Dynamic Path Generation for Multirotor Aerial Docking in Forward Flight

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

***Abstract*— In-flight docking between unmanned aerial systems (UASs) is an essential capability for extending collaborative long-range missions. This work presents a planning strategy for a smaller multirotor UAS to autonomously dock with a non-stationary carrier/leader UAS in forward flight. Our method assumes the leader aircraft to be another multirotor, and first projects the hypotheses for its pose forward in time. Using a multi-objective cost function, we then solve an optimal control problem to obtain an interception trajectory to all these possible locations. We employ a cost formulation that allows us to generate piecewise smooth curves that favor different objectives during the course of the mission. Through a greedy strategy, the paths are iteratively refined online as the prediction is improved with new observations. We demonstrate and evaluate our method through a series of physics-based simulations with different operating conditions for both vehicles.**

I. Introduction

Aerial docking of unmanned aerial systems (UAS) can enable longer missions through mid-air vehicle refueling [1], faster missions through aerial payload transfer [2], and larger coverage of simultaneous events by having a large carrier vehicle that drops and then retrieves smaller ones. We envision a system where a large UAS or manned aircraft can carry multiple smaller multi-rotors and deploy them in remote locations and then recover them again for the return trip. Using this type of transportation overcomes the multi-rotor's inherent energy limitations while leveraging their agility once deployed. Deploying multi-rotors from larger aircraft has been previously demonstrated [3]. Docking, however, poses additional challenges and most existing techniques for multi-rotor docking make the assumption that the leading vehicle is stationary [4]. While this can work in some situations, it limits the mobility of the carrier (e.g. it cannot be a fixed wing aircraft).

In this paper, we address the challenge of docking a multi-rotor UAS with a *moving* carrier aircraft. We take into account uncertainty in the state estimation of the relative positions of the UASs. Figure 1 shows a depiction of our approach. At each time step, the multi-rotor UAS trying to dock, called "Follower," estimates the relative position and trajectory of the "Leader." For each position and trajectory hypothesis, the follower calculates an optimal trajectory to intersect the leader. Throughout this process, to successfully dock, the follower must: 1) estimate the current position and trajectory of the leader, 2) maintain a safe distance for as long as possible (e.g. to avoid the propeller wash from the leader until the final docking stage), 3) align its relative final velocity with the leader, 4) only execute trajectories that remain feasible at all times (e.g. ones that do not saturate the control), and 5) reserve alternative trajectories that account for the uncertainties in the leader's pose.

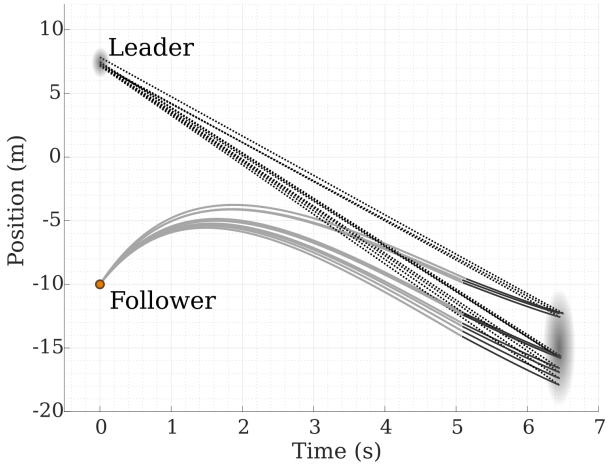

Fig. 1: A snapshot depicting multiple hypotheses for the leader UAS's trajectory (moving at $4\,\mathrm{m/s}$), and the follower's two-part optimal paths for docking developed in this work.

In this work, we generate optimal interception paths using a weighted multi-objective cost function that incorporates the above requirements for aerial docking of moving multi-rotor vehicles under uncertainty. Using a known motion model for the leader, we employ a particle filter-based approach to represent the multiple hypotheses for the leader's relative position and trajectories. Using analytical methods from optimal control, for each hypothesis, we compute the closed-form expression to find the extremal path in $\mathbb{R}^3$. We exploit the fact that the multirotor dynamics are differentially flat, and obtain solutions to this minimization problem in the flat output space. Finally, the planned trajectory is used as the feedforward component for a linear quadratic regulator (LQR) control, coupled with a non-linear mapping to the UAS's input space to execute the generated plan.

The key contributions of this work are:

- An optimal-path generation strategy for multirotor docking that consolidates the operating constraints of the mission into one single unified cost function;
- A greedy path selection algorithm that considers path constraints, multiple hypotheses, and uncertainty in the states of the carrier UAS; and,
- Exemplary evaluations through various high-fidelity simulations that demonstrate our algorithmic approach.

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

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

This work was supported in part by grants: NSF IIS-1925052, IIS-1638099, 1924777 and USDA-NIFA 2017-67021-25924.

II. Related Work

Airborne docking for fixed-wing UASs has been studied in the past under various scenarios, typically using a drogue mechanism. It has been used and demonstrated for refueling smaller aircraft through a larger "mothership" vehicle. Mathematical models for a passive towed-cable system, coupled with guidance laws that drive the chasing UAS to make contact with the drogue have shown some successful attempts [5], [6]. A simpler leader-follower scheme has also been shown to achieve repeated successful docking by Wilson et al. [1]. However, multirotor docking and undocking has only recently seen practical developments. While this enables several novel long-range exploration missions, it can potentially be more hazardous due to the propeller motion on all sides, and the constraints on flight time. Miyazaki et al. use a carrier aircraft in stationary hover that suspends a bar on a winch mechanism which is grasped by the docking UAS [4]. The authors report a low success rate due to environmental disturbances, but since the approach uses a reactive control strategy, it can be difficult to assess guarantees on time-to-dock or the energy cost of the mission. More recent work on self-assembling flying structures of UASs [7] has focused on the dynamics and control of different hybrid configurations, but circumvents the challenge of docking in non-stationary hover, which is our focus.

Previous work has considered the task of intercepting airborne objects and recovering a free-drifting parachute in the air [8], [2]. The closely related problem of quadrotor landing (on a stationary or moving platform) has also received some attention [9], [10], [11]. However, the terminal criteria for these missions, and therefore the sensing and the terminal control required as part of a successful mission are considerably different. This is, in part, due to the nature and severity of aerodynamic effects between *two* aerial vehicles, which can be harder to capture adequately using mathematical representations. Most approaches for landing/intercepting problems implement a state-machine with a sense-and-react control applied to the UAS. Guidance principles such as pure-pursuit, proportional navigation, and rendezvous guidance have been applied for aerial chasing and interception problems [10], [12]. We are interested in the ability to predict a future location of the target and *plan* a suitable trajectory towards it. In this context, some approaches have utilized a fixed-horizon prediction combined with a trajectory planner to follow and land on a platform [13], [14]. However, we re-emphasize that the landing problem has different terminal criteria and mission constraints, and as such, these approaches cannot be used directly for our docking problem.

Planning trajectories that satisfy certain criteria and minimize some objective have also been studied extensively. The approach presented in this paper is closely related to trajectory optimization methods that consider problems with custom cost functions [15], [16], [17], [18]. Chaumette *et al.* [15], [16] have presented an approach that utilizes a multi-objective cost function that guides a UAS towards a visual target while ensuring its visibility. Similarly, a probabilistic framework to plan trajectories that maximize the visibility of multiple targets simultaneously has also been explored [18]. Several of these approaches build on recursive numerical optimization methods, which can be slow and often suffer from ill-conditioned approximations. In our approach, we use analytical methods from optimal control that allow us to rapidly obtain closed-form expressions for continuous paths. In this regard, our approach bears most in common with the optimal control methods explored by Geisert & Mansard [17]. Two significant distinctions in this work are in the way we define a multi-objective cost function suitable for rapid in-air interactions, and our handling of the uncertainty in relative states. Since we generate a full plan for the docking mission, it is possible to assess the cost of the mission and the potential for failure.

III. Technical Approach

Our objective is to develop a trajectory planning strategy that enables aerial docking between two moving multi-rotor UASs in flight. We split our technical approach into three main subsystems: a trajectory generation framework, an iterative trajectory selection method, and finally, a control strategy that guides the UAS along the path. Figure 2 shows a graphical representation of our system architecture. The motion model of the leader aircraft is a fundamental assumption we make in the remainder of the paper, and is a part of our problem definition. In Section III-B we will describe our method to generate an optimal trajectory for each hypothesis of the leader's position. We will then describe our selection strategy amongst these hypotheses in Section III-C. And finally, in Section III-D we outline how the selected trajectory is used as a feedforward for the controller that guides the UAS along the path. We now elaborate each of these components.

A. Problem Formulation

In the following subsections, the leader vehicle's trajectory in the world fixed frame, $\vec{L}^{\mathcal{W}} \in \mathbb{R}^3$, is a straight line in space, such that $\vec{L}^{\mathcal{W}} = t * \dot{\mathbf{L}} + \mathbf{L}^{\mathcal{W}}(0)$, where t denotes time, $\mathbf{L}(0)^{\mathcal{W}}$ is the initial position, and $\dot{\mathbf{L}}$ is its constant velocity. It is assumed that the follower multirotor, with position denoted by $F(t)^{\mathcal{W}}$, can make measurements of the leader's position and velocity (using perspective geometry with a camera, or direct communication from the leader), and register them in the same world frame, $\mathcal{W}$. The values of $\mathbf{L}(0)$ and $\dot{\mathbf{L}}$ may only be known with some uncertainty (possibly non-Gaussian). For a tractable solution, we require that $\dot{\mathbf{L}} < \dot{F}^{max}$, the maximum velocity of the follower UAS.

The problem statement is to find and dynamically update a trajectory starting at $F(0)$ and ending at $\mathbf{L}(t_f)$ at some final time, t_f, such that $\dot{F}(t_f) = \dot{\mathbf{L}}(t_f)$. A trajectory must be continuous and kinematically feasible for the multirotor to follow, and preferably smooth (to minimize aggressive changes in accelerations). We let t_f (and consequently $\mathbf{L}(t_f)$) as a free variable, and therefore, the trajectory generation will be formulated as an optimization problem. Furthermore, because of the uncertainty in $\mathbf{L}(t_f)$, the approach must lend

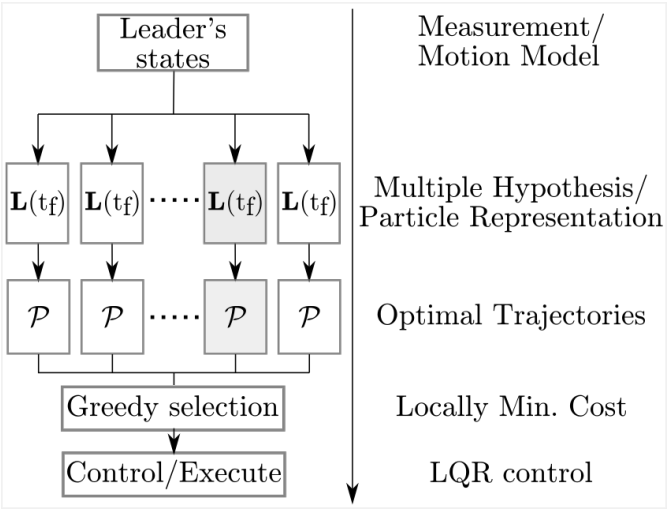

Fig. 2: A flowchart representation of the system architecture presented in this work.

itself to the selection of alternative trajectories as the estimate of $\mathbf{L}(t_f)$ converges to its true value. Maintaining a set of alternative paths is also useful in cases where obstacles may be discovered dynamically.

Although not discussed explicitly, we assume that the leader and the follower have a mechanical system in place that enables physical docking. In practice, this system will be separated from the centers of mass of the two vehicles (for instance, by using a tether/drogue on the leader). The terminal location of interest for the follower, then, is this offset point on the "docking station". However, we will use $\mathbf{L}(t)$ in our approach for the sake of notational convenience; this can readily be altered to mean the physical location of the docking station if needed. We will drop the superscript $\mathcal{W}$ throughout the rest of the paper, and assume that all states are measured in the world-fixed frame.

B. Trajectory Generation

We will first develop a method to generate an optimal trajectory towards $\mathbf{L}(t_f)$ for some final time t_f, and then extend this approach for multiple hypotheses of $\mathbf{L}(t_f)$. Here we consider the case where the leader's trajectory, $\vec{L}$, and its location, $\mathbf{L}(t_0)$, at the current time t_0 are known precisely. We show the case for a single-dimension for ease of notation, however, the same results hold for $\mathbb{R}^3$. For problems involving aerial interactions, we develop a multi-part approach that operates in at least two operation "phases". In the first phase, the follower must align its velocity with the leader's velocity, and bridge their relative distance while maintaining a safe separation between them. In general, it is desirable to permit longer mission times for this phase since the vehicles are not in close proximity to each other. The actual physical contact is made in the second phase, and consequently, this phase should aim to minimize its duration. Aligning velocities is still favorable in this phase, although for physical docking, the follower needs a higher velocity relative to the leader.

We formulate trajectory generation as an optimization problem, and obtain a family of piecewise smooth curves that lead to docking. An optimal trajectory is defined as a path, $\mathbf{x}^*$, that minimizes the weighted multi-objective functional

$$\min J(\mathbf{x}) = \int_0^{t_i} g\,\mathrm{dt} + \int_{t_i}^{t_f} h\,\mathrm{dt} \tag{1}$$

where,

$$g = w_1 + (\dot{\mathbf{x}} - \dot{\mathbf{L}})^2 + (\mathbf{x} - (\mathbf{L} - R_s))^2 \tag{2}$$

$$h = w_2 + (\dot{\mathbf{x}} - \dot{\mathbf{L}})^2 \tag{3}$$

subject to,

$$\begin{aligned} \mathbf{x}(0) &= F(0) \\ \mathbf{x}(t_i) &= \mathbf{L}(t_i) - R_s \\ \mathbf{x}(t_f) &= \mathbf{L}(t_f) \end{aligned} \tag{4}$$

where, w_1 and w_2 are weighting factors that correspond to the time spent in each phase. Both t_i and t_f are free, subject to the split boundary constraints in Equation (4). The functional J is split into two parts corresponding to each of our desired phases. In the first phase, for $0 \leq t < t_i$, the objective function g contains minimization terms corresponding to the time spent, the relative velocity, and the relative position errors (bounded to a safety distance, R_s). In the second phase, for $t_i \leq t \leq t_f$, the objective function takes the form of h and only contains time and relative velocity terms. Note that the constraint $\mathbf{x}(t_f) = \mathbf{L}(t_f)$ forces the solution to intersect $\vec{L}$. A solution that minimizes J will be continous, but only piecewise smooth (i.e., it will have a corner at the intermediate time t_i). We will later show that, since the full solution is known in advance, a trajectory feedforward controller can generate the acceleration required to accomplish the phase transition. Finally, since the factors w_1 and w_2 represent the time-cost of each phase, they influence the aggressiveness of the path.

A solution, $\mathbf{x}^*$, to the optimization problem presented in Equations (1)-(4) can be obtained analytically using variational calculus. To do so, we first note that the optimal path must satisfy the *Euler-Lagrange equation*, so that,

$$\Big[\frac{\partial g}{\partial \mathbf{x}} - \frac{\mathrm{d}}{\mathrm{dt}}\Big(\frac{\partial g}{\partial \dot{\mathbf{x}}}\Big)\Big] + \Big[\frac{\partial h}{\partial \mathbf{x}} - \frac{\mathrm{d}}{\mathrm{dt}}\Big(\frac{\partial h}{\partial \dot{\mathbf{x}}}\Big)\Big] = 0. \tag{5}$$

Substituting the expressions for g and h, and reducing, we obtain,

$$2\ddot{\mathbf{x}} - \mathbf{x} - R_s = \mathbf{L}(t), \tag{6}$$

which is a second-order non-homogeneous ordinary differential equation. The solution to Equation (6) is composed of a general solution and a particular solution, and will be of the form $\mathbf{x} = \mathbf{x}_g + \mathbf{x}_p$. Since we assume that the trajectory of the leader UAS is linear, the right-hand side of Equation (6) is known to be of the form $\mathbf{L}(t) = t \cdot \dot{\mathbf{L}} + \mathbf{L}(0)$. Therefore, we have that Equation (6) is solved by a family of curves represented by

$$\mathbf{x}^*(t) = C_1 e^{\frac{-t}{\sqrt{2}}} + C_2 e^{\frac{-t}{\sqrt{2}}} + \mathbf{L}(t) - R_s.$$

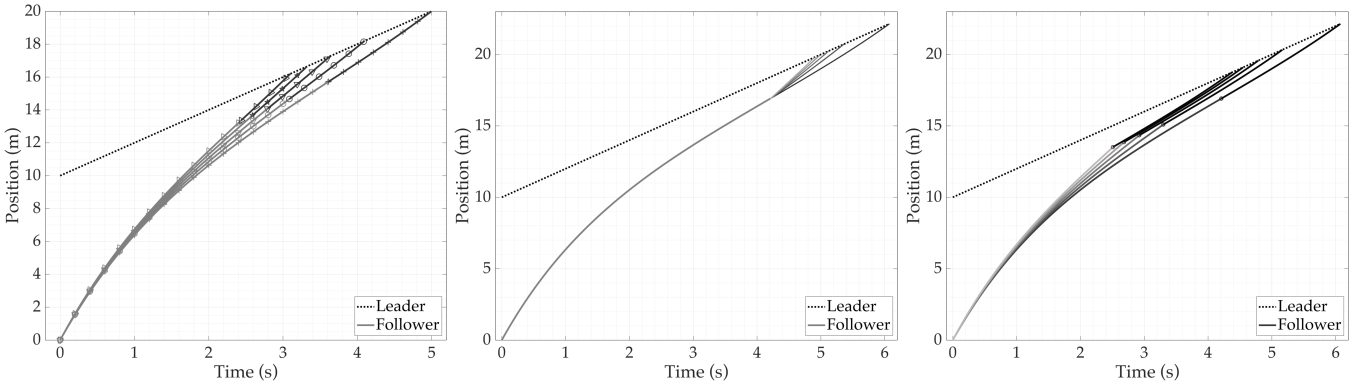

Fig. 3: Three cases illustrating the affect of changing w_1 and w_2 on the path: a) both remain equal and are changed by the same amount, b) w_1 is constant and w_2 is changed, and, c) w_1 is changed while w_2 is held constant. Observe that in (a) we additionally have that $w_1 = w_2$ and so, the resultant paths are smooth.

We split the solution curves for the two phases, such that,

$$\mathbf{x}_1^*(t) = C_1 e^{\frac{-t}{\sqrt{2}}} + C_2 e^{\frac{t}{\sqrt{2}}} + \mathbf{L}(t) - R_s, \;\; t \in [0, t_i) \tag{7}$$

$$\mathbf{x}_2^*(t) = C_3 e^{\frac{-t}{\sqrt{2}}} + C_4 e^{\frac{t}{\sqrt{2}}} + \mathbf{L}(t) - R_s, \;\; t \in [t_i, t_f]. \tag{8}$$

The problem has 4 undetermined constants, and two free variables t_i and t_f. The split boundary conditions for $t = 0, t_i, t_f$ produce four equations,

$$\mathbf{x}(0) = F(0) = C_1 + C_2 + \mathbf{L}(0) + R_s \tag{9}$$

$$\mathbf{x}(t_i) = \mathbf{L}(t_i) - R_s = C_1 e^{\frac{-t_i}{\sqrt{2}}} + C_2 e^{\frac{t_i}{\sqrt{2}}} + \mathbf{L}(t_i) - R_s, \tag{10}$$

$$\mathbf{x}(t_i) = \mathbf{L}(t_i) - R_s = C_3 e^{\frac{-t_i}{\sqrt{2}}} + C_4 e^{\frac{t_i}{\sqrt{2}}} + \mathbf{L}(t_i) - R_s, \tag{11}$$

$$\mathbf{x}(t_f) = \mathbf{L}(t_f) = C_3 e^{\frac{-t_f}{\sqrt{2}}} + C_4 e^{\frac{-t_f}{\sqrt{2}}} + \mathbf{L}(t_f) - R_s, \tag{12}$$

where the first two and the last two equations are generated from Equation (7) and Equation (8), respectively.

The two additional equations required to fully define the system are produced by noticing that, (1) while t_f is free, $\mathbf{x}(t_f)$ is constrained to lie on $\mathbf{L}(t_f)$, and, (2) the two phases must be continous at the corner at time t_i. The first condition, known as the *transversality condition*, asserts that

$$h\big|_{t_f} + \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{t_f} * \big[\frac{d}{\mathrm{dt}}\mathbf{L} - \dot{\mathbf{x}}\big]_{t_f} = 0,$$

which, after substitutions from Equations (3), (8) and (12), can be reduced to

$$R_s + \sqrt{2w_2} = 2C_4 e^{\frac{t_f}{\sqrt{2}}}. \tag{13}$$

The second condition is ascertained by extending the *Weierstrass-Erdmann corner condition*, such that,

$$g\big|_{t_i^-} + \left.\frac{\partial g}{\partial \mathbf{x}}\right|_{t_i^-} * \big[\frac{d}{\mathrm{dt}}\mathbf{L} - \dot{\mathbf{x}}\big]_{t_i^-} = h\big|_{t_i^+} + \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{t_i^+} * \big[\frac{d}{\mathrm{dt}}\mathbf{L} - \dot{\mathbf{x}}\big]_{t_i^+},$$

where t_i^- and t_i^+ denote the infinitesimally small time instances before and after t_i. After substituting Equations (7) and (8), and using Equations (10) and (11) to reduce, we obtain that

$$w_1 - w_2 = 2(C_2^2 - C_4^2)e^{\sqrt{2}t_i}. \tag{14}$$

Equations (9)-(14) represent a set of six non-linear algebraic equations with six unknowns. These can be solved analytically to fully define the family of optimal docking curves defined by Equations (7) and (8). A complete path $\mathcal{P}_{t_f}$ for an optimal final time t_f is the concatenation of $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$,

$$\mathcal{P}_{t_f} = \begin{cases} \mathbf{x}_1^*(t), & 0 \le t < t_i \\ \mathbf{x}_2^*(t), & t_i \le t \le t_f. \end{cases} \tag{15}$$

Figure 1 depicts some sample paths generated for different $\mathbf{L}(t_f)$ using Equation (15). Figure 3 further illustrates the effect of changing the scalars w_1 and w_2. A suitable combination of these is often guided by practical design principles. The values of these scalars can be adjusted to affect different profiles of trajectories, and can also be used as slack if an analytical solution yields complex numbers.

In general, this solution strategy only finds an extremum for J. To show that it indeed minimizes J, we observe that both integrands g and h contain constant scalars for penalizing time. If $\mathcal{P}$ in Equation 15 were a maximizing solution, it would be possible to select a larger t_i (and t_f) to further increase J. Clearly, since the equations include quadratic relationships, certain combinations of variables can lead to undefined results (negative time, complex trajectories etc.). This can be avoided by reselecting w_1 or w_2.

C. Iterative trajectory update

If the pose of the carrier vehicle were known apriori with no uncertainty, then the path generated in the previous step suffices. $\mathcal{P}_{t_f}$ is known to be optimal, continuous and piecewise-smooth, and meets the terminal criteria we outlined for interception. However, in practice, the measurements of the carrier UAS will be noisy, and thus the projected location of the carrier, $\mathbf{L}(t_f)$, will be be prone to accumulated errors. Additionally, a generated path may violate spatial

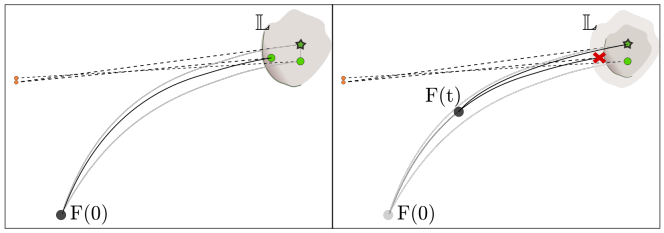

Fig. 4: Illustration describing our path selection process. The shaded region depicts the distribution of the hypotheses for the carrier's final location (the set $\mathbb{L}$), along with a few sample points. A star marks the maximum likelihood location. By greedily picking a destination closer to the inner edge of $\mathbb{L}$ in the first phase, the transition towards the final path is made less abrupt.

constraints not part of the optimization framework (such as obstacles). To accommodate this challenge, we represent the projected final location of the carrier using a set, $\mathbb{L}(t, n_L)$, of the top n_L (highest likelihood) locations at a future time $t > 0$, and refine this set with each new observation. In practice, as more observations become available (as t increases), we expect the set to shrink, i.e., the number, n_L, required to adequately capture the variance in the set of hypotheses decreases.

Our algorithmic approach, listed in Algorithm 1, uses the set $\mathbb{L}(t_f, n_L)$ to plan n_L trajectories towards each hypothesis of $\mathbf{L}(t_f)$. The trajectory selection is then split into two phases. First, the UAS must conservatively follow *some* trajectory towards $\mathbb{L}$ until it reaches a certain distance threshold, or n_L falls below a threshold. At that point, it must "commit" to one trajectory towards $\mathbf{L}^*(t_f)$, the best estimate of the leader's location. At each trajectory generation step in the first phase, we plan n_L paths using the approach outlined in the previous subsection (Equations (7),(8)). We then greedily select a path that minimizes the cost

$$\textsc{Cost}(\mathcal{P}) = ||\frac{d^4}{dt^4}\mathcal{P}_{t_f}|| + ct_f,$$

at the current planning time, where c is a scaling constant. This encourages the selection of smoother paths that terminate at those locations that lie closer to the inner edge of $\mathbb{L}$. Figure 4(left) shows a graphical illustration of this strategy, with a solid black line representing the selected path. We could alternatively use a greedy strategy that picks a path towards the $\mathbf{L}(t_f)$ that has the highest likelihood (marked with a star in Figure 4). However, doing so could result in more dramatic changes in the shape of the selected path as $\mathbb{L}$ is refined and a final $\mathbf{L}^*$ is selected in the second phase. On the other hand, as shown in Figure 4(right), using our greedy approach, if final $\mathbf{L}^*$ is not one of the edge points, then the effect of the switching is generally a "relaxation" of the current path, and thus will lead to a smoother transition.

D. Trajectory control

The selected path and its first derivative are used as reference inputs for the UAS's controller. We use a feedback-linearized system model in conjunction with a linear quadratic regulator (LQR) to control the position and velocity of the UAS along the selected trajectory. The state equations for the system are then,

$$\hat{\boldsymbol{x}}_{k+1} = A\hat{\boldsymbol{x}}_k + B\boldsymbol{u_k}, \text{and } \boldsymbol{y}_k = C\boldsymbol{x}_k, \tag{16}$$

with

$$A = \begin{pmatrix} \boldsymbol{I}_{3x3} & \Delta t \cdot \boldsymbol{I}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{I}_{3x3} \end{pmatrix}, B = \begin{pmatrix} \frac{1}{2}\Delta t^2 & 0 & 0 & 0 \\ 0 & \frac{1}{2}\Delta t^2 & 0 & 0 \\ 0 & 0 & \frac{1}{2}\Delta t^2 & 0 \\ \Delta t & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 0 \\ 0 & 0 & \Delta t & 0 \end{pmatrix} \tag{17}$$

and $C = \boldsymbol{I}$, where $\boldsymbol{x} = [F, \dot{F}]^\top$ represents the position and velocity of the UAS, $\hat{\boldsymbol{x}}$ represents its best estimate (produced by a state estimator), and, $\boldsymbol{u} = [u_1, u_2, u_3, u_4]^\top$ is the acceleration control input in each degree of freedom. Since multirotors are invariant to rotation about the yaw-axis, the fourth control input, u_4, corresponding to the yaw acceleration is arbitrarily set to zero. The control is computed using both feedback and feedforward elements, such that,

$$\boldsymbol{u} = -K\hat{\boldsymbol{x}}_e + \ddot{\mathcal{P}}_{\tilde{t}_f}, \tag{18}$$

where, $\hat{\boldsymbol{x}}_e \equiv \hat{\boldsymbol{x}} - \boldsymbol{x}_r$ is the state error, and K is the feedback gain matrix obtained by solving the algebraic Riccati equations for a standard LQR problem. The reference input, $\boldsymbol{x}_r = [\mathcal{P}_{\tilde{t}_f}, \dot{\mathcal{P}}_{\tilde{t}_f}]^\top$, and the feedforward element, $\ddot{P}_{\tilde{t}_f}$, are obtained from the selected trajectory in the previous subsection. Incorporating the feedforward component from the generated plan allows us to efficiently handle the potential corner at the phase transition.

Algorithm 1 Pseudo-code listing for a greedy path selection.

```
procedure GREEDY-PATH-SEL
    while n_L > n_L^min or ||F̄L < d_thresh|| do
        P^sel ← 0
        min_cost ← ∞
        for each L(t_f) ∈ 𝕃 do
            P ← MINCOSTPATH(F, L(t_f))          ▷ Sec. III-B
            C(L(t_f)) ← COST(P)                 ▷ Sec. III-C
            if C(L(t_f)) < min_cost then
                min_cost ← C(L(t_f))
                P^sel ← P
            end if
        end for
        EXECUTEPATH(P^sel)                      ▷ Sec. III-D
    end while
    L* ← SELECTFINAL(𝕃)
    P ← MINCOSTPATH(F, L(t_f))                  ▷ Sec. III-B
    EXECUTEPATH(P)                              ▷ Sec. III-D
end procedure

function SELECTFINAL(𝕃)
    L* ← maxlikelihood(𝕃)
    return L*
end function
```

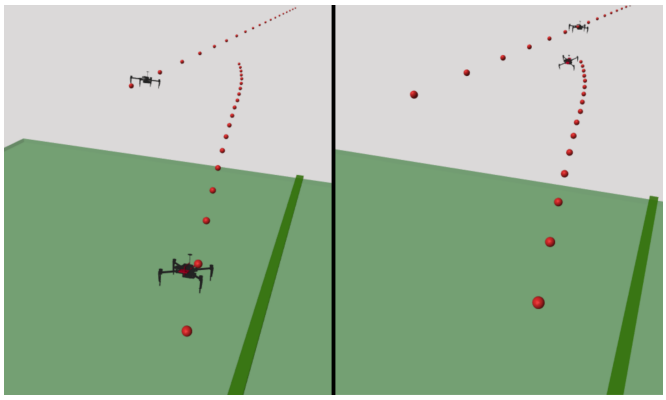

Fig. 5: Two snapshots from an ideal simulation (where the exact locations of the leader are known), where no refinement is necessary. The pictures are taken at the beginning (left) and at the end of the trajectory (right). The projected path of the leader, and the planned trajectory for the follower are indicated with markers.

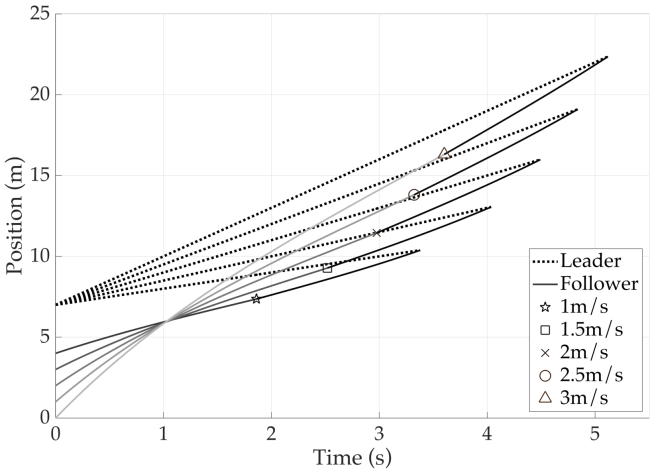

Fig. 6: Generated two-part trajectories to dock with a leader vehicle moving at five different speeds. The starting locations for the follower vehicle are also different for each case. The markers also indicate the point where the phase switch occurs.

The non-linear rigid-body dynamics of multirotors are well known in literature [19]. The acceleration control input in Equation (18) is converted to attitude and thrust commands for the vehicle by inverting a non-linear map of the vehicle's kinematics:

$$[\theta_d, \phi_d, \dot{\psi}_d, T_d] = g_{kin}(\boldsymbol{u}, m),$$

where $\theta_d, \phi_d, \dot{\psi}_d$, and T_d are the desired pitch, roll, yawrate and thrust targets, and m is the total mass. The autopilot inner-loop uses a PID controller to generate motor commands to follow the desired targets. Since the control is split into a outer/inner-loop framework, the same methods are applicable for multirotors with different number of rotors. We observe that this property also allows us to exploit the differential flatness of the system and plan trajectories entirely in the flat output space, $\boldsymbol{y}$.

IV. Evaluation

In the previous section, we have already shown the docking trajectories generated for different profiles of the leader UAS. We will evaluate the complete framework of trajectory generation, selection and execution by simulating the two UASs. In our evaluations, we do not model the effect of external disturbances (such as wind, or mechanical failures in docking), and will assume that executing the generated path leads to a successful docking.

Simulation Framework

Our simulations are designed using Freyja Simulator [1], which is a high-fidelity physics-based simulation framework developed within MATLAB and Simulink ™. Freyja models the multirotor UASs as 6-degree of freedom (DOF) rigid bodies with custom inertia matrices, and allows their aerodynamic body-drag profile to be specified. An inner-loop PID controller regulates the vehicle's attitude to the target attitude. The time-constant for the attitude controller is empirically chosen to mimic real commercially available autopilots (such as a Pixhawk). The LQR controller described in Section III-D is implemented as a library block in Freyja. This allows us to easily instantiate multiple simulated UASs with their associated inner- and outer-loop controllers. We implement the rest of the algorithm using custom function calls written in the MATLAB environment. All the simulations are run on a laptop computer with a 2.9 GHz Intel processor and 16 GB of memory.

[1] `github.com/unl-nimbus-lab/Freyja-Simulator`

Planning

Figures 1 & 3 already illustrate some of the paths obtained by our trajectory generation method. Figure 5 also shows a snapshot from a simulation with the two vehicles and their projected and planned trajectories overlaid. The results are obtained by choosing $R_s = 1.5$, $w_1 = 1.5$ and $w_2 = 1.8$. Furthermore, as shown in Figure 6, we explore a range of different speeds for the leader vehicle, while the planning parameters are all held constant. The figure illustrates the trajectories generated by the follower UAS initialized at different starting locations ($F(0)$) to dock with the leader.

We see that the resultant trajectories are continous, piecewise-smooth, and meet the location of the leader vehicle at some optimal time t_f. As expected, the planned t_f is higher for a faster moving leader, and for a larger initial separation between the two vehicles. These examples explore the scenario where the position of the leader is known accurately. We will now drop this assumption and evaluate the effectiveness of our replanning and trajectory selection strategies from Section III-C.

Replanning Effectiveness

Using Freyja, we simulate two practical effects that arise in multirotor flights: the effect of aerodynamic body-drag (which is unmodeled in Equations (16)-(18)), and the uncertainty in the leader's position. The aerodynamic drag is modeled simply as a lateral force as a function of velocity, $f_d \propto$

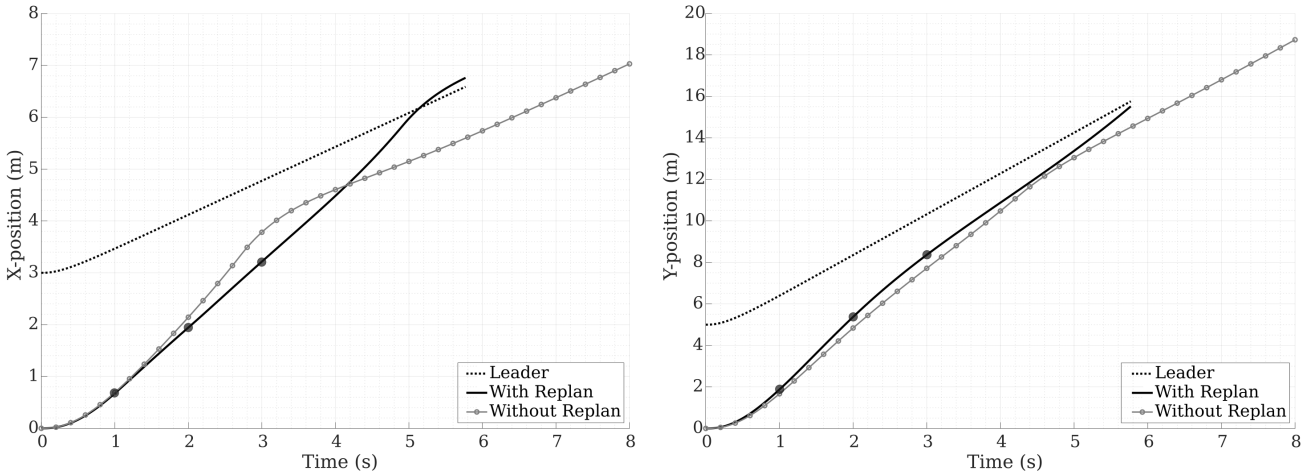

Fig. 7: Comparison of results from our simulations with and without replanning the trajectories. In both cases, the leader's position is distributed uniformly around its true value. Replanning is done once every second (highlighted by black circles). When replanning is disabled, a trajectory is planned only at $t = 0$ using only one (noisy) estimate of $\mathbf{L}(0)$. The planned trajectory never meets $\vec{L}$ because of this initial error which gets propagated with time.

$k_d v^2$. The uncertainty in the leader's state is captured using n_L particles, drawn from a uniform distribution centered around the true state. The threshold values in Algorithm 1 are set to $n_L^{min} = 5$ and $d_{thresh} = 2\,\mathrm{m}$. We mimic the decrease in this uncertainty at each planning iteration by a weighted sub-sampling of the set of particles. This simulates the convergence of an estimated state to the true state as more observations are made available from a sensor.

Figure 7a and 7b show the x and y axis plots of the trajectories with and without replanning. The trajectory replanning is triggered every second, and Algorithm 1 is used to iteratively select a path. In the case where no replanning is done, an optimal trajectory is computed only at time $t = 0$ using a sampled value of $\mathbf{L}(0)$ from the hypotheses set. Since the path is never refined again, the generated trajectory will not necessarily intersect the leader's path ($\vec{L}$). However, when replanning is enabled, we observe that the trajectory deviates from the original plan starting from the second iteration ($t = 1$), and eventually meets $\vec{L}$.

Selection Strategy

We also present an evaluation of our greedy path selection algorithm of Section III-C. Recall that in every iteration, we select a path that minimizes the weighted sum of the overall snap and the time of the mission. To show the utility of this method, we compare the algorithm against a "naïve" strategy that selects a path based simply on the maximum-weight $\mathbf{L}(t)$. Figure 8 shows the vector magnitude of the snap in the xy axis for the two approaches. The aggregated magnitude over the entire mission is $0.125\,\mathrm{m/s^4}$ for our approach compared to $0.185\,\mathrm{m/s^4}$ for the naïve method.

The magnitude is generally higher for the naïve approach, since 1) it does not prioritize snap minimization, and 2) the selected $\mathbf{L}(t)$ can have abrupt "jumps" in successive iterations. Using our greedy approach, we conservatively

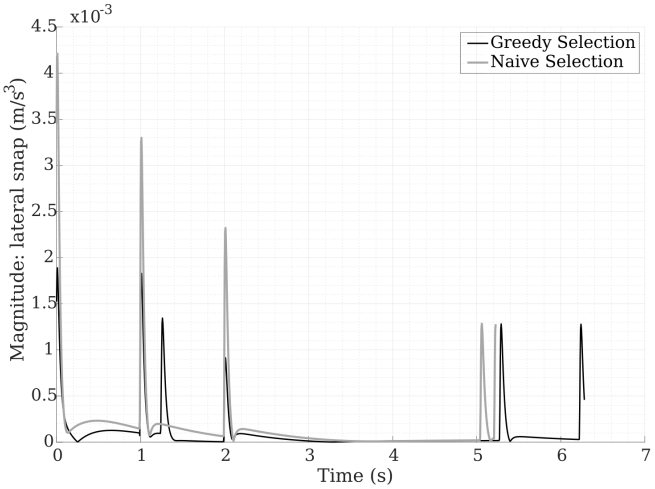

Fig. 8: The vector magnitude of snap in xy axis for the duration of one mission. Using our greedy path selection strategy, the overall snap cost is reduced compared to naive approach that chooses the one with the highest weight.

select a shorter path, and "relax" it with every iteration. Note that the naïve approach can perform better depending on the physical sensor's noise profile: it is possible that the jump in the selected $\mathbf{L}(t)$ is smaller if the sensor is extremely accurate. However, the greedy approach we implement achieves a lower cost in the general case.

Discussion

The greedy path-selection strategy is extensible beyond the jerk-cost minimization presented in this work. The ability to rapidly generate a set of paths towards multiple hypotheses of the leader's location can be useful for several other practical scenarios. For instance, since t_f and the constants $C_{1..4}$ are unconstrained, some paths could be preemptively dropped if they result in paths are infeasible in time, or in the acceleration control effort ($\ddot{\mathcal{P}}_{\bar{t}_f}$). Similarly, if an obstacle

map of the world is available (or sensed), the follower can eliminate certain paths that risk collisions.

The evaluations presented in the preceding sections indicate the feasibility of the proposed approach in generating optimal docking trajectories by simultaneously solving Equations (9)-(14). A solution to these may be obtained analytically, or by using a software package (such as MATLAB's `solve()` method). As mentioned earlier, the variables w_1 and w_2 are design parameters that can be altered to generate trajectories with varying levels of aggressiveness. We also observe that the resultant trajectory is smooth if $w_1 = w_2$, and that the smoothness of the transition depends on the difference $w_1 - w_2$. Additionally, for a well-defined solution, we require that $R_s < ||F(0) - \mathbf{L}(0)||$ and $R_s^2 < 2w_2$.

The generated paths using this approach have the general form presented in Eq (7) (8) because of the assumption that the leader's trajectory is linear. A corresponding optimization problem can be formulated for leader trajectories that follow arbitrary time-parameterized curves. In general, however, an analytical solution may not be guaranteed for such problems. We will investigate this problem in future work.

V. Conclusion

In this paper, we have presented and evaluated a strategy for generating optimal trajectories for docking between multirotor UASs in forward flight. The proposed method is used iteratively to greedily select a path that minimizes the snap cost of the trajectory until interception is affected. Our preliminary hardware tests have shown promising results in implementing the method onboard a multirotor UAS. In the future, we will extend this method to practical field tests under varying scenarios.

References

[1] D. B. Wilson, A. Göktogan, and S. Sukkarieh, "Guidance and Navigation for UAV Airborne Docking," in *Robotics: Science and Systems*, 2015. [Online]. Available: doi.org/10.15607/rss.2015.xi.017

[2] 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*. IEEE, Oct. 2018.

[3] K. Klinkmueller, A. Wieck, J. Holt, A. Valentine, J. E. Bluman, A. Kopeikin, and E. Prosser, "Airborne Delivery of Unmanned Aerial Vehicles via Joint Precision Airdrop Systems," in *American Institute of Aeronautics and Astronautics (AIAA) Scitech 2019 Forum*.

[4] R. Miyazaki, R. Jiang, H. Paul, K. Ono, and K. Shimonomura, "Airborne Docking for Multi-Rotor Aerial Manipulations," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 4708–4714.

[5] J. W. Nichols, L. Sun, R. W. Beard, and T. McLain, "Aerial Rendezvous of Small Unmanned Aircraft Using a Passive Towed Cable System," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1131–1142, Jul. 2014.

[6] M. D. Tandale, R. Bowers, and J. Valasek, "Trajectory Tracking Controller for Vision-Based Probe and Drogue Autonomous Aerial Refueling," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 4, pp. 846–857, 2006.

[7] 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 (ICRA)*, May 2018, pp. 691–698.

[8] 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.

[9] 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.

[10] 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.

[11] H. Jiang, S. Elbaum, and C. Detweiler, "Inferring and monitoring invariants in robotic systems," *Autonomous Robots*, vol. 41, no. 4, pp. 1027–1046, Apr. 2017. [Online]. Available: https://doi.org/10.1007/s10514-016-9576-y

[12] F. Kunwar, F. Wong, R. B. Mrad, and B. Benhabib, "Guidance-Based On-Line Robot Motion Planning for the Interception of Mobile Targets in Dynamic Environments," *Journal of Intelligent and Robotic Systems*, vol. 47, no. 4, pp. 341–360, Dec. 2006.

[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] T. Baca, P. Stepan, V. Spurny, D. Hert, R. Penicka, M. Saska, J. Thomas, G. Loianno, and V. Kumar, "Autonomous landing on a moving vehicle with an unmanned aerial vehicle," *Journal of Field Robotics*, vol. 36, no. 5, pp. 874–891, 2019.

[15] B. Penin, R. Spica, P. R. Giordano, and F. Chaumette, "Vision-based minimum-time trajectory generation for a quadrotor UAV," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 6199–6206.

[16] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-Based Reactive Planning for Aggressive Target Tracking While Avoiding Collisions and Occlusions," *Robotics and Automation Letters*, vol. 3, no. 4, 2018.

[17] M. Geisert and N. Mansard, "Trajectory generation for quadrotor based systems using numerical optimal control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.

[18] H. Yu, R. W. Beard, M. Argyle, and C. Chamberlain, "Probabilistic path planning for cooperative target tracking using aerial and ground vehicles," in *Proceedings of the 2011 American Control Conference*, Jun. 2011, pp. 4673–4678.

[19] 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.