Physics-guided Learning-based Adaptive Control on the SE(3) Manifold

Thai Duong **Nikolay Atanasov**
Department of Electrical and Computer Engineering
University of California San Diego
La Jolla, CA 92093, USA
e-mail: {tduong,natanasov}@ucsd.edu

Abstract

In real-world robotics applications, accurate models of robot dynamics are critical for safe and stable control in rapidly changing operational conditions. This motivates the use of machine learning techniques to approximate robot dynamics and their disturbances over a training set of state-control trajectories. This paper demonstrates that inductive biases arising from physics laws can be used to improve the data efficiency and accuracy of the approximated dynamics model. For example, the dynamics of many robots, including ground, aerial, and underwater vehicles, are described using their $SE(3)$ pose and satisfy conservation of energy principles. We design a physically plausible model of the robot dynamics by imposing the structure of Hamilton's equations of motion in the design of a neural ordinary differential equation (ODE) network. The Hamiltonian structure guarantees satisfaction of $SE(3)$ kinematic constraints and energy conservation by construction. It also allows us to derive an energy-based adaptive controller that achieves trajectory tracking while compensating for disturbances. Our learning-based adaptive controller is verified on an under-actuated quadrotor robot.

1 Introduction

Autonomous mobile robots operating in real-world complex and dynamic conditions require accurate dynamics models [1] for motion planning and control. This has motivated the development of data-driven approaches to learn dynamics models [2–9] and disturbance models [10–17] from data. As data-driven dynamics models require large amounts of data, inductive biases such as physical knowledge about the system has been imposed on the model structure [18–22], which a black-box model might struggle to infer. Such structures also simplify the design of a nominal stable regulation or tracking controller [20–22].

This paper develops physics-guided data-driven adaptive control for rigid-body systems, such as unmanned ground vehicle (UGVs), unmanned aerial vehicle (UAVs), or unmanned underwater vehicles (UUVs), that satisfy Hamilton's equations of motion over the $SE(3)$ manifold. Given a dataset of state-control trajectories with different disturbance realizations, we impose Hamilton's equations of motion on the architecture of a neural ODE network to *learn physically plausible system dynamics and disturbance model*. The learned system dynamics respect kinematic and energy conservation constraints by construction and formulated in terms of *interpretable quantities*: mass and inertia matrix, potential energy, input gain matrix, and disturbance features. *The interpretability of the model enables us to design an adaptive controller*, consisting of an energy-based tracking controller for the learned model and an adaptation law that compensates disturbances online by scaling the learned disturbance features using the $SE(3)$ tracking error. We verify our dynamics learning and control approach on an under-actuated quadrotor robot in the PyBullet simulator [23].

Physical Reasoning and Inductive Biases for the Real World at NeurIPS 2021.

2 Problem Statement

Consider a robot 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 $\mathfrak{q} = [\mathbf{p}^\top \;\; \mathbf{r}_1^\top \;\; \mathbf{r}_2^\top \;\; \mathbf{r}_3^\top]^\top \in \mathbb{R}^{12}$ 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 state $\mathbf{x} = (\mathfrak{q}, \boldsymbol{\zeta})$ evolves on the tangent bundle $TSE(3)$ of the pose manifold $SE(3)$ and is governed by the robot dynamics: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d})$, where $\mathbf{u}$ is the control input and $\mathbf{d}$ is a disturbance signal. The disturbance $\mathbf{d}$ appears as an external force applied to the system and is modeled as a linear combination of nonlinear features $\mathbf{W}(\mathfrak{q}, \mathfrak{p}) \in \mathbb{R}^{6\times p}$: $\mathbf{d}(t) = \mathbf{W}(\mathfrak{q}(t), \mathfrak{p}(t))\mathbf{a}^*$ where $\mathbf{a}^* \in \mathbb{R}^p$ are unknown weights.

As a mechanical system, the robot obeys Hamilton's equations of motion [24]. The Hamiltonian, $\mathcal{H}(\mathfrak{q}, \mathfrak{p}) = \frac{1}{2}\mathfrak{p}^\top \mathbf{M}(\mathfrak{q})^{-1}\mathfrak{p} + V(\mathfrak{q})$, captures the total energy of the system as the sum of the kinetic energy $T(\mathfrak{q}, \mathfrak{p}) = \frac{1}{2}\mathfrak{p}^\top \mathbf{M}(\mathfrak{q})^{-1}\mathfrak{p}$ and the potential energy $V(\mathfrak{q})$, where $\mathfrak{p} = \mathbf{M}(\mathfrak{q})\boldsymbol{\zeta} \in \mathbb{R}^6$ is the generalized momentum and $\mathbf{M}(\mathfrak{q}) \in \mathbb{R}^{6\times 6}$ is the generalized mass matrix. The system dynamics are governed by Hamilton's equations:

$$\begin{bmatrix} \dot{\mathfrak{q}} \\ \dot{\mathfrak{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathfrak{q}^\times \\ -\mathfrak{q}^{\times\top} & \mathfrak{p}^\times \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{H}}{\partial \mathfrak{q}} \\ \frac{\partial \mathcal{H}}{\partial \mathfrak{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}(\mathfrak{q}) \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{0} \\ \mathbf{d} \end{bmatrix}. \tag{1}$$

The cross maps $\mathfrak{q}^\times$ and $\mathfrak{p}^\times$ are defined as $\mathfrak{q}^\times = \begin{bmatrix} \mathbf{R}^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{r}}_1^\top & \hat{\mathbf{r}}_2^\top & \hat{\mathbf{r}}_3^\top \end{bmatrix}^\top$, $\mathfrak{p}^\times = \begin{bmatrix} \mathfrak{p}_{\mathbf{v}} \\ \mathfrak{p}_{\boldsymbol{\omega}} \end{bmatrix}^\times = \begin{bmatrix} \mathbf{0} & \hat{\mathfrak{p}}_{\mathbf{v}} \\ \hat{\mathfrak{p}}_{\mathbf{v}} & \hat{\mathfrak{p}}_{\boldsymbol{\omega}} \end{bmatrix}$, where the hat map $\hat{(\cdot)} : \mathbb{R}^3 \mapsto \mathfrak{so}(3)$ constructs a skew-symmetric matrix from a 3D vector. The time derivative of the generalized velocity is:

$$\dot{\boldsymbol{\zeta}} = \left(\frac{d}{dt}\mathbf{M}^{-1}(\mathfrak{q})\right)\mathfrak{p} + \mathbf{M}^{-1}(\mathfrak{q})\dot{\mathfrak{p}}. \tag{2}$$

In summary, (1) and (2) capture the structure of the system dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d})$.

Consider a collection $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_M\}$ of state-control trajectory datasets $\mathcal{D}_j$, each collected under a different unknown disturbance realization $\mathbf{a}_j^*$, for $j = 1, \ldots, M$. A trajectory dataset $\mathcal{D}_j = \{t_{0:N}^{(ij)}, \mathbf{x}_{0:N}^{(ij)}, \mathbf{u}^{(ij)}\}_{i=1}^{D_j}$ consists of D_j state sequences $\mathbf{x}_{0:N}^{(ij)}$, obtained by applying a constant control input $\mathbf{u}^{(ij)}$ to the system with initial condition $\mathbf{x}_0^{(ij)}$ at time $t_0^{(ij)}$ and sampling its state $\mathbf{x}^{(ij)}(t_n^{(ij)}) =: \mathbf{x}_n^{(ij)}$ at times $t_{0:N}^{(ij)}$. Our objective is to approximate the dynamics $\mathbf{f}$ by $\bar{\mathbf{f}}_{\boldsymbol{\theta}}$, where the parameters $\boldsymbol{\theta}$ characterize the unknowns $\mathbf{M}_{\boldsymbol{\theta}}(\mathfrak{q})$, $V_{\boldsymbol{\theta}}(\mathfrak{q})$, $\mathbf{g}_{\boldsymbol{\theta}}(\mathfrak{q})$, as well as the disturbance model by $\mathbf{W}_{\boldsymbol{\phi}}(\mathfrak{q}, \mathfrak{p})\mathbf{a}_j$, where the parameters $\boldsymbol{\phi}$, $\{\mathbf{a}_j\}_{j=1}^M$ model each disturbance sample. To optimize $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\{\mathbf{a}_j\}$, we predict the state sequence $\bar{\mathbf{x}}_{1:N}^{(ij)}$ using the approximated dynamics starting from state $\mathbf{x}_0^{(ij)}$ with a constant control $\mathbf{u}^{(ij)}$ and minimize the loss as defined in Problem 1.

Problem 1. Given $\mathcal{D} = \{\{t_{0:N}^{(ij)}, \mathbf{x}_{0:N}^{(ij)}, \mathbf{u}^{(ij)}\}_{i=1}^{D_j}\}_{j=1}^M$, find parameters $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\{\mathbf{a}_j\}_{j=1}^M$ that minimize:

$$\begin{aligned} &\min_{\boldsymbol{\theta}, \boldsymbol{\phi}, \{\mathbf{a}_j\}} \sum_{j=1}^{M} \sum_{i=1}^{D_j} \sum_{n=1}^{N} \ell(\mathbf{x}_n^{(ij)}, \bar{\mathbf{x}}_n^{(ij)}) \\ &\quad \text{s.t.} \;\; \dot{\bar{\mathbf{x}}}^{(ij)}(t) = \bar{\mathbf{f}}_{\boldsymbol{\theta}}(\bar{\mathbf{x}}^{(ij)}(t), \mathbf{u}^{(ij)}, \bar{\mathbf{d}}^{(ij)}(t)), \;\; \bar{\mathbf{d}}^{(ij)}(t) = \mathbf{W}_{\boldsymbol{\phi}}(\bar{\mathbf{x}}^{(ij)}(t))\mathbf{a}_j, \\ &\qquad\quad \bar{\mathbf{x}}^{(ij)}(t_0) = \mathbf{x}_0^{(ij)}, \;\; \bar{\mathbf{x}}_n^{(ij)} = \bar{\mathbf{x}}^{(ij)}(t_n), \forall j = 1, \ldots, M, \;\; \forall n = 1, \ldots, N, \;\; \forall i = 1, \ldots, D_j, \end{aligned} \tag{3}$$

where ℓ is a distance metric on the state space $TSE(3)$.

After the offline system identification in Problem 1, we aim to design a controller $\mathbf{u} = \boldsymbol{\pi}(\mathbf{x}, \mathbf{x}^*, \mathbf{a}; \boldsymbol{\theta}, \boldsymbol{\phi})$ that tracks a desired state trajectory $\mathbf{x}^*(t)$, $t \geq t_0$, using the learned dynamics $\bar{\mathbf{f}}_{\boldsymbol{\theta}}$ and disturbance $\mathbf{W}_{\boldsymbol{\phi}}$ models. To handle disturbance realizations $\mathbf{d} = \mathbf{W}_{\boldsymbol{\phi}}(\mathbf{x})\mathbf{a}^*$ with unknown ground-truth $\mathbf{a}^*$, we augment the tracking controller with an adaptation law $\dot{\mathbf{a}} = \boldsymbol{\rho}(\mathbf{x}, \mathbf{x}^*, \mathbf{a}; \boldsymbol{\phi})$ that estimates $\mathbf{a}^*$ online.

3 Technical Approach

We present our approach in two stages: Hamiltonian-based dynamics and disturbance model learning (Problem 1) and adaptive control design with disturbance compensation.

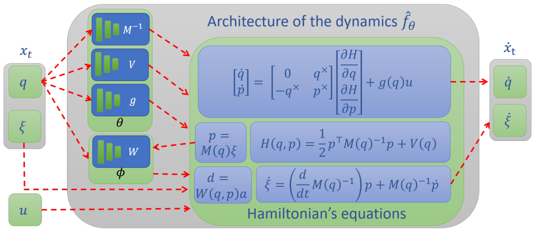

x_t
q
ξ
u
M^{-1}
V
g
θ
W
ϕ
Architecture of the dynamics $\hat{f}_\theta$
$\begin{bmatrix}\dot{q}\\ \dot{p}\end{bmatrix} = \begin{bmatrix}0 & q^\times\\ -q^\times & p^\times\end{bmatrix}\begin{bmatrix}\frac{\partial H}{\partial q}\\ \frac{\partial H}{\partial p}\end{bmatrix} + g(q)u$
$p = M(q)\xi$
$H(q,p) = \frac{1}{2}p^\top M(q)^{-1}p + V(q)$
$d = W(q,p)a$
$\dot{\xi} = \left(\frac{d}{dt}M(q)^{-1}\right)p + M(q)^{-1}\dot{p}$
Hamiltonian's equations
$\dot{x}_t$
$\dot{q}$
$\dot{\xi}$

Figure 1: Architecture of the proposed physics-guided neural ODE networks.

Hamiltonian-based dynamics learning with disturbances. We use a neural ODE network [25] whose structure (Fig. 1) respects Hamilton's equations (1)(2) and, hence, guarantees satisfaction of the $SE(3)$ kinematic constraints and energy conservation by construction. Specifically, the unknowns $\mathbf{M}_{\boldsymbol{\theta}}(\mathfrak{q})$, $\mathbf{V}_{\boldsymbol{\theta}}(\mathfrak{q})$ and $\mathbf{g}_{\boldsymbol{\theta}}(\mathfrak{q})$ are approximated by three separate neural networks. We extend the work on Hamiltonian-based neural ODE learning [22] by introducing a disturbance model, $\mathbf{d} = \mathbf{W}_{\boldsymbol{\phi}}(\mathfrak{q}, \mathfrak{p})\mathbf{a}$, described by another neural network, and estimating its parameters $\boldsymbol{\phi}$, $\mathbf{a}$ along with the system dynamics parameters $\boldsymbol{\theta}$. We first define the $TSE(3)$ distance metric ℓ in Problem 1 as $\ell(\mathbf{x}, \bar{\mathbf{x}}) = \ell_{\mathbf{p}}(\mathbf{x}, \bar{\mathbf{x}}) + \ell_{\mathbf{R}}(\mathbf{x}, \bar{\mathbf{x}}) + \ell_{\boldsymbol{\zeta}}(\mathbf{x}, \bar{\mathbf{x}})$, where the position, orientation, and velocity errors are defined respectively as $\ell_{\mathbf{p}}(\mathbf{x}, \bar{\mathbf{x}}) = \|\mathbf{p} - \bar{\mathbf{p}}\|_2^2, \ell_{\mathbf{R}}(\mathbf{x}, \bar{\mathbf{x}}) = \|\left(\log(\bar{\mathbf{R}}\mathbf{R}^\top)\right)^\vee\|_2^2, \ell_{\boldsymbol{\zeta}}(\mathbf{x}, \bar{\mathbf{x}}) = \|\boldsymbol{\zeta} - \bar{\boldsymbol{\zeta}}\|_2^2$. The log-map $\log : SE(3) \mapsto \mathfrak{so}(3)$ 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 $\hat{(\cdot)}$. Let $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \{\mathbf{a}_j\}; \mathcal{D})$ be the total loss in Problem 1. To calculate the loss, for each dataset $\mathcal{D}_j$ with disturbance $\bar{\mathbf{d}}^{(ij)}(t) = \mathbf{W}_{\boldsymbol{\phi}}(\bar{\mathbf{x}}^{(ij)}(t))\mathbf{a}_j$, we solve an ODE: $\dot{\bar{\mathbf{x}}}^{(ij)} = \bar{\mathbf{f}}_{\boldsymbol{\theta}}(\bar{\mathbf{x}}^{(ij)}, \mathbf{u}^{(ij)}, \bar{\mathbf{d}}^{(ij)}), \quad \bar{\mathbf{x}}^{(ij)}(0) = \mathbf{x}_0^{(ij)}$, using an ODE solver [25]. This generates a predicted system state trajectory $\bar{\mathbf{x}}_{1:N}^{(ij)}$ at times $t_{1:N}^{(ij)}$ for each $i = 1, \ldots, D_j$ and $j = 1, \ldots, M$, sufficient to compute $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \{\mathbf{a}_j\}; \mathcal{D})$. The parameters $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and $\mathbf{a}_j$ are updated using gradient descent by back-propagating the loss through the neural ODE solver.

Physics-guided data-driven adaptive control. Enabled by the learned Hamiltonian-based dynamics, we develop a tracking controller $\boldsymbol{\pi}$ using the interconnection and damping assignment passivity-based control (IDA-PBC) method [26]. Consider a desired state trajectory $\mathbf{x}^*(t) = (\mathfrak{q}^*(t), \boldsymbol{\zeta}^*(t))$. The desired momentum is calculated as $\mathfrak{p}^* = \mathbf{M}(\mathfrak{q})\begin{bmatrix}\mathbf{R}^\top\mathbf{R}^*\mathbf{v}^*\\ \mathbf{R}^\top\mathbf{R}^*\boldsymbol{\omega}^*\end{bmatrix}$ by transforming the desired velocity $\boldsymbol{\zeta}^* = [\mathbf{v}^{*\top} \quad \boldsymbol{\omega}^{*\top}]^\top$ to the body frame. As the Hamiltonian of the system is not necessarily minimized along $(\mathfrak{q}^*(t), \mathfrak{p}^*(t))$, the key idea of the IDA-PBC design is to choose the control input $\mathbf{u}(t)$ so that the closed-loop system has a desired Hamiltonian $\mathcal{H}_d(\mathfrak{q}, \mathfrak{p})$ that is minimized along $(\mathfrak{q}^*(t), \mathfrak{p}^*(t))$. Using a quadratic error on the tangent bundle $TSE(3)$, we design the desired Hamiltonian:

$$\mathcal{H}_d(\mathfrak{q}, \mathfrak{p}) = \frac{1}{2}k_{\mathbf{p}}(\mathbf{p} - \mathbf{p}^*)^\top(\mathbf{p} - \mathbf{p}^*) + \frac{1}{2}k_{\mathbf{R}}\operatorname{tr}(\mathbf{I} - \mathbf{R}^{*\top}\mathbf{R}) + \frac{1}{2}(\mathfrak{p} - \mathfrak{p}^*)^\top\mathbf{M}^{-1}(\mathfrak{q})(\mathfrak{p} - \mathfrak{p}^*), \quad (4)$$

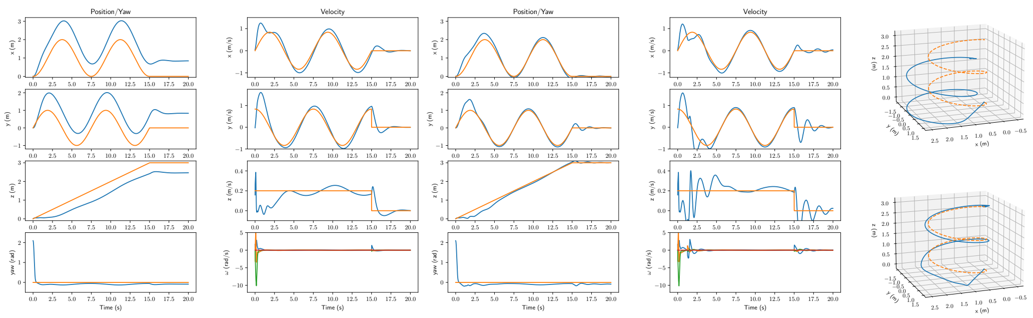
where $k_{\mathbf{p}}$ and $k_{\mathbf{R}}$ are positive gains. The controller $\boldsymbol{\pi}(\mathbf{x}, \mathbf{x}^*, \mathbf{a}; \boldsymbol{\theta}, \boldsymbol{\phi})$ consists of an energy-shaping term $\mathbf{u}_{ES}$, a damping-injection term $\mathbf{u}_{DI}$, and a disturbance compensation term $\mathbf{u}_{DC}$:

$$\begin{aligned}\mathbf{u}_{ES} &= \mathbf{g}^\dagger(\mathfrak{q})\left(\mathfrak{q}^{\times\top}\frac{\partial V}{\partial \mathfrak{q}} - \mathfrak{p}^\times\mathbf{M}^{-1}(\mathfrak{q})\mathfrak{p} - \mathbf{e}(\mathfrak{q}, \mathfrak{q}^*) + \dot{\mathfrak{p}}^*\right),\\ \mathbf{u}_{DI} &= -\mathbf{K}_{\mathbf{d}}\mathbf{g}^\dagger(\mathfrak{q})\mathbf{M}^{-1}(\mathfrak{q})(\mathfrak{p} - \mathfrak{p}^*), \text{and } \mathbf{u}_{DC} = -\mathbf{g}^\dagger(\mathfrak{q})\mathbf{W}(\mathfrak{q}, \mathfrak{p})\mathbf{a}\end{aligned} \quad (5)$$

where $\mathbf{g}^\dagger(\mathfrak{q})$ is the pseudo-inverse of $\mathbf{g}(\mathfrak{q})$, $\mathbf{K}_{\mathbf{d}} = \operatorname{diag}(k_{\mathbf{v}}\mathbf{I}, k_{\boldsymbol{\omega}}\mathbf{I})$ is a damping gain with positive terms $k_{\mathbf{v}}$, $k_{\boldsymbol{\omega}}$, the coordinate error is $\mathbf{e}(\mathfrak{q}, \mathfrak{q}^*) = \begin{bmatrix}k_{\mathbf{p}}\mathbf{e}_{\mathbf{p}}(\mathfrak{q}, \mathfrak{q}^*)\\ k_{\mathbf{R}}\mathbf{e}_{\mathbf{R}}(\mathfrak{q}, \mathfrak{q}^*)\end{bmatrix} = \begin{bmatrix}k_{\mathbf{p}}\mathbf{R}^\top(\mathbf{p} - \mathbf{p}^*)\\ \frac{1}{2}k_{\mathbf{R}}\left(\mathbf{R}^{*\top}\mathbf{R} - \mathbf{R}^\top\mathbf{R}^*\right)^\vee\end{bmatrix}$, and the momentum error is $\mathfrak{p} - \mathfrak{p}^* = \mathbf{M}(\mathfrak{q})\begin{bmatrix}\mathbf{e}_{\mathbf{v}}(\mathbf{x}, \mathbf{x}^*)\\ \mathbf{e}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{x}^*)\end{bmatrix} = \mathbf{M}(\mathfrak{q})\begin{bmatrix}\mathbf{v} - \mathbf{R}^\top\mathbf{R}^*\mathbf{v}^*\\ \boldsymbol{\omega} - \mathbf{R}^\top\mathbf{R}^*\boldsymbol{\omega}^*\end{bmatrix}$. Please refer to [22] for a detailed derivation of $\mathbf{u}_{ES}$ and $\mathbf{u}_{DI}$.

The disturbance compensation term $\mathbf{u}_{DC}$ in (5) requires online estimation of the disturbance feature weights $\mathbf{a}$. Inspired by [27], we design an adaptation law which utilizes the body-frame geometric errors $\mathbf{e}_{\mathbf{p}}, \mathbf{e}_{\mathbf{R}}, \mathbf{e}_{\mathbf{v}}$, and $\mathbf{e}_{\boldsymbol{\omega}}$ to update the disturbance feature weights:

$$\dot{\mathbf{a}} = \boldsymbol{\rho}(\mathbf{x}, \mathbf{x}^*, \mathbf{a}; \boldsymbol{\phi}) = \mathbf{W}_{\boldsymbol{\phi}}^\top(\mathfrak{q}, \mathfrak{p})\begin{bmatrix}c_{\mathbf{p}}\mathbf{e}_{\mathbf{p}}(\mathfrak{q}, \mathfrak{q}^*) + c_{\mathbf{v}}\mathbf{e}_{\mathbf{v}}(\mathbf{x}, \mathbf{x}^*)\\ c_{\mathbf{R}}\mathbf{e}_{\mathbf{R}}(\mathfrak{q}, \mathfrak{q}^*) + c_{\boldsymbol{\omega}}\mathbf{e}_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{x}^*)\end{bmatrix}, \quad (6)$$



(a)

(b)

(c)

Figure 2: Tracking a spiral trajectory with a PyBullet Crazyflie quadrotor [23] with disturbances from constant wind and defective rotors: (a)(c)-upper without adaptation and (b)(c)-lower with adaptation. With adaptation, our controller is able to estimate and compensate for the disturbances and track the trajectory successfully.

where $c_{\mathbf{p}}$, $c_{\mathbf{v}}$, $c_{\mathbf{R}}$, $c_{\boldsymbol{\omega}}$ are positive coefficients.

4 Evaluation

We consider a simulated PyBullet Crazyflie quadrotor [23], with control input $\mathbf{u} = [f, \boldsymbol{\tau}]$ including a thrust $f \in \mathbb{R}_{\geq 0}$ and a torque vector $\boldsymbol{\tau} \in \mathbb{R}^3$ generated by the 4 rotors. The disturbance $\mathbf{d}$ comes from two sources: 1) horizontal winds, simulated as an external force $\mathbf{d}_w = \begin{bmatrix} d_{wx} & d_{wy} & 0 \end{bmatrix}^\top \in \mathbb{R}^3$ in the world frame, i.e. $\mathbf{R}^\top \mathbf{d}_w$ in the body frame, applied on the quadrotor, and 2) two defective rotors 1 and 2, generating δ_1 and δ_2 percents of the nominal thrust, respectively. We collect a dataset $\mathcal{D} = \{\mathcal{D}_j\}_{j=1}^M$ with $M = 8$ realizations of the disturbance $\mathbf{d}_{wj}$, δ_{1j} and δ_{2j}. Specifically, the wind components w_{xj}, w_{yj} are chosen from the set $\{\pm 0.25, \pm 0.5\}$ 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 [23] is used to drive the quadrotor from a random starting point to 9 different desired poses, generating a dataset $\mathcal{D}_j = \{t_{0:N}^{(ij)}, \mathfrak{q}_{0:N}^{(ij)}, \boldsymbol{\zeta}_{0:N}^{(ij)}, \mathbf{u}^{(ij)}\}_{i=1}^{D_j}$ with $N = 5$ and $D_j = 1080$.

Since the mass m of the quadrotor is easily measured, we use the ground-truth value $m = 0.027$kg and form our approximated generalized mass matrix as $\mathbf{M}_{\boldsymbol{\theta}}(\mathfrak{q}) = \operatorname{diag}(m\mathbf{I}, \mathbf{J}_{\boldsymbol{\theta}}(\mathfrak{q}))$, where $\mathbf{J}_{\boldsymbol{\theta}}(\mathfrak{q})$ represents an unknown inertial matrix. The known mass m leads to a known potential energy $V(\mathfrak{q}) = mg \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{p}$, where $\mathbf{p}$ is the position of the quadrotor and $g \approx 9.8ms^{-2}$ is the gravitational acceleration. As described in Sec. 3, we learn $\mathbf{M}_{\boldsymbol{\theta}}(\mathfrak{q})$, $\mathbf{g}_{\boldsymbol{\theta}}(\mathfrak{q})$ and $\mathbf{W}_{\boldsymbol{\phi}}(\mathfrak{q}, \mathfrak{p})$ from the dataset $\mathcal{D}$.

We verify our learned adaptive controller in Sec. 3 by driving the quadrotor to track a pre-defined trajectory in the present of the aforementioned disturbance $\mathbf{d}$. The desired trajectory is specified by the desired position $\mathbf{p}^*(t)$ and the desired heading $\boldsymbol{\psi}^*(t)$, which is used to construct an appropriate choice of $\mathbf{R}^*$ and $\boldsymbol{\omega}^*$, similar to [22,28,29]. The tracking controller (5), with gains $k_{\mathbf{p}} = 0.135$, $k_{\mathbf{v}} = 0.0675$, $k_{\mathbf{R}} = 1.0$, and $k_{\boldsymbol{\omega}} = 0.08$ is paired with the adaptation law (6) with gains $c_{\mathbf{p}} = 0.04, c_{\mathbf{v}} = 0.04$, and $c_{\mathbf{R}} = c_{\boldsymbol{\omega}} = 0.5$. We test the learned adaptive controller with a task of tracking a spiral trajectory with a constant wind $\mathbf{d}_w = \begin{bmatrix} 0.75 & 0.75 & 0 \end{bmatrix}$, and defective rotors 1&2 with $\delta_1 = \delta_2 = 0.8$. The quadrotor without adaptation drifts as seen in Fig. 2a and 2c (upper). Meanwhile, Fig. 2b and 2c (lower) show that our adaptive controller is able to estimate the disturbance online after a few seconds and successfully tracks the trajectory.

5 Conclusion

This paper introduced a physics-guided gray-box model for rigid-body dynamics learning from disturbance-corrupted data. We developed a Hamiltonian neural ODE architecture which captures external force disturbances and respects the $SE(3)$ constraints and energy conservation by construction. We designed an energy-based tracking controller and an adaptation law that compensates for disturbances relying on geometric tracking errors, and verified their effectiveness on a quadrotor.

References

[1] L. Ljung, “System identification,” *Wiley encyclopedia of electrical and electronics engineering*, 1999.

[2] 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, pp. 408–423, 2015.

[3] F. Berkenkamp, A. P. Schoellig, and A. Krause, “Safe controller optimization for quadrotors with gaussian processes,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 491–496.

[4] A. H. Chang, C. M. Hubicki, J. J. Aguilar, D. I. Goldman, A. D. Ames, and P. A. Vela, “Learning to jump in granular media: Unifying optimal control synthesis with gaussian process-based regression,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2154–2160.

[5] L. Hewing, J. Kabzan, and M. N. Zeilinger, “Cautious model predictive control using gaussian process regression,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, 2019.

[6] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, “Data-driven mpc for quadrotors,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.

[7] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Multistep neural networks for data-driven discovery of nonlinear dynamical systems,” *arXiv preprint arXiv:1801.01236*, 2018.

[8] K. Chua, R. Calandra, R. McAllister, and S. Levine, “Deep reinforcement learning in a handful of trials using probabilistic dynamics models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[9] D. Nguyen-Tuong and J. Peters, “Model learning for robot control: a survey,” *Cognitive processing*, vol. 12, no. 4, 2011.

[10] R. M. Sanner and J.-J. E. Slotine, “Gaussian networks for direct adaptive control,” in *1991 American control conference*. IEEE, 1991, pp. 2153–2159.

[11] A. Gahlawat, P. Zhao, A. Patterson, N. Hovakimyan, and E. Theodorou, “L1-gp: L1 adaptive control with bayesian learning,” in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, ser. Proceedings of Machine Learning Research, vol. 120. PMLR, 10–11 Jun 2020, pp. 826–837. [Online]. Available: https://proceedings.mlr.press/v120/gahlawat20a.html

[12] 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*. IEEE, 2013, pp. 861–867.

[13] G. Joshi and G. Chowdhary, “Deep model reference adaptive control,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 4601–4608.

[14] G. Joshi, J. Virdi, and G. Chowdhary, “Asynchronous deep model reference adaptive control,” *arXiv preprint arXiv:2011.02920*, 2020.

[15] M. O’Connell, G. Shi, X. Shi, and S.-J. Chung, “Meta-learning-based robust adaptive flight control under uncertain wind conditions,” *arXiv preprint arXiv:2103.01932*, 2021.

[16] J. Harrison, A. Sharma, R. Calandra, and M. Pavone, “Control adaptation via meta-learning dynamics,” in *Workshop on Meta-Learning at NeurIPS*, vol. 2018, 2018.

[17] 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.

[18] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia, “Graph networks as learnable physics engines for inference and control,” in *International Conference on Machine Learning (ICML)*, 2018.

[19] S. Greydanus, M. Dzamba, and J. Yosinski, “Hamiltonian neural networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[20] M. Lutter, K. Listmann, and J. Peters, “Deep Lagrangian Networks for end-to-end learning of energy-based control for under-actuated systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[21] Y. D. Zhong, B. Dey, and A. Chakraborty, “Symplectic ODE-Net: learning Hamiltonian dynamics with control,” in *International Conference on Learning Representations (ICLR)*, 2019.

[22] 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.

[23] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schöllig, “Learning to fly: a pybullet gym environment to learn the control of multiple nano-quadcopters,” https://github.com/utiasDSL/gym-pybullet-drones, 2020.

[24] T. Lee, M. Leok, and N. H. McClamroch, *Global formulations of Lagrangian and Hamiltonian dynamics on manifolds*. Springer, 2017.

[25] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural ordinary differential equations,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[26] 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.

[27] J.-J. E. Slotine and W. Li, “Composite adaptive control of robot manipulators,” *Automatica*, vol. 25, no. 4, pp. 509–519, 1989.

[28] T. Lee, M. Leok, and N. H. McClamroch, “Geometric tracking control of a quadrotor UAV on SE(3),” in *IEEE Conference on Decision and Control (CDC)*, 2010.

[29] 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, p. 091007, 2015.