Towards In-flight Transfer of Payloads Between Multirotors

Ajay Shankar[1], Sebastian Elbaum[2], and Carrick Detweiler[1]

***Abstract*— Multirotor unmanned aerial systems (UASs) are often used to transport a variety of payloads. However, the maximum time that the cargo can remain airborne is limited by the flight endurance of the UAS. In this paper, we present a novel approach for two multirotors to transfer a payload between them in-air, while keeping the payload aloft and stationary. Our framework is built on a visual-feedback and grasping pipeline that enables one UAS to grasp the payload held by another, thereby allowing the UASs to act as swappable carriers. By connecting the payload outwards along a single rigid link, and allowing the UASs to maneuver about it, we let the payload remain online while it is transferred to a different carrier. Furthermore, building entirely on monocular vision, the approach does not rely on precise extrinsic localization systems. We demonstrate our proposed strategy in a variety of indoor and GPS-free outdoor experiments, and explore the range of operating limits for our system.**

I. Introduction

Multirotor UASs are immensely useful aerial platforms for carrying and transporting a variety of payloads. Their ability to hover at precise, and often hard to access locations is an added advantage when a certain sensor package may need to be "hoisted" around a volume of interest. However, the maximum duration that a payload can be held aloft, and the maximum distance across which it can be transported, is limited by the flight endurance of the vehicle. We therefore envision a scenario where a payload, such as an expensive thermal imager monitoring a wildfire, could remain "online" even while it gets transferred to another UAS. Thus the transfer must happen *in situ*, such that replacing the UAS is inconsequential to the payload. Moreover, in many cases, an aerial transfer is preferable over the simpler approach of landing and swapping the payload. For instance, a precision landing might be challenging over remote and uncertain terrains, or infeasible when flying over water bodies.

In this work, we develop and evaluate a means for multirotors to transfer payloads between them. This can be non-trivial in the absence of motion capture systems that enable very accurate relative positioning. Furthermore, coordinating such a transfer in a way that minimizes the contact duration between the UASs requires precise monitoring and feedback. Our previous work has addressed some of these challenges by completely detaching the payload on a parachute [1] and by actively swinging it about the vehicle's yaw axis [2]. These methods do not keep the payload oblivious to the transfer and require it to undergo large motions, which may not be suitable for some payloads (e.g. a camera).

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

[2] Department of Computer Science, University of Virginia, Virginia, USA. selbaum@virginia.edu

Fig. 1: A snapshot from outdoor tests where a payload support bar originally held by a UAS (right) is grasped mid-air by another for transferring a payload.

Furthermore, when deployed on a parachute, neither UAS is in full possession of the payload while being transferred. On the other hand, actively swinging a heavier payload might adversely impact the flight stability of the UASs.

In this paper, we present a novel strategy that enables the transfer of a cargo payload from one carrier multirotor to another, while they are both airborne. As shown in the snapshot in Figure 1, our proposed method involves one UAS actively grasping a support bar connecting a payload held by another. A fundamental challenge in this mission is the risk posed by the two UASs flying in close proximity without precise, external localization systems. We mitigate the risk of collisions by extending a rigidly linked support bar horizontally away from the carrier UAS. By precise closed-loop control of the second UAS and active grasping of this support bar, our approach keeps the payload in either UAS's possession, thus allowing it to stay airborne for longer. However, introducing the support bar as a grasping point also leads to non-trivial challenges in its localization and coordinated grasp/release. Furthermore, the dynamic change of mass must also be compensated for in the control strategy. The key contributions of this work are:

- A novel strategy for transferring a payload carried by one UAS to another, while keeping the payload hovering in-place;
- A fast and efficient visual-feedback control pipeline, coupled with a single-axis gripper, that allows multiple UASs to pass objects amongst them; and,
- Indoor and outdoor studies that demonstrate autonomous payload transfers between two multirotors, a characterization of the effectiveness of the proposed approach through 24 tests in total, and a detailed discussion on the elements of system design.

II. Related Work

Transfer of airborne payloads between UASs has not been studied extensively. The closely-related problem of multiple aerial vehicles interacting with the same payload object, however, has been previously demonstrated in the context of collaborative load carrying. Specifically, cooperative flight and leader-follower configurations for load transport have been shown to perform well using only visual cues within semi-constrained environments [3], [4]. More generalized n-vehicle control frameworks [5], [6] have also explored how multiple UASs can dynamically reorient themselves for safe and efficient transport of a common, suspended payload inside motion-capture environments. While these proposed approaches tackle the problem of collaborative transport, it is typically assumed that the UASs are already connected to the payload. That is, the vehicles do not dynamically transfer or share the ownership of the payload. On the other hand, several approaches have been developed and field-tested for UASs that grasp and manipulate objects while hovering airborne [7], [8], or moving at some speed [9]. However, they do not focus on systems and mechanisms that involve another UAS that is simultaneously airborne, which introduces constraints such as an uncertain motion of the payload (induced by another UAS), and challenges with proximity and mission time. As also demonstrated by Dollar *et al.* [7], the dynamic change in aircraft loading must keep the flight controller within a stable operating region.

Vision-based control approaches play a critical role for UASs interacting with extremely precise contact locations and objects with fast dynamics. These have been exploited for chasing and landing on ground-based targets [10], [11], [12], [13], as well as for intercepting airborne targets [1], [14]. Precise visual servoing has also been employed for perching, grasping objects, and navigating through cluttered spaces [15], [9]. Instead of relying on visual cues from the environment, several of these approaches use fiducial markers for state estimation. Similar to previous approaches that have used a pattern of light-emitting diodes (LEDs) to compute perspective [16], [17], we employ LEDs as the fiducial marker mounted at the tip of the support bar. Even with the penalty of added weight, we prefer it over passive tags (such as AprilTags [18]) to aid fast state estimation.

A closely related problem is the one of aerial vehicles connecting with each other mid-air to form conjoined flying structures. This has recently seen some successful demonstrations within semi-structured environments [19]. Methods for landing and autonomously swapping batteries have been presented in prior work [20]. Our previous work has also demonstrated how a cargo payload can be transfered from one airborne UAS to another by agitating the payload [2]. In this work, we present a strategy for a transfer of payload between two UASs *while* they remain airborne, so that the payload can remain online during the transfer.

III. Technical Approach

Our objective is to enable two UASs, referred to as *Alpha* and *Bravo*, to perform an autonomous aerial transfer of payload between each other. Without relying on precise motion-capture systems, the systems must ensure that the vehicles or the payload are not jeopardized during the mission. We consider such a fully autonomous mission successful when *Bravo* grasps the support bar originally held by *Alpha*, completely disengages it, and flies away with it afterwards.

Figure 2a shows a graphical illustration of the mission phases of our approach. We assume at the start of the mission that the first vehicle, *Alpha*, performs a stationary hover with the payload support bar. In practice, this support bar connects with the payload and protrudes horizontally outwards from *Alpha*'s center. The tip of this bar has active LED markers mounted on it to aid *Bravo* in relative localization and identification of the grasping location. We also assume that the two vehicles can get within relatively close proximity to each other, for instance, by agreeing on a common GPS location. Our technical approach to performing the payload hand-off between the two vehicles is split into 4 stages, shown in Figure 2a:

1. *Bravo*: Pose estimation relative to support bar
2. *Bravo*: Stable flight relative to *Alpha*
3. *Bravo*: Approach and grasp the support bar
4. *Alpha*: Coordinated release of the support bar.

We now present the details of each of these stages.

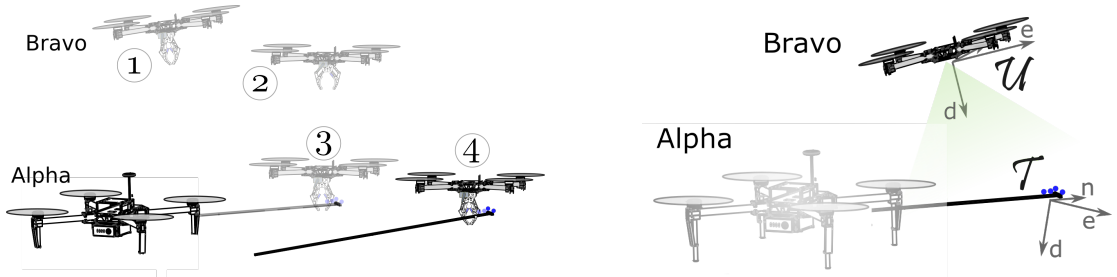
A. Bravo: pose estimation

We employ a monocular vision feedback system for *Bravo* to estimate its full pose relative to the support bar held by *Alpha*. Visual feedback control is easily adapted to a variety of scenarios (indoor vs. outdoor) and is advantageous when precise localization is critical. Our approach is independent of the choice of markers, and can easily generalize to several other monocular state estimation methods [21], [22]. We develop a position-based visual servoing (PBVS) framework using a calibrated camera to control *Bravo*'s motion over desired trajectories. Using active markers for this purpose dramatically speeds up the target identification problem by admitting very small camera exposures. This, in turn, allows fast and unambiguous full state estimation to be run entirely onboard the UAS. Note that since *Bravo* does not use any extrinsic localization method, its "full state" in this context is always relative to a local origin at the markers' center.

Consider two NED (north-east-down) frames of reference for the target and the UAS, $\mathcal{T}$ and $\mathcal{U}$, respectively, as shown in Figure 2b. We denote the location of the UAS in the $\mathcal{T}$ frame as $P^{\mathcal{T}}_{uas}$. A calibrated camera is rigidly mounted under the body-center of the UAS such that it is pointed along the positive z axis of $\mathcal{U}$. The location of a target t in the UAS's frame, $P^{\mathcal{U}}_t = [n_t, e_t, d_t]^T$, is related to its projection in the pixel-space (u_t, v_t) as,

$$\begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} = K_C * [R_C | T_C] * \begin{bmatrix} n_t \\ e_t \\ d_t \\ 1 \end{bmatrix}, \tag{1}$$

where, K_C represents the camera's intrinsics (obtained from calibration), and $R_C|T_C$ denotes its extrinsic rotation and



(a) (b)

Fig. 2: a) An illustration of the four system phases described in the mission: ① pose estimation, ② stable flight, ③ approach and grasping, and ④ coordinated release; b) The frames of reference used in this work. The $\mathcal{T}$ frame denotes the position of the markers.

translation elements stacked together. We recover the extrinsics, $R_C|T_C$, using a Perspective-from-N-Points (PnP) algorithm [23] over the known points on the target pattern. The notation $R_C|T_C$ is used to describe the rotation and translation of $P_t^{\mathcal{U}}$. The 3D pose of the UAS relative to the fixed-frame of the target is then obtained by

$$P_{uas}^{\mathcal{T}} = R_C^{-1}T_C = -R_C^T T_C. \tag{2}$$

The calculated P_{uas} and $\dot{P}_{uas}$ from this step are later used as measurements for the estimator and controller.

B. Stable flight relative to Alpha

Once *Bravo* is able to establish its pose using visual feedback, it attempts to maintain a stable hover relative to *Alpha*. Our control strategy for the UAS is based on a linear quadratic regulator (LQR) with non-linear output mapping. The state equations for the UAS are derived from a feedback-linearized system model such that,

$$\dot{\hat{\boldsymbol{x}}} = A\hat{\boldsymbol{x}} + B\boldsymbol{u}, \text{and } \boldsymbol{y} = C\boldsymbol{x}, \tag{3}$$

with

$$A = \begin{bmatrix} 0_{3x3} & I_{3x3} \\ 0_{3x3} & 0_{3x3} \end{bmatrix}, B = \begin{bmatrix} 0_{3x3} & 0 \\ I_{3x3} & 0 \end{bmatrix}, C = I,$$

where the UAS's state vector is denoted by $\boldsymbol{x} = [P_{uas}, \dot{P}_{uas}]^T$, and, $\boldsymbol{u} = [u_\theta, u_\phi, u_\psi, u_T]^T$ represents the acceleration inputs to the system in each degree of freedom. Note that we omit yaw from the state vector only for an ease of notation. We set it to zero without loss of generality, although it could be easily adapted to arbitrary orientations. The $\hat{\boldsymbol{x}}$ used in the system model is produced by a Kalman filter, and represents the best estimate of the state $\boldsymbol{x}$. The filter uses the system model in Equation (3), and runs at a rate higher than that of the camera pose estimation. The different phases of the mission highlighted in Figure 2a benefit from different closed-loop system response. We employ a gain-scheduled framework that allows the system to vary the control effort at these different stages (for instance, based on the proximity to the support bar). The control feedback law for Equation (3) is then a function of the system state, such that,

$$\boldsymbol{u} = -K_{phase}\hat{\boldsymbol{x}}_e, \tag{4}$$

where $\hat{\boldsymbol{x}}_e \equiv \hat{\boldsymbol{x}} - \boldsymbol{x}_r(t)$ and $\boldsymbol{x}_r$ is a reference function. For this phase, $\boldsymbol{x}_r$ is a constant to maintain a stable hover. The feedback gain $K_{(\cdot)}$ is precomputed by solving the algebraic Ricatti equations for different weights in the matrices Q and R for a standard LQR cost function. Finally, using the total mass, m, of the system, the acceleration inputs from Equation (4) are mapped to corresponding attitude and thrust targets for the inner attitude controller, $[\theta_d, \phi_d, \psi_d, T] = g_{kin}(\boldsymbol{u}, m)$. This non-linear map, $g_{kin} : \mathbb{R}^5 \rightarrow \mathbb{R}^4$, is obtained by inverting the UAS's kinematics [24].

Assessing a stable phase: Note that *Bravo*'s pose is measured in the $\mathcal{T}$ frame, which is affixed to the support bar and is, in general, non-inertial. As a result, *Bravo* cannot directly infer whether the bar is undergoing a turbulent phase (for instance, because of *Alpha*'s non-ideal hover). To mitigate this challenge, we use the variance in the estimator's $\hat{\boldsymbol{x}}$ over a small time interval. Assuming *Bravo*'s body-frame rotations are negligible over an interval, t_{stab}, the variance in estimated state must be due to the bar's accelerations. Hence, $\sigma \equiv var\big(\hat{\boldsymbol{x}}(t) \ldots \hat{\boldsymbol{x}}(t + t_{stab})\big)$ can be used as a metric to identify whether the bar is held steady. The system can transition into an approaching phase if $\sigma < \sigma_{th}$ for some interval t_{stab}.

C. Approach and Grasping

For grasping the support bar, the UAS must approach it and position itself accurately within relatively small error margins. We use the same underlying control strategy to guide the UAS to the target position by constructing $\boldsymbol{x}_r$ as a time-varying reference function. This trajectory is a linear constant-velocity path that terminates at the grasping point on the bar at a finite time, T_f.

Failure detection: When the vehicles are closer, small environmental disturbances and drift can more easily move the marker out of the camera's field of view. The controller uses the estimated pose, $\hat{\boldsymbol{x}}$, which can degrade rapidly if measurement updates are too sparse in time. Thus, the challenges of early failure-detection and reaction become critical at close proximities. Moreover, an unexpected motion of the target during the approach is unfavorable for a successful grasp. As a result, we employ a monitor which asserts that $|\hat{\boldsymbol{x}}_e| < x_{th}$ throughout the approach for each element in the vector, with $x_{th} \in \mathbb{R}^6$ denoting the threshold limits for estimated position and velocity. The approach phase is also constrained to a finite terminal time, T_f. Furthermore, while occlusions are rare, the monitor admits a threshold interval,

t_{miss}, for which the target is allowed to be out of the field of view. Violations of any of these constraints autonomously trigger a restart of the approach phase.

D. Coordinated Release

If the constraints during the approaching phase are met, and $||\hat{\boldsymbol{x}} - \boldsymbol{x}_r(T_f)|| < x_{tf}$, then *Bravo* attempts to grasp the support bar. When the grasp is attempted, the target has a high likelihood of being out of the field of view of the camera. In practice, therefore, an additional feedback signal indicating a successful or failed grasp is critical to the mission. An unexpected motion of the bar in the final stage can mean that a) the gripper misses the bar, or b) the grasp is unreliable or incorrectly positioned. The gripping subsystem contains a magnetic feedback element is activated only when the bar has been grasped successfully. If this feedback signal is available, *Bravo* then commands *Alpha* to release its hold, and is subsequently switched to a hover flight mode. Otherwise, the grasp attempt is aborted, and the mission is autonomously restarted in the approach phase.

The post-grasp hovering flight mode has a different mass loading for the vehicles. The inner-loop attitude controller uses a PID control strategy that remains within stable operating limits under loaded and non-loaded configurations. However, the transient response, as well as the thrust requirements, will be different under both scenarios. We use our gain-scheduled outer-loop control method (Equation (4)), and the inversion map g_{kin} to accommodate these two effects produced by this transfer of mass.

IV. Implementation and Experimental Setup

We first characterize the individual components of our visual servoing and control pipeline. We then demonstrate the capabilities of our proposed system using two commercially available quadcopters customized for our tests. Table I lists the values used for the constants in our implementation.

UASs: Alpha is a customized DJI Matrice 100 and is equipped with a GPS to maintain a stable hover outdoors. The payload bar is held in place by 3D printed supports and a servo-controlled hatch that is actuated to release the bar. *Bravo* is a DJI Flamewheel F450 frame outfitted with a gripper mechanism that is actuated by a servo. The sensing, estimation and control algorithms are run entirely onboard the vehicle on an Odroid XU4, and the visual feedback is obtained by a calibrated Bluefox MLC200c camera (752×480px) equipped with a wide-angle lens (approx. 110° horizontal field of view). The visual pose detection runs onboard the Odroid with the camera triggered at $60\,\text{fps}$, while the state estimation and control loops are implemented at $70\,\text{Hz}$ and $40\,\text{Hz}$, respectively.

The autopilot used is a Pixhawk board that runs customized Arducopter firmware. The inner attitude-control loop accepts desired attitude targets generated from Equation (4) after the mapping by g_{kin}. For the gain-scheduled implementation of Equation (4), we choose three sets of Q and R matrices for the hover, approach and post-grasp flight phases. These are chosen such that state errors are regulated

TABLE I: Listing of the values used in our implementation.

Variable	Value		Variable	Value	
	Indoor	Outdoor		Indoor	Outdoor
σ	0.0009	0.0016	t_{stab}	3.0s	3.0s
T_f	$\leq$10s	$\leq$10s	t_{miss}	1.35s	0.85s
x_{th}	$[0.50$	0.50	0.20	0.1	0.1	$0.1]^T$
x_{tf}	$[0.15$	0.15	0.10	0.0	0.0	$0.0]^T$

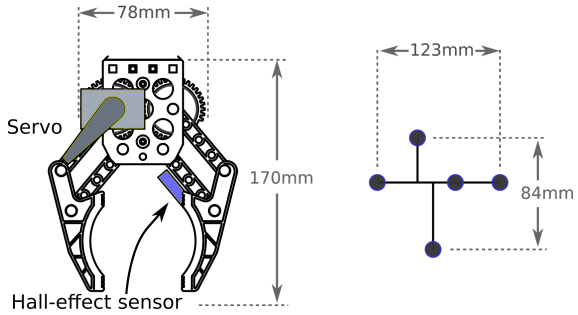


Fig. 3: [Left] A schematic drawing of our grasping mechanism[1], with the actuating servo and the hall-effect feedback sensor highlighted. [Right] The 5-point LED pattern used as the marker on the support bar.

at a faster rate during the approach phase (at the expense of a higher control effort). Furthermore, the control can be relaxed post-grasp, observing that *Bravo*'s mission no longer requires strict velocity tracking. To ensure stability, we verify that the eigenvalues of the closed-loop system are strictly on the left-half of the complex plane for each resultant $K_{(\cdot)}$. *Alpha* and *Bravo* use a pair of $2.4\,\text{GHz}$ XBee radios for a one-byte communication with each other only for the command/acknowledgment of bar release.

Support bar: The support bar carried initially by *Alpha* is a $1.1\,\text{m}$ long thin tube made out of light-weight carbon-fiber. The length of the bar is generally a tradeoff between two factors: 1) an extremely short length challenges the safety of operations, and 2) an extremely long bar might offset *Alpha*'s center of mass significantly, resulting in unstable flight dynamics. Through preliminary tests with our vehicles, we established approximately $1.5\,\text{m}$ to be a safe operational length of the bar. A predefined pattern designed using small LEDs, powered by *Alpha*'s power source, is affixed to the grasping end of the bar, along with a magnetic element. We use blue LEDs as they are easily distinguishable from the largely green and brown landscape below the UASs. *Alpha* employs a servo-actuated release mechanism that holds the bar and also simultaneously disconnects the power to the LEDs when commanded by *Bravo*. The support bar, with the LEDs and the magnetic element weighs $40\,\text{g}$, and protrudes approximately $1\,\text{m}$ outwards from the tip of *Alpha*'s propellers. In practice, a generic payload that needs to be transferred would be attached to this bar close to the center of mass of *Alpha*.

Gripper: Figure 3 shows a graphical schematic of our grasping mechanism, which is designed around a Vex Robotics model claw. A single servo provides actuation to both arms of the gripper which are connected by a gear

[1]Base gripper design from VexRobotics, `vexrobotics.com`.

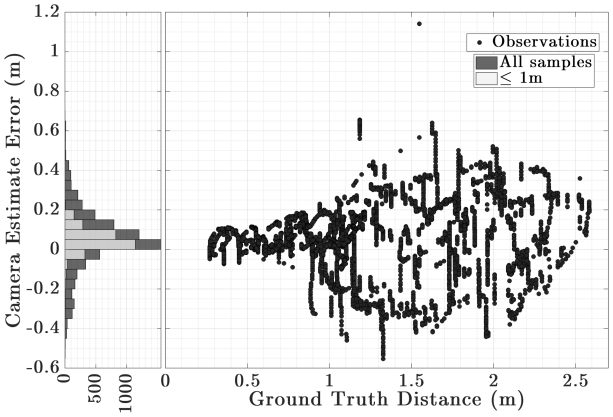


Fig. 4: The distribution of the error (euclidean distance) in camera estimated poses compared to the ground truth. The histogram on the left also shows the (count) distribution when the objects are within $1\,\mathrm{m}$ of each other.

system. The two arms therefore move symmetrically, and are mounted such that they meet the sagittal plane of the UAS simultaneously when closing. The grasping location on the payload bar contains a magnetic strip which is sensed by a hall effect sensor mounted inside the gripper arm. This allows the system to passively recognize a successful grasp, and helps reduce false and unreliable grasps. The gripper system weighs $75\,\mathrm{g}$, and has a maximal opening of $\sim 16\,\mathrm{cm}$.

Experimental Setup

We perform two types of experiments to evaluate the performance of our proposed strategy. *Indoor tests* are performed inside a closed arena with *Alpha* mounted on a raised platform, and *Bravo* executing its complete mission using only visual state feedback from the active LEDs. No external state estimation (such as a motion-capture system) is used. Furthermore, to simulate a more realistic scenario, the mounting platform for *Alpha* is not rigid, thereby admitting small motions of the bar (constrained to $\sim 20\,\mathrm{cm}$ in each axis) due to aerodynamic effects. These tests consequently evaluate the precision of the localization, control and grasping subsystems in a simulated ideal hover by *Alpha*.

Outdoor tests are then performed by flying *Alpha* in a stationary hover at a fixed GPS location. As before, *Bravo* executes its mission using purely visual-inertial feedback. The mission is entirely autonomous, and no pilot commands are used during any of the phases. All outdoor tests are conducted under low wind conditions (3–$5\,\mathrm{m/s}$) in order to reduce the impact of external disturbances on the vehicles.

V. Results

We begin by characterizing the individual components of our pipeline through indoor experiments in semi-constrained environments. Afterwards, we present results from our indoor and outdoor experiments that demonstrate the capabilities of our full system. The objective in these experiments is to 1) evaluate the controller's response in converging to the correct grasping point, and 2) determine the time and number of attempts the system takes to retrieve the payload.

A. Vision Pipeline

Our first evaluation focuses on the effectiveness of the visual feedback employed in the subsequent subsections. We rigidly fix an LED marker pattern to the world frame, while *Bravo* performs onboard pose calculation using the vision pipeline (Section III-A). *Bravo* is systematically positioned to view the pattern from a range of different poses. We record over 6500 such data points that cover a volume of locations in all three axes at a slow speed. A Vicon[2] motion capture system is used to acquire the corresponding ground truth locations of the marker and *Bravo* at $200\,\mathrm{Hz}$. Figure 4 shows the error in the 3-axis euclidean distance of *Bravo*'s relative pose produced by the visual feedback compared against the corresponding ground truth poses. Overall on average, over 74% of the sampled points fall within $0.16\,\mathrm{m}$ of the ground truth distance (roughly the maximal opening of the gripper). When the marker is in closer proximity (under $1\,\mathrm{m}$), the RMS error in the estimated distance is $0.07\,\mathrm{m}$.

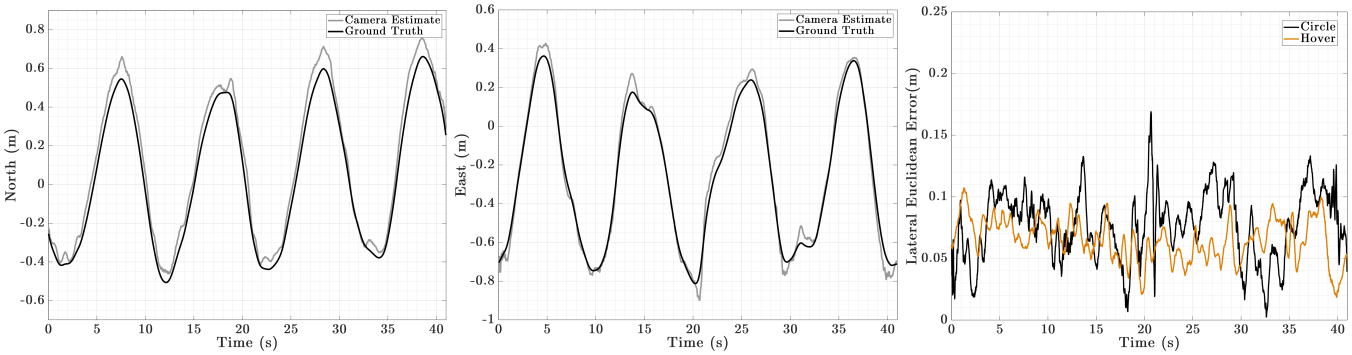
Processing a camera frame to produce P_{uas} takes approximately $12.2\,\mathrm{ms}$ on average on the Odroid. This allows us to easily trigger the camera at over $60\,\mathrm{fps}$. Since our estimation and control loops run at fixed rates ($70\,\mathrm{Hz}$ and $40\,\mathrm{Hz}$, respectively), the typical worst case latency in reacting to one camera frame is $12.2 + 1000/70 + 1000/40 \approx 51.5\,\mathrm{ms}$. On average, assuming that the arrival times for inter-process results are uniformly distributed between the process intervals, this latency drops down to $\approx 32\,\mathrm{ms}$.

B. Controller Performance

For the ultimate grasping experiments, we must ensure that the visual-control strategy enables precise maneuverability for *Bravo*. We perform this evaluation in a motion capture room with *Bravo* flying commanded trajectories using purely visual feedback from an LED pattern positioned stationary in the world frame. Figure 5(a)(b) show the results from flying a circular path around the marker at a fixed altitude with radius $0.5\,\mathrm{m}$ and a period of approximately $10.5\,\mathrm{s}$. Figure 5c shows the corresponding euclidean error in the lateral axes compared to the ground truth motion-capture data. The figure also shows the euclidean error for a stationary hover for the same duration. Through these experiments, we observe that the controller is able to maintain an average position error of less than $10\,\mathrm{cm}$ throughout. We also note a finite steady-state drift and bias for the vehicle. Since this bias can vary due to several factors such as vehicle calibration and battery voltage, in our experiments, we compensate for it by measuring it in the stable hover stage of the mission (Section III-B) *before* an approach is initiated.

This state error (and its temporal variance) can be higher in the presence of external wind. Figure 6 shows the cumulative distribution of the estimated variance in the pose during a $30\,\mathrm{s}$ hover in the presence of wind using purely onboard visual feedback. We observe that for winds up to $4-5\,\mathrm{km/h}$, the controller is able to meet the outdoor variance thresholds 75-90% of the intervals. In particular, the controller stays within $2\sigma_{th}$ for over 85% of the times under low wind

[2]https://vicon.com



(a) **(b)** **(c)**

Fig. 5: *Bravo*'s positions in (a) north and (b) east while flying a circular trajectory using our visual control pipeline. Using a motion capture system, we observe that the euclidean error shown in (c) is typically within $10\,\mathrm{cm}$.

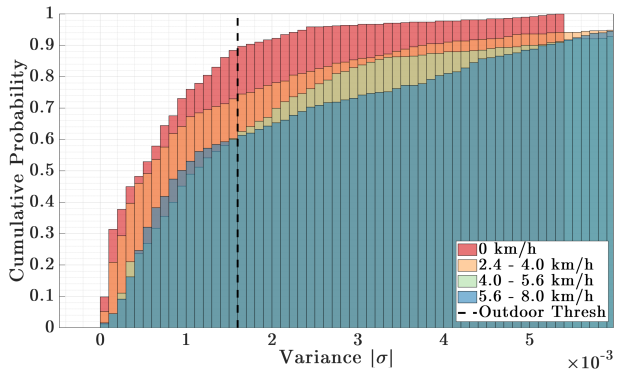


Fig. 6: Calculated variance in the pose for different wind speeds, viewed as a cumulative distribution (CDF).

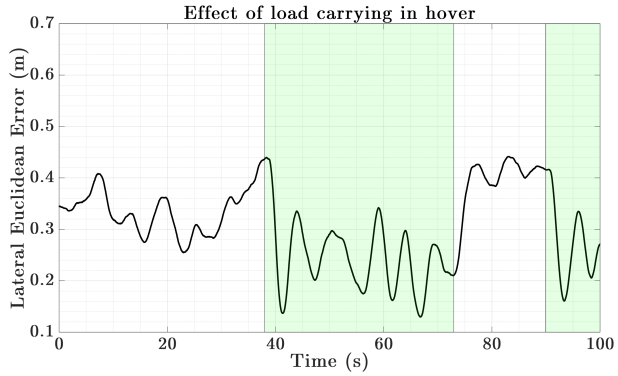


Fig. 7: Loading effect compensated using a gain-scheduled controller of the type described in Equation (4). The shaded region in green highlights the duration for which the controller switches for compensating for the added mass.

conditions. Even for higher wind speeds, nearly 60% of the samples always meet the outdoor threshold. The tail of the distribution is not shown for clarity. However, it can be observed that for all wind speeds, 90% of the samples fall within $3\sigma_{th}$.

Figure 7 shows our evaluation of the controller's compensation for the added mass on *Bravo* after the payload has been acquired. Recall that our non-linear output map, g_{kin}, from Section III-B already accounts for the mass of the system. Since the added mass is not centered around *Bravo*'s center of gravity, we employ our gain-scheduled control strategy from Equation (4). We switch the feedback gain post-grasp such that the position errors are penalized more heavily. In our characterization, we command *Bravo* to hover inside a motion-capture environment while carrying the support bar (simulating a post-grasp scenario) and switch the gains in-flight. The highlights in Figure 7 mark the regions where the post-grasp gains are applied. We observe that in doing so, the controller better reduces the effect of a biased loading from a mean error of $0.35\,\mathrm{m}$ down to $0.24\,\mathrm{m}$.

C. Indoor Grasping Tests

In all our grasping tests (indoors and outdoors), we randomize *Bravo*'s starting location relative to the support bar to validate the controller's performance for different initial conditions. The offset for these starting locations still lies within the camera's field of view.

We conducted 16 successive indoor trials in which *Alpha* is mounted on a semi-rigid mounting platform and *Bravo* localizes, grasps, and retrieves *Alpha*'s support bar. In all of our trials, *Bravo* was able to successfully grasp the bar and synchronize its release from *Alpha*. The polar plot in Figure 8a graphically illustrates the different initial positions for *Bravo* at the start of its mission. Figure 8b plots the trajectories of the error (measured as euclidean distance) between the target point and *Bravo* during the approach. As expected, the system spends some initial time in ascertaining a steady positioning before rapidly converging towards the grasping location. The final approach takes $\sim 9\,\mathrm{s}$ on average.

Figure 8c shows the number of approaching and grasping attempts made by *Bravo* before successfully finishing its mission. Recall that the system can correctly detect grasping failures and unsteady relative motions of the bar, and autonomously recover by restarting another approach phase. Consequently, the system makes an average of 3-4 attempts before it makes one with a high confidence. Figure 8c also plots the total time spent by *Bravo* in phases 2 and 3 (hover and approach). On an average, this is about $60\,\mathrm{s}$.

D. Outdoor Grasping Tests

We performed 8 outdoor field tests where both vehicles are simultaneously airborne, and *Bravo* autonomously localizes, graps and retrieves the support bar from *Alpha*. In only one instance (trial #1), the autonomous restart failed, requiring

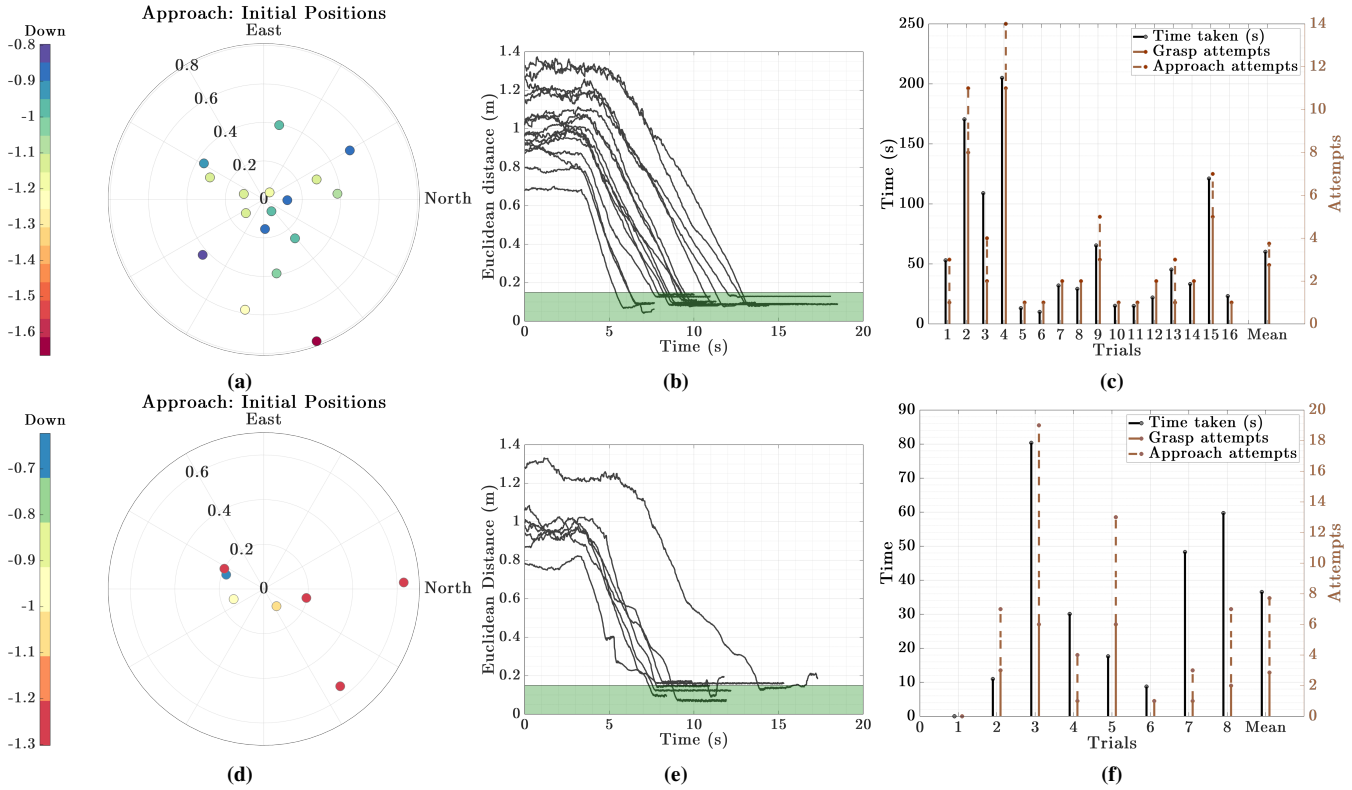


Fig. 8: Results from indoor ((a)-(c)) and outdoor ((d)-(f)) tests. (a)(d) Starting locations of the approach, with the colormap representing the vertical (z-axis) distance (meters); (b)(e) Approach trajectories of the error (euclidean dist.) in *Bravo*'s location relative to the grasping point on the bar. The highlighted area represents the acceptable region for triggering a grasp; (c)(f) A summary of key statistics.

pilot intervention. In the remaining 7 trials, *Bravo* was successful in its localization and grasping tasks. However in the last 3 trials, our gripper began experiencing mechanical fatigue, which rendered *Bravo* unable to retain the support bar after its release from *Alpha*, despite the grasper enclosing the bar.

Figure 8d shows *Bravo*'s starting location relative to the support bar. The starting locations follow a similar distribution to the indoor trials since this state is entered based on the position reported by the active markers. In Figure 8e, we show the trajectories of the euclidean distance between *Bravo* and the target point as it approaches it. These follow a trend similar to the ones observed in Figure 8b for our indoor tests. While there is some increased noise, especially noticeable when the error increases a few times, it does not significantly impact the control and estimation pipelines.

The number of trials and the time taken by the system varies significantly outdoors. Figure 8f shows the time *Bravo* spends in phases 2 and 3 (hover and approach), and on average this is approximately 126 s (more than twice compared to indoors). The average number of trials is also nearly twice (7-8) compared to indoors.

There is an expected difference in the statistics of our results from indoor and outdoor experiments. Our control and state estimation pipelines perform well in both scenarios. However, we notice that, in general, the outdoor tests take more time and number of trials. In particular, while flying outdoors, *Bravo* can make more than twice the number of approach attempts outdoors before it is confident on one and attempts to grasp. This is evident in Figure 8f, where, in trials #3 & #5, the system begins an approach 12-18 times, but aborts several of them. There are only 6 grasp attempts for either of these cases. Nevertheless, the average mission time is still close to 126 s.

We attribute these higher numbers primarily to the deviations from an ideal hover by *Alpha* (caused by environmental disturbances, downdraft from *Bravo*, etc.). Furthermore, note from Table I that we use stricter thresholds outdoors to aid the safety of operations. Consequently, the system is more conservative in its mission, and favors a retrial more frequently. However, trial #1 in outdoor tests failed such a restart in a timely manner due to errors in state estimation that allowed the system to incorrectly meet the thresholds.

VI. Discussion on Design Choices

The thresholds and parameters enumerated in Table I are selected empirically based on our choice of vehicles, the sensors and the payload, and the expected operating conditions. Some thresholds, such as T_f, are chosen to constrain the speed and the time spent in approach. Length of the support bar and the weight mounted on it are dependent on UAS's capabilities, and are identified prior through independent flights. This also influences the mission parameters such as x_{th} and x_{tf}, since larger values of these thresholds will admit larger deviations from a target system state. Similarly, σ_{th} regulates how steady the relative positioning must be

before an approach is attempted. This can be higher for systems that have longer support bars, or larger grippers. The mechanical design of the gripper also governs factors such as the variability in the grasping location, and the maximum allowable payload. Through indoor tests with the support bar, we have established that our system is capable of safe operations with an off-centered payload of up to $\sim 70-80\,\mathrm{g}$ additional to the support bar.

Our current design for *Bravo* uses a single-point gripper with a single contact region on the bar. This greatly simplifies the mechanical structure, while reducing the weight of both the gripper system and the feedback elements at the tip of the support bar. However, this poses two noticeable challenges: 1) upon release, the weight of the bar must be counteracted by frictional and normal forces at only one contact point, and 2) the bar will not remain horizontal during the post-grasp flight with *Bravo*. While this is advantageous in preventing an immediate collision with *Alpha*, it restricts the possibility of *Bravo* transferring it to another vehicle later on. In this work, we have not prioritized this element of the design. This can be extended to a synchronous two-point gripper to ensure that the grasp aids future transfers to other vehicles.

VII. Conclusions and Future Work

This paper has presented a novel strategy that allows two multirotors to transfer a payload between them while remaining simultaneously airborne. Our method requires a multirotor to simply hover in place, and extend a rigid link as a support bar to be grasped by the other. By employing precise visual servoing and active grasping, we are able to perform a transfer that requires no agitation of the payload. We demonstrate feasibility of our approach, and characterize the performance of its various elements through extensive tests.

In the next revisions of our system, we will use some of the lessons learnt from our field trials to improve the capabilities of our system. In particular, we plan on further improving the state estimation pipeline to relax the environmental constraints under which the systems operate. Modifications and updates to the gripping mechanism are also underway that allow a stronger grasp on the bar. As a result, our future work will also include the ability for *Alpha* to re-acquire the payload from *Bravo* by reversing the roles.

Acknowledgement

This work is partially supported by: NSF IIS-1638099, IIS-1924777, IIA-1539070, USDA-NIFA 2017-67021-25924, and CCF-1718040. Thanks to Alex Solokov for his contributions to the gripper modifications, and Chandima Fernando & Ashraful Islam for assisting field tests.

References

[1] A. Shankar, S. Elbaum, and C. Detweiler, "Towards Aerial Recovery of Parachute-Deployed Payloads," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[2] ——, "In-air Exchange of Small Payloads between Multirotor Aerial Systems," in *International Symposium on Experimental Robotics (ISER)*. Argentina: Springer, Nov. 2018.

[3] G. Loianno and V. Kumar, "Cooperative Transportation Using Small Quadrotors Using Monocular Vision and Inertial Sensing," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, Apr. 2018.

[4] M. Gassner, T. Cieslewski, and D. Scaramuzza, "Dynamic collaboration without communication: Vision-based cable-suspended load transport with two quadrotors," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5196–5202.

[5] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," *Robotics: Science and Systems*, 2013.

[6] V. Parra-Vega, A. Sanchez, C. Izaguirre, O. Garcia, and F. Ruiz-Sanchez, "Toward aerial grasping and manipulation with multiple UAVs," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, 2013.

[7] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 2491–2498.

[8] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two DOF robotic arm," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013.

[9] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, "Toward image based visual servoing for aerial grasping and perching," in *2014 IEEE International Conference on Robotics and Automation*, May 2014.

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

[11] A. Borowczyk, D.-T. Nguyen, A. Phu-Van Nguyen, D. Q. Nguyen, D. Saussié, and J. Le Ny, "Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle," *Ifac-Papersonline*, vol. 50, no. 1, pp. 10 488–10 494, 2017.

[12] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico, and D. Scaramuzza, "Vision-based autonomous quadrotor landing on a moving platform," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, Oct. 2017, pp. 200–207.

[13] J. Thomas, J. Welde, G. Loianno, K. Daniilidis, and V. Kumar, "Autonomous Flight for Detection, Localization, and Tracking of Moving Targets With a Small Quadrotor," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1762–1769, Jul. 2017.

[14] R. Spica, A. Franchi, G. Oriolo, H. H. Blthoff, 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.

[15] K. Mohta, V. Kumar, and K. Daniilidis, "Vision-based control of a quadrotor for perching on lines," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014.

[16] A. Breitenmoser, L. Kneip, and R. Siegwart, "A monocular vision-based system for 6D relative robot localization," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011.

[17] M. Cutler, B. Michini, and J. P. How, "Lightweight Infrared Sensing for Relative Navigation of Quadrotors," in *International Conference on Unmanned Aerial Systems (ICUAS)*, 2013, pp. 1156–1164.

[18] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*.

[19] D. Saldaa, B. Gabrich, G. Li, M. Yim, and V. Kumar, "ModQuad: The Flying Modular Structure that Self-Assembles in Midair," in *2018 IEEE International Conference on Robotics and Automation*, May 2018.

[20] M. Wang, "Systems and methods for UAV battery exchange," Patent, May, 2016, US Patent 9,346,560.

[21] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 15–22.

[22] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

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

[24] J. Ferrin, R. Leishman, R. Beard, and T. McLain, "Differential flatness based control of a rotorcraft for aggressive maneuvers," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 2688–2693.