Adaptive Control of Quadrotor UAVs: A Design Trade Study With Flight Evaluations

Zachary T. Dydek, Anuradha M. Annaswamy, and Eugene Lavretsky

***Abstract*— This brief describes the application of direct and indirect model reference adaptive control to a lightweight low-cost quadrotor unmanned aerial vehicle platform. A baseline trajectory tracking controller is augmented by an adaptive controller. The approach is validated using simulations and flight tested in an indoor test facility. The adaptive controller is found to offer increased robustness to parametric uncertainties. In particular, it is found to be effective in mitigating the effects of a loss-of-thrust anomaly, which may occur due to component failure or physical damage. The design of the adaptive controller is presented, followed by a comparison of flight test results using the existing linear and augmented adaptive controllers.**

***Index Terms*— Adaptive control, flight control, unmanned aerial vehicles (UAVs).**

I. Introduction

QUADROTOR helicopters have been an increasingly popular research platform in recent years. In designing a controller for these aircraft, there are several important considerations. There are numerous sources of uncertainty in the system—actuator degradation, external disturbances, and potentially uncertain time delays in processing or communication. These problems are amplified in the case of actuator failures, where the aircraft has lost some of its control effectiveness. Adaptive control is an attractive candidate for this type of aircraft because of its ability to generate high performance tracking in the presence of parametric uncertainties.

The first known quadrotor flight was in the 1920s [1], but due to the difficulty of controlling four motors simultaneously with sufficient bandwidth, the project was abandoned. In 1963, the Curtiss-Wright X-19A became the first manned quadrotor to leave the ground effect. Due to a lack of stability augmentation system, stationary hover was nearly impossible [2]. Smaller, unmanned quadrotor helicopters typically consist of two pairs of counter-rotating blades mounted on a carbon fiber frame, as shown in Fig. 1. The dynamics of quadrotor helicopters have been studied in detail by several groups [3]–[6]. In designing a controller for these aircraft, there are several important vehicle-specific considerations. The dynamics of quadrotors are nonlinear and multivariate. There are also several effects to which a potential controller must be robust: the aerodynamics of the rotor blades (propeller and blade flapping), inertial anti-torques (asymmetric angular speed of propellers), as well as gyroscopic effects (change in orientation of the quadrotor and the plane of the propeller).

A number of approaches to quadrotor flight control have been applied to a variety of problems, including vision-based sensing algorithms for indoor flight [7]–[9], dynamic programming approaches to task assignment [10], autonomous tracking and regulation in indoor and outdoor settings [4], and cooperative manipulation [11], [12]. Control approaches, including feedback linearization with a high-order sliding mode observer [13], and machine learning [14] have been used to perform acrobatic maneuvers in spite of the uncertainties. Simulation results have shown nonlinear controllers can also provide robustness to modeling errors in vertical take-off and landing drones [15].

While most of the above approaches acknowledge the significant uncertainties present in the system and the potential for additional uncertainty due to failures, none of the approaches explicitly accounts for these uncertainties in the controller design. Typical control laws, such as linear, proportional-integral-derivative control or linear-quadratic regulator (LQR)-based design offer some measure of robustness to uncertainties but may not be sufficient for severe uncertainties such as those that result from actuator failure or structural damage. An adaptive controller based on the MIT-rule was designed to provide robustness to partial actuator failures [16]. However, no stability proof is provided in [16] and in fact the MIT rule cannot be guaranteed to be stable in general. In this brief, we describe the application of a model reference adaptive approach that is based on Lyapunov stability arguments, and includes both direct and indirect adaptation to improve tracking and parameter estimation performance.

This brief is organized as follows. Section II describes the dynamics of the quadrotor and the development of the direct adaptive controller. Section III describes the experimental setup. Flight test results of the direct adaptive controller are presented in Section IV. Section V describes a combined/composite adaptive approach that addresses some of the shortcomings of the direct adaptive controller. In Section VI, we validate the adaptive controller design through flight testing of a quadrotor unmanned aerial vehicle (UAV). Section VII summarizes the results and gives some conclusions.

II. Adaptive Controller Design for Quadrotor Helicopters

In this section, we examine the dynamics of the quadrotor helicopter in order to gain the insight necessary for the

Manuscript received July 25, 2011; revised May 10, 2012; accepted May 13, 2012. Manuscript received in final form May 13, 2012. Date of publication June 19, 2012; date of current version June 14, 2013. This work was supported in part by the Boeing Company Strategic University Initiative. Recommended by Associate Editor A. Serrani.

Z. T. Dydek and A. M. Annaswamy are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: zac@mit.edu; aanna@mit.edu).

E. Lavretsky is with the Phantom Works Department, Boeing Company, Huntington Beach, CA 92647-2099 USA (e-mail: eugene.lavretsky@boeing.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2012.2200104

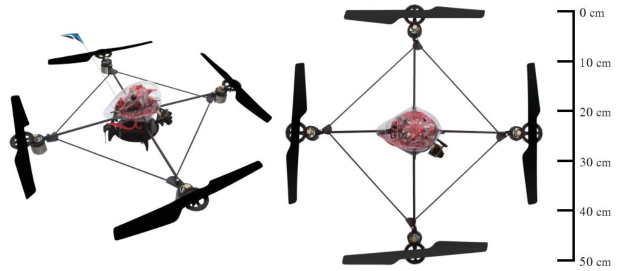


Fig. 1. Draganflyer V Ti four rotor helicopter UAV. The distance from the center of mass to the rotor shaft is 24 cm.

adaptive control design. In particular, the reference model used by the model reference adaptive control approach is generated using the linearized quadrotor dynamics and a linear baseline controller. The adaptive controller is then formulated for the problem of command tracking in the presence of parametric uncertainties in the form of actuator failures, where one or more of the four propellers loses a portion of its thrust.

A. Quadrotor Dynamics

The dynamics of quadrotor helicopters have been studied in detail by several groups [3], [4]. A simple, rigid-body model of the quadrotor, which assumes low speeds is given by

$$\begin{aligned}
\ddot{x} &= (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)\,\frac{U_1}{m}\\
\ddot{\phi} &= \dot{\theta}\dot{\psi}\left(\frac{I_y - I_z}{I_x}\right) - \frac{J_R}{I_x}\dot{\theta}\Omega_R + \frac{L}{I_x}U_2\\
\ddot{y} &= (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)\,\frac{U_1}{m}\\
\ddot{\theta} &= \dot{\phi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{J_R}{I_y}\dot{\phi}\Omega_R + \frac{L}{I_y}U_3\\
\ddot{z} &= -g + (\cos\phi \cos\theta)\,\frac{U_1}{m}\\
\ddot{\psi} &= \dot{\phi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}U_4
\end{aligned} \tag{1}$$

where x, y, and z are the position of the center of mass in the inertial frame; ϕ, θ, and ψ are the Euler angles, which describe the orientation of the body-fixed frame with respect to the inertial frame; m, I_x, I_y, and I_z are the mass and moments of inertia of the quadrotor, respectively; L is the length from the rotors to the center of mass; and J_R and Ω_R are the moments of inertia and angular velocity of the propeller blades. U_1, U_2, U_3, and U_4 are the collective, roll, pitch, and yaw forces generated by the four propellers. Since the quadrotor typically operates very near the hover position, we can make small angle approximations, neglect higher order terms and let $U_1 = mg + \Delta U_1$, resulting in the linear dynamics

$$\begin{aligned}
\ddot{x} &= g\theta, \qquad \ddot{y} = -g\phi, \qquad \ddot{z} = \frac{\Delta U_1}{m}\\
\ddot{\phi} &= \frac{L}{I_x}U_2, \qquad \ddot{\theta} = \frac{L}{I_y}U_3, \qquad \ddot{\psi} = \frac{1}{I_z}U_4.
\end{aligned} \tag{2}$$

While simple, this model captures the dominant dynamics of the quadrotor, and is accurate near the hover position. As expected, the roll, pitch, and yaw inputs command moments about their respective axes and the collective input commands acceleration in the positive z-direction. Accelerations in the x- and y-directions are achieved primarily through vectoring the collective thrust.

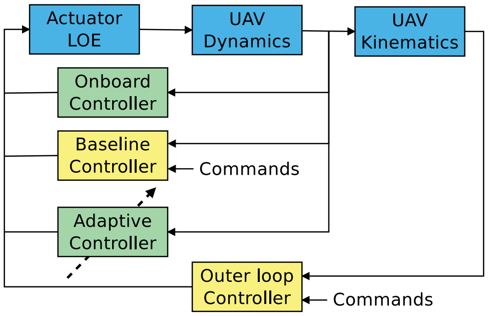


Fig. 2. Schematic block diagram of the UAV plant, control architecture, and actuator loss of effectiveness (LOE). Commands to the baseline controller are the desired trajectory of the fast output states. Commands to the outer loop controller are the desired trajectory of the slow output states.

This simplified model also sheds some light on a time-scale separation present in the system. It can be seen that the dynamics of z, ϕ, θ, and ψ are double integrators, while the dynamics of x and y are quadruple integrators. The former group can be thought of as "fast" states, or as the vehicle dynamics while the latter group can be thought of as "slow" states, or the vehicle kinematics. This time-scale separation can be seen in Fig. 2. The onboard controller for these vehicles is a simple feedback control on rate gyro measurements of the states $\dot{\phi}$ and $\dot{\theta}$. The baseline controller is off-board, and comprises feedback on z, ϕ, θ, and ψ and their derivatives. For the purposes of control design, we can use the linear dynamics given by (2).

B. Problem Statement

The primary function of the adaptive controller is to accommodate any uncertainties, which may arise in the dynamics (2). We can write the equations of motion in (2) along with these uncertainties as

$$\dot{x}_p = A_p x_p + B_p \Lambda u \tag{3}$$

where $B_p \in \Re^{n_p \times m}$ is constant and *known*, $A_p \in \Re^{n_p \times n_p}$ is constant and *unknown*, $x_p \in \Re^{n_p}$, $u \in \Re^m$, and $\Lambda \in \Re^{m \times m}$ is an unknown positive definite matrix. The state x_p thus consists of x, y, z, ϕ, θ, ψ, and their derivatives. We consider a fully coupled model of the dynamics in (2) since coupling is introduced by the uncertainties. Off-diagonal terms in the matrix A_p may contain unmodeled cross-coupling terms and the matrix Λ may represent LOE of one of the rotors, which results in a coupling of the collective, yaw, and one of the roll or pitch axes. The goal is to track a reference command $r(t) \in \Re^m$ in the presence of the unknown A_p and Λ.

We define the system output $y_p \in \Re^m$ as $y_p = C_p x_p$. In the case of the quadrotor, the output states are x, y, z, and ψ. The output tracking error is then given by $e_y = y_p - r$. Augmenting (3) with the integrated output tracking error $\dot{e}_{y_I} = e_y$, leads to the extended open-loop dynamics

$$\dot{x} = Ax + B\Lambda u + B_c r \tag{4}$$

where $x = [x_p^T \ e_{y_I}^T]^T$ is the extended system state vector. The extended open-loop system matrices are given by

$$A = \begin{bmatrix} A_p & 0_{n_p\times m} \\ C_p & 0_{m\times m} \end{bmatrix},\ B = \begin{bmatrix} B_p \\ 0_{m\times m} \end{bmatrix},\ B_c = \begin{bmatrix} 0_{n_p\times m} \\ -I_{m\times m} \end{bmatrix} \tag{5}$$

and the extended system output $y = \begin{bmatrix} C_p & 0_{m\times m} \end{bmatrix} x = Cx$. Note that the dimension of the extended system state vector is $n = n_p + m$, therefore $A \in \Re^{n\times n}$, $B, B_c \in \Re^{n\times m}$, and $C \in \Re^{m\times n}$.

C. Controller Design

A baseline controller

$$u_{bl} = K_{bl} x \tag{6}$$

can be designed for the system in (4) assuming there is no uncertainty, that is $\Lambda = I_{m\times m}$ and A is taken at some nominal value $\bar{A}$ where actually $A = \bar{A} + A^*$ and A^* contains all the uncertain terms. The feedback gains K_{bl} can be selected using LQR or classical design techniques. To account for the uncertainties in the system, we use the model reference adaptive controller (MRAC) approach. The reference model used by MRAC is the closed-loop system given by (4), again in the case of no uncertainty, along with the control input in (6)

$$\dot{x}_m = \bar{A} x_m + B u_{bl} + B_c r \tag{7}$$

where x_m is the reference model state. It is assumed that there exists a gain matrix K_x such that the closed-loop matrix $A_m = A + B\Lambda K_x$ of the actual system in (4), is Hurwitz.

An adaptive control input is added to the baseline controller as

$$u_{ad} = \hat{K}_x^T x + \hat{\theta}_r^T r + \hat{\theta}_d = \hat{\theta}^T \omega \tag{8}$$

where $\hat{\theta}^T = [\hat{K}_x^T \ \hat{\theta}_r^T \ \hat{\theta}_d^T]$ is a matrix of time-varying adaptive parameters with dimensions $m \times p$ with $p = m+n+1$ and $\omega^T = [x^T \ r^T \ 1]$ is a regressor vector of dimension p. The adaptive parameters will be adjusted in the adaptive law given in (10) below. The overall control input is thus

$$u = u_{ad} + u_{bl} = \hat{\theta}^T \omega + K_x x. \tag{9}$$

The time scale separation discussed in Section II-A is enforced through selection of 0 feedback gains on the "slow" states.

The classical adaptive law is given by

$$\dot{\hat{\theta}} = -\Gamma \omega e^T P B \tag{10}$$

where $\Gamma \in \Re^{p\times p}$ is a diagonal, positive definite matrix of adaptive gains, $e = x - x_m$ is the model tracking error, and $P \in \Re^{n\times n}$ is the unique symmetric positive definite solution of the Lyapunov equation, $A_m^T P + P A_m = -Q$, where Q is also symmetric positive definite. This adaptive controller is based on nonlinear stability theory, the details of which have been touched on by many authors throughout the years [17]–[22]. The augmented structure of the adaptive controller implies that in the nominal case, that is the case with no parameter uncertainty, the overall system is equivalent to the baseline control. However, when failures or other uncertainties arise, the adaptive controller works to assist the baseline controller in maintaining stability and performance. The overall block diagram for this system is shown in Fig. 2.

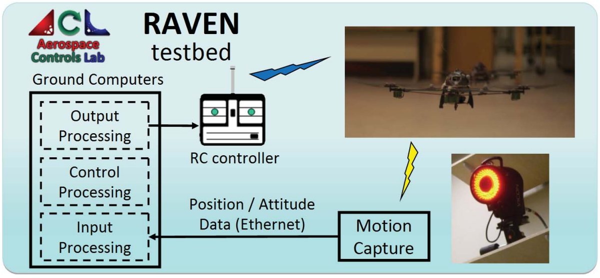


Fig. 3. Architecture of the RAVEN.

The proof of stability for the adaptive system uses a Lyapunov approach with the Lyapunov function candidate given by

$$V = e^T P e + Tr\left(\tilde{\theta}^T \Gamma^{-1} \tilde{\theta}\right) \tag{11}$$

where $\tilde{\theta} = \hat{\theta} - \theta$ is the parameter estimation error. The value of $\tilde{\theta}$ will in general be unknown; however, it is not required by the control law, it is used only in the proof. It can be shown [20] that the derivative of the Lyapunov function candidate is given by $\dot{V} = -e^T Q e \leq 0$. The system is globally asymptotically stable by Barbalat's lemma and the tracking error asymptotically converges to 0, that is $\lim_{t\to\infty} e(t) = 0$. Note that this result applies to the linear system with matched uncertainties described in (4). In the flight regime of interest, the nonlinear terms in the original system (1) are dominated by the linear terms. If the residual nonlinearities are carried through the proof, they result in terms in the derivative of the Lyapunov function candidate that are linear in $||e||$. This changes the stability result from one of global asymptotic convergence to boundedness of the error signal.

III. Experimental Setup

Flight testing was done in collaboration with the Aerospace Controls Laboratory at MIT, which utilizes a UAV testbed facility known as real-time indoor autonomous vehicle test environment (RAVEN), described in Fig. 3. RAVEN uses a Vicon motion capture system to enable rapid prototyping of aerobatic flight controllers for helicopters and aircraft; robust coordination algorithms for multiple helicopters; and vision-based sensing algorithms for indoor flight [7]–[9]. The RAVEN system essentially consists of a vision-based motion capture system that provides six-degree-of-freedom pose parameters at a fixed frequency of 100 Hz. Commands are sent to the quadrotors using a USB wireless remote-control module. The system operates on a cluster of Linux workstations, with each node assigned to the control of an individual UAV. This system is used to run the vehicle flight controller and other path planning algorithms. The adaptive controller described in Section II is then implemented in C^{++} and run alongside the fixed-gain baseline controller, both operating on feedback signals of the fast states: z, ϕ, θ, and ψ. The onboard controller is comprised of a fixed-gain feedback on roll and pitch rate from onboard rate gyros. This system allows for flight testing

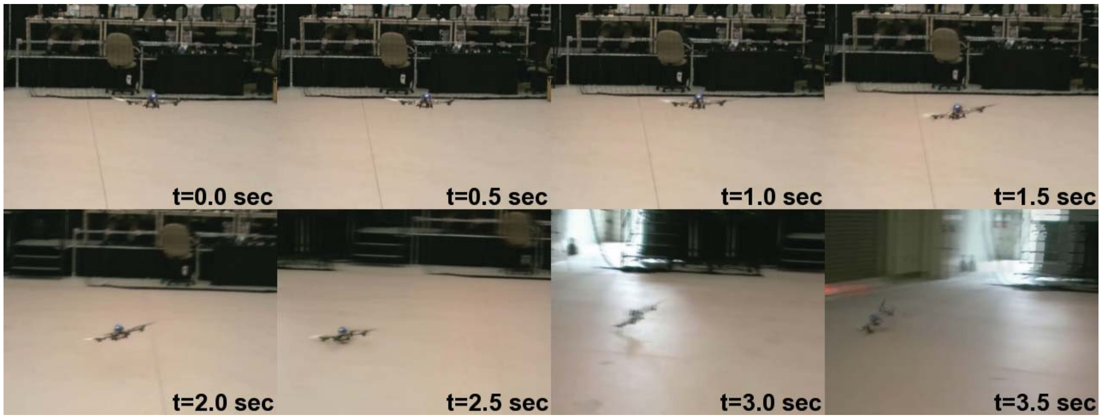


Fig. 4. Blade-cutting flight test with the linear baseline controller. The failure occurs between $t = 1.0$ s and $t = 1.5$ s.

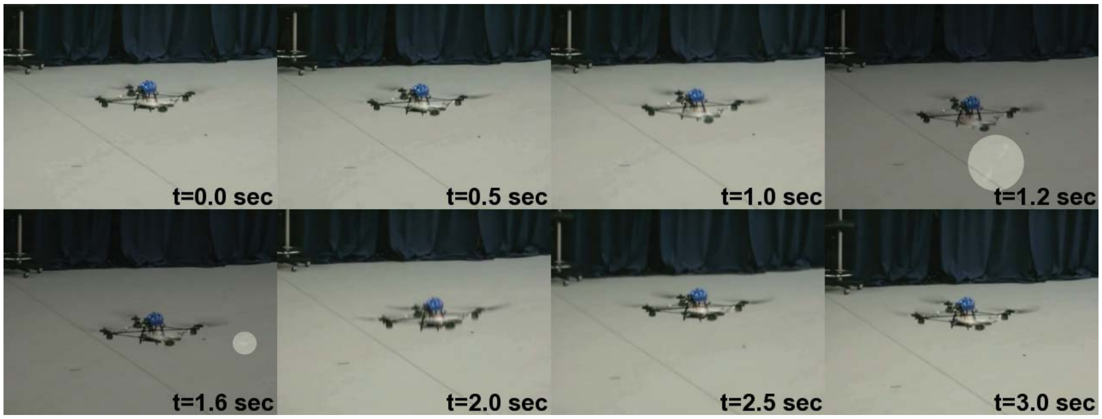


Fig. 5. Screen captures of the blade-cutting flight test with the adaptive controller. Time indices are given in the bottom right corner of each frame. At $t = 1.2$ s and $t = 1.6$ s, the liberated propeller tips are highlighted as they move quickly out of frame.

of the quadrotor using both baseline (fixed-gain) and adaptive controllers.

With the capabilities and limitations of RAVEN in mind, a number of modifications were made to the typical model reference adaptive approach. The dead-zone modification was included in the error signals to combat parameter estimate drift. The projection operator [18], [23] is added to the control law to provide an upper bound on the parameter estimates. Finally, adaptive gains were selected using an empirical guideline [24], then manually tuned to achieve desired response in each axis. For more implementation details and implications of the proof of stability, the reader is referred to [25].

IV. MRAC Flight Test Results

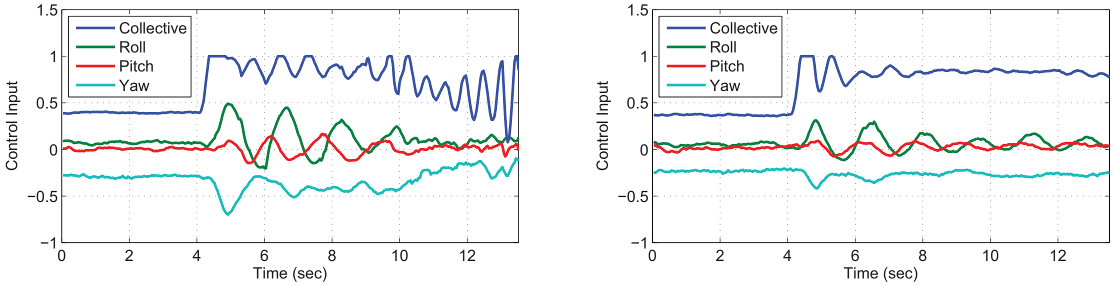
This series of tests involved a demonstration of a simulated "real-world" failure, wherein one of the propellers is cut mid-flight. To accomplish this, a propeller was modified pre-flight by slicing both tips (approximately 25% of the propeller radius on each side) and then taping them back together using masking tape. During flight, a small razor blade mounted to a radio-controlled servo motor and attached to the quadrotor frame can be remotely actuated into the path of the spinning blade, cutting the tape. This results in approximately a 40% loss-of-thrust.

This approach allows for repeatable experiments as the blade tips can be collected and reattached to the propeller. Thus, baseline and adaptive controllers can be compared easily. Videos of the flight test results have been posted online [26]. Screen captures of these videos can be seen in Figs. 4 and 5.

The goal of the maneuver shown in Figs. 4 and 5 is to remain in a hover position. The baseline controller is unable to recover from the sudden change in dynamics and the vehicle crashes into the floor and a nearby wall. On the other hand, the adaptive controller quickly accounts for the loss-of-thrust and adjusts online, allowing for safe return to hover and safe landing. Fig. 5 also shows the precise moments when the two propeller tips fly off screen.

V. Combined/Composite Adaptive Controller

Combined or composite MRAC (CMRAC) is an adaptive controller that consists of the combination of direct and indirect adaptive control. It has been observed that by adapting to both estimation and tracking errors, CMRAC systems can achieve smoother transient performance than MRAC systems for a wide variety of problems [27]–[30]. Furthermore, CMRAC accomplishes this while adding minimal computational cost, which is an important consideration for UAVs with embedded processors or limited onboard computation. The aim of the following study is to reproduce the benefits of CMRAC observed in simulation on a hardware system with a time



(a)

(b)

Fig. 6. Control inputs for (a) MRAC and (b) CMRAC with the same adaptive gains. A 25% loss of collective control effectiveness occurs at $t = 4$ s.

delay in the loop. Robustness to losses of control effectiveness during command tracking tasks will be examined.

The direct adaptive controller described in Section II uses the error between the plant state, given by (4), and the reference model, given by (7) to adjust its parameters. To add the indirect adaptive components, we must start by generating a suitable prediction error $e_Y(t) = \hat{Y}(t) - Y(t)$. We generate the signal $Y(t) \in \Re^m$ as

$$Y(t) = \left(B^T B\right)^{-1} B^T \, (\lambda_f (x(t) - x_f(t)) \\ -A_m x_f(t) - B_c r_f(t-\tau)). \tag{12}$$

This signal arises from the known portion of a filtered version of the vehicle dynamics. Note that the value of $Y(t)$ can be computed at any time t using the state $x(t)$, the filtered state $x_f(t)$, and the reference signal delayed by τ seconds $r(t-\tau)$. The filtered state x_f allows us to write (12) as an algebraic expression of x and x_f, rather than a differential expression including an $\dot{x}$ term. Thus, this approach does not require measurement of the state derivative. The bilinear predictor model is given by

$$\hat{Y}(t) = \hat{\Lambda}(t)\left(u_f(t-\tau) + \hat{\theta}^T(t)\omega_f(t)\right). \tag{13}$$

This is an estimate of the unknown signal $\Lambda(t)(u_f(t-\tau) + \theta^T(t)\omega_f(t)) = Y(t)$. Therefore, the error signal e_Y represents the error between the filtered plant dynamics and our prediction of the same. Also note that the time-shifted signals in (12) and (13) require that the time delay τ be known. For the quadrotor system, the total delay in the loop, including computation and communication delays, is 40 ms.

The indirect adaptive part is then added to the adaptive laws in (10), resulting in the CMRAC laws

$$\begin{aligned} \dot{\hat{\theta}}(t) &= \Gamma_\theta \left(\omega(t)e^T(t)PB - \omega_f(t)\gamma_c e_Y^T(t)\right) \\ \dot{\hat{\Lambda}}^T(t) &= -\Gamma_\Lambda \left(u_f(t) + \hat{\theta}^T(t)\omega_f(t)\right)\gamma_c e_Y^T(t). \end{aligned} \tag{14}$$

Note that the parameters are now adjusted according to the tracking error e, as in the MRAC case, as well as the prediction error e_Y. Additionally, the parameters $\hat{\Lambda}$ provide an estimate of the vehicle health. This signal can be used to inform higher level path planning and task allocation systems, enabling damaged UAVs to be commanded to return to base for repairs as in [31]. It can be seen that if the gain on the indirect part γ_c is set to 0, the above equations reduce to those of (10). Further details and stability proofs can be found in [30].

CMRAC has been observed in simulation of time-delay systems [32], [33] to allow for increased adaptive gains without exciting high frequencies. This can be attributed to a low-pass filtering effect on the parameter estimates that are introduced by the indirect adaptation [30]. Fig. 6 shows flight recorded control input data from the MRAC and CMRAC approaches for a 25% collective failure while executing a hover task. The MRAC and CMRAC systems are given the same adaptive gains. It can be seen that the MRAC approach exhibits undesirable oscillations which are not damped, eventually leading to instability. It was observed that the CMRAC system could support adaptive gains 25%–50% higher than those of the MRAC system.

VI. CMRAC Flight Test Results

Flight tests were conducted in the RAVEN testbed as described in Section III using the baseline controller, MRAC design, and CMRAC design. The CMRAC controller was implemented in C^{++} and run alongside the baseline controller. Using this setup, it is possible to compare baseline, MRAC, and CMRAC. Adaptive gains were selected for the MRAC and the CMRAC approaches individually to give maximum performance without exciting high frequencies in the loop.

The test scenario considered is a tracking problem, which involves the execution of a series of altitude step commands. During the test, an actuator anomaly is injected into the system in the form of a loss of collective thrust, that is, a simultaneous loss-of-thrust in all four propellers. This is representative of a sudden change in mass, or alternatively a battery failure or electrical fault. The extent of the simulated failure corresponds to a 25% reduction in thrust. More severe failures would more effectively highlight the advantages of the MRAC and CMRAC approaches. However, thrust loss of more than 25% is not possible with this vehicle because it does not have the payload capacity to maintain hover with less than 75% of nominal thrust.

Fig. 7 shows flight recorded altitude tracking data for the baseline, MRAC, and CMRAC approaches. The failure occurs at around $t = 26$ s during an aggressive upward step command. MRAC and CMRAC quickly accommodate the failure and continue to climb to the desired altitude.

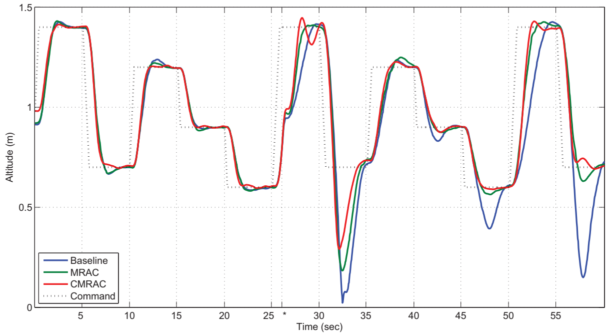


Fig. 7. Altitude tracking performance of the baseline, MRAC, and CMRAC controllers. Loss-of-thrust uncertainty occurs at approximately $t = 26$ s (denoted by the asterisk).

TABLE I

COMPARISON OF TRACKING ERROR FOR THE TRACKING PROBLEM WITH A 25% LOSS OF COLLECTIVE CONTROL EFFECTIVENESS

	Baseline	MRAC	CMRAC
Before failure	63.03	62.47	**56.21**
Transient	82.68	68.27	**59.52**
After failure	123.4	83.92	**71.96**

Even the baseline controller, which contains integral action on the altitude state, is able to complete the climb eventually.

However, the difference in performance of the three approaches is made clearer in the subsequent downward step command. The spike in the baseline controller altitude at around $t = 33$ s corresponds to the vehicle bouncing off the ground. Due to plastic feet on the bottom of the vehicle, it was not permanently damaged and was able to complete the maneuver. Both MRAC and CMRAC experience significant overshoot as well, with CMRAC being the lesser of the two. This suggests that neither adaptive controller had sufficiently learned the extent of the parameter uncertainty during the climb after the failure is inserted. This is unsurprising given that the constant altitude command is not persistently exciting and parameter convergence is therefore not guaranteed [20].

To further analyze the wealth of information contained within the flight data shown in Fig. 7, we break up the time history into three flight conditions: before failure ($t = 0-25$ s), transient ($t = 25-35$ s), and after failure ($t = 35-60$ s). Table I shows the tracking performance of the three controllers in each of these three conditions. CMRAC displays an improved performance over all three flight conditions. The improvement in tracking performance even before the failure occurs can be attributed to the fact that the reference model is, in reality, slightly faster than the baseline system. The performance increases of CMRAC during the transient and after the failure can be attributed to the improved learning of the CMRAC approach, as well as the 25%–50% higher adaptive gains that are possible with CMRAC.

TABLE II

COMPARISON OF OVERSHOOT FOR THE TRACKING PROBLEM WITH A 25% LOSS OF COLLECTIVE CONTROL EFFECTIVENESS

		Baseline	MRAC	CMRAC
Before failure	Upward	7.78%	4.26%	**1.17%**
	Downward	4.13%	5.39%	**1.07%**
Transient	Upward	1.93%	**1.32%**	5.82%
	Downward	96.9%	73.8%	**58.5%**
After failure	Upward	5.19%	6.55%	**4.46%**
	Downward	56.9%	10.2%	**4.36%**

Overshoot for the same tests is shown in Table II. In this table, the flight is further broken down into upward steps and downward steps to highlight some of the differences between the three approaches. CMRAC has the lowest overshoot with the exception of the climb portion of the transient flight condition, which is caused by a small oscillation at around $t = 28$ s.

In addition to examining the magnitude of the overshoot, it is also informational to examine the difference between upward and downward steps for the various controllers. Before the failure occurs, all three controllers have similar overshoot and “undershoot.” This is the expected result for a linear system. However, after the failure occurs, a disparity develops between the over- and under-shoot. Since the baseline controller has very limited learning, it continues to exhibit very large undershoots, particularly noticeable around $t = 57$ s on Fig. 7. In the after failure flight condition, the disparity between overshoots and undershoots of the baseline controller is 51%. For the MRAC controller, it is 3.7%, and for CMRAC, 0.1%. The CMRAC approach is able to learn the true values of the parameters better than MRAC, and it is therefore able to render the performance of the closed-loop adaptive system closer to that of the reference model (for which no distinction exists between upward and downward steps). The baseline controller, which has no learning capability outside of the integrator in the altitude loop, exhibits the largest disparity between overshoot and undershoot.

Of the three controllers compared, the CMRAC approach displays an improved tracking error and decreased overshoot over all three flight conditions.

VII. CONCLUSION

In this brief, a description of an adaptive controller based on Lyapunov stability and its application to a quadrotor UAV was presented. Flight testing was carried out in an indoor test facility using linear baseline, direct model reference adaptive control, and combined/composite model reference adaptive control. It was shown that the adaptive controller offers several benefits over the fixed-gain approach, particularly in the case of actuator failures. For a severe loss-of-thrust failure, the adaptive controller allowed for safe operation and landing, while the baseline controller failed to prevent instability and resulted in a crash landing. In the following sections, a CMRAC controller was demonstrated that delivered smoother parameter estimates, allowed for higher adaptive gains, and increased performance over that of MRAC alone. It was shown that CMRAC was more effective than MRAC in learning the true value of uncertain parameters in the system, offering numerous benefits in terms of tracking performance.

The CMRAC system also exposes vehicle health estimates that can be used by higher level autonomy.

It should be noted that the approach described herein does not exclude the application of robust control and gain scheduling techniques. Future work could involve the combination of CMRAC with gain scheduling, for example. Each of these approaches represents another tool in the toolbox that controls engineers can use to solve these difficult problems.

Acknowledgment

The authors would like to thank J. How, J. Redding, and B. Micini, Aerospace Controls Laboratory, MIT, Cambridge, for the use of the real-time indoor autonomous vehicle test environment testbed, and for their insightful comments that were invaluable in carrying out the experiments reported in this brief.

References

[1] J. Leishman, *Principles of Helicopter Aerodynamics*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[2] S. B. Anderson, "Historical overview of V/STOL aircraft technology," Ames Research Center, NASA, Moffett Field, CA, Tech. Memo. D-1060, Jun. 1981.

[3] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 5. May 2004, pp. 4393–4398.

[4] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proc. AIAA Guid., Navigat., Control Conf.*, Hilton Head, SC, 2007, pp. 1–20.

[5] P. McKerrow, "Modelling the Draganflyer four-rotor helicopter," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, Apr.–May 2004, pp. 3596–3601.

[6] P. Castillo, R. Lozano, and A. Dzul, *Modelling and Control of Mini-Flying Machines*. New York: Springer-Verlag, 2005.

[7] G. Tournier, M. Valenti, J. How, and E. Feron, "Estimation and control of a quadrotor vehicle using monocular vision and moirè patterns," in *Proc. AIAA Guid., Navigat. Control Conf.*, Keystone, CO, 2006, pp. 21–24.

[8] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron, "Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery," in *Proc. AIAA Guid., Navigat., Control Conf.*, Keystone, CO, 2006, pp. 1–18.

[9] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Syst. Mag.*, vol. 28, no. 2, pp. 51–64, Apr. 2008.

[10] B. Bethke, J. P. How, and J. Vian, "Group health management of UAV teams with applications to persistent surveillance," in *Proc. Amer. Control Conf.*, 2008, pp. 3145–3150.

[11] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," in *Proc. Robot.: Sci. Syst.*, Seattle, WA, Jun. 2009, pp. 73–86.

[12] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro-UAV testbed," *IEEE Robot. Autom. Mag.*, vol. 17, no. 3, pp. 56–65, Sep. 2010.

[13] A. Benallegue, A. Mokhtari, and L. Fridman, "High-order sliding-mode observer for a quadrotor UAV," *Int. J. Robust Nonlinear Control*, vol. 18, no. 4, pp. 427–440, 2008.

[14] S. Lupashin, A. Schollig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1642–1648.

[15] M. Hua, T. Hamel, P. Morin, and C. Samson, "A control approach for thrust-propelled underactuated vehicles and its application to VTOL drones," *IEEE Trans. Autom. Control*, vol. 54, no. 8, pp. 1837–1853, Aug. 2009.

[16] I. Sadeghzadeh, A. Mehta, and Y. Zhang, "Fault/damage tolerant control of a quadrotor helicopter UAV using model reference adaptive control and gain-scheduled PID," in *Proc. AIAA Guid., Navigat., Control Conf.*, Portland, OR, 2011, pp. 1–20.

[17] K. J. Åström and B. Wittenmark, *Adaptive Control*. Boston, MA: Addison-Wesley, 1995.

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

[19] I. D. Landau, *Adaptive Control: The Model Reference Approach*. New York: Marcel Dekker, 1979.

[20] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[21] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[22] K. Tsakalis and P. Ioannou, *Linear Time-Varying Systems: Control and Adaptation*. Upper Saddle River, NJ: Prentice-Hall, 1993.

[23] J. Pomet and L. Praly, "Adaptive nonlinear regulation: Estimation from the Lyapunov equation," *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 729–740, Jun. 1992.

[24] Z. T. Dydek, H. Jain, J. Jang, A. M. Annaswamy, and E. Lavretsky, "Theoretically verifiable stability margins for an adaptive controller," in *Proc. AIAA Conf. Guid., Navigat., Control*, Keystone, CO, Aug. 2006, no. AIAA-2006-6416, pp. 1–14.

[25] Z. T. Dydek, "Adaptive control of unmanned aerial systems," Ph.D. dissertation, Dept. Mech. Eng., Massachusetts Institute Technology, Cambridge, 2010.

[26] Active Adaptive Control Laboratory. (2010, May 27). *Adaptive Control of a Quadrotor UAV*, Cambridge, MA [Online]. Available: http://www.youtube.com/watch?v=FhgMy4ss0bw

[27] J. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, no. 4, pp. 509–519, 1989.

[28] M. Duarte-Mermoud, J. Rioseco, and R. Gonzalez, "Control of longitudinal movement of a plane using combined model reference adaptive control," *Aircraft Eng. Aerosp. Technol.: Int. J.*, vol. 77, no. 3, pp. 199–213, 2005.

[29] M. Duarte and K. Narendra, "Combined direct and indirect approach to adaptive control," *IEEE Trans. Autom. Control*, vol. 34, no. 10, pp. 1071–1075, Oct. 1989.

[30] E. Lavretsky, "Combined/composite model reference adaptive control," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2692–2697, Nov. 2009.

[31] J. Redding, Z. Dydek, J. How, M. Vavrina, and J. Vian, "Proactive planning for persistent missions using composite model-reference adaptive control and approximate dynamic programming," in *Proc. Amer. Control Conf.*, 2011, pp. 2332–2337.

[32] Z. T. Dydek, A. M. Annaswamy, J.-J. E. Slotine, and E. Lavretsky, "High performance adaptive control in the presence of time delays," in *Proc. Amer. Control Conf.*, Baltimore, MD, 2010, pp. 880–885.

[33] Z. T. Dydek, A. M. Annaswamy, J.-J. E. Slotine, and E. Lavretsky, "Time delay resistant adaptive control of mini-UAVs," in *Proc. IFAC Workshop Time Delay Syst.*, Prague, Czech Republic, 2010, pp. 27–33.