Towards Aerial Recovery of Parachute-Deployed Payloads

Ajay Shankar, Sebastian Elbaum, Carrick Detweiler[1]

***Abstract*— Sensor payloads suspended from parachutes are often used in atmospheric profiling applications. They drift freely and often end up landing in inaccessible regions that make their retrieval challenging or impossible. In this paper, we develop and evaluate an approach using a multirotor unmanned aerial system to autonomously retrieve the parachute while it is still in the air. The system relies only on the initial conditions of the parachute-payload system and feedback from the vehicle's onboard cameras to track and then intercept the parachute mid-air in under 40 seconds on average. We present the results from our field experiments where we demonstrate the feasibility of the system and discuss its applicability to long-term payload transportation systems.**

I. INTRODUCTION

Recovering parachute-suspended sensor payloads using unmanned aerial systems (UAS) has a range of potential applications. Sensors suspended by parachutes, also known as dropsondes [1], are often used for atmospheric profiling, yet they are rarely recovered since they often land in inaccessible locations. In practice, this limits the sensor payloads to inexpensive sensors such as temperature and humidity. In-air interception of payloads is also a step towards teams of aerial vehicles exchanging packages either over terrains where it may be impossible to land, or to create a longer chain of package transportation. However, the problem is challenging because of the stochasticity in the motion of the parachute-payload system and the fact that the UAS typically has under 60 seconds to intercept the payload during low altitude drops.

In this paper, we take the first steps towards developing a system for quickly and autonomously tracking and then capturing parachutes in-air using a multi-rotor UAS. Parachute-suspended payloads often have loosely modeled dynamics, which makes their tracking and estimation difficult. The overall design of the system must therefore account for the physical capabilities of the UAS and the limits on an end-effector that traps the parachute. Additionally, interception methods need to have greater considerations for the safety of the vehicle when the target is in close proximity, and these are aggravated by the erratic motion of the parachute.

Our proposed approach formulates the UAS mission first as a target identification and following problem, and then as an aerial interception problem. The design is generic, and does not require a special type of parachute or UAS, nor additional sensors and communication with the payload. The UAS is only augmented with an onboard computer, two color cameras and a passive hook to trap the parachute (Figure 1).

*This work was supported in part by NSF IIA-1539070, and USDA-NIFA 2017-67021-25924.

[1]Department of Computer Science and Engineering, University of Nebraska-Lincoln, Nebraska, USA. {ashankar, elbaum, carrick}@cse.unl.edu

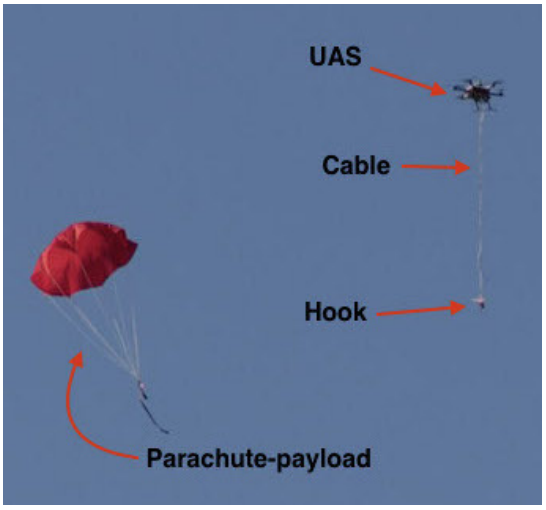

Fig. 1: A multirotor UAS chasing the parachute through its descent.

Tracking objects using aerial vehicles has been dealt with under full, partial and noisy state information about the target [2], [3], [4]. The novelty of our approach lies in the interplay between tracking a loosely modeled target and intercepting it swiftly. Our approach tackles this interplay end-to-end, while receiving no state measurements directly from the target, which makes the state estimation purely extrinsic. We impose reasonable constraints on the choice of variables such as the terminal velocity of the parachute and the maximum speed of the vehicle to demonstrate the feasibility of our approach. The main contributions of this work are:

1) An analysis of the physical requirements of a system designed to intercept a target with few constraints on the dynamics of the parachute,
2) An approach and implementation designed to intercept parachutes in-air in spite of noisy state information,
3) Preliminary field experiments that demonstrate that the UAS can track and intercept a parachute in under $40\,\mathrm{s}$ on average,
4) A detailed analysis of 10 trials, examining the conditions under which the system succeeds or fails in the first interception attempt.

II. CHALLENGE

We have argued that the aerial recovery of parachutes has enormous potential in a range of fields from atmospheric profiling to package handling. Parachutes, however, are generally hard to model with a high level of accuracy. Figure 2 shows an example illustrating this difficulty through a 24 second sequence of images captured from the downward-facing camera of a hovering UAS. The images show the path

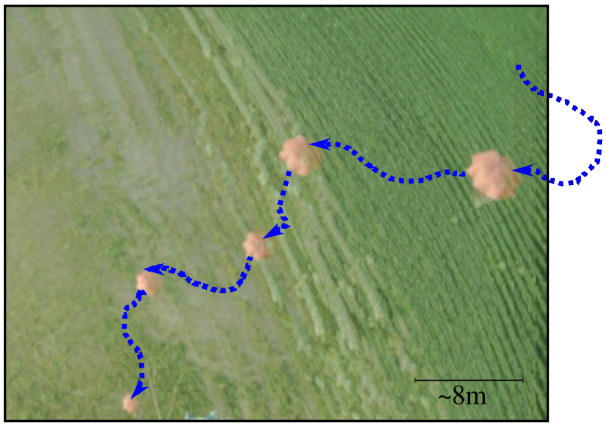

Fig. 2: A top-view blended image of the motion of the parachute over a span of 24 seconds (60 images with the parachute shown at 5 points for size reference). As marked by the dotted lines, the parachute leaves the field of view from the right edge of the frame, and then reenters soon after.

taken by a red-colored parachute canopy, 1.2m across, that starts at the top-right segment of the image, drifts outside the field of view of the camera, and then reenters by moving in the opposite direction. The images capture a fundamental issue with parachutes – they can undergo rapid changes in their trajectory in descent.

An additional difficulty is that, unlike a generic moving target localization/landing problem, in the case of parachutes, the UAS has a finite amount of time to position itself correctly (approximately 60 seconds in our experiments before it hits the ground). Additional challenges include the fact that parachutes periodically exhibit turbulent phases mixed with smooth descent phases, and a UAS hovering close to the canopy will adversely affect the descent of the parachute.

Forces on the parachute: Although the parachute's vertical and lateral velocities can be approximated, they are subject to external disturbances that introduce significant variability. Consider a payload with mass m_p, which accelerates towards the ground with an acceleration, g, due to gravity (Figure 3). As the parachute inflates due to the airflow, its canopy generates aerodynamic drag in the opposite direction, which is proportional to its surface area, A_{par}. The drag increases with the velocity of descent, and the parachute-payload system eventually attains a terminal velocity, v_p [5]. As reflected by the second term in equation (1) below, however, the descent dynamics of the parachute are affected by several external disturbances including the swinging motion of the payload ($swing$), wind gusts that change the canopy surface area thereby changing the angle of attack (AoA), the porosity of the canopy material (por), unmodelled dynamics arising from manufacturing defects and imperfections. The factors in $f_{par}()$ are particularly difficult to model, can be highly non-linear, and combined lead to erratic motions of the parachute. The terminal velocity is therefore given by,

$$v_p = \sqrt{\frac{2m_p g}{C_d \rho A_{par}}} + f_{par}(swing, A_{par}, AoA, por, dyn, ...), \quad (1)$$

where, C_d is the drag coefficient and ρ is the density of air, typically 1.75 and $1.229\mathrm{kg/m}^3$, respectively.

Similarly, the lateral dynamics of the parachute system, induced by the horizontal component of the drag, F_H, are governed primarily by the direction of wind, but still susceptible to variations in the descent. Wind conditions can change drastically during the descent and are very difficult to predict. Because of these factors, state estimation methods, such as Extended Kalman and particle filters, cannot effectively model or predict the motion of a parachute in real-world conditions well enough to facilitate an interception.

Solution Space: The space of solutions to perform such a recovery is defined by the attributes of three elements: the parachute, UAS, and payload. Our goal is to identify a solution that imposes minimal constraints on these elements, while being able to intercept parachutes. Hence, we require that the approach will:

- Work with a generic parachute consisting of a canopy of symmetric shape with shroud lines that extend down to the payload, as shown in Figure 3(left).
- Require no additional sensors incorporated into the parachute since they may add to the payload, cost, and deployment complexity.
- Utilize an off-the-shelf UAS that is agile enough (in terms of acceleration) to compensate for the stochastic nature of the parachute trajectory in all axes.
- Not rely on parachutes dropped high up in the atmosphere where there would be significant time to reestimate the dynamics.
- Maintain reliable margins of safety for the vehicle during the mission.

Our reliance on a purely extrinsic state estimation, combined with uncertainty in the motion of the parachute, makes the development of a generic approach to parachute recovery extremely challenging. In the next sections we describe the most relevant efforts related to this line of work, and then present an approach for parachute recovery that meets these constraints, and can be parameterized to work with a range of configurations of parachutes, UASs, and payloads.

III. Related Work

The literature most closely related to ours include visual tracking and following, and aerial grasping. In general, target identification in cluttered backgrounds tends to be a difficult stage in the tracking problem, especially over natural landscapes. In our implementation, we only require that the parachute be "trackable" using a monocular camera. This could be realized using several methods, for instance, using visual tags [6], [7], color-based segmentation [3], or searching for known patterns [8]. Tag-based approaches offer the advantage of obtaining a more accurate pose of the object, however, parachute packaging and its fluttering may easily deform the tag, adversely affecting the detection quality.

A variety of methods have been studied for tracking visual features when both the target and the camera are in motion [9]. Tracking targets from aerial vehicles has also been

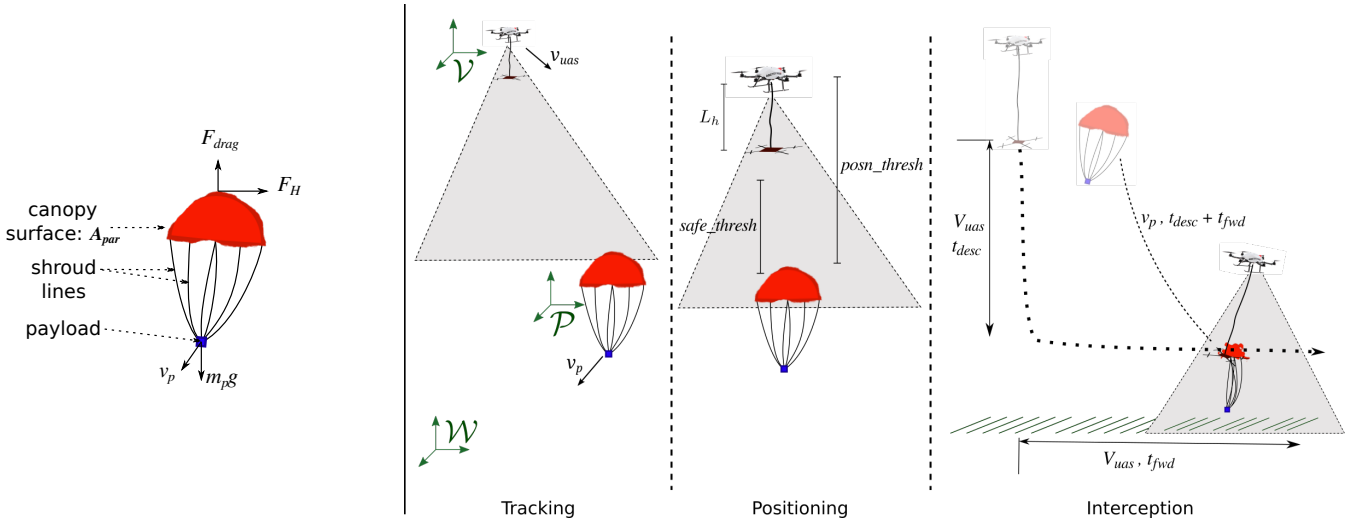

Fig. 3: A graphical representation of the parachute and its components (left) and the three phases of the proposed method (right).

studied for both fixed wing UASs [2] and for multirotors [3]. Similar to the approach presented by Teuliere *et al.* [3], in which a hierarchical P+PI control is used for position and attitude stabilization, our implementation uses a PD controller that takes the relative position of the target as its input and generates control commands for the vehicle.

Attempting to bridge the distance to a parachute in descent is closely related to vision-based algorithms for landing on a moving target, which was demonstrated in previous work by Saripalli [8], and more recently by Borowczyk [10]. The approaches demonstrated for landing typically work with a steady-moving target and a reliable source of information so that the motion can be modeled. Aerial vehicles have been used to autonomously grasp static objects using visual cues [11], [12]. Our problem of chasing and intercepting a parachute in descent does not afford these assumptions. A simulation study [13] presented results for aerial vehicles grasping moving targets by assuming full state knowledge of the target, and without any uncertainty in the estimates. Again, following the discussion in the previous section, we do not assume that the parachute's states are fully available. We draw significant inspiration for this work from the manned recovery of parachutes performed by trained fixed-wing pilots in the 1960s [14].

IV. Approach to Parachute Recovery

Understanding that a UAS must employ different strategies depending upon where it is relative to the parachute, we divide our approach into three phases: tracking, positioning and interception. These are shown in Figure 3.

We use the following notation: the world fixed frame is denoted by $\mathcal{W}$, and the respective origins of the vehicle and the parachute frames are at $\mathcal{V}$ and $\mathcal{P}$ (measured at their respective centers of mass). The vehicle is augmented with a passive end-effector in the form of an 'x'-shaped hook, which has its own frame $\mathcal{H}$ (not shown in the figure) offset from the vehicle frame by a distance L_h. This offset is the length of the cable that connects the hook to the vehicle, and it is necessary for the safety of the operation. The notation we use is such that at some given time t, $P_b^A(t)$ denotes the position of the object b measured in frame A, and $D_{A,C}(t)$ denotes the absolute distance between the origins of two frames A and C. Algorithm 1 lists the pseudo-code that governs the vehicle's software architecture, which we'll describe throughout the three phases.

A. Tracking

The mission begins by tracking the parachute in its descent. The algorithm loops (line 3) until one of the criteria for the end of the mission is met (described later). We assume that we know the initial point of the parachute release in the world frame, $P_{\mathcal{P}}^{\mathcal{W}}(0)$. The origin of the vehicle frame, $\mathcal{V}$ at that instant, $P_{\mathcal{V}}^{\mathcal{W}}(0)$, must be such that, $D_{\mathcal{V},\mathcal{P}}(0) \equiv \|P_{\mathcal{V}}^{\mathcal{W}}(0) - P_{\mathcal{P}}^{\mathcal{W}}(0)\| < B$, where $\|\cdot\|$ denotes Euclidean distance. We can select a value of B based on the maximum speeds of the parachute and the UAS to ensure that they are sufficiently close together at the start of the mission.

In our implementation, we use a parachute with a red colored canopy to ease the detection stage. A color-segmentation method separates out the parachute from the background. The detection algorithm, in line 4, uses only monocular vision estimates to generate commands for the vehicle by computing

$$D_{\mathcal{V},\mathcal{P}}(t) = f_{cam}\left(i_p(t), j_p(t), c_p(t)\right), \tag{2}$$

where f_{cam} is a function of the camera calibration, $i_p(t), j_p(t)$ specify the location of the parachute in the image plane, and $c_p(t)$ denotes the contour of the segmented region. A proportional-derivative (PD) controller uses the information from (2) to compute roll (ϕ), pitch (θ) and the vertical velocity ($V_{z,uas}$) commands (line 9) for the UAS as,

$$\begin{aligned}\{\phi(t), \theta(t)\} &= f_{K_p}(i_p(t), j_p(t), D_{\mathcal{V},\mathcal{P}}(t)) + f_{K_d}(\Delta i_p, \Delta j_p),\\ V_{z,uas}(t) &= f_{vel}(D_{\mathcal{V},\mathcal{P}}(t))\end{aligned} \tag{3}$$

for an unconstrained minimization of $D_{\mathcal{V},\mathcal{P}}$, by keeping the parachute center close to the camera center. Since we do not rely on a complete model for the parachute, the

Algorithm 1 Pseudo-code for state control

```
1:  mission_end ← false
2:  phase ← TRACKING
3:  while ¬mission_end do
4:      (i_p, j_p, c_p) ← PROCESS_CAMERASTREAM()
5:      D_{V,P} ← f_cam(i_p, j_p, c_p)                          ▷ Eqn. (2)
6:      ψ ← RC(yaw)
7:      switch phase do
8:          case TRACKING
9:              [φ, θ, V_z] ← PD(i_p, j_p, D_{V,P}, f_{K_p}, f_{K_d}, f_vel)
10:                                                             ▷ Eqn. (3)
11:             if D_{V,P} ≤ posn_thresh then
12:                 phase ← POSITIONING
13:                 timer ← 0
14:         case POSITIONING
15:             [φ, θ, V_z] ← PD(i_p, j_p, D_{V,P}, f'_{K_p}, f'_{K_d}, f'_vel)
16:                                                             ▷ Eqn. (3)
17:             INCREMENT(timer)
18:             if timer ≥ posn_time then
19:                 phase ← INTERCEPTION
20:                 t_init ← t
21:             if D_{V,P} < safe_thresh then
22:                 phase ← UNSAFE_POSE
23:             if D_{V,P} > posn_thresh then
24:                 phase ← TRACKING
25:         case INTERCEPTION
26:             [φ, θ, V_z, stop] ← MANEUVER-CONTROL(t)
27:             if stop = true then
28:                 mission_end ← true
29:         case UNSAFE_POSE
30:             mission_end ← true
31:     UAS-CONTROL(φ, θ, ψ, V_z)
32: function MANEUVER-CONTROL(t)
33:     [φ, θ, V_z, stop] ← [0, 0, 0, false]
34:     if t ≤ t_init + t_desc then
35:         θ ← 0%
36:         V_z ≈ V_uas                          ▷ Descend for t_desc sec
37:     else if t ≤ t_init + t_desc + t_fwd then
38:         θ ← 100%
39:         V_z ≈ 0%                             ▷ Pitch forward for t_fwd sec
40:     else
41:         stop ← true
    return [φ, θ, V_z, stop]
```

controller gains in the functions f_{K_p}, f_{K_d} and f_{vel} are obtained empirically. This is done by repeating different experiments with the UAS chasing a parachute placed on top of a ground vehicle moving at varying speeds.

B. Positioning

This phase (lines 14 - 24) positions the UAS with respect to the parachute in a manner that is suitable for interception. The system transitions to this stage when the estimated distance between the vehicle and the parachute ($D_{\mathcal{V},\mathcal{P}}$) is less than some predetermined threshold, $posn\_thresh$ (line 11). A value of $posn\_thresh$ is determined from our preliminary tests such that the effect of the propeller downwash on the canopy is minimal (compared to natural factors such as wind). Since the vehicle is positioned closer to the parachute, the controller functions from Equation (3) become f'_{K_p}, f'_{K_d} and f'_{vel} such that they have smaller gains so as to prevent aggressive corrections. The control commands now lead to the minimization of $|D_{\mathcal{V},\mathcal{P}}(t) - posn\_thresh|$, subject to the constraint that $D_{\mathcal{V},\mathcal{P}}(t) > safe\_thresh$. The parameter $safe\_thresh$ is a distance threshold which ensures that the vehicle does not approach th canopy too close.

The controller tries to position the vehicle within this range for a finite amount of time, as seen in lines 18-20. This duration, $posn\_time$, is required to be able to reject sensor noise and false estimates of the parachute's motion before transitioning into the next phase. This also allows the system to ascertain that the motion of the vehicle matches with that of the parachute before trying to intercept it. We note that the choice of this duration can significantly alter the behavior of the system – if $posn\_time$ is too small, the system transitions too quickly into the interception phase, whereas a large value might prevent this transition altogether.

While attempting to converge towards the location of the parachute, the vehicle must necessarily maintain a safe minimum distance, $safe\_thresh$, from the canopy. This is enforced so that the vehicle does not interfere with the motion of the parachute due to prop-wash, and to prevent collisions. The violation of this constraint causes the system to transition into the state "unsafe pose" (line 22), which is a terminal state. In reality, any state may automatically transition into this state when a predefined condition is violated (undefined state estimates, failure in image acquisition, exceeding the physical bounds etc.). The mission is aborted in the "unsafe pose" state and the control of the vehicle is returned to the pilot.

C. Interception

In the final and the most critical phase, the UAS makes an attempt to trap the parachute. A transition to this phase is triggered only once the vehicle has maintained close proximity to the parachute for a finite amount of time ($posn\_time$). In Section II, we established that there is a high amount of uncertainty in the position of the parachute while in descent. Consequently, the final interception must consider that:

- (C1) It is unsafe for the vehicle to approach the canopy too close for a prolonged period of time, and,
- (C2) The vehicle must gain velocity over the parachute to be able to trap it.

Since there might be several potential maneuvers to achieve these, Algorithm 1 separates it out as a function (MANEUVER-CONTROL in lines 32-41). We implement it as a timed swooping maneuver that is comprised of two parts: a descent for t_{desc} seconds, followed by a pitch forward for t_{fwd} seconds, both performed at the maximum vehicle velocity (V_{uas}). The goal of the motion is to position the

vehicle such that

$$D_{\mathcal{V},\mathcal{P}}(t_{init} + t_{desc} + t_{fwd}) \leq L_h, \tag{4}$$

where t_{init} is the time that the interception phase was triggered. Figure 3 (far right) shows a pictorial representation of this maneuver. The controller begins executing the maneuver when the conditions for the state transitions are met. At the start of the maneuver, the vehicle is positioned $D_{\mathcal{V},\mathcal{P}}(t_{init})$ distance away from the parachute. The maneuver aims to minimize the separation between the hook and the parachute, $D_{\mathcal{H},\mathcal{P}}(t_f)$, where $t_f (= t_{init} + t_{desc} + t_{fwd})$ is the time at which the interception is projected to happen. Note that one limitation of this strategy is that while executing this pre-planned maneuver, the vehicle does not make further corrections to its positioning relative to the parachute.

Defining the interception mechanism in this manner not only satisfies (C1) and (C2), but also allows for two advantages: 1) in case the parachute is missed, the maneuver can simply be repeated at a later time, and 2) the parameters in Equation (4) can be modified to fit the requirements of any aerial system. The values for t_{desc} and t_{fwd} are computed knowing the values for V_{uas} and v_p (maximum velocity of the UAS and terminal velocity of the parachute, respectively). We also note that the proposed phases are designed around the constraints defined in Section II, and enable us to develop an instance of the solution space.

V. Numeric Bounds: Simulations

We now perform simulations to explore and evaluate the parameters that may impact the mission success.

The parameters of our setup, along with their ranges of values, are listed in Table I. Notice that certain parameters, such as V_{uas}, are dependent on the choice of the system and will invariably affect the values of other variables. Certain other parameters such as A_{par} (surface area of the canopy) and m_p (mass of the payload) can take any arbitrary value, but they are fixed in our experiments based on our sensing and system requirements.

Two key parameters that we explore in simulation are the length of the cable, and the initial separation at the start of the maneuver, L_h and $D_{\mathcal{V},\mathcal{P}}(t_{init})$, respectively. We use the results from the simulations to identify a suitable range of parameter values that give us a higher probability of a successful interception. The rest of the variables are held constant. We note that the simulation does not try to account for all the parameters of the parachute motion (Equation (1)), and does not fully describe the highly stochastic motion. Nonetheless, it allows us to explore and understand these parameters faster than we can do with experiments alone.

For the simulations, we use MATLAB Simulink and model the UAS as a 6-DOF object with an end-effector (hook) suspended from the vehicle at the end of a series of 16 short rigid joints. The number of joints was chosen to mimic the observed physical characteristics of a connecting cable. The parachute-payload system is modeled as a 3-DOF system that has lateral and vertical forces acting on it. Equation (1) governs the descent of the parachute, with f_{par} modeled as a time-varying random noise; and the lateral motion of the parachute is affected by a constant wind (1.5m/s). The descent of the parachute is computed for $t_{desc} + t_{fwd}$ seconds, and the simulation then replicates the swooping maneuver defined in Section IV. For each value of L_h and $D_{\mathcal{V},\mathcal{P}}(t_{init})$, the simulation is run 10 times with different noise parameters, and the results are averaged.

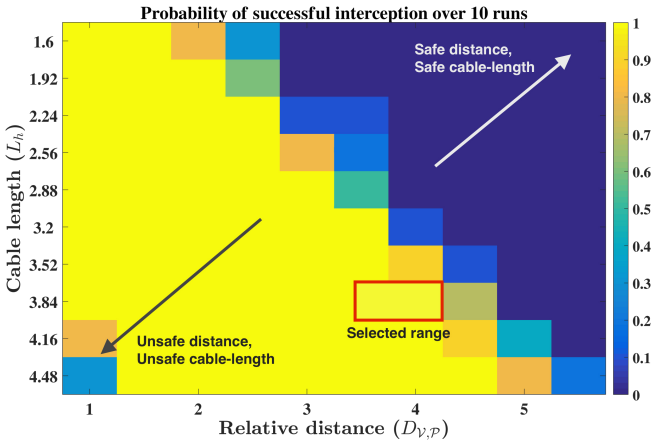

Fig. 4: Probability of a successful interception for different values of L_h and $D_{\mathcal{V},\mathcal{P}}(t_{init})$, averaged over 10 simulations each with randomized environmental disturbances. A larger value of L_h and a smaller value of the initial separation are favorable for a higher probability of success. We operate in a more conservative zone, highlighted in red.

Figure 4 shows these results for a range of values of L_h and $D_{\mathcal{V},\mathcal{P}}(t_{init})$. Clearly, given the variability in the position of the parachute, the results favor a longer cable to achieve a higher probability of a successful interception. At the same time, positioning the vehicle closer to the parachute gives it a better chance of intercepting it, unless the cable is extremely long. Additionally, there are regions in the figure which assure very high likelihoods of a success or a failure. For example, if the UAS positions itself 5m farther away from the parachute and then attempts to intercept it using a short cable, it fails. On the other hand, using a long cable and positioning closer to the parachute is almost always successful.

We note that in practice, however, there are physical restrictions on both these variables. A longer cable adds to the weight of the system, and poses a threat to the vehicle through its swinging motion. Similarly, approaching the parachute in close proximity increases the risk of the canopy getting deformed, and possibly getting stuck in the propellers. We therefore select a value for each of these parameters closer to the center in Figure 4. This is highlighted as "selected range" in the figure, and is a conservative selection that uses one of the longer cable lengths and then maximizes the relative distance for safety.

VI. Experimental Study

We conduct field experiments to assess the viability of the proposed method, and to identify factors associated with successes and failures.

TABLE I: A list of variables that affect the parachute, the UAS and the interception problem.

Variable	Description	Typical range	Experimental Value
A_{par}	Surface area of the canopy	0.125m^2-20m^2	1.2m
L_{par}	Length of the shroud lines	1-2 A_{par}	1.2m
m_p	Mass of the payload	50g-20kg	75g
W	Environmental wind velocity	0-8m/s	0.4 - 1.4 m/s
$D_{\mathcal{V},\mathcal{P}}(0)$	Initial separation from the parachute	0-100m	0m
V_{uas}	Maximum velocity of the UAS	3-10m/s	4m/s
L_h	Length of the cable suspending the end-effector	1.5m-5m	3.5m
W_h	Width (size) of the hook	0.2m-1.5m	0.4m
$D_{\mathcal{V},\mathcal{P}}(t_{init})$	Distance at which the maneuver starts	3-20m	4m

The UAS used in our experiments is an Ascending Technologies Firefly, which is a medium-scale hexrotor that measures about 66cm across the tips of the propellers, can support a payload of up to 600g and has a flight time of 10-15 minutes. We augment the vehicle with an Odroid XU4, which is a compact single-board computer with 8 processing cores and 2GB of RAM. We use two Matrix Vision BlueFox devices, mounted as a downward facing camera and a front-facing camera with overlapping field-of-views. Each of these have a resolution of 752×480 pixels and we acquire images from them at up to 20 fps. A wide-angle lens (approx. 70° vertical and 110° horizontal) is used to enhance the field of view of the sensors. All control commands for the vehicle are computed on-board the Odroid.

The parachute we use is of parasheet-type, made of red colored nylon rip-stop material and measures 1.22m in diameter when laid out flat. The mass of the suspended payload is 75 grams (this keeps the steady-state terminal velocity of the parachute below 2m/s). These parameters keep the system within the constraints introduced in Section IV.

To track the location of the parachute, we use a color-segmentation algorithm that utilizes OpenCV libraries [15] to detect the red canopy. The calibration parameters of the camera then allow us to estimate the distance to the parachute using the visible geometric area of the canopy (Equation (2)). We use an image-based visual servoing method that utilizes Equations (3) to generate attitude commands for the vehicle. In our current implementation, the control over the vehicle's yaw is kept with the remote pilot to avoid challenges that may arise due to the vehicle's slower yaw-rate. The algorithm processes images from only one camera at a time, and scans the other only if the parachute is not detected in one.

The 'x'-shaped hook is made of PLA plastic and is attached to the vehicle using a light-weight semi-elastic tube. From the simulations described in the previous section, we choose L_h to be 3.5m as the maximum length that allowed safe flight. This unactuated end-effector additionally has metallic prongs that are intended to trap the parachute when they come in contact with the parachute shroud lines.

Experimental Setup and Procedures

To aid in repeatability and ease experimental evaluation, we use the same vehicle to release and later capture the parachute. We found in our initial experiments that using two vehicles, however ideal, introduces uncertainty in coordination between pilots. The parachute is held and released by a single-servo pincer located at the end of the hook.

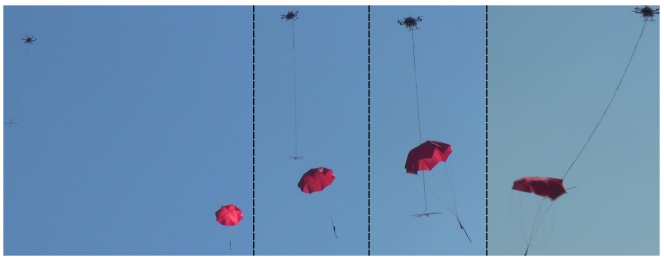

Fig. 5: Four snapshots from a successful mission. The successive images show the UAS approaching the parachute from a distance, positioning itself before the maneuver, swooping towards the parachute, and finally trapping it in the hook.

A complete mission can be described as follows. The parachute is affixed to the pincer and the vehicle is piloted to an altitude of about 50m, at which time the parachute is released and the onboard computer starts its mission. The vehicle chases the parachute through its descent and the mission completes when the interception is successful, or when the parachute reaches the ground. Table I lists the values chosen for the various parameters in the experiments.

Due to the variability introduced in the way the parachute is dropped by the vehicle (e.g., motion of the releaser, parachute packaging), we discard certain runs in which the parachute fails to open in a timely fashion, or is otherwise entangled upon release. Strong environmental factors such as gusts of wind can also cause the parachute to attain higher lateral velocities than the vehicle can achieve. Parachutes that have a parasheet-type structure also tend to be more susceptible to a swinging motion of the payload that eventually causes the canopy to not generate enough drag to slow the parachute down. In such cases, the pilot aborts the mission for safety by taking control of the vehicle.

In our evaluation of the experiments, we define a run as a viable run if 1) it does not suffer an improper release, and is therefore within the constraints we defined earlier, and 2) the vehicle is able to keep up with the parachute at its maximum velocity. The latter is important so as to ensure that the vehicle is able to get to the "Positioning" stage before considering the success or the failure of interception. We conduct our experiments during early mornings (to avoid thermal updrafts) and under wind-speeds of up to 1.5m/s (to prevent high parachute velocities).

VII. Results

We conducted 10 field runs to assess the potential of the approach and to identify the factors that affect its performance. Table II shows a summary of the runs. We label

TABLE II: Summary of results and factors involved in the mission. 100% of ϕ, θ correspond to maximum vehicle speed.

Runs	Result	Run time	Interception triggered	#phase changes	Last ϕ, θ	Miss cause
1	Success	22.7s	18.7s	1	56%, 16%	-
2	Success	62.3s	58.3s	12	15%, 14%	-
3	Success	63.1s	59.1s	13	49%, 59%	-
4	Success	14.5s	10.5s	6	21%, 16%	-
5	Success	23.6s	–	11	81%, 08%	-
6	Miss	41s	36.9s	2	38%, 88%	Relative orientation
7	Miss	32.0s	28.0s	3	20%, 63%	Relative orientation
8	Miss	35.3s	31.3s	9	62%, 24%	Relative orientation
9	Miss	53.3s	49.3s	3	26%, 89%	High v_p
10	Fail	12.3s	–	4	22%, 91%	Unsafe pose

a run as "success" when the vehicle intercepts and traps the parachute in the hook, as "miss" when the vehicle tracks and attempts the intercept but does not trap the parachute, and as "fail" when the vehicle is unable to transition into the intercept maneuver.

Five of the runs were successful, four were near misses, and one resulted in failure. Figure 5 shows four stills from one of the successful runs: the UAS autonomously approaches the parachute in its descent until the conditions for a state transition are met, positions itself, and then begins the swooping maneuver to trap the parachute in the hook.

Differentiating attributes of a successful run. We logged and analyzed an extensive set of attributes to determine the ones that distinguish a successful from a missed run. For instance, we analyzed whether or not the time spent in each phase, and the repeated transitions from tracking to positioning phases mattered (corresponding to Table II columns 3 and 5). In the end, two correlated factors seem to offer ways to characterize successful runs.

First, the position of the parachute before the vehicle begins the interception phase has a distinguishable effect on the probability of success. Figure 6 shows the last four locations of the parachute in the image frame before the vehicle attempts an interception. The figure shows the combined view from both cameras such that the vehicle's forward motion is along the positive y axis of the image. We immediately note that in all the successful runs the parachute is closer to the center of the vehicle (inside the downward camera's field of view).

Second, as a slightly weaker indicator, the last set of commanded values for either pitch or roll that the controller generated before the interception was triggered (Table II, column 6) are typically smaller for successful runs. Notice that the last values of pitch commanded to the vehicle also tend to be lower for successful runs. This corresponds directly with the location of the parachute in Figure 6 – the successful runs are closer to the origin of the pitch and roll axes. We also note that there might be instances where these values might be higher (run 3) and still lead to a successful interception owing to a favorable motion of the canopy.

Factors contributing to misses. In three of the runs (6, 7 and 8), the vehicle was either oriented slightly off the course of the parachute, or the parachute had an abrupt change in its course. This is also characterized by relatively larger values of the commanded pitch and roll angles in the positioning phase – implying that the parachute has significant lateral motion. As a result, the hook misses the parachute by swinging to its side. This behavior is similar to the one shown in Figure 2. In another instance of a miss (run 9), the parachute moves faster than expected in its descent (vertically), and is consequently outside of the reach of the end-effector during the interception.

Underlying these misses are position estimate errors that can attributed to two sources that become particularly evident in the interception phase. First, there is error in estimating the UAS position. To illustrate, Figure 7 captures the variation in the motion of the vehicle while enacting the interception phase. The figure shows the top- and the side-view of the actual GPS positions over 7 repeated enactments. We note that the altitude of the vehicle at the end of the maneuver can vary by as much as 2m, which is roughly twice the height of the selected parachute. Second, and as discussed earlier in Section II, there is uncertainty about the position of the parachute. When combined, these sources of errors compound, potentially contributing to a missed interception even in perfectly ideal conditions.

Surprises. One of the key challenges we have highlighted across the paper is the the uncertainty in estimating the motion of the parachute. In some cases this can result a successful run. In run 5, for example, the interception stage is not triggered, but the hook was able to entangle the parachute nevertheless as the motion of the parachute pushed it closer to the hook before the vehicle reacted (an analysis of the logs revealed that the interception phase would have triggered in less than 0.5s anyway). In other cases, such as in run 10, the fluttering of parachute's canopy resulted in an incorrect estimate of $D_{\mathcal{V},\mathcal{P}}$, the relative distance to the parachute. This caused the vehicle to get too close, and the state machine transitioned to "unsafe pose" to prevent a mishap. It may be possible, however, to resume the mission if the parachute continues its descent without significant disturbances.

Increasing the success rate. As noted in V, using a longer cable and approaching closer to the parachute before interception are favorable for a higher success rate. However, practical limitations prevent us from doing so. In our trials, the vehicle never transitions back into "tracking" after a missed interception because the parachute has already reached the ground. The number of retries are limited due to the low altitudes we operate at, and typically have less than $60\,\mathrm{s}$ to capture the parachute. Clearly, increasing this altitude would enable repeated interception attempts. Our current set of experiments also did not enable yaw corrections by the controller. We envision that making micro-adjustments to the UAS's orientation will minimize the probability of a miss.

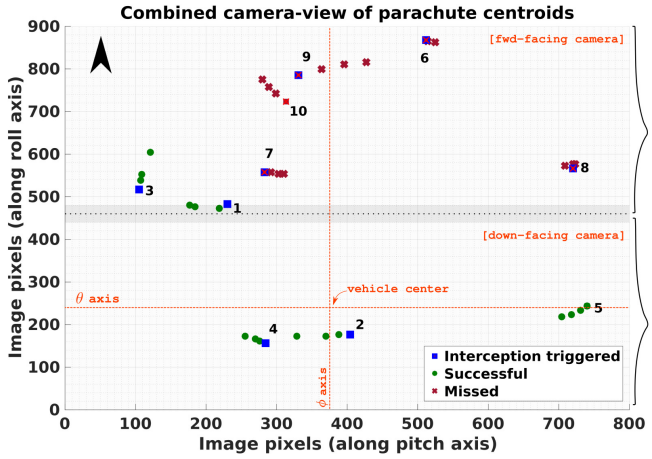

Fig. 6: The last four locations of the parachute's centroid location before the interception phase is triggered, as captured in the image frame. The vehicle pitches forward in the positive y axis of the image. The successful runs are all inside the view of the downward-facing camera.

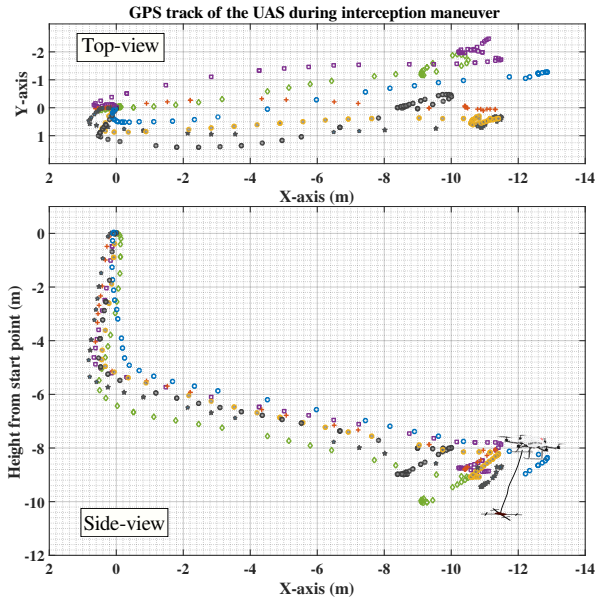

Fig. 7: A depiction of the interception maneuver that the vehicle performs in order to intercept the parachute, overlaid on GPS tracks from seven sample maneuvers by the vehicle. The variability in the total change in altitude is 2m, whereas it is 5m in the x-axis.

VIII. Conclusions and Future Work

We have presented a general approach and implementation for autonomously tracking and intercepting a loosely modeled parachute-payload system by a UAS. We first discussed the difficulties involved in the mission, and then proposed a three-phase approach to perform an interception maneuver. We then ran simulations to perform an analysis of the range of variables that contribute to the success of the mission. Finally, we evaluated the system through field tests which resulted in 5 successful and 4 missed interception and one failed mission. We also provide insights into the performance of the system, and discuss improvements that increase the success rate of the missions.

In the future, we will design experiments for releasing a parachute from one vehicle and interception by a second vehicle. We will also further relax the constraints and extend our approach to allow 1) the controller to regulate the yaw-rate of the UAS so as to reduce some of the cases of failures, and 2) multiple interception attempts by the UAS to compensate for the misses.

Acknowledgment

We would like to thank the members of NIMBUS Lab (Evan Beachly, Najeeb Najeeb, John-Paul Ore, Adam Plowcha and Andrew Rasmussen) for their invaluable help with numerous field trials. Special thanks to Seth Doebbeling for his initial design of the parachute releaser.

References

[1] NCAR Airborne Vertical Atmospheric Profiling System (AVAPS), Earth Observing Laboratory. [Online]. Available: https://www.eol.ucar.edu/observing_facilities/avaps-dropsonde-system

[2] F. Rafi, S. Khan, K. Shafiq, and M. Shah, "Autonomous target following by unmanned aerial vehicles," in *Defense and Security Symposium*. International Society for Optics and Photonics, 2006, pp. 623 010–623 010.

[3] C. Teuliere, L. Eck, and E. Marchand, "Chasing a moving target from a flying UAV," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 4929–4934.

[4] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, 2002, pp. 2799–2804.

[5] L. N. Long and H. Weiss, "The velocity dependence of aerodynamic drag: A primer for mathematicians," *The American mathematical monthly*, vol. 106, no. 2, pp. 127–135, 1999.

[6] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3400–3407.

[7] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 590–596.

[8] S. Saripalli and G. Sukhatme, "Landing on a moving target using an autonomous helicopter," in *Field and service robotics*. Springer, 2006, pp. 277–286.

[9] E. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 53 – 70, 2005, advances in Robot Vision. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889005000643

[10] A. Borowczyk, D. Nguyen, A. P. Nguyen, D. Q. Nguyen, D. Saussié, and J. L. Ny, "Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle," *CoRR*, vol. abs/1611.07329, 2016. [Online]. Available: http://arxiv.org/abs/1611.07329

[11] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, "Toward image based visual servoing for aerial grasping and perching," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2113–2118.

[12] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "The yale aerial manipulator: grasping in flight," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2974–2975.

[13] R. Spica, A. Franchi, G. Oriolo, H. H. Bülthoff, and P. R. Giordano, "Aerial grasping of a moving target with a quadrotor uav," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4985–4992.

[14] R. D. Mulcahy, *Corona Star Catchers: The Air Force Aerial Recovery Aircrews of the 6593d Test Squadron (Special), 1958-1972*. Center for the Study of National Reconnaissance, 2012.

[15] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.