Freyja: A Full Multirotor System for Agile & Precise Outdoor Flights

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

***Abstract*—Several independent approaches exist for state estimation and control of multirotor unmanned aerial systems (UASs) that address specific and constrained operational conditions. This work presents a complete end-to-end pipeline that enables precise, aggressive and agile maneuvers for multirotor UASs under real and challenging outdoor environments. We leverage state-of-the-art optimal methods from the literature for trajectory planning and control, such that designing and executing dynamic paths is fast, robust and easy to customize for a particular application. The complete pipeline, built entirely using commercially available components, is made open-source and fully documented to facilitate adoption. We demonstrate its performance in a variety of operational settings, such as hovering at a spot under dynamic wind speeds of up to 5–6 m/s (12–15 mi/h) while staying within 12 cm of 3D error. We also characterize its capabilities in flying high-speed trajectories outdoors, and enabling fast aerial docking with a moving target with planning and interception occurring in under 8 s.**

Fig. 1: Snapshots depicting instances of a multirotor UAS in different outdoors scenarios: (top) intercepting parachutes mid-air, (bottom) flying aggressive circles around a spot.

I. Introduction

Field applications of multirotor unmanned aerial systems (UASs) have become increasingly realistic and far-reaching over the last decade. This is due, in part, to a sustained development of their potential as field agents that work in real and complex environments found 'in the wild'. Modern use-cases for multirotors span the breadth of environmental sciences (profiling the lower atmosphere [1], monitoring soil and crops [2], studying water bodies [3], etc), and autonomous search and rescue operations [4]. While these have advanced the capabilities of multirotors, they do not always require precise and accurate control of the trajectories of the multirotor. The next generation of outdoor applications, such as intercepting objects in the air [5] and docking with moving aircraft [6] will require significant advances in state estimation and control implementations, demonstrated outdoors.

To realize such agile, precise and interactive field missions, we must account for natural and loosely modeled phenomena (such as wind and aerodynamic drag), and deviations from expected model parameters (such as the total mass, changing battery voltage, idealized transfer functions etc.) that pose challenges for accurate flights. These adversely affect the performance of a controller, and are more noticeably evident when flying complex time-bound trajectories. Robust compensation for such dynamic effects typically require either extremely customized solutions, or are limited to more constrained and simulated indoor/lab settings. At present, there is a gap between the research/prototype state-of-the-art approaches [7], [8], [9], and their full realization as field agents. We are currently lacking a complete and generalized end-to-end pipeline for high-level state estimation and precise control over aggressive trajectories outdoors.

In this paper, we introduce such a pipeline that we call *Freyja*, that addresses this gap through efficient, modular elements that fit together cohesively on small onboard computers. We position this work in the context of systems and components that are cost effective, commercially available, and require no specific customizations. By building on a modular architecture using robust and individually optimal elements, we show a complete system that can not only measure and reject unexpected extrinsic disturbances found in field missions, but also extend the envelope of such missions by performing precise, aggressive and feedforward maneuvers usually confined indoors. Figure 1 depicts two instances of such missions where a multirotor is required to exercise precise control for intercepting airborne parachutes, and for flying aggressive trajectories outdoors.

The system presented in this work is designed around a small-sized quadcopter frame equipped with an attitude-stabilizing autopilot (such as the popular Pixhawk). Our approach builds around three key enablers that address localization, trajectory formulation and control. For localization, we use a miniaturized low-power real-time kinematic (RTK) GPS unit for precise global and map-frame positioning. This data, fused with inertial measurements through an Extended Kalman Filter (EKF), provides the fast and accurate system state required by a controller. We allow a wide scope for trajectories, ranging from discrete waypoints and discontinuous paths, to continuous and smooth parametric curves.

[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 NSF IIS-1925052, IIS-1638099, IIS-1925368, IIS-1924777 and NASA ULI-80NSSC20M0162.

The control strategy utilizes a linear quadratic gaussian (LQG) control (which is a tandem implementation of a linear quadratic regulator (LQR) and a full-state Kalman filter) [7] along with trajectory feed-forward components to precisely track a reference trajectory in time and space. The observer in LQG is capable of measuring 3-axis extrinsic disturbances acting upon the system, which allows the feedback controller to reject them in the successive iterations. The system is feedback linearized over a nested autopilot loop, exploits the differential flatness of a multirotor system, and uses a non-linear inversion map to generate control inputs to the autopilot. This allows highly dynamic trajectories (and their feed-forward components) to be planned entirely in the output space using any of the classical planning methods. The proposed system remains oblivious to the type of multirotor (quad-, hexa- etc) by delegating the low-level attitude stabilization to a well-tuned autopilot.

The key contributions of this work are:

- A complete end-to-end pipeline that addresses state estimation, trajectory generation, and precise control under challenging outdoor conditions;
- An analysis of the impact of developing feedforward control & optimal bias observers for real environments;
- Outdoor evaluations and demonstrations of trajectory control for translational speeds over $6\,\mathrm{m/s}$, hovering with a 3D error of less than $4\,\mathrm{cm}$, and precise control for aerial docking with a moving target in under $8\,\mathrm{s}$.

II. Background

Fast and accurate estimates of the inertial position and velocity of the UAS in outdoor environments is key to precise trajectory control. The requirements in precision may vary for different applications; an initally coarse estimate might suffice for large-area applications such as search and rescue [4], [10]. An extremely high precision, on the order of a few centimeters, is necessary for closer interactions such as inspecting structures [11], landing on targets [12], or perching on power lines [13]. Consumer-grade global positioning systems are severely restrictive in such cases, with stated accuracies well above $1.5\,\mathrm{m}$ [14]. Consequently, several of these applications fuse visual-inertial data from onboard cameras and lasers. When GPS is available, differential solutions and real time kinematic (RTK) systems can offer significantly higher accuracies (on the order of 2–$3\,\mathrm{cm}$). Fusing low-rate RTK data with IMU measurements and/or visual odometry (VIO) has shown highly promising results [15], [16]. This is enabled by newer commercially available solutions that are miniaturized enough to be retrofitted to small multirotors.

Several state feedback and control approaches have been also developed for underactuated systems (for instance, [7] and references therein). For multirotors, these are developed using system model representations that are extremely detailed [8] or more abstract [17], depending on the context of the problem. Indoors, and in semi-structured environments, where motion-capture or VIO can provide reliable state information, multirotors have been used to demonstrate agile maneuvering tasks [18], grasping objects [19], and agile load transport [9]. While some of these approaches may be transferable to systems 'in the wild', we still lack detailed evaluations outdoors.

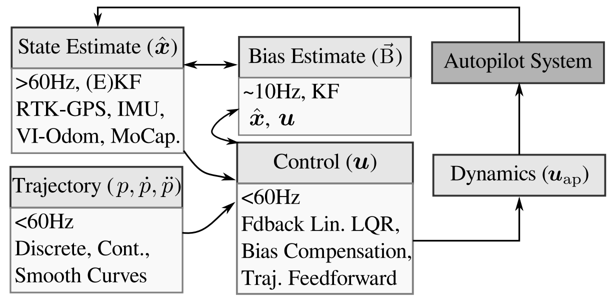


Fig. 2: A block diagram representation of the system architecture. We address each of the modules independently, and make them amenable to drop-in replacements.

Our approach here is developed using a similar high-level (point mass) representation that encapsulates nested autopilot loops so that the resultant system can generalize better. Complex system models that account for aerodynamic effects such as blade flapping and aerodynamic drag can be crucial for aggressive flight regimes [15], [20], however, their application to outdoor flight has been fairly limited. Similarly, trajectory generation methods that exploit a UAS's differential flatness and shape smooth accelerations have been demonstrated [18] primarily for constrained indoor environments. Recent work has demonstrated such methods outdoors applied to aerial docking missions [21]. Our objective is to bridge this gap with a complete system that can perform agile maneuvers outdoors under real disturbances.

III. Technical Details

Figure 2 shows a block-diagram view of our architecture, where each shaded rectangle represents a modular component of the complete pipeline. We will describe the individual modules in a logical progression in the following subsections. Note that each module is capable of having drop-in replacements in the form of alternative choices of sensors, control system and planning.

We let $\mathcal{W}$ represent the world-fixed NED (north-east-down) coordinate frame. In the following text, a local (map) frame, $\mathcal{M}$, is assumed to be rigidly fixed in $\mathcal{W}$, with its axes aligned with $\mathcal{W}$ and its origin initialized where the UAS is initialized. The translational position, $P^{\mathcal{M}}$, and the velocity, $\dot{P}^{\mathcal{M}}$, of the UAS are expressed in this local frame. We assume that the rotation angles and the rates, both expressed in the vehicle's body frame, are handled by the autopilot.

A. System Model

We develop the estimation and control pipeline on a feedback-linearized translational system model of the UAS, incorporating elements from classical approaches in literature [8], [17]. A distinguishing element in our design is the separation of the controller state from the observer state. The model is derived from the dynamics of a rigid body system (b) with six degrees of freedom (DOF) with mass m,

$$m\vec{\mathbf{a}} = -R^{b}_{\mathcal{M}} \cdot T + \hat{\mathrm{e}}_{\mathrm{d}} m g, \tag{1}$$

where $R_a^b \in \mathsf{SO}(3)$ denotes the 3×3 rotation between the frames a and b, T is the collective thrust produced by the rotors, g is the acceleration due to gravity and $\hat{\mathrm{e}}_\mathrm{d}$ denotes a unit vector along the vertical (down) axis of the inertial frame. The matrix $R^b_{\mathcal{M}}$ is obtained from the Euler roll (ϕ), pitch (θ) and yaw (ψ) angles of the UAS body in the Z-Y-X rotation order. Thus, by assuming that desired values of these angles and a collective thrust can be maintained by an autopilot's "inner loop", we can affect a desired linear acceleration, $\vec{\mathbf{a}} \in \mathbb{R}^3$, of the body in the inertial frame. We therefore define the control command sent to the autopilot as $\boldsymbol{u}_\mathrm{ap} = [\phi_d, \theta_d, \psi_d, T_d]^\top$ composed of the desired values of these quantities.

The non-linear system defined by Eqn (1) lets us model a linear system with second-order dynamics with accelerations, $\vec{\mathbf{a}}$, as its inputs. For this system, we define a state vector,

$$\begin{aligned} \boldsymbol{x} &\equiv [P^{\mathcal{M}}, \dot{P}^{\mathcal{M}}, \psi]^\top \\ &= [p_n, p_e, p_d, v_n, v_e, v_d, \psi]^\top, \end{aligned} \tag{2}$$

composed of the translational position, velocity and the heading of the UAS, all expressed in the inertial frame. The dynamics can then be expressed in the traditional form,

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

with,

$$A = \begin{pmatrix} 0_{3x3} & \mathbb{I}_{3x3} & 0_{3x1} \\ 0_{3x3} & 0_{3x3} & 0_{3x1} \\ 0_{1x3} & 0_{1x3} & 0 \end{pmatrix}, B = \begin{pmatrix} 0_{3x3} & 0_{3x1} \\ \mathbb{I}_{3x3} & 0_{3x1} \\ 0_{1x3} & 1 \end{pmatrix}, C = I.$$

The control input to this feedback-linearized system is a 4-vector composed of the translational accelerations from Eqn (1) and a body-frame rotational rate, $\dot{\psi}$, such that,

$$\boldsymbol{u} \equiv [\vec{\mathbf{a}}, \dot{\psi}]^\top. \tag{4}$$

Thus, if appropriate acceleration control inputs, $\boldsymbol{u}$, are known for the linearized system, we can decompose them into $\boldsymbol{u}_\mathrm{ap}$ by a non-linear inversion of Eqn (1).

B. State Estimation

We generally require a robust and reliable source of state information to perform accurate and high-speed maneuvers. To prevent erroneous feedback control, we further require this information to be updated faster than the control cycle. Typical GPS systems offer update rates that are too low ($\approx 10\,\mathrm{Hz}$) and are often too inaccurate. For instance, a high-end GPS accuracy of $0.8\,\mathrm{m}$ can be almost twice the diameter of medium-sized multirotors. For localized operations (within a radius of 1–2 km), we therefore switch to ground-based augmentation systems (GBAS) to achieve significantly higher accuracy in measurements. This is realized in the form of real time kinematic (RTK) GPS systems that can produce position measurements with more than $5\,\mathrm{cm}$ of accuracy at a similar rate. The accuracy also remains fairly consistent within the operational range of RTK systems.

We split the state estimation into two separate "processes" – one that estimates the controllable system states defined in model, and another that estimates a state model with biases. An optimal state estimator for both allows a controller to optimally regulate the state by *certainty equivalence*. By the separation principle, we also know the combined system will retain its stability guarantees. This also lets us design these modules independently.

Controller States. For agile maneuvering, RTK-GPS data is fused with inertial measurements from an onboard IMU (in the autopilot). We adopt an Extended Kalman filter (EKF) formulation, and rewrite the non-linear system as

$$\begin{aligned} \dot{\boldsymbol{x}} &= f(\boldsymbol{x}, \boldsymbol{u}, u), \\ z_\mathrm{pos} &= h_1(\boldsymbol{x}, v), z_\mathrm{imu} = h_2(\boldsymbol{x}_b, w) \end{aligned} \tag{5}$$

where f, h_1 and h_2 represent the state transition and measurement maps, $\boldsymbol{x}_b$ is a new state variable containing only the attitude angles in the body frame, and u, v, w are the corresponding zero-mean additive noises over a Gaussian distribution. The filter then estimates $\hat{\boldsymbol{x}}$ at a sufficiently high rate for the controller. The product of this block, eventually, is the best estimate of the state, $\hat{\boldsymbol{x}}$, as defined above and expressed in $\mathcal{M}$. Several other fusion methods, such as visual-inertial odometry (VIO), and visual pose estimation from onboard cameras [22], [23] or motion-capture systems could provide the state information at a sufficiently high rate.

Observer States. To design the state observer in LQG, we augment the state vector in Eqn (2) to include extrinsic time-varying forces. We represent these in the form of accelerations acting upon the system, so that for the bias observer, the augmented system model is represented by

$$\boldsymbol{x}_\mathrm{B} \equiv [\boldsymbol{x}^\top, \vec{\mathbf{B}}^\top]^\top \tag{6}$$

$$\dot{\boldsymbol{x}}_\mathrm{B} = A_\mathrm{B}\boldsymbol{x}_\mathrm{B} + B_\mathrm{B}\boldsymbol{u}, \text{ and, } y_\mathrm{B} = C\boldsymbol{x}_\mathrm{B} \tag{7}$$

$$\text{with, } A_\mathrm{B} = \begin{pmatrix} A & \mathbb{I}_{3x3} \\ 0_{3x7} & 0_{3x3} \end{pmatrix} \text{ and } B_\mathrm{B} = \begin{pmatrix} B \\ 0_{3x4} \end{pmatrix}$$

such that, $\vec{\mathbf{B}} = [b_n, b_e, b_d]^\top$ denotes the 3-axis external disturbances that act as biases on the system.

In aggressive maneuvering, aerodynamic drag plays a significant role in the dynamics [15], [20]. Instead of explicitly modeling it, we let the bias estimator measure it as an external force, which a controller can then compensate for. By appropriate pole-placement of the estimator, the dynamics of the estimator can be fast enough to measure other deviations from the system model such as an incorrect mass (m) variable, an off-center loading, or a changing thrust due to battery voltage.

C. Control

The control input, $\boldsymbol{u}$, from Eqn (4) applied to the system is designed with three components, such that,

$$\boldsymbol{u} \equiv \boldsymbol{u}_\mathrm{fb} + \boldsymbol{u}_\mathrm{bc} + \boldsymbol{u}_\mathrm{ff}, \tag{8}$$

where the subscripts fb, bc and ff denote the feedback, bias compensation, and the feed-forward elements of the signal. Similar feedforward designs based on differential flatness of the multirotor system have been employed previously [17]. For outdoor flights where external disturbances can manifest in several time-varying forms, the bias compensation term plays a very significant role. Our modeling of these disturbances as accelerations let us incorporate corrections directly into the the control equation.

Feedback. For a linear system model described by Eqs (2)-(3), it is possible to design a feedback control law that regulates the state vector, $\boldsymbol{x}$, and drives the error exponentially to zero. Denoting a reference state in time as $\boldsymbol{x}_r$, we write the feedback control equation as

$$\boldsymbol{u}_{\mathrm{fb}} = -K(\boldsymbol{x} - \boldsymbol{x}_r), \tag{9}$$

where K is the feedback gain matrix. Substituting $\boldsymbol{u}_{\mathrm{fb}}$ for $\boldsymbol{u}$ in Eqn (3), the resultant system dynamics can be rewritten as $\dot{\boldsymbol{x}} = (A - BK)\boldsymbol{x} = \tilde{A}\boldsymbol{x}$. For a stable system, the eigenvalues of $\tilde{A}$ must lie strictly on the left-half of the complex plane. Thus, the design matrix K can be chosen to affect a desired pole placement for the system.

Theoretically, this feedback gain matrix can be chosen to produce an arbitrarily fast convergence to the desired $\boldsymbol{x}_r$. In practice, physical constraints on the system (such as motor response time, clipped battery power, etc) limit large changes in the control effort between successive time steps. Furthermore, a smoother control is often more desirable in many practical applications such as environmental sensing and interactions. Thus, we use a Linear Quadratic Regulator (LQR) design to select an optimal feedback gain matrix K that balances the control expenditure of the system against its ability to regulate state errors. This feedback matrix, denoted K_{lqr}, is the solution for an Algebraic Ricatti Equation (ARE) that minimizes the cost functional

$$J(\boldsymbol{x}, \boldsymbol{u}) = \int_0^\infty \boldsymbol{x}_e Q \boldsymbol{x}_e^\top \mathrm{d}t + \int_0^\infty \boldsymbol{u}_{\mathrm{fb}} \mathrm{R} \boldsymbol{u}_{\mathrm{fb}}^\top \mathrm{d}t.$$

Bias Compensation. Recall from Section III-B that the state observer models external disturbances acting on the UAS as accelerations (or, equivalently as forces) in the three translational axes. Since the control input, $\boldsymbol{u}$, represents acceleration inputs to the system, we need no additional operations to transform the measured disturbances. That is, the bias vector is related to its compensation in the control law by an identity transform:

$$\boldsymbol{u}_{\mathrm{bc}} = O_b \vec{\mathbf{B}} = \begin{pmatrix} -\mathbb{I}_{3x3} \\ 0_{1x3} \end{pmatrix} \vec{\mathbf{B}}. \tag{10}$$

Feed-forward. The final element of the control input is a feedforward signal that can be derived from a trajectory, $p(t)$, that is continuous and temporally smooth up to 3rd-order. For such paths, we have that $p(t), \dot{p}(t)$ as well as $\ddot{p}(t)$ are well-defined for all time t. The reference state for the feedback regulator, $\boldsymbol{x}_r \in \mathbb{R}^7$, is still composed only of $p(t)$, and $\dot{p}(t)$ (as well as heading).

Since multirotor systems are differentially flat, we know that by carefully selecting an output, $y_{\mathrm{df}} = C_{\mathrm{df}}\boldsymbol{x}$, we can express the system states as well as the system control inputs as functions of y_{df}, $\dot{y}_{\mathrm{df}}$, $\ddot{y}_{\mathrm{df}}$ and so on. In this case, we select only the translational position in three axes as the flat output, i.e., $C_{\mathrm{df}} = \begin{pmatrix} \mathbb{I}_{3x3} & 0_{3x4} \end{pmatrix}$, and thus, $y_{\mathrm{df}} = [p_n, p_e, p_d]^\top$. Again, since the control inputs to the system are accelerations, we can directly employ $\ddot{y}_{\mathrm{df}} = \ddot{p}(t)$ as the feedforward control, such that, $\boldsymbol{u}_{\mathrm{ff}} = \begin{pmatrix} \mathbb{I}_{3x3} \\ 0_{1x3} \end{pmatrix} \ddot{p}$.

Note that we do not design a feedforward component for the heading (yaw) control of the UAS. Since multirotors are typically invariant to yaw, and high accelerations in heading are less common in trajectories, we do not prioritize yaw agility in the outer-loop control in this work. However, if required, this can be incorporated by changing C_{df} and planning smooth trajectories for yaw.

The final control input from Eqn (8) is then,

$$\boldsymbol{u} = -K_{\mathrm{lqr}}(\boldsymbol{x} - \boldsymbol{x}_r) + \begin{pmatrix} -\mathbb{I}_{3x3} \\ 0_{1x3} \end{pmatrix} \vec{\mathbf{B}}. + \begin{pmatrix} \mathbb{I}_{3x3} \\ 0_{1x3} \end{pmatrix} \ddot{p}. \tag{11}$$

This represents the desired accelerations in three translational axes and one rotational axis (yaw) for the rigid body. As mentioned in Section III-A, using the total mass, m, the actual control input to the autopilot, $\boldsymbol{u}_{\mathrm{ap}} = [\phi_d, \theta_d, \psi_d, T_d]^\top$ can now be obtained by inverting Eqn (1).

IV. Studies

We now demonstrate the capabilities of the proposed architecture, along with the impact of its individual elements. The focus in these results is the ability of this pipeline to estimate and compensate for external disturbances, and execute dynamic trajectories with high precision in the field. We therefore select three illustrative scenarios that encompass a variety of our outdoor missions: hovering at a spot, flying in a circle, and executing a planned interception mission. For each of these, we will consider the time-sensitive trajectory tracking performance of our system, and its ability to reject external disturbances in all axes. For circles and more dynamic planned trajectories, our system benefits from incorporating a feedforward element.

Implementation Details

For the purposes of a fair and replicable evaluation, we implement the presented pipeline on a commercially available and fully open-source system. The hardware frame is an off-the-shelf DJI Flamewheel quadrotor with brushless DJI

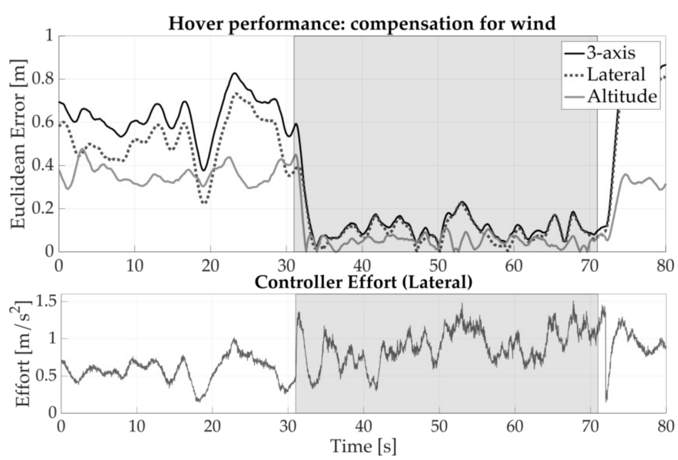
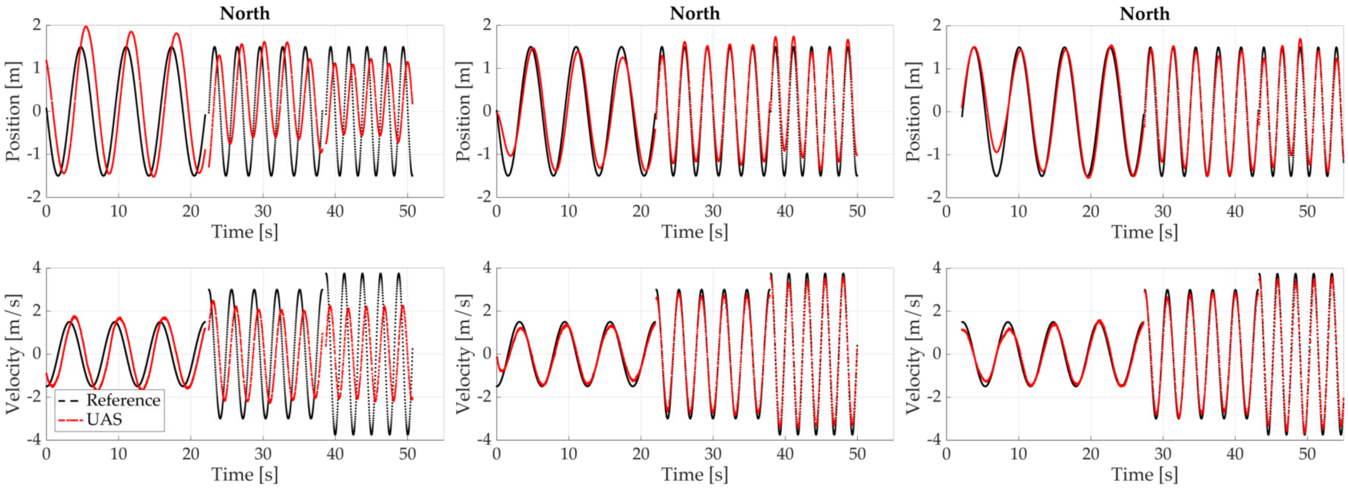


Fig. 3: Hover performance under wind speeds of up to $5.4\,\mathrm{m/s}$. Wind compensation is active during the shaded region.



(a) LQR, no FF. **(b)** LQR+FF, no compensation **(c)** LQR+FF, bias compensation

Fig. 4: Comparison of results in tracking circular trajectories of a fixed radius ($1.5\,\mathrm{m}$) and increasing angular rates with various elements of the pipeline enabled. (a) Naive LQR feedback with no feedforward and no bias compensation, (b) LQR with trajectory feedforward enabled, and, (c) LQR with trajectory feedforward and bias compensation from the full LQG system. Due to ambient wind, a steady offset can be observed in (b) which is corrected and centered in (c) by the bias estimation process. Ambient wind: 2–3 $\mathrm{m/s}$ N.

motor-ESC systems. The UAS measures $\approx 45\,\mathrm{cm}$ diagonally, weighs $1.2\,\mathrm{kg}$ with battery, and is capable of lifting more than an additional $1\,\mathrm{kg}$. The autopilot is a commercial Pixhawk board running a fork of the open-source ArduCopter firmware. We equip the UAS with a u-blox ZED-F9P board that produces precise RTK-GPS data using standard GPS antennas at $5\,\mathrm{Hz}$. The rest of the implementation is all written in C/C++ over Robot Operating System (ROS) middleware stacks, and implemented entirely onboard on an Odroid XU4. This is made publicly available[1]. The system model and feedback gains are developed on the complementary Freyja-Simulator[2]. For instance, the gain matrix K can be obtained and validated in the simulator environment using MATLAB's `place()` or `dlqr()` commands.

Our system architecture is easily adapted to several different autopilot and UAS systems by only configuring the system parameters/scalars of the model. The pipeline presented here has also been extensively employed and flight tested on Ascending Technologies' autopilot and frames, in indoor motion-capture environments over wireless telemetry, and through other sources of state information such as an Intel RealSense T265 camera [22] and monocular vision pipelines both indoors and outdoors [24].

A. Hovering, Wind Resistance

In the first evaluation, we require the UAS to be positioned at a fixed 3D point in space under the presence of varying wind disturbances. Furthermore, to increase the estimation complexity, we specify a slightly higher mass in the system model ($+0.1\,\mathrm{kg}$), which results in a higher thrust than required. These two combined effects are common in outdoor missions, specifically those which involve handling cargo.

Figure 3 shows the positioning Euclidean errors $||\boldsymbol{x} - \boldsymbol{x}_r||_2$ from a fixed reference as a function of time. The average

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

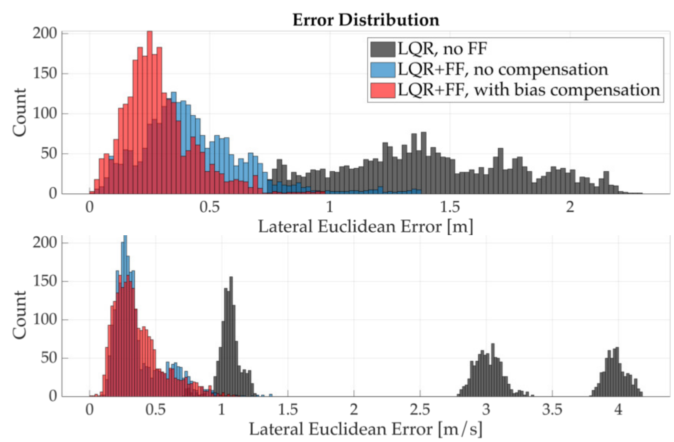


Fig. 5: Distribution of lateral trajectory tracking errors for position (top) and velocity (top) references. The three histograms represent data from the three columns in Figure 4.

wind speed during the flight is around $5\,\mathrm{m/s}$. We switch the bias compensation on mid-flight (shaded region in figure) to capture its dynamics. We notice that the lateral (2D) and the 3D errors are typically over $0.5\,\mathrm{m}$ when the compensation is inactive. When activated, the error rapidly diminishes to an average of $\approx 0.125\,\mathrm{m}$ in the shaded region. The estimator converges to its steady value within $2\,\mathrm{s}$ of activation, and also aids in reducing the vertical error due to an incorrect mass.

B. Circles

Next, we investigate the performance of the system over time-parameterized trajectories. As mentioned before, continuous and twice-differentiable paths can enable feed-forward elements in the controller, thereby aiding its temporal performance as well. Circles are well-suited for these tests, since the parametric cartesian forms are infinitely differentiable, and let us vary the translational speed targets (velocity norm in the lateral plane) in two axes.

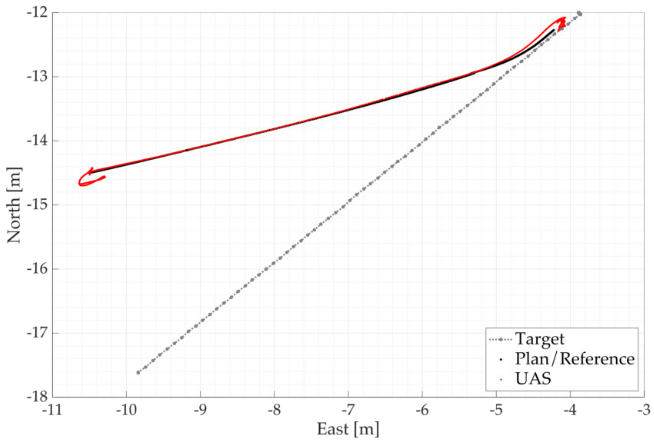


Fig. 6: Top-down (North-East) view of the docking experiment. The target and the UAS trajectories begin on the left.

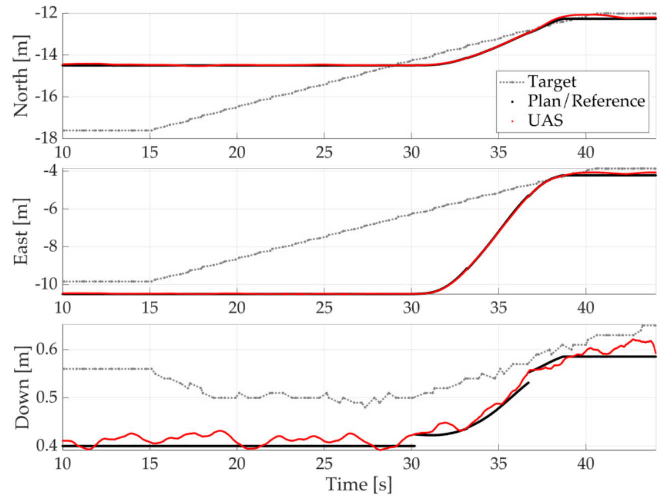


Fig. 7: Docking with a moving target by planning a smooth trajectory towards its projected (future) location.

In Figure 4 we show the North-component of the trajectories executed by the UAS outdoors in flying a reference circle of fixed radius and increasing angular rates. The vehicle is commanded peak lateral accelerations of almost $10\,\mathrm{m/s^2}$. We present results from three evaluations performed under a 2–$3\,\mathrm{m/s}$ wind from North: using only position and velocity references in a classical feedback style (Fig. 4a), incorporating trajectory feedforward (Fig. 4b), and finally with the full LQG system (Fig. 4c). As expected, without the feedforward elements, the system lags behind in time with increasing angular rates. This behaviour is exacerbated when flying outdoors and external disturbances push the system away from a desired path. With feedforward enabled, we see that the tracking is more accurate and shows negligible lag. However, without compensating for external disturbances, the UAS trajectory has an upward shift (more prominent around $30\,\mathrm{s}$). This is counteracted when bias compensation is enabled. Figure 1 shows a blended view of these aggressive trajectories with the UAS at a high lean angle.

Figure 5 also shows a histogram representation of the lateral position and velocity tracking errors seen in Figure 4. From the distribution, we see that the position errors (top) for a simple feedback system can fall between 0.75–$2\,\mathrm{m}$. When feed-forward and bias compensation from LQG are applied, the errors are reduced to less than $0.2\,\mathrm{m}$. An interesting artifact of losing phase-tracking can also be seen in the velocity distributions when no feedforward is available.

C. Aerial Docking

Finally, we demonstrate an ultimate performance objective of the UAS in outdoor applications by tracking and predicting a future location of a moving target platform to dock with it in flight. In-flight docking is extremely challenging for multirotors due to a variety of safety and mechanical constraints. In this problem, we assume only that the target is moving in a predictable path (is not evasive), and that some intermittent observations of the target are available through its GPS data. To aid a fast recovery and accurate state estimation of the target, we also equip it with a passive fiducial marker that can be observed by an onboard camera in close approaches (<2–$3\,\mathrm{m}$). The full pipeline presented here is employed for UAS control, but the relative pose estimation for the target over a horizon is accomplished by fusing these complementary modalities of information. This lets us plan (and replan) a smooth and efficient trajectory towards this projected final location, and engage a mechanical actuator to dock. Detailed and in-depth evaluations under various outdoor scenarios are available [21]; here we focus on path following capabilities.

Figure 6 shows the top-down (North-East) view of the target's path, and the interception plan generated and executed by the UAS. In this particular instance, the target is a zipline system that moves in a straight-line in the lateral plane, but affects a parabolic sag in the vertical axis. We see that the planned path meets the target's path at the highlighted region, and that the UAS also executes it correctly.

A temporal view of the same experiment is shown in Figure 7 for all three axes. The actual successful docking occurs at around the $38\,\mathrm{s}$ mark, and the UAS starts its path around $30\,\mathrm{s}$ (prior to that, observations are being collected to estimate the target's trajectory). Once again, we see that the UAS follows the reference trajectory precisely in space *and* time, which is crucial for a planned time-critical missions. Also note that the scale on 'Down' axis has more than 10x magnification; the overall 3D accuracy in hover is $\approx 4\,\mathrm{cm}$.

V. Conclusions & Remarks

We have presented a complete framework, *Freyja*, that sequentially addresses each aspect of a multirotor flight in real and challenging outdoor environments. The full open-source pipeline is structurally modular, incorporates several optimal methods from the literature to enable precise maneuvering in agile flight maneuvers, and is amenable to extension as the state of the art progresses. For instance, while *Freyja*'s state-space representation of Eqn 1 for the controller enables easy integration of 3D path planners, it currently precludes acrobatic trajectories in the rotational space (such as flips and inverted flight). Our extensive field results demonstrate the capabilities of the system in rejecting environmental disturbances and precisely executing time-critical trajectories.

References

[1] A. Islam, A. L. Houston, A. Shankar, and C. Detweiler, “Design and Evaluation of Sensor Housing for Boundary Layer Profiling Using Multirotors,” *Sensors*, vol. 19, no. 11, p. 2481, Jan. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/11/2481

[2] D. Anthony, S. Elbaum, A. Lorenz, and C. Detweiler, “On crop height estimation with UAVs,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 4805–4812.

[3] J.-P. Ore and C. Detweiler, “Sensing water properties at precise depths from the air,” *Journal of Field Robotics*, vol. 35, no. 8, pp. 1205–1221, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21807

[4] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, “Supporting wilderness search and rescue using a camera-equipped mini UAV,” *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 89–110, 2008. [Online]. Available: https://www.onlinelibrary.wiley.com/doi/abs/10.1002/rob.20226

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

[6] ——, “Dynamic path generation for multirotor aerial docking in forward flight,” in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 1564–1571.

[7] M. Hua, T. Hamel, P. Morin, and C. Samson, “Introduction to feedback control of underactuated VTOLvehicles: A review of basic control design ideas and principles,” *IEEE Control Systems Magazine*, vol. 33, no. 1, pp. 61–75, Feb. 2013.

[8] R. Mahony, V. Kumar, and P. Corke, “Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor,” *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, Sep. 2012, conference Name: IEEE Robotics Automation Magazine.

[9] P. Foehn, D. Falanga, N. Kuppuswamy, R. Tedrake, and D. Scaramuzza, “Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload,” *Robotics: Science and Systems Foundation*, 2017.

[10] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, “Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue,” *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 46–56, Sep. 2012.

[11] I. Sa, S. Hrabar, and P. Corke, “Outdoor Flight Testing of a Pole Inspection UAV Incorporating High-speed Vision,” in *Field and Service Robotics: Results of the 9th International Conference*, ser. Springer Tracts in Advanced Robotics, L. Mejias, P. Corke, and J. Roberts, Eds. Cham: Springer International Publishing, 2015, pp. 107–121. [Online]. Available: https://doi.org/10.1007/978-3-319-07488-7_8

[12] A. Borowczyk, D.-T. Nguyen, A. P.-V. Nguyen, D. Q. Nguyen, D. Saussi, and J. L. Ny, “Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle,” *CoRR*, vol. abs/1611.07329, 2016. [Online]. Available: http://arxiv.org/abs/1611.07329

[13] 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, pp. 3130–3136.

[14] “GPS Accuracy: United States.” [Online]. Available: https://www.gps.gov/systems/gps/performance/accuracy/

[15] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, “Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering,” in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 3277–3282.

[16] J. Schneider, C. Eling, L. Klingbeil, H. Kuhlmann, W. Frstner, and C. Stachniss, “Fast and effective online pose estimation and mapping for UAVs,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4784–4791.

[17] 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.

[18] M. W. Mueller, M. Hehn, and R. D’Andrea, “A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.

[19] J. Thomas, G. Loianno, J. Polin, K. Sreenath, and V. Kumar, “Toward autonomous avian-inspired grasping for micro aerial vehicles,” *Bioinspiration & Biomimetics*, vol. 9, no. 2, p. 025010, May 2014. [Online]. Available: http://stacks.iop.org/1748-3190/9/i=2/a=025010?key=crossref.10e18b0b3991c32e69cef281fde8eee5

[20] M. Faessler, A. Franchi, and D. Scaramuzza, “Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2017.

[21] A. Shankar, S. Elbaum, and C. Detweiler, “Multirotor Docking with an Aerial Platform,” in *International Symposium on Experimental Robotics (ISER)*, [To Appear], 2020.

[22] “Intel RealSense T265.” [Online]. Available: https://www.intelrealsense.com/tracking-camera-t265/

[23] J. Delmerico and D. Scaramuzza, “A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2502–2509.

[24] A. Shankar, S. Elbaum, and C. Detweiler, “Towards in-flight transfer of payloads between multirotors,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6201–6208, 2020.