Adaptive Parameter Estimation for Aerial Manipulation

Gabriel Baraban[1], Matthew Sheckells[2], Soowon Kim[1] and Marin Kobilarov[1]

***Abstract*— This paper proposes a model reference adaptive controller for aerial robots equipped with manipulators for grasping and transporting objects of unknown mass. Through a Lyapunov argument, this estimation and control algorithm is shown to be asymptotically stable when tracking a desired reference trajectory. The adaptive controller is then demonstrated on a quadrotor with a changing total mass, showing that it can track a reference trajectory with sufficient accuracy to pick up a payload and maintain flight stability, despite the sudden change in vehicle parameters induced by the payload. This new adaptive controller is slower to respond to changes in vehicle mass than control strategies that use more conventional mass estimation methods, but it is subject to less noise since it does not make use of accelerometer data, resulting in a smoother control trajectory.**

I. Introduction

Unmanned aerial vehicles (UAVs) such as quadrotors are now established platforms for aerial imaging and observations (e.g. agricultural crop monitoring, photography, and surveillance). Recent research has augmented these capabilities with onboard arms or grippers, creating aerial manipulation platforms [1], [2], [3], [4]. These systems can be used for package delivery [5], collaborative lifting [6], construction, and infrastructure maintenance [7], [8]. Methods for deriving the dynamics of such multi-body systems [9], [10], for their motion planning and trajectory generation [3], [11], [12], and control [2], [3], [13], [14] are now well established.

Applications involving the picking, lifting, or placing of a payload require control algorithms that must adapt to changes in the system mass. In most cases, mass is estimated as an isolated unknown parameter through acceleration measurements, for instance through batch least-squares regression or low-pass filtering [15], [16]. Alternatively, coupled estimation and control can be employed, specifically through an adaptive model reference controller, in order to ensure that as the mass estimate changes the system retains favorable stability properties. In this work, we follow this viewpoint and propose an estimation and control algorithm which can follow given reference trajectories while picking up a payload of unknown mass. The proposed controller is shown to be globally asymptotically stable.

A. Related Work

Adaptive methods have been applied to aerial manipulation in the past. For example, Cabecinhas *et al.* [17] used an adaptive technique to estimate a force bias on a quadcopter manipulator. Other works have used sliding mode control [9] and virtual decomposition [18] to handle these unknown dynamics.

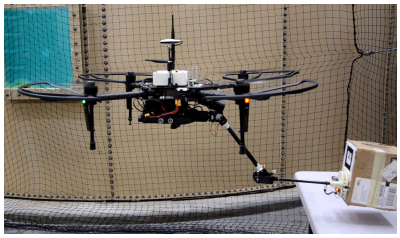

Fig. 1. The aerial manipulation system picking up a package, using the proposed adaptive model-reference controller to handle objects with unknown mass.

This work uses model reference control (MRC), a specific technique within the adaptive control toolkit. Dydek *et al.* [19] used MRC to develop a robust quadcopter control algorithm, but did not apply their work to aerial manipulation. Pierri *et al.* [20] used MRC to stabilize an aerial manipulator, but only tested the results in simulation. This work describes the development of an MRC algorithm and applies it to an experimental setting, demonstrating its ability to achieve stable flight and comparing its performance to a baseline filtered estimator.

II. Adaptive Controller Design

A. Problem Setup

Let the system state be defined as $x = (p, v)$ consisting of the position $p \in \mathbb{R}^3$ and velocity $v \in \mathbb{R}^3$ of the center of mass of the quadrotor in a fixed inertial frame. The difference between this point and the center of mass of the entire aerial manipulation system (quadrotor, arm, and payload) is handled by Kobilarov *et al.* in [21]. The dynamics of the system follow a simple force balance equation:

$$\dot{x} = Ax + Bu + \begin{bmatrix} 0 \\ g \end{bmatrix}, \tag{1}$$

where $A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ m^{-1}I \end{bmatrix}$, and g is the acceleration due to gravity.

The objective is to track a given smooth reference trajectory $x_d(t)$, i.e. to design a controller to reduce the state error $\Delta x(t) = x(t) - x_d(t)$. Ideally, the stabilizing thrust force input is

$$u_d = m(a_d - g) - K\Delta x \tag{2}$$

where m is the mass of the system, $a_d = \ddot{p}_d$ is the desired acceleration (derived from the desired trajectory), and $K = \begin{bmatrix} k_p I & k_d I \end{bmatrix}$ is the control gain matrix chosen to render the

[1]Gabriel Baraban, Soowon Kim, and Marin Kobilarov are with the Department of Mechanical Engineering, Johns Hopkins University, 3400 N Charles Str, Baltimore, MD 21218, USA gbaraban|skim386|marin@jhu.edu

[2]Matthew Sheckells is with the Department of Computer Science, Johns Hopkins University, 3400 N Charles Str, Baltimore, MD 21218, USA msheckells@jhu.edu

linear error dynamics $\dot{\Delta x} = (A - BK)\Delta x$ exponentially stable.

Unfortunately, a number of impediments prevent this solution from being implemented. First, the ideal thrust input above is a three dimensional vector, but in reality, a quadrotor can only apply thrust along the body-fixed z-axis. In order to thrust in the proper direction, the robot needs to align the thrust axis with the desired acceleration a_d by orienting itself into the desired yaw, pitch, and roll Euler angles $(\alpha_d, \beta_d, \gamma_d)$. In practice, α_d is provided by the reference trajectory and the true yaw α is stabilized to it independently. At every time step, given the current yaw α, β_d and γ_d are calculated using the following formulas, where e_i is the unit vector in the i-th dimension:

$$\gamma_d = \arcsin(a_d^T e_1 \sin\alpha - \bar{a}_d^T e_2 \cos\alpha), \tag{3}$$

$$\beta_d = \arctan\left(\frac{\cos\gamma_d (a_d^T e_1 \cos\alpha + a_d^T e_2 \sin\alpha)}{\cos\gamma_d \;\; a_d^T e_3}\right). \tag{4}$$

Of course, β and γ cannot be set instantaneously. Instead, they are controller using a high-frequency control loop which compues the required torques produced by the rotors to achieve these angles [21]. In practice, the control interface provided by many off-the-shelf quadrotor does not support torque inputs. Instead, the flight controller accepts β_d and γ_d as inputs and a low level control loop stabilizes them quickly. For yaw control, the quadrotor accepts an angular rate input, $\dot{\alpha}$, which is chosen using a simple proportional controller:

$$\dot{\alpha} = -k_\alpha(\alpha - \alpha_d) \tag{5}$$

This stabilizes yaw exponentially for a range of positive gains k_α. Thus, at each iteration of the controller, the control input takes the form $\{\dot{\alpha}, \beta_d, \gamma_d, T\}$, where T is the desired total thrust magnitude.

The second problem is the relationship between the thrust input and the resulting force produced by the rotors. Rather than controlling force directly, the autopilot accepts a normalized thrust command between 0 and 100. This value corresponds to a percentage of the battery voltage, which is then sent to the speed controller. A variety of factors (e.g. propeller design, battery charge, floor effects) influence the resulting thrust produced by the propellers, but the relationship between the command and the thrust is modelled as linear:

$$T = k_t u$$

The gain parameter k_t is unknown and dependent on a number of factors, most notably the current voltage of the battery, but for short term applications, it is assumed to be constant. With this additional factor, the matrix B is changed to be $\begin{bmatrix} 0 \\ \frac{k_t}{m} I \end{bmatrix}$. As such, equation 2 must be updated:

$$u_d = \frac{m}{k_t}(a_d - g) - K\Delta x \tag{6}$$

The value of $\frac{k_t}{m}$ is unknown, and the method developed here attempts to estimate it through an adaptive controller.

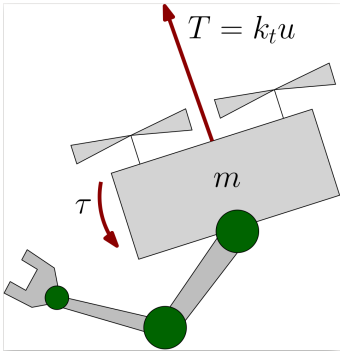

Fig. 2. A Free-Body Diagram of the control inputs of the UAV

B. Adaptive Controller

Let $\hat{\theta}$ denote the current estimate of the unknown ratio $\theta^* = \frac{m}{k_t}$. The error in the estimate is defined as $\Delta\theta = \hat{\theta} - \theta^*$. If this estimate is used to compute u as the current best estimate of u_d (defined above in equation 6), the new error dynamics are as follows:

$$\begin{aligned}\dot{\Delta x} &= \dot{x} - \dot{x}_d \\ &= A(x - x_d) + B(u - u_d) \\ &= A\Delta x + B(\theta * -\theta)(a_d - g) - BK\Delta x \\ &= (A - BK)\Delta x + \frac{\Delta\theta}{\theta^*}\begin{bmatrix} 0 \\ (a_d - g) \end{bmatrix}.\end{aligned} \tag{7}$$

To find a stable update law for $\hat{\theta}$ and thereby prove the stability of the controller, we turn to a Lyapunov function

$$V(\Delta x, \Delta\theta) = \frac{1}{2}\Delta x^T P \Delta x + \frac{1}{2k_m}\Delta\theta^2, \tag{8}$$

where $k_m > 0$ and

$$P = \begin{bmatrix} k_p & \epsilon\theta^* \\ \epsilon\theta^* & \theta^* \end{bmatrix}.$$

The positive constant, ϵ, is chosen to ensure that P is positive definite by limiting it to the interval $0 < \epsilon < \sqrt{\frac{k_p}{\theta^*}}$. In turn, the Lyapunov derivative is :

$$\begin{aligned}\dot{V}(\Delta x, \Delta\theta) = -\Delta x^T Q \Delta x + \Delta x^T P \frac{\Delta\theta}{\theta^*}\begin{bmatrix} 0 \\ a_d - g \end{bmatrix} \\ + \frac{1}{k_m}\Delta\theta\dot{\Delta\theta}.\end{aligned}$$

The matrix $Q = \begin{bmatrix} \epsilon k_p & \frac{\epsilon k_d}{2} \\ \frac{\epsilon k_d}{2} & k_d - \epsilon\theta^* \end{bmatrix}$ is ensured to be positive definite by choosing ϵ to fulfill the inequalities $(\frac{\epsilon k_d}{2})^2 < \epsilon k_p(k_d - \epsilon\theta^*)$ and $\epsilon k_p + k_d - \epsilon\theta^* > 0$. Experimentally, these inequalities are verified in section V

The second term of $\dot{V}$ is of indeterminate sign, and must be rendered negative semidefinite in order to complete the stability proof. Because $\Delta\theta = \hat{\theta} - \theta^*$ and θ^* is a constant, $\dot{\Delta\theta} = \dot{\hat{\theta}}$ and thus the error evolution can be designed through the choice of $\dot{\hat{\theta}}$. By choosing

$$\dot{\hat{\theta}} = -\frac{k_m}{\theta^*}\Delta x^T P \begin{bmatrix} 0 \\ a_d - g \end{bmatrix} \tag{9}$$

we obtain $\dot{V} = -\Delta x^T Q \Delta x$. Expanding this choice yields:

$$\begin{aligned}\dot{\hat{\theta}} = \dot{\Delta\theta} &= -\frac{k_m}{\theta^*}\Delta x^T P \begin{bmatrix} 0 \\ a_d - g \end{bmatrix} \\ &= -\frac{k_m}{\theta^*}\Delta x^T \begin{bmatrix} k_p & \epsilon\theta^* \\ \epsilon\theta^* & \theta^* \end{bmatrix} \begin{bmatrix} 0 \\ a_d - g \end{bmatrix} \\ &= -k_m(\epsilon\Delta p + \Delta v)^T(a_d - g) \end{aligned} \tag{10}$$

Note that the unknown parameter θ^* is no longer present in the control law, leaving a fully implementable solution.

Theorem 1. ***Corollary to Barbalat's Lemma: [22]*** *If $g : \mathbb{R}^+ \to \mathbb{R}$ and $g \in \mathbb{L}_2 \cap \mathbb{L}_\infty$, that is, g has bounded $\mathbb{L}_2$ and $\mathbb{L}_\infty$ norms, and if $\dot{g}$ is bounded, then $\lim_{t\to\infty} g(t) = 0$*

Proposition 1. *The control law:*

$$u = \hat{\theta}(a_d - g) - K\Delta x, \tag{11}$$

$$\dot{\hat{\theta}} = -k_m(\epsilon\Delta p + \Delta v)^T(a_d - g) \tag{12}$$

achieves stable tracking of a twice-differentiable reference trajectory $x_d(t)$.

Proof: Applying the controls defined in equations (11) and (12) results in a negative semi-definite Lyapunov derivative:

$$\dot{V}(\Delta x, \Delta\theta) = -\Delta x^T Q \Delta x. \tag{13}$$

Because $\dot{V}$ is negative semi-definite, V is inherently bounded by its initial value. Due to the definition of V, this boundedness implies that Δx and $\Delta\theta$ are bounded as well. Thus, $\Delta x \in \mathbb{L}_2 \cap \mathbb{L}_\infty$. In addition, based on equation 7, the boundedness of Δx and $\Delta\theta$ implies that $\dot{\Delta x}$ is bounded as well. Using the corollary to Barbalat's Lemma, we can therefore conclude that $\lim_{t\to\infty} \Delta x(t) = 0$. Thus, the state $x(t)$ converges to $x_d(t)$. □

Proposition 2. *Under the above conditions, the parameter estimate $\hat{\theta}$ approaches equilibrium, though not necessarily at θ^*.*

Proof: Equation (12) defines $\dot{\hat{\theta}}$. Based on the above results, Δx converges to 0, and therefore $\dot{\hat{\theta}}$ converges to zero. However, under a freefall trajectory, where $a_d(t) = g$, $\dot{\hat{\theta}} = 0$ for the entire trajectory. In that case, $\hat{\theta}(t)$ simply remains $\hat{\theta}(0)$, and does not converge to θ^*. It is important to note, that, despite this result, the tracking error will still converge to zero. □

Proposition 3. *Under the above conditions, if the reference trajectory (in particular the desired acceleration a_d) satisfies the inequality $||a_d(t) - g|| > \epsilon$ for some $\epsilon > 0$ and for all t, then the parameter error $\Delta\theta$ will converge to zero.*

Proof: LaSalle's theorem [22] states that for a system with a negative semi-definite Lyapunov function, the limit of the state as time goes to infinity will be in the largest invariant set $\mathbb{T}$ contained within the set $\mathbb{S}$, where $\dot{V} = 0$. Based on equation (13), $\mathbb{S}$ is clearly the set $\mathbb{S} = \{(\Delta x, \Delta\theta) | \Delta x = 0\}$. Using equation (7), the only condition under which Δx will remain 0 is when the second term is also zero, i.e. when $\frac{\Delta\theta}{\theta^*}\begin{bmatrix} 0 \\ (a_d - g) \end{bmatrix} = 0$. Multiplying by the constant θ^* and removing the first three (constantly zero) rows of the vector, the new definition of $\mathbb{T} = \{(\Delta x, \Delta\theta) | \Delta\theta(a_d - g) = 0\}$. Only when this condition is met will a member of $\mathbb{T}$ remain within the set for all future time. Taking the magnitude of each side, $||\Delta\theta(a_d - g)|| = |\Delta\theta|||(a_d - g)|| > |\Delta\theta|\epsilon$. Because $\epsilon > 0$, the only invariant set within $\mathbb{S}$ is $\mathbb{T} = \{(\Delta\theta, \Delta x) = (0, 0)\}$. By LaSalle's theorem, the system will converge to this point, where all errors are zero.

This additional condition on acceleration is not particularly strict for quadrotor systems. In practice, there are few desired trajectories which engage in even brief free-fall. Bounding the acceleration above gravity is often a prudent safety check, and now it is shown to also guarantee the convergence of this adaptive controller.

C. Baseline Estimator

In previous aerial manipulation systems, θ is estimated using the data from an onboard accelerometer and a low-pass filter. In particular, an estimator of this kind uses a simple yet effective formula to estimate θ at every timestep:

$$\theta_{calc} = \frac{||T||_2}{||a||_2} \tag{14}$$

This estimate is sensitive to sensor noise, and so, rather than using θ_{calc} directly, the system uses a filtered value of θ through the formula:

$$\theta_i = \alpha\theta_{calc} + (1 - \alpha)\theta_{i-1} \tag{15}$$

This low-pass filter rejects most of the sensor noise in the accelerometer while still being responsive to changes in θ^*. In this work, the value of α used was chosen to be 0.16. If the adaptive controller were not being used, this value could be plugged into the control law

$$u(t_i) = \theta_i(a_d(t_i) - g) - K\Delta x(t_i) \tag{16}$$

The success of this approach is shown in [16].

Experiments shown in section V run an estimator of this type in parallel to the adaptive estimator in order to provide a baseline for comparison. Due to the dynamic nature of the system, it is difficult to measure a true value of θ^*. Instead, it is easier to compare the measurements of θ from two different estimators, and show that the new adaptive controller meets or exceeds the performance of the baseline, well-established estimator.

III. Hardware

A. Quadrotor Platform

The aerial manipulation system is based on a DJI Matrice quadrotor. The quadrotor is equipped with a PointGrey Flea3 camera for performing visual tracking and an Intel NUCi5 computer, which communicates with the Matrice flight controller over a UART connection. A DJI Guidance autopilot system is built into the system, for use in the event of system breakdown.

B. Motion capture system

High precision position measurement is provided by an Optitrack motion capture system. The motion capture limits the experiment workspace to the coverage area of the cameras, but for the purposes of quantifying the success of the algorithm, this is not limiting. To obtain velocity measurements from the motion capture sensor, the differences between sucessive position measurements are passed through an exponential low-pass filter, smoothing out the noise introduced by this discrete approximation of the derivative.

C. Manipulator

1) Custom 2-DoF Arm: Unlike previous works that utilize a fixed position gripper [1], [23], the aerial manipulator used in this analysis uses a custom 2-DoF arm, which can extend outside the rotor footprint of the quadcopter, allowing for more complex manipulation behaviors. The arm is constructed from carbon fiber tubes and actuated by Dynamixel servo motors. The position of the end-effector is controlled using an onboard Teensy microcontroller, and is assumed to be constant during flight maneuvers.

2) Magnetic Gripper: The end of the arm is a custom designed magnetic gripper, shown in Figure 3. Permanent magnets embedded in the gripper attract magnets in a mating plate attached to the payload. This attraction force allows the gripper to tolerate 3cm error in the plane parallel to the contact and 2cm in the normal direction. To ungrip a payload, a servo motor rotates the magnets such that the polarities are reversed, repulsing the mating plate. The gripping status is measured by a momentary switch, which is depressed when the mating plate is attached.

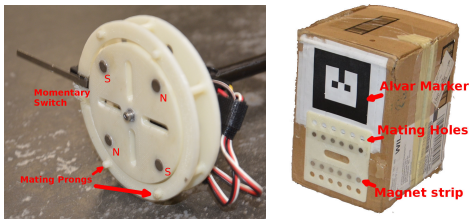

Fig. 3. The magnetic gripper (left) and a sample package (right) used in our aerial manipulation experiments. The package is instrumented with an AR marker to facilitate tracking and a magnetic mating joint so it can attach to the gripper.

IV. Testing Methods

A. Payload Picking Scenario

Using the hardware described above, the controller attempted a well-established pick and place scenario. At the beginning of the test, a cardboard box payload is set up in the testing area. The box is tagged with an AR marker [24] and equipped with a mating plate. At first, the box is empty, with a mass of 146g. Afterwards, weights were attached to the box, resulting in a total mass of 173g. Ideally, these experiments would have included a wider range of payload masses, but all further mass additions exceeded the payload capacity of the system.

The software for this controller was implemented in the Johns Hopkins University Aerial Autonomy framework [16]. When the test begins, the parameter initializes at $\hat{\theta} = 5.8$. This value was chosen because preliminary experiments found that the true values typically range from 5.5 to 6.0. Because of the initial error in $\hat{\theta}$, the trajectory tracking of the quadrotor is initially poor. To avoid collisions with the testing environment, the first phase of the test is to fly to a pre-determined staging location relative to the AR tag. Should the system lose visual tracking of the AR tag, the state machine resets, sending the quadrotor back to the home position, then beginning again with the most recent value of $\hat{\theta}$. Through these trial-reset episodes, $\hat{\theta}$ improves, allowing better position tracking. If other failures occur during the test (tag occlusions, sensor noise, etc.), this same resetting mechanism is used to recover.

When this staging finishes, the adaptive controller parameter estimate is presumed to have stabilized to an equilibrium value. From there, the robot tracks a reference trajectory toward the mating plate, extending the arm and gripping the payload. Once this connection is complete, the quadrotor retreats to a safe distance with the box still attached. The added mass of the box upsets the equilibrium in the parameter estimate, forcing it to adjust.

B. Gain Tuning

In early testing, various k_m and ϵ choices were tested. Tuning these gains was a similar process to tuning a PD controller, as the update law described in equation (10) can be expanded into a PD-like form. By adjusting k_m and the product $k_m\epsilon$ to adjust rise time and settling behavior, values of $k_m = 0.1$ and $\epsilon = 0.23$ were found to produce good results.

C. Data Analysis

The sensor measurements, control inputs, and the parameter estimate were logged during each test, along with the results of the baseline estimator. The result from the controller can be compared to a typical solution to this problem, as described in equation (15).

After collecting this data, the parameter estimates with and without the box, $\hat{\theta}_2$ and $\hat{\theta}_1$ respectively, were obtained by averaging the data of equilibrium flight. Two metrics are then used to evaluate the controller performance. The first, the rise time of the system, is defined as the time between picking up the box and when $\hat{\theta}$ reaches 90% of $\hat{\theta}_2$. The second, the mass fraction, is obtained by assuming that the mass-independent factors of θ are constant. By dividing the two estimates, we derive the formula $\frac{\hat{\theta}_2}{\hat{\theta}_1} = \frac{M+m}{M}$, where M is the mass of the aerial manipulator (3.624kg) and m is the mass of the box in that particular test. By comparing this fraction to its true value, obtained with direct measurement, the accuracy of the estimation can be evaluated.

V. Results

Once the final choices for the parameter were made, a series of four tests were run, picking up the unloaded box. On average, the rise time was 10.04 ± 2.9 seconds and the mass

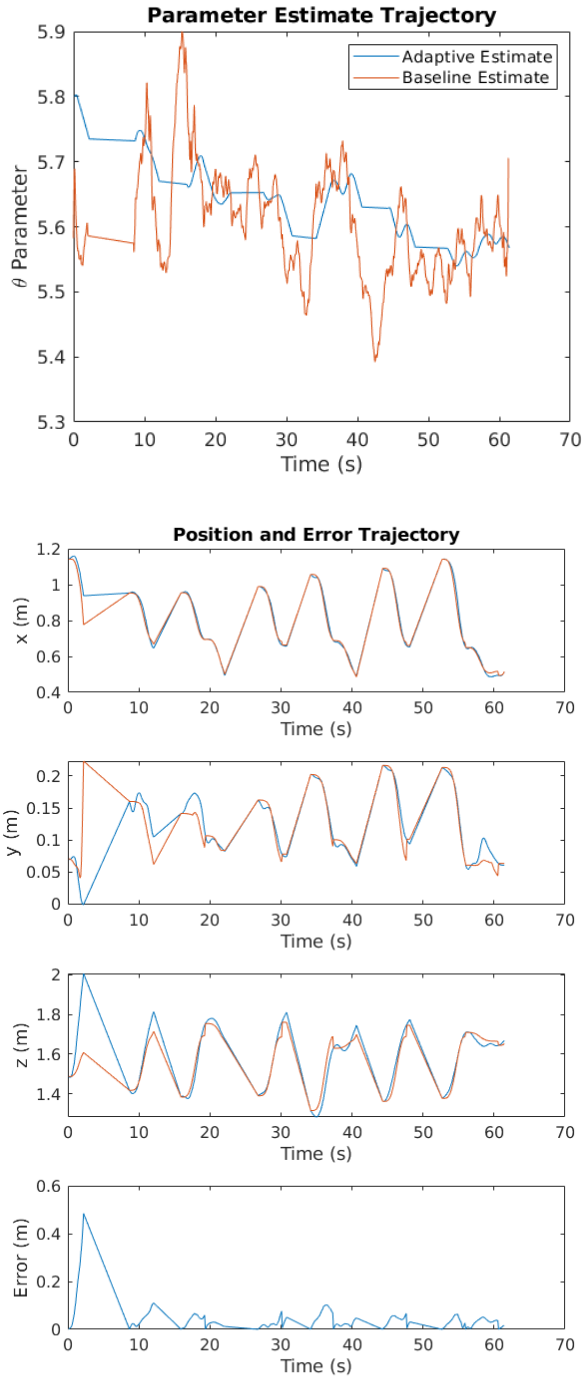

Fig. 4. An example trajectory of the quadrotor, from test initiation until the initial equilibrium is reached. In the θ plot, note the decrease in $\hat{\theta}$ from the initial guess of 5.8 to convergence with the true value (approximately 5.6). The flat portions of the trajectory demonstrate the failsafe behavior, in which the autopilot returns the quadrotor to the home position. The most recent value of $\hat{\theta}$ is retained and used in the next attempt. In the position plots, the reference trajectory is plotted in red and the quadrotor trajectory is plotted in blue. The final plot is the total RMS position error.

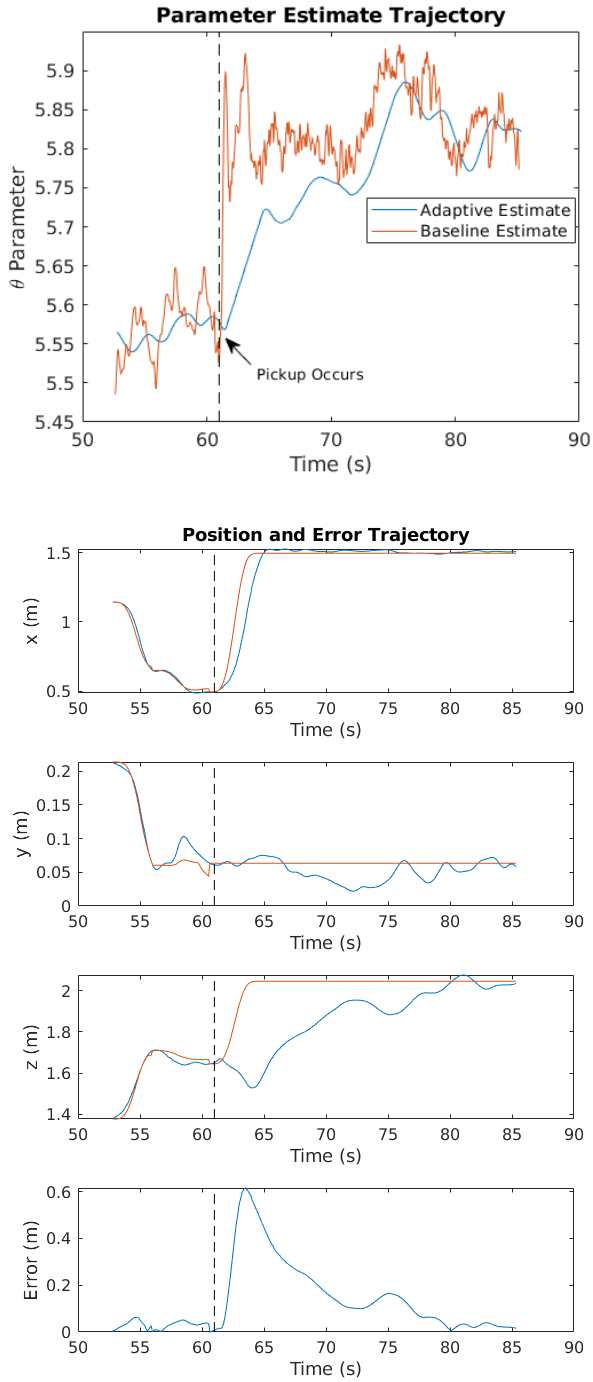

Fig. 5. An example trajectory of the quadrotor, from the initial equilibrium through the picking maneuver. The black dashed line illustrates when the mass of the system changes due to the pickup. In the position plots, the reference trajectory is plotted in red and the quadrotor trajectory is plotted in blue. The final plot is the total RMS position error.

fraction was 1.0485 ± 0.0085, compared to the true value 1.0403. Afterwards, four tests were run with a weighted box. For these weighted tests, the rise time was 10.3 ± 3.8 seconds, while the mass fraction was 1.0607 ± 0.0101, compared to the true value 1.0477. The rise time results are tabulated along with comparisons to the estimator in table I. While the rise times were slower than the near instant response from the baseline estimator, the results from the adaptive controller were less susceptible to sensor noise, even after filtering. The mass fraction results, shown in table **??** were consistently higher than the true results, which can be attributed to the overall upward trend in θ due to battery voltage changes. Because the weighted box tests took longer to converge, this error was increased.

Figure 4 shows an example trajectory of the system reaching initial equilibrium (before picking) and Figure 5 shows

an example trajectory demonstrating the system reaction to picking up a box. These plots demonstrate that the adaptive controller indeed achieves convergence to the proper value of θ. Compared to the baseline estimator, the adaptive controller has less noise, but has a slower convergence time. Possible improvements are discussed in the following section.

TABLE I

MEAN RESULTS ACROSS FOUR TRIALS.

Metric	Payload	MRAC	Std. Dev.	Baseline	Std. Dev.
Rise Time	Empty	10.04s	$\pm$2.9s	4.44s	$\pm$1.2s
Rise Time	Weighted	10.30s	$\pm$3.8s	5.96s	$\pm$3.37s
Mass Frac.	Empty	1.0485	$\pm$0.0085	1.0493	$\pm$0.0079
Mass Frac.	Weighted	1.0607	$\pm$0.0101	1.067	$\pm$0.0109

Finally, the three inequalities derived in section II-B can be verified with the measured values of θ^*. The experimental values used are shown in table V.

k_m	ϵ	k_p	k_d
0.1	0.23	4	8

θ^* varied between 5.5 and 5.8. The inequalities are therefore tested over that range. They are as follows:

1) $0 < \epsilon < \sqrt{\frac{k_p}{\theta^*}} \implies 0 < 0.23 < (0.83, 0.85)$
2) $(\frac{\epsilon k_d}{2})^2 < \epsilon k_p(k_d - \epsilon\theta^*) \implies 0.85 < (6.13, 6.20)$
3) $\epsilon k_p + k_d - \epsilon\theta^* > 0 \implies (7.59, 7.66) > 0$

Thus, we have verified that all of the conditions on the proofs above hold true for the experiment at hand.

VI. CONCLUSION

This work developed an adaptive model reference controller for an aerial manipulation system. By estimating the unknown dynamics parameter of the system, the robot can be controlled with sufficient accuracy to navigate and reach an object of interest. Then, using the same algorithm, the controller can adapt to the added mass after picking up the object and quickly stabilize again. In future work, the controller parameters could be tuned further, allowing the estimate to converge faster and with less oscillation. In addition, a future controller can circumvent the quadcopter autopilot discussed in section II-A in order to the input torques to the system. If torque-level commands were available to the controller, a similar analysis could be performed to measure the attitudinal dynamics parameters of the system, in effect measuring moments of inertia in the same way that this controller measures mass. This will provide richer information about unknown payload properties to extend the applications of aerial manipulators into more complex, unfamiliar environments.

VII. ACKNOWLEDGEMENTS

Thanks to Professor Joseph Katz and Professor Louis Whitcomb for providing facilities for conducting experiments and to Subhransu Mishra for guidance in designing the manipulator hardware. This material is based upon work supported by the National Science Foundation under grant no:1527432.

REFERENCES

[1] C. D. Bellicoso, L. R. Buonocore, V. Lippiello, and B. Siciliano, "Design, modeling and control of a 5-dof light-weight robot arm for aerial manipulation," in *Control and Automation (MED), 2015 23th Mediterranean Conference on*, pp. 853–858, IEEE, 2015.

[2] A. Suarez, G. Heredia, and A. Ollero, "Lightweight compliant arm with compliant finger for aerial manipulation and inspection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4449–4454, Oct 2016.

[3] K. Kondak, F. Huber, M. Schwarzbach, M. Laiacker, D. Sommer, M. Bejar, and A. Ollero, "Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2107–2112, May 2014.

[4] I. Palunko, P. Cruz, and R. Fierro, "Agile load transportation: Safe and efficient load manipulation with aerial robots," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 69–79, 2012.

[5] Amazon, "Prime air." `https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011`, 2017.

[6] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[7] AeroWorks. `http://www.aeroworks2020.eu/`, 2017.

[8] AEROARMS. `https://aeroarms-project.eu/`, 2017.

[9] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4990–4995, IEEE, 2013.

[10] M. Kobilarov, "Nonlinear trajectory control of multi-body aerial manipulators," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 679–692, 2014.

[11] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, *Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera*, pp. 235–252. Cham: Springer International Publishing, 2017.

[12] G. Garimella and M. Kobilarov, "Towards model-predictive control for aerial pick-and-place." submitted to International Conference on Robotics and Automation (ICRA), 2015.

[13] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2668–2673, IEEE, 2011.

[14] V. Lippiello, J. Cacace, A. Santamaria-Navarro, J. Andrade-Cetto, M. A. Trujillo, Y. R. Esteves, and A. Viguria, "Hybrid visual servoing with hierarchical task composition for aerial manipulation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 259–266, 2016.

[15] A. Suarez, G. Heredia, and A. Ollero, "Lightweight compliant arm for aerial manipulation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1627–1632, IEEE, 2015.

[16] G. Garimella, M. Sheckells, S. Kim, G. Baraban, and M. Kobilarov, "A framework for reliable aerial manipulation." Manuscript submitted for publication, 2019.

[17] D. Cabecinhas, R. Cunha, and C. Silvestre, "A nonlinear quadrotor trajectory tracking controller with disturbance rejection," *Control Engineering Practice*, vol. 26, pp. 1–10, 2014.

[18] M. Jafarinasab and S. Sirouspour, "Adaptive motion control of aerial robotic manipulators based on virtual decomposition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1858–1863, IEEE, 2015.

[19] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor uavs: A design trade study with flight evaluations," *IEEE Transactions on control systems technology*, vol. 21, no. 4, pp. 1400–1406, 2013.

[20] F. Pierri, G. Muscio, and F. Caccavale, "An adaptive hierarchical control for aerial manipulators," *Robotica*, vol. 36, no. 10, pp. 1527–1550, 2018.

[21] M. Kobilarov, "Nonlinear trajectory control of multi-body aerial manipulators," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 679–692, 2014.

[22] K. S. Narendra and A. M. Annaswamy, *Stable adaptive systems*. Dover, 2005.

[23] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "The yale aerial manipulator: grasping in flight," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2974–2975, IEEE, 2011.

[24] Alvar. `http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/index.html`, 2017.