IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

From Propeller Damage Estimation and Adaptation to Fault Tolerant Control: Enhancing Quadrotor Resilience

Jeffrey Mao*, Jennifer Yeom*, Suraj Nair, and Giuseppe Loianno

***Abstract*— Aerial robots are required to remain operational even in the event of system disturbances, damages, or failures to ensure resilient and robust task completion and safety. One common failure case is propeller damage, which presents a significant challenge in both quantification and compensation. We propose a novel adaptive control scheme capable of detecting and compensating for multi-rotor propeller damages, ensuring safe and robust flight performance. Our control scheme includes an L1 adaptive controller for damage inference and compensation of single or dual propellers, with the capability to seamlessly transition to a fault-tolerant solution in case the damage becomes severe. We experimentally identify the conditions under which the L1 adaptive solution remains preferable over a fault-tolerant alternative. Experimental results validate the proposed approach, demonstrating its effectiveness in running the adaptive strategy in real time on a quadrotor even in case of damage to multiple propellers.**

Supplementary Material

Video: https://youtu.be/MVhyvXxSR9g

I. Introduction

Micro Aerial Vehicles (MAVs) such as quadrotors are becoming ubiquitous in applications such as search and rescue [1] and aerial photography [2]. However, MAV-related accidents hinder the growth of the industry and erode public confidence in drone safety. As a result, it is necessary to develop inference and control approaches that can guarantee safe and reliable flight in case of system damage. Propellers on MAVs are susceptible to damage, especially in the event of collisions or after prolonged use, primarily due to their size, location, and lightweight construction. In addition, detecting the occurrence and location of propeller failures is challenging and difficult to estimate. Indirectly sensing individual motor thrusts is further complicated due to the nonlinear dynamics of the quadrotor and the difficulty of measuring higher order angular acceleration or moment terms. Our work specifically targets quadrotors and introduces an adaptive control scheme to autonomously infer and adjust quadrotor flight behavior in response to propeller damages (defined in our case as the percent of thrust loss from a damaged propeller compared to a healthy propeller spinning at the same speed) or failures. This is achieved by first using adaptive control to adapt to damage and then inferring the damage from the adaptive control action unlike other works which first estimate a motor damage and then perform a corrective action.

*These authors contributed equally.
The authors are with the New York University, Tandon School of Engineering, Brooklyn, NY 11201, USA. email: {jm7752, jennifer.yeom, surajkiron, loiannog}@nyu.edu.

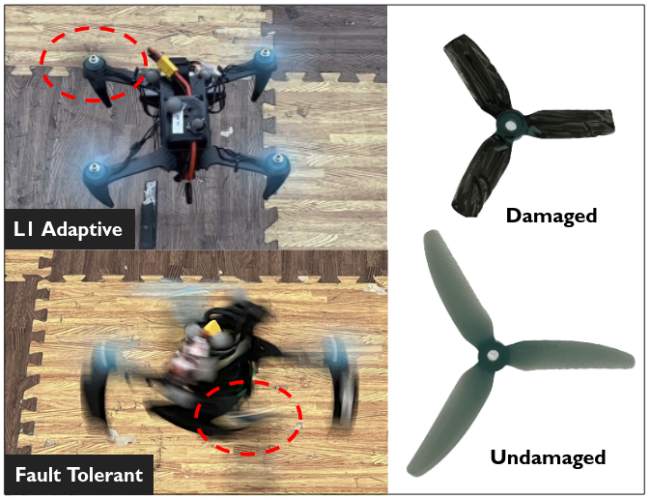

Fig. 1: L1 adaptive and fault-tolerant control with one damaged propeller (circled in red dash). The fault-tolerant controlled quadrotor is forced to spin (over 1000 deg/sec).

This work makes the following contributions. First, we introduce an adaptive control scheme in case of propeller damage, ensuring robust flight performances. Our solution combines an L1 adaptive control scheme, capable of detecting and compensating for single or dual propeller damages with minimal computational overhead and low latency, along with a fault-tolerant mode. This mode can be seamlessly activated when propeller damage reaches a critical level. Second, we study the trade-offs between adaptive control and fault-tolerant control as shown in Fig. 1 and experimentally identify the best operating modality according to the level of propeller damage. Third, we conduct a comprehensive analysis in both simulation and real-world experiments to assess the extent to which the L1 adaptive controller can improve tracking performance under various degrees of propeller damage. Finally, we validate our strategy on a real quadrotor, successfully adapting to propeller damage and transitioning in real-time to a fault-tolerant solution. This is the first work to tackle precise estimation of the propeller damage which has significant more difficulties than motor damage estimates of other works. These difficulties include unequal loss between thrust and torque coefficients along with additional noise on the system from damaged propellers.

II. Related Works

Research in fault estimation covers a variety of vehicles detailed in this survey [3] such as fixed wings and rotary aerial robots. We can divide methods handling propeller

damage into two categories: empirical and state estimation methods. Empirical works address propeller damage estimation through parameter identification [4] and sensor noise analysis [5]. Work in [4] successfully estimates the loss of effectiveness of a single motor but is only tested in simulation. The correlation between changes in accelerometer noise to variations in propeller speeds is used in [5] to estimate propeller damage, but the technique is only tested on a minuscule amount of propeller damage.

Other works rely on state estimation to accurately measure the damage [6], [7]. These typically have a slower response time than empirical methods, and are only able to handle motor damage estimation rather than propeller damage as in the proposed work. This means they assume the damage from the propeller motor pair can be represented as a scaled factor of each individual RPM. For the case of propeller damage, this assumption is not accurate. The thrust and torque coefficient scales with the prop radius, r, as a function of r^4 and r^5 respectively [8]. This increases the complexity of our propeller damage compared to [6], [7] problem. Typically a reduced state Kalman Filter [6], [7], [9] is used in either a single [6], [9] or cascaded [10], [11] form. These studies test their systems by artificially injecting a fault that decelerates the motor rather than testing true propeller damages. State estimation through cascaded Kalman Filters can infer and adapt to 60% motor damage in [11]. However, it is only tested in simulation for motor not propeller damage and does not include the possibility to switch to the fault-tolerant mode. Our work negates up to $40-60\%$ propeller damage in real-world experiments as shown in Fig. 1.

Once propeller damage is estimated, there are two options to incorporate damage compensation into control. The first option is adaptation where the damaged propeller is given a scaled up motor action [6], [7] to compensate for the damage. This adaptation is effective for small and medium damages but is unable to stabilize the robot for severe damages. The second option is to employ a fault-tolerant control strategy that completely disables the damaged propeller [4], [12], which is typically employed in case of severe damage. However, fault-tolerant control requires the robot to surrender yaw control causing rapid spins [13], [14], making it unsuitable for small or medium damages. Our proposed work combines both methods to estimate and adapt to small or medium propeller damage, and transition to fault-tolerant for severe damage.

Two notable adaptive techniques are Incremental Nonlinear Dynamic Inversion (INDI) [14] and L1 [15]. These works provide additional control actions that augment the capabilities of a baseline controller to reject damages or disturbances without calculating propeller damage estimates. However neither work provides an estimate to judge when propeller damage is too severe for adaption. Other methods exist for adaptation [16]–[18], through model identification or automatic gain tuning, but cannot adapt to rapid changes like the complimentary above. In this work, we design a L1 adaptive technique to compensate for propeller damage and give accurate propeller damage estimates. Unlike INDI [14] that requires angular acceleration which is difficult to estimate, our method only requires angular velocity which can be obtained from an on board Inertial Measurement Unit.

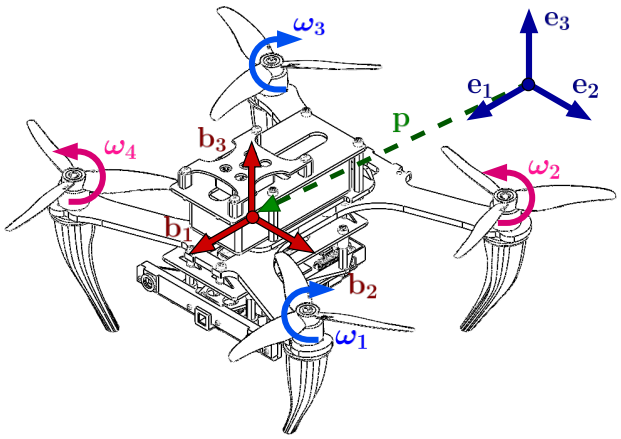

Fig. 2: MAV model with inertial and body frame definitions.

Existing fault-tolerant control strategies are adept at ensuring robust control even in the event of single or multiple rotor failures [3], [13], [14], [19]. However, these solutions require identifying the faulty propeller, and sacrificing yaw control, resulting in a rapid yaw spinning as depicted in Fig. 1. Our work introduces an L1 adaptive control scheme designed to detect, infer, and compensate for propeller damage, with the capability to smoothly transition to a fault-tolerant solution when severe damage is detected.

III. Methodology

A. Preliminaries

We use two coordinate frames to represent the dynamics of the system. As shown in Fig. 2, we choose the inertial reference frame as $\begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3\end{bmatrix}$ and the body fixed frame as $\begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3\end{bmatrix}$. The origin of the body frame is aligned with the center of mass of the MAV. The body frame follows the East North Up (ENU) coordinate system. The first axis $\mathbf{b}_1$ is aligned to the heading and the third axis, $\mathbf{b}_3$ aligns with the thrust vector of the vehicle. The dynamics of the system can be written as

$$\begin{aligned}\dot{\mathbf{p}} &= \mathbf{v},\\ m\dot{\mathbf{v}} &= mg\mathbf{e}_3 + f\mathbf{R}\mathbf{e}_3,\\ \dot{\mathbf{q}} &= \frac{1}{2}\mathbf{q}\otimes\begin{bmatrix}0\\ \boldsymbol{\Omega}\end{bmatrix},\\ \mathbf{M} &= \mathbf{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega}\times\mathbf{J}\boldsymbol{\Omega},\end{aligned} \tag{1}$$

where $\mathbf{p}\in\mathbb{R}^3$, and $\mathbf{v}\in\mathbb{R}^3$ are the position and velocity in the body frame. $\mathbf{R}\in SO(3)$ is a rotation matrix from the body fixed frame to the inertial frame, $\mathbf{q}$ is the quaternion representation of $\mathbf{R}$, and $\otimes$ is quaternion multiplication. f, m, g are the thrust, mass and gravity respectively, $\boldsymbol{\Omega}\in\mathbb{R}^3$ is the angular velocity with respect to the body frame, $\mathbf{M} = \begin{bmatrix}M_1 & M_2 & M_3\end{bmatrix}^\top$ are the moments around the three body frame axes, and $\mathbf{J}\in\mathbb{R}^{3\times3}$ is the robot inertia matrix.

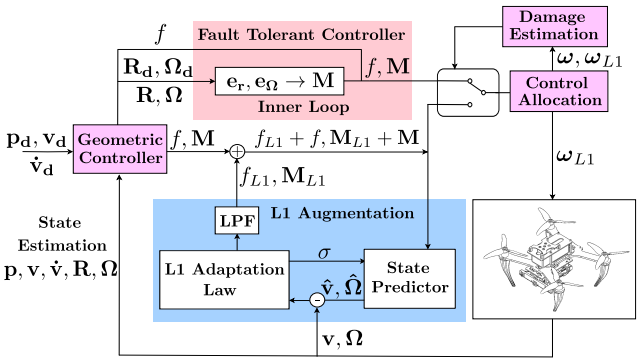

Fig. 3: Our Resilient Control Scheme.

B. Control Design

We adapt the geometric controller derived in [20] for the possibility of propeller damage, or losing one or two opposing rotors in flight. An outer controller solves for the desired attitude $\mathbf{R}$, thrust f, and angular velocities $\mathbf{\Omega}$. The inner controller solves for the moments required to control the attitude of the MAV. The geometric controller tracks position $\mathbf{p}$ and yaw ψ by generating force and moments control signals. The L1 augmentation is inserted into the controller as seen in Fig. 3. The L1 controller uses a state predictor to estimate the partial state $[\mathbf{v}^\top \ \mathbf{\Omega}^\top]^\top$ with the unknown disturbances. The adaptation law compares the prediction to the measured state and generates the control output to compensate for disturbances. If the adaptation law accurately estimates the unknown disturbances then the tracking error will converge to zero [21] assuming the augmented action is with in the actuator limits.

1) Outer-loop Controller: As per the standard geometric control algorithm, the thrust vector is normalized and chosen as the third body axis, or thrust vector $\mathbf{b}_3$

$$\mathbf{b}_3 = \frac{\dot{\mathbf{v}} + g\mathbf{e}_3}{\|\dot{\mathbf{v}} + g\mathbf{e}_3\|}. \tag{2}$$

Next, we formulate the rotation matrix based on ψ or the yaw of the vehicle. We denote the individual elements of the thrust vector as $\mathbf{b}_3 = \begin{bmatrix} b_{3x} & b_{3y} & b_{3z} \end{bmatrix}^\top$. We can formulate a quaternion representing the tilt, $\mathbf{q}_{tilt}$, and yaw rotation $\mathbf{q}_\psi$ such as in [22]. The below quaternion formulations are all scalar first notation.

$$\begin{aligned} \mathbf{q}_{tilt} &= \frac{1}{\sqrt{2(1+b_{3z})}} \begin{bmatrix} 1+b_{3z} & -b_{3y} & b_{3x} & 0 \end{bmatrix}, \\ \mathbf{q}_\psi &= \begin{bmatrix} \cos(0.5\psi) & 0 & 0 & \sin(0.5\psi) \end{bmatrix}. \end{aligned} \tag{3}$$

Multiplying the quaternion $\mathbf{q}_{tilt}$ and $\mathbf{q}_\psi$ we formulate the full vehicle's rotation with $\mathbf{q}$ being the quaternion representation of $\mathbf{R}$. A rotation matrices representation of eq. (3) is in [13]

$$\mathbf{q} = \mathbf{q}_{tilt} \otimes \mathbf{q}_\psi. \tag{4}$$

The first control input, thrust, is solved for with

$$f = m\left(\mathbf{k}_p(\mathbf{p} - \mathbf{p}_d) + \mathbf{k}_v(\mathbf{v} - \mathbf{v}_d) + \dot{\mathbf{v}}_d + g\mathbf{e}_3\right) \cdot \mathbf{b}_3, \tag{5}$$

where $\mathbf{p}_d$ and $\mathbf{v}_d$ are desired position and velocity of the vehicle and $\mathbf{k}_p, \mathbf{k}_v \in \mathbb{R}^3$ are the gains for the respective errors. Lastly, the angular velocity $\mathbf{\Omega}$, is solved using

$$\begin{bmatrix} 0 \\ \mathbf{\Omega} \end{bmatrix} = 2\mathbf{q}^{-1} \otimes \dot{\mathbf{q}}, \tag{6}$$

as a function $\dot{\mathbf{b}}_3$ and $\dot{\psi}$ such as in [22] .

2) Inner-loop Controller: The inner controller solves for the second control input, moment, from the error in attitude and error in angular velocity. In a traditional geometric controller [20], the attitude tracking error, $\mathbf{e}_R$ is calculated by using the rotation matrices as

$$\mathbf{e}_R = \frac{1}{2}(\mathbf{R}_d^\top \mathbf{R} - \mathbf{R}^\top \mathbf{R}_d)^\vee. \tag{7}$$

where the $^\vee$ term converts a skew-symmetric matrix to vector form. For the fault-tolerant controller, we use a reduced attitude error metric to decouple the yaw control that shows improved performances as demonstrated in our recenter work [13] for the fault-tolerant control case

$$\mathbf{e}_{R_{\text{reduced}}} = \mathbf{b}_{3d} \times \mathbf{b}_3, \tag{8}$$

where $\mathbf{b}_{3d}$ is the normalized desired thrust vector and $\mathbf{b}_3$ is the z axis of the body fixed frame we get from the localization. The cross product of $\mathbf{b}_{3d}$ and $\mathbf{b}_3$ measures how much the actual direction deviates from the desired direction. The angular velocity error, $\mathbf{e}_\Omega$ is calculated by

$$\mathbf{e}_\Omega = \mathbf{\Omega} - \mathbf{R}^\top \mathbf{R}_d \mathbf{\Omega}_d. \tag{9}$$

The moments are calculated using

$$\mathbf{M} = -\mathbf{k}_R \mathbf{e}_R - \mathbf{k}_\Omega \mathbf{e}_\Omega + \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega}, \tag{10}$$

where $\mathbf{k}_R$ and $\mathbf{k}_\Omega$ are rotation and angular velocity gains.

C. Adaptive Control

We supplement our controller with an L1 adaptive controller by including two elements: 1) a state predictor to estimate the evolution of the partial state and the disturbances present and 2) an adaption law to solve for an action to counter the error between the estimated and true partial state.

1) State Predictor: We adopt a partial state predictor to estimate how the partial state $\mathbf{z} = \begin{bmatrix} \mathbf{v}^\top & \mathbf{\Omega}^\top \end{bmatrix}^\top$ evolves given an input $\boldsymbol{\mu} = \begin{bmatrix} f & \mathbf{M}^\top \end{bmatrix}^\top$ and disturbances $\boldsymbol{\sigma} \in \mathbb{R}^6$. These disturbances $\sigma = \begin{bmatrix} \mathbf{f}_{\text{ext}}^\top & \mathbf{M}_{\text{ext}}^\top \end{bmatrix}^\top$ represent the additional force and moments on the vehicle both expressed in the body frame not accounted in the standard controller dynamic model. Let k denote the discrete time step of the system. To acquire the next state from derivatives we simply perform an Euler approximation given a time step dt

$$\hat{\mathbf{z}}[k] = \mathbf{z}[k-1] + dt \cdot \begin{bmatrix} \dot{\mathbf{v}}[k-1] \\ \dot{\mathbf{\Omega}}[k-1] \end{bmatrix}, \tag{11}$$

where $\hat{\mathbf{z}}[k]$ is the predicted state for time k. Let the true observed state $\mathbf{z}$ be the values observed from our state estimation. We set a diagonal gain defined by the user $\mathbf{K} \in \mathbb{R}^{6\times6}$. The state transition is

$$\dot{\mathbf{v}} = g\mathbf{e}_3 + \frac{(f + f_{L1})}{m}\mathbf{R}\mathbf{e}_3 + \frac{1}{m}\mathbf{R}\boldsymbol{\sigma}_{1:3} + \mathbf{K}_{\mathbf{1:3}}(\hat{\mathbf{z}}_{1:3} - \mathbf{z}_{1:3}), \tag{12}$$

$$\dot{\mathbf{\Omega}} = \mathbf{J}^{-1}(\mathbf{M} + \mathbf{M}_{L1} + \sigma_{4:6} - \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega}) + \mathbf{K_{4:6}}(\hat{\mathbf{z}}_{4:6} - \mathbf{z}_{4:6}). \tag{13}$$

The terms f_{L1} and $\mathbf{M}_{L1}$ are the additional force and moment commanded the L1 adaptive controller.

2) L1 Adaptation Law: We can characterize the external disturbances on our force and moments. We use the previously selected $\mathbf{K}$ diagonal matrix to generates a specific gain, $\boldsymbol{\phi}$, to drive the disturbance exponentially to zero

$$\begin{aligned} \boldsymbol{\phi} &= (\exp(\mathbf{K} \cdot dt - \mathbf{I})^{-1}\mathbf{K}) \exp(\mathbf{K} \cdot dt), \\ \boldsymbol{\sigma} &= \begin{bmatrix} m\mathbf{R}^\top & 0 \\ 0 & \mathbf{J} \end{bmatrix} \boldsymbol{\phi}(\mathbf{z}[k] - \hat{\mathbf{z}}[k]). \end{aligned} \tag{14}$$

We then estimate our acceleration and moment disturbances using eq. (14) through mapping the velocity and angular velocity errors between our state predictor and estimation to calculate disturbances $\boldsymbol{\sigma}$. We solve the additional action $\boldsymbol{\mu}_{L1} = \begin{bmatrix} f_{L1} & \mathbf{M}_{L1}^\top \end{bmatrix}^\top$ from our disturbance $\boldsymbol{\sigma}$. The moment disturbance estimates $\boldsymbol{\sigma}_{4:6}$ and forces acting along the thrust axis $\boldsymbol{\sigma}_3$ can be negated by providing an equal but negative compensation to cancel the disturbance $\boldsymbol{\mu}_{L1} = -\boldsymbol{\sigma}_{3:6}$. We also implement an exponential Low Pass Filter (LPF) to effectively mitigate the noise present in the state estimation, thereby achieving smoother results: $\boldsymbol{\mu}_{L1} = -\text{LPF}(\boldsymbol{\sigma}_{3:6})$.

D. Propeller Damage Estimation

We estimate the propellers' damage using our augmented control. We assume that in nominal flight conditions a quadratic relationship between the i^{th} propeller thrust and corresponding motor speed $f_i = k_{f_{\text{real}}}\omega_i^2$ where $k_{f_{\text{real}}}$ represents the true thrust coefficient of the propeller and ω_i is the motor speed. A damaged propeller spinning at the original RPM provides lower thrust due to the change in $k_{f_{\text{real}}}$. We quantify our propeller damages on the thrust coefficient by

$$k_{f_{\text{mis}}} = 1 - \frac{k_{f_{\text{real}}}}{k_{f_{\text{model}}}}, \tag{15}$$

where $k_{f_{\text{model}}}$ is the thrust coefficient used by our controller. When a propeller gets damaged, both the thrust coefficient k_f and torque coefficient k_m are affected. The two coefficients decay as functions of the propeller radius r^4 and r^5 respectively [8]. Due to this relationship, a damaged propeller spinning at higher RPM will provide adequate thrust but less torque than a healthy propeller. The L1 adaptive law compensates for both thrust and torque mismatch requiring the healthy propellers' RPMs to compensate for the damaged propeller's lower torque. As the thrust and torque coefficients are highly coupled, we formulate an optimization problem for damage estimation. First, we initialize a baseline estimate of our propeller damage for each propeller. Next, we identify damaged and undamaged propellers to form a prior. Finally, we perform a null-space projection optimization.

We can estimate our initial guess $k_{f_{\text{real}}}$ using the additional RPM spin that L1 provides to our system. Specifically, let ω_i now be the RPM for motor i given by the geometric controller without L1 compensation and ω_{L1i} be the RPM for motor i with L1 compensation. Supposing that L1 adaptation has compensated for the propeller damage, we will have

$$k_{f_{\text{model}}}\omega_i^2 = k_{f_{\text{real}}}\omega_{L1i}^2. \tag{16}$$

Converting terms we have an initial guess

$$k_{f_{\text{real}}} = k_{f_{\text{model}}} \cdot \frac{\omega_i^2}{\omega_{L1i}^2}. \tag{17}$$

Next, we identify the damaged propellers using our initial estimate d_i within a 5% threshold as

$$d_i = \begin{cases} 0.0 & \text{if } k_{f_{\text{mis}}} \leq 0.05 \\ k_{f_{\text{model}}} \cdot \frac{\omega_i^2}{\omega_{L1i}^2} & \text{if } k_{f_{\text{mis}}} > 0.05 \end{cases}. \tag{18}$$

This threshold is set as undamaged propellers are likely to have negative damage due to torque mismatch. However a negative damage is physically impossible to exist. This means all undamaged propellers will have negative or close to zero damage. A detailed explanation of this concept is discussed in Section V.

We then formulate our estimate as a quadratic programming problem. Let Let $\mathbf{k} = \begin{bmatrix} k_{f_{1\text{real}}} & k_{f_{2\text{real}}} & k_{f_{3\text{real}}} & k_{f_{4\text{real}}} \end{bmatrix}^\top$ represent the true thrust coefficients we wish to solve for and $\mathbf{d} \in \mathbb{R}^{4\times 1}$ is a vector where each component d_i is obtained from eq. (18). The goal of our optimization is to find a solution that matches the propeller motor dynamics while also being close to our initial estimate described in eq. (18) without torque mismatch as

$$\begin{aligned} \min_{\mathbf{k}} \quad & (\mathbf{k} - \mathbf{d})^\top(\mathbf{k} - \mathbf{d}), \\ \text{s.t.} \quad & \mathbf{A}\mathbf{k} = \mathbf{b}. \end{aligned} \tag{19}$$

Here, $\mathbf{A}$ and $\mathbf{b}$ are defined as

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \omega_{L1_1}^2 & \omega_{L1_2}^2 & \omega_{L1_3}^2 & \omega_{L1_4}^2 \\ d_x\omega_{L1_1}^2 & d_x\omega_{L1_2}^2 & -d_x\omega_{L1_3}^2 & -d_x\omega_{L1_4}^2 \\ -d_y\omega_{L1_1}^2 & d_y\omega_{L1_2}^2 & d_y\omega_{L1_3}^2 & -d_y\omega_{L1_4}^2 \end{bmatrix}, \\ \mathbf{b} &= \begin{bmatrix} f & M_1 & M_2 \end{bmatrix}^\top, \end{aligned} \tag{20}$$

where $\mathbf{b}$ represents the desired force and moments of the quadrotor commanded by the nominal control without L1 and $\boldsymbol{\omega}_{\text{L1}} = \begin{bmatrix} \omega_{L1_1} & \omega_{L1_2} & \omega_{L1_3} & \omega_{L1_4} \end{bmatrix}^\top$ are the RPMs from the combination of L1 and nominal geometric system, d_x and d_y represent the distance from the propeller to the center of mass in the body frame projected on axes $\mathbf{b}_1$ and $\mathbf{b}_2$ respectively. To prevent the torque effect from L1 control we removed the fourth row of our constraints along with M_3 (the torque components) from our optimization. In addition, torque coefficients are difficult to observe and calculate without inducing yaw movement in the quadrotor. The L1 controller drives the error to zero between desired and executed action through the use of supplemental control action [21]. We construct our equality constraint in eq. (20) to represent this behaviour where the real thrust coefficients must be consistent with the supplemental action that generates the desired action.

The optimum based on null-space projection is

$$\mathbf{k} = \mathbf{A}^\top \left(\mathbf{A}\mathbf{A}^\top\right)^{-1} (\mathbf{b} - \mathbf{A}\mathbf{d}) + \mathbf{d}. \tag{21}$$

Finally, we convert $\mathbf{k}$ to percentage form using eq. (15).

TABLE I: Tracking RMSE (in meters) where $k_{f_{\text{mis}}}$ represents the thrust loss. FT represents the fault-tolerant control. L1 Off does not have data for $k_{f_{mis}} = 60\%$ because the aerial robot is unstable outside of the hover condition.

Duration [s]	Axis	$k_{f_{\text{mis}}} = 0.0$		$k_{f_{\text{mis}}} = 20\%$		$k_{f_{\text{mis}}} = 40\%$		$k_{f_{\text{mis}}} = 60\%$		FT
Max Speed [m/s]		L1 OFF	L1 ON	L1 OFF	L1 ON	L1 OFF	L1 ON	L1 OFF	L1 ON	L1 off
12 [0.5]	x	0.042	0.031	0.072	0.017	0.127	0.020	-	0.207	0.041
	y	0.092	0.070	0.233	0.062	0.453	0.030	-	0.358	0.044
	z	0.017	0.002	0.050	0.002	0.082	0.003	-	0.005	0.011
8 [0.8]	x	0.040	0.037	0.067	0.031	0.16	0.032	-	0.208	0.075
	y	0.10	0.064	0.222	0.064	0.42	0.048	-	0.384	0.059
	z	0.022	0.002	0.051	0.004	0.09	0.003	-	0.007	0.020
5 [1.1]	x	0.051	0.045	0.079	0.060	0.181	0.046	-	0.204	0.093
	y	0.112	0.079	0.249	0.072	0.406	0.047	-	0.406	0.109
	z	0.025	0.003	0.050	0.005	0.090	0.003	-	0.009	0.019

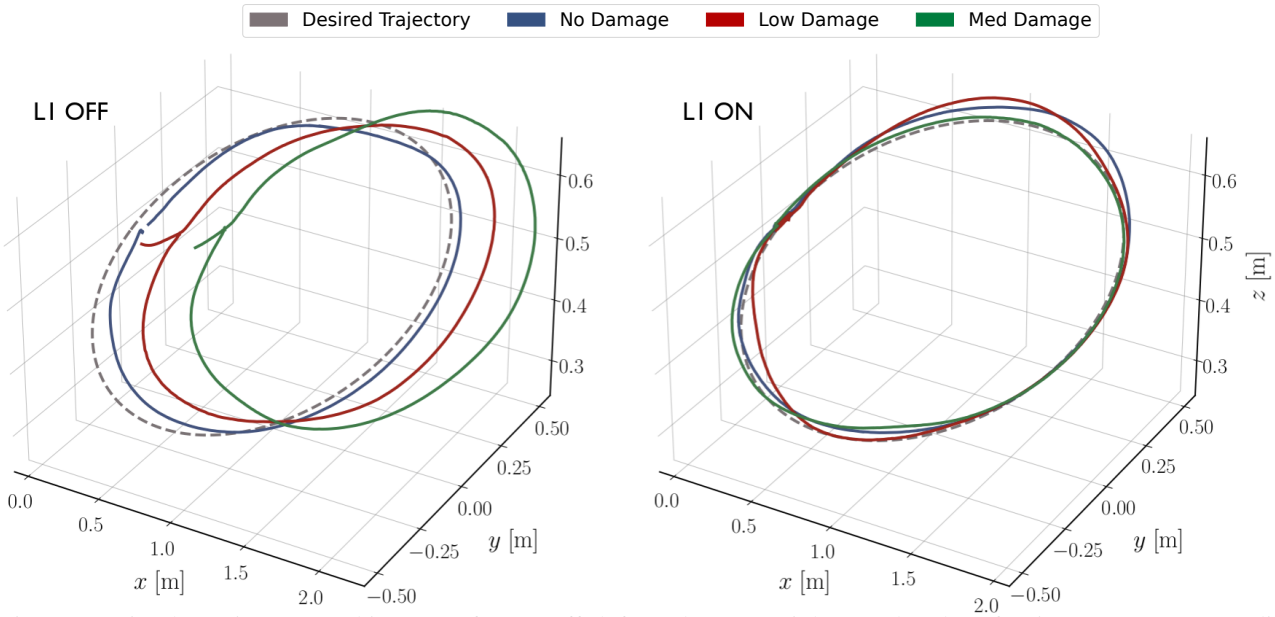

Fig. 4: 3D circular trajectory tracking trace for L1 Off (left) and L1 On (right). Each color of trajectory represents a different percentage of propeller damage as stated in the legend. The plot slows that the position drift grows as the kf mismatch gets larger. The plot on the right shows that the L1 adaptive algorithm is able to compensate for such propeller damages.

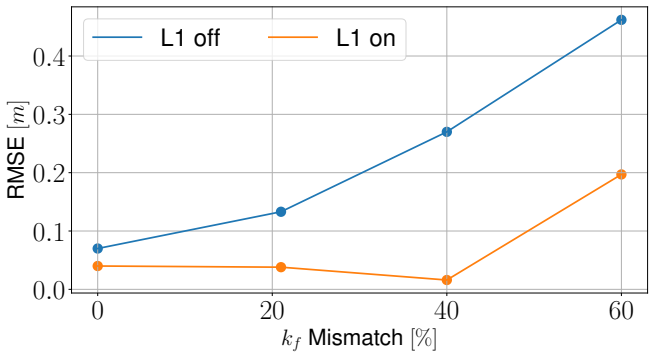

Fig. 5: RMSE averaged over 3 axes vs Propeller Damage at hover with various levels of propeller damage.

E. Fault-Tolerant Transition

We leverage our propeller damage estimate, eq. (21) to determine when to switch to fault-tolerant control. Our fault-tolerant control is implemented based on our previous work [13] where the approach is only tested in simulation. From the qualitative results with damaged propellers, we set a threshold at 50% damage. Based on our vehicle's thrust-to-weight ratio of 2.5 : 1, we experimentally verify that 50% damage provides a safety margin within the limit. A rapid response is necessary because a quadrotor could fall several meters if the estimation has a slow rise time. This is achieved by careful tuning of our LPF to balance denoising and response time. Once the threshold is met, the fault-tolerant control is transitioned from the base per our control design Fig. 3.

IV. Results

We conduct our experiments in a flying space of $10 \times 6 \times 4$ m^3 at the Agile Robotics and Perception Lab (ARPL) at New York University to validate the proposed approaches. The environment is equipped with a Vicon motion capture system which provides accurate pose estimates at 100 Hz. This is fused with IMU measurements through an Unscented Kalman Filter to provide state estimates at 500 Hz. The robot, based on our previous work [23], is equipped with

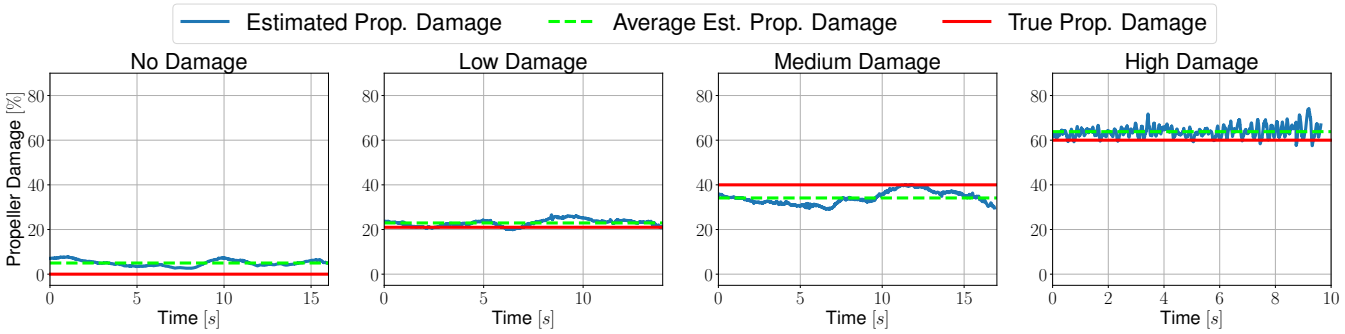

Fig. 6: A comparison between estimated damage and true damage with a real-world MAV during a 1.1 $\mathrm{m/s}$ flight. Propeller damage is estimated using eq. (17). True Propeller damage is solved by measuring the coefficient on a thrust bench.

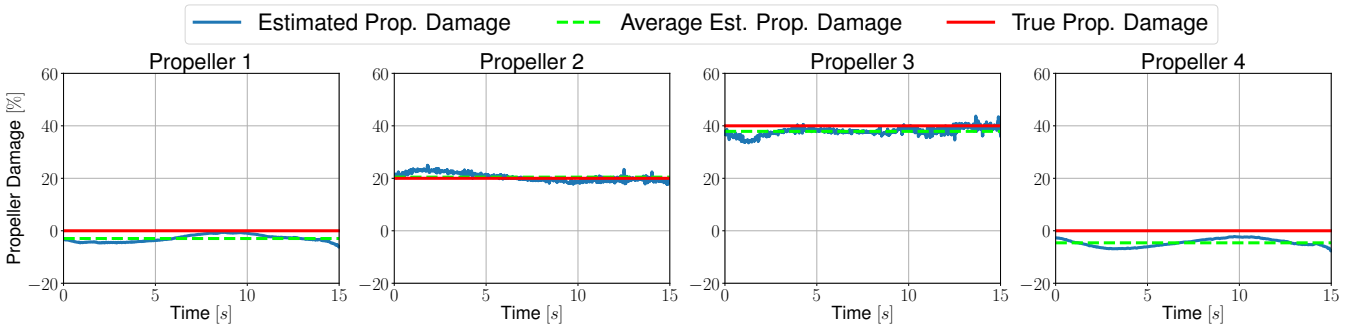

Fig. 7: Simultaneous propeller damage (propellers 2 and 3) for MAV moving $0.5\mathrm{m/s}$ in a circle

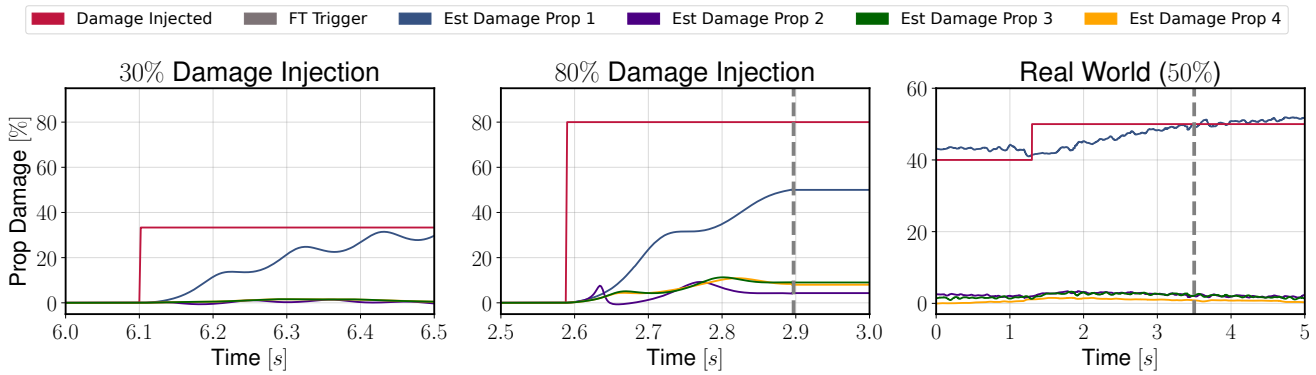

Fig. 8: Damage injected to propeller 1. Fault-tolerant transition threshold is set to 50%. Real-world experiment starts with 40% damage and transitions when 50% damage is injected. These experiments correspond to each row in Fig. (9).

a VOXL®2 ModalAI™ and four brushless motors and modified to obtain a 2.5 to 1 thrust to weight ratio with total a weight of 700 g. We classify the 4 levels of propeller damage based on $k_{f_{\text{mis}}}$: No damage refers to $k_{f_{\text{mis}}} = 0\%$. Low damage refers to $k_{f_{\text{mis}}} = 20\%$. Medium Damage refers to $k_{f_{\text{mis}}} = 40\%$. High damage refers to $k_{f_{\text{mis}}} = 60\%$.

A. L1 Performance Study

In order to evaluate the overall performance, the quadrotor is commanded to follow an ellipse of radius 1 m in x, 0.6 m in y, and 0.1 m in z for each case of propeller damage as well as a fully failed single propeller case using fault-tolerant control. The trajectory is completed with and without L1 adaptive control. This ellipse is flown at three different speeds: 12, 8, and 5 s periods. As shown in Table I, we observe that the RMSE reduces when the L1 adaptation is switched on in all cases and has lower tracking error than fault-tolerant for propeller damages below 40%. Furthermore, in Fig. 4, we depict the 8 s circular trajectory in 3D space showing that L1 adaptation compensates better to various propeller damages than with no L1 adaptation. We also see the position error grows in the L1 Off case as the severity of propeller damage increases while L1 maintains accurate tracking performance despite the degree of propeller damage. We also observe based on Fig. 5 that when the damage reaches greater than 40%, the error starts becoming too prominent for the adaptive control to negate. Therefore, it would be preferable to switch to a fault-tolerant method.

B. Propeller Damage Estimation

The L1 adaptive controller has demonstrated suitability and robustness to propeller damage compensation. In this section, we show that eq. (15) is an accurate estimate of the propeller damage ($k_{f_{\text{mis}}}$) even when the aerial robot is

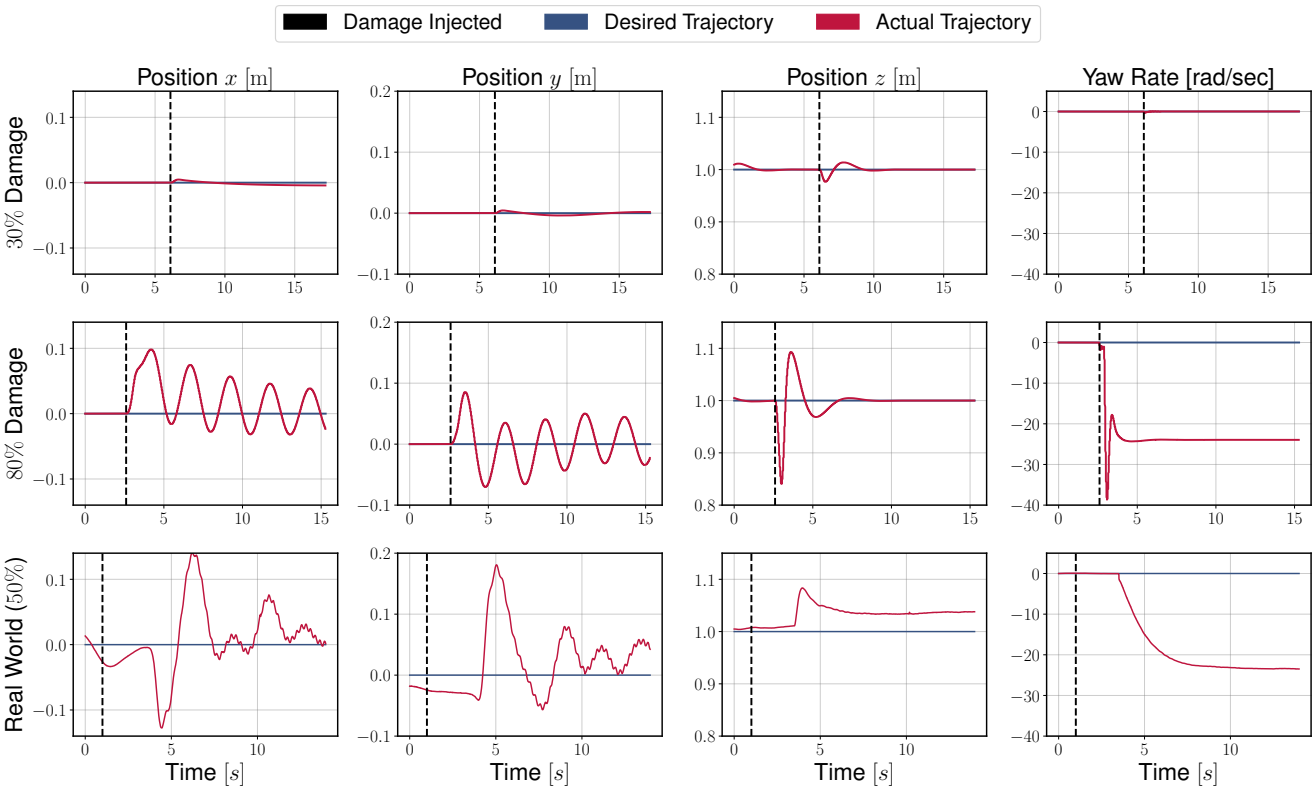

Fig. 9: Position tracking performance after propeller damage is injected. Each row corresponds to the propeller damage estimate results in Fig. 8. Top row: 30% prop damage is injected so no fault-tolerant control is activated. Middle row: 80% is injected triggering fault-tolerant control. Bottom row: 50% is injected in real world flight triggering fault control.

in motion. In Fig. 6, we report the estimates of a single damaged propeller while the quadrotor is executing an ellipse trajectory with a max speed of 1.1 m/s. This method is able to estimate the error with an error range of 4%. In terms of propeller coefficient 4% is a small absolute difference of $3.8 \cdot 10^{-10}$ or 8.3 g of thrust difference on a 700 g drone. Finally, we estimate dual propeller damages simultaneously in Fig. 7. When conducting simultaneous damage estimates, the damaged propellers, specifically 2 and 3, align closely with the true propeller damage.

C. Fault-Tolerant Transition

We study the rise-time and effectiveness of our transitioning mechanism by artificially injecting a fault mid-flight. This fault is produced by corrupting a motor input to spin slower. In simulation, we test this mechanism with a 30% and 80% fault. For real-world experiments we inject a 50% fault. Fig. 8 shows the estimation results of propeller damage. When a 30% damage is injected, the fault-tolerant control is not activated and only minor disturbances are observed. However, when an 80% damage fault occurs, the damage estimation of the affected propeller rises to 50% in 0.3 s, triggering the transition to fault-tolerant control. When the fault-tolerant controller is activated (i.e., 80% damage case), we turn off the estimate and simply extended the last value. For the real-world experiment in Fig. 8, we see a slower response time because the real-world has more noise and the low pass filter requires an increased strength. Fig. 9 shows the corresponding position tracking performances for all cases. We see that the transition to fault-tolerant control results in minor oscillations (up to 10 cm) in the x and y positions for both simulation and real-world, but is stable.

V. Discussion

The presented results show the benefit of the proposed adaptive control scheme for damage estimation and compensation. However, L1 adaptation can only correct a disturbance if the actions are within the actuator constraints [21]. Therefore a switching threshold for adaptive to fault-tolerant control should reflect this constraint. For example, our system has a 2.5 : 1 thrust to weight ratio which enables flight adaptive control with up to 60% damage without switching to fault-tolerant control. We choose to use a switching threshold of 50% to allow some safety margin. For any quadrotor capable of fault-tolerant flight, we assume a 2 : 1 thrust to weight ratio minimum is required. Therefore, a conservative switching threshold of 40% damage can be in general considered. Research has shown that L1 adaptive control is effective on a variety of quadrotors from 27g Crazyfly [24], 70g Parrot Mambo [15], to a heavy weight 700g shown in our work along with theoretical guarantees [21]. Our propeller damage estimation method is based on the steady state adaptive properties of L1 [21], making it equally generalizable to other platforms.

One of the current limitations of our approach is that we can only consider single or dual propeller damages

cases. Damaged propellers in a near hovering state will spin faster, and undamaged propellers will spin slower or remain at nominal speeds. A reverse effect in hover (lower RPMs of damaged propellers) is highly unlikely, so using the 5% threshold represents a conservative method to estimate propeller damage. Imagine a single damaged propeller that is spinning clockwise. The damaged propeller must spin faster to match the original thrust. However, this produces less torque than the original undamaged propeller creating a yaw acceleration. In order to counteract this yaw acceleration, the two counterclockwise propellers will spin slower. The opposite undamaged clockwise propeller remains consistent with slight deviations. The same principle holds for the case of two damaged propellers. In the adjacent case, a counterclockwise and clockwise torque both decrease. This results in a cancelling effect where the more damaged propeller's opposite will slow down a constant amount. In the diagonal case, while a similar thrust can be achieved, the alternative spinning undamaged propellers must reduce their corresponding speeds to counteract the yaw acceleration. It is important to note that these actions are handled automatically by our adaption law eq. (14), and we only use our optimization method for calculating the damage estimate.

VI. Conclusion

In this paper, we presented an adaptive inference and control strategy to ensure safe quadrotor flight in case of propeller damage. Our approach incorporates an L1 adaptive control technique in conjunction with a fault-tolerant mode. Furthermore, we propose a method to quantify propeller damage using L1 adaptation. Diagnosing the state of the system is vital for safety critical applications as it can help recognize and resolve faults quickly. The proposed method can estimate propeller damage with a significant degree of accuracy. We experimentally study the limits of the L1 adaptive control scheme in this context and analyze at what level of damage severity it is advisable to transition to the fault-tolerant mode.

Future works will attempt recovery and estimation in the middle of more aggressive maneuvers [25]. We would also explore alternative methods to reduce the delay on our low pass filter in real-world tests. Additionally, we would like to see if the proposed approach can be extended to other thrust curve models, especially for flight envelopes where the quadratic relation is no longer a good approximation.

References

[1] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Computer Communications*, vol. 156, pp. 1–10, 2020.

[2] J.-G. Ye, H.-T. Chen, and W.-J. Tsai, "Panorama generation based on aerial images," in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2018, pp. 1–6.

[3] G. K. Fourlas and G. C. Karras, "A survey on fault diagnosis and fault-tolerant control methods for unmanned aerial vehicles," *Machines*, vol. 9, no. 9, p. 197, 2021.

[4] P. Lu and E.-J. van Kampen, "Active fault-tolerant control for quadrotors subjected to a complete rotor failure," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4698–4703.

[5] B. Ghalamchi and M. Mueller, "Vibration-based propeller fault diagnosis for multicopters," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 1041–1047.

[6] S. P. Madruga, T. P. Nascimento, F. Holzapfel, and A. M. N. Lima, "Estimating the loss of effectiveness of uav actuators in the presence of aerodynamic effects," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1335–1342, 2023.

[7] R. C. Avram, X. Zhang, and J. Muse, "Quadrotor actuator fault diagnosis and accommodation using nonlinear adaptive estimators," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 2219–2226, 2017.

[8] A. S. Sanca, P. J. Alsina, and J. d. J. F. Cerqueira, "Dynamic modelling of a quadrotor aerial vehicle with nonlinear inputs," in *IEEE Latin American Robotic Symposium*, 2008, pp. 143–148.

[9] S. S. Alex, A. E. Daniel, and B. Jayanand, "Reduced order extended kalman filter for state estimation of brushless dc motor," in *Sixth International Symposium on Embedded Computing and System Design (ISED)*, 2016, pp. 239–244.

[10] A. G. Rot, A. Hasan, and P. Manoonpong, "Robust actuator fault diagnosis algorithm for autonomous hexacopter uavs," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 682–687, 2020, 21st IFAC World Congress.

[11] Y. Zhong, Y. Zhang, W. Zhang, J. Zuo, and H. Zhan, "Robust actuator fault detection and diagnosis for a quadrotor uav with external disturbances," *IEEE Access*, vol. 6, pp. 48 169–48 180, 2018.

[12] A. Abbaspour, S. Mokhtari, A. Sargolzaei, and K. K. Yen, "A survey on active fault-tolerant control systems," *Electronics*, vol. 9, p. 1513, 2020.

[13] J. Yeom, G. Li, and G. Loianno, "Geometric fault-tolerant control of quadrotors in case of rotor failures: An attitude based comparative study," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[14] S. Sun, X. Wang, Q. Chu, and C. d. Visser, "Incremental nonlinear fault-tolerant control of a quadrotor with complete loss of two opposing rotors," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 116–130, 2021.

[15] Z. Wu, S. Cheng, K. A. Ackerman, A. Gahlawat, A. Lakshmanan, P. Zhao, and N. Hovakimyan, "L1 adaptive augmentation for geometric tracking control of quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1329–1336.

[16] A. Saviolo, J. Frey, A. Rathod, M. Diehl, and G. Loianno, "Active learning of discrete-time dynamics for uncertainty-aware model predictive control," *arXiv preprint arXiv:2210.12583*, 2022.

[17] F. Crocetti, J. Mao, A. Saviolo, G. Costante, and G. Loianno, "Gapt: Gaussian process toolkit for online regression with application to learning quadrotor dynamics," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 308–11 314.

[18] A. Loquercio, A. Saviolo, and D. Scaramuzza, "Autotune: Controller tuning for high-speed flight," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4432–4439, 2022.

[19] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 45–52.

[20] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5420–5425.

[21] M. Khammash, V. Vittal, and C. Pawloski, "Analysis of control performance for stability robustness of power systems," *IEEE Transactions on Power Systems*, vol. 9, no. 4, pp. 1861–1867, 1994.

[22] M. Watterson and V. Kumar, "Control of quadrotors using the hopf fibration on so(3)," in *Robotics Research: The 18th International Symposium ISRR*. Springer, 2019, pp. 199–215.

[23] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, April 2017.

[24] K. Huang, R. Rana, G. Shi, A. Spitzer, and B. Boots, "Datt: Deep adaptive trajectory tracking for quadrotor control," in *7th Annual Conference on Robot Learning*, 2023.

[25] J. Mao, S. Nogar, C. M. Kroninger, and G. Loianno, "Robust active visual perching with quadrotors on inclined surfaces," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1836–1852, 2023.