

Robust Feedback Motion Planning via Contraction Theory

Sumeet Singh¹, Benoit Landry¹, Anirudha Majumdar²,
Jean-Jacques Slotine³, and Marco Pavone¹

¹Department of Aeronautics and Astronautics, Stanford University *

²Department of Mechanical and Aerospace Engineering, Princeton University, †

³Department of Mechanical Engineering, Massachusetts Institute of Technology ‡

September 10, 2019

Abstract

We present a framework for online generation of robust motion plans for robotic systems with nonlinear dynamics subject to bounded disturbances, control constraints, and online state constraints such as obstacles. In an offline phase, one computes the structure of a feedback controller that can be efficiently implemented online to track *any* feasible nominal trajectory. The offline phase leverages *contraction theory* and *convex optimization* to characterize a fixed-size “tube” that the state is guaranteed to remain within while tracking a nominal trajectory (representing the center of the tube). In the online phase, when the robot is faced with obstacles, a motion planner uses such a tube as a robustness margin for collision checking, yielding nominal trajectories that can be safely executed, i.e., tracked without collisions under disturbances. In contrast to recent work on robust online planning using funnel libraries, our approach is not restricted to a fixed library of maneuvers computed offline and is thus particularly well-suited to applications such as UAV flight in densely cluttered environments where complex maneuvers may be required to reach a goal. We demonstrate our approach through numerical simulations of planar and 3D quadrotors, and hardware results on a quadrotor platform navigating a complex obstacle environment while subject to aerodynamic disturbances. The results demonstrate the ability of our approach to jointly balance motion safety and efficiency for agile robotic systems.

1 Introduction

Despite significant progress in the field of motion planning, the problem of safe real-time planning for robots with nonlinear and underactuated dynamics subject to uncertainty has remained an outstanding challenge. A key difficulty is that uncertainty and disturbances in the dynamics force us to reason about the “funnel” (or tube) of possible outcomes (see Figs. 1 and 2) that the disturbances may drive the system to, rather than a single planned trajectory. Consequently, the key challenges for planning translate into the ability to guarantee both *safety* (with respect to

*{ssingh19, blandry, pavone}@stanford.edu

†ani.majumdar@princeton.edu

‡jjs@mit.edu

Jean-Jacques Slotine and Marco Pavone share senior co-authorship.

constraint satisfaction and collision avoidance) and *performance* (with respect to some cost function optimality) for the robotic system in cluttered and possibly dynamically changing environments.

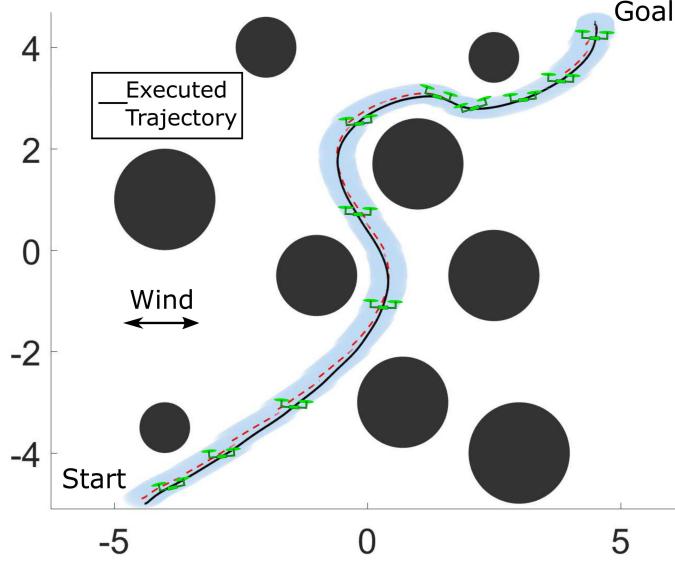


Figure 1: A planar quadrotor navigating in real-time through a previously unseen cluttered environment in the presence of a horizontal cross-wind disturbance. A nominal (disturbance-free) trajectory (dashed-red line) is generated online in response to obstacles reported in the environment such that the invariant tube (computed offline) centered around the trajectory does not intersect obstacles. The breakpoints in the tube mark the instances where the nominal path is *locally* re-optimized as it adjusts to the actual executed trajectory veering to the edges of the tube due to the cross-wind (but still remains within it as guaranteed by the tracking controller). The spacing between the edge of the tube and the obstacles accounts for the size of the vehicle itself.

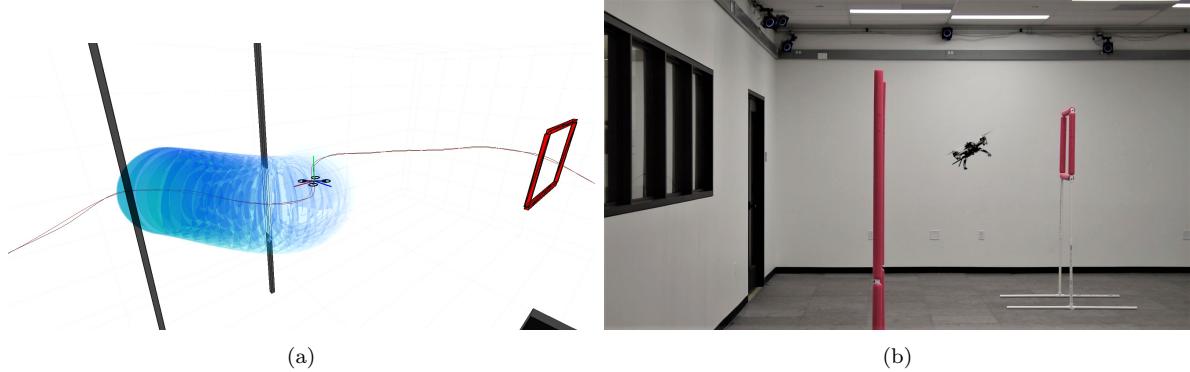


Figure 2: Illustration of the robust planning with invariant tubes methodology on a full 3D quadrotor in simulation and experiment.

1.1 Outline of Approach

The goal of this paper is to propose an approach for generating motion plans with certifiable *margins of safety*. In particular, in an *offline* stage one synthesizes the structure of a tracking controller which can be efficiently implemented online to guarantee exponential convergence to *any* feasible

nominal trajectory in the absence of disturbances. Additionally, the offline computation yields a fixed-size invariant “tube” (akin to a funnel) that can be centered around any nominal trajectory as a guaranteed collision-free envelope in the presence of bounded disturbances. In the *online* phase, when the robot is faced with obstacles, one can use such a tube as a robustness margin during collision checking, thus leading to nominal trajectories that can be safely executed. We stress that this paper does not propose a new motion planning algorithm. Instead, it proposes a framework for using a standard motion planner as a “black box” or “primitive routine,” combined with a feedback control design methodology, that together enable the online generation of safe nominal motion plans that can be robustly executed.

The key idea behind our approach is to leverage *contraction theory* (Lohmiller and Slotine, 1998), a method for analyzing nonlinear systems by studying convergence between *pairs of trajectories*. This makes it particularly well-suited to the problem we consider here since it does not require us to commit to a particular nominal trajectory in order to analyze the stability properties of a feedback controller designed to track it. In particular, we design tracking controllers by using *control contraction metrics* (CCMs) (Manchester and Slotine, 2017), a generalization of control Lyapunov functions that can be computed using convex optimization.

2 Problem Statement and Related Work

Consider robotic systems whose dynamics are described by the nonlinear differential equation:

$$\dot{x}(t) = f(x(t)) + B(x(t))u(t) + B_w(x(t))w(t), \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, and $w(t) \in \mathbb{R}^{n_w}$ is the disturbance. The function $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ captures the *drift*, $B : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ is the input matrix mapping, depicted in column-stacked form as (b_1, \dots, b_m) , and $B_w : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n_w}$ is the disturbance matrix mapping with $\bar{\sigma}(B_w(x)) = 1$ for all x , where $\bar{\sigma}(\cdot)$ denotes the maximum singular value. In other words, B_w simply selects the channels where the disturbance is active. A state-input trajectory satisfying (1) is denoted as a pair (x, u) . Additionally, we enforce state constraints (e.g., arising from obstacles in the robot’s environment and physical constraints such as joint limits) and input constraints, that is: $x(t) \in \mathcal{X}$ and $u(t) \in \mathcal{U}$ for all t , where \mathcal{X} and \mathcal{U} are defined to be the closures of bounded, open, and connected sets in Euclidean space.

The motion planning problem we wish to address is to find a (possibly non-stationary) policy $\pi : \mathcal{X} \times \mathbb{R} \rightarrow \mathcal{U}$ that (i) drives the state x to a compact region $\mathcal{X}_{\text{goal}} \subseteq \mathcal{X}$, (ii) satisfies the state and input constraints, and (iii) minimizes a quadratic cost:

$$J(x(t), \pi) := \int_0^{T_{\text{goal}}} 1 + \|\pi(x(t), t)\|_R^2 dt,$$

where $\|\cdot\|_R := \sqrt{(\cdot)^T R(\cdot)}$ denotes the weighted norm with respect to R , a strictly positive definite matrix, and T_{goal} is the first time $x(t)$ enters $\mathcal{X}_{\text{goal}}$.

The topic of planning under uncertainty has been approached from two general methodologies within the robotics community. In the first approach, one seeks probabilistic guarantees on safety (e.g., collision probabilities) for a stochastic model of uncertainty. This is elegantly described by the chance-constrained programming framework (Charnes and Cooper, 1959). Typical solution methods generally consider linear systems affected by Gaussian noise (Blackmore et al., 2006, 2011;

Ono et al., 2013) or more generally, exploit linear-Gaussian dependencies (Luders et al., 2016). The extension to nonlinear dynamics and/or non-Gaussian noise is inherently difficult; typical methods employ sampling (Monte Carlo) techniques (Janson et al., 2015b; Sun et al., 2015), stochastic Lyapunov theory (Battilotti and De Santis, 2003; Buehler et al., 2016), or Gaussian processes with (probabilistically) robust feedback linearization (Helwa et al., 2019). In the more general case with partial observability and/or noisy sensing, the stochastic formulation is lifted to the *belief space* planning framework (Kaelbling et al., 1998; Kurniawati et al., 2008; Prentice and Roy, 2009; Platt et al., 2012; Agha-mohammadi et al., 2014). This approach has been typically constrained by limitations such as Gaussian belief state assumptions, linear dynamics, and/or small state and action spaces, for instance in the Partially Observable Markov Decision Process (POMDP) framework.

In the second approach, in contrast to the stochastic methodology, one considers *bounded* models of uncertainty, where $w(t)$ is assumed to be piecewise \mathcal{C}^1 and norm-bounded, i.e., there exists some strictly positive real number \bar{w} such that $\|w(t)\|_2 \leq \bar{w}$ for all $t \geq 0$. This is typically described as the robust planning problem, and is the formulation adopted within this work. A fundamental component of solving this problem is based around computing and/or optimizing over reachable sets, or their *over*-approximations. Figure 3 depicts a rough classification of various methods for computing bounds on the reachable sets, ranked by the conservativeness of the approximation. The figure is by *no means* exhaustive and additional references are provided following the figure, but it is useful for understanding the overarching *categorization* of various methods, ranging from approximative linear analysis, through Lyapunov-based certificates computed using convex optimization, to formal verification using dynamic programming and differential games; see also the recent review (Bansal et al., 2017).

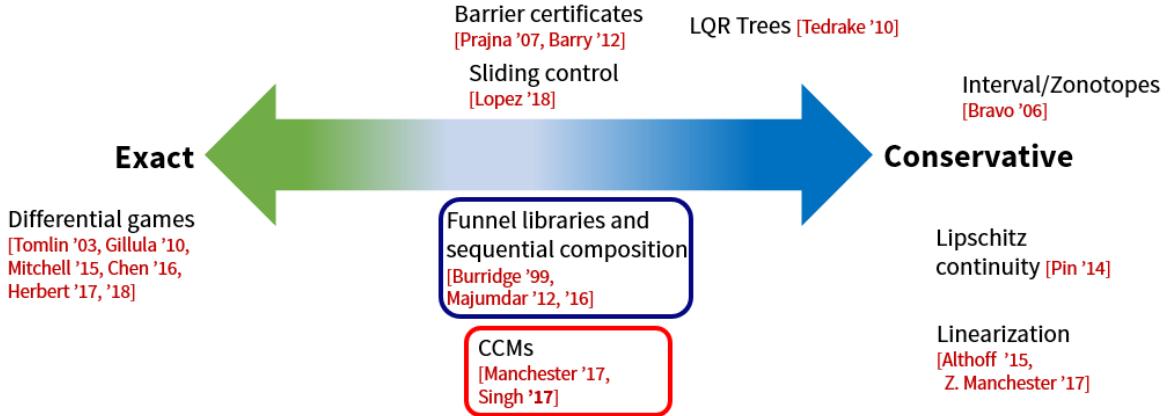


Figure 3: An overview of methods for computing exactly or outer-approximating the reachability set for nonlinear systems. The references in the blue-outline box constitute the primary motivation for our planning framework, i.e., sequential composition of invariant tubes. The red-outline box indicates the *method* for computing these tubes, i.e., via CCMs.

On the exact end of the spectrum, one may leverage logic-based methods (e.g., quantifier elimination) to recursively compute the backward reachable set of a given goal set (Raković et al., 2006a; Kong et al., 2015). In a similar vein, the differential game formulation treats any admissible disturbance as an adversarial agent and one can compute the backward reachable set of unsafe sets (e.g., obstacle locations) as the solution to a Hamilton-Jacobi PDE using level-set methods (Tomlin et al., 2003; Gillula et al., 2010; Chen et al., 2016; Herbert et al., 2017; Fridovich-Keil et al., 2018).

While these methods are exact in that they *precisely* characterize a collision-free “roadmap” to the goal set, and in the differential game formulation, also yield the optimal closed-loop controller, implementation of these methods for dynamically changing or unknown environments and/or high-dimensional systems is computationally prohibitive. A computational relaxation in this spirit is the barrier certificates method (Prajna et al., 2007; Barry et al., 2012) in which one characterizes the unsafe regions in the state-space as the zero-superlevel set of a function whose time derivative on the zero-level set is negative for a given controller and all admissible disturbances. While this yields a more conservative sufficient condition, the method is again incompatible with generalizing to unknown environments discovered in real-time.

As presented earlier, the motion planning problem entails an optimization over the class of state-feedback functions – a computationally intractable task in general with a solution that is incompatible with changing environments (e.g., obstacle locations). In an effort to reduce computational complexity, the prevailing solution approach, and indeed the strategy adopted within this work, is to parameterize general state-feedback policies as a sum of a nominal (open-loop) input u^* and a feedback term designed to track the nominal state trajectory x^* (induced by u^* assuming no disturbances):

$$\pi(x(t), t) = u^*(t) + k(x^*(t), x(t)), \quad (2)$$

where $k(\cdot, \cdot)$ is the feedback tracking controller. Commonly referred to as *feedback motion planning*, such a solution approach represents a compromise between the general class of state-feedback control laws and a purely open-loop formulation (i.e., no tracking). In order to ensure satisfaction of all constraints in the presence of disturbances, one needs to characterize the reachable set or “funnel” (Burridge et al., 1999), around the trajectory being tracked. Formally, this is defined through the notion of *robust control invariant* (RCI) tubes.

Suppose (x^*, u^*) is a state-input trajectory satisfying the nominal dynamics (i.e., (1) with $w \equiv 0$) and (x, u) is a state-input trajectory satisfying (1) under the action of a parameterized policy (2). Let T_{goal}^* be the first time $x^*(t)$ enters $\mathcal{X}_{\text{goal}}$. An RCI tube is defined as follows.

Definition 2.1 (RCI Tube). *Let $\Omega : \mathbb{R}^n \rightarrow 2^{\mathbb{R}^n}$ be a mapping s.t. $x \in \Omega(x)$ and $\Omega(x)$ is a closed and bounded set for every x . Then, $\Omega(\cdot)$ is an RCI mapping (additionally $\Omega(x)$ is an RCI set centered on x) if there exists a tracking controller $k(x^*, x)$ s.t. if $x(t_0) \in \Omega(x^*(t_0))$, then for all allowable realizations of the disturbance $w(t)$, $x(t) \in \Omega(x^*(t))$ for all $t_0 \leq t \leq T_{\text{goal}}^*$. Given an RCI mapping $\Omega(\cdot)$, an RCI tube centered on the trajectory $x^*(t)$, $t_0 \leq t \leq T_{\text{goal}}^*$, is the swept region $\bigcup_{t_0 \leq t \leq T_{\text{goal}}^*} \Omega(x^*(t))$.*

Intuitively, a tracking controller with an associated RCI tube $\Omega(\cdot)$ guarantees that the state of the system is always “close” to its nominal value $x^*(t)$ (precisely, within set $\Omega(x^*(t))$). Thus, by planning a nominal state-input trajectory satisfying the *tightened* constraints:

$$x^*(\cdot) \in \bar{\mathcal{X}} := \mathcal{X} \ominus \Omega, \quad (3a)$$

$$u^*(\cdot) \in \bar{\mathcal{U}} := \{\bar{u} \in \mathcal{U} : \forall x^*(t) \in \bar{\mathcal{X}}, \forall x(t) \in \mathcal{X} \text{ s.t. } x(t) \in \Omega(x^*(t)), \bar{u} + k(x^*(t), x(t)) \in \mathcal{U}\}, \quad (3b)$$

where \ominus denotes the Minkowski set difference, one can ensure that the robotic system will safely reach the goal region $\mathcal{X}_{\text{goal}}$ (modulo the size of the RCI set) in the presence of disturbances. Note that one can ensure that the system reaches $\mathcal{X}_{\text{goal}}$ (without the extra buffer from the RCI set) by constraining the RCI set to be contained within $\mathcal{X}_{\text{goal}}$ at T_{goal}^* . Constraint (3a) ensures that the

RCI tube around the trajectory does not intersect any obstacles, while (3b) is defined for a given tracking controller and ensures that the net applied control satisfies the input limit.

For fully-actuated (i.e., feedback linearizable) systems, RCI tubes or funnels may be computed and optimized using sliding control (Slotine, 2007; Lopez et al., 2018). More conservative approximations of these sets may be found using *linear* reachability analysis, where one computes continuous linearizations of the dynamics about reference trajectories and treats nonlinearities as bounded disturbances (Althoff and Dolan, 2014; Althoff et al., 2015). Alternatively, by bounding the dynamics’ Lipschitz constant or Jacobian, one obtains an exponentially growing outer approximation of the reachable set. For instance, in (Pin et al., 2009), a bound on the global Lipschitz constant is used to compute sequentially tightened constraint sets for a reference trajectory to ensure robust constraint satisfaction, while (Bravo et al., 2006) presents a general framework for computing outer approximations of reachable sets using zonotopes, which, while possessing favorable computational properties, again yield overly conservative approximations. The differential inequality approach in (Scott and Barton, 2013; Villanueva et al., 2017) attempts to alleviate the conservativeness resulting from such linear methods through online co-optimization of the reference trajectory and its associated reachable set. In similar spirit, (Manchester and Kuindersma, 2017, 2019) leverage a measure of the size of the approximate invariant funnels computed using linear analysis (i.e., propagation of ellipsoids under linearized dynamics) within the formulation of the cost function for the nominal trajectory itself. However, these problems are posed as one-off offline computations and not suitable for real-time re-planning on fast robotic systems. In general, treating nonlinearities as bounded disturbances naturally leads to overly conservative approximations.

Convex programming-based verification methods such as sum-of-squares (SOS) programming have gained increasing popularity in feedback motion planning. For instance, the LQR-Trees algorithm (Tedrake et al., 2010) constructs a tree of local LQR feedback controllers, however, it cannot handle scenarios in which the task and environment are unknown until runtime. Recently, the *funnel library* approach (Majumdar and Tedrake, 2012, 2017) has been proposed to handle online geometric constraints (e.g., obstacles) that force the robot to re-plan in real-time. The approach leverages SOS programming to compute, offline, a library of funnels around a set of nominal trajectories in which the state is *guaranteed* to remain despite bounded disturbances. These funnels are then sequentially composed online to avoid obstacles. However, this approach is restricted to employing a *fixed set* of trajectories computed offline. While the richness of the funnel library may be increased by exploiting invariances in the dynamics (Majumdar and Tedrake, 2017) or pre-computing a family of funnels parameterized by shifts to a nominal trajectory (Majumdar et al., 2012), one would ideally like to generate a funnel around *any* nominal trajectory generated online.

The concept of feedback motion planning is also fundamental within Tube Model Predictive Control (TMPC), whereby one computes a tracking feedback (also termed ancillary) controller that keeps the state within an invariant “tube” around the nominal MPC trajectory despite disturbances. TMPC has been studied extensively for linear systems with bounded disturbances or model uncertainties (Langson et al., 2004; Mayne et al., 2005; Limon et al., 2010; Farina and Scattolini, 2012; Rakovic et al., 2012), and for linear systems with stochastic disturbances (Fleming et al., 2015) (see also the recent review (Mayne, 2014)). The application of TMPC to nonlinear systems is certainly not new, see for instance (Raković, 2009) where the properties of TMPC for nonlinear systems are explored via lifting the analysis to set dynamics and employing the Banach fixed-point theorem. However, the construction of invariant tubes and the design of the associated ancillary controller in the nonlinear setup is significantly more complicated than in the linear case.

In (Raković, 2009) for instance, the existence of a stabilizing (nonlinear) ancillary controller (that results in contracting set iterates) is simply assumed, while in (Kögel and Findeisen, 2015), a static linear state feedback ancillary controller is used to stabilize the “linear” component of the nonlinear dynamics and Lipschitz continuity is used to bound the effect of disturbance propagation. Some techniques to construct the ancillary controller and accompanying invariant tube proposed in existing literature include systems with matched nonlinearities and linear ancillary feedback (Raković et al., 2006b); integral sliding mode ancillary feedback with Lipschitz-based reachability analysis (Rubagotti et al., 2011); ellipsoidal invariant tubes constructed using linear-matrix-inequalities (LMI) and bounds on the Lipschitz constant (Yu et al., 2013) or assuming a polytopic linear differential inclusion model for the dynamics (Yu et al., 2010); ancillary-MPC with Lipschitz-based reachability analysis (Mayne et al., 2011); systems with incrementally conic uncertainties stabilized using linear ancillary feedback (Carson III et al., 2013); and linearization with static or time-varying linear ancillary feedback with nonlinearities treated as bounded disturbances (Cannon et al., 2011). As with the robotic planning literature, methods leveraging linearization or Lipschitz constants are inherently overly conservative. The notion of incremental input-to-state stability (δ -ISS) for discrete-time systems was used in (Bayer et al., 2013) to derive the invariant tube as a sublevel set of the associated δ -ISS Lyapunov function, which was assumed to be given. More recently, (Köhler et al., 2019) leverages the assumed *existence* of such a function and implicitly incorporates constraint tightening via constraining the growth of this function (using a nonlinear dynamical representation of the scalar defining the Lyapunov sublevel sets) as part of the MPC optimization problem. Again, these bounding functions are either assumed given, or constructed assuming local linear feedback. In contrast, the work presented herein focuses on the *design and optimization* of the functions themselves (and the associated feedback controllers), subsequently permitting the use of algorithms such as the one presented in (Köhler et al., 2019).

2.1 Statement of Contributions

In this paper, we leverage recent advances in contraction theory for control design through the use of *control contraction metrics* (CCM) (Manchester and Slotine, 2017). Contraction theory (Lohmiller and Slotine, 1998) is a method of analyzing nonlinear systems in a differential framework, i.e., via the associated variational system (Crouch and van der Schaft, 1987, Chp 3), and is focused on the study of convergence between pairs of state trajectories towards each other. Thus, at its core, contraction explores a stronger notion of stability – that of incremental stability between solution trajectories, instead of the stability of an equilibrium point or invariant set. While the analysis in (Lohmiller and Slotine, 1998) and most other works on contraction theory focus on analyzing the stability of closed-loop vector fields – (Jouffroy, 2003; Sontag, 2010; Sontag et al., 2014; Simpson-Porco and Bullo, 2014; Forni and Sepulchre, 2014); see also the recent review in (Aminzarey and Sontag, 2014) and references therein, recent results demonstrate the applicability of contraction theory for *constructive control design*, e.g., control via backstepping (Sharma and Kar, 2009; Zamani et al., 2013), control for singularly perturbed systems using multiple time-scales (Rayguru and Kar, 2015), and control via CCMs (Manchester et al., 2015; Manchester and Slotine, 2017; Singh et al., 2017). The concept of δ -ISS has also been studied within a contraction theory framework, e.g., in (Zamani et al., 2013), where contraction metrics are derived for a class of nonlinear systems stabilized using backstepping. Compared to works on establishing incremental stability through suitable δ -ISS Lyapunov functions (Angeli, 2002, 2009), contraction metrics are an *intrinsic* characterization of incremental stability (i.e., coordinate invariant) and the search for a suitable metric and associated

stabilizing controller via convex optimization boasts obvious practical benefits.

The contributions of this paper are the following. First, on the theoretical side, while the CCM approach is directly inspired by (Manchester and Slotine, 2017), we present an alternative proof (Section 3.2) of incremental exponential stability between trajectories using a suitable CCM-derived controller. This proof employs techniques from calculus of variations and differential geometry in order to (i) derive a tighter characterization of the controller’s disturbance rejection properties and the size of the corresponding invariant tube (Section 3.5), and (ii) simplify the online implementation of the controller (Section 5.1), under significantly weaker conditions for the CCM as compared to (Manchester and Slotine, 2017) and (Zamani et al., 2013). Second, we derive a bound on the magnitude of the tracking controller by leveraging properties of CCMs and exploiting analogies with *differential* control Lyapunov functions (Section 5.3). Third, on the algorithmic side, we present an offline/online framework for robust motion planning where in the offline phase, we formulate a quasiconvex optimization problem using SOS programming that searches for an *optimal* CCM in order to minimize the cross-section of the invariant tube (Section 4), and derive the structure of a tracking controller that can be efficiently implemented online. The optimized tube is used online to obtain nominal motion plans that can be robustly tracked (Section 6). Fourth, on the experimental front, we illustrate our method and several variations on simulated 6-state planar-quadrotor and 10-state 3D quadrotor systems (Section 7). Finally, we present experimental results on a quadrotor hardware testbed by generating robust motion plans using sampling-based planning and polynomial smoothing for a cluttered lab environment and evaluate the performance of the quadrotor in the presence of aerodynamic disturbances induced by unmodeled drag forces (Section 8).

A preliminary version of this work was presented at ICRA 2017 (Singh et al., 2017). In this revised and extended version, we additionally present (i) full proofs for all technical results, (ii) a tutorial on CCM computation with illustrative examples, (iii) extensions to the online planning algorithm to leverage less conservative time-varying tubes, (iv) extensive additional numerical experiments, including failure analysis, and (v) validation on a hardware testbed.

Our approach has several advantages over prior work on robust motion planning and MPC. First, by explicitly enforcing safety in the online planning process, the method is particularly suited to planning in previously unseen and tightly-constrained environments, where it might be difficult to find feasible solutions by sequencing a pre-computed set of maneuvers (Fig. 1). Second, in contrast to the class of techniques that employ *linear* reachability analysis to conservatively approximate funnels/tubes for nonlinear systems by treating nonlinearities as bounded disturbances, our analysis directly reasons about intrinsic nonlinearities in the dynamics and thus has the potential to be less conservative for highly nonlinear systems. *Notably, our approach allows for both the design and optimization of invariant tubes and tracking feedback controllers.* Third, the method is computationally rooted in convex optimization, which in turn carries a smaller computational burden than differential game formulations for computing reachable tubes, which require numerical solutions to PDEs.

Notation: Let \mathbb{S}_j be the set of symmetric matrices in $\mathbb{R}^{j \times j}$ and denote $\mathbb{S}_j^{\geq 0}$, respectively $\mathbb{S}_j^{> 0}$, to be the set of symmetric positive semi-definite, respectively positive definite matrices in $\mathbb{R}^{j \times j}$. Given a matrix X , let $\widehat{X} := X + X^T$. The set of \mathcal{C}^2 functions from \mathcal{D} to \mathcal{R} is denoted by $\mathcal{C}^2(\mathcal{D}, \mathcal{R})$. We denote the components of a vector $y \in \mathbb{R}^n$ as $y[j]$, $j = 1, \dots, n$, and its Euclidean norm as $\|y\|$. Let $\|y\|_A = \sqrt{y^T A y}$ denote a weighted norm for $A \in \mathbb{S}_n^{> 0}$. Let $\bar{\sigma}(A)$ denote the maximum, and $\underline{\sigma}(A)$ the minimum, singular values of a matrix A . Let $\lambda(A)$ denote the maximum, and $\underline{\lambda}(A)$ the minimum, real eigenvalues of a symmetric matrix A . Let $\partial_y F(x)$ denote the Lie derivative of the

matrix-valued function F at x along the vector y . Finally, given sets \mathcal{A} and \mathcal{B} , the set $\mathcal{A} \ominus \mathcal{B}$ is the Minkowski difference.

3 Trajectory Tracking with Contraction Theory

In this section, we demonstrate how to derive trajectory tracking controllers with (i) exponential convergence properties in the absence of disturbances, and (ii) strong boundedness properties in the presence of *bounded* disturbances, by leveraging contraction theory. In particular, this section of the paper is concerned with the task of *robust nonlinear feedback control design*. The integration of the results of this section into a feedback planning algorithm will be the focus of Section 6. The core principle behind contraction theory (Lohmiller and Slotine, 1998) is to study the evolution of distance between any two *infinitesimally close* neighboring trajectories and draw conclusions on the distance between any *finitely apart* pair of trajectories. We begin by first introducing the fundamentals of contraction theory, in the absence of control.

3.1 Introduction to Contraction

Given an autonomous system of the form: $\dot{x}(t) = f(x(t))$, consider two neighboring trajectories separated by an infinitesimal (virtual) displacement δ_x ; formally, δ_x is a vector in the tangent space $T_x \mathcal{X}$ at x . The dynamics of this virtual displacement are given by:

$$\dot{\delta}_x = \frac{\partial f}{\partial x} \delta_x,$$

where $\partial f / \partial x$ is the Jacobian of f . The dynamics of the infinitesimal squared distance $\delta_x^T \delta_x$ between these two trajectories is then given by:

$$\frac{d}{dt} (\delta_x^T \delta_x) = 2\delta_x^T \frac{\partial f}{\partial x} \delta_x.$$

Then, if the (symmetric part) of the Jacobian matrix $\partial f / \partial x$ is *uniformly* negative definite, i.e.,

$$\sup_x \bar{\lambda} \left(\frac{1}{2} \widehat{\frac{\partial f(x)}{\partial x}} \right) \leq -\lambda < 0,$$

for some $\lambda \in \mathbb{R}_{>0}$, one has that the squared infinitesimal length $\delta_x^T \delta_x$ is exponentially convergent to zero at rate 2λ . By path integration of δ_x between *any* pair of trajectories, one has that the distance between any two trajectories shrinks exponentially to zero. The vector field f is thereby referred to be *contracting at rate λ* , and λ is referred to as the *contraction rate*.

Contraction metrics generalize this observation by considering an infinitesimal squared length distance, a symmetric positive definite function $V(x, \delta_x) = \delta_x^T M(x) \delta_x$, where $M : \mathcal{X} \rightarrow \mathbb{S}_n^{>0}$ is a mapping from \mathcal{X} to the set of uniformly positive definite $n \times n$ symmetric matrices. Formally, $M(x)$ may be interpreted as a Riemannian metric tensor, endowing the space \mathcal{X} with the Riemannian squared length element $V(x, \delta_x)$. A fundamental result in contraction theory (Lohmiller and Slotine, 1998) is that *any* contracting system admits a contraction metric $M(x)$ such that the associated function $V(x, \delta_x)$ satisfies:

$$\dot{V}(x, \delta_x) \leq -2\lambda V(x, \delta_x), \quad \forall (x, \delta_x) \in \mathcal{T}\mathcal{X},$$

for some positive contraction rate λ . Thus, the function $V(x, \delta_x)$ may be interpreted as a *differential Lyapunov function*; see also (Forni and Sepulchre, 2014) for such an analogy, albeit using Finsler metrics.

3.2 Control Contraction Metrics

Control contraction metrics (CCMs) generalize contraction analysis to the controlled dynamical setting, in the sense that the analysis searches *jointly* for a controller design and the metric that describes the contraction properties of the resulting closed-loop system. To introduce this concept, we first define the notion of incremental exponential stabilizability.

Definition 3.1 (Incremental Exponential Stability). *Consider a nominal state-input trajectory pair $(x^*(t), u^*(t))$ for the unperturbed controlled dynamics (i.e., (1) with $w \equiv 0$). Suppose there exist $\lambda, C > 0$ and a feedback controller $k(x^*(t), x(t))$ such that the resulting state trajectory $x(t)$ for the unperturbed dynamics with control $u(x(t)) = u^*(t) + k(x^*(t), x(t))$ satisfies*

$$\|x^*(t) - x(t)\| \leq Ce^{-\lambda t} \|x^*(0) - x(0)\|. \quad (4)$$

Then, the trajectory $x^(t)$ is said to be incrementally exponentially stabilizable (IES) with rate λ and overshoot constant C .*

We now illustrate how to leverage contraction theory to derive trajectory tracking controllers that guarantee IES for any nominal state trajectory $x^*(t)$. Denote the tangent space of \mathcal{X} at $x \in \mathcal{X}$ by $T_x \mathcal{X}$ ¹ and the tangent bundle of \mathcal{X} by $T\mathcal{X} = \bigcup_{x \in \mathcal{X}} \{x\} \times T_x \mathcal{X}$. The variational dynamics (i.e., dynamics of the virtual displacement δ_x along any nominal state-input trajectory $(x(t), u(t))$) for the unperturbed controlled system are given by (Crouch and van der Schaft, 1987, Chp 3):

$$\dot{\delta}_x = \overbrace{\left(\frac{\partial f(x)}{\partial x} + \sum_{j=1}^m u[j] \frac{\partial b_j(x)}{\partial x} \right) \delta_x + B(x) \delta_u}^{:= A(x, u)}, \quad (5)$$

where $\delta_x \in T_x \mathcal{X}$ is a tangent vector to a smooth path of states at $x \in \mathcal{X}$, and $\delta_u \in T_u \mathcal{U}$ is a tangent vector to a smooth path of controls at $u \in \mathcal{U}$. Let $M : \mathcal{X} \rightarrow \mathbb{S}_n^{>0}$ be a smooth matrix function that is uniformly bounded (i.e., there exist constants $0 < \underline{\alpha} < \bar{\alpha}$ such that $\underline{\alpha}I_n \preceq M(x) \preceq \bar{\alpha}I_n$). Continuing the interpretation of $V(x, \delta_x) = \delta_x^T M(x) \delta_x$ as a Riemannian squared differential length element, for a given smooth curve $c : [0, 1] \rightarrow \mathcal{X}$, we define its length $l(c)$ and energy $\mathcal{E}(c)$ as $l(c) := \int_0^1 \sqrt{V(c(s), c_s(s))} ds$, $\mathcal{E}(c) := \int_0^1 V(c(s), c_s(s)) ds$, where $c_s(s) = \partial c(s)/\partial s$.

Let $\Gamma(p, q)$ be the set of smooth curves on \mathcal{X} that connect points p and q . The Riemann distance between points p and q is defined as the quantity $d(p, q) := \inf_{c \in \Gamma(p, q)} l(c)$, and denote $\mathcal{E}(p, q) := d^2(p, q)$ to be the Riemann energy. Let the curve $\gamma \in \Gamma(p, q)$ be the (possibly non-unique) *minimizing geodesic* which achieves this infimum. Notice that $\mathcal{E}(\gamma) = d^2(p, q)$.

A smooth, uniformly positive definite matrix-valued function $M(x)$ is a CCM for the system $\dot{x} = f(x) + B(x)u$, if there exists a *differential controller* $\delta_u : T\mathcal{X} \rightarrow T_u \mathcal{U}$, such that $\dot{V}(x, \delta_x) < 0$, $\forall (x, \delta_x) \in T\mathcal{X}$. Thus, $V(x, \delta_x)$ may be interpreted as a *differential control Lyapunov function* (CLF) on the tangent bundle $T\mathcal{X}$. In the next section, we study the necessary and sufficient conditions for the *existence* of CCMs.

¹Since \mathcal{X} is the closure of an open set in \mathbb{R}^n , the tangent space $T_x \mathcal{X}$ for all x in the interior of \mathcal{X} is simply \mathbb{R}^n , while $T_x \mathcal{X}$ on the boundary of \mathcal{X} is a half-space in \mathbb{R}^n .

3.3 Conditions for CCMs

Suppose that the following equality holds for all $x \in \mathcal{X}$:

$$\partial_{b_j} M(x) + M(x) \widehat{\frac{\partial b_j(x)}{\partial x}} = 0, \quad j = 1, \dots, m. \quad (6)$$

Condition (6) implies that the vectors b_j form a Killing vector field for the metric tensor $M(x)$. Under this condition, \dot{V} reduces to

$$\dot{V}(x, \delta_x, \delta_u) = \delta_x^T \left(\partial_f M(x) + M(x) \widehat{\frac{\partial f(x)}{\partial x}} \right) \delta_x + 2\delta_x^T M(x) B(x) \delta_u. \quad (7)$$

Then, if the following property holds for some constant $\lambda > 0$ and all $(x, \delta_x) \in T\mathcal{X}$:

$$\begin{aligned} \delta_x^T \left(\partial_f M(x) + M(x) \widehat{\frac{\partial f(x)}{\partial x}} \right) \delta_x &\leq -2\lambda \delta_x^T M(x) \delta_x \\ \text{for all } \delta_x \text{ such that } \delta_x^T M(x) B(x) &= 0, \end{aligned} \quad (8)$$

then (Manchester and Slotine, 2017) shows that there always exists an integrable *differential* feedback controller of the form $\delta_u(x, \delta_x) = K(x)\delta_x$ such that the following inequality holds for all² $(x, \delta_x) \in \mathcal{X} \times \mathbb{R}^n$:

$$\begin{aligned} \dot{V}(x, \delta_x) &= \delta_x^T \left(\partial_f M(x) + M(x) \widehat{\frac{\partial f(x)}{\partial x}} + \widehat{M(x)B(x)K(x)} \right) \delta_x \\ &\leq -2\lambda \delta_x^T M(x) \delta_x = -2\lambda V(x, \delta_x). \end{aligned} \quad (9)$$

Notice that condition (8) simply indicates that for all directions where the *variational* system lacks controllability (given by the nullspace of $B^T(x)M(x)$), the system is naturally contracting with rate λ . In the scenario where (6) fails to be true, one may leverage the following weaker alternative to conditions (6) and (8):

$$\begin{aligned} \delta_x^T \left(\partial_{f+Bu} M(x) + \widehat{M(x)A(x, u)} \right) \delta_x &\leq -2\lambda \delta_x^T M(x) \delta_x \\ \text{for all } \delta_x \text{ such that } \delta_x^T M(x) B(x) &= 0. \end{aligned} \quad (10)$$

In (Manchester and Slotine, 2017), it is proven, by construction, that the weaker condition above is still sufficient to guarantee the existence of a differential feedback controller. However, in this case, the differential controller becomes a function of u as well, i.e., $\delta_u = \delta_u(x, \delta_x, u)$.

We now illustrate how to leverage such a differential controller, which guarantees stability on an infinitesimal scale by the property $\dot{V}(x, \delta_x) \leq -2\lambda V(x, \delta_x)$ along any nominal trajectory, to derive a stabilizing controller for any pair of finitely-apart trajectories.

²Note that we drop the distinction between $T_x\mathcal{X}$ at the boundary versus the interior and simply assume that $T_x\mathcal{X} = \mathbb{R}^n$.

3.4 Incrementally Stabilizing Controllers

Given a desired nominal state-input trajectory pair $(x^*(t), u^*(t))$, let $\gamma(\cdot, t) : [0, 1] \rightarrow \mathcal{X}$ denote a minimizing geodesic connecting $x^*(t)$ and $x(t)$. Consider the following control law:

$$\begin{aligned}\pi(x(t), t) &= u^*(t) + \int_{\gamma(\cdot, t)} \delta_u(\gamma(s, t), \delta_\gamma(s, t)) ds \\ &= u^*(t) + \underbrace{\int_0^1 K(\gamma(s, t)) \delta_\gamma(s, t) ds}_{= k(x^*(t), x(t))},\end{aligned}\quad (11)$$

where $\delta_\gamma(s, t) := \partial \gamma(s, t) / \partial s$. The geometric interpretation of (11) and inequality (9) is illustrated in Fig. 4.

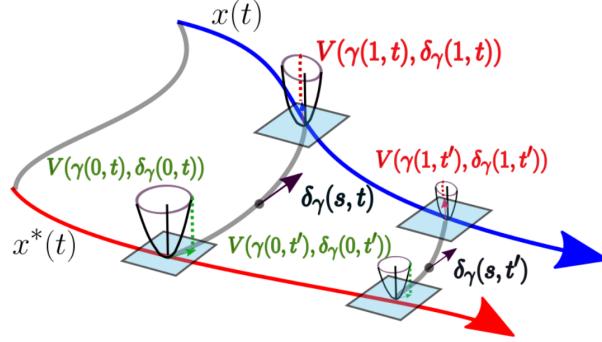


Figure 4: Schematic of the differential CLF $V(x, \delta_x)$ at the endpoints of the geodesics $\gamma(\cdot, t)$ and $\gamma(\cdot, t')$, at times t and $t' > t$ respectively, and the geodesic velocity vector at the position $s \in (0, 1)$ along the geodesic. The contours of V are shaped according to the metric tensor $M(x)$. The differential controller ensures that at all points along the geodesic, $V(x, \delta_x)$ is shrinking in the direction tangent to the geodesic.

In case the weaker condition (10) is used, $\delta_u = \delta_u(x, \delta_x, u)$, and thus $\pi(x(t), t)$ will be given by the solution to the differential equation

$$\pi(x(t), t) = u^*(t) + \int_{\gamma(\cdot, t)} \delta_u(\gamma(s, t), \delta_\gamma(s, t), \pi(\gamma(s, t), t)) ds,\quad (12)$$

where $\delta_u(\cdot)$ is designed such that $\dot{V}(x, \delta_x, u) \leq -2\lambda V(x, \delta_x)$ for all (x, δ_x) along the minimizing geodesic. Theorem 3.2, which is central to the approach, proves that control law (11) ensures the trajectory $x(t)$ is IES with respect to $x^*(t)$ in the sense of Definition 3.1. The proof differs significantly from the one presented in (Manchester and Slotine, 2017) as it is later adapted in the next section for deriving the RCI mapping.

Theorem 3.2 (Incrementally stabilizing controller). *Let (x^*, u^*) and (x, u) be state-input trajectory pairs for the nominal dynamics where $\pi(x(t), t)$ is given by (11) using a CCM which satisfies equations³ (6) and (8). Then, $x(t)$ is IES with respect to $x^*(t)$ in the sense of Definition 3.1.*

Proof. Notice that at each time instant t , (11) implicitly defines a smooth virtual parameterized surface of solutions $c_t : (s, t') \in [0, 1] \times (-\varepsilon, \varepsilon) \mapsto \mathcal{X}$ (where ε is an arbitrarily small positive value),

³Alternatively, $\pi(x(t), t)$ is given by (12) using a CCM satisfying the weaker condition (10).

to the nominal dynamics as follows:

$$c'_t(s, t') := \frac{\partial c_t}{\partial t'}(s, t') = f(c_t(s, t')) + B(c_t(s, t'))\pi_t(c_t(s, t'), t'), \quad c_t(\cdot, 0) = \gamma(\cdot, t), \quad (13)$$

where

$$\pi_t(c_t(s, t'), t') = u^*(t + t') + \int_0^s \delta_u(c_t(\xi, t'), \delta_{c_t}(\xi, t')) d\xi, \quad (14)$$

where $\delta_{c_t}(\xi, t') = \partial c_t(\xi, t')/\partial \xi$. Consider the time derivative of the Riemann energy $\mathcal{E}(x^*(t), x(t))$ between $x^*(t)$ and $x(t)$. If $x(t) \notin \text{Cut}(x^*(t))$ (where $\text{Cut}(x)$ denotes the cut-locus at x), then the derivative of $\mathcal{E}(x^*(t), x(t))$ is well-defined at t (i.e., the exponential map is a local diffeomorphism) and given by (Spivak, 1999, Chp 9):

$$\dot{\mathcal{E}}(x^*(t), x(t)) = \left[\left(\frac{\partial V}{\partial \delta_x}(\gamma(s, t), \delta_\gamma(s, t)) \right)^T \frac{\partial \gamma(s, t)}{\partial t} \right] \Big|_{s=0}^{s=1}. \quad (15)$$

This result is true for all $t \geq 0$ such that $x(t) \notin \text{Cut}(x^*(t))$. For completeness, suppose now that $x(t) \in \text{Cut}(x^*(t))$, a Lebesgue measure zero set. In this scenario, the minimizing geodesic $\gamma(\cdot, t)$ may not be unique and thus the Riemannian energy $\mathcal{E}(x^*(t), x(t))$ is *not* smooth. However, it can be shown that the upper Dini derivative of $\mathcal{E}(x^*(t), x(t))$, defined as:

$$D^+ \mathcal{E}(x^*(t), x(t)) := \limsup_{t' \searrow t} \frac{\mathcal{E}(x^*(t'), x(t')) - \mathcal{E}(x^*(t), x(t))}{t' - t}$$

exists, and satisfies the inequality⁴ (Adelstein and Epstein, 2017):

$$D^+ \mathcal{E}(x^*(t), x(t)) \leq \left[\left(\frac{\partial V}{\partial \delta_x}(\gamma(s, t), \delta_\gamma(s, t)) \right)^T \frac{\partial \gamma(s, t)}{\partial t} \right] \Big|_{s=0}^{s=1}. \quad (16)$$

Notice that the parameterized set of solutions in (13) satisfies: (i) $c_t(\cdot, 0) = \gamma(\cdot, t)$, (ii) $\delta_{c_t}(0, 0) = \delta_\gamma(0, t)$ and $\delta_{c_t}(1, 0) = \delta_\gamma(1, t)$, and (iii) $\partial \gamma(0, t)/\partial t = c'_t(0, 0) = \dot{x}^*(t)$ and $\partial \gamma(1, t)/\partial t = c'_t(1, 0) = \dot{x}(t)$. Thus, we have the following equality

$$\begin{aligned} \left[\left(\frac{\partial V}{\partial \delta_x}(\gamma(s, t), \delta_\gamma(s, t)) \right)^T \frac{\partial \gamma(s, t)}{\partial t} \right] \Big|_{s=0}^{s=1} &= \left[\left(\frac{\partial V}{\partial \delta_x}(c_t(s, 0), \delta_{c_t}(s, 0)) \right)^T c'_t(s, 0) \right] \Big|_{s=0}^{s=1} \\ &= \int_0^1 \frac{\partial}{\partial s} \left(\left(\frac{\partial V}{\partial \delta_x}(c_t(s, 0), \delta_{c_t}(s, 0)) \right)^T c'_t(s, 0) \right) ds. \end{aligned} \quad (17)$$

Now since $\gamma(\cdot, t)$ (and thus $c_t(\cdot, 0)$) is a geodesic, it satisfies the Euler-Lagrange equation, which in coordinates reads as:

$$\frac{\partial}{\partial s} \left(\frac{\partial V}{\partial \delta_x}(c_t(s, 0), \delta_{c_t}(s, 0)) \right) [i] = 2 \frac{\partial}{\partial s} \left(\sum_{j=1}^n M_{ij}(c_t(s, 0)) \delta_{c_t}(s, 0)[j] \right) = \delta_{c_t}(s, 0)^T \frac{\partial M}{\partial x[i]}(c_t(s, 0)) \delta_{c_t}(s, 0),$$

⁴In actual fact, it is shown in (Adelstein and Epstein, 2017) that the one-sided derivative of $\mathcal{E}(x^*(t), x(t))$ along any tangent vectors at $x^*(t)$ and $x(t)$ (defined using $\lim_{t' \searrow t} (\cdot)$) exists, from which it follows that the limit $\limsup_{t' \searrow t} (\cdot)$ exists.

for all $i \in \{1, \dots, n\}$ and $s \in (0, 1)$. By definition

$$\frac{\partial c'_t(s, t')}{\partial s} = A(c_t(s, t'), \pi(c_t(s, t'), t))\delta_{c_t}(s, t') + B(c_t(s, t'))\delta_u(s, t'),$$

i.e., the variational dynamics. Leveraging this equality with the Euler-Lagrange equation, one obtains

$$\begin{aligned} & \frac{\partial}{\partial s} \left(\left(\frac{\partial V}{\partial \delta_x}(c_t(s, 0), \delta_{c_t}(s, 0)) \right)^T c'_t(s, 0) \right) \\ &= \delta_{c_t}(s, 0)^T \left(\partial_{c'_t(s, 0)} M(s, 0) + \widehat{M(s, 0)A(s, 0)} \right) \delta_{c_t}(s, 0) + 2\delta_{c_t}(s, 0)^T M(s, 0)B(s, 0)\delta_u(s, 0) \\ &= V'(c_t(s, 0), \delta_{c_t}(s, 0), \delta_u(s, 0)) =: \frac{\partial V}{\partial t'}(c_t(s, 0), \delta_{c_t}(s, 0), \delta_u(s, 0)), \end{aligned}$$

where the last line follows by definition, and we use the abbreviation $(s, 0)$ for sake of clarity. Now by deliberate design of δ_u , we have:

$$V'(c_t(s, 0), \delta_{c_t}(s, 0), \delta_u(s, 0)) \leq -2\lambda V(c_t(s, 0), \delta_{c_t}(s, 0)),$$

for all $s \in [0, 1]$. Combining the inequality above and equations (16) and (17), one obtains the following chain of inequalities:

$$\begin{aligned} D^+ \mathcal{E}(x^*(t), x(t)) &\leq \int_0^1 V'(c_t(s, 0), \delta_{c_t}(s, 0), \delta_u(s, 0)) \, ds \\ &\leq -2\lambda \int_0^1 V(c_t(s, 0), \delta_{c_t}(s, 0)) \, ds \\ &= -2\lambda \int_0^1 V(\gamma(s, t), \delta_\gamma(s, t)) \, ds = -2\lambda \mathcal{E}(x^*(t), x(t)). \end{aligned} \tag{18}$$

It follows that $\mathcal{E}(x^*(t), x(t)) \leq \mathcal{E}(x^*(0), x(0))e^{-2\lambda t}$. Since $d^2(x^*(t), x(t)) = \mathcal{E}(x^*(t), x(t))$ and the metric tensor $M(x)$ is uniformly bounded, one concludes

$$\|x(t) - x^*(t)\| \leq \sqrt{\frac{\bar{\alpha}}{\underline{\alpha}}} \|x(0) - x^*(0)\| e^{-\lambda t}.$$

Thus, $x^*(t)$ is IES with rate λ and overshoot $\sqrt{\bar{\alpha}/\underline{\alpha}}$. \square

The construction of the *virtual* parameterized set of solutions induced by (13) is needed as one *cannot* directly reason about the evolution of the minimizing geodesic. Instead, we leverage the Euler-Lagrange equation to relate the time derivative of the Riemann energy between the trajectories $x^*(\cdot)$ and $x(\cdot)$ and the Lie derivative of the metric tensor $M(x)$ with respect to the nominal dynamics. Thus, the Riemannian energy between $x^*(t)$ and $x(t)$ may be viewed as an *incremental* CLF, thereby defining the structure of the feedback controller as any piecewise continuous (in time) function that satisfies inequality (18).

Remark 3.3. While the interpretation of the energy as an incremental CLF is an observation also made in (Manchester and Slotine, 2017), by showing equivalence between the right hand sides of (16), (17), and (18) and using this to establish IES for $x^*(t)$, we are able to: (i) derive the

robustness guarantees for controller (11) under significantly weaker conditions than those necessary in (Manchester and Slotine, 2017); we explore this distinction in greater detail in Section 3.5, and (ii) derive a significantly simpler form of the controller than the integral construction in eq. (11); see Section 5.1.

Remark 3.4. *It can be shown that conditions (6) and (8) are invariant under state diffeomorphism and metric pushforward (Manchester and Slotine, 2017). This invariance allows one to relax the topological assumptions on the state space \mathcal{X} . In particular, one may take \mathcal{X} to be any embedded smooth n -manifold in \mathbb{R}^k where $k \geq n$, as long as the contraction conditions hold on one (and thus, any) choice of local coordinates in \mathbb{R}^n . This highlights the coordinate-free (i.e., intrinsic) nature of CCM-based stabilization.*

3.5 Contraction-Based Tubes

We now deduce the disturbance rejection properties of the feedback controller given by (11) and derive the resulting RCI mapping for the closed-loop system stabilized with this controller. Henceforth, $(x^*(t), u^*(t))$ is assumed to satisfy the unperturbed dynamics, while $x(t)$ denotes the actual state trajectory (i.e., with disturbances) using the control law $u^*(t) + k(x^*, x)$, where $k(x^*, x)$ is a CCM-derived tracking controller.

Theorem 3.5 (Disturbance Rejection). *Assume there exists a CCM $M(x)$ satisfying conditions (6) and (8) that is uniformly bounded, i.e., $\underline{\alpha}I_n \preceq M(x) \preceq \bar{\alpha}I_n$, for all $x \in \mathcal{X}$ where $\underline{\alpha} > 0$. Factorize $M(x)$ as $\Theta(x)^T\Theta(x)$ and define*

$$\bar{\alpha}_w := \sup_{x \in \mathcal{X}} \bar{\sigma}(\Theta(x)B_w(x)).$$

Then, the geodesic energy between trajectories $x(t)$ and $x^(t)$, i.e., $\mathcal{E}(x^*(t), x(t))$, satisfies the differential inequality:*

$$D^+\mathcal{E}(x^*(t), x(t)) \leq -2\lambda\mathcal{E}(x^*(t), x(t)) + 2d(x^*(t), x(t))\bar{\alpha}_w \|w(t)\|. \quad (19)$$

Proof. Inequality (16) implies:

$$\begin{aligned} D^+\mathcal{E}(x^*(t), x(t)) &\leq 2\delta_\gamma^T(1, t)M(\gamma(1, t)) \left[f(x(t)) + B(x(t))u(x(t)) \right] \\ &\quad - 2\delta_\gamma^T(0, t)M(\gamma(0, t)) \left[f(x^*(t)) + B(x^*(t))u^*(t) \right] \\ &\quad + 2\delta_\gamma^T(1, t)M(\gamma(1, t))B_w(x(t))w(t), \end{aligned} \quad (20)$$

where $u(x(t))$ is given by (11). By the IES property of the nominal system, i.e., inequality (18), the expression above can be bounded as:

$$D^+\mathcal{E}(x^*(t), x(t)) \leq -2\lambda\mathcal{E}(x^*(t), x(t)) + 2\delta_\gamma^T(1, t)M(\gamma(1, t))B_w(x(t))w(t).$$

Defining $\delta_z(s, t) := \Theta(\gamma(s, t))\delta_\gamma(s, t)$, we obtain

$$D^+\mathcal{E}(x^*(t), x(t)) \leq -2\lambda\mathcal{E}(x^*(t), x(t)) + 2\delta_z^T(1, t)\Theta(\gamma(1, t))B_w(x(t))w(t).$$

Recall that the velocity field of a geodesic is *parallel* along the geodesic (Spivak, 1999) and thus $V(\gamma(s, t), \delta_\gamma(s, t)) = \mathcal{E}(x^*(t), x(t))$ for all $s \in [0, 1]$. This implies that $\sqrt{V(\gamma(s, t), \delta_\gamma(s, t))} = \|\delta_z(s, t)\| = d(x^*(t), x(t))$. Using the Cauchy-Schwarz inequality one obtains the stated differential inequality. \square

Remark 3.6. Notice that the differential inequality derived in (19) may be integrated in time to yield a δ -ISS statement similar to Definition 2.3 in (Zamani et al., 2013). Specifically, in (Zamani et al., 2013), the authors prove δ -ISS by leveraging a decrescent condition similar to inequality (9) with respect to tangent vectors δ_w defined on the tangent space of the disturbance manifold \mathcal{W} , yielding a differential ISS condition. The key difference here then is by leveraging the separability of the disturbance term (i.e., inequality (20)), and by using a tracking controller that enforces contraction along the minimizing geodesic at each time (see Fig. 4), we do not need to impose similar differential ISS like conditions on $V(x, \delta_x)$. We further note that a similar bound is also proved in (Manchester and Slotine, 2017) but under stronger conditions, which resemble the differential ISS conditions in (Zamani et al., 2013). Indeed, the relaxation of such conditions for Theorem 3.5 is the primary motivation for our alternative proof strategy for Theorem 3.2.

Notice that the equality in (19) yields the well known Bernoulli differential equation. Then, since $\|w(t)\| \leq \bar{w}$ for all t , if $d(x^*(0), x(0)) \in (0, \bar{\alpha}_w \bar{w}/\lambda]$, it follows by the Comparison Lemma (Khalil, 2002), that the geodesic distance is upper bounded by $\bar{\alpha}_w \bar{w}/\lambda$, for all time $t \geq 0$. Thus, the RCI mapping may be expressed as:

$$\Omega(x^*) = \{x \in \mathcal{X} : d(x^*, x) \leq \bar{\alpha}_w \bar{w}/\lambda := \bar{d}\}, \quad (21)$$

Notice that the mapping above is given using the Riemann distance which may depend upon a spatially varying metric. In order to efficiently plan a nominal trajectory whose associated RCI tube does not collide with obstacles, we would prefer a mapping that is *independent* of x^* . To this end, consider the following technical lemma.

Lemma 3.7. (Geodesic Boundedness) Consider points $x^* \in \bar{\mathcal{X}}, x \in \mathcal{X}$ s.t. $x \in \Omega(x^*)$ where $\Omega(x^*)$ is given in (21). Suppose the CCM $M(x)$ satisfies $M(x) \succeq \underline{M}$ for all $x \in \mathcal{X}$, where $\underline{M} \succeq \underline{\alpha} I_n$. Define the ellipsoid

$$\tilde{\Omega}(x^*) := \{x \in \mathcal{X} : \|x - x^*\|_{\underline{M}}^2 \leq \bar{d}^2\}, \quad (22)$$

and suppose $\tilde{\Omega}(x^*) \subset \mathcal{X}$. Then, the minimizing geodesic γ is contained entirely within $\tilde{\Omega}(x^*)$, i.e., $\gamma(s) \in \tilde{\Omega}(x^*)$ for all $s \in [0, 1]$, and thus $\Omega(x^*) \subseteq \tilde{\Omega}(x^*)$.

Proof. Consider the following chain of inequalities:

$$\begin{aligned} \mathcal{E}(x^*, x) &= \int_0^1 \delta_\gamma(s)^T M(\gamma(s)) \delta_\gamma(s) ds \\ &\geq \int_0^1 \delta_\gamma(s)^T \underline{M} \delta_\gamma(s) ds \\ &\geq \|x - x^*\|_{\underline{M}}^2, \end{aligned}$$

where the first inequality follows from the Lemma assumptions, and the second inequality follows from the fact that $\underline{M} \in \mathbb{S}_n^{>0}$ defines a flat Riemannian metric under which geodesics are straight lines. Thus, we obtain the implication:

$$\mathcal{E}(x^*, x) \leq \bar{d}^2 \Rightarrow \|x - x^*\|_{\underline{M}}^2 \leq \bar{d}^2.$$

To complete the proof we note that since γ is a minimizing geodesic, it follows that $d(x^*, \gamma(s)) = sd(x^*, x)$ for all $s \in [0, 1]$. \square

Equation (22) gives an ellipsoidal outer approximation of the RCI tube as defined in (21), and is thus independent of x^* (see Figure 5 for an illustration). This is essential for two reasons: (i) it drastically simplifies collision checking with respect to obstacles by avoiding geodesic computations, and (ii) the tightened state constraint set $\bar{\mathcal{X}}$ in problem MPC must be taken to be $\mathcal{X} \ominus \tilde{\Omega}$ to ensure that the minimizing geodesic lies within \mathcal{X} , i.e., where the CCM conditions hold.

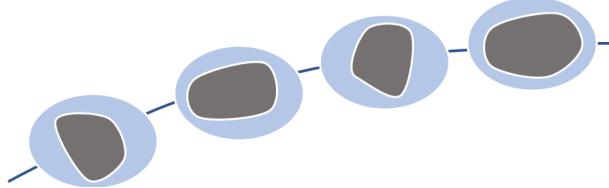


Figure 5: Illustration of the sets $\Omega(x^*)$ and $\tilde{\Omega}(x^*)$ along a trajectory $x^*(t)$. Due to the spatially varying $M(x)$, the set $\Omega(x^*)$ (shaded grey) continuously changes shape along the trajectory, making rapid collision checking difficult. The outer ellipsoidal approximation (shaded blue) $\tilde{\Omega}(x^*)$ is a fixed-size, easier to use collision margin.

So far we have shown how the *existence* of a feasible CCM $M(x)$ allows us to construct exponentially stabilizing controllers with bounded-input-bounded-output disturbance rejection guarantees. In the next section, we demonstrate how one can leverage convex optimization techniques, specifically SOS programming, to compute offline, an *optimized* CCM that minimizes the size of the RCI tube and the outer ellipsoidal approximation.

4 Offline Synthesis of Optimized Contraction-Based Tubes

In this section we show how to compute CCMs *offline* that *minimize* a certain measure of the size of the RCI set which in turn minimizes the deviation of the perturbed trajectory from the nominal, thereby reducing the amount by which we must tighten the set \mathcal{X} . Specifically, we demonstrate how to transform the CCM conditions (6), (8), and (10) into convex constraints, and formulate the synthesis problem as a quasiconvex optimization problem with appropriate objective functions. Additionally, we illustrate the resulting methodology with a representative example. We begin first with a brief review of SOS programming.

4.1 Sum-of-Squares Programming

Our computational approach is rooted in SOS programming, for which we provide a brief review here. For a more detailed review of SOS programming and its applications, please refer to (Parrilo, 2000; Ahmadi and Majumdar, 2016; Majumdar and Tedrake, 2017). We begin by discussing semi-definite programs (SDPs), a class of convex optimizations problems formulated over the space of symmetric positive semi-definite matrices. A symmetric matrix $X \in \mathbb{S}_n^{\geq 0}$ is positive semi-definite (psd) if $x^\top X x \geq 0$ for all $x \in \mathbb{R}^n$, $x \neq 0$, and is denoted as $X \succeq 0$. An SDP in standard form is

written as:

$$\begin{aligned} & \underset{\substack{X \in \mathbb{S}_n^{\geq 0}}}{\text{minimize}} \quad \text{Tr}(C^\top X) \\ & \text{subject to} \quad A_i^\top X = b_i, \quad i = 1, 2, \dots, m, \\ & \quad X \succeq 0 \end{aligned} \tag{23}$$

where C and $\{A_i\}_i$ are elements of \mathbb{S}_n , and ‘‘Tr’’ denotes the trace operator. SOS programs provide a means of certifying nonnegativity of polynomials either globally or over basic semialgebraic sets. A basic semialgebraic set \mathcal{S} is a subset of the Euclidean space characterized by a finite set of polynomial inequalities and equalities, that is,

$$\mathcal{S} := \{x \in \mathbb{R}^n : \phi_i(x) \geq 0, \psi_j(x) = 0\}, \tag{24}$$

where $\{\phi_i\}, \{\psi_j\}$ are multivariate polynomials in x . The simplest task within this class of problems is verifying the nonnegativity of a given polynomial p over \mathcal{S} , already NP-hard (Parrilo, 2000). SOS programming provides a convex relaxation approach to accomplish this task. Specifically, a polynomial p is termed SOS if it can be written in the form $\sum_k z_k^2$ for some other polynomials z_k . While such a decomposition is not necessary, it is sufficient to guarantee (global) nonnegativity of p . Moreover, if one can find a set of SOS polynomials L_i and ordinary polynomials q_j such that

$$p - \sum_i L_i \phi_i + \sum_j q_j \psi_j \quad \text{is SOS}, \tag{25}$$

then one obtains a *certificate* of nonnegativity of $p(x)$ over \mathcal{S} . Indeed, in the above equation when $\phi_i(x) \geq 0$ and $\psi_j(x) = 0$, i.e., $x \in \mathcal{S}$, one has that $p(x) \geq \sum_i L_i(x) \phi_i(x) \geq 0$, as required. Such a certificate is the extension of the generalized S-procedure (Iwasaki and Hara, 2005) to the setting of real-valued polynomials (Parrilo, 2000), and additionally constitutes a necessary condition for a subclass of semialgebraic sets (Putinar, 1993). More complex necessary and sufficient conditions for verifying nonnegativity over any semialgebraic set using SOS decompositions also exist, and leverage the Stengle Positivstellensatz involving products of ϕ_i (Parrilo, 2000). For our purposes, however, the certificate of nonnegativity as per equation (25) is sufficient.

The computational advantage of SOS programming stems from its intrinsic link to SDPs. Specifically, a polynomial p of degree $2d$ is SOS if and only if $p(x) = z(x)^T Q z(x)$, where $Q \succeq 0$ and z is a vector of monomials up to order d . Thus, certifying that a polynomial is SOS reduces to the task of finding a psd matrix Q subject to a finite set of linear equalities, thus taking the form in (23). Certificates of the form in (25) will form the building block for the computation of CCMs, as discussed next.

4.2 Optimized CCMs

As shown in (Manchester and Slotine, 2017), condition (8) can be written as a pointwise Linear-Matrix-Inequality (LMI) by introducing the *dual* metric $W(x) := M(x)^{-1}$ and the change of variables $\eta_x := M(x)\delta_x$. Specifically, define a matrix $B_\perp(x)$ whose columns form a basis for the

null space of $B(x)^T$ (i.e., $B(x)^T B_{\perp}(x) = 0$). Then, conditions (6) and (8) are equivalent to:

$$\partial_{b_j} W(x) - \widehat{\frac{\partial b_j(x)}{\partial x}} W(x) = 0, \quad j = 1, \dots, m \quad (26)$$

$$\underbrace{B_{\perp}^T \left(-\partial_f W(x) + \widehat{\frac{\partial f(x)}{\partial x}} W(x) + 2\lambda W(x) \right) B_{\perp}}_{:=G(x)} \preceq 0. \quad (27)$$

The equivalent reformulation of the weaker condition (10) in terms of the dual metric is given by LMI (27) and:

$$B_{\perp}^T \left(\partial_{b_j} W(x) - \widehat{\frac{\partial b_j(x)}{\partial x}} W(x) \right) B_{\perp} = 0, \quad j = 1, \dots, m. \quad (28)$$

In this section we replace the CCM *feasibility* problem, i.e., conditions (26) and (27), with a quasiconvex optimization problem to minimize the size of the outer approximation for the RCI mapping. A trade-off between the overshoot constant $\sqrt{\bar{\alpha}/\underline{\alpha}}$ and contraction rate λ was briefly discussed in (Manchester and Slotine, 2014). Here, we further explore this aspect by formulating a global *optimization* program to characterize such a trade-off and minimize conservatism. Ideally, one would directly like to minimize the bound in (22) subject to conditions (26) and (27). However, this problem is non-convex and infinite-dimensional.

First, to address the infinite-dimensionality of the problem, we consider a finite-dimensional approximation whereby the dual metric $W(x)$ is parameterized as a polynomial matrix and the LMIs are written as SOS constraints, enforced over the semi-algebraic set \mathcal{X} using the relaxations introduced in the previous section. Second, we propose an objective function that can be solved using line-search and quasiconvex optimization. Formally, we define the offline synthesis problem $\mathcal{OPT}_{\widehat{CCM}}$:

Optimization Problem $\mathcal{OPT}_{\widehat{CCM}}$ — Solve

$$\min_{\lambda \in \mathbb{R}_{>0}} J_{CCM}(\lambda) := \min_{\substack{W \in \mathcal{C}^\infty(\mathcal{X}, \mathbb{S}_n^{>0}) \\ \overline{W} \in \mathbb{S}_n^{>0}, \underline{\beta}, \bar{\beta} \in \mathbb{R}_{>0}}} \frac{1}{\lambda^2} \begin{pmatrix} \bar{\beta} \\ \underline{\beta} \end{pmatrix} \quad (29)$$

subject to

$$\text{eq. (26), eq. (27)} \quad (29)$$

$$\underline{\beta} I_n \preceq W(x) \preceq \overline{W} \preceq \bar{\beta} I_n, \quad (30)$$

where the conditions hold uniformly for all $x \in \mathcal{X}$.

The cost function above is an upper-bound on the *worst-case* (normalized) Euclidean distance within the ellipsoid defined in (22) since from (30), we have: $\bar{\alpha} = 1/\underline{\beta}$, $\underline{\alpha} = 1/\bar{\beta}$, and $\underline{M} = \overline{W}^{-1}$. Thus,

$$\sup_{x \in \tilde{\Omega}(x^*)} \frac{\|x - x^*\|^2}{\bar{w}^2} = \frac{\bar{\alpha}_w^2}{\lambda^2 \underline{\alpha}} \leq \frac{1}{\lambda^2} \begin{pmatrix} \bar{\alpha} \\ \underline{\alpha} \end{pmatrix} = \frac{1}{\lambda^2} \begin{pmatrix} \bar{\beta} \\ \underline{\beta} \end{pmatrix}.$$

Recognizing that for a fixed contraction rate λ , the CCM conditions define a convex feasibility region for $W(x)$, and that minimization of the condition number of a positive definite matrix over a closed convex set (i.e., the inner minimization in problem $\mathcal{OPT}_{\widehat{CCM}}$) is quasiconvex (Lu and

Pong, 2011), problem $\mathcal{OPT}_{\widehat{CCM}}$ can then be solved using line search on λ and bisection search over the condition number of $\bar{W}(x)$ (i.e., a sequence of feasibility problems). In the implementation of the bisection search we further minimized the expression $\text{Tr}(W_s \bar{W})$ where $W_s \in \mathbb{S}_n^{>0}$ is a diagonal scaling matrix. This is motivated by the following observation:

Remark 4.1. One may alternatively choose the following objective for problem $\mathcal{OPT}_{\widehat{CCM}}$:

$$\frac{1}{\lambda^2} \left(\frac{\text{Tr}(W_s \bar{W})}{\underline{\beta}} \right),$$

where $W_s \in \mathbb{S}_n^{>0}$ is a diagonal scaling matrix. This cost function is motivated by the observation:

$$\left(\frac{\bar{d}}{\bar{w}} \right)^2 = \frac{\bar{\alpha}_w^2}{\lambda^2} \leq \frac{\bar{\alpha}}{\lambda^2} = \frac{1}{\underline{\beta} \lambda^2},$$

and that the projection of the ellipsoid (22) along the i^{th} state is the interval: $[-\bar{d}\sqrt{\bar{W}_{ii}}, \bar{d}\sqrt{\bar{W}_{ii}}]$. Thus, this cost function attempts to minimize the size of this interval along each state dimension (relatively weighted by the matrix W_s). Since $\text{Tr}(\cdot)$ is convex and $\underline{\beta} = \lambda_{\min}(W)$ is concave, for a fixed λ , we obtain another quasiconvex objective for the inner minimization in problem $\mathcal{OPT}_{\widehat{CCM}}$, which can be solved using bisection search.

4.3 Illustrative Example: Planar Quadrotor

Consider the 6-state planar quadrotor system depicted in Figure 6. The state vector is defined as

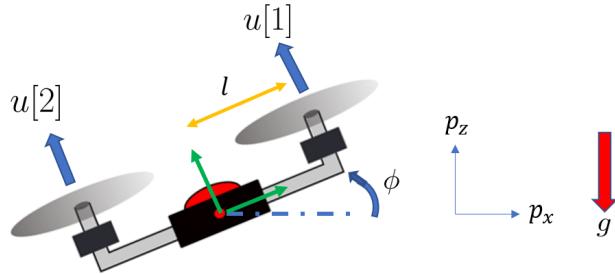


Figure 6: Definition of planar quadrotor state variables: l denotes the thrust moment arm (symmetric), and u_1 and u_2 denote the right and left thrust forces respectively.

$(p_x, p_z, \phi, \dot{p}_x, \dot{p}_z, \dot{\phi})^T$. For synthesizing the dual CCM however, it will be helpful to consider the alternative state-space representation: $(p_x, p_z, \phi, v_x, v_z, \dot{\phi})^T$ where (v_x, v_z) describe the velocity in the body frame of the vehicle with v_x representing the slip velocity (lateral) and v_z representing the velocity along the thrust axis. The dynamics in control affine form for this state representation are given as:

$$\dot{x} = \begin{bmatrix} \dot{p}_x \\ \dot{p}_z \\ \dot{\phi} \\ \dot{v}_x \\ \dot{v}_z \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} v_x \cos(\phi) - v_z \sin(\phi) \\ v_x \sin(\phi) + v_z \cos(\phi) \\ \dot{\phi} \\ v_z \dot{\phi} - g \sin(\phi) \\ -v_x \dot{\phi} - g \cos(\phi) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/m & 1/m \\ l/J & -l/J \end{bmatrix} \begin{bmatrix} u[1] \\ u[2] \end{bmatrix}, \quad (31)$$

where m and J denote the mass and moment of inertia about the out-of-plane axis. The inertial properties were taken from (Steinhardt and Tedrake, 2012), and were $m = 0.486$ kg, $J = 0.00383$ Kg m², and $l = 0.25$ m. By translation invariance of the dynamics, we expect that W will not be a function of (p_x, p_z) . Furthermore, by leveraging this state representation, condition (26) requires that W is not a function of v_z or $\dot{\phi}$. Thus, for a 6-state system, the CCM must necessarily only be a function of (v_x, ϕ) .

Let us impose the state-space bounds: $(v_x, v_z) \in [-2, 2] \times [-1, 1]$ m/s and $(\phi, \dot{\phi}) \in [-45^\circ, 45^\circ] \times [-60, 60]^\circ/\text{s}$, which may be concatenated together as the vector constraint $h(x) \geq 0$. A simple choice for B_\perp is the matrix

$$\begin{bmatrix} I_4 \\ 0_{2 \times 2} \end{bmatrix}.$$

Parameterize $W(x)$ as a polynomial matrix in (v_x, ϕ) with up to degree 4 monomials. With condition (26) dealt with, the remaining constraints in problem $\mathcal{OPT}_{\widehat{CCM}}$ may be written as:

$$\underline{\beta} \geq 1 \tag{32}$$

$$\overline{\beta}I_6 - \overline{W} \succeq 0 \tag{33}$$

$$h(x) \geq 0 \Rightarrow \overline{W} - W(x) \succeq 0 \tag{34}$$

$$h(x) \geq 0 \Rightarrow W(x) - \underline{\beta}I_6 \succeq 0 \tag{35}$$

$$h(x) \geq 0 \Rightarrow -B_\perp^T G(x) B_\perp \succeq \epsilon I_4. \tag{36}$$

where ϵ is a small positive number and the constraint $\underline{\beta} \geq 1$ is imposed to ensure uniform definiteness. To enforce the semi-definite constraints of the type $h(x) \geq 0 \Rightarrow F(x) \succeq 0$ where $F : \mathcal{X} \rightarrow \mathbb{R}^{n_F \times n_F}$ is some matrix-valued function, we leverage the quadratic form of the matrices by introducing auxiliary indeterminates y of dimension n_F so that the constraint may be equivalently written as $h(x) \geq 0 \Rightarrow y^T F(x) y \geq 0$, for all y . Leveraging certificates of the form in (25), we pose the constraints:

$$\begin{aligned} y^T F y - \sum_i L_i h_i &\text{ is SOS} \\ \{L_i\} &\text{ is SOS,} \end{aligned}$$

where $\{L_i\}$ is a collection of SOS functions in (x, y) . To keep the scale of the problem under control, one may wish to enforce simplifying structural properties for the multipliers $L(x, y)$ such as only retaining monomials quadratic in y (to reflect that $y^T F(x) y$ only contains quadratic terms in the auxiliary indeterminate y). The trigonometric terms in $G(x)$ from the dynamics function were approximated using Chebyshev polynomial expansions up to third order.

Having cast the optimization as a SOS program, we may now proceed with solving problem $\mathcal{OPT}_{\widehat{CCM}}$. We swept through a range of values for λ , using the Spotless polynomial optimization toolbox (Tobenkin et al., 2013) and MOSEK SDP solver (ApS, 2017) to solve the SOS programs, each of which took about 40 seconds. Figure 7 plots the optimal curve for J_{CCM} as a function of λ . The optimal contraction rate was determined to be $\lambda = 0.83$ and the corresponding dual metric $W(x)$ contained 15 unique monomials in (ϕ, v_x) . Assuming a cross-wind acting along either direction of the inertial p_x axis with effective acceleration up to 0.1 m/s², we used gridding to determine $\bar{d} = 0.038$. The resulting projections of the ellipsoid (22) are shown below in Figure 8.

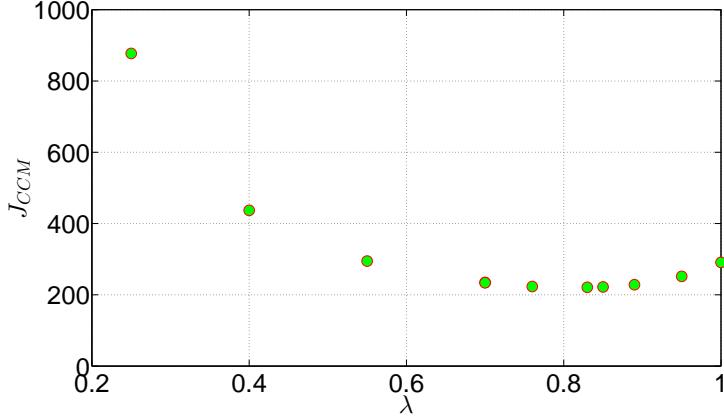
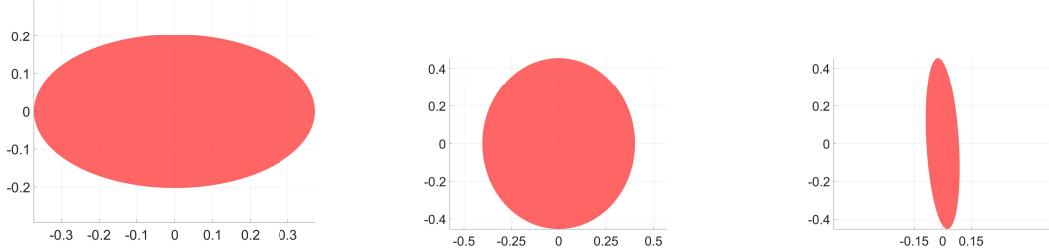


Figure 7: Problem $\mathcal{OPT}_{\widehat{CCM}}$ objective as a function of λ .



(a) Projection of ellipsoidal tube onto the position states p_x-p_z . (b) Projection of ellipsoidal tube onto the velocity states v_x-v_z . (c) Projection of ellipsoidal tube onto the rotational states $\dot{\phi}-\phi$.

Figure 8: Projections of the ellipsoidal tube upon various state-dimensions.

Notice that the projection of the outer ellipsoidal approximation of the RCI set onto the p_x-p_z plane has a major axis equal to 74 cm, and a minor axis equal to 41 cm, which compare quite favorably with the quadrotor wingspan, equal to 50 cm. That is, the size of the RCI tube is rather small and thus the nominal motion planner for the quadrotor is not overly constrained by the tightening of the state constraints. We next discuss an example for which the CCM synthesis problem can be drastically simplified using non-traditional state representations.

4.4 Case Study: Control of Mechanical Lagrangian Systems

In this section we demonstrate how to synthesize CCMs for fully-actuated Lagrangian systems by borrowing concepts from sliding mode control. The analysis presented here demonstrates an alternative method of designing CCMs via elegant state-space descriptions. Indeed the use of contraction theory for *fully* actuated systems is addressed quite recently both in (Manchester et al., 2015) and (Reyes-Báez et al., 2017). In (Manchester et al., 2015) the authors demonstrate that a *constant* CCM can always be constructed for fully actuated systems using the state-space description $x = (q, \dot{q})^T$ where $q \in \mathbb{R}^n$ is the vector of generalized coordinates, thereby reducing the dual metric search problem to a finite-dimensional SDP. In (Reyes-Báez et al., 2017), the authors instead leverage ideas from sliding control (and feedback linearization), and derive a suitable contraction metric for an alternate state-space description and a *given* feedback controller. In the following, we

also borrow ideas from sliding control but additionally provide a synthesis procedure for optimizing the controller's disturbance rejection properties and consequently, obtain tighter bounds than those resulting from the procedure in (Manchester et al., 2015), without using feedback linearization as in (Reyes-Báez et al., 2017). To do this, we first introduce the notion of *partial contraction* (Wang and Slotine, 2005).

4.4.1 Partial Contraction using Virtual Systems

Consider the autonomous system $\dot{x}(t) = f(x(t))$ and let $x(t; x_0)$ denote the solution at time t , starting from x_0 at $t = 0$. Define the following system:

$$\dot{y}(t) = \hat{f}(y(t), x(t; x_0)), \quad y(0) = y_0, \quad (37)$$

which satisfies the property:

$$\hat{f}(x, x) = f(x),$$

and with solutions denoted as $y(t; y_0)$. Thus setting $y_0 = x_0$ recovers the trajectory $x(t; x_0)$. Then, if there exists a uniformly (in x) stable differential Lyapunov function $V(y, \delta_y, t)$ for the above system, with associated variational dynamics

$$\dot{\delta}_y(t) = \frac{\partial \hat{f}}{\partial y}(y, x)\delta_y,$$

then $y(t; y_0)$ is contracting towards $y(t; x_0)$, i.e., $x(t; x_0)$. The crux of the argument is based on defining the *time-varying* system:

$$\dot{y}(t) = \tilde{f}_{x_0}(t, y(t)) := \hat{f}(y(t), x(t; x_0)), \quad y(0) = y_0,$$

and observing that the contracting properties of $\hat{f}(\cdot, \cdot)$ imply that solutions of the above time-varying system with arbitrary initial conditions y_0 and y'_0 converge towards each other. Setting $y'_0 = x_0$ proves the desired claim. The system (37) is defined as the “virtual” system for $\dot{x} = f(x)$.

4.4.2 Fully Actuated Systems

Consider the fully actuated Lagrangian dynamical system:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} = u,$$

where $q \in \mathbb{R}^n$ is the vector of generalized coordinates, $H(q) \in \mathbb{S}_n^{>0}$ is the inertia matrix, $C(q, \dot{q})$ contains the damping and Coriolis terms, and without loss of generality, we neglect the potential term. Let $(q^*(t), \dot{q}^*(t))^T$ represent the desired trajectory in generalized coordinates. Instead of using the state representation $x = (q, \dot{q})^T$, consider the change of variables:

$$x(t) := \begin{bmatrix} \tilde{q}(t) \\ \sigma(t) \end{bmatrix} := \begin{bmatrix} q(t) - q^*(t) \\ p(t) - p_r(t) \end{bmatrix},$$

where $p(t) := H(q(t))\dot{q}(t)$ is the generalized momentum, and $p_r(t) := H(q(t))\dot{q}^*(t) - H(q(t))\Lambda\tilde{q}(t)$ where $\Lambda \in \mathbb{S}_n^{>0}$ is a constant, positive definite, diagonal (for simplicity) matrix. Then, the dynamics

in this state-representation may be written as:

$$\begin{aligned}\dot{x} &= \begin{bmatrix} H^{-1}(q)(\sigma + p_r) - \dot{q}^* \\ \dot{H}(q)\dot{q} - C(q, \dot{q})\dot{q} + u - \dot{p}_r \end{bmatrix} \\ &= \begin{bmatrix} -\Lambda\tilde{q} + H^{-1}(q)\sigma \\ \left(\dot{H}H^{-1} + H\Lambda H^{-1} - CH^{-1}\right)\sigma + (-H\Lambda^2 + C\Lambda)\tilde{q} - (C\dot{q}^* + H\ddot{q}^*) \end{bmatrix} + \begin{bmatrix} O_n \\ I_n \end{bmatrix} u,\end{aligned}\tag{38}$$

where \ddot{q}^* is a function of (q^*, \dot{q}^*, u^*) . One may interpret σ as a *pseudo-sliding* variable since $\sigma = 0$ implies the stable first order dynamics for \tilde{q} . Notice that the state/control trajectory pair $(x^*(t), u^*(t)) = (0, u^*(t))$ is a feasible solution for the dynamics above.

Recall that $C = C(q(t), \dot{q}(t)) = C(q^*(t) + \tilde{q}(t), \dot{q}^*(t) + \dot{\tilde{q}}(t)) =: C_t(\tilde{q}, \dot{\tilde{q}})$ where the subscript $(\cdot)_t$ is used to denote the dependence upon the fixed signal $q^*(t)$. Similarly, write $H = H(q) = H_t(\tilde{q})$. Using the expression above for $\dot{\tilde{q}} = -\Lambda\tilde{q} + H_t^{-1}(\tilde{q})\sigma$, we further delineate $C_t(\tilde{q}, \dot{\tilde{q}})$ as $C_t(\tilde{q}, \sigma)$. Consider, then, the following virtual system for the above dynamics:

$$\dot{y} = \begin{bmatrix} -\Lambda y_1 + H_t^{-1}y_2 \\ \left(-H_t\Lambda^2 + C_t\Lambda\right)y_1 + \left(\dot{H}_tH_t^{-1} + H_t\Lambda H_t^{-1} - C_tH_t^{-1}\right)y_2 - (C_t(y_1, y_2)\dot{q}^* + H_t(y_1)\ddot{q}^*) \end{bmatrix} + \begin{bmatrix} O_n \\ I_n \end{bmatrix} u,$$

where we omit the explicit dependences (\tilde{q}) and (\tilde{q}, σ) for clarity. Clearly, setting $y(0) = (\tilde{q}(0), \sigma(0))$ and $u = u(t)$ recovers the true state trajectory $x(t)$ while setting $y(0) = (0, 0)$ and $u(t) = u^*(t)$ recovers the desired nominal state trajectory $x^*(t) = (0, 0)$ for all $t \geq 0$.

We now design a *constant* CCM for the virtual system. The result will also yield a stabilizing controller in the virtual space where as highlighted in the paragraph above, the endpoints of the geodesic in virtual space coincide with the nominal trajectory $x^*(t) = (0, 0)$ and actual trajectory $x(t)$. First, the variational dynamics for the virtual system above are given by:

$$\dot{\delta}_y = \begin{bmatrix} -\Lambda & H_t^{-1} \\ * & * \end{bmatrix} \delta_y + \begin{bmatrix} O_n \\ I_n \end{bmatrix} \delta_u,$$

where the $(*)$ indicate quantities that will become irrelevant in the CCM synthesis problem. Taking

$$B_\perp = \begin{bmatrix} I_n \\ O_n \end{bmatrix},$$

and assuming a *constant* dual metric W with the following block structure:

$$W = \begin{bmatrix} W_\perp & W_u \\ * & W_\parallel \end{bmatrix},$$

where $W_\perp, W_\parallel \in \mathbb{S}_n^{>0}$ and $W_u \in \mathbb{R}^{n \times n}$, the stability condition (27) requires that:

$$-\Lambda W_\perp - W_\perp \Lambda + H_t^{-1}W_u^T + W_u H_t^{-1} \preceq -2\lambda W_\perp.$$

Condition (26) simply requires that W is not a function of y_2 , which is automatically satisfied given W is parameterized as a *constant* matrix. As the CCM should be independent of the desired nominal trajectory, we set $W_u = O_n$, yielding, for given λ, Λ , the following LMI:

$$-\Lambda W_\perp - W_\perp \Lambda \preceq -2\lambda W_\perp.$$

This is an insightful result since the inequality above simply requires that the bandwidth of the pseudo-sliding variable σ , encapsulated by Λ , is faster than the contraction rate λ . With respect to (Manchester et al., 2015), by avoiding the construction of the dual metric in (q, \dot{q}) space, the resulting disturbance bound (for input disturbances) will depend upon $\bar{\sigma}(W_{\parallel}^{-1})$ instead of an expression that includes bounding the singular values of $H^{-1}(q)$, which can be very poor indeed⁵. As compared with (Reyes-Báez et al., 2017), we *do not* use feedback linearization to cancel out the nonlinearities for $\dot{\sigma}$ in (38), nor do we assume a fixed feedback controller and associated resulting contraction metric. In contrast, the proposed method explicitly allows a practitioner to balance the robustness properties of the controller and the bandwidth tradeoff between Λ and λ .

Let us examine this design tradeoff. Parameterize $W_{\perp} = w_{\perp} I_n$ and $W_{\parallel} = w_{\parallel} I_n$ where $w_{\perp}, w_{\parallel} \in \mathbb{R}_{>0}$. Then taking $B_w = B$, we have that

$$\bar{d} = \frac{\bar{w}}{\lambda \sqrt{w_{\parallel}}},$$

where, as before, \bar{w} bounds the Euclidean norm of the disturbance inputs. Projecting the bound into the virtual space (consequently bounding the state x), we obtain:

$$|\tilde{q}[i]| \leq \frac{\bar{w}}{\lambda} \sqrt{\frac{w_{\perp}}{w_{\parallel}}}, \quad \text{and} \quad |\sigma[i]| \leq \frac{\bar{w}}{\lambda} \quad \forall i = 1, \dots, n. \quad (39)$$

Re-writing σ as $H(q)(\dot{\tilde{q}} + \Lambda \tilde{q})$ and leveraging the bounds above on $\dot{\tilde{q}}$ and σ , one may further deduce bounds on $\dot{\tilde{q}}$, which can be shown to be proportional to $\bar{\sigma}(\Lambda)\|\tilde{q}\| + \bar{\sigma}(H^{-1}(q))\|\sigma\|$. While (39) highlights the design insight into reducing the error bounds on \tilde{q} and σ , it is apparent that a more aggressive Λ will result in weaker bounds. A strong case can be made, however, for the importance of bounding the error in generalized momentum as opposed to generalized rates.

Remark 4.2. One may further customize the dual metric for this class of systems by choosing a set of positive scalars $\{w_{\perp}[i]\}_{i=1}^n$ corresponding to the diagonal matrix W_{\perp} , such that the bound on the i^{th} generalized coordinate q_i may be expressed as:

$$|\tilde{q}[i]| \leq \frac{\bar{w}}{\lambda} \sqrt{\frac{w_{\perp}[i]}{w_{\parallel}}}.$$

The simplicity of such a customization and design freedom highlights the benefits of the virtual CCM method outlined in this case study.

5 Online Computation of CCM Controller

The solution to problem $\mathcal{OPT}_{\widehat{\text{CCM}}}$ provides an optimized CCM and an RCI tube, computed *offline*. In this section we describe the online component of the robust planning framework, namely, the implementation of the CCM-based tracking controller (Section 5.1). In addition, we also derive a bound on the tracking control effort allowing us to compute the tightened control constraint set $\bar{\mathcal{U}}$ (Section 5.3).

⁵This is easily seen since for input disturbances, the matrix $B_w = B = [O_n, H^{-1}(q)]^T$ and thus $\bar{\alpha}_w$ will depend upon $\bar{\sigma}(H^{-1}(q))$.

5.1 Tracking Controller

Equation (11) provides an example of how to construct a stabilizing feedback controller that guarantees the associated stability and robustness properties. That is, it establishes the *existence* of an exponentially stabilizing feedback controller with respect to the computed CCM. However, this feedback controller is not unique. Given the existence of this controller, we construct an alternative feedback controller that still satisfies the desired stability properties with respect to the computed CCM, but additionally, also minimizes the net control effort in order to curtail the suboptimality introduced by the “nominal plus tracking feedback” parameterization in (2). Given a CCM computed offline by solving problem $\mathcal{OPT}_{\widehat{\text{CCM}}}$, the feedback controller is computed as a solution to the following *analytical* QP:

Optimization Problem $\mathcal{OPT}_{\text{online}}$ — At time $t \geq 0$, given a desired/current state pair $(x^*(t), x(t))$ and a minimizing geodesic $\gamma(\cdot, t)$ connecting these two states (i.e., $\gamma(0, t) = x^*(t)$ and $\gamma(1, t) = x(t)$), solve

$$\begin{aligned} k^*(x^*(t), x(t)) &= \underset{k \in \mathbb{R}^m}{\operatorname{argmin}} \quad \|k\|^2 \\ \text{subject to} \quad &2\delta_\gamma^T(1, t)M(x(t))\hat{x}(t) - 2\delta_\gamma^T(0, t)M(x^*(t))\dot{x}^*(t) \\ &\leq -2\lambda\mathcal{E}(x^*(t), x(t)), \end{aligned} \quad (40)$$

where $\hat{x}(t) = f(x(t)) + B(x(t))(u^*(t) + k)$ represents the nominal dynamics evaluated at $x(t)$ and $\dot{x}^*(t) = f(x^*(t)) + B(x^*(t))u^*(t)$.

A few comments are in order. First, the existence of the dual metric $W(x)$ ensures that there exists a differential feedback controller such that inequality (9) holds for all (x, δ_x) along the minimizing geodesic between x^* and x . Then, by the equivalence shown in (17), problem $\mathcal{OPT}_{\text{online}}$ is always feasible. Second, the linear inequality (40) is essentially a relaxation of (9), in that it only enforces contraction *tangent to the given geodesic*. In contrast, the differential controller proposed in (Manchester and Slotine, 2017), obtained by solving a *feasibility* problem, must ensure that the system contracts in all directions with *at least* rate λ (implemented online by integrating the controller along the minimizing geodesic as in (11)). Such a relaxation still guarantees IES as only the flow *along the geodesic* affects the convergence of $x(t)$ to $x^*(t)$. On the other hand, one can often dramatically decrease control effort as compared with computing the controller using (11). Third, problem $\mathcal{OPT}_{\text{online}}$ is a QP subject to a single linear inequality and thus may be solved analytically (given the geodesic $\gamma(\cdot, t)$). Indeed, the QP above strongly resembles the min-norm formulation of Sontag’s generalized formula for CLF-based stabilization (Primbs et al., 1999), thereby underscoring the interpretation of the Riemann energy of the minimizing geodesic as an *incremental* CLF. We now address online computation of the geodesic.

5.2 Computing Geodesics Online

Computation of the geodesic between two points $p, q \in \mathcal{X}$ can be framed as the following functional optimization problem:

Optimization Problem \mathcal{OPT}_γ — At time $t \geq 0$, given desired state $x^*(t)$ and current state $x(t)$, solve

$$\min_{c \in \Gamma(x^*(t), x(t))} \mathcal{E}(c) \quad (41)$$

Following the approach in (Leung and Manchester, 2017), such a problem can be efficiently solved by applying the Chebyshev global pseudospectral method, i.e., by discretizing the interval $[0, 1]$ using the Chebyshev-Gauss-Lobatto nodes and using Chebyshev interpolating polynomials up to degree N to approximate the solution. The integral in (41) is approximated using the Clenshaw-Curtis quadrature scheme with $K > N$ nodes. As in (Leung and Manchester, 2017), we choose $K > N$ since the integral involves the inverse of the dual metric W . Thus, the integrand in $\mathcal{E}(c)$ is not guaranteed to be polynomial.

Given the solution to the geodesic problem \mathcal{OPT}_γ , parameterized by a set of values $\{\gamma(s_k)\}_{k=0}^K$ and $\{\delta_\gamma(s_k)\}_{k=0}^K$, $s_k \in [0, 1]$, problem $\mathcal{OPT}_{\text{online}}$ may now be solved as an analytical QP using $\delta_\gamma(s_0)$ and $\delta_\gamma(s_K)$.

5.3 Bounding Feedback Control Effort

There are two ways to compute bounds on the CCM feedback controller. For systems that satisfy the stronger Killing field condition given by (26), Theorem 5.2 provides a bound on the magnitude of the optimized tracking controller computed using problem $\mathcal{OPT}_{\text{online}}$. We first require the following technical lemma:

Lemma 5.1 (Norm Bound for Tracking Controller). *Let S be a symmetric matrix in $\mathbb{R}^{n \times n}$ and Y a full row-rank matrix in $\mathbb{R}^{m \times n}$. Construct matrices $\bar{Y} \in \mathbb{R}^{n \times m}$ and $\bar{Y}_\perp \in \mathbb{R}^{n \times (n-m)}$ such that the columns of \bar{Y} form an orthonormal basis for the column space of Y^T , i.e., $\text{Col}(Y^T)$, and the columns of \bar{Y}_\perp form an orthonormal basis for the nullspace of Y , i.e., $\mathcal{N}(Y)$. Suppose, then, that the following conditions hold:*

$$\eta_z^T S \eta_z \leq 0 \quad \forall \eta_z \in \mathcal{N}(Y) \subset \mathbb{R}^n \quad (42)$$

$$\kappa \leq \frac{2\bar{\delta}_u}{\theta}, \quad \text{where} \quad \kappa = \begin{cases} 0 & \text{if } \bar{\lambda}(S_Y) \leq 0 \\ \sqrt{\left(\frac{\bar{\lambda}(S_Y)}{\underline{\sigma}_{>0}(Y\bar{Y})}\right)^2 + 4\left(\frac{\bar{\sigma}(\bar{Y}_\perp^T S \bar{Y})}{\underline{\sigma}_{>0}(Y\bar{Y})}\right)^2} & \text{else} \end{cases}, \quad (43)$$

for some constants $\bar{\delta}_u \in \mathbb{R}_{\geq 0}$ and $\theta \in \mathbb{R}_{>0}$, where $\underline{\sigma}_{>0}(\cdot)$ denotes the smallest non-zero singular value, and $S_Y := \bar{Y}^T S \bar{Y}$. Then,

$$\theta \eta_z^T S \eta_z \leq 2\bar{\delta}_u \|Y \eta_z\| \quad \forall \eta_z \text{ s.t. } \|\eta_z\| \leq 1. \quad (44)$$

Proof. See Appendix A. □

We now leverage Lemma 5.1 to derive the bound on the magnitude of optimized tracking controller.

Theorem 5.2 (Tracking Control Effort). *Define*

$$F(x) := -\partial_f W(x) + \widehat{\frac{\partial f(x)}{\partial x}} W(x) + 2\lambda W(x).$$

Assume the dual CCM $W(x)$ satisfies conditions (26) and (27). Factorize $W(x)$ as $L(x)^T L(x)$ and define $S(x) = L^{-T} F L^{-1}$ and $Y(x) = B^T L^{-1}$. Let $\bar{\delta}_u$ be a positive real number such that the matrices $S(x)$ and $Y(x)$ satisfy property (43) for all $x \in \mathcal{X}$ with $\theta = \bar{d} = \bar{\alpha}_w \bar{w}/\lambda$, where \bar{Y} and \bar{Y}_\perp

are defined as stated in Lemma 5.1. Then, the optimized feedback controller $k^*(x^*, x)$ satisfies the bound:

$$\|k^*(x^*, x)\| \leq \bar{\delta}_u, \quad (45)$$

for all $x^*, x \in \mathcal{X}$ such that $x \in \Omega(x^*)$.

Proof. As a consequence of CCM condition (27), the matrices $S(x)$ and $Y(x)$ satisfy property (42) for all $x \in \mathcal{X}$. Then, in conjunction with property (43), it follows from the conclusions of Lemma 5.1 that

$$\begin{aligned} \bar{d}^2 \eta_z^T (L^{-T} F L^{-1}) \eta_z &\leq 2\bar{d}\bar{\delta}_u \|B^T L^{-1} \eta_z\|, \\ \forall \eta_z \text{ s.t. } \|\eta_z\| &\leq 1, \end{aligned} \quad (46)$$

for all $x \in \mathcal{X}$. Let $\eta_x := \bar{d}L^{-1}\eta_z$. Then, the set $\{\eta_z \in \mathbb{R}^n : \|\eta_z\| \leq 1\}$ is equivalent to the set $\{\eta_x \in \mathbb{R}^n : \|\eta_x\|_{W(x)}^2 \leq \bar{d}^2\}$ and inequality (46) may be written as

$$a(x, \eta_x) \leq \bar{\delta}_u \|r(x, \eta_x)\|, \quad \forall \eta_x \text{ s.t. } \eta_x^T W(x) \eta_x \leq \bar{d}^2, \quad (47)$$

for all $x \in \mathcal{X}$, where

$$\begin{aligned} a(x, \eta_x) &:= \eta_x^T F(x) \eta_x, \\ r(x, \eta_x) &:= 2B(x)^T \eta_x. \end{aligned}$$

Notice that statement (47) along with the CCM condition (27) is equivalent to the feasibility of the following CLF-like condition (with respect to bounded controls) stated in (Lin and Sontag, 1991):

$$\inf_{\|\delta_u\| \leq \bar{\delta}_u} (a(x, \eta_x) + r(x, \eta_x)^T \delta_u) \leq 0, \quad (48)$$

for all η_x satisfying $\|\eta_x\|_{W(x)}^2 \leq \bar{d}^2$. Then, by Theorem 1 in (Lin and Sontag, 1991) there exists an almost-smooth function $\delta_u(x, \eta_x)$, bounded in Euclidean norm by $\bar{\delta}_u$, such that condition (48) (equivalently the dual form of inequality (9)) is satisfied for all (x, η_x) along the minimizing geodesic connecting any $x^* \in \bar{\mathcal{X}}$ and $x \in \Omega(x^*)$. For completeness, this function is given below:

$$\delta_u(x, \eta_x) = \begin{cases} 0 & \text{if } r = 0, \\ -\frac{a + \sqrt{a^2 + \bar{\delta}_u^4 \|r\|^4}}{\bar{\delta}_u \|r\|^2 \left(1 + \sqrt{1 + \bar{\delta}_u^2 \|r\|^2}\right)} r & \text{else,} \end{cases}$$

where we have dropped the parenthesis (x, η_x) for clarity. For each $x \in \mathcal{X}$, the function above is continuous for all η_x (requisite for integrability) and smooth for $\eta_x \neq 0$.

By the equivalence shown through (17), the tracking controller given by integrating the function above along the minimizing geodesic connecting x^* and x is indeed a feasible solution to problem $\mathcal{OPT}_{\text{online}}$ that satisfies the bound claimed in (45), completing the proof. \square

A few comments regarding the computation of the bound $\bar{\delta}_u$ are in order. Note that from Theorem 5.2, one needs to show that inequality (47) holds for all $x \in \mathcal{X}$. Rewriting this inequality as (46), one may deduce a simpler, yet loose approximation of $\bar{\delta}_u$ as:

$$\bar{\delta}_u = \frac{\bar{d}}{2} \sup_{x \in \mathcal{X}} \left(\frac{\bar{\lambda}(L^{-T} F L^{-1})}{\underline{\sigma}_{>0}(B^T L^{-1})} \right), \quad (49)$$

where we omit the explicit dependence on x for notational clarity. A better approximation may be obtained by leveraging Lemma 5.1, specifically, inequality (43):

$$\bar{\delta}_u = \frac{\bar{d}}{2} \sup_{x \in \mathcal{X}} \kappa(x), \quad (50)$$

where κ is as given in (43) and $S, Y, \bar{Y}, \bar{Y}_\perp$ are as defined in Theorem 5.2.

For systems that do not satisfy the strong Killing field condition in (26), and instead satisfy the alternative pair of conditions, i.e., (27) and (28), one may bound the feedback control by augmenting the state with u , and treating \dot{u} as the actual input. The RCI tube obtained via the analysis in Section 3.5 then captures a bound on both x and u . We will shortly illustrate both methods for bounding the control effort.

6 Robust Planning

We are finally ready to formalize the robust planning algorithm. The core part of the algorithm relies on computing nominal motion plans (x^*, u^*) for the unperturbed dynamics using tightened constraints. Given the tubes derived in Section 3.5 and the control effort bounds computed in the previous section, these tightened constraints are given as:

$$x^*(\cdot) \in \bar{\mathcal{X}} := \mathcal{X} \ominus \tilde{\Omega}, \quad (51a)$$

$$u^*(\cdot) \in \bar{\mathcal{U}} := \{\bar{u} \in \mathcal{U} : \forall x^*(t) \in \bar{\mathcal{X}}, \forall x(t) \in \mathcal{X} \text{ s.t. } x(t) \in \Omega(x^*(t)), \bar{u} + k(x^*(t), x(t)) \in \mathcal{U}\}. \quad (51b)$$

Notice that the state constraint is shrunk by the fixed-size ellipsoid $\tilde{\Omega}$ to ease computation (e.g., collision-checking). On the other hand, since $x(t)$ is guaranteed to lie in the true RCI tube $\Omega(\cdot)$ defined using the geodesic distance, the control constraint is shrunk based on the bounds computed in the previous section. Having computed such a plan, one may adopt two different frameworks for its robust execution. In the first approach, one could simply execute the controller derived from problem $\mathcal{OPT}_{\text{online}}$ until the robot enters $\mathcal{X}_{\text{goal}}$. In the second approach, provided there exist sufficient online computational resources, one can use a receding-horizon algorithm in which the nominal trajectory is periodically and *locally* re-updated over a short time-horizon $T < T_{\text{goal}}$. This allows one to reduce the tracking cost (as information about the realized disturbances is taken into account). We outline such a receding-horizon strategy next.

6.1 Receding Horizon Implementation

Given a robust motion plan (i.e., a nominal state-input trajectory (x^*, u^*) such that the RCI tube centered on x^* does not intersect any obstacles), one can make *local* updates to it using the following MPC problem solved at the discrete time instants t_i , $i \in \mathbb{N}_{\geq 0}$:

Optimization Problem MPC — Given current state $x(t_i)$ and a robust motion plan (x^*, u^*) with associated RCI mapping $\Omega(\cdot)$, solve

$$\min_{\bar{u}(\cdot) \in \mathcal{C}^2([t_i, t_i+T], \bar{\mathcal{U}}), \bar{x}(t_i) \in \bar{\mathcal{X}}, T_i \in \mathbb{R}_{>0}} \int_{t_i}^{t_i+T} \|\bar{u}(\tau)\|_R^2 d\tau - \mu T_i \quad (52)$$

subject to $x(t_i) \in \Omega(\bar{x}(t_i))$,

$$\dot{\bar{x}}(\tau) = f(\bar{x}(\tau)) + B(\bar{x}(\tau))\bar{u}(\tau), \quad (53)$$

$$\bar{x}(\tau) \in \bar{\mathcal{X}}, \quad \bar{u}(\tau) \in \bar{\mathcal{U}} \quad \forall \tau \in [t_i, t_i + T] \quad (54)$$

$$\bar{x}(t_i + T) = x^*(T_i), \quad t_i + \delta \leq T_i \leq T_{\text{goal}}^*, \quad (55)$$

where $\mu > 0$ is a weighting factor. The time T_i (hereby termed as the “re-join” time), also an optimization variable within problem MPC, marks the point where the MPC state trajectory rejoins the initial motion plan and will be used to ensure persistent feasibility (see Lemma 6.1).

Problem MPC should be understood as a *local* re-optimization step – thus it should be solved using *local* methods such as trajectory optimization techniques (Betts, 2010) or elastic bands (Quinlan and Khatib, 1993) (as opposed to fully-fledged global planners). Notice that the initial value of the updated nominal state trajectory, i.e., $\bar{x}(t_i)$, is also an optimization variable above subject to the RCI constraint (52). This permits more drastic updates to the nominal trajectory, e.g., to counteract consistently large disturbances. The terminal constraint given by (55) is used to ensure *recursive feasibility* for the MPC problem, as addressed by the next lemma. The optimal re-join time for problem MPC is denoted as T_i^* and the corresponding optimal state-input pair is denoted as $(x_T^*(\cdot; x(t_i)), u_T^*(\cdot; x(t_i))) : [t_i, t_i + T] \rightarrow \bar{\mathcal{X}} \times \bar{\mathcal{U}}$, of which the segment $[t_i, t_i + \delta]$ is implemented using (2), before MPC is re-solved (with $\delta < T$). This defines the *sampled* MPC strategy commonly employed for continuous-time systems. The following lemma establishes recursive feasibility for problem MPC.

Lemma 6.1 (Recursive Feasibility for MPC). *Suppose problem MPC is feasible at the initial solve step $t_0 = 0$. Then, the problem is feasible for all $t_i, i \in \mathbb{N}_{\geq 0}$.*

Proof. Let $(x_T^*(t; x(t_i)), u_T^*(t; x(t_i))) : [t_i, t_i + T] \rightarrow \bar{\mathcal{X}} \times \bar{\mathcal{U}}$ and T_i^* denote the solution to problem MPC at time-step t_i . Then at solve time $t_{i+1} = t_i + \delta$, due to the RCI property associated with the tracking controller, one is guaranteed that the actual state $x(t_{i+1})$ lies within the set $\Omega(x_T^*(t_{i+1}; x(t_i)))$. Consider then the following feasible, but possibly suboptimal solution to problem MPC at solve time t_{i+1} :

$$\bar{x}(\tau) = \begin{cases} x_T^*(\tau; x(t_i)) & \text{for } \tau \in [t_{i+1}, t_i + T] \\ x^*(\tau) & \text{for } \tau \in [T_i^*, T_i^* + \delta], \end{cases}$$

$$\bar{u}(\tau) = \begin{cases} u_T^*(\tau; x(t_i)) & \text{for } \tau \in [t_{i+1}, t_i + T] \\ u^*(\tau) & \text{for } \tau \in [T_i^*, T_i^* + \delta], \end{cases}$$

The state-input trajectory above is simply a concatenation of the tail portion of the previous solution with the nominal motion plan solution, and represents a feasible solution for problem MPC due to the terminal constraint (55) (which guarantees that the end-point of $x_T^*(\cdot, x(t_i))$ rejoins the nominal motion plan x^* at time T_i^*). Hence, the feasible set of the MPC problem at solve time t_{i+1} is not empty, which proves recursive feasibility. \square

Remark 6.2. For $\mathcal{E}(x^*(0), x(0)) > 0$, the Comparison Lemma gives the following analytical, time-varying bound corresponding to the differential inequality (19):

$$\mathcal{E}(x^*(t), x(t)) \leq \left[\sqrt{\mathcal{E}(x^*(0), x(0))} e^{-\lambda t} + \bar{d} (1 - e^{-\lambda t}) \right]^2.$$

Thus, for the MPC problem solved at time t_i , one can modify constraint (52) to:

$$\mathcal{E}(\bar{x}(t_i), x(t_i)) \leq \mathcal{E}(x_T^*(t_i; x(t_{i-1})), x(t_i)). \quad (56)$$

This constraint states that the Riemann energy between the start of the new nominal MPC trajectory $\bar{x}(t_i)$ and current state $x(t_i)$ is at most the Riemann energy between the current MPC reference (as computed at the previous solve time t_{i-1}) and current state $x(t_i)$. This constraint may then be leveraged along with the shifted time-varying bound:

$$\mathcal{E}(x^*(t), x(t)) \leq \left[\sqrt{\mathcal{E}(\bar{x}(t_i), x(t_i))} e^{-\lambda(t-t_i)} + \bar{d} (1 - e^{-\lambda(t-t_i)}) \right]^2, \quad t \in [t_i, t_i + T],$$

within the ellipsoid (22) to obtain a time-varying tube that is smaller than the static ellipsoid and therefore, will be less conservative when incorporated within the tightened constraint set given in (51a). We demonstrate such an implementation in Section 7.

Algorithm 1 provides pseudocode for the overall approach.

Algorithm 1 Robust planning algorithm

```

1: OFFLINE:
2: Inputs: dynamics model,  $\mathcal{U}$  (control input constraints)
3: Compute:  $\Omega$  (RCI set),  $k(\cdot, \cdot)$  (controller structure)
4: ONLINE:
5: Inputs:  $x(0)$  (initial state),  $\mathcal{X}$  (state constraints),  $\mathcal{X}_{\text{goal}}$ 
6: Compute nominal  $(x^*, u^*)$ , such that  $x^*(\cdot) \in \bar{\mathcal{X}}$ 
7: Initialize:  $t_{\text{plan}} \leftarrow 0$ 
8: At each time  $t$ :
9: if New obstacles reported or goal region is changed then
10:   Re-plan nominal  $(x^*, u^*)$ 
11: else
12:   if  $t - t_{\text{plan}} = \delta$  then
13:      $(x_T^*(\cdot; x(t)), u_T^*(\cdot; x(t)), T^*) \leftarrow \text{MPC}(x(t), x^*, u^*, \Omega)$ 
14:     Update  $t_{\text{plan}} \leftarrow t$ 
15:   end if
16: end if
17: Apply control  $u_T^*(t; x(t_{\text{plan}})) + k(x_T^*(t; x(t_{\text{plan}})), x(t))$ 
```

7 Numerical Experiments

We now verify our approach in simulation by continuing with the 6-state planar-quadrotor system in Section 7.1, and then apply the framework to the more challenging 10-state 3D quadrotor model in Section 7.2. All simulation code (MATLAB) is available at <https://github.com/StanfordASL/RobustMP>.

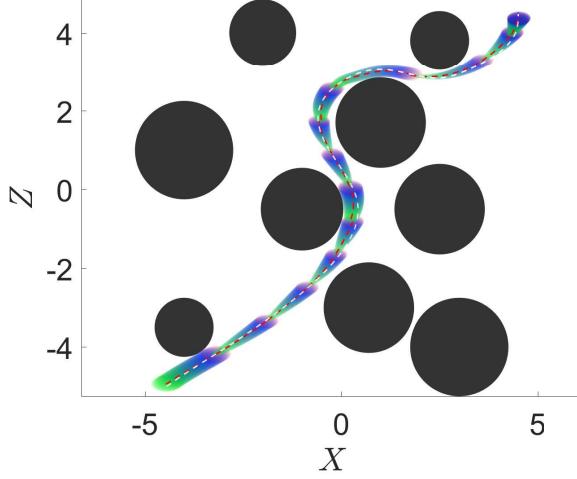
7.1 Planar Quadrotor

We continue with the 6-state planar-quadrotor system subject to horizontal cross-wind with effective acceleration up to 0.1 m/s^2 , whose dual metric $W(x)$ was designed in Example 4.3. Notably, this system is underactuated and has unstable zero dynamics, and thus represents a challenging system to benchmark the approach. In addition to the state-space constraints introduced during the metric synthesis, we imposed limits on the thrust for each propeller to the range $[0.1, 2]mg$. The feedback control effort bound $\bar{\delta}_u$ was determined using (49) to be 0.6 N ($0.13 mg$), a fraction of the overall control range.

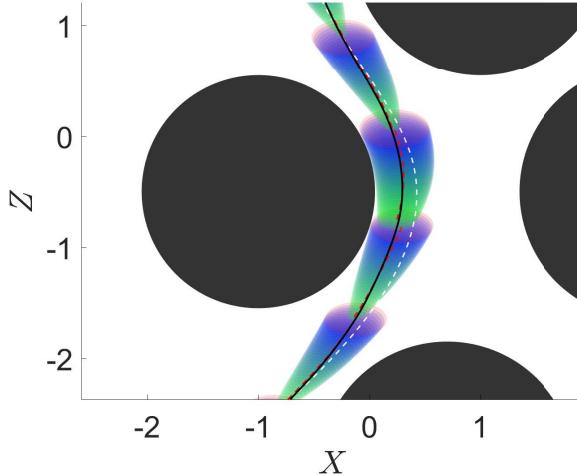
Having computed (offline) the RCI mapping, we tested Algorithm 1 on the previously unseen densely cluttered environment in Figure 1. The disturbance direction fluctuated (continuously) between left and right to try to push the vehicle into an obstacle. Problem MPC was re-solved every $\delta = 1\text{s}$ with horizon $T = 2\text{s}$ using the pseudospectral collocation method and the SNOPT solver, and leveraged the static bound given in (22). The tracking controller was implemented using zero-order-hold at 200 Hz. Leveraging a suboptimal implementation in MATLAB, the time required to compute the tracking controller (on a 3.5 GHz Intel equipped with 16 GB of RAM) is on the order of 3.5 ms. On average, each MPC problem took 0.35 s to solve. This compares favorably with the re-solve time of 1 s. Furthermore, we expect that this computational performance can be significantly improved with a more efficient implementation and by using trajectory optimization methods that fully exploit the *local* nature of the problem. We do not report the computation of the nominal trajectory (line 6 in Algorithm 1) since it highly depends on the motion planner used and is not a focus of this paper.

This example provides evidence that Algorithm 1 can be used for the online generation of safe motion plans that can be reliably executed (provided that the nominal motion plan can also be computed in real-time). This example also illustrates the benefits of our method as compared to the funnel library approach (Majumdar and Tedrake, 2017). A pre-computed set of trajectories (as required by (Majumdar and Tedrake, 2017)) would be unlikely to contain a sequence leading from the start to goal while maneuvering through the very tight spaces between obstacles.

In Figure 9, we illustrate the use of time-varying tubes as derived in Remark 6.2 within the online MPC problem. The MPC lookahead and re-solve times were increased to $T = 4\text{s}$ and $\delta = 1.5\text{s}$ respectively. We draw attention to the MPC solver making effective use of the time-varying tube to generate tighter nominal trajectories that would have been deemed infeasible using the fixed-size tube.



(a) Comparison of the initial motion plan (dashed-white line) and the online locally re-optimized trajectory (dashed-red line) leveraging time-varying tube bounds.



(b) Zoomed in view near an obstacle where the locally re-optimized trajectory is able to take a much tighter cut around the obstacle as consequence of significantly tighter time-varying bounds. Also shown is the actual trajectory (solid-black line) which can be seen to remain inside the time-varying tube centered around the re-optimized trajectory (dashed-red line).

Figure 9: Implementation of the planar quadrotor example with time-varying tubes. For clarity, the quadrotor graphic has been removed and the obstacles have been inflated by the vehicle size.

7.2 3D Quadrotor

7.2.1 Dynamics and Constraints

We adopt the state-space representation $x = (p_x, p_y, p_z, \dot{p}_x, \dot{p}_y, \dot{p}_z, f, \phi, \theta, \psi)^T$ where position $p = (p_x, p_y, p_z)^T \in \mathbb{R}^3$ and corresponding velocities are expressed in the global inertial (vertical axis pointing down) frame. Adopting the North-East-Down frame convention for the quadrotor body

and the XYZ Euler-angle rotation sequence, the attitude (roll, pitch, yaw) is parameterized as (ϕ, θ, ψ) and $f \in \mathbb{R}_{>0}$ is the net (normalized by mass) thrust generated by the four rotors. For the purposes of controller design, we consider as control inputs $u := (\dot{f}, \dot{\phi}, \dot{\theta}, \dot{\psi})^T$. Actual implementation embeds the \dot{f} term within an integrator and the resulting thrust and angular velocity reference are passed to a lower-level controller on the quadrotor that is assumed to operate at a much faster time-scale. Given this parameterization, the dynamics of the quadrotor may be written as:

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{p}_z \end{bmatrix} = ge_3 - f\hat{b}_z = \begin{bmatrix} -f \sin(\theta) \\ f \cos(\theta) \sin(\phi) \\ g - f \cos(\theta) \cos(\phi) \end{bmatrix}, \quad (57)$$

where g is the local gravitational acceleration, $e_3 = (0, 0, 1)^T$, and \hat{b}_z is the body-frame z-axis. The dynamics of $(\tau, \phi, \theta, \psi)$ reduce trivially to first-order integrators. We impose the bounds: $(\phi, \theta) \in [-60^\circ, 60^\circ]^2$ and $f \in (0.5, 2)g$, sufficient for executing fairly aggressive maneuvers.

7.2.2 CCM Computation

Notice that yaw is completely decoupled from these equations of motion. Consider, then, the following block partition model for $W(x)$:

$$W(x) = \left[\begin{array}{c|c|c} W_{\perp} & W_u & 0_{6 \times 1} \\ \hline * & W_{\parallel} & 0_{3 \times 1} \\ \hline * & * & W_{\psi} \end{array} \right],$$

where $W_{\perp} \in \mathbb{S}_6^{>0}$, $W_{\parallel} \in \mathbb{S}_3^{>0}$, $W_{\psi} \in \mathbb{R}_{>0}$ and $W_u \in \mathbb{R}^{3 \times 3}$. Given $B = [0_{4 \times 6}, I_4]^T$, take $B_{\perp} = [I_6, 0_{4 \times 6}]^T$. The weak Killing field condition in (28) requires that W_{\perp} is not a function of (f, ϕ, θ, ψ) . Further by noting the translational and yaw invariance of the dynamics, we only parametrize W_u and W_{\parallel} as functions of (f, ϕ, θ) and pick W_{\perp} to be a constant positive definite matrix and W_{ψ} a constant positive scalar. Such a parameterization also reflects the intuition that *differential stabilizability of the quadrotor, as captured by the CCM, should be independent of the quadrotor's position, velocity, and yaw orientation*. Given the block diagonal structure of W , we solve for the top left 9×9 block of the metric using the SOS formulation discussed in Section 4 and independently design W_{ψ} . Once again, the trigonometric terms in the dynamics were approximated using Chebyshev polynomial expansions up to third order. Figure 10 plots the optimal curve for J_{CCM} as a function of λ (for the top left 9×9 block of $W(x)$), using the alternative objective given in Remark 4.1. The scaling matrix W_s was chosen in order to prioritize the Euclidean error, and set as $\text{diag}(15, 15, 15, 1, 1, 1, 1, 1, 1)$.

The optimal contraction rate can be seen to be $\lambda = 1.29$ and the corresponding dual metric $W(x)$ contained 35 unique monomials in (f, ϕ, θ) . Assuming a cross-wind acting in *any* direction with effective acceleration up to 0.1 m/s^2 , we used gridding to compute a value of 0.0432 for \bar{d} . The resulting projections of ellipsoid (22) onto the position coordinates corresponded roughly to a sphere of radius $\approx 8.8 \text{ cm}$. The projection onto the thrust axis yielded the interval $[-0.05, 0.05]g$ and the projections in the (ϕ, θ) axes corresponded to the intervals $[-11.1^\circ, 11.1^\circ]$ and $[-7.45^\circ, 7.455^\circ]$ respectively. Since the projection of the bound along the yaw axis is given by $\bar{d}\sqrt{W_{\psi}}$ (see Remark 4.1), we set $W_{\psi} = 17.3$, yielding the interval $[-10^\circ, 10^\circ]$. The resulting tightened constraints are still permissive of aggressive maneuvers within cluttered obstacle environments, as demonstrated next.

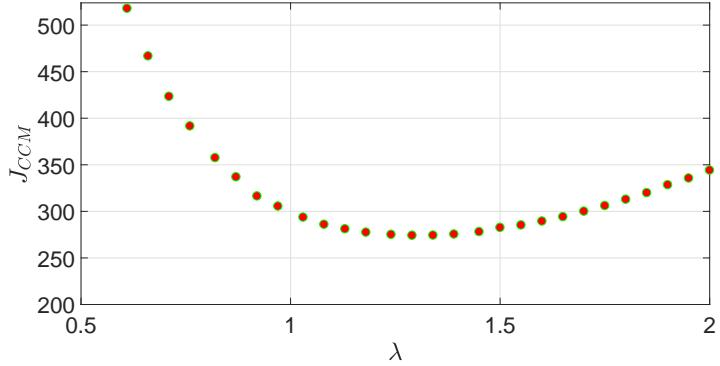


Figure 10: Problem $\mathcal{OPT}_{\widehat{CCM}}$ alternative objective as a function of λ .

7.2.3 Simulation under Nominal Disturbances

To verify the controller performance, we randomly initialized obstacle environments for the quadrotor, an example is depicted in Figure 11. Trajectory planning was performed by first computing a waypoint path using geometric FMT* (Janson et al., 2015a), and then smoothing this path using polynomial splines with the min-snap algorithm in (Richter et al., 2016). Finally, differential flatness was leveraged to recover the open-loop state and control trajectories. Collision checking was performed by leveraging the configuration space representation of the obstacles, i.e., polytopes, inflated by the size of the quadrotor (approximated as a 20 cm radius ball) and the projection of the tube bound onto position coordinates (a further 8.8 cm radius ball).

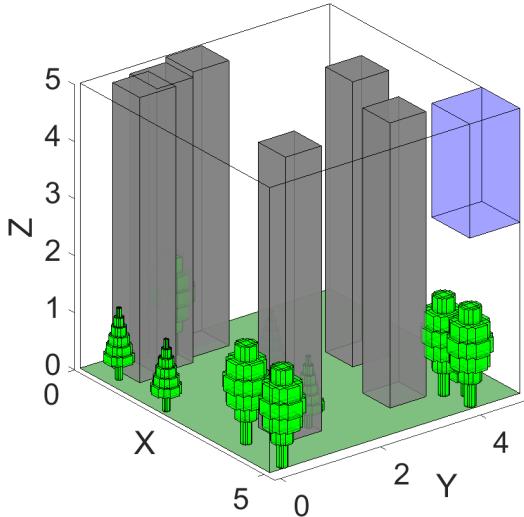


Figure 11: Randomly generated obstacle environment with towers and trees; initial position of the quadrotor is $(0, 0, 1)$, corresponding to the leftmost corner. The goal set is depicted as the light blue box.

Figure 12 shows an example of such a computed trajectory, along with the surrounding tube margin. The maximum speed along this trajectory is 3.00 m/s and the maximum pitch angle

is approximately 21.5° . The yaw trajectory was designed to follow the horizontal plane velocity. Note that both these aspects of planning (waypoint generation using sampling-based planning and polynomial smoothing) are *real-time* algorithms and therefore can be efficiently executed in receding horizon fashion. Robustness is easily accounted for via inflating the obstacles by the tube margin.

Figure 13 illustrates the simulation results for the computed trajectory in Figure 12 for a variety of disturbance time-series (sinusoidal varying and fixed direction signals with constant magnitude set to \bar{w}). The feedback controller is implemented using zero-order-hold at 250 Hz. All errors are observed to respect the theoretically computed bounds.

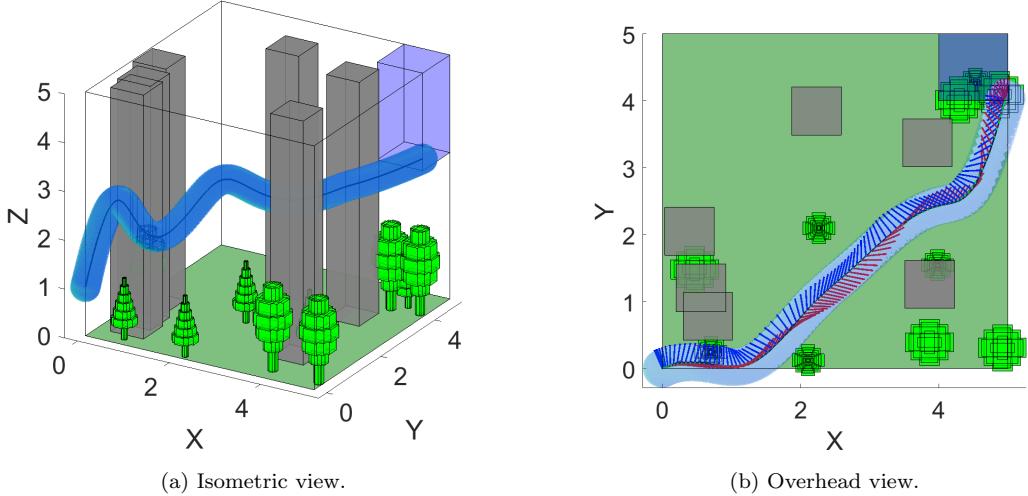


Figure 12: Computed nominal trajectory with attitude depicted using the body-fixed coordinate frame (forward: red, left: blue). The trajectory itself is centered within the depicted invariant ellipsoidal tube (shown inflated by the size of the quadrotor). The overhead view illustrates the tight margins near the beginning and end of the trajectory.

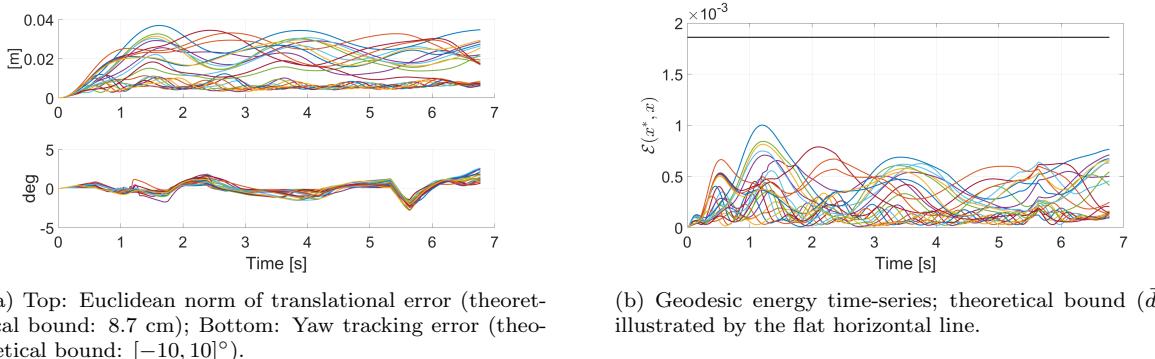


Figure 13: Time-series tracking error plots for 24 different disturbance time-series for the nominal trajectory in Figure 12. As expected, all errors remain within the theoretically computed bounds, ensuring safe execution of the planned path.

7.2.4 Assessing Conservatism

Within this section, we empirically evaluate the conservatism in the computed tracking bounds by planning trajectories assuming the disturbance bound of 0.1 m/s^2 and increasing the *actual*

disturbance level during simulation. We randomly generated 100 new trajectories similar to Figure 11 with varying obstacle placements. For each of the 100 nominal trajectories, we simulated the 24 disturbance time-series from Figure 13 with $\bar{w} \in \{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$ m/s². Figure 14 plots the empirical cdf of the tracking errors for different simulation disturbance thresholds. For reference, we also delineate the theoretical tracking bounds computed with the simulation \bar{w} values. The conservatism evaluations, therefore, are: (i) gap between actual tracking errors and the theoretical upper-bound, and (ii) extent of violation of a bound computed with a lower \bar{w} .

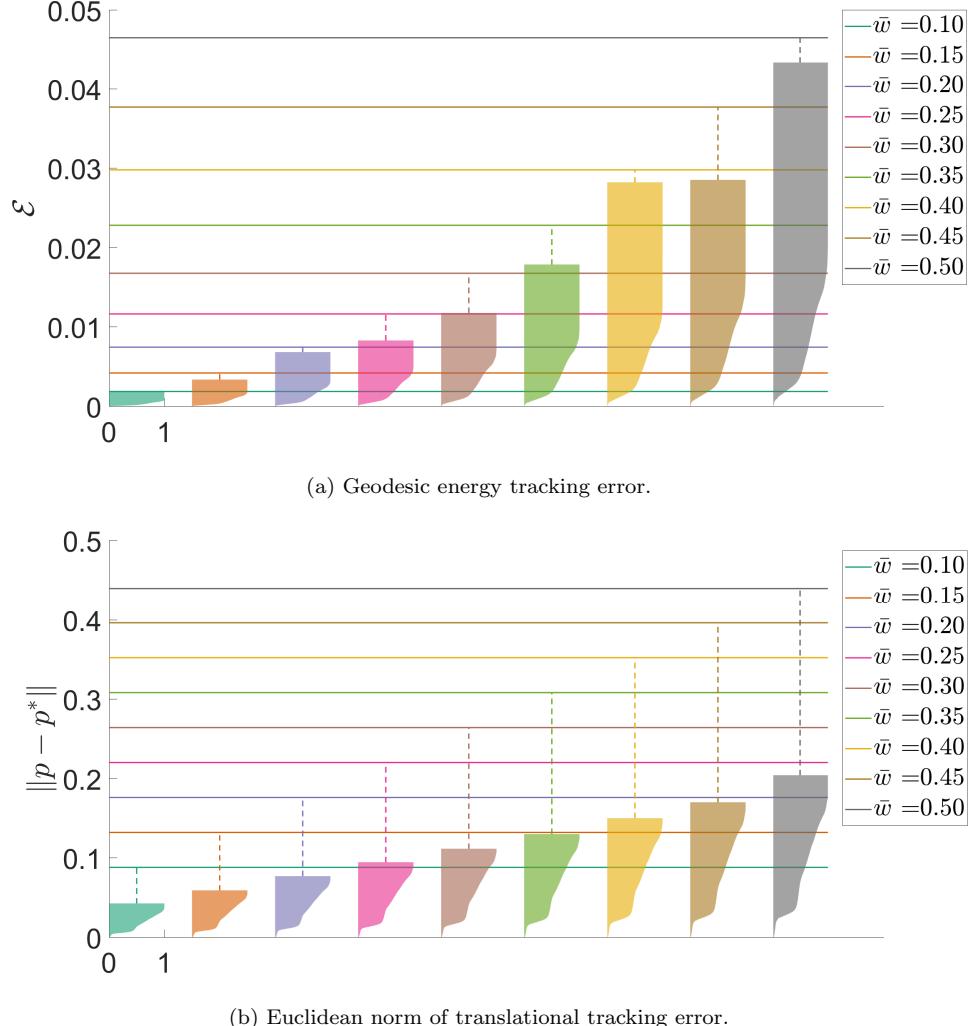


Figure 14: Empirical cdfs (rotated 90° for clarity) of tracking errors with varying simulation disturbance thresholds. Each cdf corresponds to data generated from 100 nominal trajectories and 24 disturbance time-series, for a total of approximately 4.1 million datapoints (equivalently, 4.5 hrs of simulation time) at each disturbance threshold. The horizontal lines indicate the theoretical tracking bounds as a function of \bar{w} , linked to the relevant cdfs via the vertical dotted lines. For clarity, the 0-1 range for each cdf is only depicted for the lowest disturbance threshold.

It can be noted in Figure 14(a), that the gap between the largest geodesic energy tracking error observed for a fixed disturbance threshold and the theoretical upper-bound is impressively tight.

Indeed, this gap ranged between 1.56 – 30 % of the theoretical bounds. Additionally, in all cases, the tracking errors crossed the theoretical bound computed using a value of \bar{w} that was 0.05 m/s² smaller than the simulated level. Both these indicators validate that the analysis in Theorem 3.5 is in fact practically useful and not overly conservative. The larger gaps between the maximum translational error and the theoretical bound, as observed in Figure 14(b), are to be expected since the geodesic energy is a full state error measure.

8 Hardware Experiments

Finally, we illustrate the approach on an open-source quadrotor platform, shown in Figure 15. The quadrotor consists of (i) a standard DJI F330 frame, (ii) Pixhawk autopilot running the estimator and lower-level thrust and angular rate controllers, and (iii) a companion on-board ODROID-XU4 computer running a ROS node for computing the CCM controller (which generates the thrust and angular rate setpoints for Pixhawk). The code for the ROS node is available for download at https://github.com/ssingh19/asl_flight. There is also an Optitrack motion capture system providing inertial position and yaw estimates at 120 Hz, which is fused with the onboard EKF on the Pixhawk.

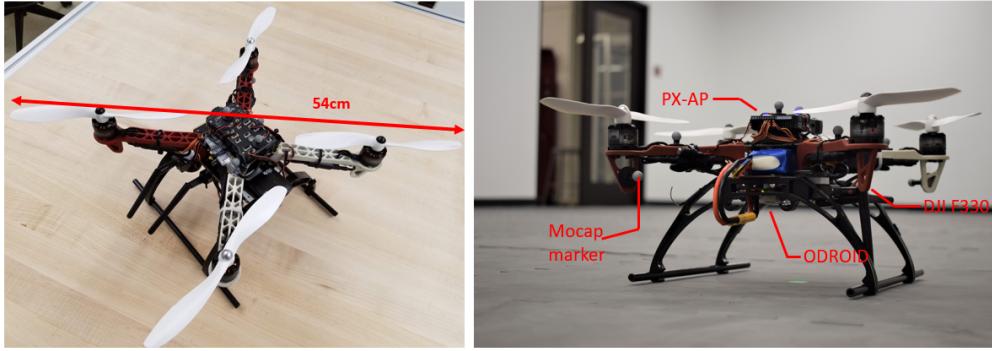


Figure 15: Quadrotor experimental platform, equipped with Pixhawk autopilot (PX-AP) for low-level (thrust and angular rate) control, and ODROID companion computer for planning and CCM controller.

8.1 Controller Implementation

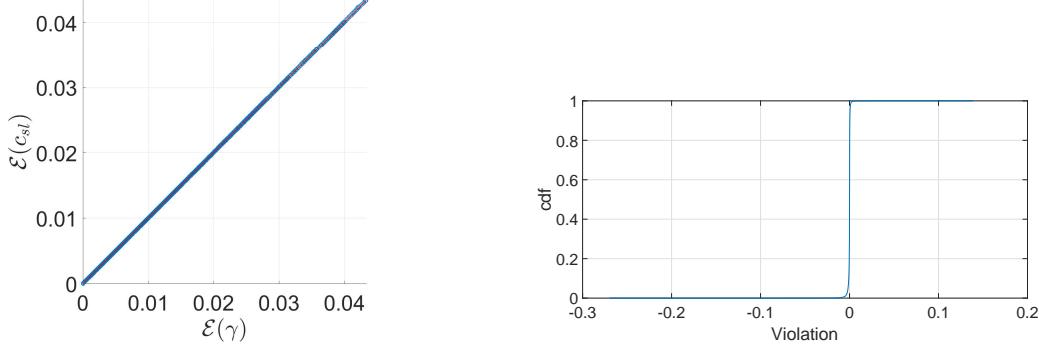
8.1.1 CCM Controller

During simulation it was often observed that the computed geodesics at each sampling instant were nearly a straight line between $x^*(t)$ and $x(t)$. Therefore, to further improve the runtime efficiency of the controller (i.e., avoid having to solve problem \mathcal{OPT}_γ on the embedded hardware), we investigated the performance of the controller using the straight-line approximation of the geodesic.

While it is theoretically possible to bound the numerical error resulting from this approximation and the induced error within inequality (40) via bounding the Christoffel symbols associated with $M(x)$, such an error analysis is likely to be conservative. Instead, we extracted tuples of $(x^*(t), x(t))$ from the simulations in Section 7.2.4, and performed the following evaluations – see Figure 16:

- Compare $\mathcal{E}(\gamma)$ and $\mathcal{E}(c_{\text{sl}})$, where $c_{\text{sl}}(\cdot, t)$ is the straight-line connecting $x^*(t)$ and $x(t)$.

- Magnitude of the induced error (i.e., violation) in inequality (40) with u_x set to the feedback computed using c_{sl} . This is the main evaluation criterion, from a stability performance perspective.



(a) Compare $\mathcal{E}(\gamma)$ and $\mathcal{E}(c_{\text{sl}})$. The datapoints lie almost perfectly along the $\mathcal{E}(c_{\text{sl}}) = \mathcal{E}(\gamma)$ diagonal line.
(b) cdf of violation of inequality (40) with k set to the feedback computed using c_{sl} .

Figure 16: Validating use of straight line approximation of geodesic to compute the online tracking controller. In the right subplot, a negative value indicates slack, while a positive value indicates violation of the stability inequality (40). The steep saturation of the curve just past 0 indicates relatively inconsequential implications for the violation of the stability inequality.

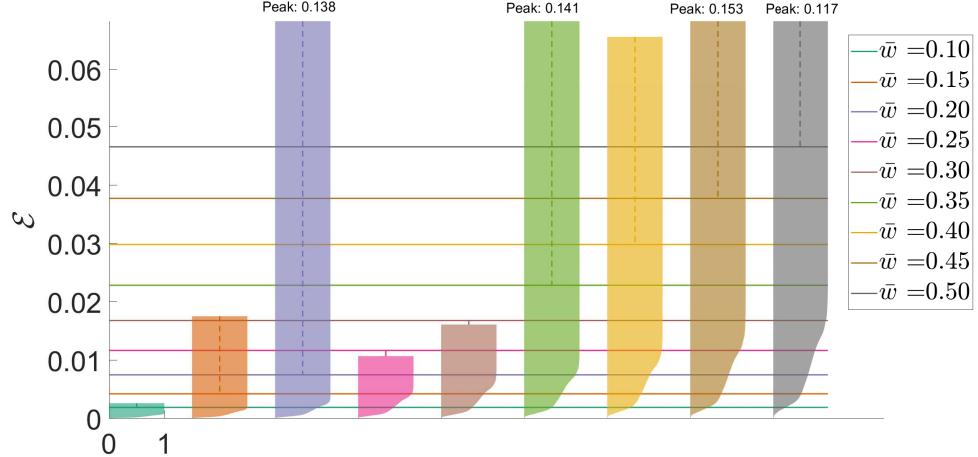
From Figure 16(a), we observe that $\mathcal{E}(c_{\text{sl}})$ almost perfectly matches $\mathcal{E}(\gamma)$, while Figure 16(b) illustrates that the resulting violation in the stability inequality (40) is practically negligible. To account for potential time-compounding effects of using the straight-line approximation (the evaluations in Figure 16 are pointwise in time), we repeated all simulations from Section 7.2.4 using c_{sl} in place of the geodesic to compute the tracking controller. Figure 17 presents the resulting tracking error cdfs. From Figure 17, one notices that the theoretical tracking bounds are indeed violated due to the compounding effect of the error induced within the stability inequality (40). However, the cdfs evaluated at the theoretical tracking bounds ranged from 0.999940 – 1.0. Equivalently, the cumulative time spent in violation of the geodesic energy theoretical bound, over all the simulations represented within Figure 17, was 7.044 s. Compared to the 40.71 hrs of simulation time represented in this figure, this is equivalent to a proportional violation of 0.004%. The translational tracking error however, remained always below the theoretical upper bound. Therefore, the straight-line approximation of the geodesic is a relatively mild simplification for this quadrotor example.

8.1.2 Lower-Level Pixhawk Controller

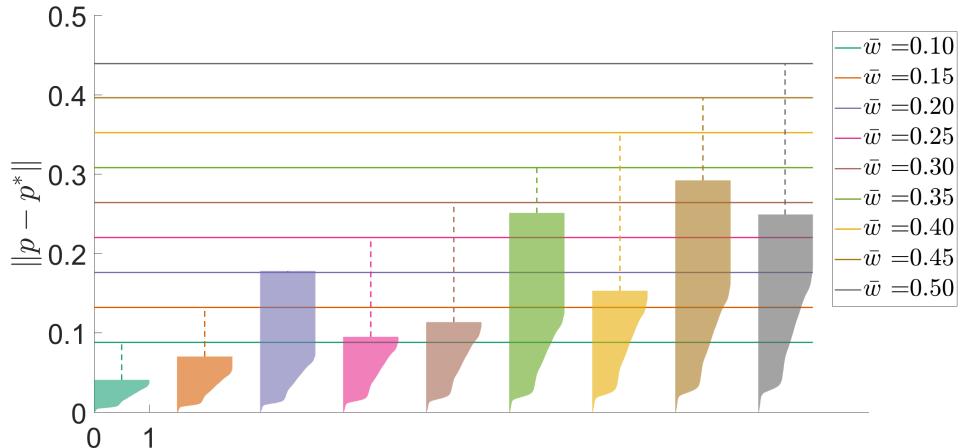
The lower-level controller on the Pixhawk autopilot is tasked with tracking the thrust and angular rate control setpoints generated by the CCM controller. The Euler rate commands from the CCM controller are converted into desired body-rates onboard Pixhawk and fed into the existing PID control loop. For thrust control, the commanded normalized thrust f_c from the CCM controller and the estimated inertial acceleration \hat{p} (obtained via moving-average finite-differencing) are used to first compute an error:

$$e_f = (f_c - g e_3 \cdot \hat{b}_z) - (-\hat{p} \cdot \hat{b}_z),$$

which is simply the difference in the desired and actual inertial accelerations, projected along the instantaneous \hat{b}_z axis. This error is then converted into a normalized throttle command $\tau \in [0, 1]$



(a) Geodesic energy tracking error.



(b) Euclidean norm of translational tracking error.

Figure 17: Empirical cdfs (rotated 90° for clarity) of tracking errors with varying simulation disturbance thresholds, from using a straight-line approximation of the geodesic for computing the tracking controller. As before, the horizontal lines indicate the theoretical tracking bounds as a function of \bar{w} , linked to the relevant cdfs via the vertical dotted lines.

using the following discrete-time recursion:

$$\tau(t_k) = \tau(t_{k-1}) + k_p e_f(t_k),$$

where the time indices $t_k, k = 0, 1, 2, \dots$ delineate the Pixhawk sampling times and k_p is a constant positive gain. The CCM controller on board the ODROID is run at 250 Hz, while the Pixhawk control loop runs between 350-400 Hz. The controller above is similar to the thrust controller proposed in (Tal and Karaman, 2018), and corresponds to an *iterative* scheme where the proportional error term is used to simply *correct* the previous throttle command, precluding the need to estimate the complex mapping between desired thrust and throttle, or rely on the popular yet oversimplifying quadratic model of this relationship.

8.2 Calibrating Disturbance Bound

The role of aerodynamic disturbance for experiments was played by the (neglected in eq. (57)) drag force. To calibrate an upper bound for planning, we flew two calibration trajectories: a figure-eight (presented in this section) and a “race-course” (similar to the one in the next section). We describe the figure-eight trajectory here. The nominal trajectory was set as: $p_x(t) = r_x - r_x \cos(\omega t)$, $p_y(t) = r_y \sin(2\omega t)$, $p_z(t) = h$, where we set $r_x = 1$ m, $r_y = 0.7$ m, $\omega = 2\pi/10$, and a constant altitude $h = 1.5$ m. The drag-force was estimated based on the following augmented translational dynamics:

$$\ddot{p} = g e_3 - f \hat{b}_z + R \underbrace{(d_0 - D R^T \dot{p})}_{:= f_d}, \quad (58)$$

where R is the rotation matrix, $D = \text{diag}(\mu_x, \mu_y, 0)$, and $d_0 = (d_{0x}, d_{0y}, 0)^T$. Here d_0 plays the role of a fixed perturbing force stemming from propeller misalignment, and D models the linear velocity drag coefficients, commonly accepted to be the dominant component of drag within the expected flight regime; for instance, see (Kai et al., 2017; Faessler et al., 2018). As in (Faessler et al., 2018), we neglect the body z component of drag.

To estimate the net body-frame drag f_d , we leveraged a smoothed finite-differenced estimate of \ddot{p} , the estimated rotation matrix R , and the *commanded* normalized thrust f_c in place of f . This last substitution is justified courtesy of the lower-level thrust controller presented in the preceding section. Figure 18 shows a series of plots from the figure-eight experiment, including (a) XY desired and actual traces, (b) translational tracking errors, (c) body-frame velocity, (d) estimated body-frame drag, (e) Euclidean norm of net drag, and (f) geodesic energy.

Concatenating the data from the figure-eight and race-course trajectories, we used least-squares to compute the following estimates: $\mu_x = 0.302$, $\mu_y = 0.288$, $d_{0x} = -0.008$, $d_{0y} = -0.026$. Figure 19 plots an overlap of the estimated and predicted drag for the figure-eight trajectory, and illustrates good agreement. In the next section, we present additional plots of estimated and predicted drag from the planning experiments to further validate the use of this drag model as a viable disturbance model.

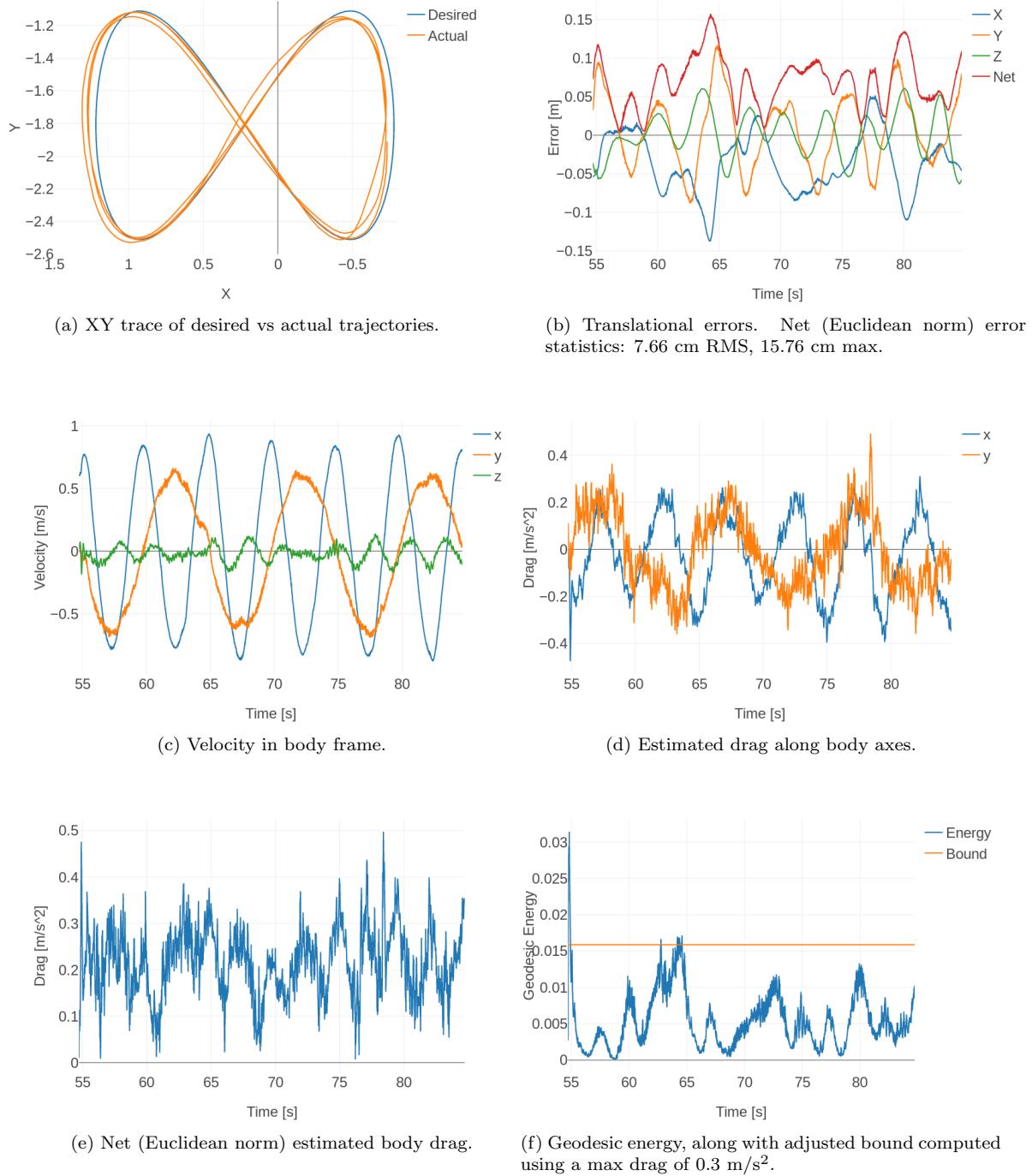


Figure 18: Experiment results for drag calibration flight using fixed-yaw figure-eight trajectory. The experiment was performed by ramping up speed over two cycles; the plots shown correspond to three cycles at the final speed (period of 10s). In subplot (f), corresponding to geodesic energy, the initial spike around 55 s corresponds to a jump in speed for the nominal (reference) trajectory. Following the spike, the error stays below the (drag-adjusted) theoretical bound illustrated by the horizontal line.

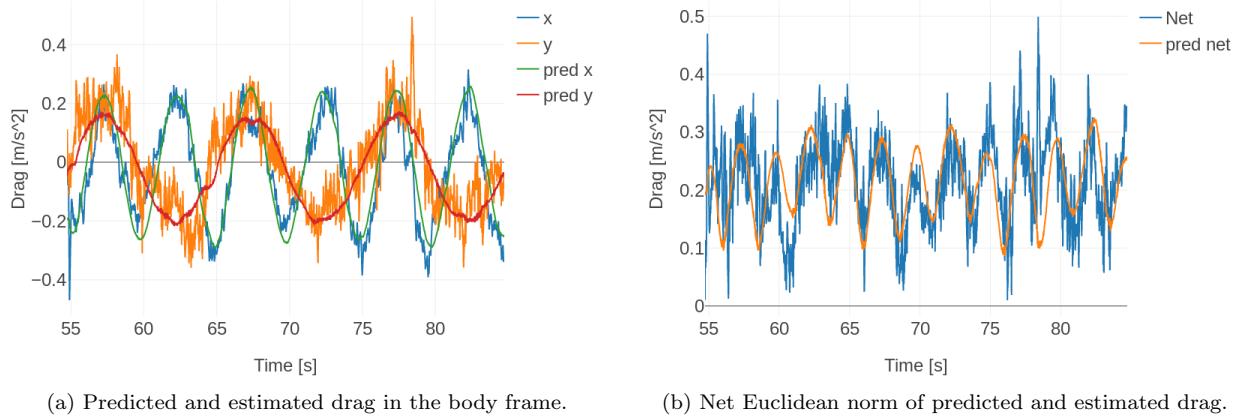


Figure 19: Comparison of linear drag model predictions and estimated drag for the figure-eight calibration flight. Note that the entire flight (including the cycles at slower speed) and the race-course trajectory were used for estimating the parameters of the linear drag model.

Remark 8.1. In these experiments, the unknown but bounded perturbing force was taken to be the linear drag model. However, given the recent advancements in leveraging drag models within planning, for instance, as in (Faessler et al., 2018), one may also incorporate the learned drag model within the nominal dynamics and compute a drag-compensated CCM. While this would necessitate including coupled velocity and yaw dependence within the CCM, it would allow the separation of drag from the unmodeled aerodynamic disturbances, thereby reserving the disturbance bound margin for purely exogenous effects. For the purposes of illustrating the methodology in this paper, we reserve this extension for future work.

8.3 Robust Planning

Equipped with a calibrated disturbance model, we ran the robust trajectory planner (geometric FMT* plus polynomial spline smoothing) introduced in the simulation section on the “race-course” test environment shown below in Figure 20. The computed trajectory was setup to create a challenging loop through the obstacle course through intermediate waypoints placed, for example, at the center of gates and in between the poles. To ensure robustness with respect to the drag model, a single constant was used to scale time (effectively, speed) along the trajectory such that the drag-adjusted invariant tube was collision free. Importantly, the drag model was not re-estimated for these test trajectories. The computed trajectory along with the invariant tube is shown in Figure 21. Figure 22 plots the nominal body-frame velocity and expected drag.



Figure 20: Quadrotor ‘‘race-course’’ test environment.

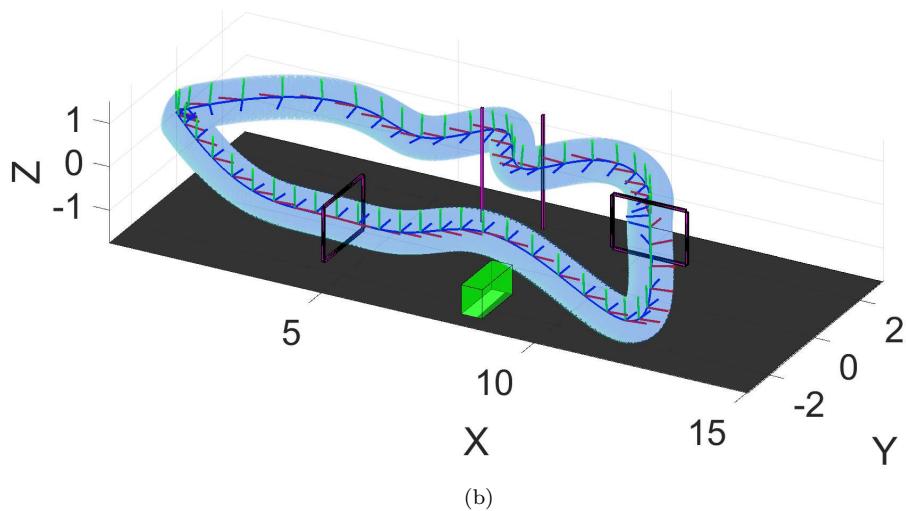
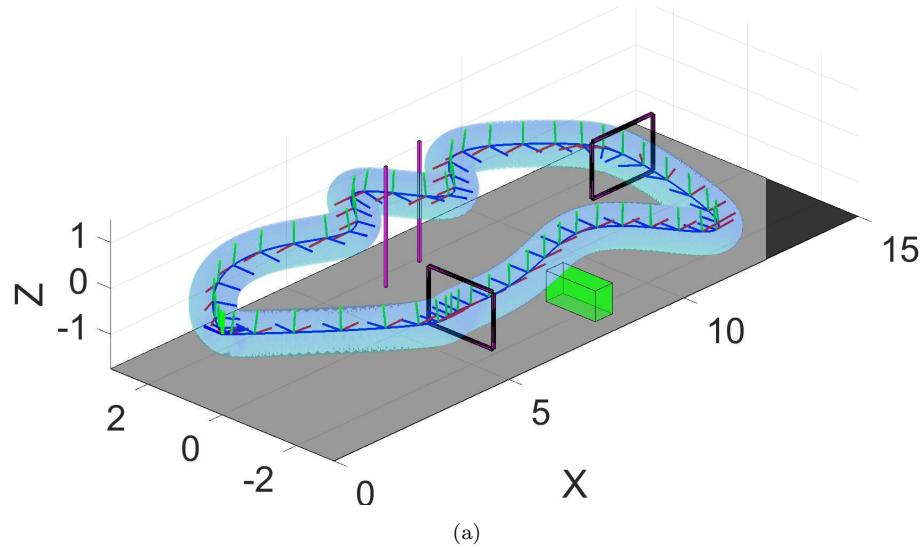


Figure 21: Computed nominal trajectory with attitude depicted using the body-fixed coordinate frame (forward: red, left: blue). The trajectory itself is centered within the depicted invariant ellipsoidal tube. The views illustrate the tight margins through the obstacles.

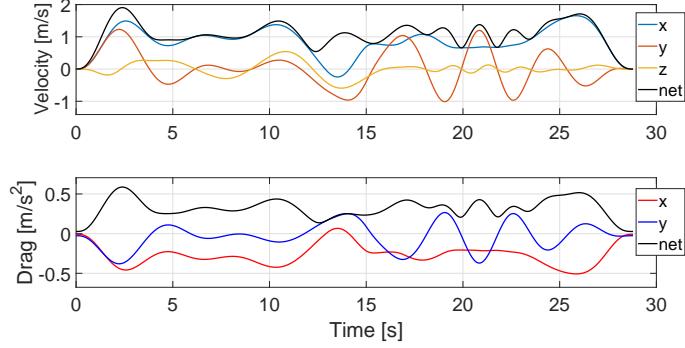
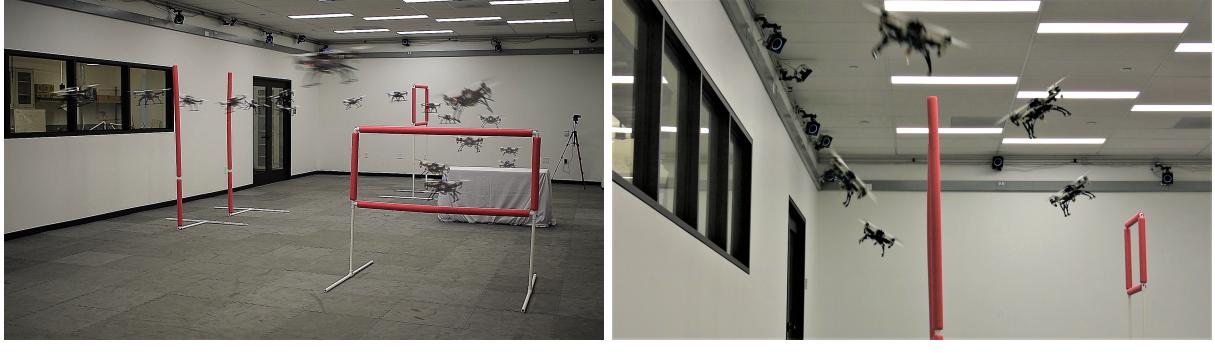


Figure 22: Top: Body-frame and net velocity; Bottom: Predicted body-frame and net drag force, along nominal trajectory. Maximum expected drag: 0.59 m/s^2 .

The maximum expected drag along the nominal trajectory is 0.59 m/s^2 , corresponding to a top speed of 1.9 m/s . This results in an adjusted translation error bound of 52 cm and a geodesic energy bound of $\bar{d}^2 = 0.0648$. The trajectory in Figure 21 was computed with a combined collision margin of 52 cm plus an additional 27 cm to account for the size of the quadrotor (see Figure 15).

Following the computation of the nominal trajectory, the quadrotor was flown through the obstacle loop three times, with a 25% increase in speed (respectively, decrease in lap time) with each successive lap. A time-lapse⁶ during the fastest lap is shown in Figure 23.



(a) Time-lapse through the full obstacle course. (b) Time-lapse through the slalom maneuver past the poles.

Figure 23: Time-lapse of quadrotor through the obstacle course during the fastest lap. Top speed achieved: 3.81 m/s .

Note that the trajectory was not re-computed during the experiment since the purpose here was to (i) illustrate that the theoretically computed tracking error bounds are indeed physically meaningful and achievable on an actual hardware testbed, and (ii) evaluate the conservatism in the tracking bounds via greater experienced disturbance than planned for. In particular, only the first speed setting is theoretically robustly collision-free, in that the invariant tube computed using the max expected drag is collision-free. The subsequent laps executed at higher speeds have significantly higher expected drag and the resulting scaled tubes are not collision-free. Given the simulation results presented in Figure 14(b) however, we anticipated (and observed) collision-free execution at the higher speeds as well.

⁶Videos of the flights can be found at <https://www.youtube.com/playlist?list=PL8-2mtI1FIJo5n4J4wgP6zdtEGKYeTY01>.

The experiment results for the three laps are presented in Figures 24– 26, and comprise (i) XY desired and actual traces, (ii) translational and geodesic energy tracking errors, and (iii) estimated and predicted body-frame and net drag. The statistics for the three laps are presented in Table 1.

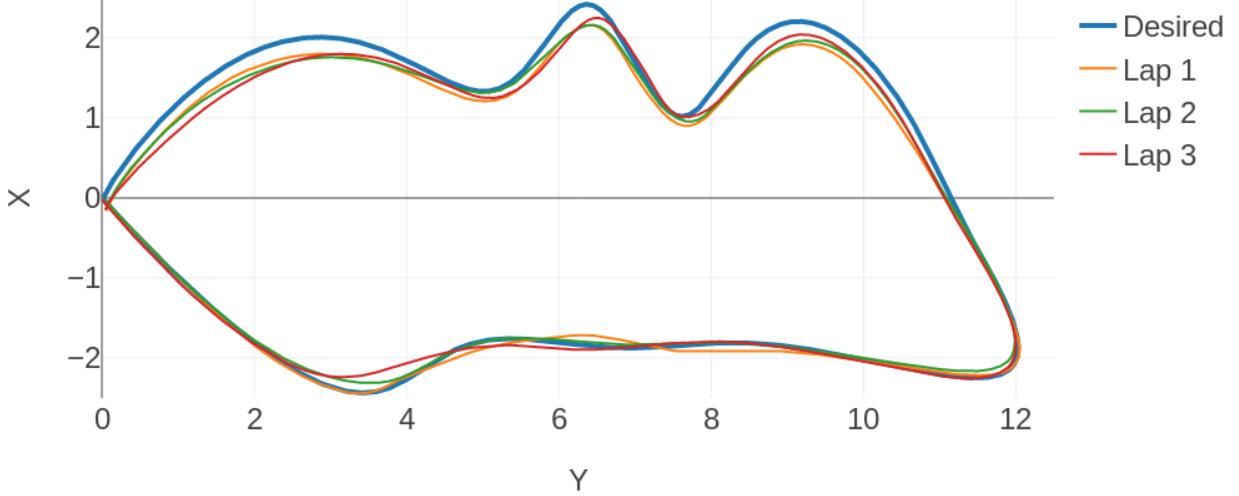
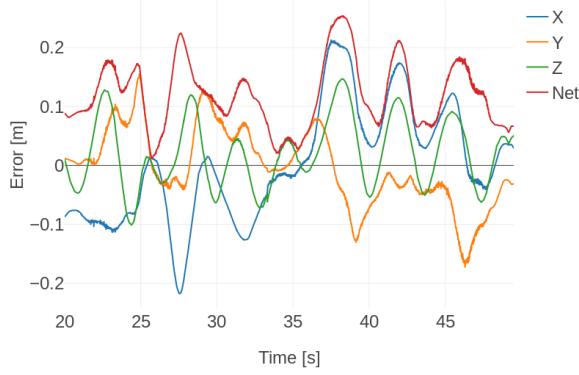


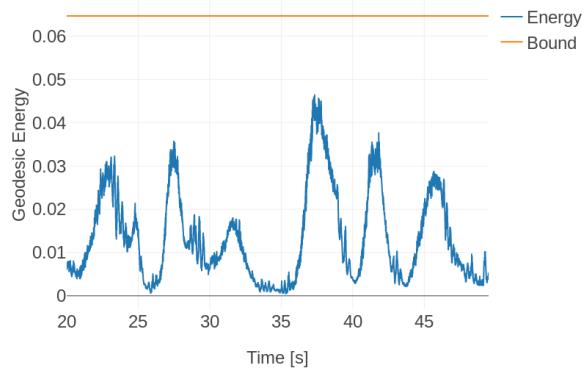
Figure 24: XY Trace of desired and actual followed trajectories. Lap direction: counter-clockwise.

Flight Extremes			Bounds		Errors rms (max)		
	Speed [m/s]	Bank [deg]	Drag [m/s ²]	\mathcal{E}	$\ p - p^*\ $ [cm]	\mathcal{E}	$\ p - p^*\ $ [cm]
Lap 1	1.91	11.46	0.59	0.0648	52	0.017 (0.044)	13.18 (25.4)
Lap 2	2.54	19.8	0.78	0.112	68	0.025 (0.074)	14.51 (30.17)
Lap 3	3.81	39.04	1.15	0.247	101	0.036 (0.124)	16.24 (38.91)

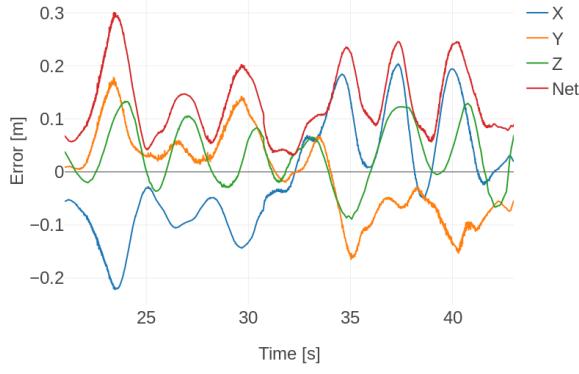
Table 1: Nominal trajectory extremes (max drag corresponds to the prediction from the linear drag model), drag-adjusted bounds, and actual flight error statistics. All three laps respect the theoretical upper-bounds computed prior to the flights.



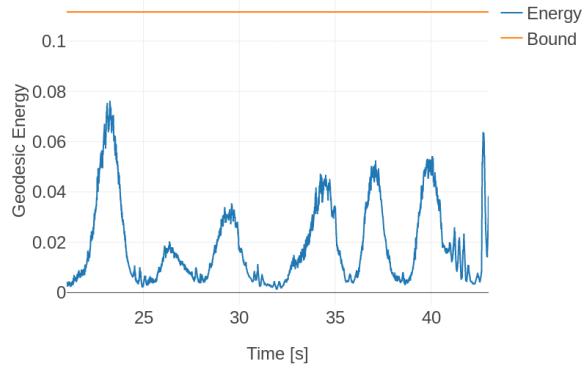
(a) Lap 1 translational error. Error rms: 13.18 cm, max: 25.4 cm, bound: 52 cm.



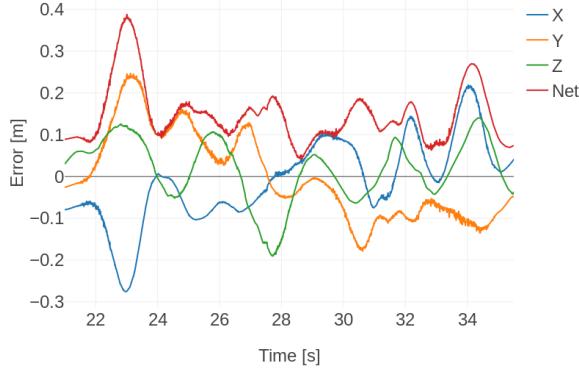
(b) Lap 1 geodesic energy. Error rms: 0.017, max: 0.044, bound: 0.0648.



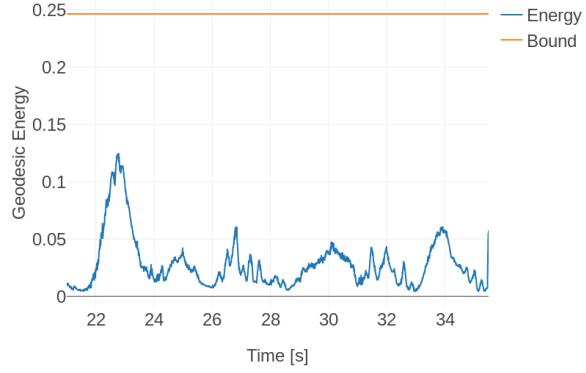
(c) Lap 2 translational error. Error rms: 14.51 cm, max: 30.17 cm, bound: 68 cm.



(d) Lap 2 geodesic energy. Error rms: 0.025, max: 0.074, bound: 0.112.



(e) Lap 3 translational error. Error rms: 16.24 cm, max: 38.91 cm, bound: 101 cm.



(f) Lap 3 geodesic energy. Error rms: 0.036, max: 0.124, bound: 0.247.

Figure 25: Validation of translational (left column) and geodesic energy (right column) error upper-bounds.

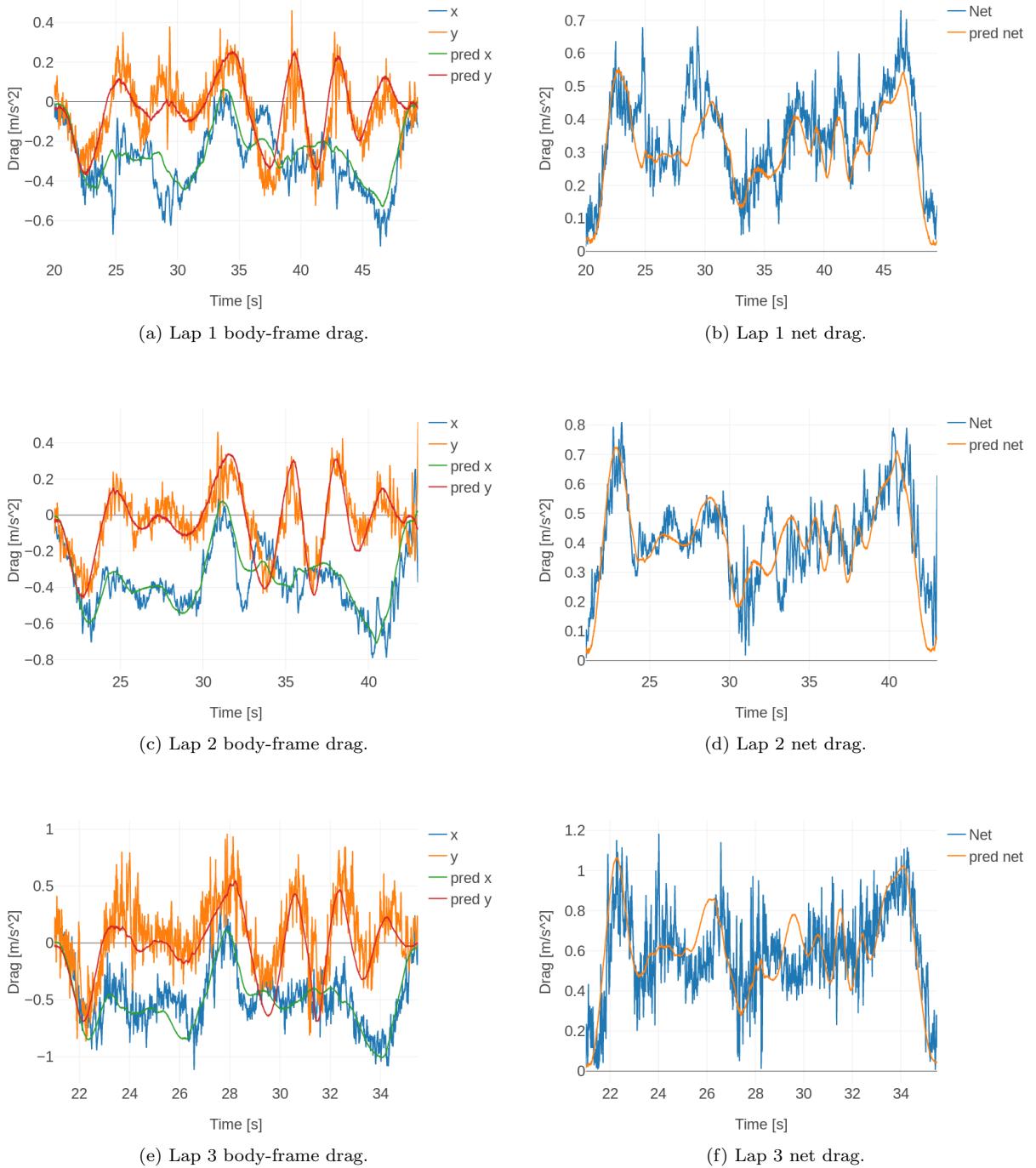


Figure 26: Validation of predicted drag model with parameters fixed *a priori* to all laps.

The key takeaway from the preceding numbers and plots is that all theoretical (a priori) geodesic energy and translational error tracking bounds computed using the calibrated drag model are validated within all three flights. Additionally, we note the following interesting observations.

First, the peaks in the geodesic energy curves line up quite well with the local peaks in aerodynamic disturbance due to either drag model or ground/interaction effects. Indeed, as the video illustrates, there are notable perturbations in the vicinity of the box (see also Figure 27), ground, and poles. Second, on each successive lap, as the trajectory becomes more aggressive, the geodesic energy tracking error becomes more concentrated at smaller fractions of the theoretical upper bounds. This is potentially due to the fact that the min-norm formulation of the feedback controller in Section 5.1 possesses a “minimally invasive” property akin to trigger or event-based control, in that, the optimal feedback can be zero. However, for the faster trajectories, the feedback is non-zero for a greater proportion of the time, thereby resulting in proportionally tighter tracking. Further, we observe that the geodesic energy at higher velocities would violate the bounds generated for lower velocities. This suggests that the bounds computed during CCM synthesis are not overly conservative, which confirms the numerical results presented in Section 7.2.4.

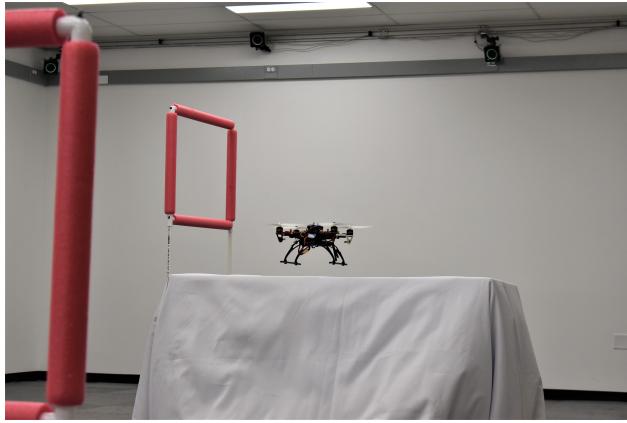


Figure 27: Close-up of quad passing near the box obstacle and experiencing complex (unmodeled) aerodynamic disturbances due to the downwash interaction.

Finally, we stress that the purpose of this experiment was to illustrate the feasibility of converting the rich theoretical analysis into a practical tool capable of running on modern hardware, and obtain true validation of the complete planning methodology. Further performance improvements may be obtained through incorporating richer dynamic models and leveraging higher rate dedicated controllers.

9 Conclusions and Future Work

We presented a framework for robust motion planning for robots with nonlinear dynamics subject to bounded disturbances, input constraints, and online state constraints. Our approach allows one to generate certifiably safe trajectories *online* when faced with a priori unknown environment. We leveraged recent advances in contraction theory in the form of CCMs to synthesize a tracking feedback controller and an optimized invariant tube, valid for *any* dynamically feasible trajectory. Such an invariant tube is consequently used as a robustness margin during online trajectory generation. Importantly, our approach is modular in the sense that we do *not* put any constraints on the planner itself; indeed we demonstrated our approach using a variety of planning techniques such as pseudospectral discretization with nonlinear programming and sampling-based planning with polynomial spline smoothing. The two relevant components of the approach are the tracking

feedback controller and the invariant tube – both of which can be interfaced with the vast majority of planning techniques. We validated our approach in simulation and on a quadrotor hardware testbed, and extensively tested the *limits of performance* of the synthesized controller to study potential failure modes. This work introduces several promising future directions, discussed next.

Computation: The primary computational challenge stems from the offline synthesis of the CCM. Computing CCMs requires solving a semi-infinite optimization problem (Hettich and Kortanek, 1993) – a daunting task. SOS programming allows us to obtain sufficient certificates for the infinite-dimensional constraints but at a cost of scalability. The size of the equivalent SDP re-formulation of a SOS constraint scales as $O(n^d)$ where n is the state-space dimension and $2d$ is the degree of the polynomial. While there are some alternative relaxations available based on LP and SOCP (Ahmadi and Majumdar, 2014), these are more conservative than the SDP re-formulation. Looking forward, two interesting avenues of work involve: (i) leveraging non-polynomial SOS decompositions, and (ii) leveraging sampling-based algorithms to obtain high probability asymptotic guarantees, e.g., as in (Zhang et al., 2010).

Conservatism: Our analysis is conservative due to the fact that we derive a *globally valid* invariant tube, as opposed to the funnel library approach in which one computes bounds valid locally around a trajectory. This issue was slightly tempered by the introduction of less conservative *time-varying* tubes, used within a local receding-horizon re-planner as for the planar quadrotor example. A more promising method for reducing the conservatism is to partition the state space into regions in which CCMs are computed *locally*, while ensuring continuity/smoothness at the region boundaries. Such a computational approach may also allow the use of lower-order polynomial expansions, thereby reducing the size of the SOS programs.

A second limitation of the approach is that the invariant tubes are computed based on a worst-case disturbance bound assumption. A less conservative solution may be achieved by decomposing the disturbance term into an unknown constant (or slowly-varying) mean, estimated online, plus zero-mean stochastic noise. In this way, one may leverage modern adaptive and disturbance estimation techniques to counteract the constant disturbance term, and a stochastic modification of CCM theory to obtain *probabilistic* invariance guarantees, e.g., based on super-martingale analysis (Steinhardt and Tedrake, 2012).

Perception Uncertainty: In this work, we assumed full state knowledge and the ability to perfectly sense obstacles in the neighborhood of the robot. In reality, noisy sensors inject uncertainty into both these terms. In this case, providing *almost sure* guarantees is practically infeasible and one must additionally incorporate perception/sensing uncertainty, for instance, in a Bayesian sense, to obtain (at most) probabilistic guarantees. This is, however, a well-known open problem.

We believe that the modular approach presented herein for automatically synthesizing feedback controllers that are optimized for robust performance, and accompanied by guarantees readily integrable into existing planning algorithms, serves as a valuable tool for a practitioner to systematically balance performance and safety.

References

- Adelstein IM and Epstein J (2017) Morse theory for the uniform energy. *Journal of Geometry* 108(3): 1193–1205.

- Agha-mohammadi A, Chakravorty S and Amato NM (2014) FIRM: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *Int. Journal of Robotics Research* 33(2): 268–304.
- Ahmadi AA and Majumdar A (2014) DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization. In: *IEEE Annual Conf. on Information Sciences and Systems*.
- Ahmadi AA and Majumdar A (2016) Some applications of polynomial optimization in operations research and real-time decision making. *Optimization Letters* 20(4): 709–729.
- Althoff D, Althoff M and Scherer S (2015) Online safety verification of trajectories for unmanned flight with offline computed robust invariant sets. In: *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*.
- Althoff M and Dolan J (2014) Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics* 30(4): 903–918.
- Aminzarey Z and Sontag ED (2014) Contraction methods for nonlinear systems: A brief introduction and some open problems. In: *Proc. IEEE Conf. on Decision and Control*.
- Angeli D (2002) A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control* 47(3): 410–422.
- Angeli D (2009) Further results on incremental input-to-state stability. *IEEE Transactions on Automatic Control* 54(6): 1386–1391.
- ApS M (2017) MOSEK optimization software. Available at <https://mosek.com/>.
- Bansal S, Chen M, Herbert S and Tomlin CJ (2017) Hamilton-Jacobi reachability: A brief overview and recent advances. In: *Proc. IEEE Conf. on Decision and Control*.
- Barry AJ, Majumdar A and Tedrake R (2012) Safety verification of reactive controllers for uav flight in cluttered environments using barrier certificates. In: *Proc. IEEE Conf. on Robotics and Automation*.
- Battilotti S and De Santis A (2003) Stabilization in probability of nonlinear stochastic systems with guaranteed region of attraction and target set. *IEEE Transactions on Automatic Control* 48(9): 1585–1599.
- Bayer F, Bürger M and Allgöwer F (2013) Discrete-time incremental ISS: A framework for robust NMPC. In: *European Control Conference*.
- Betts JT (2010) *Practical Methods for Optimal Control and Estimation using Nonlinear Programming*. Second edition. SIAM.
- Blackmore L, Li H and Williams B (2006) A probabilistic approach to optimal robust path planning with obstacles. In: *American Control Conference*.
- Blackmore L, Ono M and Williams BC (2011) Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics* 27(6): 1080–1094.

- Bravo JM, Alamo T and Camacho EF (2006) Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica* 42(10): 1745–1751.
- Buehler EA, Paulson JA and Mesbah A (2016) Lyapunov-based stochastic nonlinear model predictive control: Shaping the state probability distribution functions. In: *American Control Conference*.
- Burridge RR, Rizzi AA and Koditschek DE (1999) Sequential composition of dynamically dexterous robot behaviors. *Int. Journal of Robotics Research* 18(6): 534–555.
- Cannon M, Buerger J, Kouvaritakis B and Rakovic S (2011) Robust tubes in nonlinear model predictive control. *IEEE Transactions on Automatic Control* 56(8): 1942–1947.
- Carson III JM, Açıkmese B, Murray RM and MacMartin DG (2013) A robust model predictive control algorithm augmented with a reactive safety mode. *Automatica* 49(5): 1251–1260.
- Charnes A and Cooper WW (1959) Chance-constrained programming. *Management Science* 6(1): 73–79.
- Chen M, Herbert SL, Vashishtha MS, Bansal S and Tomlin CJ (2016) Decomposition of reachable sets and tubes for a class of nonlinear systems. Available at <https://arxiv.org/abs/1611.00122>.
- Crouch PE and van der Schaft AJ (1987) *Variational and Hamiltonian Control Systems*. Springer.
- Faessler M, Franchi A and Scaramuzza D (2018) Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories. *IEEE Robotics and Automation Letters* 3(2): 620–626.
- Farina M and Scattolini R (2012) Tube-based robust sampled-data MPC for linear continuous-time systems. *Automatica* 48(7): 1473–1476.
- Fleming J, Kouvaritakis B and Cannon M (2015) Robust tube MPC for linear systems with multiplicative uncertainty. *IEEE Transactions on Automatic Control* 60(4): 1087–1092.
- Forni F and Sepulchre R (2014) A differential Lyapunov framework for contraction analysis. *IEEE Transactions on Automatic Control* 59(3): 614–628.
- Fridovich-Keil D, Herbert SL, Fisac JF, Deglurkar S and Tomlin CJ (2018) Planning, fast and slow: A framework for adaptive real-time safe trajectory planning. In: *Proc. IEEE Conf. on Robotics and Automation*.
- Gillula JH, Huang H, Vitus MP and Tomlin CJ (2010) Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice. In: *Proc. IEEE Conf. on Robotics and Automation*.
- Helwa MK, Heins A and Schoellig AP (2019) Provably robust learning-based approach for high-accuracy tracking control of lagrangian systems. *IEEE Robotics and Automation Letters* 4(2): 1587–1594.

- Herbert SL, Chen M, Han S, Bansal S, Fisac JF and Tomlin CJ (2017) FaSTrack: a modular framework for fast and guaranteed safe motion planning. In: *Proc. IEEE Conf. on Decision and Control*.
- Hettich R and Kortanek KO (1993) Semi-infinite programming: Theory, methods, and applications. *SIAM Review* 35(3): 380–429.
- Iwasaki T and Hara S (2005) Generalized KYP lemma: unified frequency domain inequalities with design applications. *IEEE Transactions on Automatic Control* 50(1): 41–59.
- Janson L, Schmerling E, Clark A and Pavone M (2015a) Fast Marching Tree: a fast marching sampling-based method for optimal motion planning in many dimensions. *Int. Journal of Robotics Research* 34(7): 883–921.
- Janson L, Schmerling E and Pavone M (2015b) Monte Carlo motion planning for robot trajectory optimization under uncertainty. In: *Int. Symp. on Robotics Research*.
- Jouffroy J (2003) A simple extension of contraction theory to study incremental stability properties. In: *European Control Conference*.
- Kaelbling LP, Littman ML and Cassandra AR (1998) Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1-2): 99–134.
- Kai JM, Allibert G, Hua MD and Hamel T (2017) Nonlinear feedback control of quadrotors exploiting first-order drag effects. In: *IFAC World Congress*.
- Khalil HK (2002) *Nonlinear Systems*. Third edition. Prentice Hall.
- Kögel M and Findeisen R (2015) Discrete-time robust model predictive control for continuous-time nonlinear systems. In: *American Control Conference*.
- Köhler J, Soloperto R, A MM and Allgöwer F (2019) A computationally efficient robust model predictive control framework for uncertain nonlinear systems. Submitted to IEEE Transactions on Automatic Control; Available at: https://www.ist.uni-stuttgart.de/de/institut/team/PDFs_MA-Seiten/JK/Robust_Nonlin.pdf .
- Kong S, Gao S, Chen W and Clarke E (2015) dreach: δ -Reachability analysis for hybrid systems. In: *Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems* .
- Kurniawati H, Hsu D and Lee WS (2008) SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In: *Robotics: Science and Systems*.
- Langson W, Chryssochoos I, Raković SV and Mayne DQ (2004) Robust model predictive control using tubes. *Automatica* 40(1): 125–133.
- Leung K and Manchester IR (2017) Chebyshev pseudospectral method for nonlinear stabilization using control contraction metrics. In: *American Control Conference*. To Appear.
- Limon D, Alvarado I, Alamo T and Camacho EF (2010) Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control* 20(3): 248–260.

- Lin Y and Sontag ED (1991) A universal formula for stabilization with bounded controls. *System and Control Letters* 16(6): 393–397.
- Lohmiller W and Slotine JJE (1998) On contraction analysis for non-linear systems. *Automatica* 34(6): 683–696.
- Lopez BT, Slotine JJE and How JP (2018) Robust collision avoidance via sliding control. In: *Proc. IEEE Conf. on Robotics and Automation*.
- Lu Z and Pong TK (2011) Minimizing condition number via convex programming. *SIAM Journal on Matrix Analysis and Applications* 32(4): 1193–1211.
- Luders B, Ellertson A, How JP and Sugel I (2016) Wind uncertainty modeling and robust trajectory planning for autonomous parafoils. *AIAA Journal of Guidance, Control, and Dynamics* 39(7): 1614–1630.
- Majumdar A and Tedrake R (2012) Robust online motion planning with regions of finite time invariance. In: *Workshop on Algorithmic Foundations of Robotics*.
- Majumdar A and Tedrake R (2017) Funnel libraries for real-time robust feedback motion planning. *Int. Journal of Robotics Research* 36(8): 947–982.
- Majumdar A, Tobenkin MM and Tedrake R (2012) Algebraic verification for parameterized motion planning libraries. In: *American Control Conference*.
- Manchester I, Tang JZ and Slotine JJE (2015) Unifying classical and optimization-based methods for robot tracking control with control contraction metrics. In: *Int. Symp. on Robotics Research*.
- Manchester IR and Slotine JJE (2014) Output-feedback control of nonlinear systems using control contraction metrics and convex optimization. In: *Australian Control Conference*.
- Manchester IR and Slotine JJE (2017) Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Transactions on Automatic Control* In Press.
- Manchester Z and Kuindersma S (2017) DIRTREL: Robust trajectory optimization with ellipsoidal disturbances and LQR feedback. In: *Robotics: Science and Systems*.
- Manchester Z and Kuindersma S (2019) Robust direct trajectory optimization using approximate invariant funnels. *Autonomous Robots* 43(2): 375–387.
- Mayne DQ (2014) Model predictive control: Recent developments and future promise. *Automatica* 50(12): 2967–2986.
- Mayne DQ, Kerrigan EC, van Wyk EJ and Falugi P (2011) Tube-based robust nonlinear model predictive control. *Int. Journal of Robust and Nonlinear Control* 21(11): 1341–1353.
- Mayne DQ, Seron MM and Raković SV (2005) Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica* 41(2): 219–224.
- Nguyen VB, Shew RL and Xia Y (2016) Maximizing the sum of a generalized Rayleigh quotient and another Rayleigh quotient on the unit sphere via semidefinite programming. *Journal of Global Optimization* 64(2): 399–416.

- Ono M, Williams BC and Blackmore L (2013) Probabilistic planning for continuous dynamic systems under bounded risk. *Journal of Artificial Intelligence Research* 46(1): 511–577.
- Parrilo PA (2000) *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD Thesis, Massachusetts Inst. of Technology.
- Pin G, Raimondo DM, Magni L and Parisini T (2009) Robust model predictive control of nonlinear systems with bounded and state-dependent uncertainties. *IEEE Transactions on Automatic Control* 54(7): 1681–1687.
- Platt R, Kaelbling L, Tomas LP and Tedrake R (2012) Non-Gaussian belief space planning: Correctness and complexity. In: *Proc. IEEE Conf. on Robotics and Automation*.
- Prajna S, Jadbabaie A and Pappas GJ (2007) A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control* 52(8): 1415–1428.
- Prentice S and Roy N (2009) The belief roadmap: Efficient planning in linear POMDPs by factoring the covariance. *Int. Journal of Robotics Research* 28(11-12): 1448–1465.
- Primbs JA, Nevistić V and Doyle JC (1999) Nonlinear optimal control: A control Lyapunov function and receding horizon perspective. *Asian Journal of Control* 1(1): 14–24.
- Putinar M (1993) Positive polynomials on compact semi-algebraic sets. *Indiana Univ. Mathematics Journal* 42(3): 969–984.
- Quinlan S and Khatib O (1993) Elastic bands: Connecting path planning and control. In: *Proc. IEEE Conf. on Robotics and Automation*.
- Raković SV (2009) Set theoretic methods in model predictive control. In: *Nonlinear Model Predictive Control*. Springer Berlin Heidelberg.
- Raković SV, Kerrigan EC, Mayne DQ and Lygeros J (2006a) Reachability analysis of discrete-time systems with disturbances. *IEEE Transactions on Automatic Control* 51(4): 546–561.
- Rakovic SV, Kouvaritakis B, Cannon M, Panos C and Findeisen R (2012) Parameterized tube model predictive control. *IEEE Transactions on Automatic Control* 57(11): 2746–2761.
- Raković SV, Teel AR, Mayne DQ and Astolfi A (2006b) Simple robust control invariant tubes for some classes of nonlinear discrete time systems. In: *Proc. IEEE Conf. on Decision and Control*.
- Rayguru MM and Kar IN (2015) Contraction based stabilization of approximate feedback linearizable systems. In: *European Control Conference*.
- Reyes-Báez R, van der Schaft A and Jayawardhana B (2017) Tracking control of fully-actuated port-hamiltonian mechanical systems via sliding manifolds and contraction analysis. In: *IFAC World Congress*.
- Richter C, Bry A and Roy N (2016) Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In: *Robotics Research*. Springer.

- Rubagotti M, Raimondo DM, Ferrara A and Magni L (2011) Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems. *IEEE Transactions on Automatic Control* 56(3): 556–570.
- Scott JK and Barton PI (2013) Bounds on the reachable sets of nonlinear control systems. *Automatica* 49(1): 93–100.
- Sharma BB and Kar IN (2009) Contraction based adaptive control of a class of nonlinear systems. In: *American Control Conference*.
- Simpson-Porco JW and Bullo F (2014) Contraction theory on Riemannian manifolds. *System and Control Letters* 65(1): 74–80.
- Singh S, Majumdar A, Slotine JJE and Pavone M (2017) Robust online motion planning via contraction theory and convex optimization. In: *Proc. IEEE Conf. on Robotics and Automation*. Extended Version, Available at <http://asl.stanford.edu/wp-content/papercite-data/pdf/Singh.Majumdar.Slotine.Pavone.ICRA17.pdf>.
- Slotine JJE (2007) Sliding controller design for non-linear systems. *Int. Journal of Control* 38(2): 421–434.
- Sontag ED (2010) Contractive systems with inputs. In: *Perspectives in Mathematical System Theory, Control, and Signal Processing*. Springer Berlin Heidelberg.
- Sontag ED, Margaliot M and Tuller T (2014) On three generalizations of contraction. In: *Proc. IEEE Conf. on Decision and Control*.
- Spivak M (1999) *A Comprehensive Introduction to Differential Geometry, Vol. I*. Third edition. Publish or Perish, Inc.
- Steinhardt J and Tedrake R (2012) Finite-time regional verification of stochastic non-linear systems. *Int. Journal of Robotics Research* 31(7): 901–923.
- Sun W, Patil S and Alterovitz R (2015) High-frequency replanning under uncertainty using parallel sampling-based motion planning. *IEEE Transactions on Robotics* 31(1): 104–116.
- Tal E and Karaman S (2018) Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness. In: *Proc. IEEE Conf. on Decision and Control*.
- Tedrake R, Manchester IR, Tobenkin M and Roberts JW (2010) LQR-trees: Feedback motion planning via sums-of-squares verification. *Int. Journal of Robotics Research* 29(8): 1038–1052.
- Tobenkin MM, Permenter F and Megretski A (2013) SPOTLESS polynomial and conic optimization. Software available from <https://github.com/spottoolbox/spotless>.
- Tomlin CJ, Mitchell I, Bayen AM and Oishi M (2003) Computational techniques for the verification of hybrid systems. *Proc. of the IEEE* 91(7): 986–1001.
- Villanueva ME, Quirynen R, Diehl M, Chachuat B and Houska B (2017) Robust MPC via minmax differential inequalities. *Automatica* 77(1): 311–321.

Wang W and Slotine JJE (2005) On partial contraction analysis for coupled nonlinear oscillators. *Biological Cybernetics* 92(1): 38–53.

Yu S, Böhm C, Chen H and Allgöwer F (2010) Robust model predictive control with disturbance invariant sets. In: *American Control Conference*.

Yu S, Maier C, Chen H and Allgöwer F (2013) Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems. *System and Control Letters* 62(2): 194–200.

Zamani M, van de Wouw N and Majumdar R (2013) Backstepping controller synthesis and characterizations of incremental stability. *System and Control Letters* 62(10): 949–962.

Zhang L, Wu SY and López MA (2010) A new exchange method for convex semi-infinite programming. *SIAM Journal on Optimization* 20(6): 2959–2977.

Appendix A Technical Lemma

Proof of Lemma 5.1. Decompose η_z as $\eta_{z_Y} + \eta_{z_{Y^\perp}}$, where $\eta_{z_Y} \in \text{Col}(Y^T)$ and $\eta_{z_{Y^\perp}} \in \mathcal{N}(Y)$. Now write $\eta_{z_Y} = \epsilon \hat{\eta}_{z_Y}$ and $\eta_{z_{Y^\perp}} = \epsilon_\perp \hat{\eta}_{z_{Y^\perp}}$ where $\epsilon, \epsilon_\perp \geq 0$, $\epsilon^2 + \epsilon_\perp^2 \leq 1$, and $\hat{\eta}_{z_Y}$ and $\hat{\eta}_{z_{Y^\perp}}$ are (non-zero) unit vectors contained in $\text{Col}(Y^T)$ and $\mathcal{N}(Y)$, respectively. Substituting these expressions into inequality (44) above yields

$$\theta \left(\epsilon^2 \hat{\eta}_{z_Y}^T S \hat{\eta}_{z_Y} + \epsilon_\perp^2 \hat{\eta}_{z_{Y^\perp}}^T S \hat{\eta}_{z_{Y^\perp}} + 2\epsilon\epsilon_\perp \hat{\eta}_{z_{Y^\perp}}^T S \hat{\eta}_{z_Y} \right) \leq 2\bar{\delta}_u \epsilon \|Y \hat{\eta}_{z_Y}\|. \quad (59)$$

Now, if $\eta_z \in \mathcal{N}(Y)$, i.e., $\epsilon = 0$, then condition (42) is necessary and sufficient for inequality (44). Thus, we consider the case where $\epsilon > 0$. Notice that

$$\max_{\hat{\eta}_{z_{Y^\perp}} \in \mathcal{N}(Y)} \hat{\eta}_{z_{Y^\perp}}^T S \hat{\eta}_{z_Y} = \|\bar{Y}_\perp^T S \hat{\eta}_{z_Y}\|,$$

and by condition (42), $\hat{\eta}_{z_{Y^\perp}}^T S \hat{\eta}_{z_Y} < 0$. Thus, by upper-bounding the left hand side of the inequality in (59) and rearranging, we obtain the following *sufficient* condition:

$$\left(\epsilon \frac{\hat{\eta}_{z_Y}^T S \hat{\eta}_{z_Y}}{\|Y \hat{\eta}_{z_Y}\|} + 2\epsilon_\perp \frac{\|\bar{Y}_\perp^T S \hat{\eta}_{z_Y}\|}{\|Y \hat{\eta}_{z_Y}\|} \right) \leq \frac{2\bar{\delta}_u}{\theta}, \quad (60)$$

for all $\hat{\eta}_{z_Y} \in \text{Col}(Y^T)$, and ϵ, ϵ_\perp . Now, given that the columns of \bar{Y} are an orthonormal basis for $\text{Col}(Y^T)$, then $\hat{\eta}_{z_Y}$ may be expressed as $\bar{Y} \bar{\eta}_{z_Y}$ where $\bar{\eta}_{z_Y} \in \mathcal{S}^{m-1}$, the $(m-1)$ -unit sphere. Substituting this expression into the inequality above yields the following sufficient condition for inequality (44):

$$\max_{\substack{\bar{\eta}_{z_Y} \in \mathcal{S}^{m-1} \\ \epsilon, \epsilon_\perp \geq 0 \\ \epsilon^2 + \epsilon_\perp^2 \leq 1}} \epsilon \frac{\bar{\eta}_{z_Y}^T S Y \bar{\eta}_{z_Y}}{\|Y \bar{Y} \bar{\eta}_{z_Y}\|} + 2\epsilon_\perp \frac{\|\bar{Y}_\perp^T S \bar{Y} \bar{\eta}_{z_Y}\|}{\|Y \bar{Y} \bar{\eta}_{z_Y}\|} \leq \frac{2\bar{\delta}_u}{\theta}.$$

For fixed $(\epsilon, \epsilon_\perp)$, the maximization over $\bar{\eta}_{z_Y} \in \mathcal{S}^{m-1}$ above belongs to the class of sum-of-ratios fractional programming and is in general, NP-complete. Recently in (Nguyen et al., 2016), the

authors presented a two-stage algorithm using tight SDP relaxations of parameterized subproblems for maximizing the sum of a generalized Rayleigh quotient and another Rayleigh quotient on the unit sphere. For our purposes, we derive a simpler yet suboptimal approximation by decoupling the maximization as

$$\max_{\substack{\epsilon, \epsilon_\perp \geq 0 \\ \epsilon^2 + \epsilon_\perp^2 \leq 1}} \left(\epsilon \max_{\bar{\eta}_{z_Y} \in \mathcal{S}^{m-1}} \frac{\bar{\eta}_{z_Y}^T S_Y \bar{\eta}_{z_Y}}{\|Y\bar{Y}\bar{\eta}_{z_Y}\|} + 2\epsilon_\perp \max_{\bar{\eta}_{z_Y} \in \mathcal{S}^{m-1}} \frac{\|\bar{Y}_\perp^T S\bar{Y} \bar{\eta}_{z_Y}\|}{\|Y\bar{Y}\bar{\eta}_{z_Y}\|} \right).$$

The two inner maximizations may be further upper-bounded as:

$$\max_{\bar{\eta}_{z_Y} \in \mathcal{S}^{m-1}} \frac{\bar{\eta}_{z_Y}^T S_Y \bar{\eta}_{z_Y}}{\|Y\bar{Y}\bar{\eta}_{z_Y}\|} \leq \frac{\bar{\lambda}(S_Y)}{\underline{\sigma}_{>0}(Y\bar{Y})} \quad \text{and} \quad \max_{\bar{\eta}_{z_Y} \in \mathcal{S}^{m-1}} \frac{\|\bar{Y}_\perp^T S\bar{Y} \bar{\eta}_{z_Y}\|}{\|Y\bar{Y}\bar{\eta}_{z_Y}\|} \leq \frac{\bar{\sigma}(\bar{Y}_\perp^T S\bar{Y})}{\underline{\sigma}_{>0}(Y\bar{Y})}.$$

Then, the outer maximization over $(\epsilon, \epsilon_\perp)$ is of an affine expression over a convex set:

$$\max_{\substack{\epsilon, \epsilon_\perp \geq 0 \\ \epsilon^2 + \epsilon_\perp^2 \leq 1}} \epsilon \frac{\bar{\lambda}(S_Y)}{\underline{\sigma}_{>0}(Y\bar{Y})} + 2\epsilon_\perp \frac{\bar{\sigma}(\bar{Y}_\perp^T S\bar{Y})}{\underline{\sigma}_{>0}(Y\bar{Y})}.$$

Now, since the coefficient of ϵ_\perp is nonnegative, the optimal $(\epsilon, \epsilon_\perp)$ is either $(0, 1)$ (this occurs when the coefficient of ϵ is nonpositive), or lies along the nonnegative quadrant of the circle $\epsilon^2 + \epsilon_\perp^2 = 1$ with $\epsilon > 0$. We can ignore the solution $(0, 1)$ since this corresponds to the case where $\eta_z \in \mathcal{N}(Y)$. For the case $\epsilon > 0$, the maximization above evaluates to

$$\sqrt{\left(\frac{\bar{\lambda}(S_Y)}{\underline{\sigma}_{>0}(Y\bar{Y})} \right)^2 + 4 \left(\frac{\bar{\sigma}(\bar{Y}_\perp^T S\bar{Y})}{\underline{\sigma}_{>0}(Y\bar{Y})} \right)^2}.$$

Thus, we arrive at the sufficient condition stated in (43). \square