostelec and Rockmore, 2008) provide a fast method using a grid of Euler angles. This method makes use of the fast spherical harmonic transform, sampling f and g onto a spherical grid. We found a 64×64 spherical grid (a bandwidth of 32) to be sufficient for our experiments.

We can now fully describe Steps 1-3 of Algorithm 1. We spherically parametrize each of the piecewise linear surfaces, obtaining sphere-inscribed piecewise linear surfaces $\mathsf{M}^{(1)}_{\text{inscr}}, \ldots, \mathsf{M}^{(n)}_{\text{inscr}}$ and log conformal factor interpolations $\bar{u}^{(1)}, \ldots, \bar{u}^{(n)} : S^2 \to \mathbb{R}$. For $i \in \{2, \ldots, n\}$, we approximate the maximum correlation rotation matrix $R^{(i)} \in SO(3)$, maximizing the correlation function:

$$C^{(i)}(R) = \int_{S^2} \bar{u}^{(1)}(x) \bar{u}^{(i)}(R^\top x) \, \mathrm{d}x, \tag{2.12}$$

over all rotation matrices $R \in SO(3)$. With identity rotation $R^{(1)} = I_3$, we rotate each of the inscribed surfaces by the corresponding rotation, obtaining rotated surfaces

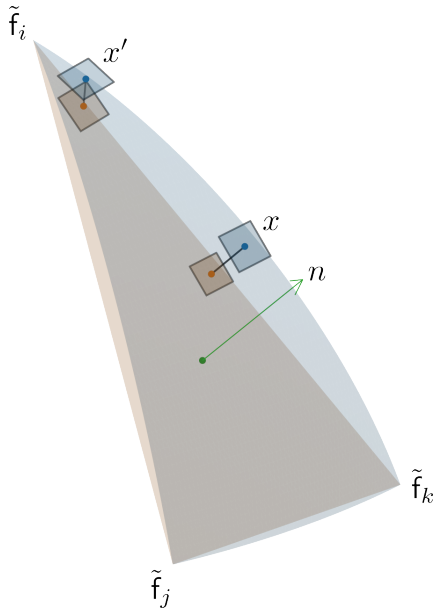


Figure 2.4: Tangent squares (in blue) and pushed forward parallelograms (in orange) used in differential change of area computation.

$\tilde{\mathsf{M}}^{(1)}_{\text{inscr}}, \ldots, \tilde{\mathsf{M}}^{(n)}_{\text{inscr}}$. In Step 2, we map each data point in each $\mathcal{D}^{(i)}$ to $\tilde{\mathsf{M}}^{(i)}_{\text{inscr}}$ and normalize the result to lie on the unit sphere, obtaining a spherical dataset $\tilde{\mathcal{D}}^{(i)}$.

We can now aggregate the spherical datasets and train a generative model using the resulting dataset (Step 3). However, to train such models with maximum likelihood estimation on the original surfaces $\mathsf{M}^{(1)}, \ldots, \mathsf{M}^{(n)}$, we require the corresponding changes of area incurred by each of the maps to the unit sphere (Step 4).

2.3.2 Differential Change of Area

For mesh index $l \in \{1, \ldots, n\}$, consider a data point $x \in \tilde{\mathcal{D}}^{(l)}$. The data point x is mapped from the sphere to $\mathsf{M}^{(l)}$ (the inverse transformation) in three stages; first x is mapped onto a Euclidean triangle of $\tilde{\mathsf{M}}^{(l)}_{\text{inscr}}$, then rotated by the inverse rotation ${R^{(l)}}^{\top}$ to lie on $\mathsf{M}^{(l)}_{\text{inscr}}$, and finally moved to the point on $\mathsf{M}^{(l)}$ with the same barycentric coordinates in the corresponding face. The rotation is isometric, incurring no change of area, and the final map from $\mathsf{M}^{(l)}_{\text{inscr}}$ to $\mathsf{M}^{(l)}$ is piecewise linear, incurring a multiplicative change of area equal to the area ratio of corresponding triangles (original triangle area divided by inscribed triangle area).

The first mapping, however, is nonlinear. To compute the corresponding change of area, suppose x is in the spherical triangle characterized by $(i, j, k) \in \mathsf{F}^{(l)}$, where $\mathsf{F}^{(l)}$ is the set of faces in $\mathsf{T}^{(l)}$. Let n denote the normalized cross product of edge

vectors in the face. That is, consider the unit vector perpendicular to the face, defined by:

$$n = \frac{(\tilde{\mathbf{f}}_j^{(l)} - \tilde{\mathbf{f}}_i^{(l)}) \times (\tilde{\mathbf{f}}_k^{(l)} - \tilde{\mathbf{f}}_i^{(l)})}{\|(\tilde{\mathbf{f}}_j^{(l)} - \tilde{\mathbf{f}}_i^{(l)}) \times (\tilde{\mathbf{f}}_k^{(l)} - \tilde{\mathbf{f}}_i^{(l)})\|_2}. \tag{2.13}$$

From Equation (2.8), we find that x is mapped to:

$$y = \frac{n^\top \tilde{\mathbf{f}}_i^{(l)}}{n^\top x} \cdot x, \tag{2.14}$$

with Jacobian matrix given as:

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{(n^\top x)(n^\top \tilde{\mathbf{f}}_i^{(l)}) I_3 - (n^\top \tilde{\mathbf{f}}_i^{(l)}) x n^\top}{(n^\top x)^2}. \tag{2.15}$$

Now, consider any tangent vector to the sphere $v \in \mathbb{R}^3$, with $x^\top v = 0$. Suppose additionally that v has unit magnitude. The cross product $x \times v$ is another tangent vector, the result of rotating v in the tangent plane counterclockwise by a quarter turn. The pair $(v, x \times v)$ form an orthonormal basis for the tangent plane, and the triple $(v, x \times v, x)$ forms a right-handed orthonormal basis for $\mathbb{R}^3$. The parallelogram spanned by v and $x \times v$ is a square with area 1, so we can compute the multiplicative change of area incurred when mapping x to y simply by computing the area of the parallelogram obtained by pushing forward v and $x \times v$ by the Jacobian $\mathrm{d}y/\mathrm{d}x$. These tangent squares and parallelograms are shown in Figure 2.4. This area is:

$$\begin{aligned}
&\left\| \left(\frac{\mathrm{d}y}{\mathrm{d}x} \cdot v \right) \times \left(\frac{\mathrm{d}y}{\mathrm{d}x} \cdot (x \times v) \right) \right\|_2 \\
&= \frac{(n^\top \tilde{\mathbf{f}}_i^{(l)})^2}{(n^\top x)^4} \cdot \left\| \left((n^\top x) v - (n^\top v) x \right) \times \left((n^\top x)(x \times v) - n^\top (x \times v) x \right) \right\|_2 \\
&= \frac{(n^\top \tilde{\mathbf{f}}_i^{(l)})^2}{|n^\top x|^3} \cdot \left\| (x^\top n) x + ((x \times v)^\top n)(x \times v) + (v^\top n) v \right\|_2 = \frac{(n^\top \tilde{\mathbf{f}}_i^{(l)})^2}{|n^\top x|^3}.
\end{aligned} \tag{2.16}$$

The last equality follows since the term inside the norm is just the sum of the projections of n onto x, $x \times v$, and v; since these three vectors form an orthonormal basis, the resulting sum has unit norm. Since:

$$((\tilde{\mathbf{f}}_j^{(l)} - \tilde{\mathbf{f}}_i^{(l)}) \times (\tilde{\mathbf{f}}_k^{(l)} - \tilde{\mathbf{f}}_i^{(l)}))^\top \tilde{\mathbf{f}}_i^{(l)} = (\tilde{\mathbf{f}}_j^{(l)} \times \tilde{\mathbf{f}}_k^{(l)})^\top \tilde{\mathbf{f}}_i^{(l)}, \tag{2.17}$$

we can write the final expression in Equation (2.16) as:

$$\frac{((\tilde{\mathbf{f}}_j^{(l)} \times \tilde{\mathbf{f}}_k^{(l)})^\top \tilde{\mathbf{f}}_i^{(l)})^2 \cdot \|(\tilde{\mathbf{f}}_j^{(l)} - \tilde{\mathbf{f}}_i^{(l)}) \times (\tilde{\mathbf{f}}_k^{(l)} - \tilde{\mathbf{f}}_i^{(l)})\|_2}{|((\tilde{\mathbf{f}}_j^{(l)} - \tilde{\mathbf{f}}_i^{(l)}) \times (\tilde{\mathbf{f}}_k^{(l)} - \tilde{\mathbf{f}}_i^{(l)}))^\top x|^3}, \tag{2.18}$$

where the norm in the numerator is the area of the inscribed triangle characterized by (i, j, k).

To summarize, the differential change of area when mapping x from the sphere to $\mathsf{M}^{(l)}$ is, as a multiplicative factor, the product of Equation (2.19) and the ratio of areas for face (i, j, k), represented both in $\mathsf{M}^{(l)}$ and $\mathsf{M}^{(l)}_{\text{inscr}}$. Using the observation about the norm from Equation (2.18), this differential change of area is:

$$\Delta^{(\ell)} \triangleq \frac{((\tilde{\mathsf{f}}^{(l)}_j \times \tilde{\mathsf{f}}^{(l)}_k)^\top \tilde{\mathsf{f}}^{(l)}_i)^2 \|(\mathsf{f}^{(l)}_j - \mathsf{f}^{(l)}_i) \times (\mathsf{f}^{(l)}_k - \mathsf{f}^{(l)}_i)\|_2}{|((\tilde{\mathsf{f}}^{(l)}_j - \tilde{\mathsf{f}}^{(l)}_i) \times (\tilde{\mathsf{f}}^{(l)}_k - \tilde{\mathsf{f}}^{(l)}_i))^\top x|^3}, \tag{2.19}$$

where now the norm in the numerator is instead the area of the triangle characterized by (i, j, k) in the original mesh.

When evaluating log probability densities via a generative model on the sphere, we must subsequently subtract the log changes of area to compute log probability densities (corresponding to dividing density by change of area) on the original surface.

Elaborating on this point, suppose we choose a generative model on the sphere with density $\rho : S^2 \to \mathbb{R}_+$. For a dataset $\tilde{\mathcal{D}}^{(1)} \cup \cdots \cup \tilde{\mathcal{D}}^{(n)}$ comprised of aligned spherical datasets, the data log likelihood (under the spherical density) is:

$$\sum_{l=1}^{n} \sum_{x \in \tilde{\mathcal{D}}^{(l)}} \log \rho(x). \tag{2.20}$$

while the corrected data log likelihood (under the corresponding mesh densities) is simply:

$$\sum_{l=1}^{n} \sum_{x \in \tilde{\mathcal{D}}^{(l)}} \left(\log \rho(x) - \log \Delta^{(l)}(x) \right), \tag{2.21}$$

where $\Delta^{\ell}(x)$ denotes the change of area from Equation (2.19) for data point $x \in \tilde{\mathcal{D}}^{(l)}$. Note that the change of area terms only need to be computed once at the start of training for such a generative model via likelihood maximization; the only terms which cannot be precomputed include ρ, as this density changes throughout training.

The remaining Step 5 is to choose an appropriate generative model to train on the sphere, which we discuss next.

2.3.3 Choosing a Base Generative Model

In principle, one could choose any Riemannian generative model that trains on canonical manifolds such as the unit sphere. In this paper, we instantiate our frame-

work using two approaches: Riemannian continuous normalizing flows (CNFs) (Mathieu and Nickel, 2020) and Moser flows (Rozen et al., 2021).

In the sections that follow, for convenience we adopt the convention of diffusion and score-based generative models (Sohl-Dickstein et al., 2015; Song, Sohl-Dickstein, et al., 2020), for which the *forward* direction of a generative model maps a data distribution to a noise distribution.

2.3.3.1 Riemannian Continuous Normalizing Flows

In the Euclidean setting, a *continuous normalizing flow* (CNF) (Chen, Rubanova, et al., 2018) is characterized by a parametric time-varying vector field $f : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$. For an initial probability density $\rho_0 : \mathbb{R}^d \to \mathbb{R}_+$, the time-varying density $\rho : \mathbb{R}^d \times [0, 1] \to \mathbb{R}_+$ solving (with initial condition ρ_0) the probability mass continuity equation:

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \cdot f) = 0, \tag{2.22}$$

describes how the probability density of a random initial condition (distributed according to ρ_0) evolves subject to the vector field f. The product $\rho \cdot f$ is the *probability mass flux*, and its divergence can be expanded as:

$$\operatorname{div}(\rho \cdot f) = \nabla \rho^\top f + \rho \cdot \operatorname{div} f. \tag{2.23}$$

This form allows us to express the rate of change of (log) probability density along deterministic trajectories governed by the vector field f. That is, for $x \in \mathbb{R}^d$ and a trajectory $\gamma : [0, 1] \to \mathbb{R}^d$ satisfying:

$$\frac{\mathrm{d}\gamma}{\mathrm{d}t} = f(\gamma(t), t), \tag{2.24}$$

for all $t \in (0, 1)$, we also have (from Chen, Rubanova, et al. (2018)):

$$\frac{\mathrm{d}}{\mathrm{d}t} \log \rho(\gamma(t), t) = -\operatorname{div}(\rho \cdot f)(\gamma(t), t). \tag{2.25}$$

This ordinary differential equation represented by Equation (2.24) and Equation (2.25) can be solved simultaneously. With this augmented system, we can map a data distribution to a noise distribution (solve forward) or a noise distribution to a data distribution (solve backward). Training the model requires querying log likelihoods of data; to compute likelihoods, data are propagated forward under Equation (2.24), the log probability densities of the corresponding terminal states (under a chosen noise distribution) are computed, and the terminal states and log probability densities are propagated backward under both Equation (2.24) and Equation (2.25).

Differentiable ODE solvers are employed to update parameters of the vector field, maximizing data likelihood.

From Euclidean to Spherical Gradients. In the spherical setting, vector fields specify tangent vectors. That is, at a point $x \in S^2$, a vector field specifies a tangent vector $v \in \mathbb{R}^3$ such that $x^\top v = 0$. We can compute differential operators like ∇ and div and solve ordinary differential equations by extending functions defined on the sphere to functions defined on all of $\mathbb{R}^3 \setminus \{0\}$ as in Rozen et al. (2021). Specifically, for tangent vector field $f : S^2 \times [0,1] \to \mathbb{R}^3$ and probability density $\rho : S^2 \times [0,1] \to \mathbb{R}_+$, we define extensions:

$$\tilde{f}(x,t) = f\left(x/\|x\|_2, t\right), \qquad \tilde{\rho}(x,t) = \rho\left(x/\|x\|_2, t\right), \tag{2.26}$$

for all nonzero $x \in \mathbb{R}^3$ and times $t \in [0,1]$. The spherical gradient of f is the Euclidean gradient of $\tilde{f}$. The spherical divergences of f and $\rho \cdot f$ are the respective Euclidean divergences of $\tilde{f}$ and $\tilde{\rho} \cdot \tilde{f}$. As in (Rozen et al., 2021), we can solve Equation (2.24) by substituting f with $\tilde{f}$ and deploying any differentiable (adaptive-step) Euclidean solver. Alternatively, we can solve Equation (2.24) using charts on the sphere (Lou et al., 2020), which locally represent the sphere and the vector field in Euclidean space (for example, by using spherical exponential and log maps). While this latter approach has the advantage of constructively restricting the solution of Equation (2.24) to the sphere, in our experiments the former approach maintained proximity to the sphere within tight tolerances while executing much faster.

2.3.3.2 Moser Flows

(Rozen et al., 2021) bypasses the need to solve ordinary differential equations to query log likelihoods by modeling the probability mass flux directly (as opposed to the vector field f) and restricting the class of vector fields used to generate data samples. For a data density $\mu : S^2 \to \mathbb{R}_+$ and a noise density $\nu : S^2 \to \mathbb{R}_+$, the probability mass flux is modeled as a parametric time-invariant tangent vector field $F : S^2 \to \mathbb{R}^3$ for which:

$$\mu(x) - \nu(x) = -\mathrm{div} F(x), \tag{2.27}$$

for all $x \in S^2$. That is, when ν is selected as well as parameters for F, the modeled data density is given by Equation (2.27). To generate data samples from noise, the vector field:

$$f(x,t) = \frac{F(x)}{\nu(x) + (1-t) \cdot \mathrm{div} F(x)}, \tag{2.28}$$

is used, and the probability density solving the corresponding continuity equation is:

$$\rho(x,t) = (1-t)\mu(x) + t\nu(x), \tag{2.29}$$

for all $x \in S^2$ and $t \in [0,1]$. Note that this probability density interpolates linearly (in time) between μ and ν.

To ensure that the flux model corresponds to a valid probability density, the divergence of the flux field must integrate to 0 and be greater than $-\nu$ everywhere. The first requirement follows from the divergence theorem, replacing the integral of divergence with a boundary integral (where the boundary is empty). The second requirement must be enforced via an integral constraint, approximated with Monte Carlo integration (using a uniform samples from the original mesh).

2.3.3.3 Riemannian Score-Based Generative Models

While we focus on training continuous normalizing flows (including Moser flows) with likelihood maximization, we also provide an overview of possible stochastic differential equation-based approaches.

In the Euclidean setting again, a *score-based generative model* (Song, Sohl-Dickstein, et al., 2020) is characterized by a *parameterless* drift-diffusion process. In this context, the drift-diffusion process is specified through a stochastic differential equation (SDE), written as:

$$dX_t = f(X_t, t) \cdot dt + \sigma(t) \cdot dW_t, \tag{2.30}$$

where $f : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is a time-varying vector field and $\sigma : [0,1] \to \mathbb{R}_+$ is a time-varying diffusion coefficient. The stochastic process W is a d-dimensional Brownian motion. For an initial probability density $\rho_0 : \mathbb{R}^d \to \mathbb{R}_+$, the time-varying density $\rho : \mathbb{R}^d \times [0,1] \to \mathbb{R}_+$ solving (with initial condition ρ_0) the *Fokker-Planck equation*:

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \cdot f) - \frac{\sigma^2}{2} \cdot \Delta\rho = 0, \tag{2.31}$$

described how the probability density of a random initial condition (distributed according to ρ_0) evolves under the chosen drift-diffusion process. Note that if:

1. f is divergence-free at all times, then this equation reduces to a linear advection-diffusion equation,

2. f is 0 at all times, then this equation reduces to the heat equation,

3. σ is 0 at all times, then this equation reduces to the mass continuity equation.

We choose f and σ such that the terminal condition $\rho(\cdot, 1) : \mathbb{R}^d \to \mathbb{R}_+$ approximates a tractable noise distribution. A common choice of vector field f is spatially linear, yielding a time-varying Ornstein-Uhlenbeck (OU) process. When the initial distribution is Gaussian, an OU process is a Gaussian process (Särkkä and Solin, 2019, Example 5.14), meaning at every time, X is Gaussian-distributed. When the OU process is time-invariant, we can characterize a steady-state distribution and speed of convergence to this distribution (Särkkä and Solin, 2019, Example 6.8).

We can approximate the flow applied to data samples with an Euler-Maruyama scheme, with:

$$X_{k+1} = X_k + \Delta t \cdot f(X_k, k\Delta t) + \sqrt{\Delta t} \cdot \sigma(k\Delta t) \cdot Z_k \tag{2.32}$$

where $Z_k \sim \mathcal{N}(0, I)$ and $\Delta t \in \mathbb{R}_{++}$ is a fixed time step. The noise vectors Z_k at each time step are independent. Note that the product $\sqrt{\Delta t} \cdot \sigma(k\Delta t) \cdot Z_k$ is a Gaussian random vector with mean 0 and covariance $\Delta t \cdot \sigma(k\Delta t)^2 \cdot I$. However, *reversing* the flow is more complicated than in the ODE case. This requires modeling the *(Stein) score* $s : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$, a vector field defined as:

$$s(x, t) = \nabla \log \rho(x, t), \tag{2.33}$$

for all $x \in \mathbb{R}^d$ and $t \in [0, 1]$. Typically, this score is modeled as a neural network via *score matching*. Letting $\bar{\rho} : \mathbb{R}^d \times \mathbb{R}^d \times [0, 1] \to \mathbb{R}_+$ denote a time-varying *conditional* density, where $\bar{\rho}(x, z, t)$ is the probability density of x at time t when the initial state is z, score matching attempts to minimize:

$$\int_0^1 \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \|s_\theta(x, t) - \nabla \log \bar{\rho}(x, t)\|_2^2 \cdot \rho_0(z) \cdot \bar{\rho}(x, z, t) \, \mathrm{d}z \, \mathrm{d}x \, \mathrm{d}t. \tag{2.34}$$

Here, $s_\theta : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ is the modeled score and the product $\rho_0 \cdot \bar{\rho}$ is the joint density of x (at time t) and z. This loss is approximated via Monte Carlo integration; each sample is drawn by selecting a data point z uniformly at random, selecting a time t uniformly between 0 and 1, and drawing a *noised* sample x from $\bar{\rho}(\cdot, z, t)$. In general, this last step of sampling requires Euler-Maruyama simulation, but in the OU process case, we can skip simulation and sample directly from an appropriately conditioned Gaussian distribution (Song, Sohl-Dickstein, et al., 2020, Equation 29).

Once the score is modeled, the reverse flow is approximated via:

$$X_{k-1} = X_k - \Delta t \cdot \Big(f(X_k, k\Delta t) - \sigma(k\Delta t)^2 \cdot s_\theta(X_k, k\Delta t) \Big) + \sqrt{\Delta t} \cdot \sigma(k\Delta t) \cdot Z_k, \tag{2.35}$$

where again, the noise vectors $Z_k \sim \mathcal{N}(0, I)$ at each time step are independently distributed.

From Euclidean to Spherical SDEs.

To generalize this procedure to spherical generative modeling (De Bortoli et al., 2022), Brownian motion, Euler-Maruyma schemes, and score matching must all be appropriately modified. Isotropic Brownian motion on a sphere (or any closed Riemannian manifold) can be described via a time-varying probability density that solves the heat equation (with a Dirac delta initial condition). However, (De Bortoli et al., 2022) simulate approximations of Brownian motions via *geodesic random walks*, modifying an Euler-Maruyama scheme as:

$$X_{k+1} = \exp_{X_k}\left(\Delta t \cdot f(X_k, k\Delta t) + \sqrt{\Delta t} \cdot \sigma(k\Delta t) \cdot Z_k\right), \tag{2.36}$$

where Z_k is a standard Gaussian random vector restricted to the tangent space at X_k. In the case of $f = 0$ and $\sigma = 1$ (pure diffusion), (De Bortoli et al., 2022) use the approximate score matching loss:

$$\int_0^1 \int_{S^2} \int_{S^2} \left\| s_\theta(x, t) - \frac{\log_x z}{t} \right\|_2^2 \cdot \rho_0(z) \cdot \bar{\rho}(x, z, t) \, \mathrm{d}z \, \mathrm{d}x \, \mathrm{d}t. \tag{2.37}$$

The question of how to properly generalize score-based generative modeling to the manifold case is still somewhat open. For example, an entirely alternative approach is outlined in (Huang et al., 2022).

2.4 Experiments

We first demonstrate our method without the need for spherical alignment. We demonstrate our approach on 8 different meshes (see Table 2.1), which is significantly more than has been empirically demonstrated by other Riemannian generative modeling approaches. We train continuous normalizing flows and Moser flows on each mesh. The meshes and data used are from the ContactDB dataset (Brahmbhatt et al., 2019). Second, we show how our approach enables the sharing of data from distinct meshes approximating the same underlying manifold. We generate 5 similar but distinct meshes (derived from `stanford-bunny`) to show how a single generative model can be trained using several meshes corresponding to the same underlying manifold. We show that log likelihood tested on a held-out mesh improves with additional data, even when the data comes from different meshes. We also show how all 5 meshes can be used to simultaneously generate data on each of

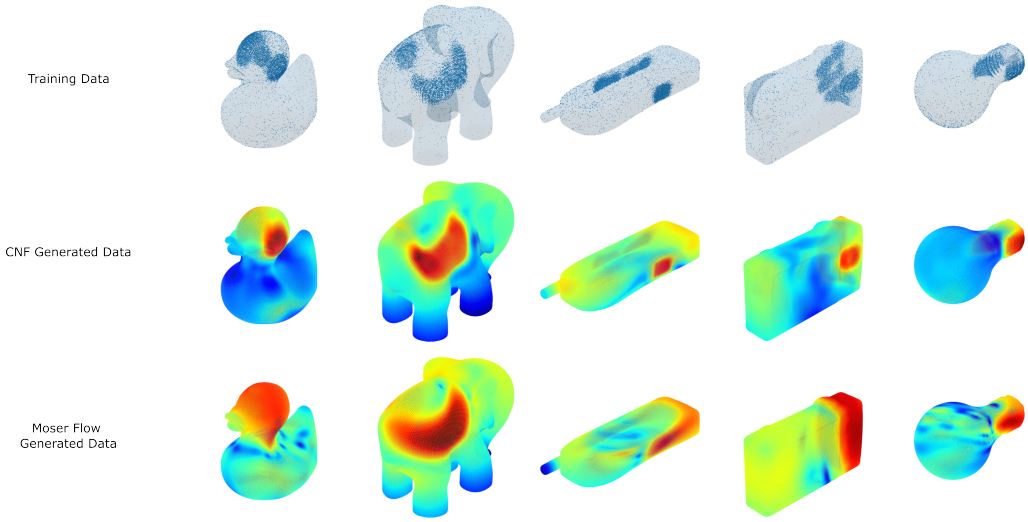


Figure 2.5: Showing five meshes: training data (top row), learned CNF densities (middle row), and learned Moser flow densities (bottom row). From left to right, the meshes are `rubber-duck`, `elephant`, `cell-phone`, `camera`, and `light-bulb`.

the 5 meshes. All experiments were run on a single NVIDIA RTX A6000 GPU. In all experiments, noise distributions are uniform on the sphere. Our code is available at github.com/vdorbs/spherical-generative-modeling.git.

2.4.1 Training Data

ContactDB (Brahmbhatt et al., 2019) is a collection of grasping and manipulation data for applications in human-robot interaction. The data were collected by thermally imaging 3D printed household objects after they were grasped by human subjects. We study 8 objects, each handled by a single participant. The data take the form of triangle meshes with a single feature (intensity) at each vertex, indicating thermal energy transferred to the object. As in (Brahmbhatt et al., 2019), we process the contact maps by passing the intensities through a sigmoid function, assigning 0.999 probability to the maximum intensity and 0.001 to the minimum. As in (Rozen et al., 2021), we average the corresponding probabilities on each face to obtain an unnormalized probability distribution over faces. We then draw 10000 samples from each mesh, each sample generated by sampling a face according to the unnormalized distribution, then by sampling a uniformly distributed point on the sampled face.

Table 2.1: Log likelihoods for 8 meshes (higher is better), averaged over 5 runs per mesh, plus or minus one standard deviation.

Mesh	CNF	Moser Flow
`camera`	4 ± 0.01703	3.726 ± 0.1048
`stanford-bunny`	4.354 ± 0.0435	4.168 ± 0.1937
`light-bulb`	4.44 ± 0.04114	4.239 ± 0.07521
`elephant`	4.472 ± 0.01215	4.462 ± 0.0361
`mouse`	4.769 ± 0.009795	4.726 ± 0.06756
`cell-phone`	5.465 ± 0.01964	5.302 ± 0.1207
`rubber-duck`	5.75 ± 0.007362	5.368 ± 0.1799
`banana`	6.591 ± 0.02889	6.318 ± 0.1723

2.4.2 Distribution Modeling Results

We run 5 trials for each of the 8 meshes, training both a CNF and a Moser flow. Validation results are listed in Table 2.1. In both cases we use 5000 training samples and 5000 validation samples. The CNF vector field and Moser flow flux field are both parametrized as 3-hidden layer neural networks with hidden dimensions of 32 and tanh nonlinearities. The inputs to both neural networks are vectors in $\mathbb{R}^3$, with the CNF taking a fourth input of time. Both models are trained with the Adam optimizer (Kingma and Ba, 2014). The CNF is trained for 100 epochs with a batch size of 256 and a learning rate of 10^{-2}. The Moser flow is trained for 4000 epochs with a batch size of 256 and a learning rate of 10^{-4}. These results show that our approach can reliably model distributions on multiple manifolds using multiple Riemannian generative models as subroutines.

2.4.3 Spherical Alignment

We next demonstrate our method when we have access to data on distinct meshes. For a single mesh, we sample data as in Section 2.4.1. We generate 5 distinct meshes by sampling 5 sets of vertices at random, each with only 1% of the vertices in the original mesh. The meshes are then obtained from Poisson surface reconstruction (Kazhdan, Bolitho, and Hoppe, 2006) with each point cloud. As reconstruction requires normal vectors at each point in a cloud, we use the vertex normals derived from the original mesh at each point rather than estimate normals from nearest neighbor tangent planes. The data from the original mesh is then partitioned into 5 equal subsets, with each subset projected onto a corresponding reconstructed mesh. First, we hold out data from one mesh and run 20 trials training a CNF with data

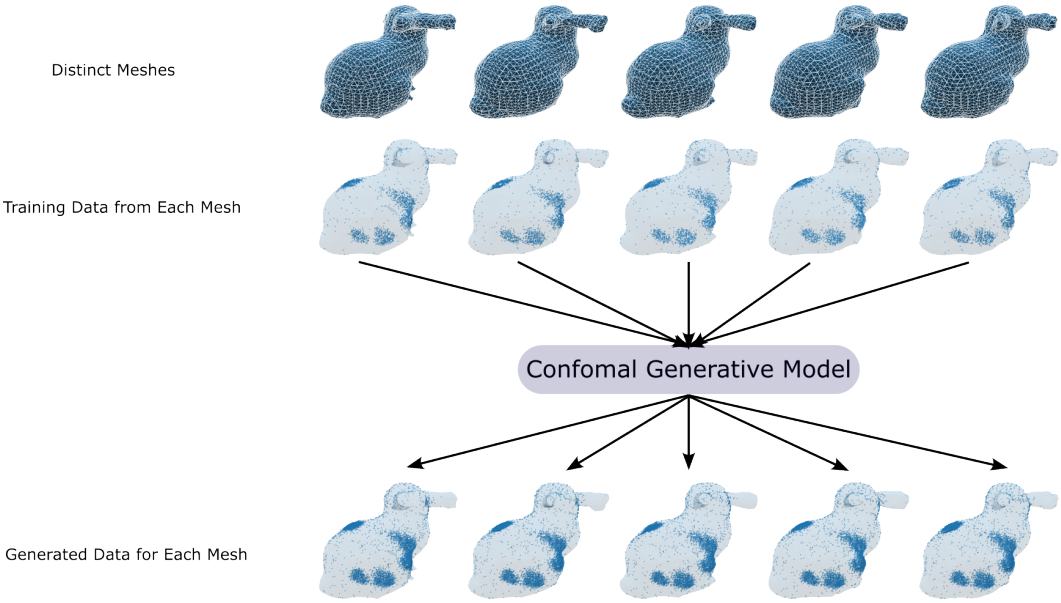


Figure 2.6: Distinct meshes (top row), training data on each mesh (second row) and samples generated from learned model (bottom row). The same model is used to generate data on each of the five meshes, trained using data aggregated from all meshes.

from 1, 2, 3, and all 4 of the remaining meshes. Each mesh has 2000 training data samples; we only use the first 1000 from each. Log likelihoods of data on the held-out mesh are shown in Figure 2.7. The CNF used has the same training configuration as in Section 2.4.2. Second, we train a single conformal generative model that can generate data on each of the five meshes. We use data from each of the five meshes to train the model, again using 1000 training samples and 1000 validation samples from each mesh. Generated samples are visualized in Figure 2.6. The CNF used has the same training configuration. These results demonstrate that data can be shared between distinct meshes for improved training, and that such models can readily generate data on qualitatively similar but unseen meshes.

2.5 Other Related Work

Generative Modeling on Meshes. The concurrent work of (Chen and Lipman, 2023) adopts a flow matching approach to generative modeling on Riemannian manifolds and, as a special case, triangle meshes. In this paradigm, a vector field is trained to match an average (weighted by the data distribution) of conditional vector fields, which are chosen for their relative simplicity. For triangle meshes, the conditional vector fields are constructed using kernel functions encoding approximate spectral distances. Both training and generation with the trained model

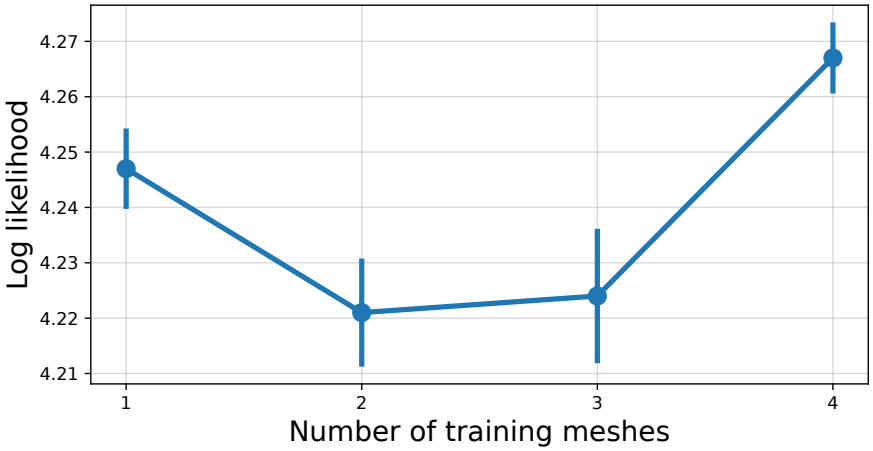


Figure 2.7: Performance of generative model on held-out mesh as more meshes and data are added.

require solving ODEs directly on the meshes, which can be difficult for complicated geometries.

Generative Modeling on Implicit Surfaces. An alternative approach to generative modeling on complicated manifolds is to a use a learned signed distance function (SDF) (Rozen et al., 2021), which defines an implicit surface (i.e., the SDF is zero if the point is on the surface). In general, learning SDFs can be quite complicated, and also lead to uncontrolled approximations of the original manifold. In contrast, our framework sidesteps this issue by leveraging tools in computational geometery.

Conformal Embedding Flows. Conformal transformations have also been applied to generative modeling problems in high dimensions under the manifold hypothesis (Ross and Cresswell, 2021). In this case, a (left) invertible map from a low-dimensional latent space to a high-dimensional data space must be learned jointly with a normalizing flow. The embedding is required to be conformal to tractably compute the corresponding change of density; this puts restrictions on the architecture used for manifold learning. Also, as with learning SDFs, manifold learning can be complicated with uncontrolled approximations of the original manifold.

Conformal Prediction. The term "conformal" is also used in the context of conformal prediction (Shafer and Vovk, 2008). Both conformal prediction and conformal geometry use a measure of how conformal a point is under a transformation—the former in terms of statistical calibration and the latter in terms of geometrical calibration. Otherwise, the two lines of research are unrelated.

Chapter 3

DATA-DRIVEN ROBUST OPTIMIZATION PROBLEMS

3.1 Introduction

Ensuring properties such as stability and safety is of significant importance in many modern control applications, including autonomous driving, industrial robotics, and aerospace vehicles. In practice, the models used to design these controllers are imperfect, with model uncertainty arising due to unmodeled dynamics and parametric errors. In the presence of such uncertainty, controllers may fail to render systems stable or safe. As real world control systems become increasingly complex, the potential for detrimental modeling errors increases, and thus it is critical to study control synthesis in the presence of uncertainty.

We propose a control synthesis process using *control certificate functions* (CCFs) (Dimitrova and Majumdar, 2014; Boffi et al., 2020) that incorporates a data-driven approach for capturing model uncertainty. CCFs generalize popular tools from nonlinear control for achieving stability and safety such as Control Lyapunov Functions (CLFs) (Artstein, 1983), Control Barrier Functions (CBFs) (Ames, Grizzle, and Tabuada, 2014), and Control Barrier-Lyapunov Functions (Prajna and Jadbabaie, 2004). CLFs and CBFs have been successfully deployed in the context of bipedal robotics (Ames, Galloway, et al., 2014; Nguyen et al., 2016), adaptive cruise control (Ames, Grizzle, and Tabuada, 2014), robotic manipulators (Khansari-Zadeh and Billard, 2014), and multi-agent systems (Pickem et al., 2017). Data-driven and machine learning based approaches have shown great promise for controlling systems with an uncertain model or with no model at all (Kober, Bagnell, and Peters, 2013; Shi et al., 2019; Cheng et al., 2019; Lee et al., 2020). The integration of techniques from nonlinear control theory for achieving stability and safety with data-driven methods has become increasingly popular (Aswani et al., 2013; Beckers, Kulić, and Hirche, 2019; Berkenkamp and Schoellig, 2015; Gillula and Tomlin, 2012; Qu et al., 2020), with many approaches relying on certificate functions for theoretical guarantees (Khansari-Zadeh and Billard, 2014; Lederer, Capone, and Hirche, 2020; Choi et al., 2020; Cohen and Belta, 2020; Castañeda et al., 2020).

Uncertainty in the effect of actuation remains a major challenge in achieving control-theoretic guarantees with data-driven methods. Many existing approaches assume

certainty in how actuation enters the dynamics (Fisac et al., 2018; Umlauft, Pöhler, and Hirche, 2018; Zheng et al., 2020), use structured controllers requiring strong characterizations of this uncertainty (Beckers, Kulić, and Hirche, 2019; Lederer, Capone, Umlauft, et al., 2020), or require high coverage datasets with (nearly) complete characterizations of the input-to-state relationship (Berkenkamp, Moriconi, et al., 2016; Berkenkamp, Turchetta, et al., 2017). In practice, collecting such data can be prohibitively costly or damage the system, suggesting a need for data-driven approaches that accommodate actuation uncertainty without requiring this complete characterization.

The contribution of this work is a novel approach for robust data-driven control synthesis via CCFs for control-affine systems with model uncertainty, including actuation uncertainty, which is broadly applicable in many real-world settings such as robotic (Murray, Li, and Sastry, 1994) and automotive systems (Ioannou and Chien, 1993). In Section 3.3 we incorporate data into a convex optimization-based control synthesis problem as affine inequality constraints which restrict possible model uncertainties. This enables the choice of robust control inputs over convex uncertainty sets. Rather than requiring a full characterization of how input enters the system, this approach utilizes the affine structure of CCF dynamics to choose inputs. This reduces the impact of actuation uncertainty on the evolution of the certificate function and allows for guarantees of stability and safety. The proposed approach provides a unique perspective for unifying nonlinear control and non-parametric machine learning that is well positioned to study both theoretical and application oriented questions at this intersection.

We then further build towards robust control of control-affine systems in the *sampled data* setting, in which control inputs are held constant over fixed frequency sample periods. Here, we make use of recent work (Taylor, Dorobantu, Yue, et al., 2021; Taylor, Dorobantu, Cosner, et al., 2022) that enables control synthesis with a variety of approximate discrete-time models. We demonstrate how our uncertainty modeling can be applied to these settings as well, and identify some unifying structures in all of our problem settings of interest. Through Lagrangian duality, we find reformulations and relaxations that enable us to carry stability and safety guarantees over from continuous-time synthesis. We conclude by demonstrating how complex uncertainty sets can be outer approximated to simplify robust optimization while maintaining robustness.

3.2 Background

This section provides a review of certificate-based nonlinear control synthesis and an overview of how model uncertainty impacts these synthesis methods.

3.2.1 Control Certificate Functions

Consider a nonlinear control affine system given by:

$$\dot{x} = f(x) + g(x)u, \tag{3.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz continuous on $\mathbb{R}^n$. Further assume that the origin is an equilibrium point of the uncontrolled system ($f(0) = 0$). In this work we assume that u may be chosen unbounded as in (Khansari-Zadeh and Billard, 2014; Jankovic, 2018). Given a locally Lipschitz continuous state-feedback controller $k : \mathbb{R}^n \to \mathbb{R}^m$, the closed-loop system dynamics are:

$$\dot{x} = f_{\text{cl}}(x) \triangleq f(x) + g(x)k(x). \tag{3.2}$$

The assumption on local Lipschitz continuity of f, g, and k implies that f_{cl} is locally Lipschitz continuous. Thus for any initial condition $x_0 \triangleq x(0) \in \mathbb{R}^n$ there exists a maximum time interval $I(x_0) = [0, t_{\max})$ such that $x(t)$ is the unique solution to (3.2) on $I(x_0)$ (Perko, 2013).

The qualitative behavior (such as stability or safety) of the the closed-loop system (3.2) can be certified via the notion of a continuously differentiable *certificate function* $\phi : \mathbb{R}^n \to \mathbb{R}$. Given a *comparison function* $\alpha : \mathbb{R} \to \mathbb{R}$ (specific to the qualitative behavior of interest), certification is specified as an inequality on the derivative of the certificate function along solutions to the closed-loop system:

$$\dot{\phi}(x) = \nabla\phi(x)^\top f_{\text{cl}}(x) \leq -\alpha(\phi(x)) \,. \tag{3.3}$$

Synthesis of controllers that satisfy (3.3) by design motivates the following definition of a *Control Certificate Function*:

Definition 1 (Control Certificate Function (CCF))**.** A continuously differentiable function $\phi : \mathbb{R}^n \to \mathbb{R}$ is a *Control Certificate Function (CCF)* for (3.1) with comparison function $\alpha : \mathbb{R} \to \mathbb{R}$ if for all $x \in \mathbb{R}^n$:

$$\inf_{u\in\mathbb{R}^m} \underbrace{\nabla\phi(x)^\top f(x)}_{L_f\phi(x)} + \underbrace{\nabla\phi(x)^\top g(x)}_{L_g\phi(x)^\top} u \leq -\alpha(\phi(x)), \tag{3.4}$$

where $L_f\phi : \mathbb{R}^n \to \mathbb{R}$ and $L_g\phi : \mathbb{R}^n \to \mathbb{R}^m$.

The control-affine nature of the system dynamics are preserved by the CCF, such that the only component of the input that impacts the evolution of the certificate function lies in the direction of $L_g\phi(x)$. Given a CCF ϕ for (3.1) and a corresponding comparison function α, we define the set-valued function of all control values that satisfy the inequality in (3.4):

$$K_{\text{ccf}}(x) \triangleq \left\{u \in \mathbb{R}^m : L_f\phi(x) + L_g\phi(x)u \leq -\alpha(\phi(x))\right\}. \tag{3.5}$$

Any nominal locally Lipschitz continuous controller $k_d : \mathbb{R}^n \to \mathbb{R}^m$ can be modified to take values in the set $K_{\text{ccf}}(x)$ via the certificate-critical CCF-QP:

$$\begin{aligned} k(x) \in \underset{u\in\mathbb{R}^m}{\arg\min}\ & \frac{1}{2}\|u - k_d(x)\|_2^2 \\ \text{s.t. }\ & \nabla\phi(x)^\top (f(x) + g(x)u) \leq -\alpha(\phi(x)). \end{aligned} \tag{CCF-QP}$$

Before providing particular examples of certificate functions useful in control synthesis, we review the following definitions. We denote a continuous function $\alpha : [0, a) \to \mathbb{R}_+$, with $a > 0$, as *class* $\mathcal{K}$ ($\alpha \in \mathcal{K}$) if $\alpha(0) = 0$ and α is strictly monotonically increasing. If $a = \infty$ and $\lim_{r\to\infty} \alpha(r) = \infty$, then α is *class* $\mathcal{K}_\infty$ ($\alpha \in \mathcal{K}_\infty$). A continuous function $\alpha : (-b, a) \to \mathbb{R}$, with $a, b > 0$, is *extended class* $\mathcal{K}$ ($\alpha \in \mathcal{K}_e$) if $\alpha(0) = 0$ and α is strictly monotonically increasing. If $a, b = \infty$, $\lim_{r\to\infty} \alpha(r) = \infty$, and $\lim_{r\to-\infty} \alpha(r) = -\infty$, then α is *extended class* $\mathcal{K}_\infty$ ($\alpha \in \mathcal{K}_{\infty,e}$). Finally, we note that $c \in \mathbb{R}$ is referred to as a *regular value* of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ if $h(x) = c \implies \nabla h(x) \neq 0$.

Example 1 (Stability via Control Lyapunov Functions)**.** In the context of stabilization to the origin, a control certificate function $V : \mathbb{R}^n \to \mathbb{R}$ with a class $\mathcal{K}$ comparison function $\alpha \in \mathcal{K}$ that satisfies:

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \tag{3.6}$$

for $\alpha_1, \alpha_2 \in \mathcal{K}$, is a *Control Lyapunov Function (CLF)* (Artstein, 1983; Sontag, 1989a), with stabilization to the origin achieved by controllers taking values in the set-valued function K_{ccf} given by (3.5) (Ames, Xu, et al., 2016).

Example 2 (Safety via Control Barrier Functions)**.** In the context of safety, defined as forward invariance (Blanchini, 1999) of a set $\mathcal{S}$, a control certificate function $h : \mathbb{R}^n \to \mathbb{R}$ with 0 a regular value and a comparison function $\alpha \in \mathcal{K}_{\infty,e}$ that satisfies:

$$x \in \mathcal{S} \implies h(x) \leq 0, \tag{3.7}$$

is a *Control Barrier Function (CBF)* (Ames, Grizzle, and Tabuada, 2014; Ames, Xu, et al., 2016), with safety of the set $\mathcal{S}$ achieved by controllers taking values in the set-valued function K_{ccf} given by (3.5) (Ames, Coogan, et al., 2019). We adopt the opposite sign convention for h so satisfying (3.4) guarantees safety.

3.2.2 Model Uncertainty

In practice, uncertainty in the system dynamics (3.1) exists due to parametric error and unmodeled dynamics, such that the functions f and g are not precisely known. Control affine systems are a natural setting to study actuation uncertainty as the function g can be seen as an uncertain gain multiplying the input. In this context, control synthesis is done with a nominal model that estimates the true system dynamics:

$$\widehat{\dot{x}} = \widehat{f}(x) + \widehat{g}(x)u, \tag{3.8}$$

where $\widehat{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\widehat{g} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz continuous. Adding and subtracting this expression to and from (3.1) implies the system evolution is described by:

$$\dot{x} = \widehat{f}(x) + \widehat{g}(x)u + \underbrace{f(x) - \widehat{f}(x)}_{\widetilde{f}(x)} + \underbrace{(g(x) - \widehat{g}(x))}_{\widetilde{g}(x)}\, u, \tag{3.9}$$

where $\widetilde{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\widetilde{g} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are the unmodeled dynamics. This uncertainty in the dynamics additionally manifests in the time derivative of a CCF for the system:

$$\dot{\phi}(x,u) = \overbrace{\underbrace{\nabla\phi(x)^\top \widehat{f}(x)}_{L_{\widehat{f}}\phi(x)} + \underbrace{\nabla\phi(x)^\top \widehat{g}(x)}_{L_{\widehat{g}}\phi(x)^\top} u}^{\widehat{\dot{\phi}}(x,u)} + \underbrace{\nabla\phi(x)^\top \widetilde{f}(x)}_{L_{\widetilde{f}}\phi(x)} + \underbrace{\nabla\phi(x)^\top \widetilde{g}(x)}_{L_{\widetilde{g}}\phi(x)^\top} u, \tag{3.10}$$

where $L_{\widehat{f}}\phi, L_{\widetilde{f}}\phi : \mathbb{R}^n \to \mathbb{R}$, and $L_{\widehat{g}}\phi, L_{\widetilde{g}}\phi : \mathbb{R}^n \to \mathbb{R}^m$. The presence of uncertainty in the CCF time derivative makes it impossible to verify whether a given control input is in the set $K_{\text{ccf}}(x)$ given in (3.5), and can lead to failure to achieve the desired qualitative behavior.

Assumption 1. The function $\phi : \mathbb{R}^n \to \mathbb{R}$ is a valid CCF with comparison function $\alpha : \mathbb{R} \to \mathbb{R}$ for the true dynamic system (3.9). Mathematically this assumption appears as:

$$\inf_{u\in\mathbb{R}^m} \nabla\phi(x)^\top \left(f(x) + g(x)u\right) \leq -\alpha(\phi(x)).$$

This assumption is structural in nature and can be met for feedback linearizable systems (such as robotic systems).

Remark 1. Though many approaches to CCF design (linearization, energy-based, numerical sums-of-squares methods) rely on knowledge of the true dynamics, it is also possible to design CCFs without explicit knowledge of the true system dynamics. For example, a valid CLF and comparison function for the true system can be designed via feedback linearization assuming only knowledge of the degree of actuation (see (Taylor, Dorobantu, Le, et al., 2019) for full details). This method also works to specify CBFs which are defined by sublevel sets of CLFs (as in our simulation results). We emphasize the difference between choosing a qualitative behavior that the system can be made to satisfy (e.g. the CCF) and actually designing the control inputs which achieve the behavior. Our work focuses on the latter: solving the problem of choosing stable/safe inputs in the presence of uncertainty.

Assumption 2. The functions $\widetilde{f}$ and $\widetilde{g}$ are globally Lipschitz continuous with known Lipschitz constants $\mathcal{L}_{\widetilde{f}}$ and $\mathcal{L}_{\widetilde{g}}$.

Remark 2. Knowledge of minimal Lipschitz constants is not necessary, but smaller Lipschitz constants are associated with improved performance of data-driven robust control methods.

3.3 Data-Driven Robust Control

In this section we explore how data can be incorporated directly into an optimization-based controller to robustly achieve a desired qualitative behavior specified via a Control Certificate Function.

Consider a dataset consisting of N tuples of states, inputs, and corresponding state time derivatives, $\mathcal{D} = \{(x_i, u_i, \dot{x}_i)\}_{i=1}^{N}$, with $x_i \in \mathbb{R}^n$, $u_i \in \mathbb{R}^m$, and $\dot{x}_i \in \mathbb{R}^n$ for $i = 1, \ldots, N$. It may not be possible to directly measure the state time derivatives $\dot{x}_i$, but they can approximated from sequential state observations x_i (Taylor, Dorobantu, Le, et al., 2019; Taylor, Singletary, et al., 2020). For simplicity of exposition, we do not consider noise in this formulation. However, the resulting construction may be modified to account for the impact of bounded noise in the data. Considering the uncertain model (3.9) evaluated at a state and input pair (x_i, u_i) in the data set yields:

$$\widetilde{F}_i \triangleq \dot{x}_i - (\widehat{f}(x_i) + \widehat{g}(x_i)u_i) = \widetilde{f}(x_i) + \widetilde{g}(x_i)u_i, \tag{3.11}$$

where $\widetilde{F}_i \in \mathbb{R}^n$ can be interpreted as the error between the true state time derivative and the nominal model (3.8) evaluated at the state and input pair (x_i, u_i).

Considering a state $x \in \mathbb{R}^n$ (not necessarily present in the dataset $\mathcal{D}$), the second equality in (3.11) implies:

$$\widetilde{f}(x) + \widetilde{g}(x)u_i - \widetilde{F}_i = \widetilde{f}(x) - \widetilde{f}(x_i) + (\widetilde{g}(x) - \widetilde{g}(x_i))\, u_i. \tag{3.12}$$

This expression provides a relationship between the possible values of the unmodeled dynamics $\widetilde{f}$ and $\widetilde{g}$ at the state x and the values of the unmodeled dynamics at the data point x_i. Using the local Lipschitz continuity of the unmodeled dynamics yields the following bound:

$$\begin{aligned}\left\|\widetilde{f}(x) + \widetilde{g}(x)u_i - \widetilde{F}_i\right\|_2 &= \left\|\widetilde{f}(x) - \widetilde{f}(x_i) + (\widetilde{g}(x) - \widetilde{g}(x_i))\, u_i\right\|_2, \\ &\leq \left(\mathcal{L}_{\widetilde{f}} + \mathcal{L}_{\widetilde{g}}\|u_i\|_2\right)\|x - x_i\|_2 \triangleq \varepsilon_i(x).\end{aligned} \tag{3.13}$$

We see that the bound grows with the magnitude of the Lipschitz constants $\mathcal{L}_{\widetilde{f}}$ and $\mathcal{L}_{\widetilde{g}}$ and distance of the state x from the data point x_i. The values of $\mathcal{L}_{\widetilde{f}}$ and $\mathcal{L}_{\widetilde{g}}$ are not explicitly data dependent, and thus the bound can be improved for a given data set by reducing the possible model uncertainty through improved modeling. Given this construction we may define the uncertainty set-valued function:

$$\mathcal{U}_i(x) \triangleq \left\{(A, b) \in \mathbb{R}^{n\times m} \times \mathbb{R}^n \;:\; \left\|b + Au_i - \widetilde{F}_i\right\|_2 \leq \epsilon_i(x)\right\}, \tag{3.14}$$

noting that $(\widetilde{g}(x), \widetilde{f}(x)) \in \mathcal{U}_i(x)$ and $\mathcal{U}_i(x)$ is convex. Considering this construction over the entire data set $\mathcal{D}$ we define:

$$\mathcal{U}(x) \triangleq \bigcap_{i=1}^{N} \mathcal{U}_i(x) \subset \mathbb{R}^{n\times m} \times \mathbb{R}^n, \tag{3.15}$$

noting that $(\widetilde{g}(x), \widetilde{f}(x)) \in \mathcal{U}(x)$ and $\mathcal{U}(x)$ is convex. Therefore, $\mathcal{U}(x)$ consists of all possible model errors that are consistent with the observed data. This allows us to pose the following data robust control problem:

Definition 2 (Data Robust Control Certificate Function Optimization Problem)**.**

$$\begin{aligned} k_{\text{rob}}(x) \in \underset{u\in\mathbb{R}^m}{\arg\min}\;\; & \frac{1}{2}\|u - k_d(x)\|_2^2 \\ \text{s.t.}\;\; & \widehat{\dot{\phi}}(x, u) + \nabla\phi(x)^\top (b + Au) \leq -\alpha(\phi(x)) \text{ for all } (A, b) \in \mathcal{U}(x). \end{aligned} \tag{DR-CCF-OP}$$

By construction we have that $(\widetilde{g}(x), \widetilde{f}(x)) \in \mathcal{U}(x)$, implying that $k_{\text{rob}}(x) \in K_{\text{ccf}}(x)$ when the problem is feasible. Thus the closed-loop system (3.2) under k_{rob} satisfies inequality (3.3).

3.4 Data-Driven Robust Sampled Data Control

We next consider the setting in which control inputs are held constant for fixed frequency sample periods, referred to as the *sampled data* setting. We developed approaches for sampled data stability in (Taylor, Dorobantu, Yue, et al., 2021) and sampled data safety in (Taylor, Dorobantu, Cosner, et al., 2022); both approaches are outlined in Appendix A.

We can use the same uncertainty model and uncertainty set functions developed in the previous section, even in the sampled data setting. However, the inequality constraints will need to be modified, and since sampled data stability and safety currently comprise two separate mathematical approaches, we will not consider the case of a sampled data "certificate."

Unlike the continuous-time case, the type of approximate map used and Lyapunov/barrier function used must be chosen in a compatible manner to yield a tractable optimization problem. Specifically, we will consider quadratic Lyapunov functions, affine barrier functions, and quadratic *configuration* barrier functions. We refer to a barrier function as a *configuration* barrier function if it is purely a function of the configuration of a robotic system (not the velocity). In the first two cases, we use the approximate Euler family (Equation (A.5)), and in the last case we use the "midpoint" two-stage Runge-Kutta family (Equation (A.11)).

3.4.1 Constraint Equations without Uncertainty

The constraint equations are derived from Equation (A.95) and Equation (A.98). Without uncertainty, the sampled data stability constraint is:

$$h(f(x)+g(x)u)^\top P(f(x)+g(x)u)+2x^\top P(f(x)+g(x)u)\leq -c\lambda_{\min}(Q)\|x\|_2^2, \tag{3.16}$$

for all $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, where $P \in \mathbb{S}^n_{++}$ is the matrix characterizing a Lyapunov function for linear closed-loop dynamics (obtained from full-state feedback linearization), $Q \in \mathbb{S}^n_{++}$ and $c \in (0,1)$ are user choices, and $h \in \mathbb{R}_{++}$ is the sample period. Together with the closed-loop dynamics, P should solve the continuous time Lyapunov equation (with right hand side $-Q$).

A affine barrier function $s : \mathbb{R}^n \to \mathbb{R}$ of the form:

$$s(x) = c^\top x + d, \tag{3.17}$$

for all states $x \in \mathbb{R}^n$ generates a corresponding sampled data safety constraint (again without uncertainty) as:

$$c^\top(x + h(f(x) + g(x)u)) + d - (c^\top x + d) \geq -h\alpha(c^\top x + d), \tag{3.18}$$

or:

$$c^\top(f(x) + g(x)u) \geq -\alpha(c^\top x + d), \tag{3.19}$$

for all $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, where $\alpha \in \mathcal{K}$.

Consider a quadratic configuration barrier $s : \mathbb{R}^\ell \to \mathbb{R}$ (where ℓ is the dimension of the configuration space and $n = 2\ell$) of the form:

$$s(q) = c - q^\top P q, \tag{3.20}$$

for all configurations $q \in \mathbb{R}^\ell$, where c is a constant and $P \in \mathbb{S}^\ell_{++}$ characterizes ellipsoidal sublevel sets. This function generates a corresponding sampled data safety constraint (again without uncertainty) as:

$$s\left(q + h\dot{q} + \frac{h^2}{2}\ddot{q}\right) - s(q) \geq -h\alpha(s(q)), \tag{3.21}$$

for all $q, \dot{q} \in \mathbb{R}^\ell$ and $u \in \mathbb{R}^m$, where $\alpha \in \mathcal{K}$. The acceleration $\ddot{q}$ has a control affine form, with:

$$\ddot{q} = f_2(q, \dot{q}) + g_2(q, \dot{q})u. \tag{3.22}$$

3.4.2 Constraint Equations with Uncertainty

To introduce our uncertainty model and perform robust optimization, note that:

$$f(x) + g(x)u = \begin{bmatrix} u_1 I & \cdots & u_m I & I \end{bmatrix} \begin{bmatrix} g_1(x) \\ \vdots \\ g_m(x) \\ f(x) \end{bmatrix}, \tag{3.23}$$

for all states $x \in \mathbb{R}^n$ and control inputs $u \in \mathbb{R}^m$, where $g_1, \ldots, g_m$ denote the columns of g and $u_1, \ldots, u_m$ denote the coordinates of u. As such, define the affine functions $U : \mathbb{R}^m \to R^{n \times (m+1)n}$ and the nonlinear functions $z, \hat{z}, \tilde{z} : \mathbb{R}^n \to \mathbb{R}^{(m+1)n}$ as:

$$U(u) = \begin{bmatrix} u_1 I & \cdots & u_m I & I \end{bmatrix}, \tag{3.24}$$

$$z(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_m(x) \\ f(x) \end{bmatrix}, \qquad \widehat{z}(x) = \begin{bmatrix} \widehat{g}_1(x) \\ \vdots \\ \widehat{g}_m(x) \\ \widehat{f}(x) \end{bmatrix}, \qquad \widetilde{z}(x) = \begin{bmatrix} \widetilde{g}_1(x) \\ \vdots \\ \widetilde{g}_m(x) \\ \widetilde{f}(x) \end{bmatrix}, \tag{3.25}$$

for all states $x \in \mathbb{R}^n$ and control inputs $u \in \mathbb{R}^m$. Note that:

$$f(x) + g(x)u = U(u)z(x), \tag{3.26}$$

$$\widehat{f}(x) + \widehat{g}(x)u = U(u)\widehat{z}(x), \tag{3.27}$$

$$\widetilde{f}(x) + \widetilde{g}(x)u = U(u)\widetilde{z}(x), \tag{3.28}$$

for all states $x \in \mathbb{R}^n$ and control inputs $u \in \mathbb{R}^m$. We can now write our uncertainty set-valued function as:

$$\mathcal{U}(x) = \left\{\theta \in \mathbb{R}^{(m+1)n} : \|U(u_i)\theta - (\dot{x}_i - U(u_i)\widehat{z}(x_i))\|_2 \leq \varepsilon_i(x), i = 1, \ldots, N\right\}, \tag{3.29}$$

for all query states $x \in \mathbb{R}^n$. We can now rewrite every sampled data constraint as:

$$hz(x)^\top U(u)^\top PU(u)z(x) + 2x^\top PU(u)z(x) \leq -c\lambda_{\min}(Q)\|x\|_2^2, \tag{3.30}$$

$$c^\top U(u)z(x) \geq -\alpha(c^\top x + d), \tag{3.31}$$

$$s\left(q + h\dot{q} + \frac{h^2}{2}U(u)z_2(q, \dot{q})\right) - s(q) \geq -h\alpha(s(q)), \tag{3.32}$$

where z_2 is defined analogously to z. In this form, we can more easily make use of the fact that $z(x) = \widehat{z}(x) + \widetilde{z}(x)$. In particular, at a query state $x \in \mathbb{R}^n$, we know that there is some ground truth uncertainty vector $\theta \in \mathcal{U}(x)$ such that $z(x) = \widehat{z}(x) + \theta$. Therefore, we can write:

$$h(\widehat{z}(x) + \theta)^\top U(u)^\top PU(u)(\widehat{z}(x) + \theta) + 2x^\top PU(u)(\widehat{z}(x) + \theta) \leq -c\lambda_{\min}(Q)\|x\|_2^2 \tag{3.33}$$

$$c^\top U(u)(\widehat{z}(x) + \theta) \geq -\alpha(c^\top x + d), \tag{3.34}$$

$$s\left(q + h\dot{q} + \frac{h^2}{2}U(u)(\widehat{z}_2(q, \dot{q}) + \theta)\right) - s(q) \geq -h\alpha(s(q)). \tag{3.35}$$

To solve robust optimization problems with these constraints, we will have to compute global extrema of the constraint equations over *all* possible values of θ that are consistent with the data. Therefore, we rewrite the constraint equations to more clearly expose their quadratic and affine (in θ) forms. First, in the sampled data stabilization case, we have:

$$\begin{aligned}\theta^\top (hU(u)^\top PU(u))\theta + 2(U(u)^\top P(x + \widehat{z}(x)))^\top \theta \\ + h\widehat{z}(x)^\top U(u)^\top PU(u)\widehat{z}(x) + 2x^\top PU(u)\widehat{z}(x) + c\lambda_{\min}(Q)\|x\|_2^2 \leq 0.\end{aligned} \tag{3.36}$$

Note that the constant (in θ) term is simply Equation (3.30) when the nominal model is assumed to be perfect. For the affine barrier, we have:

$$(U(u)^\top c)^\top \theta + c^\top U(u)\widehat{z}(x) + \alpha(c^\top x + d) \leq 0. \tag{3.37}$$

For the quadratic configuration barrier, we have:

$$
\begin{aligned}
&-\theta^\top \left(\frac{h^4}{4}U(u)^\top P U(u)\right)\theta - 2\left(\frac{h^2}{2}U(u)^\top P\left(q + h\dot{q} + \frac{h^2}{2}U(u)\widehat{z}_2(q,\dot{q})\right)\right)^\top \theta \\
&\quad - \left(q + h\dot{q} + \frac{h^2}{2}U(u)\widehat{z}_2(q,\dot{q})\right)^\top P\left(q + h\dot{q} + \frac{h^2}{2}U(u)\widehat{z}_2(q,\dot{q})\right) \\
&\quad + q^\top P q \geq -h\alpha(c - q^\top P q). \qquad (3.38)
\end{aligned}
$$

By negating this inequality and factoring out h, we obtain:

$$
\begin{aligned}
&\theta^\top \left(\frac{h^3}{4}U(u)^\top P U(u)\right)\theta + 2\left(\frac{h}{2}U(u)^\top P\left(q + h\dot{q} + \frac{h^2}{2}U(u)\widehat{z}_2(q,\dot{q})\right)\right)^\top \theta \\
&\quad + h\left(\dot{q} + \frac{h}{2}U(u)\widehat{z}_2(q,\dot{q})\right)^\top P\left(\dot{q} + \frac{h}{2}U(u)\widehat{z}_2(q,\dot{q})\right) \\
&\quad + 2q^\top P\left(\dot{q} + \frac{h}{2}U(u)\widehat{z}_2(q,\dot{q})\right) - \alpha(c - q^\top P q) \leq 0. \qquad (3.39)
\end{aligned}
$$

3.4.3 Control Certificate Functions Revisited

To help unify constraint equation structures, note that:

$$\nabla\phi(x)^\top (f(x) + g(x)u) = \nabla\phi(x)^\top (U(u)\widehat{z}(x) + \theta), \qquad (3.40)$$

for some $\theta \in \mathcal{U}(x)$, so we can rewrite the corresponding inequality constraint as:

$$(U(u)^\top \nabla\phi(x))^\top \theta + \nabla\phi(x)^\top U(u)\widehat{z}(x) + \alpha(\phi(x)) \leq 0. \qquad (3.41)$$

Note that this structure mirrors that of the sampled data affine barrier.

3.4.4 Unifying Structures

Ultimately, the constraints we consider are either affine or quadratic in the uncertainty and affine or quadratic in the control input. Specifically, letting $p \in \mathbb{N}$ denote the uncertainty vector dimension, we either consider constraints of the form:

$$a(x,u)^\top \theta + b(x,u) \leq 0, \qquad (3.42)$$

where $a : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and $b : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are affine in their second argument, or:

$$\theta^\top P(x,u)\theta + 2q(x,u)^\top \theta + r(x,u) \leq 0, \qquad (3.43)$$

where $P : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{S}^p_+$, $q : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are quadratic in their second arguments.

3.5 Robust Optimization

We now outline how to extremize (in particular, maximize) the constraint equations over all possible uncertainties in their unified forms. In both cases, for fixed state $x \in \mathbb{R}^n$ and control input $u \in \mathbb{R}^m$, the constraint equations are convex in the uncertainty θ. For affine constraints (Equation (3.42)), this maximization is still realizable as a convex program, but quadratic constraints (Equation (3.43)) will necessitate a convex relaxation and a reformulation of the uncertainty set. We therefore outline the robust optimization procedures separately.

3.5.1 Affine Constraints

For affine constraints (Equation (3.42)), we write the uncertainty set as:

$$\mathcal{U}(x) = \{\theta \in \mathbb{R}^p : \|A_i\theta + b_i\|_2 \leq d_i(x), i = 1, \ldots, N\}, \tag{3.44}$$

where for each $i \in \{1, \ldots, N\}$, $A_i \in \mathbb{R}^{k \times p}$, $b_i \in \mathbb{R}^k$, and $d_i(x) \in \mathbb{R}_+$. Here, $k \in \mathbb{N}$ is the dimension of a second-order cone.

In a state $x \in \mathbb{R}^n$, an input $u \in \mathbb{R}^m$ is robustly feasible if the optimal value of the optimization problem:

$$\begin{aligned} \sup_{\theta \in \mathbb{R}^p} \quad & a(x,u)^\top \theta + b(x,u) \\ \text{s.t.} \quad & \|A_i\theta + b_i\|_2 \leq d_i(x), \qquad i = 1, \ldots, N, \end{aligned} \tag{3.45}$$

is nonpositive. The corresponding Lagrangian is:

$$\begin{aligned} \mathcal{L}(\theta, (\lambda_1, \nu_1), \ldots, (\lambda_N, \nu_N); x, u) &= a(x,u)^\top \theta + b(x,u) + \sum_{i=1}^N \begin{bmatrix} \lambda_i \\ \nu_i \end{bmatrix}^\top \begin{bmatrix} A_i\theta + b_i \\ d_i(x) \end{bmatrix} \\ &= \left(a(x,u) + \sum_{i=1}^N A_i^\top \lambda_i \right)^\top \theta + b(x,u) + \sum_{i=1}^N (\lambda_i^\top b_i + \nu_i d_i(x)), \end{aligned} \tag{3.46}$$

where for each $i \in \{1, \ldots, N\}$, the Lagrange multiplier vector $(\lambda_i, \nu_i) \in \mathbb{R}^{k+1}$ is in the k-dimensional second order cone, with $\|\lambda_i\|_2 \leq \nu_i$. To formulate upper bounds on the optimal value of the optimization problem (through weak duality), for every choice of Lagrange multipliers, we maximize the Lagrangian over θ. These upper bounds are vacuously infinite unless $a(x,u) + \sum_{i=1}^N A_i^\top \lambda_i = 0$. The minimal upper

bound we can determine is given by the optimal value of the optimization problem:

$$\begin{aligned} \inf_{\substack{\lambda_1,\ldots,\lambda_N \in \mathbb{R}^k \\ \nu \in \mathbb{R}^N}} \quad & b(x,u) + \sum_{i=1}^{N} (\lambda_i^\top b_i + \nu_i d_i(x)) \\ \text{s.t.} \quad & a(x,u) + \sum_{i=1}^{N} A_i^\top \lambda_i = 0, \\ & \|\lambda_i\|_2 \leq \nu_i, \qquad i = 1, \ldots, N. \end{aligned} \tag{3.47}$$

We can simplify this problem further by noting that for each $i \in \{1, \ldots, N\}$, since $d_i(x) \geq 0$, there is no reason to choose ν_i any greater than $\|\lambda_i\|_2$. Therefore, our minimal upper bound is given by the optimal value of:

$$\begin{aligned} \inf_{\lambda_1,\ldots,\lambda_N \in \mathbb{R}^k} \quad & b(x,u) + \sum_{i=1}^{N} (\lambda_i^\top b_i + \|\lambda_i\|_2 d_i(x)) \\ \text{s.t.} \quad & a(x,u) + \sum_{i=1}^{N} A_i^\top \lambda_i = 0. \end{aligned} \tag{3.48}$$

With a cost function $c : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ that is convex in its second argument, we can instantiate the following robust optimization problem:

$$\begin{aligned} \inf_{\substack{u \in \mathbb{R}^m \\ \lambda_1,\ldots,\lambda_N \in \mathbb{R}^k}} \quad & c(x,u) \\ \text{s.t.} \quad & b(x,u) + \sum_{i=1}^{N} (\lambda_i^\top b_i + \|\lambda_i\|_2 d_i(x)) \leq 0 \\ & a(x,u) + \sum_{i=1}^{N} A_i^\top \lambda_i = 0. \end{aligned} \tag{3.49}$$

This robust optimization problem is a generalization of that in (Taylor, Dorobantu, Dean, et al., 2021). Note that because c is convex in u and a and b are affine in u, this is a convex problem and can be solved efficiently.

3.5.2 Quadratic Constraints

The quadratic constraint case (Equation (3.43)) provides additional challenges. In a state $x \in \mathbb{R}^n$, an input $u \in \mathbb{R}^m$ is robustly feasible if the optimal value of the optimization problem:

$$\begin{aligned} \sup_{\theta \in \mathbb{R}^p} \quad & \theta^\top P(x,u)\theta + 2q(x,u)^\top \theta + r(x,u) \\ \text{s.t.} \quad & \|A_i\theta + b_i\|_2 \leq d_i(x), \qquad i = 1, \ldots, N, \end{aligned} \tag{3.50}$$

is nonpositive. As previously mentioned, since P takes positive semidefinite values, this is a nonconvex problem. However, we can still search for upper bounds via weak duality. To facilitate this, we slightly modify the above optimization problem to:

$$\begin{aligned} \sup_{\theta \in \mathbb{R}^p} \quad & \theta^\top P(x,u)\theta + 2q(x,u)^\top \theta + r(x,u) \\ \text{s.t.} \quad & \theta^\top A_i^\top A_i \theta + 2(A_i^\top b_i)^\top \theta + \|b_i\|_2^2 - d_i(x)^2 \leq 0, \qquad i = 1, \ldots, N. \end{aligned} \tag{3.51}$$

While this modification does not affect the optimal value, it does affect the dual optimization problem. The corresponding Lagrangian is:

$$\begin{aligned} \mathcal{L}(\theta, \lambda; x, u) &= \theta^\top P(x,u)\theta + 2q(x,u)^\top \theta + r(x,u) \\ &\quad - \sum_{i=1}^N \lambda_i \left(\theta^\top A_i^\top A_i \theta + 2(A_i^\top b_i)^\top \theta + \|b_i\|_2^2 - d_i^2 \right) \\ &= \theta^\top \left(P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i \right) \theta + 2 \left(q(x,u) - \sum_{i=1}^N \lambda_i A_i^\top b_i \right)^\top \theta \\ &\quad + r(x,u) - \sum_{i=1}^N \lambda_i (\|b_i\|_2^2 - d_i(x)^2), \end{aligned} \tag{3.52}$$

where $\lambda \in \mathbb{R}_+^N$ is a vector of nonnegative Lagrange multipliers. For every choice of Lagrange multipliers, we maximize the Lagrangian over θ. These upper bounds are vacuously infinite if the symmetric matrix $P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i$ has a single positive eigenvalue. Even if this matrix has any zero eigenvalues, the corresponding upper bound can still be vacuously infinite if $q(x,u) - \sum_{i=1}^N \lambda_i A_i^\top b_i$ has *any* component in the nullspace of $P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i$. Therefore, we require $P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i$ to be negative semidefinite and $q(x,u) - \sum_{i=1}^N \lambda_i A_i^\top b_i$ to be *orthogonal* to its nullspace. Since this matrix is symmetric, its range is the orthogonal complement of the nullspace. To maximize the Lagrangian over θ, we differentiate with respect to θ and obtain:

$$\nabla_\theta \mathcal{L}(\theta, \lambda; x, u) = 2 \left(P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i \right) \theta + 2 \left(q(x,u) - \sum_{i=1}^N \lambda_i A_i^\top b_i \right). \tag{3.53}$$

Since we require $q(x,u) - \sum_{i=1}^N \lambda_i A_i^\top b$ to be in the range of the matrix $P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i$, we can make use of the left inverse of the matrix (with reciprocals of nonzero eigenvalues), yielding critical θ^*:

$$\theta^* = - \left(P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i \right)^\dagger \left(q(x,u) - \sum_{i=1}^N \lambda_i A_i^\top b_i \right). \tag{3.54}$$

Our minimal upper bound is given by the optimal value of:

$$\begin{aligned}
\inf_{\lambda \in \mathbb{R}^N} \quad & \mathcal{L}(\theta^*, \lambda; x, u) \\
\text{s.t.} \quad & P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i \preceq 0 \\
& q(x,u) - \sum_{i=1}^N \lambda_i A_i^\top b_i \in \operatorname{range}\left(P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i \right) \quad \lambda_1, \ldots, \lambda_N \geq 0,
\end{aligned}$$

where $\preceq$ denotes the positive semidefinite partial order. With a cost function $c : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ that is convex in its second argument, we can write the following robust optimization problem:

$$\begin{aligned}
\inf_{\substack{u \in \mathbb{R}^m \\ \lambda \in \mathbb{R}^N}} \quad & c(x,u) \\
\text{s.t.} \quad & \mathcal{L}(\theta^*, \lambda; x, u) \leq 0 \\
& P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i \preceq 0 \\
& q(x,u) - \sum_{i=1}^N \lambda_i A_i^\top b_i \in \operatorname{range}\left(P(x,u) - \sum_{i=1}^N \lambda_i A_i^\top A_i \right) \\
& \lambda_1, \ldots, \lambda_N \geq 0.
\end{aligned} \tag{3.55}$$

We can make use of the generalized Schur complement (Boyd and Vandenberghe, 2004, Section A.5.5) to simplify the form of this problem, as:

$$\begin{aligned}
\inf_{\substack{u \in \mathbb{R}^m \\ \lambda \in \mathbb{R}^N}} \quad & c(x,u) \\
\text{s.t.} \quad & \begin{bmatrix} P(x,u) & q(x,u) \\ q(x,u)^\top & r(x,u) \end{bmatrix} - \sum_{i=1}^N \lambda_i \begin{bmatrix} A_i^\top A_i & A_i^\top b_i \\ b_i^\top A_i & \|b_i\|_2^2 - d_i(x) \end{bmatrix} \preceq 0 \\
& \lambda_1, \ldots, \lambda_N \geq 0.
\end{aligned} \tag{3.56}$$

We can show that this robust optimization problem is convex using *generalized convexity* of symmetric matrix-valued functions. However, such problems remain difficult to solve computationally. The central difficulty lies in the terms that are purely quadratic in u. To mitigate this issue, we note that the particular cases of quadratic constraints we consider have even more structure. There are functions $P_{\text{aff}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{S}_+^p$, $q_{\text{aff}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$, and $r_{\text{aff}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ that are each

affine in their second arguments, as well as pure functions of the state $b : \mathbb{R}^n \to \mathbb{R}^p$ and $P_0\mathbb{R}^n \to \mathbb{S}^p_{++}$ such that:

$$P(x,u) = U(u)^\top P_0 U(u) + P_{\text{aff}}(x,u), \tag{3.57}$$

$$q(x,u) = U(u)^\top P_0 U(u) b(x) + q_{\text{aff}}(x,u), \tag{3.58}$$

$$r(x,u) = b(x)^\top U(u)^\top P_0 U(u) b(x) + r_{\text{aff}}(x,u), \tag{3.59}$$

for all states $x \in \mathbb{R}^n$ and inputs $u \in \mathbb{R}^m$. This allows us to write:

$$\begin{bmatrix} P(x,u) & q(x,u) \\ q(x,u)^\top & r(x,u) \end{bmatrix} = \begin{bmatrix} P_{\text{aff}}(x,u) & q_{\text{aff}}(x,u) \\ q_{\text{aff}}(x,u)^\top & r_{\text{aff}}(x,u) \end{bmatrix} + \begin{bmatrix} L & Lb(x) \\ b(x)^\top L & b(x)^\top L b(x) \end{bmatrix}, \tag{3.60}$$

where $L = U(u)^\top P_0(x) U(u)$. It is straightforward to see that any matrix $M \succeq L$ causes:

$$\begin{bmatrix} M & Mb(x) \\ b(x)^\top M & b(x)^\top M b(x) \end{bmatrix} \succeq \begin{bmatrix} L & Lb(x) \\ b(x)^\top L & b(x)^\top L b(x) \end{bmatrix}, \tag{3.61}$$

since:

$$\begin{aligned} &\begin{bmatrix} \alpha \\ \beta \end{bmatrix}^\top \begin{bmatrix} M & Mb(x) \\ b(x)^\top M & b(x)^\top M b(x) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &\quad = \alpha^\top M \alpha + \alpha^\top M b(x) \beta + \beta b(x)^\top M \alpha + \beta b(x)^\top M b(x) \beta \\ &\quad = (\alpha + b(x)\beta)^\top M (\alpha + b(x)\beta) \qquad (3.62) \\ &\quad \geq (\alpha + b(x)\beta)^\top L (\alpha + b(x)\beta) \\ &\quad = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^\top \begin{bmatrix} L & Lb(x) \\ b(x)^\top L & b(x)^\top L b(x) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \qquad (3.63) \end{aligned}$$

for any vector $(\alpha, \beta) \in \mathbb{R}^p \times \mathbb{R}$. Therefore, instead of imposing $L = U(u)^\top P_0(x) U(u)$ in our robust optimization problem, we can relax this equality constraint to $L \succeq U(u)^\top P_0(x) U(u)$. Again using the Schur complement, this is equivalent to:

$$\begin{bmatrix} L & U(u)^\top \\ U(u) & P_0(x)^{-1} \end{bmatrix} \succeq 0, \tag{3.64}$$

so we can finally instantiate the convex robust optimization problem:

$$\begin{aligned}
\inf_{\substack{u\in\mathbb{R}^m \\ \lambda\in\mathbb{R}^N \\ L\in\mathbb{S}^p}} \quad & c(x,u) \\
\text{s.t.} \quad & \begin{bmatrix} P_{\text{aff}}(x,u) & q_{\text{aff}}(x,u) \\ q_{\text{aff}}(x,u)^\top & r_{\text{aff}}(x,u) \end{bmatrix} + \begin{bmatrix} L & Lb(x) \\ b(x)^\top L & b(x)^\top Lb(x) \end{bmatrix} \\
& \qquad - \sum_{i=1}^N \lambda_i \begin{bmatrix} A_i^\top A_i & A_i^\top b_i \\ b_i^\top A_i & \|b_i\|_2^2 - d_i(x) \end{bmatrix} \preceq 0 \\
& \begin{bmatrix} L & U(u)^\top \\ U(u) & P_0(x)^{-1} \end{bmatrix} \succeq 0 \\
& \lambda_1, \ldots, \lambda_N \geq 0.
\end{aligned} \tag{3.65}$$

In general, we should not expect strong duality to hold for the maximization over uncertainty. Therefore, this convex robust optimization problem may be more conservative than necessary. However, a feasible solution to this problem *certainly* satisfies the desired constraint under the ground truth uncertainty.

3.6 Outer Bounding Uncertainty Sets

When the number of data points N is very large, the robust optimization problems in Equation (3.49) and Equation (3.65) can still be slow to solve; ultimately, this is due to the complexity of the uncertainty set description. If we can instead find simple *outer bounding* uncertainty sets, incorporating *more* potential uncertainties but with fewer necessary Lagrange multipliers, we can trade off solution speed with conservativeness. Note that if we are too conservative, the resulting optimization problem may be infeasible.

We will briefly study the computation of appropriate outer bounds for uncertainty sets of the form:

$$\mathcal{U}(x) = \{\theta \in \mathbb{R}^p : \|A_i\theta + b_i\|_2^2 \leq d_i(x)^2, i = 1, \ldots, N\}. \tag{3.66}$$

This is our uncertainty set represented as the intersection of N ellipsoids in $\mathbb{R}^p$. When $p > k$ (as is true in our case, where $p = (m+1)n$ and $k = n$), the matrices $A_i \in \mathbb{R}^{k\times p}$ are wide with nontrivial nullspaces. Therefore, each ellipsoid is degenerate, with directions of infinite extent.

However, if the *intersection* of all such degenerate ellipsoids is bounded, then we can compute an outer bounding ellipsoid via semidefinite programming (Calafiore and

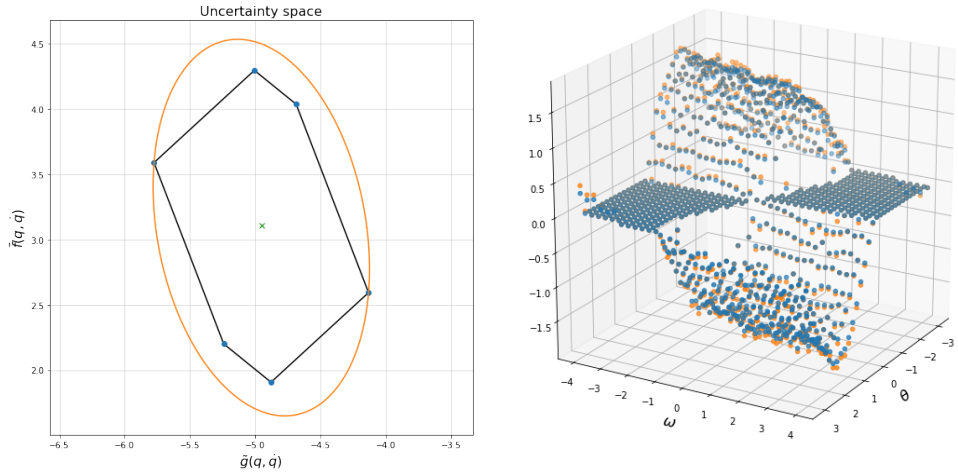


Figure 3.1: Data-driven robust sampled data controller for a pendulum system. Here, $q = \theta$ and $\dot{q} = \omega$. Uncertainty is only modeled in the acceleration. (Left) An example of the uncertainty set and outer approximating ellipsoid. The green cross shows the ground truth uncertainty. (Right) Control inputs evaluated on a grid, with smaller inputs indicating better costs. The blue surface is the control profile of the robust controller using all 3888 data points. The orange surface is the control profile of the robust controller using the outer approximating ellipsoid.

El Ghaoui, 2014, Section 11.4.2.5). The resulting bounding ellipsoid is a suboptimal approximation of the *minimum volume* outer bounding ellipsoid, circumscribing the intersection of ellipsoids. To compute the outer bound, we make use of an extension of the S-proceduce (Calafiore and El Ghaoui, 2014, p. 11.3.3.1), which provides a sufficient condition for a quadratic sublevel set (at level 0) to contain the intersection of N quadratic sublevel sets (all also at level 0).

First, we write $f_1, \ldots, f_N : \mathbb{R}^p \to \mathbb{R}$ as:

$$\begin{aligned} f_i(\theta; x) &= \theta^\top A_i^\top A_i \theta + 2(A_i^\top b_i)^\top \theta + \|b_i\|_2^2 - d_i(x)^2 \\ &= \begin{bmatrix} \theta \\ 1 \end{bmatrix}^\top \begin{bmatrix} A_i^\top A_i & A_i^\top b_i \\ b_i^\top A_i & \|b_i\|_2^2 - d_i(x)^2 \end{bmatrix} \begin{bmatrix} \theta \\ 1 \end{bmatrix}, \end{aligned} \tag{3.67}$$

for all $\theta \in \mathbb{R}^p$ and $i \in \{1, \ldots, N\}$. The ith degenerate ellipsoid is the 0-sublevel set of f_i. Similary, for $F_0 \in \mathbb{S}^p$, $g_0 \in \mathbb{R}^p$, and $h_0 \in \mathbb{R}$ (to be determined), define $f_0 : \mathbb{R}^p \to \mathbb{R}$ as:

$$f_0(\theta) = \begin{bmatrix} \theta \\ 1 \end{bmatrix}^\top \begin{bmatrix} F_0 & g_0 \\ g_0^\top & h_0 \end{bmatrix} \begin{bmatrix} \theta \\ 1 \end{bmatrix}, \tag{3.68}$$

for all $\theta \in \mathbb{R}^p$. A *sufficient* (but not neccesary) condition for the 0-sublevel set of f_0 to contain the intersection of degenerate ellipsoids is the existence of some $\tau \in \mathbb{R}^N_+$ with:

$$\begin{bmatrix} F_0 & g_0 \\ g_0^\top & h_0 \end{bmatrix} \preceq \sum_{i=1}^N \tau_i \begin{bmatrix} A_i^\top A_i & A_i^\top b_i \\ b_i^\top A_i & \|b_i\|_2^2 - d_i(x)^2 \end{bmatrix}. \tag{3.69}$$

If we prescribe that the 0-sublevel set of f_0 is:

$$\{\theta \in \mathbb{R}^p : \|A_0\theta + b_0\|_2 \leq 1\}, \tag{3.70}$$

for some $A_0 \in \mathbb{S}^p_{++}$ and $b_0 \in \mathbb{R}^p$ (again, to be determined), then we obtain:

$$\begin{bmatrix} A_0^2 & A_0 b_0 \\ b_0^\top A_0 & \|b_0\|_2^2 - 1 \end{bmatrix} \preceq \sum_{i=1}^N \tau_i \begin{bmatrix} A_i^\top A_i & A_i^\top b_i \\ b_i^\top A_i & \|b_i\|_2^2 - d_i(x)^2 \end{bmatrix}, \tag{3.71}$$

which we can rearrange as:

$$\begin{bmatrix} A_0^2 & A_0 b_0 \\ b_0^\top A_0 & -1 \end{bmatrix} - \sum_{i=1}^N \tau_i \begin{bmatrix} A_i^\top A_i & A_i^\top b_i \\ b_i^\top A_i & \|b_i\|_2^2 - d_i(x)^2 \end{bmatrix} - \begin{bmatrix} 0 \\ b_0^\top A_0 \end{bmatrix} (-A_0^{-1}) \begin{bmatrix} 0 & A_0 b_0 \end{bmatrix} \preceq 0. \tag{3.72}$$

Again, we can employ the Schur complement to equivalently write:

$$\begin{bmatrix} A_0^2 - \sum_{i=1}^N \tau_i A_i^\top A_i & A_0 b_0 - \sum_{i=1}^N \tau_i A_i^\top b_i & 0 \\ b_0^\top A_0 - \sum_{i=1}^N \tau_i b_i^\top A_i & -1 - \sum_{i=1}^N \tau_i (\|b_i\|_2^2 - d_i(x)^2) & b_0^\top A_0 \\ 0 & A_0 b_0 & -A_0^{-1} \end{bmatrix} \preceq 0, \tag{3.73}$$

which is equivalent provided that $A_0 \succeq 0$.

The volume of an ellipsoid characterized by A_0 and b_0 is $\det A_0^{-1} \cdot \mathrm{Vol}(B_{\mathbb{R}^n})$, where $B_{\mathbb{R}^n}$ is the Euclidean unit ball in $\mathbb{R}^n$. Therefore, for a fixed state $x \in \mathbb{R}^n$, we can approximate the minimum volume ellipsoid containing the uncertainty set $\mathcal{U}(x)$ with the solution to following convex optimization problem:

$$\begin{aligned} \inf_{\substack{A_0 \in \mathbb{S}^p \\ b_0 \in \mathbb{R}^p}} \quad & -\log\det A_0 \\ \text{s.t.} \quad & \begin{bmatrix} A_0^2 - \sum_{i=1}^N \tau_i A_i^\top A_i & A_0 b_0 - \sum_{i=1}^N \tau_i A_i^\top b_i & 0 \\ b_0^\top A_0 - \sum_{i=1}^N \tau_i b_i^\top A_i & -1 - \sum_{i=1}^N \tau_i (\|b_i\|_2^2 - d_i(x)^2) & b_0^\top A_0 \\ 0 & A_0 b_0 & -A_0^{-1} \end{bmatrix} \preceq 0 \\ & A_0 \succeq 0. \end{aligned} \tag{3.74}$$

A smaller volume outer bounding ellipsoid may exist that does not admit a certificate vector $\tau \in \mathbb{R}^N_+$; this optimization problem will not find such an ellipsoid.

Computing the solution to this optimization problem is still similarly difficult to solving the robust optimization problems to begin with. However, by solving such problems *offline* and training parametric models to approximate such outer bounds, the corresponding simplified robust optimization solvers can be accelerated. Moreover, differentiable convex optimization solvers (like those in (Agrawal et al., 2019)) can be employed to update parameters based on the optimal cost attained in the robust optimization problem when the outer bound approximators are employed. Example behavior is shown on a pendulum system in Figure 3.1, where the exact outer approximating ellipsoids are employed.

Chapter 4

COMPACTLY-RESTRICTABLE POLICY OPTIMIZATION PROBLEMS

4.1 Introduction

Policy optimization is a cornerstone in planning, control and reinforcement learning. Classic approaches include value iteration and policy iteration (Dynkin and Yushkevich, 1979; Puterman, 2014; Bertsekas, 2011; Bertsekas, 2019; Sutton and Barto, 2018). From a theoretical standpoint, a key step is establishing when policy optimization is well-posed, i.e., when optimal policies exist, and when they can be found algorithmically. While such results for value and policy iteration are well established for discrete systems with finitely many states and actions (also known as the tabular setting), relatively few foundational results have been established for continuous control.

We study Metric Policy Optimization Problems (MPOPs), which come endowed with metric state and action spaces. Compared to tabular MDPs, several new challenges arise in the continuous setting. Even for deterministic problems, rewards may be unbounded, maxima of functions (over actions) may not exist, and the domains of value functions may be different for different policies. Without addressing these challenges, optimal policies need not exist and value iteration or policy iteration may be impossible.

In order to establish well-posedness of dynamic programming approaches, we define the class of Compactly Restrictable MPOPs (CR-MPOPs). We show that CR-MPOPs arise naturally when imposing forward-invariance constraints on the policy class one optimizes over. As such, CR-MPOPs are well suited for characterizing many systems which rely on forward-invariance for controller design.

Sampled-data design (Nešić, Teel, and Kokotović, 1999) allows us to synthesize policies for continuous-time systems when inputs are passed through a zero-order hold (held constant over fixed frequency time intervals), as is realistic for many physical systems. We leverage recent results (Taylor, Dorobantu, Yue, et al., 2021; Taylor, Dorobantu, Cosner, et al., 2022) to certify the existence of a compact subset of the state space that can be rendered forward-invariant through control. These results are readily applicable to feedback linearizable control affine systems,

allowing us to design CR-MPOPs for a wide array of complex systems.

Many frameworks extend the theory of discrete MDPs to general state and action spaces. Most relevant for our work, (Dynkin and Yushkevich, 1979) develops semi-continuous MDPs with Borel measurable policy classes, (Blackwell, Freedman, and Orkin, 1974) and (Bertsekas and Shreve, 1996) develop results for analyticially and universally measurable policy classes, respectively, and (Feinberg, Kasyanov, and Zadoianchuk, 2012) refines conditions for the well-posedness of value iteration. For a more complete summary, see (Yu and Bertsekas, 2015). A relevant concurrent work is (Feinberg and Kasyanov, 2021), which makes use of inf-compactness of cost function action sections to demonstrate well-posedness for problems with unbounded state and action spaces and unbounded rewards. Our work focuses on developing classes of policy optimization problems that can naturally connect theoretical results from reinforcement learning with nonlinear continuous-time control, and can be directly translated to checking certain properties in control systems that arise naturally in synthesis, such as forward-invariance of sets.

Methods that maintain the forward-invariance of subsets of the state space of interest have been well-studied (Blanchini, 1999; Prajna, 2005; Ames, Xu, et al., 2016). These and related methods have recently found applications in safe reinforcement learning (Berkenkamp, Turchetta, et al., 2017; Rosolia and Borrelli, 2017; Cheng et al., 2019; Choi et al., 2020). Physical systems and robotic platforms have been popular applications for reinforcement learning methods recently, with the majority of methods employing function approximation, discretizations (such as fixed or adaptive gridding or state aggregation), or direct policy search (Kober, Bagnell, and Peters, 2013; Levine et al., 2016; Lillicrap et al., 2015). While policy search methods generally have no guarantees of global optimality, they have received much attention due to the scalability problems of dynamic programming methods (especially after discretization) and convergence problems of approximate dynamic programming (for a more complete discussion of the relative merits of these methods, see (Kober, Bagnell, and Peters, 2013, Section 2.3)). In contrast, our goal is to identify settings that are compatible with value iteration and to use control theoretic tools to guide solution methods so we can expect good performance.

We develop our results in the following phases:

1. We describe a generic property (Assumption 3) that admits the well-posedness of value iteration for MPOPs (Theorem 1).

2. We leverage the forward-invariance of compact sets to design MPOPs and policy classes that satisfy Assumption 3, allowing us to prove well-posedness of a large class of policy optimization problems (Theorem 2). Such problem settings are called CR-MPOPs.

3. We apply our results on CR-MPOPs to analyze control affine systems with time-sampled control inputs.

4. We further apply our results to analyze robotic systems with time-sampled control inputs, which comprise a large class of control affine systems. These results translate the generic requirements for well-posedness to concrete requirements on the robotic system.

5. We finally show that full-state feedback linearizable control affine systems can render a compact subset of the state space forward invariant with continuous controllers (thus satisfying conditions of Theorem 2), demonstrating practically relevant instances of CR-MPOPs.

4.1.1 Notation, Conventions, and Definitions

We consider nonempty metric spaces throughout this chapter and endow each such space with the σ-algebra generated by its topology, called the Borel σ-algebra. In this case, open sets and closed sets are measurable sets and continuous functions are measurable functions. A metric space is separable if it contains a countable dense subset. Finite or countable sets can be regarded as separable metric spaces when equipped with the discrete metric (Kechris, 2012, Section 3.A). For nonempty metric spaces X and Y, the set of measurable functions from X to Y is denoted $\mathcal{L}^0(X;Y)$. The set of bounded measurable functions from X to $\mathbb{R}$ is denoted $\mathcal{L}^0_b(X)$, which is a Banach space when equipped with the norm $\|\cdot\|_{\sup} : \mathcal{L}^0_b(X) \to \mathbb{R}$, defined as:

$$\|f\|_{\sup} = \sup_{x \in X} |f(x)|, \tag{4.1}$$

for all bounded measurable functions $f : X \to \mathbb{R}$. A bounded measurable function $f : X \to \mathbb{R}$ is upper semicontinuous if for any $c \in \mathbb{R}$, the preimage $f^{-1}([c, \infty)) = \{x \in X : f(x) \geq c\}$ is closed. The set of bounded upper semicontinuous functions from X to $\mathbb{R}$ is denoted $C^u_b(X)$, the set of bounded continuous functions from X to $\mathbb{R}$ is denoted $C^0_b(X)$, and note that $C^0_b(X) \subseteq C^u_b(X) \subseteq \mathcal{L}^0_b(X)$.

The set of all subsets of Y is the powerset of Y, denoted $\mathcal{P}(Y)$. A set-valued function $C : X \to \mathcal{P}(Y)$ is called a correspondence. The graph of C is defined as:

$$\text{Graph}(C) = \{(x, y) \in X \times Y : y \in C(x)\}. \tag{4.2}$$

If $C(x) \neq \emptyset$ for all $x \in X$, then a selector of C is a function $f : X \to Y$ satisfying $f(x) \in C(x)$ for all $x \in X$. For any set $B \subseteq Y$, the lower preimage of B under C is defined as:

$$C^{\ell}(B) = \{x \in X : C(x) \cap B \neq \emptyset\}. \tag{4.3}$$

The correspondence C is upper hemicontinuous[1] if the lower preimage of each closed set is a closed set (Aliprantis and Border, 2006, Lemma 17.4). C is measurable if the lower preimage of each closed set is a measurable set, and C is weakly measurable if the lower preimage of each open set is a measurable set. If C is measurable, then it is weakly measurable (Aliprantis and Border, 2006, Lemma 18.2).

The product $X \times Y$ is a metric space, with a topology generated by the basis $\{U \times V : U \subseteq X \text{ and } V \subseteq Y \text{ are open}\}$, and, if X and Y are separable, a σ-algebra generated by the collection $\{A \times B : A \subseteq X \text{ and } B \subseteq Y \text{ are measurable sets}\}$ (Billingsley, 1999, Appendix M).

A nonempty subset $X_0 \subseteq X$ is a metric space when equipped with the restriction of the metric on X to $X_0 \times X_0$. For any open set $U \subseteq X_0$, there is an open set $V \subseteq X$ such that $U = X_0 \cap V$. For any continuous function $f : X \to Y$, the restriction $f|_{X_0} : X_0 \to Y$ is continuous. Similarly, for any measurable set $A \subseteq X_0$, there is a measurable set $B \subseteq X$ such that $A = X_0 \cap B$, and for any measurable function $f : X \to Y$, the restriction $f|_{X_0}$ is measurable.

Consider a nonempty metric space Z and a function $f : X \times Y \to Z$. For $x \in X$ and $y \in Y$, the x-section $f_x : Y \to Z$ and the y-section $f^y : X \to Z$ are defined as:

$$f_x(y') = f(x, y'), \qquad f^y(x') = f(x', y), \tag{4.4}$$

for all $x' \in X$ and $y' \in Y$. If f is continuous, then f has continuous sections, and if f is measurable, then f has measurable sections. If f_x is continuous for each $x \in X$ and f^y is measurable for each $y \in Y$, then f is called a Carathéodory function; if Y is also separable, then f is measurable (Aliprantis and Border, 2006, Lemma 4.51). We also define sections for functions on graphs. Consider a correspondence $C : X \to \mathcal{P}(Y)$ with $C(x) \neq \emptyset$ for all $x \in X$, and a function $f : \text{Graph}(C) \to Z$. For $x \in X$, we define the x-section $f_x : C(x) \to Z$ as above.

[1] Some authors call such correspondences upper semicontinuous; we use hemicontinuous to distinguish correspondences from real-valued functions.

4.2 Finite State and Action Spaces

We first outline deterministic Markov Decision Processes (MDPs) for finite state and action spaces. An MDP with finite state action action spaces is also referred to as *tabular*. Consider a system characterized by a finite set of states $\mathcal{S}$. An agent interacts with this system; at each time step, the agent observes the system state and specifies an action from a finite set of actions $\mathcal{A}$. The actions available to the agent may depend on the state, and the state and specified action determine the next state of the system. Specifically, in each state $s \in \mathcal{S}$, there is a nonempty subset $\mathcal{A}_s \subseteq \mathcal{A}$ of actions available to the agent. We define a correspondence $C : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ with:

$$C(s) = \mathcal{A}_s, \tag{4.5}$$

for all states $s \in \mathcal{S}$. The transition dynamics of the system $f : \text{Graph}(C) \to \mathcal{S}$ tell us how the system state and specified action determine the next system state; when the system is in state $s \in \mathcal{S}$ and the agent specifies action $a \in C(s)$, the system transitions to state $f(s, a) \in \mathcal{S}$.

Given an initial state $s_0 \in \mathcal{S}$, a sequence of actions $\{a_t \in \mathcal{A} : t \in \mathbb{Z}_+\}$ is *admissible* if there is a corresponding sequence of states $\{s_t \in \mathcal{S} : t \in \mathbb{N}\}$ such that:

$$a_t \in C(s_t), \qquad\qquad s_{t+1} = f(s_t, a_t), \tag{4.6}$$

for all $t \in \mathbb{Z}_+$. Such a sequence may admit a corresponding sequence of *rewards*. Specifically, let $r : \text{Graph}(C) \to \mathbb{R}$ be a reward function, giving a reward $r(s, a)$ to the agent when the system is in state $s \in \mathcal{S}$ and the agent takes action $a \in C(s)$. The sequence of rewards corresponding to the above sequences of states and actions is simply $\{r(s_t, a_t) \in \mathbb{R} : t \in \mathbb{Z}_+\}$. Note that while this reward function only depends on a current state and a current action, it can also encode a reward that depends on the next state since the transition dynamics are deterministic.

The total reward accumulated over $T \in \mathbb{N}$ steps is:

$$\sum_{t=0}^{T-1} r(s_t, a_t). \tag{4.7}$$

Since $\mathcal{S}$ and $\mathcal{A}$ are finite, so is $\text{Graph}(C) \subseteq \mathcal{S} \times \mathcal{A}$, as is the image $r(\text{Graph}(C)) \subset \mathbb{R}$. This means the reward sequence is bounded. However, the sequence need not converge to 0, which is necessary for convergence of the accumulated reward as $T \to \infty$. Therefore, we make use of a *discount factor* $\gamma \in [0, 1)$, and consider the total *discounted* reward accumulated over an infinite horizon:

$$\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t). \tag{4.8}$$

At a high level, we aim to select actions that maximize this quantity. The tuple $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ is called a (deterministic) *Markov decision process* (MDP).

One way we can simplify the maximization total accumulated discounted reward is by introducing *state-feedback policies*. Specifically, we consider memoryless and time-invariant policies $\pi : \mathcal{S} \to \mathcal{A}$ that satisfy $\pi(s) \in C(s)$ for all states $s \in \mathcal{S}$. Such functions are *selectors* of C. When the agent follows the policy π, whenever the system is in state $s \in \mathcal{S}$, the agent specifies action $\pi(s) \in \mathcal{A}$. The *value function* under a policy π specifies the total discounted reward accumulated by the agent from any initial state of the system. Specifically, define the closed-loop transition map $f_\pi : \mathcal{S} \to \mathcal{S}$ and single-step reward function $r_\pi : \mathcal{S} \to \mathbb{R}$ as:

$$f_\pi(s) = f(s, \pi(s)), \qquad r_\pi(s) = r(s, \pi(s)), \tag{4.9}$$

for all states $s \in \mathcal{S}$. For any $t \in \mathbb{Z}_+$, let $f_\pi^t : \mathcal{S} \to \mathcal{S}$ denote the t-iterated composition of f_π. The value function $V_\pi : \mathcal{S} \to \mathbb{R}$ is then defined as:

$$V_\pi(s) = \sum_{t=0}^{\infty} \gamma^t r_\pi(f_\pi^t(s)), \tag{4.10}$$

for all states $s \in \mathcal{S}$. Importantly, V_π satisfies the implicit equation:

$$V_\pi(s) = r_\pi(s) + \gamma V_\pi(f_\pi(s)) = r(s, \pi(s)) + \gamma V_\pi(f(s, \pi(s))), \tag{4.11}$$

for all states $s \in \mathcal{S}$.

The set of all policies Π is partially ordered by value functions. Specifically, if $\pi, \pi' \in \Pi$ satisfy:

$$V_\pi(s) \geq V_{\pi'}(s), \tag{4.12}$$

for all states $s \in \mathcal{S}$, then $\pi \geq \pi'$. Since $\mathcal{S}$ and $\mathcal{A}$ are finite, there are finitely many policies, with no more than $\prod_{s \in \mathcal{S}} |C(s)| \leq |\mathcal{S}| \cdot |\mathcal{A}|$ policies total, so we can define the *optimal value function* $V^* : \mathcal{S} \to \mathbb{R}$ as:

$$V^*(s) = \max_{\pi \in \Pi} V_\pi(s), \tag{4.13}$$

for all states $s \in \mathcal{S}$. An *optimal policy* $\pi^* \in \Pi$ satisfies:

$$V_{\pi^*}(s) = V^*(s), \tag{4.14}$$

for all states $s \in \mathcal{S}$. From the definition of V^*, this means $\pi^* \geq \pi$ for all policies $\pi \in \Pi$.

Two classical methods for finding optimal policies are *value iteration* and *policy iteration*. Since its generalization beyond finite state and action spaces is more straightforward, we focus on value iteration. Intuitively, value iteration produces a sequence of approximations of the optimal value function, converging to the optimal value function itself. The set of all functions from $\mathcal{S}$ to $\mathbb{R}$ is a real vector space of dimension $|\mathcal{S}|$. The *optimal Bellman operator* is defined on this space as:

$$[\mathcal{T}(V)](s) = \max_{a \in C(s)} \left(r(s,a) + \gamma V(f(s,a)) \right), \tag{4.15}$$

for all state $s \in \mathcal{S}$, for all $V : \mathcal{S} \to \mathbb{R}$. Value iteration simply starts from any initial approximation $V_0 : \mathcal{S} \to \mathbb{R}$ and produces the sequence $\{V_n : n \in \mathbb{N}\}$ via:

$$V_{n+1} = \mathcal{T}(V_n), \tag{4.16}$$

for each $n \in \mathbb{Z}_+$. To show that value iteration converges to the optimal value function, it is sufficient to show that:

1. For an appropriate choice of norm, the space of functions from $\mathcal{S}$ to $\mathbb{R}$ is a Banach space and $\mathcal{T}$ is a γ-contraction, so value iteration always converges to a unique fixed point V_∞, and
2. $V_\infty = V^*$.

We will illustrate this in detail for infinite state and action spaces.

4.3 Metric Policy Optimization Problems

In the general case, we will still characterize a deterministic MDP by a tuple $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, action-admissibility correspondence $C : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ satisfying $C(s) \neq \emptyset$ for each state $s \in \mathcal{S}$, transition map $f : \mathrm{Graph}(C) \to \mathcal{S}$, reward function $r : \mathrm{Graph}(C) \to \mathbb{R}$, and discount factor $\gamma \in [0, 1)$. Note while r may be unbounded, r cannot assume the values $\pm\infty$. We refer to an MDP as a metric MDP if $\mathcal{S}$ and $\mathcal{A}$ are nonempty separable metric spaces and f and r are measurable functions. We will focus solely on metric MDPs, and often refer to them simply as MDPs for brevity.

We will limit our consideration to deterministic, Markovian (memoryless), and stationary (time-invariant) policies. In this case, a policy $\pi : \mathcal{S} \to \mathcal{A}$ is a selector of C; that is, $\pi(s) \in C(s)$ for all states $s \in \mathcal{S}$. The corresponding closed-loop

transition map $f_\pi : \mathcal{S} \to \mathcal{S}$ and single-step reward function $r_\pi : \mathcal{S} \to \mathbb{R}$ are defined as before. Define the subset $\mathcal{S}_\pi \subseteq \mathcal{S}$ as:

$$\mathcal{S}_\pi = \left\{ s \in \mathcal{S} : \sum_{t=0}^{\infty} \gamma^t r_\pi(f_\pi^t(s)) \text{ converges absolutely} \right\}. \tag{4.17}$$

If r_π is bounded, then $\mathcal{S}_\pi = \mathcal{S}$. If $\gamma > 0$, then $f_\pi(\mathcal{S}_\pi) \subseteq \mathcal{S}_\pi$, as a convergent series still converges after the first term is removed. If $\mathcal{S}_\pi \neq \emptyset$, then the corresponding value function $V_\pi : \mathcal{S}_\pi \to \mathbb{R}$ is defined explicitly and implicitly, for all states $s \in \mathcal{S}_\pi$, as:

$$V_\pi(s) = \sum_{t=0}^{\infty} \gamma^t r_\pi(f_\pi^t(s)) = r_\pi(s) + \gamma V_\pi(f_\pi(s)). \tag{4.18}$$

A set of policies Π admits a partial order if $\mathcal{S}_\pi = \mathcal{S}$ for all policies $\pi \in \Pi$. In this case, for policies $\pi, \pi' \in \Pi$, if $V_\pi(s) \geq V_{\pi'}(s)$ for all states $s \in \mathcal{S}$, then $\pi \geq \pi'$.

Definition 3 (Metric Policy Optimization Problem)**.** We refer to the pair of an MDP $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ and a policy class Π admitting a partial order as a *metric policy optimization problem* (MPOP). We call an MPOP well-posed if there is an optimal policy $\pi^* \in \Pi$ with $\pi^* \geq \pi$ for all policies $\pi \in \Pi$.

Our goal in this section is to analyze well-posedness of policy optimization for MDPs. We first describe sufficient conditions for well-posedness of value iteration (Theorem 1). We then establish conditions under which ill-posed MPOPs can be restricted to well-posed problems by only considering policies that render the same subset of the state space forward-invariant (Theorem 2), resulting in CR-MPOPs. We will show how to apply our results on value iteration to control affine systems in Section 4.4.

Throughout this section, we refer to and modify the following example to ground our presentation:

Example 3. Consider the state space $\mathcal{S} = \mathbb{R}$, action space $\mathcal{A} = \mathbb{R}$, and the constant action-admissibility correspondence $C = \mathcal{A}$. Additionally, consider the transition function $f : \text{Graph}(C) \to \mathcal{S}$ and reward function $r : \text{Graph}(C) \to \mathbb{R}$ defined as:

$$f(s, a) = s + \tanh(a), \qquad r(s, a) = -s^2 - (\tanh(a))^2, \tag{4.19}$$

for all state-action pairs $(s, a) \in \text{Graph}(C)$, and a discount factor $\gamma \in [0, 1)$. Note that $|f(s, a)| \leq |s| + |\tanh(a)| \leq |s| + 1$ for all state-action pairs $(s, a) \in \text{Graph}(C)$.

Consider any policy $\pi : \mathcal{S} \to \mathcal{A}$. By induction, we have $|f_\pi^t(s)| \leq |s| + t$ for all states $s \in \mathcal{S}$ and $t \in \mathbb{Z}_+$. For any state $s \in \mathcal{S}$ and $T \in \mathbb{Z}_+$, we have:

$$\begin{aligned}\sum_{t=0}^{T} \gamma^t r_\pi(f_\pi^t(s)) &\geq -\sum_{t=0}^{\infty} \gamma^t ((f_\pi^t(s))^2 + (\tanh(\pi(f_\pi^t(s))))^2) \\ &\geq -\sum_{t=0}^{\infty} \gamma^t (s^2 + 2|s|t + t^2 + 1) \\ &= -\frac{s^2+1}{1-\gamma} - \frac{2\gamma|s|}{(1-\gamma)^2} - \frac{\gamma(\gamma+1)}{(1-\gamma)^3}. \end{aligned} \tag{4.20}$$

The sequence of partial sums as $T \to \infty$ is monotone and bounded below, so $V_\pi(s)$ is well-defined. As s was arbitrary, we have $\mathcal{S}_\pi = \mathcal{S}$.

4.3.1 Value Iteration

Consider a policy class Π with $\mathcal{S}_\pi = \mathcal{S}$ for each policy $\pi \in \Pi$ and $\sup_{\pi \in \Pi} V_\pi(s) < \infty$ for each state $s \in \mathcal{S}$. Accordingly, we define the optimal (with respect to Π) value function $V^* : \mathcal{S} \to \mathbb{R}$ as:

$$V^*(s) = \sup_{\pi \in \Pi} V_\pi(s), \tag{4.21}$$

for all states $s \in \mathcal{S}$. If a policy $\pi^* \in \Pi$ satisfies $V_{\pi^*} = V^*$, then π^* is optimal and the policy optimization problem is well-posed.

As with value iteration for finite state and action spaces, given an initial guess $V_0 : \mathcal{S} \to \mathbb{R}$, we seek a sequence of real-valued functions $\{V_n : n \in \mathbb{N}\}$ satisfying:

$$V_{n+1}(s) = \sup_{a \in C(s)} \{r(s,a) + \gamma V_n(f(s,a))\}, \tag{4.22}$$

for all states $s \in \mathcal{S}$ and $n \in \mathbb{Z}_+$.

We now describe conditions under which MPOPs are well-posed and can be solved with value iteration (Theorem 1). When assumptions are strengthened by requiring $\mathcal{S}$ and $\mathcal{A}$ to be Polish spaces, we can make use of (Feinberg, Kasyanov, and Zadoianchuk, 2012, Theorem 4.1) (the setting of this theorem is called a semicontinuous model in (Dynkin and Yushkevich, 1979, Section 6.7)).

Assumption 3. The action admissibility correspondence C has compact values and is upper hemicontinuous, the transition function f is continuous, the reward function r is upper semicontinuous and bounded, and the policy class Π is the set of all measurable selectors of C. Moreover, for each state-action pair $(s,a) \in \mathrm{Graph}(C)$, there is a corresponding policy $\pi_{s,a} \in \Pi$ satisfying $\pi_{s,a}(s) = a$.

Remark 3. If the projection of $\text{Graph}(C)$ onto the action space is compact, then the assumption that for any state-action pair $(s, a) \in \text{Graph}(C)$, there exists a policy $\pi_{s,a} \in \Pi$ with $\pi_{s,a}(s) = a$ can be removed. See Lemma 2.

Remark 4 (Tabular MDP)**.** If $\mathcal{S}$ and $\mathcal{A}$ are finite sets equipped with discrete metrics and Π is the set of all selectors of C, then these assumptions are immediately met. This follows since the discrete metric renders all sets are open, closed, and compact, and all functions defined on such sets are continuous (and bounded if real-valued).

Example 4. Consider the MDP $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ from Example 3. While C is upper hemicontinuous, it is not compact-valued. While f and r are continuous, r is unbounded. Consider the constant correspondence $C' = [-1, 1]$, and the reward $r' : \text{Graph}(C') \to \mathbb{R}$:

$$r'(s) = e^{-s^2} - (\tanh(a))^2, \tag{4.23}$$

for all state-action pairs $(s, a) \in \text{Graph}(C')$. Note that $\text{Graph}(C') = \mathcal{S} \times [-1, 1]$, so r' is bounded. Since f is continuous, its restriction to $\text{Graph}(C')$ is as well. With Π the set of all measurable selectors of C', note that for every pair $(s, a) \in \text{Graph}(C')$, the constant policy $\pi_{s,a} = a$ is a measurable selector of C', so $\pi_{s,a} \in \Pi$. Thus, the MPOP defined by the MDP $(\mathcal{S}, \mathcal{A}, C', f|_{\text{Graph}(C')}, r', \gamma)$ and policy class Π satisfies Assumption 3.

The remainder of this section is dedicated to demonstrating well-posedness under Assumption 3, as stated in the following theorem.

Theorem 1. *Consider an MPOP characterized by an MDP $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ and a policy class Π that satisfies Assumption 3. There is an optimal policy $\pi^* \in \Pi$ satisfying $V_{\pi^*} = V^*$, and V^* is the limit of value iteration when the initial guess is bounded and upper semicontinuous.*

4.3.1.1 Measurability and Boundedness

To establish the measurability of the value function under a policy $\pi \in \Pi$, we begin by establishing measurability of the corresponding closed-loop transition map f_π and single-step reward function r_π. To this end, we define $z : \mathcal{S} \to \mathcal{S} \times \mathcal{A}$ as:

$$z(s) = (s, \pi(s)), \tag{4.24}$$

for all states $s \in \mathcal{S}$. Consider measurable sets $A \subseteq \mathcal{S}$ and $B \subseteq \mathcal{A}$; the preimage of the product $A \times B$ under z is $z^{-1}(A \times B) = \{s \in \mathcal{S} : s \in A, \pi(s) \in B\} =$

$A \cap \pi^{-1}(B)$, which is a measurable set since π is a measurable function. Since measurable products generate the σ-algebra on $\mathcal{S} \times \mathcal{A}$, z is measurable, implying the compositions $f_\pi = f \circ z$ and $r_\pi = r \circ z$ are measurable functions. It follows that for any $t \in \mathbb{Z}_+$, the t-iterated composition f_π^t is a measurable function, as is the composition $r_\pi \circ f_\pi^t$. Note that $\mathcal{S}_\pi = \mathcal{S}$ since r_π is bounded (by the same bound on r); therefore, the corresponding value function $V_\pi : \mathcal{S} \to \mathbb{R}$ is well-defined on all of $\mathcal{S}$. Moreover, V_π is measurable since it is the pointwise limit of a sequence of partial sums of measurable functions. Finally, V_π is bounded by $(1-\gamma)^{-1}$ times the bound on r, so $V_\pi \in \mathcal{L}_b^0(\mathcal{S})$.

4.3.2 Bellman Operators

The Bellman operator $\mathcal{T}_\pi : \mathcal{L}_b^0(\mathcal{S}) \to \mathcal{L}_b^0(\mathcal{S})$ under policy π generalizes the implicit structure of the value function V_π, satisfying:

$$[\mathcal{T}_\pi(g)](s) = r_\pi(s) + \gamma g(f_\pi(s)), \tag{4.25}$$

for all states $s \in \mathcal{S}$, for all bounded measurable functions $g : \mathcal{S} \to \mathbb{R}$. Indeed, for any bounded measurable function $g : \mathcal{S} \to \mathbb{R}$, the function $\mathcal{T}_\pi(g) = r_\pi + \gamma g \circ f_\pi$ is bounded and measurable. The Bellman operator $\mathcal{T}_\pi$ is a γ-contraction (Bertsekas, 2011, Section 1.4.1), so by the contraction mapping principle (Aliprantis and Border, 2006, Theorem 3.48), V_π is the only bounded measurable function with $\mathcal{T}_\pi(V_\pi) = V_\pi$.

We use Bellman operators to characterize the optimal Bellman operator. Since each value function corresponding to a policy in Π is bounded by $(1-\gamma)^{-1}$ times the bound on r, the optimal value function V^* is well-defined and bounded by the same bound. The optimal Bellman operator $\mathcal{T} : C_b^u(\mathcal{S}) \to C_b^u(\mathcal{S})$ satisfies:

$$[\mathcal{T}(g)](s) = \sup_{\pi \in \Pi} [\mathcal{T}_\pi(g)](s) = \sup_{a \in C(s)} (r(s,a) + \gamma g(f(s,a))), \tag{4.26}$$

for all states $s \in \mathcal{S}$, for all bounded upper semicontinuous functions $g : \mathcal{S} \to \mathbb{R}$. We cannot define $\mathcal{T}$ on all of $\mathcal{L}_b^0(\mathcal{S})$, and ensuring that $\mathcal{T}$ is well-defined (the suprema are finite and equal) and has codomain $C_b^u(\mathcal{S})$ is the subject of the following lemma.

Lemma 1. *Suppose an MDP $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ and policy class Π satisfy Assumption 3. Let $\varphi : \mathrm{Graph}(C) \to \mathbb{R}$ be upper semicontinuous and bounded. The function $g_\varphi : \mathcal{S} \to \mathbb{R}$ specified as:*

$$g_\varphi(s) = \max_{a \in C(s)} \varphi(s,a), \tag{4.27}$$

for all states $s \in \mathcal{S}$ is well-defined, upper semicontinuous, and bounded. Moreover, there is a policy $\pi_\varphi \in \Pi$ satisfying:

$$\varphi(s, \pi_\varphi(s)) = g_\varphi(s), \tag{4.28}$$

for all states $s \in \mathcal{S}$.

This lemma resembles Theorem 2.5.A of (Dynkin and Yushkevich, 1979), in which the function being maximized is only a function of actions. We will follow the proof carefully, making the necessary modifications where they are needed. Equipped with Lemma 1, we will then only need to show that the suprema in (4.26) are equal.

Proof. The function g_φ is well-defined and upper semicontinuous by (Aliprantis and Border, 2006, Lemma 17.30) and is bounded by the bound on φ.

Define the correspondence $C_\varphi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ as:

$$C_\varphi(s) = \arg\max_{a \in C(s)} \varphi(s, a) \subseteq C(s), \tag{4.29}$$

for all states $s \in \mathcal{S}$. For any state $s \in \mathcal{S}$, since g_φ is well-defined, we have $C_\varphi(s) \neq \emptyset$. The singleton set $\{s\}$ and the set of admissible actions $C(s)$ are both compact, implying the product $\{s\} \times C(s)$ is compact. Therefore, the set of maximizing state-action pairs $\arg\max_{(s',a') \in \{s\} \times C(s)} \varphi(s', a')$ is nonempty and compact (Aliprantis and Border, 2006, Theorem 2.43). Since:

$$C_\varphi(s) = p_{\mathcal{A}} \left(\arg\max_{(s',a') \in \{s\} \times C(s)} \varphi(s', a') \right), \tag{4.30}$$

where $p_{\mathcal{A}} : \mathcal{S} \times \mathcal{A} \to \mathcal{A}$ is the continuous function projecting $\mathcal{S} \times \mathcal{A}$ onto $\mathcal{A}$, the set of maximizing actions $C_\varphi(s)$ is compact.

The remainder of the proof is a modification of the proof of (Dynkin and Yushkevich, 1979, Theorem 2.5.A). Since C is upper hemicontinuous, it is measurable. Therefore, C is weakly measurable (Aliprantis and Border, 2006, Lemma 18.2). Let $\rho_{\mathcal{A}} : \mathcal{A} \times \mathcal{A} \to \mathbb{R}_+$ denote the metric on $\mathcal{A}$, and define the distance function $\rho_C : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$ as:

$$\rho_C(s, a) = \inf_{a' \in C(s)} \rho_{\mathcal{A}}(a, a'), \tag{4.31}$$

for all states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$. Since C is weakly measurable, ρ_C is a Carathéodory function (Aliprantis and Border, 2006, Theorem 18.5); that is, for

every state $s \in \mathcal{S}$, the s-section $(\rho_C)_s$ is continuous, and for every action $a \in \mathcal{A}$, the a-section $(\rho_C)^a$ is a measurable function.

Since C is upper hemicontinuous and compact valued and $\mathcal{A}$ is a Hausdorff space (as it is a metric space), $\text{Graph}(C)$ is a closed subset of $\mathcal{S} \times \mathcal{A}$ (Aliprantis and Border, 2006, Theorem 17.10). As in (K, n.d., (https://mathoverflow.net/users/117299/m-reza-k)) (and in a similar manner to the proof of (Askoura, 2008, Corollary 2)), let $M \in \mathbb{R}_+$ denote the bound on φ, and define the extension $\bar{\varphi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as:

$$\bar{\varphi}(s,a) = \begin{cases} \varphi(s,a) & (s,a) \in \text{Graph}(C), \\ -M & (s,a) \in (\mathcal{S} \times \mathcal{A}) \setminus \text{Graph}(C), \end{cases} \tag{4.32}$$

for all states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$. To see that $\bar{\varphi}$ is upper semicontinuous, consider any real number $c \in \mathbb{R}$. If $c > -M$, then $\bar{\varphi}^{-1}([c,\infty)) = \varphi^{-1}([c,\infty))$. Since φ is upper semicontinuous, there is a closed set $F_c \subseteq \mathcal{S} \times \mathcal{A}$ satisfying $\varphi^{-1}([c,\infty)) = \text{Graph}(C) \cap F_c$, making $\bar{\varphi}^{-1}([c,\infty))$ a closed subset of $\mathcal{S} \times \mathcal{A}$. Otherwise, if $c \leq -M$, then $\bar{\varphi}^{-1}([c,\infty)) = \mathcal{S} \times \mathcal{A}$, which is a closed set.

Consider a monotonically nonincreasing sequence of bounded real-valued continuous functions $\{h_n \in C^0(\mathcal{S} \times \mathcal{A}) : n \in \mathbb{N}\}$ converging pointwise to $\bar{\varphi}$; such a sequence is guaranteed to exist (Dynkin and Yushkevich, 1979, Section 2.4). For each $n \in \mathbb{N}$, define the correspondence $C_n : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ as:

$$C_n(s) = \{a \in \mathcal{A} : \rho_C(s,a) < 1/n \text{ and } g_\varphi(s) < h_n(s,a) + 1/n\}, \tag{4.33}$$

for all states $s \in \mathcal{S}$. Now, for any state $s \in \mathcal{S}$, recall that the s-section $(\rho_C)_s$ is continuous. For any $n \in \mathbb{N}$, since h_n is continuous, the corresponding s-section $(h_n)_s$ is continuous. Thus, $C_n(s)$ is an open subset of $\mathcal{A}$, and can be represented as:

$$\begin{aligned} &C_n(s) \\ &\quad = \{a \in \mathcal{A} : (\rho_C)_s(a) \in (-\infty, 1/n)\} \cap \left\{a \in \mathcal{A} : (h_n)_s(a) \in \left(g_\varphi(s) - 1/n, \infty\right)\right\} \\ &\quad = \left((\rho_C)_s\right)^{-1}((-\infty, 1/n)) \cap \left((h_n)_s\right)^{-1}\left(\left(g_\varphi(s) - 1/n, \infty\right)\right). \end{aligned} \tag{4.34}$$

Since the sequence of functions $\{h_n : n \in \mathbb{N}\}$ is monotonically nonincreasing, these correspondences satisfy $C_{n+1}(s) \subseteq C_n(s)$ for all states $s \in \mathcal{S}$ and $n \in \mathbb{N}$. For any state $s \in \mathcal{S}$ and any action $a \in C_\varphi(s)$, we have $\rho_C(s,a) = 0 < 1/n$ and $g_\varphi(s) = \varphi(s,a) \leq h_n(s,a) < h_n(s,a) + 1/n$ for all $n \in \mathbb{N}$. This implies $C_\varphi(s) \subseteq \bigcap_{n=1}^{\infty} C_n(s)$ for all states $s \in \mathcal{S}$.

Next, note that g_φ is a measurable function as it is bounded and upper semicontinuous. For any action $a \in \mathcal{A}$, recall that the a-section $(\rho_C)^a$ is a measurable function.

Additionally, for any $n \in \mathbb{N}$, since h_n is continuous, the corresponding a-section $(h_n)^a$ is a measurable function (as it is continuous). This means the following set is measurable:

$$\begin{aligned}
&\{s \in \mathcal{S} : a \in C_n(s)\} \\
&\quad = \{s \in \mathcal{S} : (\rho_C)^a(s) \in (-\infty, 1/n)\} \cap \left\{s \in \mathcal{S} : g_\varphi(s) - ((h_n)^a)(s) \in (-\infty, 1/n)\right\} \\
&\quad = ((\rho_C)^a)^{-1}\left((-\infty, 1/n)\right) \cap (g_\varphi - (h_n)^a)^{-1}\left((-\infty, 1/n)\right). \qquad (4.35)
\end{aligned}$$

Lastly, for any state $s \in \mathcal{S}$, consider a sequence of actions $\{a_n \in \mathcal{A} : n \in \mathbb{N}\}$ satisfying $a_n \in C_n(s)$ for all $n \in \mathbb{N}$. For each $n \in \mathbb{N}$, since $\rho_C(s, a_n) < 1/n$, there is an action $a'_n \in C(s)$ satisfying $\rho_{\mathcal{A}}(a'_n, a_n) < 2/n$. Since $C(s)$ is compact, the sequence of actions $\{a'_n \in C(s) : n \in \mathbb{N}\}$ has a limit point $a^* \in C(s)$. That is, there is a monotonically increasing sequence $\{n_k \in \mathbb{N} : k \in \mathbb{N}\}$ such that the subsequence $\{a'_{n_k} \in C(s) : k \in \mathbb{N}\}$ converges to a^*. Since $\rho_{\mathcal{A}}(a'_{n_k}, a_{n_k}) < 2/n_k \leq 2/k$ for all $k \in \mathbb{N}$, the subsequence $\{a_{n_k} \in \mathcal{A} : k \in \mathbb{N}\}$ also converges to a^*. For any $k, l \in \mathbb{N}$ with $k > l$, we have:

$$h_{n_l}(s, a_{n_k}) \geq h_{n_k}(s, a_{n_k}) > g_\varphi(s) - 1/n_k \geq g_\varphi(s) - 1/k, \qquad (4.36)$$

and since h_{n_l} is continuous, this means $\lim_{k\to\infty} h_{n_l}(s, a_{n_k}) = h_{n_l}(s, a^*) \geq g_\varphi(s)$. Since the sequence of functions $\{h_n : n \in \mathbb{N}\}$ converges pointwise to $\bar{\varphi}$ from above, we have $\lim_{l\to\infty} h_{n_l}(s, a^*) = \bar{\varphi}(s, a^*) = \varphi(s, a^*) \geq g_\varphi(s)$, implying $a^* \in C_\varphi(s)$. That is, the sequence of actions $\{a_n : n \in \mathbb{N}\}$ has a limit point in $C_\varphi(s)$. By the measurability criterion in (Dynkin and Yushkevich, 1979, Section 2.6), the correspondence C_φ satisfies the assumptions of (Dynkin and Yushkevich, 1979, Theorem 2.6.B); therefore, there exists a measurable selection $\pi_\varphi : \mathcal{S} \to \mathcal{A}$ of C_φ, implying $\pi_\varphi \in \Pi$ and $\varphi(s, \pi_\varphi(s)) = g_\varphi(s)$ for all states $s \in \mathcal{S}$. □

To show that the suprema in (4.26) are equal, fix a bounded measurable function $g : \mathcal{S} \to \mathbb{R}$ and a state $s \in \mathcal{S}$. For any policy $\pi \in \Pi$, we have:

$$\begin{aligned}
[\mathcal{T}_\pi(g)](s) &= r(s, \pi(s)) + \gamma g(f(s, \pi(s))) \\
&\leq \sup_{a \in C(s)} \left(r(s, a) + \gamma g(f(s, a))\right). \qquad (4.37)
\end{aligned}$$

Conversely, for any action $a \in C(s)$, we have:

$$\begin{aligned}
r(s, a) + \gamma g(f(s, a)) &= r_{\pi_{s,a}}(s) + \gamma g(f_{\pi_{s,a}}(s)) = [\mathcal{T}_{\pi_{s,a}}(g)](s) \\
&\leq \sup_{\pi \in \Pi} [\mathcal{T}_\pi(g)](s), \qquad (4.38)
\end{aligned}$$

where $\pi_{s,a} \in \Pi$ satisfies $\pi_{s,a}(s) = a$. Considering all policies in (4.37) and all actions in (4.38) establishes the equality of the suprema. Since r and g are bounded and upper semicontinuous and f is continuous, the function $r+\gamma g \circ f$ is bounded and upper semicontinuous, so applying Lemma 1 to this function allows us to conclude that $\mathcal{T}(g) \in C_b^u(\mathcal{S})$.

To show that the MPOP is well-posed, we will show that the optimal value function V^* is the fixed point of $\mathcal{T}$; that is $\mathcal{T}(V^*) = V^*$. However, though $\mathcal{T}$ is a γ-contraction (Bertsekas, 2011, Section 1.4.1), we cannot immediately apply the contraction mapping principle (Aliprantis and Border, 2006, Theorem 3.48) to show the existence of a unique fixed point of $\mathcal{T}$ since $C_b^u(\mathcal{S})$ is not a closed subset of the Banach space $\mathcal{L}_b^0(\mathcal{S})$.

Instead, we consider the sequence of bounded upper semicontinuous function $\{V_n \in C_b^u(\mathcal{S}) : n \in \mathbb{N}\}$ generated by value iteration when the initial guess is chosen as the constant function $V_0 = M/(1-\gamma)$, where $M \in \mathbb{R}_+$ denotes the bound on r. To verify that the functions in this sequence are bounded and upper semicontinuous, note that the initial guess is bounded and upper semicontinuous, and by induction, the value iteration update can be written as:

$$V_{n+1} = \mathcal{T}(V_n), \tag{4.39}$$

for all $n \in \mathbb{Z}_+$. The sequence of functions is also monotonically nonincreasing. To see this, note that:

$$\begin{aligned} V_1(s) = [\mathcal{T}(V_0)](s) &= \sup_{a \in C(s)} \left(r(s,a) + \gamma V_0(f(s,a))\right) \leq M + \gamma M/(1-\gamma) \\ &= M/(1-\gamma) \leq V_0(s), \end{aligned} \tag{4.40}$$

for all states $s \in \mathcal{S}$. Since $\mathcal{T}$ preserves ordering (Bertsekas, 2011), if for some $n \in \mathbb{Z}_+$ we have $V_{n+1}(s) \leq V_n(s)$ for all states $s \in \mathcal{S}$, then $[\mathcal{T}(V_{n+1})](s) \leq [\mathcal{T}(V_n)](s)$ for all states $s \in \mathcal{S}$. Equivalently, $V_{n+2}(s) \leq V_{n+1}(s)$ for all states $s \in \mathcal{S}$. By induction, it follows that the sequence of functions is monotonically nonincreasing. Finally, the sequence of functions is bounded by $M/(1-\gamma)$. By definition, V_0 satisfies this bound, and since the sequence of functions is nonincreasing, it is upper bounded by $M/(1-\gamma)$. To show the lower bound of $-M/(1-\gamma)$, note that:

$$\begin{aligned} V_1(s) &= \sup_{a \in C(s)} \left(r(s,a) + \gamma V_0(f(s,a))\right) \geq \inf_{a \in C(s)} \left(r(s,a) + \gamma V_0(f(s,a))\right) \\ &\geq -M - \frac{\gamma}{1-\gamma} M = -\frac{M}{1-\gamma}, \end{aligned} \tag{4.41}$$

for all states $s \in \mathcal{S}$. Similarly, if for some $n \in \mathbb{Z}_+$ we have $V_n(s) \geq -M/(1-\gamma)$ for all states $s \in \mathcal{S}$, then:

$$V_{n+1}(s) \geq \inf_{a \in C(s)} \left(r(s,a) + \gamma V_n(f(s,a))\right) \geq -M - \frac{\gamma}{1-\gamma} M = -\frac{M}{1-\gamma}, \tag{4.42}$$

for all states $s \in \mathcal{S}$. Therefore, the sequence of functions is bounded below by induction.

The pointwise limit of a monotonically nonincreasing sequence of bounded upper semicontinuous functions is upper semicontinuous (Dynkin and Yushkevich, 1979, Section 2.4), and since the sequence is bounded, so is the limit. Denote the limit by $g^* \in C_b^u(\mathcal{S})$. To show that g^* is a fixed point of $\mathcal{T}$, note that:

$$\begin{aligned} \|\mathcal{T}(g^*) - g^*\|_{\sup} &\leq \|\mathcal{T}(g^*) - V_{n+1}\|_{\sup} + \|V_{n+1} - g^*\|_{\sup} \\ &= \|\mathcal{T}(g^*) - \mathcal{T}(V_n)\|_{\sup} + \|V_{n+1} - g^*\|_{\sup} \\ &\leq \gamma \|g^* - V_n\|_{\sup} + \|V_{n+1} - g^*\|_{\sup}, \end{aligned} \tag{4.43}$$

for all $n \in \mathbb{Z}_+$. By choosing n sufficiently large, we make $\|\mathcal{T}(g^*) - g^*\|_{\sup}$ arbitrarily small, implying $\|\mathcal{T}(g^*) - g^*\|_{\sup} = 0$, or $\mathcal{T}(g^*) = g^*$. Now, consider any other sequence of bounded upper semicontinuous functions $\{W_n \in C_b^u(\mathcal{S}) : n \in \mathbb{N}\}$ generated by value iteration from a bounded and upper semicontinuous initial guess $W_0 \in C_b^u(\mathcal{S})$. Since $\mathcal{T}$ is a γ-contraction, we have:

$$\|W_{n+1} - g^*\|_{\sup} = \|\mathcal{T}(W_n) - \mathcal{T}(g^*)\|_{\sup} \leq \gamma \|W_n - g^*\|_{\sup}, \tag{4.44}$$

for all $n \in \mathbb{Z}_+$, or:

$$\|W_n - g^*\|_{\sup} \leq \gamma^n \|W_0 - g^*\|_{\sup}, \tag{4.45}$$

for all $n \in \mathbb{Z}_+$. This implies $\lim_{n \to \infty} W_n = g^*$.

We prove Theorem 1 by showing that $V^* = g^*$.

Proof of Theorem 1. First, for any policy $\pi \in \Pi$, we have:

$$g^*(s) = \sup_{a \in C(s)} \left(r(s,a) + \gamma g^*(f(s,a))\right) \geq r_\pi(s) + \gamma g^*(f_\pi(s)), \tag{4.46}$$

for all states $s \in \mathcal{S}$. For any $T \in \mathbb{N}$, assume:

$$g^*(s) \geq \sum_{t=0}^{T-1} \gamma^t r_\pi(f_\pi^t(s)) + \gamma^T g^*(f_\pi^T(s)), \tag{4.47}$$

for all states $s \in \mathcal{S}$. Then:

$$g^*(s) \geq \sum_{t=0}^{T-1} \gamma^t r_\pi(f_\pi^t(s)) + \gamma^T (r_\pi(f_\pi^T(s)) + \gamma g^*(f_\pi(f_\pi^T(s))))$$
$$= \sum_{t=0}^{T} \gamma^t r_\pi(f_\pi^t(s)) + \gamma^{T+1} g^*(f_\pi^{T+1}(s)), \tag{4.48}$$

for all states $s \in \mathcal{S}$. By induction, (4.47) holds for all $T \in \mathbb{N}$, and since g^* is bounded, we have:

$$g^*(s) \geq \sum_{t=0}^{\infty} \gamma^t r_\pi(f_\pi^t(s)) = V_\pi(s), \tag{4.49}$$

for all states $s \in \mathcal{S}$. Taking the supremum over all policies, we obtain $g^*(s) \geq \sup_{\pi \in \Pi} V_\pi(s) = V^*(s)$ for all states $s \in \mathcal{S}$.

We show the reverse inequality by following a modification of the proof of (Puterman, 2014, Theorem 6.3.1). Consider the sequence of bounded upper semicontinuous functions $\{V_n \in C_b^u(\mathcal{S}) : n \in \mathbb{N}\}$ generated by value iteration from an arbitrary initial guess $V_0 \in C_b^u(\mathcal{S})$. Since $\lim_{n\to\infty} V_n = g^*$, for any $\varepsilon \in \mathbb{R}_{++}$, there is a corresponding $N_\varepsilon \in \mathbb{N}$ such that:

$$\|V_n - g^*\|_{\sup} < \frac{1-\gamma}{1+\gamma}\varepsilon, \tag{4.50}$$

for all $n \in \mathbb{N}$ with $n \geq N_\varepsilon$. By Lemma 1, there is a policy $\pi_\varepsilon \in \Pi$ satisfying:

$$[\mathcal{T}_{\pi_\varepsilon}(V_{N_\varepsilon+1})](s) = r(s, \pi_\varepsilon(s)) + \gamma V_{N_\varepsilon+1}(f(s, \pi_\varepsilon(s)))$$
$$= \max_{a \in C(s)} (r(s,a) + \gamma V_{N_\varepsilon+1}(f(s,a)))$$
$$= [\mathcal{T}(V_{N_\varepsilon+1})](s), \tag{4.51}$$

for all states $s \in \mathcal{S}$. Therefore:

$$\|V_{\pi_\varepsilon} - V_{N_\varepsilon+1}\|_{\sup} \leq \|V_{\pi_\varepsilon} - \mathcal{T}(V_{N_\varepsilon+1})\|_{\sup} + \|\mathcal{T}(V_{N_\varepsilon+1}) - V_{N_\varepsilon+1}\|_{\sup}$$
$$= \|\mathcal{T}_{\pi_\varepsilon}(V_{\pi_\varepsilon}) - \mathcal{T}_{\pi_\varepsilon}(V_{N_\varepsilon+1})\|_{\sup} + \|\mathcal{T}(V_{N_\varepsilon+1}) - \mathcal{T}(V_{N_\varepsilon})\|_{\sup}$$
$$\leq \gamma\|V_{\pi_\varepsilon} - V_{N_\varepsilon+1}\|_{\sup} + \gamma\|V_{N_\varepsilon+1} - V_{N_\varepsilon}\|_{\sup}, \tag{4.52}$$

which implies:

$$\|V_{\pi_\varepsilon} - V_{N_\varepsilon+1}\|_{\sup} \leq \frac{\gamma}{1-\gamma}\|V_{N_\varepsilon+1} - V_{N_\varepsilon}\|_{\sup} \tag{4.53}$$
$$\leq \frac{\gamma}{1-\gamma}\|V_{N_\varepsilon+1} - g^*\|_{\sup} + \frac{\gamma}{1-\gamma}\|g^* - V_{N_\varepsilon}\|_{\sup}$$
$$< 2\frac{\gamma}{1-\gamma}\frac{1-\gamma}{1+\gamma}\varepsilon = \frac{2\gamma}{1+\gamma}\varepsilon. \tag{4.54}$$

Finally, we have:

$$\begin{aligned}\|V_{\pi_\varepsilon} - g^*\|_{\sup} &\leq \|V_{\pi_\varepsilon} - V_{N_\varepsilon+1}\|_{\sup} + \|V_{N_\varepsilon+1} - g^*\|_{\sup} \\ &< \frac{2\gamma}{1+\gamma}\varepsilon + \frac{1-\gamma}{1+\gamma}\varepsilon = \varepsilon. \end{aligned} \tag{4.55}$$

This means:

$$g^*(s) < V_{\pi_\varepsilon}(s) + \varepsilon \leq \left(\sup_{\pi\in\Pi} V_\pi(s)\right) + \varepsilon = V^*(s) + \varepsilon, \tag{4.56}$$

for all states $s \in \mathcal{S}$. Since ε was arbitrary, we have $g^*(s) \leq V^*(s)$ for all states $s \in \mathcal{S}$.

Since $V^* = g^*$, we conclude that V^* is bounded and upper semicontinuous. By Lemma 1, there is a policy $\pi^* \in \Pi$ satisfying:

$$\begin{aligned} V_{\pi^*}(s) &= [\mathcal{T}_{\pi^*}(V_{\pi^*})](s) = r(s,\pi^*(s)) + \gamma V^*(f(s,\pi^*(s))) \\ &= \sup_{a\in C(s)} \{r(s,a) + \gamma V^*(f(s,a))\} = [\mathcal{T}(V^*)](s) = V^*(s), \end{aligned} \tag{4.57}$$

for all states $s \in \mathcal{S}$. This means π^* is an optimal policy. Moreover, since g^* is the limit of value iteration from any bounded and upper semicontinuous initial guess, V^* is as well. □

We conclude this section with a lemma mentioned in Remark 3, allowing Assumption 3 to be relaxed slightly.

Lemma 2. *Suppose an MDP* $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ *and policy class* Π *satisfy Assumption 3, suppose the projection of* $\mathrm{Graph}(C)$ *onto* $\mathcal{A}$ *is compact, and let* $\rho_{\mathcal{A}} : \mathcal{A}\times\mathcal{A} \to \mathbb{R}_+$ *denote the metric on* $\mathcal{A}$*. For any state-action pair* $(s,a) \in \mathrm{Graph}(C)$*, the function* $\varphi_{s,a} : \mathrm{Graph}(C) \to \mathbb{R}$ *defined as:*

$$\varphi_{s,a}(s',a') = -\rho_{\mathcal{A}}(a',a), \tag{4.58}$$

for all state-action pairs $(s',a') \in \mathrm{Graph}(C)$ *admits a maximizing policy* $\pi_{s,a} \in \Pi$ *for which* $\pi_{s,a}(s) = a$*.*

Proof. Since $\rho_{\mathcal{A}}$ is continuous, so is $\varphi_{s,a}$. Since the projection of $\mathrm{Graph}(C)$ onto $\mathcal{A}$ is compact, $\varphi_{s,a}$ is bounded. Since $\varphi_{s,a} \in C_b^0(\mathcal{S}) \subseteq C_b^u(\mathcal{S})$, Lemma 1 justifies the existence of a policy $\pi_{s,a} \in \Pi$ satisfying:

$$\varphi_{s,a}(s',\pi_{s,a}(s')) = \max_{a'\in C(s')} \varphi_{s,a}(s',a') = -\min_{a'\in C(s')} \rho_{\mathcal{A}}(a',a), \tag{4.59}$$

for all states $s' \in \mathcal{S}$. In particular, we have $\pi_{s,a}(s) = a$ as $\varphi_{s,a}(s,\pi_{s,a}(s)) = -\min_{a'\in C(s)} \rho_{\mathcal{A}}(a',a) = 0$. □

4.3.3 Compactly Restrictable MPOPs

While Assumption 3 allows us to determine when MPOPs are well-posed and can be solved using value iteration, it is not met for many systems of interest. We now outline an alternative assumption that not only mitigates theoretical shortcomings, but also better fits the problem settings for physical systems. Our main result here is Theorem 2, showing how an MPOP satisfying these new assumptions can generate an MPOP satisfying Assumption 3 through a systematic transformation. We call the original MPOP a CR-MPOP.

4.3.3.1 Motivation

Consider an MPOP defined by an MDP $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ and a policy class Π that satisfies Assumption 3. The proof of Theorem 1 requires a bounded reward function r and a compact-valued action-admissibility correspondence C. For many examples, these requirements are too strict, as shown in Example 4. If C can be modified such that its graph is compact and if r is continuous, then r will be bounded over the graph of the modified correspondence. This guides our systematic construction of a well-posed policy optimization problem from an ill-posed one.

If the graph of the modified correspondence is compact, the projection of the graph onto the state space must also be compact. Therefore, if the state space is not already compact, then the modified correspondence must be defined on a compact strict subset $\mathcal{S}_0 \subset \mathcal{S}$; that is, the modified correspondence must have the form $C_0 : \mathcal{S}_0 \to \mathcal{P}(\mathcal{A})$, where $C_0(s) \subseteq C(s)$ is nonempty and compact for all states $s \in \mathcal{S}_0$. Moreover, while restricting admissible actions to compact sets is reasonable for physical systems (as forces, voltages, currents, etc. are bounded by physical constraints), these restrictions must be chosen carefully to enforce the nonemptiness condition. For the restriction of f to $\mathrm{Graph}(C_0)$ to be a well-defined transition function, we require $f(s, a) \in \mathcal{S}_0$ for all pairs $(s, a) \in \mathrm{Graph}(C_0)$. Finally, we must ensure that C_0 admits measurable selectors. Restricting our attention to a compact subset $\mathcal{S}_0$ is also reasonable for many physical systems, as dynamics models are often well-understood only in a bounded set around an operating condition.

4.3.3.2 Compact Restriction

For this construction, we use policies that render compact subsets of the state space forward-invariant. A policy $\pi \in \Pi$ renders a subset $\mathcal{S}_0 \subseteq \mathcal{S}$ forward-invariant if $f_\pi(\mathcal{S}_0) \subseteq \mathcal{S}_0$. By induction, we have $f_\pi^t(\mathcal{S}_0) \subseteq \mathcal{S}_0$ for any $t \in \mathbb{Z}_+$.

Definition 4 (Compactly Restrictable Metric Policy Optimization Problem)**.** An MPOP characterized by an MDP $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ and a policy class Π is a *compactly restrictable metric policy optimization problem* (CR-MPOP) if there is a nonempty compact subset $\mathcal{S}_0 \subseteq \mathcal{S}$ and a continuous policy $\pi_0 \in \Pi$ rendering $\mathcal{S}_0$ forward-invariant.

Assumption 4. The action-admissibility correspondence C has closed values and $\text{Graph}(C)$ is closed, the transition function f and reward function r are continuous, and the policy class Π is the set of all measurable selectors of C.

To show how an CR-MPOP satisfying Assumption 4 can lead to an MPOP satisfying Assumption 3, we need the following short lemma.

Lemma 3. *Consider an MDP $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ and a policy class Π satisfying Assumption 4. For any state $s \in \mathcal{S}$, the s-section $f_s : C(s) \to \mathcal{S}$ is continuous. Additionally, for any state $s \in \mathcal{S}$ and any closed set $G \subseteq \mathcal{S}$, the preimage $(f_s)^{-1}(G)$ is a closed subset of $\mathcal{A}$.*

Proof. Let $\text{id} : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \times \mathcal{A}$ denote the identity function on $\mathcal{S} \times \mathcal{A}$. For any state $s \in \mathcal{S}$, the s-section $\text{id}_s : \mathcal{A} \to \mathcal{S} \times \mathcal{A}$ is continuous, meaning the restriction $\text{id}_s|_{C(s)} : C(s) \to \mathcal{S} \times \mathcal{A}$ is continuous. We can write $f_s = f \circ (\text{id}_s|_{C(s)})$, implying f_s is continuous. For any closed set $G \subseteq \mathcal{S}$, there is a corresponding closed set $F_{s,G} \subseteq \mathcal{A}$ satisfying $(f_s)^{-1}(G) = C(s) \cap F_{s,G}$. Since $C(s)$ is also a closed subset of $\mathcal{A}$, we conclude that $(f_s)^{-1}(G)$ a closed subset of $\mathcal{A}$. □

We now present our guarantee of the existence of a well-posed value iteration settings under Assumption 4.

Theorem 2. *Consider a CR-MPOP characterized by an MDP $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ and a policy class Π that satisfies Assumption 4. There exists an MPOP characterized by an MDP $(\mathcal{S}_0, \mathcal{A}_0, C_0, f_0, r_0, \gamma)$ and a policy class Π_0 satisfying Assumption 3, with $\mathcal{S}_0 \subseteq \mathcal{S}$ rendered forward-invariant by a continuous policy in Π, $\mathcal{A}_0 \subseteq \mathcal{A}$, $C_0 : \mathcal{S}_0 \to \mathcal{P}(\mathcal{A}_0)$ satisfying $C_0(s) \subseteq C(s)$ for all states $s \in \mathcal{S}_0$, $f_0 = f|_{\text{Graph}(C_0)}$, $r_0 = r|_{\text{Graph}(C_0)}$, and:*

$$\Pi_0 = \{\pi|_{\mathcal{S}_0} : \pi \in \Pi, \pi(\mathcal{S}_0) \subseteq \mathcal{A}_0, \text{ and } f_\pi(\mathcal{S}_0) \subseteq \mathcal{S}_0\}. \tag{4.60}$$

Proof. For some $n \in \mathbb{N}$, consider continuous policies $\pi_1, \ldots, \pi_n \in \Pi$ satisfying $f_{\pi_i}(\mathcal{S}_0) \subseteq \mathcal{S}_0$ for all $i \in \{1, \ldots, n\}$. Such policies are guaranteed to exist by the

existence of the continuous policy $\pi_0 \in \Pi$. Consider the union of the images of $\mathcal{S}_0$ under each of these policies, defining the compact set $\mathcal{A}_0 = \bigcup_{i=1}^{n} \pi_i(\mathcal{S}_0)$. Since $\mathcal{S}$ and $\mathcal{A}$ are separable metric spaces, so are $\mathcal{S}_0$ and $\mathcal{A}_0$ (Aliprantis and Border, 2006, Corollary 3.5).

Define the correspondence $C_0 : \mathcal{S}_0 \to \mathcal{P}(\mathcal{A}_0)$ as:

$$C_0(s) = \{a \in C(s) \cap \mathcal{A}_0 : f(s,a) \in \mathcal{S}_0\} = (f_s)^{-1}(\mathcal{S}_0) \cap \mathcal{A}_0, \tag{4.61}$$

for all states $s \in \mathcal{S}_0$. Since $\pi_1(s), \ldots, \pi_n(s) \in C_0(s)$, $C_0(s) \neq \emptyset$ for all $s \in \mathcal{S}_0$. By Lemma 3 and since $\mathcal{S}_0$ is closed (it is compact in a metric space), the preimage $(f_s)^{-1}(\mathcal{S}_0)$ is a closed subset of $\mathcal{A}$. $C_0(s)$ is thus compact, as it is a closed subset of $\mathcal{A}_0$. For any closed set $G \subseteq \mathcal{A}_0$, the lower preimage of G under C_0 satisfies:

$$\begin{aligned}(C_0)^{\ell}(G) &= \{s \in \mathcal{S}_0 : f(s,a) \in \mathcal{S}_0 \text{ for some } a \in C(s) \cap G\} \\ &= p_{\mathcal{S}}(f^{-1}(\mathcal{S}_0) \cap (\mathcal{S}_0 \times G)),\end{aligned} \tag{4.62}$$

where $p_{\mathcal{S}} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ denotes the canonical projection onto $\mathcal{S}$. Since $\mathcal{S}_0$ is closed and f is continuous, there is a closed set $F \subseteq \mathcal{S} \times \mathcal{A}$ satisfying $f^{-1}(\mathcal{S}_0) = \text{Graph}(C) \cap F$. Since $\text{Graph}(C)$ is also closed, $f^{-1}(\mathcal{S}_0)$ is a closed subset of $\mathcal{S} \times \mathcal{A}$. Since G is a closed subset of the compact set $\mathcal{A}_0$, G is compact, and since $\mathcal{S}_0$ is compact, the product $\mathcal{S}_0 \times G$ is compact. As a closed subset of a compact set, $f^{-1}(\mathcal{S}_0) \cap (\mathcal{S}_0 \times G)$ is compact. Since the projection operator $p_{\mathcal{S}}$ is continuous, $(C_0)^{\ell}(G)$ is compact. Therefore, $(C_0)^{\ell}(G)$ is a closed subset of $\mathcal{S}_0$, and since G was arbitrary, C_0 is upper hemicontinuous.

As restrictions of continuous functions, f_0 and r_0 are continuous, implying r_0 is upper semicontinuous. Note that:

$$\begin{aligned}\text{Graph}(C_0) &= \{(s,a) \in \mathcal{S}_0 \times \mathcal{A}_0 : f(s,a) \in \mathcal{S}_0\} \\ &= f^{-1}(\mathcal{S}_0) \cap (\mathcal{S}_0 \times \mathcal{A}_0).\end{aligned} \tag{4.63}$$

Since $\mathcal{A}_0$ is trivially a closed subset of $\mathcal{A}_0$, we have already shown that $f^{-1}(\mathcal{S}_0) \cap (\mathcal{S}_0 \times \mathcal{A}_0)$ is compact, meaning $\text{Graph}(C_0)$ is compact. Therefore, since r_0 is continuous, it is bounded.

The policy class Π_0 satisfies:

$$\begin{aligned}\Pi_0 &= \{\pi|_{\mathcal{S}_0} : \pi \in \Pi, \pi(s) \in C_0(s) \text{ for all states } s \in \mathcal{S}_0\} \\ &= \{\pi \in \mathcal{L}^0(\mathcal{S}_0; \mathcal{A}_0) : \pi(s) \in C_0(s) \text{ for all states } s \in \mathcal{S}_0\}.\end{aligned} \tag{4.64}$$

To verify the last equality, first consider a policy $\pi \in \Pi$ satisfying $\pi(s) \in C_0(s)$ for all states $s \in \mathcal{S}_0$. The restriction $\pi|_{\mathcal{S}_0}$ is a measurable function from $\mathcal{S}_0$ to $\mathcal{A}_0$ selecting from C_0. Conversely, consider a measurable function $\pi : \mathcal{S}_0 \to \mathcal{A}_0$ selecting from C_0. Pick any policy $\pi_e \in \Pi$ and define the extension $\bar{\pi} : \mathcal{S} \to \mathcal{A}$ as π on $\mathcal{S}_0$ and π_e on $\mathcal{S} \setminus \mathcal{S}_0$. For any measurable set $B \subseteq \mathcal{A}$, we have:

$$\begin{aligned}\bar{\pi}^{-1}(B) &= \{s \in \mathcal{S}_0 : \pi(s) \in \mathcal{A}_0 \cap B\} \cup \{s \in \mathcal{S} \setminus \mathcal{S}_0 : \pi_e(s) \in B\} \\ &= \pi^{-1}(\mathcal{A}_0 \cap B) \cup (\pi_e^{-1}(B) \setminus \mathcal{S}_0),\end{aligned} \tag{4.65}$$

which is a measurable set; therefore, $\bar{\pi}$ is a measurable function. Note that $\bar{\pi}(s) = \pi(s) \in C_0(s) \subseteq C(s)$ for all states $s \in \mathcal{S}_0$ and $\bar{\pi}(s) = \pi_e(s) \in C(s)$ for all states $s \notin \mathcal{S}_0$. Since $\bar{\pi}$ is a measurable selector of C, we have $\bar{\pi} \in \Pi$, and inclusion follows since $\bar{\pi}|_{\mathcal{S}_0} = \pi$.

Finally, since $\text{Graph}(C_0)$ is compact, its projection onto $\mathcal{A}_0$ is compact. Therefore, by Lemma 2, for every state-action pair $(s, a) \in \text{Graph}(C_0)$, there is a policy $\pi_{s,a} \in \Pi_0$ satisfying $\pi_{s,a}(s) = a$. □

Remark 5. An MPOP characterized by an MDP $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ and a policy class Π that satisfies Assumption 3 is trivially a CR-MPOP satisfying Assumption 4 if the following sufficient conditions are met:

1. The state space $\mathcal{S}$ is compact,
2. The reward function r is continuous,
3. The policy class Π contains a continuous policy.

Example 5. Consider the MDP $(\mathcal{S}, \mathcal{A}, C, f, r, \gamma)$ defined in Example 3 with a policy class Π comprised of measurable selectors of C, and let $\mathcal{S}_0 = [-1, 1]$. Note that $|f(0, a)| = |\tanh(a)| < 1$ for all actions $a \in \mathbb{R}$. Consider any state $s \in [-1, 0)$. For any nonnegative action $a \in \mathbb{R}_+$, we have $-1 \leq s \leq f(s, a) = s + \tanh(a) < s + 1 < 1$, and $f(s, -a) = s + \tanh(-a) \in [-1, s)$ if and only if $-a \in [\tanh^{-1}(-1 - s), 0)$ since $\tanh^{-1}$ is monotonically increasing. By a similar argument for states in $(0, 1]$, we determine that, for all $s \in \mathcal{S}_0$ and $a \in \mathcal{A}$, we have $f(s, a) \in \mathcal{S}_0$ if and only if:

$$a \in \begin{cases} [\tanh^{-1}(-1 - s), \infty) & -1 \leq s < 0, \\ \mathcal{A} & s = 0, \\ (-\infty, \tanh^{-1}(1 - s)] & 0 < s \leq 1. \end{cases} \tag{4.66}$$

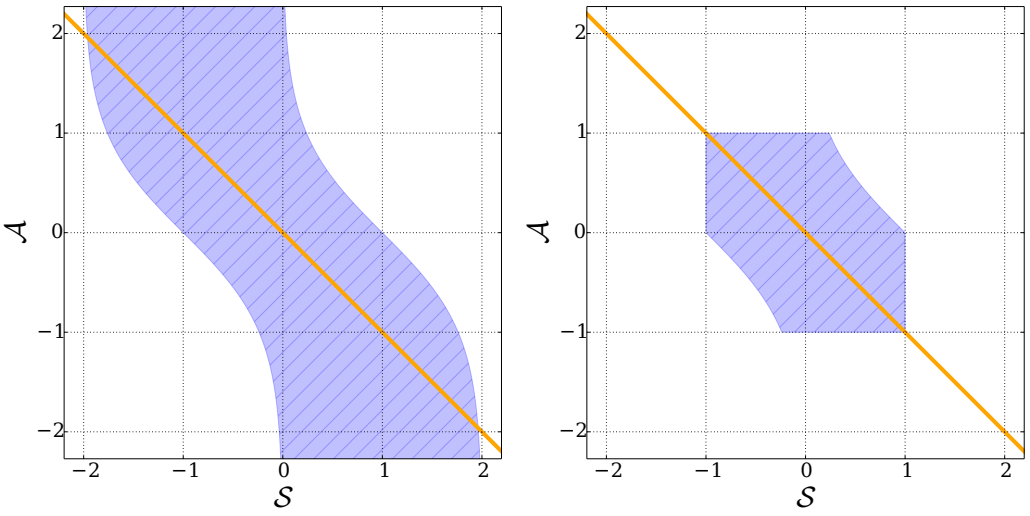


Figure 4.1: Illustration of compact restriction. (Left) The preimage $f^{-1}(\mathcal{S}_0)$ is shown in blue, which contains all state-action pairs mapped into $\mathcal{S}_0 = [-1, 1]$ by f. For any state in $\mathcal{S}_0$, the set of actions mapping that state into $\mathcal{S}_0$ is unbounded, therefore not compact. (Right) The graph of π_0 is shown in orange and the graph of C_0 is shown in blue. The graph of C_0 is compact, and for every state $s \in \mathcal{S}_0$, the set $C_0(s)$ is compact.

A correspondence coinciding with the requirements on actions in (4.66) would not have compact values; this is illustrated by the preimage $f^{-1}(\mathcal{S}_0)$ in Figure 4.1. Therefore, consider the continuous policy $\pi_0 \in \Pi$ defined as $\pi_0(s) = -s$ for all states $s \in \mathcal{S}$. For any state $s \in [-1, 0)$, we have $\pi_0(s) = -s > 0 \geq \tanh^{-1}(-1 - s)$. Similarly, for any state $s \in (0, 1]$, we have $\pi_0(s) < \tanh^{-1}(1 - s)$. Therefore, $f_{\pi_0}(\mathcal{S}_0) \subseteq \mathcal{S}_0$. The image of $\mathcal{S}_0$ under π_0 is $[-1, 1]$. Therefore, we can define $\mathcal{A}_0 = [-1, 1]$ and $C_0 : \mathcal{S}_0 \to \mathcal{P}(\mathcal{A}_0)$ as, for all states $s \in \mathcal{S}_0$:

$$C_0(s) = \begin{cases} [\tanh^{-1}(-1 - s), 1] & s \leq -1 - \tanh(-1), \\ [-1, \tanh^{-1}(1 - s)] & s \geq 1 - \tanh(1), \\ [-1, 1] & \text{otherwise.} \end{cases} \tag{4.67}$$

4.3.4 Policy Iteration

We briefly mention the difficulties of policy iteration for MPOPs, as noted in (Yu and Bertsekas, 2015). For a well-posed MPOP, policy iteration generates a monotonically nondecreasing sequence of policies. Given an initial policy $\pi_0 \in \Pi$, we seek a sequence of policies $\{\pi_n \in \Pi : n \in \mathbb{N}\}$ satisfying:

$$r(s, \pi_{n+1}(s)) + \gamma V_{\pi_n}(f(s, \pi_{n+1}(s))) = \sup_{a \in C(s)} \{r(s, a) + \gamma V_{\pi_n}(f(s, a))\}, \tag{4.68}$$

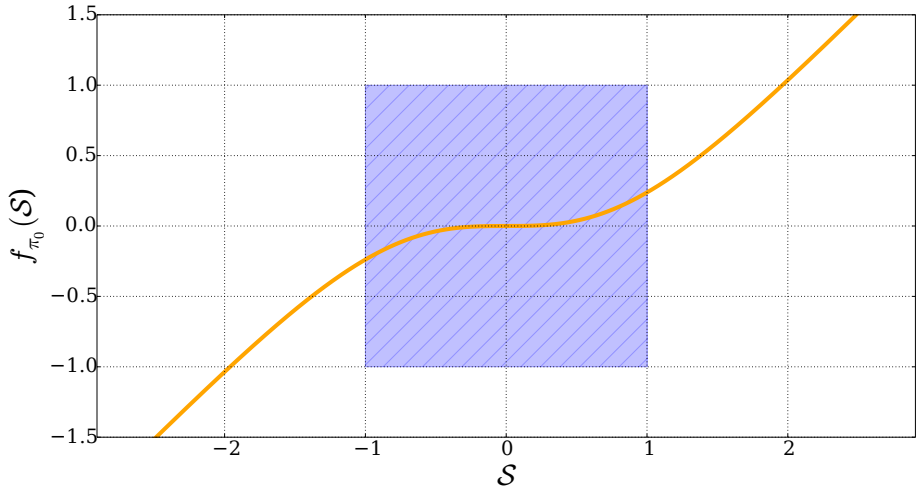


Figure 4.2: The graph of f_{π_0} is shown in orange, and the rectangle $\mathcal{S}_0 \times \mathcal{S}_0 = [-1, 1] \times [-1, 1]$ is shown in blue. Importantly, the image of $\mathcal{S}_0$ under f_{π_0} satisfies $f_{\pi_0}(\mathcal{S}_0)$; that is, f_{π_0} renders $\mathcal{S}_0$ forward invariant.

for all states $s \in \mathcal{S}$ and $n \in \mathbb{Z}_+$. If f, r, and π_0 are continuous, then so are f_{π_0} and r_{π_0} which we can show renders V_{π_0} continuous. This implies $r + \gamma V_{\pi_0} \circ f$ is continuous. If r is bounded, then so are V_{π_0} and $r + \gamma V_{\pi_0} \circ f$. Since $r + \gamma V_{\pi_0} \circ f$ is upper semicontinuous and bounded, we can show that for any state $s \in \mathcal{S}$, the optimization problem:

$$\sup_{a \in C(s)} \{r(s, a) + \gamma V_{\pi_0}(f(s, a))\}, \tag{4.69}$$

is solved by the next policy π_1, a measurable selector of C. However, we cannot conclude that $r + \gamma V_{\pi_1} \circ f$ is upper semicontinuous and bounded, so the same argument cannot be applied iteratively.

Demonstrating that policy iteration can be applied requires special knowledge that the policy improvement step in (4.68) continues to produce policies with value functions that permit maximization and appropriate selection (for the chosen policy class). Such knowledge is available in linear-quadratic regulator (LQR) problems with linear policy classes (see (Tu, 2019) for the discounted case).

4.3.5 Computation

There are 3 central difficulties in computationally implementing value iteration for a well-posed MPOP:

1. Representing iterates during value iteration,

2. Determining admissible actions, and

3. Approximating the update step (4.22).

Typically, iterates are represented using function approximators such as neural networks (Sutton and Barto, 2018) or Gaussian process regression models (Kuss and Rasmussen, 2003). Training such approximators involves sampling sufficiently many states from the state space region of interest. For any given state, the maximization in the update step (4.22) is generally a nonconvex optimization, with possible approximations including (projected) gradient ascent from several initial action seeds or maximization over a finite but sufficiently dense sampling of admissible actions. If the function approximation and optimization approximation errors can be controlled, approximate convergence of value iteration can be guaranteed (Bertsekas, 2022, Proposition 2.3.2).

Any approximate optimization approach requires checking admissibility of actions, which, in the case of a CR-MPOP, requires checking whether or not a state-action pair is mapped to the correct compact set under the transition map. This necessitates efficient set membership checking, and for nonconvex compact sets, proximity-based membership approximations may be needed.

Finally, in Section 4.4.3, we will use the closure of a reachable set (under a specific policy) to restrict problems with feedback linearizable control affine systems to well-posed MPOPs. To sample a state from the reachable set, we can sample a state from the appropriate set of initial conditions and follow the policy for a number of steps sampled from a geometric distribution (with success probability $1 - \gamma$). The resulting distribution is the discounted state distribution under the policy (Silver et al., 2014), which is supported on the entirety of the reachable set (thus, on a dense subset of its closure).

4.4 Control Affine Systems

We now show how to apply the results from Section 4.3 to a general class of control systems called control affine systems (LaValle, 2006, Section 13.2.3). For state and action space dimensions $d, m \in \mathbb{N}$, respectively, consider a set $\mathcal{D} \subseteq \mathbb{R}^d$ and vector fields $f_0, g_1, \ldots, g_m : \mathcal{D} \to \mathbb{R}^d$. Define the matrix-valued function $G : \mathcal{D} \to \mathbb{R}^{d \times m}$ with columns $g_1, \ldots, g_m$. Define $F : \mathcal{D} \times \mathbb{R}^m \to \mathbb{R}^d$ as:

$$F(x, u) = f_0(x) + G(x)u, \tag{4.70}$$

for all $x \in \mathcal{D}$ and $u = (u_1, \ldots, u_m) \in \mathbb{R}^m$. An initial value problem with constant control input $u \in \mathbb{R}^m$ is characterized by an initial condition $x \in \mathcal{D}$ and an open time interval $I \subseteq \mathbb{R}$ with $0 \in I$; a corresponding solution is a differentiable function $\phi : I \to \mathcal{D}$ satisfying $\phi(0) = x$ and $\dot{\phi}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\phi(t) = F(\phi(t), u) = f_0(\phi(t)) + G(\phi(t))u$ for all times $t \in I$.

4.4.1 Time Sampling

In contrast to continuous-time control design, we consider sampled-data control design (see (Nešić, Teel, and Kokotović, 1999; Taylor, Dorobantu, Yue, et al., 2021; Taylor, Dorobantu, Cosner, et al., 2022)), in which initial value problems with constant control characterize the evolution of a system over fixed sample intervals, resulting in control input trajectories that are piecewise constant in time. Such assumptions are realistic for physical systems interacting with digital controllers, which measure states and compute control inputs at nearly fixed frequencies. This setting requires the time intervals over which solutions are defined to be sufficiently long, uniformly for all initial conditions and control inputs under consideration.

Specifically, fix a sample period $h \in \mathbb{R}_{++}$, and define the subset $\mathcal{S}_h \subseteq \mathcal{D}$ and correspondence $C_h : \mathcal{S}_h \to \mathcal{P}(\mathbb{R}^m)$ such that the following properties are satisfied:

- For every initial condition $x \in \mathcal{S}_h$, the corresponding set of control inputs $C_h(x) \subseteq \mathbb{R}^m$ is nonempty,
- For every $(x, u) \in \mathrm{Graph}(C_h)$, there is a unique solution to the initial value problem characterized by initial condition x, control input u, and open time interval $I \subseteq \mathbb{R}$ with $[0, h] \subset I$, with the solution denoted $\phi_{x,u} : I \to \mathcal{D}$,
- $\mathcal{S}_h$ and C_h are maximal (cannot be contained in a superset and containing correspondence).

Define $f_h : \mathrm{Graph}(C_h) \to \mathcal{D}$ for all pairs $(x, u) \in \mathrm{Graph}(C_h)$ as:

$$f_h(x, u) = \phi_{x,u}(h). \tag{4.71}$$

Example 6 (Linear System)**.** Let $\mathcal{D} = \mathbb{R}^d$. For matrices $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times m}$, suppose $f_0(x) = Ax$ and $G(x) = B$ for all initial conditions $x \in \mathbb{R}^d$. Then $\mathcal{S}_h = \mathbb{R}^d$ and $C_h(x) = \mathbb{R}^m$ for all $x \in \mathbb{R}^d$. Moreover, f_h can be expressed explicitly as a linear function in terms of the matrix exponential of A.

Example 7 (Continuously Differentiable and Lipschitz Continuous Vector Fields)**.** Suppose $\mathcal{D} = \mathbb{R}^d$ and $f_0, g_1, \ldots, g_m$ are continuously differentiable and (globally) Lipschitz continuous. For a control input $u \in \mathbb{R}^m$, since the u-section $F^u : \mathbb{R}^d \to \mathbb{R}^d$ is a linear combination of continuously differentiable and Lipschitz continuous vector fields, it is continuously differentiable and Lipschitz continuous. Therefore, for any initial condition $x \in \mathbb{R}^d$ the initial value problem characterized by initial condition x, control input u, and time interval $\mathbb{R}$ has a unique solution (Perko, 2013, Theorem 3.1.3). Therefore, $\mathcal{S}_h = \mathbb{R}^d$ and $C_h(x) = \mathbb{R}^m$ for all initial conditions $x \in \mathbb{R}^d$. In general, f_h does not have an closed-form representation.

Assumption 5. The vector fields $f_0, g_1, \ldots, g_m$ are locally Lipschitz continuous. Moreover, there exist subsets $\mathcal{S} \subseteq \mathcal{S}_h$ and $\mathcal{A} \subseteq \mathbb{R}^m$ and a correspondence $C : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ such that:

- For every initial condition $x \in \mathcal{S}$, the set of control inputs $C(x) \subseteq \mathcal{A}$ is nonempty with $C(x) \subseteq C_h(x)$,
- $\text{Graph}(C)$ is a closed subset of $\mathcal{S} \times \mathcal{A}$,
- $f_h(\text{Graph}(C)) \subseteq \mathcal{S}$.

Both Examples 6 and 7 satisfy Assumption 5 with $\mathcal{S} = \mathcal{S}_h$, $\mathcal{A} = \mathbb{R}^m$, and $C(x) = \mathcal{A}$ for all $x \in \mathcal{S}$. The assumption of local Lipschitz continuity in Assumption 5 ensures that the restriction of f_h to $\text{Graph}(C)$ is continuous; this follows from (Khalil, 2002, Theorem 3.5) by treating control inputs as parameters.

4.4.2 Robotic Systems

Many robotic systems can be modeled as control affine systems, and we will demonstrate how to show that such systems can satisfy Assumption 4, establishing well-posedness of value iteration for a large class of practically relevant MPOPs. Examples of such systems include manipulators, automobiles, aircraft, and spacecraft (Murray, Li, and Sastry, 1994; Olfati-Saber, 2001).

4.4.2.1 Dynamics

An unconstrained robotic system with $n \in \mathbb{N}$ degrees of freedom is characterized by an n-dimensional C^2 manifold Q called the configuration manifold. In this work, we consider open subsets $Q \subseteq \mathbb{R}^n$ for which we can identify the tangent bundle TQ with $Q \times \mathbb{R}^n$. The inertia matrix function $D : Q \to \mathbb{S}^n_{++}$ characterizes the kinetic energy

function $T : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}_{++}$, defined as $T(q, \dot{q}) = \frac{1}{2}\dot{q}^\top D(q)\dot{q}$ for all configurations $q \in \mathcal{Q}$ and velocities $\dot{q} \in \mathbb{R}^n$. The assumption that D takes positive definite values ensures that no configurations exist that admit arbitrarily large velocities without affecting the kinetic energy of the system. We require D to be differentiable, allowing us to express the matrix-valued function $C : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^{n \times n}$ of Coriolis and centrifugal terms[2] as:

$$(C(q, \dot{q}))_{ij} = \sum_{k=1}^{n} \left(\frac{\partial}{\partial q_j}(D(q))_{ik} - \frac{1}{2}\frac{\partial}{\partial q_i}(D(q))_{jk} \right) \dot{q}_k, \tag{4.72}$$

for all configurations $q = (q_1, \ldots, q_n) \in \mathcal{Q}$, velocities $\dot{q} = (\dot{q}_1, \ldots, \dot{q}_n) \in \mathbb{R}^n$, and indices $i, j \in \{1, \ldots, n\}$ (Murray, Li, and Sastry, 1994, Equation 4.21). We also require the potential energy function $U : \mathcal{Q} \to \mathbb{R}^n$ to be differentiable, and we denote external nonconservative forces and torques applied to the system with the vector-valued function $F_{\text{ext}} : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^n$. Finally, if the system is controlled with $m \in \mathbb{N}$ actuators, then $B : \mathcal{Q} \to \mathbb{R}^{n \times m}$ denotes the actuation matrix function, converting control inputs to forces and torques. With $d = 2n$ and $\mathcal{D} = \mathcal{Q} \times \mathbb{R}^n$, we define $F : \mathcal{D} \times \mathbb{R}^m \to \mathbb{R}^d$ as:

$$F(x, u) = \begin{bmatrix} \dot{q} \\ D(q)^{-1}(F_{\text{ext}}(q, \dot{q}) - C(q, \dot{q})\dot{q} - \nabla U(q)) \end{bmatrix} + \begin{bmatrix} 0 \\ D(q)^{-1}B(q) \end{bmatrix} u, \tag{4.73}$$

for all $x = (q, \dot{q}) \in \mathcal{Q} \times \mathbb{R}^n$ and $u \in \mathbb{R}^m$.

4.4.2.2 Regularity

We now establish regularity conditions that enable us to define a valid transition function that can be used in an MDP. We first establish sufficient conditions such that every initial condition and control input under consideration correspond to initial value problems with unique solutions. We then establish sufficient conditions for these solutions to exist for all nonnegative time.

Assumption 6. The configuration manifold satisfies $\mathcal{Q} = \mathbb{R}^n$. The functions D, C, ∇U, F_{ext}, and B are each locally Lipschitz continuous. There is a strictly positive lower bound $\lambda_{\min}$ such that $\lambda_{\min} I_n \preceq D(q)$ for all configurations $q \in \mathbb{R}^n$. There is some closed set $G \subseteq \mathbb{R}^m$ such that for any control input $u \in G$ there are corresponding constants $c_0 \in \mathbb{R}_+$ and $c_1 \in \mathbb{R}_{++}$ such that:

$$\|F_{\text{ext}}(q, \dot{q}) - \nabla U(q) + B(q)u\|_2 \leq c_0 + c_1 \|\dot{q}\|_2, \tag{4.74}$$

[2] The use of C for this term is standard, but we will not use it in the same context as an action-admissibility correspondence.

for all configurations $q \in \mathbb{R}^n$ and velocities $\dot{q} \in \mathbb{R}^n$.

Remark 6. If the derivative of D is locally Lipschitz continuous, then so is D itself as it is continuously differentiable, and so is C as it is bilinear in the derivative of D and the velocities. If D is twice continuously differentiable, then the derivative of D is locally Lipschitz continuous, implying D and C are as well. If U is twice continuously differentiable, then ∇U is locally Lipschitz continuous.

Remark 7. If B is bounded by some $M \in \mathbb{R}_+$ and there are constants $d_0 \in \mathbb{R}_+$ and $d_1 \in \mathbb{R}_{++}$ such that:

$$\|F_{\text{ext}}(q, \dot{q}) - \nabla U(q)\|_2 \leq d_0 + d_1 \|\dot{q}\|_2, \tag{4.75}$$

for all configurations $q \in \mathbb{R}^n$ and velocities $\dot{q} \in \mathbb{R}^n$, then for any $u \in \mathbb{R}^m$, we have:

$$\|F_{\text{ext}}(q, \dot{q}) - \nabla U(q) + B(q)u\|_2 \leq d_0 + M\|u\|_2 + d_1 \|\dot{q}\|_2, \tag{4.76}$$

for all configurations $q \in \mathbb{R}^n$ and velocities $\dot{q} \in \mathbb{R}^n$. Therefore, choose $c_0 = d_0 + M\|u\|_2$ and $c_1 = d_1$, as well as $G = \mathbb{R}^m$.

The proof of (Rudin, 1976, Theorem 9.8b) shows that matrix inversion is locally Lipschitz continuous. Fixing a control input $u \in G$, the u-section F^u is locally Lipschitz continuous as it is comprised of sums and products of locally Lipschitz continuous functions. Fix an initial configuration $q_0 \in \mathbb{R}^n$ and velocity $\dot{q}_0 \in \mathbb{R}^n$. There is a corresponding maximal open interval $I_{\max} \subseteq \mathbb{R}$ with $0 \in I_{\max}$ such that the initial value problem characterized by initial condition $(q_0, \dot{q}_0)$, control input u, and time interval $I_{\max}$ has a unique solution $\phi : I_{\max} \to \mathbb{R}^{2n}$ (Perko, 2013, Theorem 2.4.1) (the proof of this theorem applies to locally Lipschitz continuous vector fields, not just continuously differentiable vector fields). Let $\psi : I_{\max} \to \mathbb{R}^n$ satisfy $\phi(t) = (\psi(t), \dot{\psi}(t))$ for all times $t \in I_{\max}$, and note that $\psi(0) = q_0$ and $\dot{\psi}(0) = \dot{q}_0$. Let $I^+_{\max} = \mathbb{R}_+ \cap I_{\max}$. The kinetic energy satisfies:

$$\begin{aligned}
&T(\psi(t), \dot{\psi}(t)) - T(q_0, \dot{q}_0) \\
&\quad = \int_0^t \dot{\psi}(s)^\top (F(\psi(s), \dot{\psi}(s)) - \nabla U(\psi(s)) + B(\psi(s))u) \, \mathrm{d}s \\
&\quad \leq \int_0^t \|\dot{\psi}(s)\|_2 (c_0 + c_1 \|\dot{\psi}(s)\|_2) \, \mathrm{d}s,
\end{aligned} \tag{4.77}$$

for all times $t \in I^+_{\max}$. Since $T(q, \dot{q}) \geq \frac{1}{2}\lambda_{\min}\|\dot{q}\|_2^2$ for all configurations $q \in \mathbb{R}^n$ and velocities $\dot{q} \in \mathbb{R}^n$, we have:

$$\frac{1}{2}\left(\sqrt{\lambda_{\min}}\|\dot{\psi}(t)\|_2\right)^2 \leq \frac{1}{2}\left(\sqrt{2T(q_0, \dot{q}_0)}\right)^2 + \int_0^t \frac{c_0 + c_1\|\dot{\psi}(s)\|_2}{\sqrt{\lambda_{\min}}}(\sqrt{\lambda_{\min}}\|\dot{\psi}(s)\|_2)\, \mathrm{d}s, \tag{4.78}$$

for all times $t \in I^+_{\max}$. For any upper bound $t_f \in I^+_{\max}$ with $t_f > 0$, the functions $\phi_1, \phi_2 : [0, t_f] \to \mathbb{R}_+$ defined as:

$$\phi_1(t) = \sqrt{\lambda_{\min}}\|\dot{\psi}(t)\|_2, \qquad \phi_2(t) = \frac{c_0 + c_1\|\dot{\psi}(t)\|_2}{\sqrt{\lambda_{\min}}}, \tag{4.79}$$

for all $t \in [0, t_f]$ are continuous and bounded, implying ϕ_2 is absolutely integrable. Therefore, from (Ballard, 2000, Lemma 17) (originally (Brezis, 1973, Lemma A.5)), we have:

$$\sqrt{\lambda_{\min}}\|\dot{\psi}(t)\|_2 \leq \sqrt{2T(q_0, \dot{q}_0)} + \int_0^t \frac{c_0 + c_1\|\dot{\psi}(s)\|_2}{\sqrt{\lambda_{\min}}}\, \mathrm{d}s, \tag{4.80}$$

for all times $t \in [0, t_f]$. By the Gronwall-Bellman inequality (Khalil, 2002, Lemma A.1), we have:

$$\|\dot{\psi}(t)\|_2 \leq \left(\sqrt{\frac{2T(q_0, \dot{q}_0)}{\lambda_{\min}}} + \frac{c_0 t}{\lambda_{\min}}\right) e^{\frac{c_1}{\lambda_{\min}}t}, \tag{4.81}$$

for all times $t \in [0, t_f]$. Since t_f was arbitrary, the bound (4.81) holds for all times $t \in I^+_{\max}$.

We now use (4.81) to show that $I^+_{\max} = \mathbb{R}_+$. For contradiction, assume there is some time $t_f \in \mathbb{R}_{++}$ with $t_f \notin I^+_{\max}$. By upper bounding the right-hand side of (4.81) by its value at the time t_f, we conclude that $\dot{\psi}$ is bounded on $I^+_{\max}$. Also:

$$\|\psi(t) - q_0\|_2 \leq \int_0^t \|\dot{\psi}(s)\|_2\, \mathrm{d}s \leq t_f\left(\sqrt{\frac{2T(q_0, \dot{q}_0)}{\lambda_{\min}}} + \frac{c_0 t_f}{\lambda_{\min}}\right) e^{\frac{c_1}{\lambda_{\min}}t_f}, \tag{4.82}$$

for all times $t \in I^+_{\max}$; that is, ψ is bounded on $I^+_{\max}$. Therefore, ϕ is bounded on $I^+_{\max}$, contradicting (Perko, 2013, Theorem 2.4.3) (again, the proof of this theorem applies to locally Lipschitz continuous vector fields). This implies $I^+_{\max} = \mathbb{R}_+$, meaning for any sample period $h \in \mathbb{R}_{++}$, there is an open interval $I \subseteq \mathbb{R}$ with $[0, h] \subset I$.

Thus, any robotic system satisfying Assumption 6 also satisfies Assumption 5, no matter which sample period h is chosen.

4.4.3 Sampled-Data Control

We now determine sufficient conditions for the existence of a continuous policy rendering a compact subset of the state space forward-invariant. This construction uses the methods of (Taylor, Dorobantu, Yue, et al., 2021; Taylor, Dorobantu, Cosner, et al., 2022), based on the original work (Nešić, Teel, and Kokotović, 1999) in sampled-data control. These methods are detailed in Appendix A.

A control affine system is full-state feedback linearizable if $\mathcal{D}$ is open and there exist a function $\Phi : \mathcal{D} \to \mathbb{R}^d$ that is a diffeomorphism between $\mathcal{D}$ and an open subset of $\mathbb{R}^d$, a continuous controller $k_{\text{fbl}} : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^m$ that accepts an auxiliary input, and a controllable pair $(A, B) \in \mathbb{R}^{d\times d} \times \mathbb{R}^{d\times m}$ such that:

$$\frac{\partial \Phi}{\partial x} F(x, k_{\text{fbl}}(x, v)) = A\Phi(x) + Bv, \tag{4.83}$$

for all $x \in \mathcal{D}$ and auxiliary control inputs $v \in \mathbb{R}^m$.

Remark 8. Any robotic system satisfying Assumption 6 with $n = m$ and $B(q)$ invertible for all configurations $q \in \mathbb{R}^n$ is full-state feedback linearizable. The feedback linearizing controller $k_{\text{fbl}} : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^n$ is defined as:

$$k_{\text{fbl}}(x, v) = B(q)^{-1}(C(q, \dot{q})\dot{q} + \nabla U(q) - F_{\text{ext}}(q, \dot{q}) + D(q)v), \tag{4.84}$$

for all $x = (q, \dot{q}) \in \mathbb{R}^n \times \mathbb{R}^n$ and $v \in \mathbb{R}^n$, and:

$$F(x, k_{\text{fbl}}(x, v)) = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} v, \tag{4.85}$$

for all $x = (q, \dot{q}) \in \mathbb{R}^n \times \mathbb{R}^n$ and $v \in \mathbb{R}^n$.

Consider a full-state feedback linearizable control affine system, and suppose $0 \in \mathcal{D}$ and $\Phi(0) = 0$. Consider any gain matrix $K \in \mathbb{R}^{m\times d}$ making $A - BK$ Hurwitz stable with all eigenvalues having strictly negative real parts. Suppose the system satisfies Assumption 5 and $0_d \in \text{int}(\mathcal{S})$. Additionally, suppose $\pi_0 : \mathcal{S} \to \mathbb{R}^m$ defined as:

$$\pi_0(x) = k_{\text{fbl}}(x, -K\Phi(x)), \tag{4.86}$$

for all states $x \in \mathcal{S}$ satisfies $\pi_0(x) \in C(x)$ for all states $x \in \mathcal{S}$. Since k_{fbl} and Φ are continuous, π_0 is continuous.

By (Taylor, Dorobantu, Yue, et al., 2021, Lemmas 3, 4), there is a continuous (class $\mathcal{KL}$) function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ with monotonically increasing s-sections β^s for

all $s \in \mathbb{R}_+$ and monotonically nonincreasing r-sections β_r for all $r \in \mathbb{R}_+$, as well as some $\overline{R} \in \mathbb{R}_{++}$ and a bounded open set $N \subset \mathbb{R}^d$ with $0 \in N$ and:

$$\{x \in \mathbb{R}^d : \|x\|_2 \leq \sup_{x' \in N} \beta(\|x'\|_2, 0) + \overline{R}\} \subseteq \mathcal{S}. \tag{4.87}$$

Additionally, for any $R \leq \overline{R}$, if the sample period is sufficiently small, we have:

$$\|(f_h)^t_{\pi_0}(x)\|_2 \leq \beta(\|x\|_2, th) + R \leq \sup_{x' \in N} \beta(\|x'\|_2, 0) + \overline{R}, \tag{4.88}$$

for all $x \in N$ and $t \in \mathbb{Z}_+$. Let $\mathcal{R} \subseteq \mathcal{S}$ denote the set of all states reachable from N when following π_0, defined as $\mathcal{R} = \{(f_h)^t_{\pi_0}(x) \in \mathcal{S} : x \in N, t \in \mathbb{Z}_+\}$, and note that $(f_h)_{\pi_0}(\mathcal{R}) \subseteq \mathcal{R}$ and $\mathrm{cl}(\mathcal{R}) \subseteq \mathcal{S}$. Since $(f_h)_{\pi_0}(\mathrm{cl}(\mathcal{R})) \subseteq \mathrm{cl}(\mathcal{R}) \subseteq \mathcal{S}$, we can set $\mathcal{S}_0 = \mathrm{cl}(\mathcal{R})$. The set $\mathcal{S}_0 \subseteq \mathcal{S}$ is closed and bounded since:

$$\sup_{x \in N} \beta(\|x\|_2, 0) + \overline{R} = \beta\left(\sup_{x \in N} \|x\|_2, 0\right) + \overline{R} < \infty. \tag{4.89}$$

Therefore, $\mathcal{S}_0$ is a compact set rendered forward-invariant by the continuous policy π_0, implying the system satisfies Assumption 4.

Finally, we can construct a compact subset of the state space in a similar manner by appealing to (Taylor, Dorobantu, Cosner, et al., 2022, Theorem 3), employing the notion of practical safety. In this case, if a compact 0-superlevel set of a family of sampled-data control barrier functions (Taylor, Dorobantu, Cosner, et al., 2022, Definition 7) can be rendered forward-invariant when the transition map is approximated by an appropriate Euler/Runge-Kutta scheme, then an enlarged (but bounded) set can be rendered forward-invariant for the exact transition map when the sample period is sufficiently small. The enlargement can be controlled by decreasing the sample period.

Chapter 5

NONLINEAR MODEL PREDICTIVE CONTROL

We introduce an approach for geometric model predictive control, outlining our construction for a 3D hopping robot in a way that facilitates generalization to other systems evolving on Lie groups. The hopping setting is of particular interest as a *hybrid system*, governed by families of ODEs, switching between ODEs at discrete impact events.

5.1 Introduction

Hopping has been a benchmark challenge in the field of robotic locomotion dating back to the seminal work of Marc Raibert in the 1980's (Raibert, Brown, and Chepponis, 1984). The control of hopping robots is particularly challenging due to intermittent continuous and discrete dynamics, periods of extreme underactuation, and exceptionally short ground phases during which the robot can apply forces to regulate its global position. These pose unique difficulties for conventional control algorithms, and necessitate the ability to decide control actions based on predictions of where the robot will be in the future. Besides the work of Raibert, many methods have been developed to stabilize hopping robots (Sayyad, Seth, and Seshu, 2007), including reinforcement learning based approaches (Tedrake and Seung, 2002; Maier, 2001), nonholonomic motion planning (Murray and Sastry, 1990), a mix of offline and online hierarchical motion planning strategies (Zeglin and Brown, 1998; Albro and Bobrow, 2001), and model predictive control of simplified models (Zamani and Bhounsule, 2020; Rutschmann et al., 2012). Compared to the abundance of work that exists for planar hopping robots, the literature for the control of 3D hopping robots is comparatively sparse.

Recently, model predictive control (MPC) (Borrelli, Bemporad, and Morari, 2017) has been used effectively for the control of dynamic robotic systems, including hybrid systems (systems with continuous dynamics and discrete impacts (Grizzle et al., 2014)) like legged robots (Galliker et al., 2022; Farshidian et al., 2017; Di Carlo et al., 2018). MPC has been brought into the realm of real-time dynamic robotic control due to modern computing power and increased algorithmic efficiency (Carpentier et al., 2019); however, the implementation of MPC on hardware platforms, as well as theoretical justifications of its performance for nonlinear systems remains

an active area of research. As nonlinear MPC is predicated on taking local approximations of system dynamics, its success heavily relies on correctly constructing these approximations and remaining within the regions in which they are valid.

When the system states are manifold-valued, these local approximations must be carefully constructed. We will specifically be concerned with Lie groups, groups with smooth manifold structure whose operator and inverse are also smooth (see (Celledoni, Marthinsen, and Owren, 2014; Kobilarov, Crane, and Desbrun, 2009)), as they are often used in the field of robotics to model the space of orientations. Lie groups have additionally been studied from the perspectives of discrete mechanics (Junge, Marsden, and Ober-Blöbaum, 2005) and numerical analysis (Iserles et al., 2000; Hairer et al., 2006). The estimation of robotic systems on Lie groups was outlined in (Solà, Deray, and Atchuthan, 2021), and the control of legged robotic systems on Lie groups has recently been investigated in (Teng et al., 2022). Finally, the application of optimal control techniques over differentiable manifolds to the control of hybrid systems was explored in (Mastalli, Budhiraja, et al., 2020) with experimental results achieved in (Mastalli, Merkt, et al., 2022; Mastalli, Chhatoi, et al., 2023).

We develop a framework for hybrid nonlinear MPC in a geometrically consistent fashion via Lie group integrators. Theoretically, we consider both the manifold structure of the configuration space and the hybrid nature of the dynamics (Sec. 5.2), wherein an MPC problem is synthesized through the use of sequential linearizations that leverage Lie group and Lie algebra structures (Sec. 5.3).

We experimentally demonstrated this framework on a novel 3D robot: ARCHER (Ambrose, 2022) (which builds upon earlier generations of hopping robots (Ambrose, Ma, and Ames, 2021; Ambrose and Ames, 2020; Ambrose, Csomay-Shanklin, et al., 2019)). This robot has three reaction wheels for attitude control, and one motor connected via a rope to control foot spring compression. The MPC problem was translated to hardware via a multi-rate control paradigm. Full experimental details can be found in (Csomay-Shanklin, Dorobantu, and Ames, 2022), with examples of simulated path following and flipping, as well as the first demonstration of 3D hopping on a hardware platform using online motion planning strategies.

5.2 Preliminaries

5.2.1 Hybrid System Dynamics

The configuration of the hopping robot is given by $q = (p, Q, \theta, \ell) \in \mathcal{Q}$, where $p \in \mathbb{R}^3$ is the Cartesian position, $Q \in S^3$ is the unit quaternion representing the orientation, $\theta \in \mathbb{R}^3$ represents the flywheel angles, and $\ell \in \mathbb{R}$ is the foot deflection. Next, let $v = (\dot{p}, \omega, \dot{\theta}, \dot{\ell}) \in \mathcal{V} \triangleq \mathbb{R}^3 \times \mathfrak{s}^3 \times \mathbb{R}^3 \times \mathbb{R}$, where $\omega \in \mathfrak{s}^3$ is a purely imaginary quaternion representing the angular rate of the body. The complete robot state can then be written as $x = (q, v) \in \mathcal{X} \triangleq \mathcal{Q} \times \mathcal{V}$.

Hopping consists of alternating sequences of continuous and discrete dynamics; therefore, it is naturally modeled as a hybrid system. Two distinct continuous phases of dynamics exist, the *flight* phase f when the robot is in the air, and the *ground* phase g when the foot is contacting the floor. We can construct a directed graph with vertices $v \in V \triangleq \{f, g\}$ and edges $e \in E \triangleq \{f \to g, g \to f\}$ to characterize how the robot traverses the hybrid modes, as shown in Figure 5.1.

For each vertex $v \in V$, let $\mathcal{D}_v \subset \mathcal{X}$ represent the admissible domain in which the system state evolves, and n_v denote the number of holonomic constraints restricting the motion of the robot. Note that $n_f = 0$, and $n_g = 3$, which pin the foot to the ground. Omitting the details for manifold value variables (which can be found in (Abraham and Marsden, 2008)), let $J_v : \mathcal{Q} \to \mathbb{R}^{n_v \times n}$ denote the Jacobian of the holonomic constraints where n is the dimension of $\mathcal{Q}$. We can define the dynamics as:

$$\begin{aligned} D(q)\ddot{q} + H(q, \dot{q}) &= Bu + J_v^\top(q)\lambda_v, \\ \dot{J}_v(q, \dot{q})\dot{q} + J_v(q)\ddot{q} &= 0. \end{aligned}$$

where $D : \mathcal{Q} \to \mathbb{R}^{n \times n}$ is the mass-inertia matrix, $H : \mathcal{X} \to \mathbb{R}^n$ contains the Coriolis and gravity terms, $B \in \mathbb{R}^{n \times m}$ is the actuation matrix, $u \in \mathbb{R}^m$ is the control input, and $\lambda_v \in \mathbb{R}^{n_v}$ are the Lagrange variables describing the constraint forces. These equations can be rearranged and forces λ_v solved for in order to produce the constraint implicit dynamics:

$$\dot{x} = f_v(x, u), \tag{5.1}$$

where $f_v : \mathcal{X} \times \mathbb{R}^m \to \mathbb{R}^{2n}$ is of control-affine form. Each hybrid transition $e \in E$ occurs when the system state intersects the *guard*, defined as:

$$\begin{aligned} \mathcal{S}_f &= \{x \in \mathcal{D}_f : z = 0, \dot{z} < 0\} & e &= f \to g, \\ \mathcal{S}_g &= \{x \in \mathcal{D}_g : \ell = 0, \dot{\ell} < 0\} & e &= g \to f. \end{aligned}$$

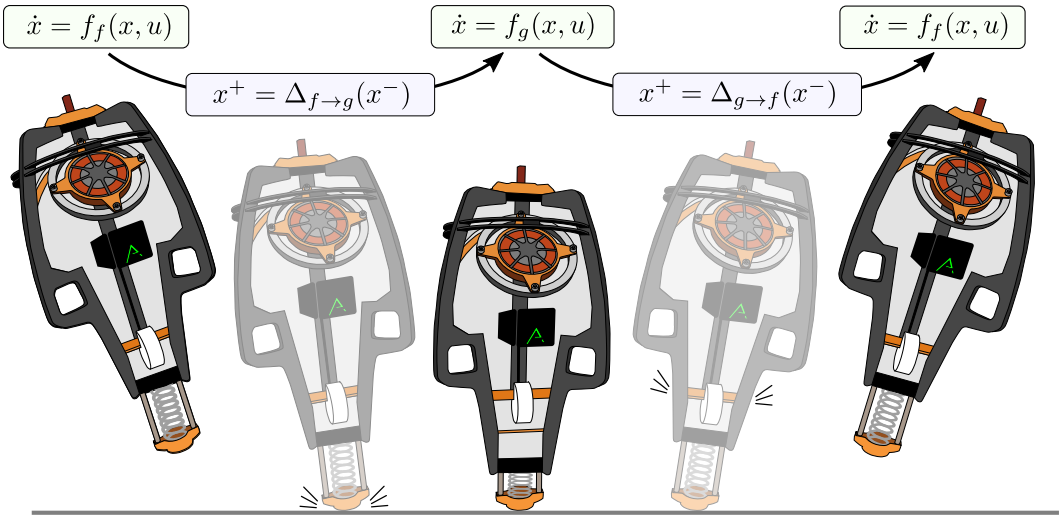


Figure 5.1: The robot traversing the various hybrid domains.

For an edge $e = v_1 \to v_2$, upon striking the guard $\mathcal{S}_{v_1}$ the system undergoes a discrete jump in states as described by:

$$x^+ = \Delta_e(x^-),$$

where $\Delta_e : \mathcal{S}_{v_1} \to \mathcal{D}_{v_2}$ is the *reset map* describing the momentum transfer though impact, and $x^- \in \mathcal{D}_{v_1}$ and $x^+ \in \mathcal{D}_{v_2}$ are the pre and post-impact states, respectively. Collecting the various objects $\mathcal{D} = \{\mathcal{D}_v\}_{v\in V}$, $\mathcal{S} = \{\mathcal{S}_v\}_{v\in V}$, $\Delta = \{\Delta_e\}_{e\in E}$ and $F = \{f_v\}_{v\in V}$, we can describe the hybrid control system of the hopping robot via the tuple:

$$\mathcal{HC} = (V, E, \mathcal{D}, \mathcal{S}, \Delta, F).$$

5.2.2 Lie Group Integrators

In this section, we focus our attention to the orientation coordinates of the robot and discuss how to perform one form of Lie group integration. We represent the orientation of a rigid body via a unit quaternion $Q \in S^3 = \{Q \in \mathbb{H} : |Q| = 1\}$; quaternionic representations of orientation have been extensively explored in attitude control of spacecrafts (Kalabić et al., 2017; Yang, 2012). As opposed to Euler angles, quaternions do not suffer from issues of singularities, and provide a straightforward interpolation method—this will be helpful when constructing continuous-time signals from the discrete points that MPC produces. Importantly, quaternions and their associated quaternion multiplication define a Lie group structure on S^3. The angular rate of the body is given by an element of the associated Lie algebra $\omega \in \mathfrak{s}^3$, the

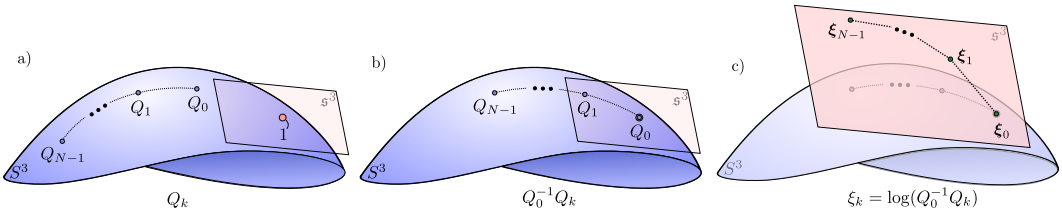


Figure 5.2: A depiction of Lie groups, Lie algebras, and the log operation. a) The trajectory Q_k, b) pulling the trajectory back to the a neighborhood of identity via Q_0^{-1}, and c) taking the log map near identity to obtain elements in the Lie algebra.

tangent space of S^3 at the identity quaternion 1, the elements of which are purely imaginary quaternions.

The time rate of change of a unit quaternion at a point Q is given by an element of the tangent space at Q, i.e., $\dot{Q} \in T_Q S^3$. Given the angular rate of the body ω, we can calculate $\dot{Q}$ as:

$$\dot{Q} = Q\omega, \tag{5.2}$$

using standard quaternion multiplication. The formulation (5.2) is possible since the tangent map of left multiplication by a quaternion is also given by left multiplication, mapping from the Lie algebra to the tangent space at Q.

Integrating equation (5.2) results in:

$$Q(t) = Q(0)\exp(\omega(t)) \tag{5.3}$$

where $\exp : \mathfrak{s}^3 \to S^3$ maps elements of the Lie algebra $\mathfrak{s}^3$ back to the Lie group S^3. This map is injective for imaginary quaternions with magnitude less than π; over this neighborhood, the inverse map is given by the log map. If instead of directly integrating we take a *Lie-Euler step*, (5.3) becomes:

$$Q_{k+1} = Q_k \exp(\omega_k h) \tag{5.4}$$

for a (small) time step $h \in \mathbb{R}_{++}$. This is a simple example of a *Lie group integrator*.

5.3 Geometric MPC

5.3.1 Linearized Dynamics

To mitigate the nonlinearity present in (5.4), we propose the following change of coordinates:

$$\xi_k = \log(Q_0^{-1}Q_k), \tag{5.5}$$

which first pulls the variables back to the vicinity of Q_ε, and then to the Lie algebra as shown in Figure 5.2. Substituting in the Lie-Euler step from (5.4) into the above expression for the first few values of k, we have:

$$\begin{aligned}
\xi_0 &= 0 \\
\xi_1 &= \log(Q_0^{-1} Q_1) = \log(Q_0^{-1} Q_0 \exp(\omega_0 h)) \\
&= \omega_0 h \\
\xi_2 &= \log(Q_0^{-1} Q_2) = \log(Q_0^{-1} Q_1 \exp(\omega_1 h)) \\
&= \log(Q_0^{-1} Q_0 \exp(\omega_0 h) \exp(\omega_1 h)) \\
&= \omega_0 h + \omega_1 h + \frac{1}{2}[\omega_0 h, \omega_1 h] + \cdots
\end{aligned}$$

where $[\cdot, \cdot]$ is the Lie bracket on $\mathfrak{s}^3$, the last line follows from the Campbell-Baker-Hausdorff theorem Stillwell, 2008, and the higher order terms consist of linear combinations of iterated Lie brackets, which due to linearity are multiplied by terms of order h^3 or higher. This means that, neglecting terms of order h^2 or higher, we are able to write our dynamics update law as:

$$\xi_{k+1} = \sum_{i=0}^{k} \omega_k h = \xi_k + \omega_k h,$$

which is linear and will therefore be straightforward to include in the MPC program.

The next challenge concerns the construction of local approximations of the acceleration dynamics. As most of the coordinates lie in Euclidean space and therefore have straightforward Taylor approximations, we will limit our attention on the manifold valued variables. Specifically, in a continuous domain v consider a function $f : S^3 \times \mathfrak{s}^3 \times \mathbb{R}^m \to \mathfrak{s}^3$ satisfying:

$$\dot{\omega} = f(Q, \omega, u).$$

In a neighborhood of Q, we have a chart map given by:

$$Q' \mapsto \log(Q^{-1} Q'). \tag{5.6}$$

In this way, we can locally represent f near (Q, ω, u) using sufficiently small coordinates $(\eta, \Delta\omega, \Delta u)$, where $\eta \in \mathfrak{s}^3$, $\Delta\omega \in \mathfrak{s}^3$ as well, and $\Delta u \in \mathbb{R}^m$. This local representation is then:

$$\tilde{f}(\eta, \Delta\omega, \Delta u) \triangleq f(Q \exp(\eta), \omega + \Delta\omega, u + \Delta u). \tag{5.7}$$

This has the first-order Taylor expansion:

$$\tilde{f}(\eta, \Delta\omega, \Delta u) \approx f(Q, \omega, u) + \left.\frac{\partial \tilde{f}}{\partial \eta}\right|_{(0,0,0)} \cdot \eta + \left.\frac{\partial f}{\partial \omega}\right|_{(Q,\omega,u)} \cdot \Delta\omega + \left.\frac{\partial f}{\partial u}\right|_{(Q,\omega,u)} \cdot \Delta u. \quad (5.8)$$

We can now write the continuous-time linearized dynamics of ω about the point (Q, ω, u) as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\omega = \left.\frac{\partial \tilde{f}}{\partial \eta}\right|_{(0,0,0)} \cdot \eta + \left.\frac{\partial f}{\partial \omega}\right|_{(Q,\omega,u)} \cdot \delta\omega + \left.\frac{\partial f}{\partial u}\right|_{(Q,\omega,u)} \cdot \delta u, \quad (5.9)$$

Next, we consider the dynamics of the variables ξ_k around a reference trajectory $\bar{Q}_k \in S^3$, $\bar{\omega}_k \in \mathfrak{s}^3$, and $\bar{u}_k \in \mathbb{R}^m$. Define $\bar{\xi}_k = \log(\bar{Q}_0^{-1}\bar{Q}_k)$ with $\bar{\xi}_0 = 0$ and suppose the reference trajectory satisfies the Lie-Euler step, i.e.:

$$\bar{\xi}_{k+1} = \bar{\xi}_k + \bar{\omega}_k h.$$

For a trajectory $\xi_k \in \mathfrak{s}^3$ similarly satisfying the Lie-Euler step, and vectors $\omega_k \in \mathfrak{s}^3$, and $u_k \in \mathbb{R}^m$, we have:

$$(\xi_{k+1} - \bar{\xi}_{k+1}) = (\xi_k - \bar{\xi}_k) + (\omega_k - \bar{\omega}_k)h + \underbrace{\bar{\xi}_k + \bar{\omega}_k h - \bar{\xi}_{k+1}}_{=0},$$

yielding continuous-time linearized dynamics:

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\xi = \delta\omega. \quad (5.10)$$

Combining expressions (5.9) and (5.10), we obtain:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\delta\xi \\ \delta\omega\end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ \left.\frac{\partial \tilde{f}}{\partial \eta}\right|_{(0,0,0)} & \left.\frac{\partial f}{\partial \omega}\right|_{(Q,\omega,u)}\end{bmatrix}}_{\triangleq A}\begin{bmatrix}\delta\xi \\ \delta\omega\end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \left.\frac{\partial f}{\partial u}\right|_{(Q,\omega,u)}\end{bmatrix}}_{\triangleq B}\delta u. \quad (5.11)$$

For base trajectories $\bar{\xi}$, $\bar{\omega}$, and $\bar{u}$ and perturbed trajectories ξ, ω and u, our linearization yields:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\begin{bmatrix}\xi \\ \omega\end{bmatrix} - \begin{bmatrix}\bar{\xi} \\ \bar{\omega}\end{bmatrix}\right) \approx A\left(\begin{bmatrix}\xi \\ \omega\end{bmatrix} - \begin{bmatrix}\bar{\xi} \\ \bar{\omega}\end{bmatrix}\right) + B(u - \bar{u}), \quad (5.12)$$

or:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\xi \\ \omega\end{bmatrix} \approx A\begin{bmatrix}\xi \\ \omega\end{bmatrix} + Bu + \underbrace{\tilde{f}(0,0,0) - \begin{bmatrix}\bar{\xi} \\ \bar{\omega}\end{bmatrix} - B\bar{u}}_{\triangleq C}. \quad (5.13)$$

The extra C term is needed since we are not linearizing about an equilibrium.

We can similarly construct local approximations of the impact maps. On an edge e, consider a function $\Delta : \mathcal{S} \to \mathfrak{s}^3$ which satisfies:

$$\omega^+ = \Delta(Q^-, \omega^-). \tag{5.14}$$

Here, $\mathcal{S}$ represents the guard as a submanifold of $S^3 \times \mathfrak{s}^3$ (though the guard is actually a submanifold of $\mathcal{Q}$). As defined, this reset map is the restriction of the momentum transfer of the system at impact to the guard. Therefore, we can naturally extend the domain of the reset map by considering the same momentum transfer applied anywhere in the state space, yielding $\Delta_{\text{ext}} : S^3 \times \mathfrak{s}^3 \to \mathfrak{s}^3$. This is needed because in our Talyor expansion of the discrete dynamics, we consider perturbations of the full system state (not just perturbations tangent to the guard). As before, we can locally represent the function Δ_{ext} as:

$$\tilde{\Delta}_{\text{ext}}(\eta, \Delta\omega^-) \triangleq \Delta_{\text{ext}}(Q^- \exp(\eta), \omega^- + \Delta\omega^-), \tag{5.15}$$

which has first-order Taylor expansion:

$$\tilde{\Delta}_{\text{ext}}(\eta, \Delta\omega^-) \approx \Delta_{\text{ext}}(Q^-, \omega^-) + \left.\frac{\partial \tilde{\Delta}_{\text{ext}}}{\partial \eta}\right|_{(0,0)} \cdot \eta + \left.\frac{\partial \Delta_{\text{ext}}}{\partial \omega^-}\right|_{(Q^-,\omega^-)} \cdot \Delta\omega^-. \tag{5.16}$$

Noting that the $Q^+ = Q^-$, we can represent the linearization of the discrete map as:

$$\begin{bmatrix} \delta\xi^+ \\ \delta\omega^+ \end{bmatrix} = \underbrace{\begin{bmatrix} I & 0 \\ \left.\frac{\partial \tilde{\Delta}_{\text{ext}}}{\partial \eta}\right|_{(0,0)} & \left.\frac{\partial \Delta_{\text{ext}}}{\partial \omega^-}\right|_{(Q^-,\omega^-)} \end{bmatrix}}_{\triangleq D} \begin{bmatrix} \delta\xi^- \\ \delta\omega^- \end{bmatrix}. \tag{5.17}$$

For base pre-impact states $\bar{\xi}^-$ and $\bar{\omega}^-$ and perturbed pre-impact states ξ^- and ω^-, we have:

$$\begin{bmatrix} \xi^+ \\ \omega^+ \end{bmatrix} - \begin{bmatrix} \bar{\xi}^+ \\ \bar{\omega}^+ \end{bmatrix} \approx D \left(\begin{bmatrix} \xi^- \\ \omega^- \end{bmatrix} - \begin{bmatrix} \bar{\xi}^- \\ \bar{\omega}^- \end{bmatrix} \right), \tag{5.18}$$

or:

$$\begin{bmatrix} \xi^+ \\ \omega^+ \end{bmatrix} \approx D \begin{bmatrix} \xi^- \\ \omega^- \end{bmatrix} + \underbrace{\tilde{\Delta}_{\text{ext}}(0,0) - D \begin{bmatrix} \bar{\xi}^- \\ \bar{\omega}^- \end{bmatrix}}_{\triangleq E}. \tag{5.19}$$

5.3.2 Sequential Quadratic Programming

We now detail our solution using sequential quadratic programming (SQP). Planning optimal state and action trajectories directly, ensuring that the states and actions

all satisfy exact discrete-time dynamics, is computationally intractable. However, suppose we have access to a finite-horizon (length N) state trajectory $\bar{x}_0, \ldots, \bar{x}_N$ and input trajectory $\bar{u}_0, \ldots, \bar{u}_{N-1}$. We can formulate a single stage of a sequential optimization procedure by optimizing *perturbations* of these states and inputs, ensuring that the perturbed dynamics satisfy linearized dynamics (that is, agree with the true dynamics up to first order) at each of the pairs $(\bar{x}_k, \bar{u}_k)$, $k = 0, \ldots, N-1$.

At time index k, consider the state $\bar{x}_k$, with:

$$\bar{x}_k = (\bar{p}_k, \bar{Q}_k, \bar{\theta}_k, \bar{\ell}_k, \dot{\bar{p}}_k, \bar{\omega}_k, \dot{\bar{\theta}}_k, \dot{\bar{\ell}}_k). \tag{5.20}$$

Define $\bar{z}_k$ as:

$$\bar{z}_k = (\bar{p}_k, \log(\bar{Q}_0^{-1}\bar{Q}_k), \bar{\theta}_k, \bar{\ell}_k, \dot{\bar{p}}_k, \bar{\omega}_k, \dot{\bar{\theta}}_k, \dot{\bar{\ell}}_k), \tag{5.21}$$

simply replacing $\bar{Q}_k$ with $\bar{\xi}_k \triangleq \log(\bar{Q}_0^{-1}\bar{Q}_k)$.

We decide *a priori* which times will correspond to flight and ground phases. We estimate impact times assuming ballistic trajectories in the vertical direction. If the flight (or ground) phase is selected for the *entire* horizon, we obtain the following quadratic program:

$$\begin{aligned}
\min_{\substack{z_0,\ldots,z_N \\ u_0,\ldots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} (z_k - z_{\text{ref}})^\top S (z_k - z_{\text{ref}}) + u_k^\top R u_k + (z_N - z_{\text{ref}})^\top P (z_N - z_{\text{ref}}) \\
\text{s.t.} \quad & z_{k+1} = A_k z_k + B_k u_k + C_k, \qquad k = 0, \ldots, N-1 \\
& u_k \in \mathcal{U}, \qquad k = 0, \ldots, N-1 \\
& z_0 \text{ given}
\end{aligned} \tag{5.22}$$

where S, R, and Q are positive definite cost matrices penalizing deviation from a reference state z_{ref}, control usage, and terminal deviation from the reference, respectively, and $\mathcal{U}$ is a set specifying control input bounds. The matrices A_k, B_k, and C_k are obtained from discretized linearizations about $(\bar{x}_k, \bar{u}_k)$. The discretization can be based on forward Euler steps or evolution via the matrix exponential (we used the former in experiments). The resulting optimal trajectories $z_0^*, \ldots, z_N^*$ and $u_0^*, \ldots, u_{N-1}^*$ seed the next stage of SQP. Specifically, the corresponding trajectory $x_0^*, \ldots, x_N^*$ is attained by replacing ξ_k^* with $Q_0 \exp(\xi_k^*)$ for each $k = 0, \ldots, N$. We then replace $\bar{x}_k \leftarrow x_k^*$ and $\bar{u}_k \leftarrow u_k^*$, and repeat the optimization.

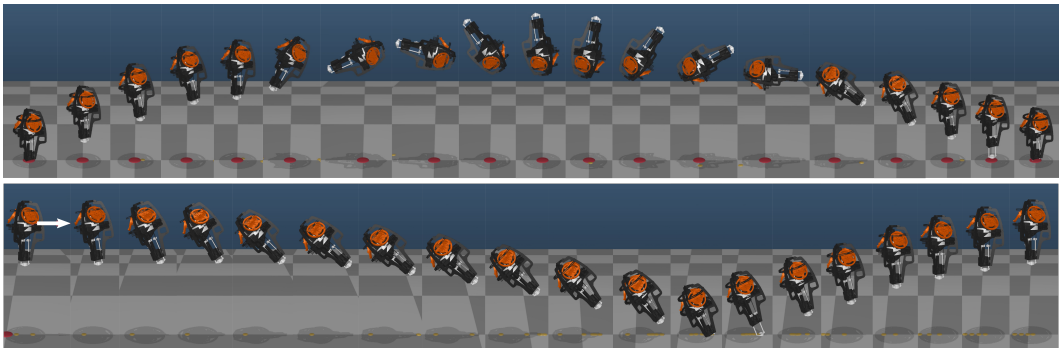


Figure 5.3: Dynamic motions explored in simulation, including flipping (above) and disturbance rejection (below).

If the phase changes during the horizon, we modify the quadratic program as:

$$\begin{aligned}
\min_{\substack{z_0,\dots,z_N \\ u_0,\dots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} (z_k - z_{\text{ref}})^\top S (z_k - z_{\text{ref}}) + u_k^\top R u_k + (z_N - z_{\text{ref}})^\top P (z_N - z_{\text{ref}}) \\
\text{s.t.} \quad & z_{k+1} = \begin{cases} A_k z_k + B_k u_k + C_k & \text{no impact} \\ D_k z_k + E_k & \text{impact} \end{cases} \qquad k = 0, \dots, N-1 \\
& u_k \in \mathcal{U}, \qquad k = 0, \dots, N-1 \\
& z_0 \text{ given}
\end{aligned} \tag{5.23}$$

where *impact* or *no impact* indicate whether of not we expect a transition from time index k to index $k+1$.

After sufficiently many iterations of optimization (this is typically user-specified or based on time constraints), the first action u_0^* is supplied to the actuators. The system undergoes its true transition, likely evolving to a state different than x_1^*. Subsequently, the initial seed of SQP at the next time step is selected as $\bar{x}_k \leftarrow x_{k+1}^*$ and $\bar{u}_k \leftarrow u_{k+1}$ for all $k = 1, \dots, N-1$. The final state $\bar{x}_N$ can be chosen based on a reference state or via linearization at the penultimate state and action. Experimental details are given in (Csomay-Shanklin, Dorobantu, and Ames, 2022); in this case, we used a horizon length of $N = 20$, a time step $h = 0.01$ for flight and $h = 0.001$ for ground phases, and 2 SQP iterations per time step.

Ultimately, this methodology can be adopted for other types of hybrid systems evolving on Lie groups. Possible future directions of work include a closer connection with discrete mechanics and variational integrators, mixed-integer optimization to plan impacts, pulling manifold states back only to the previous state in a trajectory (not all the way to the initial state), and further hardware validation. Of note, despite

pulling each manifold state back to the initial state (which would expectedly cause problems when pulling back almost antipodal points), our method was still able to realize extreme dynamic motions such as flipping (Figure 5.3).

BIBLIOGRAPHY

Abraham, Ralph and Jerrold E. Marsden (2008). *Foundations of Mechanics*. American Mathematical Soc. ISBN: 978-0-8218-4438-0.

Agrawal, Akshay, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter (2019). "Differentiable convex optimization layers." In: *Advances in Neural Information Processing Systems*.

Albro, J. V. and James E. Bobrow (2001). "Optimal motion primitives for a 5 DOF experimental hopper." In: vol. 4, 3630–3635 vol.4. DOI: 10.1109/ROBOT.2001.933181.

Aliprantis, Charalambos D. and Kim C. Border (2006). *Infinite Dimensional Analysis: A Hitchhiker's Guide*. Springer Science & Business Media.

Ambrose, Eric (2022). "Creating ARCHER: A 3d hopping robot with flywheels for attitude control." PhD. Caltech. DOI: 10.7907/gbts-va63.

Ambrose, Eric and Aaron D. Ames (2020). "Improved performance on moving-mass hopping robots with parallel elasticity." In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2457–2463. DOI: 10.1109/ICRA40945.2020.9197070.

Ambrose, Eric, Noel Csomay-Shanklin, Yizhar Or, and Aaron Ames (2019). "Design and comparative analysis of 1d hopping robots." In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5717–5724. DOI: 10.1109/IROS40897.2019.8967692.

Ambrose, Eric, Wen-Loong Ma, and Aaron D. Ames (2021). "Towards the unification of system design and motion synthesis for high-performance hopping robots." In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7073–7078. DOI: 10.1109/ICRA48506.2021.9561322.

Ames, Aaron D., Samuel Coogan, et al. (2019). "Control barrier functions: Theory and applications." In: *European Control Conference (ECC)*. IEEE, pp. 3420–3431.

Ames, Aaron D., Kevin Galloway, et al. (2014). "Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics." In: *Transactions on Automatic Control* 59.4, pp. 876–891.

Ames, Aaron D., Jessy W. Grizzle, and Paulo Tabuada (2014). "Control barrier function based quadratic programs with application to adaptive cruise control." In: *Conference on Decision & Control (CDC)*. IEEE, pp. 6271–6278.

Ames, Aaron D., Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada (2016). "Control barrier function based quadratic programs for safety critical systems." In: *IEEE Transactions on Automatic Control* 62.8, pp. 3861–3876.

Artstein, Zvi (1983). “Stabilization with relaxed controls.” In: *Nonlinear Analysis: Theory, Methods & Applications* 7.11, pp. 1163–1173.

Askoura, Youcef (2008). “About extension of upper semicontinuous multi-valued maps and applications.” In: *arXiv preprint arXiv:0810.3127*.

Aswani, Anil, Humberto Gonzalez, S. Shankar Sastry, and Claire J. Tomlin (2013). “Provably safe and robust learning-based model predictive control.” In: *Automatica* 49.5, pp. 1216–1226.

Baden, Alex, Keenan Crane, and Misha Kazhdan (2018). “Möbius registration.” In: *Computer Graphics Forum*. Vol. 37. 5. Wiley Online Library, pp. 211–220.

Ballard, Patrick (2000). “The dynamics of discrete mechanical systems with perfect unilateral constraints.” In: *Archive for Rational Mechanics and Analysis* 154.3, pp. 199–274.

Beckers, Thomas, Dana Kulić, and Sandra Hirche (2019). “Stable Gaussian process based tracking control of Euler–Lagrange systems.” In: *Automatica* 103, pp. 390–397.

Berkenkamp, Felix, Riccardo Moriconi, Angela P. Schoellig, and Andreas Krause (2016). “Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes.” In: *Conference on Decision & Control (CDC)*. IEEE, pp. 4661–4666.

Berkenkamp, Felix and Angela P. Schoellig (2015). “Safe and robust learning control with Gaussian processes.” In: *European Control Conference (ECC)*. IEEE, pp. 2496–2501.

Berkenkamp, Felix, Matteo Turchetta, Angela P. Schoellig, and Andreas Krause (2017). “Safe model-based reinforcement learning with stability guarantees.” In: *Advances in Neural Information Processing Systems* 30.

Bertsekas, Dimitri (2022). *Abstract Dynamic Programming*. Athena Scientific, Belmont, MA.

Bertsekas, Dimitri and Steven E. Shreve (1996). *Stochastic Optimal Control: The Discrete-Time Case*. Vol. 5. Athena Scientific, Belmont, MA.

Bertsekas, Dimitri P (2011). *Dynamic Programming and Optimal Control 3rd Edition*. Vol. 2. Athena Scientific, Belmont, MA.

– (2019). *Reinforcement Learning and Optimal Control*. Athena Scientific, Belmont, MA.

Billingsley, Patrick (1999). *Convergence of Probability Measures*. John Wiley & Sons, New York, NY.

Blackwell, David, David Freedman, and Michael Orkin (1974). “The optimal reward operator in dynamic programming.” In: *The Annals of Probability* 2.5, pp. 926–941. DOI: 10.1214/aop/1176996558.

Blanchini, Franco (1999). "Set invariance in control." In: *Automatica* 35.11, pp. 1747–1767. DOI: 10.1016/S0005-1098(99)00113-2.

Bobenko, Alexander I., Stefan Sechelmann, and Boris Springborn (2016). "Discrete conformal maps: Boundary value problems, circle domains, Fuchsian and Schottky uniformization." In: *Advances in Discrete Differential Geometry*. Springer, Berlin, Heidelberg, pp. 1–56.

Boffi, Nicholas M., Stephen Tu, Nikolai Matni, Jean-Jacques E. Slotine, and Vikas Sindhwani (2020). "Learning stability certificates from data." In: *arXiv preprint arXiv:2008.05952*.

Borrelli, Francesco, Alberto Bemporad, and Manfred Morari (2017). *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, Cambridge, UK. DOI: 10.1017/9781139061759.

Boyd, Stephen and Lieven Vandenberghe (2004). *Convex Optimization*. Cambridge University Press, Cambridge, UK.

Brahmbhatt, Samarth, Cusuh Ham, Charles C. Kemp, and James Hays (2019). "ContactDB: Analyzing and predicting grasp contact via thermal imaging." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8709–8719.

Brehmer, Johann and Kyle Cranmer (2020). "Flows for simultaneous manifold learning and density estimation." In: *Advances in Neural Information Processing Systems* 33, pp. 442–453.

Brezis, Haim (1973). *Opérateurs Maximaux Monotones et Semi-groupes de Contractions dans les Espaces de Hilbert*. Elsevier.

Calafiore, Giuseppe C. and Laurent El Ghaoui (2014). *Optimization Models*. Cambridge University Press, Cambridge, UK.

Carpentier, Justin, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard (2019). "The Pinocchio C++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives." In: *2019 IEEE/SICE International Symposium on System Integration (SII)*, pp. 614–619. DOI: 10.1109/SII.2019.8700380.

Castañeda, Fernando, Jason J. Choi, Bike Zhang, Claire J. Tomlin, and Koushil Sreenath (2020). "Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects." In: *arXiv preprint arXiv:2011.07183*.

Celledoni, Elena, Håkon Marthinsen, and Brynjulf Owren (2014). "An introduction to Lie group integrators–basics, new developments and applications." In: *Journal of Computational Physics* 257, pp. 1040–1061.

Chen, Minxin, Bin Tu, and Benzhuo Lu (2012). "Triangulated manifold meshing method preserving molecular surface topology." In: *Journal of Molecular Graphics and Modelling* 38, pp. 411–418.

Chen, Ricky T. Q. and Yaron Lipman (2023). "Riemannian flow matching on general geometries." In: *arXiv preprint arXiv:2302.03660*.

Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud (2018). "Neural ordinary differential equations." In: *Advances in Neural Information Processing Systems* 31.

Cheng, Richard, Gábor Orosz, Richard M. Murray, and Joel W. Burdick (2019). "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks." In: *Conference on Artificial Intelligence*. Vol. 33. AAAI, pp. 3387–3395.

Choi, Jason J., Fernando Castañeda, Claire J. Tomlin, and Koushil Sreenath (2020). "Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions." In: *Robotics: Science and Systems*. Corvalis, Oregon, USA. DOI: 10.15607/RSS.2020.XVI.088.

Cohen, Max and Calin Belta (2020). "Approximate optimal control for safety-critical systems with control barrier functions." In: *arXiv preprint arXiv:2008.04122*.

Crane, Keenan (2020). "Conformal geometry of simplicial surfaces." In: *Proceedings of Symposia in Applied Mathematics*. American Mathematical Society.

Csomay-Shanklin, Noel, Victor D. Dorobantu, and Aaron D. Ames (2022). "Nonlinear model predictive control of a 3d hopping robot: Leveraging Lie group integrators for dynamically stable behaviors." In: *arXiv preprint arXiv:2209.11808*.

De Bortoli, Valentin, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet (2022). "Riemannian score-based generative modeling." In: *arXiv preprint arXiv:2202.02763*.

Di Carlo, Jared, Patrick M. Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim (2018). "Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control." In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9. DOI: 10.1109/IROS.2018.8594448.

Dimitrova, Rayna and Rupak Majumdar (2014). "Deductive control synthesis for alternating-time logics." In: *International Conference on Embedded Software (EMSOFT)*. IEEE, pp. 1–10.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2016). "Density estimation using real nvp." In: *arXiv preprint arXiv:1605.08803*.

Dorobantu, Victor D., Kamyar Azizzadenesheli, and Yisong Yue (2023). "Compactly restrictable metric policy optimization problems." In: *IEEE Transactions on Automatic Control* 68.5, pp. 3115–3122. DOI: 10.1109/TAC.2022.3217269.

Dorobantu, Victor D., Charlotte Borcherds, and Yisong Yue (2023). "Conformal generative modeling on triangulated surfaces." In: *arXiv preprint arXiv:2303.10251*.

Dynkin, Evgenii B. and Aleksandr A. Yushkevich (1979). *Controlled Markov Processes*. Vol. 235. Springer, New York, NY.

Farshidian, Farbod, Edo Jelavic, Asutosh Satapathy, Markus Giftthaler, and Jonas Buchli (2017). "Real-time motion planning of legged robots: A model predictive control approach." In: *2017 IEEE-RAS 17th International Humanoids Conference*, pp. 577–584. DOI: 10.1109/HUMANOIDS.2017.8246930.

Feinberg, Eugene A. and Pavlo O. Kasyanov (2021). "MDPs with setwise continuous transition probabilities." In: *Operations Research Letters* 49.5, pp. 734–740.

Feinberg, Eugene A., Pavlo O. Kasyanov, and Nina V. Zadoianchuk (2012). "Average cost Markov decision processes with weakly continuous transition probabilities." In: *Mathematics of Operations Research* 37.4, pp. 591–607.

Feiten, Wendelin, Muriel Lang, and Sandra Hirche (2013). "Rigid motion estimation using mixtures of projected Gaussians." In: *Proceedings of the 16th International Conference on Information Fusion*. IEEE, pp. 1465–1472.

Fisac, Jaime F., Anayo K. Akametalu, Melanie N. Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J. Tomlin (2018). "A general safety framework for learning-based control in uncertain robotic systems." In: *Transactions on Automatic Control*.

Galliker, Manuel Y., Noel Csomay-Shanklin, Ruben Grandia, Andrew J. Taylor, Farbod Farshidian, Marco Hutter, and Aaron D. Ames (2022). "Bipedal locomotion with nonlinear model predictive control: Online gait generation using whole-body dynamics." In: *arXiv preprint arXiv:2203.07429*.

Gao, Angela, Jorge Castellanos, Yisong Yue, Zachary Ross, and Katherine Bouman (2021). "DeepGEM: Generalized expectation-maximization for blind inversion." In: *Advances in Neural Information Processing Systems* 34, pp. 11592–11603.

Gerber, Samuel, Tolga Tasdizen, P Thomas Fletcher, Sarang Joshi, Ross Whitaker, and Alzheimers Disease Neuroimaging Initiative (ADNI) (2010). "Manifold modeling for brain population analysis." In: *Medical Image Analysis* 14.5, pp. 643–653.

Gillespie, Mark, Boris Springborn, and Keenan Crane (2021). "Discrete conformal equivalence of polyhedral surfaces." In: *ACM Transactions on Graphics* 40.4.

Gillula, Jeremy H. and Claire J. Tomlin (2012). "Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2723–2730.

Grizzle, Jessy W., Christine Chevallereau, Ryan W. Sinnet, and Aaron D. Ames (2014). "Models, feedback control, and open problems of 3d bipedal robotic walking." In: *Automatica* 50.8, pp. 1955–1988. DOI: 10.1016/j.automatica.2014.04.021.

Hairer, Ernst, Marlis Hochbruck, Arieh Iserles, and Christian Lubich (2006). "Geometric numerical integration." In: *Oberwolfach Reports* 3, pp. 805–882.

Huang, Chin-Wei et al. (2022). "Riemannian diffusion models." In: *Advances in Neural Information Processing Systems* 35, pp. 2750–2761.

Ioannou, Petros A. and C. C. Chien (1993). "Autonomous intelligent cruise control." In: *IEEE Transactions on Vehicular technology* 42.4, pp. 657–672.

Iserles, Arieh, Hans Z. Munthe-Kaas, Syvert P. Nørsett, and Antonella Zanna (2000). "Lie-group methods." In: *Acta Numerica* 9, pp. 215–365.

Jankovic, Mrdjan (2018). "Robust control barrier functions for constrained stabilization of nonlinear systems." In: *Automatica* 96, pp. 359–367.

Junge, Oliver, Jerrold E. Marsden, and Sina Ober-Blöbaum (2005). "Discrete mechanics and optimal control." In: *IFAC Proceedings Volumes* 38.1, pp. 538–543.

K, M. Reza. (n.d.). *Tietze extension theorem for lower semi continuous functions.* MathOverflow. URL: https://mathoverflow.net/q/295497.

Kalabić, Uroš V., Rohit Gupta, Stefano Di Cairano, Anthony M. Bloch, and Ilya V. Kolmanovsky (2017). "MPC on manifolds with an application to the control of spacecraft attitude on SO(3)." In: *Automatica*, pp. 293–300. DOI: 10.1016/j.automatica.2016.10.022.

Katsman, Isay, Aaron Lou, Derek Lim, Qingxuan Jiang, Ser Nam Lim, and Christopher M De Sa (2021). "Equivariant manifold flows." In: *Advances in Neural Information Processing Systems* 34, pp. 10600–10612.

Kazhdan, Michael, Matthew Bolitho, and Hugues Hoppe (2006). "Poisson surface reconstruction." In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. Vol. 7.

Kechris, Alexander (2012). *Classical Descriptive Set Theory*. Vol. 156. Springer Science & Business Media.

Khalil, Hassan K. (2002). *Nonlinear Systems*. Vol. 3. Prentice Hall, Upper Saddle River, NJ.

Khansari-Zadeh, Seyed M and Aude Billard (2014). "Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions." In: *Robotics and Autonomous Systems* 62.6.

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980*.

Kober, Jens, J. Andrew Bagnell, and Jan Peters (2013). "Reinforcement learning in robotics: A survey." In: *The International Journal of Robotics Research* 32.11, pp. 1238–1274.

Kobilarov, Marin, Keenan Crane, and Mathieu Desbrun (2009). "Lie group integrators for animation and control of vehicles." In: *ACM transactions on Graphics (TOG)* 28.2, pp. 1–14.

Kostelec, Peter J. and Daniel N. Rockmore (2008). "FFTs on the rotation group." In: *Journal of Fourier Analysis and Applications* 14.2, pp. 145–179.

Kuss, Malte and Carl Rasmussen (2003). "Gaussian processes in reinforcement learning." In: *Advances in Neural Information Processing Systems* 16.

LaValle, Steven M. (2006). *Planning Algorithms*. Cambridge University Press, Cambridge, UK.

Lederer, Armin, Alexandre Capone, and Sandra Hirche (2020). "Parameter optimization for learning-based control of control-affine systems." In: *Proceedings of Machine Learning Research (PMLR)* 120, pp. 465–475.

Lederer, Armin, Alexandre Capone, Jonas Umlauft, and Sandra Hirche (2020). "How training data impacts performance in learning-based control." In: *arXiv preprint arXiv:2005.12062*.

Lee, Joonho, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter (2020). "Learning quadrupedal locomotion over challenging terrain." In: *Science Robotics* 5.47.

Levine, Sergey, Chelsea Finn, Trevor Darrell, and Pieter Abbeel (2016). "End-to-end training of deep visuomotor policies." In: *Journal of Machine Learning Research (JMLR)* 17.1, pp. 1334–1373.

Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra (2015). "Continuous control with deep reinforcement learning." In: *arXiv preprint arXiv:1509.02971*.

Lou, Aaron, Derek Lim, Isay Katsman, Leo Huang, Qingxuan Jiang, Ser Nam Lim, and Christopher M. De Sa (2020). "Neural manifold ordinary differential equations." In: *Advances in Neural Information Processing Systems* 33, pp. 17548–17558.

Maier, Klaus D. (2001). "Neural network based control of legged hopping systems." In: *Proceeding of the 2001 IEEE International Symposium on Intelligent Control (ISIC)*, pp. 115–120. DOI: 10.1109/ISIC.2001.971494.

Mastalli, Carlos, Rohan Budhiraja, Wolfgang Merkt, Guilhem Saurel, Bilal Hammoud, Maximilien Naveau, Justin Carpentier, Ludovic Righetti, Sethu Vijayakumar, and Nicolas Mansard (2020). "Crocoddyl: An efficient and versatile framework for multi-contact optimal control." In: *IEEE International Conference on Robotics and Automation (ICRA)*.

Mastalli, Carlos, Saroj Prasad Chhatoi, Thomas Corbères, Steve Tonneau, and Sethu Vijayakumar (2023). "Inverse-dynamics MPC via nullspace resolution." In: *arXiv preprint arXiv:2209.05375*.

Mastalli, Carlos, Wolfgang Merkt, Josep Marti-Saumell, Henrique Ferrolho, Joan Solà, Nicolas Mansard, and Sethu Vijayakumar (2022). "A feasibility-driven approach to control-limited DDP." In: *Autonomous Robots* 46.8, pp. 985–1005. DOI: 10.1007/s10514-022-10061-w.

Mathieu, Emile and Maximilian Nickel (2020). "Riemannian continuous normalizing flows." In: *Advances in Neural Information Processing Systems* 33, pp. 2503–2515.

Murray, Richard M., Zexiang Li, and S. Shankar Sastry (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL.

Murray, Richard M. and S. Shankar Sastry (1990). "Steering nonholonomic systems using sinusoids." In: *29th IEEE Conference on Decision and Control*, 2097–2101 vol.4. DOI: 10.1109/CDC.1990.203994.

Nešić, Dragan, Andrew R. Teel, and Petar V. Kokotović (1999). "Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations." In: *Systems & Control Letters* 38.4-5, pp. 259–270.

Nguyen, Quan, Ayonga Hereid, Jessy W. Grizzle, Aaron D. Ames, and Koushil Sreenath (2016). "3d dynamic walking on stepping stones with control barrier functions." In: *Conference on Decision and Control (CDC)*. IEEE, pp. 827–834.

Olfati-Saber, Reza (2001). "Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles." PhD thesis. Massachusetts Institute of Technology.

Papamakarios, George, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan (2021). "Normalizing flows for probabilistic modeling and inference." In: *Journal of Machine Learning Research (JMLR)* 22.57, pp. 1–64.

Perko, Lawrence (2013). *Differential Equations and Dynamical Systems*. Vol. 7. Springer Science & Business Media.

Pickem, Daniel, Paul Glotfelter, Li Wang, Mark Mote, Aaron D. Ames, Eric Feron, and Magnus Egerstedt (2017). "The robotarium: A remotely accessible swarm robotics research testbed." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1699–1706.

Prajna, Stephen (2005). "Optimization-based methods for nonlinear and hybrid systems verification." PhD thesis. California Institute of Technology.

Prajna, Stephen and Ali Jadbabaie (2004). "Safety verification of hybrid systems using barrier certificates." In: *International Workshop on Hybrid Systems: Computation and Control (HSCC)*. Springer, pp. 477–492.

Puterman, Martin L. (2014). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, NY.

Qu, Guannan, Chenkai Yu, Steven Low, and Adam Wierman (2020). "Combining model-based and model-free methods for nonlinear control: A provably convergent policy gradient approach." In: *arXiv preprint arXiv:2006.07476*.

Raibert, Marc H., H. Benjamin Brown, and Michael Chepponis (1984). "Experiments in balance with a 3d one-legged hopping machine." In: *The International Journal of Robotics Research* 3.2, pp. 75–92. DOI: 10.1177/027836498400300207.

Rezende, Danilo and Shakir Mohamed (2015). "Variational inference with normalizing flows." In: *International Conference on Machine Learning (ICML)*. PMLR, pp. 1530–1538.

Rosolia, Ugo and Francesco Borrelli (2017). "Learning model predictive control for iterative tasks. a data-driven control framework." In: *IEEE Transactions on Automatic Control* 63.7, pp. 1883–1896.

Ross, Brendan and Jesse Cresswell (2021). "Tractable density estimation on learned manifolds with conformal embedding flows." In: *Advances in Neural Information Processing Systems* 34, pp. 26635–26648.

Rozen, Noam, Aditya Grover, Maximilian Nickel, and Yaron Lipman (2021). "Moser flow: Divergence-based generative modeling on manifolds." In: *Advances in Neural Information Processing Systems* 34, pp. 17669–17680.

Rudin, Walter (1976). *Principles of Mathematical Analysis*. Vol. 3. McGraw-Hill, New York, NY.

Rutschmann, Martin, Brian Satzinger, Marten Byl, and Katie Byl (2012). "Nonlinear model predictive control for rough-terrain robot hopping." In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1859–1864. DOI: 10.1109/IROS.2012.6385865.

Särkkä, Simo and Arno Solin (2019). *Applied Stochastic Differential Equations*. Vol. 10. Cambridge University Press, Cambridge, UK.

Sayyad, Ajij, B. Seth, and P. Seshu (2007). "Single-legged hopping robotics research—A review." In: *Robotica* 25.5, pp. 587–613. DOI: 10.1017/S0263574707003487.

Shafer, Glenn and Vladimir Vovk (2008). "A tutorial on conformal prediction." In: *Journal of Machine Learning Research (JMLR)* 9.3, pp. 371–421.

Shapovalov, Maxim V. and Roland L. Dunbrack Jr. (2011). "A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions." In: *Structure* 19.6, pp. 844–858.

Shi, Guanya, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung (2019). "Neural lander: Stable drone landing control using learned dynamics." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 9784–9790.

Silver, David, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller (2014). "Deterministic policy gradient algorithms." In: *International Conference on Machine Learning (ICML)*, pp. 387–395.

Sohl-Dickstein, Jascha, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli (2015). "Deep unsupervised learning using nonequilibrium thermodynamics." In: *International Conference on Machine Learning (ICML)*, pp. 2256–2265.

Solà, Joan, Jeremie Deray, and Dinesh Atchuthan (2021). "A micro Lie theory for state estimation in robotics." In: *arXiv preprint arXiv:1812.01537*.

Song, Jiaming, Lantao Yu, Willie Neiswanger, and Stefano Ermon (2022). "A general recipe for likelihood-free Bayesian optimization." In: *International Conference on Machine Learning (ICML)*. PMLR, pp. 20384–20404.

Song, Yang, Liyue Shen, Lei Xing, and Stefano Ermon (2021). "Solving inverse problems in medical imaging with score-based generative models." In: *arXiv preprint arXiv:2111.08005*.

Song, Yang, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole (2020). "Score-based generative modeling through stochastic differential equations." In: *International Conference on Learning Representations*.

Sontag, Eduardo D. (1989a). "A 'universal' construction of Artstein's theorem on nonlinear stabilization." In: *Systems & Control Letters* 13.2, pp. 117–123.

– (1989b). "Smooth stabilization implies coprime factorization." In: *IEEE Transactions on Automatic Control* 34.4, pp. 435–443. DOI: 10.1109/9.28018.

Springborn, Boris, Peter Schröder, and Ulrich Pinkall (2008). "Conformal equivalence of triangle meshes." In: *ACM SIGGRAPH 2008 papers*, pp. 1–11.

Stillwell, John (2008). "The matrix logarithm." In: *Naive Lie Theory*. Ed. by John Stillwell. Undergraduate Texts in Mathematics. New York, NY: Springer, pp. 139–159. DOI: 10.1007/978-0-387-78214-0_7.

Sutton, Richard S. and Andrew G. Barto (2018). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Taylor, Andrew J., Victor D. Dorobantu, Ryan K. Cosner, Yisong Yue, and Aaron D. Ames (2022). "Safety of sampled-data systems with control barrier functions via approximate discrete time models." In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, pp. 7127–7134.

Taylor, Andrew J., Victor D. Dorobantu, Sarah Dean, Benjamin Recht, Yisong Yue, and Aaron D. Ames (2021). "Towards robust data-driven control synthesis for nonlinear systems with actuation uncertainty." In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 6469–6476. DOI: 10.1109/CDC45484.2021.9683511.

Taylor, Andrew J., Victor D. Dorobantu, Hoang M. Le, Yisong Yue, and Aaron D. Ames (2019). "Episodic learning with control Lyapunov functions for uncertain robotic systems." In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 6878–6884.

Taylor, Andrew J., Victor D. Dorobantu, Yisong Yue, Paulo Tabuada, and Aaron D. Ames (2021). "Sampled-data stabilization with control Lyapunov functions via quadratically constrained quadratic programs." In: *IEEE Control Systems Letters* 6, pp. 680–685.

Taylor, Andrew J., Andrew Singletary, Yisong Yue, and Aaron D. Ames (2020). "Learning for safety-critical control with control barrier functions." In: *Proceedings of Machine Learning Research (PMLR)* 120, pp. 708–717.

Tedrake, Russ and H. Sebastian Seung (2002). "Improved dynamic stability using reinforcement learning." In: *International Conference on Climbing and Walking Robots (CLAWAR)*, pp. 341–348.

Teng, Sangli, Dianhao Chen, William Clark, and Maani Ghaffari (2022). "An error-state model predictive control on connected matrix Lie groups for legged robot control." In: *arXiv preprint arXiv:2203.08728*.

Tu, Stephen L. (2019). "Sample complexity bounds for the linear quadratic regulator." PhD thesis. UC Berkeley.

Umlauft, Jonas, Lukas Pöhler, and Sandra Hirche (2018). "An uncertainty-based control Lyapunov approach for control-affine Systems modeled by Gaussian process." In: *Control Systems Letters* 2.3, pp. 483–488. DOI: 10.1109/LCSYS.2018.2841961.

Wang, Sen, Yang Wang, Miao Jin, Xianfeng Gu, and Dimitris Samaras (2006). "3D surface matching and recognition using conformal geometry." In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE, pp. 2453–2460.

Yang, Yaguang (2012). "Spacecraft attitude determination and control: Quaternion based method." In: *Annual Reviews in Control* 36, pp. 198–219. DOI: 10.1016/j.arcontrol.2012.09.003.

Yu, Huizhen and Dimitri P Bertsekas (2015). "A mixed value and policy iteration method for stochastic control with universally measurable policies." In: *Mathematics of Operations Research* 40.4, pp. 926–968.

Zamani, Ali and Pranav A. Bhounsule (2020). "Nonlinear model predictive control of hopping model using approximate step-to-step models for navigation on complex terrain." In: pp. 3627–3632. DOI: 10.1109/IROS45743.2020.9340791.

Zeglin, Garth and Ben Brown (1998). "Control of a bow leg hopping robot." In: *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*. Vol. 1. IEEE, pp. 793–798. DOI: 10.1109/ROBOT.1998.677082.

Zheng, Lei, Jiesen Pan, Rui Yang, Hui Cheng, and Haifeng Hu (2020). "Learning-based safety-stability-driven control for safety-critical systems under model uncertainties." In: *arXiv preprint arXiv:2008.03421*.

Appendix A

SAMPLED-DATA CONTROL

We now cover the theoretical foundations of our approach to *sampled-data control* (Taylor, Dorobantu, Yue, et al., 2021; Taylor, Dorobantu, Cosner, et al., 2022), in which we synthesize discrete-time controllers for continuous-time systems. Specifically, we assume that control inputs must be held constant over fixed-frequency sample periods. Moreover, we assume that sensors can only measure system states at a fixed frequency; for simplicity, we assume that these frequencies are the same. The resulting discrete-time evolution arises from sensing the system state, choosing a control input, holding the input constant while the state evolves under the flow of the governing ODE, and sensing the terminal state at the end of the sample period.

Precisely, consider a nonlinear control system governed by the differential equation:

$$\dot{x} = f(x) + g(x)u, \tag{A.1}$$

for state signal x and control input signal u taking values in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, drift dynamics $f : \mathbb{R}^n \to \mathbb{R}^n$, and actuation matrix function $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$. Consider an open subset $\mathcal{Z} \subseteq \mathbb{R}^n \times \mathbb{R}^m$ and its projection onto the state space $\mathcal{X} \triangleq \{x \in \mathbb{R}^n \ : \ \exists\, u \in \mathbb{R}^m \text{ s.t. } (x,u) \in \mathcal{Z}\} \subseteq \mathbb{R}^n$. Assume there exists $T_{\max} \in \mathbb{R}_{++}$ such that for every state-input pair $(x,u) \in \mathcal{Z}$, there exists a unique solution $\gamma : [0, T_{\max}] \to \mathbb{R}^n$ satisfying:

$$\dot{\gamma}(t) = f(\gamma(t)) + g(\gamma(t))u, \qquad \gamma(0) = x, \tag{A.2}$$

for all $t \in (0, T_{\max})$. Given an $h \in (0, T_{\max}]$, a controller $k : \mathcal{X} \to \mathbb{R}^m$ is *h-admissible* if for any $x \in \mathcal{X}$, the state-input pair $(x, k(x))$ satisfies $(x, k(x)) \in \mathcal{Z}$ and the corresponding solution γ satisfies $\gamma(t) \in \mathcal{X}$ for all $t \in [0, h]$.

A.1 Families of Controller-Map Pairs

Our sampled-data approach uses families of discrete-time dynamics rules and controllers, each adapted for specific sample periods. Denote the set of possible sample periods by $I = (0, T_{\max}]$. Given a sample period $h \in I$ and an h-admissible controller $k : \mathcal{X} \to \mathbb{R}^m$, the state and control input signals in (A.1) satisfy:

$$u(t) = k(x(t_i)) \quad \forall t \in [t_i, t_{k+i}), \tag{A.3}$$

with sample times satisfying $t_{i+1} - t_i = h$ for all $i \in \mathbb{Z}_+$. The evolution of the system over a sample period is given by the *exact map* $F_h^e : \mathcal{Z} \to \mathbb{R}^n$:

$$F_h^e(x,u) = x + \int_0^h (f(\gamma(\tau)) + g(\gamma(\tau))u) \ \mathrm{d}\tau, \tag{A.4}$$

for all state-input pairs $(x,u) \in \mathcal{Z}$. We call $\{(k_h : \mathcal{X} \to \mathbb{R}^m) \ : \ h \in I\}$ a *family of admissible controllers* if there is an $h^* \in I$ such that for each $h \in (0, h^*)$, k_h is h-admissible. This enables the following definition:

Definition 5 (*Exact Family*)**.** We define the *exact family of maps* $\{F_h^e \ : \ h \in I\}$, and for a family of admissible controllers $\{k_h \ : \ h \in I\}$, we define the *exact family of controller-map pairs* $\{(k_h, F_h^e) \ : \ h \in I\}$.

For all $h \in I$ such that k_h is h-admissible, the recursion $x_{k+1} = F_h^e(x_k, k_h(x_k)) \in \mathcal{X}$ is well-defined for all $x_0 \in \mathcal{X}$ and $k \in \mathbb{Z}_+$. Closed-form expressions for the exact family of maps are rarely obtainable, suggesting the use of approximations in control synthesis. While there are many approaches to approximating this family of maps, we will use the following common class of approximations:

Definition 6 (*Euler Approximation Family*)**.** For every sample period $h \in I$, define the map $F_h^a : \mathcal{Z} \to \mathbb{R}^n$ as:

$$F_h^a(x_0,u_0) = x_0 + h(f(x_0) + g(x_0)u_0), \tag{A.5}$$

for all $(x_0,u_0) \in \mathcal{Z}$. For a family of admissible controllers $\{k_h : \mathcal{X} \to \mathbb{R}^m \ : \ h \in I\}$, the corresponding *Euler approximation family* of controller-map pairs is:

$$\{(k_h, F_h^a) \ : \ h \in I\}. \tag{A.6}$$

We also consider the following higher-order class of approximations:

Definition 7 (*Runge-Kutta Approximation Family*)**.** Let $p \in \mathbb{N}$. We define the *Runge-Kutta approximation family of maps* $\{F_h^{a,p} \ : \ h \in I\}$, where for every sample period $h \in I$, define $F_h^{a,p} : \mathcal{Z} \to \mathbb{R}^n$ recursively as:

$$F_h^{a,p}(x,u) = x + h \sum_{i=1}^{p} b_i(f(z_i) + g(z_i)u), \tag{A.7}$$

$$z_i = x + h \sum_{j=1}^{i-1} a_{i,j}(f(z_j) + g(z_j)u), \tag{A.8}$$

for all pairs $(x, u) \in \mathcal{Z}$, with $z_1 = x$. Here, $b_1, \ldots, b_p \in \mathbb{R}_+$ satisfy $\sum_{i=1}^{p} b_i = 1$ and $a_{i,j} \in \mathbb{R}$ for each $i \in \{1, \ldots p\}$ and $j \in \{1, \ldots, i-1\}$. For a family of admissible controllers $\{k_h \ : \ h \in I\}$, we may define the *Runge-Kutta approximation family of controller-map pairs* $\{(k_h, F_h^{a,p}) \ : \ h \in I\}$.

Note that the Euler approximation family corresponds to the unique Runge-Kutta approximation family when $p = 1$. Note further that for our work, there are no requirements on the coefficients $a_{i,j}$ (additional requirements on Runge-Kutta coefficients are typically needed for higher-order accuracy ODE solves). However, in practice, we use a standard midpoint rule scheme when needed for robotic systems, with:

$$z_1 = x, \tag{A.9}$$

$$z_2 = x + \frac{h}{2}(f(z_1) + g(z_1)u) = x + \frac{h}{2}(f(x) + g(x)u), \tag{A.10}$$

$$\begin{aligned} F_h^{a,p}(x, u) &= x + h(f(z_2) + g(z_2)u) \\ &= x + h\left(f\left(x + \frac{h}{2}(f(x) + g(x)u)\right) + g\left(x + \frac{h}{2}(f(x) + g(x)u)\right)u\right), \end{aligned} \tag{A.11}$$

for all pairs $(x, u) \in \mathcal{Z}$. When the system in question is governed by:

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu, \tag{A.12}$$

with:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \underbrace{\begin{bmatrix} \dot{q} \\ -D(q)^{-1}H(q, \dot{q}) \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ D(q)^{-1}B \end{bmatrix}}_{g(x)} u, \tag{A.13}$$

we obtain:

$$z_1 = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}, \tag{A.14}$$

$$z_2 = \begin{bmatrix} q + \frac{h}{2}\dot{q} \\ \dot{q} + \frac{h}{2}D(q)^{-1}(Bu - H(q, \dot{q})) \end{bmatrix}, \tag{A.15}$$

$$\begin{aligned} &F_h^{a,p}(x, u) \\ &\quad = \begin{bmatrix} q + h\dot{q} + \frac{h^2}{2}D(q)^{-1}(Bu - H(q, \dot{q})) \\ \dot{q} + hD\left(q + \frac{h}{2}\dot{q}\right)^{-1}\left(Bu - H\left(q + \frac{h}{2}\dot{q}, \dot{q} + \frac{h}{2}D(q)^{-1}(Bu - H(q, \dot{q}))\right)\right) \end{bmatrix} \\ &\quad = \begin{bmatrix} q + h\dot{q} + \frac{h^2}{2}\ddot{q} \\ \cdots \end{bmatrix}. \end{aligned} \tag{A.16}$$

Note that the first block of $F_h^{a,p}(x,u)$ is simply the second-order Taylor approximation of the configuration q at the next time step.

The following definition characterizes how accurately an approximate map captures the exact map:

Definition 8 (*One-Step Consistency*)**.** A family $\{(k_h, F_h) : h \in I\}$ is *one-step consistent* with $\{(k_h, F_h^e) : h \in I\}$ over a set $A \subseteq \mathcal{X}$ if there exist $\rho \in \mathcal{K}_\infty$ and $h^* \in I$ such that for all $x \in A$ and $h \in (0, h^*)$, we have:

$$\|F_h^e(x, k_h(x)) - F_h(x, k_h(x))\| \leq h\rho(h). \tag{A.17}$$

We simply say *one-step consistent* if this is true over every *compact* subset of $\mathcal{X}$, where ρ and h^* are allowed to vary for each compact set.

To demonstrate a condition under which one-step consistency is attained, we first need a short lemma (we will refer to this lemma again on occasion).

Lemma 4. *For any compact set $K \subset \mathcal{X}$, there is an $\varepsilon \in \mathbb{R}_{++}$ such that $K + \overline{B}_\varepsilon \subset \mathcal{X}$* [1] *and $K + \overline{B}_\varepsilon$ is compact, where $\overline{B}_\varepsilon$ is the closed norm-ball of radius ε.*

Proof. As $\mathcal{X}$ is open, for every $x \in K$, there is a corresponding open ball centered at x with radius $\delta_x \in \mathbb{R}_{++}$ that is contained in $\mathcal{X}$. Let $B_x \subset \mathcal{X}$ be the open ball centered at x of radius $\delta_x/2$. Consider the collection $\{B_x : x \in K\}$; this is an open cover for the compact set K, so some finite collection $B_{x_1}, \ldots, B_{x_N}$ for some $x_1, \ldots, x_N \in K$, respectively, also covers K. Let $\delta = \min_i \delta_{x_i}$, and consider any $z \in K + \overline{B}_{\delta/4}$. There is some $x \in K$ such that $\|z - x\| \leq \delta/4$ and some $i \in \{1, \ldots, N\}$ such that $\|x - x_i\| < \delta_{x_i}/2$. Thus, $\|z - x_i\| < \delta/4 + \delta_{x_i}/2 < \delta_{x_i}$, so $z \in \mathcal{X}$. As z was arbitrary, $K + \overline{B}_{\delta/4} \subseteq \mathcal{X}$, so pick $\varepsilon \leq \delta/4$. The set $K + \overline{B}_\varepsilon$ is compact as $K \times \overline{B}_\varepsilon$ is compact and $(x, y) \mapsto x + y$ is continuous. □

Theorem 3. *Suppose f and g are locally Lipschitz continuous over $\mathcal{X}$. Let $K \subset \mathcal{X}$ be compact, consider a family of admissible controllers $\{k_h : h \in I\}$, and suppose there exists $h_1 \in I$ and a bound $M_K \in \mathbb{R}_+$ such that for every sample period $h \in (0, h_1)$, the controller k_h is bounded in norm by M_K over K. Then the family $\{(k_h, F_h^{a,p}) : h \in I\}$ is one-step consistent with $\{(k_h, F_h^e) : h \in I\}$ over K.*

Proof. Consider a compact set $K \subset \mathcal{X}$ and corresponding $h_1 \in I$ and $M_K \in \mathbb{R}_{++}$, and fix a sample period $h \in (0, h_1)$. By Lemma 4, there exists an $\varepsilon \in \mathbb{R}_{++}$ such that

[1] Here, + denotes the Minkowski sum.

the compact set $N = K + \overline{B}_\varepsilon$ satisfies $N \subset \mathcal{X}$. By assumption, k_h is bounded on K, and f and g are bounded on N by continuity, implying there exists an $M \in \mathbb{R}_{++}$ such that:

$$\|f(z) + g(z)k_h(y)\| \leq M, \tag{A.18}$$

for all $y \in K$ and $z \in N$. As f and g are locally Lipschitz over $\mathcal{X}$, they are globally Lipschitz over N. Therefore:

$$\begin{aligned} &\|f(z) + g(z)k_h(y) - (f(y) + g(y)k_h(y))\| \\ &\quad \leq \|f(z) - f(y)\| + \|g(z) - g(y)\|\|k_h(y)\| \\ &\quad \leq (L_f + L_g M_K)\|z - y\| = \rho(\|z - y\|), \end{aligned} \tag{A.19}$$

for all $y \in K$ and $z \in N$, with $L_f, L_g \in \mathbb{R}_{++}$ Lipschitz constants of f and g, respectively, and $\rho \in \mathcal{K}_\infty$ satisfying $\rho(r) = (L_f + L_g M_K)r$ for all $r \in \mathbb{R}_+$. Let $x \in K$. Then:

$$\begin{aligned} &F_h^e(x, k_h(x)) - F_h^{a,p}(x, k_h(x)) \\ &\quad = \int_0^h (f(\gamma(t)) + g(\gamma(t))k_h(x))\ \mathrm{d}t - h\sum_{i=1}^p b_i(f(z_i) + g(z_i)k_h(x)) \\ &\quad = \int_0^h (f(\gamma(t)) + g(\gamma(t))k_h(x) - (f(x) + g(x)k_h(x)))\,\mathrm{d}t \\ &\qquad + h\sum_{i=1}^p b_i\left(f(x) + g(x)k_h(x) - (f(z_i) + g(z_i)k_h(x))\right), \end{aligned} \tag{A.20}$$

where we make use of the fact $\sum_{i=1}^p b_i = 1$.

To bound the first term in (A.20), let $h_2 \in (0, h_1)$ satisfy $h_2 < \varepsilon/M$. By continuity, if $\gamma(t_0) \notin N$ for any $t_0 \in I$, then there is a minimal $t^* \in (0, t_0)$ such that $\|\gamma(t) - x\| < \varepsilon$ for all $t \in [0, t^*)$ and $\|\gamma(t^*) - x\| = \varepsilon$. We have:

$$\|\gamma(t) - x\| \leq \int_0^t \|f(\gamma(s)) + g(\gamma(s))k_h(x)\|\,\mathrm{d}s \leq Mt, \tag{A.21}$$

for all $t \in [0, t^*]$. Since $\varepsilon = \|\gamma(t^*) - x\| \leq Mt^*$, we know that $t^* \geq \varepsilon/M > h_2$. Thus if $h \in (0, h_2)$, then:

$$\|\gamma(t) - x\| \leq Mt \leq Mh < Mh_2 < \varepsilon, \tag{A.22}$$

for all $t \in [0, h]$, implying $\gamma(t) \in N$ for all $t \in [0, h]$.

To bound the second term in (A.20), we show by induction that if h is sufficiently small, then $z_i \in N$ for all $i \in \{1, \ldots, p\}$. First, since $z_1 = x$, we have $z_1 \in N$.

Next, for $i \in \{1, \ldots, p\}$, suppose $z_j \in N$ for all $j \in \{1, \ldots, i-1\}$. Considering the definition of z_i in (A.8) and the bound (A.18):

$$\begin{aligned} \|z_i - x\| &\leq h \sum_{j=1}^{i-1} |a_{i,j}| \|f(z_j) + g(z_j) k_h(x)\| \\ &\leq Mh \sum_{j=1}^{i-1} |a_{i,j}| \leq Mh(p-1) \max_{j,k} |a_{j,k}| \triangleq Lh. \end{aligned} \tag{A.23}$$

Let $h^* \in (0, h_2)$ satisfy $h^* < \varepsilon/L$. Then for $h \in (0, h^*)$, we have $\|z_i - x\| < \varepsilon$, or $z_i \in N$. Since this choice of h^* does not depend on i, we can conclude by induction that if $h \in (0, h^*)$, then $z_i \in N$ for all $i \in \{1, \ldots, p\}$.

We have shown that if $h \in (0, h^*)$, then $\gamma(t) \in N$ for all $t \in [0, h]$, and $z_i \in N$ for $i \in \{1, \ldots, p\}$. Thus, using the bound (A.19) in (A.20), we have that:

$$\begin{aligned} \|F_h^e(x, k_h(x)) - F_h^{a,p}(x, k_h(x))\| &\leq \int_0^h \rho(\|\gamma(t) - x\|) \, \mathrm{d}t + h \sum_{i=1}^{p} b_i \rho(\|z_i - x\|) \\ &\leq h\rho(Mh) + h \sum_{i=1}^{p} b_i \rho(Lh) \leq h\tilde{\rho}(h), \end{aligned} \tag{A.24}$$

and $\tilde{\rho} \in \mathcal{K}$ satisfies $\tilde{\rho}(r) = \rho(Mr) + \rho(Lr)$ for all $r \in \mathbb{R}_+$. □

A.2 Practical Stability and Safety

With sampled-and-held control inputs, we will typically be unable to certify stability (marginal, asymptotic, or exponential) or safety (forward invariance of a safe set). In this section, we describe graceful degredations of these properties that yield useful design criteria while maintaining much of our desired behaviors.

A.2.1 Practical Stability

Definition 9 (*Practical Stability*)**.** Let $\beta \in \mathcal{KL}_\infty$ and $N \subseteq \mathbb{R}^n$ be an open set containing the origin. A family $\{(k_h, F_h) : h \in I\}$ is (β, N)-practically stable if for each $R \in \mathbb{R}_{++}$, there exists an $h^* \in I$ such that for each sample period $h \in (0, h^*)$, initial state $x_0 \in N$, and number of steps $k \in \mathbb{Z}_+$, the recursion $x_{i+1} = F_h(x_i, k_h(x_i))$ is well-defined and:

$$\|x_i\| \leq \beta(\|x_0\|, ih) + R. \tag{A.25}$$

If this property were true for $R = 0$ and a particular sample period h, this would mean the origin can be asymptotically stabilized and reached from any initial condition in

N, even when control inputs are held constant for periods of time h. Practical stability relaxes the condition of asymptotic convergence to the origin; only asymptotic convergence to a *neighborhood* of the origin is required, but we must be able to reduce the size of the neighborhood *arbitrarily* by reducing the interval of acceptable sample periods accordingly.

We now establish practical stability of an exact family of controller-map pairs through a Lyapunov-type construction.

Theorem 4. *Consider an open set $N \subseteq \mathcal{X}$ containing the origin, a family of admissible controllers $\{k_h : h \in I\}$, and a corresponding family of controller-map pairs $\{(k_h, F_h) : h \in I\}$. Suppose that for any compact set $K \subseteq N$, we can find a family of function $\{(V_h : \mathbb{R}^n \to \mathbb{R}_+) : h \in I\}$, some $h^* \in I$, comparison functions $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ and $\alpha_3 \in \mathcal{K}$, and a Lipschitz constant $M \in \mathbb{R}_{++}$ such that:*

$$\alpha_1(\|x_1\|) \leq V_h(x_1) \leq \alpha_2(\|x_1\|), \tag{A.26}$$

$$V_h(F_h(x_2, k_h(x_2))) - V_h(x_2) \leq -h\alpha_3(\|x_2\|), \tag{A.27}$$

$$|V_h(x_3) - V_h(x_4)| \leq M|x_3 - x_4|, \tag{A.28}$$

for all states $x_1 \in \mathbb{R}^n$, $x_2 \in N$, and $x_3, x_4 \in K$, and sample periods $h \in (0, h^*)$. If the family $\{(k_h, F_h) : h \in I\}$ is one-step consistent with the exact family $\{(k_h, F_h^e) : h \in I\}$, then there is some $\beta \in \mathcal{KL}_\infty$ and bounded open set $U \subseteq \mathcal{X}$ containing the origin such that the exact family is (β, N')-practically stable for any open set $N' \subseteq U$ containing the origin.

Proof. This theorem is a local variant of (Nešić, Teel, and Kokotović, 1999, Theorem 2). Let $K \subset N$ be a compact set containing the origin. There exist a $\rho \in \mathcal{K}_\infty$, a Lipschitz constant $M \in \mathbb{R}_+$, and an $h_0^* \in I$ such that for all $h \in (0, h_0^*)$, (A.17), (A.26), (A.27), and (A.28) hold for all states $x, x_1, \ldots, x_4 \in K$. There exists a radius $R \in \mathbb{R}_{++}$ such that the closed norm-ball around the origin of radius R is contained in K. We modify the claim of (Nešić, Teel, and Kokotović, 1999, Equation 37) for the local setting in this work as follows:

For any $d, D \in \mathbb{R}_{++}$ with $D \leq \alpha_2^{-1}(\alpha_1(\frac{R}{2}))$ and $d \leq 2\alpha_2(R)$, there exists an $h^* \in (0, h_0^*)$ such that for every state $x \in \mathcal{X}$ and sample period $h \in (0, h^*)$, if $\|x\| \leq D$ and $\max\{V_h(F_h^e(x, k_h(x))), V_h(x)\} \geq d$, then:

$$V_h(F_h^e(x, k_h(x))) - V_h(x) \leq -\frac{h}{2}\alpha_3(\|x\|). \tag{A.29}$$

We follow the proof closely, adding additional clarification where we deem appropriate. To prove this claim, as in (Nešić, Teel, and Kokotović, 1999), we define:

$$\delta = \frac{1}{2}\alpha_2^{-1}\left(\frac{d}{2}\right), \qquad \Delta = \alpha_1^{-1}(\alpha_2(D)) + \frac{1}{2}\alpha_2^{-1}\left(\frac{d}{2}\right). \tag{A.30}$$

By our assumptions on d and D, we have:

$$\Delta \leq \alpha_1^{-1}\left(\alpha_2\left(\alpha_2^{-1}\left(\alpha_1\left(\frac{R}{2}\right)\right)\right)\right) + \frac{1}{2}\alpha_2^{-1}\left(\frac{2\alpha_2(R)}{2}\right) = \frac{R}{2} + \frac{R}{2} = R. \tag{A.31}$$

Pick $h_1^* \leq h_0^*$ sufficiently small such that:

$$M\rho(h_1^*) \leq \frac{1}{2}\alpha_3(\alpha_2^{-1}(\alpha_1(\delta))), \tag{A.32}$$

and pick $h_2^* \leq h_1^*$ sufficiently small such that:

$$h_2^*\rho(h_2^*) \leq \frac{1}{2}\alpha_2^{-1}\left(\frac{d}{2}\right). \tag{A.33}$$

We now work with any $h \leq h_2^* \leq h_0^*$. Consider any $x \in \mathcal{X}$ with $\|x\| \leq D$. We first show that both $F_h(x, k_h(x))$ and $F_h^e(x, k_h(x))$ have norm less than $\Delta \leq R$, indicating that both are contained in K and enabling the use of Equation (A.28). Since:

$$\alpha_1(\|F_h(x, k_h(x))\|) \leq V_h(F_h(x, k_h(x))), \tag{A.34}$$

we have:

$$\begin{aligned}
\|F_h(x, k_h(x))\| &\leq \alpha_1^{-1}(V_h(F_h(x, k_h(x)))) \\
&\leq \alpha_1^{-1}(V_h(x)) \\
&\leq \alpha_1^{-1}(\alpha_2(\|x\|)) \\
&\leq \alpha_1^{-1}(\alpha_2(D)) \leq \Delta.
\end{aligned} \tag{A.35}$$

We can then use Equation (A.17) to conclude that:

$$\begin{aligned}
\|F_h^e(x, k_h(x))\| &\leq \|F_h(x, k_h(x))\| + h\rho(h) \\
&\leq \alpha_1^{-1}(\alpha_2(D)) + h_2^*\rho(h_2^*) \\
&\leq \alpha_1^{-1}(\alpha_2(D)) + \frac{1}{2}\alpha_2^{-1}\left(\frac{d}{2}\right) \leq \Delta.
\end{aligned} \tag{A.36}$$

We now know that both $F_h(x, k_h(x))$ and $F_h^e(x, k_h(x))$ are contained in K, so we can conclude that:

$$\begin{aligned}
|V_h(F_h(x, k_h(x))) - V_h(F_h^e(x, k_h(x)))| &\leq M\|F_h(x, k_h(x)) - F_h^e(x, k_h(x))\| \\
&\leq Mh\rho(h).
\end{aligned} \tag{A.37}$$

We now consider both terms in the maximum of the claim. Supposing that $V_h(F_h^e(x, k_h(x))) \geq d/2$ (this is stricter than we need), we have:

$$\frac{d}{2} \leq V_h(F_h^e(x, k_h(x))) \leq \alpha_2(\|F_h^e(x, k_h(x))\|), \tag{A.38}$$

so:

$$\|F_h^e(x, k_h(x))\| \geq \alpha_2^{-1}(V_h(F_h(x, k_h(x)))) \geq \alpha_2^{-1}\left(\frac{d}{2}\right) = 2\delta. \tag{A.39}$$

Since:

$$\|F_h(x, k_h(x)) - F_h^e(x, k_h(x))\| \leq h\rho(h) \leq \delta, \tag{A.40}$$

we have:

$$\|F_h(x, k_h(x))\| \geq \|F_h^e(x, k_h(x))\| - \delta \geq 2\delta - \delta = \delta. \tag{A.41}$$

From Equation (A.35), this means:

$$\alpha_1^{-1}(\alpha_2(\|x\|)) \geq \|F_h(x, k_h(x))\| \geq \delta, \tag{A.42}$$

so:

$$\alpha_2^{-1}(\alpha_1(\delta)) \leq \|x\|. \tag{A.43}$$

We use this in:

$$\begin{aligned} V_h(F_h^e(x, k_h(x))) - V_h(x) &= V_h(F_h(x, k_h(x))) - V_h(x) \\ &\quad + V_h(F_h(x, k_h(x))) - V_h(F_h^e(x, k_h(x))) \\ &\leq -h\alpha_3(\|x\|) + Mh\rho(h) \\ &\leq -h\alpha_3(\|x\|) + \frac{h}{2}\alpha_3(\alpha_2^{-1}(\alpha_1(\delta))) \\ &\leq -h\alpha_3(\|x\|) + \frac{h}{2}\alpha_3(\|x\|) = -\frac{h}{2}\alpha_3(\|x\|). \end{aligned} \tag{A.44}$$

If instead $V_h(F_h^e(x, k_h(x))) < d/2$, we consider the case when $V_h(x) \geq d$. Negating Equation (A.27), we obtain:

$$V_h(x) \geq V_h(x) - V_h(F_h(x, k_h(x))) \geq h\alpha_3(\|x\|), \tag{A.45}$$

so:

$$\begin{aligned} V_h(F_h^e(x, k_h(x))) - V_h(x) &\leq \frac{d}{2} - V_h(x) \\ &= \frac{d - V_h(x)}{2} - \frac{1}{2}V_h(x) \\ &\leq -\frac{1}{2}V_h(x) \leq -\frac{h}{2}\alpha_3(\|x\|). \end{aligned} \tag{A.46}$$

This verifies the claim.

We can now return to the original proof. First, let U be the open ball of radius $\alpha_2^{-1}(\alpha_1(\alpha_2^{-1}(\alpha_1(R/2))))$. Since V_h is bounded both below and above by α_1 and α_2, respectively, we have:

$$\alpha_1(r) \leq \alpha_2(r) \implies \alpha_2^{-1}(\alpha_1(r)) \leq r, \tag{A.47}$$

so the radius of U is no more than $R/2$. This means U is completely contained in K, and we can employ Equation (A.17), Equation (A.27), and Equation (A.28) over U.

Pick an open set $N' \subseteq U$ containing the origin. Pick D as:

$$\begin{aligned} D &= \sup_{x \in N'} \alpha_1^{-1}(\alpha_2(\|x\|)) \\ &\leq \sup_{x \in U} \alpha_1^{-1}(\alpha_2(\|x\|)) \\ &\leq \alpha_1^{-1}(\alpha_2(\alpha_2^{-1}(\alpha_1(\alpha_2^{-1}(\alpha_1(R/2)))))) \\ &= \alpha_2^{-1}(\alpha_1(R/2)), \end{aligned} \tag{A.48}$$

satisfying our requirement on the choice of D. We can freely pick $d \in (0, \alpha_1(D))$, since:

$$d \leq \alpha_1(D) \leq \alpha_2(D) \leq \alpha_2(\alpha_2^{-1}(\alpha_1(R/2))) \leq \alpha_1(R/2) \leq \alpha_2(R/2) \leq 2\alpha_2(R), \tag{A.49}$$

again, satisfying our requirement on the choice of d. We can now make use of the claim. For any $h \leq h_2^* \leq h_0^*$, from an initial condition $x_0 \in N'$, we generate the exact sequence of states:

$$x_{i+1} = F_h^e(x_i, k_h(x_i)), \tag{A.50}$$

for all $i \in \mathbb{Z}_+$. By induction, we show that:

$$V_h(x_i) \leq \max\{V_h(x_0), d\}. \tag{A.51}$$

Assume this is true; we then have:

$$\begin{aligned} \|x_i\| &\leq \alpha_1^{-1}(V_h(x_i)) \\ &\leq \max\{\alpha_1^{-1}(V_h(x_0)), \alpha_1^{-1}(d)\} \\ &\leq \max\{\alpha_1^{-1}(\alpha_2(\|x_0\|)), \alpha_1^{-1}(\alpha_1(D))\} \leq D. \end{aligned} \tag{A.52}$$

Since $\|x_i\| \leq D$, if $V_h(x_{i+1}) = V_h(F_h^e(x, k_h(x))) \geq d$, then we can employ the claim to show that:

$$V_h(x_{i+1}) \leq -\frac{h}{2}\alpha_3(\|x_i\|) + V_h(x_i) \leq V_h(x_i) \leq \max\{V_h(x_0), d\}. \tag{A.53}$$

Otherwise, if $V_h(x_{i+1}) < d$, then simply $V_h(x_{i+1}) \leq \max\{V_h(x_0), d\}$.

We can also define a new comparison function $\alpha \in \mathcal{K}$ with some additional manipulation. If:

$$\max\{V_h(x_{i+1}), V_h(x_i)\} = \max\{V_h(F_h^e(x_i, k_h(x_i))), V_h(x_i)\} \geq d, \tag{A.54}$$

then we can again employ the claim to show that:

$$V_h(x_{i+1}) - V_h(x_i) \leq -\frac{h}{2}\alpha_3(\|x\|) \leq -\frac{h}{2}\alpha_3(\alpha_2^{-1}(V_h(x_i))). \tag{A.55}$$

As such, define $\alpha \in \mathcal{K}$ as:

$$\alpha(r) = \frac{1}{2}\alpha_3(\alpha_2^{-1}(r)). \tag{A.56}$$

Letting $y_h : \mathbb{R}_+ \to \mathbb{R}$ denote the linear interpolation of values $V_h(x_i)$, note that:

$$\dot{y}_h(t) = \frac{V_h(x_{i+1}) - V_h(x_i)}{h}, \tag{A.57}$$

for all $t \in (ih, ih + h)$. We can now analyze y_h with continuous-time tools. Specifically, whenever $y_h(t) \geq d$, we have:

$$\max\{V_h(x_{i+1}), V_h(x_i)\} \geq d, \tag{A.58}$$

for whichever $i \in \mathbb{Z}_+$ satisfies $t \in [ih, ih + h]$. This means:

$$\dot{y}_h(t) = \frac{V_h(x_{i+1}) - V_h(x_i)}{h} \leq \frac{-h\alpha(V_h(x_i))}{h} = -\alpha(V_h(x_i)). \tag{A.59}$$

Moreover, since:

$$V_h(x_{i+1}) - V_h(x_i) \geq -h\alpha(V_h(x_i)) \geq 0 \implies V_h(x_i) = \min\{V_h(x_{i+1}), V_h(x_i)\}, \tag{A.60}$$

we know $y_h(t) \geq V_h(x_i)$ and:

$$\dot{y}_h(t) \leq -\alpha(y_h(t)). \tag{A.61}$$

Essentially, whenever y_h sufficiently large, its rate of change is large and negative, indicating that we can reason about y_h using concepts like input-to-state stability (Sontag, 1989b). In particular, there is some $\beta \in \mathcal{KL}$ satisfying:

$$y_h(t) \leq \max\{\beta(y_h(0), t), d\} \tag{A.62}$$

for all $t \in \mathbb{R}_+$. Since Equation (A.61) is no longer explicitly dependent on the sample period (the same inequality holds for all sufficiently small sample periods), we can choose β independent of the sample period.

We use this result to show that:

$$\alpha_1(\|x_i\|) \leq V_h(x_i) \leq \max\{\beta(V_h(x_0), ih), d\}, \tag{A.63}$$

for all $i \in \mathbb{Z}_+$, so:

$$\begin{aligned} \|x_i\| &\leq \max\{\alpha_1^{-1}(\beta(V_h(x_0), ih)), \alpha_1^{-1}(d)\} \\ &\leq \alpha_1^{-1}(\beta(V_h(x_0), ih)) + \alpha_1^{-1}(d). \end{aligned} \tag{A.64}$$

The composition $\alpha_1^{-1} \circ \beta \in \mathcal{KL}$ does not depend on d, so choosing d however small is necessary, we achieve $(\alpha_1^{-1} \circ \beta, N')$-practical stability. □

A.2.2 Practical Safety

We first adapt a definition of forward invariance for a controller-map pairs.

Definition 10 (*Forward Invariance*)**.** A set $C \subseteq \mathcal{X}$ is *forward invariant* for a controller-map pair (k, F) if for every $x_0 \in C$ and number of steps $i \in \mathbb{Z}_+$, the recursion $x_{i+1} = F(x_i, k(x_i))$ is well-defined and satisfies $x_i \in C$.

The following definition will describe the safety properties of the exact family of controller-map pairs when design uses approximations:

Definition 11 (*Practical Safety*)**.** A family $\{(k_h, F_h) \ : \ h \in I\}$ is *practically safe* with respect to a set $C \subseteq \mathcal{X}$ if for each $R \in \mathbb{R}_{++}$, there exists an $h^* \in I$ such that for each sample period $h \in (0, h^*)$, there is a corresponding set $C_h \subseteq \mathcal{X}$ that is forward invariant for the controller-map pair (k_h, F_h) and satisfies $C \subseteq C_h \subseteq C + \overline{B}_R$.

Again, if this property were true for $R = 0$ and a particular sample period h, this would mean $C \subseteq C_h \subseteq C$, so $C_h = C$ is forward invariant, even when control inputs are held constant for periods of time h. Practical safety relaxes the condition of forward invariance; only forward invariance of an expanded set is required, but we must be able to reduce the size of the expansion *arbitrarily* by reducing the interval of acceptable sample periods accordingly. Note further that the burden of proof lies with small values of R. If $R' \geq R$ and C_h is a forward invariant subset of $C + \overline{B}_R$, then it is automatically a forward invariant subset of $C + \overline{B}_{R'}$.

We now establish practical safety of an exact family of controller-map pairs through a barrier-type construction.

Theorem 5. *Consider a set $C \subseteq \mathcal{X}$, a family of functions $\{(s_h : \mathcal{X} \to \mathbb{R}) : h \in I\}$, some $h^* \in I$, a comparison function $\alpha \in \mathcal{K}^e$, a radius $\varepsilon \in \mathbb{R}_{++}$, and a Lipschitz constant $M \in \mathbb{R}_{++}$ such that:*

$$s_h(x_1) > 0, \qquad s_h(x_2) = 0, \qquad s_h(x_3) < 0, \tag{A.65}$$

$$h\alpha(s_h(x_4)) \leq s_h(x_4), \tag{A.66}$$

$$|s_h(x_5) - s_h(x_6)| \leq M\|x_5 - x_6\|, \tag{A.67}$$

for all states $x_1 \in \operatorname{int} C$, $x_2 \in \partial C$, $x_3 \in \mathcal{X} \setminus C$, $x_4 \in C$, $x_5, x_6 \in \mathcal{X} \cap (C + \overline{B}_\varepsilon)$, and sample periods $h \in (0, h^*)$. Additionally, we require that for each $\eta \in \mathbb{R}_{++}$, there is a corresponding $\delta \in \mathbb{R}_{++}$ such that:

$$d_C(x) \triangleq \inf_{y \in C} \|y - x\| > \eta \implies s_h(x) < -\delta, \tag{A.68}$$

for all $x \in \mathcal{X} \cap (C + \overline{B}_\varepsilon)$ and $h \in (0, h^*)$. For admissible controllers $\{k_h : h \in I\}$ and a corresponding family of controller-map pairs $\{(k_h, F_h) : h \in I\}$, suppose:

$$s_h(F_h(x, k_h(x))) - s_h(x) \geq -h\alpha(s_h(x)), \tag{A.69}$$

for all states $x \in \mathcal{X}$ and sample times $h \in (0, h^*)$. If there is a radius $\varepsilon' \in \mathbb{R}_{++}$ such that the family $\{(k_h, F_h) : h \in I\}$ is one-step consistent with the exact family $\{(k_h, F_h^e) : h \in I\}$ over the set $\mathcal{X} \cap (C + \overline{B}_{\varepsilon'})$, then the exact family is practically safe with respect to C.

Proof. By assumption, there exists an $h_2^* \in I$ and $\rho \in \mathcal{K}_\infty$ such that (A.17) holds for all $x \in \mathcal{X} \cap (C + \overline{B}_{\varepsilon'})$ and $h \in (0, h_2^*)$. As the family of controllers is assumed admissible, there is an $h_3^* \in I$ such that k_h is h-admissible for each $h \in (0, h_3^*)$.

Let $R \in \mathbb{R}_{++}$, and pick $R' \in \mathbb{R}_{++}$ such that $R' \leq \min\{\varepsilon, \varepsilon', R\}$. By (A.68), there exist $\delta, \Delta \in \mathbb{R}_{++}$ such that:

$$d_C(x) > R'/2 \implies s_h(x) < -\delta, \tag{A.70}$$

$$d_C(x) > \delta/(2M) \implies s_h(x) < -\Delta, \tag{A.71}$$

for all $x \in \mathcal{X} \cap (C + \overline{B}_\varepsilon)$ and $h \in (0, h_1^*)$. Fix $h \in I$ with $h < \min\{h_1^*, h_2^*, h_3^*\}$. For any $c \in \mathbb{R}$, we denote the c-superlevel set of s_h as:

$$\Omega_{c,h} = \{x \in \mathcal{X} \mid s_h(x) \geq c\}. \tag{A.72}$$

For any state $x \in \Omega_{-\delta,h}$, we have $d_C(x) \leq R'/2$, and thus $C \subseteq \Omega_{-\delta,h} \subseteq \mathcal{X} \cap (C + \overline{B}_{R'/2}) \subseteq C + \overline{B}_R$.

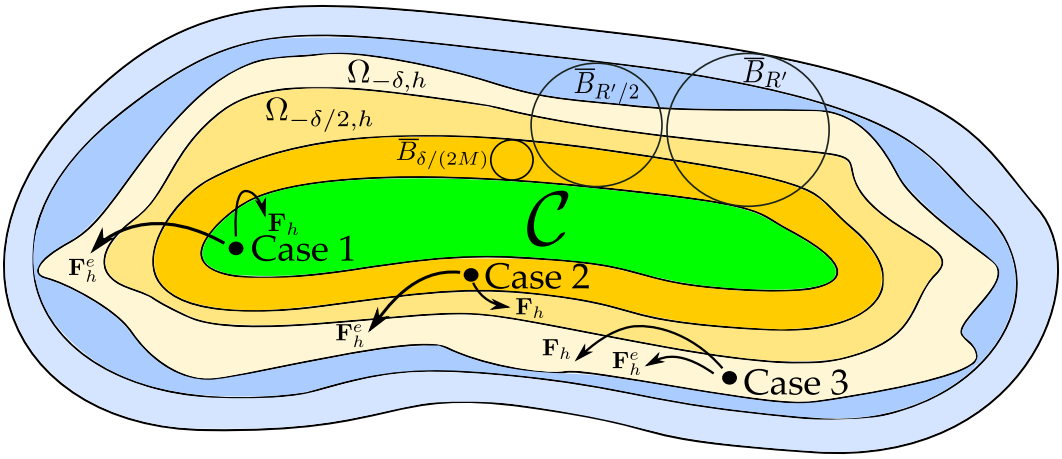


Figure A.1: A visual representation of the main sets and three cases discussed in the proof of Theorem 5.

We will prove that for small enough h, the set $\Omega_{-\delta,h}$ is forward invariant for the controller-map pair (k_h, F_h^e). We denote three cases (see Fig. A.1), considering a state $x \in \mathcal{X}$ such that either

1. $x \in C$,
2. $x \in \Omega_{-\delta,h} \setminus C$ and $d_C(x) \leq \delta/(2M)$, or
3. $x \in \Omega_{-\delta,h} \setminus C$ and $d_C(x) > \delta/(2M)$.

Case 1: Suppose $x \in C$. From (A.69) and (A.66), we have:

$$s_h(F_h(x, k_h(x))) - s_h(x) \geq -h\alpha(s_h(x)) \geq -s_h(x), \tag{A.73}$$

so $s_h(F_h(x, k_h(x))) \geq 0$, or $F_h(x, k_h(x)) \in C$. By one-step consistency, we have:

$$\|F_h^e(x, k_h(x)) - F_h(x, k_h(x))\| \leq h\rho(h), \tag{A.74}$$

so if $h\rho(h) \leq \varepsilon$, then $F_h^e(x, k_h(x)) \in \mathcal{X} \cap (C + \overline{B}_\varepsilon)$. Thus:

$$|s_h(F_h^e(x, k_h(x))) - s_h(F_h(x, k_h(x)))| \leq Mh\rho(h), \tag{A.75}$$

and if $Mh\rho(h) \leq \delta$ as well, then:

$$s_h(F_h^e(x, k_h(x))) \geq s_h(F_h(x, k_h(x))) - Mh\rho(h) \geq -\delta, \tag{A.76}$$

giving us $F_h^e(x, k_h(x)) \in \Omega_{-\delta,h}$. The analysis of this case gives us the requirement $h\rho(h) \leq \min\{\varepsilon, \delta/M\}$.

Before continuing to cases 2 and 3, we establish some additional properties. First, note that the superlevel sets have the containment property $\Omega_{-\delta/2,h} \subseteq \Omega_{-\delta,h}$. Next, for any $\eta \in \mathbb{R}_{++}$ and any $x \in \mathcal{X} \cap (C + \overline{B}_\varepsilon)$ with $x \notin C$, there is a state $y \in C$ such that $\|x - y\| < d_C(x) + \eta$. Therefore:

$$s_h(x) \geq s_h(y) - M\|x - y\| \geq -Md_C(x) - M\eta, \tag{A.77}$$

since $s_h(y) \geq 0$. Since η can be chosen arbitrarily small, we have $s_h(x) \geq -Md_C(x)$. If $d_C(x) \leq \delta/(2M)$, then $s_h(x) \geq -\delta/2$, so $\mathcal{X} \cap (C + \overline{B}_{\delta/(2M)}) \subseteq \Omega_{-\delta/2,h} \subseteq \Omega_{-\delta,h}$.

Next, consider $x \in \Omega_{-\delta,h} \setminus C$. Since $x \notin C$, meaning $s_h(x) < 0$ and thus $\alpha(s_h(x)) < 0$, we have from (A.69) that:

$$s_h(F_h(x, k_h(x))) \geq s_h(x) - h\alpha(s_h(x)) > -\delta. \tag{A.78}$$

Thus $F_h(x, k_h(x)) \in \Omega_{-\delta,h} \subseteq \mathcal{X} \cap (C + \overline{B}_{R'/2})$ so we can apply one step consistency to achieve:

$$\|F_h^e(x, k_h(x)) - F_h(x, k_h(x))\| \leq h\rho(h). \tag{A.79}$$

If $h\rho(h) \leq R'/2$, then $F_h^e(x, k_h(x)) \in \mathcal{X} \cap (C + \overline{B}_{R'})$, in which case the Lipschitz property of s_h yields the bound:

$$|s_h(F_h^e(x, k_h(x))) - s_h(F_h(x, k_h(x)))| \leq Mh\rho(h). \tag{A.80}$$

Note that because $R'/2 < \varepsilon$, the requirement from Case 1 can be replaced by $h\rho(h) \leq \min\{R'/2, \delta/M\}$.

Case 2: Suppose $x \in \Omega_{-\delta,h} \setminus C$ and $d_C(x) \leq \delta/(2M)$. Since $x \notin C$ and $\mathcal{X} \cap (C + \overline{B}_{\delta/(2M)}) \subseteq \Omega_{-\delta/2,h}$, we have $-\delta/2 \leq s_h(x) < 0$. Therefore:

$$s_h(F_h(x, k_h(x))) \geq s_h(x) - h\alpha(s_h(x)) \geq -\delta/2, \tag{A.81}$$

so $F_h(x, k_h(x)) \in \Omega_{-\delta/2,h}$. By adding and subtracting $s_h(F_h^e(x, k_h(x)))$ and using (A.80), we have:

$$s_h(F_h^e(x, k_h(x))) \geq -Mh\rho(h) - \delta/2, \tag{A.82}$$

when $h\rho(h) \leq R'/2$. If $Mh\rho(h) \leq \delta/2$ as well, then $s_h(F_h^e(x, k_h(x))) \geq -\delta$, or $F_h^e(x, k_h(x)) \in \Omega_{-\delta,h}$. Thus we update the requirements to be $h\rho(h) \leq \min\{R'/2, \delta/(2M)\}$.

Case 3: Suppose $x \in \Omega_{-\delta,h} \setminus C$ and $d_C(x) > \delta/(2M)$. From (A.71), we have:

$$s_h(F_h(x, k_h(x))) - s_h(x) > -h\alpha(-\Delta). \tag{A.83}$$

Adding and subtracting $s_h(F_h^e(x, k_h(x)))$ and (A.80) yield:

$$\begin{aligned} s_h(F_h^e(x, k_h(x))) &> s_h(x) - Mh\rho(h) - h\alpha(-\Delta), \\ &= s_h(x) - h(M\rho(h) + \alpha(-\Delta)), \end{aligned} \tag{A.84}$$

when $h\rho(h) \leq R'/2$. If $M\rho(h) \leq -\alpha(-\Delta)$ as well, then $s_h(F_h^e(x, k_h(x))) > s_h(x) \geq -\delta$, or $F_h^e(x, k_h(x)) \in \Omega_{-\delta,h}$.

To conclude, if both:

1. $h < \min\left\{h_1^*, h_2^*, h_3^*, \rho^{-1}(-\alpha(-\Delta)/M)\right\}$,
2. $h\rho(h) \leq \min\{R'/2, \delta/(2M)\}$,

then the set $C_h \triangleq \Omega_{-\delta,h} \subseteq C + \overline{B}_R$ is forward invariant for the controller-map pair (k_h, F_h^e), and thus the family $\{(k_h, F_h^e) \ : \ h \in I\}$ is practically safe with respect to C. □

A.3 Controller Synthesis

A.3.1 Stabilization

The nonlinear control system is *full-state feedback linearizable* if there is a choice of control $k_{\text{fbl}} : \mathcal{X} \times \mathbb{R}^m \to \mathbb{R}^m$ (with an auxiliary input) such that:

$$f(x) + g(x)k_{\text{fbl}}(x, v) = Ax + Bv, \tag{A.85}$$

for all $x \in \mathcal{X}$ and auxiliary inputs $v \in \mathbb{R}^m$, where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ form a controllable pair (A, B). With (A, B) controllable, we can find an auxiliary control gain matrix $K \in \mathbb{R}^{m\times n}$ such that $A - BK$ is Hurwitz (all eigenvalues have negative real parts). Choosing the controller $k_{\text{fbl},K} : \mathcal{X} \to \mathbb{R}^m$ as:

$$k_{\text{fbl},K}(x) = k_{\text{fbl}}(x, -Kx), \tag{A.86}$$

the closed-loop dynamics of the system are stable linear dynamics:

$$\dot{x} = (A - BK)x \triangleq A_{\text{cl}}x. \tag{A.87}$$

With sampled-and-held control inputs, we may be unable to achieve these closed-loop dynamics, but we can still use the closed-loop dynamics to generate a family of functions to use with Theorem 4. Specifically, for any positive definite matrix

$Q \in \mathbb{S}^n_{++}$, there is a unique solution $P \in \mathbb{S}^n_{++}$ to the continuous time Lyapunov equation:

$$A_{\text{cl}}^\top P + P A_{\text{cl}} = -Q. \tag{A.88}$$

We can then define $\{(V_h : \mathbb{R}^n \to \mathbb{R}_+) : h \in I\}$ with:

$$V_h(x) = x^\top P x, \tag{A.89}$$

for all states $x \in \mathbb{R}^n$. Note that these functions do not actually vary with the sample period h. Immediately, we have:

$$\lambda_{\min}(P)\|x\|_2^2 \leq V_h(x) \leq \lambda_{\max}(P)\|x\|_2^2, \tag{A.90}$$

for all states $x \in \mathbb{R}^n$ and sample period $h \in I$, where $\lambda_{\min}$ and $\lambda_{\max}$ denote minimum and maximum eigenvalues. Moreover, for each sample period $h \in I$, the function V_h is continuously differentiable, therefore locally Lipschitz continuous. This means over any compact set, we can find a global Lipschitz constant for V_h. Finally, we show that the Euler approximate family with feedback linearizing control satisfies Equation (A.27):

$$\begin{aligned}
V_h(F_h^a(x, k_{\text{fbl},K}(x))) - V_h(x) &= V_h(x + h \cdot A_{\text{cl}} x) - V_h(x) \\
&= x^\top (I + hA_{\text{cl}})^\top P (I + hA_{\text{cl}}) x - x^\top P x \\
&= x^\top \left(h(A_{\text{cl}}^\top P + P A_{\text{cl}}) + h^2 A_{\text{cl}}^\top P A_{\text{cl}} \right) x \\
&= -h x^\top \left(Q - h A_{\text{cl}}^\top P A_{\text{cl}} \right) x \\
&\leq -h(\lambda_{\min}(Q) - h\lambda_{\max}(A_{\text{cl}}^\top P A_{\text{cl}}))\|x\|_2^2,
\end{aligned} \tag{A.91}$$

for all states $x \in \mathbb{R}^n$. For any $c \in (0, 1)$, we know that if:

$$h \leq (1 - c)\frac{\lambda_{\min}(Q)}{\lambda_{\max}(A_{\text{cl}}^\top P A_{\text{cl}})}, \tag{A.92}$$

we have:

$$\begin{aligned}
V_h(F_h^a(x, k_{\text{fbl},K}(x))) - V_h(x) &\leq -h(\lambda_{\min}(Q) - (1 - c)\lambda_{\min}(Q))\|x\|_2^2 \\
&= -hc\lambda_{\min}(Q)\|x\|_2^2.
\end{aligned} \tag{A.93}$$

This condition can be incorporated into a convex optimization problem if we wish to add additional constraints or regularizers. Consider the following stabilization optimization problem, in a state $x \in \mathbb{R}^n$ with a sample period $h \in I$:

$$\begin{aligned}
\inf_{u \in \mathbb{R}^m} \quad & \|u\|_2^2 \\
\text{s.t.} \quad & V_h(x + h(f(x) + g(x)u)) - V_h(x) \leq -hc\lambda_{\min}(Q)\|x\|_2^2.
\end{aligned} \tag{A.94}$$

Expanding V_h and dividing by h, we obtain the following stabilization quadratically-constrained quadratic program (QCQP):

$$\begin{aligned} \inf_{u\in\mathbb{R}^m} \quad & \|u\|_2^2 \\ \text{s.t.} \quad & u^\top (hg(x)^\top P g(x))u + 2(g(x)^\top P(x + hf(x)))^\top u \\ & + hf(x)^\top P f(x) + 2x^\top P f(x) + c\lambda_{\min}(Q)\|x\|_2^2 \leq 0. \end{aligned} \tag{A.95}$$

Extensions requiring a change of variables for feedback linearization and/or systems with exponentially stable zero dynamics can be found in (Taylor, Dorobantu, Yue, et al., 2021).

A.3.2 Safety

The following result establishes how for a system with a block integrator structure, a Runge-Kutta approximation family of maps of the appropriate order can preserve a convexity property of a family $\{s_h \ : \ h \in I\}$:

Theorem 6. *Consider $\ell, \gamma, q \in \mathbb{N}$ such that $n = \ell\gamma$ and $q \leq \gamma$. Suppose the system dynamics have the form:*

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & I & & \\ & \ddots & \ddots & \\ & & 0 & I \\ & & & 0 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ f_\gamma(x) + g_\gamma(x)u \end{bmatrix}}_{r(x,u)}, \tag{A.96}$$

where $f_\gamma : \mathbb{R}^n \to \mathbb{R}^\ell$ and $g_\gamma : \mathbb{R}^n \to \mathbb{R}^{\ell\times m}$. For each $h \in I$, consider a function $s_h : \mathbb{R}^n \to \mathbb{R}$, and suppose there exists a function $\tilde{s}_h : (\mathbb{R}^\ell)^q \to \mathbb{R}$ satisfying:

$$s_h(x) = \tilde{s}_h(\zeta_1, \ldots, \zeta_q), \tag{A.97}$$

for all $x = (\zeta_1, \ldots, \zeta_\gamma) \in (\mathbb{R}^\ell)^\gamma \simeq \mathbb{R}^n$. If the function $\tilde{s}_h$ is concave with respect to its last argument and $p = \gamma - q + 1$, then for $\alpha \in \mathcal{K}^e$, the function $\phi_h : \mathcal{Z} \to \mathbb{R}$ defined as:

$$\phi_h(x, u) = -s_h(F_h^{a,p}(x, u)) + s_h(x) - h\alpha(s_h(x)), \tag{A.98}$$

is convex in its second argument.

Proof. For all $(x, u) \in \mathcal{Z}$, denote:

$$F_h^{a,p}(x, u) = ((F_1)_h^{a,p}(x, u), \ldots, (F_\gamma)_h^{a,p}(x, u)), \tag{A.99}$$

where $(F_i)_h^{a,p} : \mathcal{Z} \to \mathbb{R}^\ell$ for all $i \in \{1, \ldots, \gamma\}$. For $(x, u) \in \mathcal{Z}$, the block vector $r(x, u)$ can be nonzero only in the last (γth) block. Noting the block chain-of-integrators structure of A, for any degree $d \in \{0, \ldots, \gamma - 1\}$, $A^d r(x, u)$ can be nonzero only in the $(\gamma - d)$th block, and for a degree d polynomial ρ_d, $\rho_d(A)r(x, u)$ can be nonzero only in the last $d + 1$ blocks (that is, blocks $\gamma - d$ through γ).

Consider a state-input pair $(x, u) \in \mathcal{Z}$. We have:

$$F_h^{a,p}(x, u) = x + h \sum_{i=1}^{p} b_i (A z_i + r(z_i, u)), \tag{A.100}$$

$$z_i = x + h \sum_{j=1}^{i-1} a_{i,j} (A z_j + r(z_j, u)), \tag{A.101}$$

with $z_1 = x$. By induction, for any $i \in \{1, \ldots, p\}$, we show:

$$z_i = \rho_{i,i-1}(A)x + \sum_{j=1}^{i-1} \sigma_{i,i-j-1}(A) r(z_j, u), \tag{A.102}$$

where $\rho_{i,i-1}$ is a degree $i - 1$ polynomial, and for $j \in \{1, \ldots, i - 1\}$, $\sigma_{i,i-j-1}$ is a degree $i - j - 1$ polynomial. Indeed, $z_1 = I \cdot x$, and assuming (A.102) holds for $0, \ldots, i - 1$, substituting (A.102) into (A.101) yields the following:

$$\begin{aligned} z_i = \overbrace{\underbrace{\left(I + h \sum_{j=1}^{i-1} a_{i,j} \overbrace{A \rho_{j,j-1}(A)}^{\text{degree } j} \right)}_{\triangleq \rho_{i,i-1}(A)} x}^{\text{degree } i-1} + h \sum_{j=1}^{i-1} a_{i,j} r(z_j, u) \\ + h \sum_{k=1}^{i-1} \sum_{j=1}^{k-1} a_{i,k} A \sigma_{k,k-j-1}(A) r(z_j, u), \end{aligned} \tag{A.103}$$

which we may further manipulate to obtain:

$$z_i - \rho_{i,i-1}(A)x = \sum_{j=1}^{i-1} \underbrace{h \left(a_{i,j} + \sum_{k=j+1}^{i-1} a_{i,k} \underbrace{A \sigma_{k,k-j-1}(A)}_{\text{degree } k-j} \right)}_{\text{degree } i-j-1} r(z_j, u), \tag{A.104}$$

$$\triangleq \sum_{j=1}^{i-1} \sigma_{i,i-j-1}(A) r(z_j, u), \tag{A.105}$$

establishing (A.102) holds for i. Substituting the expression (A.102) into (A.100) and following a similar sequence of steps, we find a degree p polynomial $\tilde{\rho}_p$, and for each $i \in \{1, \ldots, p\}$, a degree $p - i$ polynomial $\tilde{\sigma}_{p-i}$ such that:

$$F_h^{a,p}(x, u) = \tilde{\rho}_p(A)x + \sum_{i=1}^{p} \tilde{\sigma}_{p-i}(A)r(z_i, u). \tag{A.106}$$

For $i \in \{1, \ldots, p\}$, the term $\tilde{\sigma}_{p-i}(A)r(z_i, u)$ can be nonzero only in blocks $\gamma - (p - i) = q + i - 1$ through γ. The highest-order polynomial multiplying the block vectors $r(z_1, u), \ldots, r(z_p, u)$ is $\tilde{\sigma}_{p-1} = \tilde{\sigma}_{\gamma-q}$. Therefore, the functions $(F_1)_h^{a,p}, \ldots, (F_{q-1})_h^{a,p}$ are independent of their second argument (they depend only on state). Moreover, $(F_q)_h^{a,p}(x, u)$ depends on the block vector $r(z_1, u) = r(x, u)$, which depends on u affinely, and does not depend on $r(z_2, u), \ldots, r(z_p, u)$, which may depend on u nonlinearly.

The composition $s_h \circ F_h^{a,p} : \mathcal{Z} \to \mathbb{R}$ satisfies:

$$s_h(F_h^{a,p}(x, u)) = \tilde{s}_h((F_1)_h^{a,p}(x, u), \cdots, (F_q)_h^{a,p}(x, u)),$$

for all $(x, u) \in \mathcal{Z}$. The composition of concave and affine functions is concave, so $s_h \circ F_h^{a,p}$ is concave in its second argument, and ϕ_h in (A.98) is convex in its second argument. □

We can leverage this result to formulate a safe convex optimization, in a state $x \in \mathbb{R}^n$ with a sample period $h \in I$:

$$\begin{aligned} \inf_{u \in \mathbb{R}^m} \quad & c(u) \\ \text{s.t.} \quad & -s_h(F_h^{a,p}(x, u)) + s_h(x) - h\alpha(s_h(x)) \leq 0, \end{aligned} \tag{A.107}$$

where $c : \mathbb{R}^m \to \mathbb{R}$ is a convex cost function.