Unnamed Systems, Vol. 14, No. 1 (2026) 181–200

DOI: 10.1142/S2301385024500390

UNMANNED SYSTEMS
https://www.worldscientific.com/worldscinet/us

Collaborative Unmanned Aerial Package Pickup and Delivery System

Stephen Haviland [*,||], Dmitry Bershadsky [*,**], Jintasit Pravitra [†,††], Takuma Nakamura [‡,‡‡], Eric Johnson [§,§§], Thanakorn Khamvilai [¶,¶¶]

[*] *optimAero, Tehachapi, CA 93561, USA*

[†] *Department of Mechanical Engineering, Navaminda Kasatriyadhiraj Royal Air Force Academy, Saraburi 18180, Thailand*

[‡] *Independent Researcher, Seattle, WA 98109*

[§] *Department of Aerospace Engineering, The Pennsylvania State University, University Park, PA 16802, USA*

[¶] *Department of Mechanical Engineering, Texas Tech University, Lubbock, TX 79409, USA*

A system using aerial vehicles to autonomously locate and retrieve ground packages of various colors and shapes within a designated area was developed. This system was notable for its use of an innovative rotor configuration that offered a higher degree of control compared to traditional designs. To accurately identify and track the packages, a multi-sensor approach combining vision and Light Detection and Ranging technologies was implemented, integrating data into an extended Kalman filter framework for precise position and velocity estimates. An electro-permanent magnet was employed to securely grab and release the packages, which have a ferrous material. The process of path planning and collision avoidance was conducted in a decentralized manner, leveraging a shared global map among the airborne vehicles. This paper details system technical design, including the integration of various technologies, and shares insights and outcomes derived from its application.

Keywords: Unmanned aerial system; package pickup and delivery; autonomous collaboration; simulation; system integration.

1. Introduction

The development of unmanned aerial vehicles (UAVs) has surged in the past 20 years, driven by affordable micro-electro-mechanical systems (MEMS) and powerful compact computers. Companies like Zipline [1], Amazon [2], Google, DHL and UPS have leveraged this technology to enhance package delivery systems [3, 4]. Drones offer faster delivery with reduced human involvement, though several technical challenges still need to be addressed.

This paper examines a team of one to four UAVs collaborating to search an arena for ground objects, pick them up, and deliver them to specified locations. The package delivery challenge encompasses several complex areas of ongoing research, including multi-UAV collaboration in outdoor environments, visual detection and tracking of ground objects, autonomous landing, grasping, releasing and carrying payloads up to 2 kg. The UAVs must perform these tasks autonomously in a dynamic outdoor environment with varying sun and wind conditions.

The search for ground targets relies on image processing techniques. Once a target is detected, typically by a camera

Received 14 May 2024; Accepted 6 November 2024; Published 12 December 2024. This paper was recommended for publication in its revised form by editorial board member, Jose Martinez-Carranza.
Email Addresses: ||shaviland@optim.aero, **dbershadsky@optim.aero, ††jintasit_p@rtaf.mi.th, ‡‡takuma.nakamur@gmail.com, §§eric.johnson@psu.edu, ¶¶thanakorn.khamvilai@ttu.edu
¶¶ Corresponding author.

due to known visual features, the system must determine the object's location and execute a landing despite wind conditions. Upon landing, the vehicle must grasp the target and transport it to the destination, posing another location challenge. The UAVs are designed and tuned to fly for approximately 30 min (a self-imposed requirement) and carry up to 2 kg of payload, a difficult task given their small size. Even if this capability is demonstrated in a single vehicle, the collaborative nature of the challenge requires duplicating the system. Variations in hardware, manufacturing tolerances, settings and software make it challenging to replicate a functioning prototype. Collaborative search requires multiple UAVs to share information, autonomously determine search strategies, and execute them without collision. The systems described in this paper are developed for a Mohamed Bin Zayed International Robotics Challenge competition [5–8] where the mission is illustrated in Fig. 1.

The requirements and our design choices are

(1) **Autonomous Multi-UAV Collaboration**

- *Requirement*: The system needs to handle collaboration between one to four UAVs for searching, picking and delivering packages.
- *Design Choice*: We implemented a decentralized approach, leveraging a shared global map among airborne vehicles to avoid collisions and redundant searches. Each UAV autonomously determines its search strategy and executes it independently.

(2) **Visual Detection and Object Tracking**

- *Requirement*: The UAVs must reliably detect and track ground objects in an outdoor environment with varying conditions.
- *Design Choice*: A multi-sensor approach combining vision and LIDAR was implemented, integrating data into an Extended Kalman Filter (EKF) for precise position and velocity estimation. The vision system detects packages using color filtering and contour detection.

(3) **Grasping and Payload Handling**

- *Requirement*: The UAVs must autonomously land, grasp, release and carry payloads of up to 2 kg.
- *Design Choice*: The UAVs use an electro-permanent magnet (EPM) mounted on a soft plate to grab and release ferrous packages.

(4) **Endurance**

- *Requirement*: The UAVs should have an endurance of approximately 30 min with a payload.
- *Design Choice*: The vehicle is powered by a 6S (22.2 V) 16000 mAh battery. The system is optimized to hover for 30 min and carry the payload for 5 min using a Tarot 680 frame with 700 KV motors.

(5) **Collision Avoidance**

- *Requirement*: The system must avoid collisions during multi-UAV operations.
- *Design Choice*: Priority assignment and avoidance guidance rules were implemented, with a focus on priority-based navigation. Velocity adjustments are made for lower-priority UAVs to prevent collisions.

(6) **Map Sharing and Search Strategy**

- *Requirement*: UAVs need to explore the environment efficiently without duplicating efforts.
- *Design Choice*: A grid map system is used, where each cell is marked as explored, unexplored, or ignored. UAVs share map updates with others. Waypoint generation and reassignment are optimized using market-based coordination principles.

(7) **Stability while Performing Maneuvers**

- *Requirement*: UAVs must maintain stability during flight, especially when grasping and carrying payloads.
- *Design Choice*: We implemented direct force control (DFC), where the motor tilt angles allow the UAV to

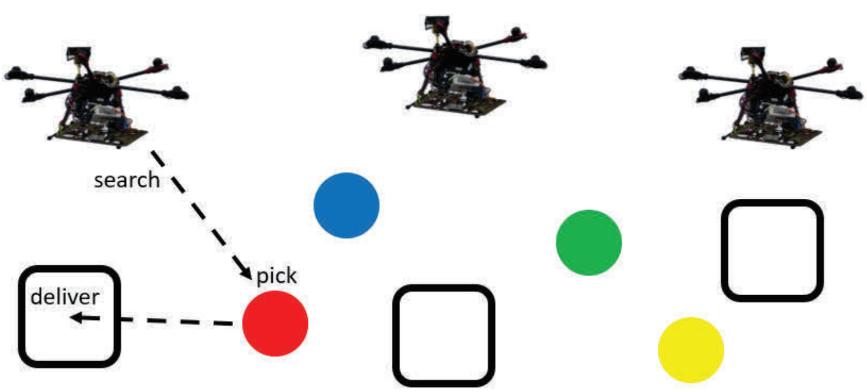


Fig. 1. Mission overview: A team of aerial robots collaboratively performs searching, picking and delivering packages. This represents a core capability for urban package delivery and aerial manipulation in general.

control all six degrees of freedom. This enables precise control over the vehicles position and attitude.

(8) **Autonomous Takeoff, Landing and Delivery**

- *Requirement*: UAVs must accurately detect the drop-off bin and deliver packages autonomously.
- *Design Choice*: The system uses LIDAR for centering the UAV during descent and detecting the drop-off bin. A GPS is used for navigation, and the perception system handles bin detection during the final approach.

For detailed technical approach, the system architecture and vehicle design are described in detail in Secs. 2 and 3, respectively. The autonomy that enabled package delivery is described in Sec. 4. Experiments and competition results are discussed in Secs. 5 and 6, respectively.

1.1. *Previous work*

Many of the tasks essential to the mission can be research topics, including controller architecture for variable mass and unknown CG offsets, target detection and tracking, multi-vehicle operations and vehicle sizing and propulsion optimization. A UAV is considered fully autonomous if it can successfully complete its assigned mission without human intervention or external assistance, while adapting to operational and environmental conditions [9].

The package pickup and delivery task requires the reliable detection, tracking, grasping and release of ground objects in an outdoor environment. Common sensors used for object detection include vision, laser and radar. While radar sensors are typically large and expensive, visual sensors are smaller and more affordable but demand greater processing power. Light Detection and Ranging (LIDAR) sensors offer lower computational costs but can be heavy and expensive depending on the model.

In [10], a review of combined sensors (vision/laser) for detecting and tracking multiple objects (DATMO) was presented, primarily focused on larger objects like people and vehicles. Detecting smaller objects presents a greater challenge, but the problem could benefit from a multi-sensor approach, leveraging the complementary strengths of different sensors. For example, at higher altitudes where laser scanners may be less reliable, visual sensors could detect objects. As the UAV descends, the laser scanner's precision becomes advantageous, providing better object location measurements when cameras may struggle due to partial occlusion. Cameras have proven effective for object detection and tracking, but visual-based autonomous landings on static and moving objects remain an active research area. Standard helipads have been detected and tracked using methods like Hu moments and Haar-like features [11]. For object tracking and estimating their locations, [12] employed a multi-EKF to track multiple objects and enable landings on moving targets.

Aerial vehicles offer the potential for faster and more cost-effective delivery compared to ground transportation. Developing systems that can reliably grasp and release packages is an ongoing research area. Various approaches have been explored, including mechanical grabbers [13], suction systems [14] and magnetic solutions [15]. For packages with ferrous surfaces, EPMs are advantageous due to their selectable operational state and low power consumption. The designed system uses an EPM mounted on a custom soft-mounted plate, enabling the grasping of objects even at nonlevel attitudes.

The challenge of simultaneously mapping an area and searching for objects has been a recurring focus in UAV competitions since the early 2000s, such as the UAV Outback Challenge [16] and the International Aerial Robotics Competition [17]. This problem remains relevant, particularly in scenarios involving multiple UAVs working collaboratively. Multi-vehicle exploration is faster and more robust than single-vehicle approaches [18]; if one vehicle fails, others can continue the mission [19, 20]. However, multi-vehicle operations often face challenges with data communication links [21], particularly when sharing vehicle states and maps. Accurate vehicle state sharing is crucial for collision avoidance, and issues like time delays and temporary communication outages must be effectively managed. Maintaining a consistent and updated global map can also be complex, depending on the mission and environment.

Various methods for map sharing in simultaneous localization and mapping (SLAM) are reviewed in [18]. Frontier-based multi-robot exploration using shared occupancy maps is discussed in [20, 22], where occupancy maps are used to calculate the cost-to-go values for reaching cells. In [19], a collaboratively generated terrain map is later employed to develop a pursuit-evasion strategy. An effective exploration strategy is as crucial as the mapping algorithm itself. Methods based on market economy principles to enhance multi-robot exploration strategies are presented in [23] and [24]. In our approach, map sharing is solely aimed at avoiding redundant searches, with the map only tracking whether a cell has been explored.

To enable aerial vehicles to carry large payloads while maintaining extended endurance, careful consideration of propulsion sizing is essential. Additionally, factors such as transporting multiple vehicles across seas and compensating for significant wind conditions must be addressed. Currently, there are few general, rigorous and comprehensive design methods for electric propulsion systems. Many existing methods are either too specific to be applicable to general electric vehicles or tailored to standard or nonstandard geometries. For larger, turbine-powered, manned

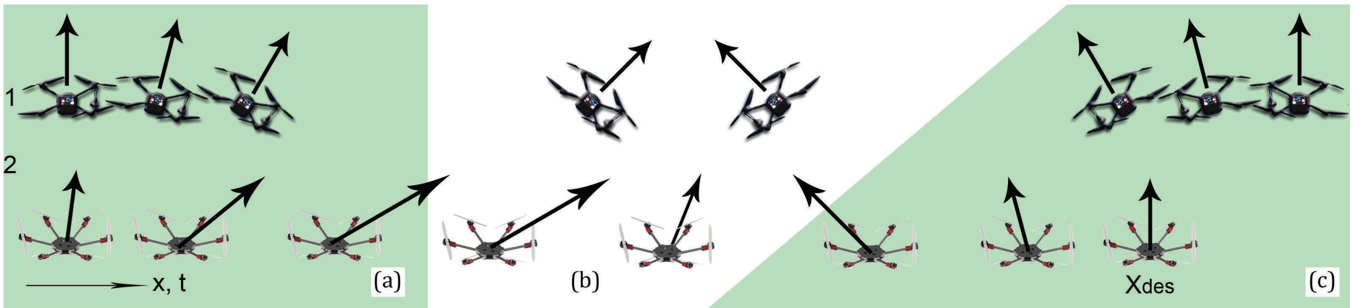


Fig. 2. Planar multirotor (1) versus a higher degree-of-freedom multirotor (2) in moving to a new desired horizontal position (x_{des}). Regions (a)–(c) are at different times during the maneuver with arrows indicating the force vector used to achieve it. As a result, attitude and position can be effectively controlled independently.

helicopters, the required fuel fraction (RF) method [25, 26] has been used for years, though it is suited for gas-/jet-fueled aircraft. However, a comparable method for sizing electric vehicle drive components is lacking in the literature. Several rigorous studies on propulsion system design have been presented. Some use blade element and momentum theory (BEMT) for an iterative selection of motors, propellers, and batteries [27]. Multi-disciplinary optimization (MDO) methods have also been explored for designing propulsion systems [28], aiming to meet performance goals such as loiter time and rate of climb while analyzing motor and battery requirements.

Standard multirotor platforms are underactuated because their thrusters are aligned. To move from one position to another, the vehicle must command a torque to adjust the orientation of the overall thrust vector, as illustrated in Fig. 2. Smaller multirotors can track trajectories more easily due to their higher agility [29]. However, as vehicles grow larger, precise position control becomes more challenging because position and attitude control are coupled in underactuated systems. Higher degree-of-freedom (DoF) multirotors, such as vehicle 2 in Fig. 2, can generate forces in additional directions, allowing them to decouple position and attitude control. This capability enables the vehicle to translate without rotating and results in quicker position changes, as shown in Fig. 2(a). While most higher DoF multirotors can only generate a limited horizontal body force compared to vertical force, they can achieve greater horizontal acceleration by tilting, as depicted in Fig. 2(b). During deceleration, however, a planar vehicle must execute more aggressive maneuvers compared to a higher DoF vehicle, which often leads to overshooting the target position (x_{des}), as seen in Fig. 2(c). Higher DoF vehicles can also perform the same maneuvers while flying at nonlevel attitudes. Additionally, they offer the advantage of maintaining position and orientation in windy conditions. Conventional planar multirotors must fly at nonlevel attitudes to compensate for disturbances, which can be problematic for high-precision tasks or environmental interactions, especially in strong winds.

Another challenge with under-actuated designs is their limited ability to counteract reaction forces when manipulating objects. Research has explored fully actuated multirotors by adding servos to each arm, allowing for motor orientation adjustments [30–32]. While this increases complexity and reduces survivability, it can be advantageous depending on mission requirements. However, servoed designs often have limitations that restrict orientation and trajectory tracking. To address this, some systems use modified electronic speed controller (ESC) firmware and symmetrical propellers to create omnidirectional vehicles [33, 34]. These vehicles, while offering enhanced tracking capabilities, may have reduced efficiency and increased power requirements for hovering due to nonconventional rotor orientations.

Another approach fixes rotors to produce horizontal force components [35], which improves functionality for level flight and hovering in nonlevel orientations without adding complexity. Though there is some efficiency loss, the added tracking ability can be beneficial. Under windy conditions, tracking and landing on moving objects become challenging if constant attitude adjustments are needed. The rotor orientations in this design are optimized for lateral movement, considering the self-imposed operational requirement of picking up discs in strong winds [36].

2. System Architecture

The system architecture, illustrated on the right side of Fig. 3, enables UAVs to handle all perception, planning and control tasks autonomously. On the left side, ground control stations serve as relays, sharing a global map and providing telemetry for vehicle collision avoidance. While the scenario involves three vehicles, the system is scalable and can

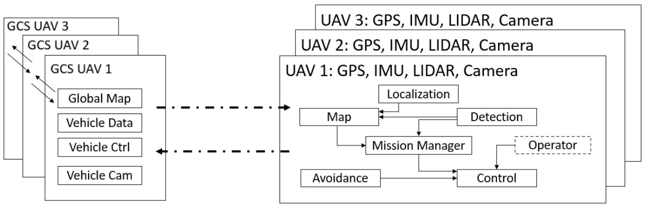


Fig. 3. An illustration of system architecture.

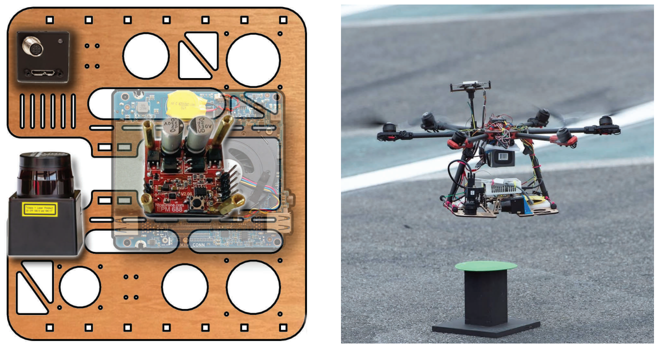
operate individually or on a larger scale. In summary, the system can be described as

- Direct force control (DFC) UAV — vehicle has downward facing camera (Pointgrey Blackfly), LIDAR (Hokuyo URG04LX), Inertial Measuring Unit (IMU), Global Positioning System (GPS), barometer, magnetometer, electro-permanent magnet (EPM) and contact switch. Sensor data are fused to estimate object positions on a global map. The ground station provides updates to the map and other vehicle data for collision avoidance. Additionally, the operator can optionally control each UAV.
- Wireless communications — each UAV has an outdoor 5.8 GHz WiFi router that is set up in bridge mode to send and receive information from the ground station.
- Ground control station (GCS) — single or daisy-chained computers on the ground act as relays for sending and receiving all vehicles and map information. The global map is displayed on the GCS where an operator can interact with each vehicle.

The communication between UAVs and GCSs is designed to minimize onboard communication load. Each UAV communicates only with GCS 1, while the GCSs are connected in a daisy-chain fashion — GCS 1 communicates with GCS 2, GCS 2 with both GCS 1 and GCS 3, and so on. This setup allows data to be relayed from UAVs to the last GCS without excessive direct communication.

Each airborne vehicle faces two perception challenges: detecting and tracking ground-based objects, and detecting the drop-off bin. The drop-off bin is $1\,\mathrm{m}^2$ with known coordinates. Depending on the choice between single-point or real-time kinematic (RTK) GPS, detection of the bin may or may not rely on GPS, as standard GPS lacks precision. Although affordable RTK solutions exist, they often prove unreliable, so a single-point GPS receiver is used. Consequently, the perception module must be capable of detecting the drop-off bin after an object is grabbed.

Figure 4(a) shows the base plate with sensor mountings used on each vehicle. The EPM is positioned at the vehicle's center of gravity (CG) to minimize the impact of mass and CG changes when lifting heavy targets, which can weigh up to 2 kg (over 50% of the vehicle's gross takeoff weight). The EPM is mounted on four loose rails, allowing it to adjust its attitude, which is useful when the EPM and package make contact at an angle. Two electrical switches beside the EPM inform the autopilot when the package is grabbed and released. The camera is offset by 225 mm to avoid being obstructed by the carried package, enabling it to detect the drop-off bin. However, this offset causes the camera to lose sight of the target at lower altitudes. To mitigate this, a downward-facing LIDAR centers on the target during the descent, and a large section of the baseplate is removed to improve the LIDARs field of view. A 5.8 GHz WiFi router is mounted directly on the baseplate with antennae pointing upward. The vehicle layout is shown in Fig. 4(a).



(a) (b)

Fig. 4. (a) Top-down view of vehicle baseplate with sensor configuration including camera, LIDAR, computer and magnet (b) DFC vehicle with baseplate mounted executing a package pickup, the package is tracked with LIDAR and camera.

An Intel NUC i7 runs all perception, in-house guidance, navigation and control (GNC) software, and interfaces with the sensor board. The sensor board, a Pixhawk modified with custom firmware for new motor mixing logic, functions as a sensor and motor interface in 'Auto' mode. In 'Manual' mode, the original Pixhawk code is used. Pixhawk is an open-source autopilot popular with hobbyists, known for its customizable software and hardware.

Two perception modules onboard the NUC handle detection tasks. Laser data are processed at 10 Hz, matching the frequency of above-ground-level (AGL) and object estimates. The camera outputs 320×240 images at 30 Hz, with target tracking running at the same rate. The Pixhawk sends IMU (100 Hz), barometer (25 Hz), magnetometer (25 Hz), and switch (25 Hz) data via USB to the NUC for GNC and mission manager updates. GPS data (10 Hz) connect via USB-UART. All sensor data are fused into a single navigation solution using an EKF [37].

A contact switch, an analog voltage button, detects whether an object has been grabbed or released, as the EPM lacks feedback on its 'On' or 'Off' state. The mission manager updates the planner, which sets new position and velocity commands for the controller to track. The controller outputs three torque and three force commands at 100 Hz, sent to the Pixhawk autopilot. Rate feedback for the torque commands is processed onboard the Pixhawk at 400 Hz to minimize latency between commands and gyroscope data. This architecture was initially developed in [38], proving effective with low-cost MEMS devices.

3. Direct Force Control Hexarotor: Design, Sizing and Control

A vehicle capable of controlling all six-DoF would offer significant advantages. Precise control is crucial for accurate positioning, especially during touchdown on an object. Upon contact, reaction forces and moments may arise that an underactuated vehicle might struggle to counter. Additionally, if the vehicle is not level during touchdown, the grabbing mechanism might fail. With DFC, the motor tilt provides added DoF, enabling the vehicle to maintain level flight even in windy conditions or when the C.G. is offset.

A vehicle capable of carrying the necessary avionics, an additional 2 kg payload, and achieving up to 30 min of endurance is desired. In our previous research [39], we developed a tool for sizing multirotors for specific missions, which was used to design this vehicle. The vehicle is sized to take off, hover for 30 min, and carry a 2 kg payload for 5 min, assuming a 100% depth of discharge. The large payload represents the worst-case scenario in terms of power requirements. The optimization algorithm's output, shown in Table 1, led to the selection of the Tarot 680 frame with Tiger Motors U3-700 KV. The battery chosen is a 6S (22.2 V) 16000 mAh 15C GensAce Tattu. The optimization software also provided an estimated motor configuration, recommending 6S, even though the motors are rated for 3-4S (11.1–14.8 V). The fully enclosed motors are advantageous for operation in sandy environments.

The vehicle is designed as an underactuated hexarotor, with modifications to enable DFC. Instead of creating a chassis from scratch, we chose the Tarot 680 frame and adapted it for DFC. The motor tilt angles were determined to maintain position in known wind conditions. Previous work [36] optimized motor orientation angles to maximize horizontal acceleration limits, which we enhanced with updates to propulsion, aerodynamics and dynamics modeling. Figure 5(b) shows how changing motor tilt affects body-axis acceleration. As tilt is added, body x and y acceleration capabilities increase, while body z decreases. With a tilted hexarotor, symmetry is lost, causing different capabilities in the x- and y-directions, though symmetry can be restored with eight rotors. However, a trade-off exists with fixed motor tilt: hovering power increases because thrust vectors are no longer planar. Despite this, rotor downwash interactions slightly improve overall efficiency [36]. Some DFC-like vehicles use servos to tilt rotors only when needed [32], but this introduces complexity. Our motor tilt design is robust to wind conditions.

Assuming the attitude control system can maintain a level vehicle, Eq. (1) represents the forces acting on the vehicle in the horizontal plane. For steady-level hover, the DFC forces must counteract the wind drag forces. Using Eq. (2) with estimated area (S), drag coefficient (C_D), and accounting for a 38 kph wind (doubled for a safety margin),

Table 1. Optimizer output for rubber propulsion system requirements 30-min hover including 5-min hover with additional 2 kg payload compared to the built design.

Parameter	Value (rubber)	Value (built)
Battery configuration	6S	6S
Battery capacity	17423 mAh	16000 mAh
Propeller diameter	14”	13”
Propeller pitch	4”	4.4”
K_v	658 RPM/V	700 RPM/V

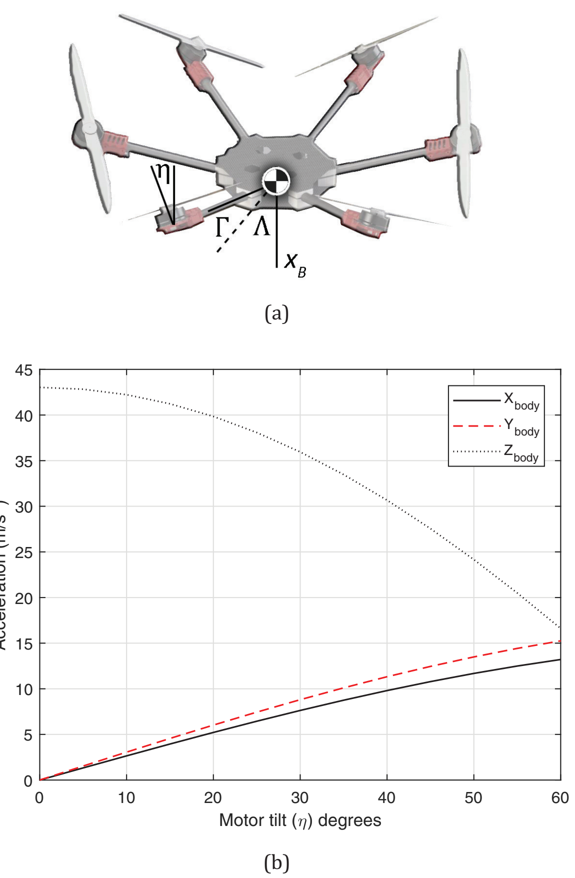


Fig. 5. (a) The fixed angles defining the DFC motors and (b) Optimization routine to get maximum body acceleration values while changing motor tilt (η).

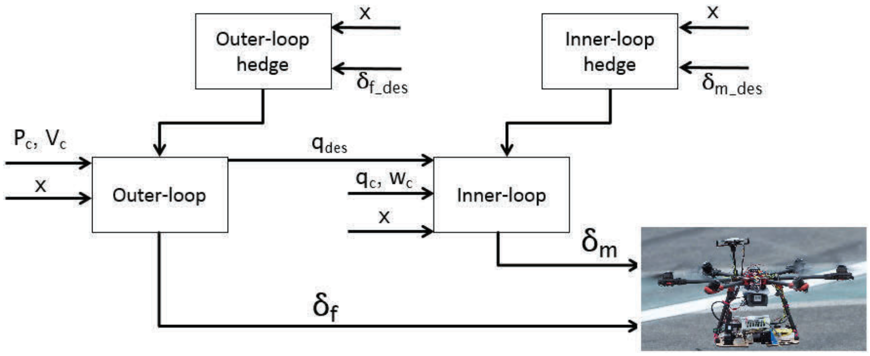


Fig. 6. High-level block diagram of the model inversion controller; outer loop controls position, inner loop controls attitude, daisy-chain architecture manages to interconnect between them.

the required motor tilt angle is approximately 30°.

$$m\ddot{x} = F_{dfc} + D, \tag{1}$$

$$F_{dfc} = \frac{\rho v_{\text{wind}}^2 S C_D}{2}. \tag{2}$$

3.1. *Daisy chain control*

By rotating the motors into a fixed but nonplanar orientation, the multirotor gains additional degrees of freedom, enabling it to translate without rotating or rotate without translating. DFC-like designs allow for direct body acceleration commands in the horizontal plane, effectively adding two extra control inputs. However, as shown in Fig. 5(b), the horizontal body acceleration inputs are typically much smaller than the vertical ones. To utilize these DFC inputs effectively, modifications to our standard controller [40] were necessary. Figure 6 illustrates the high-level architecture of the controller. The outer-loop controller generates acceleration commands ($\delta_f = [dfc_x, dfc_y, \delta_{\text{thr}}]$) and attitude commands (q_{des}) based on the error between the trajectory generator's position and velocity commands (P_c, V_c) and the estimated states ($x = [p^{\mathrm{T}} \quad v^{\mathrm{T}} \quad q^{\mathrm{T}} \quad \omega^{\mathrm{T}}]^{\mathrm{T}}$). These commands are then processed by the inner-loop controller, which generates moment commands ($\delta_m = [\delta_{\text{ail}}, \delta_{\text{ele}}, \delta_{\text{rud}}]$).

If the outerloop controller demands larger horizontal acceleration commands than the vehicle's capabilities, the vehicle's attitude must be adjusted. The daisy chain approach [41] addresses this by adjusting the outer-loop control to account for the vehicle's horizontal acceleration limits. If necessary, it uses the available DFC actuator limits and requests a change in attitude to meet the acceleration command. This daisy chain scheme is implemented on top of the adaptive controller architecture described in [40]. The adaptive control addresses actuator dynamics and saturation issues, and pseudocontrol hedging [42] is employed to modify both inner and outer loop dynamics.

Algorithm 3.1. Daisy-Chain acceleration command modification

```
1: dfc_x ← a_des_x, dfc_y ← a_des_y
2: δ_h ← sqrt(dfc_x^2 + dfc_y^2)
3: if δ_h > dfc_max then
4:     dfc_x ← (dfc_x dfc_max)/δ_h, dfc_y ← (dfc_y dfc_max)/δ_h
5: end if
6: ā_des_x ← a_des_x − dfc_x, ā_des_y ← a_des_y − dfc_y
```

This technique helps the system adapt even when facing input saturation.

According to the Daisy Chain Algorithm 3.1., the outerloop acceleration commands ($\bar{a}_{\text{des}}$) are adjusted so that all horizontal acceleration commands utilize the available DFC limits, if necessary. If there are still remaining horizontal acceleration commands, the vehicle must adjust its attitude to accommodate them. To convert the remaining acceleration commands into attitude commands, a similar algorithm to the one described in [43] is used.

4. Enabling Autonomy

4.1. *Collision avoidance*

Collision avoidance is crucial for the safety of multi-vehicle operations. Our approach comprises two components: priority assignment and avoidance guidance. Priority assignment is inspired by right-of-way rules commonly used in air transportation, where less maneuverable aircraft are given priority. In our system, an aircraft performing a higher-priority task is granted right-of-way. For instance, an aircraft heading toward a drop-off bin is prioritized over one engaged in a search mission. When two aircraft have the same priority, a unique 'tail number' is used to break the tie.

The priorities for determining right-of-way are as follows, with higher numbers indicating higher priority:

(5) Executing a pickup attempt
(4) Dropping off an object
(3) Has the target and is flying toward the drop bin
(2) Taking off
(1) Searching

Once priorities are assigned to all aircraft, those with lower priority will adjust their velocity commands to avoid aircraft with higher priority. If multiple avoidance maneuvers are required, the velocity commands will be combined. The following logic is applied when an aircraft needs to avoid a higher-priority aircraft.

We first try to encapsulate the idea that aircraft do not need large horizontal separation if there is already a sufficient altitude separation. Define the altitude difference between an ownship and a higher priority aircraft to be $\Delta h = h_{\mathrm{own}} - h_{\mathrm{high}}$. A separation cylinder of radius Δr_{sep}, and half-height Δh, form a cylinder inscribed inside an ellipsoid of parameters Δr_{safe} and Δh_{safe}.

$$\Delta r_{\mathrm{sep}} = \Delta r_{\mathrm{safe}} \sqrt{1 - \left(\frac{\Delta h}{\Delta h_{\mathrm{safe}}}\right)^2}. \tag{3}$$

The two tuning parameters Δr_{safe} and Δh_{safe} both have physical meanings. The parameter Δr_{safe} is the distance to keep apart if two aircraft are at the same altitude. The parameter Δh_{safe} quantifies altitude separation. If $|\Delta h| > |h_{\mathrm{safe}}|$, the traffic is considered no-factor. Collision avoidance is not needed. Note that Eq. (3) is always valid since $1 - (\frac{\Delta h}{\Delta h_{\mathrm{safe}}})^2$ is strictly positive for all cases where avoidance is necessary.

Let $\mathbf{r_{own}}$ and $\mathbf{v_{own}}$ be the ownship horizontal position and velocity. Let $\mathbf{r_{high}}$ and $\mathbf{v_{high}}$ be the higher priority aircraft's horizontal position and velocity. The states for both aircraft are obtained from EKF-based navigation solutions mentioned in Sec. 2. The higher-priority aircraft states are relayed via the ground station. Additionally, the higher-priority aircraft's position is modified to be $\mathbf{r_{high}} + \mathbf{v_{high}}\Delta t$ instead of only $\mathbf{r_{high}}$ to account for the aircraft's movement during the communication relay. This allows for the prediction of the aircraft's position based on its velocity during the time interval Δt.

The approach speed between ownship and the higher priority aircraft is defined as

$$V_{\mathrm{app}} = -\frac{\Delta \mathbf{r} \cdot \Delta \mathbf{v}}{\|\Delta \mathbf{r}\|}, \tag{4}$$

where $\Delta \mathbf{r} = \mathbf{r_{own}} - \mathbf{r_{high}}$ and $\Delta \mathbf{v} = \mathbf{v_{own}} - \mathbf{v_{high}}$. Positive V_{app} means aircraft are approaching each other, while negative V_{app} means aircraft are flying away from each other.

The allowable approach speed is a function of the separation cylinder radius, the distance between aircraft, and the tuning parameter ω_d, which can be thought of as the desired frequency response.

$$V_{\mathrm{app,all}} = (\|\Delta \mathbf{r}\| - \Delta r_{\mathrm{sep}})\omega_d. \tag{5}$$

If $V_{\mathrm{app}} > V_{\mathrm{app,all}}$, aircraft are approaching each other faster than allowed, and it is necessary to deconflict by changing ownship velocity command by $\Delta \mathbf{v_{avoid}}$.

$$\Delta \mathbf{v_{avoid}} = (V_{\mathrm{app}} - V_{\mathrm{app,all}})\frac{\Delta \mathbf{r}}{\|\Delta \mathbf{r}\|}. \tag{6}$$

The direction $\frac{\Delta \mathbf{r}}{\|\Delta \mathbf{r}\|}$ ensures that aircraft is commanded away from higher priority aircraft. This logic holds true regardless of the sign of V_{app} and $V_{\mathrm{app,all}}$. The algorithm can handle various scenarios. Examples of various cases are shown in Figs. 7(a)–7(e). The overall avoidance algorithm is also summarized in Algorithm 4.1.

The complexity of this algorithm in the worst case is $\mathcal{O}(n)$ where n is the total number of aircraft in operation.

4.2. *Collaborative map and search*

We approach multi-vehicle operations with a strategy that each aircraft will grab the first target it sees. This method [19], summarized in Fig. 8, avoids the need to store and share target states among multiple aircraft.

4.2.1. *Map update*

Each aircraft uses a grid map with a 2×2 ft resolution to cover the arena. Each grid cell can be in one of three states: explored, unexplored or ignored. All cells are initialized as unexplored, except for launch and drop zones, which are marked as ignored. As an aircraft moves through the arena, it marks cells as explored based on its altitude and camera field of view. Ignored zones ensure that the aircraft does not attempt to pick up a target if it is in an ignored area, such as a drop zone. Additionally, if an aircraft fails to pick up a target after multiple attempts, it marks the cell as ignored to prevent other aircraft from attempting to pick up the target. Failures may occur due to various reasons, such as the target falling off or being undetectable. Updated map cells are sent to the Ground Control Station (GCS), which then relays these updates to all other aircraft. Each aircraft updates the shared map when 50 cells have changed or after 2 s, whichever comes first. This approach ensures consistent maps across all aircraft while minimizing communication bandwidth.

With an implementation of a hash map, the complexity of this map update procedure is $\mathcal{O}(1)$.

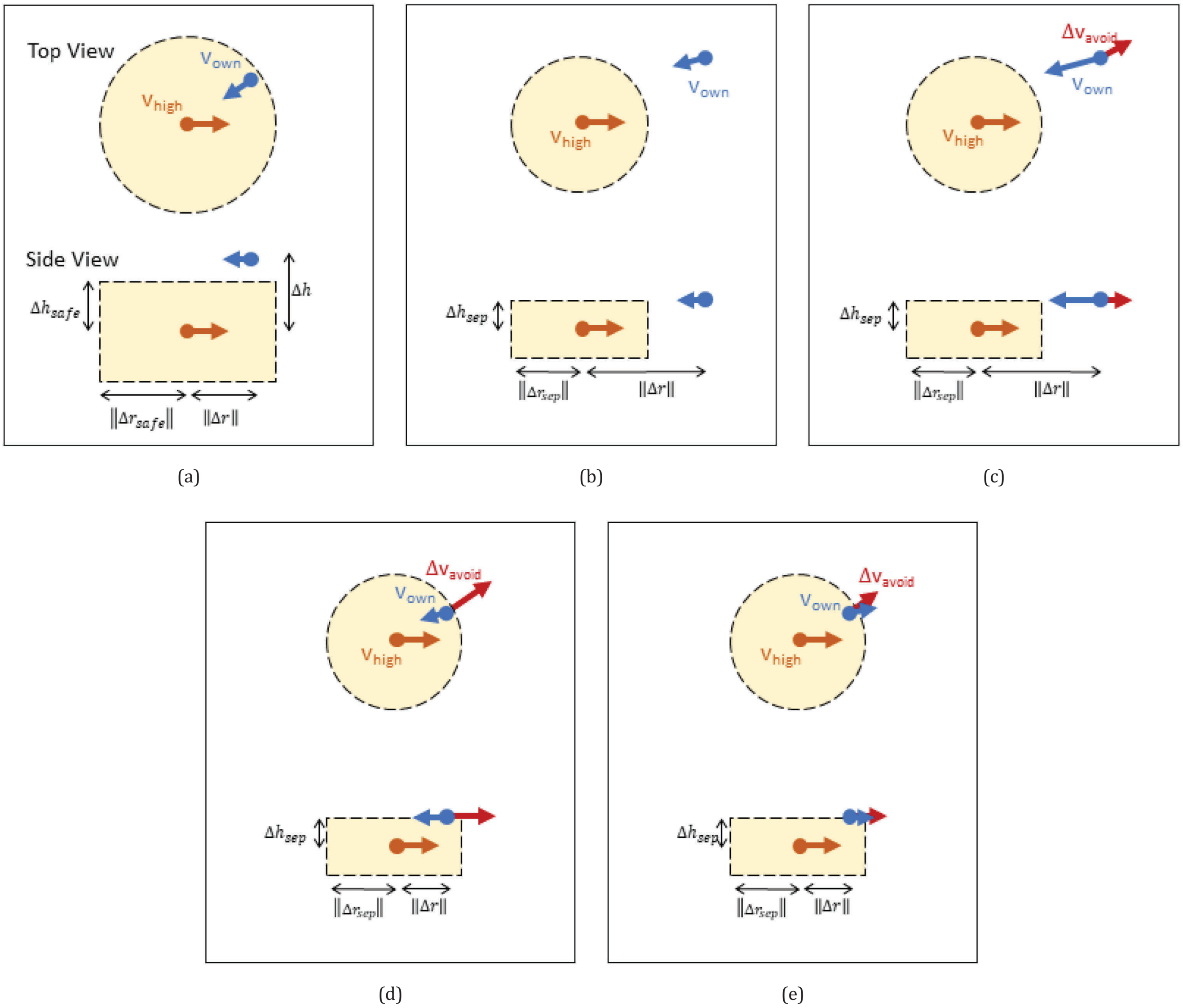


Fig. 7. Top view and side view for different collision avoidance scenarios (a) $|\Delta h| > |h_{\text{safe}}|$. There is a sufficient altitude separation. Avoidance is not needed. (b) $V_{\text{app}} < V_{\text{app,all}}$ and $\|\Delta \mathbf{r}\| - \Delta r_{\text{sep}} > 0$. The approach speed is slow, and the higher-priority aircraft is outside the separation cylinder. Avoidance is not needed. (c) $V_{\text{app}} > V_{\text{app,all}}$ and $\|\Delta \mathbf{r}\| - \Delta r_{\text{sep}} > 0$. The higher-priority aircraft is outside the separation cylinder, but the approach speed is too fast. Avoidance is needed. (d) $V_{\text{app}} > V_{\text{app,all}}$, $V_{\text{app}} > 0$ and $\|\Delta \mathbf{r}\| - \Delta r_{\text{sep}} < 0$. The higher-priority aircraft is inside the separation cylinder. The aircraft are approaching each other. Avoidance is needed. (e) $V_{\text{app}} > V_{\text{app,all}}$, $V_{\text{app}} < 0$ and $\|\Delta \mathbf{r}\| - \Delta r_{\text{sep}} < 0$. The higher-priority aircraft is inside the separation cylinder. The aircraft are already moving away from each other but not fast enough. Avoidance is needed.

4.2.2. *Waypoint generation*

Each aircraft explores the arena using waypoint guidance. A new waypoint is generated randomly. If this waypoint is within an explored area, an ignored zone, or too close to another aircraft, it will be discarded, and a new waypoint will be generated. The aircraft will proceed toward the waypoint until it either encounters a target or reaches the waypoint. If the waypoint is reached without encountering a target, a new waypoint is generated. If a target is found, the aircraft will stop searching and attempt to pick it up. After a successful drop or if several pickup attempts fail, the aircraft will resume search mode.

The complexity of this algorithm in the worst case is $\mathcal{O}(n)$ where n is the total number of grid cells in the map. However, the worst-case scenario is unlikely to happen as it implies that the map has been almost thoroughly explored.

Algorithm 4.1. Collision Avoidance Algorithm

for all other aircraft in operation **do**
  $\boldsymbol{\Delta}\mathbf{v}_{\mathbf{avoid}} \leftarrow 0$
  if Lost right of way to the higher priority aircraft **then**
    if $|\Delta h| > |h_{\text{safe}}|$ **then** ▷ sufficient altitude separation
      Do Nothing
    else
      $V_{\text{app}} \leftarrow$ Calculate the actual approach speed
      $V_{\text{app,all}} \leftarrow$ Calculate the allowable approach speed
      if $V_{\text{app}} < V_{\text{app,all}}$ **then** ▷ the approach speed is within allowable value
        Do Nothing
      else ▷ necessary to modify ownship velocity
        $\boldsymbol{\Delta}\mathbf{v}_{\mathbf{avoid}} \leftarrow \text{getAvoidVelocity}(V_{\text{app,all}}, V_{\text{app}}, \boldsymbol{\Delta}\mathbf{r})$
      end if
    end if
  end if
  $\mathbf{v}_{\mathbf{cmd}} \leftarrow \mathbf{v}_{\mathbf{cmd}} + \boldsymbol{\Delta}\mathbf{v}_{\mathbf{avoid}}$ ▷ Superimpose velocity correction
end for

4.2.3. *Promoting random exploration*

To encourage random exploration and prevent an aircraft from becoming stuck at a location, a new waypoint is automatically generated after a specific timeout. This mechanism ensures that an aircraft does not remain indefinitely at a point due to obstacles or conflicts. The timeout period is set to be longer than the time it would take for the aircraft to fly diagonally across the entire arena at its typical exploration speed. For example, an aircraft could become trapped if it is avoiding another aircraft engaged in a pickup operation, as the stationary aircraft could block its path. The automatic waypoint generation helps to mitigate such situations by continuously redirecting the aircraft, promoting effective exploration of the arena.

A waypoint swap logic has been implemented to enhance efficiency in multi-vehicle operations. Since waypoints are generated randomly, an aircraft might end up crossing another aircraft's path, leading to inefficiencies. Additionally, if a waypoint is closer to another aircraft, the further aircraft should relinquish the waypoint to the closer one. This reassignment logic is managed at the Ground Control Station (GCS), where waypoints are reallocated to the nearest vehicle every second. This approach is inspired by market-based robot coordination [23]. Note that this replanning is intended to optimize search efficiency; if communication is unreliable, each aircraft can still conduct its own search without the reassignment.

Simulation and flight tests indicate that with the correct search altitude, the entire arena can be explored in under

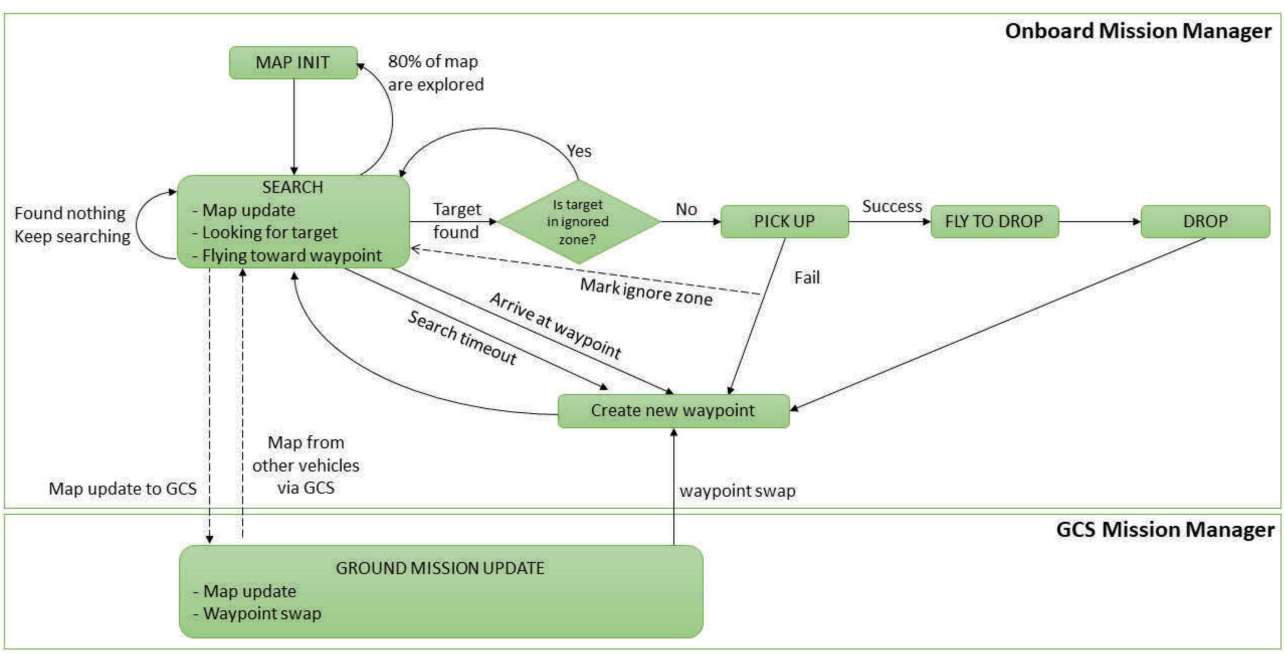


Fig. 8. Mission level state machine for a single vehicle. This architecture enables collaborative package pick-up and delivery.

20 min. However, flight tests also reveal that targets may be missed due to various uncertainties. To address this, the algorithm includes map resets after 80% of the arena has been explored. This 80% threshold is chosen as a balance between aircraft endurance and the likelihood of a target being in the explored area. If a target is missed and remains in the explored area, continuing to explore all remaining unexplored areas could deplete the aircraft's batteries and increase the risk of mid-air collisions, as all aircraft would converge on the smaller unexplored regions.

4.3. *Vision system*

The vision system is responsible for detecting and tracking packages and providing visual feedback for autonomous landing. To achieve this, it generates criteria to initiate, continue, or abort the mission based on the reliability of visual tracking. To detect packages, the system uses a color filter in the Hue, Saturation and Value (HSV) space and constructs a contour tree [44]. The package's dimensions in the image are calculated using camera calibration parameters and the vehicle's above-ground level (AGL) altitude. False positives are filtered out by comparing the detected results with the expected package area and aspect ratio. Once this filtering process is complete, the system retains the 2D measurements of the packages location and size, along with its color label.

The color filtering step involves taking onboard images and applying thresholds based on HSV values. A contour tree is then computed from the binary image, and bounding boxes of the contours are used to size measurement candidates, as shown in Fig. 9(a). Only bounding boxes with valid aspect ratios and areas are considered for measurements in the EKF. The colored diagonal crossings in Fig. 9(b) represent the final measurements, while the rectangles indicate the EKFs estimations. Although the algorithm was intended for both circular and rectangular packages, simulations revealed that measuring the center of rectangular packages is more challenging due to significant changes in bounding box size and aspect ratio with orientation, as depicted in Fig. 10. Figure 11 displays results from a Monte Carlo simulation where the vehicle flew over packages with various initial attitudes and positions. The simulation, which included 7000 samples per experiment, showed that the measurement error covariance for rectangular packages was larger than for circular packages.

The position and the velocity estimates of the package are expressed in the local north-east-down (NED) frame, which is denoted as p^n_{np} and v^n_{np}, respectively. The superscript indicates the coordinate frame, and the subscripts represent the coordinate frames the vector is measured with respect to. For instance p^n_{np} is the position vector of a package (p) with respect to the NED frame expressed in the NED frame. The system also obtains estimated distances to the packages from the method described in Sec. 4.4. The state vector of the EKF is expressed as follows:

$$\hat{x} = [\hat{p}^{\,n}_{np} \quad \hat{v}^{\,n}_{np}]^T, \tag{7}$$

where $p^n_{np} \in \mathbb{R}^3$ and $v^n_{np} \in \mathbb{R}^2$. The positions on the x–y plane (denoted $p^b_{np,x}$ and $p^n_{np,y}$) are propagated using

$$\hat{p}^{\,n}_{np,x}(t+dt) = \hat{p}^{\,n}_{np,x}(t) + \hat{v}^{\,n}_{np,x}(t)dt, \tag{8}$$

$$\hat{p}^{\,n}_{np,y}(t+dt) = \hat{p}^{\,n}_{np,y}(t) + \hat{v}^{\,n}_{np,y}(t)dt. \tag{9}$$

The estimated error covariance matrix (P) is propagated using the discrete Lyapunov equation

$$P^-_k = F_{k-1}P^+_{k-1}F^T_{k-1} + Q, \tag{10}$$

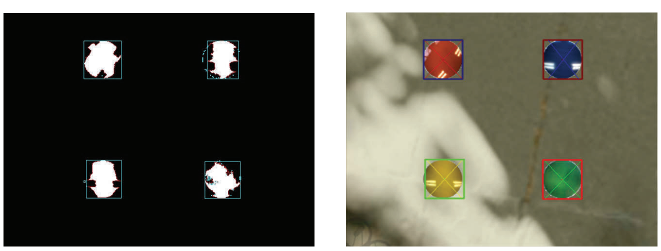
where F is a discrete-time state matrix, and Q is a process noise matrix. Since the dynamics model (static or mobile) is measured by color, a small Q is given to the static packages, and a large Q is given to the mobile packages.

When a new onboard image is available, the positions of packages are detected and used to update the estimated states of an EKF. This measurement correction is conducted using the standard Kalman equations

$$K_k = P^-_k H^T_k (H_k P^-_k H^T_k + R)^{-1}, \tag{11}$$

$$\hat{x}^{\,+}_k = \hat{x}^{\,-}_k + K_k(z - H_k\hat{x}_k), \tag{12}$$

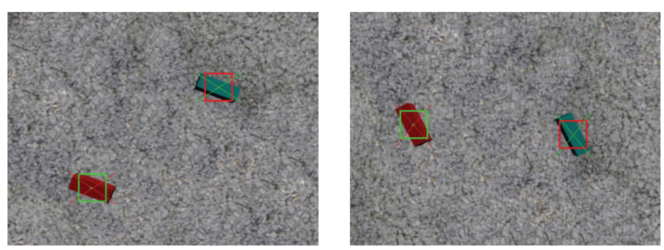
$$P^+_k = (I - K_k H_k)P^-_k, \tag{13}$$



(a) Image thresholding using HSV filter & bounded contour boxes

(b) Measurements (cross) & EKF estimated positions

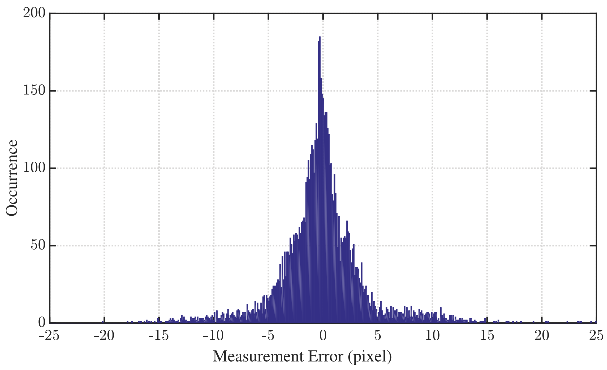
Fig. 9. Circular package detection and tracking.



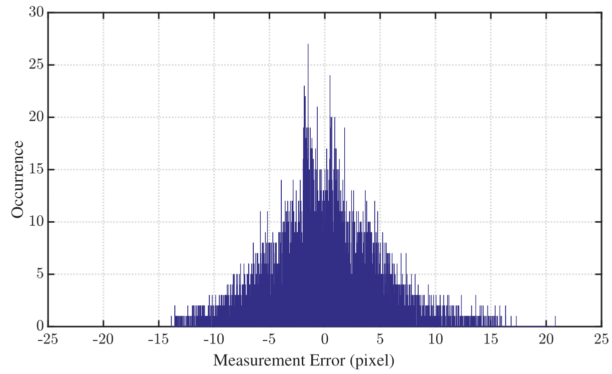
(a)

(b)

Fig. 10. Tracking of the rectangular packages in an image-in-the-loop simulation.



(a) Circular package



(b) Rectangular package

Fig. 11. Measurement error distribution of the vision system in a Monte Carlo simulation for circular and rectangular packages. 7000 samples were obtained in each experiment.

where R is the measurement error covariance matrix obtained from an experiment shown in Fig. 11, and H is the discrete-time output matrix obtained from linearization. Let $z = [u \quad v]^T$ denote the observed location of a package expressed in pixels. Then, the measurement model is

$$z = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_x \dfrac{p^c_{cp,y}}{p^c_{cp,z}} \\ f_y \dfrac{-p^c_{cp,x}}{p^c_{cp,z}} \end{bmatrix} + \nu, \tag{14}$$

where f_x and f_y are the focal lengths of the onboard camera, and ν is a measurement noise $\nu \sim N(0, R)$. Using this measurement model, H is derived from

$$\begin{aligned} H &= \frac{\partial z}{\partial x}\Big|_{x=\hat{x}} \\ &= \begin{bmatrix} \dfrac{\partial z}{\partial \hat{p}^n_{np}} & \dfrac{\partial z}{\partial \hat{v}^n_{np}} \end{bmatrix} \qquad (15) \\ &= \begin{bmatrix} \dfrac{\partial z}{\partial \hat{p}^c_{cp}} \dfrac{\partial \hat{p}^c_{cp}}{\partial \hat{p}^n_{np}} & 0 \end{bmatrix} \qquad (16) \\ &= \left[\begin{bmatrix} 0 & f_x \dfrac{1}{\hat{p}^c_{cp,z}} & -f_x \dfrac{\hat{p}^c_{cp,y}}{\hat{p}^{c2}_{cp,z}} \\ -f_y \dfrac{1}{\hat{p}^c_{cp,z}} & 0 & f_y \dfrac{\hat{p}^c_{cp,z}}{\hat{p}^{c2}_{cp,z}} \end{bmatrix} \hat{R}^c_n \quad 0 \right], \qquad (17) \end{aligned}$$

where $\hat{R}^c_n = R^c_b \hat{R}^b_n$ is an estimated rotation matrix from the navigation frame to the camera frame. Note $\frac{\partial \hat{p}^c_{cp}}{\partial \hat{p}^n_{np}} = \hat{R}^c_n$ because $\hat{p}^c_{cp} = -R^c_b p^b_{bc} + \hat{R}^c_n(\hat{p}^n_{np} - \hat{p}^n_{nb})$.

The vision system needs to have the ability of handling multiple targets in view. The color label and the χ^2 test gating help the system separate multiple targets. The Mahalanobis distance is computed from

$$Z^2 = \epsilon^T \Sigma^{-1} \epsilon, \tag{18}$$

where $\epsilon = z - H_k \hat{x}^-_k$ is the measurement residual. The Mahalanobis distance is an adaptive factor to separate multiple targets. When Z^2 is greater than a designated threshold, the system recognizes the measurement as a different target and initiates another EKF rather than using the measurement to update the states. Σ is the estimated covariance matrix of the residual

$$\Sigma = E[\epsilon \epsilon^T] = H_k P^-_k H^T_k + R. \tag{19}$$

The more confident the estimator is, the more strict this gating is. The error covariance matrix of the estimated state, P, is also used to initiate a mission. Algorithm 4.2. shows the overall procedure of the vision system. Note that each EKF does a loop over all of the measurements in order to find the available measurements.

The complexity of this algorithm in the worst case is $\mathcal{O}(mnp)$ where m is the number of EKFs, n is the number of colors and p is the number of measurements.

4.4. *LIDAR system*

The LIDAR system onboard the vehicle serves two main purposes: providing AGL altitude estimates for the navigation filter and offering accurate low AGL object detection. While laser rangefinders could be used for AGL estimates, they may fail if directly over an object. A 2D LIDAR, on the other hand, can potentially provide both AGL estimates and the C.G. of objects in view. To estimate AGL and object locations, Random Sample Consensus (RANSAC) is employed. RANSAC is an iterative method used to fit a model to data by repeatedly selecting random subsets and estimating the model parameters, which helps in determining AGL and object positions despite noise and outliers.

Algorithm 4.2. The EKF-based algorithm to determine a navigation target.

for $i = 1$:NUMBER OF EKFS **do**
  for $j = 1$:NUMBER OF COLORS **do**
    for $k = 1$:NUMBER OF MEASUREMENTS **do**
      $Z_k^2 = \epsilon_k^T (H_k P_k^- H_k^T + R_j)^{-1} \epsilon_k$ ▷ Compute the Mahalanobis distance
      if Z_k is less than a threshold **then**
        use Measurement(k) to update EKF(i)
      else
        A new EKF is started at measurement(k) with a small random velocity
      end if
    end for
  end for
  if EKF is started **then**
    Propagate EKF
    for $k = 1$:NUMBER OF VALID MEASUREMENTS **do**
      Update EKF with measurement(k)
    end for
  end if
  $|P|_i = \det(P[i])$
  if $|P|_i$ is less than a threshold **then**
    EKF(i) is a valid navigation target
  end if
end for

Algorithm 4.3. RANSAC AGL and Object detection estimation

$\Lambda_{\text{full}} \leftarrow$ Laser scan input (681 points) [–120° 120°]
$\Lambda_d \leftarrow$ Downsize laser scan to usable set [–85° 85°]
[inliers, outliers, model] $\leftarrow$ RANSAC (Λ_d)
if number inliers $\geq$ min inliers and model axis threshold is met **then**
  $\Lambda_{\text{rem}} \leftarrow$ Remove all outlier points within threshold
  if Λ_{rem} is not empty **then**
    [inliers, outliers, model] $\leftarrow$ RANSAC (Λ_{rem})
    if number inliers $\geq$ min inliers and model axis threshold is met **then**
      $z_{agl} \leftarrow$ Model with longer distance
      if Model with shorter distance is close to height of object (200 mm) **then**
        Object converged
        Calculate centroid of object
      else
        Object nonconverged
      end if
    else
      $z_{agl} \leftarrow \Lambda_d$ model is ground
    end if
  else
    $z_{agl} \leftarrow \Lambda_d$ model is ground
  end if
else
  $z_{agl} \leftarrow$ out of range
end if

It is reasonable to assume that the ground for package pickup and delivery will be relatively flat, with packages elevated above it. Consequently, a 2D laser scanner at range typically detects one or two lines: one from the ground and one from the package. RANSAC is employed to estimate the parameters of the model that best fit the data through multiple iterations. It identifies the points that best fit the model, known as inliers, while other points are considered outliers. This approach is used to determine the AGL and object location, as detailed in Algorithm 4.3. The Point Cloud Library (PCL) provides RANSAC algorithms for different models, including lines, and is integrated into our simulation environment (described in Sec. 5). The line model is parameterized with six coefficients: three for a point on the line and three for the lines direction. When RANSAC is executed, the output direction of the line is checked. If this direction is nearly parallel to the ground, it indicates that the line could represent either the ground or an object resting on it.

Algorithm 4.3. outlines a method for estimating AGL and object location. It is crucial to accurately identify which RANSAC model corresponds to which feature. At higher altitudes, most points will reflect off the ground, so the first RANSAC function call typically models the ground. However, as the vehicle descends and the laser scans objects, the initial RANSAC call might model the object instead of the ground. Simulations and flight tests validate this approach, as detailed in Sec. 5.

The time complexity of this algorithm depends on the implementation of the RANSAC function. With a vanilla implementation, the RANSAC time complexity is linear in the product of the number of iterations by the number of data points [45].

5. Experiments Conducted

5.1. *Simulations*

We developed a simulation environment using the Penn State—Georgia Tech UAV Software Tool (PSU-GUST) [46]. PSU-GUST supports software-in-the-loop (SITL), hardware-in-the-loop (HITL), and ground station operations. It features DFC vehicle dynamics, target dynamics, sensor models and communication emulation. This allowed us to test mission managers, guidance systems, DFC controllers, LIDAR and vision-tracking algorithms without physical vehicles.

Table 2 presents results from 10 20-min simulation runs. Each run begins with 10 stationary disks, and yellow moving targets are not detected. Waypoint reassignment is not used, and each vehicle attempts a maximum of seven grabs before resuming the search. The simulations demonstrate effective searching, grabbing and delivery of disks. Figure 12 shows a snapshot of one run, with three vehicles "*Vulture*, *Crow* and *Kite*" searching for packages. Red areas in the figure denote explored regions, while purple areas indicate zones to ignore. From Table 2, it is evident that most disks are located and delivered successfully in most runs. On average, the map was reset four to five times. However, without tracking the drop bin or using RTK GPS, the navigation alone is insufficient for accurate disk delivery into the drop bin.

5.2. *Package detection and DFC validation*

Flight tests were conducted to validate the package detection software. Two colored circular disks were constructed to ensure the vision and laser algorithms function correctly. Figure 13 displays detection results and EKF outputs overlaid on the onboard images from the test. Diagonal

Table 2. Disk pickup results when running simulation 10 times.

	Vulture	Crow	Kite	Total drops	
Run	Pick/attempt	Pick/attempt	Pick/attempt	In bin/in zone	Map reset
1	4/38	2/7	3/26	1/9	5
2	1/35	3/45	2/40	0/6	4
3	2/23	5/31	3/22	1/10	5
4	3/23	4/24	2/43	1/9	5
5	2/30	2/34	2/41	1/6	4
6	2/15	5/47	3/37	1/10	5
7	4/28	3/29	3/35	2/8	4
8	2/46	2/22	1/51	0/5	4
9	3/23	5/52	1/37	1/9	4
10	4/31	2/30	1/45	0/7	4

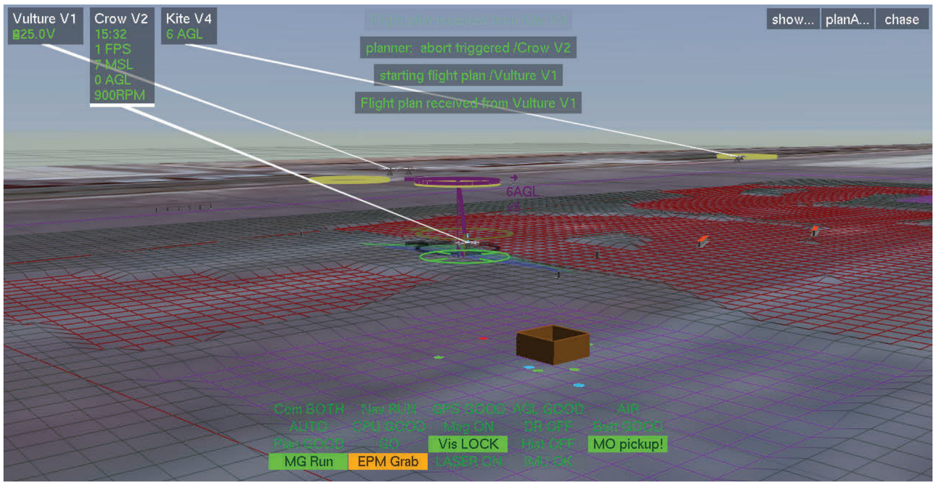
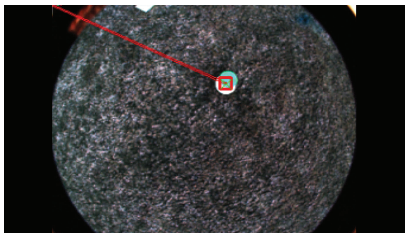
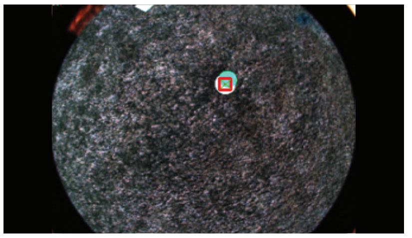


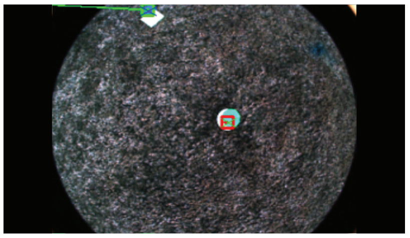
Fig. 12. Screenshot of full scenario simulated in PSU-GUST, enabling the detailed development of the entire mission.



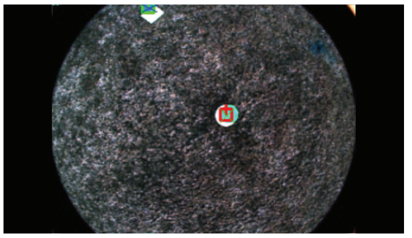
(a) Initialization of green disk EKF



(b) Tracking of green disk



(c) Initialization of blue disk EKF



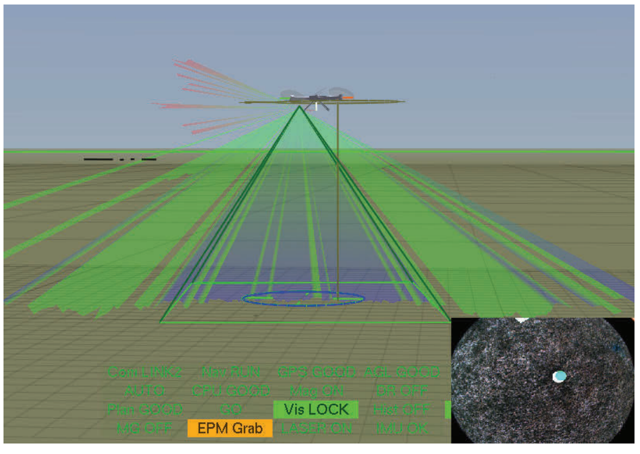
(d) Tracking both green and blue disks

Fig. 13. Detection and EKF results overlaid on the images obtained in the flight test.

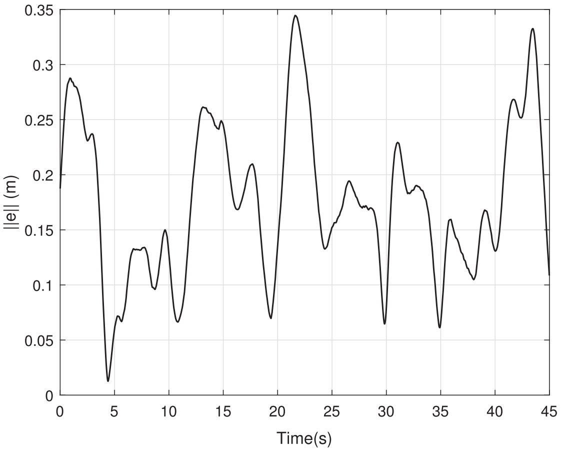
crossings indicate detection locations, with colors representing the detected disk colors. The boxes represent EKF positions, with red and green indicating the first and second EKFs, respectively. The HSV bound values are as follows:

- Blue: Hue 105–115, Saturation 220–240, Value 245–255
- Green: Hue 80–92, Saturation 155–175, Value 235–255

To validate the RANSAC method for AGL sensing, the Hokuyo sensor was moved along a ladder at known distances. The method consistently provides accurate AGL measurements within its practical maximum range of 4 m. Once accuracy was confirmed, the vehicle was flown over targets. Figure 14(a) shows the ground station view of the vehicle hovering over a package. LIDAR colors indicate AGL-RANSAC method usage: blue for ground points, green for points in range but not considered ground, and red for out-of-range points. A blue circle on the ground indicates package identification. Preliminary flight tests revealed difficulties in obtaining consistent LIDAR hits on the package. Figure 14(b) shows the normalized position error, with a mean error of 0.17 m compared to the package diameter of 0.02 m. Scaled DFC commands and raw IMU data are shown in Fig. 15(a), with commands scaled by the estimated maximum DFC capability ($7\,\mathrm{m/s^2}$). Roll and pitch during hover were near zero, as seen in Fig. 15(b),

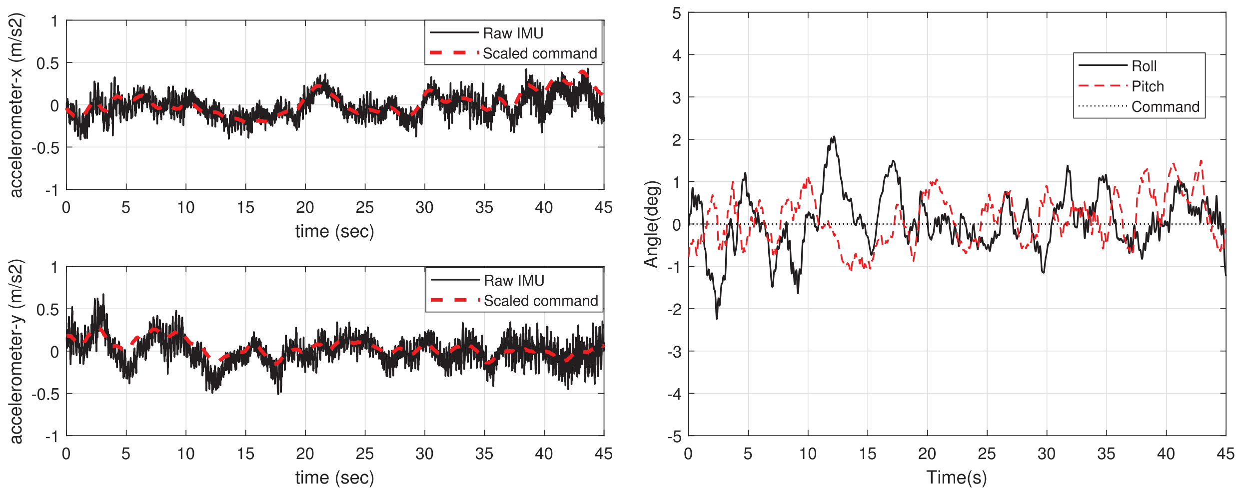


(a)



(b)

Fig. 14. (a) Captured still from PSU-GUST ground station during hover over target. (b) Normalized hover error over target.



(a) (b)

Fig. 15. (a) Scaled DFC commands with raw IMU comparison. (b) Estimates of roll and pitch over target.

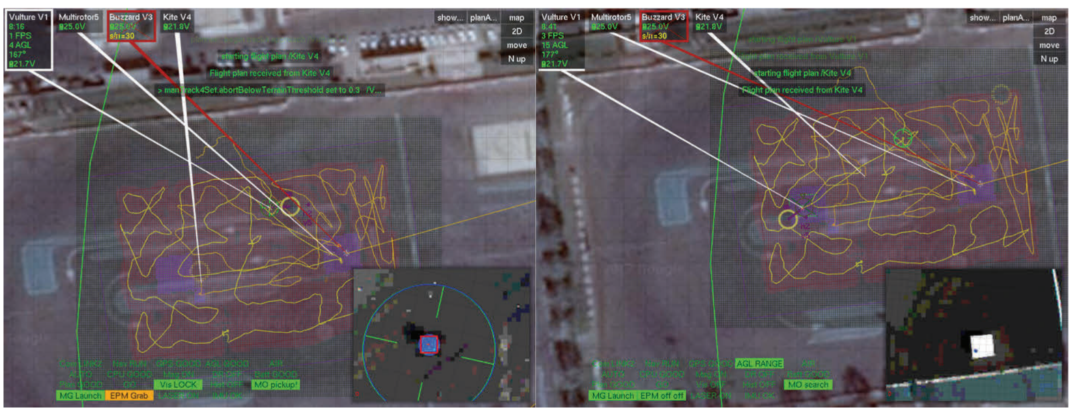


Fig. 16. Ground station stills from the practice attempt. Note the even map coverage by the aircraft based on their traced paths.

indicating that the daisy-chain limit was not reached during this flight segment.

6. Flight Demonstration

We performed flight demonstrations of our presented work at the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) competition [5–8]. Four identical vehicles were built—*Buzzard*, *Crow*, *Kite* and *Vulture*.

We completed a practice run on the competition arena with two aircraft: *Vulture* and *Kite*. The collaborative flight lasted 23 min, during which *Vulture* successfully dropped one package in the bin and another inside the drop zone. Figure 16 shows the ground station view of the practice. The RANSAC algorithm was effective for 98.75% of the flight. *Vulture* made five attempts to grab packages, with

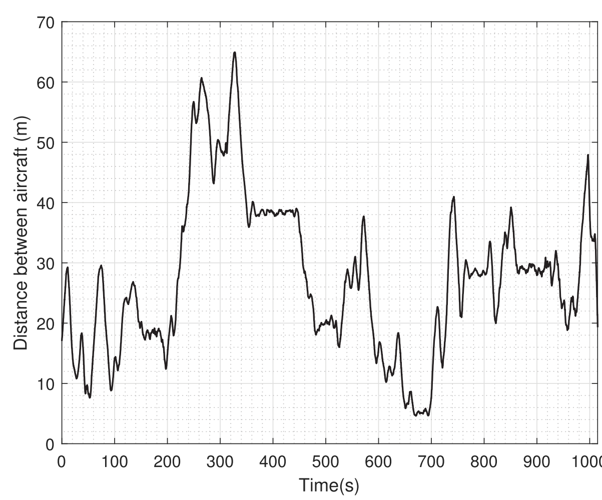


Fig. 17. Distance between aircraft (two in air) during the competition. The minimum specified distance of 4 m was met.

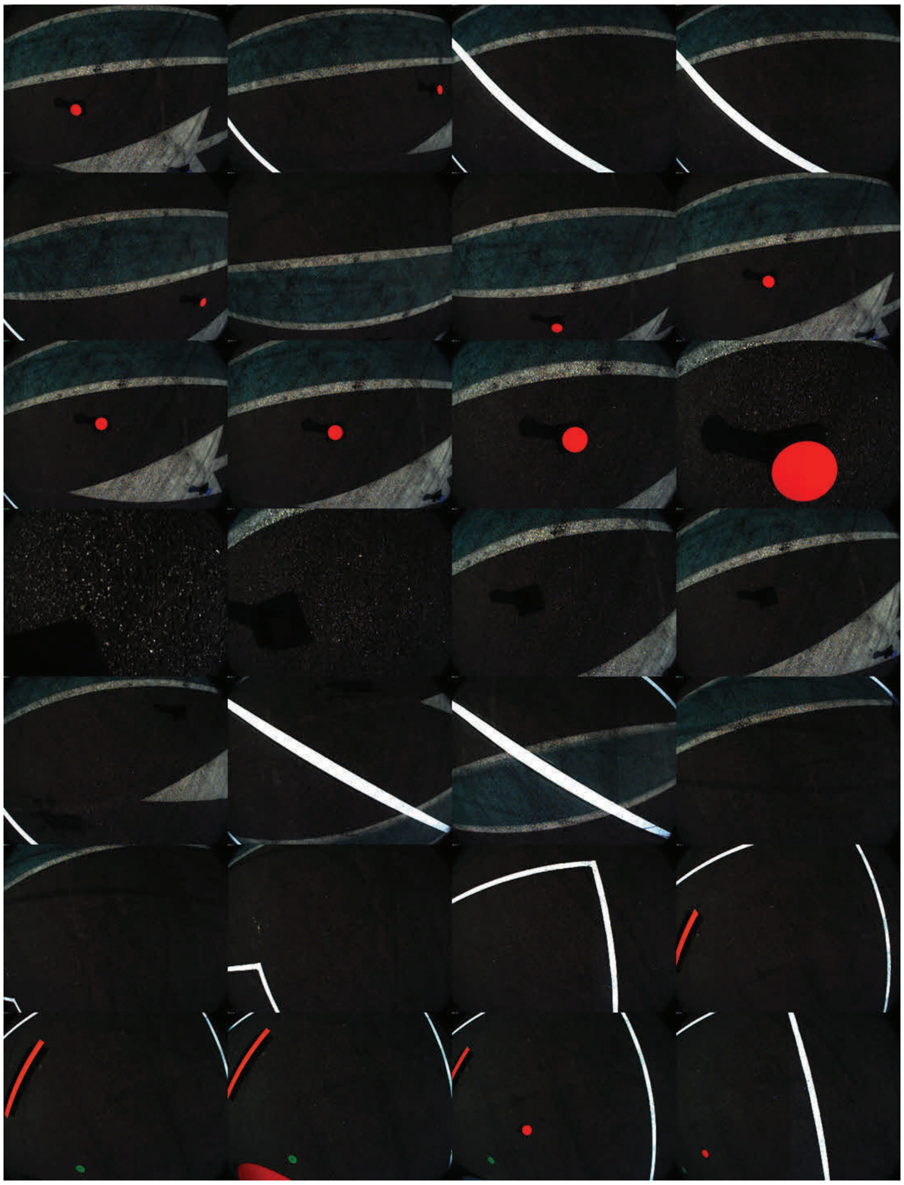


Fig. 18. Autonomous package searching, picking and delivering.

two successes. *Kite* detected and attempted to pick up one of two targets it flew over. The mission manager limited the aircraft to three descent attempts per package. This was our teams first test at the competition field. Figure 17 shows the distance between the two aircraft for that attempt. The set avoidance distance was set to 4 m.

On the last grand challenge attempt, *Vulture* and *Crow* each grabbed the first package they saw (first 2 min of flight). Video of the aircraft performing the competition tasks has been uploaded online[a] and the snapshots are shown in Fig. 18.

[a] https://youtu.be/o4Rg55rRVIs

7. Conclusion

The system described in this paper was fully capable of completing autonomous package pickup and delivery using a group of aerial vehicles. The combined laser and vision system provided a method to locate and grab packages. The DFC vehicle allowed the vehicle to stay level during wind and position changes. A map-sharing method was developed to allow multiple vehicles to search an area for packages while avoiding each other. The EPM enabled packages to be mechanically picked up and released. The team was able to successfully deliver five packages into the drop zone autonomously during missions flown in a competition setting.

Acknowledgments

Dr. Evangelos Theodorou, Dr. Daniel Magree, Dr. Jeremy Epps, Jimmy Tang, Manan Gandhi, Adrian Marrero, Matt Busch, Apoorva Beedu, Gerard Bennett, Lee Whitcher, Marcus Pereira, Sharath Yalla, Akash Patel, Eduardo Maya, Josh Peterson, Hyun Ki Lee, Kamil Saigol, Ziyi Wang, Ranjit Bijoy, Sharon Ang and Justin Lee all helped the team. The authors would like to thank the MBZIRC committee, organizers, Khalifa University, and fellow competitors for making it a great experience. A special shout out to fellow alum Dr. Suresh Kannan for being a great support during the actual competition.

ORCID

Stephen Haviland https://orcid.org/0000-0002-7152-5889
Jintasit Pravitra https://orcid.org/0009-0009-4773-921X
Eric Johnson https://orcid.org/0000-0001-5845-3641
Thanakorn Khamvilai https://orcid.org/0000-0001-5739-1338

References

[1] E. F. Griffith, J. M. Schurer, B. Mawindo, R. Kwibuka, T. Turibyarive and J. H. Amuguni, The use of drones to deliver rift valley fever vaccines in rwanda: Perceptions and recommendations, *Vaccines* **11**(3) (2023) 605.

[2] S. R. R. Singireddy and T. U. Daim, Technology roadmap: Drone delivery–amazon prime air, in *Infrastructure and Technology Management: Contributions from the Energy, Healthcare and Transportation Sectors* (Springer, Cham, 2018), pp. 387–412.

[3] X. Li, J. Tupayachi, A. Sharmin and M. Martinez Ferguson, Drone-aided delivery methods, challenge, and the future: A methodological review, *Drones* **7**(3) (2023) 191.

[4] N. T. K. Chi *et al.*, The drone delivery services: An innovative application in an emerging economy, *Asian J. Shipp. Logist.* **39**(2) (2023) 39–45.

[5] M. Beul, M. Nieuwenhuisen, J. Quenzel, R. A. Rosu, J. Horn, D. Pavlichenko, S. Houben and S. Behnke, Team nimbro at mbzirc 2017: Fast landing on a moving target and treasure hunting with a team of micro aerial vehicles, *J. Field Robot.* **36**(1) (2019) 204–229.

[6] R. Bähnemann, M. Pantic, M. Popović, D. Schindler, M. Tranzatto, M. Kamel, M. Grimm, J. Widauer, R. Siegwart and J. Nieto, The ETH-MAV team in the MBZ international robotics challenge, *J. Field Robot.* **36**(1) (2019) 78–103.

[7] Á. R. Castaño, F. Real, P. Ramón-Soria, J. Capitán, V. Vega, B. C. Arrue, A. Torres-González and A. Ollero, Al-robotics team: A cooperative multi-unmanned aerial vehicle approach for the Mohamed Bin Zayed international robotic challenge, *J. Field Robot.* **36**(1) (2019) 104–124.

[8] V. Spurnỳ, T. Báča, M. Saska, R. Pěnička, T. Krajník, J. Thomas, D. Thakur, G. Loianno and V. Kumar, Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles, *J. Field Robot.* **36**(1) (2019) 125–148.

[9] F. Kendoul, Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems, *J. Field Robot.* **29**(2) (2012) 315–378.

[10] C. Mertz *et al.*, Moving object detection with laser scanners, *J. Field Robot.* **30**(1) (2013) 17–43.

[11] T. Nakamura, S. Haviland, D. Bershadsky and E. N. Johnson, Vision sensor fusion for autonomous landing, in *AIAA Information Systems-AIAA Infotech@ Aerospace* (Grapevine, TX, USA, 2017), p. 0674.

[12] T. Nakamura, S. Haviland, D. Bershadsky, D. Magree and E. N. Johnson, Vision-based closed-loop tracking using micro air vehicles, in *IEEE Aerospace Conf.* (IEEE, 2016), pp. 1–12.

[13] D. Mellinger, Q. Lindsey, M. Shomin and V. Kumar, Design, modeling, estimation and control for aerial grasping and manipulation, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2011), pp. 2668–2673.

[14] C. C. Kessens, J. Thomas, J. P. Desai and V. Kumar, Versatile aerial grasping using self-sealing suction, in *IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2016), pp. 3249–3254.

[15] A. Gawel, M. Kamel, T. Novkovic, J. Widauer, D. Schindler, B. P. von Altishofen, R. Siegwart and J. Nieto, Aerial picking and delivery of magnetic objects with MAVs (2016), arXiv:1612.02606.

[16] J. Roberts, D. Frousheger, B. Williams, D. Campbell and R. Walker, How the outback challenge was won: The motivation for the UAV challenge outback rescue, the competition mission, and a summary of the six events, *IEEE Robot. Autom. Mag.* **23**(4) (2016) 54–62.

[17] N. Rooz *et al.*, Experience with highly automated unmanned aircraft performing complex missions, in *AIAA Guidance, Navigation, and Control Conf.*, Chicago, Illinois, 2009, p. 6157.

[18] S. Saeedi, M. Trentini, M. Seto and H. Li, Multiple-robot simultaneous localization and mapping: A review, *J. Field Robot.* **33**(1) (2016) 3–46.

[19] J. G. Mooney, S. Haviland and E. N. Johnson, Collaborative autonomy for mapping, search, and pursuit, *Unmanned Syst.* **4**(2) (2016) 167–184.

[20] W. Burgard, M. Moors, C. Stachniss and F. E. Schneider, Coordinated multi-robot exploration, *IEEE Trans. Robot.* **21**(3) (2005) 376–386.

[21] H. C. Christmann, A path planning aid for single operator UAV swarms in structured environments, Ph.D. Dissertation, Georgia Institute of Technology (2015).

[22] D. Bershadsky and E. N. Johnson, Indoor GPS-denied context based SLAM aided guidance for autonomous unmanned aerial systems, in *AIAA Guidance, Navigation, and Control (GNC) Conf.*, Boston, MA, 2013, p. 4989.

[23] R. Zlot, A. Stentz, M. B. Dias and S. Thayer, Multi-robot exploration controlled by a market economy, in *IEEE Int. Conf. Robotics and Automation*, Vol. 3 (IEEE, 2002), pp. 3016–3023.

[24] M. B. Dias and A. Stentz, A free market architecture for distributed control of a multirobot system, in *6th Int. Conf. Intelligent Autonomous Systems (IAS-6)* (Venice, Italy, 2000), pp. 115–122.

[25] P. Bates and D. Schrage, The configuration and conceptual design for rotary wing aircraft, in *Proc. AIAA/AHS/ASEE Aircraft Design Systems and Operations Meeting*, St. Louis, Missouri, USA, 1987, p. 10.

[26] A. S. Khalid, Development and implementation of rotorcraft preliminary design methodology using multidisciplinary design optimization, Ph.D. Dissertation, Georgia Institute of Technology (2006).

[27] E. S. Latorre, Propulsion system optimization for an unmanned lightweight quadrotor, Master's Thesis Universitat Politecnica de Catalunya, Master in Aerospace Science Technology (2011).

[28] O. Gur and A. Rosen, Optimizing electric propulsion systems for unmanned aerial vehicles, *J. Aircraft* **46**(4) (2009) 1340–1353.

[29] Y. Mulgaonkar, M. Whitzer, C. Kroninger and M. Aaron, Power and weight considerations in small, agile, quadrotors, *Proc. SPIE* **9083** (2014) 90831Q.

[30] M. Ramp and E. Papadopoulos, On modeling and control of a holonomic vectoring tricopter, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2015), pp. 662–668.

[31] M. Ryll, H. H. Bülthoff and P. R. Giordano, A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation, *IEEE Trans. Control Syst. Technol.* **23**(2) (2015) 540–556.

[32] M. Ryll, D. Bicego and A. Franchi, Modeling and control of fast-hex: A fully-actuated by synchronized-tilting hexarotor, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2016), pp. 1689–1694.

[33] D. Brescianini and R. D'Andrea, Design, modeling and control of an omni-directional aerial vehicle, in *IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2016), pp. 3261–3266.

[34] S. Park, J. Her, J. Kim and D. Lee, Design, modeling and control of omni-directional aerial robot, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2016), pp. 1570–1575.

[35] G. Jiang and R. Voyles, A nonparallel hexrotor UAV with faster response to disturbances for precision position keeping, in *IEEE Int. Symp. Safety, Security, and Rescue Robotics (SSRR)* (IEEE, 2014), pp. 1–5.

[36] D. Bershadsky, Electric multirotor design and optimization, Ph.D. Dissertation, Georgia Institute of Technology (2017).

[37] J. S. Dittrich, Design and integration of an unmanned aerial vehicle navigation system, Master's Thesis, School of Aerospace Engineering, Georgia Institute of Technology (2002).

[38] S. Haviland, D. Bershadsky, D. Magree and E. N. Johnson, Development of a 500 gram vision-based autonomous quadrotor vehicle capable of indoor Navigation, in *American Helicopter Society (AHS) 71st Annual Forum* (Georgia Institute of Technology, 2015).

[39] D. Bershadsky, S. Haviland and E. N. Johnson, Electric multirotor propulsion system sizing for performance prediction and design optimization, in *Proc. AIAA Scitech Design Processes and Tools Conf.* (San Diego, CA, USA, 2016).

[40] E. N. Johnson and S. K. Kannan, Adaptive trajectory control for autonomous helicopters, *J. Guid. Control Dyn.* **28**(3) (2005) 524–538.

[41] H. Mehmood and E. N. Johnson, A daisy-chain control design for a multirotor UAV with direct force capabilities, in *AIAA Guidance, Navigation, and Control Conf.*, Grapevine Texas, (2017), p. 1043.

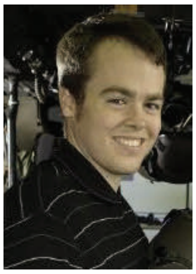
[42] E. N. Johnson, Limited authority adaptive flight control, Ph.D. Dissertation, School of Aerospace Engineering, Georgia Institute of Technology (2000).

[43] B. L. Stevens, F. L. Lewis and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, (John Wiley & Sons, 2015).

[44] H. Carr, J. Snoeyink and U. Axen, Computing contour trees in all dimensions, *Comput. Geom.* **24**(2) (2003) 75–94.

[45] J. M. Martínez-Otzeta, I. Rodríguez-Moreno, I. Mendialdua and B. Sierra, Ransac for robotic applications: A survey, *Sensors* **23**(1) (2022) 327.

[46] E. N. Johnson and D. P. Schrage, System integration and operation of a research unmanned aerial vehicle, *AIAA J. Aerosp. Comput., Inf. Commun.* **1**(1) (2004) 5–18.



Stephen Haviland received his M.S. degree from the Georgia Institute of Technology. He is Co-founder of optim.Aero, a UAS design, build, and test company, providing consultation services including design and simulation focusing on aerobotic vertical takeoff and landing (VTOL) vehicles.

Dmitry Bershadsky received his M.S. and Ph.D. degrees from the Georgia Institute of Technology. He is a founder of optim.Aero, a UAS design, build and test company, providing consultation services including design and simulation focusing on aerobotic vertical takeoff and landing (VTOL) vehicles.

Jintasit Pravitra is a Faculty at the Department of Mechanical Engineering at the Navaminda Kasatriyadhiraj Royal Air Force Academy. He earned his Bachelor's degree and Master's degree in Aerospace Engineering at Georgia Tech. He earned his Ph.D. in Aerospace Engineering at Penn State under the supervision of Professor Eric Johnson. His research interest focuses on guidance, navigation and control for unmanned aircraft. At the Royal Thai Air Force, he is currently working on developing indigenous solutions for small unmanned aerial systems tailored specifically for the Thai military.

Takuma Nakamura is Senior Staff Autonomy Engineer at Wisk Aero. He received his M.S. and Ph. D. degrees from the Georgia Institute of Technology. His primary research interests include aircraft navigation, computer vision systems, sensor fusion and autonomous flight control.

Eric Johnson is Professor of Aerospace Engineering at the Pennsylvania State University (PSU), and Director of the PSU UAS Research Laboratory (PURL). He received a B.S. degree from the University of Washington, M.S. degrees from MIT and The George Washington University, and a Ph.D. from Georgia Tech. He also has five years of industry experience working at Lockheed Martin and Draper. As a faculty since 2001, he has performed research in unmanned aircraft fault-tolerant control, aided inertial navigation and autonomy.

Thanakorn Khamvilai is Assistant Professor of Aerospace Engineering in the Department of Mechanical Engineering at Texas Tech University. He received a B.S. degree from Kasetsart University, a M.S., and a Ph.D. degrees from the Georgia Institute of Technology. His research interest is in unmanned aerial systems, avionics and optimization.