$\mathcal{L}_1$ Adaptive Control for Indoor Autonomous Vehicles: Design Process and Flight Testing

Buddy Michini* and Jonathan P. How†
Aerospace Controls Laboratory
Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Adaptive control techniques have the potential to address many of the special performance and robustness requirements of flight control for unmanned aerial vehicles. $\mathcal{L}_1$ adaptive control offers potential benefits in terms of performance and robustness. An $\mathcal{L}_1$ adaptive output feedback control design process is presented here in which control parameters are systematically determined based on intuitive desired performance and robustness metrics set by the designer. Flight test results verify the process for an indoor autonomous quadrotor helicopter, demonstrating that designer specifications correspond to the expected physical responses. In flight tests comparing it with the baseline linear controller, the augmented adaptive system shows definite performance and robustness improvements confirming the potential of $\mathcal{L}_1$ adaptive control as a useful tool for autonomous aircraft.

I. Introduction

Unmanned aerial vehicles (UAVs) have become increasingly prominent in a variety of aerospace applications. The need to operate these vehicles in potentially constrained environments and make them robust to actuator failures and plant variations has brought about a renewed interest in adaptive control techniques. Model Reference Adaptive Control (MRAC) has been widely used, but can be particularly susceptible to time delays.[1] A filtered version of MRAC, termed $\mathcal{L}_1$ adaptive control, was developed to address these issues and offer a more realistic adaptive solution.[2,3].

The main advantage of $\mathcal{L}_1$ adaptive control over other adaptive control algorithms such as MRAC is that $\mathcal{L}_1$ cleanly separates performance and robustness. The inclusion of a low-pass filter not only guarantees a bandwidth-limited control signal, but also allows for an arbitrarily-high adaptation rate limited only by available computational resources. This parameterizes the adaptive control problem into two very realistic constraints: actuator bandwidth and available computation. In this paper we consider the *output feedback* version of $\mathcal{L}_1$ described in [4]. This single-input single-output (SISO) formulation has several advantages. Foremost, the internal system states need not be modeled or measured. All that is required is a SISO input-output model that can encompass the entire closed-loop system and be acquired using simple system identification techniques. Thus the adaptive controller can be "wrapped around" an already-stable closed-loop system, adding performance and robustness in the face of plant variations. It is also easy to predict the time-delay margin using standard linear systems analysis, and this margin has been confirmed experimentally. Finally, output-feedback $\mathcal{L}_1$ is relatively easy to implement in practice as will be seen in the experimental sections.

*Ph.D. Candidate, Aerospace Controls Laboratory, MIT, bmich@mit.edu
†Professor, MIT Dept. of Aeronautics and Astronautics, jhow@mit.edu, Associate Fellow AIAA

One consequence of using this output feedback (as opposed to full-state feedback) form of $\mathcal{L}_1$ is that the expected closed-loop response becomes somewhat complex. Whereas in full-state form the reference model sets the desired system behavior, with output feedback $\mathcal{L}_1$ it is not immediately clear how to choose the design parameters to achieve some desired response. Previous efforts have focused on norm minimization and time delay optimization via modification of only the low-pass control signal filter.[2,3,5] In [6], more than just the low-pass filter is considered, but the analysis again relies on a system norm as the only performance metric. Metrics are validated through extensive flight testing in [7], but these metrics are not considered in an *a priori* control design process. None of the approaches listed comprises a systematic design process that considers both transient performance and robustness simultaneously. Such a design process would be a key step for further application of $\mathcal{L}_1$ adaptive output feedback control in real-world applications including indoor autonomous flight.

The work presented here proposes a design process by which the designer specifies important performance and robustness metrics for the control task at hand. A multi-objective optimization is then performed to systematically choose the $\mathcal{L}_1$ design parameters. The design process is validated through flight testing using the RAVEN testbed, and flight results are presented that demonstrate qualitative performance and robustness improvements when a baseline linear controller is augmented with $\mathcal{L}_1$ adaptive control.

The paper is structured as follows. Section II provides an overview of $\mathcal{L}_1$ adaptive output feedback and describes the challenge of choosing the $\mathcal{L}_1$ parameters. In Section III, a design process is proposed by which these parameters can be chosen based on specified performance and robustness metrics. Section IV describes the experimental setup used to conduct flight tests. Section V presents flight test results that both validate the design process and demonstrate noticeable improvements with the use of $\mathcal{L}_1$ control. Finally, Section VI summarizes relevant conclusions and lays out a plan for future work in this area.

II. $\mathcal{L}_1$ Adaptive Output Feedback

This section provides a brief overview of $\mathcal{L}_1$ adaptive output feedback control,[4,8] which is the adaptive algorithm used herein. Derivation of the predicted time-delay margin is presented, as well as the expected closed-loop system response. Challenges in choosing the $\mathcal{L}_1$ control parameters $C(s)$ and $M(s)$ are discussed as a motivation for the proposed design process.

II.A. Overview

Figure 1 shows the block diagram for the adaptive controller. The disturbance $d(s)$ is used to represent any type of non-linear disturbance, and thus can represent not only external disturbances but also changes in the plant $A(s)$ due to parameter variations or actuation failures. Note that if $C(s) = 1$, a high-gain PI controller is recovered. If this were the case, $\hat{\sigma}$ (the adaptive signal) would simply be "whatever it takes" to make the output of $A(s)$ match the output of $M(s)$. The τ block represents a typical sensor measurement time delay, and will be used later in the characterization of robustness metrics.

Adding the low-pass filter $C(s)$ does two important things. First, it limits the bandwidth of the control signal u being sent to the plant. This prevents high-bandwidth oscillatory control signals (as are often seen with fast-adapting MRAC controllers) from being commanded. Second, the portion of $\hat{\sigma}$ that gets sent into the reference model is the *high-frequency portion* (note that the low-pass version is subtracted from the full signal before being sent to the reference model $M(s)$). This signal, in a sense, corresponds to the portion of the disturbance $d(s)$ that can not be canceled given the limited actuator bandwidth. The fact that this is sent into the reference model $M(s)$

along with the reference signal implies that the output of $M(s)$ is the *achievable* system output, a realistic goal that the system should be able to match given its bandwidth constraints.

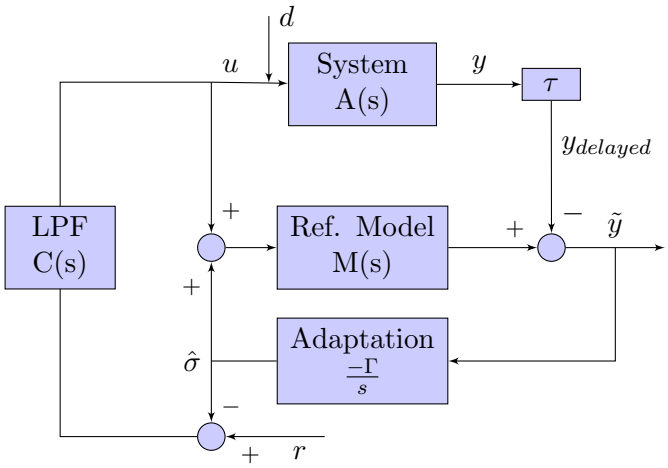

Figure 1: $\mathcal{L}_1$ adaptive output feedback control block diagram.

II.B. Control Parameters

The user-specified parameters of the $\mathcal{L}_1$ controller are the low-pass filter $C(s)$, the reference model $M(s)$, and the adaptation rate Γ. It is clear that $C(s)$ should be chosen such that its bandwidth does not exceed that of the available actuators. The adaptation rate Γ is essentially the gain of the adaptive estimator, and since the control signal is low-pass filtered a very large value can be used (for example, $\Gamma = 10000$ in the flight tests in Section V). Since the controller must still be implemented in real-time on a computer, Γ is limited in practice by the stability of the numerical integration which is determined largely by available computational capabilities. As will be discussed below, the choice of $M(s)$ is not so straightforward for achieving the desired specifications.

II.C. Predicted Time Delay Margin

It is helpful to be able to predict the adaptive controller's margin to a time delay on the output measurement $y(s)$ (corresponding to some known sensor delay). While this analysis has previously been done for the general $\mathcal{L}_1$ control setup[5], it has not been shown explicitly for the output feedback case considered here. Since the output feedback system is comprised of SISO linear blocks, the time delay margin of the system can be calculated as the ratio of phase margin to cross-over frequency of the appropriate system. From Figure 1, the system of interest is the system whose input is y_{delayed} and whose output is y, assuming that $r = 0$ and $d = 0$. It is easiest to analyze this system using state space techniques instead of transfer functions. Let $[A_A, B_A, C_A, 0]$, $[A_C, B_C, C_C, 0]$, and $[A_M, B_M, C_M, 0]$ be the state space representations of $A(s)$, $C(s)$, and $M(s)$, respectively. It can be verified that the system with input y_{delayed} and output y has the following state space representation:

$$A = \begin{bmatrix} A_A & 0 & B_A C_C & 0 \\ 0 & A_M & B_M C_C & B_M \\ 0 & 0 & A_C & -B_C \\ 0 & -\Gamma C_M & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \Gamma \end{bmatrix} \quad C = \begin{bmatrix} C_A & 0 & 0 & 0 \end{bmatrix} \quad D = [0] \tag{1}$$

The time delay margin is then calculated by taking the ratio of the phase margin to the cross-over frequency, both of which can be deduced from a Bode plot of the system. Determination of the time-delay margin using this method has been confirmed both in simulation and experiment.

II.D. Expected Closed-loop Response

An analysis of the $\mathcal{L}_1$ output feedback system is provided in [6] where it is shown that, if the disturbance is known exactly (i.e. Γ is sufficiently large such that the adaptive estimator is doing its job perfectly), the expected closed-loop response becomes:

$$y(s) = \underbrace{H(s)C(s)r(s)}_{\text{Response to reference } r(s)} - \underbrace{H(s)(1-C(s))d(s)}_{\text{Response to disturbance } d(s)} \tag{2}$$

$$\text{where } H(s) = \frac{A(s)M(s)}{C(s)A(s) + (1-C(s))M(s)} \tag{3}$$

Recall that in MRAC the expected closed-loop response is simply $y(s) = M(s)r(s)$. Now, even if the disturbance is known perfectly, the expected closed-loop response is a complicated function of $A(s)$, $C(s)$, and $M(s)$.

II.E. Selection of $C(s)$ and $M(s)$

As seen from Eq. 2, the inclusion of a low-pass filter in the control path obscures the expected closed-loop dynamics. Since $M(s)$ no longer acts as a reference model, it is not obvious how its choice affects the system response. In other words, while it is a design parameter of the $\mathcal{L}_1$ controller, it cannot simply be chosen as the reference model like in MRAC. While the bandwidth of $C(s)$ should be upper-bounded by the bandwidth of the corresponding actuator, there is no intuitive lower bound.

This brings about the need for a systematic method of choosing the filters $C(s)$ and $M(s)$ based on desired performance and robustness metrics. Performance measures may include transient response characteristics such as rise time, overshoot, or cross-over frequency. Robustness measures may include metrics such as time delay margin and disturbance rejection.

III. Design Process

In this section, a design process is proposed by which $C(s)$ and $M(s)$ are selected in a systematic way based on desired performance and robustness metrics. This is achieved by performing multi-objective optimization on a weighted cost function comprised of these metrics, which also requires basic system identification of the baseline plant $A(s)$.

III.A. Specifying Performance and Robustness Metrics

The first step in this design process is to identify performance and robustness metrics relevant to the control task at hand. For the dynamic control of indoor autonomous flight vehicles, three important metrics are considered. **Transient performance** characteristics such as rise time and overshoot provide intuitive measures of the closed-loop system response. For instance, these can be chosen to set the aggressiveness of the controller based on mission requirements. For application to real-world physical systems, **time delay margin** is a very important measure of robustness. Some minimum time delay margin must be achieved based on known delays in the sensing, estimation, computation, and actuation of the physical system. Finally, the general goal of adaptive control is to maintain nominal performance in lieu of disturbances such as actuation failures or plant parameter variations. Thus, it is important to somehow specify a desired degree of **disturbance rejection**.

III.A.1. Transient Performance

As shown in Eq. 2, the expected closed-loop response $y(s)$ to the reference input $r(s)$ is given by the transfer function $H(s)C(s)$. Thus, the design process should shape the transient response of this system. The response can be affected by weighting a combination of standard step-response characteristics which may include rise time, overshoot, or settling time. Frequency-domain metrics may also be used such as the bandwidth of $H(s)C(s)$, which in this case would roughly represent the frequency range through which the output $y(s)$ matches the reference input $r(s)$. Calculation of these metrics is fairly straight-forward either by simulating the closed-loop system (more time-consuming) or via second-order approximation (less accurate).

III.A.2. Time Delay Margin

The time delay margin of the adaptive controller can be calculated as in Section II.C. As will be seen, $A(s)$ will often represent the nominal *closed-loop* system which may include internal feedback loops from the baseline controller. It should be noted that, for analytical simplicity, the margin considered here only considers a time delay on the output signal $y(s)$ being sent back to the adaptive controller, and not on any of the feedback signals internal to $A(s)$. Even so, this time delay margin still provides a good indication of how robust the adaptive controller will be to feedback delays in general. In addition, since only the adaptive controller is being designed here, use of this measure also offers analytical separation between the effect of time delay on the adaptive controller and the potentially complex effects on the nominal system.

III.A.3. Robustness to Disturbances

As shown in Eq. 2, the expected closed-loop response $y(s)$ to the disturbance $d(s)$ is given by the transfer function $H(s)(1-C(s))$. Note that $d(s)$ can be an arbitrary signal (restricted only in its Lipschitz norm[6]), thus it can represent many types of disturbances including actuation failures, parameter variations, and exogenous factors. To minimize the effects of these disturbances, the norm of $H(s)(1-C(s))$ should be minimized. The $\mathcal{L}_1$ norm has typically been used,[2,3] so the measure of disturbance rejection used here is chosen to be $\|H(s)(1-C(s))\|_{\mathcal{L}_1}$, a smaller norm corresponding to increased robustness to disturbances.

III.B. System Identification

Since the expected closed-loop response from Eq. 2 includes the nominal system $A(s)$, some knowledge of this system is required to design the adaptive controller. As will be seen in Section IV.C, $A(s)$ can be taken to be the nominal closed-loop system, which includes the baseline controller. Since this system is typically stable and well-behaved by design, system identification from experimental data is relatively easy.

Take, for example, the quadrotor helicopters used in Section V. The system $A(s)$ in this case represents the closed-loop response of the system from x-velocity reference command to x-velocity measured output. Using data from routine flight tests, the following second-order system model was fit using an ARMAX-type regression:

$$\frac{V_{x_{meas}}}{V_{x_{ref}}}(s) = \frac{k(s-z_1)}{(s-p_1)(s-p_2)} = \frac{1.80(s+0.44)}{(s+0.89)(s+0.89)} \tag{4}$$

As can be seen in Figure 2, this simple second-order model adequately characterizes the closed-loop response.

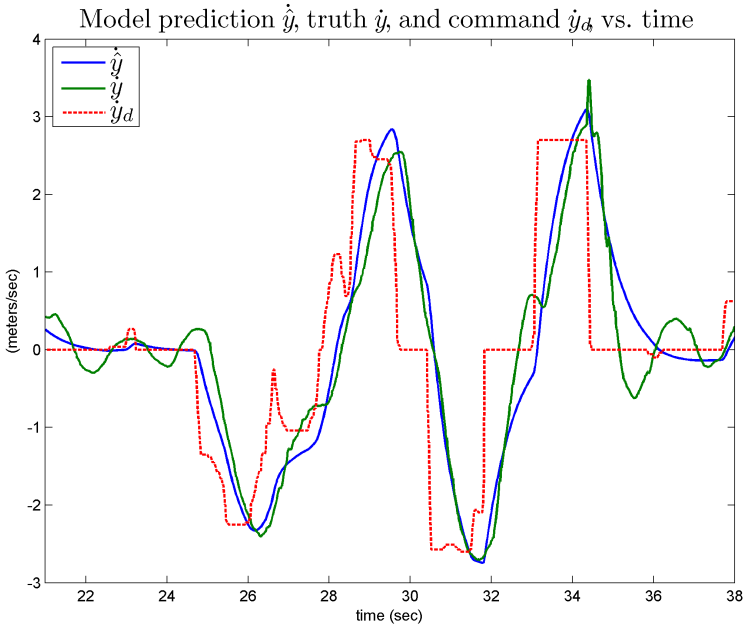

Figure 2: System identification results for quadrotor closed-loop velocity system. User-generated reference command (red), measured output (green), and predicted output based on system ID (blue).

III.C. Multi-Objective Optimization

The desired performance and robustness metrics are combined into a cost function that can be minimized in an effort to calculate $C(s)$ and $M(s)$. The optimization process must be provided with the cost function, relevant constraints (like system stability), a model of $A(s)$ from system identification, and some parametrization of $C(s)$ and $M(s)$. Figure 3 shows a general diagram of the optimization process.

III.C.1. Cost Function and Constraints

The cost function considered here is a simple weighted combination of the performance and robustness metrics listed above. For example, a possible cost function might be:

$$J = \alpha_1(\text{Rise Time}) + \alpha_2(\text{Overshoot}) + \alpha_3\left(\frac{1}{\text{TD margin}}\right) + \alpha_4\left(\|H(s)(1-C(s))\|_{\mathcal{L}_1}\right) \tag{5}$$

Since the cost function is to be minimized, α_i represents the *penalty* on the associated metric. Two constraints must be considered. First, the bandwidth of $C(s)$ must be limited to the bandwidth of the associated actuator. In the case of the quadrotor velocity controller, the "actuator" is the closed-loop system given by (4), thus $C(s)$ must be limited in bandwidth to that of (4). The second constraint to be considered is the stability of the expected system response $H(s)C(s)$. One simple way to implement this constraint is to augment the cost function with a term that is arbitrarily large if the system is unstable. The test for stability used in the experimental validation below is simply a check of whether the roots of $H(s)C(s)$ are strictly negative.

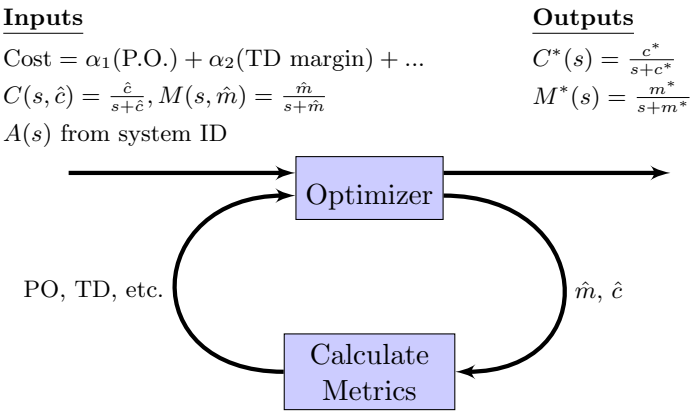

Figure 3: Multi-objective optimization diagram

III.C.2. Filter Parametrization

$C(s)$ and $M(s)$ must be parameterized such that the cost function can be minimized over these parameters. A particularly straightforward parametrization is:

$$C(s,\hat{c}) = \frac{\hat{c}}{s+\hat{c}} \tag{6}$$

$$M(s,\hat{m}) = \frac{\hat{m}}{s+\hat{m}} \tag{7}$$

Using this parametrization, each transfer function has only one parameter associated with it. While this may limit performance in some ways, it greatly simplifies the optimization process and provides for an intuitive initial design iteration. Use of higher-order filters is an open topic of research[6]. It will be shown in Section V, however, that these simple filter parameterizations are intuitive, easy to implement, and work quite well in practice.

III.C.3. Solution Methods and Limitations

Once the cost function, constraints, and parameterized solution forms are specified, the designer is free to use any constrained optimization technique to solve for $C(s)$ and $M(s)$. One obvious drawback to this process, though, is that the cost function presented is almost certainly non-convex with respect to the parameterized transfer functions. This is not surprising considering that characteristics like overshoot and time-delay margin are summed in the same function. Typical solvers, like Matlab®'s `fmincon`, work well for a small number of parameters, but exhibit poor convergence properties as the parametrization complexity increases.

Using filters in the form of (7) is thus advantageous since typical solvers usually converge to a global minimum. It is easy to then verify this minimum if necessary using exhaustive search methods when there are only two parameters. Furthermore, the cost as a function of the two parameters ($\hat{c}$ and $\hat{m}$) can be visualized using a contour plot, making the process more intuitive to the designer. Ongoing work (see Section VI) aims to address the non-convexity issue by attempting to cast each performance and robustness metric as a linear matrix inequality (LMI) constraint and performing the optimization with much more efficient search methods. This has been done for the state feedback $\mathcal{L}_1$ adaptive control formulation[9], but has not been extended to the output feedback case considered here.

IV. Experimental Setup

This section describes the experimental setup used for the flight test results to follow. The Realtime Autonomous Vehicle test ENvironoment (RAVEN) testbed is presented along with the specific vehicles flown. The baseline control strategy is discussed, as well as the augmented adaptive controller.

IV.A. The RAVEN Testbed

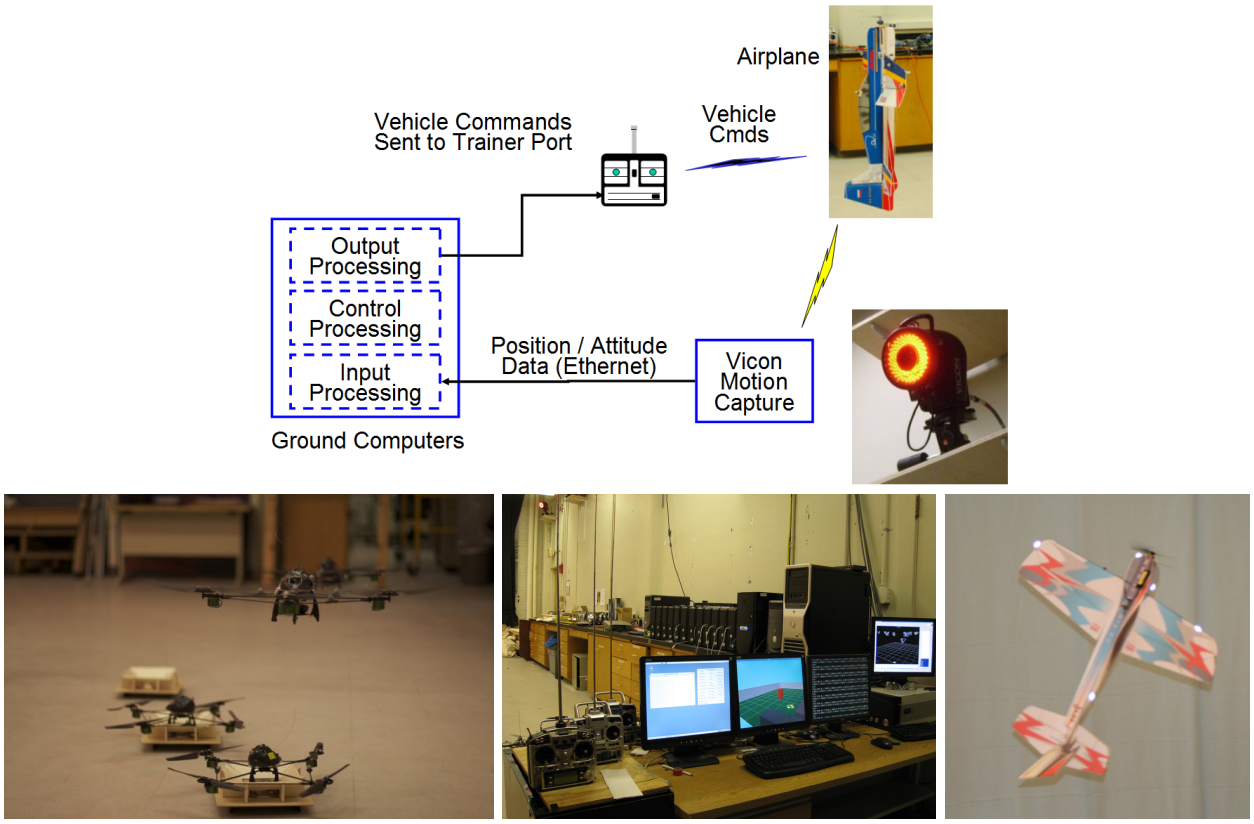

Figure 4: RAVEN system architecture (top), Quadrotor helicopters (bottom left), control computers (bottom middle), Clik aerobatic aircraft (bottom right).

Figure 4 presents the control architecture used for all of the flight experiments given in this paper. Note that while only several flying vehicles are shown, the system can support up to ten (10) vehicles flying simultaneously[10]. A key feature of RAVEN is the motion capture system[10–12] that can accurately track all vehicles in the room in real-time. With lightweight reflective balls attached to each vehicle's structure, the motion capture system can measure the vehicle's position and attitude information at rates up to 120 Hz, with approximately a 15-25 ms delay, and sub-mm accuracy[10,11]. RAVEN currently has motion capture systems from both Vicon[a] and Motion Analysis[b].

Flight control commands are computed using ground-based computers at rates that exceed 50 Hz and sent to the vehicles via standard Radio Control (R/C) transmitters. An important feature of this setup is that small, inexpensive, essentially unmodified, radio-controlled vehicles can

[a] http://www.vicon.com/
[b] http://www.motionanalysis.com/

be used. This enables researchers to avoid being overly conservative during flight testing. The computer configuration is shown in Figure 4 (bottom middle), with input (vehicle and environment state estimation), planning and control, and output (conversion to R/C commands) processing all done in linked ground computers, as if it were being done onboard.

The combination of simple vehicles, a fast and accurate external metrology/control system, modular onboard payloads, and a well-structured software infrastructure provides a very robust testbed environment that has enabled the demonstration of more than 3000 multi-UAV flights in the past 36 months.

IV.B. Flight Vehicles

For the purposes of this work, two types of indoor autonomous flight vehicles are used, a quadrotor helicopter and a fixed-wing aerobatic aircraft.

IV.B.1. Quadrotor Helicopters

The Draganflyer V Ti Pro quadrotor (Figure 4, bottom left) is a small (∼500g), capable flight vehicle that has been used extensively in RAVEN. While the dynamics of quadrotor can be modeled reasonably well, its four motors are subject to performance variations and partial failures. The goal of adaptive control in this case is to make quadrotor flight more robust to these types of actuator failures. Since the vehicles are controlled autonomously, failures can be simulated mid-flight by scaling the control commands to individual rotors.

IV.B.2. Aerobatic Fixed-Wing Aircraft

The Clik F3P competition plane is shown in Figure 4 (bottom right). It is an extremely light (< 200g) airframe designed for aggressive aerobatic maneuvering. The Clik's high thrust-to-weight ratio (1.5:1) give it the ability to hover in a "prop-hang" configuration and transition smoothly to forward flight. The controller tested here is based on a model linearized about the hover configuration. However, as the aircraft approaches forward flight and its translational speed increases the dynamics change substantially. Adaptive control is applied here in an effort to push the envelope of the hover controller towards forward flight in lieu of these rapidly-changing dynamics.

IV.C. Control Setup

The nominal controller for both the quadrotor and the fixed-wing consists on an outer-loop velocity controller wrapped around an inner-loop attitude controller. The vehicle's translational velocity in the X-Y (horizontal) plane is affected by commanding an appropriate attitude. For example, if a positive x-velocity is desired, the outer-loop velocity controller will command an attitude that "tilts" the vehicle in the x-direction, and the inner-loop attitude controller will attempt to track this commanded attitude. Independent controllers for x- and y-velocity are used, and attitude commands are combined and sent to a single inner-loop attitude controller. The velocity controller is linear Proportional+Integral, while the attitude controller is quaternion-based linear Proportional+Derivative similar to that presented in [13].

Figure 5 shows the baseline x-velocity controller for the quadrotor helicopter augmented with an $\mathcal{L}_1$ adaptive controller. The system in the dashed box represents $A(s)$, the baseline closed-loop system that takes a velocity reference input and produces some measured velocity as an output as described above. This is the system identified in (4). Note that the $\mathcal{L}_1$ controller is completely external to the nominal closed-loop system, augmenting the velocity reference command sent to the baseline controller.

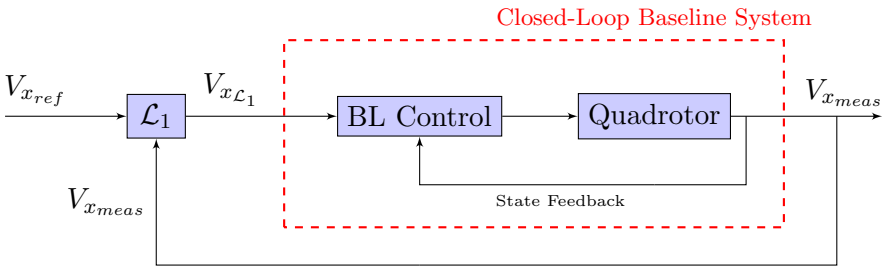

Figure 5: Setup for the quadrotor x-velocity controller with $\mathcal{L}_1$ adaptive augmentation. The system in the dashed lines represents $A(s)$, the baseline closed-loop system.

V. Experimental Results

The design process described in Section III would normally be carried out by weighting the appropriate performance and robustness metrics in accordance with mission requirements. However, to validate the design process experimentally, results are shown that vary *only one* parameter of the cost function at a time. In doing so, modifications to individual components of the cost function can be clearly verified to have the expected result. The cost function chosen to do this is:

$$J = \alpha_1(\text{Rise Time}) + \alpha_2\left(\frac{1}{\text{TD margin}}\right) + \alpha_3(\|H(s)(1-C(s))\|_{\mathcal{L}_1}) \tag{8}$$

Thus α_1 penalizes slow transient performance, α_2 penalizes small time delay margin, and α_3 penalizes poor disturbance rejection. Three flight scenarios are used to test these metrics with the quadrotor: nominal response to a step command, response to a step command with time-delayed sensor measurements, and response to a single-rotor actuator failure to 50% of its original capacity.

Note that while the $\mathcal{L}_1$ adaptive controller is acting on the velocity controller of the baseline system, a simple proportional position controller has been added outside of this. This is done primarily due to space constraints of the testing area, since step commands in position are more predictable than steps in velocity. This also provides insight as to how adaptation on the velocity controller impacts position control. As a result, the following results will show not only the velocity response of the system, but also the position response.

V.A. Nominal Transient Performance

To test nominal transient performance, α_1 was increased while keeping α_2 and α_3 constant, corresponding to an increasing penalty on rise time. Figure 6 shows that the expected behavior is achieved. As the penalty is increased, the position and velocity responses show faster rise times at the cost of more overshoot and less damping.

V.B. Robustness to Time Delay

To test robustness to time delay, α_2 was increased while keeping α_1 and α_3 constant. As can be seen in Figure 7, the increased penalty yields a slower, more well-damped response in lieu of a 90ms sensor measurement delay while the smaller penalty yields a marginally stable response. It should be noted that measurement delay is applied to *all* feedback loops (including the inner-loop controllers) as would be the case for realistic time-delayed systems. Recall from Section II.C that the time delay analysis for the $\mathcal{L}_1$ controller only considered delay on the output feedback signal, not the internal feedback loops. The fact that the expected result is still achieved here implies that the $\mathcal{L}_1$ time delay analysis yields the correct trend even when all of the feedback loops are delayed.

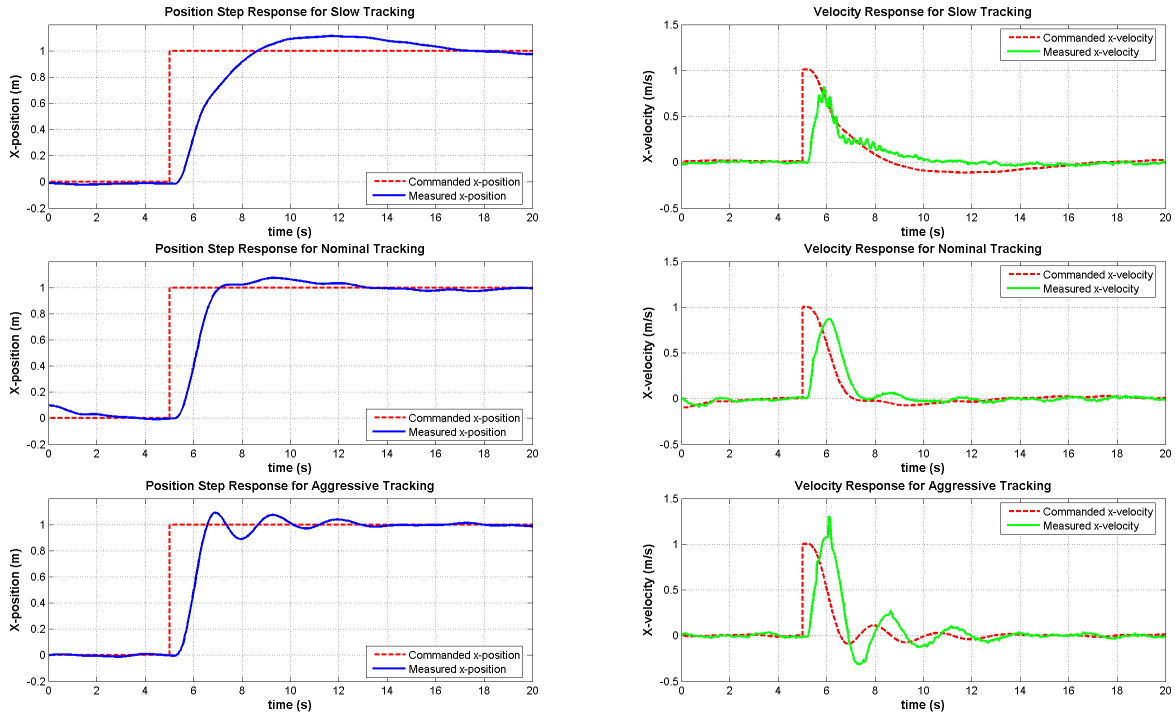

Figure 6: Nominal transient performance flight results, position response (left) and velocity response(right). Parameters chosen for slow tracking (top), nominal tracking (middle), aggressive tracking (bottom).

V.C. Robustness to Actuator Failure

To test robustness to disturbances, α_3 was increased while keeping α_1 and α_2 constant, corresponding to an increased penalty on poor disturbance rejection. In Figure 8, it is clear that the increased penalty yields a stable response after a 50% single-rotor failure, while the lower penalty leads to an unstable response in which the vehicle eventually crashes. While the increased penalty yields stability, it comes at the cost of a slower and more conservative response (note that it initially has a greater maximum error in position). This result highlights the fundamental tradeoff between performance and robustness.

V.D. Comparison to Baseline Controller

As a useful benchmark, flight tests are performed comparing the baseline controller to the augmented $\mathcal{L}_1$ adaptive system. To provide a fair comparison, the adaptive controller is tuned so that the nominal step response for both systems is sufficiently similar (see Figure 9 top). The same adaptive controller is then compared to the baseline controller in the measurement delay (Figure 9 middle) and actuator failure (Figure 9 bottom) scenarios. In the time delay case, the adaptive controller shows noticeably improved tracking in both position and velocity. The same can be said for the actuator failure, as the adaptive controller yields a smaller maximum error in position as well as improved damping in the transient response for both position and velocity. These results confirm that $\mathcal{L}_1$ adaptive control can offer useful performance improvements in the context of indoor autonomous flight.

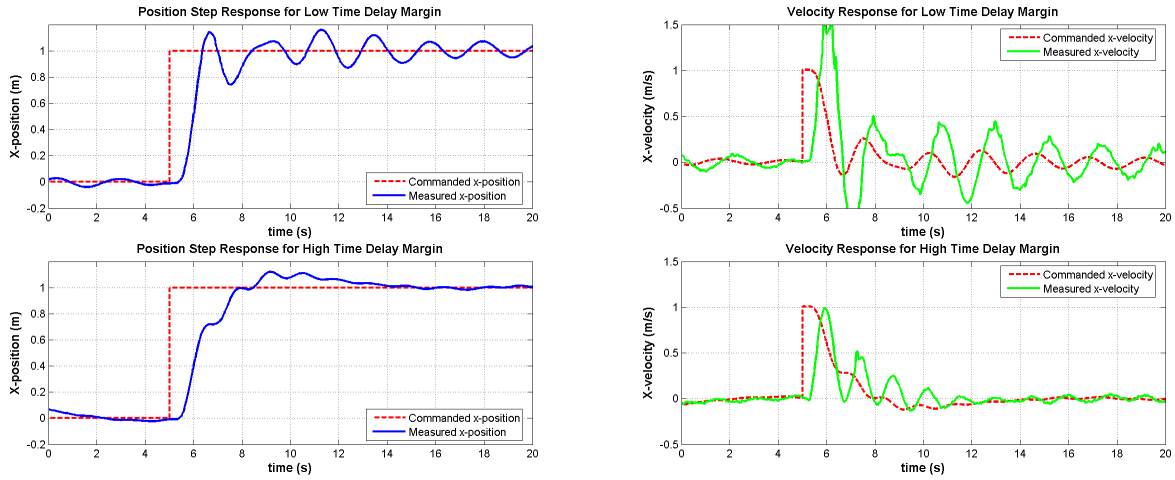

Figure 7: Measurement time delay flight results, position response (left) and velocity response(right). Parameters chosen for less time delay margin (top), and more time delay margin (bottom).

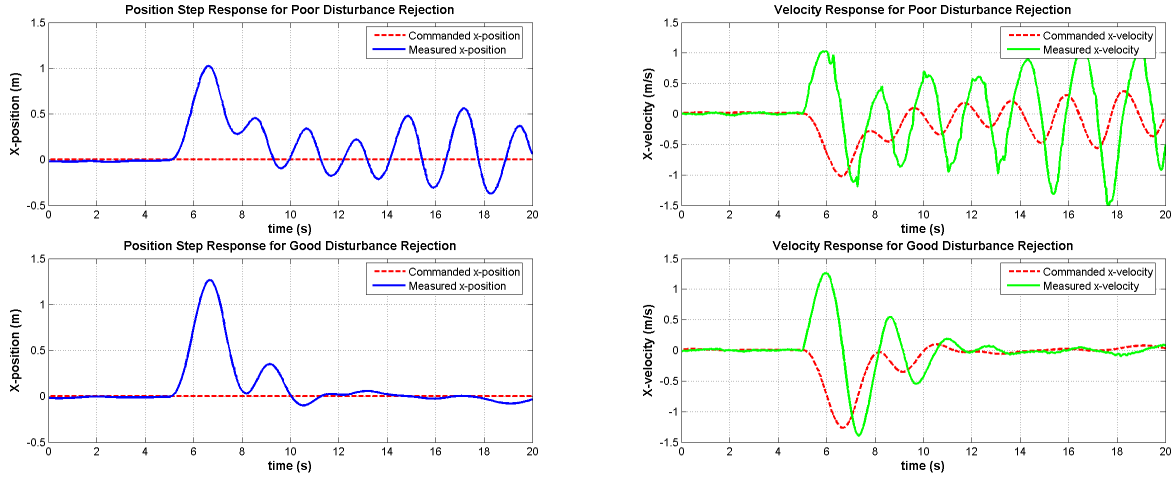

Figure 8: Actuator failure flight results, position response (left) and velocity response(right). Parameters chosen for poor disturbance rejection (top), and good disturbance rejection (bottom).

V.E. Application to Aggressive Flight

As discussed in Section IV.B.2, the dynamics of the Clik aerobatic aircraft change rapidly as translational speed increases from a hover configuration. Figure 10 (left) shows the inability of the linear controller to accurately track forward velocities greater than 1.5 m/s. In situations like this, the gains are typically increased to improve performance of the linear controller. However, an increase in the nominal gains is not possible in this case due to destabilization of the aircraft about hover (the linearization point). $\mathcal{L}_1$ adaptive control is applied here in an attempt to address this issue. In Figure 10 (right), the augmented adaptive controller enables aggressive tracking of up to 3m/s. Note that the large observed overshoot is a function of cost function weighting parameters corresponding to aggressive tracking performance (as is desired in this case).

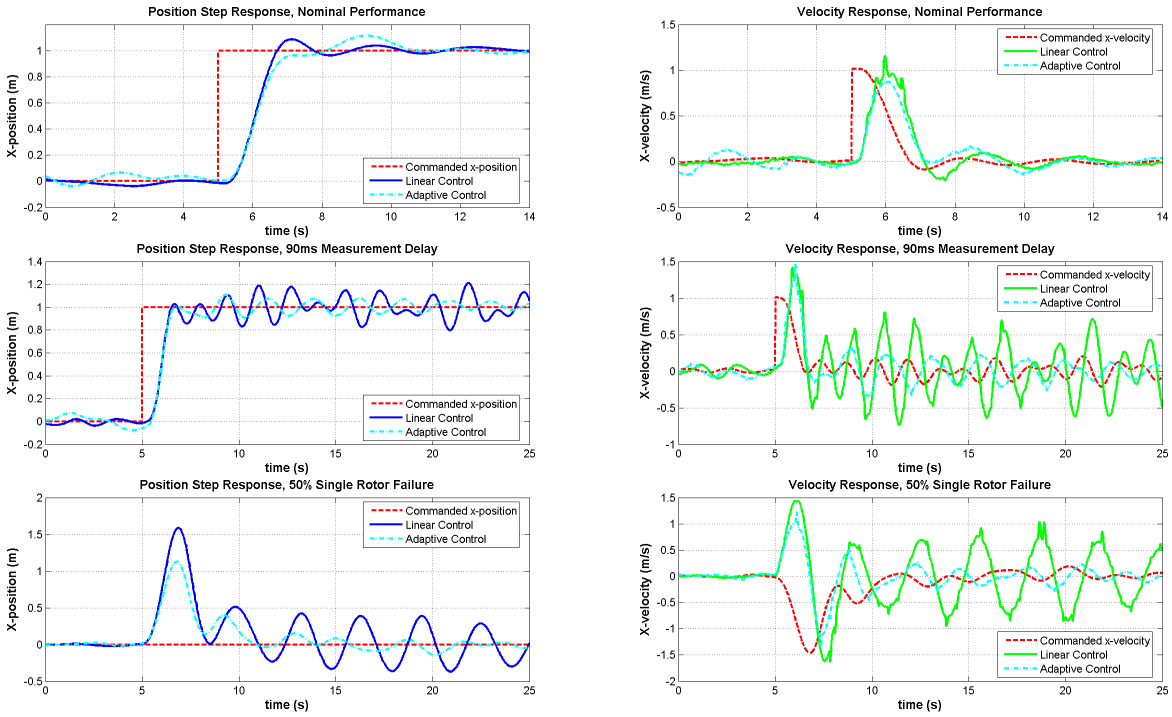

Figure 9: Flight test comparison of baseline linear controller to $\mathcal{L}_1$ adaptive controller, position response (left) and velocity response (right). While both show similar nominal performance (top), the $\mathcal{L}_1$ adaptive controller shows improved performance for both a 90ms measurement delay (middle) and a 50% single-rotor failure (bottom).

VI. Conclusions and Future Work

This paper attempts to provide a systematic design process for the use of $\mathcal{L}_1$ adaptive output feedback control in realistic flight control applications. The proposed method provides the control designer with an intuitive method linking relevant performance and robustness metrics to the selection of the $\mathcal{L}_1$ parameters $C(s)$ and $M(s)$. This design process represents a step in the direction of more readily applying $\mathcal{L}_1$ adaptive control to real-world flight systems and taking advantage of its potential benefits.

Flight test results verify the process for an indoor autonomous quadrotor, demonstrating that variations in the specified cost function produce the expected and desired physical responses. In flight tests comparing it with the baseline linear controller, the augmented $\mathcal{L}_1$ adaptive system shows definite performance and robustness improvements. Also, adaptive augmentation is shown to help enable aggressive flight for a fixed-wing aerobatic aircraft. Both of these results confirm the potential of $\mathcal{L}_1$ adaptive control as a useful tool for autonomous aircraft.

Several limitations of the design process have been identified, most stemming from the non-convexity of the cost function. This acts to limit the complexity of the assumed forms of $C(s)$ and $M(s)$, preventing the potential benefits of higher-order filters from being explored. Future work is currently focused on converting the performance and robustness metrics to a set of linear matrix inequality(LMI) constraints. Such a system is much more efficiently solved, thus having the potential to handle more complex solution forms. Some problems currently being faced are conservatism in conversion of the metrics to LMIs, and the inability of available numerical solvers to find initial feasible solutions.

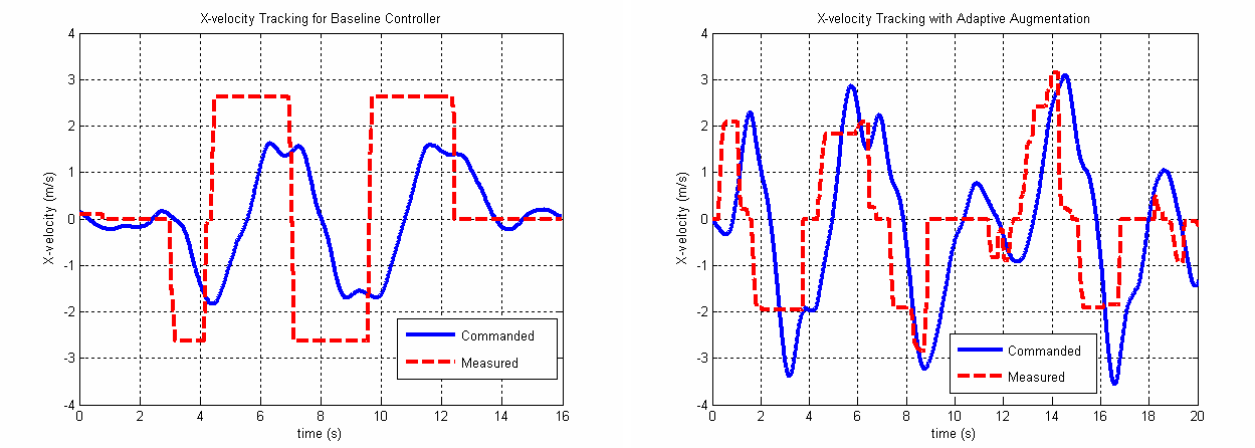

Figure 10: Flight results showing improved aggressive tracking performance of a fixed-wing aerobatic aircraft with $\mathcal{L}_1$ augmentation (right) over the baseline linear controller (left).

Acknowledgments

The authors would like to give special thanks to Naira Hovakimyan, Dapeng Li, and Eugene Lavretsky for their patient and expert advice regarding all matters adaptive. Thanks also to Frantisek Sobolic for his helpful feedback throughout the writing process. This research was funded in part under AFOSR Grant FA9550-08-1-0086.

References

[1] Anderson, B. D. and Dehghani, A., "Challenges of adaptive control: past, permanent and future," Vol. 32, No. 2, December 2008, pp. 123–135.

[2] Cao, C. and Hovakimyan, N., "Design and Analysis of a Novel L1 Adaptive Controller, Part I: Control Signal and Asymptotic Stability," *Proc. American Control Conference*, 14–16 June 2006, pp. 3397–3402.

[3] Cao, C. and Hovakimyan, N., "Design and Analysis of a Novel L1 Adaptive Controller, Part II: Guaranteed Transient Performance," *Proc. American Control Conference*, 14–16 June 2006, pp. 3403–3408.

[4] Cao, C. and Hovakimyan, N., "L1 Adaptive Output Feedback Controller for Systems with Time-varying Unknown Parameters and Bounded Disturbances," *Proc. American Control Conference ACC '07*, 9–13 July 2007, pp. 486–491.

[5] Li, D., Patel, V., Cao, C., and Hovakimyan, N., "Optimization of the Time-Delay Margin of L1 Adaptive Controller via the Design of the Underlying Filter," *Proc. AIAA Guidance, Navigation and Control*, August 2007.

[6] Hindman, R., Cao, C., and Hovakimyan, N., "Designing a High Performance, Stable L1 Adaptive Output Feedback Controller," *Proc. AIAA Guidance, Navigation and Control*, August 2007.

[7] Dobrokhodov, V., Xargay, E., Kaminer, I. I., Lizarraga, M., Cao, C., Hovakimyan, N., Gregory, I., and Kitsios, I., "Flight Validation of Metrics Driven L1 Adaptive Control," *Proc. AIAA Guidance, Navigation, and Control*, August 2007.

[8] Cao, C. and Hovakimyan, N., "L1 Adaptive Output Feedback Controller to Systems of Unknown Dimension," *Proc. American Control Conference ACC '07*, 9–13 July 2007, pp. 1191–1196.

[9] Li, D., Hovakimyan, N., Cao, C., and Wise, K., "Filter Design for Feedback-loop Trade-off of L1 Adaptive Controller: A Linear Matrix Inequality Approach," *Proc. AIAA Guidance, Navigation and Control*, August 2008.

[10] 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.

[11] Valenti, M., Bethke, B., Fiore, G., How, J., and Feron, E., "Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery," *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, August 2006, AIAA-2006-6200.

[12] Frank, A., McGrew, J. S., Valenti, M., Levine, D., and How, J. P., "Hover, Transition, and Level Flight Control Design for a Single-Propeller Indoor Airplane," *AIAA Guidance, Navigation and Control Conference and Exhibit*, No. AIAA 2007-6318, Hilton Head, South Carolina, 20 - 23 August 2007.

[13] Knoebel, N., *Adaptive Control of a Miniature Tailsitter UAV*, Master's thesis, Brigham Young University, December 2007.