# Rotor Fault Detection and Isolation in Aerial Vehicles with Dozens of Rotors

Neeraj Gandhi[1], Jiawei Xu[2], David Saldaña[2], and Linh Thi Xuan Phan[1]

***Abstract*—Aerial vehicles with dozens of rotors are becoming increasingly common in important applications such as transportation and construction. One challenge with building such a system is to ensure that the system is robust against faults: as the number of rotors increases, the likelihood of a rotor failing during operation also increases; despite the spare thrust capacity provided by the redundant rotors, a rotor fault can significantly impact the motion and safety of the system.**

**This paper presents an efficient fault detection and isolation (FDI) method for aerial vehicles with a large number of rotors. Our approach relies on two key insights: First, the effect of a faulty rotor directly affects the tracking error in roll and in pitch. This property can be used to order our faulty rotor search space. Second, the error in either roll or pitch is related to both the distance from the (relevant) axis and the severity of a fault. With these observations, we can use *probe faults* to isolate faulty rotors. Evaluation results show that our technique can efficiently detect and isolate faults in multi-rotor aerial vehicles with up to 64 rotors (8× more rotors than in existing FDI work), and that it can help improve robustness. To the best of our knowledge, our FDI method is the first that scales to several dozens of rotors.**

## I. Introduction

Multirotor aerial vehicles (MAV) with dozens of rotors are becoming increasingly common for important applications such as transportation [3], [8], [13]. As the number of rotors increases, the likelihood of any rotor becoming faulty at runtime also increases. Therefore, it is critical to efficiently and effectively handle faults in these systems to avoid undesirable consequences such as potential crashes [2].

The standard approach to handling faults is fault-tolerant control, either by designing the controller to be able to stabilize itself in the presence of a fault (passive methods) or by isolating the fault and changing the control parameters to adapt (active methods). While fault-tolerant control has been explored extensively in control systems, we are not aware of any existing work that applies to faulty rotors in multirotor systems with dozens of rotors. A a first step towards active fault-tolerant control for such systems, we present an efficient method for online fault detection and isolation (FDI) of faulty rotors in MAVs with many rotors.

**Challenges.** Within a MAV, rotors are bound together in a single rigid body; thus, a fault in one rotor can impact the motion of the entire system. Consider the examples shown in Fig. 1. The left figure shows a non-modular chAIR system [8], which can lift a human using a 76-rotor MAV. The right figure shows ModQuad [18], a modular aerial vehicle composed of cuboid modules that are bound together (e.g., using magnets) to form a rigid body, with each module being propelled by a quadrotor. In both examples, a fault in one rotor can substantially impact the motion of the structure as a whole. It is, therefore, important to isolate the rotor that is faulty, and to estimate how severe it is, to perform appropriate actions.

Fig. 1: (Left) The 76-rotor chAIR can lift a human [1]. (Right) ModQuad structures with 12 and 28 rotors [18]. All vehicles have redundant rotors.

There are several challenges towards this goal, however. First, a rotor may be *partially faulty*, so the fault severity lies along a continuous scale; thus, searching over the infinitely many possible fault severity levels would be impractical. Second, the magnitude of the impact of a fault depends on multiple factors, including e.g., structure shape, fault location and fault severity. In addition, different fault scenarios may have similar impact on the system, making them indistinguishable from one another; for instance, a less severe fault in a higher-leverage rotor may cause the same error in motion as a more severe fault in a lower-leverage rotor. Together, these make it hard to efficiently isolate the faulty rotor and to correctly estimate its fault severity, especially as the number of rotors increases.

**Related work.** Fault-tolerant control methods have been studied extensively; for a recent survey on the topic, see [17].

Active fault-tolerant techniques [10], [15] first use an FDI method to identify the faulty rotor, and then tune the controller to adapt to the identified fault. Some solutions assume that an FDI method is given [10], [21], whereas others (like ours) focus on the FDI problem. Existing FDI methods typically fall into two categories: model-based approaches [9], [15] and machine learning approaches [7], [14]. The former is often system-specific and does not scale well, whereas the latter tends to require adding hardware and lots of data.

Recently, other data-driven FDI methods have also been developed, but they are either too slow or do not scale to many rotors. For example, IMU data have been used to find faulty propellers, but it can take over 100 seconds to detect a single fault in a quadrotor [4], [5]. Saied et al. [15] uses the signs of orientation error vector elements to quickly identify faulty rotors in octorotors, but their technique cannot extend to systems with more than eight rotors.

Passive fault-tolerant methods, in contrast, do not need an FDI method, because the controller is designed to work

We gratefully acknowledge the support of NSF grants CNS-1750158, CNS-1955670, CNS-2111688, CCF-2326606, and FRR-544926.

[1]Neeraj Gandhi and Linh Thi Xuan Phan are with the University of Pennsylvania, Philadelphia, PA, USA. {ngandhi3, linhphan}@seas.upenn.edu

[2]Jiawei Xu and David Saldaña are with Lehigh University, Bethlehem, PA, USA. {jix519, dsaldana}@lehigh.edu

even in the presence of faults. Several passive approaches exist, including sliding mode control, backstepping, H-infinity control, nonlinear dynamic inversion, and fractional-order control [17]. Sliding mode control is robust, but more-than-expected faults can exceed the fault tolerance capabilities of the system without warning and cause the entire system to fail abruptly [16]. Backstepping and nonlinear dynamic inversion both require highly accurate (or even full) knowledge of state, and H-infinity control and fractional order control are complex, even in systems with eight or fewer rotors [17].

To the best of our knowledge, our work is the first to consider MAVs with a single rigid body containing more than eight rotors. Our solution only uses data from a pre-planned trajectory and an IMU, both of which are commonly available in robots. The insights underlying our approach also apply to systems with fewer rotors, such as in 6-rotor VTOLs [6].

**Contributions.** As the first step towards active fault-tolerant control for MAVs, we present an online FDI method that can be used to efficiently isolate rotor faults in multirotor vehicles with dozens of rotors. We make three main contributions: First, we establish that roll error and pitch error magnitudes are consistently related to each other for each individual rotor in a MAV, independent of fault severity, planned trajectory, and even structure shape. This can be used to order the faulty rotor search space. Second, we introduce a bivariate quadratic model to relate the faulty rotor location, fault severity, and the magnitude of error in roll and pitch. Given a candidate faulty rotor, we can compute by what severity it must have failed if it is indeed the faulty rotor. Third, we present a way to use *probe faults* to determine whether a candidate faulty rotor with a candidate fault severity, computed using our bivariate quadratic model, is the faulty rotor. Our evaluation in CoppeliaSim using ModQuad shows that our method can efficiently detect and isolate faults in MAVs with up to sixty-four rotors (8× more than the state of the art). Furthermore, by enabling prompt mitigating actions, our real-time detection can help increase the system's robustness against future faults (even when using just a simple mitigation strategy).

## II. Model and Approach Overview

Our approach is designed for any MAV with many rotors; however, for concreteness we present our solution based on ModQuad [18], [19]. In ModQuad, a module is a cuboid propelled by a quadrotor. Modules can self-assemble to produce a larger MAV. A set of rigidly attached modules is called a *structure*; two example structures are shown in the right image in Fig. 1. For simplicity, we focus on settings without external disturbances such as wind, but – as we will show in Section V – our technique is robust to thrust noise. The dynamics model we use is largely based on [18]; however, our fault model and FDI approach are completely new.

### A. Dynamics of MAV

The standard basis in $\mathbb{R}^3$ is given by the three unit vectors $\hat{\mathbf{b}}_1 = [1,0,0]^\top, \hat{\mathbf{b}}_2 = [0,1,0]^\top$, and $\hat{\mathbf{b}}_3 = [0,0,1]^\top$. The fixed world coordinate frame, in which the $z$-axis points up, is denoted by $\{W\}$. The set of rigidly attached modules that form a particular structure are associated with a structure coordinate frame $\{S\}$. Within a structure, the $j^{th}$ module has a module reference frame $\{M_j\}$ with its origin at the module's center of mass. The unit vectors of the bases of module frames are parallel and identically oriented to those of $\{S\}$. The location and orientation of $\{S\}$ in $\{W\}$ is specified by vector $\mathbf{r} \in \mathbb{R}^3$ and rotation matrix ${}^W R_S \in \mathtt{SO(3)}$.

The four propellers associated with each respective module are coplanar on the $xy$-plane. The $k^{th}$ propeller of the $j^{th}$ module generates force $\mathbf{f}_{jk} = \mathsf{f}_{jk}\hat{\mathbf{h}}_{jk}$, where $\mathsf{f}_{jk}$ is less than or equal to the maximum producible thrust $\mathsf{f}_{\mathsf{max}}$ and we define for compactness $\hat{\mathbf{h}}_{jk} = {}^S R_{M_j}\, {}^{M_j} R_k \hat{\mathbf{b}}_3$. The torque due to propeller drag is given by $\boldsymbol{\tau}^{\mathsf{d}}_{jk} = \mathsf{f}_{jk}(-1)^{j+k}\left(\frac{k_m}{k_f}\right)\hat{\mathbf{h}}_{jk}$, where $k_m$ and $k_f$ are experimentally obtained constants. The total force can be represented by the vector sum $\mathbf{f}_T = \sum_{jk}\mathbf{f}_{jk}$ and $\boldsymbol{\tau}_T = \sum_{jk}\boldsymbol{\tau}^{\mathsf{f}}_{jk} + \boldsymbol{\tau}^{\mathsf{d}}_{jk}$. Here, $\boldsymbol{\tau}^{\mathsf{f}}_{jk} = \mathbf{p}_{jk} \times \mathbf{f}_{jk}$ is the torque generated by the thrust force of the $k^{th}$ propeller in the $j^{th}$ module, where $\mathbf{p}_{jk} \in \mathbb{R}^3$ is the position of each propeller in $\{S\}$.

The control vector, denoted by $\mathbf{u} = [\mathsf{f}_{11}, \mathsf{f}_{12}, \ldots, \mathsf{f}_{N4}]^\top$, where $N$ is the number of modules, represents the thrust commanded to be outputted from each rotor in the system. Together with the *design matrix* $A$ (see [20, Sec. III]), this can be used to produce the wrench vector $\mathbf{w} = A\mathbf{u}$ in $\{S\}$, where $A = \begin{bmatrix} \ldots & \hat{\mathbf{h}}_{jk} & \ldots \\ \ldots & \mathbf{p}_{jk} \times \hat{\mathbf{h}}_{jk} + (-1)^{j+k}\left(\frac{k_m}{k_f}\right)\hat{\mathbf{h}}_{jk} & \ldots \end{bmatrix}$.

The dynamics of the structure can be represented by the Lagrangian for robot motion [11], [12]

$$M \begin{bmatrix} \ddot{\mathbf{r}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + C \begin{bmatrix} \dot{\mathbf{r}} \\ \boldsymbol{\omega} \end{bmatrix} + \mathbf{g} = B\mathbf{w} \tag{1}$$

where $\ddot{\mathbf{r}}$ is the linear acceleration, $\dot{\mathbf{r}}$ is the linear velocity, $\dot{\boldsymbol{\omega}}$ is the angular acceleration, $\boldsymbol{\omega}$ is the angular velocity, $M = \begin{bmatrix} \mathsf{m} I_3 & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix}$, $C = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\omega}^\times J \end{bmatrix}$, $\mathbf{g} = \begin{bmatrix} \mathsf{m} g \hat{\mathbf{b}}_3 \\ \mathbf{0} \end{bmatrix}$, and $B = \begin{bmatrix} {}^W R_S & \mathbf{0} \\ \mathbf{0} & I_3 \end{bmatrix}$. Here, $M$ collects the mass $\mathsf{m}$ and inertia tensor $J$ of the structure, $C$ is the Coriolis matrix and the operator $(.)^\times$ converts a vector into its skew-symmetrix matrix form, $\mathbf{g}$ is the gravitational force vector ($g$ is gravitational acceleration), $B$ maps the wrench vector $\mathbf{w}$ from $\{S\}$ to $\{W\}$, and $I_3$ is the $3 \times 3$ identity matrix.

**Faults and errors.** A rotor's current 'fault level' can be quantified as a severity $\mathsf{s}_{jk} \in [0,1]$, where $(j,k)$ is the index of the rotor. For notational convenience, we additionally define $\Gamma_{jk} = 1 - \mathsf{s}_{jk}$, and we coaelesce these quantities acrosss all rotors in the structure into the vector $\boldsymbol{\Gamma} = [\Gamma_{11}, \Gamma_{12}, \ldots, \Gamma_{N4}]^\top$. An individual rotor's thrust can now be redefined as $u'_{jk} = \min\{\Gamma_{jk}u_{jk}, \Gamma_{jk}\mathsf{f}_{\mathsf{max}}\}$, where $u_{jk}$ is the thrust commanded for this rotor by the control algorithm. We consider a single rotor fault at a time; however, multiple rotors can become faulty sequentially, in which case we assume that a subsequent fault only occurs after the current one has been isolated.

The trajectory planner for the structure outputs a desired state, and we can extract orientation information by using a gyroscope to measure angular velocity. Since our approach is mainly concerned with roll and pitch, we define a limited state error vector $\mathbf{e} = [e_\phi, e_\theta]^\top$ consisting of roll angle error $e_\phi$ and

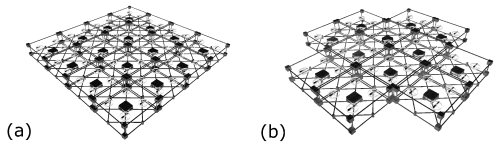

Fig. 2: $4 \times 4$ (a) square structure and (b) plus structure.

pitch angle error $e_\theta$. We denote $\boldsymbol{\eta} = [\eta_\phi, \eta_\theta]^\top$ as the vector of error magnitudes, where $\eta_\phi = |e_\phi|$ and $\eta_\theta = |e_\theta|$. The median value of $\boldsymbol{\eta}$ over a time window $\Delta_x$ is given by $\overline{\boldsymbol{\eta}}_x$.

**Objective.** Given a flying MAV, our goal is to detect whether any rotor is faulty and, if so, isolate the faulty rotor and estimate its fault severity. Our solution should work even when some prior faults have been previously identified in the MAV.

### *B. Fault detection and isolation approach overview*

Detection can be done by thresholding state error data, but this approach is unreliable under noise in the rotor thrust output. To avoid this problem, our detection method performs error thresholding on a function of the standard deviation in roll and pitch error over a baseline time window of data. When a fault is isolated, we collect a new baseline window.

Fault isolation aims to answer two key questions: (1) How to efficiently find the faulty rotor as the search space grows with more rotors? and (2) Given a candidate rotor, how to determine whether it is indeed the faulty one? We address these questions using four key insights: *I*1) By adapting the technique in [15], we can quickly identify the structure quadrant containing the faulty rotor. *I*2) The roll-error-to-pitch-error ratio is stable across fault severity levels, planned trajectories, and structure shapes; using this ratio, we can sort the candidate rotors by their likelihood of being faulty. *I*3) Using a bivariate quadratic model that relates the error in roll (or pitch), the distance from the axis roll (or pitch) is found w.r.t., and the fault severity, we can estimate the fault severity of a candidate rotor based on its structure-frame position and the observed error. *I*4) For a candidate rotor, injecting a *probe fault* can help determine whether the candidate is indeed faulty. Specifically, if the change in state error is consistent with the estimation (in *I*3) plus the probe fault severity, the search can terminate; otherwise, it moves to the next rotor.

Due to space limitations, our experimental studies focus on the two ModQuad structures shown in Fig. 2, referred to as the 'square' and 'plus' structures. However, the core technique is designed to work for other symmetric ModQuad structure shapes as well. We now discuss our approach in detail.

## III. Detecting the presence of a rotor fault

One way to detect the presence of a fault is to set up thresholds on (roll and pitch) errors: if the error observed at runtime exceeds the set threshold, a fault is considered present. This simple strategy is insufficient, however, since we can expect that noise, inertia, and the control gains will result in imperfect tracking of the desired state. Instead, we take the statistical median over a window of time. Specifically, we compute $\overline{\boldsymbol{\eta}}_b = [\eta_\phi \;\; \eta_\theta]^\top$ over a time window $\Delta_b$, where $\Delta_b$ is a 'baseline' window to compare newer windows against.

We periodically collect a new window $\Delta_n$ and check if

$$||\overline{\boldsymbol{\eta}}_n - \overline{\boldsymbol{\eta}}_b > m\boldsymbol{\sigma}_{\boldsymbol{\eta}_b}||_1 = 2, \text{ or} \tag{2}$$

$$||\overline{\boldsymbol{\eta}}_n - \overline{\boldsymbol{\eta}}_b||_1 > m'||\boldsymbol{\sigma}_{\boldsymbol{\eta}_b}||_1, \tag{3}$$

where $m \in \mathbb{R} : m > 0$ and $m' \in \mathbb{R} : m' > m$ are tunable constants, $\boldsymbol{\sigma}_{\boldsymbol{\eta}_b}$ is the standard deviation in $\boldsymbol{\eta}$ over $\Delta_b$, $>$ returns 0 (false) or 1 (true), and $||\cdot||_1$ is the L1-norm. This not only makes fault detection more robust to noise but can also be used to detect sequential faults (i.e., the next fault can be detected based on a new baseline $\Delta_{b'}$). Our approach uses both (2) and (3) because sometimes the roll error is much larger than the pitch error, and vice versa. (3) can detect the fault even if one of the errors is low; in contrast, (2) is better at detecting faults in the presence of noise. This is also the reason for $m' > m$.

## IV. Isolating the faulty rotor

Once a fault is detected, our technique 1) identifies the quadrant containing the fault, 2) performs sub-quadrant search by examining rotors of the identified quadrant in decreasing likelihood of being faulty, 3) computes the severity with which a candidate rotor should have failed to produced the observed errors, and 4) uses probe faults to verify faultiness.

**Quadrant isolation.** Quadrant isolation is inspired by [15], which uses the errors in orientation to create an inference table for finding the faulty rotor in an octorotor. We adapt this idea to identify the quadrant of the structure that contains a fault. Since $(\mathsf{sgn}(e_\phi), \mathsf{sgn}(e_\theta))$ is unique for each quadrant, we can use Table I to narrow down the search space.

TABLE I: Inference table for identifying the faulty quadrant.

| $e_\phi$ | $e_\theta$ | Quadrant with Fault |
|---|---|---|
| + | - | I |
| + | + | II |
| - | + | III |
| - | - | IV |

**Sub-quadrant candidate ordering with error ratios.** Next, we search through the candidate faulty rotors within the quadrant. The first step is to order rotors by their likelihood of being faulty. We start by modeling the effect of a fault on the dynamics of the structure. The wrench vector computation is rewritten as $\mathbf{w}' = AH\mathbf{u}$, where $H = \mathsf{diag}(\boldsymbol{\Gamma})$ is a $4N \times 4N$ diagonal matrix defined as having elements of $\boldsymbol{\Gamma}$ along its diagonal and zeros elsewhere. Thus, the dynamical model described in (1) can be reformulated as $M\begin{bmatrix}\ddot{\mathbf{r}}\\ \dot{\boldsymbol{\omega}}\end{bmatrix} + C\begin{bmatrix}\dot{\mathbf{r}}\\ \boldsymbol{\omega}\end{bmatrix} + \mathbf{g} = B\mathbf{w}$.

We next look specifically at the angular acceleration expression: $J\dot{\boldsymbol{\omega}} + \boldsymbol{\omega}^\times J\boldsymbol{\omega} = \boldsymbol{\tau}_T$. By rearranging, we get $\dot{\boldsymbol{\omega}} = J^{-1}(\boldsymbol{\tau}_T - \boldsymbol{\omega}^\times J\boldsymbol{\omega})$.[1] This is the expected angular acceleration (i.e., what we would expect if $\boldsymbol{\Gamma} = \mathbf{1}$). We can similarly compute the *achieved* angular acceleration as $\dot{\boldsymbol{\omega}}' = J^{-1}(\boldsymbol{\tau}'_T - \boldsymbol{\omega}'^\times J\boldsymbol{\omega}')$; this is the acceleration achieved given the fault(s) in the system. We can double-integrate each of the elements of $\dot{\boldsymbol{\omega}}$ and $\dot{\boldsymbol{\omega}}'$ to find the expected and achieved

[1]The inertia matrix must be invertible; the structure has mass distributed in 3D space, meaning that in the body-fixed frame, where the basis is defined along the principal axes, the inertia matrix is diagonal and full rank, implying the invertibility of $J$.

roll, pitch, and yaw: $\int\int \dot{\boldsymbol{\omega}} dt = J^{-1}(\int\int (A_{\boldsymbol{\tau}_T}\mathbf{u} - \boldsymbol{\omega}^{\times} J\boldsymbol{\omega})\, dt)$, where $A_{\boldsymbol{\tau}_T}$ is the bottom row of $A$. We abstract the Taylor series of this result as $\int\int \dot{\boldsymbol{\omega}} dt = D_1 \boldsymbol{\ell}_1^{(n)}(t) - D_2 \ell_2^{(n')}(t)$, where $D_1 \boldsymbol{\ell}_1^{(n)}(t) = J^{-1}\int\int A_{\boldsymbol{\tau}_T}\mathbf{u} dt$, $D_2 \boldsymbol{\ell}_2^{(n')}(t) = J^{-1}\int\int \boldsymbol{\omega}^{\times} J\boldsymbol{\omega} dt$, $D_1$ and $D_2$ are $(n+1)\times(n+1)$ coefficient matrices, $\boldsymbol{\ell}_1^{(n)}(t) = \begin{bmatrix} t^n & t^{n-1} & \dots & 1 \end{bmatrix}^{\top}$, $\boldsymbol{\ell}_1^{\prime(n)}(t) = \begin{bmatrix} t^{n'} & t^{n'-1} & \dots & 1 \end{bmatrix}^{\top}$, and $n$ and $n'$ represent the order of the polynomial Taylor series. Similarly, $\int\int \dot{\boldsymbol{\omega}}' dt = D_1' \boldsymbol{\ell}_1^{\prime(n)}(t) - D_2' \ell_2^{\prime(n')}(t)$. We can then find the orientation error as $\int\int \dot{\boldsymbol{\omega}} dt - \int\int \dot{\boldsymbol{\omega}}' dt = \left(D_1 \boldsymbol{\ell}_1^{(n)}(t) - D_2 \boldsymbol{\ell}_2^{(n')}(t)\right) - \left(D_1' \boldsymbol{\ell}_1^{\prime(n)}(t) - D_2' \boldsymbol{\ell}_2^{\prime(n')}(t)\right)$. For compactness, we define $\Omega_\phi(t) = \left(D_{1_\phi} \boldsymbol{\ell}_1^{(n)}(t) - D_{2_\phi} \boldsymbol{\ell}_2^{(n')}(t)\right) - \left(D'_{1_\phi} \boldsymbol{\ell}_1^{\prime(n)}(t) - D'_{2_\phi} \boldsymbol{\ell}_2^{\prime(n')}(t)\right)$ and $\Omega_\phi(t) = \left(D_{1_\theta} \boldsymbol{\ell}_1^{(n)}(t) - D_{2_\theta} \boldsymbol{\ell}_2^{(n')}(t)\right) - \left(D'_{1_\theta} \boldsymbol{\ell}_1^{\prime(n)}(t) - D'_{2_\theta} \boldsymbol{\ell}_2^{\prime(n')}(t)\right)$.

Then, we can take the limit of the ratio of the roll and pitch errors: $\lim_{t\to\infty} \frac{\Omega_\phi(t)}{\Omega_\theta(t)}$ . Let $\boldsymbol{\zeta}$ be the vector with the first element being one and all others zero. If $n > n'$, the limit becomes $\frac{(D_{1_\phi} - D'_{1_\phi})\boldsymbol{\zeta}}{(D_{1_\theta} - D'_{1_\theta})\boldsymbol{\zeta}}$ . If $n < n'$, the limit becomes $\frac{(D_{2_\phi} - D'_{2_\phi})\boldsymbol{\zeta}}{(D_{2_\theta} - D'_{2_\theta})\boldsymbol{\zeta}}$ . Finally, if $n = n'$, the limit becomes $\frac{(D_{1_\phi} - D_{2_\phi} - D'_{1_\phi} + D'_{2_\phi})\boldsymbol{\zeta}}{(D_{1_\theta} - D_{2_\theta} - D'_{1_\theta} + D'_{2_\theta})\boldsymbol{\zeta}}$ . In all cases, the result of the limit is a constant. Thus, given a fault at a particular location, we should expect to see that the roll-to-pitch *error ratio* $\eta_\phi/\eta_\theta$ is stable. Note that the severity does not impact this ratio, since the severity term will appear in both the numerator and denominator.

Our isolation approach is based on the constant-ness of the limit observed above. Specifically, the error ratio is (mostly) agnostic to fault severity, planned trajectory, and even structure shape. (Our simulation further confirms this property, but due to space constraints we omitted the details here.) The error ratios provide a much finer-grained information about the rotors than the quadrant-level analysis. At runtime, we subtract (element-wise) the observed error ratio from a saved *error ratio profile*, flatten the result, and sort it to generate a list of candidate faulty rotors in decreasing likelihood of being faulty.

**Computing severity.** Suppose a particular rotor in a particular position in the structure is faulty. Given the $\eta_\phi$ and $\eta_\theta$ observed at runtime, we need to determine which severity level this rotor must have undergone.

Given a specific arrangement of rotors in a structure, $\eta_\phi$ and $\eta_\theta$ are a function of the distance of the rotor from the relevant axis and the severity of the fault. This is illustrated by Fig. 3, which shows the error magnitudes observed for both the square and plus structures. As before, each plot represents the data across three trajectories and four fault severities, with noise added as described in Sec. V. These figures combine data points for $\eta_\phi$ and $\eta_\theta$, since they follow the same trends.

The magnitude of orientation error observed at runtime can be modeled using the bivariate quadratic model

$$\begin{bmatrix} \eta_\phi & \eta_\theta \end{bmatrix} = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \end{bmatrix} \begin{bmatrix} 1 & \mathsf{d}_\phi & \mathsf{s} & \mathsf{d}_\phi^2 & \mathsf{d}_\phi \mathsf{s} & \mathsf{s}^2 \\ 1 & \mathsf{d}_\theta & \mathsf{s} & \mathsf{d}_\theta^2 & \mathsf{d}_\theta \mathsf{s} & \mathsf{s}^2 \end{bmatrix}^{\top}, \tag{4}$$

where $\mathsf{d}_\phi$ and $\mathsf{d}_\theta$ are the distance from the axes roll and pitch are computed w.r.t., $\mathsf{s}$ is the fault severity, and $c_0, .. c_5$ are coefficients (unique to each structure). Overlaid on the stem plots of the errors observed is a wireframe that plots the bivariate quadratic models fitted to each respective data set. Table II shows the coefficients found for each structure. The coefficients can be quite different across different structures, and the maximum $\eta$ varies by a factor of almost $3\times$.

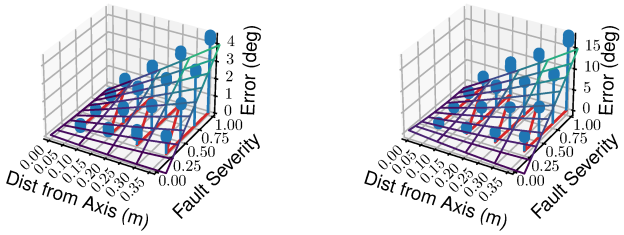


Fig. 3: Models used to compute fault severity. (Note: the z-scales in the two plots are not the same.)

TABLE II: Coefficients found for (4).

| | $c_0$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ |
|---|---|---|---|---|---|---|
| Square | 0.103 | −0.941 | −0.312 | 1.542 | 12.786 | 0.169 |
| Plus | 0.902 | −6.580 | −2.875 | 5.983 | 51.420 | 1.966 |

Thus, we can compute the severity of a fault given a) the candidate faulty rotor, for which the position in a given structure is fixed, and thus $d_\phi$ and $d_\theta$ are known, and b) the error magnitudes $\eta_\phi$ and $\eta_\theta$, which are collected as part of the previous step of finding error ratios.

**Determining if a candidate is the faulty rotor.** Once we have selected a candidate rotor to check for fault and have computed the corresponding fault severity $\mathsf{s}'$, we need to determine whether this rotor is indeed faulty. This is done via *probe fault* injection. Suppose we wish to check a rotor $(j,k)$ for having undergone a fault of severity $\mathsf{s}' = 1 - \Gamma'_{jk}$, and we wish to check whether $\mathsf{s}' \approx \mathsf{s}$, where $\mathsf{s}$ is the actual fault severity. We inject a probe fault into $(j,k)$ with severity $\mathsf{s}_p = 1 - \Gamma^p_{jk}$, where $\mathsf{s}_p$ is constant. If $(j,k)$ is the faulty rotor, then the actual thrust output $u'_{jk} = \Gamma^p_{jk}\Gamma_{jk} u_{jk}$; otherwise, $u'_{jk} = \Gamma^p_{jk} u_{jk}$. After applying the probe fault, we can collect a new window of data $\Delta_p$, from which we can compute $\overline{\boldsymbol{\eta}}_p$. We can also compute the expected $\widehat{\boldsymbol{\eta}}_p$ by plugging in $\mathsf{s}'' = 1 - \Gamma^p_{jk}\Gamma'_{jk}$ into (4) (since we already have all the coefficients and we know the location of the candidate faulty rotor, from which $d_\phi$ and $d_\theta$ are derived).

We next compute $\delta^{\overline{\boldsymbol{\eta}}}_{p,n} = (\overline{\boldsymbol{\eta}}_p - \overline{\boldsymbol{\eta}}_b) - (\overline{\boldsymbol{\eta}}_n - \overline{\boldsymbol{\eta}}_b) = \overline{\boldsymbol{\eta}}_p - \overline{\boldsymbol{\eta}}_n$ and $\delta^{\widehat{\boldsymbol{\eta}}}_{p,n} = \widehat{\boldsymbol{\eta}}_p - (\overline{\boldsymbol{\eta}}_n - \overline{\boldsymbol{\eta}}_b)$. We subtract $\overline{\boldsymbol{\eta}}_b$ from observed errors to isolate the impact of the probe fault and the actual fault on $\overline{\boldsymbol{\eta}}$ from that of pre-existing error (e.g., from previous faults). Finally, to determine whether the current fault candidate is the faulty rotor, check whether $||\delta^{\widehat{\boldsymbol{\eta}}}_{p,n} - \delta^{\overline{\boldsymbol{\eta}}}_{p,n}||_1 < m'' \sigma_{||\boldsymbol{\eta}_n||_1}$, where $m'' \in \mathbb{R} : m'' > 0$ is a tunable constant analogous to the $m$ used in fault detection and $\sigma_{||\boldsymbol{\eta}_n||_1}$ is the standard deviation in $||\boldsymbol{\eta}||_1$ over $\Delta_n$. If the condition is satisfied, the search terminates.

**Limitation.** When the structure is small, injecting probe faults can be limiting since smaller systems have fewer redundant rotors and less redundant thrust in each quadrant. Thus, while we can use the error ratio and severity computation parts of our approach for systems with few rotors (as demonstrated

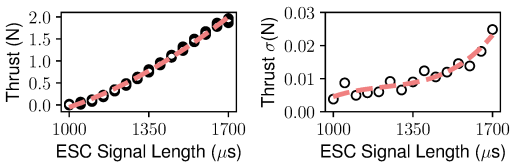


Fig. 4: Noise models used in the simulation.

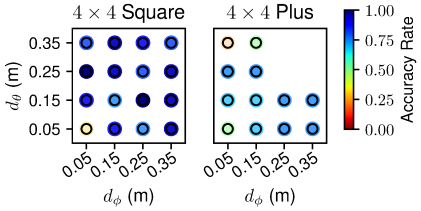


Fig. 5: Isolation accuracy tends to increase as the impact a rotor has on the planned trajectories increases.

in [6] for 6-rotor VTOL systems), our approach for validating a rotor's fault status is best suited for systems with enough spare thrust capacity to stay airborne despite faults. This is why our approach works well for MAVs with many rotors.

## V. Implementation

We developed atop ModQuad's CoppeliaSim simulator [20] (https://github.com/Jarvis-X/HModQuad-sim). We set $m = 1.75$, $m' = 2m$, $m'' = 1.75$, $\varepsilon = 0.25$, $s_p = 0.5$, and $\Delta = 8$s, where $\varepsilon$ is a tuning factor needed for multiple sequential faults. We used four trajectories: a diagonal line, a spiral trajectory, a zigzag trajectory in the $xy$-plane, and a zigzag trajectory in the $xz$-plane (the last of which was not included in our profiling). We tested $s \in \{0.25, 0.5, 0.75, 1.0\}$. The fault detector ran at 1Hz; fault detection time was measured from the time of fault injection to the time of detection. Fault isolation time was likewise measured from fault injection to isolation. Due to structure symmetry, we tested faults in one of the four quadrants of each structure.

**Thrust noise.** For simplicity, we assume external forces such as wind are negligible, and techniques such as Kalman filter are used to handle noise in sensor measurements. A remaining major source of noise is in the thrust output of each rotor. To model this noise, we ran experiments on the EMax RS-1108/5200KV motor equipped with a type-3020 bi-blade three-inch propeller, as used in [20]. The electronic speed controller outputs a signal of varying length (denoted $\ell$), where a larger $\ell$ corresponds to a faster rotational velocity, which in turn produces a larger thrust. Fig. 4 shows the results in black, along with overlaid regression models in dashed light-red lines. The equations $u = a_0 + a_1\ell + a_2\ell^2$ and $\sigma_u = b_0 - b_1\ell + b_2\ell^2 + b_3\ell^3$ related $\ell$ to $u$ and $\sigma_u$ well, respectively, where $u$ is thrust in Newtons and $\sigma_u$ is standard deviation in $u$. We found $a_0 = -0.2269$, $a_1 = -0.0014$, $a_2 = 1.6 \times 10^{-6}$, $b_0 = -0.2$, $b_1 = -4.9 \times 10^{-4}$ $b_2 = 3.98 \times 10^{-7}$, and $b_3 = -1.09 \times 10^{-10}$. In simulation, given the desired thrust $u$ for a rotor, the quadratic formula is used to find the $\ell$-value that would produce $u$, after which we can compute $\sigma_u$. The output of rotor $(j,k)$ is thus $u'_{jk} = u_{jk} + G(\mu, \sigma_{u_{jk}})$, where $G$ is Gaussian noise and $\mu = 0$N.

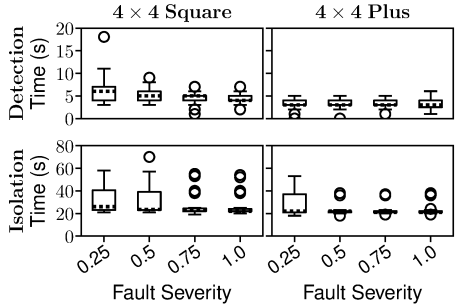


Fig. 6: Lower severities have less impact on motion and take longer to detect and isolate.

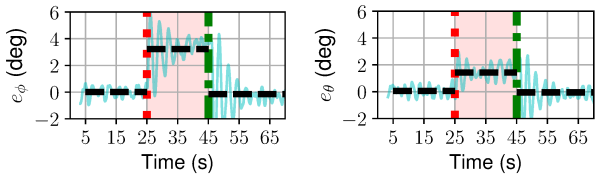


Fig. 7: $K_{j,k}$ applied at about $t = 45$s results in lower error despite a fault.

## VI. Evaluation

Our evaluation aims to answer three key questions: 1) How accurate is our isolation technique? 2) How much time do detection and isolation take? 3) When used with a recovery strategy, how well does it improve robustness against faults?

**Isolation accuracy.** Fig 5 shows our accuracy results for one quadrant of both the square and plus structures. The lighter colors shown for some rotors usually stems from inaccuracies at lower fault severities, which are harder to isolate. The square structure tends to perform better than the plus structure, which is partly because the square structure is better able to handle both the thrust loss from the rotor fault itself and the thrust loss from the probe faults. The silver lining is that, typically, we would expect it to be harder to achieve high accuracy when the system has more rotors to consider as fault candidates, but we actually see better accuracy with more rotors.

**Detection and isolation time.** Fig. 6 shows the time-to-detect and time-to-isolate for the square and plus structures, for those rotors that were correctly isolated. Lower-severity faults take a bit longer to detect and isolate, but this is expected since a smaller fault disrupts motion less, and is thus less distinct from normal behavior. Once detected, however, the time to isolate the fault is similar for most rotors. We can observe that the median detection time is virtually identical across severities, although the variance is still a bit larger for lower-severity faults. Note that these times are affected by the period with which we ran our detector (once per second); if it is run more often, the observed times should decrease.

**Impact on robustness when combining with recovery.** A benefit of our FDI technique is that, by efficiently isolating the fault, we can promptly apply recovery actions to improve performance and handle multiple sequential faults. While our technique works with any recovery strategy, for illustration purposes, we implemented a simple recovery strategy for low-severity faults. Specifically, once we have

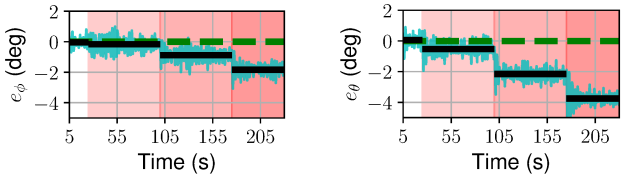


Fig. 8: Fault impact on motion compounds.

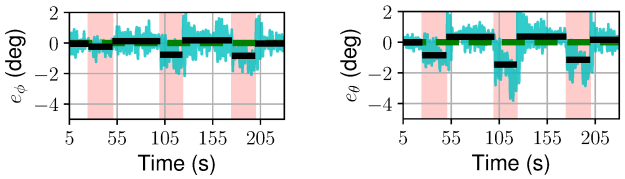


Fig. 9: Our approach can keep median error close to 0°.

isolated the rotor $(j,k)$ as being faulty and have checked that $\mathsf{s}'_{j,k} \approx \mathsf{s}_{j,k}$ is a reasonable estimate for the fault severity, we apply the compensatory gain $K_{j,k} = 1/\Gamma'_{j,k}$ to the rotor thrust, where $\Gamma'_{j,k} = 1 - \mathsf{s}'$. Thus, the actual thrust output is $u'_{j,k} = \min\{K_{j,k}\Gamma_{j,k}u_{j,k}, \Gamma_{j,k}\mathsf{f}_{\mathsf{max}}\}$. This does not affect the limits from Sec. IV, since all we are changing are constant multipliers, so far as the analysis is concerned.

Fig. 7 shows how our FDI combined with recovery can significantly reduce $\eta_\phi$ and $\eta_\theta$. The dotted, red, vertical line shows the time of fault injection, the dash-dotted, green, vertical line shows the time of fault compensation, and the dashed, black, horizontal line shows median error in segments divided by vertical lines. Here, we injected the fault with severity $\mathsf{s} = 0.5$ into a rotor of high leverage. We observe that isolating and compensating for the rotor can reduce median roll error from over 2 degrees to a value close to 0 degrees, with similar results for pitch error.

Fig. 7 also shows that while the compensatory gain is quite effective, it can leave a 'residue' offset relative to the ideal 0° error. This occurs because at runtime, we typically will find $\mathsf{s}' \approx \mathsf{s}$, but not $\mathsf{s}' = \mathsf{s}$. Thus, the compensatory gain compensates for $\mathsf{s}'$ and not $\mathsf{s}$, which in turn means that a fault of severity $|\mathsf{s} - \mathsf{s}'|$ is left in the system (although this value is typically small). Thus, checking for future faults can be affected by artifacts of previous fault recovery procedures. To address this, once we apply $K_{j,k}$, we increment $m$ in (2) by $\varepsilon$ to reduce the chance of false positives in subsequent fault detection, and collect a new baseline window of data $\Delta_{b'}$. The new baseline allows us to isolate the impact that a new fault has on the system, thereby enabling re-use of the fault detection, isolation, and simple recovery approaches for future faults as we did for already-compensated-for faults.

**Multiple sequential faults.** Finally, we sequentially injected three faults of severity $\mathsf{s} \in \{0.3, 0.4, 0.5\}$ in three rotors of a quadrant in the square structure traversing a spiral trajectory. Fig. 8 shows results *without* our technique. The dashed green line shows the ideal error of 0°, and the solid black horizontal lines overlaid on the raw data show the median for the relevant time spans. Translucent rectangles show when each fault was impacting the system; darker shades indicate compounded fault impact. Without our method, the compound effect of the faults pushes error to $-2°$ and $-4°$ error in roll and pitch, respectively. In contrast, with our method in Fig. 9, despite the presence of three faults, we can keep the median error in both roll and pitch close to 0°.

## VII. Conclusion

Fault detection and isolation is a difficult problem in multirotor aerial vehicles, especially as more rotors are added. We have presented the first FDI technique for MAVs with dozens of rotors. Our evaluation shows that our technique can efficiently detect and isolate faulty rotors in systems with 8× more rotors than the state of the art, and that it enables prompt recovery actions to improve the system's robustness and handle multiple sequential faults.

## References


[1] amazingDIYprojects. chAIR -Manned drone Episode 24 -Playtime! Electric VTOL Axel Borg, 2017. YouTube.

[2] Associated Press. Report: Propeller blade broke, causing military plane crash in Mississippi. *WREG*, 2020.

[3] N. Gandhi, D. Saldaña, V. Kumar, and L. T. X. Phan. Self-reconfiguration in response to faults in modular aerial systems. *RA-L*, 5(2):2522–2529, 2020.

[4] B. Ghalamchi, Z. Jia, and M. W. Mueller. Real-time vibration-based propeller fault diagnosis for multicopters. *TMECH*, 25(1):395–405, 2020.

[5] B. Ghalamchi and M. W. Mueller. Vibration-based propeller fault diagnosis for multicopters. In *ICUAS*, 2018.

[6] J. Lian, N. Gandhi, Y. Wang, and L. T. X. Phan. Online rotor fault detection and isolation for vertical takeoff and landing vehicles. In *IROS*, 2024.

[7] W. Liu, Z. Chen, and M. Zheng. An audio-based fault diagnosis method for quadrotors using convolutional neural network and transfer learning. In *ACC*, 2020.

[8] C. MacDonald. The $10,000 Swedish 'flying carpet' drone powerful enough to carry its creator (and his armchair). *DailyMail*, 2017.

[9] S. P. Madruga, T. P. do Nascimento, F. Holzapfel, and A. M. N. Lima. Estimating the loss of effectiveness of UAV actuators in the presence of aerodynamic effects. *RA-L*, 8:1335–1342, 2023.

[10] M. W. Mueller and R. D'Andrea. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In *ICRA*, 2014.

[11] R. M. Murray, S. Sastry, and Z. Li. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[12] R. Ortega and M. W. Spong. Adaptive motion control of rigid robots: A tutorial. In *CDC*, 1988.

[13] R. Oung and R. D'Andrea. The distributed flight array: Design, implementation, and analysis of a modular vertical take-off and landing vehicle. *IJRR*, 33(3):375–400, 2014.

[14] R. Puchalski, A. Bondyra, W. Giernacki, and Y. Zhang. Actuator fault detection and isolation system for multirotor unmanned aerial vehicles. In *MMAR*, 2022.

[15] M. Saied, B. Lussier, I. Fantoni, C. Francis, H. Shraim, and G. Sanahuja. Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor. In *ICRA*, 2015.

[16] M. Saied, B. Lussier, I. Fantoni, H. Shraim, and C. Francis. Active versus passive fault-tolerant control of a redundant multirotor UAV. *The Aeronautical Journal*, 124:385 – 408, 2019.

[17] M. Saied, H. Shraim, and C. Francis. A review on recent development of multirotor uav fault-tolerant control systems. *Aerospace and Electronic Systems Magazine*, 39:146–180, 2023.

[18] D. Saldaña, B. Gabrich, G. Li, M. Yim, and V. Kumar. ModQuad: The flying modular structure that self-assembles in midair. In *ICRA*, 2018.

[19] J. Xu, D. S. D'Antonio, and D. Saldaña. Modular multi-rotors: From quadrotors to fully-actuated aerial vehicles, 2022. arXiv:2202.00788.

[20] J. Xu, D. F. Salazar-D'Antonio, and D. Saldaña. H-ModQuad: Modular multi-rotors with 4, 5, and 6 controllable DOF. In *ICRA*, 2021.

[21] B. Yu, Y. Zhang, and Y. Qu. Fault tolerant control using PID structured optimal technique against actuator faults in a quadrotor UAV. In *ICUAS*, 2014.