Autonomous Navigation System for a Delivery Drone

Victor R. F. Miranda[1] · Adriano M. C. Rezende[1] · Thiago L. Rocha[1] · Héctor Azpúrua[2,3] ·
Luciano C. A. Pimenta[1] · Gustavo M. Freitas[1]

Abstract

The use of delivery services is an increasing trend worldwide, further enhanced by the COVID pandemic. In this context, drone delivery systems are of great interest as they may allow for faster and cheaper deliveries. This paper presented a navigation system that makes feasible the delivery of parcels with autonomous drones. The system generates a path between a start and a final point and controls the drone to follow this path based on its localization obtained through GPS, 9DoF IMU, and barometer. In the landing phase, information of poses estimated by a marker (ArUco) detection technique using a camera, ultrawideband (UWB) devices, and the drone's software estimation are merged by utilizing an extended Kalman filter algorithm to improve the landing precision. A vector field-based method controls the drone to follow the desired path smoothly, reducing vibrations or harsh movements that could harm the transported parcel. Real experiments validate the delivery strategy and allow the evaluation of the performance of the adopted techniques. Preliminary results state the viability of our proposal for autonomous drone delivery.

Keywords Autonomous drones · Unmanned aerial vehicle delivery · Path planning · Localization · Vector field control

1 Introduction

Autonomous robots have been studied for a long time, and the rising demand for automated solutions for real-world problems has accelerated research in the area. Such problems involve housekeeping tasks (vacuum and lawnmower robots), military missions (rescue, patrol, attacks), security applications (surveillance, exploration), and industrial operations (production and logistics), among others.

In the past few years, the online shopping market has grown significantly, and ever since, retail and mail companies seek to make autonomous drone delivery a reality. The global online food delivery services reached \$136.4bn in 2020, a 27% increase from the same period in 2019 (AJOT 2021). This increase should continue in the following years, with an expectation of \$182.3bn in 2024. As a result, delivery services face a high demand for orders and sometimes cannot maintain a short delivery time. Thus, new and innovative means of transportation must be developed to increase efficiency and cope with the increasing demand.

Autonomous delivery presents itself as a very convenient alternative, particularly in social isolation periods, such as those experienced by many countries in 2020 and 2021 due to the COVID-19 pandemic. In situations like this, physical contact between people should be reduced as much as possible, especially with people from risk groups. Therefore, the transportation of goods by autonomous quadrotors could prevent any physical contact with the customer, thus maintaining the World Health Organization (WHO) recommendations and preventing the proliferation of the virus.

✉ Victor R. F. Miranda
victormrfm@ufmg.br

Adriano M. C. Rezende
adrianomcr@ufmg.br

Thiago L. Rocha
thiagolages@ufmg.br

Héctor Azpúrua
hector.azpurua@dcc.ufmg.br ; hector.azpurua@itv.org

Luciano C. A. Pimenta
lucpim@ufmg.br

Gustavo M. Freitas
gustavomfreitas@ufmg.br

1 Escola de Engenharia, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

2 Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

3 Instituto Tecnológico Vale (ITV) - Mining, Ouro Preto, Brazil

Some of the world's biggest companies, such as Amazon, UPS, and Alphabet, are making advances toward drone delivery services. According to Schneider (2020), Flight-Forward, the UPS branch responsible for drone flights, has already achieved air carrier certification, which allows it to deliver small packages with drones. Wing, a division of Alphabet, launched the USA's first small commercial delivery service in Christiansburg, Virginia. These achievements happened in late 2019 and show that the market for drone delivery is undoubtedly gaining ground worldwide. In late 2020, Brazil's National Aviation Agency (ANAC) granted the first authorization to a private company, SpeedBird, to perform drone cargo tests in Brazillian urban areas (ANAC 2020).

In this context, this paper presents a methodology for enabling autonomous drone deliveries. After generating a path between two points of interest, the drone's location is obtained through its GPS (Global Positioning System), 9DoF IMU (Inertial Measurement Unit), and a barometer. A vector field control algorithm then guides the quadrotor into the desired path. However, in drone delivery, reproducible safe landings in urban areas are a critical challenge. In that respect, we propose an Extended Kalman Filter (EKF) algorithm that fuses planar visual marker and ultrawideband (UWB) localization strategies with the drone's software pose estimation to improve landing accuracy. The visual localization uses the ArUco markers, and the UWB localization is estimated via multilateration with multiple UWB anchors over the landing area. The proposed method explores the techniques described in Rezende et al. (2019), initially developed for high-performance autonomous drone racing, to create a practical and robust real-world system for drone delivery services. Real experiments validate the feasibility of the proposed strategies.

The remainder of the paper is structured as follows. First, the related works are presented in Sect. 2. Section 3 describes the problem of delivery with autonomous drones, and Sect. 4 presents the methodologies used to accomplish this task. The results obtained are discussed in Sect. 5. Finally, conclusions are presented in Sect. 6, together with future research perspectives.

2 Related Works

Given the increasing demand involving autonomous air transport of cargo over short distances, several studies address different strategies for load transportation and other common problems in this type of task. Drone delivery is an emerging field, gaining attention in the academy and industry given the numerous challenges to overcome to perform successful missions in urban environments, such as trajectory planning, localization, guidance and control, obstacle avoid-

ance, and safe landing, especially when global localization is not available or is unreliable (Yoo et al. 2018).The study of parcel delivery using drones is also motivated by the environmental benefit of aerial platforms against standard truck delivery (Koiwanit 2018).

Regarding the control methods for load transportation applications, Villa et al. (2020) present a survey addressing different control techniques and strategies for load transportation using multirotor Unmanned Aerial Vehicles (UAVs) from the literature.Raffo and de Almeida (2016) propose a robust nonlinear control technique for load transportation using quadrotors. Despite proving asymptotic stability, they consider a cable-suspended transport system susceptible to external disturbances due to the wind and drone maneuvers. Similarly, Zúñiga et al. (2018) present cooperative cable-suspended load transportation, using multiples drones with consensus strategies. This approach reduces in 60% the cable oscillations. Also, Rossomando et al. (2020) present a control strategy for cable-suspended transportation using multiple quadrotors cooperation. In this case, the authors combine two strategies to control the formation and maintain load stability. Unlike these approaches, we use a transportation system with the load attached to the drone's body, reducing disturbances due to the cargo movements, and a control method that is known for not having aggressive maneuvers when following the path.

With a focus on the landing process, Gonçalves et al. (2020) present a control method based on vector fields for autonomous landing in a fixed platform. They use visual feedback of a marker on the target to estimate the relative distance and compute a velocity vector field to follow a path to landing. The authors in de Souza et al. (2019) present a different approach for the UAV landing using artificial neural networks (ANNs). There, the model training bases on fuzzy logic to define reference velocities for the robot, which uses visual detection of a marker for distance feedback. Xuan-Mung et al. (2020) present an algorithm for quad-rotors autonomous landing in a moving target, with a robust control strategy to minimize the ground effect and other disturbances. They also propose a state estimation based on visual detection of the landing platform and use this estimation to plan a route to landing. Similarly, de Santana et al. (2019) present a visual-based method for landing in moving targets, tracking the target movements, and predicting a point to land. However, moving targets are not common in delivery tasks, and we have not addressed these special cases.

Obstacle avoidance appears as the main problem investigated in several papers on autonomous drone navigation in the literature. However, in works related to cargo transportation and landing, this issue is not usually addressed. Falanga et al. (2020) present a method for dynamic obstacle avoidance using event cameras for fast reactions of quadrotors. Experiments show the ability to avoidance of multiples

obstacles with speeds up to 10 m/s. This strategy generates control commands according to the detection of the events during flight and can be adopted in the delivery task. Still focusing on drone navigation with obstacle avoidance, the authors in Chiella et al. (2019) present a localization approach and a vector field-based strategy for path following in sparse forests. They use GNSS and LiDAR data for location improvement and detection of trees, which are avoided using a probabilistic planner along with the vector field. Their collision avoidance strategy can also be incorporated into our solution, given that we also use a similar vector field-based control method. Despite the importance of considering obstacles in drone navigation tasks, the problem is not commonly treated in drone delivery, transportation, and landing researches, since the environments are usually known and controlled.

Traditional protocols and drone landing methods rely on expensive equipment such as DGPS or RTK GPS or do not satisfy the precision and robustness needed for drone landings in urban areas. Visual localization methods could aid in locating the landing area accurately, even in partially cluttered scenarios, using equipment already deployed in the platform, for instance, RGB cameras. Planar markers such as the ArUco (Garrido-Jurado et al. 2014) can generate robust pose estimation from heights up to 30 m and are a feasible solution for drone landing (Wubben et al. 2019; Marut et al. 2019; Gonçalves et al. 2020). Many landing solutions using planar markers change the flight behavior to use only ArUco localization when in range instead of combining information from different sources. Despite the low-cost, low-power consumption of visual localization using planar markers, they are more prone to environmental interference such as low light conditions, snow, dust, rain, and fog. Therefore, visual planar markers require constant maintenance to keep them fully functional. A different visual approach for robust landing in vision-compromised environments uses active infrared (IR) beacons located at the landing platform and a special IR camera for detecting them from afar (Nowak et al. 2017). These methods can work without external illumination but could be imprecise at direct sunlight or other IR emission sources.

Other types of localization systems that are less prone to environmental interference are wireless-based localization systems. Robust wireless localization can be achieved with ultrawideband (UWB) systems using the time-of-flight (ToF) principle to estimate distances with centimeter precision. Drones could exploit these localization systems for indoor localization in cluttered environments such as in Tiemann and Wietfeld (2017) and Tiemann et al. (2018). Key benefits of these types of wireless localization technologies are that they could work even in visually degraded situations, are easily scalable to multiple platforms (Nguyen et al. 2016), and are particularly robust to walls and reflections, increasing the possible range of real-world situations where they

can be applied. Recent works have fused UWB and vision localization for drone landing based on recursive least square optimization (Nguyen et al. 2019).

Recent works dealing with drone delivery have focused on route optimization (Chiang et al. 2019), optimal charging station location (Hong et al. 2018), and the mixture of traditional aerial routes with drone-carrying truck routes (Chang and Lee 2018; Boysen et al. 2018). However, a holistic analysis of the requirements of a suitable delivery platform is often overlooked. Our work differs from the previously mentioned ones since we propose a complete platform for drone delivery and compare popular localization methods for UAV platforms such as GPS, visual, and UWB localization in the landing phase. We also propose a sensory fusion of multiple external localization techniques given the sensing capabilities already available on the UAV, including GPS, 9DoF IMU, and barometer. An extended Kalman filter improves landing accuracy considering fixed location platforms in urban areas.

Besides that, most of the control strategies present in the literature focus on ensuring the drone's stability at the lowest level, acting on the motors to follow a trajectory (Almakhles 2020). The present work considers that the drone already has the lowest level controller properly tuned to guarantee the desired angular velocities. Therefore, we adopt the approaches presented in Rezende et al. (2020) and Gonçalves et al. (2010) to define our vector field-based high-level control strategy, commonly called guidance.

Table 1 presents a qualitative comparison of the differences between our approach for autonomous drone delivery and some related works mentioned in this section. Note that most of the related works deal with a specific part of the delivery task, while ours proposes a navigation system for complete autonomous delivery tasks using drones, focusing on the practical aspects of safe drone landing.

3 Autonomous Drone Delivery

This section describes the autonomous delivery problem with drones and specifies the hardware and software used for development.

3.1 Problem Description

The problem addressed involves transporting small parcels between two points of interest using a drone in autonomous mode without receiving commands from a human pilot. Assuming that the landing stations are defined at safe locations and the route is planned for a height above buildings and trees, we have considered obstacle-free environments. Furthermore, we assumed distances compatible with the drone endurance (maximum time of flight). Besides, since it is essential for delivery drones to carry fragile objects without

Table 1 Qualitative comparison of the related works

Work	Delivery Path Plan.	Landing	Load	Image proc.	UWB	Validation	Control	Moving platf.	Obst. Avoidance
Present paper*	✓	✓	✓	✓	✓	✓	✓	×	×
Rossomando et al. (2020)	×	×	✓	×	×	×	✓	×	×
Gonçalves et al. (2020)	×	✓	×	✓	×	✓	✓	×	×
Xuan-Mung et al. (2020)	×	✓	×	✓	×	✓	✓	✓	×
Falanga et al. (2020)	×	×	×	×	×	✓	✓	×	✓
de Souza et al. (2019)	×	✓	×	✓	×	✓	✓	×	×
de Santana et al. (2019)	×	✓	×	✓	×	×	✓	✓	×
Chiella et al. (2019)	×	×	×	×	×	✓	✓	×	✓
Wubben et al. (2019)	×	✓	×	✓	×	✓	×	×	×
Raffo and de Almeida (2016)	×	×	✓	×	×	×	✓	×	×

abrupt movements, we have adopted a grasped transportation mechanism that reduces the risks of vibrations or unexpected package drops.

At first, residential regions will be the primary landing locality for delivery tasks; therefore, the drone must have the ability to land with accuracy in restricted and narrow areas to prevent unexpected accidents or injuries. The tests were performed considering a 1x1m landing platform.

3.2 Hardware

The DJI Matrice 100[1], illustrated in Fig. 1 allows developers to implement their codes to control the drone. All sensors that come with the drone by default are used, including GPS, 9DoF IMU, and a barometer. These are responsible for helping to estimate the drone's position in global coordinates (latitude and longitude), orientation, and height. The drone's flight board is responsible for transmitting information from the sensors to another device, in addition to receiving control commands and sending them to the brushless motors. In addition to the flight controller board already available on the drone, a Jetson Nano[2] is responsible for data processing, path planning, and the quadrotor high-level control.

We also developed a 3D model of the box used for delivery, with dimensions $110 \times 105 \times 90$ mm, as shown in Fig. 2a. Its coupling mechanism uses a servo motor, as illustrated in Fig. 2b. An Arduino Nano is connected to the Jetson board to control this servo motor, placing it in the position of coupling or decoupling the box on the drone. The schematic of Fig. 3 illustrates the connections between the equipment and drone used during experiments. The drone has a total mass of approximately 3000 g with all devices connected and an empty box. Therefore, the cargo mass must be limited to 600 g, following the DJI Matrice 100 takeoff technical specifications. The current system is mostly intended to validate the algorithms that we propose in Sect. 4. Further, it could be scaled up to transport heavier loads.

Considering that the landing site used in the experiments has 1x1 m and that the horizontal accuracy in the drone's location using GPS is approximately 2m, it is necessary to use additional sensors to assist the drone in making a safer and more accurate landing. For that, we verified the application of the following sensors in experiments using the real drone: (i) a RaspberryPi Camera v2.0 pointing downwards, together with ArUco markers combination on the landing platform, and (ii) ultrawideband (UWB) devices anchored at the landing site, and a device of the same type attached to the drone. In both cases, it is possible to obtain additional information on the drone's position with respect to the plat-

---

[1] DJI Matrice 100 - https://www.dji.com/br/matrice100.

[2] Jetson Nano - https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/.
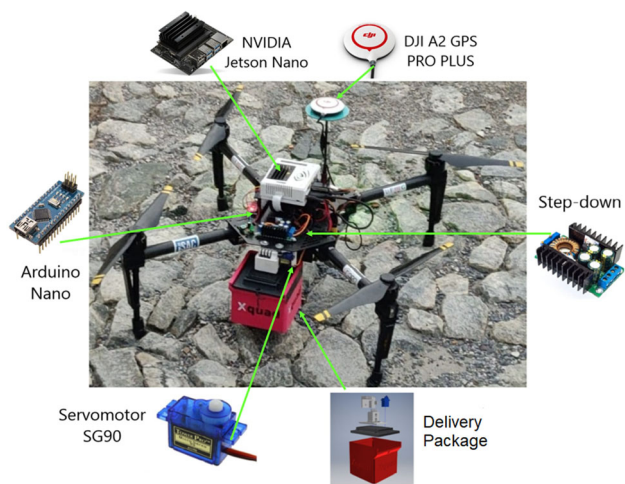
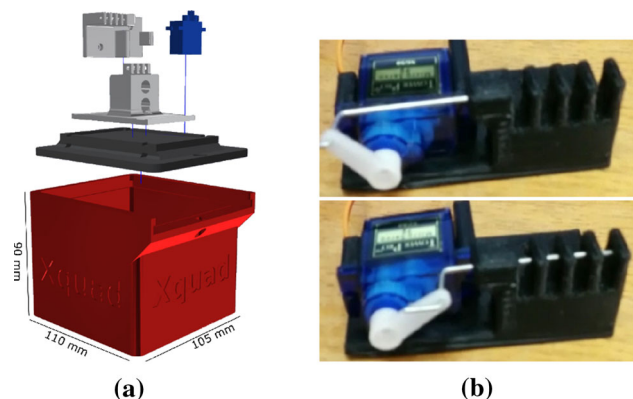Fig. 1 Drone used for the autonomous delivery and its components.

Fig. 2 Coupling mechanism using a servo motor with a package used for cargo delivery: (a) exploded delivery box and (b) the servo coupling mechanism.

Fig. 3 Connection between the drone embedded equipment.
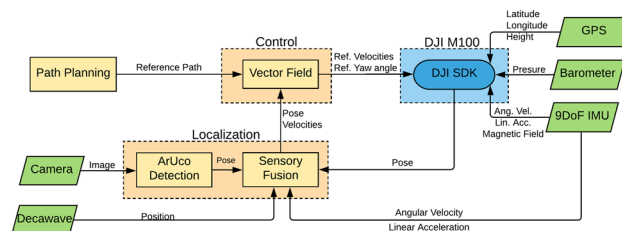
Fig. 4 System overview with the information flow between the modules.

form during landing and improve the localization by fusing all these data.

3.3 Software

The Operating System used is Ubuntu 18.04 together with ROS 1 (Robot Operating System). The ROS package provided by DJI called Onboard-SDK-ROS[3] establishes communication between the drone and the Jetson Nano. The package allows sending commands to the embedded control system and provides data from the sensors present on the drone, such as GPS, 9DoF IMU, and barometer, in addition to estimating the drone's position and orientation in an online fashion. For the identification and estimation of the ArUco's pose, specific algorithms from the OpenCV[4] are used. In the case of UWB devices, an algorithm uses the difference

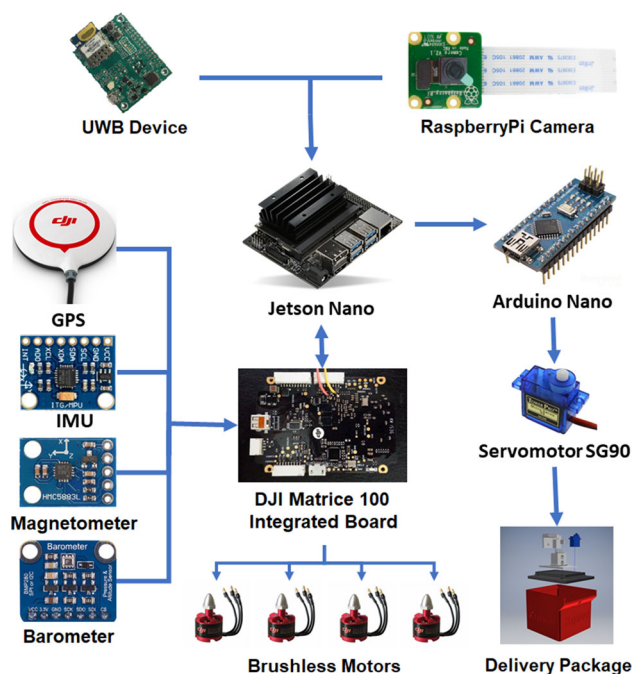3 Onboard-SDK-ROS - https://github.com/dji-sdk/Onboard-SDK-ROS
4 OpenCV - https://opencv.org/

between the time of arrival (TDoA) of the signal for each device to compute the drone's position with respect to the landing site.

4 Methodology

In order to satisfy the problem requirements, such as reducing the delivery time while avoiding abrupt movements and with a safe landing, we propose a solution divided into three distinct tasks: (i) path planning, (ii) localization, and (iii) control. Figure 4 illustrates the information flow during the system operation.

4.1 Path Planning

The proposed path planning strategy simplifies autonomous drone delivery. First, the method considers the altitude, latitude, and longitude data to define the drone's geographical

location. For test purposes, the covered distances are short, such that a flat Earth model can be considered. Thus, we transform the angles of latitude and longitude into measurements of distance. The drone's position is then initially represented with respect to the Earth's reference frame $\mathcal{F}_E$.

Consider that the drone's starting point is $\mathbf{p}_s \in \mathbb{R}^3$ and that the load delivery point is $\mathbf{p}_f \in \mathbb{R}^3$. Without loss of generality, it is possible to assume an inertial coordinate system $\mathcal{F}_I$ that respects two conditions:

1. The path's end point is the origin, i.e. $\mathbf{p}_f = \mathbf{0}$;
2. The **x** axis of the coordinate system $\mathcal{F}_I$ is in the horizontal plane, pointing in the direction of the final point, i.e. $\hat{x} \parallel \Pi_{xy}(\mathbf{p}_f - \mathbf{p}_s)$, where $\Pi_{xy}(\cdot)$ represents the projection in the **xy** plane.

The inertial reference frame $\mathcal{F}_I$ is easily obtained through two operations: a translation with respect to $\mathcal{F}_E$, in order to satisfy condition 1; and a simple rotation in the **z** axis to satisfy condition 2.

The proposed path planning method computes a smooth reference path connecting the two points $\mathbf{p}_s$ and $\mathbf{p}_f$. The strategy consists of creating 5 path sections: (i) vertical ascending line; (ii) arc of a circle; (iii) horizontal line towards the platform; (iv) arc of a circle; (v) vertical descending line. In order to allow a smooth transition between sections, space is divided into 5 sectors $\mathcal{S}_i$, $i = 1, 2, 3, 4, 5$. Each path section is associated with one sector. The definition of the sectors is presented below:

$$\mathcal{S}_1 = \left\{ (x, y, z) \in \mathbb{R}^3 : z \leq h - r,\ x \leq -d/2 \right\},$$
$$\mathcal{S}_2 = \left\{ (x, y, z) \in \mathbb{R}^3 : z > h - r,\ x < -d + r \right\},$$
$$\mathcal{S}_3 = \left\{ (x, y, z) \in \mathbb{R}^3 : z > h - r,\ -d + r \leq x \leq -r \right\},$$
$$\mathcal{S}_4 = \left\{ (x, y, z) \in \mathbb{R}^3 : z > h - r,\ x > -r \right\},$$
$$\mathcal{S}_5 = \left\{ (x, y, z) \in \mathbb{R}^3 : z \leq h - r,\ x > -d/2 \right\}. \tag{1}$$

where h is the drone flight height (with respect to $\mathcal{F}_I$), r is the radius of the transition arcs, and d is the horizontal separation between $\mathbf{p}_s$ and $\mathbf{p}_f$. Starting and final points, sectors, and variables defined here are illustrated in Fig. 5.

4.2 Localization

The Onboard-SDK-ROS package provides a georeferenced estimate of the drone's global orientation and position (DJI-SDK Pose). This information comes from the sensory fusion of the available GPS, 9DoF IMU, and barometer, which results in an accuracy of approximately 2m. Such an estimate is good enough when on a cruise flight and was therefore
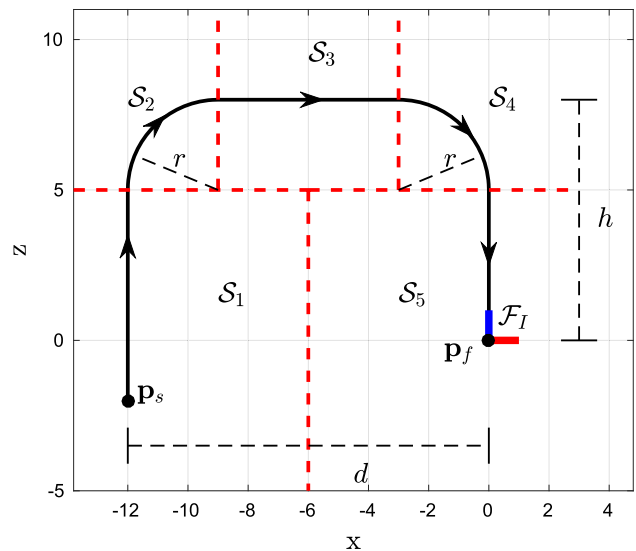
Fig. 5 Sectors $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, $\mathcal{S}_4$ e $\mathcal{S}_5$, defined in Eq. (1)

used throughout the flight. However, this may be insufficient when landing on a 1x1 m platform like the one proposed in this work, whose dimensions are less than the position estimate's accuracy. Besides, if the landing pad position changes or the georeference is not precise enough, the drone might not be able to land at the right location using GPS localization alone.

For these reasons, it is necessary to obtain additional information that allows the improvement of the drone's estimated position with respect to the landing site. This paper presents a sensor fusion strategy to improve localization by merging the DJI-SDK pose estimation with information from: (i) an ArUco marker detection technique and (ii) multilateration using ultrawideband (UWB) communication devices.

4.2.1 ArUco

For the use of the marker detection technique for localization, ArUco markers were printed and placed on top of the landing platform. A camera attached to the drone, pointing downwards, provides images of the marker during landing. ArUco markers have features that facilitate their identification in the image, such as well-defined borders and high color contrast. In addition, the markers do not present ambiguities in their orientation.

Thus, specific OpenCV algorithms identify the ArUco, as well as estimate the relative pose of the marker with respect to the camera. This last step is done by solving the problem of *PnP (Perspective-n-Point)*, which proposes to estimate the three-dimensional pose of a calibrated camera given a set of 3D points and their corresponding 2D projections on the camera plane.

Fig. 6 Modified ArUco marker. An smaller marker is placed in a bigger one.

It is possible to find the pose that minimizes the projection errors of the points on the camera plane by knowing the actual size of the marker and the intrinsic calibration parameters of the camera. These points must be distinguishable from each other, and in this case, the corners of the ArUco and its orientation allow for differentiating each of its four corners before sending to a PnP solver algorithm (Lepetit et al. 2009; Hesch and Roumeliotis 2011).

There are works in the literature that address strategies to improve the detection of the markers. A common method is to merge different AR markers to create a group that provides a better pose estimation, minimizing noise and occlusion, as presented in de Santana et al. (2019). Large ArUco markers can be detected from high altitudes. However, when the drone is approaching the platform, this ArUco is quickly lost by the camera. On the other hand, a smaller ArUco has the advantage of being detectable when the drone is close to the platform (if there is no high horizontal error), even though it is difficult to be detected at high altitudes. To improve the marker detection range at high and low altitudes, we considered a modified ArUco marker that has a smaller marker (0.09×0.09m) inside a larger one (0.8×0.8 m). Figure 6 shows this modified ArUco. The inclusion of the smaller marker may harm the detection of the larger one. Nonetheless, in our tests, this problem did not occur.

4.2.2 Ultrawideband (UWB) Devices

Although the ArUco marker detection provides a good pose estimation, the method is not robust for detection in low-light environments or under visual occlusion situations. For this reason, we consider using another localization based on ultrawideband devices, which works on these conditions and increases the landing strategy robustness.

Devices based on ultrawideband wireless technology are commonly used for low-energy IoT communication or localization. This technology uses radio waves with a band-

width greater than 500 MHz, which reduces the loss due to obstructions and reflections of the environment, consequently increasing the security of transmissions (Sahinoglu 2008).

UWB-based localization systems can be used indoors and outdoors, with an accuracy of up to $20cm$, according to some manufacturers. In this method, multilateration algorithms estimate the position x_T, y_T and z_T of a mobile device (called tag) with respect to a fixed reference, where other devices (called anchors) are located.

In this paper, we use Decawave DWM1001[5] UWB devices to estimate the position of the drone with greater precision when approaching the landing platform. A minimum of five devices are required for the algorithm to work, one tag embedded in the drone and four anchors in known positions; one of these devices is set as the base anchor.

The position is calculated based on the distance of the tag with respect to the anchors, which comes from the Time Difference of Arrival (TDoA) of the transmitted signal, multiplied by the speed of signal propagation (speed of light), as presented in Sayed et al. (2005). Consider a set of enumerated UWB devices, where index 0 represents the tag placed on the robot, index 1 the base anchor (or main anchor), and the higher indexes the other anchors used on the system. The distance d_{i1} from the ith anchor to the base anchor is given by:

$$d_{i1} = (t_i - t_1)c, \quad i = 2, ..., N, \tag{2}$$

where t_i is the instant of time the signal sent by the tag reaches anchor i, and t_1 is the instant of time this signal reaches the base anchor. Light speed is c and the number of anchors is N, such that $N \geq 4$. Distances in equation (2) result on an intersection region that represents the Tag's position, obtained as the solution of the following set of equations:

$$
\begin{aligned}
(d_{21}^2 + d_1^2)^2 &= (x_2 + y_2 + z_2)^2 - 2J_2 + d_1^2, \\
(d_{31}^2 + d_1^2)^2 &= (x_3 + y_3 + z_3)^2 - 2J_3 + d_1^2, \\
&\vdots \\
(d_{N1}^2 + d_1^2)^2 &= (x_N + y_N + z_N)^2 - 2J_N + d_1^2,
\end{aligned}
\tag{3}
$$

where $J_i = (x_i x_T + y_i y_T + z_i z_T)$ and d_1 is the distance from the tag to the base anchor. Considering t_0 the instant of time the signal is sent by the tag, d_1 can be computed as follows:

$$d_1 = (t_1 - t_0)c. \tag{4}$$

To use Decawave devices, a maximum distance of 10m must be kept between the anchors and the tag. This method-

ology does not estimate orientation, whereas the ArUco estimation does.

4.2.3 Sensory Fusion

One way to improve the localization and more accurately estimate the drone's position, orientation, and speed states, is to use information from several sensors. Sensory fusion methods use data from different devices to obtain more accurate information on the states of interest. For instance, the software on the DJI Matrice 100 uses data from the GPS, 9DoF IMU, and barometer to provide the drone's pose and speeds (DJI-SDK pose).

One of the most common fusion methods is the Extended Kalman filter (EKF) (Thrun et al. 2000). In this sense, the PnP estimation of the ArUco's marker location and the UWB localization are merged with the DJI-SDK pose data to allow a more accurate landing. As the sensors provide data referring to coincident states, a bias is considered in the position estimated by GPS. It is possible to divide Kalman's extended fusion and filtering method into two stages: prediction and correction. For simplicity, the equations are presented with the notation $b \leftarrow a$ indicating that b is updated with the value of a.

The prediction step in the discrete EKF involves the state vector $\bar{\mathbf{x}}$ and the covariance matrix P:

$$\bar{\mathbf{x}} \leftarrow f(\bar{\mathbf{x}}, \mathbf{u}, \Delta t), \tag{5}$$

$$P \leftarrow FPF^T + GQ_uG^T + Q_f, \tag{6}$$

where f represents the state propagation model, which involves current estimate $\bar{\mathbf{x}}$, the input vector $\mathbf{u}$ and the timestep Δt. Matrix $F \equiv F(\bar{\mathbf{x}}, \mathbf{u}, \Delta t)$ is the partial derivative of f with respect to $\bar{\mathbf{x}}$, and matrix G is the partial derivative of f with respect to $\mathbf{u}$. Matrix Q_u is the covariance matrix associated with input vector $\mathbf{u}$, and Q_f is a covariance matrix associated with the model.

The correction step is defined as follows:

$$\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}} + K \left(\mathbf{w} - h(\bar{\mathbf{x}})\right), \tag{7}$$

$$P \leftarrow (I - KH) P, \tag{8}$$

where $\mathbf{w}$ is the measurement vector and $h(\bar{\mathbf{x}})$ is the measurement model, which represents the expected value of $\mathbf{w}$ given the current estimated state $\bar{\mathbf{x}}$. Matrix H is the Jacobian of $h(\bar{\mathbf{x}})$, and I is an identity matrix. Finally K represents the Kalman gain, given by:

$$K = PH^T \left(HPH^T + R\right)^{-1}, \tag{9}$$

where R is the covariance of the measurement data $\mathbf{w}$.

The strategy defines the input vector as $\mathbf{u} = [\mathbf{u}_v^T \ \mathbf{u}_\omega^T]^T = [v_x \ v_y \ v_z \ \omega_x \ \omega_y \ \omega_z]^T$, where v_x, v_y, and v_z are the linear velocities of the drone with respect to the world frame, and ω_x, ω_y, and ω_z are the angular velocities in the body frame. In the correction steps, the measurement $\mathbf{w}$ may assume three different values: (i) position and Euler angles from the DJI-SDK pose; (ii) position and Euler angles from the ArUco-PnP; and (iii) position from the UWB system.

In general, the data collected by DJI-SDK position (GPS), compared to the ArUco and UWB, have an unknown nonzero average error. Therefore, the direct use of these measurements in the correction step causes an oscillation in the estimation. Despite containing a displacement, the data from DJI-SDK is the only available throughout the whole experiment and cannot be discarded. In order to merge the data from these two sensors properly, we consider extra states representing a bias associated with the DJI-SDK position (GPS) measurements with respect to the landing platform's location (ArUco or Decawave). In fact, the proposed filter incorporates the following 12 states:

$$\bar{\mathbf{x}} = [\ \underbrace{\bar{x} \ \bar{y} \ \bar{z}}_{position} \ \underbrace{\bar{\phi} \ \bar{\theta} \ \bar{\psi}}_{angles} \ \underbrace{\bar{b}_x \ \bar{b}_y \ \bar{b}_z}_{biasGPS} \ \underbrace{\bar{b}_{\omega x} \ \bar{b}_{\omega y} \ \bar{b}_{\omega z}}_{bias \ gyro}]^T, \tag{10}$$

where $\bar{\mathbf{p}} \equiv [\bar{x} \ \bar{y} \ \bar{z}]^T$ is the drone's position, $\bar{\mathbf{r}} \equiv [\bar{\phi} \ \bar{\theta} \ \bar{\psi}]^T$ the drone's orientation in Euler angles, $\bar{\mathbf{b}}_p \equiv [\bar{b}_x \ \bar{b}_y \ \bar{b}_z]^T$ is the GPS bias, and $\bar{\mathbf{b}}_\omega \equiv [\bar{b}_{\omega x} \ \bar{b}_{\omega y} \ \bar{b}_{\omega z}]^T$ is the drone's gyro bias. More precisely, $\bar{\mathbf{b}}_p$ is the GPS bias with respect to the (predefined) location of the landing platform. Note that the filter does not have bias states related to the ArUco orientation. For this reason, the marker geographical orientation must be known with relative precision.

The propagation model $f(\bar{\mathbf{x}}, \mathbf{u}, \Delta t)$ is given by:

$$f(\bar{\mathbf{x}}, \mathbf{u}, \Delta t) = \begin{bmatrix} \bar{\mathbf{p}} \\ \bar{\mathbf{r}} \\ \bar{\mathbf{b}}_p \\ \bar{\mathbf{b}}_\omega \end{bmatrix} + \begin{bmatrix} \mathbf{u}_v \\ J_r(\mathbf{u}_\omega - \bar{\mathbf{b}}_\omega) \\ 0 \\ 0 \end{bmatrix} \Delta t, \tag{11}$$

in which $J_r \equiv J_r(\bar{\phi}, \bar{\theta})$ is the Jacobian matrix that transforms the angular velocities $\mathbf{u}_\omega$ in the derivatives of the Euler angles $\bar{\mathbf{r}}$. Note that the prediction model assumes constant bias states.

The filter considers three measurement models. The first one represents the expected measurement of position and orientation data provided by the DJI SDK. In this correction step, there is a binary variable that determines whether the landing platform local data is already being collected or not. This variable ξ determines the inclusion of bias in the GPS measurement model. This variable is 1 when the local data

has already been observed and 0 if not. Thus, we have:

$$h_{SDK}(\bar{\mathbf{x}}) = \begin{bmatrix} \bar{\mathbf{p}} + \xi \bar{\mathbf{b}}_p \\ \bar{\mathbf{r}} \end{bmatrix}. \tag{12}$$

Let $H_c^d \in SE(3)$ be a constant homogeneous matrix that represents the pose of the camera with respect to the drone and $H_a^w \in SE(3)$ be a constant homogeneous matrix that represents the pose of the ArUco in the world. The matrix $\bar{H}_d^w \equiv \bar{H}_d^w(\bar{\mathbf{p}}, \bar{\mathbf{r}})$ represents the pose of the drone in the world frame. Then, the expected pose of the ArUco with respect to the camera (data provided by the PnP algorithm) can be written as:

$$\bar{H}_a^c = (H_c^d)^{-1}(\bar{H}_d^w)^{-1}H_a^w. \tag{13}$$

The matrices H_c^d and H_a^w are known *a priori*, whereas the matrix $\bar{H}_d^w$ is obtained from the filter states $\bar{\mathbf{p}}$ and $\bar{\mathbf{r}}$. Since the filter works with Euler angles, the measurement model of the ArUco information is given by:

$$h_{ArUco}(\bar{\mathbf{x}}) = \begin{bmatrix} get\_pos(\bar{H}_a^c) \\ get\_euler(\bar{H}_a^c) \end{bmatrix}, \tag{14}$$

in which *get_pos()* and *get_euler()* are functions that return the position and the Euler angles associated with a given homogeneous transformation matrix.

It is important to comment about the reason we feed the filter with the original pose of the ArUco (direct from PnP, with no transformation) with respect to the camera. A different strategy would consider a sequence of homogeneous transformations in order to obtain the pose of the drone with respect to the world and feed the filter with this transformed data. In this case, model $h_{ArUco}(\bar{\mathbf{x}})$ would correspond to the identity function. The justification for the choice of our strategy relies on the noise levels of the ArUco measurement. In fact, the position and the yaw angle of the marker with respect to the camera are estimated with good precision, while the roll and pitch have significantly higher uncertainty. In our approach, we are able to provide these levels of accuracy in a diagonal covariance matrix R. Using a sequence of homogeneous transformations, the high noise in the roll and pitch reflects on the computed position of the drone, and consequently, would worse the filter response. The consideration of the original pose measurement allows the filter to properly treat the signals according to their correct covariances.

Finally, since the UWB system only provides a position measurement, its measurement model is given by:

$$h_{UWB}(\bar{\mathbf{x}}) = \bar{\mathbf{p}}. \tag{15}$$

In this way, the estimation of the drone's pose improves, increasing safety and accuracy during landing. The results in Sect. 5 demonstrate the effectiveness of the methodology.
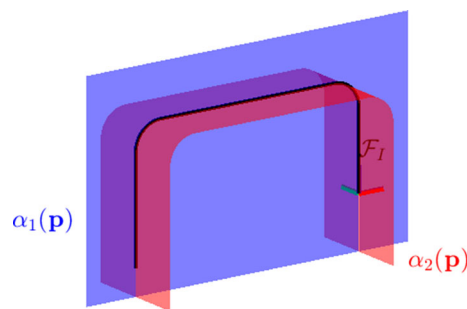

Fig. 7 Graphical representation of α_1 and α_2 functions, equations (16) and (17), respectively.

4.3 Control with Vector Fields

The quadrotor control is divided into two levels: high and low. The main objective is to enforce the drone to follow the path presented in Sect. 4.1. In order to accomplish this task, the proposed high-level controller is an artificial vector field created to allow path following assuming the drone behaves as a simple integrator. The low-level controller can be any controller able to impose this desired vector field-based velocity behavior to the real system. For instance, in the experiments presented in this work, the drone uses the low-level controller proposed in Rezende et al. (2020). That controller has the drone's mass as a parameter. Here we consider a mass $m = m_{vehicle} + m_{package}$, accounting for the drone and the package. After the package is released, we consider $m = m_{vehicle}$. Next, we describe how the desired vector field, used by the controller, can be defined by means of the strategy proposed in Gonçalves et al. (2010).

Traditional control techniques, or even flight modes that use waypoints on the map could have been used, however with disadvantages. In the proposed technique, the generated path is smooth, eliminating unwanted effects caused by switching between control laws at each waypoint. In addition, paths such as the ones in sectors S_2 and S_4 (Fig. 5) can be optimized to generate less abrupt movements with the load.

The control by vector fields is used to follow paths, not trajectories. The proposed control law is a function only of the drone's state $\mathbf{p}$, therefore, it does not directly depend on time t. This property is particularly interesting because it does not present two problems associated to trajectory control: (i) if the reference starts very far from the drone's initial position, it may pass through an aggressive transient, which is unwanted; (ii) in the event of a temporary system failure that causes the drone to stop responding for a while, when the drone returns, the reference may be too far away, generating an additional transient state.

To represent a curve $\mathcal{C}$, it is necessary to define scalar functions $\alpha_i : \mathbb{R}^3 \to \mathbb{R}$, $i = 1, 2$, for each sector $\mathcal{S}_j$, $j = 1, 2, 3, 4, 5$, in a way that the intersection of their zero-level

surfaces, $\alpha_i(\mathbf{p}) = 0$, generates the desired curve (Gonçalves et al. 2010). This means that $\mathcal{C}$ is defined by $\mathcal{C} = \{\mathbf{p} \in \mathbb{R}^3 : \alpha_1(\mathbf{p})=0 \wedge \alpha_2(\mathbf{p})=0\}$. Figure 7 illustrates this representation in the case of the path defined in Sect. 4.1. The functions $\alpha_1 \equiv \alpha_1(y)$ and $\alpha_2 \equiv \alpha_1(x, z)$ are defined as:

$$\alpha_1 = y, \qquad (16)$$

$$\alpha_2 = \begin{cases} -x - d, & if\ \mathbf{p} \in \mathcal{S}_1 \\ \sqrt{(x+d-r)^2 + (z-h+r)^2} - r, & if\ \mathbf{p} \in \mathcal{S}_2 \\ z - h, & if\ \mathbf{p} \in \mathcal{S}_3 \\ \sqrt{(x+r)^2 + (z-h+r)^2} - r, & if\ \mathbf{p} \in \mathcal{S}_4 \\ x, & if\ \mathbf{p} \in \mathcal{S}_5 \end{cases} \qquad (17)$$

This way, for each point in space, a convergent and a tangential component to the curve are given by:

$$F_{conv} = \frac{\nabla V}{\|\nabla V\|}, \qquad F_{tang} = s\frac{\nabla\alpha_1 \times \nabla\alpha_2}{\|\nabla\alpha_1 \times \nabla\alpha_2\|}, \qquad (18)$$

where $V = \frac{1}{2}\alpha_1^2 + \frac{1}{2}\alpha_2^2$ is a Lyapunov function and $\times$ denotes the cross product. We consider $s = 1$ when the drone is going to delivery the package (moving from $\mathbf{p}_s$ to $\mathbf{p}_f$) and $s = -1$ when it is returning (from $\mathbf{p}_f$ to $\mathbf{p}_s$). The parameter s is responsible for inverting the sense of motion by changing the direction of the tangent component F_{tang}.

As seen in Gonçalves et al. (2010), the functions $G \equiv G(V) = -(2/\pi)\arctan(k_f\sqrt{V})$ and $H \equiv H(V) = \sqrt{1 - G^2}$ are defined, where $k_f > 0$ is a convergence weight. This functions are part of the vector field $F(\mathbf{p})$ definition, used in the control strategy, as seen next:

$$F(\mathbf{p}) = v_r(GF_{conv} + HF_{tang}), \qquad (19)$$

where $v_r \equiv v_r(\mathbf{p}) > 0$ is the desired robot velocity, defined by $v_r(\mathbf{p}) = v_i$ for $\mathbf{p} \in \mathcal{S}_i$, $i = 1, 2, 3, 4, 5$, i.e. the reference velocity for the drone is dependent on the sector it is.

The drone's orientation can be controlled in a way to keep its ψ angle (around $\mathbf{z}$ axis) in $0°$ with respect to reference frame $\mathcal{F}_I$. Thus, a reference $\psi_r = 0$ is passed to the lower level controller.

The delivery task in autonomous mode can be accomplished by following the logic demonstrated in Algorithm 1.

5 Results

In this section, real experimental results are presented, considering a parcel delivery task with a drone in autonomous operation. The carried out experiments aim to evaluate the

---

Algorithm 1: Quadrotor high-level control

$delivered \leftarrow false$
$landingDone \leftarrow false$
$taskFinished \leftarrow false$
while $landingDone == false$ **do**
  $states \leftarrow getEKFstates()$
  $sector \leftarrow getSector(position)$
  $cmd \leftarrow vectorFieldController(states, sector)$
  $sendCommands(cmd)$
  if $position == landingPosition$ **then**
    $landingDone \leftarrow true$
  end
end
$releasePackage()$
$delivered \leftarrow true$
$updateMass()$
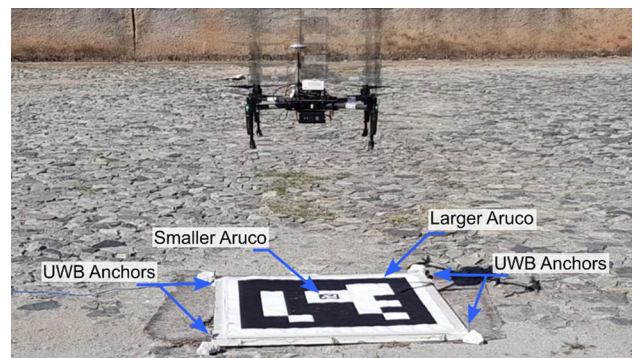$takeOff()$
$taskFinished \leftarrow true$

Fig. 8 Experimental landing setup with planar Aruco markers and UWB tags.

complete autonomous delivery method, as well as the localization strategy in the landing phase. We have considered the platform center as the landing reference point $[x, y] = [0, 0]$, and the distance for the drone's landing point was measured manually using a measuring tape, basing on the drone's center. The considered experimental setup is depicted in Fig. 8 and videos of the experiments are available online[6].

5.1 Complete Delivery Task Evaluation

The proposed strategy was validated in a delivery task with the drone following a planned path, from a start point to the endpoint on the landing platform, autonomously. The object transported has a mass of 200 g, respecting the 600 g limit of the drone. In these experiments, we consider that the control system knows the load mass. In future works, we intend to study the effects of changes in the mass.

Figure 9 shows a comparison between the path performed by the drone and the planned one, using the proposed EKF algorithm with data from the DJI-SDK, and including ArUco

---

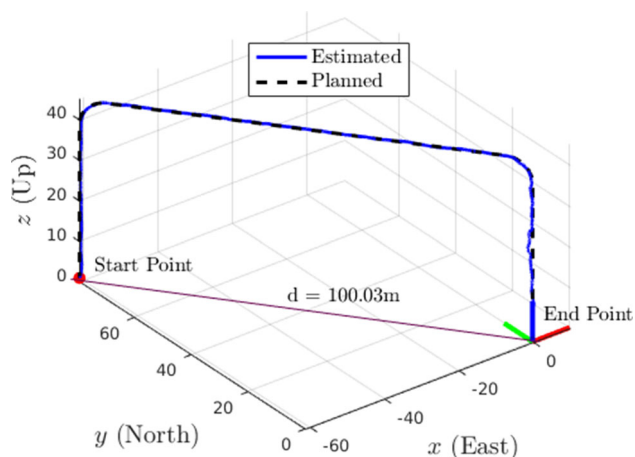[6] Experiment Video - https://youtu.be/wJYaKBytC2g

Fig. 9 Autonomous delivery task results. The performed path is the solid green line and the planned is dashed black.
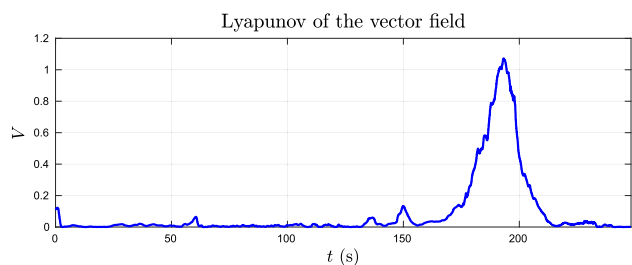

Fig. 10 Lyapunov function of the vector field strategy. It indicates the distance from the drone to the curve.

Fig. 11 Results of 3D position estimation in landing.
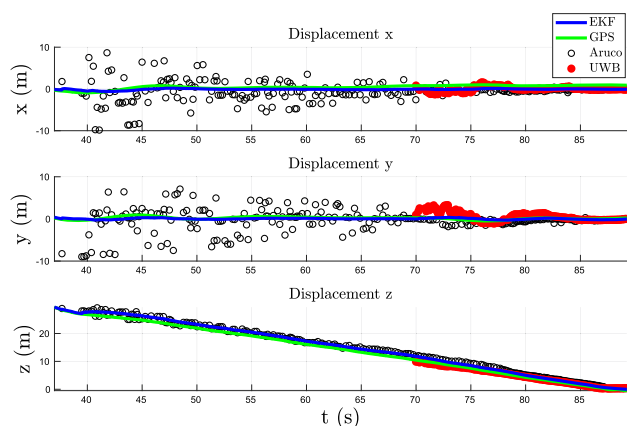
Fig. 12 Results of position estimation in landing for each axis.

markers and UWB on landing. The experiment considers a height $h = 45$m and arcs of the circumference with radius $r = 6$m. The parameter d is computed as the distance of drone at its initial position and the expected position of the platform, and in the experiment, $d = 100.03$m. Figure 10 illustrates the Lyapunov function of the vector field control, which indicates the distance between the drone and the planned path according to the EKF estimation. This distance increases after 150s because of the corrections that start when the platform is detected. These changes show to the system that it is not exactly on the path as expected before. In sequence, this distance decreases again with the actions of the controller. As shown in Fig. 9, these corrections happened in sector $\mathcal{S}_5$ at the height of approximately 30m. Although the camera detected the marker from 45m of height, given the high uncertainties in higher altitudes, the filter only considers measurements with height below 30m.

Figure 11 illustrates the results of the drone's position estimation in another experiment. It shows the data obtained from each sensor separately and from the proposed EKF fusion algorithm during the landing phase. It was a local experiment with $r = 2$m and $h = 30$m focused on the landing phase. In order to plot the position of the ArUco, we considered a direct computation of the position of drone
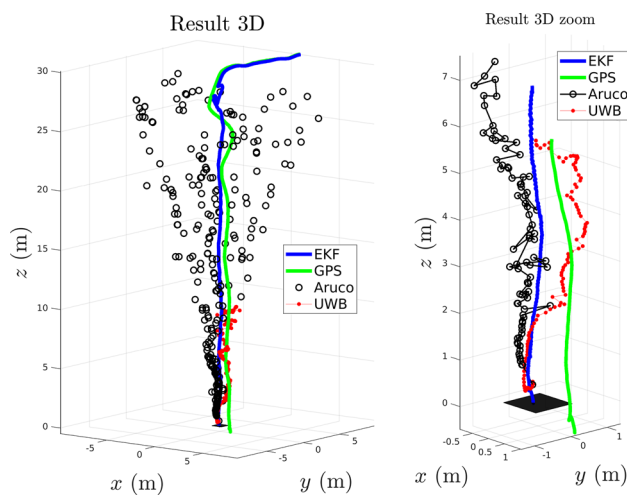
with respect to the frame $\mathcal{F}_I$ (see Fig. 5). It is important to emphasize that this transformed data was not supplied to the filter (see Sect. 4.2.1). There is considerable noise associated with the ArUco and UWB estimates. These noises are directly proportional to the distance between the drone and the platform, as we can observe in Fig. 12. Besides that, the UWB sensors' information starts to be computed only around 10m of distance from the landing base. The system was able to make the drone land almost in the center of the platform, position $[-0.072 \ -0.103]^T$ m in the filter's estimation and $[-0.03 \ -0.11]^T$ m in the ground truth measure. Thus, the distance error from the filter estimation to the ground truth is 0.043m, and the ground truth distance to the center of the platform is 0.114m.

Note in Fig. 11 that, despite the noisy data from the ArUco and the UWB, the filter was able to estimate a clear trajectory for the drone. Also, the GPS (from the DJI-SDK) signal has a shift. According to this measurement, the drone did not land correctly on the platform. The filter bias is responsible to correct the GPS shift while neglecting the noise in the
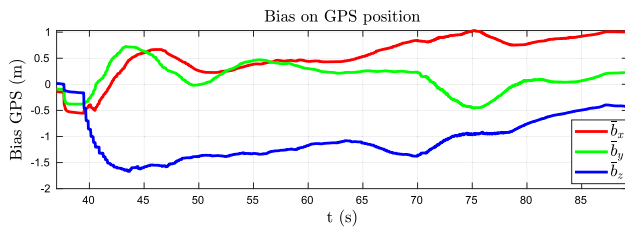
Fig. 13 Bias estimation for GPS position
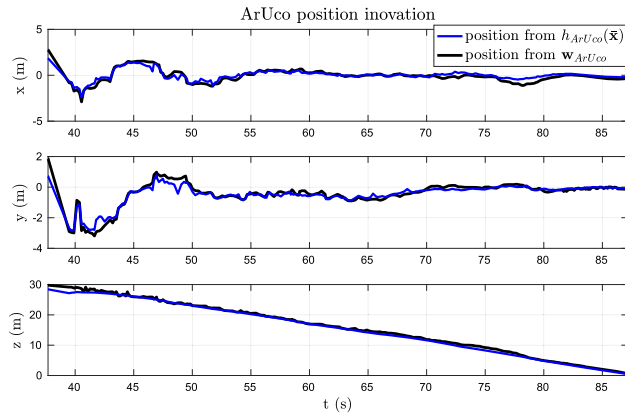
Fig. 14 Comparison of the position measurement of the ArUco with the position of the measurement model $h_{ArUco}(\bar{\mathbf{x}})$. The original data has significantly less noise.

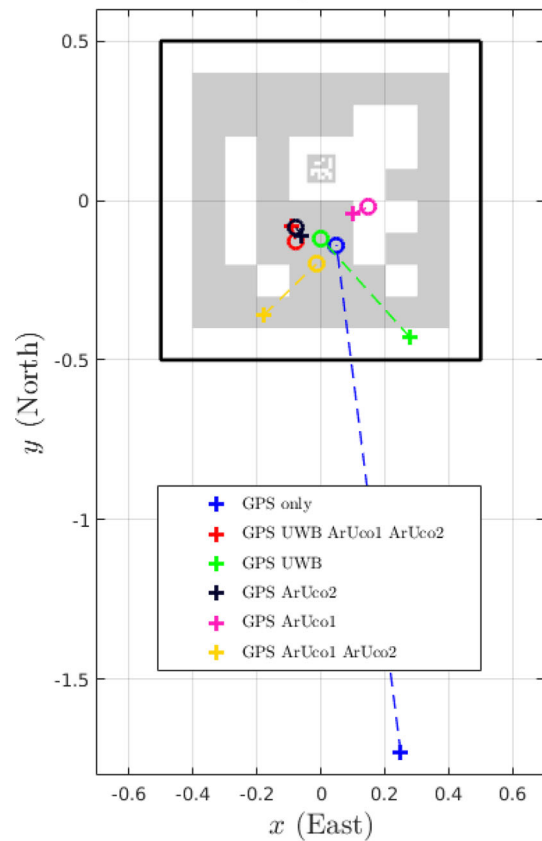**Fig. 15** Results of 6 landing experiments that used different combination of the localization strategies.

ArUco and UWB measurements. The bias estimated by the filter is depicted in Fig. 13.

As commented in Sect. 4.2.3, the noise of the ArUco measurement (ArUco frame with respect to the camera frame) is significantly larger in the estimated roll and pitch angles. When homogeneous transformations are applied to this data to estimate the drone's position, the high noise in the roll and pitch manifest on the x and y position of the drone, as observed in Figs. 11 and 12. Figure 14 presents a comparison of the original position estimation of the ArUco with respect to the camera ($\mathbf{w}_{ArUco}$) with the expected measurement ($h_{ArUco}(\bar{\mathbf{x}})$). This signal has much less noise than the signal observed in Fig. 12.

5.2 Failures Evaluation

The EKF was tested with different combinations of localization methods. The objective is to show that the approach is robust, in the case that if one of the methods fails, the drone still lands on the platform. This experiment simulates real situations including visual occlusion of the ArUco marker, in addition to failures when using UWB devices. Figure 15 shows the results of the experiments. The crosses represent the ground truth positions, while the circles represent the estimation of the EKF. Errors in the ground truth positions are due to the imprecision from both the localization and the

controller. As we can see, the only method not capable of landing the drone on the platform was the GPS alone. The other strategies consisted of combinations using: UWB, large marker (ArUco1) and small marker (ArUco2).

Still evaluating robustness, we have also conducted another experiment inserting noise on the landing platform position from the defined landing point. Using only the DJI-SDK data as input for the EKF, the drone does not reach the platform, and the platform displacement increases the distance error observed in the previous experiment. Considering the inclusion of ArUco detection and the UWB estimation, the drone lands on the platform. In this case, the EKF algorithm considers the displacement on the platform as an error on the GPS data and includes it in the bias estimation.

5.3 Precision Evaluation

This section presents a comparison between the landing accuracy of the proposed system. The EKF merging information from all localization systems was considered in 6 landing experiments. In another 6 landings, the EKF does not use the

Filter X GPS only comparison
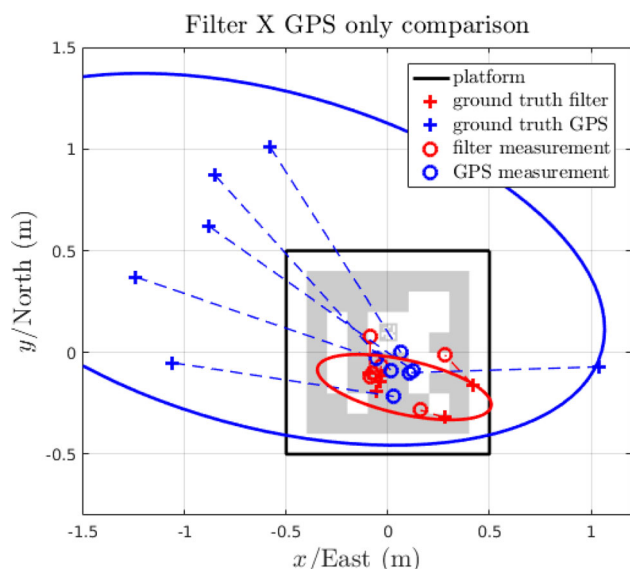
Fig. 16 Comparison between the system's precision when the ArUco and UWB are considered or not.

Table 2 Error between estimated location and ground truth at landing.

Considered sensors	Mean error (m)	Stand. Deviation (m)
All (filter)	0.121	0.0641
DJI	1.166	0.130

ArUco and the UWB data, thus, relying only on the GPS for position estimation.

Figure 16 shows the result of these 12 experiments. The true landing positions correspond to the crosses, while the circles correspond to the position estimation of the filter after the landing. Red represents the data obtained when the filter considers all information, and blue when it uses only the GPS data. The two ellipses correspond to the covariance of the ground truth landing results and assume 2 standard deviations.

Note that in none of the experiments trusting only on the GPS, the drone landed on the platform (blue crosses), despite the filter estimated that the drone was close to the platform's center (blue circles). This error was not corrected since that no local information (ArUco or UWB) was available. In these experiments, the distance from the platform's center reached 1.29m. When the filter counted with data from the ArUco and the UWB, the drone landed on the platform all 6 experiments and the result has a mean distance of 0.19m from the platform's center.

Table 2 presents, in the first column, the mean of the norm of the error (distance) between the ground truth and the estimated pose. We consider a comparison between our strategy with all considered sensors and with only the DJI sensors. On the second column, we show the associated standard deviation of the values.

6 Conclusion and Future Work

This paper presented a navigation strategy with techniques of path planning, localization, and control for autonomous delivery tasks using drones. Computer Vision algorithms and UWB devices provide pose estimates for the quadrotor to an extended Kalman filter, allowing the sensory fusion with the drone's DJI-SDK pose. A vector field-based controller defines commands for the drone to follow a planned path connecting a start point to the landing platform.

Experimental results validate the proposed method in a complete autonomous delivery task. In the drone landing experiments, it is possible to note the advantage of the proposed localization method over the DJI-SDK pose estimation alone. A robustness analysis shows that the system works in case of failure in one of the localization methods, which can occur in low-light situations, camera vision occlusion, and power supply failure for the UWB devices. Besides, a precision evaluation compares the accuracy of the landing phase using only the GPS for position estimation and using the proposed EKF with all localization methods, showing the advantages of the adopted strategy. Most of the delivery operations occur in urban and dense populated areas, and all these results show the effectiveness of the adopted method and increase confidence for a safe landing.

Despite the advantages presented, there are still some limitations to overcome. The proposed strategy works only with GPS available and in obstacle-free environments. Besides, the load mass must be known and invariant until landing. These points will be addressed in future works.

During the mechanical design of the cargo holding system, it is important to avoid placing the box so that it blocks the propellers' winds. This has a considerable negative effect in the generated total thrust and also creates an undesirable asymmetry. It is also fundamental to adjust the parameters of the controller in Rezende et al. (2020) so that the drone has a smooth flight. Aggressive turns would reflect in shaking on the camera image and harm the localization based on the ArUco. Moreover, it may not be appropriate to shake the cargo. Another important detail that has a significant effect on the systems performance is the camera's aperture. It should be wide enough to identify the ArUco even when GPS errors are high and the drone enters stage S_5 far from the platform. A wider aperture also enables the marker to be identified from a closer distance, improving the landing precision.

Future works include improving safety by using the images to detect obstructions on landing caused by people or animals, for example. Another goal is the increase of the localization precision, for instance, by exploring methods for GPS-denied moments during the transportation phase and including other sensors for the landing. Other option is to improve the EKF considering delay in the pose estimated by the ArUco method and measurement covariance matri-

ces dependent on the vehicle's height. Besides, a complete 360-degrees obstacle avoidance system could improve the flight's overall security. We also intend to reach improvements in path planning, considering obstacle avoidance and minimum power consumption. Other future work is to implement fault detection methods using proprioceptive sensors to activate parachutes or other types of security equipment, considering a forced landing. In addition, we intend to investigate situations of landing on a moving platform and the effects in the control system for unknown load mass.

Declarations

Ethical Approval Not applicable

Consent to Participate: Not applicable.

Consent to Publish Not applicable.

Conflict of interest The authors declare that they have no conflict of interest.

Availability of data and material Algorithms and data sets can be made available upon acceptance and publication.

References

AJOT (2021) Online food delivery market to hit $151.5b in revenue and 1.6b users in 2021, a 10% jump in a year. https://bityli.com/6XP0z

Almakhles, D. J. (2020). Robust backstepping sliding mode control for a quadrotor trajectory tracking application. *IEEE Access, 8,* 5515–5525. https://doi.org/10.1109/ACCESS.2019.2962722.

ANAC, (2020). ANAC autoriza teste para entrega de produtos com drones. https://bit.ly/2O1C5U0.

Boysen, N., Briskorn, D., Fedtke, S., & Schwerdfeger, S. (2018). Drone delivery from trucks: Drone scheduling for given truck routes. *Networks, 72*(4), 506–527. https://doi.org/10.1002/net.21847.

Chang, Y. S., & Lee, H. J. (2018). Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. *Expert Systems with Applications, 104,* 307–317. https://doi.org/10.1016/j.eswa.2018.03.032.

Chiang, W. C., Li, Y., Shang, J., & Urban, T. L. (2019). Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization. *Applied Energy, 242,* 1164–1175. https://doi.org/10.1016/j.apenergy.2019.03.117.

Chiella, A. C. B., Machado, H. N., Teixeira, B. O. S., & Pereira, G. A. S. (2019). Gnss/lidar-based navigation of an aerial robot in sparse forests. *Sensors*. https://doi.org/10.3390/s19194061.

de Santana, R.O., Mozelli, L.A., & Neto, A.A. (2019). Vision-based autonomous landing for micro aerial vehicles on targets moving in 3d space. In 2019 19th International Conference on Advanced Robotics (ICAR), pp 541–546, https://doi.org/10.1109/ICAR46387.2019.8981643.

de Souza, J. P. C., Marcato, A. L. M., de Aguiar, E. P., Jucá, M. A., & Teixeira, A. M. (2019). Autonomous landing of UAV based on artificial neural network supervised by fuzzy logic. *Journal of Control, Automation and Electrical Systems, 30*(4), 522–531. https://doi.org/10.1007/s40313-019-00465-y.

Falanga, D., Kleber, K., & Scaramuzza, D. (2020). Dynamic obstacle avoidance for quadrotors with event cameras. Science Robotics 5(40), https://doi.org/10.1126/scirobotics.aaz9712. https://robotics.sciencemag.org/content/5/40/eaaz9712.full.pdf.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., & Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition, 47*(6), 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005.

Gonçalves, V. M., McLaughlin, R., & Pereira, G. A. S. (2020). Precise landing of autonomous aerial vehicles using vector fields. *IEEE Robotics and Automation Letters, 5*(3), 4337–4344. https://doi.org/10.1109/LRA.2020.2994485.

Gonçalves, V. M., Pimenta, L. C. A., Maia, C. A., Dutra, B. C. O., & Pereira, G. A. S. (2010). Vector fields for robot navigation along time-varying curves in n-dimensions. *IEEE Transactions on Robotics, 26*(4), 647–659. https://doi.org/10.1109/TRO.2010.2053077.

Hesch, J. A., & Roumeliotis, S.I. (2011). A Direct Least-Squares (DLS) method for PnP. In 2011 International Conference on Computer Vision, pp 383–390, https://doi.org/10.1109/ICCV.2011.6126266.

Hong, I., Kuby, M., & Murray, A. T. (2018). A range-restricted recharging station coverage model for drone delivery service planning. *Transportation Research Part C: Emerging Technologies, 90,* 198–212. https://doi.org/10.1016/j.trc.2018.02.017.

Koiwanit, J. (2018). Analysis of environmental impacts of drone delivery on an online shopping system. *Advances in Climate Change Research, 9*(3), 201–207. https://doi.org/10.1016/j.accre.2018.09.001.

Lepetit, V., Moreno-Noguer, F., & Fua, P. (2009). EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision, 81*(2), 155. https://doi.org/10.1007/s11263-008-0152-6.

Marut, A., Wojtowicz, K., & Falkowski, K. (2019). ArUco markers pose estimation in UAV landing aid system. In 2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace), pp 261–266, https://doi.org/10.1109/MetroAeroSpace.2019.8869572.

Nguyen, T. M., Nguyen, T. H., Cao, M., Qiu, Z., & Xie, L. (2019). Integrated UWB-vision approach for autonomous docking of UAVs in GPS-denied environments. In 2019 International Conference on Robotics and Automation (ICRA), pp 9603–9609, https://doi.org/10.1109/ICRA.2019.8793851.

Nguyen, T. M., Zaini, A. H., Guo, K., & Xie, L. (2016). An ultra-wideband-based multi-UAV localization system in GPS-denied environments. In 2016 International Micro Air Vehicles Conference.

Nowak, E., Gupta, K., & Najjaran, H. (2017). Development of a plug-and-play infrared landing system for multirotor unmanned aerial vehicles. In 2017 14th Conference on Computer and Robot Vision (CRV), pp 256–260, https://doi.org/10.1109/CRV.2017.23.

Raffo, G. V., & de Almeida, M. M. (2016). Nonlinear robust control of a quadrotor UAV for load transportation with swing improvement. In 2016 American Control Conference (ACC), pp 3156–3162, https://doi.org/10.1109/ACC.2016.7525403.

Rezende, A.M.C., Gonçalves, V.M., Nunes, A.H.D., & Pimenta, L.C.A. (2020). Robust quadcopter control with artificial vector fields*. In

2020 IEEE International Conference on Robotics and Automation (ICRA), pp 6381–6387, https://doi.org/10.1109/ICRA40945.2020.9196605.

Rezende, A.M.C, Miranda, V.R.F., Machado, H.N., Chiella, A.C.B, Gonçalves, V.M., & Freitas, G.M. (2019). Autonomous system for a racing quadcopter. In 2019 19th International Conference on Advanced Robotics (ICAR), pp 1–6, https://doi.org/10.1109/ICAR46387.2019.8981660.

Rossomando, F., Rosales, C., Gimenez, J., Salinas, L., Soria, C., Sarcinelli-Filho, M., & Carelli, R. (2020). Aerial load transportation with multiple quadrotors based on a kinematic controller and a neural SMC dynamic compensation. *Journal of Intelligent & Robotic Systems, 100*(2), 519–530. https://doi.org/10.1007/s10846-020-01195-z.

Sahinoglu, Z. (2008). *Ultra-wideband positioning systems*. Cambridge: Cambridge University Press.

Sayed, A., Tarighat, A., & Khajehnouri, N. (2005). Network-based wireless location: Challenges faced in developing techniques for accurate wireless location information. *IEEE Signal Processing Magazine, 22*(4), 24–40. https://doi.org/10.1109/MSP.2005.1458275.

Schneider, D. (2020). The delivery drones are coming. *IEEE Spectrum, 57*(1), 28–29. https://doi.org/10.1109/MSPEC.2020.8946304.

Thrun, S., Burgard, W., & Fox, D. (2000). *Probabilistic robotics* (Vol. 1). Cambridge: MIT press.

Tiemann, J., & Wietfeld, C. (2017). Scalable and precise multi-UAV indoor navigation using TDOA-based UWB localization. In: 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp 1–7, https://doi.org/10.1109/IPIN.2017.8115937.

Tiemann, J., Ramsey, A., & Wietfeld, C. (2018). Enhanced UAV indoor navigation through SLAM-Augmented UWB localization. In 2018 IEEE International Conference on Communications Workshops (ICC Workshops), pp 1–6, https://doi.org/10.1109/ICCW.2018.8403539.

Villa, D. K., Brandao, A. S., & Sarcinelli-Filho, M. (2020). A survey on load transportation using multirotor UAVs. *Journal of Intelligent & Robotic Systems, 98*(2), 267–296. https://doi.org/10.1007/s10846-019-01088-w.

Wubben, J., Fabra, F., Calafate, C. T., Krzeszowski, T., Marquez-Barja, J. M., Cano, J. C., & Manzoni, P. (2019). Accurate landing of unmanned aerial vehicles using ground pattern recognition. *Electronics*. https://doi.org/10.3390/electronics8121532.

Xuan-Mung, N., Hong, S. K., Nguyen, N. P., Ha, L. N. N. T., & Le, T. L. (2020). Autonomous quadcopter precision landing onto a heaving platform: New method and experiment. *IEEE Access, 8,* 167192–167202. https://doi.org/10.1109/ACCESS.2020.3022881.

Yoo, W., Yu, E., & Jung, J. (2018). Drone delivery: Factors affecting the public's attitude and intention to adopt. *Telematics and Informatics, 35*(6), 1687–1700. https://doi.org/10.1016/j.tele.2018.04.014.

Zúñiga, N.S., Muñoz, F., Márquez, M.A., Espinoza, E.S., & Carrillo, L.R.G. (2018). Load transportation using single and multiple quadrotor aerial vehicles with swing load attenuation. In 2018 International Conference on Unmanned Aircraft Systems (ICUAS), pp 269–278, https://doi.org/10.1109/ICUAS.2018.8453485.