WATCH AND LEARN: LEARNING TO CONTROL FEEDBACK LINEARIZABLE SYSTEMS FROM EXPERT DEMONSTRATIONS*

ALIMZHAN SULTANGAZIN[1], LUCAS FRAILE[1], LUIGI PANNOCCHI[1] AND PAULO TABUADA[1]

ABSTRACT. In this paper, we revisit the problem of learning a stabilizing controller from a finite number of demonstrations by an expert. By focusing on feedback linearizable systems, we show how to combine expert demonstrations into a stabilizing controller, provided that demonstrations are sufficiently long and there are at least $n+1$ of them, where n is the number of states of the system being controlled. When we have more than $n+1$ demonstrations, we discuss how to optimally choose the best $n+1$ demonstrations to construct the stabilizing controller.

1. INTRODUCTION

1.1. **Motivation and related work.** The usefulness of learning from demonstrations has been well-argued in the literature (see, e.g., [1–3]). In the context of control, there are many tasks where providing examples of the desired behaviour is easier than defining such behaviour mathematically (e.g., driving a car in a way that is comfortable to passengers, teaching a robot to manipulate objects or play sports). The growing research interest in learning from demonstrations (LfD) for robot control [3] reflects the need for a well-defined controller design methodology for such tasks.

In this section, we present the previous work in learning from demonstrations, briefly discuss how our approach is partially inspired by the behavioural systems theory perspective, and review other works that apply the same perspective to various problems in data-driven control. This is in no way a comprehensive account of the literature on learning from demonstrations, but rather an overview of the approaches most related to ours (please refer to the surveys in [3] or [4] for a more detailed description of the literature on LfD).

Policy-learning LfD methods, to which this work belongs, assume that there exists a mapping from state (or observation) to control input that dictates the expert's behaviour. This mapping is referred to as the expert's policy. The goal of these methods is to find (or approximate) the expert's policy given a set of expert demonstrations. In many machine-learning-based LfD methods, policy learning is viewed as a supervised-learning problem where states and control inputs are treated as features and labels, respectively. We refer to these methods as *behavioural cloning* methods. Pioneered in the 80s by works like [5], this class of methods is still popular today because of their conceptual simplicity. Behavioural cloning methods are typically agnostic to the nature of the expert — demonstrations can be provided by a human (see [6, 7]), an offline optimal controller (see [8, 9]), or a controller with access to privileged state information (see [10]). They do, however, require a large number of demonstrations to work well in practice and, if trained solely on data from unmodified expert demonstrations, generate unstable policies that cannot recover from drifts or disturbances [6]. It also needs to be mentioned that the works on behavioural cloning typically provide few formal stability guarantees and quality of performance is mainly illustrated by extensive experimental results.

*This work was partially supported by the NSF grant 1705135 and by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. .

[1]Alimzhan Sultangazin, Lucas Fraile, Luigi Pannocchi and Paulo Tabuada are with Department of Electrical and Computer Engineering, University of California - Los Angeles, USA {asultangazin, lfrailev, lpannocchi, tabuada}@ucla.edu.

The authors would like to express sincere gratitude to Tzanis Anevlavis for the title and fruitful discussions.

Currently, there is a concerted effort to develop policy-learning LfD methods that improve on existing techniques using tools from control theory. In this line of effort, the work by Palan et al. [11] is conceptually the closest to ours — the authors use convex optimization to fit a linear policy to expert demonstrations stabilizing a linear system. By adding an additional set of constraints from [12] to the optimization problem, they are able to guarantee that the learned policy also stabilizes this linear system. Our methodology is different from that in [11] because we do not assume the expert's policy to be linear with respect to the current state.

In control theory, there has recently been a considerable interest in data-driven techniques. In the context of this work, we are interested in discussing the data-driven techniques that use a behavioural systems theory perspective [13,14]. The key observation used in these works is that a system can be represented by persistently exciting input-output trajectories. Although, at first glance, the problems addressed by these data-driven techniques and learning from demonstrations may appear similar, this is not exactly the case. The important distinction is that these data-driven techniques do not attempt to construct a controller that emulates the provided input-output trajectories. The data from demonstrations there serves only as a form of system representation, so it is not important whether it comes from an expert controller or not. Both our work and these data-driven techniques, however, are based on the insight that, for linear systems, any trajectory can be constructed as a linear combination of a sufficient number of trajectories.

1.2. **Contributions.** In this work, we propose a methodology for constructing a controller for a nonlinear system from a finite number of expert demonstrations of desired behaviour, provided the number of demonstrations is greater than the number of states by one and the demonstrations are sufficiently long. The approach proposed in this paper is two-fold:

- use feedback linearization to transform the nonlinear system into a chain of integrators;
- use affine combinations of the demonstrations in the transformed coordinates to construct a control law stabilizing the original system.

We formally prove the learned controller asymptotically stabilizes the system. Furthermore, we show the optimality of the proposed controller's approximation error and illustrate with simulation the application of the approach to the trajectory tracking problem. It is important to note that, unlike [11], our methodology produces a controller that is *not* linear neither in the original nor in the transformed state. This reflects our belief that, in many cases, the expert demonstration is produced by a nonlinear controller.

2. Problem Statement and Preliminaries

2.1. **Notations and basic definitions.** The notation used in this paper is fairly standard. The integers are denoted by $\mathbb{Z}$, the natural numbers, including zero, by $\mathbb{N}_0$, the real numbers by $\mathbb{R}$, and the non-negative real numbers by $\mathbb{R}_0^+$. We denote by $\|\cdot\|$ (or by $\|\cdot\|_2$ for clarity) the standard Euclidean norm or the induced matrix 2-norm; and by $\|\cdot\|_F$ the matrix Frobenius norm. A set of vectors $\{v_1, \dots, v_k\}$ in $\mathbb{R}^n$ is *affinely independent* if the set $\{v_2 - v_1, \dots, v_k - v_1\}$ is linearly independent.

A function $\alpha : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is of class $\mathcal{K}$ if α is continuous, strictly increasing, and $\alpha(0) = 0$. If α is also unbounded, it is of class $\mathcal{K}_\infty$. A function $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is of class $\mathcal{KL}$ if, for fixed $t \geq 0$, $\beta(\cdot, t)$ is of class $\mathcal{K}$ and $\beta(r, \cdot)$ decreases to 0 as $t \to \infty$ for each fixed $r \geq 0$.

The Lie derivative of a function $h : \mathbb{R}^n \to \mathbb{R}$ along a vector field $f : \mathbb{R}^n \to \mathbb{R}^n$, given by $\frac{\partial h}{\partial x} f$, is denoted by $L_f h$. We use the notation $L_f^k h$ for the iterated Lie derivative, i.e., $L_f^k h = L_f(L_f^{k-1} h)$, with $L_f^0 h = h$.

Consider a continuous-time dynamical system of the form:

$$\dot{x} = f(t, x), \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state and $f : \mathbb{R}_0^+ \times \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function. The origin of (2.1) is *uniformly asymptotically stable* if there exists $\beta \in \mathcal{KL}$ such that the following is satisfied [15]:

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0), \quad \forall t \geq t_0 \geq 0. \tag{2.2}$$

Let $C = \{c_1, \ldots, c_k\}$ be an indexed set. For any $\mathcal{I} \subset \{1, \ldots, k\}$, we define the subset $C^{\mathcal{I}} = \{c_i \in C \mid i \in \mathcal{I}\}$. A Cartesian product of two sets $C_1 \times C_2$ has a natural left projection map $\pi_1 : C_1 \times C_2 \to C_1$ (resp., right projection map $\pi_2 : C_1 \times C_2 \to C_2$) given by $\pi_1(c_1, c_2) = c_1$ (resp., $\pi_2(c_1, c_2) = c_2$).

Let $\mathcal{X} = \{x_1, \ldots, x_k\}$ be a finite set of points in $\mathbb{R}^n$. A point $x = \sum_{i=1}^k \theta_i x_i$ with $\sum_{i=1}^k \theta_i = 1$ is called a *an affine combination* of points in $\mathcal{X}$. If, in addition, $\theta_i \geq 0$ is satisfied for all $i \in \{1, \ldots, k\}$, then x is a *convex combination* of points in P. The *convex hull* of a set $\mathcal{X}$, denoted $\operatorname{conv} \mathcal{X}$, is the set of all convex combinations of points in $\mathcal{X}$ [16].

2.2. **Problem Statement.** Consider a continuous-time control-affine system:

$$\Sigma : \quad \dot{x} = f(x) + g(x)u, \tag{2.3}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and the input, respectively; and $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are smooth functions. Assume that the origin is an equilibrium point of (2.3). We call a pair $(x, u) : \mathbb{R}_0^+ \to \mathbb{R}^n \times \mathbb{R}^m$ a solution of the system (2.3) if, for all $t \in \mathbb{R}_0^+$, the equation (2.3) is satisfied. Furthermore, we refer to the functions x and u as a trajectory and a control input of the system (2.3).

Definition 2.1. A controller $u = k(x)$ is *asymptotically stabilizing* for system (2.3) if the origin is uniformly asymptotically stable for the system (2.3) with $u = k(x)$.

Suppose there exists an unknown asymptotically stabilizing controller $u = k(x)$, which we call the expert controller. We assume that $k : \mathbb{R}^n \to \mathbb{R}^m$ is smooth. Towards the goal of learning a controller $\hat{k} : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^m$ that asymptotically stabilizes the origin of the system (2.3), we assume that we are given a set of M finite-length expert solutions $\mathcal{D} = \{(x^i, u^i)\}_{i=1}^M$ of (2.3), where: for each i, the trajectory $x^i : [0, T] \to \mathbb{R}^n$ and the control input $u^i : [0, T] \to \mathbb{R}^m$ are smooth and satisfy $u^i(t) = k(x^i(t))$ for all $t \in \mathbb{R}_0^+$; $T \in \mathbb{R}$ is the length of a solution; and $M \geq n + 1$. We also ascertain that the "trivial" expert solution, wherein $x(t) = 0$ and $u(t) = 0$ for all $t \in [0, T]$, is included in $\mathcal{D}$.

Remark 2.2. In practice, we cannot record continuous solutions provided by the expert — we can only record the values of these solutions at certain sampling instants. In this work, however, we choose to work in continuous-time to simplify theoretical analysis. We can do this without sacrificing the practical applicability because it is well-known that continuous-time controller designs can be implemented via emulation and still guarantee stability [17].

We make the assumption that the system (2.3) is feedback linearizable on $\mathbb{R}^n$. To avoid the cumbersome notation that comes with feedback linearization of multiple-input systems, we assume that $m = 1$, that is, the system (2.3) has only a single input. Readers familiar with feedback linearization can verify that all the results extend to multiple-input case, mutatis mutandis (refer to [18, Ch. 4] for a complete introduction to feedback linearization). In the single-input case, the system (2.3) is feedback linearizable if there is an output function $h : \mathbb{R}^n \to \mathbb{R}$ that has relative degree n, i.e., $L_g L_f^i h(x) = 0$ for $i = 0, \ldots, n-2$ and $L_g L_f^{n-1} h(x) \neq 0$ for all $x \in \mathbb{R}^n$. We further assume, without loss of generality, that $h(0) = 0$.

3. Learning a stabilizing controller from $n + 1$ expert demonstrations

In this section, we describe the proposed methodology for constructing an asymptotically stabilizing controller based on a set of $n + 1$ demonstrations and present some of the main results.

3.1. **Feedback linearization.** Recall that using the feedback linearizability assumption, we can rewrite the nonlinear system dynamics (2.3) in the coordinates:

$$z = \Phi(x) = \begin{bmatrix} h(x) & L_f h(x) & \cdots & L_f^{n-1} h(x) \end{bmatrix}^T, \tag{3.1}$$

resulting in:

$$\begin{aligned} \dot{z}_1 &= z_2, \\ &\vdots \\ \dot{z}_{n-1} &= z_n, \\ \dot{z}_n &= a(z) + b(z)u, \end{aligned} \tag{3.2}$$

where $a = \left(L_f^n h\right) \circ \Phi^{-1}$ and $b = \left(L_g L_f^{n-1} h\right) \circ \Phi^{-1}$. The feedback law:

$$u = b(z)^{-1}(-a(z) + v), \tag{3.3}$$

further transforms the system (2.3) into a linear time-invariant (LTI) controllable system:

$$z = Az + Bv, \tag{3.4}$$

where (A, B) is a Brunovsky pair.

Remark 3.1. The expert controller $\kappa : \mathbb{R}^n \to \mathbb{R}$ in the transformed state and input coordinates is given by $\kappa(z) = a(z) + b(z)k(\Phi^{-1}(z))$. The smoothness of k implies that the function κ is also smooth.

3.2. **Expert demonstrations.** Recall that the set of demonstrations $\mathcal{D}$ satisfies the nonlinear system dynamics (2.3). Using (3.1) and (3.3), we can represent the demonstrations $\mathcal{D}$ in (z, v)-coordinates. We denote the resulting set by $\mathcal{D}_{(z,v)} = \{(z^i, v^i)\}_{i=1}^M$, where the functions $z^i : [0, T] \to \mathbb{R}^n$ and $v^i : [0, T] \to \mathbb{R}$ are:

$$z^i(t) \triangleq \Phi(x^i(t)) \tag{3.5}$$

$$v^i(t) \triangleq L_f^n h(x^i(t)) + L_g L_f^{n-1} h(x^i(t)) u^i(t), \tag{3.6}$$

for all $i \in \{1, \cdots, M\}$ and for all $t \in [0, T]$. We define the set of demonstrations $\mathcal{D}_{(z,v)}$ evaluated at time t as:

$$\mathcal{D}_{(z,v)}(t) = \{(z^i(t), v^i(t))\}_{i=1}^M. \tag{3.7}$$

It can be easily verified that the demonstrations in $\mathcal{D}_{(z,v)}$ satisfy the dynamics (3.4) and $v^i(t) = \kappa(z^i(t))$.

3.3. **Constructing the learned controller.** We denote by $v = \hat{\kappa}(t, z)$ the controller learned from the expert demonstrations. For the clarity of exposition, we assume the the number of expert demonstrations is $M = n+1$. We will consider the case when $M > n + 1$ in Section 4.

We begin by partitioning time into intervals of length T and indexing these intervals with $p \in \mathbb{N}_0$. Let us define $\mathcal{Z}(t) = \pi_1\left(\mathcal{D}_{(z,v)}(t)\right)$ and $\mathcal{V}(t) = \pi_2\left(\mathcal{D}_{(z,v)}(t)\right)$, and construct the following matrices:

$$Z(t) \triangleq \left[z^2(t) - z^1(t) \mid \cdots \mid z^{n+1}(t) - z^1(t) \right] \tag{3.8}$$

$$V(t) \triangleq \left[v^2(t) - v^1(t) \mid \cdots \mid v^{n+1}(t) - v^1(t) \right], \tag{3.9}$$

for $t \in [0, T]$. A first attempt at constructing the learned controller, which we improve upon later in the paper, would be to use the piecewise-continuous control law $v(t) = \hat{\kappa}(t - pT, z(pT))$ for all $t \in [pT, (p+1)T)$, where the value of $\hat{\kappa}(t, z)$ is given by:

$$\hat{\kappa}(t, z(pT)) = V(t - pT)\zeta(p), \tag{3.10}$$

where $\zeta(p) = Z^{-1}(0)z(pT)$, and $Z(t), V(t)$ are defined in (3.8) and (3.9), respectively.

The next lemma formally shows that an affine combination of trajectories of (3.4) is a valid trajectory for (3.4).

Lemma 3.2. *Suppose we are given a set of finite-length solutions* $\{(z^i, v^i)\}_{i=1}^{n+1}$ *of the system* (3.4)*, where each* (z^i, v^i) *is defined for* $0 \le t \le T$, $T \in \mathbb{R}$*. Assume that* $\{z^i(0)\}_{i=1}^{n+1}$ *is an affinely independent set. Then, under the control law* $v(t) = V(t - t_0)\zeta$ *with* $\zeta = Z^{-1}(0)z_0$*, the solution of the system* (3.4) *with the initial state* $z(t_0) = z_0$ *is:*

$$z(t) = Z(t - t_0)\zeta,$$

for $t_0 \le t \le T + t_0$*, where the matrices* $Z(t)$ *and* $V(t)$ *are defined in* (3.8) *and* (3.9)*, respectively.*

Proof. This lemma can be verified by substitution. □

Remark 3.3. The requirement that $\{z^i(0)\}_{i=1}^{n+1}$ is an affinely independent set is a generic property, i.e., this is true for almost all expert demonstrations. In practice, if this condition is violated, a user can eliminate one of the affinely dependent demonstrations and ask the expert to provide additional demonstrations.

We note, however, that the control law (3.10) samples the state z with a sampling time T and essentially operates in open loop in between these samples. To allow for more frequent sampling, we improve the controller (3.10) by further partitioning each interval $[pT, (p+1)T)$ into $\ell \in \mathbb{N}$ equal intervals of length $\Delta = T/\ell$ and sampling the state at the boundaries of such smaller intervals. The improved controller has, for all $t \in [pT + q\Delta, pT + (q+1)\Delta)$, the following form:

$$v(t) = \hat{\kappa}(t, z(pT + q\Delta)) = V(t - pT)\zeta(p, q), \tag{3.11}$$

where $p = \lfloor t/T \rfloor$, $q = \lfloor (t - pT)/\Delta \rfloor$, and

$$\zeta(p, q) = Z^{-1}(q\Delta)z(pT + q\Delta).$$

Note that, in the absence of uncertainties and disturbances, by Lemma 3.2, the coefficients ζ satisfy:

$$\begin{aligned} \zeta(p, q) &= Z^{-1}(q\Delta)z(pT + q\Delta) \\ &= Z^{-1}(0)z(pT), \end{aligned} \tag{3.12}$$

for all $q \in \{0, 1, \ldots, \ell - 1\}$ (i.e., the controller (3.11) applies the input equal to that applied by the controller (3.10)). In practice, however, the systems are often subject to uncertainties and disturbances and, therefore, using the controller (3.11) instead of (3.10) significantly improves robustness in realistic scenarios. The interested reader can check this by comparing the disturbance-to-state L2-gains of the system (3.4) when using (3.10) and when using (3.11).

3.4. Stability of the learned controller.

Assuming (3.12) holds, the system (3.4) in closed loop with (3.11) has the following form:

$$\dot{z} = Az + BV(t - pT)Z^{-1}(0)z(pT), \tag{3.13}$$

for all $t \in [pT, (p+1)T)$. Integrating the dynamics, we show that the sequence $\{z(pT)\}_{p \in \mathbb{N}_0}$ satisfies:

$$z((p+1)T) = \Psi(T)z(pT), \tag{3.14}$$

where

$$\Psi(T) \triangleq e^{AT} + \int_0^T e^{A(T-\tau)}BV(\tau)Z^{-1}(0)\mathrm{d}\tau. \tag{3.15}$$

By adopting a term from Floquet's theory, we refer to $\Psi(T)$ in (3.15) as the closed-loop monodromy matrix [19].

The main result of this section presents sufficient conditions for asymptotic stability of the system (2.3) in closed loop with (3.3)-(3.11).

Theorem 3.4. *Consider the feedback linearizable system* (2.3) *under the transformation* (3.1) *and the feedback law* (3.3)*. Suppose we are given a finite set of solutions* $\mathcal{D} = \{(x^i, u^i)\}_{i=1}^{n+1}$ *generated by the system* (2.3) *in closed loop with a smooth asymptotically stabilizing controller* $k : \mathbb{R}^n \to \mathbb{R}$*. Assume that* $\{\Phi(x^i(t))\}_{i=1}^{n+1}$ *is affinely independent for all* $t \in [0, T]$*, and define* $Z(t)$ *and* $V(t)$ *as in* (3.8) *and* (3.9)*, respectively. Then, there exists* $\tilde{T} \in \mathbb{R}_0^+$ *such that for all* $T \geq \tilde{T}$*, the origin of system* (2.3) *in closed-loop with controller* (3.3)-(3.11) *is uniformly asymptotically stable.*

Proof. The asymptotic stability of (2.3) and (3.4) are equivalent [20], and, therefore, the set $\mathcal{D}_{(z,v)}$ given by (3.5) and (3.6) also consists of asymptotically stable solutions, i.e., there exists $\beta \in \mathcal{KL}$ such that:

$$\|z^i(t)\| \leq \beta(\|z^i(0)\|, t), \quad \forall t \in \mathbb{R}_0^+, \tag{3.16}$$

for all $i \in \{1, ..., n+1\}$.

Consider the closed-loop system:

$$\dot{z} = Az + BV(t)Z^{-1}(0)z(pT).$$

By Lemma 3.2, we have that:

$$z((p+1)T) = Z(T)Z^{-1}(0)z(pT), \quad \forall T \in \mathbb{R}_0^+.$$

At the same time, by (3.14), we have $z(T) = \Psi(T)z(pT)$. This implies that:

$$\Psi(T) = Z(T)Z^{-1}(0). \tag{3.17}$$

We claim that, for any constants $a, b, c > 0$, there exists $t \in \mathbb{R}_0^+$ such that $\beta(r, t) < c$ for all $r \in [a, b]$. This claim will be shown using an argument similar to that of the proof of Lemma 16 in [21]. Using Lemma 4.3 from [22], there exist class $\mathcal{K}_\infty$ functions σ_1, σ_2 such that $\beta(r, t) \leq \sigma_1(\sigma_2(r)e^{-t})$ for all $r, t \in \mathbb{R}_0^+$. Let $0 < \varepsilon < c$. Define, for all $r \in \mathbb{R}_0^+$, $t(r)$ to be the solution of $\sigma_1(\sigma_2(r)e^{-t}) = c - \varepsilon$ and obtain:

$$t(r) = -\log \frac{\sigma_1^{-1}(c - \varepsilon)}{\sigma_2(r)}.$$

Since $t(r)$ is a continuous function and $[a, b]$ is compact, the extreme value theorem implies that $t^* = \max_{r \in [a,b]} t(r)$ is well-defined. For all $r \in [a, b]$, it is true that:

$$\beta(r, t^*) \leq \sigma_1(\sigma_2(r)e^{-t^*}) \leq c - \varepsilon < c.$$

Using the previous claim with $a = \min_{i \in \{1,...,n+1\}} \|z^i(0)\|$, $b = \max_{i \in \{1,...,n+1\}} \|z^i(0)\|$ and $c = 1/\left(2\sqrt{n}\left\|Z^{-1}(0)\right\|\right)$, we conclude the existence of $\tilde{T} \in \mathbb{R}$ for which the following inequality holds:

$$\beta(\|z^i(0)\|, T) < \frac{1}{2\sqrt{n}\|Z^{-1}(0)\|},$$

for all $i \in \{1, \ldots, n+1\}$ and for all $T \geq \tilde{T}$. Therefore, by (3.16), we have:

$$\|z^i(T)\| < \frac{1}{2\sqrt{n}\|Z^{-1}(0)\|}, \tag{3.18}$$

for all $i \in \{1, \ldots, n+1\}$ and for all $T \geq \tilde{T}$. Using (3.17), we have:

$$
\begin{aligned}
\|\Psi(T)\| &\leq \|Z(T)\| \left\|(Z(0))^{-1}\right\| \\
&\leq \|Z(T)\|_F \left\|(Z(0))^{-1}\right\| \\
&= \left(\sum_{i=2}^{n+1} \|z^i(T) - z^1(T)\|^2\right)^{\frac{1}{2}} \left\|(Z(0))^{-1}\right\| \\
&< \frac{\sqrt{n}}{\sqrt{n}\left\|(Z(0))^{-1}\right\|} \cdot \left\|(Z(0))^{-1}\right\| \\
&< 1,
\end{aligned}
\tag{3.19}
$$

for all $T \geq \tilde{T}$. The second to last inequality follows from (3.18) and the triangle inequality.

According to stability conditions for linear discrete-time systems, equation (3.19) implies that, for all $T > \tilde{T}$, the system (3.4) in closed loop with the controller (3.11) is uniformly exponentially stable. From [19], we know that uniform exponential stability of the sampled-data system (3.14) implies uniform exponential stability of the system (3.4)-(3.11) because the matrices $\Psi(t)$ are bounded for $t \in [0, T]$. Uniform asymptotic stability of the origin for the system (3.4)-(3.11) in the (z, v)-coordinates implies uniform asymptotic stability of the origin for the feedback equivalent system (2.3)-(3.3)-(3.11) in (x, u)-coordinates [20]. □

Remark 3.5. Theorem 3.4 shows the existence of $\tilde{T} \geq 0$ such that $\|\Psi(T)\| < 1$ for all $T \geq \tilde{T}$. In practice, a user can determine the upper bound on $\|\Psi(T)\|$ by finding $s_{\min}(Z(0))$, the smallest singular value of $Z(0)$, and $s_{\max}(Z(T))$, the largest singular value of $Z(T)$. From (3.17), it can be seen that $\|\Psi(T)\| \leq s_{\max}(Z(T))s_{\min}(Z(0))$. Thus, whenever the expert demonstrations are such that $s_{\max}(Z(T))s_{\min}(Z(0)) < 1$, we have a stable monodromy matrix $\Psi(T)$.

Remark 3.6. Up to this point, we strictly assumed that the expert controller k aims to stabilize the system at the origin and provided a guarantee that the learned controller $\hat{k}$ does the same. The aforementioned results easily extend to the case where the objective of the learned controller is to track a trajectory. The key idea is to recast the problem of trajectory tracking into that of stabilizing the error dynamics similar to what is done in Section 4.5 in [18]. We consider this generality of the learned controller to be a strength of this approach since a user cannot ask the expert to provide demonstrations of all the trajectories they might want to track in the future. We will illustrate this idea using a simulation in Section 5.

What Theorem 4.3 shows is that the proposed controller (4.5) only requires a finite set of finite-length demonstrations to drive the state to the origin. As long as, over the course of a demonstration, the expert leaves the system in a "better" state than the one they got it in, it is sufficient for the controller to use this demonstration.

4. Learning from more than $n+1$ expert demonstrations

In this section, we extended the results in the previous section to the case where more than $n+1$ trajectories are available. We will show that it is possible to select a subset of $n+1$ demonstrations that results in the best approximation of the unknown controller. Readers not interested in the results on approximation optimality can skip this section.

4.1. **Preliminaries.** We begin by reviewing several key concepts in multivariate linear interpolation. Let $\mathcal{X} = \{x_1, \ldots, x_k\}$ be a finite set of points in $\mathbb{R}^n$. An *n-simplex* S is the convex hull of a set $\mathcal{X}' = \{x'_1, \ldots, x'_{n+1}\}$ of $n+1$ affinely independent points. A *triangulation* of points in $\mathcal{X}$, denoted $\mathcal{T}(\mathcal{X})$, is a collection of n-simplices such that their vertices are points in $\mathcal{X}$, their interiors are disjoint, and their union is $\operatorname{conv} \mathcal{X}$. We denote the n-simplex in $\mathcal{T}(\mathcal{X})$ containing $x \in \operatorname{conv} \mathcal{X}$ by $S_{\mathcal{T}}(x)$ and define a vertex index set associated with x in $\mathcal{T}(\mathcal{X})$, denoted $\mathcal{I}_{\mathcal{T}}(x)$, as $S_{\mathcal{T}}(x) = \operatorname{conv} \mathcal{X}^{\mathcal{I}_{\mathcal{T}}(x)}$. The *Delauney triangulation* of a finite set of points P, denoted

$\mathcal{DT}(P)$, is the triangulation such that the circum-hypersphere of every n-simplex in the triangulation contains no point from the set in its interior. It is unique if no $n+1$ points are on the same hyperplane and no $n+2$ points are on the same hypersphere [23].

Let $\psi : \mathbb{R}^n \to \mathbb{R}^m$ be an unknown function. Given a finite set of points $\mathcal{X} = \{x_1, \dots, x_k\} \subset \mathbb{R}^n$ and a set of function values $\mathcal{Y} = \{y_1, \dots, y_k\} \triangleq \{\psi(x_1), \dots, \psi(x_k)\}$, an *interpolant* $\hat{\psi}_{\mathcal{X},\mathcal{Y}} : \operatorname{conv}\mathcal{X} \to \mathbb{R}^m$ is an approximation of ψ that satisfies $\hat{\psi}(x) = \psi(x)$ for all $x \in \mathcal{X}$. We define an interpolant $\hat{\psi}^{\mathcal{T}}_{\mathcal{X},\mathcal{Y}} : \operatorname{conv}\mathcal{X} \to \mathbb{R}^m$, called a *piecewise-linear interpolant* based on $\mathcal{T}(\mathcal{X})$, as:

$$\hat{\psi}^{\mathcal{T}}_{\mathcal{X},\mathcal{Y}}(x) = \sum_{i \in \mathcal{I}_{\mathcal{T}}(x)} \theta_i y_i, \tag{4.1}$$

where $\theta_i \geq 0$ satisfy:

$$x = \sum_{i \in \mathcal{I}_{\mathcal{T}}(x)} \theta_i x_i, \qquad \sum_{i \in \mathcal{I}_{\mathcal{T}}(x)} \theta_i = 1. \tag{4.2}$$

4.2. **Constructing the learned controller.** We now describe the construction of the controller $v = \hat{\kappa}(t, z)$ for $M \geq n+1$. Define $\mathcal{Z}(t) = \pi_1\left(\mathcal{D}_{(z,v)}(t)\right)$ and $\mathcal{V}(t) = \pi_2\left(\mathcal{D}_{(z,v)}(t)\right)$. Recall that we partition time into intervals of length T, indexed by $p \in \mathbb{N}_0$, and further partition those into ℓ equal intervals, indexed by $q \in \{0, \dots, \ell-1\}$. For each of intervals $[pT + q\Delta, pT + (q+1)\Delta)$, we propose using the piecewise-continuous control law

$$v(t) = \hat{\kappa}(t, z(pT + q\Delta)),$$

where the value of $\hat{\kappa}(\tau, \xi)$ defined as follows:

(i) For $\xi \in \operatorname{conv}\mathcal{Z}(q\Delta)$, the value of $\hat{\kappa}(\tau, \xi)$ is given by the value at z of a piecewise-linear interpolant $\hat{\psi}^{\mathcal{T}}_{\mathcal{Z}(q\Delta),\mathcal{V}(\tau - pT)}$. Since a piecewise-linear interpolant is determined by an associated triangulation $\mathcal{T}(\mathcal{Z}(q\Delta))$ [23], this implies that there is a family of possible learned controller we can construct from the given data $\mathcal{D}_{(z,v)}$. Moreover, the value of the interpolant depends only on the values of $\mathcal{Z}^{\mathcal{I}_{\mathcal{T}}(\xi)}(q\Delta)$ and $\mathcal{V}^{\mathcal{I}_{\mathcal{T}}(\xi)}(t - pT)$, where $\mathcal{I}_{\mathcal{T}}(\xi)$ is a vertex set associated with ξ in $T(\mathcal{Z}(q\Delta))$.

(ii) For $\xi \notin \mathcal{Z}(q\Delta)$, let ξ^* be the Euclidean projection of ξ onto $\operatorname{conv}\mathcal{Z}(q\Delta)$. Define the index set $\mathcal{I}_{\mathcal{T}}(\xi) = \mathcal{I}_{\mathcal{T}}(\xi^*)$ and express ξ as an affine combination $\xi = \sum_{i \in \mathcal{I}_{\mathcal{T}}(\xi)} \theta_i z^i(0)$. Then, the value of $\hat{\kappa}(\tau, \xi)$ is given by $\hat{\kappa}(\tau, \xi) = \sum_{i \in \mathcal{I}_{\mathcal{T}}(\xi)} \theta_i v_i(\tau - pT)$.

In both cases, the controller can be concisely expressed if, given a vertex index set $\mathcal{I}$ of cardinality $n+1$ for $\mathcal{Z}(t)$ and $\mathcal{V}(t)$, we construct the following matrices:

$$Z^{\mathcal{I}}(t) \triangleq \left[z^{i_2}(t) - z^{i_1}(t) \;\middle|\; \cdots \;\middle|\; z^{i_{n+1}}(t) - z^{i_1}(t)\right] \tag{4.3}$$

$$V^{\mathcal{I}}(t) \triangleq \left[v^{i_2}(t) - v^{i_1}(t) \;\middle|\; \cdots \;\middle|\; v^{i_{n+1}}(t) - v^{i_1}(t)\right] \tag{4.4}$$

for $t \in [0, T]$. Then, using (4.3) and (4.4), the proposed control law, for all $t \in [pT + q\Delta, pT + (q+1)\Delta)$, is:

$$\begin{aligned} v(t) &= \hat{\kappa}(t, z(pT + q\Delta)) \\ &= V^{\mathcal{I}_{\mathcal{T}}(z(pT+q\Delta))}(t - pT)\zeta(p, q), \end{aligned} \tag{4.5}$$

where $p = \lfloor t/T \rfloor$, $q = \lfloor (t - pT)/\Delta \rfloor$, and

$$\zeta(p, q) = \left(Z^{\mathcal{I}_{\mathcal{T}}(z(pT+q\Delta))}(q\Delta)\right)^{-1} z(pT + q\Delta).$$

Note that, in the absence of uncertainties and disturbances, by Lemma 3.2, the coefficients satisfy:

$$\begin{aligned} \zeta(p, q) &= \left(Z^{\mathcal{I}_{\mathcal{T}}(z(pT+q\Delta))}(q\Delta)\right)^{-1} z(pT + q\Delta) \\ &= \left(Z^{\mathcal{I}_{\mathcal{T}}(z(pT))}(0)\right)^{-1} z(pT), \end{aligned} \tag{4.6}$$

for all $q \in \{0, 1, \ldots, \ell - 1\}$. Therefore, for all $t \in [pT, (p+1)T)$, the controller (4.5) applies the input equal to that applied by the following controller:

$$v(t) = \hat{\kappa}(t, z(pT)) = V^{\mathcal{I}_{\mathcal{T}}(z(pT))}(t - pT)\zeta(p), \tag{4.7}$$

where $p = \lfloor t/T \rfloor$ and:

$$\zeta(p) = \left(Z^{\mathcal{I}_{\mathcal{T}}(z(pT))}(0)\right)^{-1} z(pT). \tag{4.8}$$

Incidentally, this corresponds to the value of the piecewise-linear interpolant $\hat{\psi}^{\mathcal{T}}_{\mathcal{Z}(0),\mathcal{V}(t-pT)}$ at $z(pT)$. In what follows, we assume that (4.6) holds.

Recall that the piecewise-linear interpolant depends on the choice of the triangulation $\mathcal{T}(\mathcal{Z}(0))$. Typically, there are several triangulations one can define given a set of points $\mathcal{Z}(0)$. We want to choose the triangulation that leads to the piecewise-linear interpolant that approximates the expert controller well at the points in $\operatorname{conv} \mathcal{Z}(0)$ distinct from $\mathcal{Z}(0)$. We have previously mentioned that the expert controller κ is a smooth function. Therefore, there exists some $H \in \mathbb{R}^+_0$ such that the function κ satisfies $\|\nabla^2 \kappa(z)\| \le H$ for all $z \in \operatorname{conv} \mathcal{Z}(0)$. We denote the class of functions with the Hessian norm smaller or equal to H by $\mathcal{F}(H)$. Formally, we seek to find the triangulation $\mathcal{T}(\mathcal{Z}(0))$ that leads to the piecewise-linear interpolant $\hat{\psi}^{\mathcal{T}}_{\mathcal{Z}(0),\mathcal{V}(t-pT)}$ that gives the best approximation of the expert controller $\kappa \in \mathcal{F}(H)$ even in the worst-case scenario, i.e., we seek the solution to the following problem:

$$\min_{\mathcal{T}(\mathcal{Z}(0))} \max_{\kappa \in \mathcal{F}(H)} \max_{x \in \operatorname{conv} \mathcal{X}} \|\kappa(z) - \hat{\psi}^{\mathcal{T}}_{\mathcal{Z}(0),\mathcal{V}(t-pT)}(z)\|.$$

We can view this as a game where we pick $\mathcal{T}(\mathcal{Z}(0))$, and the adversary, after seeing our choice of $\mathcal{T}(\mathcal{Z}(0))$, picks (κ, x) to maximize the cost. The following lemma by Omohundro [24] shows that the Delauney triangulation leads to the best worst-case piecewise-linear interpolation for the class of functions with the bounded Hessian norm. For an efficient implementation of piecewise-linear interpolation based on Delauney triangulation, we refer the reader to [23].

Lemma 4.1 ([24]). *Let $\psi : \mathbb{R}^n \to \mathbb{R}^m$ be a function satisfying the bounded Hessian norm property, i.e., $\psi \in \mathcal{F}(H)$. Given a set of points $\mathcal{X} = \{x_1, \ldots, x_k\} \subset \mathbb{R}^n$ and a set of function values $\mathcal{Y} = \{y_1, \ldots, y_k\} \subset \mathbb{R}^m$, the piecewise-linear interpolant with the smallest maximum approximation error is based on the Delauney triangulation, i.e.:*

$$\mathcal{DT}(\mathcal{X}) = arg \min_{\mathcal{T}(\mathcal{X})} \max_{\psi \in \mathcal{F}} \max_{x \in \operatorname{conv} \mathcal{X}} \|\psi(x) - \hat{\psi}^{\mathcal{T}}_{\mathcal{X},\mathcal{Y}}(x)\|.$$

Remark 4.2. While we justify the construction of the controller for $z \in \operatorname{conv} \mathcal{Z}(0)$ with optimality in terms of approximation error, we cannot provide a similar justification for $z \notin \operatorname{conv} \mathcal{Z}(0)$. Therefore, we suggest collecting the expert demonstrations in such a way that the normal region of operation belongs to the convex hull of the demonstrations.

Let us define the collection of index sets $\mathcal{P} = \{\mathcal{I}_1, \ldots, \mathcal{I}_P\}$, where each $\mathcal{I}_j$ selects the vertices of an n-simplex in the triangulation $\mathcal{T}(\mathcal{Z}(0))$ and $P = |\mathcal{T}(\mathcal{Z}(0))|$. Note that $\mathcal{P}$ is a finite set because there are only finitely many n-simplices in $\mathcal{T}(\mathcal{Z}(0))$.

Suppose that the vertex index set associated with $z(pT)$ in $\mathcal{P}$ is $\mathcal{I}_{\mathcal{DT}}(z(pT)) = \mathcal{I}_j$ for some $j \in \{1, \ldots, P\}$. Assuming (4.6) holds, the system (3.4) in closed loop with (4.5) is given by:

$$\dot{z} = Az + BV^{\mathcal{I}_j}(t - pT)\left(Z^{\mathcal{I}_j}(0)\right)^{-1} z(pT), \tag{4.9}$$

for all $t \in [pT, (p+1)T)$. Integrating the dynamics shows that the sequence $\{z(pT)\}_{p \in \mathbb{N}_0}$ satisfies:

$$z((p+1)T) = \Psi_j(T) z(pT), \tag{4.10}$$

where

$$\Psi_j(T) \triangleq e^{AT} + \int_0^T e^{A(T-\tau)} B V^{\mathcal{I}_j}(\tau) \left(Z^{\mathcal{I}_j}(0)\right)^{-1} \mathrm{d}\tau. \tag{4.11}$$

Note that now, instead of a single monodromy matrix, we have a set of monodromy matrices $\{\Psi_j\}_{j=1}^P$.

The following result is a version of Theorem 3.4 for $M \geq n+1$ demonstrations.

Theorem 4.3. *Consider the feedback linearizable system* (2.3) *under the transformation* (3.1) *and the feedback law* (3.3)*. Suppose we are given a finite set of demonstrations $\mathcal{D} = \{(x^i, u^i)\}_{i=1}^M$ generated by the system* (2.3) *in closed loop with an asymptotically stabilizing controller $k : \mathbb{R}^n \to \mathbb{R}$. Assume that $\{\Phi(x^i(t))\}_{i=1}^M$ is affinely independent for all $t \in [0, T]$, and define $Z^{\mathcal{I}}(t)$ and $V^{\mathcal{I}}(t)$ as in* (4.3) *and* (4.4)*, respectively. Then, there exists $\tilde{T} \in \mathbb{R}_0^+$ such that for all $T \geq \tilde{T}$, the origin of system* (2.3) *in closed-loop with controller* (3.3)-(4.5) *is uniformly asymptotically stable.*

Proof. The arguments in the proof of Theorem 3.4 show the existence of $\tilde{T}_j \in \mathbb{R}$ such that $\|\Psi_j(t)\| < 1$ for all $t \geq \tilde{T}_j$. We choose $\tilde{T} = \max_{j \in \{1,\ldots,P\}} T_j$. The system (3.4) in closed loop with the controller (4.5) can be represented as a switched system $z((p+1)T) = \Psi_{j(p)}(T) z(pT)$, where $j(p) \in \{1, \ldots, P\}$ is some switching sequence. By Theorem 3 in [25], the fact that $\|\Psi_j(T)\| < 1$ for all $T \geq \tilde{T}$ and for all $j \in \{1, \ldots, P\}$ implies that, for any switching signal $j(p)$, the system $z((p+1)T) = \Psi_{j(p)}(T) z(pT)$ is uniformly exponentially stable. Since the matrices $\Psi_j(t)$ are bounded for $t \in [0, T]$, the system (3.4) in closed loop with the controller (4.5) is uniformly exponentially stable. Uniform asymptotic stability of the origin for the system (3.4)-(3.11) in the (z, v)-coordinates implies uniform asymptotic stability of the origin for the feedback equivalent system (2.3)-(3.3)-(3.11) in (x, u)-coordinates [20]. □

5. Simulation example

We illustrate the performance of our proposed controller (3.10) on a quadrotor described by the nonlinear model:

$$\ddot{p} = \frac{1}{m}\left(\tau R e_3 - [\omega]_\times J \omega\right), \tag{5.1}$$

$$\dot{R} = R[\omega]_\times, \tag{5.2}$$

$$\dot{\omega} = J^{-1}(\eta - [\omega]_\times J\omega), \tag{5.3}$$

where: $p \in \mathbb{R}^3$, $R \in SO(3)$, $\omega \in \mathbb{R}^3$ are the position, orientation, and angular velocity of the quadrotor, respectively; $\tau \in \mathbb{R}$ and $\eta \in \mathbb{R}^3$ are thrust and torque inputs, respectively; $m \in \mathbb{R}$, and $J \in \mathbb{R}^{3\times3}$ are the mass and the inertia matrix of the quadrotor; and $[\,\cdot\,]_\times$ is the skew-symmetric matrix form of the vector cross product.

We split the dynamics (5.1)-(5.3) into two subsystems: one described by (5.1)-(5.2) with the state $x = (p, \dot{p}, R, \tau)$ and the virtual inputs $u = (\dot{\tau}, \omega)$, and the other described by (5.3) with the state $x' = \omega$ and the virtual inputs $u' = \eta$. Typically, quadrotors have high-frequency internal controllers, which we model as linear in simulation, that track the desired angular velocity based on state feedback and, therefore, it is reasonable to assume that we can directly control the angular velocity [27].

It is known that the dynamics (5.1)-(5.3) are differentially flat with respect to position and yaw angle [26]. In what follows, we focus on controlling the position p, whereas the yaw angle is controlled to remain constant. Differential flatness allows us to transform the dynamics (5.1)-(5.2) into linear dynamics $\dot{z}_1 = z_2,\ \dot{z}_2 = z_3,\ \dot{z}_3 = v$ via a coordinate transformation:

$$z = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T \triangleq \begin{bmatrix} p & \dot{p} & \ddot{p} \end{bmatrix}^T, \tag{5.4}$$

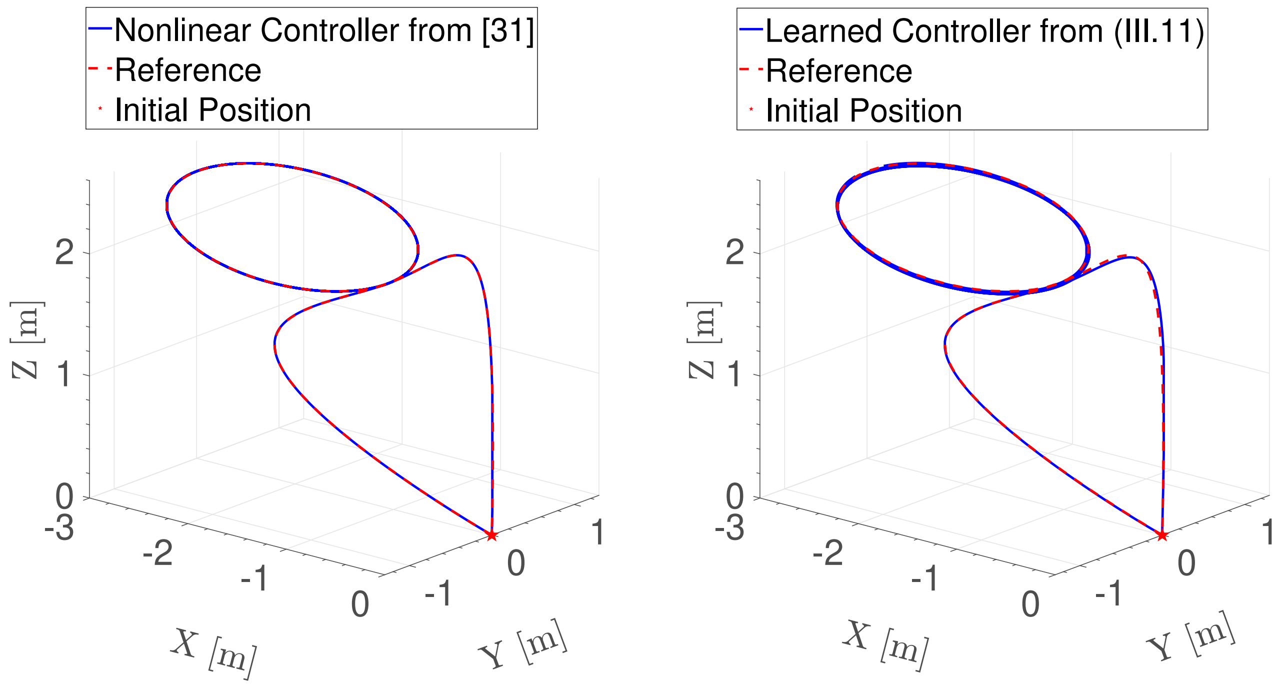

FIGURE 1. Trajectory tracking of the nonlinear controller from [26] (left) and the learned controller from (3.11) (right).

and the feedback law:

$$v = \frac{1}{m}(\dot{\tau} R e_3 - \tau R \omega_1 e_2 + \tau R \omega_2 e_1) \triangleq b(z)u. \tag{5.5}$$

We apply the controller design from [26] to the dynamics (5.1)-(5.3) in simulation and use the resulting solutions as the expert demonstrations. The expert is commanded to stabilize the quadrotor at the origin, starting from various positions, velocities and accelerations. Given the dimension of the state z equals to 9, we record 10 pairs of expert demonstrations $\{(z^i, v^i)\}_{i=1}^{10}$ from simulations, including the pair corresponding to the trivial solution $(z^1, v^1) \equiv (0, 0)$[1]. Please note that the pairs (z^i, v^i) in this context are merely evolutions of position, velocity, acceleration, and jerk. The recorded data is studied to ensure that the sufficient conditions of Theorem 3.4 are satisfied, i.e., the matrix $Z(t)$ in (3.8) is always invertible and $\|Z(T)Z^{-1}(0)\| < 1$, and a well-behaved fragment of length $T = 2.4$ s is subsequently used to construct a stabilizing controller (3.11).

As a simulation benchmark, we use the learned controller (3.11) to track the reference trajectory depicted on Figure 1, which consists of four parts: a polynomial minimum snap trajectory from $(0, 0, 0)$ to $(-1.0607, 0, 1.8371)$ from $t = 0$ s to $t = 5$ s; twelve revolutions around a slanted circle given by:

$$p_R(t) = \left(a_1 \cos\frac{2\pi t}{5} + a_2, \sin\frac{2\pi t}{5}, a_3 \cos\frac{2\pi t}{5} + a_4\right),$$

where $a_1 = 0.9659$, $a_2 = -2.1560$, $a_3 = -0.2588$, $a_4 = 2.0959$, from $t = 5$ s to $t = 65$ s; a polynomial minimum snap trajectory from $(-1.0607, 0, 1.8371)$ to $(0, 0, 0)$ from $t = 65$ s to $t = 70$ s; and, finally, a setpoint at the origin after $t \geq 70$ s. To control the quadrotor, we use the learned controller $\hat{\kappa}$ from (3.11) to control the tracking error with $v(t) = \hat{\kappa}(t, z(t) - z_R(t))$, where $z_R = (p_R, \dot{p}_R, \ddot{p}_R)$, together with the feedback law:

$$u(t) = (\dddot{p}_R(t) + v(t))/b(z(t)). \tag{5.6}$$

[1]Please refer to `https://github.com/cyphylab/watch-and-learn-LfD-controller` for the recorded expert demonstrations.

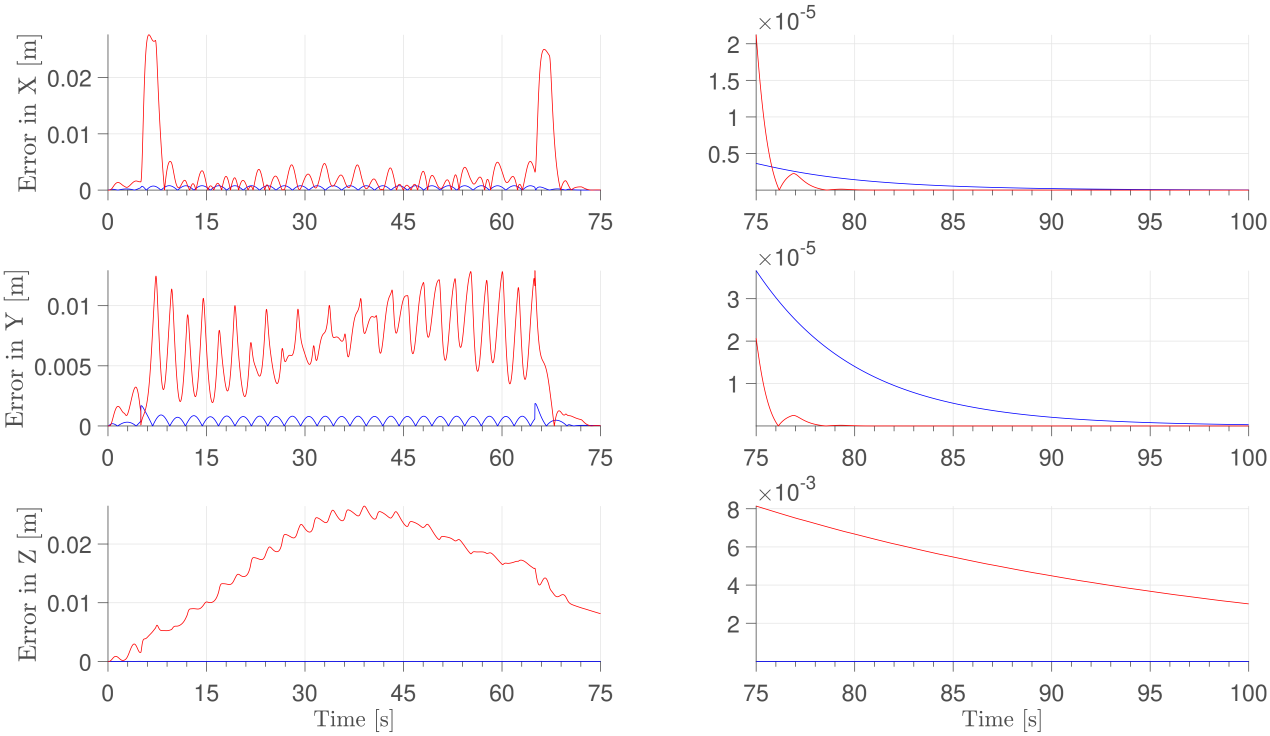

FIGURE 2. Comparison of tracking errors in X, Y and Z coordinates of learned controller from (3.11) (red) and nonlinear controller from [26] (blue).

In Figure 1, we depict the quadrotor trajectories for both the nonlinear controller in [26] and the learned controller (3.11) tracking the aforementioned trajectory. In Figure 2 we compare the tracking errors of the learned controller with those of the nonlinear controller from [26]. The learned controller appears to track the trajectory well (the error is of the order of centimeters), but not as well as the nonlinear controller does. Looking at the plots more closely, we observe that the learned controller is better at tracking minimum snap trajectories than at tracking the circle, which is also the case for the expert controller [26], indicating that the two are indeed related. Given the theoretical results in the previous sections, one would expect the tracking error to exponentially decrease to zero. This does not happen here, however, because recall that we give the desired angular velocity as reference to the internal controller, which does not track it perfectly. This is also evident from the fact that when the system stabilizes at the origin in the end, it exhibits exponential stability. It might be interesting to study how the learned controller behaves in cascade with other controllers as part of the future work. We also want to emphasize that the learned controller (3.11) can be used for more complex tasks, and we chose tracking this particular reference trajectory merely as an example.

6. CONCLUSION

In this work, we have presented a methodology for constructing a stabilizing controller from expert demonstrations. Compared to machine-learning approaches, this methodology requires few demonstrations (i.e., the minimal number of demonstrations is $n + 1$) and provides formal stability guarantees. As part of our future work, we intend to examine if the same methodology can be applied when system controlled by the expert is unknown. This will be an important extension to this work because, typically, for the tasks where learning from demonstrations is required, it is rarely the case that the underlying dynamical system is completely known.

References

[1] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233 – 242, 1999.

[2] S. Chernova and A. L. Thomaz, *Robot Learning from Human Teachers*. Morgan & Claypool Publishers, 2014.

[3] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent Advances in Robot Learning from Demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 297–330, 2020.

[4] O. Kroemer, S. Niekum, and G. D. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," 2019. [Online]. Available: http://arxiv.org/abs/1907.03146

[5] D. A. Pomerleau, *ALVINN: An Autonomous Land Vehicle in a Neural Network*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, p. 305–313.

[6] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4693–4700.

[7] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.

[8] S. Levine and V. Koltun, "Learning complex neural network policies with trajectory optimization," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, p. II–829–II–837.

[9] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating explicit model predictive control using constrained neural networks," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 1520–1527.

[10] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Deep drone acrobatics," *CoRR*, vol. abs/2006.05768, 2020. [Online]. Available: https://arxiv.org/abs/2006.05768

[11] M. Palan, S. Barratt, A. McCauley, D. Sadigh, V. Sindhwani, and S. Boyd, "Fitting a Linear Control Policy to Demonstrations with a Kalman Constraint," *arXiv e-prints*, p. arXiv:2001.07572, Jan. 2020.

[12] R. Kálmán, "When is a linear control system optimal," *Journal of Basic Engineering*, vol. 86, pp. 51–60, 1964.

[13] J. C. Willems, "From time series to linear system—part i. finite dimensional linear time invariant systems," *Automatica*, vol. 22, no. 5, pp. 561 – 580, 1986.

[14] I. Markovsky, J. C. Willems, S. V. Huffel, and B. D. Moor, *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach (Mathematical Modeling and Computation) (Mathematical Modeling and Computation)*. USA: Society for Industrial and Applied Mathematics, 2006.

[15] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization*. USA: Cambridge University Press, 2004.

[17] D. Nesic and A. R. Teel, "A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models," *IEEE Transactions on Automatic Control*, vol. 49, no. 7, pp. 1103–1122, 2004.

[18] A. Isidori, *Nonlinear Control Systems*, ser. Communications and Control Engineering. Springer-Verlag London, 1995.

[19] P. T. Kabamba, "Control of Linear Systems Using Generalized Sampled-Data Hold Functions," *IEEE Transactions on Automatic Control*, vol. 32, no. 9, pp. 772–783, 1987.

[20] A. V. Kavinov and A. P. Krischenko, "Stability of solutions in different variables," *Differential Equations*, vol. 43, pp. 1505–1509, Nov. 2007.

[21] R. Geiselhart, R. H. Gielen, M. Lazar, and F. R. Wirth, "An alternative converse lyapunov theorem for discrete-time systems," *Systems & Control Letters*, vol. 70, pp. 49 – 59, 2014.

[22] A. R. Teel and L. Praly, "A smooth Lyapunov function from a class-KL estimate involving two positive semidefinite functions," *ESAIM - Control Optimization and Calculus of Variations*, p. 313–367, 2000.

[23] T. H. Chang, L. T. Watson, T. C. H. Lux, B. Li, L. Xu, A. R. Butt, K. W. Cameron, and Y. Hong, "A polynomial time algorithm for multivariate interpolation in arbitrary dimension via the delaunay triangulation," in *Proceedings of the ACMSE 2018 Conference*, ser. ACMSE '18, New York, NY, USA, 2018.

[24] S. M. Omohundro, "The Delaunay Triangulation and Function Learning," International Computer Science Institute, Tech. Rep., 1989.

[25] Guisheng Zhai, Bo Hu, K. Yasuda, and A. N. Michel, "Qualitative analysis of discrete-time switched systems," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 3, 2002, pp. 1880–1885 vol.3.

[26] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2520–2525.

[27] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1485–1491, 2011, 18th IFAC World Congress.