

# 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. In this paper, we propose a novel adaptive control scheme capable of detecting and compensating for multi-rotor propeller damages, ensuring safe and robust flight performances. Our solution combines an L1 adaptive controller with an optimization routine 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 the ability of our solution to adapt and compensate onboard in real time on a quadrotor for damages even when multiple propellers are damaged.

**Index Terms**—Aerial Systems: Mechanics and Control; Aerial Systems: Applications

SUPPLEMENTARY MATERIAL

**Video:** <https://youtu.be/eIV00-tkPs0>

## I. INTRODUCTION

**M**ICRO Aerial Vehicles (MAVs) such as quadrotors are becoming ubiquitous in applications such as search and rescue and aerial photography [1]. 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. 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 introduces an adaptive control scheme for quadrotors to autonomously infer and adjust flight behavior in response to propeller damages or failures.

\*These authors contributed equally and are listed in alphabetic order.

Manuscript received: October 19, 2023; Accepted February 28, 2024.

This paper was recommended for publication by Editor Cesar Cadena and Editor Lucia Pallottino upon evaluation of reviewers' comments. This work was supported by the DEVCOM ARL grant DCIST CRA W911NF-17-2-0181, NSF CAREER Award 2145277, DARPA YFA Grant D22AP00156-00, Qualcomm Research, Nokia, and NYU Wireless.

The authors are with the New York University, Tandon School of Engineering, Brooklyn, NY 11201, USA. email: {jm7752, jennifer.yeom, surajkiron, loiannog}@nyu.edu. (Jeffrey Mao and Jennifer Yeom are co-first authors.) (Corresponding author: Jeffrey Mao.)

Digital Object Identifier (DOI): see top of this page

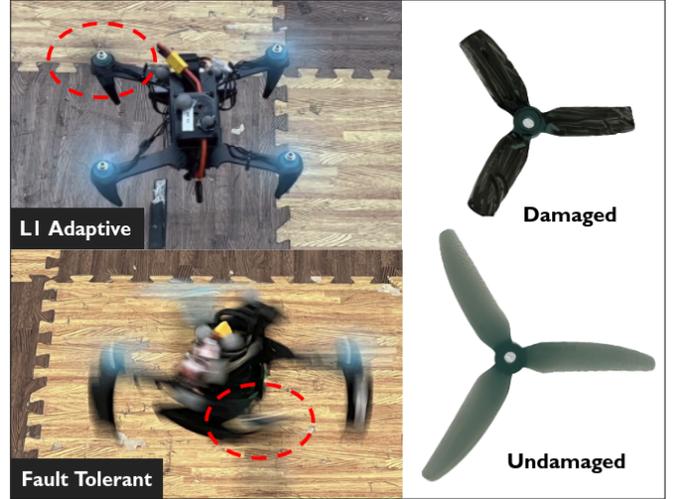


Figure 1: L1 adaptive and fault-tolerant control with one damaged propeller (circled in red dash). The fault-tolerant controlled quadrotor is forced to spin (over 1000 deg/sec).

This work makes the following contributions. First, we propose a propeller damage estimation technique by combining an L1 Adaptive algorithm and an optimization routine with no need for direct RPM (Rotations Per Minute) measurements or estimation of angular accelerations, making it suitable for low-cost systems. Second, we propose a holistic control scheme coupling the L1 adaptive and fault-tolerant controller modules for autonomous transition from L1 adaptation to the fault-tolerant mode in situations where adaptation alone proves to be insufficient for ensuring safe and effective flight. This approach enhances the system's capability to manage the full range of propeller damage. Finally, we demonstrate the performance of the damage estimation and controller design in extensive real-world experiments and compare our estimation method to using a standard PID (Proportional Integral Derivative) controller. This is the first work to tackle precise estimation and compensation of propeller damage rather than considering a reduction of effectiveness of a motor-propeller pair. Challenges include unequal loss between thrust and torque coefficients along with additional noise on the system from damaged propellers as well as the unavailability of direct RPM measurements.

## II. RELATED WORKS

**Damage Estimation.** Research in fault estimation covers a variety of vehicles detailed in this survey [2] 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 [3] or

sensor noise analysis [4]. The approach proposed in [3] successfully estimates the loss of effectiveness of a single motor but is only tested in simulation, whereas [4] uses accelerometer noise to estimate propeller damage, but is only tested on a minuscule amount of propeller damage.

Other works rely on state estimation to accurately measure the damage [5], [6]. 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. They typically 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 coefficients scale with the propeller radius,  $r$ , as a function of  $r^4$  and  $r^5$  respectively [7]. This increases the complexity of our propeller damage compared to the problems in [5], [6], [8]. Typically a reduced state Kalman Filter [5], [6], [9] is used in either a single [5], [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 up to 60% motor damage in [11]. However, it is only tested in simulation for motor 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. Uniquely, [8] implements a Kalman filter method to detect a complete loss of a propeller and is able to detect a fault within 0.1 s of occurrence. However, this method requires accurate RPM measurements and cannot adapt to propeller damage.

**Damage Compensation.** 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 [5], [6] 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 [3], [12] but sacrifices yaw control.

A few adaptive techniques are Incremental Nonlinear Dynamic Inversion (INDI) [13], dual loop disturbance observers [14], and L1 [15]–[17]. These works provide additional control actions that augment the capabilities of a baseline controller to reject some actuator damages or disturbances. However, none of the above works provides an estimate to judge when propeller damage is too severe for adaptation as shown in the proposed work. Other methods exist for adaptation [18]–[20], through model identification or automatic gain tuning, but cannot adapt to rapid changes. In this work, we design an L1 adaptive technique to compensate and accurately estimate propeller damage. Unlike INDI [13] 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 (IMU).

Existing fault-tolerant control strategies are adept at ensuring robust control even in the event of single or multiple rotor failures [2], [13], [21], [22]. However, these solutions require identifying the faulty propeller, and sacrificing yaw

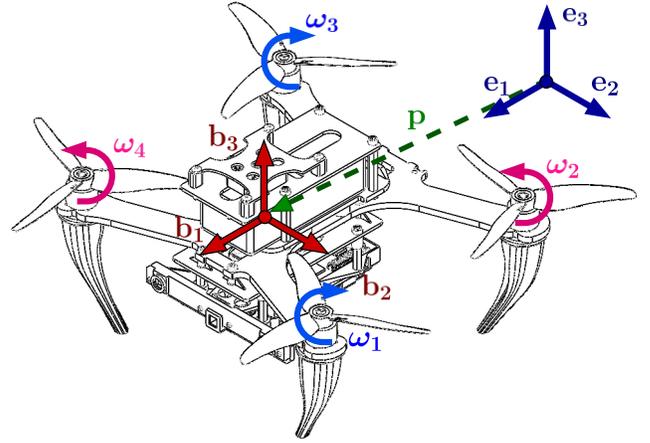


Figure 2: MAV model with inertial and body frame definitions.

control which results in a rapid yaw spin as depicted in Fig. 1 making them unsuitable for small to medium damages. Our work introduces a holistic 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 allowing effective operations for safe recovery and navigation.

### 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  $[e_1 \ e_2 \ e_3]$  and the body fixed frame as  $[b_1 \ b_2 \ b_3]$ . 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  $b_1$  is aligned to the heading and the third axis,  $b_3$  aligns with the thrust vector of the vehicle. The dynamics of the system can be written as

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

where  $\mathbf{p} \in \mathbb{R}^3$ , and  $\mathbf{v} \in \mathbb{R}^3$  are the position and velocity in the world 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,  $\mathbf{M} = [M_1 \ M_2 \ M_3]^\top$  are the moments around the three body frame axes, and  $\mathbf{J} \in \mathbb{R}^{3 \times 3}$  is the robot inertia matrix.

#### B. Control Design

We adapt the geometric controller derived in [23] for the possibility of propeller damage, or fully losing one or two opposing rotors in flight. An outer controller solves for the desired attitude  $\mathbf{R}$ , thrust  $f$ , and angular velocities  $\boldsymbol{\Omega}$ . The inner controller solves for the moments required to control the



$(\mathbf{v}_m, \boldsymbol{\Omega}_m)$ . Let  $k$  denote the discrete time step of the system. We set a diagonal gain defined by the user  $\mathbf{K} \in \mathbb{R}^{6 \times 6}$  to generate a specific gain,  $\mathbf{A}$ , based on the time step,  $dt$ , between  $k$  and  $k + 1$  time instants with

$$\mathbf{A} = (\exp(\mathbf{K} \cdot dt - \mathbf{I})^{-1} \mathbf{K}) \exp(\mathbf{K} \cdot dt),$$

$$\boldsymbol{\sigma}[k] = \begin{bmatrix} m\mathbf{R}^\top & 0 \\ 0 & \mathbf{J} \end{bmatrix} \mathbf{A} \left( \begin{bmatrix} \mathbf{v}_m[k] \\ \boldsymbol{\Omega}_m[k] \end{bmatrix} - \begin{bmatrix} \mathbf{v}_p[k] \\ \boldsymbol{\Omega}_p[k] \end{bmatrix} \right). \quad (11)$$

We solve the additional action  $\boldsymbol{\mu}_{L1} = [f_{L1} \quad \mathbf{M}_{L1}^\top]^\top$  from our disturbance  $\boldsymbol{\sigma}$ . The moment disturbance estimates  $\boldsymbol{\sigma}_{4:6}$  and disturbance 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 implement an exponential Low Pass Filter (LPF) to mitigate the noise present in the state estimation, achieving smoother results:  $\boldsymbol{\mu}_{L1} = -\text{LPF}(\boldsymbol{\sigma}_{3:6})$ .  $f_{L1}$  and  $\mathbf{M}_{L1}$  are added to eq. (5) and eq. (10) respectively and reflected in Fig. (3).

2) *State Predictor*: Separately, once we solve for the action  $\boldsymbol{\mu}_{L1}$  and external disturbances  $\boldsymbol{\sigma}$ , we solve for the next predicted velocities of our system,  $(\mathbf{v}_p[k], \boldsymbol{\Omega}_p[k]) \rightarrow (\mathbf{v}_p[k+1], \boldsymbol{\Omega}_p[k+1])$ . First, we solve for the linear and angular acceleration  $\dot{\mathbf{v}}_p[k]$  and  $\dot{\boldsymbol{\Omega}}_p[k]$  as

$$\dot{\mathbf{v}}_p[k] = \underbrace{\frac{(f[k] + f_{L1}[k])}{m} \mathbf{R} \mathbf{e}_3}_{\text{actuation}} + \underbrace{\frac{1}{m} \mathbf{R} \boldsymbol{\sigma}_{1:3}[k]}_{\text{disturbance}} + \underbrace{g \mathbf{e}_3}_{\text{gravity}}, \quad (12)$$

$$\dot{\boldsymbol{\Omega}}_p[k] = \mathbf{J}^{-1} \left( \underbrace{\mathbf{M}[k] + \mathbf{M}_{L1}[k]}_{\text{actuation}} + \underbrace{\boldsymbol{\sigma}_{4:6}[k]}_{\text{disturbance}} - \underbrace{\boldsymbol{\Omega}_m[k] \times \mathbf{J} \boldsymbol{\Omega}_m[k]}_{\text{cross}} \right). \quad (13)$$

Finally, we integrate over a time step  $dt$  with a weighted average of the measurement and the predicted velocities. A weighted average is used to filter the noise inherent in the measured velocity

$$\begin{bmatrix} \mathbf{v}_p[k+1] \\ \boldsymbol{\Omega}_p[k+1] \end{bmatrix} = (1 - \lambda) \begin{bmatrix} \mathbf{v}_p[k] \\ \boldsymbol{\Omega}_p[k] \end{bmatrix} + \lambda \begin{bmatrix} \mathbf{v}_m[k] \\ \boldsymbol{\Omega}_m[k] \end{bmatrix} + dt \begin{bmatrix} \dot{\mathbf{v}}_p[k] \\ \dot{\boldsymbol{\Omega}}_p[k] \end{bmatrix}. \quad (14)$$

The weighted average term is  $\lambda = \mathbf{K} \cdot dt$ . This is predicted value is looped backed into our L1 control law eq. (11) in the next time step  $k + 1$

#### 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  $i^{\text{th}}$  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  $\omega_i$  is the motor speed of the  $i^{\text{th}}$  propeller. 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 the following mismatch index

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

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

[7]. 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  $\omega_i$  now be the RPM for motor  $i$  given by the geometric controller without L1 compensation and  $\omega_{L1i}$  be the RPM for motor  $i$  with L1 compensation. Supposing that L1 adaptation has compensated for the propeller damage, we will have

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

Converting terms we have an initial guess

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

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

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

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

We then formulate our estimate as a quadratic programming problem. Let  $\mathbf{k} = [k_{f1_{\text{real}}} \quad k_{f2_{\text{real}}} \quad k_{f3_{\text{real}}} \quad k_{f4_{\text{real}}}]^\top$  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

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

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

$$\mathbf{A} = \begin{bmatrix} \omega_{L11}^2 & \omega_{L12}^2 & \omega_{L13}^2 & \omega_{L14}^2 \\ d_x \omega_{L11}^2 & d_x \omega_{L12}^2 & -d_x \omega_{L13}^2 & -d_x \omega_{L14}^2 \\ -d_y \omega_{L11}^2 & d_y \omega_{L12}^2 & d_y \omega_{L13}^2 & -d_y \omega_{L14}^2 \end{bmatrix}, \quad (20)$$

$$\mathbf{b} = [f \quad M_1 \quad M_2]^\top,$$

where  $\mathbf{b}$  represents the desired force and moments of the quadrotor commanded by the nominal control without L1 and  $\boldsymbol{\omega}_{L1} = [\omega_{L11} \quad \omega_{L12} \quad \omega_{L13} \quad \omega_{L14}]^\top$  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 remove the fourth

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  $k_{f_{\text{mis}}} = 60\%$  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
		L1 ON	L1 OFF	L1 ON	L1 OFF	L1 ON	L1 OFF	L1 ON	L1 OFF	L1 ON
12 [0.5]	$x$	<b>0.031</b>	0.042	<b>0.017</b>	0.072	<b>0.020</b>	0.127	<b>0.207</b>	-	0.041
	$y$	<b>0.070</b>	0.092	<b>0.062</b>	0.233	<b>0.030</b>	0.453	<b>0.358</b>	-	0.044
	$z$	<b>0.002</b>	0.017	<b>0.002</b>	0.050	<b>0.003</b>	0.082	<b>0.005</b>	-	0.011
8 [0.8]	$x$	<b>0.037</b>	0.040	<b>0.031</b>	0.067	<b>0.032</b>	0.16	<b>0.208</b>	-	0.075
	$y$	<b>0.064</b>	0.10	<b>0.064</b>	0.222	<b>0.048</b>	0.42	<b>0.384</b>	-	0.059
	$z$	<b>0.002</b>	0.022	<b>0.004</b>	0.051	<b>0.003</b>	0.09	<b>0.007</b>	-	0.020
5 [1.1]	$x$	<b>0.045</b>	0.052	<b>0.060</b>	0.079	<b>0.046</b>	0.181	<b>0.204</b>	-	0.093
	$y$	<b>0.079</b>	0.112	<b>0.072</b>	0.249	<b>0.047</b>	0.406	<b>0.406</b>	-	0.109
	$z$	<b>0.003</b>	0.025	<b>0.005</b>	0.050	<b>0.003</b>	0.090	<b>0.009</b>	-	0.019

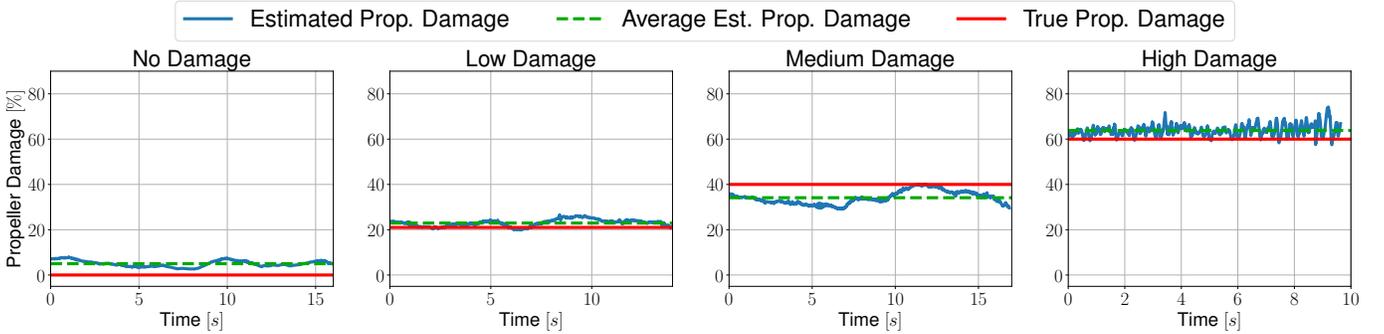


Figure 4: A comparison between estimated damage and true damage with a real-world MAV during a 1.1 m/s flight. Propeller damage is estimated using eq. (17). True Propeller damage is solved by measuring the coefficient on a thrust bench.

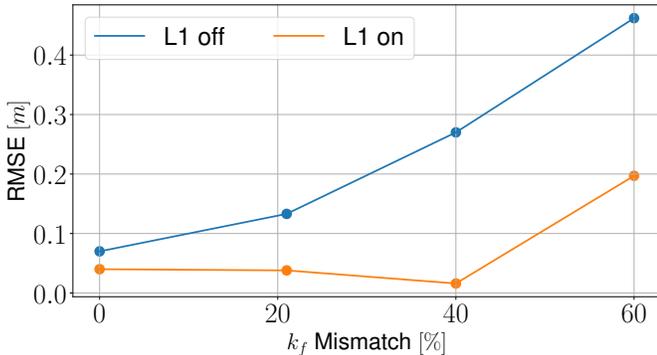


Figure 5: Tracking RMSE averaged over 3 axes vs Propeller Damage at hover with various levels of propeller damage.

row of our constraints along with  $M_3$  (the torque components) from our optimization. The L1 controller drives the error to zero between desired and executed action through the use of supplemental control action [24]. 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

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

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

### 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 [21] which was 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. Once the threshold is met, the system automatically transitions to the fault-tolerant control per our control design as shown in Fig. 3.

## IV. RESULTS

Our experiments are conducted in a flying space of  $10 \times 6 \times 4 \text{ m}^3$  at the Agile Robotics and Perception Lab (ARPL) at New York University. 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 [26], is equipped with a VOXL<sup>®</sup>2 ModalAI<sup>™</sup> 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 set the  $\mathbf{K}$  such that  $\lambda = \text{Diag}[0.4 \ 0.4 \ 0.4 \ 0.1 \ 0.1 \ 0.1]$ .

### A. L1 Performance Study

To evaluate the tracking 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 under fault-tolerant control with one disabled propeller. The trajectory is completed with and without L1 adaptive control. This ellipse is flown at three different speeds: 12, 8, and 5 s

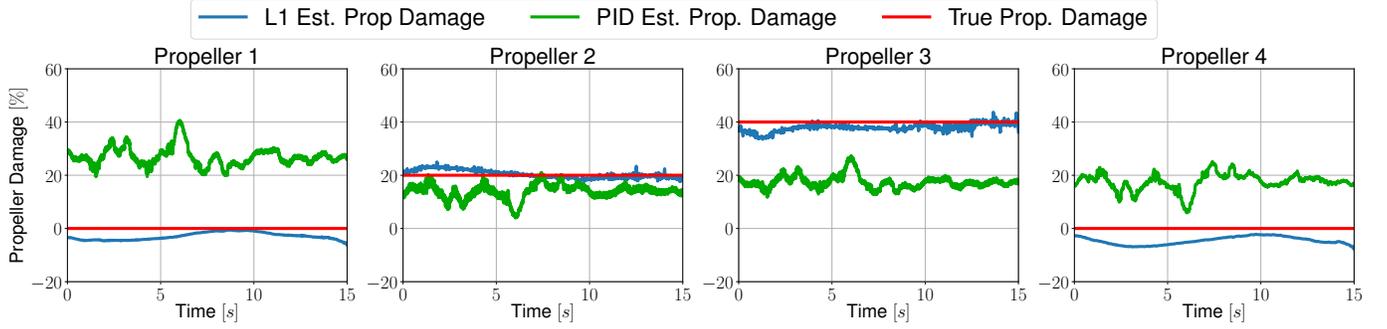


Figure 6: Simultaneous propeller damage (propellers 2 and 3) for MAV moving 0.5m/s in a circle. Both PID (green) and L1 (blue) approaches for propeller damage estimation are shown.

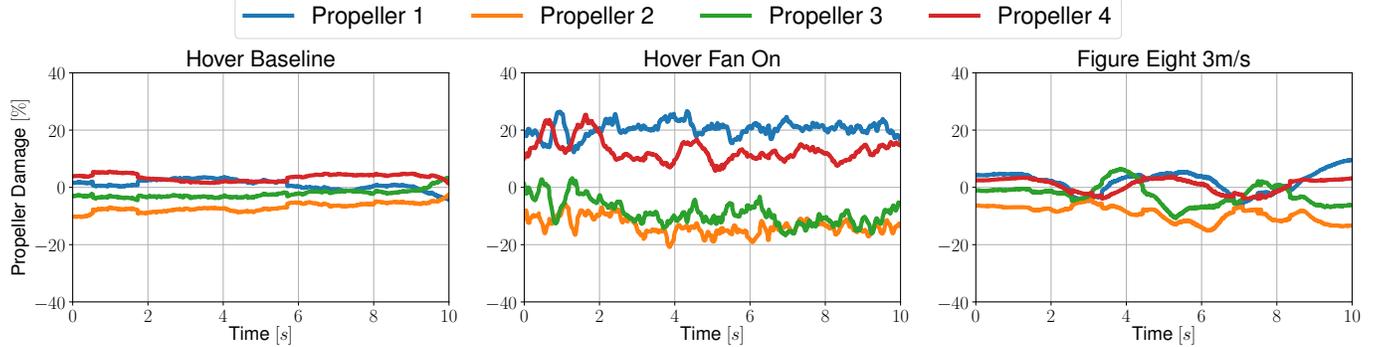


Figure 7: Propeller damage estimate of an undamaged vehicle under 3 different experimental conditions. Hover Baseline refers to a hover with no external forces. Figure Eight refers to flying a trajectory in the shape of an eight with a max velocity of 3 m/s. Hover fan on refers to stabilizing the vehicle under a fan blowing wind at 3 m/s

periods. We observe in Table I 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%. We also see in Fig. 5 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. However when the damage reaches greater than 40%, the error starts becoming too prominent for the adaptive control to negate. In these cases, transitioning to a fault-tolerant controller is preferable.

### B. Propeller Damage Estimation

The L1 adaptive controller has demonstrated suitability and robustness to propeller damage compensation. In this section, we show that Sec. III-D represents an accurate estimate of the propeller damage,  $k_{f_{mis}}$ , even when the aerial robot is in motion. To calculate the various levels of propeller damage, we use a thrust bench to calculate the true thrust coefficient of the damaged propeller,  $k_{f_{real}}$  and undamaged propeller,  $k_{f_{model}}$ , coefficient. We then use eq. (15) to calculate the damage,  $k_{f_{mis}}$ . We test 4 levels of propeller damage based on  $k_{f_{mis}}$ : No damage refers to  $k_{f_{mis}} = 0\%$ . Low damage refers to  $k_{f_{mis}} = 20\%$ . Medium Damage refers to  $k_{f_{mis}} = 40\%$ . High damage refers to  $k_{f_{mis}} = 60\%$ . In Fig. 4, we report the estimates of a single damaged propeller at the aforementioned levels 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 individual actuator forces, this represents 8.3 g of thrust on a 700 g drone. Next,

we estimate dual propeller damage simultaneously in Fig. 6. We compare two methods for damage estimation. First, we use L1 complimentary actions shown in blue. Next, we replace the L1 adaptive control with an integral term in a PID controller for complimentary action in green. We substitute  $\omega_{L1i}$  with the actions generated by an additional integral controller to estimate the propeller damage following the same inference and optimization procedure as Section III-D. The integral term is placed over the velocity expressed in the body frame. While the integral term can compensate some damage, it is unsuitable for damage estimation. The L1 complimentary actions on the other hand can be used to accurately estimate the propeller's damage.

Furthermore, we test the effects of medium speed flight (3 m/s) and external disturbances on our propeller damage estimation. In Fig. 7, we show the results of 3 experiments. First, we show a control case where the quadrotor is in hover without external disturbances and no damages. Second, we show the same vehicle in hover with 3 m/s wind disturbance. Third, we show a flight without external disturbances flying at 3 m/s. Ideally, the desired estimated damage should be 0%. Overall, we notice that the fan has a significant effect on damage estimation around 20%, but it is not enough to trigger a transition. Medium speed motions at 3 m/s instead cause minimal but noticeable deviations from around 0% to  $\pm 10\%$ .

### C. Fault-Tolerant Transition

We study the rise-time and effectiveness of our transitioning mechanism by artificially injecting a fault mid-flight. This fault

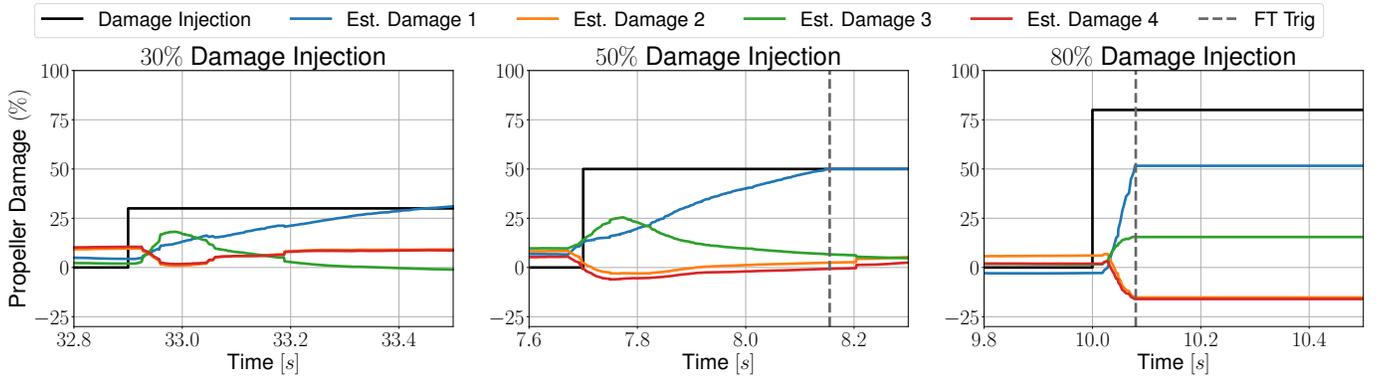


Figure 8: Real-world experiments with damage injected to propeller 1. Fault-tolerant transition threshold is set to 50%.

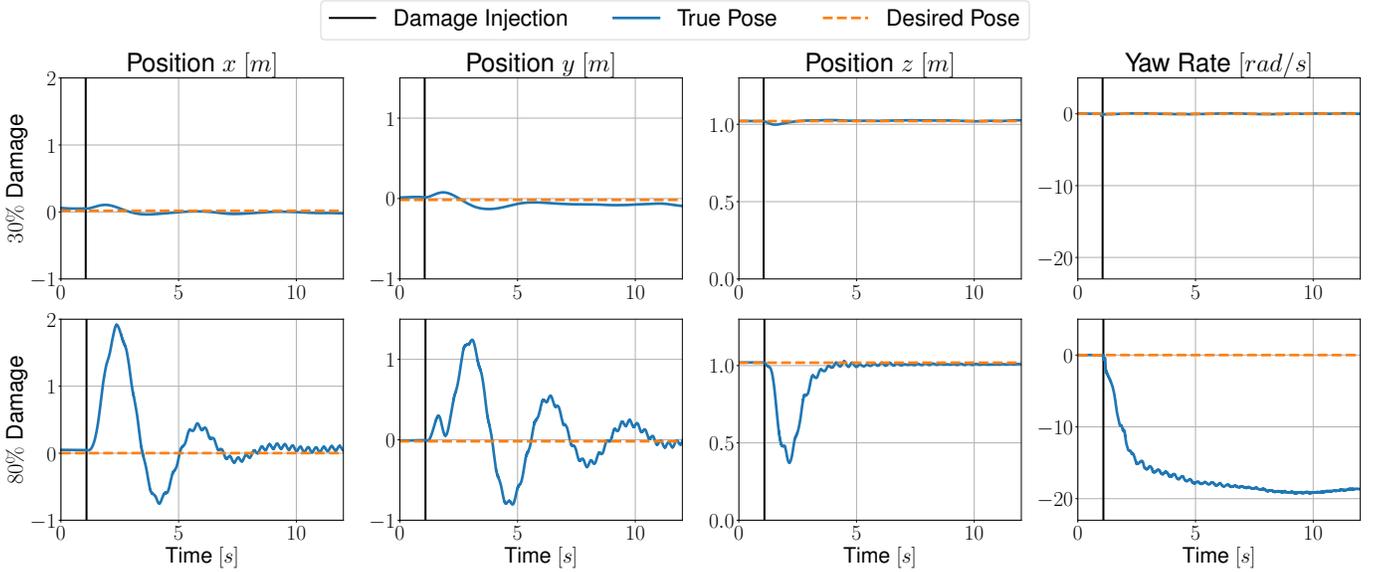


Figure 9: Position tracking performance after propeller damage is injected in real-world flight. The 80% damage case shows a high yaw rate because it triggers fault-tolerant control, causing a rapid spin unlike the 30% damage case.

is produced by corrupting a motor input to spin slower. In our experiments, we tested the reliability of our system given 40 trials by injecting 30%, 40%, and 50% damages in a real drone. 30% damage injection has a 100% success rate of not transitioning whereas the 50% case shows a 100% successful transitions. The 40% injection has a 87.5% success rate for not transitioning and a 12.5% false positive transition. Our method did not produce any false negative.

Additionally, we inject an 80% damage fault in a real-world experiment to show that our latency is low enough for practical uses. The damage estimation of the affected propeller rises to 50% in 0.15 s, triggering the transition to fault-tolerant control. Fig. 8 shows the damage estimation in real-world experiments with 30%, 50%, and 80% damage injections along with transition times when relevant. When the fault-tolerant controller is activated (i.e.,  $> 50\%$  damage cases), we stop the inference and estimation process. Fig. 9 shows the corresponding position tracking performances for the non transitioning 30% case and the transitioning 80% damage case. We see our methodology can autonomously transition and stabilize the drone despite the injection of heavy propeller damage. For visualization purposes, the last values are repeated instead of cut when the inference stops after the

fault tolerant control is triggered.

## 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 [24]. Therefore a switching threshold for adaptive to fault-tolerant control should reflect this constraint. 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, a 2 : 1 thrust to weight ratio minimum is required. Therefore, a conservative switching threshold of 40% damage can be considered. Research has shown that L1 adaptive control is effective on a variety of quadrotors from 27 g Crazyflie [27], 70 g Parrot Mambo [15], to our heavy weight platform 700 g along with theoretical guarantees [24]. Our propeller damage estimation method is based on the steady state adaptive properties of L1 [24], making it equally generalizable to other platforms.

One limitation of our approach is that we 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 detect 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. These actions are handled by our adaption law eq. (11), and our optimization method is only for damage estimation.

## 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. We propose a method to quantify propeller damage using L1 adaptation that allows smooth transitioning to fault tolerant control in the case of severe damage. The proposed method can effectively account for all range damages from low to severe levels.

Future works will attempt recovery and estimation during more aggressive maneuvers [28]. We would also like to see if the proposed approach can be extended to other thrust curve models, especially for flight envelopes where the quadratic relation is an inaccurate approximation.

## REFERENCES

- [1] M. Idrissi, M. Salami, and F. Annaz, "A review of quadrotor unmanned aerial vehicles: applications, architectural design and control algorithms," *Journal of Intelligent & Robotic Systems*, vol. 104, no. 2, p. 22, 2022.
- [2] G. K. Furlas 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.
- [3] 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.
- [4] B. Ghalamchi and M. Mueller, "Vibration-based propeller fault diagnosis for multicopters," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 1041–1047.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] B. A. S. van Schijndel, S. Sun, and C. C. de Visser, "Fast loss of effectiveness detection on a quadrotor using onboard sensors and a kalman estimation approach," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2023, pp. 1–8.
- [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] 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.
- [14] X. Yu, X. Zhou, K. Guo, J. Jia, L. Guo, and Y. Zhang, "Safety flight control for a quadrotor uav using differential flatness and dual-loop observers," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 12, pp. 13 326–13 336, 2022.
- [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] M. Mühlegg, P. Niermeyer, G. P. Falconi, and F. Holzapfel, "L1 fault tolerant adaptive control of a hexacopter with control degradation," in *2015 IEEE Conference on Control Applications (CCA)*, 2015, pp. 750–755.
- [17] M. Li, Z. Zuo, H. Liu, C. Liu, and B. Zhu, "Adaptive fault tolerant control for trajectory tracking of a quadrotor helicopter," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 12, pp. 3560–3569, 2018.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrotor despite the complete loss of one, two, or three propellers," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 45–52.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.