2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Watch and Learn: Learning to control feedback linearizable systems from expert demonstrations*

Alimzhan Sultangazin[1], Luigi Pannocchi[1], Lucas Fraile[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. The results are experimentally demonstrated on a CrazyFlie 2.0 quadrotor.**

I. Introduction

A. 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) [3] reflects the need for a well-defined controller design methodology for such tasks. In this work, we propose a methodology that uses expert demonstrations to construct a stabilizing controller.

In what follows, 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 performance is mainly illustrated with experimental results.

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 can 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 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.

B. Contributions

In this work, we propose a methodology for constructing a controller for a known nonlinear system from a finite number of expert demonstrations of desired behaviour, provided the number of demonstrations is greater than the number of states

*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]A. Sultangazin, L. Pannocchi, L. Fraile, and P. Tabuada are with Department of Electrical and Computer Engineering, University of California - Los Angeles, USA {asultangazin, lpannocchi, lfrailev, tabuada}@ucla.edu

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

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9812054

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 demonstrate the feasibility of this approach by applying it to the problem of quadrotor control. 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.

Remark I.1. Please note that while it is possible to design a controller that stabilizes a feedback linearizable system without expert demonstrations, we want to emphasize that our goal is to stabilize the system and, at the same time, imitate the behaviour of the expert controller.

II. Problem Statement and Preliminaries

A. 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:

$$\dot{x} = f(t, x), \tag{1}$$

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 (1) is *uniformly asymptotically stable* if there exists $\beta \in \mathcal{KL}$ such that the following is satisfied:

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0), \quad \forall t \geq t_0 \geq 0. \tag{2}$$

Let $\mathcal{X} = \{x_1, \dots, x_k\}$ 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 $\mathcal{X}$.

B. Problem Statement

Consider a known continuous-time control-affine system:

$$\Sigma : \quad \dot{x} = f(x) + g(x)u, \tag{3}$$

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 (3). We call a pair $(x, u) : \mathbb{R}_0^+ \to \mathbb{R}^n \times \mathbb{R}^m$ a solution of the system (3) if, for all $t \in \mathbb{R}_0^+$, the equation (3) is satisfied. Furthermore, we refer to the functions x and u as a trajectory and a control input of the system (3).

Definition II.1. A controller $u = k(x)$ is *asymptotically stabilizing* for system (3) if the origin is uniformly asymptotically stable for the system (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 (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 (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 II.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 [15].

We make the assumption that the system (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 (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 [16, Ch. 4] for a complete introduction to feedback linearization). In the single-input case, the system (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 $i = 0, \dots, n-2$ 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 $h(0) = 0$.

III. 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 $M = n + 1$ demonstrations and present some of the main results. Due to the space limit, we will not consider the case when $M \geq n + 1$ in this paper, but the interested reader can refer to Section 4 of [17] for a detailed discussion of that case.

A. Feedback linearization

Recall that using the feedback linearizability assumption, we can rewrite the nonlinear system dynamics (3) in the coordinates:

$$z = \Phi(x) = \begin{bmatrix} h(x) & L_f h(x) & \cdots & L_f^{n-1} h(x) \end{bmatrix}^T, \quad (4)$$

resulting in:

$$\begin{aligned} \dot{z}_1 &= z_2, \\ &\vdots \\ \dot{z}_{n-1} &= z_n, \\ \dot{z}_n &= a(z) + b(z)u, \end{aligned} \quad (5)$$

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:

$$u = b(z)^{-1}(-a(z) + v), \quad (6)$$

further transforms the system (3) into a linear time-invariant (LTI) controllable system:

$$\dot{z} = Az + Bv, \quad (7)$$

where (A, B) is a Brunovsky pair.

Remark III.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.

B. Expert demonstrations

Recall that the set of demonstrations $\mathcal{D}$ satisfies the nonlinear system dynamics (3). Using (4) and (6), we can represent the demonstrations $\mathcal{D}$ in (z, v)-coordinates. We denote the resulting set by $\mathcal{D}_{(z,v)} = \{(z^i, v^i)\}_{i=1}^{n+1}$, where the functions $z^i : [0, T] \to \mathbb{R}^n$ and $v^i : [0, T] \to \mathbb{R}$ are:

$$z^i(t) \triangleq \Phi(x^i(t)) \quad (8)$$

$$v^i(t) \triangleq L_f^n h(x^i(t)) + L_g L_f^{n-1} h(x^i(t)) u^i(t), \quad (9)$$

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

$$\mathcal{D}_{(z,v)}(t) = \{(z^i(t), v^i(t))\}_{i=1}^{n+1}. \quad (10)$$

It can be easily verified that the demonstrations in $\mathcal{D}_{(z,v)}$ satisfy the dynamics (7) and $v^i(t) = \kappa(z^i(t))$.

C. Constructing the learned controller

We denote by $v = \hat{\kappa}(t, z)$ the controller learned from the expert demonstrations. 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:

$$Z(t) \triangleq \left[z^2(t) - z^1(t) \;\middle|\; \cdots \;\middle|\; z^{n+1}(t) - z^1(t)\right] \quad (11)$$

$$V(t) \triangleq \left[v^2(t) - v^1(t) \;\middle|\; \cdots \;\middle|\; v^{n+1}(t) - v^1(t)\right], \quad (12)$$

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:

$$\hat{\kappa}(t, z(pT)) = V(t - pT)\zeta(p), \quad (13)$$

where $\zeta(p) = Z^{-1}(0)z(pT)$, and $Z(t), V(t)$ are defined in (11) and (12), respectively.

The next lemma formally shows that an affine combination of trajectories of (7) is a valid trajectory for (7).

Lemma III.2. *Suppose we are given a set of finite-length solutions $\{(z^i, v^i)\}_{i=1}^{n+1}$ of the system (7), 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 (7) 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 (11) and (12), respectively.

Proof. This lemma can be verified by substitution. □

Remark III.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 ask the expert to provide additional demonstrations until there is a subset of $n+1$ demonstrations that is affinely independent.

We note, however, that the control law (13) 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 (13) 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:

$$v(t) = \hat{\kappa}(t, z(pT + q\Delta)) = V(t - pT)\zeta(p, q), \quad (14)$$

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 III.2, the coefficients ζ satisfy:

$$\zeta(p, q) = Z^{-1}(q\Delta)z(pT + q\Delta) = Z^{-1}(0)z(pT), \quad (15)$$

for all $q \in \{0, 1, \ldots, \ell - 1\}$ (i.e., the controller (14) applies the input equal to that applied by the controller (13)). In practice, however, the systems are often subject to uncertainties and disturbances and, therefore, using the controller (14) instead of (13) significantly improves robustness in realistic scenarios. The interested reader can check this by comparing the disturbance-to-state L2-gains of the system (7) when using (13) and when using (14).

D. Stability of the learned controller

Assuming (15) holds, the system (7) in closed loop with (14) has the following form:

$$\dot{z} = Az + BV(t - pT)Z^{-1}(0)z(pT), \tag{16}$$

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

$$z((p+1)T) = \Psi(T)z(pT), \tag{17}$$

where

$$\Psi(T) \triangleq e^{AT} + \int_0^T e^{A(T-\tau)}BV(\tau)Z^{-1}(0)\mathrm{d}\tau. \tag{18}$$

By adopting a term from Floquet's theory, we refer to $\Psi(T)$ in (18) as the closed-loop monodromy matrix [18].

The main result of this section presents sufficient conditions for asymptotic stability of the system (3) in closed loop with (6)-(14).

Theorem III.4. *Consider the feedback linearizable system* (3) *under the transformation* (4) *and the feedback law* (6)*. Suppose we are given a finite set of solutions* $\mathcal{D} = \{(x^i, u^i)\}_{i=1}^{n+1}$ *generated by the system* (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* (11) *and* (12)*, respectively. Then, there exists* $\tilde{T} \in \mathbb{R}_0^+$ *such that for all* $T \geq \tilde{T}$*, the origin of system* (3) *in closed-loop with controller* (6)-(14) *is uniformly asymptotically stable.*

Proof. The asymptotic stability of (3) and (7) are equivalent [19], and, therefore, the set $\mathcal{D}_{(z,v)}$ given by (8) and (9) also consists of asymptotically stable solutions, i.e., there exists $\beta \in \mathcal{KL}$ such that:

$$\|z^i(t)\| \leq \beta(\|z^i(0)\|, t), \quad \forall t \in \mathbb{R}_0^+, \tag{19}$$

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

Consider the closed-loop system:

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

By Lemma III.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 (17), we have $z(T) = \Psi(T)z(pT)$. This implies that:

$$\Psi(T) = Z(T)Z^{-1}(0). \tag{20}$$

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 [20]. Using Lemma 4.3 from [21], 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 (19), we have:

$$\|z^i(T)\| < \frac{1}{2\sqrt{n}\|Z^{-1}(0)\|}, \tag{21}$$

for all $i \in \{1, \ldots, n+1\}$ and for all $T \geq \tilde{T}$. Using (20), we have:

$$\begin{aligned}\|\Psi(T)\| &\leq \|Z(T)\| \left\|(Z(0))^{-1}\right\| \leq \|Z(T)\|_F \left\|(Z(0))^{-1}\right\| \\ &= \left(\sum_{i=2}^{n+1} \|z^i(T) - z^1(T)\|^2\right)^{\frac{1}{2}} \left\|(Z(0))^{-1}\right\| \\ &< \frac{\sqrt{n}}{\sqrt{n}\left\|(Z(0))^{-1}\right\|} \cdot \left\|(Z(0))^{-1}\right\| < 1,\end{aligned} \tag{22}$$

for all $T \geq \tilde{T}$. The second to last inequality follows from (21) and the triangle inequality.

According to stability conditions for linear discrete-time systems, equation (22) implies that, for all $T > \tilde{T}$, the system (7) in closed loop with the controller (14) is uniformly exponentially stable. From [18], we know that uniform exponential stability of the sampled-data system (17) implies uniform exponential stability of the system (7)-(14) because the matrices $\Psi(t)$ are bounded for $t \in [0, T]$. Uniform asymptotic stability of the origin for the system (7)-(14) in the (z, v)-coordinates implies uniform asymptotic stability of the origin for the feedback equivalent system (3)-(6)-(14) in (x, u)-coordinates [19]. □

Remark III.5. Theorem III.4 shows the existence of $\tilde{T} \in \mathbb{R}^+$ such that $\|\Psi(T)\| < 1$ for all $T \geq \tilde{T}$. In practice, a user can determine $T \in \mathbb{R}^+$ satisfying this condition by directly computing $\|\Psi(t)\| = \|Z(t)Z^{-1}(0)\|$ for various $t \in \mathbb{R}_0^+$.

Remark III.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 [16]. 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. We will experimentally illustrate this in Section IV.

IV. Experiments

We illustrate the performance of our framework using the example of quadrotor dynamics:

$$\ddot{p} = \frac{1}{m}\left(\tau R e_3 - [\omega]_\times J\omega\right), \tag{23}$$

$$\dot{R} = R[\omega]_\times, \tag{24}$$

$$\dot{\omega} = J^{-1}(\eta - [\omega]_\times J\omega), \tag{25}$$

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 $J \in \mathbb{R}^{3\times 3}$ are the mass and the inertia matrix; and $[\,\cdot\,]_\times$ denotes the matrix form of the vector cross product.

We split the dynamics (23)-(25) into two subsystems: one described by (23)-(24) with the state $x = (p, \dot{p}, R, \tau)$ and the virtual inputs $u = (\dot{\tau}, \omega)$, and the other described by (25) with the state $x' = \omega$ and the virtual inputs $u' = \eta$. Typically, quadrotors have high-frequency internal controllers 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 [22].

It is known that the dynamics (23)-(25) are differentially flat with respect to position and yaw angle [23]. 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 (23)-(24) into linear dynamics $\dot{z}_1 = z_2,\ \dot{z}_2 = z_3,\ \dot{z}_3 = v$ via a coordinate transformation:

$$z = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T \triangleq \begin{bmatrix} p & \dot{p} & \ddot{p} \end{bmatrix}^T, \tag{26}$$

and the feedback law:

$$v = \frac{1}{m}(\dot{\tau} R e_3 - \tau R\omega_1 e_2 + \tau R\omega_2 e_1) \triangleq b(z)u. \tag{27}$$

We apply the controller design from [24] to the dynamics (23)-(25) in simulation[1] and use the resulting solutions as the expert demonstrations. The controller parameters are chosen as follows: $K_P = \mathrm{diag}(7.0, 7.0, 16.5)$, $K_I = \mathrm{diag}(0, 0, 15.5)$, $K_D = \mathrm{diag}(5.0, 5.0, 3.4)$, $K_{rp} = 6.0$, $K_y = 2.0$. 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 $(z^1, v^1) \equiv (0, 0)$. 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 III.4 are satisfied, i.e., the matrix $Z(t)$ in (11) is always invertible and $\|Z(T)Z^{-1}(0)\| < 1$, and a fragment of length $T = 2$ s is used to construct a stabilizing controller (14).

Next, we compare the learned controller (14) and the expert controller from [24] by using them to control a BitCraze CrazyFlie 2.0 quadrotor. In these experiments, the control inputs (τ, ω) are supplied by a computer via a USB radio at the average rate of 300 Hz. The internal PD controller of the CrazyFlie tracks (τ, ω) by controlling angular speeds of individual rotors. For state estimation, we use a Kalman filter that gets the position and attitude measurements from the OptiTrack motion capture system.

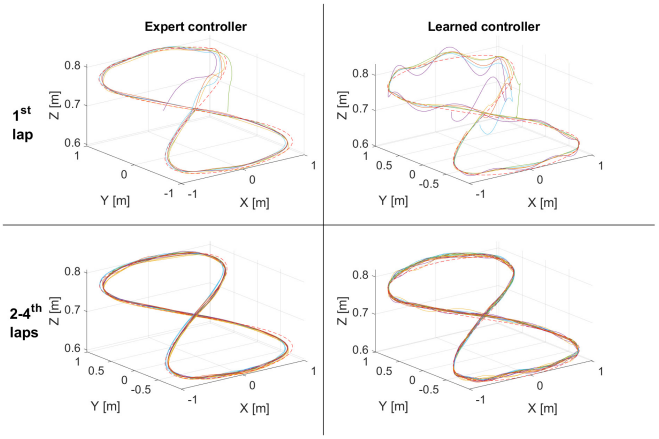

Fig. 1. Trajectory tracking of the nonlinear controller from [24] (left column) and the learned controller from (14) (right column) under five different initial conditions. Each experiment is plotted with a different color. The trajectories in the first lap (top row) are plotted separately from the trajectories in the subsequent laps (bottom row).

The experimental benchmark[2] we choose to compare the controllers is to track the reference depicted on Figure 1, which consists of two parts: a figure of eight given by:

$$p_R(t) = \left(\sin\frac{4\pi t}{T}, \sin\frac{2\pi t}{T}, 0.1\sin\frac{2\pi t}{5} + 0.7\right),$$

where $T = 10$ s, from $t = 0$ s to $t = 40$ s; and a setpoint at the origin after $t \geq 40$ s. We use the learned controller $\hat{\kappa}$ from (14) 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:

$$u(t) = (\dddot{p}_R(t) + v(t))/b(z(t)). \tag{28}$$

For each controller, we perform five experiments — each with a different initial position[3].

In Figure 1, we depict the quadrotor trajectories for both the nonlinear controller in [24] and the learned controller (14) tracking the aforementioned trajectory. We plot the position trajectories in the first lap separately from those in the subsequent laps to decouple the transient behaviour of a controller from the steady-state behaviour. In Figure 2 we compare the tracking errors of the learned controller with those of the nonlinear controller from [24] for all five experiments. The learned controller appears to track the trajectory well — the error is of the order of centimeters. It can be seen qualitatively, however, from Figure 1 that,

[1]The only difference of the expert controller used in this work and that used in [24] is that here the low-level controller is a linear PD controller.

[2]Code used in the experiments can be found at `https://github.com/cyphylab/cyphy_testbed/tree/LFD`.

[3]The initial positions used are $(0, 0, 0.7)$, $(0.3, 0.3, 0.7)$, $(0.3, -0.3, 0.7)$, $(-0.3, 0.3, 0.7)$, $(-0.3, -0.3, 0.7)$.

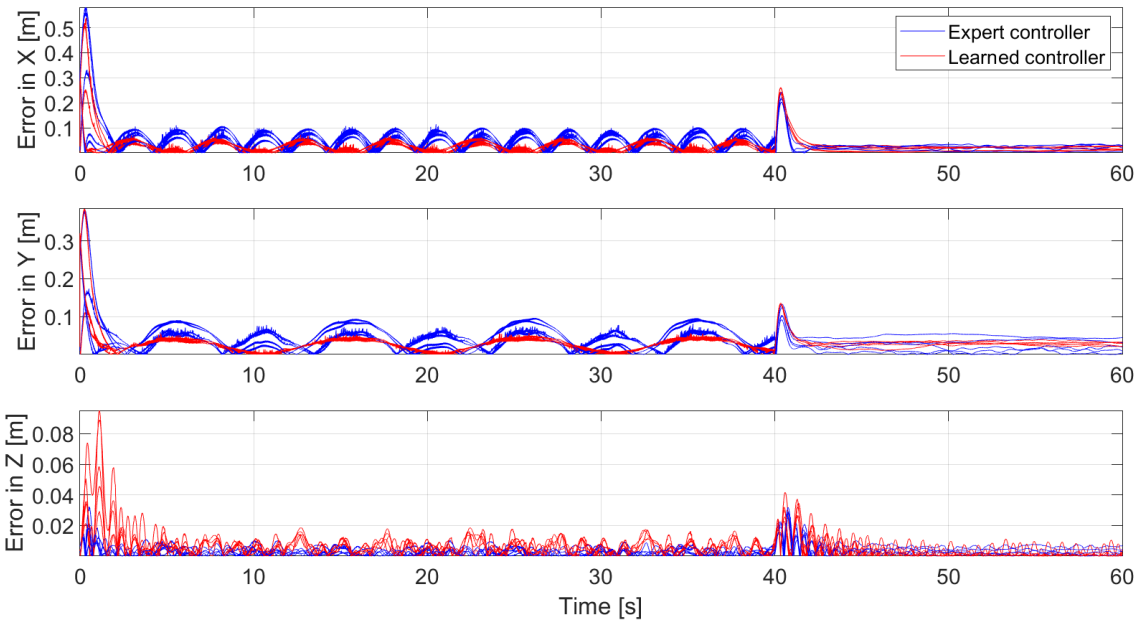

Fig. 2. Comparison of tracking errors in X, Y and Z coordinates of learned controller from (14) (red) and nonlinear controller from [24] (blue) for five different initial conditions.

in comparison to the expert controller, the learned controller takes a longer time to settle — this is especially noticeable in the experiments where the initial position of the quadrotor does not match that of the reference. From Figure 2, we observe that the errors of the learned controller and the expert controller are comparable, with the errors of the learned controller being slightly smaller in X and Y coordinates, whereas being slightly larger in Z coordinates. For $t \geq 40$ s, the error in position does not tend to zero for neither of the controllers, which appears to contradict the theoretical results. We attribute this to the several milliseconds of delay with which the control input is sent to the quadrotor[4].

V. 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 fewer demonstrations (i.e., the minimal number of demonstrations is $n + 1$) and provides formal stability guarantees. As part of future work, we intend to examine if the same methodology can be applied when the 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. In addition, it would be interesting to consider how this methodology changes if a different method for system linearization is used.

[4]Even in the ideal conditions of a simulation, an introduction of such a delay into the control loop has resulted in the trajectory stabilizing at a non-zero steady-state error.

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," *arXiv e-prints*, 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] 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.
[16] A. Isidori, *Nonlinear Control Systems*, ser. Communications and Control Engineering. Springer-Verlag London, 1995.
[17] A. Sultangazin, L. Fraile, L. Pannocchi, and P. Tabuada, "Watch and Learn: Learning to control feedback linearizable systems from expert demonstrations," UCLA, Tech. Rep., Mar 2021. [Online]. Available: http://www.cyphylab.ee.ucla.edu/Home/publications/UCLA-CyPhyLab-2021-03.pdf
[18] 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.
[19] A. V. Kavinov and A. P. Krischenko, "Stability of solutions in different variables," *Differential Equations*, vol. 43, pp. 1505–1509, Nov. 2007.
[20] 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.
[21] 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.
[22] 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.
[23] 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.
[24] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1722–1729.