Guaranteed Contraction Control in the Presence of Imperfectly Learned Dynamics

Pan Zhao  
Ziyao Guo  
Yikun Cheng  
Aditya Gahlawat  
Naira Hovakimyan  

University of Illinois at Urbana Champaign, Urbana, IL, 61801

Abstract

This paper presents an approach for trajectory-centric learning control based on contraction metrics and disturbance estimation for nonlinear systems subject to matched uncertainties. The approach allows for the use of a broad class of model learning tools including deep neural networks to learn uncertain dynamics while still providing guarantees of transient tracking performance throughout the learning phase, including the special case of no learning. Within the proposed approach, a disturbance estimation law is proposed to estimate the pointwise value of the uncertainty, with pre-computable estimation error bounds (EEBs). The learned dynamics, the estimated disturbances, and the EEBs are then incorporated in a robust Riemannian energy condition to compute the control law that guarantees exponential convergence of actual trajectories to desired ones throughout the learning phase, even when the learned model is poor. On the other hand, with improved accuracy, the learned model can be incorporated in a high-level planner to plan better trajectories with improved performance, e.g., lower energy consumption and shorter travel time. The proposed framework is validated on a planar quadrotor navigation example.

Keywords: Safe learning, contraction theory, disturbance estimation, neural networks

1. Introduction

Robots or autonomous systems (ASs) generally have nonlinear dynamics and often need to operate in uncertain environments subject to significant dynamic uncertainties and disturbances. Planning and executing a trajectory is one of the most common ways for a robot or AS to achieve a mission. However, the presence of dynamic uncertainties and disturbances, together with the nonlinear dynamics, brings significant challenges to safely plan and execute a trajectory. In this paper, we propose a robust learning control approach that allows for the use of a broad class of model learning tools such as deep neural networks (DNNs) to learn uncertain dynamics while providing guaranteed tracking performance in the form of exponential trajectory convergence throughout the learning phase.

1.1. Related Work

Control theoretic methods to deal with uncertain dynamics can be roughly classified into adaptive/robust approaches and learning-based approaches. Robust approaches such as $H_{\infty}$ control (Zhou and Doyle, 1998), sliding mode control (Edwards and Spurgeon, 1998), and robust/tube
model predictive control (MPC) (Mayne et al., 2005; Mayne, 2014), usually consider parametric uncertainties or bounded disturbances, and design controllers to achieve certain performance despite the presence of such uncertainties. Disturbance-observer (DOB) based control and related methods such as active disturbance rejection control (ADRC) (Han, 2009) usually lump parametric uncertainties, unmodeled dynamics, and external disturbances together as a “total disturbance”, estimate it via an observer such as DOB and extended state observer (ESO) (Han, 2009), and then compute control actions to compensate for the estimated disturbance (Chen et al., 2015). On the other hand, adaptive control methods such as model reference adaptive control (MRAC) (Ioannou and Sun, 2012) and $L_1$ adaptive control (Hovakimyan and Cao, 2010) rely on online adaptation to estimate parametric or non-parametric uncertainties and use of the estimated value in the control design to provide stability and performance guarantees. While significant progress has been made in the linear setting, trajectory tracking for nonlinear uncertain systems with transient performance guarantees has been less successful in terms of analytical quantification, yet it is required for safety guarantees of robots and ASs.

On the other hand, recent years have witnessed increased use of machine learning (ML) tools for control, both for dynamics learning and control design. This is partially due to the expressive power associated with these ML tools that has led to the success of applying these tools in a variety of domains such as computer vision (LeCun et al., 2015) and games (Mnih et al., 2015). In terms of controlling uncertain systems, learning-based methods can be further classified into model-free and model-based approaches. Model-free approaches are mainly investigated in the reinforcement learning (RL) community (Sutton and Barto, 2018) and seek to directly learn a controller (or a policy) from the data collected during interaction with the controlled system. In contrast, model-based approaches try to first learn a model for the uncertain dynamics and then incorporate the learned model into control-theoretic approaches to generate the control law. Because of the sampling efficiency, the easiness to incorporate prior knowledge (e.g., a physics based model) and provide performance or safety guarantees, model-based approaches are more applicable to robots and ASs that operate in the real world. Along this direction, a few ML tools such as Gaussian process regression (GPR) and (deep) neural networks (NNs) have been used to learn uncertain dynamics in the context of robust control (Berkenkamp and Schoellig, 2015; Shi et al., 2019), adaptive control (Chowdhary et al., 2014; Gahlawat et al., 2020, 2021), MPC (Williams et al., 2017; Hewing et al., 2019; Chua et al., 2018) and barrier-certified control (Khojasteh et al., 2020).

However, for model-based learning control with safety and/or performance guarantees, researchers have largely relied on GPR (Khojasteh et al., 2020; Berkenkamp and Schoellig, 2015; Berkenkamp et al., 2016; Chowdhary et al., 2014; Gahlawat et al., 2020, 2021; Hewing et al., 2019) due to the inherent ability to quantify the uncertainties in the prediction of the learned model. DNNs were used in the context of MRAC in an online-update fashion to learn state-dependent uncertainties in (Joshi and Chowdhary, 2019; Joshi et al., 2020). However, only asymptotic (i.e., no transient) performance guarantee was provided; additionally, the nominal dynamics is limited to a linear system. In (Shi et al., 2019), the authors used DNNs to model ground effects and showed good performance when the learned model well approximated the true uncertainties. However, the tracking performance during the learning transients cannot be guaranteed.

Contraction theory (Lohmiller and Slotine, 1998) provides a tool for analyzing nonlinear systems in a differential framework and is focused on studying the convergence between pairs of state trajectories towards each other, i.e., incremental stability. It has recently been extended for constructive control design, e.g., via control contraction metrics (CCMs) for both deterministic (Manch-
ester and Slotine, 2017) and stochastic systems (Tsukamoto and Chung, 2020a,b). Compared to incremental Lyapunov function approaches for studying incremental stability, contraction metrics presents an intrinsic characterization of incremental stability (i.e., invariant under change of coordinates); additionally, the search for a CCM can be formulated as a convex optimization problem and solved using sum of squares (SOS) programming (Singh et al., 2019) and DNN optimization (Tsukamoto and Chung, 2020a; Sun et al., 2020). CCM-based adaptive control has been developed to address parametric uncertainties in (Lopez and Slotine, 2020) and non-parametric uncertainties in (Lakshmanan et al., 2020). The case of bounded disturbances in contraction-based control has also been addressed by leveraging input-to-stability analysis (Singh et al., 2019) or robust CCM (Zhao et al., 2021a; Manchester and Slotine, 2018). The problem we consider in this paper (nonlinear systems subject to matched state-dependent uncertainties) is mostly close to the one addressed in (Gahlawat et al., 2021), which uses GPR to learn the uncertain dynamics while relying on a contraction based $L_1$ adaptive controller proposed in (Lakshmanan et al., 2020) to provide transient tracking performance guarantees throughout the learning phase. In contrast with (Gahlawat et al., 2021), our learning control framework allows for use of DNNs to learn the uncertain dynamics while still providing transient tracking performance guarantees. Moreover, our transient performance guarantees are in the form of exponential convergence to the desired state trajectory despite the presence of uncertain dynamics or model errors, which is achieved by incorporating an error bound for the estimated disturbance into a robust condition to constrain the decreasing rate of the Riemannian energy, as opposed to a bounded distance from the the desired state trajectory provided in (Gahlawat et al., 2021).

1.2. Statement of Contributions

We present a robust trajectory-centric learning control framework for nonlinear systems subject to matched state-dependent uncertainties based on contraction metrics and disturbance estimation. To the best of our knowledge, our learning control framework is the first one that allows for the use of DNNs to learn uncertain dynamics while still providing transient tracking performance guarantees throughout the learning phase, including the case of no learning. Our framework leverages an estimation law to estimate the pointwise value of the the dynamic uncertainty, with pre-computable estimation error bounds (EEBs). The uncertainty estimation, and the EEBs are then incorporated into a robust inequality condition constraining the decreasing rate of the Riemannian energy, which guarantees exponential convergence of actual trajectories to desired trajectories (planned using the learned dynamics). Due to the uncertainty estimation and compensation mechanism, our approach provides exponential convergence to the desired trajectory throughout the learning phase, even when the learned model is poor. On the other hand, with improved accuracy, the learned model can help optimize other performance criteria, e.g., energy consumption, beyond trajectory tracking. We demonstrate the proposed framework using a planar quadrotor example.

Notations. Let $\mathbb{R}^n$, $\mathbb{R}^+$ and $\mathbb{R}^{m \times n}$ denote the $n$-dimensional real vector space, the set of non-negative real numbers, and the set of real $m$ by $n$ matrices, respectively. $I$ and $0$ denote an identity matrix, and a zero matrix of compatible dimensions, respectively. $\|\cdot\|$ denotes the 2-norm of a vector or a matrix. Let $\partial_y F(x)$ denote the Lie derivative of the matrix-valued function $F$ at $x$ along the vector $y$. For symmetric matrices $P$ and $Q$, $P > Q$ ($P \geq Q$) means $P - Q$ is positive definite (semidefinite). $\langle X \rangle$ is the shorthand notation of $X + X^\top$. Finally, $\ominus$ denotes the Minkowski set difference.
2. Problem Statement

Consider a nonlinear control-affine system subject to state-dependent uncertainties

\[ \dot{x}(t) = f(x(t)) + B(x(t))(u(t) + d(x(t))), \quad x(0) = x_0, \tag{1} \]

where \( x(t) \in \mathcal{X} \subset \mathbb{R}^n, u(t) \in \mathcal{U} \subset \mathbb{R}^m, f : \mathbb{R}^n \rightarrow \mathbb{R}^n \) and \( B : \mathbb{R}^n \rightarrow \mathbb{R}^m \) are known and locally Lipschitz continuous functions, \( d(x) \) represents the matched model uncertainties. We assume that \( B(x) \) has full column rank for any \( x \in \mathcal{X} \). Suppose \( \mathcal{X} \) is a compact set that contains the origin, and the control constraint set \( \mathcal{U} \) is defined as \( \mathcal{U} \triangleq \{ u \in \mathbb{R}^m : u \leq \bar{u} \leq \hat{u} \} \), where \( \bar{u}, \hat{u} \in \mathbb{R}^m \) denote the lower and upper bounds of all control channels, respectively. Furthermore, we make the following assumption for \( d(x) \).

**Assumption 1** There exist known positive constants \( L_B, L_d \) and \( b_d \) such that for any \( x, y \in \mathcal{X} \), the following inequalities hold:

\[ \| B(x) - B(y) \| \leq L_B \| x - y \|, \quad \| d(x) - d(y) \| \leq L_d \| x - y \|, \quad \| d(x) \| \leq b_d. \tag{2} \]

**Remark 1** Assumption 1 indicates that the uncertain function \( d(x) \) is locally Lipschitz continuous with a known Lipschitz constant and is uniformly bounded by a known constant in the compact set \( \mathcal{X} \). In fact, given the local Lipschitz constant \( L_d \), the uniform bound can always be derived by using Lipschitz continuity if the bound for \( d(x) \) for any \( x \) in \( \mathcal{X} \) is known. For instance, assuming \( \| d(0) \| \leq b_d \), we have \( \| d(x) \| \leq b_d + L_d \max_{x \in \mathcal{X}} \| x \| \). In practice, some prior knowledge about the actual system and the uncertainty may be leveraged to obtain a tighter bound than the one based on the Lipschitz continuity explained earlier, which is why we directly make an assumption on the uniform bound.

With Assumption 1, we will show (in Section 4.3) that the pointwise value of \( d(x(t)) \) at any time \( t \) can be estimated with pre-computable estimation error bounds (EEBs).

We would like to learn the uncertain function \( d(x) \) using model learning tools, while providing guaranteed trajectory tracking performance throughout the learning phase. The performance guarantees provided by the proposed framework are agnostic to the model learning tools used. As a demonstration purpose, we choose to use DNNs, due to their significant potential in dynamics learning attributed to their expressive power and the fact that they have been rarely explored for dynamics learning in the presence of safety and/or performance guarantees. Denoting the DNN-learned function as \( \hat{d}(x) \) and the model error as \( \hat{d}(x) \triangleq d(x) - \hat{d}(x) \), the actual dynamics (1) can be rewritten as

\[ \dot{x} = f(x) + B(x)(u + \hat{d}(x) + \hat{d}(x)) = F_l(x, u) + B(x)\hat{d}(x), \tag{3} \]

where

\[ F_l(x, u) \triangleq f(x) + B(x)\hat{d}(x) + B(x)u. \tag{4} \]

The learned dynamics including the DNN model can now be represented as

\[ \dot{x}^* = F_l(x^*, u^*). \tag{5} \]

**Remark 2** The above setting includes the special case of no learning, corresponding to \( \hat{d}(x) \triangleq 0. \)
The learned dynamics (5) (including the special case of \( \hat{d} = 0 \)) can be incorporated in a motion planner or trajectory optimizer to plan a desired trajectory \((x^*, u^*)\) to minimize a specific cost function, e.g., energy consumption and travel time. Additionally, we would like to design a feedback controller to track the desired state trajectory \(x^*\) with guaranteed tracking performance despite the presence of the model error \(\hat{d}(x)\) throughout the learning phase. As the quality of DNN model \(\hat{d}(x)\) improves with continuous learning, it is expected that the cost associated with the actual state and input trajectory \((x, u)\) will decrease. To summarize, this paper aims to

1. Use DNNs to learn the uncertainty \(d(x)\), and incorporate the learned model, \(\hat{d}(x)\), in planning desired trajectories to minimize specific cost functions subject to constraints;
2. Design a feedback control law to track the planned trajectory with guaranteed tracking performance throughout the learning phase;
3. Enhance the performance, i.e., reducing the costs associated with the actual control and state trajectories, through improved quality of the learned uncertainty model.

To achieve these goals, we will (1) leverage DNNs with bounded Lipschitz constants (by use of spectral normalization) to be consistent with Assumption 1 to learn the uncertainty; (2) use robust CCMs to design the feedback tracking controller; (3) use a disturbance estimation scheme to estimate the pointwise value of the uncertainty \(d(x(t))\) at each time \(t\) with pre-computable EEBs; (4) incorporate the estimated value of \(d(x(t))\) at each time \(t\) and the EEBs into a robust Riemannian energy condition associated with the robust CCMs to compute the control signal that guarantees exponential convergence of actual state trajectories to the desired trajectories despite the existence of the model error.

3. Preliminaries

Since the local Lipschitz constant for the uncertainty \(d(x)\) is already known according to Assumption 1, we would like to ensure that the learned model respects such prior knowledge. To this purpose, we will use spectral-normalized DNNs (SN-DNN) to learn \(d(x)\). Section 3.1 presents a brief overview of SN-DNNs. Additionally, since the proposed control framework is built upon CCM, we give an overview of CCM for uncertainty-free systems in Section 3.2.

3.1. ReLU DNNs with Spectral Normalization

A ReLU DNN is a composition of layers, which uses rectified linear unit activation function \(\phi(\cdot) = \max(\cdot, 0)\) in the layers. It maps from the input \(x\) to the output \(\hat{d}(x, \theta)\) with the internal NN parameters \(\theta\). The mapping \(\hat{d}(x, \theta)\) can be written as combination of activation functions and weights \(\theta = W^1, \ldots, W^{l+1}; \phi(W^{l+1}\phi(W^l(\phi(W^{l-1}(\cdots \phi(W^1 x) \cdots))))\), where \(l\) is the number of layers. Spectral normalization (SN) is a technique for stabilizing the training of DNNs, first introduced in (Miyato et al., 2018). In SN, the Lipschitz constant is the only hyperparameter to be tuned. Theoretically, the Lipschitz constant of a function \(d(x)\) is defined as the smallest value, called Lipschitz norm \(\|d\|_\text{Lip}\), such that \(\forall x, x' : \|d(x) - d(x')\| \leq \|d\|_\text{Lip}\|x - x'\|\). For a linear NN layer \(g(x) = Wx\), the Lipschitz norm is given by \(\|g\|_\text{Lip} = \sup_x (\sigma(\nabla g(x)) = \sup_x (\sigma(W)) = \sigma(W)\), where \(\sigma(\cdot)\) denotes the largest singular value. Hence, the bound of \(\|\hat{d}\|_\text{Lip}\) could be generated by using the fact that the Lipschitz norm of ReLU activation function \(\phi(\cdot)\) is equal to 1 and the inequality...
\[ \| g_1 \circ g_2 \|_{\text{Lip}} \leq \| g_1 \|_{\text{Lip}} \cdot \| g_2 \|_{\text{Lip}}, \] as follows:
\[ \| \hat{d} \|_{\text{Lip}} \leq \| g_{t+1} \|_{\text{Lip}} \| \phi \|_{\text{Lip}} \cdots \| g_1 \|_{\text{Lip}} = \prod_{i=1}^{t+1} \sigma \left( W^i \right). \] (6)

Further, SN could be applied to weight matrices \( W \) of each layer as \( \bar{W} = W/\sigma(W) \cdot \beta_1 \), where \( \beta \) is the pre-designed Lipschitz contant for the ReLU DNN. Considering Assumption 1, we simply set \( \beta = L_d \) when using the ReLU DNNs to learn \( \hat{d} \). See (Shi et al., 2019) for more details.

### 3.2. Control Contraction Metrics (CCMs)

Contraction theory (Lohmiller and Slotine, 1998) draws conclusions on the convergence between pairs of state trajectories towards each other, by studying the evolution of the distance between any two infinitesimally close neighbouring trajectories. CCM generalizes contraction analysis to the controlled dynamic setting, in which the analysis jointly searches for a controller and a metric that describes the contraction properties of the resulting closed-loop system. Following (Lopez and Slotine, 2020; Manchester and Slotine, 2017), we now briefly review CCMs by considering the nominal, i.e., uncertainty-free, system

\[ \dot{x} = f(x) + B(x)u, \] (7)

where \( x(t) \in \mathbb{R}^n \) and \( u(t) \in \mathbb{R}^m \). The differential form of (7) is given by
\[ \dot{\delta} = A(x,u)\delta + B(x)\delta u, \]
where \( A(x,u) \triangleq \partial f/\partial x + \sum_{i=1}^{m} \partial b_i/\partial x u_i \). We first recall some basic results related to CCM.

**Definition 3** (Manchester and Slotine, 2017) The system (7) is said to be universally exponentially stabilizable if, for any feasible desired trajectory \( x^*(t) \) and \( u^*(t) \), a feedback controller can be constructed that for any initial condition \( x(0) \), a unique solution to (7) exists and satisfies
\[ \| x(t) - x^*(t) \| \leq R \| x(0) - x^*(0) \| e^{-\lambda t}, \]
where \( \lambda, R \) are the convergence rate and overshoot, respectively, independent of the initial conditions.

**Lemma 4** (Manchester and Slotine, 2017) If there exists a uniformly bounded metric \( M(x) \), i.e., \( \alpha_1 I \leq M(x) \leq \alpha_2 I \) for some positive constants \( \alpha_1 \) and \( \alpha_2 \), such that
\[ \delta_x^\top MB = 0 \Rightarrow \]
\[ \delta_x^\top \left( M \frac{\partial f}{\partial x} + \partial f M + 2\lambda M \right) \delta_x \leq 0 \text{ and } \left( M \frac{\partial b_i}{\partial x} + \partial b_i M = 0, \text{ for } i = 1, \ldots, m, \right. \] (8)

holds for all \( x \neq 0 \), then the system (7) is universally exponentially stabilizable in the sense of Definition 3 via continuous feedback defined almost everywhere, and everywhere in the neighborhood of the target trajectory with the convergence rate \( \lambda \) and overshoot \( R = \sqrt{\alpha_2/\alpha_1} \).

The condition (8) ensures that each column of \( B(x) \) form a Killing vector field for the metric \( M(x) \) and the dynamics orthogonal to the input are contracting, and is often termed as the strong CCM condition (Manchester and Slotine, 2017). The metric \( M(x) \) satisfying (8) is termed as a (strong) CCM. If the Killing vector condition cannot be satisfied, then a weaker condition can be
used to replace (8), and is given by $\delta_A^\top MB = 0 \implies \delta_A^\top ((MA) + \partial f M + 2\lambda M) \delta_x \leq 0$ for all $\delta_x \neq 0$, $x$, $u$. Due to the dependence of $A(x, u)$ on $u$, the weaker condition can lead to complex feedback controllers. The CCM condition (8) can be transformed into a convex constructive condition for the metric $M(x)$ by a change of variables. Let $W(x) = M^{-1}(x)$ (commonly referred to as the dual metric), and $B_\perp(x)$ be a matrix whose columns span the null space of the input matrix $B$ (i.e., $B_\perp^\top B = 0$). Then, the condition (8) can be cast as convex constructive conditions for $W(x)$:

$$B_\perp^\top \left( \frac{\partial f}{\partial x} W - \partial f W + 2\lambda W \right) B_\perp \leq 0 \quad \text{and} \quad \left\langle \frac{\partial b_i}{\partial x} W \right\rangle - \partial b_i W = 0, \text{ for } i = 1, \ldots, m. \tag{9}$$

The existence of a contraction metric $M(x)$ is sufficient for stabilizability via Lemma 4. What remains is constructing a feedback controller that achieves the universal exponential stabilizability (UES). As mentioned in (Manchester and Slotine, 2017; Singh et al., 2019), one way to derive the controller is to interpret the Riemann energy, $E(x^*(t), x(t))$, as an incremental control Lyapunov function and use it to construct a min-norm controller that renders for any time $t$

$$\dot{E}(x^*(t), x(t)) \leq -2\lambda E(x^*(t), x(t)). \tag{10}$$

Specifically, at any time $t > 0$, given the metric $M(x)$ and a desired/actual state pair $(x^*(t), x(t))$, a geodesic $\gamma(\cdot, t)$ connecting these two states (i.e., $\gamma(0, t) = x^*(t)$ and $\gamma(1, t) = x(t)$) can be computed (e.g., using the pseudospectral method in (Leung and Manchester, 2017)). Consequently, the Riemannian energy of the geodesic, defined as $E(x^*(t), x(t)) = \int_0^1 \gamma_s(s, t)^\top M(\gamma(s, t)) \gamma_s(s, t) ds$, where $\gamma_s(s) \triangleq \frac{\partial \gamma}{\partial s}$, can be calculated. As noted in (Singh et al., 2019), from the formula for the first variation of energy (Do Carmo and Flaherty Francis, 1992),

$$\dot{E}(x^*(t), x(t)) = 2\gamma_s^\top (1, t) M(x(t)) \dot{x}(t) - 2\gamma_s^\top (0, t) M(x^*(t)) \dot{x}^*(t).$$

Therefore, (10) becomes

$$\gamma_s^\top (1, t) M(x(t)) \dot{x}(t) - \gamma_s^\top (0, t) M(x^*(t)) \dot{x}^*(t) \leq -\lambda E(x^*(t), x(t)), \tag{11}$$

where $\dot{x}(t) = f(x(t)) + B(x(t)) u(t)$ and $\dot{x}^*(t) = f(x^*(t)) + B(x^*(t)) u^*(t)$. Therefore, the control signal with a minimal norm for $u(t) - u^*(t)$ can then be obtained by solving the following quadratic programming (QP) problem:

$$u(t) = \arg\min_{k \in \mathbb{R}^m} \|k - u^*(t)\|^2 \text{ subject to } \tag{12}$$

at each time $t$, which is guaranteed to be feasible under the condition (8) (Manchester and Slotine, 2017). The minimization problem (12) is often termed as the pointwise min-norm control problem and has an analytic solution (Freeman and Kokotovic, 2008). The above discussions can be summarized in the following theorem. The proof is trivial by following Lemma 4 and the subsequent discussions, and is thus omitted.

**Theorem 5** (Manchester and Slotine, 2017) Given a nominal system (7), assume that there exists a uniformly bounded metric $W(x)$ that satisfies (9) for all $x \in \mathbb{R}^n$. Then, the control law constructed by solving (12) with $M(x) = W^{-1}(x)$, universally exponentially stabilizes the system (7) in the sense of Definition 3, where $R = \sqrt{\frac{\alpha_2}{\alpha_1}}$ with $\alpha_1$ and $\alpha_2$ being two positive constants satisfying $\alpha_1 I \leq M(x) \leq \alpha_2 I$. 


4. Robust Contraction Control in the Presence of (Imperfectly) Learned Dynamics

In Section 3.2, we have shown that existence of a CCM for a nominal (i.e., uncertainty-free) system can be used to construct a feedback control law to guarantee the universal exponential stabilizability (UES) of the system. In this section, we present an approach based on CCM and disturbance estimation to ensure the UES of the uncertain system (3) even when the learned model ̂(UES) of the system. In this section, we present an approach based on CCM and disturbance estimation to ensure the UES of the uncertain system (3) even when the learned model ̂(UES) of the system. In this section, we present an approach based on CCM and disturbance estimation to ensure the UES of the uncertain system (3) even when the learned model ̂(UES) of the system.

4.1. CCMs and Feasible Trajectories for the True System

In order to apply the contraction method to design a controller to guarantee the UES of the uncertain system (3), we need to first search a valid CCM for it. Following Section 3.2, we can derive the counterparts of the strong CCM condition (8) or (9). Due to the particular structure with (3) attributed to the matched uncertainty assumption, we have the following lemma. A similar observation has been made in (Lopez and Slotine, 2020) for the case of matched parametric uncertainties. The proof is straightforward. One can refer to (Lopez and Slotine, 2020) for more details.

Lemma 6 The strong (dual) CCM condition for the uncertain system (3) with the learned model ̂(x) is the same as , the strong (dual) CCM condition, i.e., (8) (9), for the nominal system.

Remark 7 As a result of Lemma 6, a metric \( M(x) \) (dual metric \( W(x) \)) satisfying the strong condition (8) (9)) for the nominal system (7) is always a CCM (dual CCM) for the true system (3).

Define \( \mathcal{D} = \{ y \in \mathbb{R}^m : \|d(y)\| \leq b_d \} \). Assumption 1 indicates \( d(x) \in \mathcal{D} \) for any \( x \in \mathcal{X} \). As mentioned in Section 3.2, given a CCM and a feasible desired trajectory \( x^*(t) \) and \( u^*(t) \) for a nominal system, a control law can be constructed to ensure exponential convergence of the actual state trajectory \( x(t) \) to the desired state trajectory \( x^*(t) \). In practice, we have access to only learned dynamics (5) instead of the true dynamics to plan a trajectory \( x^*(t) \) and \( u^*(t) \). The following lemma gives the condition when \( x^*(t) \) planned using the (potentially imperfectly) learned dynamics (5) is also a feasible state trajectory for the true system.

Lemma 8 Given a desired trajectory \( x^*(t) \) and \( u^*(t) \) satisfying the learned dynamics (5) with \( x^*(t) \in \mathcal{X} \), if
\[
\begin{align*}
    u^*(t) + ̂d(x^*(t)) &\in \mathcal{U} \ominus \mathcal{D} \quad \forall t \geq 0,
\end{align*}
\]
then, \( x^*(t) \) is also a feasible state trajectory for the true system (1).

Proof. Note that \( ̂u^*(t) \triangleq u^*(t) - ̂d(x^*(t)) = u^*(t) + ̂d(x^*(t)) - (d(x^*(t)) + ̂d(x^*(t))) = u^*(t) + ̂d(x^*(t)) - d(x^*(t)) \). Since \( u^*(t) + ̂d(x^*(t)) \in \mathcal{U} \ominus \mathcal{D} \) and \( -d(x^*(t)) \in \mathcal{D} \), which is due to \( x^*(t) \in \mathcal{X} \) and Assumption 1, we have \( ̂u^*(t) \in \mathcal{U} \). By comparing the dynamics in (5) and (3), we conclude that \( x^*(t) \) and \( ̂u^*(t) \) satisfying the true dynamics (3) and thus are a feasible state and input trajectory for the true system.

Lemma 8 provides a way to verify whether a trajectory planned using the learned dynamics is a feasible trajectory for the true system in the presence of actuator limits. In the absence of such limits, any feasible trajectory for the learned dynamics is also a feasible trajectory for the true dynamics, due to the particular structure of (3) associated with the matched uncertainty assumption.
4.2. Robust Riemannian Energy Condition

Section 3.2 shows that, given a nominal system and a CCM for such a system, a control law can be constructed via solving a QP problem (12) with a condition to constrain the decreasing rate of the Riemannian energy, i.e., condition (11). When considering the uncertain dynamics with the learned model in (3), the condition (11) becomes

$$\gamma_{s^T} (1, t) M(x(t)) \dot{x}(t) - \gamma_{s^T} (0, t) M(x^*(t)) \dot{x}^*(t) \leq -\lambda E(x^*, x),$$  \hspace{1cm} (14)

where $$\dot{x}(t) = f(x(t)) + B(x(t))(u(t) + d(x(t)))$$ represents the true dynamics evaluated at $$x(t)$$ and $$\dot{x}^*(t) = F_1(x^*(t), u^*(t))$$ with $$F_1(x, u)$$ defined in (4). Several observations follow immediately. First, it is clear that (14) is not implementable due to its dependence on the true uncertainty $$d(x(t))$$ through $$\dot{x}(t)$$. Second, if we could have access to the pointwise value of $$d(x(t))$$ at each time $$t$$, (14) will become implementable even when we do not know the exact functional representation of $$d(x)$$. Third, if we could estimate the pointwise value of $$d(x(t))$$ at each time $$t$$ with a bound to quantify the estimation error, then we could derive a robust condition for (14). Specifically, assume $$d(x(t))$$ is estimated as $$\hat{d}(t)$$ at each time $$t$$ with a uniform estimation error bound (EEB) $$\delta$$, i.e.,

$$\left\| \hat{d}(t) - d(x(t)) \right\| \leq \delta, \forall t \geq 0.$$  \hspace{1cm} (16)

Moreover, since $$M(x)$$ satisfies the CCM condition (8), $$u(t)$$ that satisfies (15) is guaranteed to exist for any $$t \geq 0$$, regardless of the size of $$\delta$$, if there are no control limits, i.e., $$\mathcal{U} = \mathbb{R}^m$$. We call condition (15) the robust Riemannian energy (RRE) condition.

4.3. Disturbance Estimation with Computable EEBs

We now introduce a disturbance estimation scheme to estimate the pointwise value of the uncertainty $$d(x)$$ with pre-computable EEBs, which can be systematically improved by tuning a parameter in the estimation law. The estimation scheme is based on the piecewise-constant estimation (PWCE) law in (Zhao et al., 2020), which was originally from (Cao and Hovakimyan, 2008). The PWCE law consists of two elements, namely a state predictor and a piecewise-constant update law. The state predictor is defined as:

$$\hat{x}(t) = f(x(t)) + B(x(t))u(t) + \hat{\sigma}(t) - a \tilde{x}(t), \ \hat{x}(0) = x_0,$$  \hspace{1cm} (17)

where $$\tilde{x}(t) \triangleq \hat{x}(t) - x(t)$$ is the prediction error, $$a$$ is a positive constant. A discussion about the role of $$a$$ is available in (Zhao et al., 2021b). The estimation, $$\hat{\sigma}(t)$$, is updated in a piecewise-constant way:

$$\left\{ \begin{array}{ll}
\hat{\sigma}(t) = \hat{\sigma}(iT), & t \in [iT, (i+1)T), \\
\hat{\sigma}(iT) = -\frac{a}{e^{aT} - 1} \tilde{x}(iT), & 
\end{array} \right.$$  \hspace{1cm} (18)

where $$T$$ is the estimation sampling time, and $$i = 0, 1, 2, \cdots$$. Finally, the pointwise value of $$d(x)$$ at time $$t$$ is estimated as

$$\hat{d}(t) = B^T(x(t))\hat{\sigma}(t),$$  \hspace{1cm} (19)
where $B^\dagger(x(t))$ is the pseudoinverse of $B(x(t))$. The following lemma establishes the EEBs associated with the estimation scheme in (17) and (18). The proof is inspired by (Wang et al., 2017), and similar to that in (Zhao et al., 2020). For completeness, it is given in Appendix A.

**Lemma 9** Given the dynamics (1) subject to Assumption 1, and the estimation law in (17) and (18), if $x \in \mathcal{X}$ and $u \in \mathcal{U}$ for any $t \geq 0$, the estimation error can be bounded as

$$
\|\tilde{d}(t) - d(x(t))\| \leq \delta(t, T) \triangleq \begin{cases} 
\beta_d, & \forall 0 \leq t < T, \\
\alpha(T) \max_{x \in \mathcal{X}} B^\dagger(x), & \forall t \geq T,
\end{cases}
$$

where

$$
\alpha(T) \triangleq 2\sqrt{n} \phi T (L_d \max_{x \in \mathcal{X}} \|B(x)\| + L_B b_d) + (1 - e^{-aT}) \sqrt{n} b_d \max_{x \in \mathcal{X}} \|B(x)\| 
$$

$$
\phi \triangleq \max_{x \in \mathcal{X}, u \in \mathcal{U}} \|f(x) + B(x)u\| + b_d \max_{x \in \mathcal{X}} \|B(x)\|,
$$

with constants $L_B$, $L_d$ and $b_d$ from Assumption 1, and $\phi$ defined in (22). Moreover, $\lim_{T \to 0} \delta(t, T) = 0$, for any $t \geq T$.

**Proof.** See Appendix A.

**Remark 10** Lemma 9 implies that theoretically, for $t \geq T$, the disturbance estimation after a single sampling interval can be made arbitrarily accurate by reducing $T$, which further indicates that the conservatism with the RRE condition can be arbitrarily reduced after a sampling interval.

In practice, the value of $T$ is subject to the limitations related to computational hardware and sensor noise. Additionally, using a very small $T$ tends to introduce high frequency components in the control loop, potentially harming the robustness of the closed-loop system, e.g., against time delay. This is similar to the use of a high adaptation rate in model reference adaptive control schemes, as discussed in (Hovakimyan and Cao, 2010). Therefore, one should avoid the use of a very small $T$ for the sake of robustness, unless a low-pass filter is used to filter the estimated disturbance before fed into (15), as suggested by the $L_1$ adaptive control architecture (Hovakimyan and Cao, 2010).

**Remark 11** The estimation in $(0, T)$ cannot be arbitrarily accurate. This is because the estimation in $[0, T)$ depends on $\tilde{x}(0)$ according to (18). Considering that $\tilde{x}(0)$ is purely determined by the initial state of the system, $x_0$, and the initial state of the predictor, $\tilde{x}_0$, it does not contain any information of the uncertainty. Since $T$ is usually very small in practice, lack of a tight estimation error bound for the interval $[0, T)$ will not cause an issue from a practical point of view. Additionally, the estimation of $\phi$ defined in (22) could be quite conservative. Further considering the frequent use of Lipschitz continuity and inequalities related to matrix/vector norms in deriving the constant $\alpha(T)$, $\alpha(T)$ can be overly conservative. Therefore, for practical implementation, one should leverage some empirical study, e.g., doing simulations under a few user-selected functions for $d(x)$ and determining a bound for $\delta(t, T)$. In our experiments, we found the theoretical bound $\delta(t, T)$ computed according to (20) was usually at least 10 and could be $10^4$ times more conservative.

**Remark 12** From Sections 4.2 and 4.3, we see that the uncertainty estimated by the estimation law (17) and (18) and incorporated in the RRE condition (15) is the discrepancy between the true
dynamics and the nominal dynamics without the learned model (i.e., \(d(x)\)), instead of the learned dynamics (5) (i.e., \(\tilde{d}(x)\)). Alternatively, we can easily adapt the estimation law (17) and (18) to estimate \(\tilde{d}(x)\), and adjust the RRE condition accordingly. However, as characterized in (20), the EEB depends on the local Lipschitz bound of the uncertainty to be estimated, which indicates that a Lipschitz bound for \(\tilde{d}(x)\) is needed to establish the EEB for it. However, except for a few model learning tools such as GPR (Lederer et al., 2019; Gahlawat et al., 2021), it is not easy, if not impossible, to establish the Lipschitz bound of the model error. When using SN-DNNs, we can ensure that the learned model \(\tilde{d}(x)\) has the same Lipschitz bound of \(L_d\) as \(d(x)\), which yields a Lipschitz bound of \(2L_d\) for \(\tilde{d}(x)\). In such a case, by inspecting (20), we can easily see that the EEB for \(\tilde{d}(x)\) is larger than that for \(d(x)\), which indicates that the RRE condition would be more conservative if we chose to use the estimated uncertainty of \(\tilde{d}(x)\) in this condition. Based on these observations, we choose to estimate the original uncertainty \(d(x)\) and use the estimated value in the RRE condition.

4.4. Guaranteed Contraction Control in the Presence of Imperfectly Learned Dynamics

Based on the review of contraction control in Section 3.2 and the discussions in Sections 4.2 and 4.3, the control law can be obtained by solving the following QP problem at each time \(t\):

\[
u(t) = \underset{k \in \mathbb{R}^m}{\text{argmin}} \|k - u^*(t)\|^2
\]

subject to

\[
\gamma^T_\delta(1, t)M(x) \dot{x} - \gamma^T_\delta(0, t)M(x^*) \dot{x}^* + \|\gamma^T_\delta(1, t)M(x)B(x)\| \delta(t, T) \leq -\lambda E(x^*, x),
\]

where \(\dot{x}(t) = f(x) + B(x)(k + \tilde{d}(t))\), according to, (16) depends on \(\tilde{d}(t)\), which is from the disturbance estimation law defined by (17)–(19), \(\delta(t, T)\) is defined in (20), and \(\dot{x}^* = F_l(x^*, u^*)\) with \(F_l(\cdot, \cdot)\) defined in (4). Similar to (12), the problem (23) is a pointwise min-norm control problem and has an analytic solution (Freeman and Kokotovic, 2008). Specifically, denoting \(\phi_0(t, x^*, x) \triangleq \gamma^T_\delta(1, t)M(x)(f(x) + B(x)(u^*(t) + \tilde{d}(t))) + \|\gamma^T_\delta(1, t)M(x)B(x)\| \delta(t, T) - \gamma^T_\delta(0, t)M(x^*) \dot{x}^* + \lambda E(x^*, x)\) and \(\phi_1(x^*, x) \triangleq \gamma^T_\delta(1, t)M(x)B(x)\), (24) can be written as \(\phi_0(t, x^*, x) + \phi_1(x^*, x)(k - u^*(t)) \leq 0\), and the solution for (23) is given by

\[
u(t) = k^* = \begin{cases}
u^*(t) & \text{if } \phi_0(t, x^*, x) \leq 0, \\
u^*(t) - \frac{\phi_0(t, x^*, x)\phi_1(x^*, x)}{\|\phi_1(x^*, x)\|^2} & \text{if } \phi_0(t, x^*, x) > 0.
\end{cases}
\]

We now make another assumption related to the control input space.

**Remark 13** To move forward with analysis, we need to verify that when \(x(t), x^*(t) \in \mathcal{X}\), the control signal \(u(t)\) from solving the QP problem (23) satisfies \(u(t) \in \mathcal{U}\). Deriving verifiable conditions to ensure this set bound is outside the scope of this paper and will be addressed as future work.

We are now ready to state the main result of the paper.

**Theorem 14** Given an uncertain system with learned dynamics represented by (3) satisfying Assumption 1, assume that there exists a metric \(W(x)\) that satisfies (9) for all \(x \in \mathcal{X}\). Furthermore,
suppose that a trajectory \((x^*(t), u^*(t))\) planned using the learned dynamics (5) and the initial actual states \(x(0)\) satisfy (13) and

\[
\Omega(t) \triangleq \left\{ y \in \mathbb{R}^n : y \leq \|x^*(t)\| + \sqrt{\frac{\alpha_2}{\alpha_1}} \|x(0) - x^*(0)\| e^{-\lambda t} \right\} \subset \mathcal{X},
\]

for any \(t \geq 0\), where \(\alpha_1 I \leq M(x) = W^{-1}(x) \leq \alpha_2 I\). Then, if \(u(t)\) from solving (23) satisfies \(u(t) \in \mathcal{U}\) for any \(t \geq 0\), the control law constructed by solving (23) ensures \(x(t) \in \mathcal{X}\) for any \(t \geq 0\), and furthermore, universally exponentially stabilizes the uncertain system (3) in the sense of Definition 3.

**Proof.** We use contradiction to show \(x(t) \in \mathcal{X}\) for all \(t \geq 0\). Assume this is not true. According to (25), \(x(0) \in \mathcal{X}\). Since \(x(t)\) is continuous, there must exist a time \(\tau\) such that

\[
x(t) \in \mathcal{X}, \ \forall t \in [0, \tau^-] \text{ and } x(\tau) \notin \mathcal{X}.
\]

Now let us consider the system evolution in \([0, \tau^-]\). Since \(u(t) \in \mathcal{U}\) by assumption and \(x(t) \in \mathcal{X}\) for any \(t \in [0, \tau^-]\), the EEB in (20) holds in \([0, \tau^-]\). As a result, the control law obtained from solving (23) ensures satisfaction of the RRE condition (15), and thus universally exponentially stabilizes the uncertain system (3) in \([0, \tau^-]\), in the sense of Definition 3. On the other hand, satisfaction of (13) implies that \(x^*(t)\) is a feasible state trajectory for the uncertain system (1) according to Theorem 8. Further considering Theorem 5, we have \(\|x(t)\| \leq \|x^*(t)\| + \sqrt{\frac{\alpha_2}{\alpha_1}} \|x(0) - x^*(0)\| e^{-\lambda t}\) for any \(t \in [0, \tau^-]\). Due to (25), the preceding inequality indicates that \(x(t)\) remains in the interior of \(\mathcal{X}\) for \(t \in [0, \tau^-]\). This, together with the continuity of \(x(t)\), immediately implies \(x(\tau) \in \mathcal{X}\), which contradicts (26). Therefore, we conclude that \(x(t) \in \mathcal{X}\) for all \(t \geq 0\). From the development of the proof, it is clear that the the UES of the closed-loop system in \(\mathcal{X}\) with the control law given by the solution of (23) is achieved. The proof is complete. 

A few remarks are in order.

**Remark 15** Theorem 14 essentially states that under certain assumptions, the proposed disturbance estimation based robust contraction controller guarantees exponential convergence of the actual state trajectory \(x(t)\) to a desired one \(x^*(t)\), planned using the learned dynamic model, regardless of the quality of the learned model. With the exponential guarantee, if the actual trajectory meets the desired trajectory at certain time \(\tau\), then these two trajectories will stay together afterward. The exponential convergence guarantee, is stronger than the performance guarantees provided by existing adaptive CCM-based approaches (Lopez and Slotine, 2020; Lakshmanan et al., 2020) that deal with similar settings (i.e., matched uncertainties). Specifically, in (Lopez and Slotine, 2020) asymptotic stability and bounded tracking error are guaranteed, while the approach in (Lakshmanan et al., 2020) ensures the actual trajectory always stays in a tube around the desired trajectory. Moreover, unlike sliding mode control schemes (Shiessel et al., 2014) that can potentially drive tracking error to zero in the presence of bounded uncertainties, the proposed scheme does not rely on discontinuous control inputs.

Our approach is related to the robust control Lyapunov based approaches (Freeman and Kokotovic, 2008), which provide robust stabilization around an equilibrium point (as opposed to a trajectory) in the presence of uncertainties.
Remark 16 The proposed control approach is inspired by the $L_1$ adaptive control theory (Hovakimyan and Cao, 2010). In fact, we adopt the estimation mechanism (the PWCE law in (17) and (18)) used within an $L_1$ controller. However, instead of directly cancelling the estimated disturbance as one would do with an $L_1$ controller, the proposed approach incorporates the estimated uncertainty and the EEBs into the robust Riemannian energy condition (24) to compute the control signal, which ensures exponential convergence of actual trajectories to desired ones. On the other hand, an $L_1$ adaptive controller can be used to augment a broad class of baseline controllers such as PID, linear quadratic regulator (LQR), model predictive controller (Pravitra et al., 2020) and an RL policy (Cheng et al., 2021), while the proposed approach is limited to the settings where the control signal can be derived by solving some inequality conditions, e.g., the RRE condition (15).

Remark 17 The exponential convergence guarantee stated in Theorem 14 is based on a continuous-time implementation of the controller. In practice, a controller is normally implemented on a digital processor or controller with a fixed sampling time. As a result, the property of exponential convergence may be slightly violated, as observed in Section 5.

5. Simulation Results

We validate the proposed learning control framework on a planar quadrotor system borrowed from (Singh et al., 2019). The state vector is defined as $x = [p_x, p_z, \phi, v_x, v_z, \phi]^{\top}$, where $p_x$ and $p_z$ are the position in $x$ and $z$ directions, respectively, $v_x$ and $v_z$ are the slip velocity (lateral) and the velocity along the thrust axis in the body frame of the vehicle, $\phi$ is the angle between the $x$ direction of the body frame and the $x$ direction of the inertia frame. The input vector $u = [u_1, u_2]$ contains the thrust force produced by each of the two propellers. The dynamics of the vehicle are given by

$$
\dot{x} = \begin{bmatrix}
p_x \\
p_z \\
\phi \\
v_x \\
v_z \\
\phi
\end{bmatrix} = \begin{bmatrix}
v_x \cos(\phi) - v_z \sin(\phi) \\
v_x \sin(\phi) + v_z \cos(\phi) \\
\phi \\
v_z \phi - g \sin(\phi) \\
v_x \phi - g \cos(\phi) \\
0
\end{bmatrix} + \begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
\frac{1}{m} & \frac{1}{m} \\
\frac{1}{J} & \frac{1}{J}
\end{bmatrix} (u + d(x)),
$$

where $m$ and $J$ denote the mass and moment of inertia about the out-of-plane axis and $l$ is the distance between each of the propellers and the vehicle center, and $d(x)$ denotes the known disturbances exerted on the propellers. The parameters were set as $m = 0.486$ kg, $J = 0.00383$ Kg m$^2$, and $l = 0.25$ m. The uncertainty $d(x)$ is chosen to be $d(x) = \rho(x, z) \cdot 0.5(v_x^2 + v_z^2)[1, 1]^{\top}$, where $\rho(x, z) = 1/(x^2 + y^2 + 1)$ represents the disturbance intensity whose values in a specific location are denoted by the color at this location in Fig. 1. We imposed the following constraints:

\begin{align*}
x \in \mathcal{X} & \triangleq [0, 15] \times [0, 15] \times [-\frac{\pi}{4}, \frac{\pi}{4}] \times [-2, 2] \times [-1, -1] \times [-\frac{\pi}{4}, \frac{\pi}{4}], \\
u \in \mathcal{U} & \triangleq [0, \frac{2}{3}mg] \times [0, \frac{2}{3}mg].
\end{align*}

When searching for CCM, we parameterized the CCM $W$ by $\phi$ and $v_x$, and imposed the constraint $W \geq 0.01I$. The convergence rate $\lambda$ was chosen to be 0.8. More details about synthesizing the CCM can be found in (Zhao et al., 2021a). All the subsequent computations and simulations except DNN training (which was done in Python using PyTorch) were done in Matlab R2021b. For estimating the disturbance using (17)–(19), we set $a = 10$. It is easy to verify that $L_d = 4$, $b_d = 3.54$, and $L_B = 0$ (due to the fact that $B$ is constant) satisfy (2). By gridding the space $\mathcal{X}$, the constant $\phi$ in (22) can be determined as $\phi = 783.96$. According to (20), if we want to achieve an
EEB $\delta(t, T) = 0.1$, then the estimation sampling time needs to satisfy $T_s \leq 2.04 \times 10^{-7}$ s. However, as noted in Remark 11, the EEB computed according to (20) could be overly conservative. By simulations, we found that the estimation sampling time of 0.002 s was more than enough to ensure the desired EEB and therefore simply set $T_s = 0.002$ s.

We consider three navigation tasks: Task 1: flying from point $(2, 0)$ to $(8, 10)$, Task 2: flying from point $(8, 0)$ to $(2, 10)$, and Task 3: flying from point $(0, 6)$ to $(10, 6)$, while avoiding the three circular obstacles as illustrated in Fig. 4. The planned trajectories were generated using OptimTraj (Kelly, 2017), to minimize the cost $J = \int_{T_a}^T u(t)\|u(t)\|^2 \, dt + 5T_a$, where $T_a$ is the arrival time. OPTI (Currie and Wilson, 2012) and Matlab fmincon solvers were used to solve the geodesic optimization problem (see Section 3.2). The actual start points for Tasks 1~3 were $(0, 0)$, $(10, 0)$ and $(0, 4)$, respectively, which were intentionally set to be different from the desired ones used for trajectory planning. Figure 1 shows the planned and actual trajectories under the CCM controller using only the learned dynamics and the proposed CCM controller based on the RRE condition and distur-
Guaranteed Control with Learned Dynamics

In the presence of no, poor, moderate and good learned model for the uncertain dynamics, SN-DNNs (see Section 3.1) with four inputs, two outputs and four hidden layers were used for all the learned models. The data used for training these models were generated as follows. First, without learning, when the RD-CCM controller was used to conduct the three tasks, data associated with the three actual trajectories were collected and used for training the DNN model, which gave the poor model due to the limited amount of data used to train the DNN. Under the same configuration, three different trajectories with lower limits on $v_2$ were generated and tracked by the RD-CCM controller. The data from the actual three trajectories were added to previously connected data for training, which gave the moderate model. In practice, the quality of the learned model directly depends on the amount of information-rich data. Therefore, sufficient exploration of the state space is necessary to learn a good model for the uncertainty $d(x)$. Thanks to the performance guarantee (in terms of exponential trajectory convergence), the proposed RD-CCM controller can be used to control the system to safely explore the state space. For illustration purpose, we directly used true uncertainty model to generate the data and used the generated data for training, which yield the good model.

![Graphs](image)

(a) Riemannian energy

(b) Control input

Figure 2: Trajectories of Riemannian energy $E(x^*, x)$ (left) and the first element of control input (right), i.e., $u_1$, in the presence of no, poor and good learning for Task 1

One can see that the actual trajectories yielded by the CCM controller deviated quite a lot from the planned ones and collided with obstacle sometimes, except in the case of good learning. On the other hand, the actual trajectories yielded by the RD-CCM controller converged to the desired trajectory as expected and almost overlapped with it afterward, throughout the learning phase, even for some weird trajectories, e.g., the one for Task 3 planned using the poor model. In fact, the slight deviations of actual trajectories from the desired ones under the RD-CCM controller were due to the finite step size associated with the ODE solver used for the simulations (see Remark 17). Figure 2 shows trajectories of Riemannian energy $E(x^*, x)$ in the presence of no, poor and good learning for Task 1, while the trajectories for Tasks 2 and 3 are similar and thus omitted. One can see that the value of $E(x^*, x)$ under the RD-CCM controller decreased exponentially in all cases, regardless of the quality of the learned model. Figure 3 depicts the trajectories of true, learned and estimated...
Figure 3: Trajectories of true, learned and estimated disturbances in the presence of no and good learning for Task 1. The notations $d_1$, $\hat{d}_1$ and $\tilde{d}_1$ denote the first element of $d$, $\hat{d}$ and $\tilde{d}$, respectively.

Figure 4: Costs of the actual trajectories yielded by the RD-CCM controller throughout the learning phase

disturbances in the presence of no and good learning for Task 1, while the trajectories for Tasks 2 and 3 are similar and thus omitted. One can see that the estimated disturbances were always fairly close to the true disturbances. Also, the area with high disturbance intensity was avoided during path planning with good learning, which explains the smaller disturbance encountered.

We further evaluated the costs $J$ associated with the actual trajectories given by the RD-CCM controller under different learning qualities. The results are shown in Fig. 4. As expected, the good model helped plan better trajectories, which led to reduced costs for all the three tasks. It is not a surprise that the poor and moderate models led to temporal increase of the costs for some tasks. In practice, we may never use the poorly learned model, due to lack of a sufficient exploration of the state space, directly for planning trajectories. The proposed control approach ensures that in case one really does so, the planned trajectories can still be well tracked.
6. Conclusions

This paper presents a trajectory-centric learning control framework based on contraction metrics and disturbance estimation for uncertain nonlinear systems. The framework allows for the use of a broad class of model learning tools, such as deep neural networks (DNNs), to learn uncertain dynamics, while still providing guarantees of transient tracking performance in the form of exponential convergence of actual trajectories to desired ones throughout the learning phase, including the special case of no learning. On the other hand, with improved accuracy, the learned model can help plan better trajectories with improved performance beyond tracking. The proposed framework is demonstrated on a planar quadrotor example.

There are a few directions we would like to explore in the future. First, we would like to consider more general uncertainties, in particular, unmatched uncertainties that widely exist in practical systems. Second, our performance guarantees build upon the assumption that the computed control inputs are within the actuator limits of the controlled system (Remark 13). In practice, a poorly learned model, if naively incorporated in a trajectory planner, may produce a trajectory that deviates much from the trajectories with which the collected data is associated, and necessitates large control inputs exceeding the actuator limits to track. To mitigate this issue, we will incorporate mechanisms to ensure that newly planned trajectories are close to data-collection trajectories. Finally, we would like to experimentally validate the proposed framework on real hardware.

Appendix A. Proof of Lemma 9

Hereafter, we use the notations $\mathbb{Z}_i$ and $\mathbb{Z}_i^n$ to denote the integer sets $\{i, i + 1, i + 2, \cdots\}$ and $\{1, 2, \cdots, n\}$, respectively. Additionally, for notation brevity, we define $\sigma(t) \triangleq B(x(t))d(x(t))$. From (1) and (17), the prediction error dynamics are obtained as

$$\dot{x}(t) = -a \ddot{x}(t) + \dot{\sigma}(t) - \sigma(t), \quad \ddot{x}(0) = 0. \quad (27)$$

Note that $\dot{\sigma}(t) = 0$ (and thus $\ddot{d}(t) = 0$ due to (19)) for any $t \in [0, T)$ according to (18). Further considering the bound on $d(x)$ in (2), we have

$$\| \ddot{d}(t) - \ddot{d}(x(t)) \| \leq b_d, \quad \forall t \in [0, T). \quad (28)$$

We next derive the bound on $\| \dot{\sigma}(t) - \sigma(t) \|$ for $t \geq T$. For any $t \in [iT, (i+1)T)$ ($i \in \mathbb{Z}_0$), we have

$$\ddot{x}(t) = e^{-a(t-iT)} \ddot{x}(iT) + \int_{iT}^{t} e^{-a(t-\tau)} (\ddot{\sigma}(\tau) - \sigma(\tau))d\tau. \quad (29)$$

Since $\ddot{x}(t)$ is continuous, the preceding equation implies

$$\ddot{x}((i+1)T) = e^{-aT} \ddot{x}(iT) + \int_{iT}^{(i+1)T} e^{-a((i+1)T-\tau)} d\tau \ddot{\sigma}(iT) - \int_{iT}^{(i+1)T} e^{-a((i+1)T-\tau)} \sigma(\tau)d\tau$$

$$= e^{-aT} \ddot{x}(iT) + \frac{1 - e^{-aT}}{a} \ddot{\sigma}(iT) - \int_{iT}^{(i+1)T} e^{-a((i+1)T-\tau)} \sigma(\tau)d\tau$$

$$= - \int_{iT}^{(i+1)T} e^{-a((i+1)T-\tau)} \sigma(\tau)d\tau, \quad (29)$$

17
where the first and last equalities are due to the estimation law (18).

Since \(x(t)\) is continuous, \(\sigma(t) = B(x(t))\) is also continuous given Assumption 1. Furthermore, considering that \(e^{-a((i+1)T - \tau)}\) is always positive, we can apply the first mean value theorem in an element-wise manner\(^1\) to (29), which leads to

\[
\hat{x}((i+1)T) = -\int_{iT}^{(i+1)T} e^{-a((i+1)T - \tau)} d\tau [\sigma_j(\tau_j^*)] = -\frac{1}{a} (1 - e^{-aT}) [\sigma_j(\tau_j^*)],
\]

(30)

for some \(\tau_j^* \in (iT, (i+1)T)\) with \(j \in \mathbb{Z}_1^n\) and \(i \in \mathbb{Z}_0\), where \(\sigma_j(t)\) is the \(j\)-th element of \(\sigma(t)\), and

\[
[\sigma_j(\tau_j^*), \cdots, \sigma_n(\tau_n^*)]^T.
\]

The estimation law (18) indicates that for any \(t \in [(i+1)T, (i+2)T)\), we have \(\hat{\sigma}(t) = -\frac{a}{e^{aT} - 1} \hat{x}((i+1)T)\). The preceding equality and (30) imply that for any \(t \in [(i+1)T, (i+2)T)\) with \(i \in \mathbb{Z}_0\), there exist \(\tau_j^* \in (iT, (i+1)T)\) \((j \in \mathbb{Z}_1^n)\) such that

\[
\hat{\sigma}(t) = e^{-aT} [\sigma_j(\tau_j^*)].
\]

(31)

Note that

\[
\|\sigma(t) - [\sigma_j(\tau_j^*)]\| \leq \sqrt{n} \|\sigma(t) - [\sigma_j(\tau_j^*)]\|_\infty = \sqrt{n} \|\sigma_j(t) - \sigma_j(\tau_j^*)\| \leq \sqrt{n} \|\sigma(t) - \sigma(\tau_j^*)\|,
\]

(32)

where \(j_t = \arg \max_{j \in \mathbb{Z}_1^n} |\sigma_j(t) - \sigma_j(\tau_j^*)|\). Similarly,

\[
\|\sigma(t)\|_\infty \leq \sqrt{n} \|\sigma_j(\tau_j^*)\| \leq \sqrt{n} \|\sigma(\tau_j^*)\| \leq \sqrt{nb_d} \max_{x \in \mathcal{X}} \|B(x)\|,
\]

(33)

where \(\hat{j} = \arg \max_{j \in \mathbb{Z}_1^n} |\sigma_j(\tau_j^*)|\), and the last inequality is due to the fact \(\|B(x)d(x)\| \leq \|B(x)\|\|d(x)\|\) and Assumption 1. Therefore, for any \(t \in [(i+1)T, (i+2)T)\) \((i \in \mathbb{Z}_0)\), we have

\[
\|\sigma(t) - \hat{\sigma}(t)\| = \|\sigma(t) - e^{-aT} [\sigma_j(\tau_j^*)]\| \leq \|\sigma(t) - [\sigma_j(\tau_j^*)]\| + (1 - e^{-aT}) \|\sigma_j(\tau_j^*)\| \\
\leq \sqrt{n} \|\sigma(t) - \sigma(\tau_j^*)\| + (1 - e^{-aT}) \sqrt{nb_d} \max_{x \in \mathcal{X}} \|B(x)\|,
\]

(34)

for some \(\tau_j^* \in (iT, (i+1)T)\), where the equality is due to (31), and the last inequality is due to (32) and (33). The dynamics (1) indicates that

\[
\|\dot{x}\| \leq \|f(x) + B(x)u\| + \|B(x)\|\|d(x)\| \leq \phi,
\]

(35)

where \(\phi\) is defined in (22). As a result, the inequality (35) implies that

\[
\|x(t) - x(\tau_{j_t}^*)\| \leq \int_{\tau_{j_t}^*}^t \|\dot{x}(\tau)\| d\tau \leq \int_{\tau_{j_t}^*}^t \phi d\tau = \phi (t - \tau_{j_t}^*) \leq 2\phi T,
\]

\(1\). Note that the mean value theorem for definite integrals only holds for scalar valued functions.
where the last inequality is due to the fact that $t \in [(i + 1)T, (i + 2)T)$ and $\tau_{ji}^* \in (iT, (i + 1)T)$. The preceding inequality and (2) indicate that

\[
\left\| \sigma(x(t)) - \sigma(x(\tau_{ji}^*)) \right\| = \left\| B(x(t)) \left( d(x(t)) - d(x(\tau_{ji}^*)) \right) + \left( B(x(t)) - B(x(\tau_{ji}^*)) \right) d(x(\tau_{ji}^*)) \right\|
\leq \left\| B(x(t)) \right\| \left\| d(x(t)) - d(x(\tau_{ji}^*)) \right\| + \left\| B(x(t)) - B(x(\tau_{ji}^*)) \right\| \left\| d(x(\tau_{ji}^*)) \right\|
\leq L_d \max_{x \in \mathcal{X}} \left\| B(x) \right\| + L_B \left\| x(t) - x(\tau_{ji}^*) \right\| b_d
\leq 2\phi T(L_d \max_{x \in \mathcal{X}} \left\| B(x) \right\| + L_B b_d).
\] (36)

Finally, plugging (36) into (34) leads to

\[
\left\| \sigma(t) - \hat{\sigma}(t) \right\| \leq 2\sqrt{n}\phi T(L_d \max_{x \in \mathcal{X}} \left\| B(x) \right\| + L_B b_d) + (1 - e^{-\alpha T}) \sqrt{n} b_d \max_{x \in \mathcal{X}} \left\| B(x) \right\| = \alpha(T),
\] (37)

for any $t \geq T$. From (28), (37) and the relation between $\hat{\sigma}(t)$ and $\tilde{d}(t)$ in (19), we arrive at (20). Considering Assumption 1 and the assumption that $\mathcal{X}$ and $\mathcal{U}$ are compact, the constants involved in the definition of $\alpha(T)$ in (21) are all finite. As a result, we have $\lim_{T \to 0} \alpha(T) = 0$, which further indicates that $\lim_{T \to 0} \delta(t, T) = 0$, for any $t \geq T$. The proof is complete. \hfill \blacksquare

Acknowledgments

This work is supported by AFOSR, NASA and NSF under the RI grant #2133656, NRI grant #1830639, CPS grant #1932529 and AI Institute Planning grant #2020289. The authors would like to thank Jing Wu for helpful discussions.

References

Felix Berkenkamp and Angela P. Schoellig. Safe and robust learning control with Gaussian processes. In Proceedings of 2015 European Control Conference (ECC), pages 2496–2501, 2015.

Felix Berkenkamp, Riccardo Moriconi, Angela P Schoellig, and Andreas Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes. In IEEE Conference on Decision and Control (CDC), pages 4661–4666, December 2016.

Chengyu Cao and Naira Hovakimyan. $L_1$ adaptive output feedback controller for non strictly positive real reference systems with applications to aerospace examples. In AIAA Guidance, Navigation and Control Conference and Exhibit, page 7288, 2008.

Wen-Hua Chen, Jun Yang, Lei Guo, and Shihua Li. Disturbance-observer-based control and related methods—An overview. IEEE Transactions on Industrial Electronics, 63(2):1083–1095, 2015.

Yikun Cheng, Pan Zhao, Fanxin Wang, Jerome Daniel Block, and Naira Hovakimyan. Improving the robustness of reinforcement learning policies with $L_1$ adaptive control. arXiv preprint arXiv:2106.02249, 2021. URL https://arxiv.org/abs/2112.01953.

Girish Chowdhary, Hassan A. Kingravi, Jonathan P. How, and Patricio A. Vela. Bayesian nonparametric adaptive control using Gaussian processes. IEEE Transactions on Neural Networks and Learning Systems, 26(3):537–550, 2014.
Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In Advances in Neural Information Processing Systems, pages 4754–4765, 2018.

Jonathan Currie and David I. Wilson. OPTI: Lowering the barrier between open source optimizers and the industrial MATLAB user. In Nick Sahinidis and Jose Pinto, editors, Foundations of Computer-Aided Process Operations, Savannah, Georgia, USA, 2012.

Manfredo Perdigao Do Carmo and J Flaherty Francis. Riemannian Geometry. Springer, Boston, MA, USA, 1992.

Christopher Edwards and Sarah Spurgeon. Sliding mode control: theory and applications. CRC Press, 1998.

Randy Freeman and Petar V. Kokotovic. Robust Nonlinear Control Design: State-Space and Lyapunov Techniques. Springer Science & Business Media, 2008.

Aditya Gahlawat, Pan Zhao, Andrew Patterson, Naira Hovakimyan, and Evangelos A Theodorou. $\mathcal{L}_1$-GP: $\mathcal{L}_1$ adaptive control with Bayesian learning. In The 2nd Annual Conference on Learning for Dynamics and Control, Proceedings of Machine Learning Research, volume 120, pages 1–12, 2020.

Aditya Gahlawat, Arun Lakshmanan, Lin Song, Andrew Patterson, Zhuohuan Wu, Naira Hovakimyan, and Evangelos Theodorou. Contraction $\mathcal{L}_1$ adaptive control using Gaussian processes. In Proceedings of the 3rd Conference on Learning for Dynamics and Control, Proceedings of Machine Learning Research, volume 144, pages 1027–1040, 2021.

J. Han. From PID to active disturbance rejection control. IEEE Transactions on Industrial Electronics, 56(3):900–906, 2009.

Lukas Hewing, Juraj Kabzan, and Melanie N. Zeilinger. Cautious model predictive control using Gaussian process regression. IEEE Transactions on Control Systems Technology, 2019.

Naira Hovakimyan and Chengyu Cao. $\mathcal{L}_1$ Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2010.

Petros A. Ioannou and Jing Sun. Robust Adaptive Control. Dover Publications, Inc., Mineola, NY, 2012.

Girish Joshi and Girish Chowdhary. Deep model reference adaptive control. In IEEE 58th Conference on Decision and Control (CDC), pages 4601–4608. IEEE, 2019.

Girish Joshi, Jasvir Virdi, and Girish Chowdhary. Asynchronous deep model reference adaptive control. arXiv preprint arXiv:2011.02920, 2020.

Matthew Kelly. An introduction to trajectory optimization: How to do your own direct collocation. SIAM Review, 59(4):849–904, 2017.

Mohammad Javad Khojasteh, Vikas Dhiman, Massimo Franceschetti, and Nikolay Atanasov. Probabilistic safety constraints for learned high relative degree system dynamics. In Annual Conference on Learning for Dynamics and Control, pages 781–792, 2020.
Arun Lakshmanan, Aditya Gahlawat, and Naira Hovakimyan. Safe feedback motion planning: A contraction theory and $L_1$-adaptive control based approach. In Proceedings of 59th IEEE Conference on Decision and Control (CDC), pages 1578–1583, 2020.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, 2015.

Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform error bounds for Gaussian process regression with application to safe control. arXiv preprint arXiv:1906.01376, 2019.

Karen Leung and Ian R. Manchester. Nonlinear stabilization via control contraction metrics: A pseudospectral approach for computing geodesics. In American Control Conference, pages 1284–1289, 2017.

Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for non-linear systems. Automatica, 34(6):683–696, 1998.

Brett T Lopez and Jean-Jacques E Slotine. Adaptive nonlinear control with contraction metrics. IEEE Control Systems Letters, 5(1):205–210, 2020.

Ian R Manchester and Jean-Jacques E Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. IEEE Transactions on Automatic Control, 62(6):3046–3053, 2017.

Ian R Manchester and Jean-Jacques E Slotine. Robust control contraction metrics: A convex approach to nonlinear state-feedback $H_{\infty}$ control. IEEE Control Systems Letters, 2(3):333–338, 2018.

David Q Mayne. Model predictive control: Recent developments and future promise. Automatica, 50(12):2967–2986, 2014.

David Q Mayne, María M Seron, and SV Raković. Robust model predictive control of constrained linear systems with bounded disturbances. Automatica, 41(2):219–224, 2005.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In International Conference on Learning Representations (ICLR), 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

Jintasit Pravitra, Kasey A. Ackerman, Chengyu Cao, Naira Hovakimyan, and Evangelos A. Theodorou. $L_1$-adaptive MPPI architecture for robust and agile control of multirotors. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 7661–7666, 2020. doi: 10.1109/IROS45743.2020.9341154.

Guanya Shi, Xichen Shi, Michael O’Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. In International Conference on Robotics and Automation (ICRA), pages 9784–9790, 2019.
Yuri Shtessel, Christopher Edwards, Leonid Fridman, Arie Levant, et al. *Sliding Mode Control and Observation*, volume 10. Springer, 2014.

Sumeet Singh, Benoit Landry, Anirudha Majumdar, Jean-Jacques Slotine, and Marco Pavone. Robust feedback motion planning via contraction theory. *The International Journal of Robotics Research*, 2019.

Dawei Sun, Susmit Jha, and Chuchu Fan. Learning certified control using contraction metric. In *Conference on Robot Learning (CoRL)*, 2020.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT press, Cambridge, MA, 2018.

Hiroyasu Tsukamoto and Soon-Jo Chung. Neural contraction metrics for robust estimation and control: A convex optimization approach. *IEEE Control Systems Letters*, 5(1):211–216, 2020a.

Hiroyasu Tsukamoto and Soon-Jo Chung. Robust controller design for stochastic nonlinear systems via convex optimization. *IEEE Transactions on Automatic Control*, 2020b.

Xiaofeng Wang, Lixing Yang, Yu Sun, and Kun Deng. Adaptive model predictive control of nonlinear systems with state-dependent uncertainties. *Int. J. Robust Nonlinear Control*, 27(17):4138–4153, 2017.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic MPC for model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721, 2017.

Pan Zhao, Yanbing Mao, Chuyuan Tao, Naira Hovakimyan, and Xiaofeng Wang. Adaptive robust quadratic programs using control Lyapunov and barrier functions. In *59th IEEE Conference on Decision and Control*, pages 3353–3358, 2020.

Pan Zhao, Arun Lakshmanan, Kasey Ackerman, Aditya Gahlawat, Marco Pavone, and Naira Hovakimyan. Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics. *arXiv preprint arXiv:2109.04453*, 2021a.

Pan Zhao, Steven Snyder, Naira Hovakimyan, and Chengyu Cao. Robust adaptive control of linear parameter-varying systems with unmatched uncertainties. *arXiv:2010.04600*, 2021b.

Kemin Zhou and John Comstock Doyle. *Essentials of Robust Control*. Prentice Hall, Upper Saddle River, NJ, 1998.