Rapid Transfer Of Controllers Between UAVs Using Learning-Based Adaptive Control

Girish Chowdhary, Tongbin Wu, Mark Cutler, Jonathan P. How

Abstract— **Commonly used Proportional-Integral-Derivative based UAV flight controllers are often seen to provide adequate trajectory-tracking performance, but only after extensive tuning. The gains of these controllers are tuned to particular platforms, which makes transferring controllers from one UAV to other time-intensive. This paper formulates the problem of control-transfer from a source system to a transfer system and proposes a solution that leverages well-studied techniques in adaptive control. It is shown that concurrent learning adaptive controllers improve the trajectory tracking performance of a quadrotor with the baseline linear controller directly imported from another quadrotor whose inertial characteristics and throttle mapping are very different. Extensive flight-testing, using indoor quadrotor platforms operated in MIT's RAVEN environment, is used to validate the method.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have been heavily investigated recently for aerial robotics with several potential applications. The research focus is shifting to developing teams of smaller networked UAVs capable of robustly outperforming a single, larger, and more expensive UAV. Fueled by speedy manufacturing techniques, several small UAV platforms that are specifically outfitted for a particular mission are being developed. Due to their simplicity and reliability, the autopilots for these vehicles are often designed using Proportional-Integral-Derivative (PID) based techniques (see e.g. [1]–[5]). Well tuned PID controllers have been shown to yield excellent flight performance on quadrotors and other UAVs. Notably, PID based quadrotor UAVs have been used to perform aggressive maneuvers by various groups [5]–[8]. PID based controllers however, need significant tuning of the gains for extracting good performance. Furthermore, since the gains of such controllers are tuned to particular platforms, therefore transferring controllers from one UAV to another becomes time-intensive. This paper tackles the problem of efficiently transferring controllers between different UAV platforms using ideas from adaptive control.

The problem of control transfer is rigorously framed using ideas from feedback linearization and adaptive control, and it is shown that techniques similar to those employed in the widely studied framework of Model Reference Adaptive Control (MRAC) can be used to transfer controllers between systems that have "similar" control structure. The notion of similarity is formalized through two assumptions. The first assumption requires the existence of an isomorphic control law on the source system that can be inverted w.r.t. the control input of the transfer system, and the second assumption requires that the sign of control effectiveness be same on both source and transfer systems. It is further argued that for true control transfer, the adaptive controller must learn the model of the transfer system over the long-term to improve the controller's predictive ability. This motivates the use of learning-focused MRAC techniques such as concurrent learning MRAC. The theoretical results are complimented by extensive experimental demonstration. It is shown that a concurrent learning adaptive controller and a budgeted nonparametric adaptive control technique improves the trajectory tracking performance of a smaller quadrotor whose controller has been directly imported (without changing gains) from a larger quadrotor whose inertial characteristics and throttle mapping are very different (but satisfy the two above mentioned assumptions). The flight-test results are presented on indoor quadrotor platforms operated in MIT's RAVEN environment.

A. Related Work

Motivated by the fact that it is difficult or costly to obtain an exact model of system dynamics, the widely studied field of adaptive control has investigated several methods for mitigating modeling error in control design. In the context of flight vehicle control, Calise [9], Johnson and Kannan [10] and others have developed model reference adaptive controllers for both fixed wing and rotary wing UAVs by employing Neural Network adaptive elements. Cao, Yang, Hovaykiman, and others have developed the L_1 adaptive control method [11]. Lavretsky [12], Nguyen [13], Steinberg [14], Jourdan et al. [15] and others have extended direct adaptive control methods to fault tolerant control and developed techniques in composite/hybrid adaptation. Chowdhary and Johnson developed concurrent learning adaptive controllers, which use online selected and recorded data concurrently with current data for adaptation for guaranteeing improved learning and stability properties. Their methods have been tested on fixed- and rotary-wing UAVs [16,17]. Kingravi et al. further extended Concurrent Learning to accommodate unknown domains of operation [18] by using ideas from reproducing kernel Hilbert spaces to maintain and update a dictionary of kernels online, so that the internal parameters of the kernel based adaptive element are changed according to the domain of operation. However, the problem of using adaptive-control like techniques for control transfer has not been explicitly studied, although it is clear in the literature that this application has been envisioned. One notable

The authors are with the Laboratory of Information and Decision Systems and with Dept. of Aeronautics and Astronautics at the Massachusetts Institute of Technology, Cambrige MA girishc,twu1,cutlerm,jhow@mit.edu.

exception is [19], authors of which noted in passing that controllers were transferred between two aircraft with little modification. However, a formal transfer technique was not pursued. This paper contributes to the field by formulating the control-transfer problem and rigorously incorporating techniques from adaptive control for its solution.

The problem of transfer-learning has been investigated in some detail in the field of machine learning, an excellent survey in the context of reinforcement learning is [20]. Authors have explored learning through similar tasks [21], shared features for transfer-learning [22], learning from solving multiple MDPs [23], and hierarchical Bayes approaches [24]. However, the focus of these problems has been on higher-level decision making problems involving discrete state-action spaces. Because the control-transfer problem is characterized by continuous state-spaces, and because the generated policy is safety-critical, existing transfer learning methods are not well suited to this task. The learning-based direct adaptive control method pursued here can be viewed as a subclass of fixed-policy model-based reinforcement learning techniques for continuous MDPs [25]. Therefore, the techniques in this paper contribute to the transfer learning literature by providing a specific technique for solving the control-transfer problem.

Finally, some preliminary results relating to this paper were presented in [26]. The main differences here include a rigorous formulation of the control transfer problem, flight testing of the nonparametric Budgeted Kernel Restructuring - Concurrent Learning (BKR-CL) method, and flight testing over different flight scenarios.

II. The Control Transfer Problem

The source-system, that is the system from which the controller is to be transferred, is assumed to be modeled by the following differential equation:

$$\ddot{x}_s(t) = f_s(x_s(t), \dot{x}_s(t), \delta_s(t)), \tag{1}$$

while the transfer-system, that is the system to which the controller is being transferred to, is also modeled as:

$$\ddot{x}_t(t) = f_t(x_t(t), \dot{x}_t(t), \delta_t(t)). \tag{2}$$

The functions f_s, f_t, are assumed to be Lipschitz continuous in $x, \dot{x} \in D_x \subset \mathbb{R}^n$, and have an equilibrium at $(0,0,0)$. The control inputs $\delta_{s_t} \in D_\delta \subset \mathbb{R}^n$ are assumed to belong to the set of admissible control inputs consisting of measurable bounded functions, and the systems are assumed to be finite input controllable. Therefore, existence and uniqueness of piecewise solutions to (1) are guaranteed. In addition, a condition on controllability must be assumed. These assumptions are typically satisfied by most UAV platforms, including quadrotors. Furthermore, note that we have assumed that the dimension of control is equal to the dimension of the state, this assumption is valid for most UAVs with an inner-outer loop control structure [27] (e.g. two attitude and one throttle commands correspond to three dimensional outerloop position control of a quadrotor UAV).

The desired trajectory to be tracked by both the source and the transfer system is characterized by the bounded twice continuously differential function of time $x_{\text{ref}}(t)$. The following assumption characterizes the source-controller on the source system:

Assumption 1 There exists a control law $g : D_x \to D_\delta$ such that $\delta_s = g(x_s, \dot{x}_s, \ddot{x}_{\text{des}})$ drives $x_s \to x_{\text{des}}$ as $t \to \infty$. Furthermore, the control law is invertible w.r.t. δ_s, hence the relation $\ddot{x}_{\text{des}} = g^{-1}(x_s, \dot{x}_s, \delta_s)$ holds.

Letting $e = x_{\text{ref}} - x_s$, a highly succesful technique in UAV control has been to map the tracking error to a desired acceleration $\ddot{x}_{\text{des}}$ which can then be converted to actuator commands through control inversion models (see e.g. [4,5,10,16,28]). The desired acceleration is typically found through a linear PD feedback term and a feedforward term: $x_{\text{des}} = K_p e + K_d \dot{e} + \ddot{x}_{\text{ref}}$. With an additional integral term, a set of well tuned PID gains result in a controller satisfying Assumption 1 (see e.g. [1,2,5]–[8]).

Let $z_t = (x_t, \dot{x}_t, \delta_t)$. If for the transferred system, we let $\delta_t = g(x_t, \dot{x}_t, \ddot{x}_{\text{des}})$ (with the same PID gains as for the source system) the following discrepancy could arise:

$$\ddot{x}_t = \ddot{x}_{\text{des}} + \Delta(z_t) \tag{3}$$

where the unknown error $\Delta(.)$ between the desired and the actual behavior is given by

$$\Delta(x_t, \dot{x}_t, \delta_t) = f_t(z_t) - g^{-1}(z_t). \tag{4}$$

An approximate model inversion-based MRAC approach is leveraged here to adapt to this unknown modeling error by directly modifying the desired acceleration $\ddot{x}_{\text{des}_t}$ for the transfer system as follows:

$$\ddot{x}_{\text{des}_t}(z_t) = \ddot{x}_{\text{des}}(z_t) - \nu_{ad}(z_t), \tag{5}$$

where ν_{ad} is the output of an adaptive element designed to cancel the modeling uncertainty in (4). Note that since Δ is a function of ν_{ad} and ν_{ad} needs to be designed to cancel Δ, the following assumption needs to be satisfied:

Assumption 2 The existence and uniqueness of a fixed-point solution to $\nu_{ad} = \Delta(., \nu_{ad})$ is assumed. Sufficient conditions are available for satisfying this assumption [29,30].

From the results in [29] it can be further argued that this condition implicitly requires that the sign of control effectiveness derivative is same for both the source and the transfer system, that is $\text{sgn}\frac{\partial g}{\partial \delta_s} = \text{sgn}\frac{\partial g}{\partial \delta_t}$.

Noting that $\ddot{x}_{\text{ref}} - \ddot{x}_t = \ddot{x}_{\text{ref}} - (\ddot{x}_{\text{des}_t} + \Delta(z_t))$ due to (3) and using (5) the tracking error dynamics can be derived as

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = A \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + B[\nu_{ad}(x, \dot{x}, \delta) - \Delta(x, \dot{x}, \delta)], \tag{6}$$

where $A = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix}$ contains the source control PD gains and $B = \begin{bmatrix} 0 \\ I \end{bmatrix}$. Note that for $K_p > 0$ and $K_d > 0$ A is Hurwitz, therefore, a unique positive definite solution $P \in$

$\Re^{2n\times 2n}$ exists to the Lyapunov equation $0 = A^TP+PA+Q$ for $Q > 0$.

Radial Basis Function Networks (RBFN) have been widely used as universally approximating adaptive elements to capture continuous modeling uncertainties [29,31,32]. One reason RBFN have received significant popularity is because of their linear-in-parameters structure. When using RBFN, the adaptive element is represented by

$$\nu_{ad}(z_t) = W^T\sigma(z_t), \tag{7}$$

where $W \in \mathbb{R}^{q\times n_2}$ and $\sigma(z_t) = [1, \sigma_2(z_t), ..., \sigma_q(z)]^T$ is a q dimensional vector of chosen radial basis functions. For $i = 2, 3..., q$ let c_i denote the RBF centroid and μ_i denote the RBF widths then for each i the radial basis functions are given as $\sigma_i(z_t) = e^{-\|z_t-c_i\|^2/\mu_i}$. Appealing to the universal approximation property of Radial Basis Function Networks [33] we have that given a fixed number of radial basis functions q there exists ideal parameters $W^* \in \mathbb{R}^{q\times n_2}$ and a vector $\tilde{\epsilon} \in \mathbb{R}^n$ such that the following approximation holds for all $z \in D \subset \mathbb{R}^{n+l}$ where D is compact

$$\Delta(z_t) = {W^*}^T\sigma(z_t) + \tilde{\epsilon}(z_t). \tag{8}$$

Furthermore $\bar{\epsilon} = \sup_{z_t\in D_x}\|\tilde{\epsilon}(z)\|$ can be made arbitrarily small given sufficient number of radial basis functions.

Therefore, as formulated here, the control-transfer problem can be reduced to finding a weight update law $\dot{W}(t)$ such that $x_t(t)$ can be made to follow $x_{\text{ref}}(t)$. The design of such adaptive laws is a well-studied problem in the framework of MRAC. The following theorem formalizes this using the well known projection based weight update law [34]:

$$\dot{W}(t) = -\text{proj}(W, \Gamma_W\sigma(z(t))e^T(t)PB), \tag{9}$$

where Γ_W is a positive definite learning rate matrix.

Theorem 1 Let the source and transfer systems be given by (1),(2), and assume that the source controller satisfies assumption 1. Assume that the desired acceleration for the transfer system is given by 5, with the adaptive element ν_{ad} being the output of an RBFN (7), the weights of the RBF are updated using the projection based adaptive law in 9, $D_x \subset D$, and Assumption 2 is satisfied. Then the control input δ_t obtained using the source controller $g(.)$ with the modified desired acceleration $\ddot{x}_{\text{des}}$: $\delta_t = g(x_t, \dot{x}_t, \ddot{x}_{\text{des}})$, guarantees that the transfer system's states are uniformly ultimately bounded around the desired trajectory x_{ref}.

Proof: Let $\tilde{W} = W - W^*$ Choose a positive definite quadratic Lyapunov candidate $V(e, \tilde{W}) = \frac{1}{2}(e^TPe + \text{tr}(\tilde{W}^T\Gamma_W^{-1}\tilde{W}))$. The Lie derivative of the Lyapunov candidate is

$$\dot{V}(e, \tilde{W}) = -e^TQe + e^TPB(\nu_ad - \Delta) + tr(\tilde{W}^T\Gamma_W^{-1}\dot{W}). \tag{10}$$

Substituting the adaptive law (9), and noting that since $D_x \subset D$ (8) is valid, results in

$$\dot{V}(e, \tilde{W}) \le -\lambda_{\min}(Q)\|e\|^2 + \|e\|\|PB\|\|\tilde{\epsilon}(z_t)\|, \tag{11}$$

where $\lambda_{\min}(Q)$ is the minimum eigenvalue of Q. Therefore, the Lyapunov candidate is negative definite outside of a compact set, furthermore, the adaptive weights W are bounded due to the use of the projection operator (see [34]). Hence $(e, \tilde{W})$ are uniformly ultimately bounded. ■

The above theorem however, does not guarantee that the adaptive weights will approach and stay bounded within a compact domain of the ideal weights W^* as given by (7) unless the system's states are persistently excited [31,32]. Furthermore, it is difficult to guarantee a good response in the transient phase, potentially leading to unsafe adaptation. Safe envelop exploration techniques similar to those experimented with in [**?**] can be useful in this case. On the other hand, it was shown in [35,36] that for linearly parameterized uncertainties the requirement on persistency of excitation can be relaxed if online recorded data is used concurrently with instantaneous data for adaptation. It was further shown that a concurrent learning adaptive controller guarantees exponential tracking error and weight error convergence, thereby providing bounded response during the learning transient. In particular, for a linearly parameterized representations of the uncertainty, the following theorem can be proven [35]–[37]

Theorem 2 Let the source and transfer systems be given by (1),(2), and assume that the source controller satisfies assumption 1. Assume that the desired acceleration for the transfer system is given by 5, with the adaptive element ν_{ad} being the output of an RBFN (7). For each online recorded data point i, let $\epsilon_i(t) = W^T(t)\phi(x_i, \delta_i) - \hat{\Delta}(x_i, \delta_i)$, with $\hat{\Delta}(x_i, \delta_i) = \dot{\hat{x}}_i - \nu(x_i, \delta_i)$, where $\dot{\hat{x}}_i$ is the bounded estimate of $\dot{x}_i$ and let the weight update be given by the following concurrent learning adaptive law:

$$\dot{W} = -\Gamma_W\sigma(z)e^TPB - \frac{1}{p}\sum_{j=1}^{p}\Gamma_{W_b}\sigma(x_i, \delta_i)\epsilon_j^T, \tag{12}$$

with Γ_{W_b} the learning rate for training on online recorded data. Assume that $Z = [\phi(z_1), ..., \phi(z_p)]$ and $rank(Z) = l$, $D_x \subset D$, and Assumption 2 is satisfied. Then the control input δ_t obtained using the source controller $g(.)$ with the modified desired acceleration $\ddot{x}_{\text{des}}$: $\delta_t = g(x_t, \dot{x}_t, \ddot{x}_{\text{des}})$, guarantees that the transfer system's tracking error e and the weight error $\tilde{W} = W - W^*$ is exponentially uniformly ultimately bounded and converge exponentially fast to a compact ball around the origin with the rate of convergence directly proportional to the minimum singular value of the history stack matrix Z.

The proof of this theorem is avoided for brevity, it can be formulated by using arguments similar to that of Theorem 1 and the proof of the theorems in [35]–[37].

Remark 1 The size of the compact ball around the origin where the weight and tracking error converge is dependent on the representation error $\bar{\epsilon}$ and the estimation error $\breve{\epsilon} = \max_i\|\dot{x}_i - \dot{\hat{x}}_i\|$. The former can be reduced by choosing an appropriate number of RBFs across the operating domain, and the latter can be reduced by an appropriate implementation of a fixed point smoother. A fixed point smoother uses

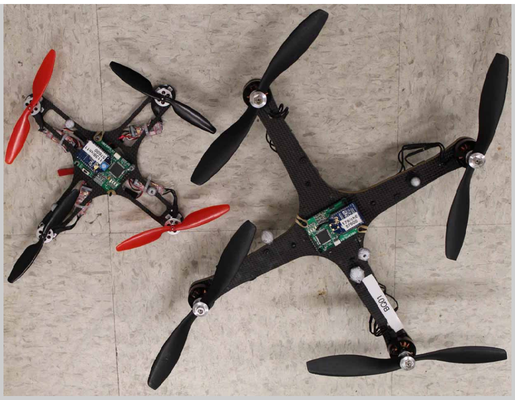

Fig. 1. Two MIT quadrotors equipped to fly in the ACL Real Time Indoor Autonomous Vehicle Test Environment (RAVEN) [40]. The baseline controller on both quadrotors is PID. The gains have been tuned for the bigger quadrotor. The small quadrotor uses gains from the bigger one, resulting in relatively poor trajectory tracking performance.

data before and after a data point is recorded to form very accurate estimates of $\dot{\hat{x}}_i$ using a forward-backward Kalman filter [16,38]. Note that $\dot{\hat{x}}(t)$ is not needed at a current instant t. Therefore, an appropriate implementation of a fixed point smoother alleviates the time-delay often observed in estimating $\dot{\hat{x}}(t)$ with forward Kalman filter (or a low pass filter) only.

Remark 2 The main limitation of the linearly parameterized RBFN adaptive element is that the RBF centers need to be preallotted over an estimated compact domain of operation D, and $D_x \subset D$. However, if D_x, the operating domain of the transfer system, is not known a-priori, then the stability guarantees provided by the above adaptive law may not hold. This can be addressed by evolving the RBF basis to reflect the current domain of operation, a Reproducing Kernel Hilbert Space (RKHS) approach for accomplishing this was presented by Kingravi et al. in [18]. Kingravi et al. presented the Budgeted Kernel Restructuring - Concurrent Learning (BKR-CL) algorithm for adaptive control which uses a linear independence test in the associated RKHS to determine when to add/remove centers without exceeding a predefined budget of allowable center-dictionary size. The BKR-CL approach can be used for control transfer through a formulation similar to that in Theorem 2.

III. Flight Test Results on MIT Quadrotors

The Aerospace Controls Laboratory (ACL) at MIT maintains the Real Time Indoor Autonomous Vehicle Test Environment (RAVEN) [39,40]. RAVEN uses a motion capture system (VICON http://www.vicon.com/) to obtain accurate estimates of the position and attitude of autonomous vehicles. The quadrotors shown in Figure 1 were developed in-house and are equipped to fly within the RAVEN environment. A detailed description of the PID-based baseline control architecture and the corresponding software infrastructure of RAVEN is in [5,41].

A. Hardware Details

The flight experiments in this paper were performed on the smaller of the two quadrotors shown in Figure 1. This vehicle weighs 96 grams without the battery and measures 18.8 cm from motor to motor. The larger quadrotor weighs 316 grams and measures 33.3 cm from motor to motor. Both quadrotors utilize standard hobby brushless motors, speed controllers, and fixed-pitch propellers, all mounted to custom-milled carbon fiber frames. Onboard attitude control is performed on a custom autopilot, with attitude commands being calculated at 1 kHz. Due to limitations of the speed controllers, the smaller quadrotor motors only accept motor updates at 490 Hz. More details on the autopilot and attitude control are in [41].

The large differences in size and mass between the two quadrotors results in relatively poor position and velocity tracking with the small quadrotor when it uses the gains from the bigger quadrotor. In particular, the lack of knowledge of the mapping from unit-less speed controller commands to propeller thrust for the small quadrotor requires significant input from the altitude integral PID term and leads to poor vertical and horizontal position tracking, as is shown in Section III-C.

B. Augmentation of baseline linear control with adaptation

The source control mapping $g(.)$ is described in [5] and was found to satisfy assumptions stated in Section II.

In the results presented here, the control-transfer is performed only on the position and velocity controller (outer-loop) using methods described in Section II, while the inner-loop quaternion-based attitude controller is left unchanged. This introduces further uncertainty due to ill-tuned inner-loop that must be accounted for by the control-transfer techniques. In the following discussion, baseline PID refers to the outer-loop PID controller on the smaller quadrotors with gains directly transferred from the larger (well-tuned) quadrotor. MRAC refers to augmentation of the baseline law with RBFN adaptive law of (9), CL-MRAC refers to Concurrent Learning - MRAC of Theorem 2, and BKR-CL refer to the Budgeted Kernel Restructuring CL adaptive control algorithm. The limit of the projection operator (200) was not reached in any of the flights.

The controllers were separated into three different loops corresponding to x, y, z positions. The input to the RBF (7) for the x loop was $z_t = [x, \dot{x}, \vec{q}]$, y loop was $z_t = [y, \dot{y}, \vec{q}]$, and z loop was $z_t = [z, \dot{z}, \vec{q}]$, where $\vec{q}$ is the attitude quaternion. The separation of the position loops is motivated by the symmetric nature of the quadrotor flight platform. Note, however, that the controller adapts on the attitude quaternion for all three loops. This presents the controller with sufficient information to account for attitude based couplings.

C. Flight-Test results

1) Figure 8 maneuvers: The quadrotor performed five sets of three identical “figure 8” maneuvers with a pause of 0.2 s in between. Several different values of learning rates were analyzed through simulation and preliminary flight testing. The results presented here correspond to values that resulted in good performance without over-learning. The best initial MRAC learning was found to be $\Gamma_W = 2$, and the learning rate of the adaptive law that trains on recorded data was found to be $\Gamma_{W_b} = 0.5$. Theoretically, MRAC and CL-MRAC learning rates can be kept constant in adaptive control, as suggested by deterministic Lyapunov based stability analysis. However, classical stochastic stability results (e.g. [42]) indicate that driving the learning rates to zero is required for guaranteeing convergence in presence of noise. The problem with this approach is that the effect of adaptation would be eventually removed. Therefore, in practice, the learning rate is set to decay to a small constant to avoid unstable or oscillatory behaviors. Learning rates for MRAC and CL-MRAC were decayed by dividing it by 1.5 for Γ_W and 2 for Γ_{W_b} after each set of three figure 8 maneuvers. The decay limit of these learning rates are $\Gamma_W = 0.5$ and $\Gamma_{W_b} = 0.001$. For BKR-CL, Γ_W decays from 0.5 to 0.02 and Γ_{W_b} decays from 0.5 to 0.001.

For MRAC and CL-MRAC 50 RBF centers were generated using a uniform random distribution over a domain where the states were expected to evolve. The centers for the position and velocity for the x,y axes were spread across $[-2, 2]$. For the vertical z axis, the position and velocity were spread across $[-0.5, 0.5]$ and $[-0.6, 0.6]$ respectively. The centers for quaternions were placed within $[-1, 1]$. The bandwidth for each radial basis function is set to $\mu_i = 1$. For CL, the last-point-difference technique was used to record all data points that satisfy $\frac{\|\sigma(x(t))-\sigma(x_k)\|}{\sigma(x(t))} \geq 0.01$, where k is the index of the last recorded point [37]. The size of the cyclic history stack was set to 51. For BKR-CL, an a-priori assignment of centers is not required. The budget for the online generated dictionary of centers was set to 20 (see [18]). The center selection tolerance was set to $\epsilon_{tol} = 1 \times 10^{-6}$. The history-stack for BKR-CL was updated using the same algorithm as that for CL-MRAC.

The plots of the quadrotor's flight performances with baseline PID, MRAC and CL-MRAC are available in [26], they are not presented here due to space limitations. The baseline outer-loop PID controller had poor position tracking performance, in particular, the x and y positions were more than 30 cm off from the maximum and minimum commanded position. Even though MRAC shows improved tracking performance over time, it was not able to track the extremities of the reference command. Furthermore, increasing the learning-rate did not seem to help and often resulted in oscillatory behavior. Figure 2 shows the trajectory in x, y and z of the BKR-CL controller. It can be seen that BKR-CL's position tracking performance improves over time. This is a result of the BKR-CL algorithm *learning* a representation of the error $\Delta()$, unlike simply reactively suppressing it like MRAC. The learning process is better illustrated by Figure 3 which shows that the adaptive element's ability to capture the modeling uncertainty improves over time as the algorithm learns the modeling error. This indicates that BKR-CL is indeed driving the weights and adding RBFs online to best capture the uncertainty. Furthermore, the good tracking performance of CL-MRAC is in agreement with theoretical and experimental results previously presented in [16,17,36] on different UAV platforms. Note however that those results were not concerned with the control-transfer problem.

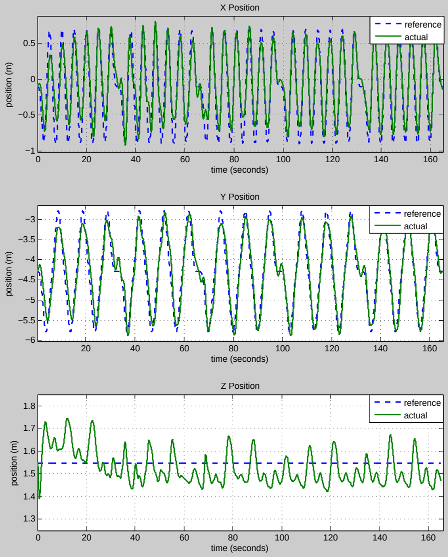

Fig. 2. Trajectory tracking performance of BKR-CL shows significant improvement over the baseline PID and MRAC controller. BKR-CL can be seen to learn to track the position better over time.

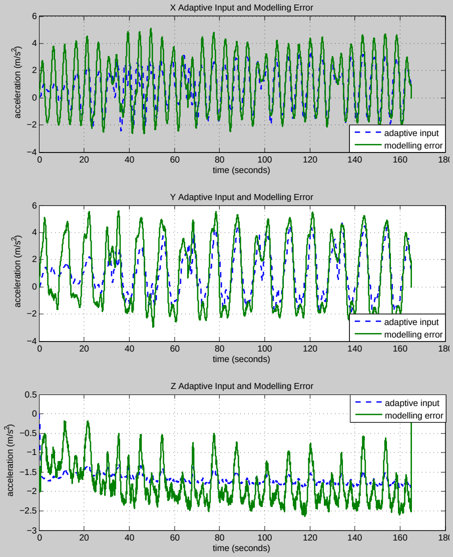

Fig. 3. The estimated modeling error is compared with the output of RBFN adaptive element when using BKR-CL. It is seen that BKR-CL finds the correct weights and the RBF bases online required to learn the modeling uncertainty over time in x and y axes, however, the z axis performance suffers possibly due to unmodeled motor dynamics.

Figure 4 plots the RMS error per set of three maneuvers in all three dimensions of the vehicle's trajectories for baseline PID, MRAC, CL-MRAC and BKR-CL. In x and y, BKR-CL and CL-MRAC showed significant improvement over PID and MRAC as their RMS errors was decreasing over time. Despite having similar performances, BKR-CL has an advantage over CL-MRAC because its centers can be selected online instead of being pre-allocated manually. In z axis however, BKR-CL had higher RMS error than the other three controllers. One reason for this could be unmodeled uncertainties or time delays in the z axis. Another reason could be that because the position command in the vertical direction was constant, BKR-CL selected RBFs close to one another. This may have caused the adaptive controller's

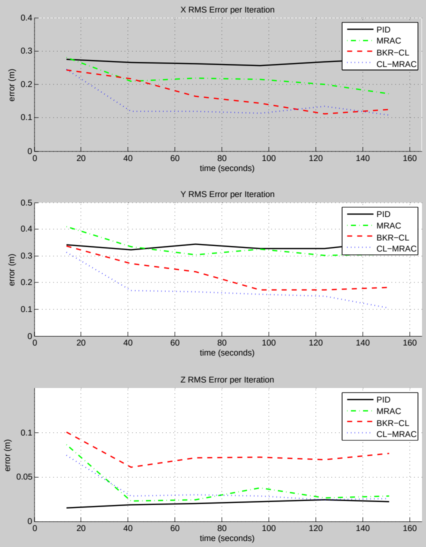

Fig. 4. RMS error in x, y, and z while tracking figure 8 maneuvers. MRAC, and CL-MRAC control transfer techniques outperform the baseline PID transferred directly from the source controller in x and y axes. The BKR-CL performance in z axis is poor. However, the BKR-CL controller needs very little domain knowledge and assigns RBF centers online.

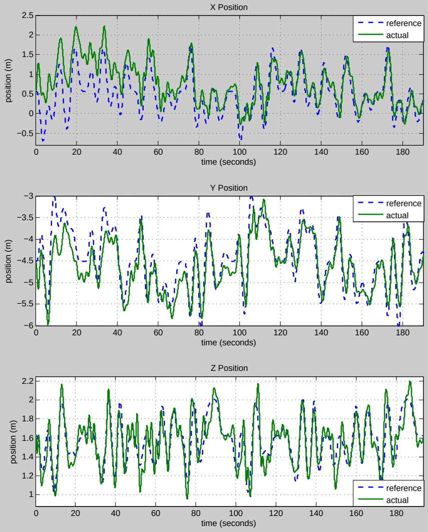

Fig. 5. Un-tuned PID controller's trajectory in x, y and z is compared with the reference command. The quad-rotor was unable to track the command closely in x and y in the first 100 seconds, indicating that the integral term was slow in becoming effective.

learning to be less effective than MRAC and CL-MRAC. Future work will involve training the vehicle on a larger state-space than the commanded position to render better learning performance in the z direction. The video accompanying this paper depicts a flight-test scenario comparing control transfer with CL-MRAC and MRAC.

2) Random waypoints with PID: In this set of maneuvers, the UAV tracks a set of eight trajectories generated from random waypoints created using the spline based trajectory generator [5]. The last four sets are a repetition of the first four and are used for evaluating long-term learning. The purpose of executing random paths is to validate that control-transfer techniques over generic paths representative of the UAV's typical operation.

The parameters of the different controllers for this set of maneuvers are similar to the Figure 8 case. Additionally, the PID controller's integrator term was manually tuned to render the smallest RMS errors in the x, y and z directions. This PID controller will be referred to as the tuned PID as opposed to the un-tuned PID (directly imported source controller); although it should be noted that the PD gains were not tuned because the adaptive control formulation in Section II assumes the same PD gains for both the augmented adaptive controller and the linear PD controller. The goal of this effort was to compare the adaptive control-transfer approach with a manual tuning of gains.

Figure 5 clearly illustrates the discrepancy between source PID's position and the reference command. This controller took over 100 seconds before the integral term became effective and the vehicle began to track the position in x accurately. On the other hand, with the tuned PID, the initial tracking performance of the vehicle was good. However, neither PID controllers could track sharp changes in the position, indicating the presence of nonlinearities not well handled by the linear PID control law. CL-MRAC on the other hand tracked the commanded trajectory better, and was able to capture most sudden changes. MRAC's tracking performance was comparable to tuned PID but remained inferior to CL-MRAC. Furthermore, MRAC's performance did not improve over time, as is visible through Figure 8, which shows that MRAC adaptive output does not learn the model uncertainty $\Delta(.)$. Furthermore, increasing MRAC learning rate did not seem to help. Figure 9 shows that CL-MRAC once again was able to demonstrate long-term learning as characterized by better ability of the adaptive element to predict and correct for the uncertainty $\Delta(.)$. Figure 10 compares the RMS performance of all the controllers discussed above for this set of maneuvers. It can be seen that CL-MRAC based control-transfer approach outperforms all other approaches, clearly indicating the power of learning based control-transfer approaches.

In summary, these results indicate that adaptation can be used to improve tracking performance of UAVs whose controllers have been transferred from other UAVs with similar control assignment but different inertial and force generation properties. Furthermore, these results together confirm that the long term learning ability in adaptive controllers such as CL-MRAC and BKR-CL aids in improving performance of the transferred controller. Furthermore, to the best of our knowledge, the BKR-CL results presented in this paper represent the first time a budgeted nonparametric adaptive control approach has been tested in flight.

IV. Conclusion

The problem of control-transfer was defined as the task of transferring controllers from one UAV to another without sacrificing tracking performance. A rigorous formulation using dynamical systems theory was developed, and it was shown that ideas from adaptive control and feedback linearization could be used for transferring controllers between

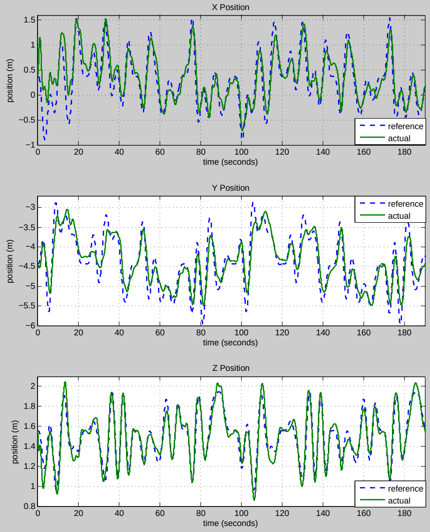

Fig. 6. Tuned PID controller showed significant improvement over un-tuned PID. It was able to track the position command better within 20 seconds from beginning of the flight. However, it could not track sharp changes in position commands.

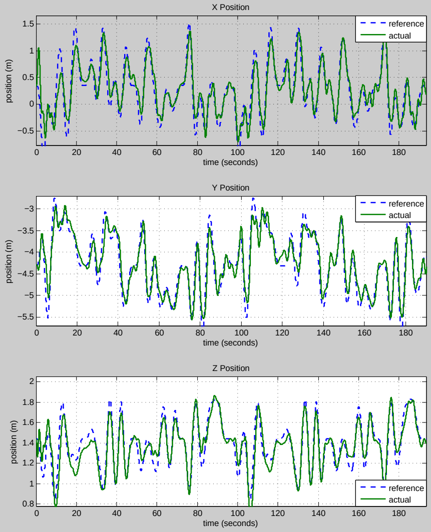

Fig. 7. CL-MRAC demonstrated long-term learning as characterized by better tracking over time. Furthermore, it could track most sharp changes in the position command.

systems that have "similar" control structure. Existence of an isomorphic control law on the source system that can be inverted w.r.t. the control input of the transfer system, and the same sign of control effectiveness on both source and transfer systems were shown to guarantee similarity. Three adaptive control-transfer laws were evaluated, including two learning based MRAC laws: Concurrent Learning MRAC, and Budgeted Kernel Restructuring CL (BKR-CL). These results indicate the feasibility of transferring controllers between UAVs using learning-based control.

V. ACKNOWLEDGMENTS

The authors acknowledge Boeing Research & Technology for support of the RAVEN [39,40] indoor flight facility in which the flight experiments were conducted.

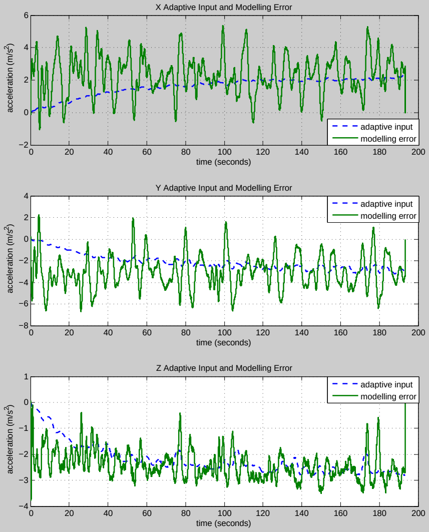

Fig. 8. MRAC's adaptive output is compared against the estimated model error. It can be seen that MRAC does not learn the modeling uncertainty.

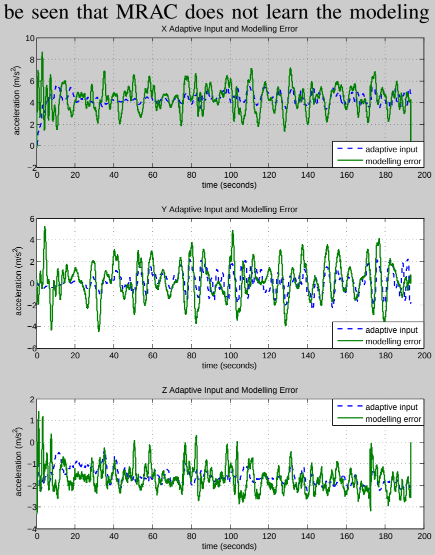

Fig. 9. CL-MRAC's ability to learn the transfer system modeling uncertainty over the long-term is characterized by improvement in its ability to predict the uncertainty resulted due to control transfer.

REFERENCES

[1] S. Bouabdallah, A. Noth, and S. R., "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[2] J. N. Portlock and S. N. Cubero, "Dynamics and control of a vtol quad-thrust aerial robot," in *Mechatronics and Machine Vision in Practice* (J. Billingsley and R. Bradbeer, eds.), pp. 27–40, Springer Berlin Heidelberg, 2008. 10.1007 978-3-540-74027-83.

[3] W. Guo and J. Horn, "Modeling and simulation for the development of a quad-rotor UAV capable of indoor flight," in *Modeling and Simulation Technologies Conference and Exhibit*, 2006.

[4] B. Michini, J. Redding, N. K. Ure, M. Cutler, and J. P. How, "Design and flight testing of an autonomous variable-pitch quadrotor," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2978 – 2979, IEEE, May 2011.

[5] M. Cutler and J. P. How, "Actuator constrained trajectory generation and control for variable-pitch quadrotors," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, (Minneapolis, Minnesota), August 2012.

[6] H. Huang, G. Hoffmann, S. Waslander, and C. Tomlin, "Aerodynam-

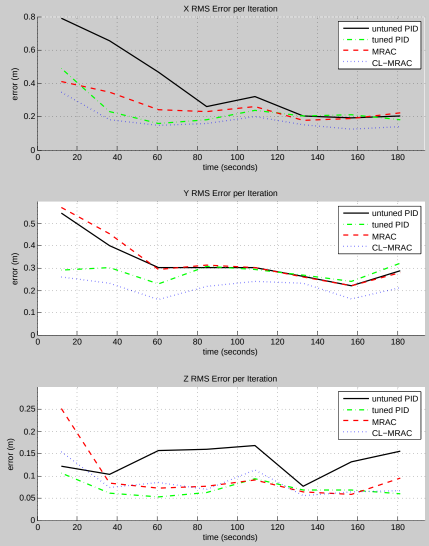

Fig. 10. The Root Mean Square is plotted as a function of time. Performance of MRAC and well tuned PID are comparable towards the end. Furthermore, CL-MRAC has the lowest error when performing random maneuvers, clearly highlighting the benefit of learning.

ics and control of autonomous quadrotor helicopters in aggressive maneuvering," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3277 –3282, May 2009.

[7] S. Lupashin, A. Schollig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1642–1648, IEEE, 2010.

[8] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-UAV testbed," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[9] A. Calise, N. Hovakimyan, and M. Idan, "Adaptive output feedback control of nonlinear systems using neural networks," *Automatica*, vol. 37, no. 8, pp. 1201–1211, 2001. Special Issue on Neural Networks for Feedback Control.

[10] E. Johnson and S. Kannan, "Adaptive trajectory control for autonomous helicopters," *Journal of Guidance Control and Dynamics*, vol. 28, pp. 524–538, May 2005.

[11] C. Cao and N. Hovakimyan, "Design and analysis of a novel adaptive control architecture with guaranteed transient performance," *Automatic Control, IEEE Transactions on*, vol. 53, pp. 586 –591, march 2008.

[12] E. Lavretsky and K. Wise, "Flight control of manned/unmanned military aircraft," in *Proceedings of American Control Conference*, 2005.

[13] N. Nguyen, K. Krishnakumar, J. Kaneshige, and P. Nespeca, "Dynamics and adaptive control for stability recovery of damaged asymmetric aircraft," in *AIAA Guidance Navigation and Control Conference*, (Keystone, CO), 2006.

[14] M. Steinberg, "Historical overview of research in reconfigurable flight control," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 219, no. 4, pp. 263–275, 2005.

[15] D. B. Jourdan, M. D. Piedmonte, V. Gavrilets, and D. W. Vos, *Enhancing UAV Survivability Through Damage Tolerant Control*, pp. 1–26. No. August, AIAA, 2010. AIAA-2010-7548.

[16] G. Chowdhary and E. N. Johnson, "Theory and flight test validation of a concurrent learning adaptive controller," *Journal of Guidance Control and Dynamics*, vol. 34, pp. 592–607, March 2011.

[17] G. Chowdhary and E. N. Johnson, "Concurrent learning for improved parameter convergence in adaptive control," in *AIAA Guidance Navigation and Control Conference*, (Toronto, Canada), 2010.

[18] H. A. Kingravi, G. Chowdhary, P. A. Vela, and E. N. Johnson, "Reproducing kernel hilbert space approach for the online update of radial bases in neuro-adaptive control," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, pp. 1130 –1141, july 2012.

[19] E. Johnson, M. Turbe, A. Wu, and S. Kannan, "Flight results of autonomous fixed-wing uav transitions to and from stationary hover," in *Proceedings of the AIAA GNC Conference*, August 2006.

[20] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, no. 1, pp. 1633–1685, 2009.

[21] M. E. Taylor and P. Stone, "An introduction to inter-task transfer for reinforcement learning," *AI Magazine*, vol. 32, no. 1, pp. 15–34, 2011.

[22] G. Konidaris, I. Scheidwasser, and A. G. Barto, "Transfer in reinforcement learning via shared features," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1333–1371, 2012.

[23] A. Lazaric and M. Restelli, "Transfer from multiple mdps," in *NIPS*, pp. 1746–1754, 2011.

[24] A. Wilson, A. Fern, and P. Tadepalli, "Transfer learning in sequential decision problems: A hierarchical bayesian approach," in *JMLR Workshop and Conference Proceedings*, pp. 217–227, 2012.

[25] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 19–22, 1992.

[26] G. Chowdhary, T. Wu, M. Cutler, N. K. Üre, and J. How, "Experimental results of concurrent learning adaptive controller," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, (Minneapolis, MN), AIAA, August 2012. Invited.

[27] J. P. How, E. Frazzoli, and G. Chowdhary, *Handbook of Unmanned Aerial Vehicles*, ch. Linear Flight Contol Techniques for Unmanned Aerial Vehicles. Springer, 2012 (to appear).

[28] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[29] N. Kim, *Improved Methods in Neural Network Based Adaptive Output Feedback Control, with Applications to Flight Control*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2003.

[30] T. Zhang, S. Ge, and C. Hang, "Direct adaptive control of non-affine nonlinear systems using multilayer neural networks," in *American Control Conference, 1998. Proceedings of the 1998*, vol. 1, pp. 515 –519 vol.1, jun 1998.

[31] R. Sanner and J.-J. Slotine, "Gaussian networks for direct adaptive control," *Neural Networks, IEEE Transactions on*, vol. 3, pp. 837 –863, nov 1992.

[32] K. Narendra, "Neural networks for control theory and practice," *Proceedings of the IEEE*, vol. 84, pp. 1385 –1406, oct 1996.

[33] J. Park and I. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computatations*, vol. 3, pp. 246–257, 1991.

[34] J.-B. Pomet and L. Praly, "Adaptive nonlinear regulation: estimation from the lyapunov equation," *Automatic Control, IEEE Transactions on*, vol. 37, pp. 729 –740, jun 1992.

[35] G. Chowdhary, *Concurrent Learning for Convergence in Adaptive Control Without Persistency of Excitation*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2010.

[36] G. Chowdhary and E. N. Johnson, "Concurrent learning for convergence in adaptive control without persistency of excitation," in *49th IEEE Conference on Decision and Control*, pp. 3674–3679, 2010.

[37] G. Chowdhary and E. N. Johnson, "A singular value maximizing data recording algorithm for concurrent learning," in *American Control Conference*, (San Francisco, CA), June 2011.

[38] A. Gelb, *Applied Optimal Estimation*. Cambridge: MIT Press, 1974.

[39] M. Valenti, D. Dale, J. How, and J. Vian, "Mission health management for 24/7 persistent surveillance operations," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, (Myrtle Beach, SC), August 2007.

[40] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Systems Magazine*, vol. 28, pp. 51–64, April 2008.

[41] M. Cutler, "Design and Control of an Autonomous Variable-Pitch Quadrotor Helicopter," Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, August 2012.

[42] L. Ljung, "Analysis of recursive stochastic algorithms," *Automatic Control, IEEE Transactions on*, vol. 22, pp. 551 – 575, aug 1977.