2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Quadrotor Collision Characterization and Recovery Control

Gareth Dicker, Fiona Chui, and Inna Sharf

***Abstract*— Collisions between quadrotor UAVs and the environment often occur, for instance, under faulty piloting, from wind gusts, or when obstacle avoidance fails. Airspace regulations are forcing drone companies to build safer drones; many quadrotor drones now incorporate propeller protection. However, propeller protected quadrotors still do not detect or react to collisions with objects such as walls, poles and cables. In this paper, we present a collision recovery pipeline which controls propeller protected quadrotors to recover from collisions. This pipeline combines concepts from impact dynamics, fuzzy logic, and aggressive quadrotor attitude control. The strategy is validated via a comprehensive Monte Carlo simulation of collisions against a wall, showing the feasibility of recovery from challenging collision scenarios. The pipeline is implemented on a custom experimental quadrotor platform, demonstrating feasibility of real-time performance and successful recovery from a range of pre-collision conditions. The ultimate goal of the research is to implement a general collision recovery solution as a safety feature for quadrotor flight controllers.**

SUPPLEMENTARY MATERIAL

A supplementary video for this work is available at:
`https://youtu.be/TJsZum_XKX4`

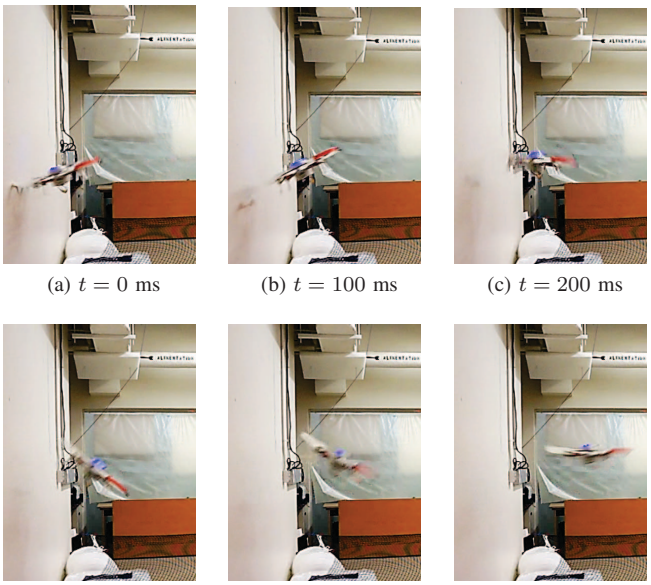
I. INTRODUCTION

A. Background and motivation

Currently, unmanned aerial vehicles (UAVs) in North America are only allowed to be manually flown within the line of sight of a pilot. In order to encourage airspace regulatory bodies to permit more autonomous UAV flight, the most important step is to improve UAV safety.

Quadrotor UAVs crash when, for instance, a gust of wind, poor piloting, or communication loss causes the vehicle - typically its propellers - to collide with another object. Propeller-environment collisions cause almost immediate failure of the onboard flight controller, which requires all four propellers to remain stable. To avoid this, drone companies are implementing propeller protection. When propeller protected quadrotors collide at low velocities, they can generally bounce away safely. However, at higher velocities or steep collision angles the flight controller can become compromised by the impulsive disturbance from the impact or by continued contact between the propeller protection (bumpers) and an object. The ability to recover from challenging collisions is a safety feature that will be critical to increasing overall UAV autonomy.

All authors are with the Department of Mechanical Engineering, McGill University, Montreal, QC, Canada. {gareth.dicker, fiona.chui}@mail.mcgill.ca, inna.sharf@mcgill.ca



(a) $t = 0$ ms (b) $t = 100$ ms (c) $t = 200$ ms

(d) $t = 300$ ms (e) $t = 400$ ms (f) $t = 500$ ms

Fig. 1. Quadrotor recovering from a collision with a wall. In (a), collision characterization is performed and recovery begins. The quadrotor re-orients (b-d) away from the wall, and then (e-f) returns to hover and stabilizes altitude.

B. Related work

Controllers for interaction between UAVs and their environment have been developed by several researchers. Docking and sliding control have been developed for ducted-fan miniature UAVs [1] and quadrotors with wire airframes [2] for inspection tasks. Notably, external wrench estimation was used for controlling 'collision reflexes' of a quadrotor, where the authors demonstrated a successful reflex away from a polystyrene block obstacle after an impact at 1.5 m/s, at level attitude [3].

Numerous fuzzy logic processes have been implemented and evaluated in robotics control, showing the effectiveness of fuzzy logic approaches in controlling nonlinear, uncertain systems. Robotic manipulators [4] and wheeled mobile robots [5] are some examples where a fuzzy logic process (FLP) has provided robust control, and also motivated our use of a FLP in the present context.

The recovery controller must be able to handle the quadrotor being in any orientation; therefore, linearized attitude control laws cannot be used [6]. A quaternion formulation of the attitude allows for nearly global attitude control of a quadrotor [7] and therefore suffices for recovery control, although other attitude descriptions in SE(3) could also be used [8]. Precise aerobatic maneuvers have been demonstrated

978-1-5090-4633-1/17/$31.00 ©2017 IEEE

under external motion capture feedback [9, 10]. However, they track the differentially flat parameters of position and yaw angle, which require a full pose estimate [11]. Our recovery control method is based on the recent work of Faessler et. al [12], which stabilizes a quadrotor from a wide range of unstable initial conditions to a hover attitude, assuming that only an attitude estimate is initially available.

C. Contributions

In this paper, we demonstrate a collision characterization and recovery control strategy for a propeller protected quadrotor in the case of a collision with a vertical wall, as depicted in Figure 1. This particular scenario was chosen for experimental testing; however, the strategy is formulated to be extensible for collisions with other types of obstacles.

In Section II we present an overview of the collision recovery strategy. Section III describes the collision characterization method using a FLP. Section IV provides a review of the control method and presents simulation results, and Section V presents experimental results. In addition, the supplementary video shows several successful experimental trials.

II. Collision Recovery Strategy: An Overview

The collision recovery pipeline consists of three consecutive phases, as shown in Figure 2. The first phase detects that a collision has occurred. The intensity of the collision is then characterized using a FLP. Finally, the output of the FLP characterization is sent as input to the recovery controller which returns the quadrotor to stable flight. These three phases occur in fast succession; the collision is detected and characterized within 20 ms of the actual occurrence, and the recovery control generally takes under 2 s.

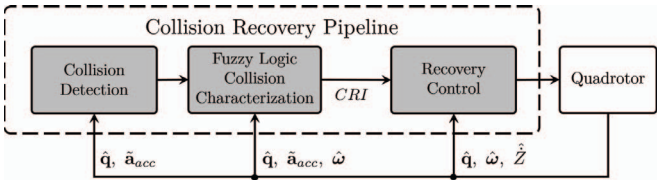


Fig. 2. Collision recovery pipeline and quadrotor plant

This strategy for collision recovery combines the two domains of fuzzy logic and aggressive/aerobatic attitude control to develop a recovery pipeline that is robust to the dynamic uncertainty of aerial collisions with walls. The choice of using a FLP for collision characterization creates a flexible framework for weighting the importance of different incoming conditions. The recovery control method was chosen to generate aggressive reactions which result in fast recovery time and consequently minimal altitude loss.

We note that simulation of the quadrotor collision contact dynamics, as described in detail in [13], was essential for the development and validation of the collision characterization and recovery control. All parameters used for simulation were based on those of the experimental platform, as described in Section V.

A. Nomenclature

The inertial frame is $\mathcal{F}_I = \{\mathbf{e}_X \ \mathbf{e}_Y \ \mathbf{e}_Z\}$ with $\mathbf{e}_Z$ pointing upwards and the body-fixed (quadrotor) frame is $\mathcal{F}_Q = \{\mathbf{e}_x \ \mathbf{e}_y \ \mathbf{e}_z\}$. The body-fixed frame is centered at the quadrotor center of mass, and is oriented such that $\mathbf{e}_x$ points outwards from the front of the vehicle and $\mathbf{e}_z$ points downwards when in the orientation of hover. We represent the attitude of the quadrotor relative to the inertial frame with the quaternion $\mathbf{q} = [q^{(w)} \ q^{(x)} \ q^{(y)} \ q^{(z)}]^T$. The quadrotor's position is $\mathbf{p} = [X \ Y \ Z]^T$ in the inertial frame, its acceleration is $\mathbf{a} = [a_X \ a_Y \ a_Z]^T$ as components in the inertial frame, and the body-frame angular velocity is $\boldsymbol{\omega} = [p \ q \ r]^T$. Vectors in the body-fixed frame are rotated into the inertial frame via the quaternion rotation operator $\odot$. Estimated values are denoted with a hat (e.g., $\hat{\mathbf{q}}$) and measured values are given a tilde (e.g., $\tilde{\mathbf{a}}_{acc}$).

B. Collision detection

Since the collisions are assumed to be only with vertical walls, as per expected impact dynamics, the collision detection phase checks for spikes in the horizontal component of acceleration. To estimate the horizontal acceleration, the accelerometer measurement $\tilde{\mathbf{a}}_{acc}$ is rotated into the inertial frame, and compensated with gravity $\mathbf{g}$ as:

$$\hat{\mathbf{a}} = \hat{\mathbf{q}} \odot \tilde{\mathbf{a}}_{acc} + \mathbf{g} \tag{1}$$

Then, the collision is detected when the magnitude of the inertial acceleration horizontal components reaches a threshold:

$$\text{Collision detection flag} = \begin{cases} 1, & \left\| [\hat{a}_X, \ \hat{a}_Y]^T \right\| > 1 \ g \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where the threshold of 1 g was chosen to be higher than the normal free-flight noisy horizontal acceleration reading, and still detect very mild collisions.

At the time of collision detection t_d, the wall normal $\hat{\mathbf{e}}_N$ is also estimated for use in the remaining two pipeline phases as follows:

$$\hat{\mathbf{e}}_N = [\hat{a}_X, \ \hat{a}_Y]^T \ / \left\| [\hat{a}_X, \ \hat{a}_Y]^T \right\| \tag{3}$$

C. Fuzzy logic collision characterization

When the collision is detected, the characterization phase begins. This phase executes a FLP on four measured or estimated inputs, these having been determined with simulations to be significant indicators of the post-collision response intensity if recovery control were not engaged. The response is characterized by the single FLP output: Collision Response Intensity (CRI).

D. Recovery control

The CRI and $\hat{\mathbf{e}}_N$ are sent to the final recovery pipeline phase where they are mapped to a reference acceleration of the center of mass of the quadrotor. The recovery control passes through three stages, the first of which tracks this reference acceleration. In the second and third stages, the quadrotor attempts to achieve an upright attitude and zero

vertical velocity before exiting the recovery pipeline. In this work, we do not attempt to stabilize horizontal velocity, since horizontal motion estimation requires additional sensing, and it was successfully demonstrated with vision-based feedback in [12].

III. Collision Characterization Using Fuzzy Logic

The collision characterization phase provides an estimate of the control effort required to successfully recover from the collision. Using Mamdani-type fuzzy logic to characterize the response provides the recovery control phase with important information about the intensity of the impact, which is a challenging task given the unexpected and unpredictable nature of the collisions [14]. To implement the FLP, a range of simulated quadrotor collisions were studied. Particularly, correlations between the simulated sensor data and the quadrotor response were studied in order to gain the 'expertise' necessary to design and tune the FLP inputs and parameters.

Following t_d, the four inputs for the FLP are calculated and treated as the 'crisp' or numerical inputs into the process. Then, the three steps of the FLP: fuzzification, inference mechanism, and defuzzification, produce the 'crisp' output, or CRI. While this FLP was designed and tuned for our experimental platform, adjustments can easily be made to the input calculation and fuzzification parameters for use on other propeller protected quadrotors, while the inference mechanism and defuzzification parameters are generally usable cross-platform.

A. Fuzzy logic process inputs

Given that intense collisions can cause the quadrotor to spin out of control as its angular momentum increases over time, the FLP characterization should commence as close to the collision time as possible. However, there is a trade-off between how soon the FLP inputs are calculated after t_d, and how well they act as indicators of collision response.

For our platform, the minimum first bumper contact duration seen in uncontrolled collision experiments was ~ 30 ms, so it is reasonable that the FLP inputs can be calculated up to as much as 30 ms after the bumper first touches the wall. For other platforms, this minimum contact duration would need to be tested and reflected in the calculation times for the inputs below.

Input 1) Accelerometer horizontal magnitude: This FLP input is an indicator of the collision normal force magnitude. It is the magnitude of the x and y components of $\tilde{\mathbf{a}}_{acc}$ and is calculated 8 ms after t_d.

Input 2) Pre-collision inclination: The inclination angle ζ is the angle between the projection of the body-fixed $-z$ axis onto the vertical plane normal to the wall, and the inertial Z axis. This angle is positive if the quadrotor is directed into the wall, and negative if directed away. In order to calculate the *pre-collision* inclination, the quaternion $\hat{\mathbf{q}}$ used in (4) must be stored from a few estimation samples before t_d. The computation for ζ in (4) makes use of the tangent direction to the contact plane $\hat{\mathbf{e}}_T = \mathbf{e}_Z{}^{\times}\hat{\mathbf{e}}_N$, and the body-fixed $-z$ axis rotated into the inertial frame $\mathbf{e}_z^I = \mathbf{q} \odot (-\mathbf{e}_z)$. Since $\hat{\mathbf{q}}$ is saved from before the collision, and $\hat{\mathbf{e}}_N$ is available at t_d, this input can be generated at t_d.

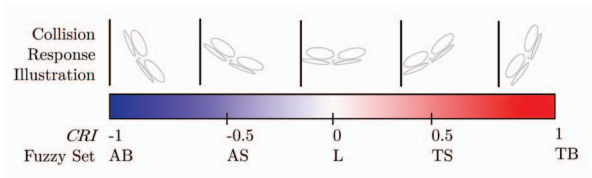


Fig. 3. Collision Response Intensity (CRI) scale and FLP output fuzzy sets with corresponding post-collision quadrotor illustrations

$$\zeta = \text{sign}(\zeta) \cdot \cos^{-1}\left(\frac{\left(\mathbf{e}_z^I - \left((\mathbf{e}_z^I)^T\hat{\mathbf{e}}_T\right)\hat{\mathbf{e}}_T\right)^T \mathbf{e}_Z}{\left\|\mathbf{e}_z^I - \left((\mathbf{e}_z^I)^T\hat{\mathbf{e}}_T\right)\hat{\mathbf{e}}_T\right\|}\right) \quad (4)$$

Input 3) Flipping direction angle: The flipping direction angle indicates if the quadrotor is flipping towards or away from the wall. The angle is between $\hat{\mathbf{e}}_N$ and the flipping direction $(\hat{\mathbf{q}} \odot \hat{\boldsymbol{\omega}})^{\times}\mathbf{e}_Z$ projected onto the XY plane. It is calculated 8 ms after t_d.

Input 4) Angular velocity horizontal component magnitude: This input is an indicator of the quadrotor's flipping motion, as larger horizontal angular velocities are seen when the vehicle spins out of control. The FLP input is simply the magnitude of $[\hat{p} \;\; \hat{q}]^T$, and is taken 12 ms after t_d.

B. Fuzzy logic process output: Collision Response Intensity

The output of the FLP is the CRI, a numerical value on a scale from -1 to 1. Negative and positive CRI values characterize the quadrotor collision response as flipping away and towards the wall respectively. A larger CRI magnitude represents a more intense response: the quadrotor spins out of control more rapidly and is more difficult to recover. Figure 3 illustrates the immediate post-collision orientation of the quadrotor which corresponds to different values on the CRI scale.

C. Fuzzy logic process parameters

1) Fuzzification: The first step in the FLP takes each of the four inputs and assigns them degrees of membership for each of the fuzzy sets, depending on the membership functions defined in Figure 4. For example, an Accelerometer Horizontal Magnitude of 5 g would have a degree of membership of 0.5 for both the Medium and High fuzzy sets. These degrees of membership for each of the four 'crisp' inputs are now said to be the 'fuzzy' inputs.

Triangular and trapezoidal membership functions were chosen for both the fuzzification and defuzzification steps because they are sufficient for segmenting the membership and are more computationally efficient than other non-linear membership functions, which is important for real-time flight control.

The specific fuzzy set definitions (i.e. where triangle and trapezoid vertices are placed) were chosen based on the range of 'crisp'input values seen during simulated and experimental collisions without recovery control. Inputs 1 and 4 are directly affected by the input calculation times chosen in III-A, as they are both sensor magnitudes that change over time. With the exception of input 3, all the membership functions are specific to our platform, and would need to be re-tuned for collision characterization of a different platform.

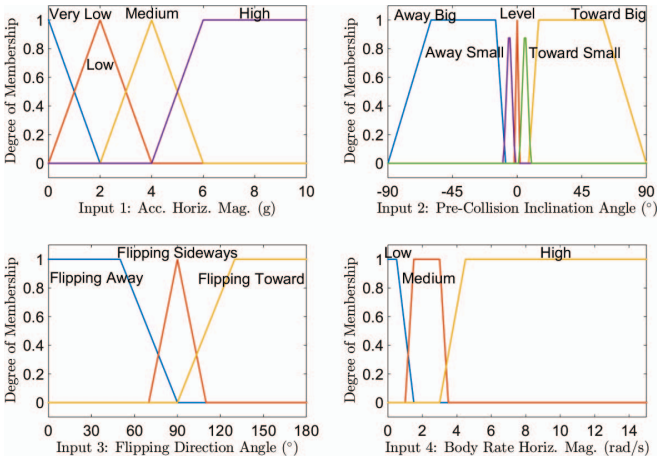


Fig. 4. Fuzzy logic process input membership functions

2) Inference mechanism: The Mamdani-type inference mechanism turns the 'fuzzy' inputs from the first step into a 'fuzzy' output for the last step. This is done via IF-THEN rules, where certain combinations of the input fuzzy sets will result in an output fuzzy set. The degrees of membership from the fuzzification step translate to this step, where there are degrees of membership for the fuzzy output sets as well. In our inference system, two rule sets as defined in Table I are used with equal weighting, where the output fuzzy sets abbreviations are defined in Figure 5. The use of different rule sets and their associated weightings were investigated in [15], and the inference system producing the most effective CRI, as evaluated with the method in III-D, are presented here.

3) Defuzzification: This step turns the fuzzy output degrees of membership into a 'crisp' output, via the membership function defined in Figure 5. Collision response correlations to the output fuzzy sets are also illustrated in Figure 3.

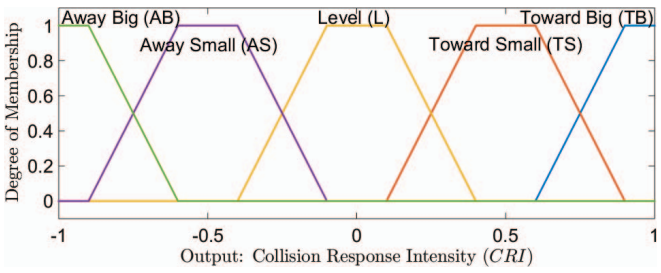


Fig. 5. Fuzzy logic process output membership function

TABLE I
FUZZY RULE SETS

Fuzzy Rule Set 1

Input 1	**Input 2**				
	Towards Big	Towards Small	Level	Away Small	Away Big
Very Low	L	L	L	L	L
Low	TS	TS	L	AS	AS
Med.	TB	TS	L	AS	AB
High	TB	TB	L	AB	AB

Fuzzy Rule Set 2

Input 4	**Input 3**		
	Flipping Towards	Flipping Sideways	Flipping Away
Low	L	L	L
Med	TS		AS
High	TB		AB

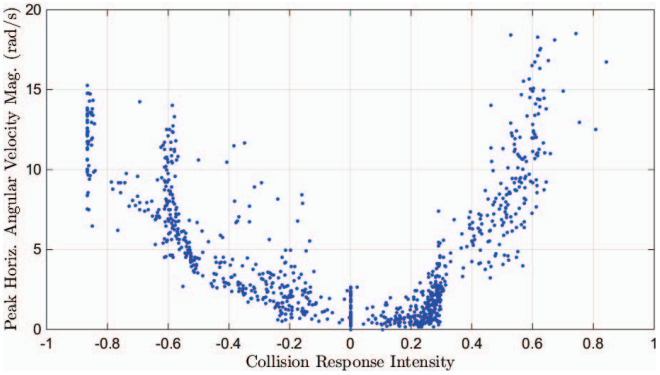


Fig. 6. Monte Carlo simulation results for quadrotor collisions without recovery control

D. Collision characterization validation

The FLP collision characterization was validated using a Monte Carlo simulation of 1000 quadrotor collisions with a wall, not using recovery control. The Euler angles roll, pitch and yaw, $[\phi\ \theta\ \psi]$, at impact, were randomized to be within $-15° < \phi < 15°$ and $-45° < \theta\ ,\ \psi < 45°$. The incoming velocities were limited between 0.5 and 2.5 m/s, since velocities faster than this would damage the propeller protection of the experimental platform. Figure 6 plots the CRI against the first peak $||[\hat{p}\ \ \hat{q}]^T||$ value after the first bumper contact. This peak occurs well after the CRI is computed, and is a good numerical indicator of the quadrotor's response. The plot shows that larger horizontal body rates are measured for collisions with a higher CRI magnitude, and that the CRI is a suitable measure of quadrotor response that can be used by the recovery controller.

IV. COLLISION RECOVERY CONTROLLER

A. Recovery control law

We chose to adapt the control law developed by Faessler et. al. [12] to directly track a reference *acceleration*, $\mathbf{a}_{ref}$,

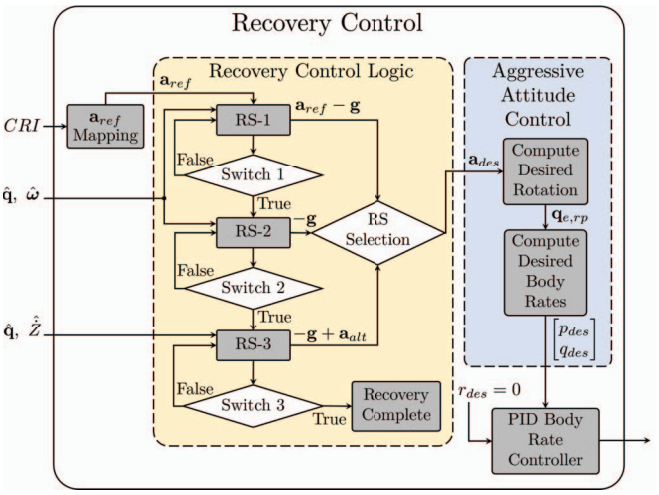


Fig. 7. Recovery stage logic and control flow

as opposed to a reference position or velocity. Given that a quadrotor is an under-actuated and non-holonomic system [16], in order to track a reference acceleration, the vehicle must first orient itself to provide thrust in the correct direction. This is accomplished during the first stage of the recovery control through the following sequence of steps which results in the desired angular velocity set-point for the vehicle.

1) Mapping CRI to reference acceleration: The magnitude of $\mathbf{a}_{ref}$ is mapped from the CRI output of the FLP by a proportional gain as:

$$\|\mathbf{a}_{ref}\| = k_a \cdot CRI \tag{5}$$

and the vector $\mathbf{a}_{ref}$ is then oriented along the wall normal, giving:

$$\mathbf{a}_{ref} = \|\mathbf{a}_{ref}\| \cdot \hat{\mathbf{e}}_n \tag{6}$$

2) Computing desired acceleration: The quadrotor must provide thrust in a direction and magnitude such that its resultant with the force of gravity is equivalent to $\mathbf{a}_{ref}$, and therefore, the acceleration due to this thrust, denoted as $\mathbf{a}_{des}$, is given by:

$$\mathbf{a}_{des} = \mathbf{a}_{ref} - \mathbf{g} \tag{7}$$

3) Generating desired angular velocity: Taking $\mathbf{a}_{des}$ as the control input, the method of [12] was used to generate desired body rates $\boldsymbol{\omega}_{des} = [p_{des}\ q_{des}\ r_{des}]^T$, that were tracked under PID control. This method uses the difference between the current and the desired $\mathbf{e}_z$ to compute an error quaternion in roll and pitch, denoted $\mathbf{q}_{e,rp}$. The desired roll and pitch angular velocity components are then computed as follows:

$$[p_{des}\ q_{des}]^T = \text{sign}(q^{(w)}_{e,rp}) \cdot k_{err} \cdot \left[q^{(x)}_{e,rp}\ q^{(y)}_{e,rp}\right]^T \tag{8}$$

where k_{err} is a proportional gain scaling the aggressiveness of the control. The desired yaw component, r_{des}, is set to zero because rotations about $\mathbf{e}_z$ are undesirable.

TABLE II
RECOVERY CONTROL SWITCH CONDITIONS

Stage	$\mathbf{a}_{des}$	Switch Condition
RS-1	$\mathbf{a}_{ref} - \mathbf{g}$	$abs(\mathbf{q}^{(x,y)}_{e,rp}) < \mathbf{q}^{(x,y)}_{switch} \ \wedge\ \boldsymbol{\omega} < \boldsymbol{\omega}_{switch}$
RS-2	$-\mathbf{g}$	$abs(\mathbf{q}^{(x,y)}_{e,rp}) < \mathbf{q}^{(x,y)}_{switch} \ \wedge\ \boldsymbol{\omega} < \boldsymbol{\omega}_{switch}$
RS-3	$\mathbf{a}_{alt} - \mathbf{g}$	$abs(\dot{Z}) < \dot{Z}_{switch}$

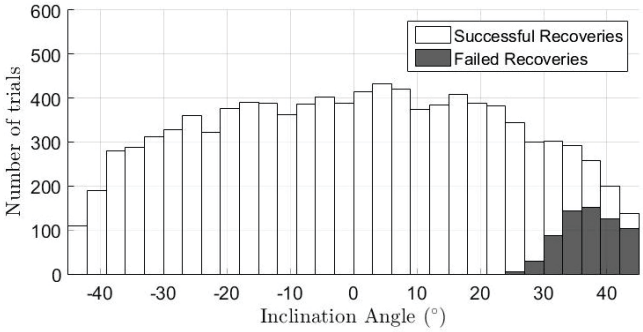


Fig. 8. Inclination angles for successful and failed recoveries in Monte Carlo simulation

B. Recovery control logic

As noted earlier, the recovery control consists of three stages, with logic to trigger switching between stages shown in Figure 7 and the corresponding desired acceleration and switch conditions summarized in Table II. Switches 1 and 2 check for bounds on orientation and angular velocity, while Switch 3 checks vertical velocity.

In the Recovery Stage 1 (RS-1), $\mathbf{a}_{des}$ is computed using $\mathbf{a}_{ref}$ as per (5) - (7). The controller attempts to track this $\mathbf{a}_{des}$ until the condition of Switch 1 is met. RS-2 then attempts to bring the quadrotor to an orientation of hover by setting $\mathbf{a}_{ref}$ to zero. Finally, the third stage (RS-3) stabilizes the quadrotor's altitude, so $\mathbf{a}_{des}$ is offset by the output $\mathbf{a}_{alt}$ of the altitude controller, expressed here as an acceleration for simplicity. Once Switch 3 evaluates True, the recovery is complete.

C. Monte Carlo simulation

The collision recovery pipeline was first validated by running a 10,000 trials Monte Carlo simulation using the same set of initial conditions in attitude and incoming velocity as employed in Section III-D and the controller parameters summarized in Table III. The results showed 93.5% of the trials completing RS-1 within 3 s and 91.9% completing RS-3, that is, fully recovered responses.

The 6.5% which did not reach RS-2 had inclination angles toward the wall of greater than $25°$, as indicated in Figure 8. Failure cases occurred when the quadrotor became vertically flush with the wall, causing a 'stuck' condition where the unidirectional propellers could not lower their RPM sufficiently to orient the vehicle away from the wall. For all other cases, the recovery control pipeline succeeded in detecting, characterizing and recovering to stable hover.

Fig. 9. Propeller protected experimental quadrotor

V. Experiments on Collision Recovery

A set of 22 experimental trials was performed on the custom quadrotor shown in Figure 9. The range of testable pre-collision conditions was limited due to tranverse flexibility of the propeller protective bumpers, which for high-inclination orientations resulted in propellers directly impacting the wall, which compromises control. Therefore, a subset of the pre-collision conditions from simulation was tested on the experimental platform.

A. Experimental platform

A 1.1 kg experimental platform was designed on a carbon fiber frame. Propeller protection consists of 3D printed nylon alloy bumpers which extend around each of the propellers, and two carbon fiber tubes connecting each bumper to its adjacent bumper (see Figure 9). 1400kV motors with 8" propellers from DJI provide thrust and torque factors of approximately 8.7×10^{-8} N s^2/rad^2 and 8.7×10^{-9} Nm s^2/rad^2, respectively. A Pixhawk micro-controller runs the open source px4 flight stack which was modified and extended to support the collision recovery pipeline. The px4's state estimator fuses the Pixhawk's embedded IMU, barometer, and magnetometer to provide an attitude and altitude estimate. Its flight controller runs at 250 Hz and all flight data is logged onboard at the same rate. A Vicon motion capture system was used to measure the quadrotor's position and orientation at 60 Hz. A bandpass filter with cut-off frequencies between 10 Hz and 50 Hz was used to estimate velocity in post-processing.

Takeoff was performed manually in the Pixhawk's altitude control mode. The quadrotor was flown to hover at varying distances away from the wall, and was then directed into the wall. The pitch angle into the wall was set manually just prior to collision, while roll and yaw were kept near zero. The velocity, $\dot{X}$, at collision was kept between 1.0 and 2.5 m/s.

Following collision characterization, which took 16 ms, the flight controller switched into fully autonomous recovery control. Upon recovery, manual control was returned to the pilot.

B. Results

The trials were divided into two sets of 11 trials each: level and inclined, as shown in Table IV. The pitch angle into the wall $\tilde{\theta}$ for the level and inclined collisions were kept near 0° and 15° respectively. In Table IV, μ is the mean and σ is the standard deviation of each trial set. The CRI values for the inclined collisions are significantly and consistently higher than those for level collisions. Level collisions produce almost no change in height ($\Delta\tilde{Z}$), while inclined collisions actually cause slight gains in height. Also, level collisions incur almost no horizontal change ($||\Delta\tilde{X}, \Delta\tilde{Y}||$) from their pre-collision positions, while inclined collisions, which had to track higher CRI values, show up to about one meter of horizontal displacement before recovery is achieved.

TABLE III

Recovery controller gains and switching parameters

Parameter	Value	Units
k_a	$\begin{cases} g, & \text{if } CRI >= 0 \\ g/2, & \text{otherwise} \end{cases}$	m/s^2
k_{err}	15	rad/s
$\boldsymbol{\omega}_{switch}$	$[0.5\ 0.5\ 0.5]^T$	rad/s
$\dot{Z}_{switch}$	0.2	m/s
$\mathbf{q}^{(x,y)}_{switch}$	$[0.3\ 0.3]^T$	

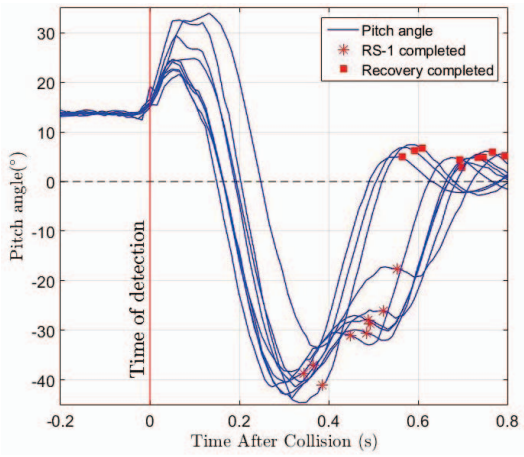


Fig. 10. Pitch angle for inclined collisions with each t_d aligned to coincide at time zero. Note that RS-3 begins and ends immediately because the vertical velocity was never outside of Switch 3's range in experiments.

TABLE IV

Experimental Result Averages

Trial Set	Value	$\tilde{\theta}$ (°)	$\dot{\hat{X}}$ (m/s)	CRI	$\Delta\tilde{Z}$ (m)	$\|\|\Delta\tilde{X}, \Delta\tilde{Y}\|\|$ (m)
Level	μ	0.03	1.49	0.08	-0.01	0.16
	σ	1.17	0.46	0.28	0.03	0.21
Inclined	μ	15.87	1.43	0.68	0.24	0.75
	σ	1.20	0.25	0.12	0.15	0.33

Figure 10 shows the pitch angle under recovery control for the inclined set, where the t_d for each trial is aligned to coincide at time zero. It can be seen that the quadrotor experiences nearly a 35° pitch into the wall before the recovery controller reorients the vehicle in the opposing direction and finally returns it to hover. In every trial, the quadrotor successfully passes through all stages of recovery,

taking about 0.4 s to complete RS-1 and 0.7 s to complete the recovery pipeline.

VI. Conclusions and Future Work

A strategy for collision recovery has been developed and implemented on a propeller protected quadrotor. We performed extensive simulations to validate the phases of the recovery pipeline, and produced a working solution on a real quadrotor. This strategy, combining fuzzy logic and aggressive attitude control, shows promising performance for collisions with walls, demonstrating high success rates of recovery for collision speeds up to 2 m/s and inclination angles up to 17°. The strategy can be extended to handle other obstacle types, such as poles, and for grazing collisions. Outlier failure cases can be reduced by increasing the upper and lower saturation limits of the propeller speeds, or by investigating the use of bi-directional propellers. Building on this work in the future, we plan to investigate iterative learning to tune both fuzzy parameters and control parameters for quadrotors with different inertia, thrusting capabilities, and geometries. Detection of motor failures resulting from collisions could also be integrated into the recovery pipeline.

Acknowledgments

This work was supported by the National Sciences and Engineering Research Council (NSERC) Canadian Field Robotics Network (NCFRN) and the McGill Engineering Undergraduate Student Masters Award (MEUSMA). The authors would like to thank Sebastian Schell and Linus Lehnert for their involvement in building and designing the experimental quadrotor platform, as well as Thomas Fuehrer for assistance on performing the experimental tests.

References

[1] L. Marconi, R. Naldi, and L. Gentili, "Modelling and control of a flying robot interacting with the environment," *Automatica*, vol. 47, no. 12, pp. 2571–2583, 2011.

[2] K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, "Aerial robotic contact-based inspection: planning and control," *Autonomous Robots*, vol. 40, no. 4, pp. 631–655, 2015.

[3] T. Tomić and S. Haddadin, "A unified framework for external wrench estimation, interaction control and collision reflexes for flying robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4197–4204, 2014.

[4] C. M. Lim and T. Hiyama, "Application of fuzzy logic control to a manipulator," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 5, pp. 688–691, 1991.

[5] T. Das and I. N. Kar, "Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots," *IEEE Trans. on Control Systems Technology*, vol. 14, no. 3, pp. 501–510, 2006.

[6] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proc. of the AIAA Guidance, Navigation, & Control Conf.*, vol. 2, 2007.

[7] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "Quaternion-based hybrid control for robust global attitude tracking," *IEEE Trans. on Automatic Control*, vol. 56, no. 11, pp. 2555–2566, 2011.

[8] T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear robust tracking control of a quadrotor uav on se (3)," *Asian J. of Control*, vol. 15, no. 2, pp. 391–408, 2013.

[9] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE Int. Conf. on Robotics & Automation (ICRA)*, pp. 2520–2525, 2011.

[10] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The Int. J. of Robotics Research*, p. 0278364911434236, 2012.

[11] D. Zhou, "Quadrotor dynamics and the differential flatness theory based aggressive control," tech. rep., Tech. Rep., Boston University, 2013.

[12] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2015.

[13] F. Chui, G. Dicker, and I. Sharf, "Dynamics of a quadrotor undergoing impact with a wall," in *IEEE Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, pp. 717–726, 2016.

[14] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. of man-machine studies*, vol. 7, no. 1, pp. 1–13, 1975.

[15] F. Chui, "Quadrotor collision dynamics and fuzzy logic characterization," Master's thesis, McGill University, 2016.

[16] H. Bouadi, M. Bouchoucha, and M. Tadjine, "Sliding mode control based on backstepping approach for an uav type-quadrotor," *World Academy of Science, Engineering and Technology*, vol. 26, no. 5, pp. 22–27, 2007.