

Adaptive Control of SE(3) Hamiltonian Dynamics with Learned Disturbance Features

Thai Duong, *Student Member, IEEE*, and Nikolay Atanasov, *Member, IEEE*

Abstract—Adaptive control is a critical component of reliable robot autonomy in rapidly changing operational conditions. Adaptive control designs benefit from a disturbance model, which is often unavailable in practice. This motivates the use of machine learning techniques to learn disturbance features from training data offline, which can subsequently be employed to compensate the disturbances online. This paper develops geometric adaptive control with a learned disturbance model for rigid-body systems, such as ground, aerial, and underwater vehicles, that satisfy Hamilton’s equations of motion over the SE(3) manifold. Our design consists of an *offline disturbance model identification stage*, using a Hamiltonian-based neural ordinary differential equation (ODE) network trained from state-control trajectory data, and an *online adaptive control stage*, estimating and compensating the disturbances based on geometric tracking errors. We demonstrate our adaptive geometric controller in trajectory tracking simulations of fully-actuated pendulum and under-actuated quadrotor systems.

Index Terms—Adaptive control, Identification for control, Machine learning.

I. INTRODUCTION

AUTONOMOUS mobile robots assisting in transportation, search and rescue, and environmental monitoring applications face complex and dynamic operational conditions. Ensuring safe operation depends on the availability of accurate system dynamics models, which can be obtained using system identification [1] or machine learning techniques [2]–[5]. When disturbances and system changes during online operation bring about new out-of-distribution data, it is often too slow to re-train the nominal dynamics model to support real-time adaptation to environment changes. Instead, adaptive control [6], [7] offers efficient tools to estimate and compensate for disturbances and parameter variations online.

A key technical challenge in adaptive control is the design of an adaptation law that estimates the disturbance online [6]. The disturbance can be non-parametric [8]–[12] or parametric [13]–[16], e.g. a linear combinations of *known* nonlinear features, and is updated based on the state errors with stability obtained by sliding-mode theory [13]–[15], assuming zero-state detectability [15], [16] or \mathcal{L}_1 -adaptation [8]–[10]. If the system evolves on a manifold (e.g., when the state contains

Manuscript received March 21, 2022; accepted May 5, 2022. We gratefully acknowledge support from NSF RI IIS-2007141 and NSF CCF-2112665 (TILOS).

The authors are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA, e-mail: {tduong, natanasov}@ucsd.edu.

orientation), an adaptation law is designed based on geometric errors, derived from the manifold constraints [17], [18]. A disturbance observer [11], [12] use the state errors introduced by the disturbances to design an asymptotically stable observer system that estimates the disturbance online. A disturbance adaptation law is paired with a nominal controller, derived using Lagrangian dynamics with feedback linearization [13], [19], Hamiltonian dynamics with energy shaping [15], [20], or model predictive control [10], [21].

Recently, there has been growing interest in applying machine learning techniques to design adaptive controllers. As the nonlinear disturbance features are actually *unknown* in practice, they can be estimated using Gaussian processes [9], [22] or neural networks [23], [24]. The features can be learned online in the control loop [9], [23], which is potentially slow for real-time operation, or offline via meta-learning from past state-control trajectories [25] or system dynamics simulation [24]. Given the learned disturbance features, an adaptation law is designed to estimate the disturbances online, e.g. using \mathcal{L}_1 -adaptation [9] or by updating the last layer of the feature neural network [23], [24], [26].

This paper develops data-driven adaptive control for rigid-body systems, such as unmanned ground vehicles (UGVs), unmanned aerial vehicles (UAVs), or unmanned underwater vehicles (UUVs), that satisfy Hamilton’s equations of motion on position and orientation manifold $SE(3)$. While adaptation laws have been developed to work with non-parametric uncertainties in the related work, we consider linearly parameterized disturbances, i.e. linear combinations of *unknown* nonlinear features. While recent techniques for disturbance feature learning and data-driven adaptive control are restricted to systems whose states evolve in Euclidean space, a unique aspect of our adaptive control design is the consideration of geometric tracking errors on the $SE(3)$ manifold. Compared to existing $SE(3)$ geometric adaptive controllers specifically designed for quadrotors with a known disturbance model [17], [18], we develop a general adaptation law that can be used for any rigid-body robot, such as a UGV, UAV, or UUV, and learn disturbance features from trajectory data instead of assuming a known model. Specifically, given a dataset of state-control trajectories with different disturbance realizations, we learn nonlinear disturbance features using a Hamiltonian-based neural ODE network [5], where the disturbances are represented by neural networks, connected in an architecture that respects the Hamiltonian dynamics. We develop a geometric adaptation law to estimate the disturbances online and compensate them by a nonlinear energy-shaping tracking controller.

In summary, our **contribution** is a learning-based adaptive geometric control for $SE(3)$ Hamiltonian dynamics that

- learns disturbance features offline from state-control trajectories using an $SE(3)$ Hamiltonian-based neural ODE network, and
- employs energy-based tracking control with adaptive disturbance compensation online based on the learned disturbance model and the geometric tracking errors.

We verify our approach using simulated fully-actuated pendulum and under-actuated quadrotor systems, and compare with a disturbance observer method to highlight the benefit of learning disturbance features from data.

II. PROBLEM STATEMENT

Consider a system modeled as a single rigid body with position $\mathbf{p} \in \mathbb{R}^3$, orientation $\mathbf{R} \in SO(3)$, body-frame linear velocity $\mathbf{v} \in \mathbb{R}^3$, and body-frame angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$. Let $\mathbf{q} = [\mathbf{p}^\top \ \mathbf{r}_1^\top \ \mathbf{r}_2^\top \ \mathbf{r}_3^\top]^\top \in SE(3)$ be the generalized coordinates, where $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ are the rows of the rotation matrix \mathbf{R} . Let $\boldsymbol{\zeta} = [\mathbf{v}^\top \ \boldsymbol{\omega}^\top]^\top \in \mathbb{R}^6$ be the generalized velocity. The generalized momentum of the system is defined as $\mathbf{p} = \mathbf{M}(\mathbf{q})\boldsymbol{\zeta} \in \mathbb{R}^6$, where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{6 \times 6}$ is the generalized mass matrix. The state is defined as $\mathbf{x} = (\mathbf{q}, \mathbf{p})$ and its evolution is governed by the system dynamics:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}), \quad (1)$$

where \mathbf{u} is the control input and \mathbf{d} is a disturbance signal. The disturbance \mathbf{d} is modeled as a linear combination of nonlinear features $\mathbf{W}(\mathbf{x}) \in \mathbb{R}^{6 \times p}$:

$$\mathbf{d}(t) = \mathbf{W}(\mathbf{x}(t))\mathbf{a}^*, \quad (2)$$

where $\mathbf{a}^* \in \mathbb{R}^p$ are unknown feature weights.

A mechanical system obeys Hamilton's equations of motion [27, Chapter 7]. The Hamiltonian, $\mathcal{H}(\mathbf{q}, \mathbf{p}) = T(\mathbf{q}, \mathbf{p}) + V(\mathbf{q})$, captures the total energy of the system as the sum of the kinetic energy $T(\mathbf{q}, \mathbf{p}) = \frac{1}{2}\mathbf{p}^\top \mathbf{M}(\mathbf{q})^{-1} \mathbf{p}$ and the potential energy $V(\mathbf{q})$. The dynamics in (1) are determined by the Hamiltonian and re-formulated as a port-Hamiltonian system [5], [28]:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{q}^\times \\ -\mathbf{q}^{\times\top} & \mathbf{p}^\times \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{H}}{\partial \mathbf{q}} \\ \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}(\mathbf{q}) \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{0} \\ \mathbf{d} \end{bmatrix}, \quad (3)$$

where $\mathbf{g}(\mathbf{q})$ is the input gain and the disturbance \mathbf{d} appears as an external force applied to the system. The operators \mathbf{q}^\times and \mathbf{p}^\times are defined as:

$$\mathbf{q}^\times = \begin{bmatrix} \mathbf{R}^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^\top, \quad \mathbf{p}^\times = \begin{bmatrix} \mathbf{p}_\mathbf{v} \\ \hat{\mathbf{p}}_\mathbf{v} \end{bmatrix}^\times = \begin{bmatrix} \mathbf{0} & \hat{\mathbf{p}}_\mathbf{v} \\ \hat{\mathbf{p}}_\mathbf{v} & \hat{\mathbf{p}}_\omega \end{bmatrix},$$

where the hat map $\hat{(\cdot)} : \mathbb{R}^3 \mapsto \mathfrak{so}(3)$ constructs a skew-symmetric matrix from a 3D vector. Note that the equation $\dot{\mathbf{q}} = \mathbf{q}^\times \frac{\partial \mathcal{H}}{\partial \mathbf{q}}$ in (3) exactly specifies the $SE(3)$ kinematics, $\dot{\mathbf{p}} = \mathbf{R}\mathbf{v}$ and $\dot{\mathbf{R}} = \mathbf{R}\hat{\boldsymbol{\omega}}$, with the rotation part written row-by-row.

Consider a collection $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_M\}$ of system state transitions \mathcal{D}_j , each obtained under a different unknown disturbance realization \mathbf{a}_j^* , for $j = 1, \dots, M$. Each $\mathcal{D}_j = \{\mathbf{x}_0^{(ij)}, \mathbf{u}^{(ij)}, \mathbf{x}_f^{(ij)}, \tau^{(ij)}\}_{i=1}^{D_j}$ consists of D_j state transitions,

each obtained by applying a constant control input $\mathbf{u}^{(ij)}$ to the system with initial condition $\mathbf{x}_0^{(ij)}$ and sampling the state $\mathbf{x}_f^{(ij)} := \mathbf{x}^{(ij)}(\tau^{(ij)})$ at time $\tau^{(ij)}$. Our objective is to approximate the disturbance model in (2) by $\bar{\mathbf{d}}_\theta(t) = \mathbf{W}_\theta(\mathbf{x}(t))\mathbf{a}_j$, where θ parameterizes the shared disturbance features and the parameters $\{\mathbf{a}_j\}_{j=1}^M$ model each disturbance realization. To optimize θ , $\{\mathbf{a}_j\}$, we predict the dynamics evolution starting from state $\mathbf{x}_0^{(ij)}$ with control $\mathbf{u}^{(ij)}$ and minimize the distance between the predicted state $\bar{\mathbf{x}}_f^{(ij)}$ and the true state $\mathbf{x}_f^{(ij)}$ from \mathcal{D}_j , for $j = 1, \dots, M$. Since the approximated disturbance $\bar{\mathbf{d}}_\theta$ does not change if the features \mathbf{W}_θ and the coefficients \mathbf{a}_j are scaled by constants γ and $1/\gamma$, respectively, we add the norms of $\mathbf{W}_\theta(\mathbf{x}_0^{(ij)})$ and $\{\mathbf{a}_j\}_{j=1}^M$ to the objective function as regularization terms.

Problem 1. Given $\mathcal{D} = \{\{\mathbf{x}_0^{(ij)}, \mathbf{u}^{(ij)}, \mathbf{x}_f^{(ij)}, \tau^{(ij)}\}_{i=1}^{D_j}\}_{j=1}^M$, find disturbance parameters θ , $\{\mathbf{a}_j\}_{j=1}^M$ that minimize:

$$\begin{aligned} \min_{\theta, \{\mathbf{a}_j\}} \sum_{j=1}^M \sum_{i=1}^{D_j} \ell(\mathbf{x}_f^{(ij)}, \bar{\mathbf{x}}_f^{(ij)}) + \\ \lambda_\theta \sum_{j=1}^M \sum_{i=1}^{D_j} \|\mathbf{W}_\theta(\mathbf{x}_0^{(ij)})\|^2 + \lambda_\mathbf{a} \sum_{j=1}^M \|\mathbf{a}_j\|^2 \\ \text{s.t. } \dot{\bar{\mathbf{x}}}^{(ij)}(t) = \mathbf{f}(\bar{\mathbf{x}}^{(ij)}(t), \mathbf{u}^{(ij)}, \bar{\mathbf{d}}_\theta^{(ij)}(t)), \\ \bar{\mathbf{d}}_\theta^{(ij)}(t) = \mathbf{W}_\theta(\bar{\mathbf{x}}^{(ij)}(t))\mathbf{a}_j, \\ \bar{\mathbf{x}}^{(ij)}(0) = \mathbf{x}_0^{(ij)}, \quad \bar{\mathbf{x}}_f^{(ij)} = \bar{\mathbf{x}}^{(ij)}(\tau^{(ij)}), \\ \forall i = 1, \dots, D_j, \quad \forall j = 1, \dots, M, \end{aligned} \quad (4)$$

where ℓ is a distance metric on the state space.

After the offline disturbance feature identification in Problem 1, we design a controller $\mathbf{u} = \pi(\mathbf{x}, \mathbf{x}^*, \mathbf{a}; \theta)$ that tracks a desired state trajectory $\mathbf{x}^*(t)$, using the dynamics \mathbf{f} and the learned disturbance model $\mathbf{W}_\theta(\mathbf{x})$. To handle a disturbance signal $\mathbf{d}(t) = \mathbf{W}_\theta(\mathbf{x}(t))\mathbf{a}^*$ with an unknown realization \mathbf{a}^* , we augment the tracking controller with an adaptation law $\dot{\mathbf{a}} = \rho(\mathbf{x}, \mathbf{x}^*, \mathbf{a}; \theta)$, estimating \mathbf{a}^* online, so that $\limsup_{t \rightarrow \infty} \ell(\mathbf{x}(t), \mathbf{x}^*(t))$ is bounded.

III. TECHNICAL APPROACH

We present our approach in two stages: disturbance feature learning to solve Problem 1 (Sec. III-A) and geometric adaptive control design for trajectory tracking (Sec. III-B).

A. SE(3) Hamiltonian-based disturbance feature learning

To address Problem 1, we use a neural ODE network [29] whose structure respects Hamilton's equations in (3) with known generalized mass $\mathbf{M}(\mathbf{q})$, potential energy $V(\mathbf{q})$ and the input gain $\mathbf{g}(\mathbf{q})$. We introduce a disturbance model, $\mathbf{d} = \mathbf{W}_\theta(\mathbf{q}, \mathbf{p})\mathbf{a}$, where $\mathbf{W}_\theta(\mathbf{q}, \mathbf{p})$ is a neural network, and estimate its parameters θ from disturbance-corrupted data. The training data $\mathcal{D}_j = \{\mathbf{x}_0^{(ij)}, \mathbf{u}^{(ij)}, \mathbf{x}_f^{(ij)}, \tau^{(ij)}\}_{i=1}^{D_j}$ may be obtained using an odometry algorithm [30] or a motion capture system. The data collection can be performed using an existing baseline controller or a human operator manually controlling

the system under different disturbance conditions (e.g., wind, ground effect, etc. for a UAV).

We define the geometric distance metric ℓ in Problem 1 as a sum of position, orientation, and momentum errors:

$$\ell(\mathbf{x}, \bar{\mathbf{x}}) = \ell_{\mathbf{p}}(\mathbf{x}, \bar{\mathbf{x}}) + \ell_{\mathbf{R}}(\mathbf{x}, \bar{\mathbf{x}}) + \ell_{\mathbf{p}}(\mathbf{x}, \bar{\mathbf{x}}), \quad (5)$$

where $\ell_{\mathbf{p}}(\mathbf{x}, \bar{\mathbf{x}}) = \|\mathbf{p} - \bar{\mathbf{p}}\|_2^2$, $\ell_{\mathbf{p}}(\mathbf{x}, \bar{\mathbf{x}}) = \|\mathbf{p} - \bar{\mathbf{p}}\|_2^2$, $\ell_{\mathbf{R}}(\mathbf{x}, \bar{\mathbf{x}}) = \|(\log(\bar{\mathbf{R}}\mathbf{R}^T))^\vee\|_2^2$, $\log : SE(3) \mapsto \mathfrak{so}(3)$ is the inverse of the exponential map, associating a rotation matrix to a skew-symmetric matrix, and $(\cdot)^\vee : \mathfrak{so}(3) \mapsto \mathbb{R}^3$ is the inverse of the hat map (\cdot) . Let $\mathcal{L}(\theta, \{\mathbf{a}_j\}; \mathcal{D})$ be the total loss in Problem 1. To calculate the loss, for each dataset \mathcal{D}_j with disturbance $\bar{\mathbf{d}}_\theta^{(ij)}(t) = \mathbf{W}_\theta(\bar{\mathbf{x}}^{(ij)}(t))\mathbf{a}_j$, we solve an ODE:

$$\dot{\bar{\mathbf{x}}}^{(ij)} = \mathbf{f}(\bar{\mathbf{x}}^{(ij)}, \mathbf{u}^{(ij)}, \bar{\mathbf{d}}_\theta^{(ij)}), \quad \bar{\mathbf{x}}^{(ij)}(0) = \mathbf{x}_0^{(ij)}, \quad (6)$$

using an ODE solver. This generates a predicted state $\bar{\mathbf{x}}_f^{(ij)}$ at time $\tau^{(ij)}$ for each $i = 1, \dots, D_j$ and $j = 1, \dots, M$:

$$\bar{\mathbf{x}}_f^{(ij)} = \text{ODESolver}(\mathbf{x}_0^{(ij)}, \mathbf{f}, \tau^{(ij)}; \theta), \quad (7)$$

sufficient to compute $\mathcal{L}(\theta, \{\mathbf{a}_j\}; \mathcal{D})$. The parameters θ and $\{\mathbf{a}_j\}$ are updated using gradient descent by back-propagating the loss through the neural ODE solver using adjoint states $\mathbf{y} = \frac{\partial \mathcal{L}}{\partial \bar{\mathbf{x}}}$ [29]. An augmented state $\mathbf{s} = \left(\bar{\mathbf{x}}, \mathbf{y}, \frac{\partial \mathcal{L}}{\partial \theta}, \left\{ \frac{\partial \mathcal{L}}{\partial \mathbf{a}_j} \right\} \right)$ satisfies $\dot{\mathbf{s}} = \mathbf{f}_s = \left(\mathbf{f}, -\mathbf{y}^\top \frac{\partial \mathbf{f}}{\partial \bar{\mathbf{x}}}, -\mathbf{y}^\top \frac{\partial \mathbf{f}}{\partial \theta}, \left\{ -\mathbf{y}^\top \frac{\partial \mathbf{f}}{\partial \mathbf{a}_j} \right\} \right)$. The gradients $\frac{\partial \mathcal{L}}{\partial \theta}$ and $\left\{ \frac{\partial \mathcal{L}}{\partial \mathbf{a}_j} \right\}$ are obtained by a call to a reverse-time ODE solver starting from $\mathbf{s}_f = \mathbf{s}_f(\tau^{(ij)})$ [29]:

$$\mathbf{s}_0 = \left(\bar{\mathbf{x}}_0, \mathbf{a}_0, \frac{\partial \mathcal{L}}{\partial \theta}, \left\{ \frac{\partial \mathcal{L}}{\partial \mathbf{a}_j} \right\} \right) = \text{ODESolver}(\mathbf{s}_f, \mathbf{f}_s, \tau^{(ij)}). \quad (8)$$

B. Data-driven geometric adaptive control

Given the learned disturbance model $\mathbf{W}_\theta(\mathbf{x})$ and a desired trajectory $\mathbf{x}^*(t)$, we develop a trajectory tracking controller $\mathbf{u} = \pi(\mathbf{x}, \mathbf{x}^*, \mathbf{a}; \theta)$ that compensates for disturbances and an adaptation law $\dot{\mathbf{a}} = \rho(\mathbf{x}, \mathbf{x}^*, \mathbf{a}; \theta)$ that estimates the disturbance realization online.

Our tracking controller for the Hamiltonian dynamics in (3) is developed using interconnection and damping assignment passivity-based control (IDA-PBC) [31]. Consider a desired pose-velocity trajectory $(\mathbf{q}^*(t), \zeta^*(t))$. Since the momentum \mathbf{p} is defined in the body inertial frame, the desired momentum $\mathbf{p}^*(t)$ should be computed by transforming the desired velocity $\zeta^* = [\mathbf{v}^{*\top} \ \omega^{*\top}]^\top$ to the body frame as $\mathbf{p}^* = \mathbf{M}(\mathbf{q}) \begin{bmatrix} \mathbf{R}^\top \mathbf{R}^* \mathbf{v}^* \\ \mathbf{R}^\top \mathbf{R}^* \omega^* \end{bmatrix}$. The Hamiltonian of the system (3) is not necessarily minimized along $\mathbf{x}^*(t) = (\mathbf{q}^*(t), \mathbf{p}^*(t))$. The key idea of an IDA-PBC design is to choose the control input $\mathbf{u}(t)$ so that the closed-loop system has a desired Hamiltonian $\mathcal{H}_d(\mathbf{q}, \mathbf{p})$, which is minimized along $\mathbf{x}^*(t)$. Using quadratic errors in the position, orientation, and momentum, we design the desired Hamiltonian:

$$\begin{aligned} \mathcal{H}_d(\mathbf{q}, \mathbf{p}) &= \frac{1}{2} k_{\mathbf{p}} (\mathbf{p} - \mathbf{p}^*)^\top (\mathbf{p} - \mathbf{p}^*) \\ &+ \frac{1}{2} k_{\mathbf{R}} \text{tr}(\mathbf{I} - \mathbf{R}^{*\top} \mathbf{R}) + \frac{1}{2} (\mathbf{p} - \mathbf{p}^*)^\top \mathbf{M}^{-1}(\mathbf{q}) (\mathbf{p} - \mathbf{p}^*), \end{aligned} \quad (9)$$

where $k_{\mathbf{p}}$ and $k_{\mathbf{R}}$ are positive gains. We solve a set of matching conditions, described in [5], [31], between the original dynamics (3) with Hamiltonian $\mathcal{H}(\mathbf{q}, \mathbf{p})$ and the desired dynamics with Hamiltonian $\mathcal{H}_d(\mathbf{q}, \mathbf{p})$ in (9) to arrive at a tracking controller $\mathbf{u} = \pi(\mathbf{x}, \mathbf{x}^*, \mathbf{a}; \theta)$. The controller consists of an energy-shaping term \mathbf{u}_{ES} , a damping-injection term \mathbf{u}_{DI} , and a disturbance compensation term \mathbf{u}_{DC} :

$$\begin{aligned} \mathbf{u}_{ES} &= \mathbf{g}^\dagger(\mathbf{q}) \left(\mathbf{q}^{\times\top} \frac{\partial V}{\partial \mathbf{q}} - \mathbf{p}^{\times} \mathbf{M}^{-1}(\mathbf{q}) \mathbf{p} - \mathbf{e}(\mathbf{q}, \mathbf{q}^*) + \dot{\mathbf{p}}^* \right), \\ \mathbf{u}_{DI} &= -\mathbf{K}_d \mathbf{g}^\dagger(\mathbf{q}) \mathbf{M}^{-1}(\mathbf{q}) (\mathbf{p} - \mathbf{p}^*), \\ \mathbf{u}_{DC} &= -\mathbf{g}^\dagger(\mathbf{q}) \mathbf{W}(\mathbf{q}, \mathbf{p}) \mathbf{a}, \end{aligned} \quad (10)$$

where $\mathbf{g}^\dagger(\mathbf{q}) = (\mathbf{g}^\top(\mathbf{q}) \mathbf{g}(\mathbf{q}))^{-1} \mathbf{g}^\top(\mathbf{q})$ is the pseudo-inverse of $\mathbf{g}(\mathbf{q})$ and $\mathbf{K}_d = \text{diag}(k_{\mathbf{v}} \mathbf{I}, k_{\omega} \mathbf{I})$ is a damping gain with positive terms $k_{\mathbf{v}}, k_{\omega}$. The controller utilizes a generalized coordinate error between \mathbf{q} and \mathbf{q}^* :

$$\mathbf{e}(\mathbf{q}, \mathbf{q}^*) = \begin{bmatrix} \mathbf{e}_{\mathbf{p}}(\mathbf{q}, \mathbf{q}^*) \\ \mathbf{e}_{\mathbf{R}}(\mathbf{q}, \mathbf{q}^*) \end{bmatrix} = \begin{bmatrix} k_{\mathbf{p}} \mathbf{R}^\top (\mathbf{p} - \mathbf{p}^*) \\ \frac{1}{2} k_{\mathbf{R}} (\mathbf{R}^{*\top} \mathbf{R} - \mathbf{R}^\top \mathbf{R}^*)^\vee \end{bmatrix} \quad (11)$$

and a generalized momentum error $\mathbf{p}_e = \mathbf{p} - \mathbf{p}^*$:

$$\mathbf{p}_e = \mathbf{M}(\mathbf{q}) \begin{bmatrix} \mathbf{e}_{\mathbf{v}}(\mathbf{x}, \mathbf{x}^*) \\ \mathbf{e}_{\omega}(\mathbf{x}, \mathbf{x}^*) \end{bmatrix} = \mathbf{M}(\mathbf{q}) \begin{bmatrix} \mathbf{v} - \mathbf{R}^\top \mathbf{R}^* \mathbf{v}^* \\ \omega - \mathbf{R}^\top \mathbf{R}^* \omega^* \end{bmatrix}. \quad (12)$$

Please refer to [5] for a detailed derivation of \mathbf{u}_{ES} and \mathbf{u}_{DI} .

The disturbance compensation term \mathbf{u}_{DC} in (10) requires online estimation of the disturbance feature weights \mathbf{a} . Inspired by [17], we design an adaptation law which utilizes the geometric errors (11), (12) to update the weights \mathbf{a} :

$$\begin{aligned} \dot{\mathbf{a}} &= \rho(\mathbf{x}, \mathbf{x}^*, \mathbf{a}; \theta) \\ &= \mathbf{W}_\theta^\top(\mathbf{q}, \mathbf{p}) \begin{bmatrix} c_{\mathbf{p}} \mathbf{e}_{\mathbf{p}}(\mathbf{q}, \mathbf{q}^*) + c_{\mathbf{v}} \mathbf{e}_{\mathbf{v}}(\mathbf{x}, \mathbf{x}^*) \\ c_{\mathbf{R}} \mathbf{e}_{\mathbf{R}}(\mathbf{q}, \mathbf{q}^*) + c_{\omega} \mathbf{e}_{\omega}(\mathbf{x}, \mathbf{x}^*) \end{bmatrix}, \end{aligned} \quad (13)$$

where $c_{\mathbf{p}}, c_{\mathbf{v}}, c_{\mathbf{R}}, c_{\omega}$ are positive coefficients. The stability of our adaptive controller (π, ρ) is shown in Theorem 1, under the assumption that the learned disturbance features \mathbf{W}_θ converge to the true ones $\mathbf{W}(\mathbf{q}, \mathbf{p})$ after the training process.

Theorem 1. Consider the Hamiltonian dynamics in (3) with disturbance model in (2). Suppose that the parameters $\mathbf{g}(\mathbf{q})$, $\mathbf{M}(\mathbf{q})$, $V(\mathbf{q})$ and $\mathbf{W}(\mathbf{q}, \mathbf{p})$ are known but the disturbance feature weights \mathbf{a}^* are unknown. Let $\mathbf{x}^*(t)$ be a desired state trajectory with bounded angular velocity, $\|\omega^*(t)\| \leq \gamma$. Assume that the initial system state lies in the domain $\mathcal{T} = \{\mathbf{x} \in T^* SE(3) \mid \Psi(\mathbf{R}, \mathbf{R}^*) < \alpha < 2, \|\mathbf{e}_{\omega}(\mathbf{x}, \mathbf{x}^*)\| < \beta\}$ for some positive constants α and β , where $\Psi(\mathbf{R}, \mathbf{R}^*) = \frac{1}{2} \text{tr}(\mathbf{I} - \mathbf{R}^{*\top} \mathbf{R})$. Consider the tracking controller in (10) with adaptation law in (13). Then, there exist positive constants $k_{\mathbf{p}}, k_{\mathbf{R}}, k_{\mathbf{v}}, k_{\omega}, c_{\mathbf{p}} = c_{\mathbf{R}} = c_1, c_{\mathbf{v}} = c_{\omega} = c_2$ such that the tracking errors $\mathbf{e}(\mathbf{q}, \mathbf{q}^*)$ and \mathbf{p}_e defined in (11) and (12) converge to zero. Also, the estimation error $\mathbf{e}_{\mathbf{a}} = \mathbf{a} - \mathbf{a}^*$ is stable in the sense of Lyapunov and uniformly bounded. An estimate of the region of attraction is $\mathcal{R} = \{\mathbf{x} \in \mathcal{T} \mid \mathcal{V}(\mathbf{x}) \leq \delta\}$, where:

$$\mathcal{V}(\mathbf{q}, \mathbf{p}) = \mathcal{H}_d(\mathbf{q}, \mathbf{p}) + \frac{c_1}{c_2} \mathbf{e}^\top \mathbf{p}_e + \frac{1}{2c_2} \|\mathbf{e}_{\mathbf{a}}\|_2^2 \quad (14)$$

and $\delta < \lambda_{\min}(\mathbf{Q}_1) \min(\alpha(2 - \alpha) k_{\mathbf{R}}^2, \beta^2 \lambda_{\min}^2(\mathbf{M}(\mathbf{q}))) / 2$ for

$$\mathbf{Q}_1 = \begin{bmatrix} \min\{k_{\mathbf{p}}^{-1}, k_{\mathbf{R}}^{-1}\} & -c_1/c_2 \\ -c_1/c_2 & \lambda_{\min}(\mathbf{M}^{-1}(\mathbf{q})) \end{bmatrix}. \quad (15)$$

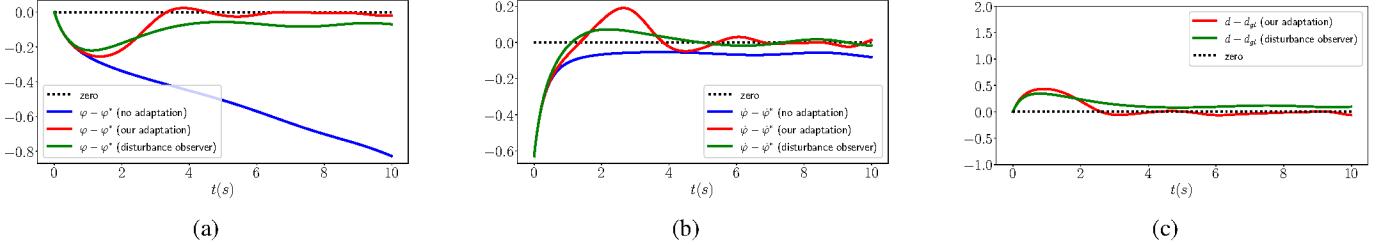


Fig. 1: Comparison of our learned adaptive controller and a disturbance observer method on a pendulum: (a) the angle error $\varphi - \varphi^*$; (b) velocity error $\dot{\varphi} - \dot{\varphi}^*$; and (c) the disturbance error $d - d_{gt}$ with the ground-truth disturbance $d_{gt} = -2.5\dot{\varphi}$.

Proof. We drop function parameters to simplify the notation. The derivative of the generalized coordinate error satisfies:

$$\begin{aligned} \dot{\mathbf{e}} &= \begin{bmatrix} \dot{\mathbf{e}}_{\mathbf{p}} \\ \dot{\mathbf{e}}_{\mathbf{R}} \end{bmatrix} = \begin{bmatrix} -\hat{\omega}\mathbf{e}_{\mathbf{p}} + k_{\mathbf{p}}\mathbf{e}_{\mathbf{v}} \\ k_{\mathbf{R}}\mathbf{E}(\mathbf{R}, \mathbf{R}^*)\mathbf{e}_{\omega} \end{bmatrix} \\ &= -\begin{bmatrix} \hat{\omega} & 0 \\ 0 & 0 \end{bmatrix}\mathbf{e} + \begin{bmatrix} k_{\mathbf{p}}\mathbf{I} & 0 \\ 0 & k_{\mathbf{R}}\mathbf{E}(\mathbf{R}, \mathbf{R}^*) \end{bmatrix}\mathbf{M}^{-1}\dot{\mathbf{p}}_e, \end{aligned} \quad (16)$$

where $\mathbf{E}(\mathbf{R}, \mathbf{R}^*) = \frac{1}{2}(\text{tr}(\mathbf{R}^T\mathbf{R}^*)\mathbf{I} - \mathbf{R}^T\mathbf{R}^*)$ satisfies $\|\mathbf{E}(\mathbf{R}, \mathbf{R}^*)\| \leq 1$. By construction of the IDA-PBC controller [5]:

$$\dot{\mathbf{p}}_e = -\mathbf{e} - \mathbf{K}_d\mathbf{M}^{-1}\mathbf{p}_e - \mathbf{W}\mathbf{e}_a. \quad (17)$$

Consider the adaptation law $\dot{\mathbf{a}} = c_1\mathbf{W}^T\mathbf{e} + c_2\mathbf{W}^T\mathbf{M}^{-1}\mathbf{p}_e$ in (13) with $c_1 = c_{\mathbf{p}} = c_{\mathbf{R}}$ and $c_2 = c_{\mathbf{v}} = c_{\omega}$. In the domain \mathcal{T} , $\Psi(\mathbf{R}, \mathbf{R}^*) < \alpha < 2$ and $\frac{k_{\mathbf{R}}^{-2}}{2}\|\mathbf{e}_{\mathbf{R}}\|_2^2 \leq \Psi(\mathbf{R}, \mathbf{R}^*) \leq \frac{k_{\mathbf{R}}^{-2}}{2-\alpha}\|\mathbf{e}_{\mathbf{R}}\|_2^2$ by [17, Prop. 1]. For $\mathbf{z} := [\|\mathbf{e}\| \ \|\mathbf{p}_e\|]^T \in \mathbb{R}^2$, the Lyapunov function candidate \mathcal{V} in (14) is bounded as:

$$\frac{1}{2}\mathbf{z}^T\mathbf{Q}_1\mathbf{z} + \frac{1}{2c_2}\|\mathbf{e}_a\|_2^2 \leq \mathcal{V} \leq \frac{1}{2}\mathbf{z}^T\mathbf{Q}_2\mathbf{z} + \frac{1}{2c_2}\|\mathbf{e}_a\|_2^2, \quad (18)$$

where the matrix \mathbf{Q}_1 is specified in (15) and \mathbf{Q}_2 is:

$$\mathbf{Q}_2 = \begin{bmatrix} \max\left\{k_{\mathbf{p}}^{-1}, \frac{2k_{\mathbf{R}}^{-1}}{2-\alpha}\right\} & c_1/c_2 \\ c_1/c_2 & \lambda_{\max}(\mathbf{M}^{-1}) \end{bmatrix}. \quad (19)$$

The time derivative of the Lyapunov candidate satisfies:

$$\begin{aligned} \dot{\mathcal{V}} &= \mathbf{p}_e^T\mathbf{M}^{-1}\dot{\mathbf{p}}_e + \mathbf{e}^T\mathbf{M}^{-1}\mathbf{p}_e + \frac{c_1\mathbf{e}^T\dot{\mathbf{p}}_e}{c_2} + \frac{c_1\dot{\mathbf{e}}^T\mathbf{p}_e}{c_2} + \frac{\mathbf{e}_a^T\dot{\mathbf{a}}}{c_2} \\ &= -\mathbf{p}_e^T\mathbf{M}^{-1}\mathbf{K}_d\mathbf{M}^{-1}\mathbf{p}_e - \frac{c_1}{c_2}\mathbf{e}^T\mathbf{e} \\ &\quad - \frac{c_1}{c_2}\mathbf{e}^T\mathbf{K}_d\mathbf{M}^{-1}\mathbf{p}_e + \frac{c_1}{c_2}\mathbf{e}^T\begin{bmatrix} \dot{\mathbf{e}}_{\omega} & 0 \\ 0 & 0 \end{bmatrix}\mathbf{p}_e \\ &\quad + \frac{c_1}{c_2}\mathbf{e}^T\begin{bmatrix} \mathbf{R}^T\mathbf{R}^*\hat{\omega}^*\mathbf{R}^{*\top}\mathbf{R} & 0 \\ 0 & 0 \end{bmatrix}\mathbf{p}_e \\ &\quad + \frac{c_1}{c_2}\mathbf{p}_e^T\mathbf{M}^{-1}\begin{bmatrix} k_{\mathbf{p}}\mathbf{I} & 0 \\ 0 & k_{\mathbf{R}}\mathbf{E}(\mathbf{R}, \mathbf{R}^*) \end{bmatrix}\mathbf{p}_e, \end{aligned}$$

where we use (16), (17), and that $\omega = \mathbf{e}_{\omega} + \mathbf{R}^T\mathbf{R}^*\omega^*$ by definition of \mathbf{e}_{ω} . Hence, in the domain \mathcal{T} , we have:

$$\frac{d}{dt}\mathcal{V} \leq -\mathbf{z}^T\mathbf{Q}_3\mathbf{z} = -\mathbf{z}^T\begin{bmatrix} q_1 & q_2 \\ q_2 & q_3 \end{bmatrix}\mathbf{z}, \quad (20)$$

where $q_1 = \frac{c_1}{c_2}$, $q_2 = -\frac{c_1}{c_2}(\lambda_{\max}(\mathbf{K}_d\mathbf{M}^{-1}) + \beta + \gamma)$, and $q_3 = \lambda_{\min}(\mathbf{M}^{-1}\mathbf{K}_d\mathbf{M}^{-1}) - \frac{c_1}{c_2}\max\{k_{\mathbf{p}}, k_{\mathbf{R}}\}\lambda_{\max}(\mathbf{M}^{-1})$.

Since $\mathbf{K}_d = \text{diag}(k_{\mathbf{p}}\mathbf{I}, k_{\omega}\mathbf{I})$ can be chosen arbitrarily large and c_1/c_2 can be chosen arbitrarily small, there exists a

TABLE I: Tracking errors and disturbance estimation error per time step (mean \pm standard deviation) with our adaptive controller, with disturbance observer (DOB), and without adaptation for 100 experiments of 10-second pendulum angle tracking.

Approach	No adaptation	Our adaptation	DOB
Angle error	0.35 ± 0.14	0.04 ± 0.02	0.08 ± 0.02
Disturbance error	0.67 ± 0.27	0.06 ± 0.03	0.10 ± 0.04

choice of constants that ensures that the matrices \mathbf{Q}_1 , \mathbf{Q}_2 , and \mathbf{Q}_3 are positive definite. Consider the sub-level set of the Lyapunov function $\mathcal{R} = \{\mathbf{x} \in \mathcal{T} | \mathcal{V}(\mathbf{x}) \leq \delta\}$ where $\delta < \lambda_{\min}(\mathbf{Q}_1)\min(\alpha(2-\alpha)k_{\mathbf{R}}^2, \beta^2\lambda_{\min}^2(\mathbf{M}))/2$. For $\mathbf{x}_0 \in \mathcal{R}$, we have $\Psi(\mathbf{R}, \mathbf{R}^*) \leq \frac{k_{\mathbf{R}}^{-2}\|\mathbf{e}_{\mathbf{R}}\|_2^2}{(2-\alpha)} \leq \frac{2\delta k_{\mathbf{R}}^{-2}}{(2-\alpha)\lambda_{\min}(\mathbf{Q}_1)} < \alpha$, and $\|\mathbf{e}_{\omega}(\mathbf{x}, \mathbf{x}^*)\|^2 \leq \frac{2\delta}{\lambda_{\min}(\mathbf{Q}_1)\lambda_{\min}^2(\mathbf{M})} \leq \beta^2$ for all $\mathbf{x}(t)$, $t > 0$, i.e., $dV/dt \leq 0$ for all $t > 0$ and \mathcal{R} is a positively invariant set. Therefore, for any system trajectory starting in \mathcal{R} , the tracking errors \mathbf{e} , \mathbf{p}_e are asymptotically stable, while the estimation error \mathbf{e}_a is stable and uniformly bounded, by the LaSalle-Yoshizawa theorem [6, Thm. A.8]. \square

IV. EVALUATION

We evaluate our data-driven geometric adaptive controller on a fully-actuated pendulum and under-actuated quadrotor.

A. Pendulum

Consider a pendulum with angle φ , scalar control input u , and dynamics $m\ddot{\varphi} = -5\sin\varphi + u + d$, where the mass is $m = 1/3$, the potential energy is $V(\varphi) = 5(1 - \cos\varphi)$, the input gain is $g(\varphi) = 1$, and the disturbance $d = -\mu\dot{\varphi}$ models a friction force with unknown friction coefficient μ . To illustrate our approach, we consider φ as a yaw angle specifying a rotation \mathbf{R} around the z axis, with angular velocity $\omega = [0, 0, \dot{\varphi}]$. We remove the position \mathbf{p} and linear velocity \mathbf{v} terms from Hamilton's equations in (3) to obtain the pendulum dynamics.

To learn the disturbance features, we consider $M = 11$ realizations of the disturbance $d_j = -\mu_j\dot{\varphi}$ with friction coefficient $\mu_j = 0.05(j-1) \in [0, 0.5]$ for $j = 1, \dots, M$. For each value μ_j , we collect transitions $\mathcal{D}_j = \{\mathbf{x}_0^{(ij)}, \mathbf{u}^{(ij)}, \mathbf{x}_f^{(ij)}, \tau^{(ij)}\}_{i=1}^{1024}$ by applying 1024 random control inputs to the pendulum for a time interval of $\tau^{(ij)} = 0.01$ s. We train the disturbance model (Sec. III-A) for 4000 iterations with learning rate 10^{-4} .

We verify our adaptive controller (π, ρ) in Sec. III-B with the task of tracking a desired angle $\varphi^*(t) = \pi t/5 + \pi t^2/50$. We simplify the controller π in (10) and the adaptation law ρ in (13) by removing the position and linear velocity

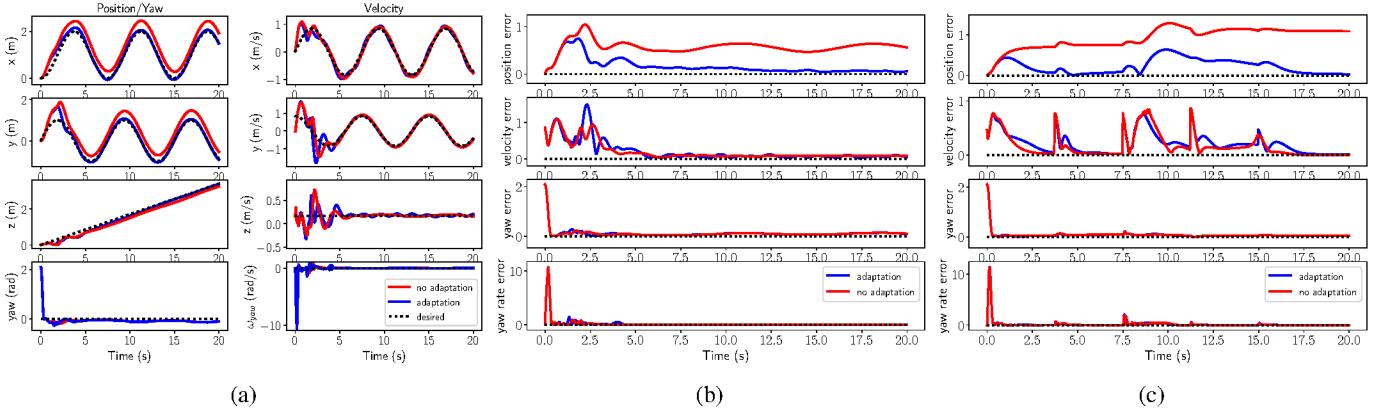


Fig. 2: Tracking spiral and diamond-shaped trajectories with a PyBullet Crazyflie quadrotor [32] under an external wind $\mathbf{d}_w = [0.075 \ 0.075 \ 0]$ and two rotors turning defective from the beginning (scenario 1) and after 8s (scenario 2) both with $(\delta_1, \delta_2) = (80\%, 80\%)$: (a) robot states in scenario 1, (b) tracking errors in scenario 1, (c) tracking errors in scenario 2.

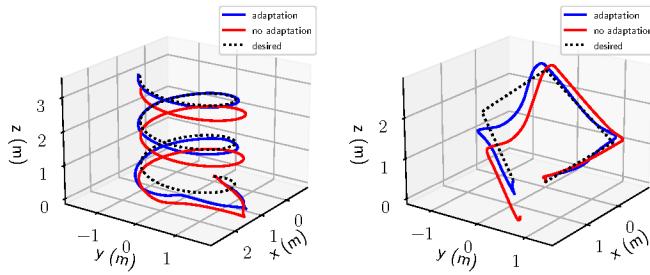


Fig. 3: Tracking visualization for scenarios 1 (left) and 2 (right).

components. The controller gains are: $k_R = 1$, $k_d = 2$, $c_R = 75$, $c_\omega = 10$. We compare our approach with a disturbance observer method [11], [12] for the pendulum. As the disturbance features in (2) are unknown, we design an observer to estimate d online. Let z be the observer state with dynamics $m\dot{z} = -l(\dot{\varphi})z - l(\dot{\varphi})(r(\dot{\varphi}) - 5 \sin(\varphi) + u)$, where $l(\dot{\varphi}) = \frac{\partial r(\dot{\varphi})}{\partial \dot{\varphi}}$ for some function $r(\dot{\varphi})$. The disturbance d is estimated as $\hat{d} = z + r(\dot{\varphi})$. The disturbance estimation error is $e_d = \hat{d} - d$ satisfying $\dot{e}_d = \dot{z} + \frac{\partial r(\dot{\varphi})}{\partial \dot{\varphi}}\dot{\varphi} = -l(\dot{\varphi})e_d/m$. We choose $r(\dot{\varphi}) = \dot{\varphi}$ so that the disturbance estimation errors converges to 0 asymptotically. While it is hard to provide a fair comparison between controllers, e.g. different control gain tuning, we try our best to match the experiment settings. For example, we use the same tracking controller π in (10) to compensate the estimated disturbance.

We run the experiments 100 times with a friction coefficient μ uniformly sampled from the range $[0.5, 3]$. Table I shows the angle tracking errors and the disturbance estimation errors with our adaptive controller, with the disturbance observer (DOB), and without adaptation. Our adaptive controller achieves better tracking error and disturbance estimation error than the DOB approach. Fig. 1 plots the tracking errors and disturbance estimation error with $\mu = 2.5$, showing that we achieve the desired angle $\varphi^*(t)$ and are able to converge to the state-dependent ground-truth disturbance $d_{gt} = -2.5\dot{\varphi}$. Without knowing the disturbance features, the DOB method lags behind the changes in ground-truth disturbances caused by the velocity $\dot{\varphi}$. This

TABLE II: Tracking errors per time step (mean \pm standard deviation) for 100 experiments of quadrotor trajectory tracking.

Experiments	Diamond-shaped	Spiral
Scenario 1 (without adaptation)	0.71 ± 0.15	0.55 ± 0.13
Scenario 1 (with adaptation)	0.12 ± 0.02	0.13 ± 0.01
Scenario 2 (without adaptation)	0.62 ± 0.13	0.64 ± 0.10
Scenario 2 (with adaptation)	0.12 ± 0.02	0.16 ± 0.02

illustrates the benefit of our approach – the learned disturbance features improve the performance of the adaptive controller.

B. Crazyflie Quadrotor

Next, we consider a Crazyflie quadrotor, simulated using the PyBullet physics engine [32], with control input $\mathbf{u} = [f, \boldsymbol{\tau}]$ including the thrust $f \in \mathbb{R}_{\geq 0}$ and torque $\boldsymbol{\tau} \in \mathbb{R}^3$ generated by the 4 rotors. The mass of the quadrotor is $m = 0.027$ kg and the inertia matrix is $\mathbf{J} = 10^{-5} \text{ diag}([1.4, 1.4, 2.2])$, leading to the generalized mass matrix $\mathbf{M}(\mathbf{q}) = \text{diag}(m\mathbf{I}, \mathbf{J})$. The potential energy is $V(\mathbf{q}) = mg [0 \ 0 \ 1] \mathbf{p}$ with $g \approx 9.8 \text{ ms}^{-2}$, where \mathbf{p} is the position of the quadrotor. We consider disturbances from three sources: 1) horizontal wind, simulated as an external force $\mathbf{d}_w = [w_x \ w_y \ 0]^\top \in \mathbb{R}^3$ in the world frame, i.e., $\mathbf{R}^\top \mathbf{d}_w$ in the body frame; 2) two defective rotors 1 and 2, generating δ_1 and δ_2 percents of the nominal thrust, respectively; and 3) near-ground, drag, and downwash effects in the PyBullet simulated quadrotor.

As described in Sec. III-A, we learn the disturbance features $\mathbf{W}_\theta(\mathbf{q}, \mathbf{p})$ from a dataset \mathcal{D} of transitions using a Hamiltonian-based neural ODE network. We collect a dataset $\mathcal{D} = \{\mathcal{D}_j\}_{j=1}^M$ with $M = 8$ realizations of the disturbances \mathbf{d}_{wj} , δ_{1j} , and δ_{2j} . Specifically, the wind components w_{xj}, w_{yj} are chosen from the set $\{\pm 0.025, \pm 0.05\}$, while the values of δ_{1j} and δ_{2j} are sampled from the range $[94\%, 98\%]$. For each disturbance realization, a PID controller provided by [32] is used to drive the quadrotor from a random starting point to 9 different desired poses, providing transitions $\mathcal{D}_j = \{\mathbf{x}_0^{(ij)}, \mathbf{u}^{(ij)}, \mathbf{x}_f^{(ij)}, \boldsymbol{\tau}^{(ij)}\}_{i=1}^{1080}$ with $\tau^{(ij)} = 1/240$ s.

We verify our geometric adaptive controller with learned disturbance features by having the quadrotor track pre-defined trajectories in the presence of the aforementioned disturbances.

The desired trajectory is specified by the desired position $\mathbf{p}^*(t)$ and the desired heading $\psi^*(t)$. We construct an appropriate choice of \mathbf{R}^* and ω^* from $\psi^*(t)$, as described in [5], [17], to be used with the adaptive controller. The tracking controller in (10) with gains $k_p = 0.135$, $k_v = 0.0675$, $k_R = 1.0$, and $k_\omega = 0.08$, is used to obtain the control input \mathbf{u} that compensates for the disturbances. The disturbances \mathbf{d} are estimated by updating the weights \mathbf{a} using the adaptation law (13) with gains $c_p = c_R = 0.08$, $c_v = c_\omega = 0.04$.

We test the controller with wind \mathbf{d}_w , rotors 1 and 2 that become defective from the beginning (scenario 1) or during flight at $t = 8$ s (scenario 2), and near-ground, drag, and downwash effects enabled in PyBullet. We track diamond-shaped and spiral trajectories 100 times with w_x and w_y uniformly sampled from $[0, 0.075]$ and δ_1 and δ_2 drawn uniformly from $[80\%, 99\%]$. Table II shows the mean and standard deviation of the tracking errors with and without adaptation from the 100 flights. The errors with adaptation are ~ 5 times lower than without adaptation, illustrating the benefit of our adaptive control design. For $\mathbf{d}_w = [0.075 \ 0.075 \ 0]$ and $(\delta_1, \delta_2) = (80\%, 80\%)$, the quadrotor in scenario 1 without adaptation drifts while our adaptive controller estimates the disturbances online after a few seconds and successfully tracks the trajectory as seen in Fig. 2a, 2b and 3 (left). For the same wind, the quadrotor in scenario 2 with our controller starts to track the trajectory, then drops down at $t = 8$ s, due to the rotors becoming defective, but recovers as our adaptation law updates the disturbances accordingly, as seen in Fig. 2c and 3 (right). The velocity error spikes in Fig. 2c are caused by sharp turns in the diamond-shaped trajectory and the defective rotors at $t = 8$ s. Without adaptation, the quadrotor drops to the ground at $t \approx 12.5$ s, shown in Fig. 3 (right). In Fig. 2 and 3, the tracking errors with adaptation stabilize close to but not exactly 0 because of the disturbance feature approximation gap between $\mathbf{W}_\theta(\mathbf{q}, \mathbf{p})$ and $\mathbf{W}(\mathbf{q}, \mathbf{p})$, and the control input discretization in time.

V. CONCLUSION

This paper introduced a neural ODE network for disturbance feature learning using disturbance-corrupted trajectory data from a rigid-body system with Hamiltonian dynamics. To enable trajectory tracking with online disturbance compensation, we designed a passivity-based tracking controller and augmented it with an adaptation law that compensates disturbances relying on the learned features and geometric tracking errors. Our evaluation showed that our adaptive controller quickly estimates disturbances online and successfully tracks desired trajectories, outperforming adaptation methods without learned disturbance features. Future work will focus on deploying the proposed controller on real robot systems.

REFERENCES

- [1] L. Ljung, “System identification,” *Wiley encyclopedia of electrical and electronics engineering*, 1999.
- [2] D. Nguyen-Tuong and J. Peters, “Model learning for robot control: a survey,” *Cognitive processing*, vol. 12, no. 4, 2011.
- [3] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, “Gaussian processes for data-efficient learning in robotics and control,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, 2015.
- [4] S. Greydanus, M. Dzamba, and J. Yosinski, “Hamiltonian neural networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.
- [5] T. Duong and N. Atanasov, “Hamiltonian-based Neural ODE Networks on the SE(3) Manifold For Dynamics Learning and Control,” in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.
- [6] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., 1995.
- [7] P. A. Ioannou and J. Sun, *Robust adaptive control*. Prentice Hall, 1996.
- [8] N. Hovakimyan and C. Cao, *\mathcal{L}_1 adaptive control theory: Guaranteed robustness with fast adaptation*. SIAM, 2010.
- [9] A. Gahlawat, P. Zhao, A. Patterson, N. Hovakimyan, and E. Theodorou, “ \mathcal{L}_1 -gp: \mathcal{L}_1 adaptive control with bayesian learning,” in *Conference on Learning for Dynamics and Control*, 2020.
- [10] D. Hanover, P. Foehn, S. Sun, E. Kaufmann, and D. Scaramuzza, “Performance, Precision, and Payloads: Adaptive Nonlinear MPC for Quadrotors,” in *arXiv cs.RO*: 2109.04210, 2021.
- [11] W.-H. Chen, D. J. Ballance, P. J. Gawthrop, and J. O'Reilly, “A nonlinear disturbance observer for robotic manipulators,” *IEEE Transactions on Industrial Electronics*, vol. 47, no. 4, pp. 932–938, 2000.
- [12] S. Li, J. Yang, W.-H. Chen, and X. Chen, *Disturbance observer-based control: methods and applications*. CRC press, 2014.
- [13] J.-J. E. Slotine and W. Li, “On the adaptive control of robot manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 3, 1987.
- [14] J.-J. E. Slotine and M. Di Benedetto, “Hamiltonian adaptive control of spacecraft,” *IEEE Transactions on Automatic Control*, vol. 35, no. 7, pp. 848–852, 1990.
- [15] D. A. Dirksz and J. M. Scherpen, “Structure preserving adaptive control of port-Hamiltonian systems,” *IEEE Transactions on Automatic Control*, vol. 57, no. 11, 2012.
- [16] S. S. Sastry and A. Isidori, “Adaptive control of linearizable systems,” *IEEE Transactions on Automatic Control*, vol. 34, no. 11, 1989.
- [17] F. A. Goodarzi, D. Lee, and T. Lee, “Geometric adaptive tracking control of a quadrotor unmanned aerial vehicle on SE(3) for agile maneuvers,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 9, 2015.
- [18] M. Bisheban and T. Lee, “Geometric adaptive control with neural networks for a quadrotor in wind fields,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1533–1548, 2020.
- [19] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991.
- [20] S. P. Nageshrao, G. A. Lopes, D. Jeltsema, and R. Babuška, “Port-Hamiltonian systems in adaptive and learning control: A survey,” *IEEE Transactions on Automatic Control*, vol. 61, no. 5, 2015.
- [21] K. Pereira and A. P. Schoellig, “Adaptive model predictive control for high-accuracy trajectory tracking in changing conditions,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.
- [22] R. C. Grande, G. Chowdhary, and J. P. How, “Nonparametric adaptive control using gaussian processes with online hyperparameter estimation,” in *52nd IEEE Conference on Decision and Control*, 2013.
- [23] G. Joshi, J. Virdi, and G. Chowdhary, “Asynchronous deep model reference adaptive control,” *arXiv preprint arXiv:2011.02920*, 2020.
- [24] S. M. Richards, N. Azizan, J.-J. Slotine, and M. Pavone, “Adaptive-Control-Oriented Meta-Learning for Nonlinear Systems,” in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.
- [25] J. Harrison, A. Sharma, and M. Pavone, “Meta-learning priors for efficient online bayesian regression,” *arXiv:1807.08912*, 2018.
- [26] G. Shi, K. Azizzadenesheli, S.-J. Chung, and Y. Yue, “Meta-adaptive nonlinear control: Theory and algorithms,” *arXiv:2106.06098*, 2021.
- [27] T. Lee, M. Leok, and N. H. McClamroch, *Global formulations of Lagrangian and Hamiltonian dynamics on manifolds*. Springer, 2017.
- [28] P. Forni, D. Jeltsema, and G. A. Lopes, “Port-Hamiltonian formulation of rigid-body attitude control,” *IFAC-PapersOnLine*, 2015.
- [29] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural ordinary differential equations,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- [30] S. A. S. Mohamed, M. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, “A survey on odometry for autonomous navigation systems,” *IEEE Access*, vol. 7, 2019.
- [31] A. Van Der Schaft and D. Jeltsema, “Port-Hamiltonian systems theory: An introductory overview,” *Foundations and Trends in Systems and Control*, vol. 1, no. 2-3, 2014.
- [32] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schoellig, “Learning to Fly—a Gym Environment with PyBullet Physics for Reinforcement Learning of Multi-agent Quadcopter Control,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2021.