(Preprint) AAS 22-41

A CAMERA AND RANGE SENSOR FUSION APPROACH FOR AUTONOMOUS NAVIGATION SYSTEMS DRIVEN BY ROBUST ADAPTIVE CONTROL

Caleb Peck[*], Julius Marshall[†], Christoper Miller[‡], Kookjin Sung[§], Manoranjan Majji[¶], Peter Arslanian[‖]

An integrated sensing approach that fuses vision and range information to land an autonomous class 1 unmanned aerial system (UAS) controlled by e-modification model reference adaptive control is presented. The navigation system uses a feature detection algorithm to locate features and compute the corresponding range vectors on a coarsely instrumented landing platform. The relative translation and rotation state is estimated and sent to the flight computer for control feedback. A robust adaptive control law that guarantees uniform ultimate boundedness of the adaptive gains in the presence of bounded external disturbances is used to control the flight vehicle. Experimental flight tests are conducted to validate the integration of these systems and measure the quality of result from the navigation solution. Robustness of the control law amidst flight disturbances and hardware failures is demonstrated. The research results demonstrate the utility of low-cost, low-weight navigation solutions for navigation of small, autonomous UAS to carryout littoral proximity operations about unprepared shipdecks.

INTRODUCTION

As opportunity for autonomous flight on Earth and in space widens, there is an increasing presence of small, unmanned vehicles among consumers and the industry as a whole. These vehicles are particularly of interest for concepts such as swarm flight, reconnaissance, and supply delivery due to their lower cost and a smaller overall footprint. As the frame of these vehicles shrink, so too must the on-board sensing and navigation hardware. This paper introduces a prototype sensor suite (see Figure 1(a)) which fuses light detection and ranging (LIDAR) and optical camera data to compute a relative navigation solution. Since this is a relative pose solution, absolute positioning is not required. Thus, it is suitable for close proximity, GPS-denied environments. This sensor is integrated onto a class 1, autonomous, unmanned aerial system (UAS) (see Figure 1(b)), which uses an adaptive control law capable of rejecting flight disturbances.

LIDARs, such as flash LIDAR and scanning LIDAR, have become a key sensing component for autonomous vehicles.[1,2] One major application is terrain relative navigation (TRN), in which the

[*]Graduate Research Assistant, Land, Air and Space Robotics (LASR) Laboratory, Aerospace Engineering
[†]Graduate Research Assistant, Industrial and Systems Engineering, Advanced Control Systems Laboratory, Virginia Tech
[‡]Graduate Research Assistant, Mechanical Engineering, Advanced Control Systems Laboratory, Virginia Tech
[§]Graduate Research Assistant, Land, Air and Space Robotics (LASR) Laboratory, Aerospace Engineering
[¶]Associate Professor, Director, Land, Air and Space Robotics (LASR) Laboratory, Aerospace Engineering
[‖]Director, Maritime Autonomous Guidance-Navigation-Control Intelligent Coupled Systems Lab (MAGIC) NAVAIR/NAWC-AD

imaged terrain is compared to a known map of the area such that a vehicles position, velocity and attitude can be obtained.[3] This capability is critical in the Entry, Descent, and Landing (EDL) phase of space missions such as the Mars 2020 Mars Perseverance Rover.[4] The DragonFlyer mission on Titan, which will take advantage of the combined low gravity and high atmospheric density, is slated to use a LIDAR for TRN since the craft will be able to hover for longer periods of time compared to a Martian or Lunar descent.[5] In a similar concept, the navigation solution presented herein uses aligned LIDAR and camera data to compute the relative position and attitude from the vehicle to a known location. In this experiment, the known location is a user-defined feature frame on a simulated ship landing deck. However, the concept can be cast to any cooperative active proximity operation, including TRN and Rendezvous, Proximity Operations, and Docking (RPOD).

Ship-board landing poses a unique environment for manned and un-manned aircraft. Airflow across the deck produces a large region of turbulent airflow behind the stern,[6,7] which renders aircraft approach to and landing on the ship a region of increased pilot workload.[8] This turbulent flow offers a rich environment of study for many different disciplines, from modeling the flow through CFD to disturbance-rejecting control of vehicles.[9,10] In addition to the chaotic flow, the ground effect phenomenon is apparent with every landing. The surface diminishes the strength of the drag-inducing downwash produced by the wingtip vortices, which effectively creates a "cushion" of lift at the instant of landing.[11] These dynamic lift characteristics near the ground offer yet another 'disturbance' for which a robust control method must be able to reject.

(a) ANS Hardware

(b) Quadcopter, Top View

Figure 1. ANS Sensor System and Quadrotor Vehicle

Model Reference Adaptive Control (MRAC) is an adaptive control method that changes the dynamics of the plant in real time to fit some desired reference model based on the state's output.[12,13] It is shown that adaptive control is robust to flight disturbances and mathematically guarantees stability against bounded disturbances.[14–16] The non-linear, coupled dynamics of a quad-rotor drone offer a rigorous platform to experiment adaptive control methods.[17,18] This paper covers the implementation of model reference adaptive control with e-modification [12, pp. 327-329] on a small

UAS subjected to flight disturbances including motor failure, mass uncertainties, and wind gusts.

The scope of this work focuses on the performance of a camera-based navigation solution and a controller that can improve the incoming images by maintaining a steady hover amidst the disturbances. The next section will introduce the Autoland Navigation Solution (ANS), followed by the mathematical theory behind e-modification model reference adaptive control (e-MRAC). Then, the flight experiment and the results will be presented.

NAVIGATION SOLUTION

The navigation solution employs an off-the-shelf sensor suite which houses a RGB camera, LIDAR and Inertial Measurement Unit (IMU). The algorithm locates features on a user-defined frame of interest and computes the relative translation and rotation of the feature vector from the feature frame to the camera frame. This is achieved by detecting features using the camera, and matching the feature location with a range measurement from the LIDAR. The experiments presented in this paper use the ArUco marker library[19,20] since there are several readily available and reliable detection algorithms for these markers. It must be said that this LIDAR/vision-based system may be applied to any structured feature or cooperative relative navigation problem where the location of features are known and can be detected.

The relative translation and rotation is computed and can serve as the tracking error, which the controller would regulate to zero during landing. This section covers the pinhole camera model, which relates the feature pixel location to a world location based on the intrinsic parameters of the camera. Then, the ANS algorithm is presented.

Pinhole Camera Model

The pinhole camera model,[21] illustrated in Figure 2, is an idealized model of a camera which assumes all light enters through a single point and then re-projected onto an image plane. This model can be used to calibrate a camera and estimate the focal lengths (f_x, f_y) and principal point (c_x, c_y), also known as the camera's *intrinsic* parameters.

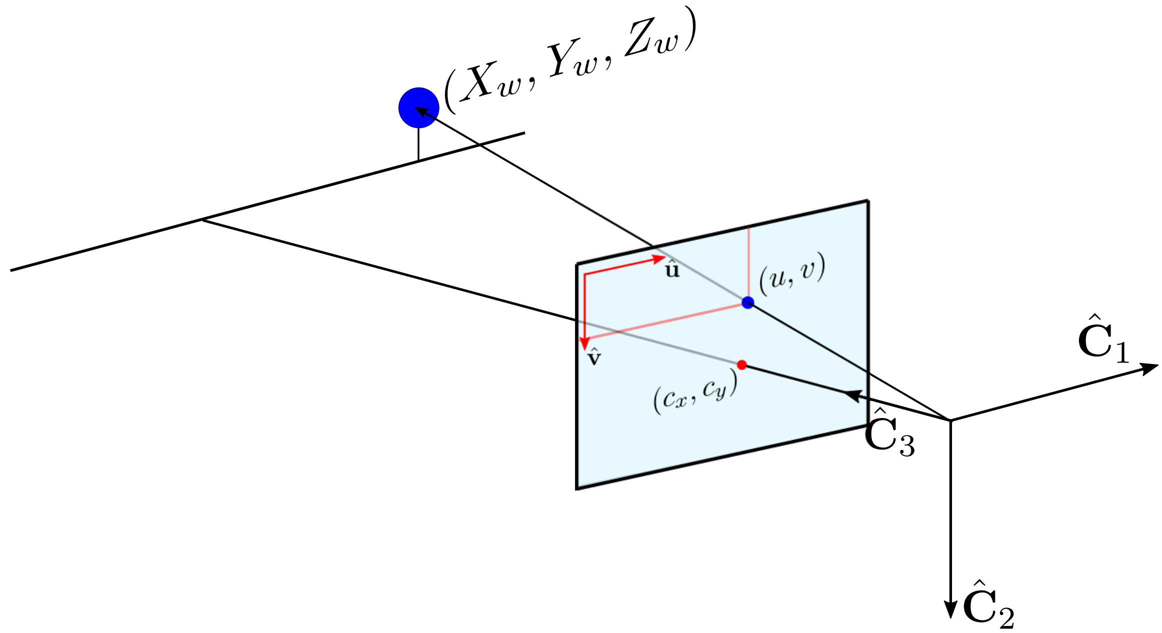

Figure 2. Pin-Hole Camera Model

The *extrinsic* parameters of a camera map a point in the world frame, $\mathcal{W}$, to the camera's frame, $\mathcal{C}$, through a rigid body rotation, R, and translation $\mathbf{t}$. Once the world point is cast to the camera frame, the intrinsic matrix transforms the camera frame vector, up to some length scale s, to a pixel location in the canonical image plane. This transformation is carried out as follows

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R \mid \mathbf{t} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{1}$$

The vector $[X_w, Y_w, Z_w]^T$ is defined as $\mathbf{r}_c$ in the following section. The application of ANS assumes the intrinsic parameters of the camera are known through calibration. Thus, the pixel location of a feature can be transformed into world frame coordinates up to some length scale, s. The ambiguity of this length scale is resolved through the line-of-sight depth data provided by the aligned LIDAR frame. It will be shown that the extrinsic parameters, and thus the relative pose of the camera, may be measured by comparing the camera's observation of a point to the point's known location in some feature frame.

Autoland Navigation Solution (ANS) Algorithm

The ANS algorithm stems from a matched 3D feature registration algorithm used in Simultaneous Localization and Mapping (SLAM) applications [22, pp. 407-411]. The algorithm begins with transforming the feature frame corner vector, $\mathbf{r}_f$, to the camera frame corner vector, $\mathbf{r}_c$ through some unknown rotation matrix $\mathbf{R}$ and translation $\mathbf{t}$, illustrated in Figure 3.

$$\mathbf{r}_c = \mathbf{R}\mathbf{r}_f + \mathbf{t} \tag{2}$$

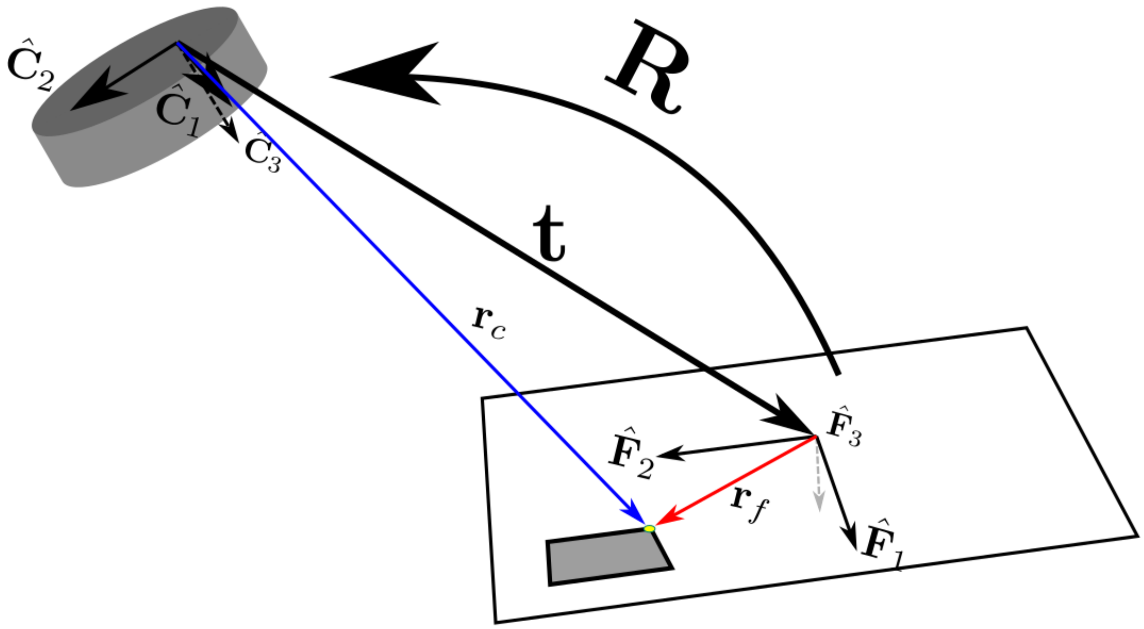

Figure 3. Autoland Navigation Solution Feature Co-Location

Cayley Transform[22] is applied to re-parameterize the rotation matrix to Classical Rodrigues Parameters (CRPs) $\mathbf{q} = [q_1, q_2, q_3]^T$ which reduces the number of independent attitude parameters from six to three.

$$\mathbf{R} = \left[\mathbb{I} + \tilde{\mathbf{q}}\right]^{-1} \left[\mathbb{I} - \tilde{\mathbf{q}}\right]; \quad \tilde{\mathbf{q}} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{3}$$

Substituting this into equation (2)

$$\left[\mathbb{I} + \tilde{\mathbf{q}}\right] \mathbf{r}_c = \left[\mathbb{I} - \tilde{\mathbf{q}}\right] \mathbf{r}_f + \left[\mathbb{I} + \tilde{\mathbf{q}}\right] \mathbf{t} \tag{4}$$

$$\mathbf{d} = -\tilde{\mathbf{q}}\mathbf{s} + \mathbf{t}' \tag{5}$$

Where

$$\mathbf{d} = (\mathbf{r}_c - \mathbf{r}_f), \quad \mathbf{s} = (\mathbf{r}_c + \mathbf{r}_f), \quad \mathbf{t}' = \left[\mathbb{I} + \tilde{\mathbf{q}}\right] \mathbf{t} \tag{6}$$

Given $n \geq 3$ detected corners, a linear matrix equation is formed.

$$\begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_n \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{s}}_1 & \mathbb{I} \\ \vdots & \vdots \\ \tilde{\mathbf{s}}_n & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{t}' \end{bmatrix} \Leftrightarrow \mathbf{y} = \mathbf{A}\mathbf{x}, \quad \mathbf{y} \in \mathbb{R}^{3n\times 1}, \ \mathbf{A} \in \mathbb{R}^{3n\times 6}, \ \mathbf{x} \in \mathbb{R}^{6\times 1} \tag{7}$$

The least squares solution for the state vector, $\mathbf{x}$, is given by

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1} \mathbf{A}^T\mathbf{y} \tag{8}$$

Finally, the relative translation vector is recovered by

$$\mathbf{t} = \left[\mathbb{I} + \tilde{\mathbf{q}}\right]^{-1} \mathbf{t}' \tag{9}$$

FLIGHT VEHICLE

ANS is attached to a quad-rotor UAS that is controlled autonomously using e-modification Model Reference Adaptive Control (e-MRAC). This section briefly outlines the dynamics of the quadcopter, e-MRAC control technique, and the hardware integration of the navigation system.

Flight Dynamics

The translational *kinematic* equation of a quadrotor is given by[17]

$$\dot{\mathbf{r}}_A^{\mathbb{I}}(t) = R(\phi(t), \theta(t), \psi(t))\mathbf{v}_A(t) \tag{10}$$

$$\mathbf{r}_{A(t_0)}^{\mathbb{I}} = \mathbf{r}_{A,0}^{\mathbb{I}} \tag{11}$$

$$\phi \in [0, 2\pi), \quad \theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right), \quad \psi \in [0, 2\pi)), \quad t \geq t_0$$

and R is the Euler 3-2-1 rotation sequence. The rotational *kinematic* equation of a quadrotor is given by [23, Th. 1.7]

$$\begin{bmatrix} \dot{\phi}(t) \\ \dot{\theta}(t) \\ \dot{\psi}(t) \end{bmatrix} = \Gamma\left(\phi(t), \theta(t)\right) \boldsymbol{\omega}(t) \tag{12}$$

$$\begin{bmatrix} \phi(t_0) \\ \theta(t_0) \\ \phi(t_0) \end{bmatrix} = \begin{bmatrix} \phi_0 \\ \theta_0 \\ \psi_0 \end{bmatrix} \tag{13}$$

where $\Gamma(\phi(t), \theta(t))$ is defined as

$$\Gamma(\phi(t), \theta(t)) \triangleq \begin{bmatrix} 1 & \sin\phi(t)\tan\theta(t) & \cos\phi(t)\tan\theta(t) \\ 0 & \cos\phi(t) & -\sin\phi(t) \\ 0 & \sin\phi(t)\sec\theta(t) & \cos\phi(t)\sec\theta(t) \end{bmatrix} \tag{14}$$

Since $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$, the matrix Γ is invertible [23, pp. 18-19]. Assuming constant quadrotor mass m, the translational *dynamic* equation is given by[24]

$$\begin{aligned} \mathbf{F}_g(\phi(t), \theta(t)) - \mathbf{F}_T(t) + \mathbf{F}_a(t) = m[&\dot{\mathbf{v}}_A(t) + \left[\boldsymbol{\omega}(t)^\times\right] \mathbf{v}_A(t) + \ddot{\mathbf{r}}_C(t) \\ &+ \left[\dot{\boldsymbol{\omega}}(t)^\times\right] \mathbf{r}_C(t) + 2\left[\boldsymbol{\omega}(t)^\times\right] \dot{\mathbf{r}}_C(t) \\ &+ \left[\boldsymbol{\omega}(t)^\times\right]\left[\boldsymbol{\omega}(t)^\times\right] \mathbf{r}_C(t)] \end{aligned} \tag{15}$$

$$\mathbf{v}_A(t_0) = \mathbf{v}_{A,0} \tag{16}$$

where $[\cdot^\times]$ denotes the skew-symmetric cross product matrix of a vector, $\mathbf{F}_T$ is the thrust force, $u_1(t)$ is the control input, the term $\mathbf{F}_a$ captures the aerodynamic forces acting on the system and $\mathbf{F}_g$ is the force due to gravity defined by

$$\mathbf{F}_g(\phi, \theta) = mg \begin{bmatrix} -\sin\theta(t) \\ \cos\theta(t)\sin\phi(t) \\ \cos\theta(t)\cos\phi(t) \end{bmatrix} \tag{17}$$

$$(\theta, \phi) \in [0, 2\pi) \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$$

The rotational *dynamic* equation of a quadrotor, whose chassis is modeled as a rigid body and propellers are modeled as thin, spinning discs, is given by[24]

$$\begin{aligned} \mathbf{M}_T(t) + \mathbf{M}_g\left(\mathbf{r}_C(t), \phi(t), \theta(t)\right) + \mathbf{M}_a(t) &= \left[\mathbf{r}_C^\times\right]\left[\dot{\mathbf{v}}_A(t) + \left[\boldsymbol{\omega}(t)^\times\right]\mathbf{v}_A(t)\right] \\ &\quad + I\dot{\boldsymbol{\omega}}(t) + \left[\boldsymbol{\omega}(t)^\times\right] I\boldsymbol{\omega}(t) + \mathbf{l}_{ct} + \mathbf{l}_g \qquad (18) \\ \mathbf{l}_{ct} &\triangleq I_P \sum_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ \dot{\Omega}_{P,i}(t) \end{bmatrix} \\ \mathbf{l}_g &\triangleq \left[\boldsymbol{\omega}(t)^\times\right] I_P \sum_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ \Omega_{P,i}(t) \end{bmatrix} \\ \mathbf{M}_g\left(\mathbf{r}_C(t), \phi(t), \theta(t)\right) &\triangleq \mathbf{r}_C \times \mathbf{F}_g \\ \boldsymbol{\omega}(t_0) &= \boldsymbol{\omega}_0, \qquad t \geq t_0 \end{aligned}$$

where $\mathbf{M}_T$ is the moment due to the propeller forces given by $[u_2(t), u_3(t), u_4(t)]^{\mathrm{T}}$, u_2 is the moment about the x-axis of the body reference frame, u_3 is the moment about the y-axis of the body reference frame, u_4 is the moment about the z-axis of the body reference frame, $\mathbf{M}_g$, is the moment due to the quadrotor's weight with respect to the reference point A, $\mathbf{M}_a$ is the moment due to aerodynamic forces with respect to A, $\mathbf{l}_{ct}$ is the inertial counter-torque, and $\mathbf{l}_g$ is the gyroscopic effect.

The control input $u(t)$ is related to the necessary thrust produced by the ith propeller $T_i(t)$ along the $z(\cdot)$ axis of the body reference frame $\mathbb{J}$ by

$$\mathbf{u}(t) = M_u \mathbf{T}(t) \tag{19}$$

where $u(t)$ is given by

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \end{bmatrix} \tag{20}$$

where M_u is the *mixer* and defined as

$$M_u \triangleq \begin{bmatrix} 1 & 1 & 1 & 1 \\ -L_y & L_y & L_y & -L_y \\ -L_x & L_x & -L_x & L_x \\ c_T & c_T & -c_T & -c_T \end{bmatrix} \tag{21}$$

where $L_x, L_y > 0$ is the distance of the propellers from the reference point A along the x-axis and y-axis, respectively; $c_T > 0$ denotes the propeller's aerodynamic drag coefficient. Let $\mathbf{T}(t)$ denote the *vector of thrust forces* given by

$$\mathbf{T}(t) = \begin{bmatrix} T_1(t) \\ T_2(t) \\ T_3(t) \\ T_4(t) \end{bmatrix} \tag{22}$$

where $T_i, i = 1, \ldots, 4$, denotes the force produced by the ith propeller along the $z(\cdot)$ axis of the body reference frame $\mathbb{J}$. The mixer M is invertible if and only if the product $L_x L_y c_T \neq 0$. Note that this mixer is defined for a quadrotor in X-configuration, with the motor labelling shown below:

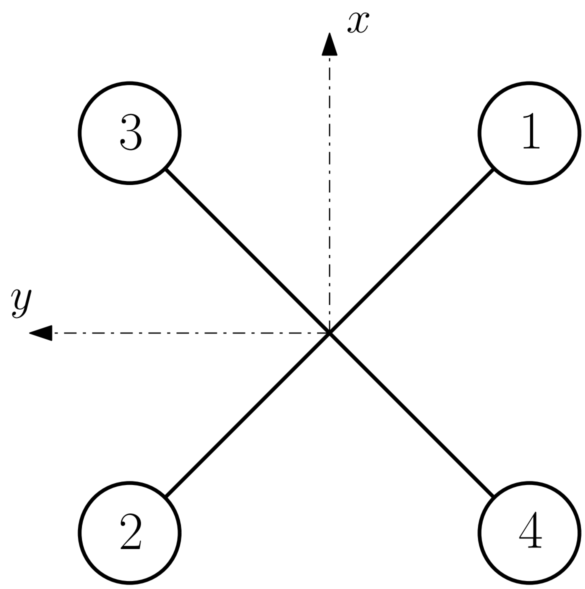

Figure 4. Motor ordering diagram for quadrotor in X-configuration.

The propeller's angular velocity is related to the thrust force generated by the *i*th propeller by

$$T_i(t) = k\Omega^2_{\mathrm{P},i}(t) \tag{23}$$

where $k > 0$[25,26]

e-MRAC Algorithm

This section presents a form of Robust MRAC based on the e-modification from [12, pp. 327-329], which is also considered as σ modification.[27] This control law enjoys the benefits of the classical MRAC, while being robust to bounded external disturbances, and guarantees uniform ultimate boundedness of the adaptive gains. Bounded adaptive gains help to alleviate an issue with classical MRAC known as "parameter drift". Furthermore, after a finite time transient, this control law guarantees that the plant's measured output will track a given reference signal with bounded error, despite uncertainties in the model and the presence of external disturbances. The main difference between the e-modification and classical MRAC is that the classical MRAC does not guarantee uniform ultimate boundedness of the adaptive gains, or trajectory tracking error *in the presence of unmatched uncertainty*, $\hat{\xi}(\cdot)$.

Consider the nonlinear, time-varying plant dynamics given by

$$\begin{aligned} \dot{\mathbf{x}}_p(t) &= A_p\mathbf{x}_p(t) + B_p\Lambda[\mathbf{u}(t) + \Theta^{\mathrm{T}}\mathbf{\Phi}(\mathbf{x}_p(t))] + \hat{\boldsymbol{\xi}}(t) \\ \mathbf{x}_p(t_0) &= \mathbf{x}_{p,0}, \quad t \geq t_0 \end{aligned} \tag{24}$$

where $\mathbf{x}_p(t)$ denotes the trajectory of the plant, $\mathbf{u}(t) \in \mathbb{R}^m$ denotes the control input, $A_p \in \mathbb{R}^{n_p \times n_p}$ is unknown, $B_p \in \mathbb{R}^{n_p \times m}$, $\Lambda \in \mathbb{R}^{m \times m}$ is diagonal, positive-definite, and unknown,$\Theta \in \mathbb{R}^{N \times m}$ is unknown, Φ is the Lipschitz continuous *regressor vector*, $\hat{\boldsymbol{\xi}}$ is continuous, unknown, and is assumed to be bounded i.e $\|\hat{\boldsymbol{\xi}}\| \leq \boldsymbol{\xi}_{\max}\ \forall\, t \geq t_0$.

It is assumed that Λ is such that the pair $(A_p, B_p\Lambda)$ is controllable and $\Lambda_{\min}\mathbb{I}_{m\times m} \leq \Lambda$, for some $\Lambda_{\min} > 0$. The plant's *parametric* and *matched* uncertainties are captured in Λ and $\Theta^{\mathrm{T}}\mathbf{\Phi}(\mathbf{x}_p)$, respectively, which includes possible malfunctions in the control system. The term $\hat{\xi}(\cdot)$ captures the *unmatched* uncertainties such as external disturbances of the plant like wind gusts.

Consider the plant's sensor dynamics given by

$$\dot{\mathbf{y}}(t) = \epsilon C_p \mathbf{x}_p(t) - \epsilon y(t), \qquad y(t_0) = C_p \mathbf{x}_{p,0} \tag{25}$$

where $\mathbf{y}(t) \in \mathbb{R}^m$ denotes the system output, $\epsilon > 0$, and $C_p \in \mathbb{R}^{m\times n_p}$. Thus, the sensor dynamics are modeled as linear dynamical systems. The uncontrolled sensor dynamics are exponentially stable and characterized by the parameter $\epsilon > 0$. Now, consider a given *reference command* $\mathbf{r}$, which has a continuous first derivative, and define $\mathbf{r}_2(t) \triangleq \dot{\mathbf{r}}(t)$, and assume that both $\mathbf{r}(t), \mathbf{r}_2(t)$ are bounded, i.e

$$\|\mathbf{r}(t)\| \leq \mathbf{r}_{\max}, \quad t \geq t_0 \tag{26}$$

$$\|\mathbf{r}_2(t)\| \leq \mathbf{r}_{\max,2} \text{for some} \quad \mathbf{r}_{\max}, \mathbf{r}_{\max,2} > 0 \tag{27}$$

The goal of Robust MRAC is design a feedback control law, $\mathbf{u}(\cdot)$, such that, after a finite-time transient, the measured output $\mathbf{y}(\cdot)$ is able to track the reference signal $\mathbf{r}(\cdot)$ with some bounded error i.e.

$$if \quad \|\mathbf{y}(t_0) - \mathbf{r}(t_0)\| \leq a \tag{28}$$

$$then \quad \|\mathbf{y}(t) - \mathbf{r}(t)\| \leq b; \quad t \geq t_0 + T, \quad a \in (0, c), \quad b, c > 0 \tag{29}$$

For this, we create an augmented system where $n \triangleq n_p + m$ and $x(t) \triangleq \left[x_p^{\mathrm{T}}(t), [y(t) - r(t)]^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^n$, $t \geq t_0$, which allows (24) and (25) to be expressed as

$$\dot{x}(t) = Ax(t) + B\Lambda[u(t) + \Theta^{\mathrm{T}}\Phi(x_p(t))] + \xi(t), \quad x(t_0) = \begin{bmatrix} x_{p,0} \\ C_p x_{p,0} - r(t_0) \end{bmatrix}, \quad t \geq 0 \tag{30}$$

$$A \triangleq \begin{bmatrix} A_p & 0_{n_p\times m} \\ \epsilon C_p & -\epsilon I_m \end{bmatrix}$$

$$B \triangleq \begin{bmatrix} B_p \\ 0_{m\times m} \end{bmatrix}$$

$$\xi(t) \triangleq \begin{bmatrix} 0_{n_p\times m} \\ -1_m \end{bmatrix} \left[r_2(t) + \epsilon r(t)\right] + \begin{bmatrix} 1_{n_p} \\ 0_{m\times n_p} \end{bmatrix} \hat{\xi}(t)$$

Also consider the *reference model* whose dynamics are given by

$$\dot{x}_{\mathrm{ref}}(t) = A_{\mathrm{ref}} x_{\mathrm{ref}}(t) + B_{\mathrm{ref}} r(t), \qquad x_{\mathrm{ref}}(t_0) = \begin{bmatrix} x_{\mathrm{p},0} \\ C_p x_{\mathrm{p},0} - r(t_0) \end{bmatrix}, \qquad t \geq t_0 \tag{31}$$

$$A_{\mathrm{ref}} = \begin{bmatrix} A_{\mathrm{ref},1} & 0_{n_p\times m} \\ 0_{m\times n_p} & A_{\mathrm{ref},2} \end{bmatrix}$$

where $x_{\mathrm{ref}}(t) \in \mathbb{R}^n$ is the *reference model system state*, $A_{\mathrm{ref},1} \in \mathbb{R}^{n_p\times n_p}$ and $A_{\mathrm{ref},2} \in \mathbb{R}^{m\times m}$ are both Hurwitz, and $B_{\mathrm{ref}} \in \mathbb{R}^{n\times m}$ is such that the pair $(A_{\mathrm{ref}}, B_{\mathrm{ref}})$ is controllable.

Define the error as $e \triangleq x(t) - x_{\text{ref}}(t)$, and let the control be of the form

$$\begin{aligned} \beta(\hat{K}, \pi) &= \hat{K}^{\text{T}}(t)\pi(t) \\ \hat{K}^{T} &= \left[K_x^{\text{T}}, K_r^{\text{T}}, -\Theta^{\text{T}}\right] \\ \pi(t) &= \left[x(t), r(t), \Phi(x(t))\right]^{\text{T}} \end{aligned} \tag{32}$$

where $\hat{K}^T \in \mathbb{R}^{m\times(n+m+N)}$ is the control gain, $K_x \in \mathbb{R}^{n\times m}$, $K_r \in \mathbb{R}^{m\times m}$, $\Theta \in \mathbb{R}^{N\times n}$ and $\pi(t) \in \mathbb{R}^{n+m+N}$.

The adaptive law is given by

$$\dot{\hat{K}}^{\text{T}}(t) = -\Gamma\left(\pi(t)e^{\text{T}}(t)PB + \sigma\|B^{\text{T}}Pe(t)\|\hat{K}^{T}(t)\right) \tag{33}$$

$$\hat{K}(t_0) = 0_{(n+m+N)\times(n+m+N)} \quad t \geq 0$$

$$\Gamma = \begin{bmatrix} \Gamma_x & 0_{n\times m} & 0_{n\times N} \\ 0_{m\times n} & \Gamma_r & 0_{m\times N} \\ 0_{N\times n} & 0_{N\times m} & \Gamma_\theta \end{bmatrix} \tag{34}$$

$$\Gamma \in \mathbb{R}^{(n+m+N)\times(n+m+N)}, \Gamma_x \in \mathbb{R}^{n\times n}, \Gamma_r \in \mathbb{R}^{m\times m}, \Gamma_\theta \in \mathbb{R}^{N\times N}$$

$$\sigma \triangleq \begin{bmatrix} \sigma_x I_n & 0_{n\times m} & 0_{n\times N} \\ 0_{m\times n} & \sigma_r I_m & 0_{m\times N} \\ 0_{N\times n} & 0_{N\times m} & \sigma_\theta I_N \end{bmatrix} \tag{35}$$

$$\sigma_x, \sigma_r, \sigma_\theta \in \mathbb{R}, \quad \sigma_x, \sigma_r, \sigma_\theta > 0$$

where Γ_i are the adaptive learning rates which are user-defined and positive-definite, and $P \in \mathbb{R}^{n\times n}$ is the positive definite solution to the Lyapunov equation given by

$$0 = A_{\text{ref}}^{\text{T}}P + PA_{\text{ref}} + Q \tag{36}$$

where $Q \in \mathbb{R}^{n\times n}$ is user-defined and positive definite. If K_x and K_r exist such that the following matching conditions are satisfied,

$$\begin{aligned} A_{\text{ref}} &= A + B\Lambda K_x^{\text{T}} \\ B_{\text{ref}} &= B\Lambda K_r^{\text{T}} \end{aligned} \tag{37}$$

then the nonlinear time-varying system (30) with $u(t) = \beta(\hat{K}, \pi)$, $t \geq t_0$, is uniformly ultimately bounded and (28) is verified.

Application

To apply this adaptive control technique to the quadcopter UAV, the quadcopter's dynamics is first cast to the form of (24). It is typical to consider the altitude dynamics and attitude dynamics separately. The coupling of translational dynamics (x, y) and attitude dynamics combined with the under-actuation of the quadrotor UAV lead to a decomposition of the control structure to an outer/inner loop architecture. The *outer loop* considers the user-defined reference trajectory, and uses a PID controller to compute the ideal acceleration to stabilize the vehicle about the trajectory. By assuming small angle approximations and small angular velocities, the corresponding reference attitude is determined. This reference attitude is the reference command sent to the *inner loop*, which employs the adaptive control law to drive the attitude of the vehicle to the desired attitude.

Outer Loop Let $\mathbf{r}_{\text{cmd}}(t) = [x_{\text{cmd}}(t), y_{\text{cmd}}(t), z_{\text{cmd}}(t), \psi_{\text{cmd}}(t)]^{\text{T}}$ denote the user-defined reference trajectory. The purpose of the outer loop is to convert $\mathbf{r}_{\text{cmd}}(t)$ to reference roll, $\phi_{\text{cmd}}(t)$, and pitch, $\theta_{\text{cmd}}(t)$, angles. Given $\mathbf{r}_{\text{cmd}}(t)$ such that $\dot{r}_{\text{cmd}}(t) \leq r_{\text{d,max}}$, the ideal accelerations $\ddot{x}_{\text{ideal}}$, $\ddot{y}_{\text{ideal}}$, $\ddot{z}_{\text{ideal}}$ are computed using a PID controller:

$$\begin{aligned}
\ddot{x}_{\text{ideal}}(t) &= -K_{\text{p,x}} e_{\text{x}}(t) - K_{\text{i,x}} \int_0^t e_{\text{x}}(t) dt - K_{\text{d,x}} \dot{e}_{\text{x}}(t) \\
\ddot{y}_{\text{ideal}}(t) &= -K_{\text{p,y}} e_{\text{y}}(t) - K_{\text{i,y}} \int_0^t e_{\text{y}}(t) dt - K_{\text{d,y}} \dot{e}_{\text{y}}(t) \\
\ddot{z}_{\text{ideal}}(t) &= -K_{\text{p,z}} e_{\text{z}}(t) - K_{\text{i,z}} \int_0^t e_{\text{z,outer}}(t) dt - K_{\text{d,z}} \dot{e}_{\text{z,outer}}(t) \\
e_x(t) &\triangleq x_{\text{q}}(t) - x_{\text{cmd}}(t) \\
\dot{e}_x(t) &= \dot{x}_{\text{q}}(t) - \dot{x}_{\text{cmd}}(t)
\end{aligned}$$

where the error dynamics for y and $z, outer$ are defined similarly and $[x_q(t), y_q(t), z_q(t)]^T$ and $[\dot{x}_q(t), \dot{y}_q(t), \dot{z}_q(t)]^T$ denote the inertial position and velocity of the body reference frame, respectively. From the small angle and angular velocity assumption, the ideal force along the z-axis of the inertial frame is given by

$$F_{\text{z,ideal}}(t) = m\left(\ddot{z}_{\text{ideal}} - g\right) \tag{38}$$

The reference attitude is determined using trigonometry

$$\begin{aligned}
\phi_{\text{cmd}}(t) &= m \frac{\ddot{x}_{\text{cmd}}(t)\sin(\psi(t)) - \ddot{y}_{\text{cmd}}(t)\cos(\psi(t))}{\text{dead}(F_{\text{z,ideal}}(t))} \\
\theta_{\text{cmd}}(t) &= m \frac{\ddot{x}_{\text{cmd}}(t)\cos(\psi(t)) + \ddot{y}_{\text{cmd}}(t)\sin(\psi(t))}{\text{dead}(F_{\text{z,ideal}}(t))}
\end{aligned}$$

where

$$\text{dead}(x) \triangleq \begin{cases} x, & \text{if } x > \epsilon_{\text{dead}} \\ \epsilon_{\text{dead}} & \text{otherwise} \end{cases}$$

is a dead-band function. This is used to avoid a singularity in the case the applied force in the z-direction $F_z(t)$ is zero. The limit for the dead-band is typically $\epsilon_{\text{dead}} = 0.1$. This reference attitude is sent to the *inner loop*, where the vehicle's attitude is driven to the reference attitude using an the adaptive control law.

Inner Loop In model reference adaptive control, the first step is to determine an ideal reference model (31) for the plant to track. For this application, reference model is split into four systems, one for each of the four inner loop "channels" which are the altitude $z(t)$, and three Euler angles $\phi(t), \theta(t)$, and $\psi(t)$. Each reference model is designed to mimic a mass-spring damper system, which has easily quantifiable properties such as rise time, overshoot, settling time, etc; and is physically intuitive. After selecting the reference model, you must solve the Lyapunov equation to obtain the sensitivity matrix P, which is a 2×2 matrix for each channel.

Given the inner loop command signal $\mathbf{r}_{\text{cmd,inner}}(t) = [z_{\text{cmd}}(t), \phi_{\text{cmd}}(t), \theta_{\text{cmd}}(t), \psi_{\text{cmd}}(t)]^{\text{T}}$, the reference model is integrated to determine the ideal trajectory for each channel using Runge-Kutta 4. This yields the altitude trajectory tracking error $e_z(t) = z(t) - z_{\text{ref}}(t)$, with $e_\phi(t), e_\theta(t)$, and $e_\psi(t)$ defined similarly. The trajectory tracking errors and inner loop command signal are two of three necessary pieces of information for computing the adaptive law. The third piece, the regressor vector, is computed below.

Consider the altitude dynamics of the quadcopter UAV

$$\begin{aligned}\dot{z}(t) &= v_z(t),\\ \ddot{z}(t) &= \frac{u_1(t)\cos(\phi(t))\cos(\theta(t))}{m} - g\end{aligned}$$

This system is cast in the form of (24) by first adding and subtracting $\frac{u_1}{m}$ on the right hand side of the $\ddot{z}(t)$ equation

$$\begin{aligned}\dot{z}(t) &= v_z(t),\\ \ddot{z}(t) &= \frac{u_1(t)\cos(\phi(t))\cos(\theta(t))}{m} - g + \frac{u_1(t)}{m} - \frac{u_1(t)}{m}\end{aligned}$$

hence, in state space form:

$$\begin{aligned}\begin{bmatrix}\dot{z}(t)\\ \ddot{z}(t)\end{bmatrix} &= \underbrace{\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}}_{A_{\mathrm{z}}}\begin{bmatrix}z(t)\\ \dot{z}(t)\end{bmatrix} + \underbrace{\begin{bmatrix}0\\ 1\end{bmatrix}}_{B_{\mathrm{z}}}\frac{1}{m}\left(u_1(t) + \Theta_{\mathrm{z}}^{\mathrm{T}}\Phi_{\mathrm{z}}(t, x_p(t))\right) + \xi_{\mathrm{z}}(t)\\ \Theta_{\mathrm{z}} &= m\\ \Phi_{\mathrm{z}}(t, x(t)) &= -g\\ \xi_{\mathrm{z}}(t) &= \frac{u_1(t)\left(\cos(\phi(t))\cos(\theta(t)) - 1\right)}{m}\end{aligned}$$

Note that the mass of the vehicle m may be subject to uncertainty, or be entirely unknown, hence $\Lambda_{\mathrm{z}} = \frac{1}{m}$ which is sign-definite, diagonal, and unknown; furthermore, $\xi_{\mathrm{z}}(t)$ is unknown. Lastly, since it is reasonable to assume $m > 0$, and the total thrust that can be applied to the quadcopter UAV by the propellers $u_1(t)$ is finite, then $\xi_{\mathrm{z}}(t)$ is bounded. Also note that if the UAV is level at time $t = t_n$, such that $\phi(t_n) = 0$ and $\theta(t_n) = 0$, then $\xi_z = 0$, and there is no unmatched uncertainty. However, if the system was modeled to account for more complicated physical phenomena such as inertial counter-torque, gyroscopic effect, or the center of mass not coinciding with the reference point (which in reality, is always the case), the unmatched uncertainty will not be equal to zero.

Assume the reference point and center of mass coincide. For Class 1 UAVs, the effects of inertial counter-torque and gyroscopic effect are negligible, and can be ignored. Consider the attitude dynamics of the quadcopter UAV

$$\begin{aligned}u(t) &= I\dot{\omega}(t) + \omega^{\times}(t)I\omega(t) + I_P\sum_{i=1}^{4}\begin{bmatrix}0\\ 0\\ \dot{\Omega}_{P,i}(t)\end{bmatrix} + \omega^{\times}(t)I_P\sum_{i=1}^{4}\begin{bmatrix}0\\ 0\\ \Omega_{P,i}(t)\end{bmatrix}\\ \omega(t_0) &= \omega_0, \qquad t \geq t_0.\end{aligned}$$

where $M_T(t) = [u_2(t), u_3(t), u_4(t)]^{\mathrm{T}}$ is the control torque. The attitude dynamics of the quadcopter

UAV can be expressed equivalently as

$$\begin{aligned}
\omega(t) &= \int \dot{\omega}(t)dt, \quad t \geq t_0 \\
\dot{\omega}(t) &= \underbrace{I^{-1}}_{B\Lambda}\left(u(t) + \Theta_{\text{att}}^{\mathrm{T}}\Phi_{\text{att}}(t, x_p(t))\right) + \underbrace{I^{-1}\left(I_P \sum_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ \dot{\Omega}_{P,i}(t) \end{bmatrix} + \omega^{\times}(t) I_P \sum_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ \Omega_{P,i}(t) \end{bmatrix}\right)}_{\xi_{\text{att}}(t)} \\
\Theta_{\text{att}} &= \text{diag}\left(\frac{I_2 - I_3}{I_1}, \frac{I_1 - I_3}{I_2}, \frac{I_1 - I_2}{I_3}\right) \\
\Phi_{\text{att}}(t, x_p(t)) &= [\omega_2\omega_3, -\omega_1\omega_3, \omega_1\omega_2]^{\mathrm{T}}
\end{aligned}$$

Note that since I is positive-definite, I^{-1} exists. Assuming the angular acceleration of the propeller $\dot{\Omega}_{P,i}(t)$ is finite (since the torque that the actuators can apply is finite), then $\Omega_{P,i}(t)$ is also finite. Hence, $\xi_{\text{att}}(t)$ is bounded. The unmatched uncertainty thus far has been a result of the modeling of the vehicle's dynamics, and not necessarily unforeseen disturbances. A common disturbance can be drag and aerodynamic moments due to wind, and can be considered as an unmatched uncertainty. Since the UAV's speed is finite, the drag and aerodynamic torque acting on the vehicle is finite, hence, the unmatched uncertainty is finite, and the e-modification of MRAC can be applied.

EXPERIMENTAL METHOD

The lab experiments were performed within an indoor flight area equipped with Cortex Motion Capture to provide truth position and attitude; see Figure 5 for a schematic of the experiment. The landing platform is a 6 DOF Stewart platform attached atop a remote-controlled cart driven by Mecanum wheels such that the cart can linearly translate forwards and sideways, as well as yaw. The platform is programmed to simulate the motion of a ship experiencing a heave sea-state value of one, corresponding to heave motion amplitudes up to six inches; the angular sea-state is set at four which corresponds to roll and pitch amplitudes up to fifteen degrees. The objective of the experiment was to exercise the adaptive control law to understand the operational limit of the controller, and also to observe the performance of the ANS relative pose solution. With this objective, the following flight test cadence was designed.

The flight test began with take-off from the Stewart platform, while carrying an 85 gram payload attached to a servo near the nose of the craft. The quadrotor was prescribed a circle trajectory given by

$$\begin{aligned}
x_{\text{ref}}(t) &= 0.75\cos(t - t_{c,0}), \quad t \geq t_{c,0} \\
y_{\text{ref}}(t) &= 0.75\cos(t - t_{c,0}), \\
z_{\text{ref}}(t) &= 1.50\text{m}, \\
\psi_{\text{ref}}(t) &= 0
\end{aligned}$$

where $t_{c,0} > 0$ denotes the start time of the circle trajectory. During the flight, the quadrotor was subjected to multiple disturbances. At $t = 55s$, a simulated motor failure of up to 20% is injected into the system as a matched uncertainty. This was done by multiplying the second row of the mixer (21) by the % failure value, i.e. 0.15 for a 15% motor failure. The payload was dropped roughly $5s$ after the motor failure. A large fan was used to introduce a wind gust into the mission, and can be considered an unmatched uncertainty. The fan was pointed such that the freestream velocity was tangent to the circle trajectory and hence, the UAV flew into and out of the freestream at certain points along the prescribed trajectory. Soon after the payload was released, the quadcopter was commanded to begin tracking the moving ground robot/Stewart platform. Let $r_q(t) = [x_q(t), y_q(t), z_q(t)]^{\mathrm{T}}$ denote the position of the UAV , and $r_s(t) = [x_s(t), y_s(t), z_s(t)]^{\mathrm{T}}$ denote the position of the landing platform. The trajectory executed by the UAV was prescribed by the following function

$$\begin{aligned} r_{\mathrm{ref}}(t) &= \begin{cases} \lambda(t) r_q(t) + (1-\lambda(t)) r_s(t), & \text{if } \{\|r_q(t) - r_s(t)\| > 1.0\} \cup \{\lambda \in [0,1]\} \\ r_s(t), & \text{otherwise.} \end{cases} \\ \psi_{\mathrm{ref}}(t) &= \psi_s(t), \end{aligned}$$

where $\lambda(t) = \frac{t - t_{s,0}}{4}$, and $t_{s,0} \geq 0$ denotes the time at which the follow ship command was given. With this $\lambda(t)$, the UAV was given 4 seconds to arrive at the ship position. Note that if $\|r_q(t_{s,0}) - r_s(t_{s,0})\| > 1.0$ the ANS solution typically had not been acquired yet, so a different solution was required to deduce the pose of the ship. For this demonstration, the motion capture solution was used. Finally, the quadrotor landed on the moving ground robot/Stewart platform, according to the landing profile:

$$\begin{aligned} x_{\mathrm{ref}}(t) &= x_s(t), \quad t \geq t_{\ell,0} \\ y_{\mathrm{ref}}(t) &= y_s(t), \\ z_{\mathrm{ref}}(t) &= \begin{cases} \lambda(t) z_q(t) + (1-\lambda(t)) z_s(t), & \text{if } \{\|r_q(t) - r_s(t)\| > \epsilon\} \cup \{\lambda(t) \in [0,1]\}, \\ z_s(t), & \text{otherwise,} \end{cases} \\ \psi_{\mathrm{ref}}(t) &= \psi_s(t). \end{aligned}$$

where $t_{\ell,0} > 0$ denotes the time at which the UAV was given the command to follow the ground robot, $\lambda(t) = \frac{t - t_{\ell,0}}{6}$ giving the UAV 6 seconds of smooth descent. This landing profile was selected to ensure that the UAV would not decrease altitude if the UAV is not at least $\epsilon = 0.15$m from the landing platform's position $r_s(t) = [x_s(t), y_s(t), z_s(t)]^{\mathrm{T}}$. If the condition $\|r_q(t) - r_s(t)\| > \epsilon$ was met, the initial time $t_{\ell,0}$ was reset to the current time t, giving the UAV extra time to finish the landing maneuver. This was done to ensure a soft landing on the motion platform.

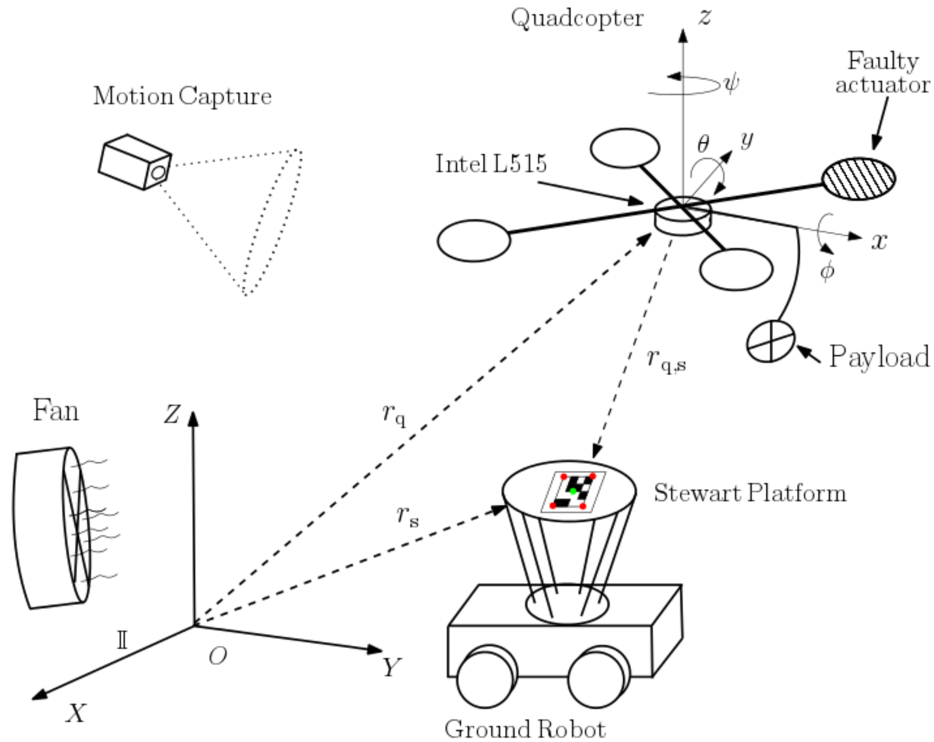

Figure 5. Schematic of Experimental Setup

Flight Data

Flight data from a representative test is shown below. Displayed in Figure 6 is the defined reference trajectory (red line) and the actual path that the UAS flew (black line). As shown, the quadrotor is able to accurately track the prescribed position and attitude despite the disturbances. In each subfigure, around time $t = 55$s, the 20% motor failure takes effect, causing large excursions in the trajectory tracking error, which is rapidly compensated for by the adaptive controller. Noticeably, the figure in the top row, third column, presents the altitude of the UAV and reference altitude. It can be seen that at the time of the motor failure, the UAV drops approximately 0.4m since the total thrust has been significantly reduced. However, the adaptive controller has adjusted to the new dynamics of the system, and quickly drives the UAV back to the reference altitude while carrying the payload.

Referring to the the top row, second column of Figure 6, the y-position overshoots the desired trajectory consistently. This is due to the behavior of the slung payload making it difficult to track the desired roll angle (bottom row, first column), and the wind disturbance. This can be improved by tuning the outer loop gains to provide better roll and pitch references that will guide the UAV to the desired position. The position tracking can also be improved by tuning the inner loop adaptive rates to compensate more aggressively to disturbances. Referring to the figure in the bottom row, third column, the measurement of the UAV's heading angle includes some systematic error, hence there is an offset.

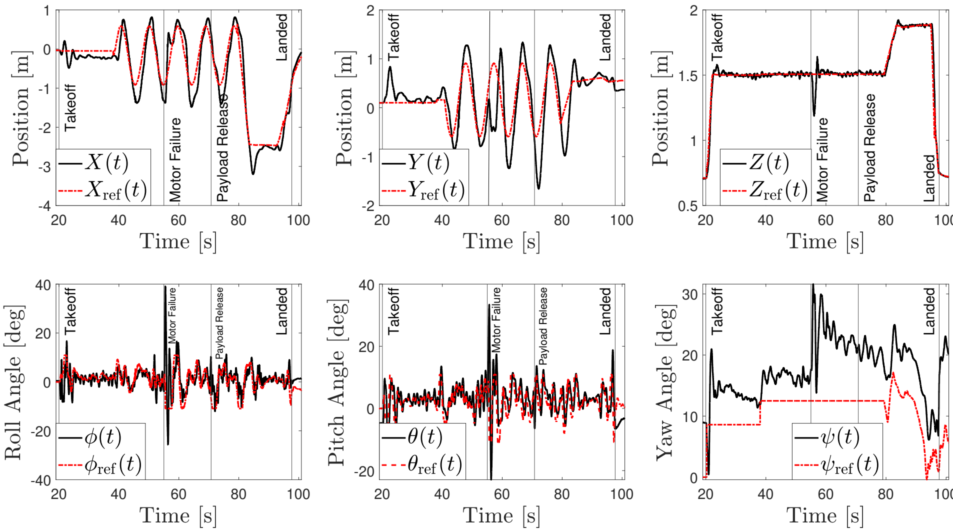

Figure 6. (*Top row*) Quadrotor position in meters; (*Bottom row*) Quadrotor attitude in degrees. The reference trajectories are represented by the red line, while the actual flight trajectories are shown in black.

Figure 7 illustrates the $\mathcal{L}_2$-norm of the trajectory tracking error over time from the experiment. At times $t = 20$s, $t = 38.25$s, $t = 55$s, $t \approx 71$s, $t \approx 79$s, there are increases in the position and attitude error due to the takeoff command, circle tracking command, motor failure disturbance, payload release, ship tracking command, and land command, respectively. The e-modification of MRAC law drives the error to a neighborhood of zero rapidly after each event. Notice that, shortly after $t = 55$s when the motor failure occurs, the attitude tracking error's $\mathcal{L}_2$-norm increases to $\approx 63°$. This exemplifies how the 20% significantly impacts the system's performance, and simultaneously, how the adaptive controller is able to quickly compensate for this failure, and prevent a crash.

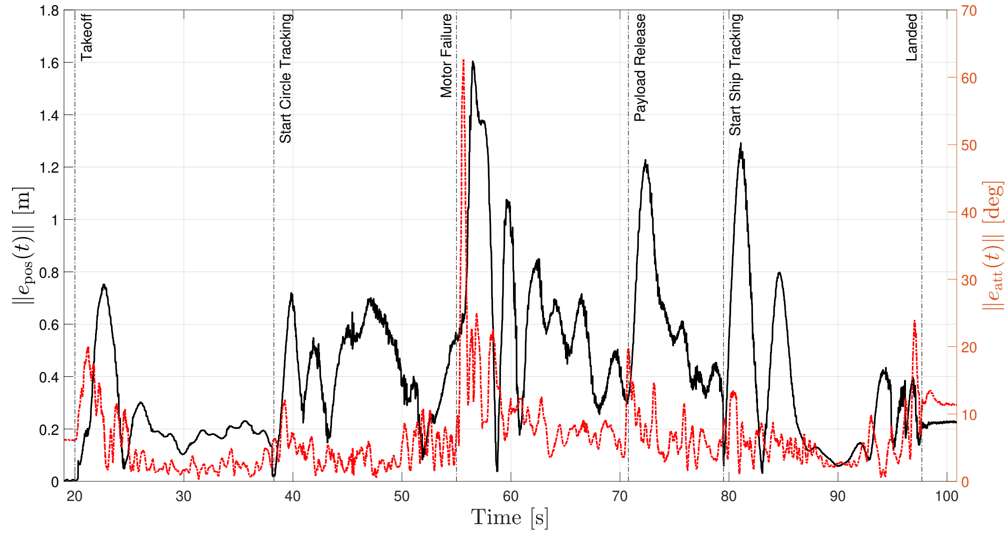

Figure 7. (*Black line*) Norm of the position error in meters; (*Red line*) Norm of the altitude tracking error in degrees.

Figure 8 shows the adaptive gains evolution over time. As can be seen in the figure in the first row, first column, the gains corresponding to the altitude control adjust rapidly after takeoff to account for the mass uncertainty and added payload. Between large disturbances, many of the adaptive gains are constant, except for the $\hat{\Theta}(\cdot)$ gain, which is used to reject matched uncertainties. To enable rejection of large disturbances such as the sudden 20% reduction in thrust of one actuator, the adaptive rates corresponding to $\hat{\Theta}(\cdot)$ are an order of magnitude larger than the adaptive rates corresponding to $\hat{K}_{\text{x}}, \hat{K}_{\text{r}}$. The adaptive gains for the roll and pitch channels change very rapidly, especially $\hat{\Theta}(\cdot)$, in response to the slung payload, wind, and motor failure. The roll channel adaptive gain even changes sign to compensate for the motor failure, implying the existence of a persistent disturbance in the roll dynamics, hence, the change in sign of the adaptive gain. The magnitude of the motor failure masks this unknown disturbance. This disturbance is the slung payload, wind, and a truly malfunctioning motor (discovered after-wards). The adaptive gains corresponding to the heading dynamics evolve slowly since their adaptive rates are significantly lower than those of the other channels. Large yaw motions have not been included as a part of our experiments, hence, tuning the adaptive rates for heading control was not necessary.

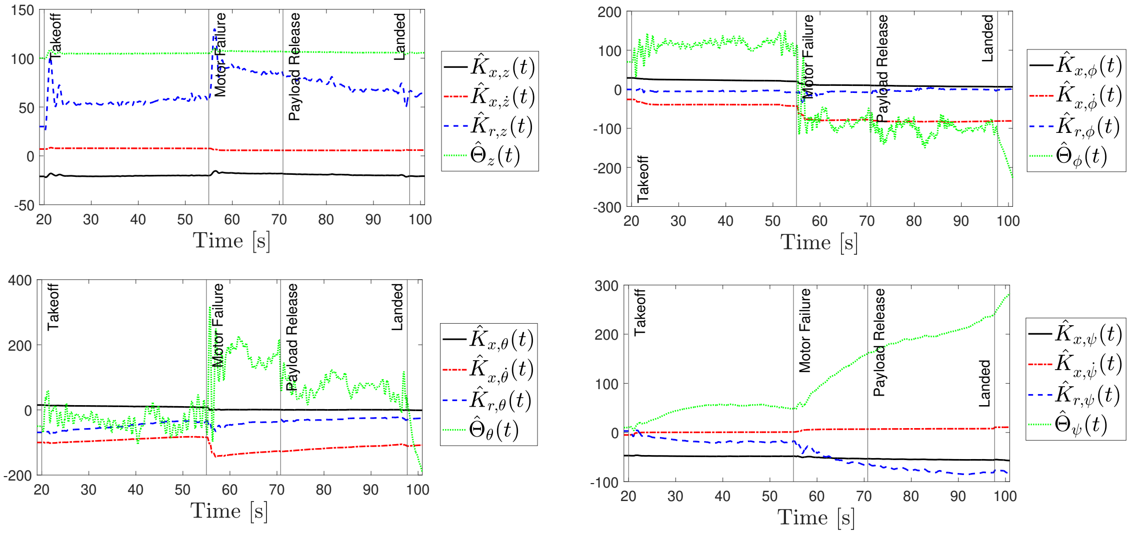

Figure 8. Adaptive gains computed using e-modification adaptive laws. (*top left*) shows the evolution of the adaptive altitude gains: $\hat{K}_{x,z}(t)$, $\hat{K}_{x,\dot{z}}(t)$, $\hat{K}_{r,z}(t)$, $\hat{\Theta}_{x,z}(t)$. Similarly, the evolution of adaptive gains are shown for the roll channel (*top right*), pitch channel (*bottom left*), and yaw channel (*bottom right*).

Figure 9 shows the control inputs over time. Over the time interval $t \in [20, 50]$, the total thrust $u_1(t)$ indicates that approximately 63% of the UAV's total thrust capability is required to stay aloft with the added 85g payload. In the same time interval, the UAV requires a constant roll moment to be applied to stay level. This is due to the slung payload being mounted slightly off of the x-axis of the body's reference frame. Similarly, a constant yawing moment is applied to keep the UAV facing forward. This is a result of the wind disturbance and a malfunctioning motor.

At time $t = 55$s, the control inputs rapidly respond to the changing behavior of the UAV as a result of the induced motor failure and slung payload. The total thrust $u_1(t)$ increases significantly to improve the altitude tracking despite the loss of available thrust. The roll, pitch, and yaw moments $u_2(t)$, $u_3(t)$, and $u_4(t)$ increase in magnitude following the motor failure as well. At time $t \approx 71$s, the payload is released, and the total thrust $u_1(t)$ decreases accordingly. At the same time, the pitching moment $u_3(t)$ increases slightly since the payload, once mounted on the nose of the aircraft, has been released, and a pitching moment needs to be applied to keep the trajectory tracking error small.

Interestingly, the adaptive controller is only able to ensure mission completeness in the face of large disturbances if there is enough excess thrust to maintain good levels of control authority. If there was not enough excess thrust, the motor failure would result in aborting the flight rather than continue the mission. This is still better than the performance provided by the common PID controller, which is unable to recover from this large disturbance.

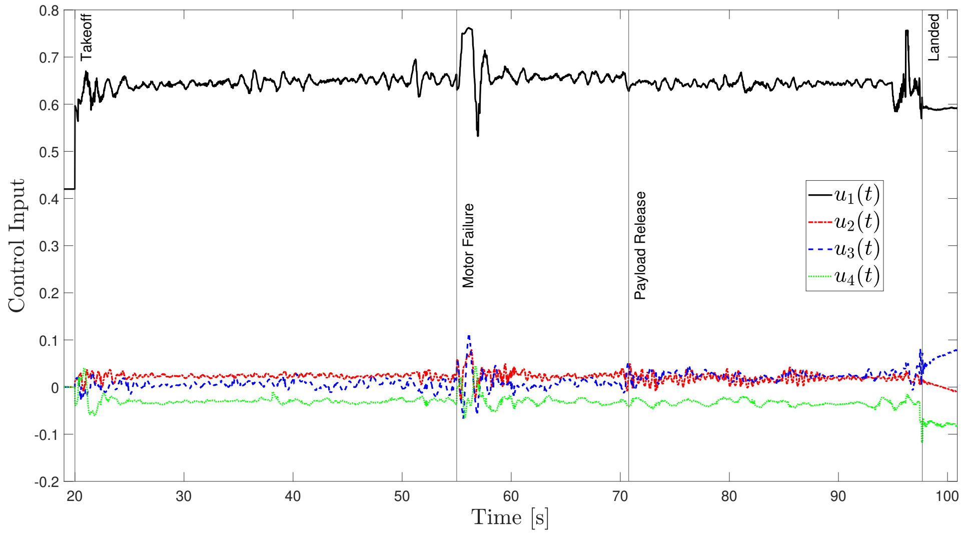

Figure 9. Total thrust $u_1(t)$, roll moment $u_2(t)$, pitch moment $u_3(t)$, and yaw moment $u_4(t)$ generated by e-modification MRAC law.

ANS Output

Figures 10 and 11 are the measured relative position and attitude, respectively, from ANS compared with the truth data from the motion capture system (MoCAP). Due to processing limitations, a measurement solution was produced at a rate of 4-8Hz which was too slow for the UAS flight computer to use as a relative pose solution for navigation purposes. As such, the flight computer used the motion capture data for the duration of the experiment.

The position estimate in Figure 10 shows the relative position of the sensor, but coordinitized in the inertial frame of the motion capture system. Thus, for instance, the negative value of relative Z shows that the UAS must decrease its altitude to land on the target platform. Note there is an altitude bias between the motion capture data and the ANS data. That is due to a roughly 10cm difference in location ANS frame and the body frame of the quadcopter as viewed by the MoCAP system.

Often, the ANS solution is able to accurately measure the relative pose, especially for the Y and Z channels. However, is a significant amount of noise present in the ANS solution. These errors highlight how the LIDAR/vision-based system is susceptible to factors such as motion blur, mechanical vibration, and image latency. These factors affect the precision of the feature detection algorithm of the camera, which cascades to the computation of the relative pose solution.

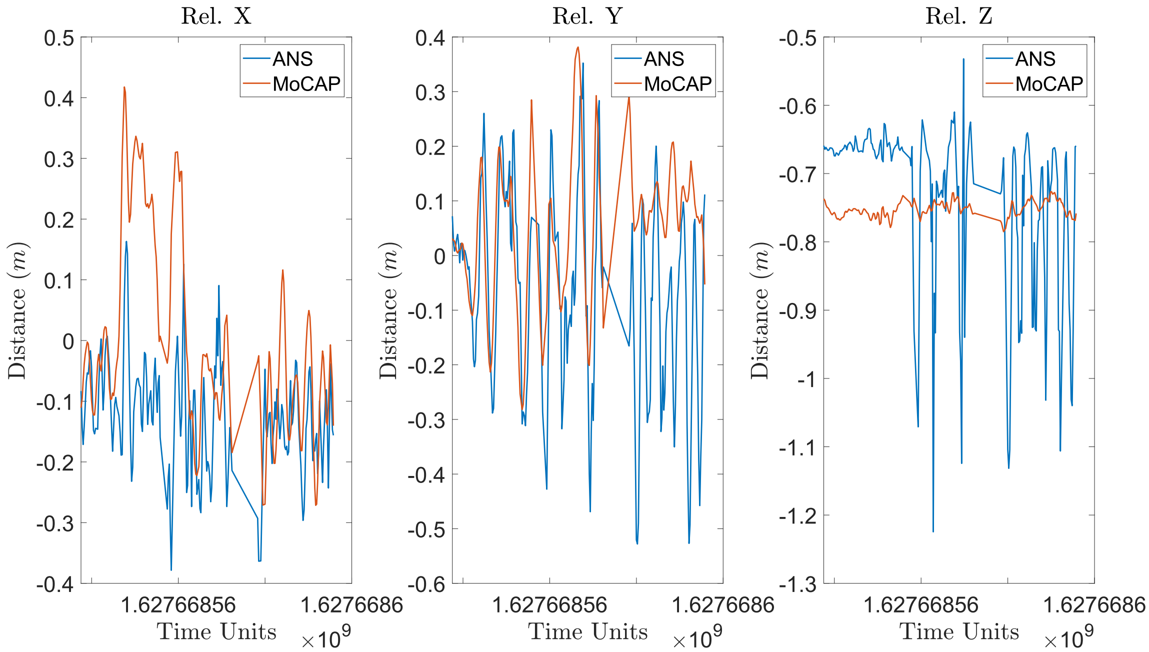

Figure 10. Relative position measurement from ANS (*blue line*) compared to cortex motion capture data(*orange line*).

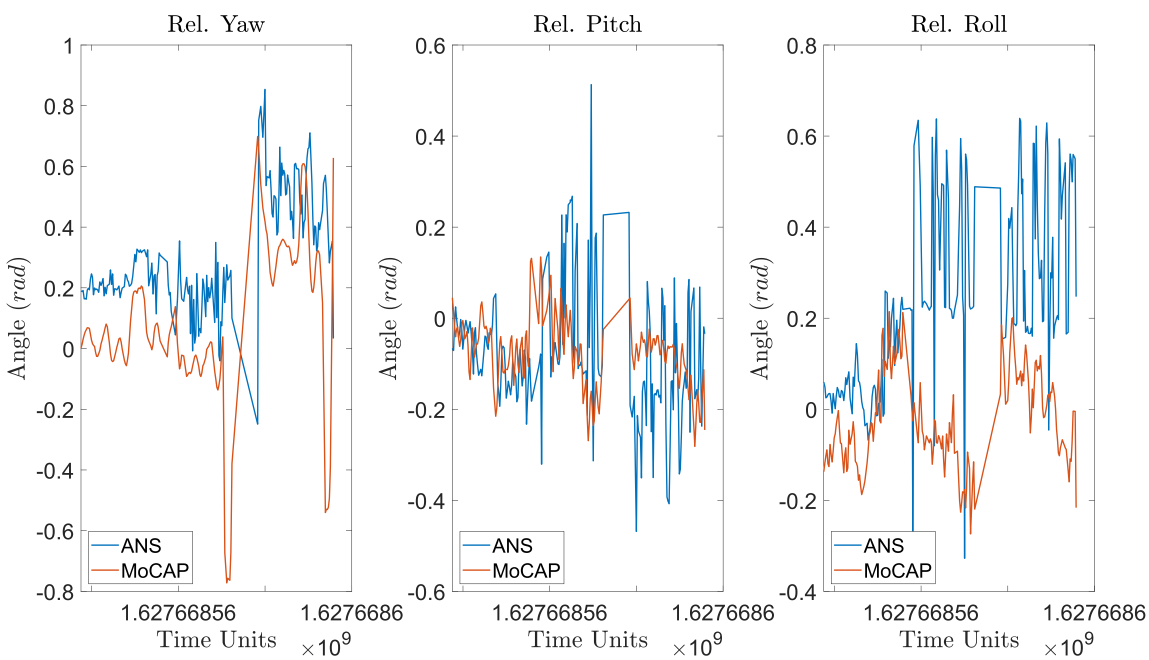

Figure 11. Relative attitude measurement from ANS (*blue line*) compared to cortex motion capture data(*orange line*).

CONCLUSION

This objective of this paper was to present a fully developed and capable integration of LIDAR-based navigation to a Class 1, autonomous UAS. The UAS is controlled using a state-of-the-art robust adaptive control law and demonstrates rejection of flight disturbances. Successful demonstration of these flight capabilities supports ongoing validation of adaptive control techniques for field use. The new sensor package offers a lightweight, and low cost navigation solution for GPS-denied e.g. extraterrestrial environments. In light of the noise and errors in the navigation solution, the flight tests were considered a success due to the successful implementation of the adaptive control law in response to flight disturbances, and relative pose measurement solution achieved through an on-board LIDAR/vision device. This data will be used to identify key areas of improvement with the overall flight package.

ACKNOWLEDGEMENTS:

The Office of Naval Research (ONR Grant Number N00014-19-1-2435) is acknowledged for their support. The authors are thankful to Dr. Brian Holm-Hansen and Dr. David Gonzales for their support. Partial support from National Geospatial Intelligence Agency (NGA) through Grant No. HM0476-19-1-2015 is gratefully acknowledged. Dr. Scott A. True is thanked for serving as a technical monitor.

REFERENCES

[1] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, Vol. 37, No. 4, 2020, pp. 50–61.

[2] S. Royo and M. Ballesta-Garcia, "An overview of lidar imaging systems for autonomous vehicles," *Applied sciences*, Vol. 9, No. 19, 2019, p. 4093.

[3] A. Johnson, A. Ansar, L. Matthies, N. Trawny, A. Mourikis, and S. Roumeliotis, "A general approach to terrain relative navigation for planetary landing," *AIAA Infotech@ Aerospace 2007 Conference and Exhibit*, 2007, p. 2854.

[4] T. P. Setterfield, R. A. Hewitt, P.-T. Chen, A. Teran Espinoza, N. Trawny, and A. Katake, "LiDAR-inertial based navigation and mapping for precision landing," 2021.

[5] T. G. McGee, D. Adams, K. Hibbard, E. Turtle, R. Lorenz, F. Amzajerdian, and J. Langelaan, "Guidance, Navigation, and Control for Exploration of Titan with the Dragonfly Rotorcraft Lander," *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 1330.

[6] K. Reddy, R. Toffoletto, and K. Jones, "Numerical simulation of ship airwake," *Computers & fluids*, Vol. 29, No. 4, 2000, pp. 451–465.

[7] C. H. Kääriä, Y. Wang, G. D. Padfield, J. S. Forrest, and I. Owen, "Aerodynamic loading characteristics of a model-scale helicopter in a ship's airwake," *Journal of Aircraft*, Vol. 49, No. 5, 2012, pp. 1271–1278.

[8] D. Lee and J. F. Horn, "Simulation of pilot workload for a helicopter operating in a turbulent ship airwake," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 219, No. 5, 2005, pp. 445–458.

[9] C. Liu and W.-H. Chen, "Disturbance rejection flight control for small fixed-wing unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 12, 2016, pp. 2810–2819.

[10] G. Cai, B. M. Chen, X. Dong, and T. H. Lee, "Design and implementation of a robust and nonlinear flight control system for an unmanned helicopter," *Mechatronics*, Vol. 21, No. 5, 2011, pp. 803–820.

[11] J. D. Anderson and M. L. Bowden, *Introduction to flight*. McGraw-Hill Higher Education, 2005.

[12] E. Lavretsky and K. A. Wise, "Robust adaptive control," *Robust and adaptive control*, pp. 317–353, Springer, 2013.

[13] P. A. Ioannou and J. Sun, *Robust adaptive control*. Courier Corporation, 2012.

[14] 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," *AIAA Guidance, Navigation, and Control Conference*, 2011, p. 6716.

[15] A. Chamseddine, Y. Zhang, C.-A. Rabbath, J. Apkarian, and C. Fulford, "Model reference adaptive fault tolerant control of a quadrotor UAV," *Infotech@ Aerospace 2011*, p. 1606, 2011.

[16] H. Liu, W. Zhao, Z. Zuo, and Y. Zhong, "Robust control for quadrotors with multiple time-varying uncertainties and delays," *IEEE Transactions on Industrial Electronics*, Vol. 64, No. 2, 2017, pp. 1303–1312.

[17] A. L'afflitto, R. B. Anderson, and K. Mohammadi, "An introduction to nonlinear robust control for unmanned quadrotor aircraft: how to design control algorithms for quadrotors using sliding mode control and adaptive control techniques [focus on education]," *IEEE Control Systems Magazine*, Vol. 38, No. 3, 2018, pp. 102–121.

[18] M. Mohammadi and A. M. Shahri, "Adaptive nonlinear stabilization control for a quadrotor UAV: Theory, simulation and experimentation," *Journal of Intelligent & Robotic Systems*, Vol. 72, No. 1, 2013, pp. 105–122.

[19] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and vision Computing*, Vol. 76, 2018, pp. 38–47.

[20] S. Garrido-Jurado, R. Munoz-Salinas, F. J. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognition*, Vol. 51, 2016, pp. 481–491.

[21] K. M. Dawson-Howe and D. Vernon, "Simple pinhole camera calibration," *International Journal of Imaging Systems and Technology*, Vol. 5, No. 1, 1994, pp. 1–6.

[22] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*, Vol. 2. Chapman & Hall/CRC Boca Raton, FL, 2004.

[23] A. L'Afflitto, *A mathematical perspective on flight dynamics and control*. Springer, 2017.

[24] A. L'Afflitto and K. Mohammadi, "Equations of Motion of Rotary-Wing Unmanned Aerial System with Time-Varying Inertial Properties," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 2, 2017, pp. 559–564.

[25] L. R. G. Carrillo, A. E. D. López, R. Lozano, and C. Pégard, *Quad rotorcraft control: vision-based hovering and navigation*. Springer Science & Business Media, 2012.

[26] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," tech. rep., Epfl, 2007.

[27] K. Mohammadi and A. L'Afflitto, "Robust Adaptive Output Tracking for Quadrotor Helicopters," *Adaptive Robust Control Systems*, InTech, 2018.