To Share or Not to Share? Performance Guarantees and the Asymmetric Nature of Cross-Robot Experience Transfer

Michael J. Sorocky, Siqi Zhou, and Angela P. Schoellig

Abstract—In the robotics literature, experience transfer has been proposed in different learning-based control frameworks to minimize the costs and risks associated with training robots. While various works have shown the feasibility of transferring prior experience from a source robot to improve or accelerate the learning of a target robot, there are usually no guarantees that experience transfer improves the performance of the target robot. In practice, the efficacy of transferring experience is often not known until it is tested on physical robots. This trial-and-error approach can be extremely unsafe and inefficient. Building on our previous work, in this paper we consider an inverse module transfer learning framework, where the inverse module of a source robot system is transferred to a target robot system to improve its tracking performance on arbitrary trajectories. We derive a theoretical bound on the tracking error when a source inverse module is transferred to the target robot and propose a Bayesian-optimization-based algorithm to estimate this bound from data. We further highlight the asymmetric nature of cross-robot experience transfer that has often been neglected in the literature. We demonstrate our approach in quadrotor experiments and show that we can guarantee positive transfer on the target robot for tracking random periodic trajectories.

Index Terms—Machine Learning, Robotics.

I. INTRODUCTION

APPROACHES to transfer experience across robots or across tasks have been explored in different robot learning-based control frameworks (e.g., [1]–[4]). The goal of experience transfer is to leverage existing data or a learned model to accelerate or improve the learning process on a new robot or new task [2]. While existing literature shows the feasibility to improve or accelerate the learning of a robot by leveraging prior experience in training, it is often implicitly assumed without further analysis that the transferred experience will lead to improved performance (i.e., positive transfer) on the target robot or task. This assumption is typically not examined until it is experimentally tested on the target robot hardware. This trial-and-error experience transfer strategy can be unsafe and inefficient for practical robot applications.

In this paper, building on our previous work [5], we consider an inverse module transfer learning framework, where we transfer an inverse dynamics module previously trained on a source robot to improve the tracking performance of a target robot. While classical approaches such as model predictive control (MPC), the linear quadratic regulator (LQR), or PID control can be applied to trajectory tracking problems, these approaches typically rely on a sufficiently accurate dynamics
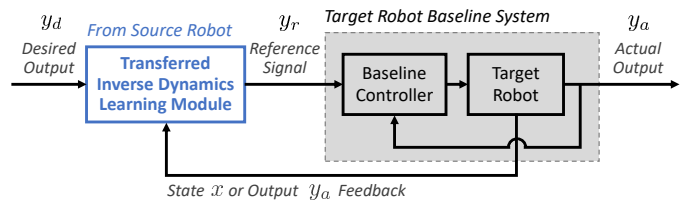
(a) Inverse module transfer learning framework

(b) Source and target quadrotor platforms used in experiments

Fig. 1: The inverse learning control architecture [5] considered in our experience transfer analysis is shown in (a). The goal is to enhance the performance of the target robot system (grey block) for tracking arbitrary trajectories. The blue block represents the inverse dynamics learning module transferred from a source robot system. The transferred inverse block adjusts the reference signal y_r sent to the target robot baseline system such that the map from the desired output y_d to the actual target system output y_a is closer to identity [5]. In this paper, we aim to provide performance guarantees for this transfer approach *prior* to testing source inverse modules on the target robot. We demonstrate our approach in quadrotor experiments on the source and target quadrotors shown in (b). A video of this work can be found here: http://tiny.cc/transfer_guarantees

model of the robot or can be time consuming to tune for achieving high-performance tracking on arbitrary trajectories. With the inverse dynamics learning framework [5], we can improve the tracking performance of a nonlinear control-affine robot system whose dynamics are not exactly known. As shown in Fig. 1a, the learned inverse module (blue) is pre-cascaded to the target robot system (grey), with the goal of achieving an identity map between the desired output y_d and the actual output y_a. In [5], [6], we show that we can train a deep neural network (DNN) to approximate the inverse dynamics of a source quadrotor's baseline control system, which reduced the tracking error of the source quadrotor by an average of 43% on arbitrary hand-drawn trajectories.

Our goal in this paper is to provide performance guarantees when a source inverse module is transferred to a target robot. With this inverse module transfer, we aim to reduce the time and data to train the target robot while guaranteeing improved performance. Our contributions are as follows:
1) derive theoretical bounds on the tracking error when a source inverse module is transferred to the target robot,
2) introduce a Bayesian Optimization (BO)-based algorithm to estimate the tracking error upper bound from simple periodic experiments on the source and target robots for guaranteeing positive transfer,

3) highlight the asymmetric nature of this inverse transfer learning framework, and
4) demonstrate our approach in quadrotor experiments.

II. RELATED WORK

The idea of experience transfer or transfer learning originates from machine learning research. It has been applied to fields such as computer vision [7], natural language processing [8], and reinforcement learning [9]. The goal of transfer learning is to develop methods that exploit information from source domains to facilitate learning in a target domain.

Transfer learning has been considered in different robot learning-based control frameworks. Experience transfer approaches in control can be categorized as cross-task transfer or cross-robot transfer [1]–[4]. Whether it is robot model learning [2], controller learning [3], or reference adaptation [1], [4], the common goal of transferring experience is to leverage prior knowledge such that the time and data required for training a new robot or on a new task can be reduced. Techniques from adaptive control are used in [3] to transfer controller parameters from a source robot to a target robot. In [2], data collected from a source robot is mapped to align with data from the target robot to speed the learning of the forward kinematics of a target manipulator. In [10], the authors consider a similar alignment-based framework and show that the optimal alignment map is a dynamical system.

While existing work demonstrates transferring prior experience can facilitate learning on new robots or tasks, there is usually no guarantee that using prior experience will improve the performance of the target robot before testing. The phenomenon of the performance of the target robot being degraded when using experience from a source robot is known as *negative transfer*. As demonstrated in machine learning literature [11], [12], negative transfer can occur when the source and target domains are dissimilar. In our previous work [13], we also showed that using experience from a dissimilar source robot can cause negative transfer.

In this work, we consider a cross-robot inverse module transfer problem and derive a procedure that allows us to guarantee positive transfer prior to transferring experience from the source robot to the target robot.

III. PROBLEM FORMULATION

In this paper, we consider the control architecture shown in Fig. 1a, where an inverse dynamics module from a source robot R_s is transferred to enhance the tracking performance of a target robot R_t. We say that the transfer is *positive* if the performance of the target robot with the source robot inverse is improved as compared to the target robot baseline system. Our goal is to provide a systematic approach that allows us to identify when we achieve positive transfer from the source robot to the target robot prior to physically testing the source inverse module on the target robot. In our analysis, we make the following assumptions:

(A1) The baseline systems of the source and the target robots are stable and have stable inverse dynamics.
(A2) The source and target baseline systems are single-input single-output (SISO), linear time-invariant (LTI) systems, where the relation between the reference signal $y_r(t)$ and the system output $y_a(t)$ can be represented

by proper transfer functions of the form $G(s) = \frac{\hat{y}_a(s)}{\hat{y}_r(s)}$ with $\hat{y}_r(s)$ and $\hat{y}_a(s)$ denoting the Laplace transforms of $y_r(t)$ and $y_a(t)$, respectively.
(A3) The desired trajectory y_d satisfies $\|y_d\|_2 < \infty$, where $\|\cdot\|_2$ denotes the l_2-norm [14].

Note that assumption (A1) is a necessary condition for safely applying the inverse dynamics learning approach to improve the tracking performance of a black-box system [5]. In (A2), we assume the baseline systems of the source and the target robots are SISO, LTI systems to simplify our analysis. For many practical robot systems, we can use linearization techniques in baseline tracking controller designs. Applying these techniques to quadrotors [15], fully-actuated manipulators [16] or wheeled ground robots [17] typically results in decoupled linear dynamics. The decoupling resulting from linearization allows us to apply our analysis to each SISO system separately and provide insights on transfer performance for a range of practical systems (e.g., [15]–[17]). Lastly, assuming the desired trajectory has finite l_2-norm (A3) is not restrictive, and can be satisfied with a desired trajectory that is bounded, and is non-zero over a finite-time horizon $[0, T]$ with $T < \infty$.

In our discussion, we denote the transfer functions of the source and target robots by $G_s(s)$ and $G_t(s)$, respectively. From (A1), we assume that $G_s(s)$ and $G_t(s)$ and their inverses are bounded-input and bounded-output (BIBO) stable. The tracking error of the target system is defined as $e(t) = y_a(t) - y_d(t)$. We consider two cases: *(i)* using an inverse module from a source robot, in which case we denote the tracking error by $e_{t,s}$, and *(ii)* using the baseline target system without an inverse module (i.e., setting $y_r = y_d$), in which case we denote the tracking error by $e_{t,b}$. We say that *positive transfer* occurs when using an inverse module from a source robot if $\|e_{t,s}\|_2 < \|e_{t,b}\|_2$.

IV. MAIN RESULTS

To ensure safe operation when using an inverse module from a particular source system R_s for the target R_t, our first goal is to find bounds on the corresponding tracking error $\|e_{t,s}\|_2$ *before* testing on the target robot as shown in Fig. 1a. In Sec. IV-A, we derive an upper bound on the tracking error $\|e_{t,s}\|_2$ and relate the bound to our goal of identifying positive transfer. In Sec. IV-B, we propose an algorithm to estimate the bound from data when the source and target dynamics are unknown. In Sec. IV-C, we demonstrate the asymmetric nature of transfer learning in our framework.

A. Bound on the Tracking Error

In this subsection, we assume that the dynamics of the source and target robot systems are known and provide an analysis of the tracking error $e_{t,s}$ of the target robot when the transferred source robot inverse module is applied. We derive an upper bound on $\|e_{t,s}\|_2$ using the signal norm analysis from [14], which relates the infinity norm of a transfer function to the ℓ_2-norm of its input and output signals.

To facilitate our discussion, we note that the relative degree of an LTI system $G(s) = \frac{n(s)}{d(s)}$ is $r = \deg(d(s)) - \deg(n(s))$, where $n(s)$ and $d(s)$ are polynomials in s, and $\deg(\cdot)$ is the degree of a polynomial. We denote the relative degree of the source system $G_s(s)$ and target system $G_t(s)$ by r_s and r_t.

Theorem 1. *Consider the control architecture in Fig. 1a, where an inverse module from robot R_s is pre-cascaded to a target robot R_t. Under the assumptions (A1)–(A3), and suppose that $r_t \geq r_s$, then the ℓ_2-norm of the tracking error of R_t using an inverse module from R_s can be bounded by*

$$\|e_{t,s}\|_2 \leq \|E_{t,s}\|_\infty \|y_d\|_2, \qquad (1)$$

where $E_{t,s}(s) = \frac{\hat{e}_{t,s}(s)}{\hat{y}_d(s)} = G_s^{-1}(s)G_t(s) - 1$ is the transfer function from the desired output $\hat{y}_d(s)$ to the target robot tracking error $\hat{e}_{t,s}(s)$, and $\|E_{t,s}\|_\infty = \sup_{\omega \in \mathbb{R}} |E_{t,s}(j\omega)|$ is the infinity norm of $E_{t,s}$ with $\sup_{\omega \in \mathbb{R}}(\cdot)$ denoting the supremum of a function over frequencies $\omega \in \mathbb{R}$.

Following the signal norm analysis in Sec. 2.3 of [14], we provide a proof of Theorem 1 below.

Proof. Consider source and target systems as defined by (A2). Following (A1), we assume that $G_s^{-1}(s)$ and $G_t(s)$ are BIBO stable. It follows that, if $r_t \geq r_s$, the error transfer function $E_{t,s}(s) = \frac{\hat{e}_{t,s}(s)}{\hat{y}_d(s)}$ is proper and BIBO stable. Since $E_{t,s}(s)$ is proper and stable, by Parseval's theorem [14], the l_2-norms of the error signal $e_{t,s}(t)$ and its Laplace transform $\hat{e}_{t,s}(s)$ are equal: $\|e_{t,s}\|_2 = \|\hat{e}_{t,s}\|_2$. Then, we obtain

$$\begin{aligned}
\|e_{t,s}\|_2^2 = \|\hat{e}_{t,s}\|_2^2 &= \frac{1}{2\pi}\int_{-\infty}^{\infty} |E_{t,s}(j\omega)|^2 |\hat{y}_d(j\omega)|^2 \, d\omega \\
&\leq \|E_{t,s}\|_\infty^2 \frac{1}{2\pi}\int_{-\infty}^{\infty} |\hat{y}_d(j\omega)|^2 \, d\omega \\
&= \|E_{t,s}\|_\infty^2 \|y_d\|_2^2,
\end{aligned}$$

which leads to (1) by taking the square root of each side. Since $\|y_d\|_2$ is finite by (A3) and $E_{t,s}(s)$ is proper and stable, the upper bound given by $\|E_{t,s}\|_\infty \|y_d\|_2$ is finite. $\qquad\square$

Remark 1. *Given the baseline tracking error of the target system $e_{t,b}(t)$, we can guarantee positive transfer using the inverse module from system R_s if $\|E_{t,s}\|_\infty \|y_d\|_2 < \|e_{t,b}\|_2$.*

Since y_d is known, we can compute $\|y_d\|_2$. To compute the upper bound in (1), it remains to compute $\|E_{t,s}\|_\infty$. In Sec. IV-B, we propose an algorithm to estimate $\|E_{t,s}\|_\infty$ for source and target systems with unknown dynamics.

B. Computing Tracking Error Bounds for Systems with Unknown Dynamics

In this subsection, we assume that the exact dynamics of the source and the target systems are unknown and introduce an algorithm to estimate $\|E_{t,s}\|_\infty$ from data. There exist methods in the literature to estimate the infinity norm of a transfer function from its input/output data, e.g. [14], [18]. However, our goal is to find an upper bound on $\|e_{t,s}\|_2$ *before* testing the transfer architecture in Fig. 1a. As a result, we are unable to directly gather input/output data from $E_{t,s}$, and cannot use the aforementioned approaches in the literature.

To this end, we introduce a Bayesian Optimization (BO)-based algorithm to estimate the upper bound on $\|e_{t,s}\|_2$ by running simple periodic experiments on the *baseline systems* of the source and the target robots. Since we consider stable closed-loop baseline source and target systems, these simple periodic experiments can be realized over a practical set of frequencies of interest $\mathcal{W} = \{\omega \in \mathbb{R} \mid \omega_{\min} \leq \omega \leq \omega_{\max}\}$.

To start our discussion, we write the estimation of $\|E_{t,s}\|_\infty$ as the following optimization problem:

$$E_{t,s}^* = \sup_{\omega \in \mathcal{W}} f_{t,s}(\omega), \qquad (2)$$

where $E_{t,s}^*$ is an estimate of $\|E_{t,s}\|_\infty$ and $f_{t,s}(\omega) = \left|G_s^{-1}(j\omega)G_t(j\omega) - 1\right|$ is the objective function to be maximized. Since G_s and G_t are assumed to be unknown, the objective function $f_{t,s}$ is unknown. To solve the black-box optimization problem in (2), we adopt a BO approach. In particular, we use a Gaussian Process (GP) model to approximate the objective function $f_{t,s}$ and optimize the objective function by iteratively sampling from $f_{t,s}$ [19].

The algorithm runs as follows: We start with an initial ω_{sample} that is randomly sampled over $\mathcal{W}$. Then, in each iteration, we proceed with the following steps:

1) *Evaluating the unknown objective function:* We evaluate $f_{t,s}$ at ω_{sample} by sending a sinusoidal reference with frequency ω_{sample} to the source and target *baseline* systems. From the magnitude M and phase θ of the baseline system input/output responses, we can estimate $G_s^{-1}(j\omega_{\text{sample}})$ by $M_s^{-1}\exp(-j\theta_s)$ and $G_t(j\omega_{\text{sample}})$ by $M_t\exp(j\theta_t)$, where subscripts s and t denote the source and target systems. The estimate of $f_{t,s}(\omega_{\text{sample}})$ can be computed from $G_s^{-1}(j\omega_{\text{sample}})$ and $G_t(j\omega_{\text{sample}})$. The data $(\omega_{\text{sample}}, f_{t,s}(\omega_{\text{sample}}))$ is added to a dataset $\mathcal{D} = \{(\omega_k, f_{t,s}(\omega_k))\}_{k=1}^K$, where K is the dataset size at the current iteration.

2) *Fitting a GP model:* Given the dataset $\mathcal{D} = \{(\omega_k, f_{t,s}(\omega_k))\}_{k=1}^K$, we use a GP model to approximate $f_{t,s}$. The GP model has a prior mean function $\mu(\omega)$, and a kernel function $\kappa(\omega, \omega')$. The GP posterior mean $\hat{f}_{t,s}(\omega)$ and posterior covariance $\hat{\sigma}_{t,s}^2(\omega)$ at a test point ω^* are

$$\begin{aligned}
\hat{f}_{t,s}(\omega^*) &= \mu(\omega^*) + (\boldsymbol{k}^*)^T(\boldsymbol{K} + \sigma_n^2 I)^{-1}(\boldsymbol{f} - \boldsymbol{\mu}) \\
\hat{\sigma}_{t,s}^2(\omega^*) &= \kappa(\omega^*, \omega^*) - (\boldsymbol{k}^*)^T(\boldsymbol{K} + \sigma_n^2 I)^{-1}\boldsymbol{k}^*,
\end{aligned} \qquad (3)$$

where $\boldsymbol{K}$ is the covariance matrix with the entry in the ith row and jth column given by $[\boldsymbol{K}]_{ij} = \kappa(\omega_i, \omega_j)$, $\boldsymbol{k}^* = [\kappa(\omega^*, \omega_1), \ldots, \kappa(\omega^*, \omega_K)]^T$, $\boldsymbol{f} = [f_{t,s}(\omega_1), \ldots, f_{t,s}(\omega_K)]^T$, $\boldsymbol{\mu} = [\mu(\omega_1), \ldots, \mu(\omega_K)]^T$, σ_n^2 is the noise variance, and I is the identity matrix [19].

3) *Determining the next sample to acquire:* In each iteration, we select the next sample ω_{sample} based on an acquisition function. A common BO acquisition function is the expected improvement (EI), which is given by

$$\text{EI}_{t,s}(\omega) = (\hat{f}_{t,s}(\omega) - f_{t,s}^{\max})\Phi(Z) + \hat{\sigma}_{t,s}(\omega)\phi(Z), \quad (4)$$

where $f_{t,s}^{\max} = \max_k f_{t,s}(\omega_k)$ is the largest sample we have obtained thus far, $Z = (\hat{f}_{t,s}(\omega) - f_{t,s}^{\max})/\hat{\sigma}_{t,s}(\omega)$, and $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative distribution and probability density functions of the standard normal distribution [19]. The next point we sample at, ω_{sample}, is the one that maximizes the expected improvement: $\omega_{\text{sample}} = \arg\max_{\omega \in \mathcal{W}} \text{EI}_{t,s}(\omega)$.

We repeat steps 1) – 3) until the maximum of the GP posterior mean $\hat{f}_{t,s}$ converges to an approximately constant value. Our estimate of $\|E_{t,s}\|_\infty$ is given by $\hat{E}_{t,s}^* = \hat{f}_{t,s}(\omega_{t,s}^*) + 3\hat{\sigma}_{t,s}(\omega_{t,s}^*)$, where $\omega_{t,s}^* = \arg\max_{\omega \in \mathcal{W}} \hat{f}_{t,s}(\omega)$. The corresponding estimate of the upper bound on $\|e_{t,s}\|_2$ on a desired trajectory y_d is $\bar{e}_{t,s} = \hat{E}_{t,s}^* \|y_d\|_2$.

Note that we can extend the algorithm to the case when we have N source robots $\{R_{s_n}\}_{n=1}^N$, and aim

Algorithm 1: Upper Bound on $\|e_{t,s_n}\|_2$

Input: N source robots $\{R_{s_n}\}_{n=1}^N$, one target robot R_t, desired trajectory y_d, and a frequency range $\mathcal{W} = \{\omega \in \mathbb{R} \mid \omega_{\min} \leq \omega \leq \omega_{\max}\}$.

Output: Upper bound $\bar{e}_{t,s_n}$ on $\|e_{t,s_n}\|_2$.

Initialize: Empty sample datasets: $\mathcal{D}_n \leftarrow \varnothing$ $(n = 1, \ldots, N)$

1 Compute initial sample: $\omega_{\text{sample}} \sim \mathcal{U}(\omega_{\min}, \omega_{\max})$
2 **while** *not converged* **do**
3   Estimate $G_t(j\omega_{\text{sample}})$
4   **for** $n = 1, \ldots, N$ **do**
5   Estimate $G_{s_n}^{-1}(j\omega_{\text{sample}})$
6   Compute $f_{t,s_n}(\omega_{\text{sample}})$
7      $\mathcal{D}_n \leftarrow \mathcal{D}_n \cup \{(\omega_{\text{sample}}, f_{t,s_n}(\omega_{\text{sample}}))\}$
8   Fit nth GP with data $\mathcal{D}_n$
9   Compute $\omega_{\text{sample}} = \arg\max_{\omega \in \mathcal{W}} \alpha(\omega)$
10 **for** $n = 1, \ldots, N$ **do**
11    $\omega_{t,s_n}^* = \arg\max_{\omega \in \mathcal{W}} \hat{f}_{t,s_n}(\omega)$
12    $\hat{E}_{t,s_n}^* = \hat{f}_{t,s_n}(\omega_{t,s_n}^*) + 3\hat{\sigma}_{t,s_n}(\omega_{t,s_n}^*)$
13    $\bar{e}_{t,s_n} = \hat{E}_{t,s_n}^* \|y_d\|_2$

---

to estimate the corresponding upper bounds on $\|e_{t,s_n}\|_2$ $(n = 1, \ldots, N)$. In this case, we have N optimization problems in the form of (2): $E_{t,s_n}^* = \sup_{\omega \in \mathcal{W}} f_{t,s_n}(\omega)$, where $f_{t,s_n}(\omega) = |G_{s_n}^{-1}(j\omega)G_t(j\omega) - 1|$. To solve them, we follow a similar approach to steps 1) – 3): we model each objective function $f_{t,s_n}(\omega)$ with its own GP, with posterior mean $\hat{f}_{t,s_n}(\omega)$ and posterior variance $\hat{\sigma}_{t,s_n}^2(\omega)$. We replace the acquisition function in (4) with $\alpha(\omega) = \max_n \text{EI}_{t,s_n}(\omega)$. The next sample point is the one that maximizes $\alpha(\omega)$: $\omega_{\text{sample}} = \arg\max_{\omega \in \mathcal{W}} \alpha(\omega)$. That is, given the expected improvement $\text{EI}_{t,s_n}(\omega)$ for each system, we sample at the point that gives the highest EI. Upon convergence, our estimate of $\|E_{t,s_n}\|_\infty$ is given by $\hat{E}_{t,s_n}^* = \hat{f}_{t,s_n}(\omega_{t,s_n}^*) + 3\hat{\sigma}_{t,s_n}(\omega_{t,s_n}^*)$, where $\omega_{t,s_n}^* = \arg\max_{\omega \in \mathcal{W}} \hat{f}_{t,s_n}(\omega)$. The upper bound on $\|e_{t,s_n}\|_2$ is given by $\bar{e}_{t,s_n} = \hat{E}_{t,s_n}^* \|y_d\|_2$. We summarize the overall algorithm in Alg. 1, and demonstrate the algorithm with quadrotor experiments in Sec. V. Note that while we assume the source and target systems are linear, Alg. 1 uses only their input/output data, without requiring knowledge of the system structure (e.g., number of poles or zeros).

C. Asymmetry in Inverse Module Transfer Learning

In this section, we explore the asymmetric nature that is inherent in transfer learning. In the following discussion, we first demonstrate the asymmetry property for two first-order systems in the inverse transfer learning framework and then extend our discussion to a more generalized setting.

We consider two first-order systems $G_1(s) = \frac{1}{\tau_1 s + 1}$ and $G_2(s) = \frac{1}{\tau_2 s + 1}$, where τ_1, $\tau_2 > 0$ are the time constants of each system. The baseline tracking error of each system is denoted $e_{1,b}(t)$ and $e_{2,b}(t)$, respectively. The tracking error when pre-cascading G_1^{-1} to G_2 and when pre-cascading G_2^{-1} to G_1 are denoted by $e_{2,1}(t)$ and $e_{1,2}(t)$, respectively.

Example 1. *Consider the two first-order systems with transfer functions $G_1(s)$ and $G_2(s)$. Without loss of generality, we assume $\tau_2 < \tau_1$. Then, for a given desired trajectory y_d, transferring the inverse G_2^{-1} to G_1 always leads to positive transfer, i.e. $\|e_{1,2}\|_2 < \|e_{1,b}\|_2$. However, the reverse is not true. For the same trajectory y_d, transferring the inverse G_1^{-1} to G_2 is positive ($\|e_{2,1}\|_2 < \|e_{2,b}\|_2$) iff τ_1 satisfies $\tau_1 < 2\tau_2$.*

Proof. By directly evaluating the convolution integral for the output of a linear system, one can show that $e_{1,b}(t) = \frac{1}{\tau_1}e^{-t/\tau_1}\mathcal{I} - y_d(t)$ and $e_{1,2}(t) = \left(1 - \frac{\tau_2}{\tau_1}\right)e_{1,b}(t)$, where $\mathcal{I} = \int_0^t y_d(z)e^{z/\tau_1}dz$. It follows that $\|e_{1,2}\|_2 < \|e_{1,b}\|_2$ iff $\tau_2 < 2\tau_1$. By assumption, $\tau_2 < \tau_1$, and thus the condition of positive transfer from G_2^{-1} to G_1 is always satisfied. For the reverse direction, one can similarly show that $\|e_{2,1}\|_2 < \|e_{2,b}\|_2$ iff $\tau_1 < 2\tau_2$. Thus, transfer from G_1^{-1} to G_2 is positive iff τ_1 is sufficiently small (i.e., $\tau_1 < 2\tau_2$). $\square$

Remark 2. *For the systems we consider in Example 1, the time constant τ of each system represents the rise time of the system in response to a step input, which can be interpreted as the system's aggressiveness with a smaller value indicating a higher aggressiveness. The result then implies that it is easier to guarantee positive transfer from a more aggressive system to a less aggressive system, but not vice versa.*

We can extend the insight by leveraging the ν-gap metric from robust control, which provides a notion of 'distance' between two systems. For two SISO minimum phase systems satisfying $\|G_2(-j\omega)G_1(j\omega)\|_\infty < 1$, the ν-gap metric can be written as $\delta_\nu(G_1, G_2) = \sup_\omega \psi(G_1(j\omega), G_2(j\omega))$, where $\psi(G_1(j\omega), G_2(j\omega)) = \frac{|G_1(j\omega) - G_2(j\omega)|}{\sqrt{(1 + |G_1(j\omega)|^2)(1 + |G_2(j\omega)|^2)}}$ is the chordal distance between the projections of $G_1(j\omega)$ and $G_2(j\omega)$ onto the Riemann sphere [20]. For a desired trajectory with frequency ω, we can decompose the error transfer function into a symmetric and an asymmetric component:

$$|E_{1,2}(j\omega)| = \underbrace{\psi(G_1(j\omega), G_2(j\omega))}_{\text{symmetric}} \underbrace{\Psi(G_1(j\omega), G_2(j\omega))}_{\text{asymmetric}},$$

where $\Psi(G_1(j\omega), G_2(j\omega)) = \frac{\sqrt{1 + |G_1(j\omega)|^2}}{\psi(G_2(j\omega), 0)}$. The term $\psi(G_1(j\omega), G_2(j\omega))$ is the symmetric term providing a notion of 'distance' between G_1 and G_2. Systems close in this sense will have reduced $|E_{1,2}(j\omega)|$. The term $\Psi(G_1(j\omega), G_2(j\omega))$ reflects the direction of transfer. Note that, since $\psi(G_2(j\omega), 0)$ represents the chordal distance between $G_2(j\omega)$ and 0, we can interpret $\psi(G_2(j\omega), 0)$ as a measure of the aggressiveness of the system. As a result, if G_2 is more aggressive, then $\Psi(G_1(j\omega), G_2(j\omega))$ and thus $|E_{1,2}(j\omega)|$ is further reduced. This implies that transferring an inverse module from a more aggressive system is more likely to reduce the tracking error of a less aggressive system and thereby results in positive transfer.

We note that the asymmetric nature of transfer learning is not limited to our inverse module transfer framework. The asymmetric behaviour is also found in frameworks such as alignment-based transfer learning [10], [21]. To guarantee positive transfer, it is important to define a measure of similarity between the source and target domains [11], [12]. Such measures of similarity should account for the direction of transfer, which has a direct impact on transfer success.

V. QUADROTOR EXPERIMENTS

In this section, we demonstrate Alg. 1 and the asymmetric nature of the transfer problem with quadrotor experiments.

A. Experiment Setup

We consider one target robot R_t, whose tracking is to be improved, and five source robots (Fig. 1b), whose inverse

modules have been previously trained and can be leveraged to improve the tracking of R_t. The target robot in our experiments is the Crazyflie 2.0 quadrotor; its mass is 36 g, with a rotor-to-rotor distance of 14 cm. The five source robots are: R_{s_1} – Crazyflie 2.0 quadrotor (light), R_{s_2} – Crazyflie simulator, R_{s_3} – Parrot ARDrone 2.0, R_{s_4} – Parrot ARDrone simulator, and R_{s_5} – Parrot Bebop 2. The simulators R_{s_2}, R_{s_4} emulate the dynamics and control of R_t and R_{s_3}, but do not account for effects such as aerodynamic drag or time delays. The mass of the source robots are 0.86, 1, 13.7, 13.7 and 14.5 times that of the target, respectively, and the rotor-to-rotor distance of the source robots are 1, 1, 3.9, 3.9, and 3.1 times that of the target, respectively. The baseline controllers for the target R_t and sources R_{s_1} and R_{s_2} are designed by exploiting the differential flatness of quadrotors [22], while the baseline controllers for sources R_{s_3}–R_{s_5} are designed based on a combination of feedback linearization and PD control [15]. We assume that the position dynamics of each quadrotor is decoupled in the x-, y-, and z-directions. The inverse module of each source system is approximated by a deep neural network (DNN) trained on 400 seconds of data collected on each source baseline system [5].

Our experiments consist of three parts: in Sec. V-B, we apply Alg. 1 to estimate the upper bound on tracking error of the target quadrotor when using the inverse module from each source quadrotor. In Sec. V-C, we verify the upper bounds on five random periodic trajectories. In Sec. V-D, we illustrate the asymmetric nature of this transfer problem. We note that in the upper bound estimation experiments and test trajectory experiments, we rely on the input/output data from the source and target quadrotors, and do not require knowledge of their exact dynamics or system structures.

B. Tracking Error Upper Bound Estimation

We apply Alg. 1 proposed in Sec. IV-B to identify the upper bounds on the tracking errors of the target quadrotor system pre-cascaded with the DNN inverse modules of the source quadrotors. Since by assumption the dynamics of the source and target systems are decoupled in the x-, y-, and z-directions, we apply Alg. 1 to each direction separately. We denote the tracking error of R_t using an inverse module from source R_{s_n} in the x-, y-, and z-directions by $e_{t,s_n,x}(t)$, $e_{t,s_n,y}(t)$, and $e_{t,s_n,z}(t)$, respectively. Our goal is to estimate $\hat{E}^*_{t,s_n,x}$, $\hat{E}^*_{t,s_n,y}$, and $\hat{E}^*_{t,s_n,z}$ to compute the error bounds on $\|e_{t,s_n,x}\|_2$, $\|e_{t,s_n,y}\|_2$, and $\|e_{t,s_n,z}\|_2$ (see (1)). As outlined in Alg. 1, we estimate the error norms $\hat{E}^*_{t,s_n,x}$, $\hat{E}^*_{t,s_n,y}$, and $\hat{E}^*_{t,s_n,z}$ iteratively by running periodic trajectories on the baseline systems along the x-, y-, and z-directions. The estimates of the error norms converged in six iterations in the x- and y-directions, and five iterations in the z-direction.

The converged estimates $\hat{E}^*_{t,s_n,x}$, $\hat{E}^*_{t,s_n,y}$, and $\hat{E}^*_{t,s_n,z}$ are summarized in Table I. Sources R_{s_1} and R_{s_2} are of similar size and agility as the target R_t, and thus the upper bound on tracking error in each direction when using their inverse modules is low. Conversely, sources R_{s_3}–R_{s_5} are much larger and less agile than the target, leading to a larger upper bound on tracking error when using their inverse modules. These differences are especially pronounced in the x- and y-directions. From Table I, we expect transferring the inverse modules from source quadrotors similarly aggressive to the

TABLE I: A summary of the error norm estimates $\hat{E}^*_{t,s_n}$ of the target quadrotor when using the inverse modules from the five source quadrotors.

Direction	$\hat{E}^*_{t,s_1}$	$\hat{E}^*_{t,s_2}$	$\hat{E}^*_{t,s_3}$	$\hat{E}^*_{t,s_4}$	$\hat{E}^*_{t,s_5}$
x	0.22	0.25	2.70	2.19	3.03
y	0.23	0.28	3.20	2.23	3.25
z	0.22	0.24	0.56	1.55	0.98

target (R_{s_1} and R_{s_2}) would lead to lower tracking error on the target quadrotor as compared to transferring the inverse modules from less aggressive source quadrotors (R_{s_3}–R_{s_5}).

Note that to compute the estimates in Table I, we required 209 seconds of data from the target quadrotor, approximately half of the data needed to train its inverse module from scratch. As we show in Sec. V-C, the estimated upper bounds allow us to safely reuse an inverse from a source quadrotor to improve the target quadrotor tracking performance.

C. Test Trajectory Tracking Experiments

We validate the estimated upper bounds on the target system tracking error (Table I) on five test trajectories. The test trajectories are parameterized as $y_d(t) = [0.25\sin(\omega_x t),\ 0.25\cos(\omega_y t),\ 0.25\sin(\omega_z t)]$, where ω_x, ω_y, and ω_z are randomly sampled between 0 and 2 rad/sec, which is the maximal range for the target quadrotor to track.

We quantify the tracking performance of the target quadrotor as $e_{t,\cdot} = (\|e_{t,\cdot,x}\|_2^2 + \|e_{t,\cdot,y}\|_2^2 + \|e_{t,\cdot,z}\|_2^2)^{1/2}$. Using the estimates in Table I, we can bound the tracking error of the target R_t when using the inverse module from R_{s_n} as $e_{t,s_n} \leq e^*_{t,s_n}$, where $e^*_{t,s_n} = ((\hat{E}^*_{t,s_n,x}\|y_{d,x}\|_2)^2 + (\hat{E}^*_{t,s_n,y}\|y_{d,y}\|_2)^2 + (\hat{E}^*_{t,s_n,z}\|y_{d,z}\|_2)^2)^{1/2}$. We guarantee positive transfer from R_{s_n} to R_t if the condition $e^*_{t,s_n} < e_{t,b}$ is satisfied.

We ran each test trajectory on the target quadrotor baseline system with and without the source inverse modules. As shown in Table II, for each trajectory and each source robot R_{s_n}, we have that $e_{t,s_n} < e^*_{t,s_n}$, verifying the efficacy of the upper bounds. By comparing the baseline tracking error $e_{t,b}$ to the upper bounds e^*_{t,s_n}, we observe that we can guarantee positive transfer for sources R_{s_1} and R_{s_2}, as $e^*_{t,s_n} < e_{t,b}$ for $n \in \{1,2\}$. From the estimated upper bounds, we can predict positive transfer for R_{s_1} and R_{s_2} before testing their inverse modules on the target quadrotor. Using inverse modules from R_{s_1} and R_{s_2} improve the target quadrotor performance by an average of 74% without requiring retraining an inverse module for the target system. Conversely, for R_{s_3}–R_{s_5}, we cannot guarantee positive transfer; indeed, when using these source inverse modules, we observe negative transfer ($e_{t,s_n} > e_{t,b}$).

D. The Asymmetric Nature of Inverse Module Transfer

Although from the tracking experiments, we see that using an inverse module from a less similar robot (e.g., R_{s_5}) can cause negative transfer to the target R_t, the reverse is not necessarily true. To demonstrate the asymmetric nature of transfer, we train an inverse module for the target robot R_t and pre-cascade this module to the baseline controller of R_{s_5}.

As an example, in the y-direction for one test trajectory from Sec. V-C, when using the inverse module from R_{s_5}, the tracking error of R_t, $\|e_{t,s_5,y}\|_2$, is *increased* by 13% relative to its baseline error. However, when transferring the inverse module from R_t to R_{s_5}, the tracking error of R_{s_5},

TABLE II: Quadrotor trajectory tracking experiment results. For each source system R_{s_n}, we display the actual tracking error e_{t,s_n}, the tracking error upper bound e_{t,s_n}^*, and the baseline tracking error $e_{t,b}$ of the target robot. Positive transfer cases (i.e., entries with values less than $e_{t,b}$) are indicated with boldface. For all cases, we have that $e_{t,s_n} < e_{t,s_n}^*$, verifying the efficacy of the upper bounds. Positive transfer is guaranteed for R_{s_1} and R_{s_2}.

| Traj.[†] | Transfer from R_{s_1} | | Transfer from R_{s_2} | | Transfer from R_{s_3} | | Transfer from R_{s_4} | | Transfer from R_{s_5} | | Baseline |
	e_{t,s_1}[m]	e_{t,s_1}^* [m]	e_{t,s_2} [m]	e_{t,s_2}^* [m]	e_{t,s_3} [m]	e_{t,s_3}^* [m]	e_{t,s_4} [m]	e_{t,s_4}^* [m]	e_{t,s_5} [m]	e_{t,s_5}^* [m]	$e_{t,b}$ [m]
y_{d_1}	**0.144**	**0.358**	**0.161**	**0.335**	0.775	2.401	**0.542**	2.440	0.610	3.263	0.562
y_{d_2}	**0.174**	**0.419**	**0.162**	**0.399**	0.834	2.990	**0.557**	2.871	0.822	3.962	0.665
y_{d_3}	**0.224**	**0.358**	**0.210**	**0.335**	0.926	2.396	1.041	2.441	0.923	3.264	0.817
y_{d_4}	**0.215**	**0.354**	**0.215**	**0.331**	0.890	2.382	1.076	2.418	0.903	3.244	0.867
y_{d_5}	**0.252**	**0.357**	**0.258**	**0.334**	**0.945**	2.389	1.110	2.434	1.081	3.252	0.972

[†]The frequencies $(\omega_x, \omega_y, \omega_z)$ of the five sinusoidal test trajectories were uniformly sampled between 0 and 2 rad/sec using the Matlab function `rand`. The values are $(\omega_x, \omega_y, \omega_z) \in \{(1.087, 0.557, 0.849), (1.690, 0.009, 0.243), (1.342, 1.652, 0.273), (1.150, 1.783, 0.418), (1.957, 1.623, 0.344)\}$ rad/sec.
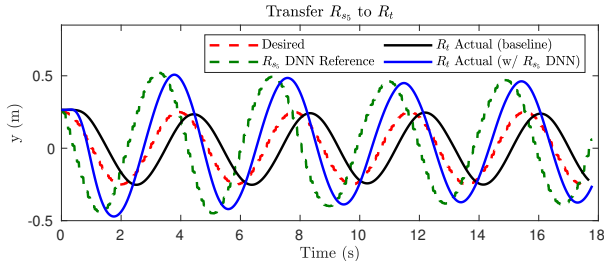

Fig. 2: Using an inverse module from R_{s_5} for R_t increases the tracking error of R_t, $\|e_{t,s_5,y}\|_2$, by 13%, relative to its baseline tracking error. The inverse module from R_{s_5} amplifies the reference y_d (red) to produce a reference y_r (green). The over-amplification of the reference adversely causes an increase in tracking error of R_t compared to its baseline (black).

$\|e_{s_5,t,y}\|_2$, is *reduced* by 9%, relative to its baseline error. The intuition of this asymmetry is discussed in Sec. IV-C. The negative transfer from R_{s_5} to R_t is a result of R_{s_5} being less aggressive. As compared to R_t, the baseline system of R_{s_5} has larger delays and damping. To compensate for the slow response, as seen in Fig. 2, given a desired trajectory (red), the inverse of R_{s_5} tends to produce a significantly amplified reference (green). When sending the amplified reference to the target robot R_t, the target robot R_t follows the R_{s_5} DNN reference (green) 'too well', causing it to overshoot the desired trajectory. Conversely, when pre-cascading the inverse of R_t, the more aggressive robot, to R_{s_5}, the less aggressive robot, the R_t inverse module slightly amplifies the reference signal to compensate for the slow response of R_{s_5}. This results in positive transfer from R_t to R_{s_5}.

VI. CONCLUSION AND FUTURE WORK

In this paper, we considered an inverse module transfer framework in which an inverse dynamics module from a source system is transferred to a target system with the goal of improving the tracking performance of the target system. We derived an upper bound on the l_2-norm of the tracking error of the target system when using a source inverse module and proposed an algorithm to estimate this bound using input/output data from the source and target systems. We also highlighted the asymmetric nature of this transfer framework. In quadrotor experiments, we demonstrated the efficacy of the upper bounds, showed how the estimated bound can be used for guaranteeing positive transfer, and illustrated the asymmetric nature in the inverse transfer framework with two quadrotors cross-sharing their inverse modules.

As future work, we would like to extend our theoretical results on transfer performance guarantees and the asymmetric nature of this transfer framework to multi-input multi-output nonlinear systems and explore the asymmetric nature of transfer learning in other robot learning-based control frameworks.

REFERENCES

[1] M. Hamer, M. Waibel, and R. D'Andrea, "Knowledge transfer for high-performance quadrocopter maneuvers," in *IEEE Intl. Conf. on Intelligent Robots and Systems*, 2013.

[2] B. Bocsi, L. Csató, and J. Peters, "Alignment-based transfer learning for robot models," in *Intl. Joint Conf. on Neural Networks*, 2013.

[3] G. Chowdhary, T. Wu, M. Cutler, and J. P. How, "Rapid transfer of controllers between uavs using learning-based adaptive control," in *IEEE Intl. Conf. on Robotics and Automation*, 2013.

[4] K. Pereida, M. K. Helwa, and A. P. Schoellig, "Data-efficient multirobot, multitask transfer learning for trajectory tracking," *IEEE Robotics and Automation Letters*, 2018.

[5] S. Zhou, M. K. Helwa, and A. P. Schoellig, "Design of deep neural networks as add-on blocks for improving impromptu trajectory tracking," in *IEEE Conf. on Decision and Control*, 2017.

[6] Q. Li, J. Qian, Z. Zhu, X. Bao, M. K. Helwa, and A. P. Schoellig, "Deep neural networks for improved, impromptu trajectory tracking of quadrotors," in *IEEE Intl. Conf. on Robotics and Automation*, 2017.

[7] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2014.

[8] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," in *IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conf.*, 2015.

[9] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, 2009.

[10] M. K. Helwa and A. P. Schoellig, "Multi-robot transfer learning: A dynamical system perspective," in *IEEE Intl. Conf. on Intelligent Robots and Systems*, 2017.

[11] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell, "Characterizing and avoiding negative transfer," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[12] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, "To transfer or not to transfer," in *Neural Information Processing Systems Workshop on Transfer Learning*, 2005.

[13] M. J. Sorocky, S. Zhou, and A. P. Schoellig, "Experience selection using dynamics similarity for efficient multi-source transfer learning between robots," in *IEEE Intl. Conf. on Robotics and Automation*, 2020.

[14] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. Courier Corporation, 2013.

[15] M. K. Helwa and A. P. Schoellig, "On the construction of safe controllable regions for affine systems with applications to robotics," *Automatica*, 2018.

[16] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley New York, 2006.

[17] J. L. Giesbrecht, H. K. Goi, T. D. Barfoot, and B. A. Francis, "A vision-based robotic follower vehicle," in *SPIE 7332, Unmanned Systems Technology XI*, 2009, pp. 73 321O1–73 321O12.

[18] K. Van Heusden, A. Karimi, and D. Bonvin, "Data-driven estimation of the infinity norm of a dynamical system," in *IEEE Conf. on Decision and Control*, 2007.

[19] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. of the IEEE*, 2015.

[20] K. Zhou and J. C. Doyle, *Essentials of Robust Control*. Prentice hall Upper Saddle River, NJ, 1998.

[21] K. V. Raimalwala, B. A. Francis, and A. P. Schoellig, "An upper bound on the error of alignment-based transfer learning between two linear, time-invariant, scalar systems," in *IEEE Intl. Conf. on Intelligent Robots and Systems*, 2015.

[22] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE Intl. Conf. on Robotics and Automation*, 2011.