2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Simplifying Aerial Manipulation Using Intentional Collisions

Mark Nail[1], Nick Jänne[2], Olivia Ma[1], Gabriel Arellano[3], Ella Atkins[4], and R. Brent Gillespie[1]

***Abstract*—Aerial manipulation describes a process that includes physical interaction between an unmanned aircraft system (UAS) and its environment. We aim to apply aerial manipulation to sample leaves and small branches from rain forest trees. Current approaches to aerial manipulation involve extended periods of UAS-environment interaction, during which forces and moments can lead to a loss in attitude or position control in underactuated multicopters. By adapting intelligent foot placement strategies found in dynamically stable hopping robots, this work proposes a strategy involving carefully managed intentional collisions between the UAS and its environment. We designed an attitude controller denoted a Velocity Matching controller that aligns a UAS-mounted pogo-stick foot with the center of mass velocity vector during collision approach to maximize UAS ability to recover a hover state after collision. We propose the use of a flight envelope involving altitude and horizontal speed states to assess recoverability prior to initiating each approach to collision. We identify this flight envelope from a simulation study built on a model of flight in Conventional Waypoint Following and Velocity Matching control modes as well as a model of collision response. Experimental flight testing evaluates the simulation-based envelope resulting in an actual envelope that is somewhat smaller but similarly shaped to the envelope identified in simulation.**

I. Introduction

Unmanned aircraft systems (UAS) are increasingly deployed to complete tasks that are out of reach or dangerous for humans to complete. UAS have been typically deployed for monitoring, surveillance, and sensing tasks [1]. UAS have also been developed to physically interact with their environment under three strategies. The first physical interaction strategy involves deploying payloads in flight, avoiding interaction forces from the environment, e.g., dropping ignition spheres for prescribed burns [2]. The second strategy involves landing prior to interaction to gain a stable base of support for more complicated tasks, e.g., deploying sensors underground [3]. The third strategy involves physical interaction with the environment while in flight, e.g., nailing shingles to a roof using a UAS as presented in [4]. The term *aerial manipulation* is reserved for in-flight physical interaction, the third strategy named above [5].

We aim to apply aerial manipulation for sampling leaves and small branches from within a dense tree canopy to improve the unreliable and often unsafe methods used to study the health of rainforests, one of the most bio-diverse ecosystems on the planet [6] [7]. UAS have been previously developed for tree sampling by the teams of Charron [8] and Krisanski [9]. The DeLeaves platform [8] performs well in forests without subcanopies, where the majority of trees can be sampled from above (e.g. evergreen forests). The UAS design documented in [9] approaches a tree for sampling from the side, allowing it to collect samples at any altitude. However, challenges arise for both systems in the rainforest, where density and stratification of vegetation make both navigation and manipulation difficult.

The simplest forms of aerial manipulation use a non-actuated, rigidly attached limb for UAS-environment interactions, similar to the "boomcopter" presented in [10]. This type of contact through a rigid limb is similar to leaning against a wall with one's arm. While rigid limb contact can provide benefits to a UAS by transferring force onto a rigid support, it compromises disturbance rejection and attitude control because of constraints between the UAS and the environment.

More complex aerial manipulation systems use actuated arms to combat disturbances as reviewed in [11]. Robotic arms make increasingly complicated tasks feasible for aerial manipulation, including opening doors [12] or picking up items while on the move [13]. However, increased actuation comes at the cost of overall system weight, energy demand, and changing flight dynamics that arise from changes in arm configuration [11].

Approaches that take inspiration from nature hold significant promise, allowing for perching and grasping to be completed with the same limb [14]. For interacting with stable platforms like industrial piping [15], perching works well and adds stability. On the other hand, the unpredictability of a tree branch's mechanical behavior makes perching in the rainforest a difficult task. Additionally, attitude control and disturbance rejection can be compromised during perching interactions. We hypothesize that minimizing overall interaction time will maximize attitude control of the UAS, thus minimizing risks induced by UAS-environment interactions.

Raibert-style, single-legged hopping robots locomote through short "hopping" interactions between the robot's

[1]Mark Nail, Olivia Ma, and R. Brent Gillespie are with the Robotics Department, University of Michigan, Ann Arbor, MI 48105, USA {mrnail, olivma, brentg}@umich.edu

[2]Nick Jänne is with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48105, USA njanne@umich.edu

[3]Gabriel Arellano is with the Department of Ecology and Evolutionary Biology, University of Michigan, Ann Arbor, MI 48105, USA garellan@umich.edu

[4]Ella Atkins is with the Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA 24061, USA {ematkins}@vt.edu

This projected was funded through the University of Michigan's MCubed Program.

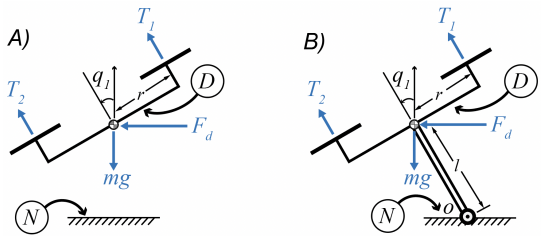

Fig. 1. A) 2-D model of UAS in flight experiencing a horizontal disturbance. B) 2-D model of UAS in one-legged perch experiencing a horizontal disturbance.

foot and the ground [16]. The hopping strategy minimizes interaction time and manages each hop's impulse, magnitude, and direction to maintain stability. Control derives from the use of intelligent foot placement. Unlike conventional multi-jointed walking robots, the hopping motion in Raibert's robots allows for simpler mechanical limbs and simpler control [17]. We take inspiration from these hopping robots, equipping our UAS with a pogo stick for short, predictable UAS-environment interactions. A pogo stick-equipped UAS was proposed in [18], although their approach used vertical bouncing as a more energy efficient substitute to hovering, as opposed to bouncing as a means of aerial manipulation.

This paper extends bouncing robots into the domain of aerial manipulation, adopting an attitude-only controller for foot placement, and proposing flight envelopes to manage the approach to collision in a manner that assures the recovery of hover after collision. This paper contributes the two-dimensional flight and perching model that drives our collision strategy (II), the attitude-only controller that enables intentional and recoverable collision (III), the design of a python-base simulator that evaluates the collision strategy (IV), and simulation and experimental flight envelopes for velocity mode safety checking (V).

II. Model Development

A. Flight Dynamics in 2-D

To quantify any compromise in the ability of a perched UAS to reject disturbances, we develop simple 2-D models of a UAS either in flight or perched with a single leg on fixed ground. Figure 1A shows a planar model of a UAS in flight while Fig. 1B shows the same UAS perched with a single central leg on the ground. Both systems are acted upon by a horizontal disturbance force F_d, a vertical force of gravity mg, and two thrusts T_1 and T_2 fixed in the UAS frame. Authority over the pitch angle q_1 is available from the thrust difference $T_{dif} = T_1 - T_2$, while the total thrust $T_{tot} = T_1 + T_2$ drives changes in positions p_x, p_z, the following comes from a simplification of the 3-D dynamics presented in [19].

A sum of forces and sum of moments acting on the UAS in flight can be used to produce the equations of motion that govern flight

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{p}_z \\ \ddot{p}_x \end{bmatrix} = \begin{bmatrix} r\frac{T_{dif}}{J} \\ g - \frac{T_{tot}}{m}\cos q_1 \\ \frac{T_{tot}}{m}\sin q_1 - \frac{F_d}{m} \end{bmatrix}, \tag{1}$$

where J is the moment of inertia, m is the UAS mass, and g is acceleration due to gravity.

In the 2-D flight case, the UAS can pitch toward the disturbance force F_d and maintain its position. Authority over pitch acceleration $\ddot{q}_1$ is available using T_{dif} while unimpeded by F_d. In turn, the disturbance F_d can be counteracted using a component of T_{tot} even while maintaining null horizontal acceleration $\ddot{p}_x$.

The situation is different while perching. The connection at the point of contact O between the UAS and ground constrains the UAS position to a function of pitch q_1 and leg length l.

$$\begin{bmatrix} p_x \\ p_z \end{bmatrix} = \begin{bmatrix} l\sin q_1 \\ l\cos q_1 \end{bmatrix}. \tag{2}$$

A sum of forces and sum of moments acting on the UAS along with a consideration of the constraint Eqs. 2 can be used to produce the equation of motion that governs pitch during perch

$$\ddot{q}_1 = \frac{F_d l\cos q_1 + mgl\sin q_1 + rT_{dif} + M_o}{J}. \tag{3}$$

During perch, all forces acting on the UAS contribute to the moment necessary to maintain a pitch angle. Due to the direct coupling between pitch q_1 and position p_x, p_y, disturbance rejection when coupled to the environment is less direct than in the flight case. Motor forces must generate moments for rejection, instead of directly using their forces as restoring forces. This complication in disturbance rejection is fundamental in motivating this paper's collision approach to aerial manipulation. By limiting total interaction time to be near instantaneous during a collision, disturbance rejection can be undertaken during flight.

B. Flight Dynamics in 3-D

For the 3-D case, we consider the full dynamics of the hexacopter platform. As presented in [19] the following set of equations govern the accelerations $\ddot{p}_x$, $\ddot{p}_y$, $\ddot{p}_z$ in the global frame, with I_{xx}, I_{yy}, I_{zz} as central moments of inertia with respect to the body axes, with ϕ, q_1, and ψ as roll, pitch, and yaw, respectively, and with α, β, γ as the angular velocities with respect to the body axes.

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{p}_z \end{bmatrix} = \begin{bmatrix} T(\cos\psi\sin q_1\cos\phi + \sin\psi\sin\phi)/m \\ T(\sin\psi\sin q_1\cos\theta - \cos\psi\sin\theta)/m \\ g - T\cos\theta\cos\phi/m \end{bmatrix}, \tag{4}$$

with the following angular rates and accelerations:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin q_1 \\ 0 & \cos\phi & \cos\theta\cos\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{q}_1 \\ \dot{\psi} \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})\beta\gamma/I_{xx} \\ (I_{zz} - I_{xx})\alpha\gamma/I_{yy} \\ (I_{xx} - I_{yy})\gamma\beta/I_{zz} \end{bmatrix} + \begin{bmatrix} \tau_\phi/I_{xx} \\ \tau_\theta/I_{yy} \\ \tau_\psi/I_{zz} \end{bmatrix}. \tag{6}$$

It is apparent from these equations that not only pitch q_1 but also roll ϕ and yaw ψ angles contribute to changes in world-frame position p_x and velocity $\dot{p}_x$, whereas only pitch contributes in the 2-D flight dynamics (see Eq. 1). However,

we are leveraging our cascaded PID controller (see Fig. 3 below) to maintain 0° pitch and yaw throughout flight tests.

C. Collision Response

For modeling collisions in simulation, we have chosen the rigid contact model described in [20]. This rigid body collision response model offers predictions that match experiment so long as backspin is not present [20]. Although the pogo stick mounted to our UAS includes a spring, we assume a rigid pogo stick in the model used for collision response, controller design, and simulation. To compute collision response, we adopt the linear complementary problem solver outlined in [21].

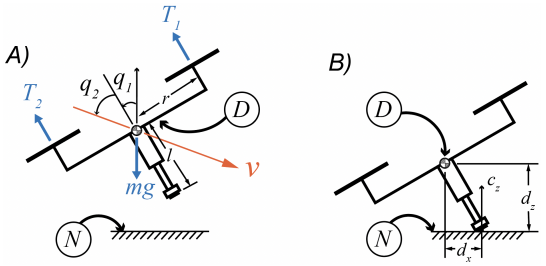

Fig. 2. A) Free Body and Configuration Diagram of 2-D UAS in Flight. B) Dimensions of Collision Contact from UAS Center of Mass and Ground Contact Normal

At the instant of collision, it is necessary to establish the point at which contact is made and its relationship to the UAS center of mass. This mapping is visualized in Fig. 2B, where d_x and d_z are the horizontal and vertical distances from the UAS center of mass to the point of contact, respectively, and c_x (not shown as it is zero on flat horizontal ground) and c_z are the horizontal and vertical contact normal components of the contact surface, respectively. With this information, a contact Jacobian J_c consisting of tangential J_t and normal J_n components can be constructed according to the following formulation:

$$J_c = \begin{bmatrix} J_t & J_n \end{bmatrix} = \begin{bmatrix} c_z & c_x \\ -c_x & c_z \\ -d_x c_x - d_z c_z & d_x c_z - d_z c_x \end{bmatrix} \tag{7}$$

At the instant of contact, the total forces acting on the UAS are

$$F_e + J_c F_c = Ma, \tag{8}$$

where $a = [\ddot{p}_x, \ddot{p}_z, \ddot{q}_1]^T$, M is the UAS inertia matrix, F_e are the external forces in the world frame and F_c is the contact force in the contact surface frame. The contact force is broken into normal and tangential components that are found in the z column matrix after the formulation of the linear complementary problem is solved as follows:

$$Vz + p \geq [0] \tag{9}$$

$$z \geq [0], \tag{10}$$

where the matrices V and p are defined as outlined in [21].

III. Hybrid System Controller Design

A. Velocity Matching

Velocity Matching is our method of foot placement. The appeal of this Velocity Matching mode is that it can run blind of altitude and position, only running with attitude control as shown in the block diagram in Fig. 4. By aligning the axis of the pogo stick with the velocity vector of the UAS, Velocity Matching ensures that the tip of the pogo stick is the first point of contact the system has with the environment. This allows for easier recovery, as the propellers are protected from striking the ground during a collision. Velocity Matching also minimizes the moments of any contact impulses about the UAS center of mass. The impulse that comes from the collision even acts as a restoring force, driving the UAS back away from the ground. The angle of the realtime velocity vector $q_3 = q_1 + q_2$ is used as the setpoint for the attitude controller such that the axis of the UAS pogo stick aligns with the velocity of the UAS center of mass. In effect, q_2 is driven to zero (see Fig. 2A). The value of q_2 is calculated from the velocity of the UAS center of mass. During Velocity Matching mode, the maximum pitch and roll of the UAS was set to 45° and the maximum angular velocity was set to 4 rad/s to prevent UAS tip-over and propeller strike.

For the 2-D simulation case, velocities are given in the world frame and pitch setpoint q_3 is calculated as follows:

$$q_3 = q_1 = \frac{\pi}{2} - \arctan\frac{\dot{p}_z}{\dot{p}_x} \tag{11}$$

For the 3-D experimental case, the additional roll setpoint $\phi_{desired}$ is calculated as follows:

$$\phi_{desired} = \arctan\frac{\dot{p}_z}{\dot{p}_y} - \frac{\pi}{2} \tag{12}$$

B. Hybrid System Integration

Our UAS operates in one of two distinct flight control modes: Conventional Waypoint Following and Velocity Matching Control. From the Conventional Waypoint Control mode, there are three possible transitions: Approach Initiation, Envelope Violation, and Recovery. The mode switching is described in a Hybrid Automaton Diagram in Fig. 5.

When in Conventional Waypoint Following, time-dependent 3-D waypoints w_{conv} are used for guidance and navigation and are generally defined with 3-D position and yaw setpoints associated with specific times, as outlined in [22].

$$w_{conv} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ \psi \end{bmatrix} \tag{13}$$

To begin, a manual trigger sends the UAS into the Approach Initiation transition with a special waypoint w_{ai}, which sets a predetermined target velocity $\dot{p}_{x|t}$. The UAS uses feedback rate control to achieve the desired $\dot{x}$ while maintaining $\dot{p_y} = \dot{p_z} = 0$.

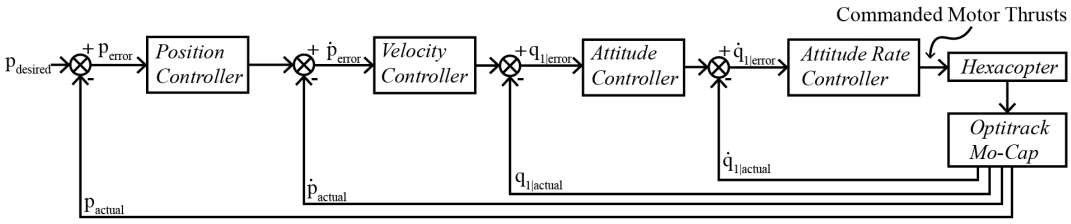

Fig. 3. Cascaded control loop for the UAS flight controller managing all flight control tasks other than Velocity Matching.

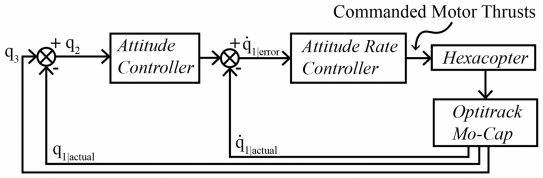

Fig. 4. Inner loop attitude controller that is solely used in the Velocity Matching flight mode.

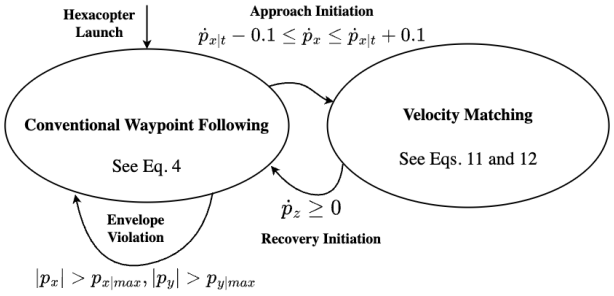

Fig. 5. Hybrid control model for UAS using physical parameters as guard conditions

If at any point the $\dot{p}_x$ is within $\dot{p}_{x|t} \pm 0.1$m/s, then the Approach Initiation transition is triggered and the UAS is in Velocity Matching Mode, see (Fig. 5)

Waypoints w_{ai} in the Approach Initiation transition are time independent, only focusing on hitting a target horizontal velocity $\dot{p}_{x|t}$ and maintaining a target altitude $p_{z|t}$, as seen in the following formulation:

$$w_{ai} = \begin{bmatrix} p_y = 0 \\ p_z = p_{z|t} \\ \dot{p}_x = \dot{p}_{x|t} \\ \dot{p}_y = 0 \\ \dot{p}_z = 0 \\ \psi = 0 \end{bmatrix} \tag{14}$$

All other waypoints in Conventional Waypoint Following mode use the normal standard of time-based position and yaw waypoints.

In Velocity Matching mode, a collision is determined to be complete if the controller detects $\dot{p_z} = 0$ (after a period of $\dot{p_z} < 0$), at which point the UAS goes through a recovery transition to the Conventional Waypoint Following mode. If at any point while in the Velocity Matching mode the controller detects a positive $\dot{p_z}$, the controller saves the system with a Recovery transition.

The Recovery transition is a simple routine to achieve straight and level flight with $\dot{\vec{p}} = \vec{0}$. This transition is triggered on the detection of a successful collision or unsafe approach.

The Envelope Violation transition can occur during any portion of the Conventional Waypoint Following, and is triggered when the UAS leaves a predefined safety barrier on p_x and p_y. When this transition is triggered, the UAS is commanded to safely hover in the center of the testing area.

The inner loop for the UAS in all flight modes is the same combination of a proportional controller for attitude and a proportional-integral-derivative controller for attitude rate, see Fig. 4. The gains for these two controllers were determined empirically through flights with the UAS and attached pogo stick.

In the Conventional Waypoint Following mode, the additional use of the outer loop of the cascaded control presented in Fig. 3 maintains altitude (for use with the Approach Initiation transition), and 3-D position (for Recovery, Envelope Violations, and normal flight).

C. Thrust Limitation in Velocity Matching Mode

When the UAS is in Velocity Matching mode, the total thrust of the system is set to an experimentally determined total thrust of 9.4 N. This limit allows for minimal lift while maintaining a degree of authority for attitude control. Minimizing lift is important to having a forceful collision, so that the pogo stick is compressed and the UAS is launched upwards after contact. For this design, we assume that the system has much more influence over its pitch q_1 than its position p_x and p_z when thrust is limited.

By linearizing the 2-D flight dynamics outlined in Section II-A, a relationship between T_{dif}, T_{tot}, q_1, $\ddot{q}_1$, $\ddot{p}_x$, and $\ddot{p}_z$ can be found. To linearize the equations governing UAS flight, we define state vector x and input vector u to express the state derivative $\dot{x}$ as a linear combination of each state in x and each input in u, as derived from the nonlinear system presented earlier in Eq. 1, with the disturbance force $F_d = 0$.

$$x = x_o + \delta x = \begin{bmatrix} q_1 \\ p_x \\ p_z \\ \dot{q}_1 \\ \dot{p}_x \\ \dot{p}_z \end{bmatrix}, \; u = u_o + \delta u = \begin{bmatrix} T_{dif} \\ T_{tot} \\ mg \end{bmatrix} \tag{15}$$

$$\dot{x} = f(x,u) = \begin{bmatrix} \dot{q_1} \\ \dot{p_x} \\ \dot{p_z} \\ \frac{rT_{dif}}{J} \\ \frac{T_{tot}\sin q_1}{m} \\ \frac{-T_{tot}\cos q_1 + mg}{m} \end{bmatrix} \tag{16}$$

We can relate local perturbation states x their derivatives $\dot{x}$ and inputs u as

$$\delta\dot{x} = A\delta x + B\delta u. \tag{17}$$

By linearizing the system with $T_{tot|0} = 9.4\,\text{N}$ and $q_{1|0} = 45°$, where $r = 0.275$ m, $m = 1.687$ kg, and $J = 0.00915$ kg m^2 [23], we can derive the following relations:

$$\begin{aligned} \delta\dot{q_1} &= \frac{r}{J}\delta T_{dif} \\ &= 30.055\, T_{dif} \end{aligned} \tag{18}$$

$$\begin{aligned} \delta\dot{p}_x &= \frac{T_{tot|0}\cos q_{1|0}}{m}\delta q_1 + \frac{\sin q_{1|0}}{m}\delta T_{tot} \\ &= 2.93\,\delta q_1 + 0.311\,\delta T_{tot} \end{aligned} \tag{19}$$

$$\begin{aligned} \delta\dot{p}_z &= \frac{T_{tot|0}\sin q_{1|0}}{m}\delta q_1 - \frac{\cos q_{1|0}}{m}\delta T_{tot} \\ &= 2.93\,\delta q_1 - 0.311\,\delta T_{tot} \end{aligned} \tag{20}$$

The coefficents in Eqs. 18, 19, and 20 govern the relationships between $\delta\dot{q_1}$, δT_{dif}, δq_1, δt_{tot}, $\delta\dot{p_x}$, and $\delta\dot{p_z}$. We see that the coefficient on δT_{dif} in Eq. 18 is two orders of magnitude greater than the coefficients on δT_{tot} in Eqs. 19 and 20. This means that the UAS should have significant authority over pitch q_1 while having lower authority over the change of the velocity vector when pitching with lower limited T_{tot}.

This lower influence over velocity change additionally makes the velocity matching easier to track. If the influence of pitch q_1 over velocity was highly coupled, velocity matched pitches would cause drastic changes in the direction of velocity vector meaning that the already moving setpoint q_3 more difficult to track. The lesser coupling outlined (see Eqs. 18, 19, and 20), means that changes in velocity and thus q_3 will be slower.

IV. Flight Testing Methods

We used a DJI F550 Hexacopter frame as the platform for testing collision-based aerial manipulation (see Fig. 6). The hexacopter used six 935kV brushless motors controlled at 48MHz each equipped with a 32-bit electronic speed controller and paired with a 1045 plastic propeller. We used a single 3000mAh 4 cell LiPo battery to power the UAS.

We fabricated the 22.86 cm long, two-part pogo stick (see Fig. 6) with SLA printed resin in the form of a cylinder frame with a rubber tip (for increased friction on contact). The springs in the pogo stick could be replaced to change the collision response of the UAS. We selected a 2.68 kgf/cm spring, stored in the upper housing of the pogo stick. The total system weight was 1.687 kg. The coefficient of static friction μ between the floor of the flight space and pogo stick was experimentally determined to be 0.42 using a specialized apparatus documented in [24]. The effective coefficient of restitution for the system with the 2.68 kgf/cm spring was experimentally determined to be 0.744 (used in collision response simulations).

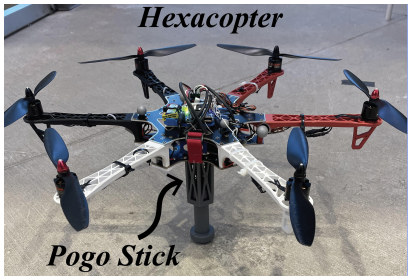

Fig. 6. Modified DJI F550 Hexacopter (550 mm wheelbase) that is acting as the UAS for boucing interactions. The attached pogo stick was used to facilitate collision.

We used an open source modified version of Ardupilot (`rc-pilot-a2sys`) to implement the flight control system described above. The flight control software ran on a BeagleBone Blue. Position and velocity information was streamed from an Optitrack system to a BeagleBone ground control station, while pitch, roll, and yaw states plus angular rates were collected from an onboard IMU.

We developed a 2-D simulation software in Python to compute rigid body collision response profiles between the UAS and a flat ground. This simulation models the complete system seen in Fig. 2. The time step for the simulation was 0.005s, the same as control loop time step on the flight controller. Simulation dynamics were computed via Euler integration.

V. Results

A. Velocity Matching Performance

Fig. 7A shows the overall trajectories of the UAS from approach through recovery. This also shows that collision is not instantaneous, as assumed in simulation. We have identified a region in time (contact) where the pogo stick spring is engaged and applying a restoring force. After the contact period, the Velocity Matching pitch setpoint, $q_{1|setpoint}$, is set to 0°, demarcated by the asterisk on Fig. 7A. Fig. 7B shows that the pitch tracking itself is performing poorly with a roughly 10° error at collision for this flight test ($p_{z|initial}$ = 1.5 m, $\dot{p}_{x|initial}$ = 1.8 m/s). This large error still allowed for a recoverable collision, and slower speeds at the same altitude show less error at collision.

B. Flight Envelopes for Collision Approach

A typical flight envelope is a constraint set based on aerial vehicle properties (ie. velocities, thrusts, control surfaces, and structures) that allows the vehicle to stay within its operational limits (ie. avoid structural failure or stalling) [25]. For aerial manipulation with collisions, we define our flight envelope as the initial altitudes and horizontal velocities that will allow for recovery from intentional collisions. A

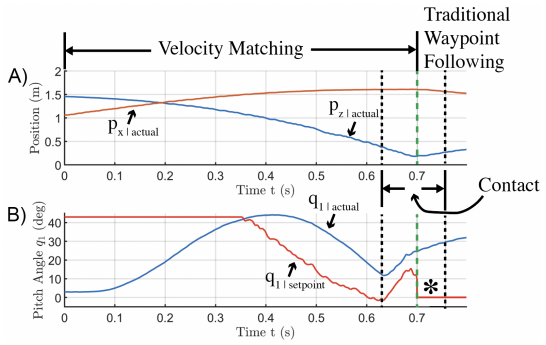

Fig. 7. A) Visualization of UAS collision with $p_{z|initial} = 1.5$m, $\dot{p}_{x|initial} = 1.8$m/s. B) Performance visualization of pitch in Velocity Matching controller over the same time period. Asterisk demarcates where the Velocity Matching controller is toggled off and a zero pitch setpoint is being commanded after a bounce has occured.

recoverable collision is defined as a collision in which a hover can be reached post-collision without propellers striking the ground.

In order to identify the *simulation* flight envelope 250,000 simulations were run at varying horizontal speeds and altitudes. The combined grey and yellow region in Fig. 8 indicates those initial altitudes $p_{z|initial}$ and horizontal speeds $\dot{p}_{x|initial}$ that are recoverable, describing the *simulation* flight envelope while the orange region is outside the *simulation* flight envelope.

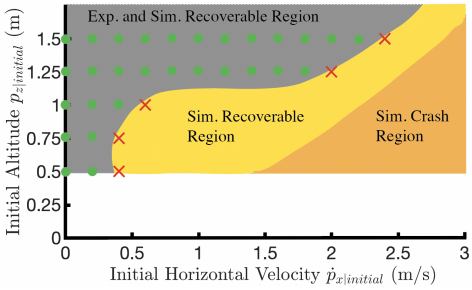

Fig. 8. Two-dimensional flight envelope mapping initial altitudes $p_{z|initial}$ and horizontal velocities $\dot{p}_{x|initial}$ to successful collisions. The area shaded in yellow and grey represents successful collisions, while the area shaded in orange are failed collisions in simulation. The green dots and red x's represent successful and unsuccessful experimental flights that each included a collision, which form the grey shaded area. The unshaded region represents altitudes that were too low to obtain consistent collision data, due to the length of the pogo stick.

Additionally, 35 experimental collision flights were conducted (in increments of 0.25 m $p_{z|initial}$ and 0.2 m/s $\dot{p}_{x|initial}$) to identify the *experimental* flight envelope. These test results can be seen in Fig. 8, with the green dots representing successful collisions and the red x's representing crashes. The experimental flight data was used to create the grey shaded experimental recoverable region that separates successful collisions and crashes. No tests were conducted past the first horizontal speeds that resulted in a crash at a given altitude.

VI. Discussion and Conclusions

We have presented and compared 2-D models for a UAS during flight and during perch to assess the impact of the constraints imposed by perch on disturbance rejection control properties. We argue that a UAS can interact with its environment without sacrificing disturbance rejection when limiting overall interaction time. By using collisions as our method of choice for environmental interaction, we limit the total interaction time to be near instantaneous and turn collision impulses into forces that move the UAS away from the ground after collision is completed. Aerial manipulation with collisions was made possible through the development of a Velocity Matching controller, which simplifies approach control to an attitude and velocity matching controller.

A 2-D, rigid body simulator was developed both for control design and validation but also for uncovering a flight envelope for the Velocity Matching flight mode. This *simulation* flight envelope covers the *experimental* flight envelope (see Fig. 8) showing that it is a predictor of safe Velocity Matching conditions for the UAS. Experiments were run in the longitudinal plane, but the Velocity Matching controller had to be 3-D to reject lateral disturbances. Discrepancies in the flight envelopes can be explained partially by the difference between rigid body and pogo stick contact with the ground (see Fig. 7A) as well as other unknowns in the physical system that were not modelled. The pogo stick in simulation has no compression and the propellers provided instant torque while this is untrue in the physical experiment. The poor performance of the velocity matching controllers at low $p_{z|initial}$ with higher horizontal speeds may also be influenced by latency in the experimental setup, as motion capture data must be initially recorded, then streamed via USB to the ground station, and finally streamed via XBee radio to the UAS during free-fall before velocity matching can run and motors can begin being commanded. This latency was not modelled in simulation, but was found to have a mean of 0.0084 s with a standard deviation of 0.0011 s during experiments. Even for low latency communications, the shortest falling time during Velocity Matching (average of 0.32 s for altitude of 0.5 m) exacerbates the problem. Nonetheless, the existence of these flight envelopes is essential for extending this work further into the domain of aerial manipulation, where fully understanding when and where the UAS can safely collide is paramount (i.e. during tree leaf sampling).

The goal of this work was to lay the foundation for solving autonomous rainforest tree sampling. We believe that aerial manipulation with collisions can be extended to any manipulation task that can completed in a single pass with a short interaction window. Follow-on work for this project includes extending the current foot placement strategy to allow for bouncing on a specified target in 3-D and developing a pogo stick manipulator that can cut and grasp during collision to further pursue the purpose of rainforest tree sampling.

REFERENCES

[1] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, "Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use," *Remote Sensing*, vol. 4, no. 6, pp. 1671–1692, 2012.

[2] E. Beachly, C. Detweiler, S. Elbaum, D. Twidwell, and B. Duncan, "Uas-rx interface for mission planning, fire tracking, fire ignition, and real-time updating," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, 2017, pp. 67–74.

[3] Y. Sun, A. Plowcha, M. Nail, S. Elbaum, B. Terry, and C. Detweiler, "Unmanned aerial auger for underground sensor installation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1374–1381.

[4] M. M. Romano, Y. Chen, P. Kuevor, O. Marshall, and E. Atkins, "Nailed it: Autonomous roofing with a nailgun-equipped octocopter," in *AIAA AVIATION 2021 FORUM*, 2021, p. 3211.

[5] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.

[6] S. P. Netto, A. L. Pelissari, V. C. Cysneiros, M. Bonazza, and C. R. Sanquetta, "Sampling procedures for inventory of commercial volume tree species in amazon forest," *Anais da Academia Brasileira de Ciências*, vol. 89, pp. 1829–1840, 2017.

[7] M. G. Barker and M. A. Pinard, "Forest canopy research: sampling problems, and some solutions," in *Tropical forest canopies: ecology and management*. Springer, 2001, pp. 23–38.

[8] G. Charron, T. Robichaud-Courteau, H. La Vigne, S. Weintraub, A. Hill, D. Justice, N. Bélanger, and A. Lussier Desbiens, "The deleaves: a uav device for efficient tree canopy sampling," *Journal of Unmanned Vehicle Systems*, vol. 8, no. 3, pp. 245–264, 2020.

[9] S. Krisanski, M. S. Taskhiri, J. Montgomery, and P. Turner, "Design and testing of a novel unoccupied aircraft system for the collection of forest canopy samples," *Forests*, vol. 13, no. 2, p. 153, 2022.

[10] D. R. McArthur, A. B. Chowdhury, and D. J. Cappelleri, "Autonomous control of the interacting-boomcopter uav for remote sensor mounting," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5219–5224.

[11] H. B. Khamseh, F. Janabi-Sharifi, and A. Abdessameud, "Aerial manipulation—a literature survey," *Robotics and Autonomous Systems*, vol. 107, pp. 221–235, 2018.

[12] D. Lee, H. Seo, D. Kim, and H. J. Kim, "Aerial manipulation using model predictive control for opening a hinged door," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1237–1242.

[13] J. Thomas, J. Polin, K. Sreenath, and V. Kumar, "Avian-inspired grasping for quadrotor micro uavs," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 55935. American Society of Mechanical Engineers, 2013, p. V06AT07A014.

[14] A. E. Gomez-Tamm, V. Perez-Sanchez, B. C. Arrue, and A. Ollero, "Sma actuated low-weight bio-inspired claws for grasping and perching using flapping wing aerial systems," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8807–8814.

[15] P. Ramon-Soria, A. E. Gomez-Tamm, F. J. Garcia-Rubiales, B. C. Arrue, and A. Ollero, "Autonomous landing on pipes using soft gripper for inspection and maintenance in outdoor environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5832–5839.

[16] M. H. Raibert, H. B. Brown Jr, M. Chepponis, J. Koechling, and J. K. Hodgins, "Dynamically stable legged locomotion," Massachusetts Inst of Tech Cambridge Artificial Intelligence Lab, Tech. Rep., 1989.

[17] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4559–4566.

[18] B. Zhu, J. Xu, A. Charway, and D. Saldaña, "Pogodrone: Design, model, and control of a jumping quadrotor," *arXiv preprint arXiv:2204.00207*, 2022.

[19] V. Artale, C. L. R. Milazzo, C. Orlando, and A. Ricciardello, "Particle swarm optimization applied to hexarotor flight dynamics," in *Progress in Industrial Mathematics at ECMI 2014*, G. Russo, V. Capasso, G. Nicosia, and V. Romano, Eds. Cham: Springer International Publishing, 2016, pp. 625–632.

[20] N. Fazeli, S. Zapolsky, E. Drumwright, and A. Rodriguez, "Fundamental limitations in performance and interpretability of common planar rigid-body contact models," in *Robotics Research*. Springer, 2020, pp. 555–571.

[21] D. E. Stewart, "Rigid-body dynamics with friction and impact," *SIAM review*, vol. 42, no. 1, pp. 3–39, 2000.

[22] G. Farid, H. T. Hamid, S. Karim, and S. Tahir, "Waypoint-based generation of guided and optimal trajectories for autonomous tracking using a quadrotor uav," *Studies in Informatics and Control*, vol. 27, no. 2, pp. 225–236, 2018.

[23] N. P. Nguyen, N. Xuan Mung, and S. K. Hong, "Actuator fault detection and fault-tolerant control for hexacopter," *Sensors*, vol. 19, no. 21, p. 4721, 2019.

[24] P. Ewen, A. Li, Y. Chen, S. Hong, and R. Vasudevan, "These maps are made for walking: Real-time terrain property estimation for mobile robots," *IEEE Robotics and Automation Letters*, 2022.

[25] N. H. McClamroch, *Steady aircraft flight and performance*. Princeton University Press, 2011.