



Delft University of Technology

## A Tactile Feedback Approach to Path Recovery After High-Speed Impacts for Collision-Resilient Drones

Bredenbeck, A.; Yang, Teaya ; Hamaza, S.; Mueller, Mark W.

**DOI**

[10.3390/drones9110758](https://doi.org/10.3390/drones9110758)

**Publication date**

2025

**Document Version**

Final published version

**Published in**

Drones

**Citation (APA)**

Bredenbeck, A., Yang, T., Hamaza, S., & Mueller, M. W. (2025). A Tactile Feedback Approach to Path Recovery After High-Speed Impacts for Collision-Resilient Drones. *Drones*, 9(11), Article 758.  
<https://doi.org/10.3390/drones9110758>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

---

*Article*

# A Tactile Feedback Approach to Path Recovery After High-Speed Impacts for Collision-Resilient Drones

Anton Bredenbeck <sup>1,\*</sup>, Teaya Yang <sup>2</sup>, Salua Hamaza <sup>1</sup> and Mark W. Mueller <sup>2</sup>

<sup>1</sup> Biomimetic Intelligence Lab, Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS Delft, The Netherlands; s.hamaza@tudelft.nl

<sup>2</sup> HiPeRLab, Department of Mechanical Engineering, University of California Berkeley, Berkeley, CA 94720, USA; teaya.yang@berkeley.edu (T.Y.); mwm@berkeley.edu (M.W.M.)

\* Correspondence: a.bredenbeck@tudelft.nl

## Highlights

### What are the main findings?

- The proposed approach exploits tactile feedback from collisions to infer obstacle locations in the environment.
- Our collision-aware estimator uses pre-collision velocities, rates and tactile feedback to predict post-collision velocities and rates alongside a vector-field-based path representation and recovery strategy to improve state estimation and ensure safe traversal of cluttered environments at low computational cost.

### What are the implications of the main findings?

- The proposed method enables robust navigation in environments where traditional vision- or range-based sensing is unreliable.
- The proposed method allows drones to recover in-flight from high-speed collisions and adapt their paths afterwards, preventing repeated impacts and improving resilience in cluttered settings.

## Abstract



Academic Editor: Chao Huang

Received: 16 September 2025

Revised: 17 October 2025

Accepted: 29 October 2025

Published: 31 October 2025

**Citation:** Bredenbeck, A.; Yang, T.; Hamaza, S.; Mueller, M.W. A Tactile Feedback Approach to Path Recovery After High-Speed Impacts for Collision-Resilient Drones. *Drones* **2025**, *9*, 758. <https://doi.org/10.3390/drones9110758>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Aerial robots are a well-established solution for exploration, monitoring, and inspection, thanks to their superior maneuverability and agility. However, in many environments, they risk crashing and sustaining damage after collisions. Traditional methods focus on avoiding obstacles entirely, but these approaches can be limiting, particularly in cluttered spaces or on weight- and computationally constrained platforms such as drones. This paper presents a novel approach to enhance drone robustness and autonomy by developing a path recovery and adjustment method for a high-speed collision-resilient aerial robot equipped with lightweight, distributed tactile sensors. The proposed system explicitly models collisions using pre-collision velocities, rates and tactile feedback to predict post-collision dynamics, improving state estimation accuracy. Additionally, we introduce a computationally efficient vector-field-based path representation that guarantees convergence to a user-specified path, while naturally avoiding known obstacles. Post-collision, contact point locations are incorporated into the vector field as a repulsive potential, enabling the drone to avoid obstacles while naturally returning to its path. The effectiveness of this method is validated through Monte Carlo simulations and demonstrated on a physical prototype, showing successful path following, collision recovery, and adjustment at speeds up to 3.7 m/s.

**Keywords:** drones; collision resilient; state estimation; tactile-based control

## 1. Introduction

Aerial robots, known as Micro Aerial Vehicles (MAVs), find evermore use cases in industry and in academia thanks to their superior mobility and agility. Examples include infrastructure monitoring [1,2], environmental monitoring [3,4], manipulation tasks [5,6], and search and rescue [7,8], all featuring individual MAVs and swarms applications [9,10]. Especially in cluttered environments, challenges still persist. Efforts to improve the dependability of MAVs operating in these spaces can be broadly classified into two categories: object avoidance and collision resilience.

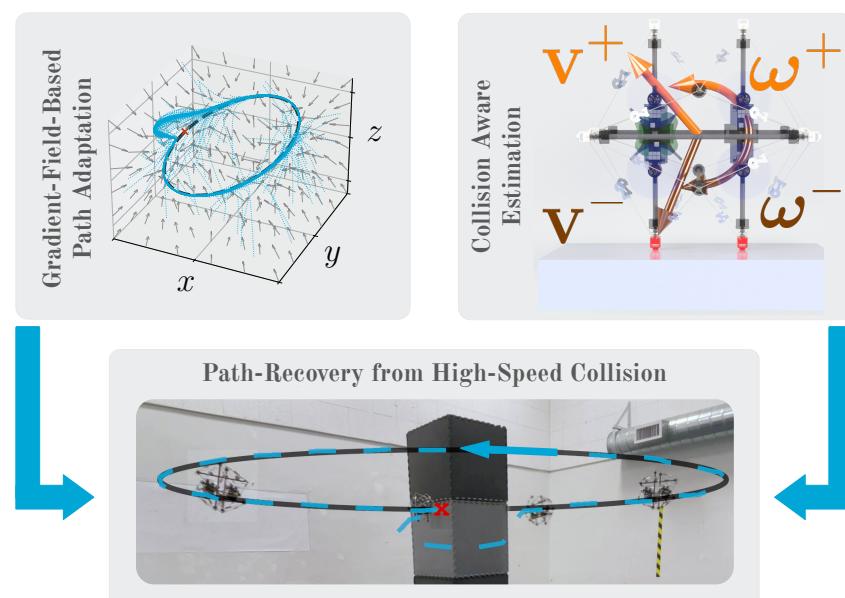
State-of-the-art object avoidance relies on sensing methods, like cameras or LiDAR, to detect objects and follow collision-free paths. In particular, advancements in computer vision, path planning and control have helped improve MAVs' reliability. Some approaches use explicit mapping pipelines to create maps and precomputed paths, which are then sent to a low-level controller [11,12]. Other approaches learn collision-free flight directly from sensor data [13,14], where [15] explicitly uses aerodynamic effects while in close proximity to sensed obstacles. Collision resilience, instead, focuses on increasing the robustness of the platform by integrating flexible enclosures and mechanical compliance into the design [16–24]. However, all mentioned strategies rely on complex and power-hungry sensors, as well as costly onboard computation to run various mapping pipelines, or perform inference of deep neural networks in real time. For example, conventional LiDAR sensors require approximately 10 W and weigh upwards of 500 g, which is unsuitable for small MAVs [25], while also being unable to detect transparent obstacles or operate in smoky or dusty conditions. On the other hand, vision-based approaches are often too slow for collisions and fail under sudden velocity jumps. Re-initialization during impact is further hindered by limited features, e.g., when the camera's field-of-view is filled by the obstacle, and reduced feature reliability from motion blur during recovery. Meanwhile, the latter strategies struggle to recover from high-speed collisions while maintaining flight. Furthermore, they often have computational requirements similar to the object avoidance strategies to adapt their path post-collision. Lastly, neither strategy exploits information gained from physical contact with the environment.

In nature, on the other hand, we observe that *physical contact* takes on the role of a low-latency and low-bandwidth source of information as an alternative to visual information, often connected to reflexive behaviors. Humans and animals use the tactile feedback they obtain from touch to trigger fast responses to recover from unforeseen deviations from their desired state. Beyond reflexes, in nature we also observe how this data feeds into their model of the surrounding environment; e.g., a human moving through a dark corridor uses touch on the walls to navigate until the visual cues become available again. In an early work on contact detection for MAVs, Bakir et al. introduce a soft sensor that detects contacts during flight at speeds up to 1 m/s [26]. In this work, we go beyond this and introduce the use of tactile feedback to improve collision resilience on drones for high-speed impacts. This enables particularly small and lightweight MAVs to travel at higher speeds in cluttered environments, even without functioning remote sensing modalities, as collisions no longer pose a significant risk, enabling a myriad of applications such as improved search and rescue operations. Here, lightweight and low-bandwidth touch sensors inform on when and where on the drone frame contact events occur during a collision, enabling state estimation through contact and real-time path recovery and trajectory adjustments to avoid the same obstacles. Consequently, the main contribution of this work is a high-speed collision path recovery strategy consisting of three components, as seen in Figure 1:

- Instead of evading obstacles, the proposed approach uses tactile feedback acquired through collisions to infer the locations of obstacles in the environment.

- A collision-aware estimator uses pre-collision velocities, rates, and tactile feedback in the form of collision locations to predict the post-collision velocities and rates, which enables improved state estimation through contact.
- A vector-field-based path representation and recovery strategy guarantees convergence to a desired path and adapts the path after contact to avoid re-collision by adding known objects as a repulsive potential.

The rest of this paper is structured as follows: Section 2 discusses related work, Section 3 details the proposed model used throughout the manuscript, Section 4 describes the proposed approach to include the contact information in estimation and control, Section 5 showcases the hardware implementation, Section 6 details the simulated and real experiments, and, finally, Section 7 concludes the paper and outlines future research directions.



**Figure 1.** A path (black dashed line) is represented by a vector field (gray arrows) that guarantees convergence from any location while avoiding known obstacles (top left). Using the pre-collision velocity ( $v^-$  and  $\omega^-$ ) and contact points on the drone frame, we predict the post-collision velocity ( $v^+$  and  $\omega^+$ ) and updated state estimate to facilitate recovery (top right). Together, these enable the MAV to withstand collisions and adapt its path to prevent future ones (bottom).

## 2. Related Work

Drones typically rely on visual and range sensing to detect and avoid obstacles, preventing collisions. However, few approaches instead embrace collisions by designing mechanically resilient systems, incorporating collision handling mechanisms in control algorithms, or combining both. Briad et al. [16] introduced a collision-resilient drone with a protective shell that decouples rotational dynamics from external impacts. Wang et al. [21,22] also use an exterior shell to protect onboard electronics while integrating collision data into the mapping pipeline. Mintchev et al. [27] employ a soft exoskeleton to enable safe collisions and protect sensitive components. Other methods employ compliant interaction tools to enable safe contact and extract collision properties such as contact forces, normals, and obstacle locations through deformation [18,19,24,26,28,29].

Zha et al. [30] embed impact protection into the drone's frame using an icosahedral tensegrity structure capable of withstanding high-force impacts—the same frame used in this work. Other approaches integrate compliance directly into the structure, reducing additional weight while maintaining collision resilience [23,31,32].

Going from mechanical design to algorithmic works, several studies explicitly consider collisions to enhance drone control and recovery. Battison et al. [33] analyze attitude observers that minimize estimation errors through impacts, improving recovery. Lew et al. [34] leverage low-velocity collisions to refine state estimation, enabling velocity-based exploration in cluttered environments. Zha et al. [17] extend their prior work by allowing a drone to reorient its propellers and resume operation after a crash.

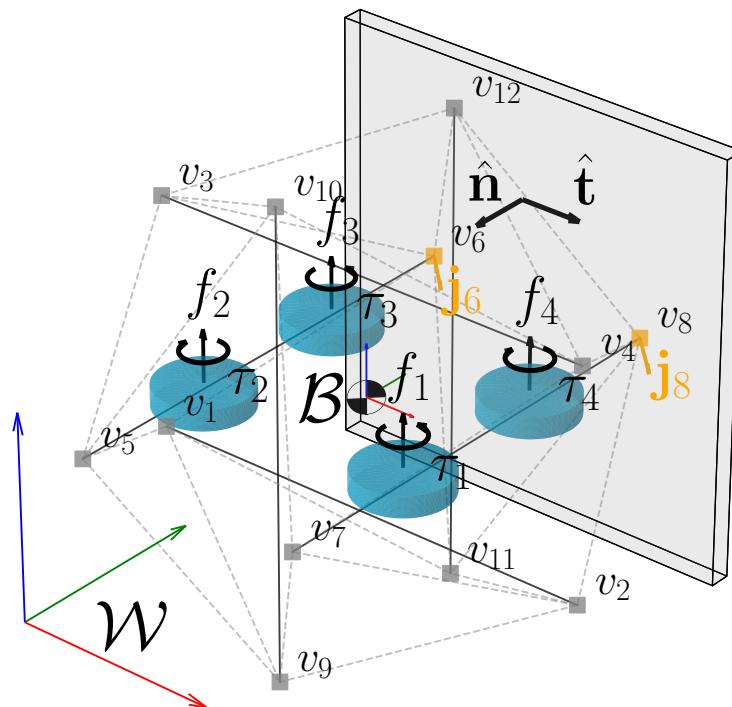
While these studies establish a foundation for collision-resilient drones, the majority focus on impact survival rather than autonomous path recovery from high-speed collisions. Furthermore, methods that adjust flight paths rely on computationally expensive mapping pipelines. This paper addresses both challenges, proposing a high-speed collision recovery strategy that enables drones to withstand impacts and adapt their trajectory with minimal computational cost, preventing repeated collisions.

### 3. Modeling

The model used is the conventional quadrotor, with its Center-of-Mass (CoM) encased in a protective icosahedral shell, as depicted in Figure 2. In free flight, the system dynamics follow the standard equations of motion for a quadrotor, as used in works such as [35]:

$$\ddot{\mathbf{p}} = m^{-1} \mathbf{f}_M + \mathbf{g} \quad \dot{\boldsymbol{\omega}} = \mathbf{I}^{-1} (\boldsymbol{\tau}_M - \boldsymbol{\omega} \times \mathbf{I} \boldsymbol{\omega}) \quad (1)$$

For all nomenclature, please refer to Table 1. The vehicle is controlled using a cascaded control approach as described in [30]. A position reference is converted into an acceleration reference, which is then translated into total thrust and attitude commands. These are subsequently tracked by a rate controller.



**Figure 2.** A free-body diagram of the forces and torques acting on the MAV as well as the frames used to model it colliding with the environment. The tensegrity drone with its twelve nodes ( $v_1$  through  $v_{12}$ ) is controlled through its control forces and torques ( $f_1$  and  $\tau_1$  through  $f_4$  and  $\tau_4$ ) and experiences the collision impulses  $j_i$  at the  $i$ -th vertex.

**Table 1.** Nomenclature.

Symbol	Definition
$\mathcal{W}, \mathcal{B}$	World and body frame.
$m \in \mathbb{R}, \mathbf{I} \in \mathbb{R}^{3 \times 3}$	Mass and Inertia matrix of the MAV.
$\zeta = \begin{bmatrix} \mathbf{p} & \mathbf{R} \end{bmatrix}^T \in \mathbb{SE}(3)$	MAV pose (position and attitude).
$\dot{\zeta} = \begin{bmatrix} \mathbf{v} & \boldsymbol{\omega} \end{bmatrix}^T \in \mathbb{R}^6$	MAV rates (linear and angular).
$f_i, \tau_i \in \mathbb{R}; \mathbf{f}_M, \boldsymbol{\tau}_M \in \mathbb{R}^3$	Force and torque produced by the $i$ -th motor and total motor force and torque.
$\mathbf{r}_i \in \mathbb{R}^3$	The vector from the MAV's center to the $i$ -th icosahedron vertex $v_i$ .
$\mathcal{C} = [\mathcal{C}_1 \dots \mathcal{C}_i] \in \mathbb{B}^{12}$	Vector of all binary contact signals.
$e, \mu \in \mathbb{R}$	Coef. of restitution and friction coef.
$\mathbf{j}_i, \mathbf{j} \in \mathbb{R}^3$	Contact impulse on the $i$ -th vertex and total impulse on the MAV.
$\hat{\mathbf{n}}, \hat{\mathbf{t}} \in \mathbb{R}^3$	Surface normal and tangential direction.
$\mathbf{v}^+, \mathbf{v}^- \in \mathbb{R}^3$	Post- and pre-collision linear velocities.
$\boldsymbol{\omega}^+, \boldsymbol{\omega}^- \in \mathbb{R}^3$	Post- and pre-collision angular rates.

We assume that contact can only occur at the twelve vertices of the icosahedral shell, with the vertices treated as point contacts. While in theory also the strings forming the edges of the icosahedron could be points of contact, in practice however these strings are soft and do not significantly contribute to collisions. Furthermore, the vertices are the most likely points of contact, as the vertices form a convex structure around the rest of the body; i.e., these are the points with maximal distance from the CoM. Therefore, unless the MAV comes into contact with very thin objects precisely aligned with an edge, the vertices will be the first points of contact. By assuming the vertices as the only possible contact points, the following model becomes computationally much more efficient: only individual points need to be queried for collision and not full meshes. Given the convex structure, this only makes a small sacrifice in accuracy as in most cases these nodes are indeed the contact points. Sensors detect the contact as a binary signal  $\mathcal{C}$ . Following the impulse-based collision modeling approach from [36] we define the impulse as the effect of an instantaneous contact force  $f_{\text{cntc}}$ , meaning  $\mathbf{j} := \lim_{\Delta t \rightarrow 0} f_{\text{cntc}} \Delta t$ . Therefore, in this model, a collision causes an instantaneous change in momentum, governed by a restitution coefficient  $e$  and a friction coefficient  $\mu$ . The change in linear velocity due to the total impulse is given by  $\mathbf{v}^+ = \mathbf{v}^- + \frac{\mathbf{j}}{m}$ . Assume a single vertex  $i$  is in collision with the pre- and post-collision velocities:

$$\mathbf{v}_i^- = \mathbf{v}^- + \boldsymbol{\omega}^- \times (\mathbf{R} \mathbf{r}_i) \quad \mathbf{v}_i^+ = \mathbf{v}^+ + \boldsymbol{\omega}^+ \times (\mathbf{R} \mathbf{r}_i) \quad (2)$$

Then we can derive the impulse  $\mathbf{j}_i$  via

$$\mathbf{v}_i^+ = \mathbf{v}_i^- + \|\mathbf{j}_i\| \left( \frac{\hat{\mathbf{n}}}{m} + \mathbf{I}^{-1}[(\mathbf{R} \mathbf{r}_i) \times \hat{\mathbf{n}}] \times \mathbf{R} \mathbf{r}_i \right) \quad (3)$$

Using the system restitution property  $\hat{\mathbf{n}} \circ \mathbf{v}_i^+ = -e \hat{\mathbf{n}} \circ \mathbf{v}_i^-$  and the relationship between the normal and frictional impulses  $\|\hat{\mathbf{t}} \circ \mathbf{j}_i\| = \mu \|\hat{\mathbf{n}} \circ \mathbf{j}_i\|$ ,

$$\mathbf{j}_i = -\frac{(1+e)\hat{\mathbf{n}} \circ \mathbf{v}_i^-}{m^{-1} + \hat{\mathbf{n}} \circ (\mathbf{I}^{-1}(\mathbf{R} \mathbf{r}_i \times \hat{\mathbf{n}}) \times \mathbf{R} \mathbf{r}_i)} (\hat{\mathbf{n}} + \mu \hat{\mathbf{t}}) \quad (4)$$

The CAD model provides the drone's mass and inertia, while a least-squares optimization on a dataset of pre- and post-collision velocities during unactuated impacts estimates the restitution and friction coefficients. The surface normal is assumed to align with the tensegrity beam on which the active contact location lies. Please note that this assumption does not always hold, but the dominant collision effects align with these axes, justifying its use. The friction is modeled as dynamic; i.e., we assume a sufficiently small friction cone, such that the node experiences sliding friction. The overall change in linear and angular velocity is then the normalized sum of the contribution of all  $k$  active vertices:

$$\Delta v = \frac{1}{k} \sum_{i=1}^k m j_i \quad \Delta \omega = \frac{1}{k} \sum_{i=1}^k I (\mathbf{R} \mathbf{r}_i \times \mathbf{j}_i) \quad (5)$$

Normalization is required, as otherwise the estimated post-collision state would violate energy conservation laws with respect to the pre-collision state. Please note that this naturally handles multiple points of contact happening in close temporal proximity to each other as the impulses behave additively.

## 4. Collision-Inclusive Estimation and Control

This section details the proposed approach for path representation and recovery after a collision. It comprises two key components: a collision-aware state estimator, which explicitly models collisions to improve estimation during impact, and a path recovery strategy that stabilizes the MAV post-collision while adjusting the path to avoid re-colliding with the same object.

### 4.1. Collision-Inclusive State Estimator

The collision-inclusive state estimator is a Kalman Filter (KF), as introduced in [37]. Additionally it uses a switching prediction function and assumes that, on average, the MAV follows the commanded angular velocity. This enables the estimator's angular velocity estimate to decay toward the commanded velocity with a time constant  $\tau_\omega$ . At each timestep  $k$ , with duration  $\Delta t$ , and given the commanded acceleration  $\mathbf{a}_{k,cmd}$  and angular velocity  $\omega_{k,cmd}$ , the prediction step proceeds as follows:

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}_k \Delta t + \mathbf{a}_{k,cmd} \frac{\Delta t^2}{2} \quad (6)$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + (1 - \kappa) \Delta t \mathbf{a}_{k,cmd} + \kappa \Delta \mathbf{v}_{\mathcal{C}} \quad (7)$$

$$\mathbf{R}_{k+1} = \mathbf{R}_k [\omega \Delta t] \times \quad (8)$$

$$\omega_{k+1} = (1 - \kappa) \left( e^{-\frac{\Delta t}{\tau_\omega}} \omega_k + \left( 1 - e^{-\frac{\Delta t}{\tau_\omega}} \right) \omega_{k,cmd} \right) + \kappa \Delta \omega_{\mathcal{C}} \quad (9)$$

$$\kappa_{k+1} = \begin{cases} 1 & \text{if } \mathcal{C}_{i,k} \text{ for any } i \in [1, 12] \\ 0 & \text{otherwise} \end{cases}, \quad (10)$$

where  $\Delta \mathbf{v}_{\mathcal{C}}$  and  $\Delta \omega_{\mathcal{C}}$  denote the changes in linear and angular velocity due to a collision at contacts  $\mathcal{C}$ ; see Equation (5). They only contribute if there is at least one active contact, i.e., if the auxiliary variable  $\kappa$  is one. This structure allows the estimator to capture the discontinuous jumps in state space (velocities and rates) that occur during contact. As a result, unlike a standard KF that requires long re-convergence times, the estimator output is representative of the true state earlier after a collision, allowing the controller to perform more effectively in the crucial post-collision stabilization phase. The update step on position and attitude measurements follows the standard form, elaborated in detail in [37]. We tune

the KF's matrices by setting the diagonal entries of the measurement noise matrix to the system's measurement accuracy and iteratively adapt the process noise matrix's diagonal entries to achieve a fast convergence without too much amplification of sensor noise [38].

#### 4.2. Collision-Inclusive Path Recovery

The path recovery procedure involves two steps: a reflexive, immediate action to stabilize the MAV and prevent a crash, followed by a path adjustment to avoid future collisions with the same object. The following sections describe both behaviors in detail.

##### 4.2.1. Collision Recovery

The reflexive collision recovery behavior operates at the highest level of the cascading controller, specifically the position controller. Upon detecting a collision (i.e.,  $C_i$  for any  $i \in [1, 12]$ ), the controller computes a new reference position  $p_{\text{rec}}$ , which is the current position shifted along the surface normal  $\hat{n}$ . The surface normal is assumed to align with the tensegrity beam where the triggered contact sensor is mounted:

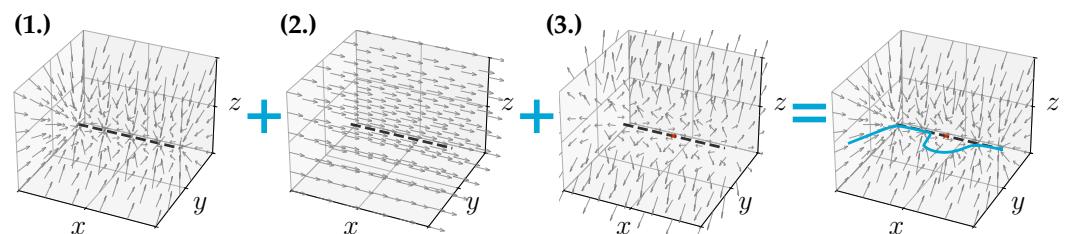
$$p_{\text{rec}} = p - \alpha \hat{n} \quad \text{where} \quad \alpha = \sqrt{\|v\|} \quad (11)$$

with  $\alpha$  being a scaling factor that determines how far the recovery position should be from the surface. Please note that the assumed surface normal may be inaccurate as the MAV might collide with a slanted wall. However, the sensor covers large offsets of a contact plane along two axes. Consequently, even when the normal is offset by large angles the escape direction still drives the MAV away from the obstacle toward free space.

##### 4.2.2. Path Adjustment

In this work we represent any path in the form of a vector field, similar to [39]. Thereby the goal is to construct a vector field such that by integrating along the field one converges to the original path and moves along it while also avoiding known objects. The vector field is initialized without any known objects, and only after the reflexive collision recovery procedure in the previous section has succeeded we add the location of the triggering sensor as a new known obstacle. The vector field function consists of three components, as shown in Figure 3:

- i. A vector pointing to the nearest point on the path.
- ii. The velocity vector at the nearest point on the path.
- iii. A repulsive force from known obstacles.



**Figure 3.** An example trajectory being represented by a vector field. For any point in space, the vector field consists of three contributions: (1.) An attractor to the nearest point on the trajectory, (2.) the velocity vector of the trajectory at that nearest point, and (3.) a repulsive potential at known collision locations. The grey arrows represent the vector field components, the black dashed line the desired path, the red cross a known obstacle and the blue path the resulting path when following the vector field.

The next sections detail each component.

### Finding the Nearest Point

For an arbitrary path parametrized by a twice-differentiable function  $\mathbf{h}(\tau)$  where  $\tau \in [0, 1]$  defines the path from start to finish, and for any arbitrary location in space  $\mathbf{x}$ , the nearest point on the trajectory can be found by solving the following optimization:

$$\tau_{\min} = \arg \min_{\tau} \|\mathbf{h}(\tau) - \mathbf{x}\|^2; \quad \text{s.t. } \tau \in [0, 1] \quad (12)$$

We solve this in two steps: First, we uniformly sample  $n$  points along  $\mathbf{h}(\tau)$  within the specified range and use a brute-force search to find the nearest point. Subsequently, we hot-start a Newton method optimization using the brute-force solution as an initial guess to refine the solution over  $i$  iterations. The Newton method update for the optimization problem above is then

$$\tau_{k+1,\min} = \tau_{k,\min} - \frac{(\mathbf{h} - \mathbf{x})^T \mathbf{h}'}{\mathbf{h}'^T \mathbf{h}' + (\mathbf{h} - \mathbf{x})^T \mathbf{h}''} \quad (13)$$

where we have omitted the dependency on  $\tau$  for readability. In practice, coarse sampling of the trajectory ( $n = 10$ ) and a few optimization iterations ( $i = 3$ ) are sufficient.

### The Vector Field

Given the differential function  $\mathbf{h}(\tau)$  and the list of  $m$  obstacle locations  $\mathcal{C}$  the goal is to construct a vector field that contains all three components described above. We start by considering the repulsion from obstacles. Rather than acting as a simple point source, the obstacle's contribution should be perpendicular to the desired path velocity to avoid interfering with forward progress. Therefore, we define the weighted ( $\kappa_c$ ) collision contribution for a collision location  $\mathbf{c}_j$  as the projection of the point source repulsion into the plane whose normal is the velocity vector of the vector field without the collision contribution spanned by the tangential vectors  $\hat{\mathbf{t}}_1(\tau)$  and  $\hat{\mathbf{t}}_2(\tau)$ :

$$\mathbf{g}_{\text{collision},j}(\mathbf{x}) = \left[ \frac{\kappa_c(\mathbf{x} - \mathbf{c}_j)}{\|\mathbf{x} - \mathbf{c}_j\|^3} \circ \hat{\mathbf{t}}_1(\tau_{\min}) \right] \hat{\mathbf{t}}_1(\tau_{\min}) + \left[ \frac{\kappa_c(\mathbf{x} - \mathbf{c}_j)}{\|\mathbf{x} - \mathbf{c}_j\|^3} \circ \hat{\mathbf{t}}_2(\tau_{\min}) \right] \hat{\mathbf{t}}_2(\tau_{\min}) \quad (14)$$

The positional and velocity components are then defined as a linear and reciprocal function of the distance to  $\mathbf{h}(\tau_{\min})$ , respectively, completing the overall vector field function,

$$\mathbf{g}(\mathbf{x}) = v_{GF} \left[ \kappa_p(\mathbf{h}(\tau_{\min}) - \mathbf{x}) + \frac{\kappa_v}{\|\mathbf{h}(\tau_{\min}) - \mathbf{x}\|} \mathbf{h}'(\tau_{\min}) + \sum_{k=1}^m \mathbf{g}_{\text{collision},k}(\mathbf{x}) \right]_{\text{norm}} \quad (15)$$

where  $\kappa_p = 10$ ,  $\kappa_v = 0.1$ , and  $\kappa_c = 2.5$  are respectively the weights that control the rate of convergence to the desired path, the influence range of the path velocity, and the avoidance distance from known obstacles, and  $[\cdot]_{\text{norm}}$  indicates the normalization operator. In other words, a higher  $\kappa_p$  to  $\kappa_v$  ratio increases the convergence rate to the path, while the reverse prioritizes velocity convergence. Similarly, a larger  $\kappa_c$  relative to  $\kappa_p$  and  $\kappa_v$  increases the obstacle avoidance radius. The normalized sum of all components is then scaled by a constant, user-defined velocity  $v_{GF}$ . At timestep  $k$ , given the previous reference, the new reference position  $\mathbf{p}_{\text{des}}$ , velocity  $\mathbf{v}_{\text{des}}$ , and yaw  $\psi_{\text{des}}$  maintaining a forward orientation are

$$\mathbf{p}_{k+1,\text{des}} = \mathbf{p}_{k,\text{des}} + \Delta t \mathbf{g}(\mathbf{p}_{k,\text{des}}) \quad (16)$$

$$\mathbf{v}_{k+1,\text{des}} = \mathbf{g}(\mathbf{p}_{k,\text{des}}) \quad (17)$$

$$\psi_{k+1,\text{des}} = \arctan 2(v_{k+1,\text{des};y}, v_{k+1,\text{des};x}) \quad (18)$$

### Proof of Convergence

To prove the convergence of the movement towards the desired path we make two key assumptions: 1. At least one point on the path is negligibly affected by obstacles, since convergence to a fully blocked path is infeasible and cannot be proven. 2. The desired path is twice-differentiable, ensuring a smooth nearest-point projection. We then define the Lyapunov candidate function:

$$V(\mathbf{x}) = \frac{1}{2} \|\mathbf{x} - \mathbf{h}(\tau_{\min}(\mathbf{x}))\|^2 \quad (19)$$

$V(\mathbf{x})$  is a suitable Lyapunov function as it is radially unbounded, positive definite when not converged, and zero at convergence. Taking the derivative,

$$\dot{V}(\mathbf{x}) = (\mathbf{x} - \mathbf{h}(\tau_{\min}))^T (\dot{\mathbf{x}} - \mathbf{h}'(\tau_{\min}) \dot{\tau}_{\min}) \quad (20)$$

Using the optimality condition  $(\mathbf{x} - \mathbf{h}(\tau_{\min}))^T \mathbf{h}'(\tau_{\min}) = 0$ ,

$$\dot{V}(\mathbf{x}) = (\mathbf{x} - \mathbf{h}(\tau_{\min}))^T \dot{\mathbf{x}} \quad (21)$$

Substituting system dynamics  $\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x})$  and omitting the dependence on  $\tau_{\min}$  for clarity,

$$\dot{V}(\mathbf{x}) = (\mathbf{x} - \mathbf{h})^T v_{GF} \left[ \kappa_p(\mathbf{h} - \mathbf{x}) + \frac{\kappa_v}{\|\mathbf{h} - \mathbf{x}\|} \mathbf{h}' + \sum_{k=1}^m \mathbf{g}_{\text{collision},k}(\mathbf{x}) \right]_{\text{norm}} \quad (22)$$

Using assumption 1, i.e., wlog  $\mathbf{g}_{\text{collision},k}(\mathbf{x}) \approx 0 \forall k$  and the optimality condition,

$$\dot{V}(\mathbf{x}) = - \frac{v_{GF} \kappa_p (\mathbf{h} - \mathbf{x})^T (\mathbf{h} - \mathbf{x})}{\|\kappa_p(\mathbf{h} - \mathbf{x}) + \kappa_v \|\mathbf{h} - \mathbf{x}\|^{-1} \mathbf{h}'\|} \quad (23)$$

This is a negated quadratic form, i.e., it will always be smaller than zero guaranteeing convergence, if all obstacles are sufficiently far away. Furthermore, we can write this as

$$\dot{V}(\mathbf{x}) = - \frac{v_{GF} \kappa_p}{\frac{1}{2} \|\kappa_p(\mathbf{h} - \mathbf{x}) + \kappa_v \|\mathbf{h} - \mathbf{x}\|^{-1} \mathbf{h}'\|} V(\mathbf{x}) \quad (24)$$

and using the AM-GM inequality we can approximate this as

$$\dot{V}(\mathbf{x}) \leq - \frac{v_{GF} \kappa_p V(\mathbf{x})}{\sqrt{\kappa_p \kappa_v \|\mathbf{h}'\|}} = -v_{GF} \sqrt{\frac{\kappa_p}{\kappa_v \|\mathbf{h}'\|}} V(\mathbf{x}) \quad (25)$$

Assuming  $\|\mathbf{h}'\|$  is upper-bounded by  $h'_{\max}$  this results in

$$\dot{V}(\mathbf{x}) \leq -v_{GF} \sqrt{\frac{\kappa_p}{\kappa_v h'_{\max}}} V(\mathbf{x}) = -\beta V(\mathbf{x}) \quad (26)$$

which guarantees an exponential convergence rate with the convergence rate  $\beta$ , defined by the heuristic variables of the vector field, matching the intuition given when introducing the vector field.

### Computational Complexity

Since this procedure runs at each controller iteration on modest onboard hardware, efficiency is a key design goal. The most computationally expensive task is finding the nearest point on the path, with a linear complexity  $\mathcal{O}(n)$  in the number of discrete points sampled  $n$  in the initial step. For highly nonlinear paths,  $n$  may be large, but for practical

paths—like splines, low-order polynomials, or trigonometric functions—a small, constant  $n$  suffices, therefore reducing the overall complexity to  $\mathcal{O}(1)$ . On the hardware used in this work (see Section 5) the chain to trigger the recovery maneuver is executed in a single logical cycle, i.e., in 2 ms, allowing the system to respond to a collision within the same time.

## 5. Hardware Implementation

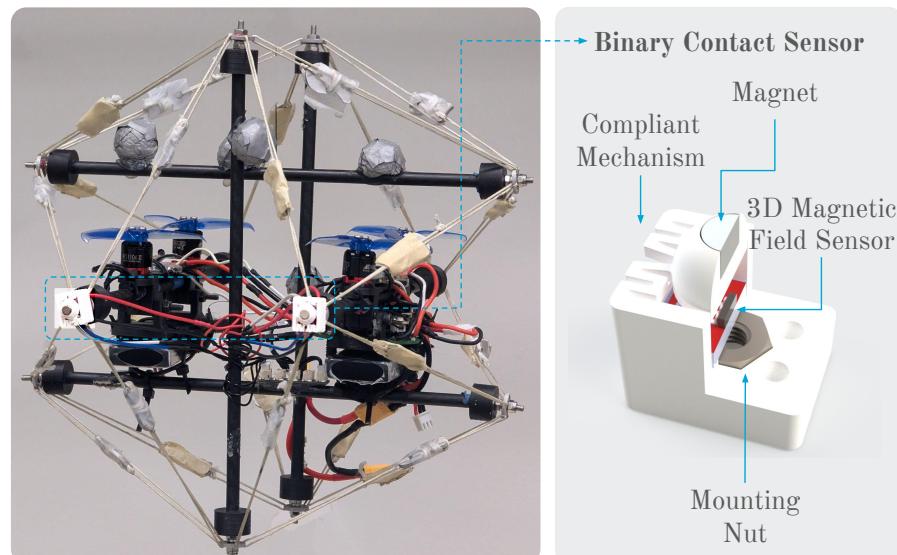
This section presents the physical prototype developed to approximate the model introduced in Section 3. Specifically, we detail the design of the contact sensor and briefly review the construction of the collision-resilient tensegrity vehicle.

### 5.1. Contact Sensor

The main purpose of the contact sensor is to trigger a binary signal when the respective corner of the tensegrity vehicle makes contact. It is designed for high-sensitivity contact detection from all directions, physical robustness, and minimal weight. At the same time since we only require a binary signal there are no range or significant resolution requirements. As shown on the right in Figure 4 the sensor consists of a 3D-printed housing with a compliant mechanism connected to a dome, a button magnet inside the dome, and a 3D Hall-effect sensor positioned below. The compliant mechanism includes four spring-like connections between the casing and the dome, allowing displacement along all three axes while returning the dome to its original position when no external force is applied. The housing is printed using TPU material making it flexible and robust. When in contact, the contact force displaces the dome, altering the magnetic field measured by the Hall sensor from its nominal value. Therefore, the binary output for sensor is

$$C_i = (||B_i - B_{nom,i}|| < \epsilon_{B,i}) \quad (27)$$

By selecting an appropriate threshold  $\epsilon_B$ , the sensor is calibrated to provide high sensitivity while minimizing false positives. One of the contact sensors weighs 3.5 g and is smaller than 8.5 cm  $\times$  6 cm  $\times$  5 cm.



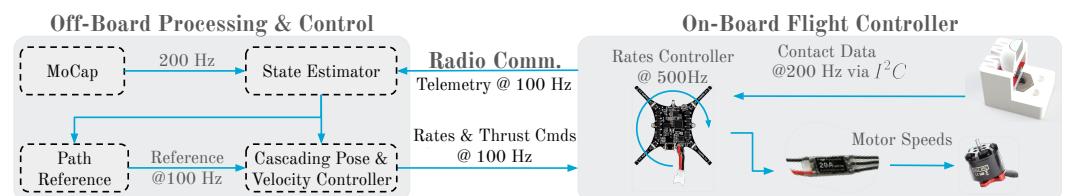
**Figure 4.** The drone (left) and a section view of the contact sensor (right). During contact, the compliant mechanism allows the magnet to move relative to the 3D magnetic field sensor without damage. The sensor measures the magnetic field and triggers contact when the deviation from the nominal field exceeds a threshold.

## 5.2. Collision-Resilient MAV

The collision-resilient MAV takes on the form of an icosahedron tensegrity vehicle as introduced in [17,40], which is shown to survive collisions up to 7.8 m/s. Tensegrity structures consist of rigid bodies suspended within a tension network. As these structures distribute external loads through tension and compression rather than bending, they are exceptionally impact-resistant. For aerial vehicles, which have fragile components like propellers and electronics that are vulnerable to collisions and whose defect is catastrophic, such a tensegrity structure provides a protective shell.

The physical prototype carries a Crazyflie 2.1 (Manufacturer: Bitcraze, Malmö, Sweden) as a flight controller, four Little Bee 20 A ESCs and EMAX RS1106 II 4500 KV motors (Manufacturer: EMAX, Dongguan, China) with  $2.5 \times 4$  three blade propellers. As a proof-of-concept the prototype carries two contact sensors at the front due to constraints on the I<sup>2</sup>C bus of the Crazyflie 2.1 (c.f. Figure 4). We choose the front two nodes of the icosahedron to place the contact sensors as these nodes are most likely to make contact during typical flight maneuvers, i.e., forward flight.

Figure 5 showcases the overall software architecture: The onboard flight controller runs a rate controller that tracks external rate commands while also reading out the contact sensors, triggering the reflexive behavior, and providing this information via radio telemetry. The off-board computer runs a motion-capture-assisted state estimator and a cascading controller that tracks the references produced by the vector-field-based path planner. The motion-capture system provides pose measurements at 200 Hz, while the state estimator yields a full state estimate (pose and twist) required for the cascading controller. Overall the system deviates from the model introduced in Section 3 by not having contact sensors on all icosahedron nodes as well as exhibiting time delay between the contact occurrence and the off-board computer receiving the correct binary signal.



**Figure 5.** The overall system operates as follows: Onboard the MAV, the flight controller runs a rate controller that tracks external body rate references, reads contact sensor data, and provides telemetry. Offboard, the vector-field-based path planner computes reference positions and velocities, which are tracked by a motion-capture-assisted state estimator and a cascaded controller [40].

## 6. Experiments

To evaluate the proposed approach, we validate it in both simulation and real-world experiments. The simulations allow testing the theoretical limits of the proposed approach under various conditions, while the physical prototype demonstrates feasibility and performance in practice.

### 6.1. Simulation

The simulated experiments test the limits of collision recovery at high speeds and the applicability of the full path recovery in cluttered environments. The simulation environment is implemented as a C++ program using a fixed-step explicit second-order integration scheme. We choose  $\Delta t = 2$  ms as it matches the onboard frequency of the real prototype. For more details on the implementation, please refer to the project page.

### 6.1.1. Contact Model

To validate the proposed approach in simulation and assess its robustness across numerous trials, a representative contact model is needed to simulate collisions. As outlined in Section 3, we focus exclusively on the twelve nodes of the icosahedron, each assumed to make point contact with the environment. Thus, we adopt a simplified version of the compliant contact model introduced in [41,42]. Specifically, the contact force  $f_{\text{cntc}}$  acting on each node  $i$  during contact is computed as a function of the surface normal  $\hat{n}$ , tangential direction  $\hat{t}$ , the penetration depth  $d$ , node velocity  $v_i$ , and the stiffness, damping and friction coefficients  $k_p, k_d$ , and  $\mu$ :

$$f_{\text{cntc},i} = f_{\hat{n}} + f_{\hat{t}} \quad (28)$$

$$f_{\hat{n}} = (k_p d - k_d ||v_i \circ \hat{n}||) \hat{n} \quad (29)$$

$$f_{\hat{t}} = -\mu ||f_{\hat{n}}|| \text{sgn}(v_i \circ \hat{t}) \hat{t} \quad (30)$$

The total contact force  $f_{\text{cntc}}$  and contact torque  $\tau_{\text{cntc}}$  on the simulated MAV from  $m$  active contact points then follows

$$f_{\text{cntc}} = \sum_{k=1}^m f_{\text{cntc},k} \quad \tau_{\text{cntc}} = \sum_{k=1}^m (\mathbf{R}r_k) \times f_{\text{cntc},k} \quad (31)$$

Note that this contact model achieves higher fidelity compared to the contact model introduced in Section 3. This allows us to benchmark the proposed approach in simulation as shown in the following sections.

### 6.1.2. Results

Two scenarios are validated using Monte Carlo simulations: the first evaluates the system's ability to recover from high-speed collisions compared to non-tactile approaches, while the second assesses the reliability of the overall path-recovery procedure.

## High-Speed Collision Recovery

Table 2 displays the collision recovery success for different velocities across three controllers: (1) a contact-agnostic cascading controller combined with a KF as briefly introduced in Section 3, (2) the same estimator and low-level controller as (1), but with the recovery maneuver introduced in Section 4.2.1 triggered when the accelerometer norm exceeds a threshold, and (3) the proposed approach. For each velocity and controller, the Monte Carlo simulation consists of 100 trials. A recovery is considered successful if the MAV does not touch the ground during the recovery maneuver.

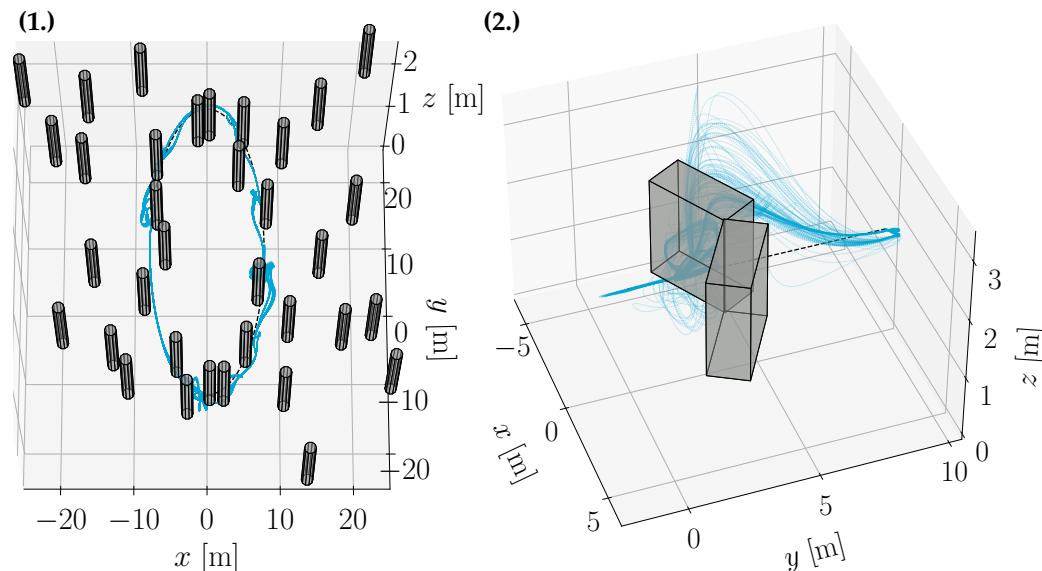
**Table 2.** Monte-Carlo evaluation: collision recovery. Monte Carlo simulations (100 trials each) at various collision velocities for three controllers: collision-agnostic, accelerometer-based recovery, and the proposed tactile approach. A trial succeeds if the drone does not touch the ground during recovery, and a Monte Carlo experiment succeeds if no trials fail. Please note that this tests the algorithmic limits, which are different from the physical limitations as addressed in Section 6.3.

As expected, the collision-agnostic approach only manages recovery at very low speeds, crashing at velocities greater than 0.5 m/s. The accelerometer-based approach actively triggers a recovery maneuver and performs significantly better, successfully recovering at speeds up to 3.0 m/s. Beyond this velocity, the non-contact-aware Kalman filter fails to track the system state. In particular, during a collision, the velocity and rate estimates initially continue in their pre-impact direction, causing the state estimate to diverge and leading to incorrect control inputs, altitude loss, and eventual crash at velocities over 3.0 m/s. In contrast, the proposed approach successfully recovers in all trials, even at velocities up to 8.0 m/s, due to improved state estimation from low-latency contact data.

### Path-Recovery Procedure

In this experiment, the MAV is commanded to take off from an initial position that is randomized for each trial by uniformly sampling from within a 50 cm cube. After takeoff we evaluate one of two scenarios: (1.) The MAV is instructed to follow an elliptical path in a cluttered environment at a speed of 4 m/s. However, this path is obstructed by cylinders along the path and randomly placed in the vicinity. (2.) The MAV is instructed to follow a straight path, blocked by a concave obstacle.

Figure 6 displays the flight paths for all 100 trials. Both scenarios show how the MAV successfully recovers from the collision in all trials. In scenario (1.), the MAV avoids obstacles along the path until a new collision occurs and continues to avoid known obstacles in subsequent passes. In scenario (2.), it demonstrates the ability to escape large obstacles with concave traps using its vertical mobility. Note that after a single collision (e.g., with a concave obstacle like in Figure 6), the vector field may not immediately guide the MAV around the obstacle. However, thanks to its collision resilience, the MAV can re-collide and gather more contact points until the MAV has a sufficiently complete representation of the obstacle and an escape route emerges. By continuously incorporating the collision location into the vector field, the MAV naturally bypasses the obstacle and eventually re-converges to the original path.

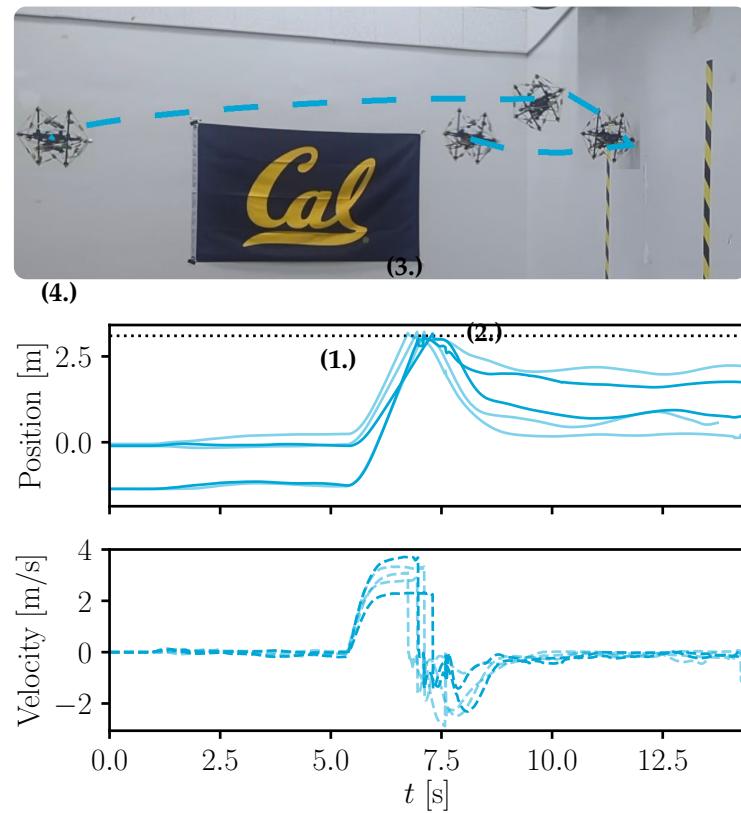


**Figure 6.** Monte Carlo simulations (100 trials each) of (1.) a MAV following a path in a cluttered environment using the proposed path recovery and adaptation approach, and (2.) a MAV following a straight path blocked by a concave obstacle. The MAV follows the desired path (black dashed line), collides with obstacles, recovers, and reconverges while avoiding known obstacles in subsequent passes. In (1.), multiple collisions and recoveries occur across trials, leading to slightly different avoidance strategies, while in (2.), several collisions may be required to generate enough repulsive potential to redirect the MAV. Animations are available on the project page.

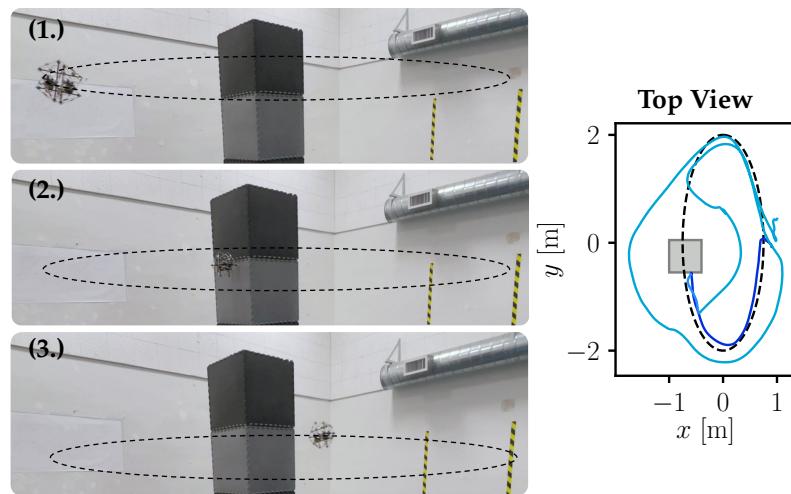
## 6.2. Flight Experiments

Using the physical prototype introduced in Section 5, we replicated the experiments from the Monte Carlo simulations to validate our proposed approach. Figure 7 shows stills from the experiment and the position and linear velocity along the approach direction for multiple trials of high-speed collision recovery. The results demonstrate that the MAV successfully recovers from collisions at very high speeds ranging from 2.3 m/s to 3.7 m/s. Additionally, in the videos available on the project page, one can observe that the MAV consistently finds a stable recovery position and maintains stabilization at that location across all trials.

Figure 8 presents stills from the video and a top-down view of the complete recovery and path adaptation pipeline executed on the physical prototype. We commanded the MAV to follow an elliptical path simulation, obstructed by a single 60 cm cubical object. The figure illustrates that after takeoff, the MAV converges to the path and follows it closely until a collision occurs. Afterwards, it successfully performs the recovery maneuver, stabilizing in front of the obstacle. The MAV then resumes the path, avoiding the obstacle on the inside of the ellipse before re-converging with the original path. During the next pass, having added the obstacle to the vector field, the MAV avoids it on the outside of the ellipse before returning to its initial takeoff position. This demonstrates the successful post-collision adjustment of the vector field and highlights the effective integration of the collision recovery procedure into the full pipeline.



**Figure 7.** (Top) High-speed recovery maneuver: (1.) approaching the obstacle, (2.) updating velocity estimate upon collision, (3.) triggering reflexive recovery maneuver, and (4.) hovering at the recovery position. (Bottom) The position and velocity of the physical MAV in approach direction during collision recovery from five high-speed collisions with velocities up to 3.7 m/s. Videos are provided on the project page.



**Figure 8.** The physical MAV (1.) follows a nominal elliptic trajectory (start until collision; dark blue), (2.) recovers from a collision (collision until recovery; brighter blue), and (3.) adapts its path in the next pass to avoid re-colliding with the same object (after recovery; light blue). During the second pass, the MAV deviates from the desired path even on the opposite side of the ellipse due to a large repulsive gain  $\kappa_p$  and the limited flight arena size, which extends the obstacle’s influence across the ellipse. A video is available on the project page.

### 6.3. Hardware Limitations and Model Mismatch

In this section we want to explicitly address the differences between the simulated and physical experiments. In particular, we want to highlight the design choices that lead to a difference between the theoretical and practical limits but also address the model mismatch between the simplified model and the real-world dynamics.

As mentioned in Section 5, compared to the idealized contact model used in simulation (Section 3), the prototype uses only two contact sensors, while the idealized version senses contact at all twelve nodes. While this simplification is particularly significant for ceiling and ground collisions, the majority of collisions happen in the direction of flight. Therefore, by using the front sensors, and by aligning the MAV with the velocity vector, most collision events are still detected. Hereby the limitation lies entirely within the chosen hardware components. Additional sensors would not increase the computational complexity of the proposed approach, as it would only add one bit of transferred data and not influence the estimation computation. I.e., in future iterations with optimized hardware that allow for more sensors on its I<sup>2</sup>C-bus, it is straightforward to increase the number of sensors.

Besides implementing a different number of sensors in the physical prototype, the model also assumes only sliding friction, while in reality sticking friction can occur. In practice, however, the contact time during collisions is very short such that both types of friction behave similarly, albeit exhibiting different friction coefficients. However, since we estimated the friction coefficient from data, this effect is captured by the system identification process.

Furthermore, there is a noticeable discrepancy between the maximal recovery velocities in simulation (Table 2) and on the physical system (Figure 7). While the simulation provides a theoretical upper limit for the performance of the proposed approach, hardware limitations still constraint the results with the physical prototype. In particular, hardware failures occur before the collision-aware estimator’s or the recovery controller’s limits are reached, alongside the cumulative effect of non-modeled dynamics in simulation. At high velocities, collision recovery commands the motors to generate large thrust values, leading to a surge in current draw from the onboard batteries. Since the batteries are optimized for minimal weight, this causes a significant voltage drop—sometimes below the flight

controller's minimum threshold—resulting in a momentary shutdown and a crash. These "brownouts" could be mitigated by using higher-capacity batteries, but this would increase the overall weight of the system, thereby reducing the maximal recovery speed. We choose to accept this tradeoff as we prioritize exploring the limit of the proposed approach. Furthermore, at very high speeds, the compliant mechanism of the tactile sensor compresses too rapidly, causing the top part to strike the underlying integrated circuit and breaking its surface-mount solder joints. This renders the sensor inoperative and leads to system failure. Lastly, as seen in Figure 4, the batteries are mounted with a weak axis parallel to the drone's forward direction, leading to them slipping out during high-speed impacts and consequently disconnecting the electronics.

#### 6.4. Comparison

In the previous sections we have demonstrated the performance of the proposed method with respect to collision resilience and path recovery. In the following we want to rigorously compare our approach to conventional MAVs as well as related work.

Conventional MAVs, equipped with propeller guards but agnostic to collisions, are unable to recover from collisions at speeds beyond 0.5 m/s, as shown in Table 2, implying a seven-fold improvement of our methodology compared to the naive baseline. This is also seen in contact-aware methods without specific hardware components [22] where collision velocities up to 0.7 m/s are reported. Table 3 summarizes the performance of the proposed approach compared to related work. It shows that while maintaining one of the lowest overall system weights, the proposed approach achieves one of the highest experimental recovery velocities. Furthermore, the proposed approach relies on only simple binary tactile sensing for collision recovery compared to a more complex compliance, which allows the additional hardware added to the system to be small and very lightweight.

**Table 3.** Comparison of collision recovery performance in related works. The proposed method achieves one of the highest experimental recovery velocities by combining tactile sensing with collision-aware estimation to enhance post-impact stability. At the same time the proposed system achieves this while maintaining one of the lowest overall system weights, making it suitable for a wide range of MAVs.

Reference	Approach / Structure	Recovery Trigger	Sensing Modality	Max. Recovery Velocity [m/s]	Overall System Weight [kg]
Briod et al. [16]	Collision-resilient flying robot (GimBall) with gimbal-suspended inner frame	Passive mechanical decoupling	IMU	~1.5	0.385
Zha et al. [17]	Icosahedral tensegrity structure for collision resilience	Passive mechanical	IMU	Survives >7.8 (no in-flight recovery)	0.30
Liu et al. [18]	Impact-resilient quadrotor (ARQ) with compliant arms	Compliant Arm Deformation	IMU + compliance deformation	~2.6	1.419
Liu et al. [19]	Contact-prioritized planning of impact-resilient aerial robots with compliant arm	Compliant Shield Deformation	IMU + compliance deformation	~3.0	1.38
Liu et al. [20]	Dynamic modeling of impact-resilient MAVs under high-speed, large-angle collisions	Compliant Arm Deformation	IMU + compliance deformation	~3.5	1.38
Wang et al. [21]	Air-Bumper collision detection and reaction framework	Acceleration-based	IMU	~1.0	1.45
Wang et al. [22]	Fly–Crash–Recover: sensor-based reactive recovery	Accelerometer norm threshold	IMU	~0.5	0.08
Patnaik et al. [23]	Foldable compliant arm for passive impact absorption	Accelerometer-based	IMU	~2.5	1.11
De Petris et al. [31]	Attitude estimation for collision recovery	Sudden attitude-rate deviation	IMU	~1.7	0.50
Battison et al. [33]	Filter-based attitude estimation	Magnetometer-Based	IMU	~4.0 (mostly manual control)	n/a
This Work	Tactile feedback + tensegrity frame	Binary tactile contact	Tactile + IMU	3.7 (exp), 8.0 (sim)	0.321

## 7. Conclusions

This paper introduces a novel approach to high-speed collision recovery and path adaptation. By leveraging tactile feedback in the form of binary contact information, the method enhances state estimation after collisions, enabling more effective recovery and dynamic adjustments to pre-planned paths based on newly acquired obstacle data.

Unlike traditional obstacle avoidance techniques that rely on heavy and computationally intensive sensors, such as cameras and LiDAR, this method introduces lightweight, low-bandwidth, and low-latency touch sensors to increase the flight robustness in cluttered environments. Results demonstrate the full recovery and re-planning pipeline through Monte Carlo simulations and flight experiments on a real prototype. In simulation the approach achieves successful recovery from collisions at speeds up to 8 m/s. On the implemented physical prototype, with a notable difference due to hardware limitations, the drone achieves in-flight collision recovery at high velocities up to 3.7 m/s. Additionally, it showcases the approach to path adaptation that relies solely on touch sensors, offering computational efficiency. Future work will focus on increasing the prototype's payload capacity to equip all vertices of the vehicle with contact sensors, maximizing the potential for path recovery from all angles. Furthermore, it will extend this approach to dynamic obstacles by constructing a time-dependent vector field  $\mathbf{g}(\mathbf{x}, t)$ , allowing each obstacles' contribution to exponentially decay while being propagated along expected velocities. The authors also plan to integrate onboard state estimation, using a downward-facing flow sensor to enable operation in natural, outdoor environments. The ability to recover from collisions uses a paradigm in which contact is no longer avoided, but *actively exploited*.

**Author Contributions:** Conceptualization, A.B. and M.W.M.; methodology, A.B.; software, A.B.; validation, A.B., T.Y.; formal analysis, A.B.; investigation, A.B.; resources, A.B.; data curation, A.B.; writing—original draft preparation, A.B.; writing—review and editing, A.B., T.Y., S.H., and M.W.M.; visualization, A.B.; supervision, S.H. and M.W.M.; project administration, S.H. and M.W.M.; funding acquisition, S.H. and M.W.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Agriculture and Food Research Initiative (AFRI) Competitive Grant no. 2020-67021-32855/project accession no. 1024262 from the USDA National Institute of Food and Agriculture, and by project “*Aerial Robots in a Tangible World: Drones with the Sense of Touch Act upon Their Surroundings*” funded by the Dutch Research Council, grant number NWO-VENI-20308.

**Data Availability Statement:** Code and raw data is linked to on the github repository: <https://github.com/BioMorphic-Intelligence-Lab/colliding-drone> (accessed on 15 October 2025). The project page can be found at: [https://antbre.github.io/Projects/colliding\\_drone.html](https://antbre.github.io/Projects/colliding_drone.html) (accessed on 15 October 2025).

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
MAV	Micro Aerial Vehicle
GNSS	Global Navigation Satellite System
EE	End-Effector
DoF	Degree-of-Freedom
CoM	Center-of-Mass
AM	Aerial Manipulator

TN	Tactile Navigation
NDT	Non-Destructive Testing
RPM	Rotations Per Minute
KF	Kalman Filter
EKF	Extended Kalman Filter

## References

1. Ollero, A.; Tognon, M.; Suarez, A.; Lee, D.; Franchi, A. Past, Present, and Future of Aerial Robotic Manipulators. *IEEE Trans. Robot.* **2022**, *38*, 626–645. [\[CrossRef\]](#)
2. Meng, J.; Buzzatto, J.; Liu, Y.; Liarokapis, M. On Aerial Robots with Grasping and Perching Capabilities: A Comprehensive Review. *Front. Robot. AI* **2022**, *8*, 739173. [\[CrossRef\]](#)
3. Zheng, L.; Hamaza, S. ALBERO: Agile Landing on Branches for Environmental Robotics Operations. *IEEE Robot. Autom. Lett.* **2024**, *9*, 2845–2852. [\[CrossRef\]](#)
4. Bogue, R. The role of robots in environmental monitoring. *Ind. Robot. Int. J. Robot. Res. Appl.* **2023**, *50*, 369–375. [\[CrossRef\]](#)
5. Ubellacker, S.; Ray, A.; Bern, J.M.; Strader, J.; Carbone, L. High-speed aerial grasping using a soft drone with onboard perception. *npj Robot.* **2024**, *2*, 5. [\[CrossRef\]](#)
6. Zeng, J.; Gimenez, A.M.; Vinitsky, E.; Alonso-Mora, J.; Sun, S. Decentralized Aerial Manipulation of a Cable-Suspended Load Using Multi-Agent Reinforcement Learning. In Proceedings of the 9th Conference on Robot Learning, PMLR, Seoul, Republic of Korea, 27–30 September 2025; Volume 305, pp. 3850–3868.
7. Hanover, D.; Loquercio, A.; Bauersfeld, L.; Romero, A.; Penicka, R.; Song, Y.; Cioffi, G.; Kaufmann, E.; Scaramuzza, D. Autonomous Drone Racing: A Survey. *IEEE Trans. Robot.* **2024**, *40*, 3044–3067. [\[CrossRef\]](#)
8. Rashida Farsath, K.; Jitha, K.; Mohammed Marwan, V.; Muhammed Ali Jouhar, A.; Muhammed Farseen, K.; Musrifa, K. AI-Enhanced Unmanned Aerial Vehicles for Search and Rescue Operations. In Proceedings of the 2024 5th International Conference on Innovative Trends in Information Technology (ICITIIT), Kottayam, India, 15–16 March 2024; pp. 1–10. [\[CrossRef\]](#)
9. Bakirci, M. A Novel Swarm Unmanned Aerial Vehicle System: Incorporating Autonomous Flight, Real-Time Object Detection, and Coordinated Intelligence for Enhanced Performance. *Trait. Signal* **2023**, *40*, 2063–2078. [\[CrossRef\]](#)
10. Yasin, J.N.; Mahboob, H.; Haghbayan, M.H.; Yasin, M.M.; Plosila, J. Energy-Efficient Navigation of an Autonomous Swarm with Adaptive Consciousness. *Remote Sens.* **2021**, *13*, 1059. [\[CrossRef\]](#)
11. Shraim, H.; Awada, A.; Younes, R. A survey on quadrotors: Configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control. *IEEE Aerosp. Electron. Syst. Mag.* **2018**, *33*, 14–33. [\[CrossRef\]](#)
12. Yasin, J.; Mohamed, S.A.S.; Haghbayan, H.; Heikkonen, J.; Tenhunen, H.; Plosila, J. Unmanned Aerial Vehicles (UAVs): Collision Avoidance Systems and Approaches. *IEEE Access* **2020**, *8*, 105139–105155. [\[CrossRef\]](#)
13. Kulkarni, M.; Alexis, K. Reinforcement Learning for Collision-free Flight Exploiting Deep Collision Encoding. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 15781–15788. [\[CrossRef\]](#)
14. Kaufmann, E.; Bauersfeld, L.; Loquercio, A.; Müller, M.; Koltun, V.; Scaramuzza, D. Champion-level drone racing using deep reinforcement learning. *Nature* **2023**, *620*, 982–987. [\[CrossRef\]](#)
15. Ding, R.; Bai, S.; Dong, K.; Chirarattananon, P. Aerodynamic effect for collision-free reactive navigation of a small quadcopter. *npj Robot.* **2023**, *1*, 2. [\[CrossRef\]](#)
16. Briad, A.; Kornatowski, P.; Zufferey, J.C.; Floreano, D. A collision-resilient flying robot. *J. Field Robot.* **2014**, *31*, 496–509. [\[CrossRef\]](#)
17. Zha, J.; Wu, X.; Dimick, R.; Mueller, M.W. Design and Control of a Collision-Resilient Aerial Vehicle With an Icosahedron Tensegrity Structure. *IEEE/ASME Trans. Mechatronics* **2024**, *29*, 3449–3460. [\[CrossRef\]](#)
18. Liu, Z.; Karydis, K. Toward impact-resilient quadrotor design, collision characterization and recovery control to sustain flight after collisions. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 183–189. [\[CrossRef\]](#)
19. Liu, Z.; Lu, Z.; Agha-mohammadi, A.a.; Karydis, K. Contact-Prioritized Planning of Impact-Resilient Aerial Robots with an Integrated Compliant Arm. *IEEE/ASME Trans. Mechatronics* **2023**, *28*, 2064–2072. [\[CrossRef\]](#)
20. Liu, Z.; Karydis, K. Dynamic Modeling and Analysis of Impact-Resilient MAVs Undergoing High-Speed and Large-Angle Collisions with the Environment. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 4285–4292. [\[CrossRef\]](#)
21. Wang, R.; Guo, Z.; Chen, Y.; Wang, X.; Chen, B.M. Air Bumper: A Collision Detection and Reaction Framework for Autonomous MAV Navigation. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 15735–15741.

22. Wang, S.; Anselmo, N.; Garrett, M.; Remias, R.; Trivett, M.; Christoffersen, A.; Bezzo, N. Fly-Crash-Recover: A Sensor-based Reactive Framework for Online Collision Recovery of UAVs. In Proceedings of the 2020 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 24 April 2020; pp. 1–6. [CrossRef]

23. Patnaik, K.; Mishra, S.; Chase, Z.; Zhang, W. Collision Recovery Control of a Foldable Quadrotor. In Proceedings of the 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Delft, The Netherlands, 12–16 July 2021; pp. 418–423. [CrossRef]

24. Schuster, M.; Bredenbeck, A.; Beitelshmidt, M.; Hamaza, S. Tactile Odometry in Aerial Physical Interaction. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; pp. 8103–8110. [CrossRef]

25. Livox Technology Inc. Livox Mid-40 & Mid-100 Technical Data. 2025. Available online: <https://www.livoxtech.com/mid-40-and-mid-100/specs> (accessed on 15 October 2025).

26. Bakir, A.; Özbeş, D.; Abazari, A.; Özcan, O. SCoReR: Sensorized Collision Resilient Aerial Robot. In Proceedings of the 2023 IEEE International Conference on Soft Robotics (RoboSoft), Singapore, 3–7 April 2023; pp. 1–7. [CrossRef]

27. Mintchev, S.; de Rivaz, S.; Floreano, D. Insect-Inspired Mechanical Resilience for Multicopters. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1248–1255. [CrossRef]

28. Hamaza, S.; Georgilas, I.; Richardson, T. 2d contour following with an unmanned aerial manipulator: Towards tactile-based aerial navigation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3 November 2019; pp. 3664–3669.

29. Nguyen, P.H.; Patnaik, K.; Mishra, S.; Polygerinos, P.; Zhang, W. A Soft-Bodied Aerial Robot for Collision Resilience and Contact-Reactive Perching. *Soft Robot.* **2023**, *10*, 838–851. [CrossRef] [PubMed]

30. Zha, J.; Mueller, M.W. Exploiting collisions for sampling-based multicopter motion planning. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 7943–7949. [CrossRef]

31. De Petris, P.; Nguyen, H.; Kulkarni, M.; Mascalich, F.; Alexis, K. Resilient collision-tolerant navigation in confined environments. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 2286–2292.

32. De Petris, P.; Nissov, M.; Harms, M.; Alexis, K. Design, Modeling, Control and Experimental Evaluation of Flexy: A Novel Compliant Quadrotor. In Proceedings of the 2024 International Conference on Unmanned Aircraft Systems (ICUAS), Chania, Greece, 4–7 June 2024; pp. 31–38.

33. Battiston, A.; Sharf, I.; Nahon, M. Attitude estimation for collision recovery of a quadcopter unmanned aerial vehicle. *Int. J. Robot. Res.* **2019**, *38*, 1286–1306. [CrossRef]

34. Lew, T.; Emmei, T.; Fan, D.D.; Bartlett, T.; Santamaría-Navarro, A.; Thakker, R.; Agha-mohammadi, A. Contact Inertial Odometry: Collisions are your Friends. In *Robotics Research*; Asfour, T., Yoshida, E., Park, J., Christensen, H., Khatib, O., Eds.; Springer: Cham, Switzerland, 2022; pp. 938–958.

35. Bredenbeck, A.; Santina, C.D.; Hamaza, S. Embodying Compliant Touch on Drones for Aerial Tactile Navigation. *IEEE Robot. Autom. Lett.* **2025**, *10*, 1209–1216. [CrossRef]

36. Schmidl, H.; Milenkovic, V. A fast impulsive contact suite for rigid body simulation. *IEEE Trans. Vis. Comput. Graph.* **2004**, *10*, 189–197. [CrossRef]

37. Mueller, M.W.; Hehn, M.; D'Andrea, R. Covariance Correction Step for Kalman Filtering with an Attitude. *J. Guid. Control Dyn.* **2017**, *40*, 2301–2306. [CrossRef]

38. Chen, Z.; Biggie, H.; Ahmed, N.; Julier, S.; Heckman, C. Kalman Filter Auto-Tuning With Consistent and Robust Bayesian Optimization. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 2236–2250. [CrossRef]

39. Wilhelm, J.P.; Clem, G. Vector Field UAV Guidance for Path Following and Obstacle Avoidance with Minimal Deviation. *J. Guid. Control Dyn.* **2019**, *42*, 1848–1856. [CrossRef]

40. Zha, J.; Wu, X.; Kroeger, J.; Perez, N.; Mueller, M.W. A collision-resilient aerial vehicle with icosahedron tensegrity structure. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 1407–1412. [CrossRef]

41. Elandt, R.; Drumwright, E.; Sherman, M.; Ruina, A. A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3 November 2019; pp. 8238–8245. [CrossRef]

42. Masterjohn, J.; Guoy, D.; Shepherd, J.; Castro, A. Velocity Level Approximation of Pressure Field Contact Patches. *IEEE Robot. Autom. Lett.* **2022**, *7*, 11593–11600. [CrossRef]