SYSTEMS ARTICLE

WILEY

A marsupial robotic system for surveying and inspection of freshwater ecosystems

Michail Kalaitzakis[1] | Brennan Cain[2] | Nikolaos Vitzilaios[1] | Ioannis Rekleitis[2] | Jason Moulton[2]

[1]Department of Mechanical Engineering, University of South Carolina, Columbia, South Carolina

[2]Department of Computer Science and Engineering, University of South Carolina, Columbia, South Carolina

Correspondence
Nikolaos Vitzilaios, Department of Mechanical Engineering, University of South Carolina, Columbia, SC 29208.
Email: vitzilaios@sc.edu

Abstract

Freshwater ecosystems are vast areas that are constantly changing and evolving. To maintain the ecosystem as well as the structures located close to bodies of water, frequent monitoring is required. Although dangerous and time consuming, manual operations are the conventional way of monitoring such areas. Recently, Autonomous Surface Vehicles (ASVs) have been proposed to undertake the monitoring task. As any other platform, ASVs have limitations, such as a restricted point of view and access only where the water is sufficiently deep. Unmanned Aerial Vehicles (UAVs) can fly over any terrain and provide a "bird's-eye-view" of the environment. However, UAVs have limited operational time due to power constraints. Heterogeneous marsupial robotic systems use different types of robots to augment their operation envelope, taking advantage of their individual strengths. A marsupial survey system comprised an ASV and a UAV for freshwater monitoring is developed and presented in this paper. This system is able to complete long missions and reach remote locations while also being able to generate detailed maps and inspections of points of interest. The system was thoroughly tested during a 6-month period in a number of field deployments in freshwater ecosystems at Lake Murray and at the Congaree River, SC, USA, to validate its capabilities.

KEYWORDS

cooperative robots, environmental monitoring, robot teaming

1 | INTRODUCTION

Monitoring the condition of bodies of water and nearby structures is an important capability for scientists, governments, and citizens. Recently, Autonomous Surface Vehicles (ASVs) have been shown to increase the efficiency of monitoring missions by removing the need for humans to conduct these missions (Karapetyan et al., 2018, 2019). These systems, however, face their own shortcomings. Because ASVs are restricted to the water's surface, they have a low point of view. In marine monitoring, specifically for algal bloom biomass characterization (Kislik, Dronova, & Kelly, 2018), a low point of view is not efficient due to reflections and the small size of the well-observable area around the ASV. For inspecting near-water structures, such as docks and bridges,

ASVs are unable to fully inspect all surfaces, particularly the tops, which can lead to improper classification of the state of wear on a structure.

On the other hand, Unmanned Aerial Vehicles (UAVs), more commonly referred as drones, are ideal platforms for conducting surveys and infrastructure inspections (Mathe et al., 2016). UAVs, particularly Vertical Take-Off and Landing (VTOL) capable ones, are able to provide a higher point of view than an ASV for marine surveying and can be used to cover all exposed surfaces of a structure. Moreover, they can be equipped with mission-specific sensors for specialized inspection tasks (Jimenez-Cano, Sanchez-Cuevas, Grau, Ollero, & Heredia, 2019; Kalaitzakis, Kattil, Vitzilaios, Rizos, & Sutton, 2019). However, drones face issues, such as a limited flight endurance (Malyuta et al., 2019) and a low payload capacity.

With a marsupial system, we can overcome the individual weaknesses of the two platforms. Marsupial systems place a smaller or "weaker" robot on another larger robot to increase the capabilities of the overall system. In our case, a UAV is placed on-board an ASV, thus extending its total mission time by having the ASV ferry it to areas of interest. Meanwhile, the drone contributes to the overall system by providing a higher point of view, that is, a bird's-eye-view and the ability to cover all areas of a near-water structure. When the system is deployed, the following measurements are obtained: the position of the ASV, the position of the UAV, and the relative pose between the two from a vision-based tracker. By taking advantage of the extra information, we can increase the accuracy of the system and its robustness in the event of Global Positioning System (GPS) signal loss; this process is termed Cooperative Localization (CL; Rekleitis, Dudek, & Milios, 1997, 1998).

The main contribution of this study lies in the development and field deployment of the proposed system. Through an extensive period of outdoor testing spanning over 6 months, we show the effectiveness and the robustness of the system for repeated data acquisition in freshwater environments.

The paper is organized as follows. Section 2 surveys existing work on marsupial robotic systems and their applications in different domains. Section 3 details the system design, including a high-level controller with safety features, flight control for the missions, and the autonomous take-off and landing procedures. Section 4 presents the CL framework used in this study, followed by a detailed description of the system control schemes in Section 5. Results from the CL method used, comparative results from two different flight controllers tested for the autonomous landing task, as well as results from a sample survey and infrastructure inspection are then given in Section 6. Section 7 provides a discussion on lessons learned and future research directions, followed by a conclusion in Section 8.

2 | BACKGROUND

Marsupial robotic systems were first introduced by Anderson, McKay, and Richardson (1996) where an Unmanned Ground Vehicle (UGV) deployed a smaller UGV to inspect areas that the larger robot could not reach. Murphy et al. (1999) were the first to use the term "Marsupial Robot" to describe a system of a parent and child robot used in this manner. A survey of marsupial systems in several environments is presented by Hourani, Wolters, Hauck, and Jeschke (2011), where the roles that the marsupial members have are explained and the advantages of such systems are discussed.

Recently, marsupial systems have gained popularity in the marine robotics domain. Murphy et al. describe the use of a non-autonomous cooperative system of a UAV and an Unmanned Surface Vehicle (USV) to survey the areas affected by hurricane Wilma (Murphy et al., 2008). The UAV, operated by a human pilot, was used to guide the USV around broken sections of bridges and piers. Several missions involved using the drone to cover areas that the USV could not access, such as roofs of buildings or tops of bridges. This system sets the basis for later papers in which UAVs are used to create cost maps and pinpoint features of interest for USVs to access.

Lindemuth et al. present an early UAV–USV marsupial system (Lindemuth et al., 2011) as an extension of Murphy et al. (2008). A drone is used to provide a higher point of view for tracking objects in the water. Unlike the prior system, this system carries the UAV on-board the USV while moving to the areas to be surveyed. This allows the drone to conserve flight time while moving to far-off locations. The drone operator is able to use cameras on board the two platforms to navigate while the USV operator can use the drone's camera to find safe paths to follow. Moreover, this system does not support landing back on the USV nor autonomous cooperative behaviors.

Marques et al. (2015) present a system of a UAV and an ASV used for environmental monitoring. The ASV acts as the carrier, while the drone is used either for data gathering or to augment the field of view of the ASV. In order for the UAV to return to the ASV, the two platforms must cooperate. While the ASV is trying to visually detect the UAV, the drone does a spiral search trying to detect the ASV. The UAV is equipped with an Augmented Reality (AR) marker that the ASV uses to locate it and track its pose. To ensure the safe docking of the UAV, they utilize a safety net around the landing area. However, they report instances where the UAV gets caught in the net, making it impossible to take-off again and continue its operation.

Marsupial systems have also been proposed in the Search and Rescue domain (Mendonça et al., 2016), where the UAV is used to provide a high point of view to search for life boats. If a potential target is identified, the drone hovers around the target while it transmits its location. Once the ASV arrives at the target location, it starts searching for heat signatures similar to those from human bodies. If a signature is found, then it relays a color image to the operation center for a human rescue team to be deployed. Since the UAV used in this application is waterproof, it can land on water to conserve battery if it cannot dock on the ASV. For the docking procedure, the drone needs to detect the landing platform on the ASV. However, during the last moments of the landing, since the drone is close to the platform it is not able to detect it. As a result, the ASV needs to also detect the UAV with an upward facing camera. This camera, however, is sensitive to lens flares and overexposure, causing difficulties in the detection of the UAV. Recent work by Hood et al. (2017) using an indoor ground robot and a UAV reported similar problems with an upward facing camera.

Another notable example of a marsupial system is the Mars 2020 rover and the Mars Helicopter (Balaram et al., 2018). While the 2020 mission will primarily serve as a technology demonstrator, the helicopter aims to improve and augment planetary exploration. The helicopter can explore large areas faster than a rover and it can also be used to provide reconnaissance on target locations and safe to traverse routes. Finally, the platform could access areas not reachable by a rover and retrieve small science samples.

Previous field work shows that a key characteristic to ensure the continuous and safe operation of a marsupial system is the effective and repeatable docking and separation procedure. In the proposed system, the separation and docking procedures are taking-off and

landing on a moving platform. While autonomous take-off is a trivial process for VTOL capable vehicles, autonomous landing is considered one of the most challenging parts of flight, and it requires precise sensing and accurate control (Brommer, Malyuta, Hentzen, & Brockers, 2018; Gautam, Sujit, & Saripalli, 2014; Malyuta et al., 2019). The landing problem is made more difficult when the landing area is not stable.

Landing on a moving platform has recently captured the interest of the scientific community. The challenges of this procedure have been demonstrated in space, capturing a satellite (Christidi-Loumpasefski, Nanos, & Papadopoulos, 2017; Rekleitis et al., 2007), underwater (Cowen, Briest, & Dombrowski, 1997; Myint, Yonemori, Yanou, Minami, & Ishiyama, 2015), and with aerial systems which is the focus of this study (Balmer, 2015; Chaves, Wolcott, & Eustice, 2015). A common characteristic in most approaches is the utilization of vision-based positioning techniques for the task. In Falanga, Zanchettin, Simovic, Delmerico, and Scaramuzza (2017) specifically, the proposed method uses only visual and inertial data for the position estimation without using any external infra-structures, such as a Global Navigation Satellite System (GNSS) or a Motion Capture (MoCap) system. Using Visual Inertial Odometry (VIO) to estimate the state of the UAV and pattern recognition to estimate the position of the landing platform they were able to land a UAV on a moving UGV.

As for the control methods used for the landing task, a variety of control schemes has been proposed. In Ghamry, Dong, Kamel, and Zhang (2016), a combination of Sliding Mode Control (SMC) and Linear Quadratic Regulator (LQR) was used to land a UAV on a moving UGV with the state of the two platforms provided by a MoCap system. On the more challenging task of landing on an in-clined moving platform, a Model Predictive Controller (MPC) was used in Vlantis, Marantos, Bechlioulis, and Kyriakopoulos (2015). In this case, visual-based position estimation was used as well. Proportional–Integral–Derivative (PID) control is used for the landing maneuvers in Araar, Aouf, and Vitanov (2017), where a landing pad designed using a number of fiducial markers of different sizes is proposed, to allow for a better position estimation from different distances from the target. Finally, in Hentzen, Stastny, Siegwart, and Brockers (2019), a nonlinear MPC that takes into account disturbances, like, wind gusts and the ground effect, is presented.

In most cases of autonomous landing on moving platforms, the moving platform utilized is a ground vehicle. While in ground vehicles the roll and pitch motion of the platform is negligible, that is not the case in boats, as the motion of the water is expected to affect them. In Sanchez-Lopez, Saripalli, Campoy, Pestana, and Fu (2013), an ef-fort to create a test-bed that models the motion of a vessel deck was made using a parallel robot. Then, using on-board computer vision and filtering techniques, they were able to estimate the motion of the platform from a UAV. A similar deck emulator platform was used in L. Wang and Bai (2018), where a small scale UAV using computer vision position estimation and a set of PID controllers was able to land on the platform.

The ability to localize one robot based on observations taken by a second robot is termed CL (Rekleitis et al., 1998), and was first in-troduced as a cooperative positioning system by Kurazume and Hirose (2000) and Kurazume, Nagata, and Hirose (1994). Many approaches were introduced based on the Kalman (Roumeliotis & Bekey, 2002) or Particle (Burgard, Moors, Fox, Simmons, & Thrun, 2000) Filters for exploration, localization, or simultaneous localization and mapping (SLAM; Rekleitis, Dudek, & Milios, 2001a). CL is an NP-hard problem (Dieudonné, Labbani-Igbida, & Petit, 2010). Improved positioning, through CL, was also used for guiding a robot generating a visual map as early as 2001 (Rekleitis, Sim, Dudek, & Milios, 2001b). In this paper the drone observes the ASV enabling navigation even when GPS signal is lost.

3 | SYSTEM DESIGN

In this study we developed a marsupial robotic system composed of an ASV, the AFRL Jetyak (Moulton et al., 2018), and a UAV, the DJI Matrice 100,[1] both shown in action in Figure 1. Next, a brief overview of the system is presented.

The AFRL Jetyak is an ASV based on the Mokai Es-Kape,[2] a 3.6-m long boat with a seven horsepower, four stroke engine (Moulton et al., 2018). Its operation time can vary between 4 and 18 hr depending on the speed of the boat and the maximum payload capacity is 163 kg. The system is augmented with a Pixhawk PX4 microcontroller and an RFD900+ radio. This allows for remote and autonomous operations of the vehicle as well as telemetry broad-casting. The Jetyak system architecture can be seen in Figure 2. For the purpose of this project, a 1.2 m × 1.2 m landing platform was designed and built in our lab. The platform is attached to the top of the Jetyak behind the mast. The platform features a quick release mechanism as well as a folding surface, for easy transportation and quick installation on the boat.

Following the example of Araar et al. (2017), a bundle of six AR tags (Fiala, 2005) of various sizes is placed on the Jetyak. Specifically, a set of four AR tags, 52 cm × 52 cm each, are placed on the landing platform while a set of two tags, 13 cm × 13 cm, is attached at the base of the mast. The landing platform bundle is used to provide a visual tether over long distances while the mast bundle is used when the drone flies in close proximity to the Jetyak. AR tags were chosen over other fiducial markers, like, April tags (Olson, 2011; J. Wang & Olson, 2016), for their efficient and fast tracking performance. Our experimentation found that in the deployed hardware on-board the drone, AR tag tracking is providing accurate position measurements at 13 Hz while April tag tracking has a lower rate of 5 Hz.

The DJI Matrice 100 is a development-focused platform for autonomous aircraft research. The on-board computer used is a DJI Manifold, based on the NVidia Tegra TK1. A Zenmuse Z3 gimbal camera is attached for data gathering and visual position estimation,

---

FIGURE 1 A view of the UAV-ASV marsupial system where the UAV follows the ASV at a safe distance during field trials. ASV, Autonomous Surface Vehicle; UAV, Unmanned Aerial Vehicle [Color figure can be viewed at wileyonlinelibrary.com]

and an RFD900+ radio is used for communications with the Jetyak and the Ground Control Station (GCS). All the peripheral devices, as well as the UAV's N1 Flight Control System, are connected to the onboard computer. The computer is able to access the sensor data and send high-level controls to the drone using the DJI Onboard SDK.

The two platforms and the GCS communicate over MAVLink. The connection allows messages to be exchanged either directly from one node to the other, or indirectly through another node. The Robot Operating System (ROS; Quigley et al., 2009) was used as a middleware to develop the systems for both platforms as well as the communications between them. Radio broadcasts can be accessed using MAVROS, an ROS package that provides the communication drivers for MAVLink.

The overall architecture of the UAV system, with its individual components, can be seen in Figure 3. The system is designed in a highly modular way that allows wrappers to be written for each hardware component, such as the gimbal camera or the flight control system. This increases the interoperability of the system.

3.1 | Visual position estimation using an independently controlled gimballed camera

The most important subsystem of the UAV is the gimballed camera. The camera is being used to gather data during infrastructure inspections and surveys and most importantly it is used to provide an estimate of the relative position between the two platforms. While in most cases where cameras are used to provide position measurements the cameras are rigidly attached on the platform (Falanga et al., 2017; Sanchez-Lopez et al., 2013; Vlantis et al., 2015), in our case the camera can move independently of the platform.

Having the camera able to move independently of the platform, it provides the advantage of being able to track the target even in situations where the drone is not facing the boat. Moreover, when the drone flies close to the platform, even small movements could lead to tracking loss with a stable camera. With a moving camera, the gimbal can rotate to keep the target always in the camera's field of view and thus maximizing the rate of data from the visual position estimation algorithm. By tracking the relative orientation of the gimbal frame to the body frame, we can transform the position measurements from the camera frame to the body frame. The UAV used and the gimballed camera with the two coordinate systems are shown in Figure 4.

In most cases where gimbals are used in UAVs, their role is to compensate for the motion of the UAV and to keep the camera's orientation. In our case, the gimbal is independently controlled to keep a point of interest in frame. Two different control scenarios were implemented. First, if the target is already in frame, the gimbal

FIGURE 2 Block diagram of the ASV system architecture. The blue blocks are open-source ROS layers, the green blocks are the developed components, the magenta blocks are the hardware peripherals of the ASV, and the cyan one is hardware peripherals of the UAV. The dotted line denotes the use of the cooperative localization data in the Jetyak controller. ASV, Autonomous Surface Vehicle; ROS, Robot Operating System; UAV, Unmanned Aerial Vehicle; RC, Remote Control [Color figure can be viewed at wileyonlinelibrary.com]
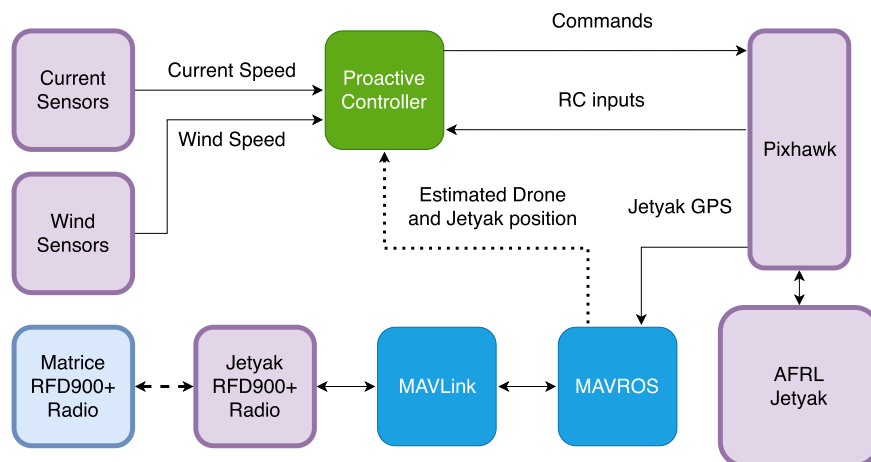
FIGURE 3 Block diagram of the developed UAV system architecture. The blue blocks are open-source ROS layers, the green blocks are components developed for this project, the magenta blocks are the hardware peripherals of the UAV, and the cyan one is hardware peripherals of the ASV. ASV, Autonomous Surface Vehicle; ROS, Robot Operating System; UAV, Unmanned Aerial Vehicle [Color figure can be viewed at wileyonlinelibrary.com]

is rotated as to always keep the target in the center of the image. When the target is not in the line of sight of the camera but its position is known, the gimbal moves so that the projection of the target aligns with the center of the image. In the case of tracking the Jetyak when it is not in the line of sight of the drone, the position of the boat from the CL framework is used to control the gimbal.

4 | COOPERATIVE LOCALIZATION

Missions in marine and especially in freshwater environments require operating close to vegetation and large structures. In such environments, GNSS positioning, like, GPS, Galileo, and so forth, can be unreliable or even unavailable. Moreover, on long missions the quality of the GPS measurements is expected to change. At the same time, these environments are visually challenging with the vegetation generating extreme lighting variations between shadows and direct sunlight resulting in lens flares and overexposed images, while the water is generally featureless and there is an increased chance of reflections on the water surface (Chahine & Pradalier, 2018). To ensure the safety of the autonomous landing and to increase the effectiveness of navigating in such environments, a robust localization framework is needed that will be able to detect and reject outliers as well as to adapt to the quality of the sensor measurements.

A CL method based on a Kalman Filter is used throughout all the experiments in this study. Both the UAV and the ASV have a built-in algorithm that fuses the GPS position and the inertial sensor measurements to get a more accurate positioning of the platform. However, since the two systems use different sensors and are expected to operate in different regions at times, localization based solely on this data is unreliable for precision maneuvers. Visual position estimation methods, such as Visual Odometry (VO), are often used to increase the

FIGURE 4 The body and camera coordinate systems shown on the UAV. Since the camera is mounted on a gimbal it is free to rotate relative to the body. UAV, Unmanned Aerial Vehicle [Color figure can be viewed at wileyonlinelibrary.com]

accuracy of the localization framework. In marine environments, however, VO is challenging and visual methods that use specific features and patterns are more appropriate in such cases.

Our main motivation for CL is to get an accurate relative position between the two platforms that is crucial for the precision landing maneuvers. CL can also be used to assist navigation in GPS-degraded or even denied environments. For example, when there is a need to traverse under a large bridge, the drone can be deployed to a position with a better GPS signal and to provide a more accurate position estimation to the Jetyak by maintaining a visual tether. Once the quality of the GPS signal has been restored on the Jetyak side, the drone can fly over the bridge and return to land on the Jetyak.

The designed framework fuses the measured position of the two platforms, the measured velocity of the drone and the relative position of the platforms acquired from a visual positioning method. As described above, the Jetyak features a bundle of six AR tags that are detected and tracked to provide the relative position of the two platforms. During the system setup, it was noted that there was a significant offset between the measured position of the Jetyak and the measured position of the drone even when the sensors were placed next to each other. Furthermore, the quality of the GPS measurements of the UAV was superior to that of the Jetyak both indoors and outdoors. As a result, an extra state was added to the filter to track the offset on a local East-North-Up (ENU) frame.

Figure 5 shows the localization scheme used with

$$p_{\text{UAV,GPS}} = p_{\text{UAV}} + w_{\text{UAV}}, \tag{1}$$

$$p_{\text{Jetyak,GPS}} = \hat{p}_{\text{Jetyak}} + w_{\text{Jetyak}}, \tag{2}$$

$$p_{\text{Jetyak}} = \hat{p}_{\text{Jetyak}} - b_{\text{GPS}}, \tag{3}$$

where $p_{\text{UAV,GPS}}$ is the measured position of the drone with p_{UAV} the true position and w_{UAV} the measurement noise. The measured position of the Jetyak is $p_{\text{Jetyak,GPS}}$, with $\hat{p}_{\text{Jetyak}}$ the Jetyak position before it is corrected, $w_{\text{Jetyak,GPS}}$ the sensor noise, p_{Jetyak} the corrected position and b_{GPS} the offset. The state of the filter is

$$x = \begin{bmatrix} p_{\text{UAV}}^T & v_{\text{UAV}}^T & \hat{p}_{\text{Jetyak}}^T & v_{\text{Jetyak}}^T & b_{\text{GPS}}^T \end{bmatrix}^T, \tag{4}$$

where v_{UAV} is the three-dimensional (3D) velocity of the drone on the same ENU frame and v_{Jetyak} the 2D velocity of the Jetyak. Since the Jetyak can only move on the surface of the water, only minor changes in its altitude are expected due to any waves. Thus, its velocity in the z-axis is assumed to be zero. While the AR tag measurements provide information about both the position and the orientation of the tag, only the position measurements were used. The attitude of the two platforms was estimated by using only their on-board sensors. The measurement from the AR tag tracking is finally defined as
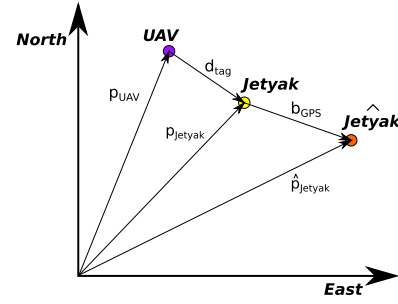

FIGURE 5 The scheme used in the cooperative localization framework. The purple dot is the estimated position of the drone in the local ENU system while the yellow dot is the estimated position of the Jetyak in the same system. The orange dot is the measured position of the Jetyak before it is corrected by subtracting b_{GPS}. ENU, East-North-Up; GPS, Global Positioning System; UAV, Unmanned Aerial Vehicle [Color figure can be viewed at wileyonlinelibrary.com]

$$d_{\text{tag}} = \hat{p}_{\text{Jetyak}} - p_{\text{UAV}} - b_{\text{GPS}} + w_{\text{tag}} \tag{5}$$

with w_{tag} the noise of the tag measurement.

4.1 | Outlier rejection and adaptive measurement covariance matrix

To operate safely in marine and freshwater environments and close to structures, the localization framework must be robust to outliers. Outliers are expected in the GPS position measurements in GPS-degraded environments but more frequently in the tag measurements. Reflections and occlusions in the image usually result in erroneous measurements that may cause the filter to diverge or the controller to fail. To reject any outlying sensor measurement, a χ^2 test is implemented (Chambers et al., 2014; Chiella, Teixeira, & Pereira, 2019; Hausman, Weiss, Brockers, Matthies, & Sukhatme, 2016). Using the innovation of the sensor measurement the χ^2 test is

$$\chi^2 = (z_i - \hat{z}_i) S_i^{-1} (z_i - \hat{z}_i), \tag{6}$$

where z_i is the sensor measurement, $\hat{z}_i$ the predicted measurement, $S_i = HPH^T + R$ the innovation covariance matrix for a sensor with observation matrix H, R the measurement noise covariance matrix, and P the a priori covariance matrix of the Kalman Filter. If the value of χ^2 is larger than a predefined confidence level, then the measurement is rejected and the filter update is skipped.

It is usually assumed that the measurement uncertainty remains constant during the operation of the system. In our case, however, the uncertainties are expected to change significantly as the mission progresses and the surrounding environment changes. For example, GPS localization accuracy is expected to degrade close to large structures and the tag measurement accuracy reduces with the distance between the platforms.

To deal with the time-varying uncertainty of measurements, an adaptive filter must be used. Covariance Matching (CM), proposed in Mehra (1972), is one of the approaches used in adaptive filters. CM uses the innovation sequence v_i to make the residuals consistent with their theoretical covariances. On a sample of N measurements, the innovation covariance matrix can be approximated by its sample covariance as

$$S_i \approx \frac{1}{N}\sum_{i=1}^{N} v_i v_i^T \qquad (7)$$

and since $S_i = HPH^T + R$, the estimated measurement covariance matrix is approximated as

$$R_i \approx \frac{1}{N}\sum_{i=1}^{N} v_i v_i^T - HPH^T, \qquad (8)$$

where N must be chosen empirically for each sensor to provide some statistical smoothing. Finally, it is important to note here that a nominal value for R_i should be measured in optimal conditions and Equation (8) must only be used to increase R_i.

5 | SYSTEM CONTROL

In this section the different control schemes used will be discussed. First, the state machine for the high-level controls of the system will be explained. Then, a brief introduction to the Jetyak controls will be presented and finally the UAV controller will be discussed in detail.
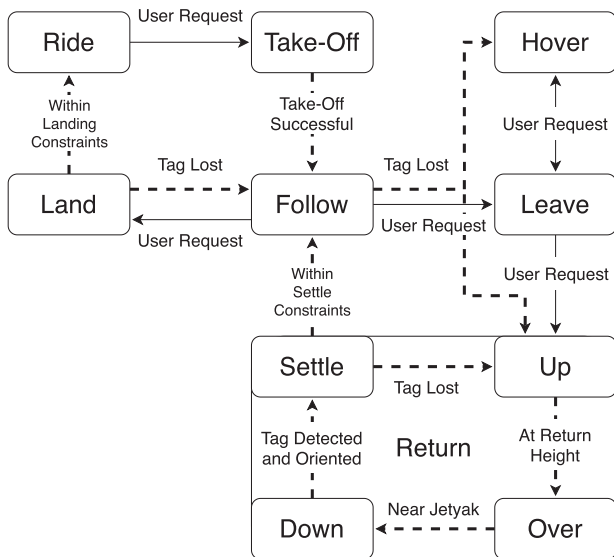

FIGURE 6 State machine diagram of the behaviors. Dashed lines signal automatic transitions with conditions. Solid lines show where user input is required for transitions. Higher level behaviors may also generate user requests

5.1 | Behaviors state machine

The high-level behaviors of the system are designed as a state machine, shown in Figure 6. Seven main behaviors are implemented: *Ride*, *Take-Off*, *Follow*, *Leave*, *Return*, *Land*, and *Hover*. Secondary behaviors are also implemented to handle specific functions, such as waypoint following and spiral search. Transitioning between the states can be either automatic or manual after a user request. The behaviors and their functionality are:

 (i) *Ride*: This is the initial state and is used while the drone is ferried by the Jetyak. In this state, the UAV is on stand-by, waiting for a request. The motors are ensured to be off to allow for a smooth ride on the Jetyak and to conserve energy. The colocalization framework is normally operating in this state to provide the ASV an augmented positioning if needed.
 (ii) *Take-Off*: Every mission starts with a *Take-Off* request. Once the *Take-Off* state is requested, the drone controller starts the motors and the UAV starts ascending. Once a certain altitude is reached, the behavior state automatically switches to *Follow*. In the event of a tag loss during the UAV's ascent the state will first switch to *Follow* and the safety features designed for that state will be utilized.
 (iii) *Follow*: In the *Follow* state, the UAV controller aims to maintain a visual tether with the ASV while flying behind it in a safe distance. Moreover, at this position, the UAV can provide a bird's eye view of the Jetyak's surroundings, as shown in Figure 7. Through in-lab trials using a MoCap system it was determined that the position of 2 m above the ASV and 1.5 m behind has the best tag reliability. In this position, the tag bundle at the mast and two out of the four tags in the landing pad are visible. As a safety feature, if the UAV is unable to track the tag for more than 10 s, then the state is either switched to *Hover* or to *Return*, depending on the configuration.
 (iv) *Leave*: This is an umbrella state that includes all the activities that the UAV completes when it does not maintain a visual tether with the Jetyak. Since there is no visual tether, the CL framework is updated using only the individual GPS data and thus the two platform states become independent. In this state, the drone can be requested to move and operate in a remote location. Using waypoint following control, the drone can fly in patterns and gather data from areas of interest either for mapping or monitoring purposes. A spiral and a quadrilateral coverage pattern trajectory generator are included in this waypoint follow method. When a mission is completed, the state automatically switches to *Hover* where the UAV remains in stand-by until it receives another request.
 (v) *Return*: When the UAV is in the *Leave* state, the *Return* state can be requested to move the drone back in a position close to the Jetyak, where a visual tether can be established. To ensure a smooth transition between the states, the *Return* state maintains an internal state machine as seen in Figure 6. Initially, the *Up* state moves the UAV to an altitude of 10 m that should provide a safe path for the return flight. Meanwhile, the gimbal controller moves the camera to point toward the estimated
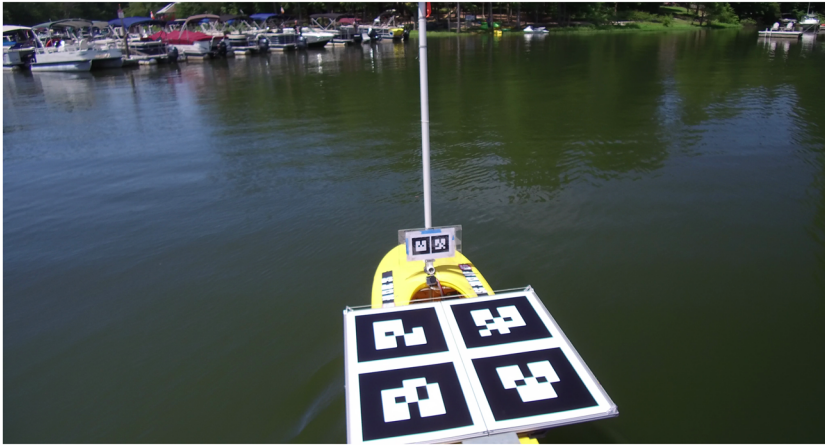
FIGURE 7 A bird's eye view image of the ASV and its surrounding environment. Captured from the drone camera while following the Jetyak. ASV, Autonomous Surface Vehicle [Color figure can be viewed at wileyonlinelibrary.com]

Jetyak position so that a visual tether can be established as soon as possible. Once in a safe altitude, the state switches to *Over* where the UAV moves toward the estimated Jetyak location. Once the UAV approaches the area over the Jetyak, the state switches to *Down* where the UAV slowly descents to an altitude of 3 m above the ASV where both tag bundles should be visible. Then, the state switches to *Settle* and the controller corrects the UAV's orientation to match the Jetyak's heading while it moves the drone to a specified position behind the Jetyak. During the final descent and positioning, a visual tether needs to be established to proceed to the next steps of the *Return* state, otherwise the state is switched back to *Up* and the process repeats. Finally, once the UAV has descended to a safe altitude above the ASV and a visual tether is established, the state transitions to *Follow*.

(vi) *Land*: This state is used for the docking procedure since it transitions from the *Follow* to the *Ride* state. The drone controller moves the UAV from the *Follow* position to a safe position above the landing pad. The UAV starts descending when the drone is within the landing constraints: (i) the UAV velocity is within 15 cm/s of the velocity of the Jetyak, while (ii) it is inside the volume of a pyramid, centered at the center of the landing platform. The apex of the pyramid is at 20 cm over the platform while the base of the pyramid is a square with an 80-cm side. Once the drone is resting safely on the Jetyak then the controller switches off the motors and the state switches to *Ride*. A landing can only be requested from the *Follow* position. If the UAV is in any other state, the *Return* state needs to be requested first to move the UAV to the *Follow* position. For safety reasons, if the visual tether is broken for more than 3 s the state is automatically switched back to *Follow*.

(vii) *Hover*: This is a stand-by state where the UAV maintains its position and altitude while it is away from the ASV. This state is used as a safety measure when the tag is lost in the *Follow* state so that the drone will hold its position and altitude until further action can be taken. Moreover, it can also be used in the case that there is a need to hold position to monitor a point of interest.

The operational cycle of the proposed system is graphically represented in Figure 8, where the concept of operations and the different phases of flight are shown and explained for a sample mission.

5.2 | Jetyak control

The Jetyak utilizes a PID controller to follow a trajectory specified by waypoints. In a recent work by the authors (Moulton et al., 2019), a controller that dynamically updates the waypoints to compensate for disturbances from wind and current was presented. The same approach can be used in GPS-degraded environments where the Jetyak will correct the waypoints according to the position reported by the CL framework.

5.3 | UAV flight control

To control the position of the UAV during all mission stages, an LQR was designed. To evaluate the performance of our controller, a PID controller was also implemented to be used as a baseline. To make the comparison fair, a significant amount of time was invested in tuning both controllers while both of them were tested in the field and were successful on landing the UAV on the Jetyak.

The drone allows for different types of control inputs. On the horizontal axis, the commands can either be velocities, roll and pitch angles, or roll and pitch rates. On the vertical axis, the control input is the velocity along the axis and for rotations around the vertical axis, the command is the yaw rate.

The tuning of the controllers started on the hardware-in-the-loop (HIL) simulator provided by DJI and then gradually moved to indoor flights using a MoCap system to finally reach field trials where they successfully managed to land the UAV on the platform. The two controllers are described in the following subsections while the results from their comparison are presented in Section 6.

FIGURE 8 Concept design highlighting the characteristics of the UAV–USV marsupial system. For each phase of the sample mission the relevant high-level behavior state is provided. ASV, Autonomous Surface Vehicle; UAV, Unmanned Aerial Vehicle; USV, Unmanned Surface Vehicle [Color figure can be viewed at wileyonlinelibrary.com]

5.3.1 | LQR controller

To implement the LQR, a model of the system under control is needed. A high-level position controller was designed on top of the built-in attitude controller. Nevertheless, a model for the closed loop attitude dynamics needs to be derived. Data from the onboard sensors along with the corresponding remote control commands were used to derive second-order models of the roll and pitch dynamics. The State-Variable Filters (SVF) approach along with the Generalized Poisson Moment Functions (GPMF) approach were used to estimate the model parameters (Garnier, Mensler, & Richard, 2003). The structure used for the model is

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -b_{\dot{q},q} & -b_{\dot{q},\dot{q}} \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ c_{\dot{q},q_r} \end{bmatrix} q_r \tag{9}$$

with q representing either the roll (ϕ) or the pitch (θ) angle and q_r the reference signal. By linearizing the system around the stable state of hovering, the horizontal position dynamics are modeled as the second-order system. Around the hovering point, the only force acting on the drone is the gravity. Assuming that there is some damping relative to the velocity and using the small angle approximation we have

$$\begin{bmatrix} \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\lambda_p \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ \pm g \end{bmatrix} q, \tag{10}$$

where p represents either the x-axis ($+g, q = \theta$) or y-axis ($-g, q = \phi$). The only unknown in the model is the parameter λ_p that captures the damping of the system. To estimate this parameter, data from an OptiTrack MoCap system were used in a nonlinear least-squares optimization algorithm.

The parameters for both roll and pitch angles are shown in Table 1. As for the altitude and yaw control, a second-order model for the vertical position z and heading ψ was derived using the vertical velocity u_z and yaw rate u_ψ as inputs:

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -b_{\dot{\psi},\psi} \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ c_{\dot{\psi},\psi^r} \end{bmatrix} u_\psi, \tag{11}$$

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -b_{\dot{z},\dot{z}} \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ c_{\dot{z},z^r} \end{bmatrix} u_z. \tag{12}$$

5.3.2 | PID controller

For the PID case, the selected control inputs were the velocities in the horizontal and vertical axes and the yaw rate. The input to the controller is the position error on the three axes as well as the desired orientation of the drone. Since the control inputs are in the body frame, the position error must be also expressed in the same frame. Thus, the orientation of the drone is used to rotate the body frame to the world frame before calculating the position error.

As discussed in Section 5.1, there are three different states where the drone needs to be in a controlled flight, *Follow*, *Land*, and

TABLE 1 Model parameters found through model identification

Parameter	Roll (ϕ)	Pitch (θ)
$b_{\dot{q},q}$ (s^{-2})	45	46
$b_{\dot{q},\dot{q}}$ (s^{-1})	7.2	7.4
$c_{\dot{q},q^r}$ (s^{-2})	42.5	44
λ_p (s^{-1})	0.05	0.05

Leave. Each of these states requires a different control approach. For example, in the *Follow* state, the drone needs to be able to keep up with a moving boat so it needs to be more aggressive while in the *Return* state a smooth flight is more preferable because it will produce better quality surveys and inspections. For that reason, a gain scheduling approach was followed with three different sets of gains for the three different states. This is however not the case for the LQR controller, since the LQR was able to handle all cases with the same tuning.

6 | EXPERIMENTAL DESIGN AND RESULTS

The proposed system and its subsystems were thoroughly tested. In this section, the experimental setup for each test will be discussed and the results will be presented. First, the CL and controller comparison experiments in the lab and at the field will be presented. Then, a number of field experiments will be discussed that demonstrate the system's effectiveness for surveying and inspection. Videos of the field experiments presented in this paper are available online at https://tinyurl.com/yxbkkdra.

6.1 | CL experiments and results

The CL framework was tested in the lab using an OptiTrack MoCap system to provide an accurate ground truth. Moreover, since the experiment was done indoors, we were able to emulate a GPS-degraded environment. The aim of this experiment was to test the accuracy of the framework in finding the relative position between the two platforms in a GPS-degraded environment.

For this experiment, the Jetyak was static and the drone was flying in different positions while maintaining a visual tether. Specifically, using the drone position from the MoCap system, the drone was flying to a number of prespecified positions of different altitudes and distances from the tag to see how the distance may affect the tag measurements and the whole CL framework. The testing range for the altitude was from 1.5 to 3.5 m and for the horizontal distance was from 2.5 to 4.5 m. During the experiment, the standard deviation for

the Jetyak position reported by the GPS was 5.66 m in the *x*-axis, 3.11 m in the *y*-axis, and 17.94 m in the *z*-axis.

Figure 9 shows the relative position of the two platforms according to the MoCap system, the proposed filter, and using just the GPS measurements. It is clear that in such an environment, using only the GPS position would not be able to provide an accurate enough relative position for an autonomous landing. On the other hand, the filter results have a significantly higher accuracy. Moreover, as it visible in Figure 10, the error does not significantly change with the distance at least at the range that it was tested.

While the GPS positioning usually has a slow drift, the tag measurements are more prone to misdetections that could cause aggressive changes in the relative position of the two platforms. Such aggressive changes in the relative position, especially during the approach and landing maneuvers, may lead to the drone crashing. Reflections are a common source of misdetections especially in the freshwater domain. Figure 11 shows the mast tag being reflected on the platform during the approach maneuver and the results on the tag detection measurements.

Using the proposed outlier rejection method, such measurements do not affect the filter, ensuring the safe operation of the system. Figure 12 shows the CL framework with and without the outlier rejection method. As it can be seen in the figure, the positions of both platforms are affected, with the Jetyak position being affected the most with an error of a few meters. The reason why the UAV position is more robust to outliers is that the UAV GPS measurements are more certain than that of the Jetyak.

As an additional test, in one of the field experiments we purposefully took the drone camera out of focus, as seen in Figure 13, to test the CL framework and also the designed fail–safe capabilities of the high-level controller. The focus of the camera was fixed so that it could clearly see the markers while close to the landing platform so that the landing procedure would not be compromised. On the other hand, in the follow position, the markers are blurry and the vision-based localization was not able to track the tags for extended periods of time while it was producing less accurate data. While the drone was initially able to follow the Jetyak, when 10 s passed without a new tag measurement then, as described in Section 5.1, the drone initially switches to *Hover* before it switches to *Return*. For the *Return*
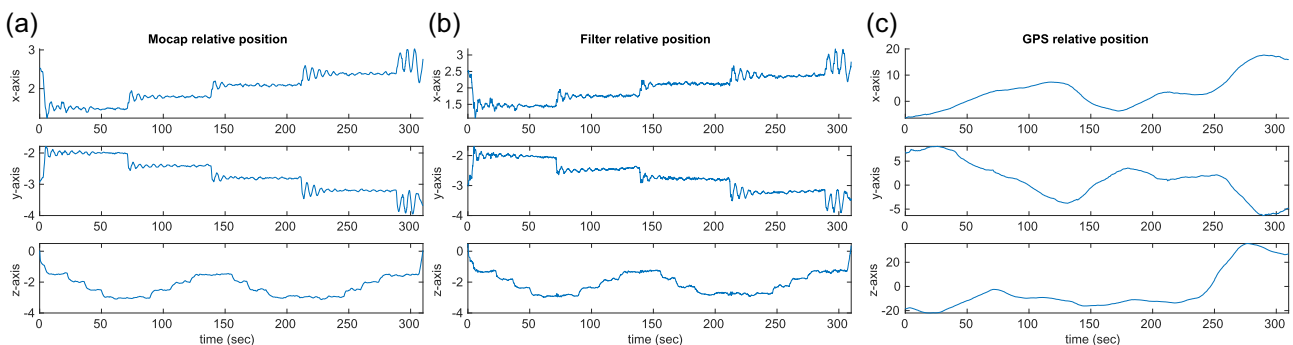

FIGURE 9 Relative position of the two platforms in a GPS-degraded environment according to (a) the OptiTrack Motion Capture system, (b) the cooperative localization framework, and (c) using only the GPS. GPS, Global Positioning System [Color figure can be viewed at wileyonlinelibrary.com]
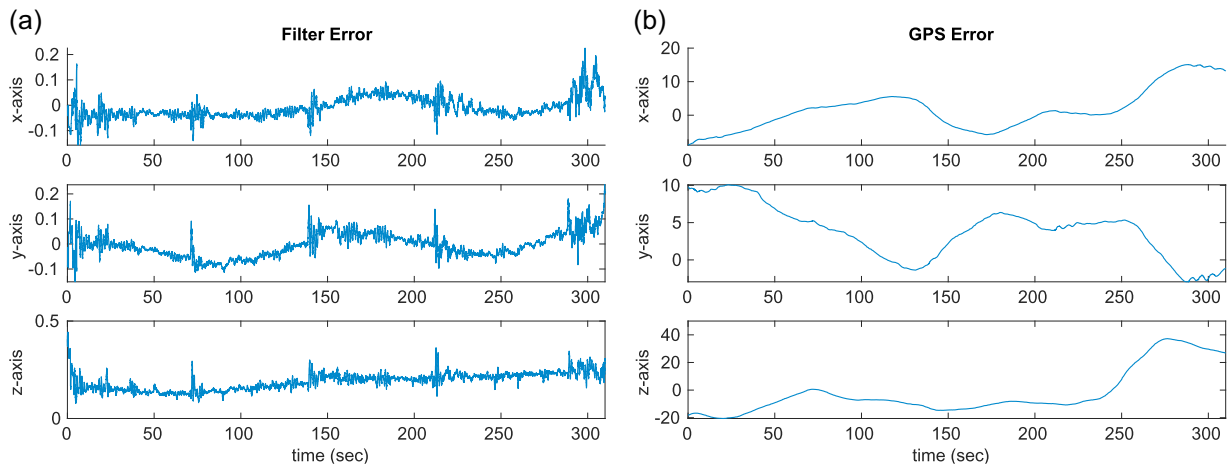
FIGURE 10 The position error according to (a) the filter results and (b) the GPS results. GPS, Global Positioning System [Color figure can be viewed at wileyonlinelibrary.com]
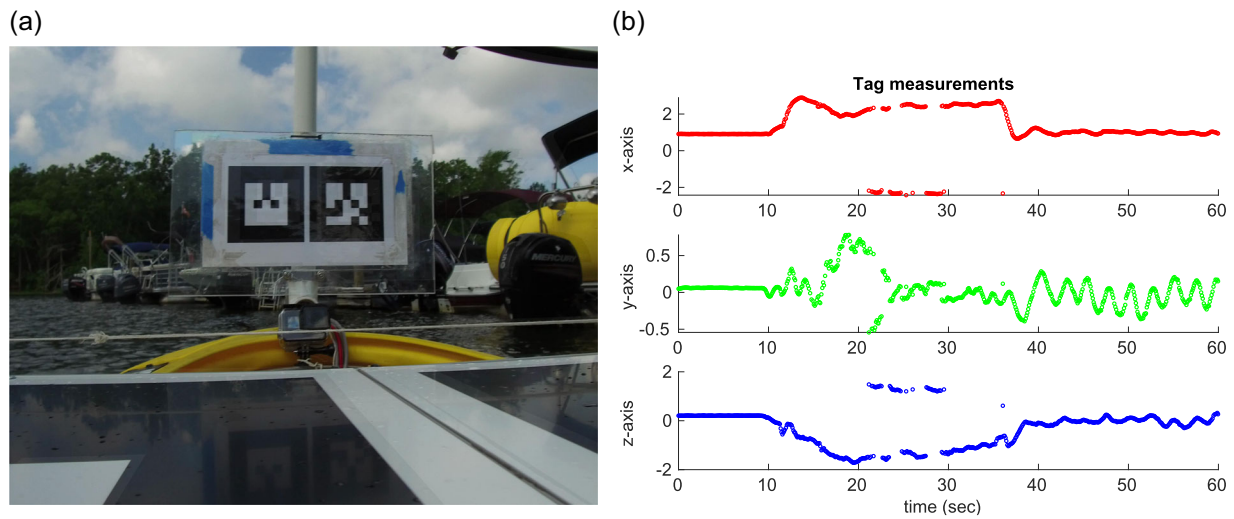

FIGURE 11 (a) Reflections of the mast tag on the platform are a cause of misdetections; (b) while the drone tries to approach the platform and land, the reflections on the platform may cause outliers in the tag measurements as seen around the 20 s and 30 s mark of the experiment [Color figure can be viewed at wileyonlinelibrary.com]
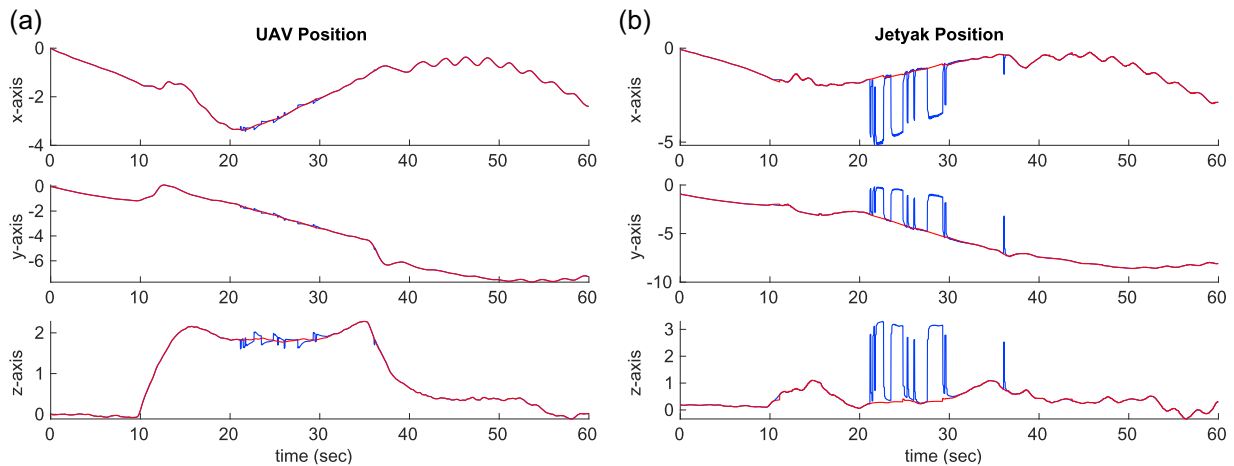
FIGURE 12 (a) The estimated UAV position and (b) the estimated Jetyak position; blue line is without outlier rejection and red line is with outlier detection. UAV, Unmanned Aerial Vehicle [Color figure can be viewed at wileyonlinelibrary.com]

(a)   (b)
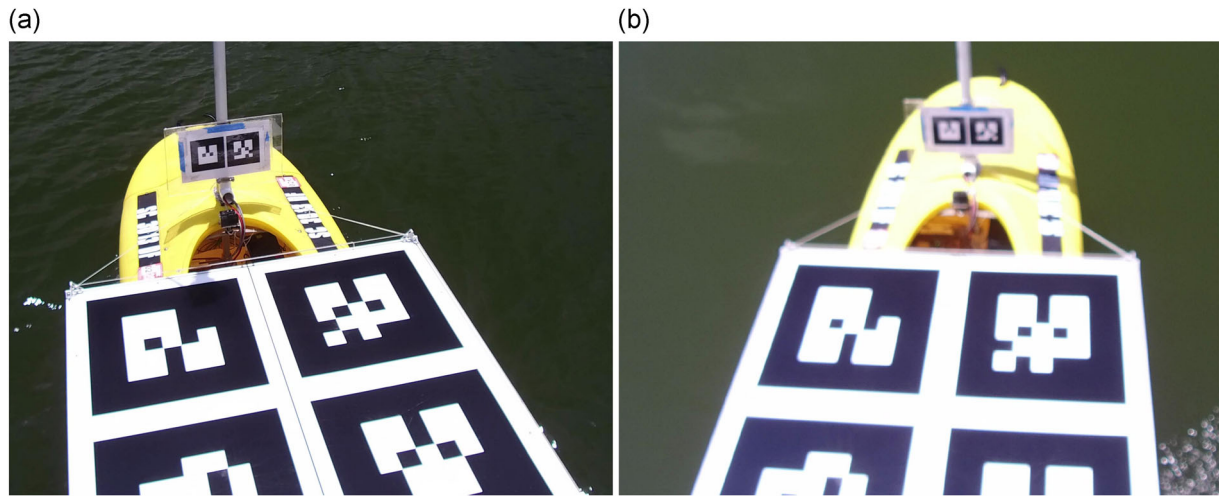
FIGURE 13 A cropped view of the Jetyak from the drone with the camera (a) in focus and (b) out of focus [Color figure can be viewed at wileyonlinelibrary.com]

behavior, the drone controller uses the platform positions estimated from the localization framework. Thus, if the framework was not able to discard outliers and adapt to any erroneous measurements provided by the camera, the system would not be able to continue its operation. However, the system was able to manage this challenging situation, switch between the different behavior states and finally land safely on the platform.

6.2 | Controller experiments and results

As described in Section 5, both of the developed controllers were tested in the field to make sure that they were able to successfully land the drone on a moving platform. Initially, both controllers were tuned using an HIL simulator provided by DJI, then in the lab using a MoCap system and finally in the field. Since the most challenging aspect of any mission is the landing, the controllers were judged by their effectiveness on this task. While the controllers were tuned to land the drone on a moving platform, a static landing was used for the comparison. The reason for this is that in a static landing there are less uncontrolled variables that might affect the results. In the field, the wind and current conditions might change during the comparison affecting the results and leading to wrong conclusions. Furthermore, in the lab there is not enough space to perform a landing in a moving platform.

The three metrics used to compare the controllers are the time it took to land the drone from the moment of the request until the drone was resting on the platform with the motors off, the maximum overshoot and finally the error on the landing position. The missions that the system undertakes are time critical, meaning that the drone should use its limited flight time effectively. Moreover, in cases where a mission needs to be abruptly terminated, the drone needs to return and dock on the boat as fast as possible. As described in Section 3, the Jetyak is equipped with a mast that holds the communication hardware and is used to host other mission-specific sensors, such as an anemometer or a lidar. As a result, the maximum overshoot in the landing

process has an increased significance to minimize the chance of the drone crashing on the mast. Finally, the position error shows how close to the center of the platform is the drone when it lands. It is worth noting here that the drone is trying to land in an area around the center of the platform which is selected to be a 15 cm × 15 cm square.

As mentioned earlier, the controllers were compared in all three domains: simulation, lab, and field trials. In all three domains, the drone starts on the platform, takes-off, and flies to the follow position that is 1.1 m back and 1.6 m up and after 20 s it starts the landing procedure. For the simulation results, a total number of 30 landings were executed for each controller in six different wind conditions. Initially the wind is set to zero and then it is increased to 3 and 4 m/s in both the lateral and longitudinal directions. In the last case, the wind is set to 3 m/s in both directions. For the lab experiment, the MoCap system was used to provide the position of the platform and the drone during five consecutive landings. Finally, two sets of five consecutive landings were done at the field, where the position of the two platforms was provided by the CL framework.

Table 2 shows the results for both controllers in all three domains. In all cases, the LQR controller was able to land the drone significantly faster than the PID controller. Moreover, the

TABLE 2 Landing results for the three different test types showing the mean values along the experiments for all three metrics

Controller	Experiment	Landing time (s)	Overshoot (m)	Landing error (m)
LQR	Simulation	8.373	0.103	0.146
	Motion capture	9.576	0.344	0.088
	Outdoors	8.382	0.331	0.133
PID	Simulation	17.032	0.281	0.020
	Motion capture	15.406	0.552	0.128
	Outdoors	16.833	0.355	0.101

Abbreviations: LQR, Linear Quadratic Regulator; PID, Proportional–Integral–Derivative.
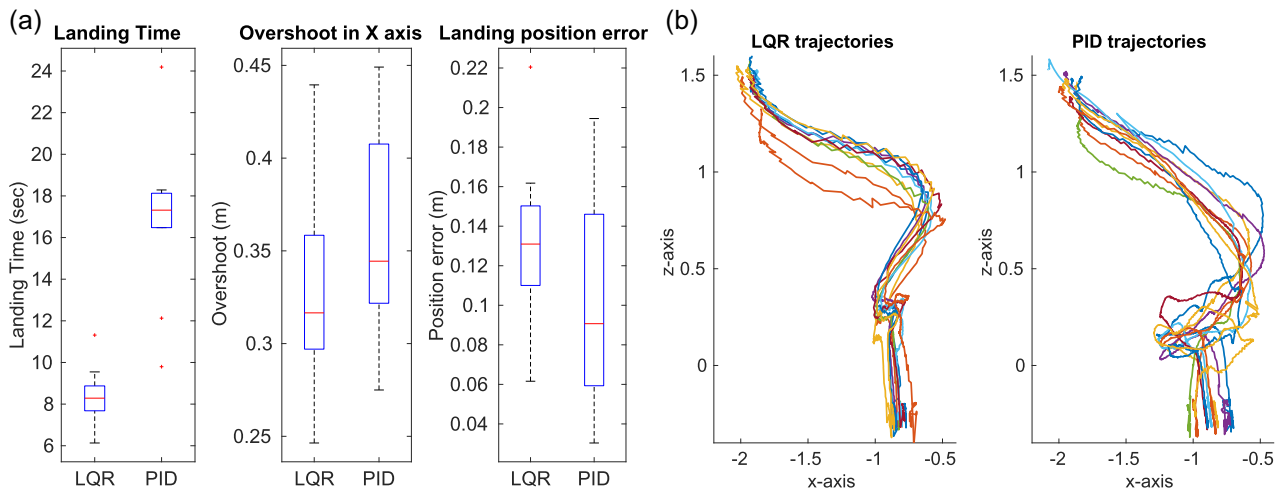
FIGURE 14 (a) Box and whiskers plots showing the landing time, overshoot, and final error for the 10 field landings. (b) The different trajectories from the follow position to the landing position. LQR, Linear Quadratic Regulator; PID, Proportional–Integral–Derivative [Color figure can be viewed at wileyonlinelibrary.com]

overshoot is lower for the LQR in all cases. The PID controller shows smaller errors on the landing position in most cases but both controllers were able to land inside the predefined area. Figure 14 shows the box and whiskers plots for the 10 field landings along with the trajectories followed in the indoor landings. One can notice that the LQR had a more consistent behavior between the different trials. Finally, it is worth mentioning that the measured overshoot on the static landings was higher than the one observed on landings in a moving platform especially for the PID. As mentioned before, the two controllers were tuned to land on a moving platform and as a result they had a more aggressive response.

Switching between the *Take-Off* to *Follow* and finally to *Land* states was thoroughly tested on the field with a moving platform.

The first field deployments aimed to test the separation and docking mechanism of the system and its safety. Moreover, to test the robustness of the system, the *Take-off*, *Follow*, and *Land* sequence was repeated in different weather conditions. The system was able to complete the sequence even when flying under significant side winds and with the boat swaying due to waves, as shown in Figure 15. Finally, the sequence was also repeated consecutive times to show that the landing position is a viable position for the drone to take-off again and repeat the process. The position of the two platforms and the control effort of the drone during one of the field landings are shown in Figure 16. In this particular experiment the drone was able to land on the boat in 14 s while the ASV was moving with a velocity of around 2 km/h.
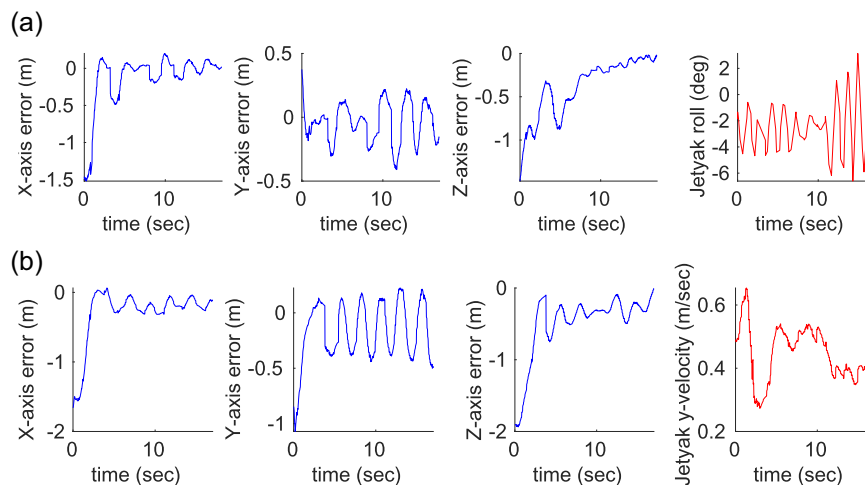
FIGURE 15 Landing in challenging conditions with the LQR controller. (a) Landing under sway. Waves in the water surface can significantly affect the Jetyak's attitude. The figure shows the errors in the *x*-, *y*-, and *z*-axes along with the roll angle of the Jetyak during landing. (b) Landing under side wind. The presence of significant side winds is affecting both platforms by introducing a velocity in the *y*-axis. The figure shows the errors in the *x*-, *y*-, and *z*-axes along with the *y*-axis velocity of the Jetyak in its body frame. LQR, Linear Quadratic Regulator [Color figure can be viewed at wileyonlinelibrary.com]
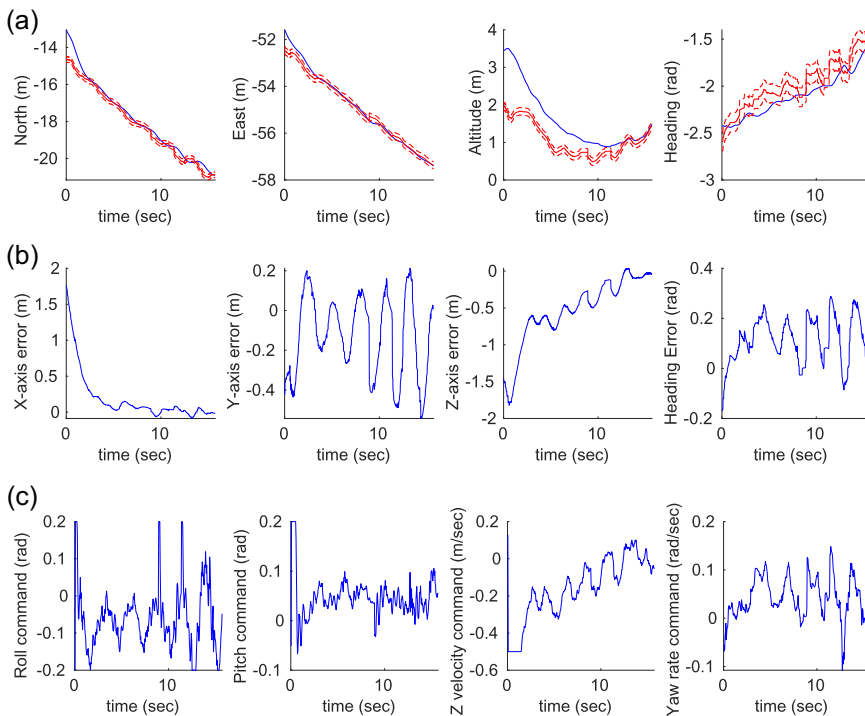
FIGURE 16 (a) The position of the Jetyak and the UAV during a field landing in a local ENU system. The blue line is the UAV position and the red solid line is the Jetyak position. The red dashed lines show the landing constraints. (b) The landing error on the UAV's body frame. Notice that the heading affects the x- and y-axes errors. (c) The control effort commands of the drone during landing using the LQR controller. ENU, East-North-Up; LQR, Linear Quadratic Regulator; UAV, Unmanned Aerial Vehicle [Color figure can be viewed at wileyonlinelibrary.com]

6.3 | Surveying and inspection results

The system was tested on a number of successful field deployments in two different freshwater environments, Lake Murray and the Congaree River, both located in South Carolina, USA. During these deployments, the capabilities of the system were tested and its inspection effectiveness evaluated. A successful deployment is one where the drone starts on the Jetyak, flies to the region of interest, completes its mission, then returns to the Jetyak, and performs a landing. While both controllers were used in the field deployments, only a small number of them were completed using the PID controller and the vast majority was done using the LQR controller.

To show the capability of the UAV to provide a bird's-eye-view perspective to the boat, a series of deployments were done with the drone following the ASV while the boat was following a trajectory autonomously using waypoint navigation. In these tests, the Jetyak would start following a desired trajectory and at some point it would deploy the drone to get the augmented field of view. The drone would follow behind the ASV at a safe distance while tracking the position of the Jetyak in a local world frame. When the ASV had finished its trajectory it would signal the UAV to start the landing procedure.

The *Leave* mode is designed to control the UAV when it operates remotely. As described in Section 5.1, the system can use predefined waypoints to cover an area of interest and gather data. In our deployments, three different types of missions were completed. First, the drone would move from the *Follow* position to a single waypoint where the mode would switch to *Hover* to fly in position and wait for further instructions. This type of mission is useful when information about a specific region needs to be gathered for a duration of time. As an example, in the case that the Jetyak needs to traverse a GPS-degraded environment, the drone would fly in a position with good GPS coverage and stay there while tracking the Jetyak's position. In the second type, the drone would move to a waypoint and then a spiral pattern generator would produce a number of waypoints to create a spiral trajectory of the desired radius and number of rotations. This type of experiment is used in surveying missions to map an area of interest. Finally, using three different waypoints, a quadrilateral trajectory can be generated with the desired number of sweeps. This type of mission can either be used for surveying and mapping or for the inspection of a structure. The trajectories of a spiral trajectory mission and a quadrilateral trajectory mission can be seen in Figure 17.

The primary application of the system is to automate surveys of near-shore environments and inspections of structures. In Lake Murray, a dock and its surrounding area were surveyed 6 months apart to show the change in the surrounding area over that period. Figure 17a shows the mission trajectory for the summer deployment while Figure 18 shows the high-definition maps created from the survey data using the automatic stitching method described in Brown and Lowe (2007). One can see the change in the water level, the difference in the vegetation as well as the boards of the dock which have been replaced in the structure.

To demonstrate the inspection capabilities of the system, a bridge at Congaree River was selected for inspection. There, the system was deployed multiple times to take videos of the structure and the surrounding area from different angles with Figure 17b, showing one of such deployments. Using COLMAP (Schönberger & Frahm, 2016; Schönberger, Zheng, Frahm, & Pollefeys, 2016), a Structure-from-Motion software, a dense 3D reconstruction of the structure was generated, as shown in Figure 19. The bridge under inspection, the surrounding environment and the nearby transmission towers can all be seen in the reconstruction. This reconstruction
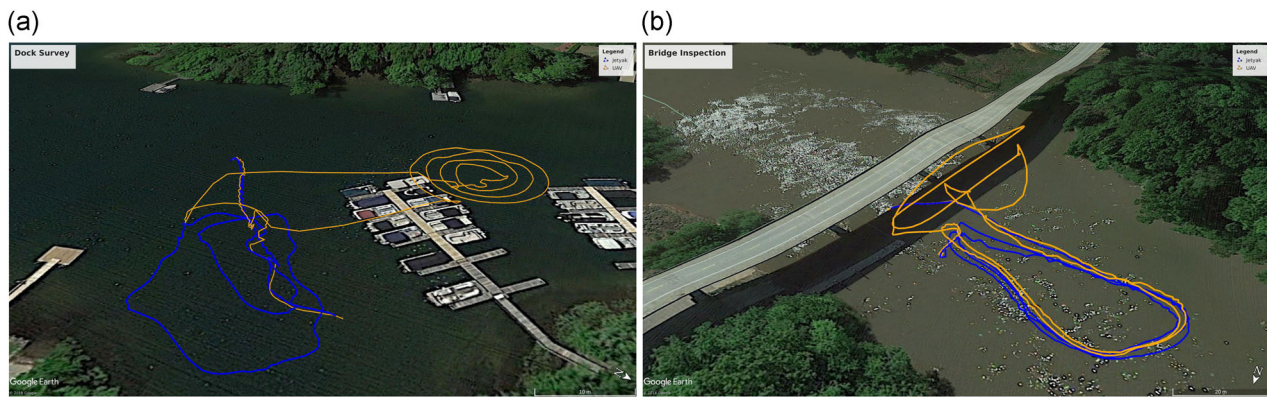
FIGURE 17 The trajectories of the UAV (orange) and ASV (blue) during (a) a spiral survey of the docks and (b) a bridge inspection. ASV, Autonomous Surface Vehicle; UAV, Unmanned Aerial Vehicle [Color figure can be viewed at wileyonlinelibrary.com]
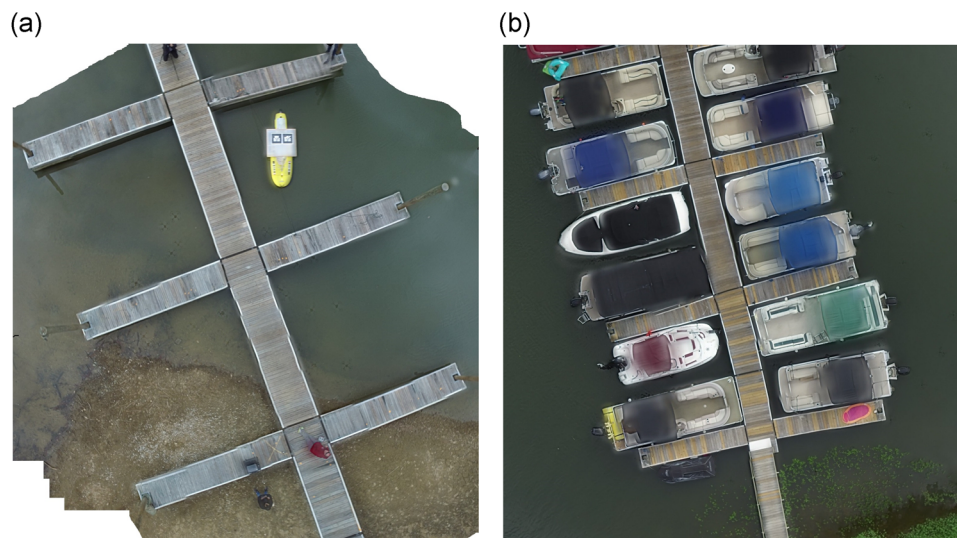

FIGURE 18 Mosaic created from spiral survey of a dock and shore in (a) winter and (b) summer. The change in the water level is noticeable as well as the blooming of the near-shore vegetation. The spiral trajectory that resulted in the mosaic in (b) can be seen in Figure 17a [Color figure can be viewed at wileyonlinelibrary.com]

FIGURE 19 Dense rendering of the inspected bridge and its surrounding environment at 33.753000°N, 80.645223°W. The trajectory of the mission that resulted in this rendering is shown in Figure 17b [Color figure can be viewed at wileyonlinelibrary.com]

can be used to assess the state of the bridge, measure the water level, and monitor the surrounding environment.

7 | LESSONS LEARNED AND FUTURE WORK

During the 6-month span of field deployments, we had the opportunity to thoroughly test the system under different conditions that shaped our final system design. One of the most critical parts of the development was the tag detection and localization system design. By choosing to use AR tags we maximized the rate of positioning measurements but this also created the need for adding a robust outlier rejection in our framework. While initially we used laminated prints for our tag bundles, later on we replaced them with prints in aluminum sheets. The latter had a less reflective surface that was not affected by flapping or warping.

The extensive period of field trials showed that the most suitable weather conditions for deployment was overcast with low wind speeds. In sunny days, the reflection of the sun in the landing pad would obscure a large portion of the tag bundle rendering it undetectable. To limit the effect of such reflections, we mounted a polarized neutral density filter to the camera. Since both of the platforms used are light for their domains, the wind was able to affect the motion of both platforms making their control harder. Even without significant wind conditions and with the ASV idling, the draft generated from the UAV propellers was able to push around the ASV making the landing process challenging even in such conditions.

The landing platform is another important part of the developed system. The platform needs to ensure that the UAV will be safely docked while it is ferried to and from positions of interest. In our current design, a thread that runs across the landing platform is used as a safety mechanism. Future work includes improvements in this area with an effective locking mechanism that will allow the Jetyak to move faster or operate in adverse weather conditions. Other improvements to the platform can be self-leveling capabilities (Conyers, Vitzilaios, Rutherford, & Valavanis, 2015) to compensate for the ASV's motion, and charging capabilities (Malyuta et al., 2019) to further increase the operation time of the UAV.

In all of our field deployments, safety pilots for both the UAV and the ASV were present on site. While all the results presented in this study are from fully autonomous deployments, there were cases that the safety pilot had to intervene and stop the process. One case is when a platform would approach an obstacle, since neither of the platforms has any active obstacle avoidance capability. The lack of obstacle avoidance capabilities is also the reason why in the *Return* state the UAV first ascends to 10 m and then moves toward the Jetyak so that it can safely fly above any obstacles or vegetation.

Another case where a safety pilot must intervene is after a number of failed attempts to successfully implement the *Return* state. As described in Section 5.1, if a visual tether is not established during the last phases of the *Return* state, the process repeats. If the colocalization framework has significantly diverged, the UAV might not be able to locate the ASV. In such a case, the UAV relies on the safety pilot to intervene and find a safe landing spot before the battery runs out. To increase the autonomy level of the proposed design, an obstacle avoidance framework needs to be added as well as a safety measure for autonomous emergency landing when the ASV cannot be detected.

While this study focuses on the UAV subsystem, future work will further explore the ASV subsystem and how it can be augmented with the use of the companion UAV. Since the communication framework between the two platforms has been already developed, what remains to be done is to develop further ASV capabilities that will use the relayed information to augment its control and operation, followed by another long-term testing campaign.

8 | CONCLUSION

A marsupial robotic system composed of a UAV and an ASV is presented in this study. This heterogeneous robotic team exploits the advantages of each platform to increase the application range of the system. Taking advantage of the long operation time of the ASV, the UAV can be ferried to a remote location and thus increase its operation range. By utilizing the increased field of view that a flying platform can provide, the ASV can safely and efficiently navigate dangerous regions. Furthermore, the UAV can be used as a surveying and inspection platform generating high-quality maps and 3D reconstructions of freshwater environments and structures. A large number of successful field trials demonstrate the capabilities of the proposed system as an autonomous surveying, inspection, and monitoring system.

ORCID
Nikolaos Vitzilaios http://orcid.org/0000-0003-1252-586X
Ioannis Rekleitis http://orcid.org/0000-0002-0655-224X

REFERENCES
Anderson, M., McKay, M., & Richardson, B. (1996). Multirobot automated indoor floor characterization team. In *Proceedings of the IEEE International Conference on Robotics and Automation* (Vol. 2, pp. 1750–1753). Minneapolis, MN.

Araar, O., Aouf, N., & Vitanov, I. (2017). Vision based autonomous landing of multirotor UAV on moving platform. *Journal of Intelligent & Robotic Systems, 85*(2), 369–384.

Balaram, B., Canham, T., Duncan, C., Grip, H. F., Johnson, W., Maki, J., ... Zhu, D. (2018). Mars helicopter technology demonstrator. In *2018 AIAA Atmospheric Flight Mechanics Conference*. Kissimmee, FL.

Balmer, G. R. (2015). *Modelling and control of a fixed-wing UAV for landings on mobile landing platforms* (Ph.D. Thesis). KTH Royal Institute of Technology, Stockholm, Sweden.

Brommer, C., Malyuta, D., Hentzen, D., & Brockers, R. (2018). Long-duration autonomy for small rotorcraft UAS including recharging. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 7252–7258). Madrid, Spain.

Brown, M., & Lowe, D. G. (2007). Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1), 59–73.

Burgard, W., Moors, M., Fox, D., Simmons, R., & Thrun, S. (2000). Collaborative multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation* (Vol. 1, pp. 476–481). San Francisco, CA.

Chahine, G., & Pradalier, C. (2018). Survey of monocular SLAM algorithms in natural environments. In *2018 15th Conference on Computer and Robot Vision (CRV)* (pp. 345–352). Toronto, ON, Canada.

Chambers, A., Scherer, S., Yoder, L., Jain, S., Nuske, S., & Singh, S. (2014). Robust multi-sensor fusion for micro aerial vehicle navigation in GPS-degraded/denied environments. In *2014 American Control Conference* (pp. 1892–1899). Portland, OR.

Chaves, S. M., Wolcott, R. W., & Eustice, R. M. (2015). NEEC research: Toward GPS-denied landing of unmanned aerial vehicles on ships at sea. *Naval Engineers Journal*, 127(1), 23–35.

Chiella, A. C. B., Teixeira, B. O. S., & Pereira, G. A. S. (2019). State estimation for aerial vehicles in forest environments. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 890–898). Atlanta, GA.

Christidi-Loumpasefski, O. O., Nanos, K., & Papadopoulos, E. (2017). On parameter estimation of space manipulator systems using the angular momentum conservation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5453–5458). Singapore.

Conyers, S. A., Vitzilaios, N. I., Rutherford, M. J., & Valavanis, K. P. (2015). A mobile self-leveling landing platform for VTOL UAVs. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 815–822). Seattle, WA.

Cowen, S., Briest, S., & Dombrowski, J. (1997). Underwater docking of autonomous undersea vehicles using optical terminal guidance. In *Oceans '97. MTS/IEEE Conference Proceedings* (Vol. 2, pp. 1143–1147). Halifax, NS, Canada: MTS/IEEE.

Dieudonné, Y., Labbani-Igbida, O., & Petit, F. (2010). Deterministic robot-network localization is hard. *IEEE Transactions on Robotics*, 26(2), 331–339.

Falanga, D., Zanchettin, A., Simovic, A., Delmerico, J., & Scaramuzza, D. (2017). Vision-based autonomous quadrotor landing on a moving platform. In *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)* (pp. 200–207). Shanghai, China.

Fiala, M. (2005). ARTag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)* (Vol. 2, pp. 590–596). San Diego, CA.

Garnier, H., Mensler, M., & Richard, A. (2003). Continuous-time model identification from sampled data: Implementation issues and performance evaluation. *International Journal of Control*, 76(13), 1337–1357.

Gautam, A., Sujit, P., & Saripalli, S. (2014). A survey of autonomous landing techniques for UAVs. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 1210–1218). Orlando, FL.

Ghamry, K. A., Dong, Y., Kamel, M. A., & Zhang, Y. (2016). Real-time autonomous take-off, tracking and landing of UAV on a moving UGV platform. In *2016 24th Mediterranean Conference on Control and Automation (MED)* (pp. 1236–1241). Athens, Greece.

Hausman, K., Weiss, S., Brockers, R., Matthies, L., & Sukhatme, G. S. (2016). Self-calibrating multi-sensor fusion with probabilistic measurement validation for seamless sensor switching on a UAV. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4289–4296). Stockholm, Sweden.

Hentzen, D., Stastny, T., Siegwart, R., & Brockers, R. (2019). Disturbance estimation and rejection for high-precision multirotor position control. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Hood, S., Benson, K., Hamod, P., Madison, D., O'Kane, J. M., & Rekleitis, I. (2017). Bird's eye view: Cooperative exploration by UGV and UAV. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 247–255). Miami, FL.

Hourani, H., Wolters, P., Hauck, E., & Jeschke, S. (2011). A marsupial relationship in robotics: A survey. In S. Jeschke, H. Liu & D. Schilberg (Eds.), *Intelligent Robotics and Applications. ICIRA 2011. Lecture Notes in Computer Science* (7101, pp. 335–345). Berlin, Heidelberg: Springer.

Jimenez-Cano, A. E., Sanchez-Cuevas, P. J., Grau, P., Ollero, A., & Heredia, G. (2019). Contact-based bridge inspection multirotors: Design, modeling, and control considering the ceiling effect. *IEEE Robotics and Automation Letters*, 4(4), 3561–3568.

Kalaitzakis, M., Kattil, S. R., Vitzilaios, N., Rizos, D., & Sutton, M. (2019). Dynamic structural health monitoring using a DIC-enabled drone. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 321–327). Atlanta, GA.

Karapetyan, N., Moulton, J., Lewis, J. S., Quattrini Li, A., Okane, J. M., & Rekleitis, I. (2018). Multi-robot Dubins coverage with autonomous surface vehicles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2373–2379). Brisbane, Australia.

Karapetyan, N., Moulton, J., & Rekleitis, I. (2019). Meander based river coverage by an autonomous surface vehicle. In *12th Conference on Field and Service Robotics (FSR)*. Tokyo, Japan.

Kislik, C., Dronova, I., & Kelly, M. (2018). UAVs in support of algal bloom research: A review of current applications and future opportunities. *Drones*, 2(4), 35.

Kurazume, R., & Hirose, S. (2000). An experimental study of a cooperative positioning system. *Autonomous Robots*, 8(1), 43–52.

Kurazume, R., Nagata, S., & Hirose, S. (1994). Cooperative positioning with multiple robots. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation* (pp. 1250–1257). San Diego, CA.

Lindemuth, M., Murphy, R., Steimle, E., Armitage, W., Dreger, K., Elliot, T., … Griffin, C. (2011). Sea robot-assisted inspection. *IEEE Robotics & Automation Magazine*, 18(2), 96–107.

Malyuta, D., Brommer, C., Hentzen, D., Stastny, T., Siegwart, R., & Brockers, R. (2019). Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition. *Journal of Field Robotics*, 37(1), 137–157.

Marques, F., Lourenço, A., Mendonça, R., Pinto, E., Rodrigues, P., Santana, P., & Jose, B. (2015). A critical survey on marsupial robotic teams for environmental monitoring of water bodies, *OCEANS 2015—Genova* (pp. 1–6). Genoa, Italy: MTS/IEEE.

Mathe, K., Busoniu, L., Barabas, L., Iuga, C.-I., Miclea, L., & Braband, J. (2016). Vision-based control of a quadrotor for an object inspection scenario. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 849–857). Arlington, VA.

Mehra, R. (1972). Approaches to adaptive filtering. *IEEE Transactions on Automatic Control*, 17(5), 693–698.

Mendonça, R., Marques, M. M., Marques, F., Lourenco, A., Pinto, E., Santana, P., … Jose, B. (2016). A cooperative multi-robot team for the surveillance of shipwreck survivors at sea, *OCEANS 2016 Monterey* (13, pp. 1–6). Monterey, CA: MTS/IEEE.

Moulton, J., Karapetyan, N., Bukhsbaum, S., McKinney, C., Malebary, S., Sophocleous, G., … Rekleitis, I. (2018). An autonomous surface vehicle for long term operations, *OCEANS 2018 Charleston* (pp. 1–10). Charleston, SC: MTS/IEEE.

Moulton, J., Karapetyan, N., Kalaitzakis, M., Quattrini Li, A., Vitzilaios, N., & Rekleitis, I. (2019). Dynamic autonomous surface vehicle controls, aiding disaster response and enabling responsible port growth. In *12th Conference on Field and Service Robotics (FSR)*. Tokyo, Japan.

Murphy, R. R., Ausmus, M., Bugajska, M., Ellis, T., Johnson, T., Kelley, N., … Pollock, L. (1999). Marsupial-like mobile robot societies. In *Proceedings of the Third Annual Conference on Autonomous Agents—AGENTS '99* (pp. 364–365). New York, NY.

Murphy, R. R., Steimle, E., Griffin, C., Cullins, C., Hall, M., & Pratt, K. (2008). Cooperative use of unmanned sea surface and micro aerial vehicles at Hurricane Wilma. *Journal of Field Robotics, 25*(3), 164–180.

Myint, M., Yonemori, K., Yanou, A., Minami, M., & Ishiyama, S. (2015). Visual-servo-based autonomous docking system for underwater vehicle using dual-eyes camera 3D-pose tracking. In *2015 IEEE/SICE International Symposium on System Integration (SII)* (pp. 989–994). Nagoya, Japan.

Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation* (pp. 3400–3407). Shanghai, China.

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., … Ng, A. (2009). ROS: An open-source robot operating system. In *ICRA Workshop on Open Source Software*. Kobe, Japan.

Rekleitis, I., Dudek, G., & Milios, E. (1997). Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *International Joint Conference in Artificial Intelligence (IJCAI)* (Vol. 2, pp. 1340–1345). Nagoya, Japan.

Rekleitis, I., Dudek, G., & Milios, E. (1998). On multiagent exploration, *Vision Interface* (pp. 455–461). Vancouver, BC, Canada: Canadian Image Processing and Pattern Recognition Society.

Rekleitis, I., Dudek, G., & Milios, E. (2001a). Multi-robot collaboration for robust exploration. *Annals of Mathematics and Artificial Intelligence, 31*(1), 7–40.

Rekleitis, I., Martin, E., Rouleau, G., L'Archevêque, R., Parsa, K., & Dupuis, E. (2007). Autonomous capture of a tumbling satellite. *Journal of Field Robotics, 24*(4), 275–296.

Rekleitis, I., Sim, R., Dudek, G., & Milios, E. (2001b). Collaborative exploration for the construction of visual maps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vol. 3, pp. 1269–1274). Maui, HI: IEEE/RSJ.

Roumeliotis, S., & Bekey, G. (2002). Distributed multirobot localization. *IEEE Transactions on Robotics and Automation, 18*(5), 781–795.

Sanchez-Lopez, J. L., Saripalli, S., Campoy, P., Pestana, J., & Fu, C. (2013). Toward visual autonomous ship board landing of a VTOL UAV. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 779–788). Atlanta, GA.

Schönberger, J. L., & Frahm, J.-M. (2016). Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4104–4113). Las Vegas, NV.

Schönberger, J. L., Zheng, E., Frahm, J.-M., & Pollefeys, M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision—ECCV 2016. Lecture Notes in Computer Science* (Vol. 9907, pp. 501–518). Cham: Springer.

Vlantis, P., Marantos, P., Bechlioulis, C. P., & Kyriakopoulos, K. J. (2015). Quadrotor landing on an inclined platform of a moving ground vehicle. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2202–2207). Seattle, WA.

Wang, J., & Olson, E. (2016). AprilTag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4193–4198). Daejeon, South Korea.

Wang, L., & Bai, X. (2018). Quadrotor autonomous approaching and landing on a vessel deck. *Journal of Intelligent & Robotic Systems, 92*(1), 125–143.