

# Contact-Prioritized Planning of Impact-Resilient Aerial Robots With an Integrated Compliant Arm

Zhichao Liu , Zhouyu Lu , Ali-akbar Agha-mohammadi, and Konstantinos Karydis , *Member, IEEE*

**Abstract**—This article develops an impact-resilient aerial robot (s-ARQ) equipped with a compliant arm to sense contacts and reduce collision impact and featuring a real-time contact force estimator and a nonlinear motion controller to handle collisions while performing aggressive maneuvers and stabilize from high-speed wall collisions. Further, a new collision-inclusive planning method that aims to prioritize contacts to facilitate aerial robot navigation in cluttered environments is proposed. A range of simulated and physical experiments demonstrate key benefits of the robot and the contact-prioritized (CP) planner. Experimental results show that the compliant robot has only a 4% weight increase but around 40% impact reduction in drop tests and wall collision tests. s-ARQ can handle collisions while performing aggressive maneuvers and stabilize from high-speed wall collisions at 3.0 m/s with a success rate of 100%. Our proposed compliant robot and CP planning method can accelerate computation time while having shorter trajectory time and larger clearances compared to A\* and RRT\* planners with velocity constraints. Online planning tests in partially known environments further demonstrate the preliminary feasibility of our method to apply in practical use cases.

**Index Terms**—Autonomous aerial vehicles, collision mitigation, motion planning, nonlinear control systems, robot motion.

## I. INTRODUCTION

MICRO aerial vehicles (MAVs) can support sensor-based exploration and navigation, and en route to robust autonomous navigation, aerial autonomy with interactive behavior has been studied [1], [2], [3]. There has been a growing interest in deploying MAVs in challenging environments, including but

not limited to confined [4], [5] and cluttered [6] ones. Collision risks get significantly higher for autonomous missions in these complex environments. Compliant resilient robots attract growing attention due to the merits of reducing impact and protecting onboard sensors [7]. Taking advantage of impact resilience, research efforts on collision-inclusive motion planning have started to be proposed [8], [9], [10].

In this work, we introduce a lightweight compliant arm to sense contacts and reduce high-speed collision impact. Equipped with the integrated compliant arm, we develop an impact-resilient aerial robot (named s-ARQ). The compliant robot has only a 4% weight increase compared to its rigid counterpart, however, experimental results show that the compliant arm can reduce around 40% impact. The compliant arm incorporates a passive spring and a laser ranging sensor to enable contact force estimation. Employing a force estimator and nonlinear motion controller, s-ARQ can stabilize from high-speed wall collisions at 3.0 m/s with a success rate of 100%. We consolidate the impact resilience by including pole obstacle collisions, as well as different yaw angles. Further, we harness s-ARQ's strong collision resilience capability to propose a novel planning method that prioritizes contacts. Physical tests and extended simulations demonstrate that our proposed compliant robot and contact-prioritized (CP) planner can accelerate computation while achieving shorter trajectory time and larger clearances compared to collision-avoidance methods with velocity constraints. Online planning tests in partially known environments were studied to support application toward practical use cases. Simulated results further validate the efficiency of the proposed CP planner.

## II. RELATED WORKS

### A. Impact-Resilient Compliant MAVs

Several compliant aerial robots have been developed over the years. Bartelds et al. [1] developed an aerial robot with a compliant contact arm. Examples of works integrating compliant protective structure onto the robot include origami-inspired mechanisms [11], [12] and an icosahedron tensegrity structure [13]. De Petris et al. [4] studied impact-resilient MAVs with external compliant flaps. Compliance has also been included into the robot chassis to reduce impact. Chen et al. [14] developed a collision-resilient insect-scale compliant flapping-wing robot, while Patnaik et al. [7] introduced a collision-resilient MAV with foldable arms. Soft aerial robots were developed for physical interactions [15], [16]. Further, insect-inspired multicopters

Manuscript received 4 January 2023; revised 29 March 2023; accepted 15 May 2023. Date of publication 19 June 2023; date of current version 16 August 2023. Recommended by Technical Editor D. Naso and Senior Editor Q. Zou. This work was supported in part by NSF under Grant #IIS-1724341, Grant #IIS-1901379, and Grant #IIS-1910087; in part by ARL under Grant #W911NF-18-1-0266; and in part by ONR under Grant #N00014-19-1-2264. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. (Corresponding author: Konstantinos Karydis.)

Zhichao Liu, Zhouyu Lu, and Konstantinos Karydis are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521 USA (e-mail: zliu157@ucr.edu; zlu044@ucr.edu; kkarydis@ece.ucr.edu).

Ali-akbar Agha-mohammadi is with Field AI, Mission Viejo, CA 92691 USA (e-mail: ali@fieldai.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TMECH.2023.3277811>.

Digital Object Identifier 10.1109/TMECH.2023.3277811

are presented with compliant frames to handle collisions [17], whereas compliant frames based on tensegrity [18] are proved helpful to reduce impact. Our prior work developed a collision-resilient MAV with compliant arms [19]. Compliant MAVs can reduce the effect of impact and help survive collisions; however, these robots cannot estimate contact force and handle impacts accordingly.

### B. Contact Force Estimation

This work focuses on estimating contact force (excluding external torque). One way is to map robot control inputs to external contact forces offline [20]. However, this applies only when the robot is in contact. In a different approach, momentum-based external wrench estimators with second-order estimation dynamics [5], [21] and a Lyapunov-based nonlinear external wrench observer including also inertia shaping [22] have been proposed. Unscented Kalman filters are also utilized for estimation [23]. A nonlinear disturbance observer has been proposed to estimate contact force [24]. Recently, the feasibility of using cameras to estimate contact force was shown [25]. Yet, external force estimation in presence of compliant frames is an open task for aerial robots.

### C. Physical Interaction

Aerial robots are equipped with end-effectors to physically interact with environments. Bartelds et al. [1] studied a compliant manipulator for impact reduction. Flying robots with end-effectors are utilized to apply a force to vertical walls [26], [27], inspection [28], and sensor placement [25]. Alejandro et al. [29] developed a compliant arm to monitor and control interaction wrench. However, these projects focus on low-speed interactions, while high-speed collisions involve large impact forces and attitude changes. The interactions fail to assist with motion tasks, such as planning and exploration.

### D. Contact-Inclusive Planning

Contrary to collision avoidance, impact-resilient robots can embrace contacts to improve overall safety and navigation task effectiveness. Mote et al. [30] incorporated the collision model into mixed integer programming for trajectory optimization. Contacts can be also used to improve velocity estimation [5] and mapping [31]. Risk reward tradeoffs have been studied for collision-inclusive trajectories [32], [33]. Local replanners with setpoint adjustment post collision can be adopted to improve global planners like A\* [9], [10] and sampling methods [8]. However, these methods directly extend global planners and can bound to their constraints.

## III. DEVELOPMENT AND KEY FEATURES OF S-ARQ

### A. Design

Inspired by prior works on compliant end-effectors [1], we introduce a lightweight compliant arm design and embed it onto a custom-made quadrotor to enable the latter to both estimate contact forces and stabilize from high-speed collisions.

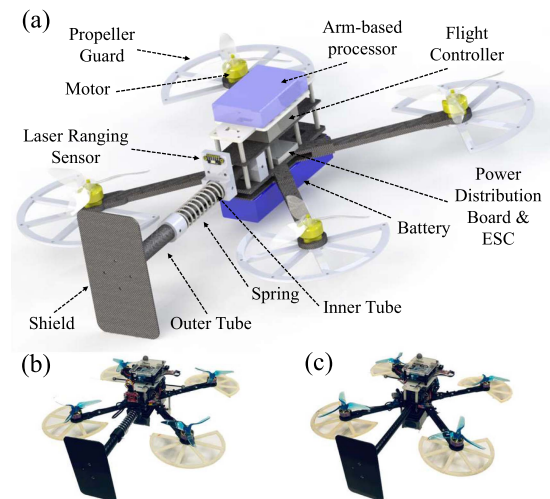


Fig. 1. Impact-resilient aerial robot introduced in this work. (a) CAD rendering of the robot. Physical prototypes of the (b) compliant robot and (c) its rigid counterpart.

The robot introduced in this work is named single-arm active resilient quadrotor (s-ARQ). When moving forward in static environments (the robot's front faces the moving direction); robots mostly have contacts only in the front direction. We revise our earlier compliant aerial robot design [19] to attach one compliant arm onto the (rigid otherwise) chassis (Fig. 1).

The new arm design consists of two carbon fiber tubes, a compression steel spring, a laser range sensor, and a carbon fiber shield. Carbon fiber tubes (tensile strength 125 000–175 000 psi) are assembled in a concentric manner. The outer tube has an outer diameter (OD) of 14.5 mm while the inner tube has an OD of 17.5 mm. The inner tube includes a linear slot to limit rotational motion with negligible friction, thus both tubes comprise a prismatic joint. A steel compression spring (OD 21.5 mm, free length 76 mm) connects both tubes. A lightweight fiber sheet (75 × 140 mm) is fabricated with a Stepcraft D.600 CNC router with enclosure and milling bath, to work as the end-effector (shield) to contact with obstacles. Multiple custom adapters are 3-D-printed with a Markforged Mark 2 printer. A laser range sensor (VL53L1X) is attached to the inner tube to measure the length of the compliant arm. Our design differs from works [1], [25] in its capacity to enable stabilization from high-speed collisions with large impact, and estimate contact forces in presence of frame compliance.

The robot chassis is shared with our prior work [34] and consists of custom carbon fiber frames, a flight controller (Pixhawk), and an ARM-based multicore processor (Odroid). The four frame arms measure 0.19 m, and the contact arm measures 0.28 m in free flight. The compliant robot weighs 890 g without batteries. For comparison purposes, we also build a rigid version (Quad), which shares the same quadrotor platform but a rigid contact arm of the same length [Fig. 1(c)]. S-ARQ is 50 g heavier (4% of the total weight with batteries) compared to its rigid counterpart. We use a 5200-mAh Lipo battery yielding a flight time of approximately 610 s.

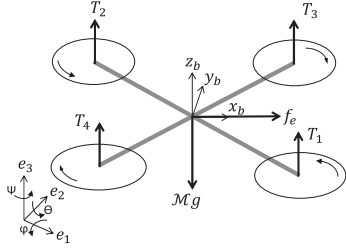


Fig. 2. Dynamic model of the system.

## B. Modeling

With reference to Fig. 2, the equations of motion for our robot following notation [21] are

$$\begin{aligned} \mathcal{M}\ddot{\mathbf{r}} &= -\mathcal{M}g\mathbf{e}_3 + \mathbf{R}\mathbf{f}_T + \mathbf{R}\mathbf{f}_e \\ \mathcal{I}\dot{\boldsymbol{\omega}} &= \mathcal{S}(\mathcal{I}\boldsymbol{\omega})\boldsymbol{\omega} + \mathbf{m}_T + \mathbf{m}_e \\ \dot{\mathbf{R}} &= \mathbf{R}\mathcal{S}(\boldsymbol{\omega}) \end{aligned} \quad (1)$$

where  $\mathbf{r} = [x \ y \ z]^T$  is the position in the inertial frame (East–North–Up),  $\mathcal{M}$  is the mass, and  $\mathbf{R} \in SO(3)$  denotes the rotation matrix from body (Forward–Left–Up) to inertial frame.  $\mathcal{S}(\cdot)$  is the skew-symmetric operator,  $g = 9.81 \text{ m/s}^2$  is the gravity constant,  $\mathbf{e}_3 = [0 \ 0 \ 1]^T$ ,  $\mathbf{f}_T$  and  $\mathbf{f}_e = [f_e \ 0 \ 0]^T$  are the thrust and external force vectors in body frame, respectively, and  $\mathbf{m}_T$  and  $\mathbf{m}_e$  are the moments generated by the thrust and external force vectors, respectively. Note that, as in [35], the external moment ( $\mathbf{m}_e$ ) is not considered herein.

## C. Contact Force Estimation

The contact force along the compliant arm can be measured utilizing the Hooke's law as

$$\hat{f}_e = -k_l(\delta l + l_0)$$

where  $\delta l = l_{\max} - \hat{l}$  is the arm length difference ( $l_{\max} = 0.28 \text{ m}$ ) and  $\hat{l}$  is the estimated arm length measured by the onboard distance sensor. To prevent oscillations, we preload the compliant arm ( $l_0 = 2 \text{ mm}$ ). The selected spring constant is  $k_l = 3.80 \text{ N/mm}$ . The distance sensor has precision of 1 mm with accuracy of  $\pm 5 \text{ mm}$ . To mitigate sensor noise we apply a recursive filter ( $w = 0.6$ ) to sensor readings ( $h_i$ ) as

$$\hat{l}_i = wh_i + (1-w)\hat{l}_{i-1} . \quad (2)$$

The distance sensor has a frequency of around 25 Hz. Admittedly, this design can only estimate forces along the contact arm ( $x_b$ -axis) in body frame. When flying toward known obstacles, the robot can face obstacles along its  $x_b$ -axis utilizing yaw control. Although the compliant arm is of no help to estimate external torques, the prismatic joint does not affect estimating methods, such as [21] and [22].

## IV. MOTION CONTROL AND COLLISION HANDLING

To stabilize after high-speed collision, the tracking controller of the robot must be able to follow aggressive trajectories with

large attitude angles. In this work, we adopt the cascaded tracking control method, as in our prior work [19]. Note that the attitude is described as roll ( $\phi$ ), pitch ( $\theta$ ), and yaw ( $\psi$ ) angles, such that  $\mathbf{R} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$ , where  $\mathbf{R}_x$ ,  $\mathbf{R}_y$ ,  $\mathbf{R}_z$  are elementary rotation matrices about the coordinate axis (see Fig. 2). The planner generates desired states (position  $\mathbf{r}_{\text{des}}$ , velocity  $\dot{\mathbf{r}}_{\text{des}}$ , acceleration  $\ddot{\mathbf{r}}_{\text{des}}$ , and yaw angle  $\psi_{\text{des}}$ ). The tracking controller comprises high-level position control, mid-level attitude and bodyrate control, and a low-level mixer to output PWM signals to actuators.

The position controller harnesses geometric constraints for nonlinear tracking [36], [37]. First we find the desired thrust force vector  $\mathbf{F}_{\text{des}} \in \mathbb{R}^3$  in the inertial frame

$$\mathbf{F}_{\text{des}} = -\mathbf{K}_d(\dot{\mathbf{r}} - \dot{\mathbf{r}}_{\text{des}}) - \mathbf{K}_p(\mathbf{r} - \mathbf{r}_{\text{des}}) + \mathcal{M}\ddot{\mathbf{r}}_{\text{des}} + \mathcal{M}g\mathbf{e}_3$$

where  $\mathbf{K}_d, \mathbf{K}_p \in \mathbb{R}^{3 \times 3}$  are diagonal, positive definite tuning matrices. Then, we can calculate the desired total thrust  $f_{T, \text{des}}$  in body frame by  $f_{T, \text{des}} = \mathbf{F}_{\text{des}}^T \cdot \mathbf{e}_3$ . Given that the robot can only produce thrust along the  $z_b$ -axis, we align  $z_{b, \text{des}}$  with  $\mathbf{F}_{\text{des}}$ , and align  $\mathbf{y}_{b, \text{des}}$  to match the desired yaw  $\psi_{\text{des}}$ . Therefore, we can calculate the desired attitude  $\mathbf{R}_{\text{des}}$  as

$$\begin{aligned} z_{b, \text{des}} &= \frac{\mathbf{F}_{\text{des}}}{\|\mathbf{F}_{\text{des}}\|} \\ \ddot{\mathbf{r}}_{\psi} &= [\cos \psi_{\text{des}}, \sin \psi_{\text{des}}, 0]^T \\ \mathbf{y}_{b, \text{des}} &= \frac{z_{b, \text{des}} \times \ddot{\mathbf{r}}_{\psi}}{\|z_{b, \text{des}} \times \ddot{\mathbf{r}}_{\psi}\|} \\ \mathbf{R}_{\text{des}} &= [\mathbf{y}_{b, \text{des}} \times z_{b, \text{des}}, \mathbf{y}_{b, \text{des}}, z_{b, \text{des}}] \end{aligned}$$

where the operator  $\times$  denotes the cross product. After calculating the desired attitude, we input it into a nonlinear attitude tracking controller to regulate the orientation of the robot. A quaternion-based controller [38] is adopted in this work, but other attitude tracking methods can achieve similar performance. See PX4 firmware [39] for details about the PID bodyrate control and mixer.

When collisions occur, the compliant arm compresses with increased estimated contact force ( $\hat{f}_e$ ). Collisions are detected when  $\hat{f}_e$  reaches a threshold ( $\hat{f}_e^* = 25 \text{ N}$ ). The maximum estimated contact force is measured ( $\hat{f}_{e, \max}$ ), and the collision handling is started when the estimated contact force falls below  $\hat{f}_e^*$  following the detection. We revise the collision handling method in our prior work [19] to generate trajectories to reach a setpoint at a distance proportional to  $\hat{f}_{e, \max}$ . We use  $\mathbf{r}_c, \dot{\mathbf{r}}_c$  to denote the position and velocity of the robot in the inertial frame when the collision handling is started, as well as the rotation matrix ( $\mathbf{R}_c$ ). The new setpoint ( $\mathbf{r}_n$ ) in the inertial frame can be written as

$$\mathbf{r}_n = \mathbf{r}_c - (\eta\hat{f}_{e, \max} + d_0)\mathbf{R}_c\mathbf{x}_b \quad (3)$$

where  $\eta$  and  $d_0$  are constants ( $\eta = 0.01 \text{ m/N}$ ,  $d_0 = 0.2 \text{ m}$ ). During collision handling, the robot tracks a smooth (polynomial) trajectory so that for  $t \in [t_0, t_T]$ ,  $\mathbf{r}(t_0) = \mathbf{r}_c$ ,  $\dot{\mathbf{r}}(t_0) = \dot{\mathbf{r}}_c$  and it stops at  $\mathbf{r}(t_T) = \mathbf{r}_n$ . The time interval is computed based on maximum accelerations and velocities [40].

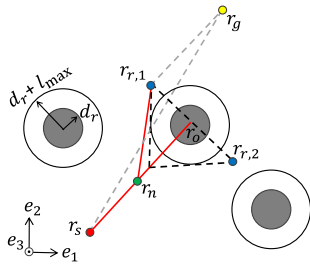


Fig. 3. Novel planning method (CP) to prioritize contact to facilitate navigation in cluttered environments.

## V. CONTACT-PRIORITIZED PLANNING

Contrary to collision-inclusive local replanners [8], [10], this work proposes an intuitive global planner to exploit impact resilience. We draw motivation from the use case of aerial robots rapidly traversing forest-like environments, whereby maps contain isolated cylindrical obstacles with constant radius ( $d_r$ ). With an intention to utilize the impact resilience, our proposed CP planner prioritizes collisions to facilitate navigation.

As shown in Fig. 3, the robot starts at  $r_s$  while the goal is at  $r_g$ , which share the same altitude ( $e_3$ ), thus the navigation is simplified as 2-D with constant  $e_3$ .<sup>1</sup> The robot is simplified as a ball with a radius of  $l_{\max}$ , thus augmented obstacles have a radius of  $d_r + l_{\max}$ . The planner starts with drawing a line  $\overrightarrow{r_s r_g}$  (gray dashed), and checks if the line intersects with any augmented obstacles. If intersections occur, the robot moves toward the center of the first obstacle  $r_o$  while controlling yaw to face  $r_o$  along  $x_b$ -axis, collides and stabilizes at  $r_n$ , as described in (3). After recovery, two added waypoints  $r_{r,\{1,2\}}$  are found, which lie on the line perpendicular to  $\overrightarrow{r_s r_o}$  (black dashed) with a distance of  $\sqrt{2}(d_r + l_{\max})$ . The robot moves to the added waypoint closer to the goal ( $r_{r,1}$  in this case), and repeats exploration with a new starting point ( $r_s = r_{r,1}$ ) until no obstacles are found along the line  $\overrightarrow{r_s r_g}$ . The robot follows minimum-snap polynomial trajectories with the desired colliding velocities at the center of the in-contact obstacles.

## VI. RESULTS

We present results from four experimental tests, namely, force estimation, impact reduction, collision resilience, and planning. A 12-camera VICON motion capture system over WiFi was used for odometry feedback at a rate of 100 Hz. The feedback is only used to estimate the state of the robot, which can also be achieved by cameras or laser sensors in outdoor environments. Note that we use accelerometer data  $\hat{a}_b$  of the rigid robot for collision detection. Hence, collisions are detected whenever  $\|\mathbf{R}\hat{a}_b + g\mathbf{e}_3\| \geq 2g$ . The rigid robot employs the same collision handling method with a constant maximum force ( $f_{e,\max} = 80$  N) as it cannot directly estimate contact forces.

<sup>1</sup> Note that the collision recovery has varying altitude but the setpoint  $r_n$  shares the same  $e_3$  value.

TABLE I  
CONTACT FORCE ESTIMATION STATISTICS (10 TRIALS)

Case \ Impact	30 N	40 N	50 N
Static (N)	$30.18 \pm 1.08$	$39.76 \pm 1.62$	$50.61 \pm 1.38$
Dynamic (N)	$30.78 \pm 3.16$	$40.66 \pm 4.06$	$52.11 \pm 4.68$

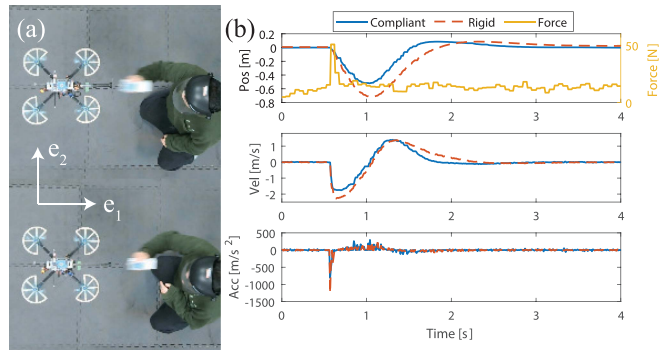


Fig. 4. (a) Snapshots and (b) states tracking of a sample test to evaluate contact force estimation.

### A. Force Estimation

First, we study the force estimation of the compliant arm. Two cases are considered: 1) when s-ARQ is placed on the ground (static) and 2) while hovering (dynamic). We use a digital force gauge for ground truth. We apply constant forces to s-ARQ at 30, 40, 50 N in the static case; in the dynamic case we apply impact forces (hit and release) of the same values. Table I shows the mean and standard deviation values of ten consecutive trials for each case. Despite sensor noise, results show relatively accurate contact force estimation when the robot is static. The estimation accuracy deteriorates in the dynamic case. This can be associated with vibrations in flight; yet, impact forces also contain larger errors when the robot is flying. Nevertheless, experimental results validate the feasibility of estimating contact forces using the developed compliant arm while in flight.

Further, we study the effect of the embedded compliance on the response to external impacts. We apply an impact force of 50 N along the  $e_1$ -axis to both s-ARQ and its rigid counterpart, Quad [Fig. 4(a) top and bottom panels, respectively]. Note that both robots have the same weight in the current and all following tests. Fig. 4(b) depicts the position, velocity, and acceleration along the  $e_1$ -axis of the compliant (blue solid) and rigid (red dashed) robots while the yellow-colored curve shown on the top panel denotes the estimated contact force on the s-ARQ robot. Note that accelerations are computed based on velocity data from motion capture in the current and all following tests.

Results show that s-ARQ can detect a contact force of about 50 N, as desired. In addition, the compliant robot has fewer changes in all states under impact forcing owing to the embedded compliance. This comparison indicates that existing methods that rely on robot states alone may underestimate impact contact forces when there is embedded compliance.

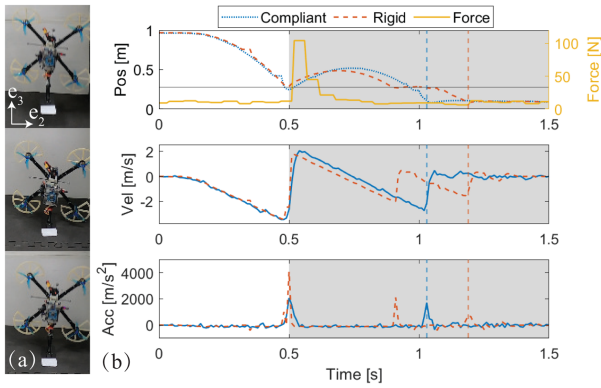


Fig. 5. (a) Snapshots and (b) states tracking of a 0.7-m drop test to study impact reduction. Grey-shaded area denotes states post impact.

TABLE II  
RECORDED MAXIMUM ACCELERATION STATISTICS (10 TRIALS)

Robot \ Drop Height	0.3 m	0.5 m	0.7 m
Compliant ( $\text{m/s}^2$ )	$1,017 \pm 103$	$1,545 \pm 129$	$2,177 \pm 140$
Rigid ( $\text{m/s}^2$ )	$1,809 \pm 186$	$2,774 \pm 169$	$3,649 \pm 175$

### B. Impact Reduction

In the second set of tests we seek to study the impact reduction afforded by the embedded compliance. To this end, we employ drop tests. Both robots are fixed vertically ( $x_b$  facing  $-e_3$  and  $z_b$  facing  $e_1$ ) before falling to the ground (hard floor mat) from 0.3, 0.5, and 0.7 m along  $e_3$  axis.

Fig. 5(a) shows snapshots from one of the 0.7-m drop tests for s-ARQ. The compliant arm touches the ground (top), compresses to the minimum length (middle), and then bounces back (bottom). Fig. 5(b) depicts position, velocity, and acceleration tracking along the  $e_3$ -axis of s-ARQ (blue solid) and Quad (red dashed) for a sample 0.7-m drop test, as well as the estimated contact force (curve in yellow at the top panel). A horizontal black line denotes the  $e_3$  value (0.28 m) when the robot is placed vertically on the ground.

Results show that both robots have identical position and velocity profiles before touching the ground with a velocity of  $-3.46$  m/s. The compliant arm length reduces to its minimum, followed by a saturated force estimation of 104 N. During the impact, s-ARQ has a maximum acceleration of  $2069$   $\text{m/s}^2$  while Quad reaches  $4063$   $\text{m/s}^2$ . Blue and red vertical dashed lines denote that the robot flips to a horizontal state ( $z_b$  facing  $e_3$ ), and therefore lower  $e_3$  values are observed.

Further, we repeat ten drop tests at three different  $e_3$  values for both robots and record the mean and standard deviation of maximum accelerations attained (Table II). Results show that the compliant arm design can help reduce impact by 43.8%, 44.3%, and 40.3% in the drop tests at 0.3, 0.5, and 0.7 m, respectively. These findings demonstrate that our compliant aerial robot design can reduce impact by around 40% with only 4% weight increase.

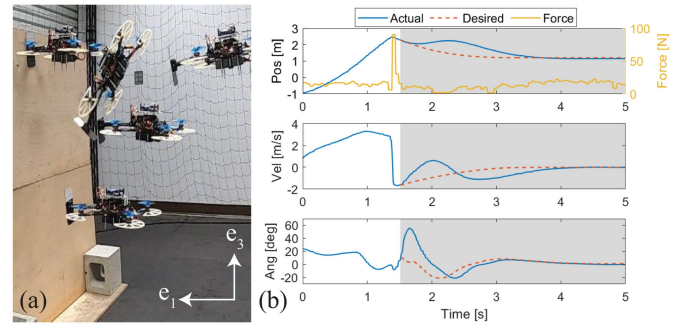


Fig. 6. (a) Snapshots and (b) states tracking of a sample wall collision test at a speed of 3.0 m/s for s-ARQ.

### C. Collision Resilience

We also study the s-ARQ robot's collision resilience using extensive physical collision tests against vertical walls and poles at different velocities and yaw and pitch angles, and compare against the rigid robot, Quad. In wall tests, we place a vertical wall at the  $e_1$  position of 2.45 m and perpendicular to  $e_1$ -axis. Both robots take off at the  $e_1$  position of  $-1$  m and fly along the  $e_1$ -axis before hitting the wall with zero Euler angles (identity rotation matrix). Owing to the embedded compliance's utility to reduce impact, s-ARQ can sustain wall collisions at a speed of 3.0 m/s with a 100% success rate for ten consecutive trials (see supplemental video). In contrast, the rigid Quad robot can fail at the highest speed collision of 3.0 m/s because of IMU malfunctions caused by the impact.

With reference to Fig. 6(a), s-ARQ has the compliant arm compressed when colliding with the wall, followed by recovering with large attitude angles and stabilizing at a safe position. Fig. 6(b) depicts the position and velocity of s-ARQ along the  $e_1$ -axis, as well as the pitch angle  $\theta$ . Blue solid curves denote the actual states while the red dashed ones represent the desired states from the planner. A yellow curve denotes the estimated contact force (top panel), and a grey-shaded area means the recovery control is enabled.

Results show that s-ARQ touches the wall at a speed of 3.0 m/s at time  $t = 1.35$  s. The robot stops and bounces back at a speed of  $-1.60$  m/s, during which a maximum contact force of 90 N is recorded. Recovery control is enabled (desired states) when the estimated force drops below the threshold ( $\hat{f}_e^* = 25$  N) at  $t = 1.51$  s. The recovery trajectory begins at current position, velocity and attitude states, and stabilizes rapidly at a distance proportional to  $\hat{f}_{e,\max}$ . The nonlinear controller tracks the aggressive recovery trajectory with large pitch ( $\theta_{\max} = 55^\circ$ ) for prompt post-collision stabilization.

In addition, we conduct wall collision tests at velocities of 1.5, 2.0, and 2.5 m/s along  $e_1$ -axis for both s-ARQ and Quad. Fig. 7(a) shows a top view of s-ARQ robot wall collision test, where the compliant arm compresses to reduce impact and protect onboard sensors. Fig. 7(b) visualizes sample trials of wall collision tests at a velocity of 2.5 m/s for both robots, where solid curves denote measured states while dashed ones represent desired states from the planner. Similarly, blue and

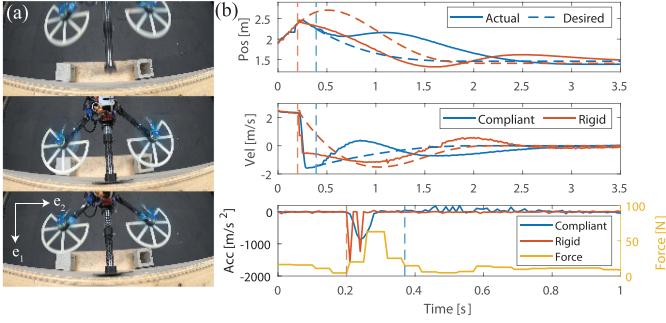


Fig. 7. (a) Compliant arm compressing during a collision. (b) State tracking of both s-ARQ and Quad for 2.5-m/s wall collisions.

red curves visualize position, velocity, and acceleration states of s-ARQ and its rigid counterpart, respectively. Blue and red vertical dashed lines denote the time when recovery controls are enabled for s-ARQ and Quad, and the yellow curve shows the estimated force on s-ARQ.

Results show that both robots have identical states before the contact at time  $t = 0.2$  s, when a collision is detected for the rigid robot. The velocity of Quad changes sharply from 2.5 to  $-0.5$  m/s in 0.05 s, resulting in a maximum acceleration of  $-1570$  m/s<sup>2</sup>. Due to the short contact time, the 100-Hz motion capture feedback causes discontinuity in the velocity tracking. On the contrary, the compliant arm elongates the contact time to 0.1 s, and reduces the impact to  $-876$  m/s<sup>2</sup>. A maximum contact force of 63 N is estimated; however, there is a slight delay in the force estimation due to the filter (2). The recovery control of s-ARQ is enabled at  $t = 0.37$  s, when the estimated contact force falls below the threshold. s-ARQ collision handling is started 0.17 s later than Quad's. Still, s-ARQ stabilizes at  $t = 3.0$  s with a settling time of 2.5 s, compared to 3.3 s of Quad. In sum, these findings demonstrate that s-ARQ can stabilize from collisions faster while also mitigating impacts, as compared to the rigid robot. In an effort to demonstrate the preliminary feasibility of our method to apply in practical use cases where high-accuracy localization feedback from motion capture is not available, we experimentally determined that the robot has same success rates when motion capture position feedback was processed (degraded) prior to be sent to the robot in a way that emulates key differences with visual inertial odometry feedback (namely, lower accuracy and larger delay).

We further study collisions against pole obstacles, as well as different yaw and pitch collision angles for s-ARQ. The pole obstacle has a radius of 0.15 m. We drive the s-ARQ robot to have yaw angles  $10^\circ$  (left),  $0^\circ$  (middle), and  $-10^\circ$  (right) collisions against wall and pole obstacles (see supplemental video). Note that the robot has a collision velocity of 2.0 m/s in all tests. Results show that the robot can stabilize from collisions against walls and poles with different yaw angles. Larger angle changes occur during wall collisions in nonzero yaw angles due to the flat geometry of the shield. The collision handling records the current yaw angle at the time of triggering and sustains the angle for stabilization. On the contrary, large angle changes are observed with zero colliding yaw angle in pole collisions, since

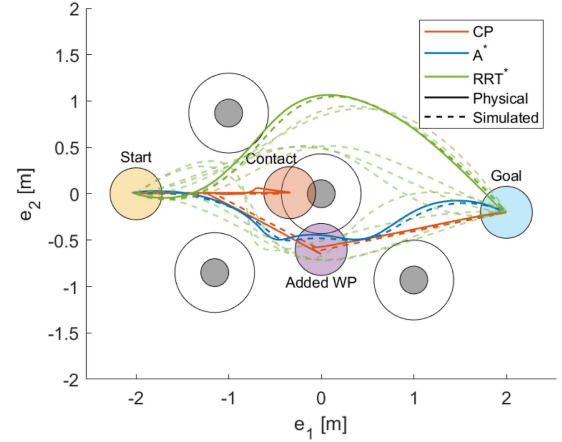


Fig. 8. Physical and simulated trajectories for different planners.

the robot is not ideally pointing to the geometric center. In addition, s-ARQ can survive wall collisions with yaw angles up to  $30^\circ$  (see supplemental video). However, direct contacts between obstacles and propellers occur at larger yaw angles, which pose danger to the robot. Further, the robot was experimentally found able to stabilize from large-pitch collisions of  $\pm 30^\circ$  as well (see supplemental video).

#### D. Contact-Prioritized Planning

In the final set of tests, we study the proposed CP planning method, and compare against A\* [41] and RRT\* [42] in both simulated and physical experiments. The latter help validate the proposed method in practice, whereas simulations help better understand the behavior of our CP planning algorithm in terms of its scalability in increasingly cluttered maps, all in relation to standard-of-practice planning algorithm baselines.

1) *Offline Planning*: The experimental map has a size of  $4 \times 3$  m. Four uniform pole obstacles with a radius of 0.15 m are located, as shown in Fig. 8. The robot starts at  $[-2, 0]$  and the goal is at  $[-2, -0.2]$ . We discretize the map with a resolution of 0.1 m, and run the three algorithms on a Windows machine with an Intel Xeon Processor (3.50 GHz). Four metrics are used to evaluate different methods: planning time, trajectory time, path length, and minimum distance to obstacles (clearances). The planning time records the time to find paths in milliseconds, excluding the time for trajectory generation. The trajectory time records the time for robots to reach the goal following the trajectories in seconds. Path lengths denote the Euclidean distance among waypoints (WP). Lastly, we record the clearances between trajectories and augmented obstacles to evaluate the safety against potential collisions, as in [43]. Note that in-contact obstacles are excluded in the CP planner due to the strong resilience to controlled collisions.

Fig. 8 depicts physical (solid) and simulated (dashed) trajectories of the three planning methods. Note that RRT\* results are stochastic with all simulated trials are visualized, however, and only one sample trial is included in the physical testing. Note that polynomials are generated based on waypoints from

**TABLE III**  
COMPARISON METRICS FOR THE PLANNING METHODS

Metrics \ Method	CP	A*	RRT*
Plan. Time (ms)	2.0	48.9	28.3 ± 0.9
Simul. Traj. Time (s)	9.4	10.0	11.3 ± 1.4
Phys. Traj. Time (s)	9.3	10.1	12.3
Path Len. (m)	5.5	4.2	4.4 ± 0.2
Simul. Cl. (m)	0.09	0.05	0.01 ± 0.03
Phys. Cl. (m)	0.10	0.01	-0.02

A\* and RRT\* planners at a maximum velocity of 1.5 m/s, while the CP method collides with obstacles at 2.5 m/s. The robot recovers at position  $r_n = [-1.25, 0]$  and moves to added WP  $r_r = [0, -0.61]$  before reaching the goal.

Comparison metrics of all methods are shown in Table III. It can be observed that simulated results generally match with the physical ones. Results show that our proposed method requires only around 4% and 7% planning time compared to A\* and RRT\* planners. In addition, results show that CP leads to the lowest trajectory time. Despite having a larger path length due to the collision recoveries, the CP planner has almost double obstacle clearances, indicating the enhanced safety of the trajectories. This is in fact a benefit of our controlled collision-inclusive planning: by selecting where to collide (safely), the risk for future (unsafe) collisions (as measured by clearances to other obstacles) can be reduced.

We extend planning tests to simulated cluttered maps. The maps have a size of  $20 \times 20$  m with 30 pole obstacles (see supplemental video). The obstacles have a uniform radius of 0.3 m, and are randomly distributed with a clearance of 2.5 m from center to center. The start is at  $[-8, -8]$  while the goal is at  $[8, 8]$ . Ten trials are run for each map and planning method with a discretization resolution of 0.5 m. Note that  $|v_{\max}|$  of simulated trajectories by A\* and RRT\* planners were capped at 1.5 m/s, and a collision speed of 3.0 m/s was used for trajectories of the CP planner as our prior work indicated higher velocities with collision resilience [10].

Simulated results are listed in Fig. 9 where different planners are evaluated in four metrics, as mentioned above. Note that two cases of the CP planner are studied in the trajectory time comparison. The compliant robot (CP C.) has a maximum velocity of 3.0 m/s and recovery time 2.5 s, while the rigid robot (CP R.) uses the velocity 2.5 m/s and time 3.3 s, as we measured in the collision tests. Results show that the CP planner cost around 30% planning time compared to other methods. In the meantime, the results show that the compliant aerial robot with the CP planner saves about 36% and 45% trajectory time compared to A\* and RRT\* planners, respectively. The compliant robot saves about 10% trajectory time compared to its rigid counterpart under the same planner. On the other, the CP planner has longer path lengths than A\* method in the simulation, similar to the observations in the physical test. However, the results show that the trajectories generated by the CP planner have doubled the clearances compared to other methods. To sum up,

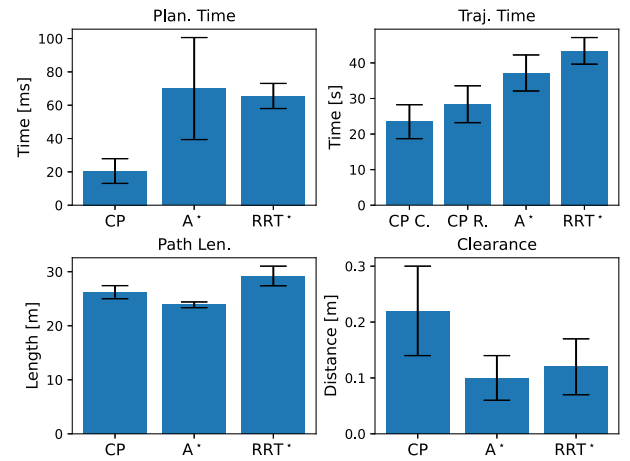


Fig. 9. Comparison metrics for simulated studies on cluttered maps.

**TABLE IV**  
COMPARISON METRICS FOR DIFFERENT ENVIRONMENTS

	Metrics	Offline	Online	Units
CP	Plan. Time	20.50 ± 6.43	16.78 ± 4.41	ms
	Traj. Gen. Time	0.079 ± 0.026	0.318 ± 0.122	s
	Traj. Time	23.48 ± 4.77	31.04 ± 5.27	s
A*	Path Len.	26.22 ± 1.21	28.64 ± 2.98	m
	Plan. Time	70.02 ± 30.61	38.51 ± 4.52	ms
	Traj. Gen. Time	32.80 ± 2.65	26.17 ± 7.40	s
A*	Traj. Time	37.17 ± 5.07	40.30 ± 4.37	s
	Path Len.	23.88 ± 0.53	24.25 ± 1.22	m

simulated results help demonstrate that our proposed CP planner can outperform collision-avoidance planning methods A\* and RRT\* in terms of planning time, trajectory time and path safety, despite longer path lengths than the A\* planner.

**2) Online Planning:** In support of our method's preliminary feasibility to apply in practical use cases, we also include a simulated study where the robot operates in partially known maps. The robot can localize obstacles only within a sensing range of 5 m, which is consistent with practice when the robot relies on cameras or (short-range lightweight and airborne) LiDAR sensors for localization. Both CP and A\* methods are run online with a replanning interval of 5 s. Similarly, ten random maps with 30 obstacles are studied for each planner.

We list the simulated results for both offline and online tasks in Table IV. Note that we also include the trajectory generation time, which stands for the process to convert waypoints to polynomial-based trajectories by solving a constrained optimization problem. Results show that both planning methods have lower planning time in the online task. The A\* planing method has a larger decrease, indicating high sensitivity to map size. Results also show that the CP method has lower planning time in both settings. Despite the increased trajectory generation time in partially known environments, the CP planner costs around 1% time of the A\* method in generating trajectories,

indicating improved efficiency. Both planners have longer trajectory time in the online task due to the limited knowledge of the map. Still, trajectories generated by the CP planner save about 23% of the time compared to A\* trajectories, despite longer path lengths in both settings.

## VII. CONCLUSION

In this work, we designed a lightweight compliant arm to sense contacts and reduce collision impact. Equipped with the integrated arm, we developed a novel impact-resilient aerial robot, named s-ARQ, to stabilize from high-speed collisions. Experimental results show that the compliant robot has only a 4% weight increase but around 40% impact reduction compared to a rigid counterpart. Further, when equipped with a real-time contact force estimator and a nonlinear motion controller, the compliant robot can handle collisions while attempting aggressive maneuvers, and stabilize from high-speed wall collisions at 3.0 m/s with a success rate of 100%. This impact resilience was also verified with pole obstacle collisions, as well as with different yaw and pitch angles. Our robot is found to result in better performance over both maximum collision speeds and success rates compared to other state-of-the-art methods reported in literature.

We also proposed and validated in both simulated and physical experiments a planning method for impact-resilient robots that prioritizes contacts to facilitate navigation. Physical tests and extended simulations demonstrate that our proposed compliant robot and CP planning method can accelerate the computation while achieving shorter trajectory time by relaxing velocity constraints. Despite having a larger path length due to the collision and follow-on recovery, the CP planner leads to higher clearances, indicating enhanced safety. Online planning tests in partially known environments were also studied. Simulated results further validated the efficiency of the proposed CP planner, with reduced planning and trajectory generation time, shorter trajectory time and increased clearances. Admittedly, the CP planner has longer trajectory time compared to collision-avoidance planning methods when applying velocities constraints. However, the significant saving in computational time and increased trajectory safety may outweigh the increasing path length limitation. The proposed CP planner thus provides positive results to study how to utilize contacts to facilitate navigation, especially when computational time is of essence.

There are multiple interesting directions to explore in future work. Herein, we assumed regular wall and pole obstacles; we plan to extend results to irregular obstacles in 3-D space. While collision handling involves basic motion control, it is possible to study force-control-based recovery methods on the compliant robot, such as impedance and admittance control [21]. Further, we plan to incorporate camera or laser distance sensors for odometry feedback and deploy impact-resilient aerial robots in outdoor environments. Lastly, we adopted a conservative velocity for traversing cluttered environments; studying collision-inclusive high-speed flight is also another direction of future research.

## REFERENCES

- [1] T. Bartelds, A. Capra, S. Hamaza, S. Stramigioli, and M. Fumagalli, "Compliant aerial manipulators: Toward a new generation of aerial robotic workers," *IEEE Robot. Automat. Lett.*, vol. 1, no. 1, pp. 477–483, Jan. 2016.
- [2] S. Kim, H. Seo, S. Choi, and H. J. Kim, "Vision-guided aerial manipulation using a multirotor with a robotic arm," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 1912–1923, Aug. 2016.
- [3] S. Park et al., "Odar: Aerial manipulation platform enabling omnidirectional wrench generation," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 4, pp. 1907–1918, Aug. 2018.
- [4] P. De Petris, H. Nguyen, M. Kulkarni, F. Mascarich, and K. Alexis, "Resilient collision-tolerant navigation in confined environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 2286–2292.
- [5] T. Lew et al., "Contact inertial odometry: Collisions are your friends," in *Proc. Robot. Res.: 19th Int. Symp.*, 2019, pp. 938–958.
- [6] Y. Mulgaonkar, A. Makineni, L. Guerrero-Bonilla, and V. Kumar, "Robust aerial robot swarms without collision avoidance," *IEEE Robot. Automat. Lett.*, vol. 3, no. 1, pp. 596–603, Jan. 2018.
- [7] K. Patnaik, S. Mishra, Z. Chase, and W. Zhang, "Collision recovery control of a foldable quadrotor," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, 2021, pp. 418–423.
- [8] J. Zha and M. W. Mueller, "Exploiting collisions for sampling-based multicopter motion planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 7943–7949.
- [9] Z. Lu, Z. Liu, and K. Karydis, "Deformation recovery control and post-impact trajectory replanning for collision-resilient mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 2030–2037.
- [10] Z. Lu, Z. Liu, M. Campbell, and K. Karydis, "Online search-based collision-inclusive motion planning and control for impact-resilient mobile robots," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 1029–1049, Apr. 2023.
- [11] P. Sareh, P. Chermprayong, M. Emmanuelli, H. Nadeem, and M. Kovac, "Rotorigami: A rotary origami protective system for robotic rotorcraft," *Sci. Robot.*, vol. 3, no. 22, 2018, Art. no. eaah5228.
- [12] J. Shu and P. Chirattananon, "A quadrotor with an origami-inspired protective mechanism," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3820–3827, Oct. 2019.
- [13] J. Zha, X. Wu, J. Kroeger, N. Perez, and M. W. Mueller, "A collision-resilient aerial vehicle with icosahedron tensegrity structure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 1407–1412.
- [14] Y. Chen, S. Xu, Z. Ren, and P. Chirattananon, "Collision resilient insect-scale soft-actuated aerial robots with high agility," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1752–1764, Oct. 2021.
- [15] P. H. Nguyen, K. Patnaik, S. Mishra, P. Polygerinos, and W. Zhang, "A soft-bodied aerial robot for collision resilience and contact-reactive perching," *Soft Robot.*, to be published, doi: [10.1089/soro.2022.0010](https://doi.org/10.1089/soro.2022.0010).
- [16] F. Ruiz, B. C. Arrue, and A. Ollero, "Sophie: Soft and flexible aerial vehicle for physical interaction with the environment," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 11086–11093, Oct. 2022.
- [17] S. Mintchev, S. de Rivaz, and D. Floreano, "Insect-inspired mechanical resilience for multicopters," *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1248–1255, Jul. 2017.
- [18] S. Savin, A. Al Badr, D. Devitt, R. Fedorenko, and A. Klimchik, "Mixed-integer-based path and morphing planning for a tensegrity drone," *Appl. Sci.*, vol. 12, no. 11, 2022, Art. no. 5588.
- [19] Z. Liu and K. Karydis, "Toward impact-resilient quadrotor design, collision characterization and recovery control to sustain flight after collisions," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 183–189.
- [20] S. Bellens, J. De Schutter, and H. Bruyninckx, "A hybrid pose/wrench control framework for quadrotor helicopters," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 2269–2274.
- [21] T. Tomić, C. Ott, and S. Haddadin, "External wrench estimation, collision detection, and reflex reaction for flying robots," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1467–1482, Dec. 2017.
- [22] B. Yüksel, C. Secchi, H. H. Büthoff, and A. Franchi, "A nonlinear force observer for quadrotors and application to physical interactive tasks," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, 2014, pp. 433–440.
- [23] C. D. McKinnon and A. P. Schoellig, "Unscented external force and torque estimation for quadrotors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 5651–5657.
- [24] D. Fan, K. Guo, S. Lyu, X. Yu, L. Xie, and L. Guo, "Quadrotor UAV: Collision resilience behaviors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 3, pp. 2092–2104, Jun. 2023, doi: [10.1109/TAES.2022.3211479](https://doi.org/10.1109/TAES.2022.3211479).



- [25] B. Stephens, H.-N. Nguyen, S. Hamaza, and M. Kovac, "An integrated framework for autonomous sensor placement with aerial robots," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 1, pp. 38–49, Feb. 2023.
- [26] H. W. Wopereis, J. J. Hoekstra, T. H. Post, G. A. Folkertsma, S. Stramigioli, and M. Fumagalli, "Application of substantial and sustained force to vertical surfaces using a quadrotor," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 2704–2709.
- [27] S. Hamaza, I. Georgilas, and T. Richardson, "An adaptive-compliance manipulator for contact-based aerial applications," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, 2018, pp. 730–735.
- [28] A. Praveen, X. Ma, H. Manoj, V. L. Venkatesh, M. Rastgaar, and R. M. Voyles, "Inspection-on-the-fly using hybrid physical interaction control for aerial manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 1583–1588.
- [29] A. Suarez, M. Perez, G. Heredia, and A. Ollero, "Cartesian aerial manipulator with compliant arm," *Appl. Sci.*, vol. 11, no. 3, 2021, Art. no. 1001.
- [30] M. Mote, M. Egerstedt, E. Feron, A. Bylard, and M. Pavone, "Collision-inclusive trajectory optimization for free-flying spacecraft," *J. Guid. Control Dyn.*, vol. 43, no. 7, pp. 1247–1258, 2020.
- [31] Y. Mulgaonkar, W. Liu, D. Thakur, K. Daniilidis, C. J. Taylor, and V. Kumar, "The tiercel: A novel autonomous micro aerial vehicle that can map the environment by flying into obstacles," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 7448–7454.
- [32] Z. Lu, Z. Liu, G. J. Correa, and K. Karydis, "Motion planning for collision-resilient mobile robots in obstacle-cluttered unknown environments with risk reward trade-offs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 7064–7070.
- [33] P. De Petris, M. Dharmadhikari, H. Nguyen, and K. Alexis, "Risk-aware motion planning for collision-tolerant aerial robots subject to localization uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 4561–4568.
- [34] Z. Liu, C. Mucchiani, K. Ye, and K. Karydis, "Safely catching aerial micro-robots in mid-air using an open-source aerial robot with soft gripper," *Front. Robot. AI*, vol. 9, 2022, Art. no. 1030515.
- [35] M. Fumagalli et al., "Developing an aerial manipulator prototype: Physical interaction with the environment," *IEEE Robot. Automat. Mag.*, vol. 21, no. 3, pp. 41–50, Sep. 2014.
- [36] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE (3)," in *Proc. IEEE Conf. Decis. Control*, 2010, pp. 5420–5425.
- [37] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2520–2525.
- [38] D. Brescianini, M. Hehn, and R. D'Andrea, "Nonlinear quadcopter attitude control: Technical report," ETH Zürich, Zürich, Switzerland, Tech. Rep. ethz-a-009970340, 2013.
- [39] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multi-threaded open source robotics framework for deeply embedded platforms," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 6235–6240.
- [40] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Proc. Int. Symp. J. Robot. Res.*, 2016, pp. 649–666.
- [41] M. Likhachev, G. J. Gordon, and S. Thrun, "Ara\*: Anytime A\* with provable bounds on sub-optimality," in *Proc. Adv. Neural. Inf. Process. Syst.*, vol. 16, 2003, pp. 767–774.
- [42] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [43] T. Schoels, P. Rutquist, L. Palmieri, A. Zanelli, K. O. Arras, and M. Diehl, "CIAO\*: MPC-based safe motion planning in predictable dynamic environments," *IFAC-Paper*, vol. 53, no. 2, pp. 6555–6562, 2020.