Multirotor Docking with an Airborne Platform

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

[1]Department of Computer Science & Engg., University of Nebraska-Lincoln, USA.
[2]Department of Computer Science, University of Virginia, USA.

Abstract. Multirotor systems have traditionally been employed for missions that ensure minimal contact with the objects in their vicinity. However, their agile flight dynamics lets them sense, plan and react rapidly, and therefore perform highly dynamic missions. In this work, we push their operational envelope further by developing a complete framework that allows a multirotor to dock with a moving platform. Our approach builds on state-of-the-art and optimal methods for estimating and predicting the state of the moving platform, as well as for generating interception trajectories for the docking multirotor. Through a total of 25 field tests outdoors, we demonstrate the capabilities of our system in docking with a platform moving at different speeds and in various operating conditions. We also evaluate the quality of our system's trajectory following at speeds over 2 m/s to effect docking within 10 s.

1 Motivation and Background

Recent advances in multirotor unmanned aerial systems' (UASs) precision in sensing and control have allowed them to be used more efficiently in a wide array of aerial manipulation tasks. These involve picking up objects, inspecting surfaces, and applying contact forces on other objects in the world [1]. Performing such aerial manipulation of stationary objects in outdoor environments still remains challenging due to the sensing and kino-dynamic constraints of the UAS. The problem becomes even more difficult when considering *non-stationary* target objects outdoors, since (a) errors and ambiguities in the target's relative motion estimates can undermine the advantages of precise maneuvering and control, and, (b) manipulating a moving target often imposes other mission contraints such as the time horizon or the number of attempts to engage successfully.

In this paper, we introduce a framework that allows a follower multirotor to dock with a non-stationary docking platform moving on a predictable (non-evasive) path, akin to that of a recovery [leader] aircraft. Illustrated in Fig. 1, we mimic a towed platform by suspending it over a moving "zipline" system, and effect motions at different speeds outdoors. Our approach minimally equips the docking platform with a fiducial marker and a real-time kinematic (RTK) GPS unit. The follower multirotor fuses the high-rate visual feedback with sporadic GPS measurements to estimate the platform's trajectory over a time horizon, h. A smooth, energy-optimal, end-to-end polynomial trajectory is then planned and refined over time with new measurements to effect docking within this constrained horizon. Our evaluations are conducted outdoors under realistic constraints and operational conditions.

Fig. 1: Snapshots depicting a multirotor docking with a suspended platform outdoors. Our outdoor setup comprises of pulleys interconnected by an actuated "zipline" mechanism to regulate the translational velocity of the platform suspended from it.

We consider docking with a moving platform as a variant of the classic problem of landing on a moving platform, coupled with the mechanical and state estimation challenges of an aerial manipulation task. UASs landings on moving platforms have been demonstrated for slow [2] and fast moving targets [3], as well as using entirely onboard estimation [4]. Aerial manipulation problems have been studied within certain applications such as payload transportation and applying contact forces on objects [1].

Unlike landing, in-flight docking necessitates a specialized mechanism to engage with the moving platform. This is akin to robotic manipulators and end-effectors that extend outward from the UAS's center of mass to safely grasp another object [5, 6]. However, for docking, the target object (the docking platform) will typically have limited surface area as it is either a part of the vehicle or an extension towed by an aerial vehicle. Consequently, the positioning tolerances in forward-flight are considerably lower for docking (compared to landing, for instance). We note that while guidance principles have been derived and studied for docking fixed-wing aircraft [7, 8], limited work has been shown for docking stationary platforms [9]. Furthermore, the leader-follower approaches mentioned prior typically employ reactive control strategies that do not accommodate trajectory plans. To the best of our knowledge, there is no previous work demonstrating planned docking for multirotors and moving platforms.

Problem Statement. In a local (fixed) world (Fig.2), $\mathcal{W}$, given a docking platform $\mathcal{D}$ at position $D(t_i)^{\mathcal{W}}$ with a true velocity vector $\mathbf{D}$, a follower multirotor $\mathcal{F}$ at position $F(t_i)^{\mathcal{W}}$, and a time horizon h, the problem is to generate and execute a valid trajectory, $\mathbb{T}(t)$ such that $dock(\mathcal{D}, \mathcal{F}, t) = true$ and $t \in [t_i, t_i + h]$.

Solution Space. We aim to develop a complete framework that enables the follower multirotor UAS to precisely estimate the trajectory of a moving docking platform, and continually refine safe trajectories to dock with it. We also seek cost-effective solutions by integrating off-the-shelf subsystems that can be mounted on a small multirotor UAS.

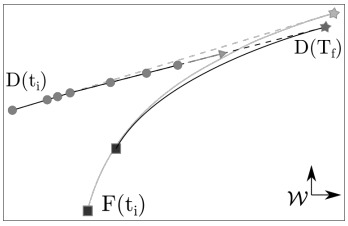

Fig. 2: A top-down graphical illustration of docking: follower UAS at F plans an interception trajectory towards $D(T_f)$, the projected final location for the leader.

2 Technical Approach

We first describe the sensing, planning, and control components needed to meet our goal of state estimation and generating interception trajectories, and then provide a description of the physical mechanism that enables the docking to complete the mission.

A mission is successful when the UAS autonomously docks with the moving platform and disengages its flight controller within the specified time horizon. In the following, let $\mathcal{W}$ denote the origin of the world-fixed (local map) frame in a North-East-Down (NED) convention, and let $F^{\mathcal{W}}, D^{\mathcal{W}} \in \mathbb{R}^3$ respectively denote the origins of the NED frames of the follower UAS ($\mathcal{F}$) and the docking platform ($\mathcal{D}$) expressed in the fixed frame. The first objective is to accurately estimate the pose of $\mathcal{D}$ in the fixed frame for some final time, T_f.

2.1 Sensing and Estimation: Leader

We equip the docking platform with two complementary sources - a real-time kinematic (RTK) GPS unit and a passive fiducial marker. Using the constant velocity assumption, a straight line trajectory is estimated for the platform using the observed positions such that, starting with an initial measurement of the platform's position, $D(t_i)$, we can express its position at some time t as

$$D(t) = D(t_i) + t\mathbf{D}, \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^3$ is the NED velocity of the platform expressed in world frame. To estimate $\mathbf{D}$, we stack k such observations, so that,

$$\begin{pmatrix} D(t_1) - D(t_i) \\ D(t_2) - D(t_i) \\ \vdots \\ D(t_k) - D(t_i) \end{pmatrix} = \begin{pmatrix} t_1\mathbb{I}_{3x3} \\ t_2\mathbb{I}_{3x3} \\ \vdots \\ t_k\mathbb{I}_{3x3} \end{pmatrix} * \mathbf{D} \tag{2}$$

where $\mathbb{I}$ is the identity matrix. The equations have been rearranged in the form of $b = Ax$, and a closed-form solution can be obtained as $\mathbf{D} \equiv x = (A^\top A)^{-1}A^\top b$. The location of the leader at time T_f can now be calculated from Equation 1. This form of linear regression is resilient to outliers that are often encountered in GPS measurements, and yields robust estimates of the constant $\mathbf{D}$.

In some cases, the velocity of the leader may not be constant, but the resultant path may still be predictable (e.g., parabolic). This is true for an emulated leader setup suspended on a gantry/zipline system, which exhibits a longitudinal sag due to its own weight. For a leader system moving along this path, the resultant trajectory in the vertical axis can be approximated as a parabola in time. We can extend the same approach to define a polynomial regression over the observed path points $D_{\hat{\imath}}$ along any axis $\hat{\imath}$. As before, k such observations can be stacked, and the set of parabolic equations can be rewritten as

$$\begin{pmatrix} D_{\hat{\imath}}(t_1) \\ D_{\hat{\imath}}(t_2) \\ \vdots \\ D_{\hat{\imath}}(t_k) \end{pmatrix} = \begin{pmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ & \vdots & \\ 1 & t_k & t_k^2 \end{pmatrix} * \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}, \tag{3}$$

where $[c_1, c_2, c_3]^\top$ are the coefficients of a parabola. Once again, the system of equations yield a closed-form solution for $[c_1, c_2, c_3]^\top$ by recognizing the form to be $b = Ax$ and solving $(A^\top A)^{-1}A^\top b$ in this case. The location and velocity in the $\hat{\imath}$ axis at the final time, T_f, can then be expressed as

$$D_i(T_f) = c_1 + c_2 T_f + c_3 T_f^2, \text{ and, } \dot{D}_i(T_f) = c_2 + 2c_3 T_f. \tag{4}$$

In this case, the $\hat{\imath}$ component of $\mathbf{D}$ is simply substituted with $\dot{D}_i(T_f)$ instead.

For dynamically adjusting to the changes in the platform's state, especially upon closer approach, we require fast and accurate observations from the calibrated monocular camera. Using a fiducial marker, the camera-frame position of the platform, $z_{\text{cam}} \equiv D(t)^{\mathcal{C}} \in \mathbb{R}^3$, is obtained directly through a fast and open-source localization framework [10]. We can then express the position in world-fixed frame as $D(t)^{\mathcal{W}} = R^{\mathcal{W}}_{\mathcal{F}} R^{\mathcal{F}}_{\mathcal{C}} D(t)^{\mathcal{C}}$, where the general notation R^A_B denotes the transform from a frame B to frame A. The final location of the platform at time T_f is computed using Eq (1).

These complementary sources of information are fused using a first-order Kalman filter, with two measurement update cycles for z_{cam} and z_{gps}. We treat the low-rate RTK-GPS measurements as ground-truth (with respect to $\mathcal{W}$ frame), and thus associate a lower measurement uncertainty to it.

2.2 Sensing and Estimation: Follower

The follower UAS is modeled as a six degree of freedom (DOF) object in a feedback-linearized system of equations. The UAS state vector, denoted as $\boldsymbol{x} \equiv [p_n, p_e, p_d, v_n, v_e, v_d, \psi]^\top$, is comprised of position ($p_{(\cdot)}$) and velocity ($v_{(\cdot)}$) elements in North, East and Down axes, and the heading (ψ). The 2nd-order dynamic system of equations can then be represented as

$$\begin{aligned} \dot{\boldsymbol{x}} &= A_F \boldsymbol{x} + B_F \boldsymbol{u} \\ y &= C_F \boldsymbol{x}. \end{aligned} \quad \Big| \quad A_F = \begin{pmatrix} 0_{3x3} & \mathbb{I}_{3x3} & 0_{3x1} \\ 0_{3x3} & 0_{3x3} & 0_{3x1} \\ 0_{1x3} & 0_{1x3} & 0 \end{pmatrix} B_F = \begin{pmatrix} 0_{3x3} & 0_{3x1} \\ \mathbb{I}_{3x3} & 0_{3x1} \\ 0_{1x3} & 1 \end{pmatrix} C_F = \mathbb{I}_{7x7}. \tag{5}$$

An extended Kalman filter (EKF) is used to produce the best estimate of the system state, $\hat{\boldsymbol{x}}$, by fusing low rate RTK-GPS and high rate attitude measurements from the onboard inertial measurement unit (IMU). The controller uses this $\hat{\boldsymbol{x}}$ to regulate $\boldsymbol{x}$ along the trajectory for the docking mission.

2.3 Trajectory Planning and Control

A trajectory for the UAS is generated at time t_i, such that it begins at $F(t_i)^{\mathcal{W}}$ and terminates at $D(T_f)$, and is smooth up to the 3rd order (to minimize drastic changes in acceleration). For a smooth terminal contact, we also require that the trajectory is parameterized in time to account for velocity constraints, such that $\dot{F}(T_f) = \mathbf{D} + \mathbf{v}_\epsilon$, where $\mathbf{v}_\epsilon \in \mathbb{R}^3$ is a small constant. The choice of $\mathbf{v}_\epsilon$ is a design parameter that allows the follower to use the transfer of momentum from the contact to trigger a grasping mechanism.

To meet these criteria, we use closed-form representations of energy optimal trajectories that can be computed algebraically [11]. Thus, given a starting state $F(t_i)$ and $\dot{F}(t_i)$, a trajectory $\mathbb{T}(t)$ is defined as a ordered collection of 3D points in time that minimize the total jerk over time,

$$\mathbb{T}(t) = \big\{F(t) \;\big|\; \min \int_{t_i}^{T} \|\frac{\mathrm{d}^3\mathbb{T}}{\mathrm{d}t^3}\|^2 \mathrm{dt}\big\}, \qquad t_i < t \leq T_f, \tag{6}$$

such that $\mathbb{T}(T_f) = D(T_f)$ and $\frac{\mathrm{d}\mathbb{T}(T_f)}{\mathrm{d}t} = \mathbf{D} + \mathbf{v}_\epsilon$. Since the translational end-states are known (estimated in Section 2.1), the resultant trajectory is fully defined and can be computed through closed-form expressions.

A linear quadratic regulator (LQR) outer-loop controller governs the execution of the generated trajectory. The resultant path from Equation 6 is a 4th-order polynomial in time. We use the path points, and their first derivative (velocity) as reference states for the controller, while the second derivative (acceleration) is used as a feed-forward element. The trajectory is undefined for $t > T_f$, and thus, the controller maintains a stable hover if docking has failed, or is disengaged if successful.

Denoting $\boldsymbol{x}_r = [\mathbb{T}(t), \dot{\mathbb{T}}(t)]^\top$ as the reference state for the controller, we compute the control input to the system as the sum of feedback and feedforward elements as,

$$\boldsymbol{u} = -K_{\mathrm{lqr}}(\hat{\boldsymbol{x}} - \boldsymbol{x}_r) + \ddot{\mathbb{T}}(t), \tag{7}$$

where K_{lqr} is the LQR feedback gain matrix. The control input, $\boldsymbol{u}$, in this case consists of acceleration commands in the three translational axes. These are converted to angle and thrust commands by a non-linear inversion map, such that, $[\phi_d, \theta_d, \psi_d, T_d] = g_{\mathrm{kin}}(\boldsymbol{u}, m)$, with m denoting the total mass of the system, and $(\cdot)_d$ denoting a desired target.

2.4 Docking Subsystem

A distinctive element in the docking problem (compared to landing, for instance) is the mechanical subsystem that ensures a physical lock. The mechanism must sense the contact with the docking platform and react rapidly to ensure a successful "grasp" of the docking platform. Since the relative velocity is non-zero at the time of contact (due to $\mathbf{v}_\epsilon$), the mechanism can utilize the transfer of momentum as a sensory input.

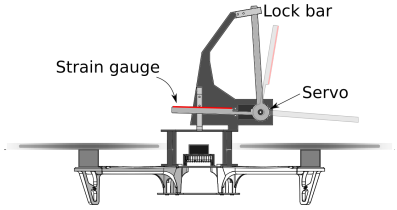

Fig. 3: A CAD schematic for the docking assembly atop the UAS.

Figure 3 shows a side-view illustration of the docking mechanism. The servo-actuated rotary "gate" is mounted on top of the UAS such that a strain gauge affixed to one of the gate arms is exposed (shown as faded). A microcontroller senses the calibrated output of the strain gauge, and actuates the gate to a "closed" position when a threshold for contact force is met. Using a fast-actuation servo, we minimize the reaction time to an external contact down to 0.2 s.

3 Field Studies

We begin by first outlining the details of our implementation setup, including the various off-the-shelf components utilized for a fully characterizable set of tests. We then present our results from outdoor field studies in two parts: first, for a stationary leader, and then, for a leader moving in linear trajectory at different velocities. Our evaluations focus on the ability of the system to (a) correctly estimate the location of the docking platform and (re)plan trajectories towards it, (b) regulate the distance to the final predicted location in a smooth and predictable fashion within the specified time horizon, and finally, (c) dock with the platform.

Implementation Details

Leader System. As mentioned earlier, we utilize a zipline-like mechanism to emulate a leader aircraft to aid consistent and repeatable missions. The zipline consists of two pulleys mounted rigidly on poles approximately 17 m apart. A light-weight string loops around the pulleys and runs through a DC motor system (driver, battery and a micro-controller). The motor can affect different translational speeds for the zipline system through a serial command interface. A 3D-printed clip and frame are affixed on the string to suspend the docking platform on it. The docking platform is a metal bar held horizontal by two strings in an 'A'-frame fashion. The bar is a hollow 20 mm square extruded to a length of 0.5 m. The total available contact surface spans approximately 0.4 m. A fiducial marker is placed separated 0.2 m from the docking platform. The RTK GPS receiver (uBlox ZED-F9P) is also mounted on this A-frame, exposed to the sky.

The platform exhibits translational velocities due to the motion of the zipline. These are constant in time in the lateral (north-east) plane, resulting in a linear trajectory. Since the strings are flexible, the weight of the platform causes the zipline to sag in the vertical (down) axis, resulting in a curved path that can be approximated as parabolic. The platform also exhibits 3-axis rotational velocities due to wind and noise from the standing waves induced in the zipline's strings. These are only naturally damped by the material properties of the strings.

Follower System. We use a DJI Flamewheel frame with the open-source Pixhawk autopilot as the follower vehicle. The docking platform described in Section 2.4 is mounted on top of the frame as shown in Figure 3. All of state estimation, prediction, trajectory generation and flight control is implemented entirely onboard on an Odroid XU4 single-board computer. No external motion-capture or ground controllers are utilized.

3.1 Stationary Leader

We first assess the quality of our state estimation, trajectory control and the docking subsystem by keeping the leader system stationary. This primarily allows us to isolate the follower's performance by removing the prediction errors that might arise when the target is in motion. It also lets us validate that the electro-mechanical elements of the docking system function as expected upon contact with the docking platform. For these tests, the zipline system is commanded a zero velocity, while the leader (docking platform) simply suspends from it.

Figure 4 shows the distribution of this stationary leader's estimated velocities as a probability density function. Due to ambient wind, the leader can exhibit a 'bobbing' motion with a small velocity (usually $\leq$0.01 m/s), which can cause erroneous predictions over a long time horizon. It is possible to use Equation (1) with the assumption that the leader remains stationary. However, to avoid over-simplifying the missions, we make no such assumptions, and carry out state estimation, projection, and trajectory (re-)planning without any simplifications. For context, an estimated velocity of 0.01 m/s over a horizon of 8 s produces an error of 0.08 m which is $\sim$ 70% the size of our docking mechanism's gate.

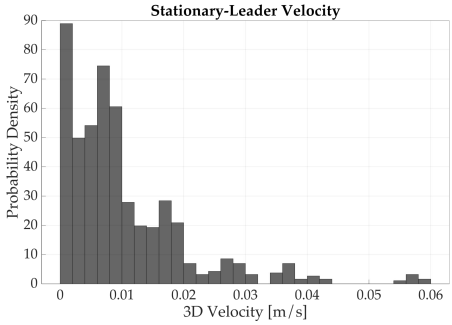

Fig. 4: Probability function of the leader's velocities even when stationary.

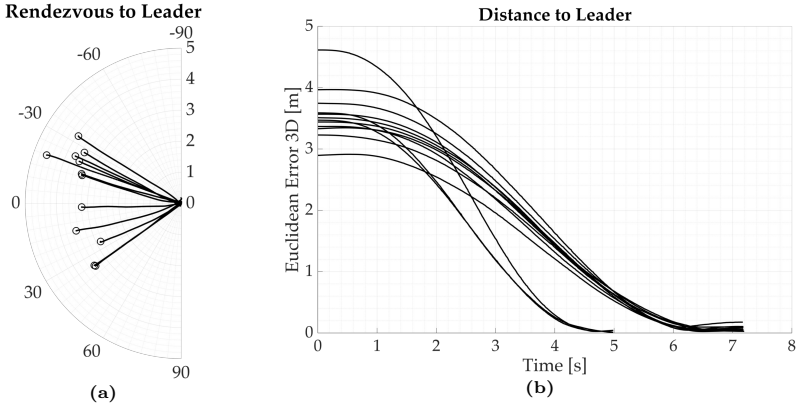

Fig. 5: Docking with a stationary leader, shown as, (a) top-down polar view of all the docking tests, framed to keep the leader at the origin; and (b) temporal evolution of the 3D Euclidean distance to the leader. Each test lead to a successful docking.

Figure 5 graphically summarizes the results from the stationary tests, conducted under ambient wind speeds of 0.75–1.8 m/s. We conduct a total of 12 trials starting at different relative positions between the leader and the follower. These are graphically shown on a polar plot of the lateral axes in Figure 5(a), rotated and re-centered to keep the leader at the origin. For 9 of these, the fixed time-horizon is set to 7 s. To stress the system further, for the final 3 tests, we shrink this horizon to 5 s. In all cases, the controller is able to regulate the UAS's position exactly within these selected time windows and effect a successful dock-

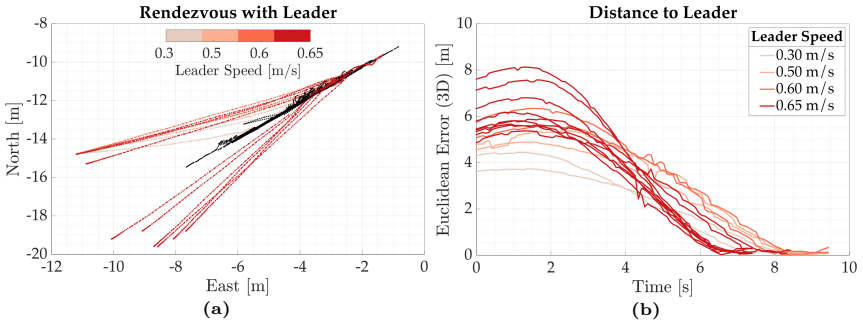

Fig. 6: Docking with a moving leader. (a) The top-down (North-East) view of trajectories (leader moves on the same path at different speeds). The initial locations are on the lower-left of the image, and time evolves from left to right. The follower's trajectories are color-coded for the 4 different speeds of the leader. (b) Temporal evolutions of the 3D Euclidean distance to the leader. Docking occurs between 7–9 s.

ing. In addition, one of the 5 s missions also begins at the largest separation to the leader, thus creating a challenging mission. This particular trial exhibits a peak velocity of almost 2 m/s, and still leads to a successful docking.

3.2 Moving Leader

We now present our evaluations from 15 tests with the leader system in motion: 8 successive tests at 4 different translational speeds, and then 7 additional tests with different initial conditions. For the speed tests, we select a common starting location for the follower. It maintains a stable hover at this location as the zipline begins to move at the commanded speed. The four speeds we test at are 0.3, 0.5, 0.6 and 0.65 m/s. We set $t_{\text{obs}} = 10$ s for all of these tests, and choose either 8.5 s or 9.5 s as the mission horizon. These numbers are chosen empirically based on the operational size of our test arena and the speed of the zipline. For instance, approximately 20 s of motion at 0.65 m/s would move the zipline from one extremity to another.

In Figure 6(a), we show a top-down (North-East) view of all these trials. The black trajectory is that of the leader/zipline, which moves on the same path (starting at lower-left) but at the different speeds we listed above. The follower UAS begins its mission at the common location at the lower-left, and the trajectories for the four leader speeds are shown with different colors. We see all the planned paths meet the leader's path correctly. As expected, longer paths are generated for a faster moving leader, since its predicted location at T_f will be farther away.

Figure 6(b) also shows the trajectory of the follower's distance errors to the leader's location. These are noisier than the corresponding error trajectories for docking with a stationary leader (in Figure 5). We note that the distance to the leader is bridged in a consistent and repeatable fashion in spite of the higher

noise. The peak velocity attained by the follower in these tests can exceed 2 m/s for the fastest moving leader.

Table 1 summarizes the key statistics from these tests. The leader's actual velocity, as well as the average and the peak velocity of the follower UAS during each mission are listed therein. Additionally, we also list the 3D error between the leader's final predicted location at $t = T_f$ and its actual location at $t = T_f$. We see that this error is typically less than 0.05 m. However, this can be occasionally much higher (for instance, 0.24 m in Exp.4). The mission still results in a successful docking since the majority of the error is situated in the lateral axes, which have a larger margin because of the width of the docking platform.

We continue at the highest speed of the leader (∼0.65 m/s) and assess the repeatability the docking missions through 7 more tests. For these, the follower is initialized at different locations and the leader moves at a constant speed over the same path. The time horizon is further reduced to 7.5 s to expose the system to a more challenging operational constraint. The resultant tracks are shown in Figure 6(a) corresponding the leader velocity of 0.65 m/s. In Figure 6(b), the corresponding evolutions of the 3D Euclidean distances to the leader can be visualized – these are the tracks that complete docking around the 7 s mark. The docking is successful in 6 out of these 7 tests. In the one failed run, the final state error is approximately 4 cm outside the tolerance, and the docking mechanism never makes contact with the platform.

Figure 7 also shows key evaluations of our system's trajectory following performance in docking with a moving target. On the left, we show the commanded and the executed trajectories in each axes for one sample mission. We see that the trajectory of the leader (target) exhibits a parabolic curvature in the vertical plane (Down), while it is linear in the lateral plane (North and East). The follower UAS begins its mission at around the 28 s mark, and successfully docks around 36.5 s. The planned and the executed trajectories are in close agreement with each other (note that the magnification on Down axis is nearly 20x).

Table 1: Summary of statistics from outdoor (moving) tests.

Exp.	Leader Vel (m/s)	Obsv (s)	Mission (s)	Pred. Err 3D (m)	UAS Err Final (m)	UAS Vel (m/s) Avg, Max	*dock*?
1	0.31	10	8.50	0.013	0.24	0.78, 1.25	*true*
2	0.30	10	9.50	0.024	0.08	0.60, 1.00	*true*
3	0.50	10	9.50	0.057	0.15	0.93, 1.67	*true*
4	0.48	10	9.50	0.049	0.24	0.90, 1.42	*true*
5	0.58	10	9.50	0.050	0.08	1.15, 1.97	*true*
6	0.57	10	9.50	0.121	0.29	1.02, 1.72	*true*
7	0.65	10	8.50	0.244	0.18	1.14, 2.06	*true*
8	0.64	10	8.50	0.050	0.15	1.17, 2.13	*true*
9-15	0.65	10	7.50	0.099 (median)	0.13 (median)	1.31, 2.35 (median)	6**true* 1**false*

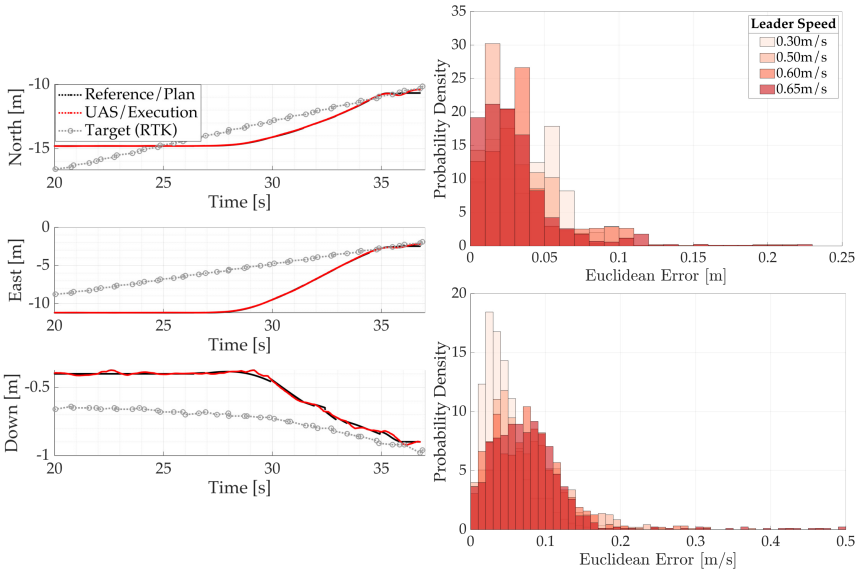

Fig. 7: Trajectory following performance represented as [left] temporal graphs for each axis, and [right] probability histograms of position and velocity tracking errors for different leader speeds. The errors are measured as $||\boldsymbol{x} - \boldsymbol{x}_r||_2$.

The tracking errors in position and velocity references for all missions are also assimilated into two histograms on the right. We see that the majority of the position errors lie within 0.05 m, while the velocity errors are usually under 0.1 m/s. We also note that velocity errors are higher when the leader is moving at a higher speed. The peak for the velocity error for a slower mission is around 0.05 m/s. These statistics collectively indicate that our controller is able to regulate the position of the follower UAS sufficiently for a docking mission.

To dock with a moving leader, an accurate estimate of its position and velocity at the specified time horizon is necessary. This is prone to measurement noise, since the velocity must be inferred by the follower UAS using its own observations. Figure 8 shows this error in the leader's estimated position at $t = T_f$ compared to its true position at $t = T_f$ for one sample mission (Exp.6 in Table 1). The estimates are obtained using Eqns. (1) - (3) for a time horizon of 9.5 s and $k = 5$. The figure shows that the er-

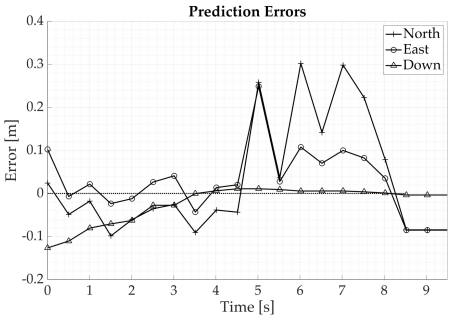

Fig. 8: The error in a 10s-horizon prediction for the final location of a platform at 0.5 m/s.

ror rapidly diminishes within 4–5 s, and then increases dramatically as noisier observations are introduced. Towards the end of the mission (at closer proximity), the error diminishes again, such that the final 3D error, as reported in Table 1 is approximately 0.12 m. Notice that since the North and East components contribute most to this final error, the system can still dock successfully.

4 Discussion and Future Work

Our outdoor tests with a stationary and moving leader have shown consistent and repeatable docking with an emulated leader system. These results indicate that we are able to achieve the accuracy in estimation and control necessary for such missions even under realistic operating constraints (such as the relative sizes of the docking platform and docking subsystem).

We observe in Figure 6(a) that a significant amount of noise can corrupt the leader's position estimate during certain intervals. These are due to the combined effect of ambient wind, and disagreements between camera and RTK-GPS measurements. The disagreements can manifest themselves due to various factors, such as latency in GPS measurements, calibration errors on the camera, and the different motions exhibited by the fiducial marker and the RTK antenna mounted on the trapeze.

In our current implementation, we require the observation and prediction horizons be specified at the start of a mission. In practice, however, these values must be automatically inferred. For instance, the system can continue to collect observations until the residual of new data (or alternatively, the covariance) falls within a certain threshold. Similarly, one strategy to pick the mission duration automatically is to choose the shortest duration that leads to feasible trajectories. Our future work will investigate these methods. We also plan to expand the operational range of our setup to allow higher speeds of the platform, carried by another unmanned system.

Acknowledgments

This work was supported in part by NSF-IIS-1925052 -1924777, IIS-1638099, IIS-1925368, and USDA-NIFA 2017-67021-25924. Thanks to the members of the Nimbus Lab (Paul Fletcher, Ji Young Lee and Daniel Rico) for assisting with the field tests, and Jacob Hogberg (Research Engineer) for his design contributions to the docking subsystem.

References

1. Ruggiero, F., Lippiello, V., Ollero, A.: Aerial Manipulation: A Literature Review. IEEE Robotics and Automation Letters 3(3), 1957–1964 (Jul 2018)
2. Saripalli, S., Sukhatme, G.: Landing on a moving target using an autonomous helicopter. In: Field and service robotics. pp. 277–286. Springer (2006)
3. Borowczyk, A., Nguyen, D.T., Nguyen, A.P.V., Nguyen, D.Q., Saussi, D., Ny, J.L.: Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle. CoRR abs/1611.07329 (2016), http://arxiv.org/abs/1611.07329
4. Falanga, D., Zanchettin, A., Simovic, A., Delmerico, J., Scaramuzza, D.: Vision-based autonomous quadrotor landing on a moving platform. In: 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR). pp. 200–207

5. Kim, S., Choi, S., Kim, H.J.: Aerial manipulation using a quadrotor with a two DOF robotic arm. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (Nov 2013)
6. Jimenez-Cano, A., Martin, J., Heredia, G., Ollero, A., Cano, R.: Control of an aerial robot with multi-link arm for assembly tasks. In: Robotics and Automation (ICRA), 2013 IEEE International Conference on. pp. 4916–4921. IEEE (2013)
7. Tandale, M.D., Bowers, R., Valasek, J.: Trajectory Tracking Controller for Vision-Based Probe and Drogue Autonomous Aerial Refueling. Journal of Guidance, Control, and Dynamics 29(4), 846–857 (2006), https://doi.org/10.2514/1.19694
8. Wilson, D.B., Gktogan, A., Sukkarieh, S.: Guidance and Navigation for UAV Airborne Docking. In: Robotics: Science and Systems (2015)
9. Miyazaki, R., Jiang, R., Paul, H., Ono, K., Shimonomura, K.: Airborne Docking for Multi-Rotor Aerial Manipulations. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4708–4714 (Oct 2018)
10. Krajnk, T., Nitsche, M., Faigl, J., Vank, P., Saska, M., Peuil, L., Duckett, T., Mejail, M.: A Practical Multirobot Localization System. Journal of Intelligent & Robotic Systems 76(3), 539–562 (Dec 2014), https://doi.org/10.1007/s10846-014-0041-x
11. Mueller, M.W., Hehn, M., D'Andrea, R.: A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation. IEEE Transactions on Robotics 31(6), 1294–1310 (Dec 2015)