Experimental Results of Concurrent Learning Adaptive Controllers

Girish Chowdhary[*], Tongbin Wu[†], Mark Cutler[‡], Nazim Kemal Üre[§], Jonathan P. How[¶]

Commonly used Proportional-Integral-Derivative based UAV flight controllers are often seen to provide adequate trajectory-tracking performance 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 suggests the use of adaptive controllers in speeding up the process of extracting good control performance from new UAVs. In particular, it is shown that a concurrent learning adaptive controller improves the trajectory tracking performance of a quadrotor with baseline linear controller directly imported from another quadrotors whose inertial characteristics and throttle mapping are very different. Concurrent learning adaptive control uses specifically selected and online recorded data concurrently with instantaneous data and is capable of guaranteeing tracking error and weight error convergence without requiring persistency of excitation. Flight-test results are presented on indoor quadrotor platforms operated in MIT's RAVEN environment. These results indicate the feasibility of rapidly developing high-performance UAV controllers by using adaptive control to augment a controller transferred from another UAV with similar control assignment structure.

I. Introduction

Due to their simplicity and reliability, Proportional-Integral-Derivative (PID) based controllers are often used for developing Unmanned Aerial Vehicle (UAV) autopilots (see e.g. [1, 2, 3, 4, 5]). Well tuned PID controllers have been shown to yield excellent flight performance on quadrotors in various indoor environments. Notably, they have been used to perform aggressive maneuvers by various groups.[6,7,8,5] 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, transferring controllers from one UAV to other becomes time-intensive. This paper demonstrates that the use of adaptive controllers can speed up the process of extracting good control performance from new UAVs on which controllers from UAVs with similar control assignment structure are transferred. In particular, it is shown that a PD controller augmented with a concurrent learning adaptive controller improves the trajectory tracking performance of a smaller quadrotor whose controller has been directly imported (without changing gains) from another quadrotor whose inertial characteristics and throttle mapping are very different. The flight-test results are presented on indoor quadrotor platforms operated in MIT's RAVEN environment. These results serve two purposes, firstly they indicate the feasibility of using adaptive control to rapidly develop high-performance UAV controllers by adapting controllers from one UAV to another with similar control assignment structure, and secondly they validate concurrent learning adaptive control on a different platform.

Often, it is difficult or costly to obtain an exact model of system dynamics. Adaptive control is one promising method that has been used for mitigating modeling error in control design. Particularity, adaptive flight control has been widely studied. For example, Calise,[9] Johnson,[10,11,12] Kannan[13,14] and others have developed model reference adaptive controllers for both fixed wing and rotary wing UAVs. Cao, Yang, Hovaykiman, and other have developed the L_1 adaptive control method.[15,16] Lavretsky,[17] Nguyen,[18] Steinberg,[19] Jourdan et al.[20] and others have extended direct adaptive control methods to fault tolerant control and developed techniques in composite/hybrid adaptation.

[*] Postdoctoral Associate, Massachusetts Institute of Technology (MIT), MA, girishc@mit.edu

[†] Visiting Student, MIT

[‡] Graduate Research Assistant, Massachusetts Institute of Technology, MA

[§] Graduate Research Assistant, Massachusetts Institute of Technology, MA

[¶] Richard C. Maclaurin Professor of Aeronautics and Astronautics, Massachusetts Institute of Technology, MA, jhow@mit.edu

Chowdhary and Johnson recently developed concurrent learning adaptive controllers, which use online selected and recorded data concurrently with current data for adaptation. They have shown that concurrent learning adaptive controllers can guarantee exponential closed loop stability without requiring persistency of excitation if the plant uncertainty can be parameterized using a set of known basis functions.[21,22] Furthermore, it was shown that when the structure of the uncertainty is unknown, concurrent learning neuro-adaptive controllers can guarantee uniform ultimate boundedness of the tracking error in a neighborhood of zero and the uniform ultimate boundedness of the neural network weights in a compact neighborhood of the ideal weights.[21,23] It was also shown that the rate of convergence is directly proportional to the minimum singular value of the history stack and a singular value maximizing data recording algorithm was also presented.[24] Concurrent learning controllers have also been extended to multivariable uncertain linear dynamical systems.[25]

In [4,5] Michini et al. and Cutler and How present the details of a trajectory generation algorithm and a PID based controller for a quadrotor built and flown at MIT's Aerospace Controls Laboratory (ACL). Their results show that with a well tuned PID, excellent flight performance can be extracted from the quadrotor platform operating indoors. A relatively smaller sized quadrotor has since been developed at ACL. The controller from the larger quadrotor (described in [4,5]) has been directly imported (without changing the gains) on the smaller quadrotor, and has been shown to enable hovering flight. However, the throttle mapping of the smaller quadrotor is not known, and neither are the gains tuned to suite the smaller quadrotor. Therefore, the trajectory tracking performance of the smaller quadrotor with the imported controller is relatively poor. In this paper, the baseline controller on the smaller quadrotor is augmented with a concurrent learning adaptive controller and it is demonstrated through flight testing that the concurrent learning controller significantly improves the trajectory tracking performance over time. The flight tests are performed in ACL's RAVEN indoor flight facility.[26]

Previously, concurrent learning adaptive controllers have been validated on an outdoor rotorcraft platform,[27] and on a fixed wing outdoor UAV.[23] In both these cases the baseline PD linear controller was well-tuned, and concurrent learning was shown to improve the performance. In this paper, concurrent learning adaptive controllers are validated on small-sized quadrotor aircraft for which the baseline PD controller is not well-tuned. This could be viewed as a more challenging situation for the adaptive controller, because the control structure does not leverage the robustness of a well-tuned baseline linear controller.

II. Concurrent Learning Model Reference Adaptive Control

The Approximate Model Inversion base MRAC (AMI-MRAC) architecture employed for adaptive control is discussed in this section. Let $D_x \subset \Re^n$ be compact, and Let $x(t)$ be the known state vector, let $\delta \in \Re^n$ denote the control input. Note that it is assumed that the dimension of the control input is same as the dimension of $\ddot{x}$. This is a reasonable assumption for outer-loop control of a rotorcraft, where the control required to achieve a desired position can be translated to roll angle, pitch angle, and the throttle input (it is assumed that heading control is independent of position control, which is a reasonable assumption for symmetric rotorcraft such as quadrotors). In context of MRAC, this assumption can be viewed as a specific matching condition. Consider the following system that describes the dynamics of an aircraft:

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

In the above equation, the function f, $f(0,0,0) = 0$, is assumed to be Lipschitz continuous in $x, \dot{x} \in D_x$, and control input δ is assumed to belong to the set of admissible control inputs consisting of measurable bounded functions. Therefore, existence and uniqueness of piecewise solutions to (1) are guaranteed. In addition, a condition on controllability must be assumed. These assumption are typically satisfied by most aircraft platforms, including quadrotors.

The goal of the AMI-MRAC controller is to track the states of a reference model given by:

$$\ddot{x}_{rm} = f_{rm}(x_{rm}, \dot{x}_{rm}, r), \tag{2}$$

where f_{rm} denotes the reference model dynamics which is assumed to be continuously differentiable in $x_{rm}, \dot{x}_{rm}$ for all $x_{rm}, \dot{x}_{rm} \in D_x \subset \Re^n$. The command $r(t)$ is assumed to be bounded and piecewise continuous. Furthermore, it is assumed that the reference model is chosen such that it has bounded outputs $(x_{rm}, \dot{x}_{rm})$ for a bounded input $r(t)$. Often a linear reference model is desirable, however the theory allows for a general nonlinear reference model (as is the case in the results presented in Section III) as long as it is bounded-input-bounded-output stable. It has been shown by Kannan that when the actuators are subjected to saturation, a nonlinear reference model may be required.[14]

Since the exact model (1) is usually not available, an approximate model $\hat{f}(x,\dot{x},\delta)$ which is invertible with respect to δ is chosen such that

$$\delta = \hat{f}^{-1}(x,\dot{x},\nu), \tag{3}$$

where $\nu \in \Re^n$ is the pseudo control input representing the desired acceleration. Note that the choice of the inversion model should capture the control assignment structure, however the parameter of that mapping need not be completely known. This is important in guaranteeing that appropriate control inputs are mapped to appropriate states. For example, the inversion model should capture the fact that pitching of the quadrotor results in motion along the x direction, although the exact parameters of this relationship need not be known. It is assumed that for every $(x,\dot{x},\nu)$ the chosen inversion model returns a unique δ. This can be realized if the dimension of the input δ is the same as the dimension of $\ddot{x}$. In many control problems, and indeed in the quadrotor position control problem studied here, this is true. Hence, the pseudo control input satisfies

$$\nu = \hat{f}(x,\dot{x},\delta). \tag{4}$$

This approximation results in a model error of the form

$$\ddot{x} = \nu + \Delta(x,\dot{x},\delta) \tag{5}$$

where the model error $\Delta : \Re^{2n+k} \to \Re^n$ is given by

$$\Delta(x,\dot{x},\delta) = f(x,\dot{x},\delta) - \hat{f}(x,\dot{x},\delta). \tag{6}$$

Define the tracking error $e \in \Re^{2n}$ as $e(t) = [x_{rm}(t) - x(t)]^T$, and $\dot{e}(t) = [\dot{x}_{rm}(t) - \dot{x}(t)]^T$. A tracking control law consisting of a linear feedback part $\nu_{pd} = -[K_p\ K_d]\begin{pmatrix} e \\ \dot{e} \end{pmatrix}$, a linear feedforward part $\nu_{rm} = \ddot{x}_{rm}$, and an adaptive part $\nu_{ad}(x)$ is chosen to have the following form

$$\nu = \nu_{rm} + \nu_{pd} - \nu_{ad}. \tag{7}$$

To derive the equation for tracking error dynamics note that $\ddot{e} = \ddot{x}_{rm} - \ddot{x} = \nu_{rm} - (\nu + \Delta)$ due to (6). Substituting (7) we see that $\ddot{e} = -\nu_{pd} + (\nu_{ad} - \Delta)$ and then, letting $A = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix}$, and $B = \begin{bmatrix} 0 \\ I \end{bmatrix}$, the tracking error dynamics are found to be[12, 14, 21, 27]

$$\dot{e} = Ae + B[\nu_{ad}(x,\dot{x},\delta) - \Delta(x,\dot{x},\delta)]. \tag{8}$$

The baseline full state feedback controller gains K_p, K_d are chosen such that A is a Hurwitz matrix. Hence for any positive definite matrix $Q \in \Re^{2n\times 2n}$, a unique positive definite solution $P \in \Re^{2n\times 2n}$ exists to the Lyapunov equation

$$0 = A^T P + PA + Q. \tag{9}$$

Letting $z = [x,\dot{x},\delta] \in \Re^{2n+k}$, it is assumed that the uncertainty $\Delta(z)$ can be approximated using a Single Hidden Layer (SHL) Neural Network (NN) or a Radial Basis Function (RBF) NN. Here we concentrate on the RBF-NN case.

RBF-NN have been widely used to represent uncertainties whose basis is not known, but it is known that the uncertainty $\Delta(z)$ is continuous and defined over a compact domain $D \subset \mathbb{R}^{n+l}$.[28, 29, 30] One reason RBF-NN have received significant popularity is because of their linear-in-parameters structure. When using RBF-NN, the adaptive element is represented by

$$\nu_{ad}(z) = W^T\sigma(z), \tag{10}$$

where $W \in \mathbb{R}^{q\times n_2}$ and $\sigma(z) = [1, \sigma_2(z), \sigma_3(z), ..., \sigma_q(z)]^T$ 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

$$\sigma_i(z) = e^{-\|z-c_i\|^2/\mu_i}. \tag{11}$$

Appealing to the universal approximation property of Radial Basis Function Neural Networks[31] 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

$$\Delta(z) = {W^*}^T \sigma(z) + \tilde{\epsilon}(z). \tag{12}$$

Furthermore $\bar{\epsilon} = \sup_{z \in D} \|\tilde{\epsilon}(z)\|$ can be made arbitrarily small given sufficient number of radial basis functions. We say that this model of the uncertainty is unstructured because the number of RBFs needed and the location of centers is not always clear. For this case it is well known that the following baseline adaptive law

$$\dot{W}(t) = -\Gamma_W \sigma(z(t)) e^T(t) PB \tag{13}$$

guarantees uniform ultimate boundedness of the tracking error, and guarantees that the adaptive parameters stay bounded within a neighborhood of the ideal parameters only if the system states are PE (see e.g. [28, 29, 32, 33]). Note that in this case $e(t) \nrightarrow 0$ since $\bar{\epsilon} \neq 0$. If the system states are not persistently exciting, additional modifications such as e-mod,[34] σ-mod,[35] or projection operator based modifications (see e.g. [36]) are required to guarantee that the adaptive parameters stay bounded around a neighborhood of an *a-priori* selected guess of the ideal parameters (usually set to 0), especially in the presence of noise.

It was shown in [21, 22] 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. In particular, for a linearly parameterized representations of the uncertainty, the following theorem can be proven:[21, 22, 24]

Theorem 1. *Consider the system given by (1), with the inverse law 3, reference models (2), the tracking error equation in (8). Assume further that the uncertainty is linearly parameterizable using an appropriate set of bases over a compact domain D, 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 $\dot{x}_i$. Now consider the following update law for the weights of the RBF NN*

$$\dot{W} = -\Gamma_W \sigma(z) e^T PB - \frac{1}{p} \sum_{j=1}^{p} \Gamma_{W_b} \sigma(x_i, \delta_i) \epsilon_j^T, \tag{14}$$

with Γ_{W_b} the learning rate for training on online recorded data, and assume that $Z = [\phi(z_1), ..., \phi(z_p)]$ and $rank(Z) = l$. Furthermore, let B_α be the largest compact ball in D, and assume $x(0) \in B_\alpha$, define $\delta = \max(\beta, \frac{2\|PB\|\bar{\epsilon}}{\lambda_{\min}(Q)} + \frac{p\bar{\epsilon}\sqrt{l}}{\lambda_{\min}(\Omega)})$, and assume that D is sufficiently large such that $m = \alpha - \delta$ is a positive scalar. If the states $x_r m$ of the bounded input bounded output reference model of (2) remains bounded in the compact ball $B_m = \{x_{rm} : \|x_{rm}\| \leq m\}$ for all $t \geq 0$ then the tracking error e and the weight error $\tilde{W} = W - W^*$ are exponentially uniformly ultimately bounded. Furthermore, if the representation in (12) is exact over the entire operating domain, that is $\bar{\epsilon} = 0$, then the tracking error and weight error converge exponentially fast to a compact ball around the origin for arbitrary initial conditions, with the rate of convergence directly proportional to the minimum singular value of the history stack matrix Z.

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 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 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.*[37, 27] *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.*

The history stack matrix $Z = [\phi(z_1), ..., \phi(z_p)]$ is not a buffer of last p states. It can be updated online by including data points that are of significant interest over the course of operation. Theoretically, convergence is guaranteed as soon as the history stack becomes full ranked. New data points could replace existing data points once the history stack reaches a pre-determined size. It was shown in [24] that the rate of convergence of the tracking error and weights is directly proportional to the minimum singular value of Z. This provides a useful metric to determine which data points are most useful for improving convergence. Consequently, an algorithm for adding points that improve the minimum singular value of Z for the case of

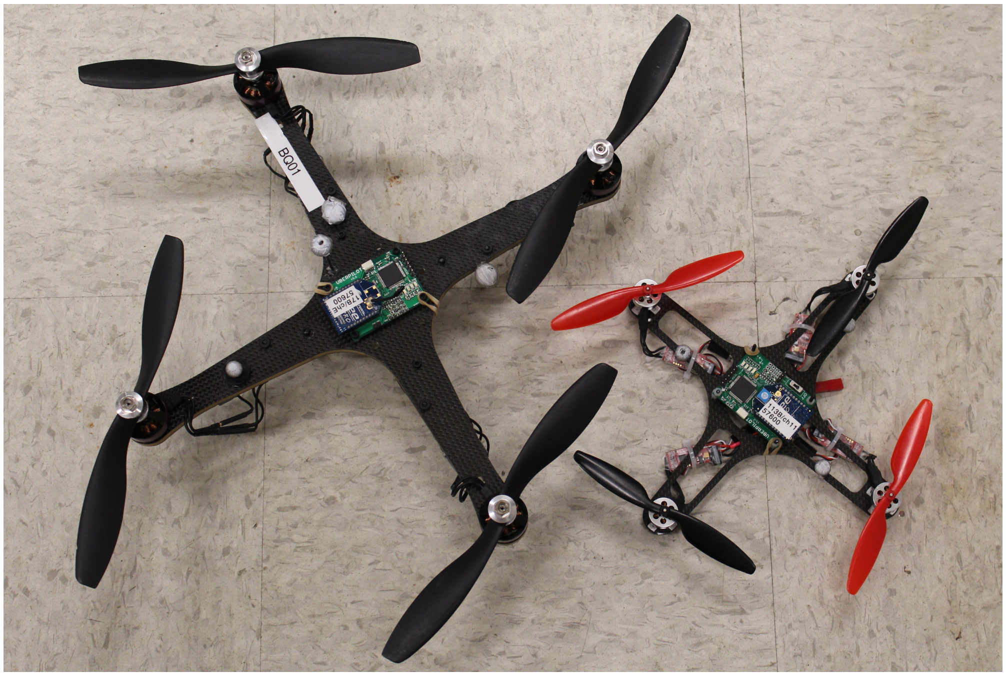

Figure 1. Two MIT quadrotors equipped to fly in the ACL Real Time Indoor Autonomous Vehicle Test Environment (RAVEN).[26] The baseline controller on both quadrotors is PID. The gains have been tuned for the bigger quadrotor. The small quadrotor uses gains from the bigger one, resulting in relatively poor trajectory tracking performance.

linearly parameterizable uncertainty was presented there. The main limitation of the linearly parameterized RBF NN representation of the uncertainty is that the RBF centers need to be preallocated over an estimated compact domain of operation D. Therefore, if the system evolves outside of D all benefits of using adaptive control are lost. This can be addressed by evolving the RBF basis to reflect the current domain of operation, a reproducing kernel Hilbert space approach for accomplishing this was presented in [38]. When the basis is fixed however, for the above adaptive laws to hold, the reference model and the exogenous reference commands should be constrained such that the desired trajectory does not leave the domain over which the neural network approximation is valid. Ensuring that the state remains within a given compact set implies an upper bound on the adaptation gain (see for example Remark 2 of Theorem-1 in [39]). Finally, note also that Δ depends on ν_{ad} through the pseudocontrol ν, whereas ν_{ad} has to be designed to cancel Δ. Hence the existence and uniqueness of a fixed-point-solution for $\nu_{ad} = \Delta(x, \dot{x}, \nu_{ad})$ is assumed. Sufficient conditions for this assumption are also available.[30,40]

III. Flight Test Results on MIT Quadrotors

The Aerospace Controls Laboratory (ACL) at MIT maintains the Real Time Indoor Autonomous Vehicle Test Environment (RAVEN).[26] RAVEN uses a motion capture system[41] to obtain accurate estimates of position and attitude. 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 can be found in [5, 42].

A. Hardware Details

The flight experiments in this paper are 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. On-board attitude control is performed on a custom autopilot, with attitude commands being calculated at 1000 Hz. 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 can be found in [42].

B. Baseline linear controller and adaptive control augmentation

Let the reference trajectory position (generated by a spline based trajectory generator[5]) be given by x_{rm}. The tracking error is defined as $e = x - x_{rm}$. The desired acceleration ν for the baseline PID control is found as

$$\nu = K_p e + K_d \dot{e} + \ddot{x}_{rm} + K_i \int_0^t e(\tau) d\tau + g^i, \tag{15}$$

where K_p, K_d, K_i are the proportional, derivative, and integral gains, and g^i is the acceleration due to gravity expressed in the inertial frame. The acceleration commands are mapped directly into desired thrust, attitude, and attitude rate values using an inversion model whose details are presented in [5]. The attitude control loop (inner-loop) consists of a high bandwidth quaternion-based on-board PD controller that uses attitude and rate estimates to track the desired attitude. See [5, 43, 44] for further details of the inner-loop control law.

In the results presented here (Section C), adaptation is performed only on the position and velocity controller (outer-loop) while the inner-loop quaternion-based attitude controller is left unchanged. The baseline outer-loop controller is augmented using a RBF-NN based MRAC control as described in Section II. In particular, instead of using (15), the the desire acceleration was found using a control law of the form (7)

$$\nu = K_p e + K_d \dot{e} + \ddot{x}_{rm} + g^i - \nu_{ad}, \tag{16}$$

where ν_{ad} is the output of the adaptive element in (10). Both the adaptive law in (13) and the concurrent learning adaptive law in (14) were implemented for testing. Projection operators were used to bound the weights of the baseline MRAC adaptive law,[36] however, 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 NN (10) for each of the loops was $z_x = [x, \dot{x}, \vec{q}]$, $z_y = [y, \dot{y}, \vec{q}]$, $z_z = [z, \dot{z}, \vec{q}]$, where $\vec{q}$ is the attitude quaternion. Note the slight abuse of notation where z denotes both the vertical position and input to the RBF NN. 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

In the following results, baseline PID refers to the outerloop PID controller on the smaller quadrotors with gains directly transferred from the bigger (well-tuned) quadrotor. MRAC refers to the adaptive law in (13) with a projection operator for bounding weights, and CL-MRAC refers to Concurrent Learning - MRAC of Theorem 1. The vehicle performs five sets of three figure 8 maneuvers with a small pause in between. The MRAC learning rate was initialized to $\Gamma_W = 2$ whereas the learning rate of the part of the adaptive law that trains on online recorded data with CL-MRAC was set to $\Gamma_{W_b} = 0.5$. Traditionally, MRAC learning rates are kept constant in adaptive control as suggested by deterministic Lyapunov based stability analysis typically performed. However, classical stochastic stability results by Ljung,[45] Benveniste,[46] and Borkar[47] for example, indicate that driving the learning rate to zero is required for guaranteeing convergence in presence of noise. However, doing this would eventually remove presence of adaptation, so, as is often done in practice, the practical fix here was to decay the learning rate to a small positive constant. The learning rates were decayed by dividing it by 1.5 for Γ_W, and 2 for Γ_{W_b} after every set of three figure 8 maneuvers. The MRAC learning rate was lower bounded to 0.5 and Γ_{W_b} was lower bounded to 0.001. The motivation behind the decaying learning rate is that due to presence of noise $e(t)$ or $\epsilon(t)$ can never be identically equal to zero, hence over-learning or unnecessary oscillations may be observed close to the origin if the learning rate is too high.

In addition to the learning rate, another tuning knob for MRAC and CL-MRAC is the number and location of RBF centers. For the results here a set of 50 RBF centers were randomly distributed using a uniform distribution in the 6 dimensional space in which z was expected to evolve. The centers for the position and velocity for the x, y axes were spread across $[-2, 2]$. For the vertical z axes the position was

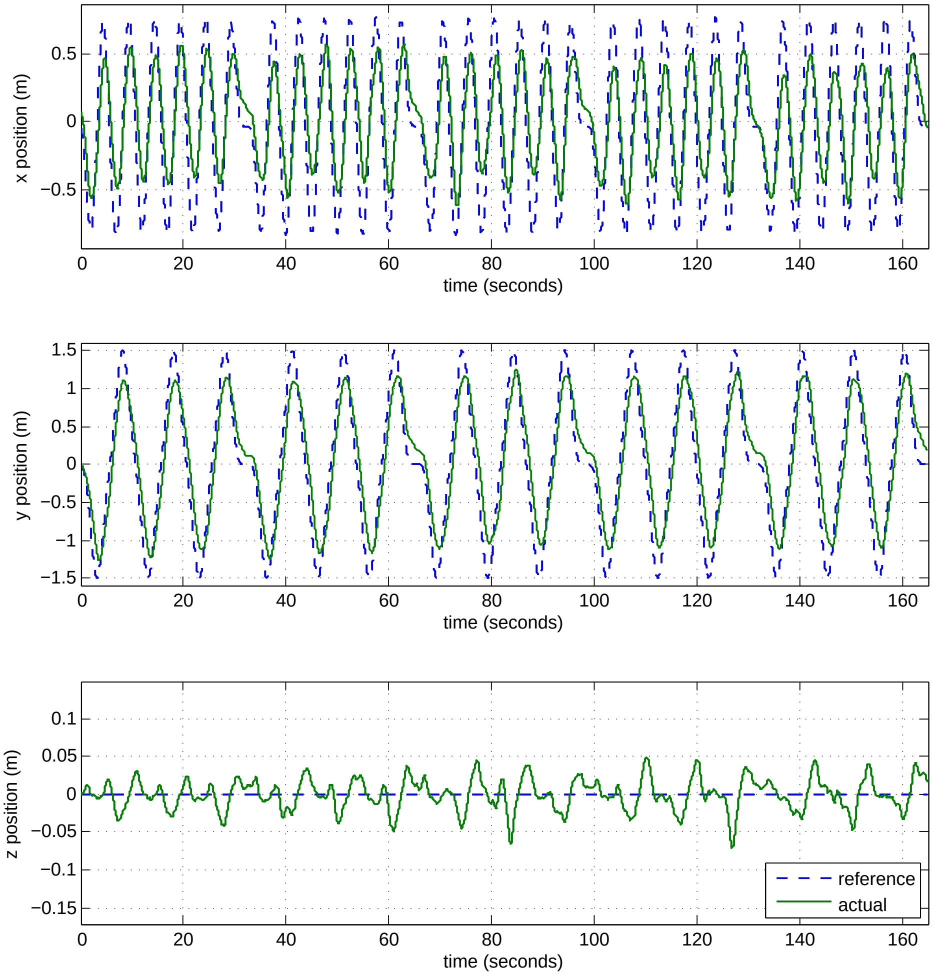

Figure 2. Trajectory tracking performance of the baseline PID controller transferred from another quadrotor with different inertial and motor characteristics. It can be seen that the controller is unable to track the commanded trajectory accurately.

spread across [-.5,-.5], and the velocity was spread across [-0.6,0.6]. The centers for quaternions for all axes were spread across $[-1, 1]$. The bandwidth of the RBF was set to 1. The location and hyperparameters of the RBF were not optimized. It should be noted that the location of the centers can affect the accuracy and numerical performance of the RBF-NN, future work will involve adaptive center placement algorithms such as those described by Kingravi et al.[38] and Chowdhary et al.[48] The final tuning knob is the selection/removal criterion for recorded point. The simple last-point-difference technique was used to record data points that satisfied $\frac{\|\sigma(x(t))-\sigma(x_k)\|}{\sigma(x(t))} \geq 0.01$,[24] where k is the index of last recorded data point. The size of the history stack was set to the number of RBFs, which was 51 (since a bias term is also included). Once the history stack was full, any new points were added by removing the oldest recorded point. Estimates of $\ddot{x}$ were found using Kalman filter based techniques, and any known delay in estimates was accounted for.

Figure 2 shows the trajectory tracking performance of the baseline outer-loop PID controller. It can be seen that the position tracking is poor, although the vehicle is stable. Figure 3 shows that MRAC improves the position tracking performance, however, the vehicle is not able to track the extremities of the commanded position. It can be seen that MRAC's performance does not show any significant improvement over time. Figure 4 shows that CL-MRAC tracking performance gets better over time and is significantly better than that of PID. Figure 5 compares the Root Mean Square (RMS) error of CL-MRAC, MRAC, and PID on the same plot for the three axes. It can be seen that the CL-MRAC exhibits the lowest RMS tracking error in x and y axes.

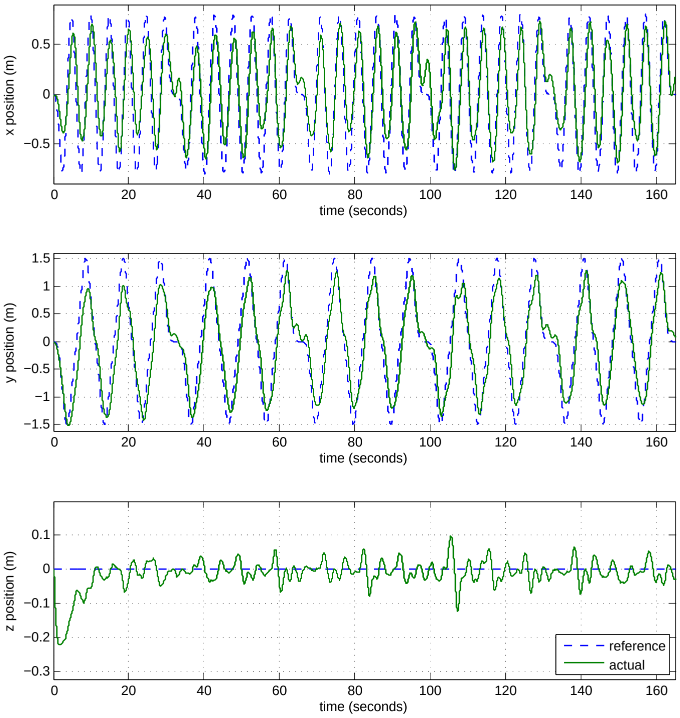

Figure 3. Trajectory tracking performance with MRAC adaptive law of (13) with projection operator. The tracking performance has improved over PID, however, the vehicle is unable to track the extremities of the commanded figure 8 maneuver accurately. Furthermore, little improvement in performance over the long term is observed.

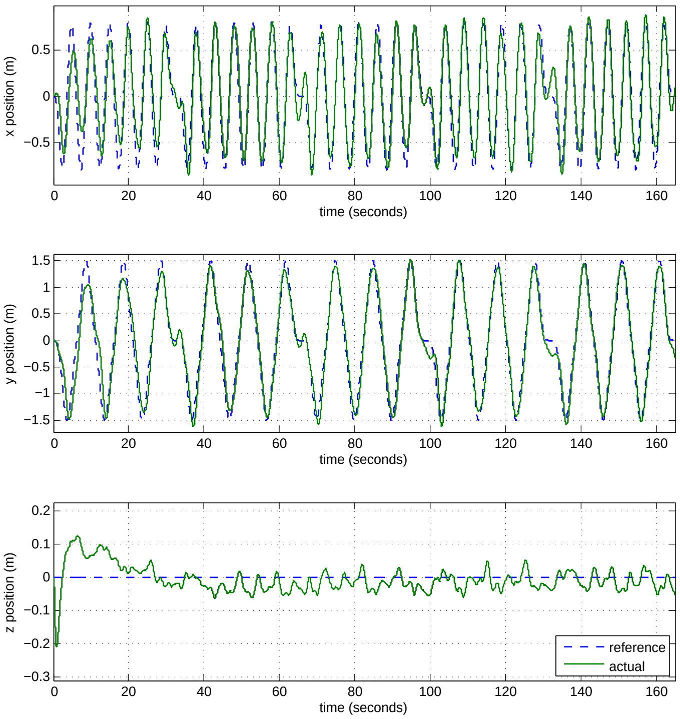

Figure 4. Trajectory tracking performance with CL-MRAC shows significant improvement over PID. Furthermore, results indicate that CL-MRAC gets better at tracking the trajectory over time. The results indicate the feasibility of using adaptive control techniques to rapidly transfer controllers from one UAV to another.

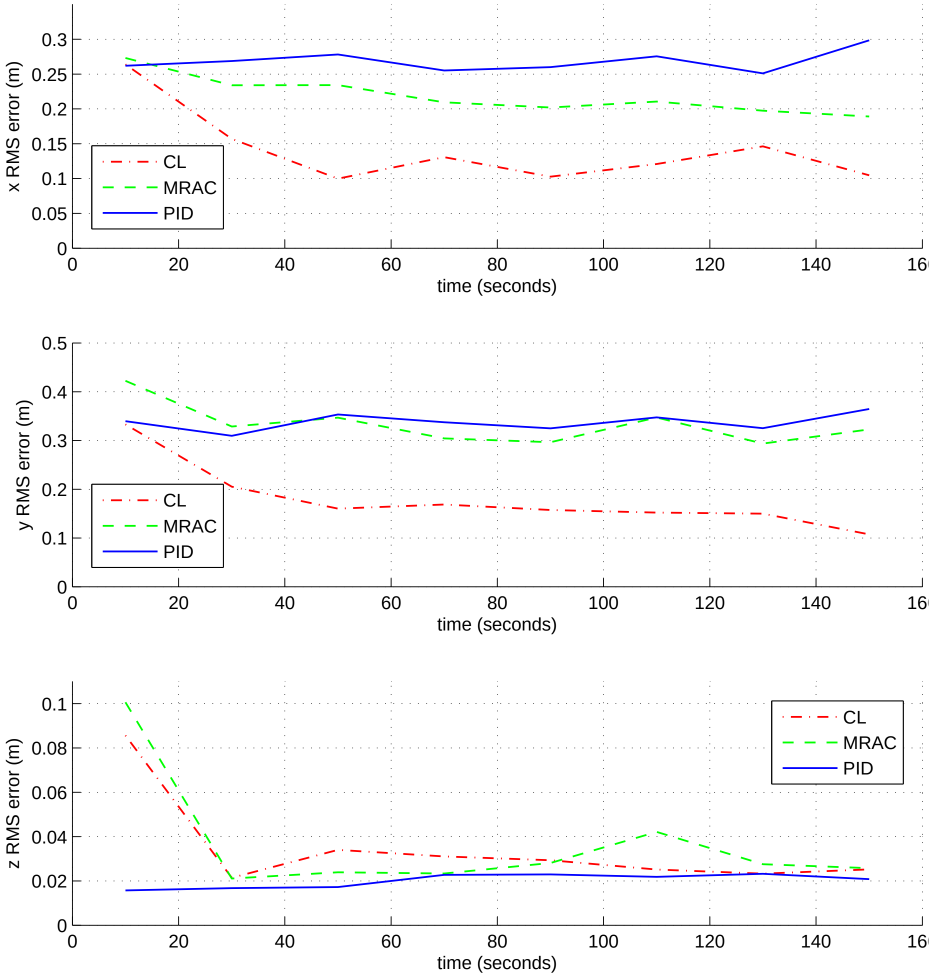

Figure 5. Comparison of RMS tracking error for the three controllers. CL-MRAC controller has significantly less RMS error in horizontal position tracking.

Figure 6 shows the velocity tracking performance of the PID controller is significantly lacking. It is seen in Figure 7 that the velocity tracking performance is improved with MRAC, although the vehicle is not able to track the rapid change in velocities at the extremities of the figure 8 maneuver. Figure 8 shows however, that the velocity tracking performance with CL-MRAC improves over time, and the vehicle learns to track the commanded velocity with little over-/under-shoot over time.

Figure 9 compares the difference between the estimated modeling error $(\hat{\Delta}(t) = \dot{x}(t) - \nu(t))$ and the output of the RBF-NN adaptive element $\nu_{ad}(t)$. It can be seen that the RBF-NN with MRAC shows slight improvement over time in its prediction of the modeling error. This relates directly to the slight improvement seen in MRAC's performance. Figure 10 shows that CL-MRAC's estimate of the modeling error improves significantly over time, until it is able to predict the modeling error much more accurately than MRAC. This is one reason behind the long-term performance improvement seen with CL-MRAC. Both controllers capture the trim in the altitude loop very accurately.

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 the long term learning ability of CL-MRAC, which is due to its judicious use of online recorded data, thereby validating the claims in [22,21] on a new platform.

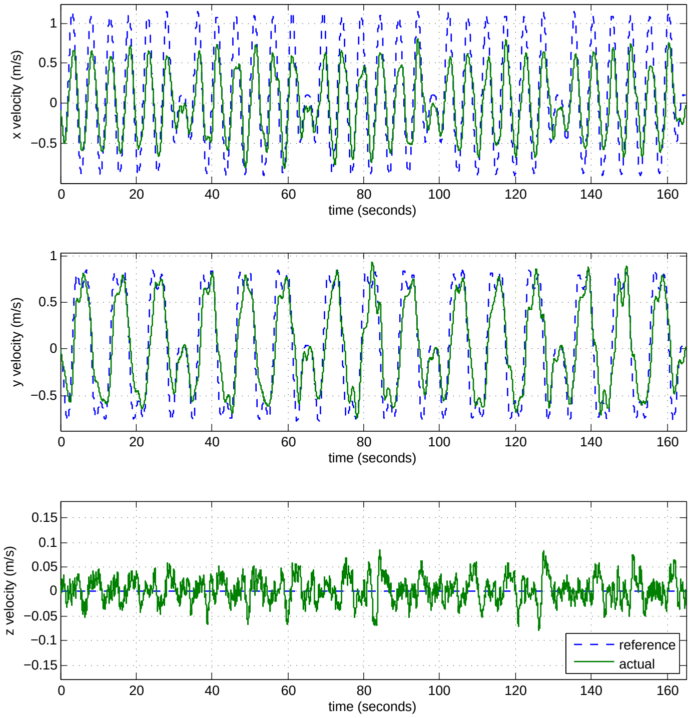

Figure 6. Velocity tracking performance of the baseline PID controller transferred from another quadrotor with different inertial and motor characteristics. The controller is unable to track the velocity commands corresponding to the commanded trajectory accurately.

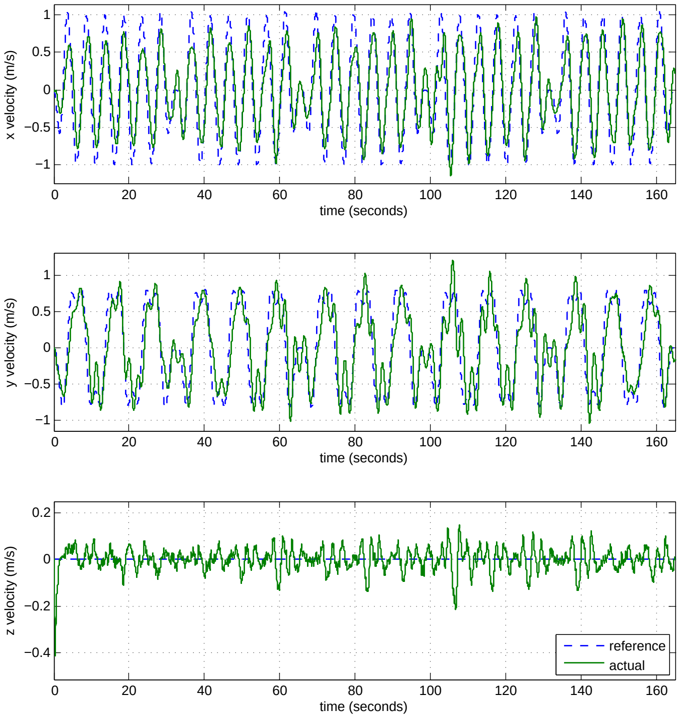

Figure 7. Velocity tracking performance of MRAC. Tracking performance improves over PID.

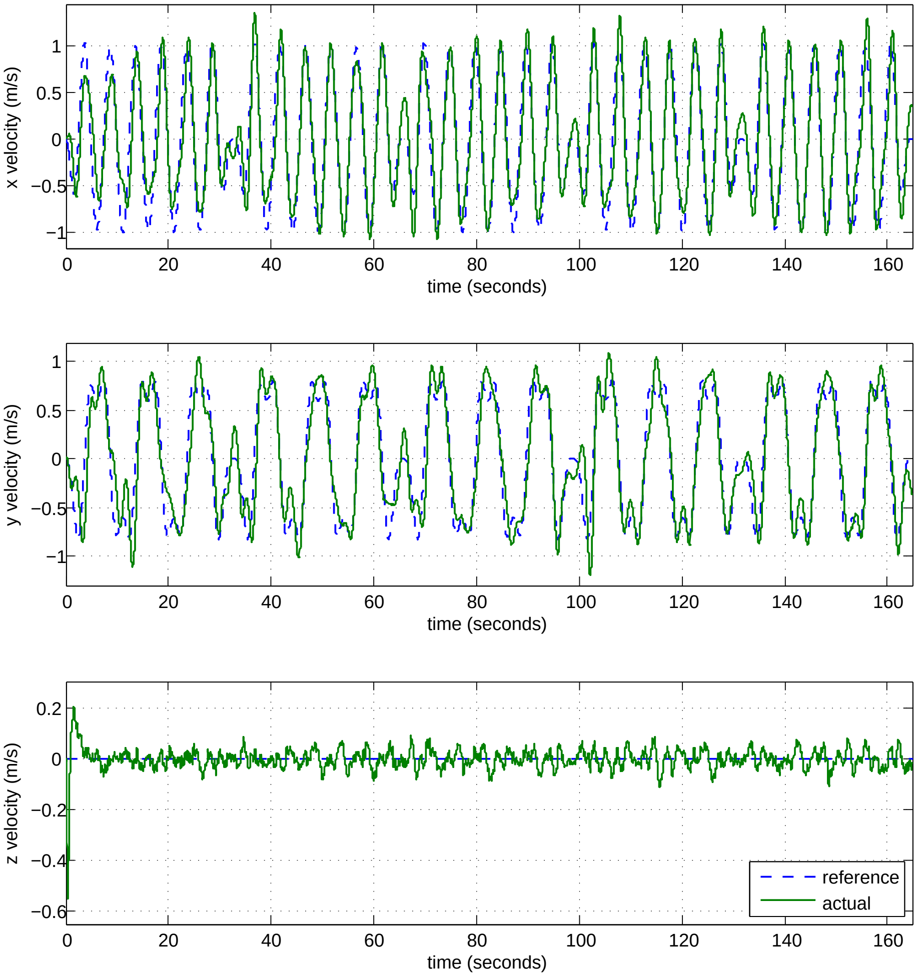

Figure 8. Velocity tracking performance of CL-MRAC. Tracking performance is significantly better than PID, and improvement over long-term is seen over MRAC.

IV. Conclusion

It was shown that adaptive control can be used to speed-up the transfer of controllers from one UAV to another with similar control assignment structure. In particular, it was shown that a concurrent learning adaptive controller improves the trajectory tracking performance of a quadrotor with baseline controller directly imported from another (bigger) quadrotor whose inertial characteristics and throttle mapping are very different. The claims were supported through flight-test results in MIT's RAVEN indoor test environment. A set of repeating maneuvers were used to verify long-term learning properties of concurrent learning controllers. However, prior results have indicated that performance improvement should be observed even if the commands are not repetitive. Future work will concentrate on validating the techniques through testing on trajectories that are more typical of a UAV mission, and investigating how the flight envelope of the UAV could be extended in light of the improved tracking capability.

Acknowledgments

This research was supported in part by ONR MURI Grant N000141110688, and is based in part on work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 0645960. The authors also acknowledge Boeing Research & Technology for support of the RAVEN indoor flight facility in which the flight experiments were conducted.

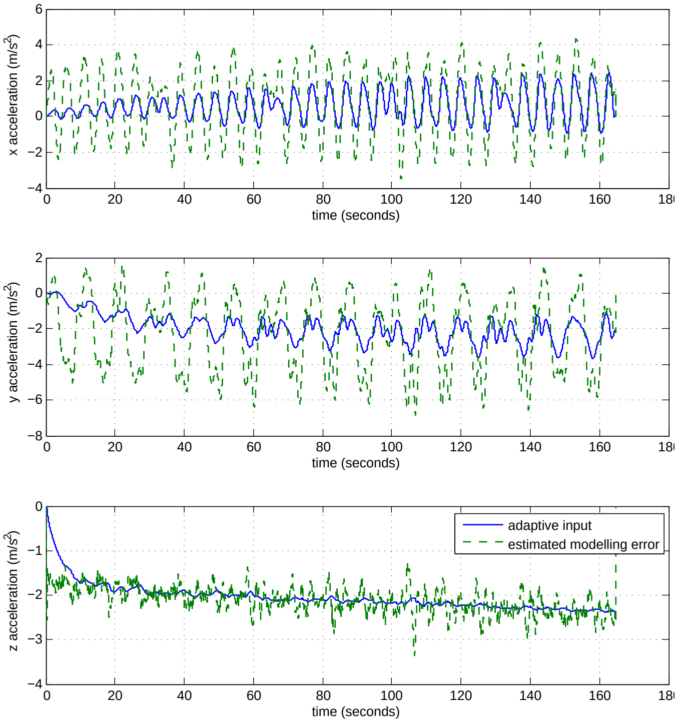

Figure 9. Comparison of estimated modeling error with the output of the RBF-NN adaptive element when using MRAC. Little improvement in the ability of the RBF-NN to predict the modeling error is seen over time.

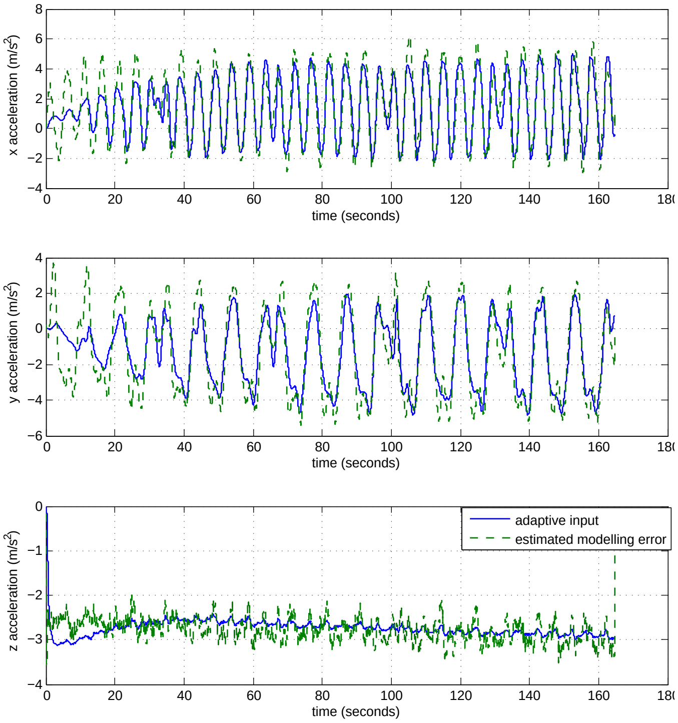

Figure 10. Comparison of estimated modeling error with the output of the RBF-NN adaptive element when using CL-MRAC. The RBF-NN adaptive element learns to predict the modeling error fairly accurately over time when trained with CL-MRAC; the improved prediction leads to better tracking as seen in Figures 4 and 8. These results indicate the presence of long-term learning.

References

[1] Bouabdallah, S., Noth, A., and R., S., "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor," *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[2] Portlock, J. N. and Cubero, S. N., "Dynamics and Control of a VTOL Quad-Thrust Aerial Robot," *Mechatronics and Machine Vision in Practice*, edited by J. Billingsley and R. Bradbeer, Springer Berlin Heidelberg, 2008, pp. 27–40, 10.1007 978-3-540-74027-83.

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

[4] Michini, B., Redding, J., Ure, N. K., Cutler, M., and How, J. P., "Design and Flight Testing of an Autonomous Variable-Pitch Quadrotor," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2011, pp. 2978 – 2979.

[5] Cutler, M. and How, J. P., "Actuator Constrained Trajectory Generation and Control for Variable-Pitch Quadrotors," *AIAA Guidance, Navigation, and Control Conference (GNC)*, Minneapolis, Minnesota, August 2012.

[6] Huang, H., Hoffmann, G., Waslander, S., and Tomlin, C., "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," *IEEE International Conference on Robotics and Automation (ICRA)*, May 2009, pp. 3277 –3282.

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

[8] Michael, N., Mellinger, D., Lindsey, Q., and Kumar, V., "The GRASP Multiple Micro-UAV Testbed," *IEEE Robotics & Automation Magazine*, Vol. 17, No. 3, 2010, pp. 56–65.

[9] Calise, A., Hovakimyan, N., and Idan, M., "Adaptive Output Feedback Control of Nonlinear Systems Using Neural Networks," *Automatica*, Vol. 37, No. 8, 2001, pp. 1201–1211, Special Issue on Neural Networks for Feedback Control.

[10] Johnson, E. and Kannan, S., "Adaptive Flight Control for an Autonomous Unmanned Helicopter," *Proceedings of the AIAA Guidance Navigation and Control Conference, held at Monterrery CA*, 2002.

[11] Johnson, E. N. and Oh, S. M., "Adaptive Control using Combined Online and Background Learning Neural Network," *Proceedings of CDC*, 2004.

[12] Johnson, E. N., *Limited Authority Adaptive Flight Control*, Ph.D. thesis, Georgia Institute of Technology, Atlanta Ga, 2000.

[13] Kannan, S. K., Koller, A. A., and Johnson, E. N., "Simulation and Development Environment for Multiple Heterogeneous UAVs," *AIAA Modeling and Simulation Technology Conference*, No. AIAA-2004-5041, Providence, Rhode Island, August 2004.

[14] Kannan, S., *Adaptive Control of Systems in Cascade with Saturation*, Ph.D. thesis, Georgia Institute of Technology, Atlanta Ga, 2005.

[15] Hovakimyan, N., Yang, B. J., and Calise, A., "An Adaptive Output Feedback Control Methodology for Non-Minimum Phase Systems," *Automatica*, Vol. 42, No. 4, 2006, pp. 513–522.

[16] Cao, C. and Hovakimyan, N., "L1 Adaptive Output Feedback Controller for Systems with Time-varying Unknown Parameters and Bounded Disturbances," *Proceedings of American Control Conference*, New York, 2007.

[17] Lavretsky, E. and Wise, K., "Flight Control of Manned/Unmanned Military Aircraft," *Proceedings of American Control Conference*, 2005.

[18] Nguyen, N., Krishnakumar, K., Kaneshige, J., and Nespeca, P., "Dynamics and Adaptive Control for Stability Recovery of Damaged Asymmetric Aircraft," *AIAA Guidance Navigation and Control Conference*, Keystone, CO, 2006.

[19] Steinberg, M., "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, 2005, pp. 263–275.

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

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

[22] Chowdhary, G. and Johnson, E. N., "Concurrent Learning for Convergence in Adaptive Control Without Persistency of Excitation," *49th IEEE Conference on Decision and Control*, 2010, pp. 3674–3679.

[23] Chowdhary, G. and Johnson, E. N., "Concurrent Learning for Improved Parameter Convergence in Adaptive Control," *AIAA Guidance Navigation and Control Conference*, Toronto, Canada, 2010.

[24] Chowdhary, G. and Johnson, E. N., "A Singular Value Maximizing Data Recording Algorithm for Concurrent Learning," *American Control Conference*, San Francisco, CA, June 2011.

[25] Chowdhary, G., Muhlegg, M., Yucelen, T., and Johnson, E., "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *International Journal of Adaptive Control and Signal Processing*, 2012.

[26] How, J. P., Bethke, B., Frank, A., Dale, D., and Vian, J., "Real-Time Indoor Autonomous Vehicle Test Environment," *IEEE Control Systems Magazine*, Vol. 28, No. 2, April 2008, pp. 51–64.

[27] Chowdhary, G. and Johnson, E. N., "Theory and Flight Test Validation of a Concurrent Learning Adaptive Controller," *Journal of Guidance Control and Dynamics*, Vol. 34, No. 2, March 2011, pp. 592–607.

[28] Sanner, R. and Slotine, J.-J., "Gaussian networks for direct adaptive control," *Neural Networks, IEEE Transactions on*, Vol. 3, No. 6, nov 1992, pp. 837 –863.

[29] Narendra, K., "Neural networks for control theory and practice," *Proceedings of the IEEE*, Vol. 84, No. 10, oct 1996, pp. 1385 –1406.

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

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

[32] Nardi, F., *Neural Network based Adaptive Algorithms for Nonlinear Control*, Ph.D. thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, nov 2000.

[33] Kim, Y. H. and Lewis, F., *High-Level Feedback Control with Neural Networks*, Vol. 21 of *Robotics and Intelligent Systems*, World Scientific, Singapore, 1998.

[34] Narendra, K. and Annaswamy, A., "A New Adaptive Law for Robust Adaptation without Persistent Excitation," *IEEE Transactions on Automatic Control*, Vol. 32, No. 2, February 1987, pp. 134–145.

[35] Ioannou, P. A. and Sun, J., *Robust Adaptive Control*, Prentice-Hall, Upper Saddle River, 1996.

[36] Tao, G., *Adaptive Control Design and Analysis*, Wiley, New York, 2003.

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

[38] Kingravi, H. A., Chowdhary, G., Vela, P. A., and Johnson, E. N., "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, No. 7, july 2012, pp. 1130 –1141.

[39] Yucelen, T. and Calise, A., "Kalman Filter Modification in Adaptive Control," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, march-april 2010, pp. 426–439.

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

[41] "Motion Capture Systems from Vicon," Tech. rep., 2011, Online http://www.vicon.com/.

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

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

[44] Chowdhary, G., Frazzoli, E., How, J. P., and Lui, H., *Handbook of Unmanned Aerial Vehicles*, chap. Nonlinear Flight Contol Techniques for Unmanned Aerial Vehicles, Springer, 2012 (to appear).

[45] Ljung, L., "Analysis of recursive stochastic algorithms," *Automatic Control, IEEE Transactions on*, Vol. 22, No. 4, aug 1977, pp. 551 – 575.

[46] Benveniste, A., Priouret, P., and Métivier, M., *Adaptive algorithms and stochastic approximations*, Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[47] Borkar, V. and Soumyanatha, K., "An analog scheme for fixed point computation. I. Theory," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, Vol. 44, No. 4, apr 1997, pp. 351 –355.

[48] Chowdhary, G., How, J., and Kingravi, H., "Model Reference Adaptive Control using Nonparametric Adaptive Elements," *Conference on Guidance Navigation and Control*, AIAA, Minneapolis, MN, August 2012, Invited.