Introducing Autonomous Aerial Robots in Industrial Manufacturing

Francisco J. Perez-Grau[a,*], J. Ramiro Martinez-de Dios[b], Julio L. Paneque[b], J. Joaquin Acevedo[b], Arturo Torres-González[b], Antidio Viguria[a], Juan R. Astorga[c] and Anibal Ollero[b]

[a]*Advanced Center for Aerospace Technologies (CATEC), Seville, Spain*
[b]*GRVC Robotics Laboratory, University of Seville, Spain*
[c]*Airbus Defence and Space, Centro Bahía de Cádiz, Cádiz, Spain*

ARTICLE INFO

Keywords:
aerial robot co-worker
factory automation
industrial application

ABSTRACT

Although ground robots have been successfully used for many years in manufacturing, the capability of aerial robots to agilely navigate in the often sparse and static upper part of factories makes them suitable for performing tasks of interest in many industrial sectors. This paper presents the design, development, and validation of a fully autonomous aerial robotic system for manufacturing industries. It includes modules for accurate pose estimation without using a Global Navigation Satellite System (GNSS), autonomous navigation, radio-based localization, and obstacle avoidance, among others, providing a fully onboard solution capable of autonomously performing complex tasks in dynamic indoor environments in which all necessary sensors, electronics, and processing are on the robot. It was developed to fulfill two use cases relevant in many industries: *light object logistics* and *missing tool search*. The presented robotic system, functionalities, and use cases have been extensively validated with Technology Readiness Level 7 (TRL-7) in the Centro Bahía de Cádiz (CBC) Airbus D&S factory in fully working conditions.

1. Introduction

The growing levels of flexibility and efficiency required by industries demand the massive introduction of robotic technologies into manufacturing tasks [1, 2, 3]. The majority of the robot co-workers in industrial sectors that have been reported refer mainly to manipulators collaborating with humans, ground robots used for transportation, or exoskeletons. Ground robots and Autonomously Guided Vehicles (AGVs) have been largely used in manufacturing centers for decades. Although ground robots are suitable tools for many tasks such as transportation of heavy components and dexterous manipulation, the ground of factories is a dynamic environment with a high density of objects, workers, and other ground robots, which constrain the robot's motion and speed. Besides, ground robots can perturb workers who should be concentrated on their tasks. In contrast, the upper part of the factories is often sparse and static, and in general, they are more suitable for autonomous robot navigation, and some industrial tasks could greatly benefit from utilizing this unused space, for tasks such as the inspection of large areas or structures, or the search for missing items [4].

There is an increasing interest in the use of drones in warehousing and manufacturing operations. However, most systems are manually operated by skilled pilots, and the few autonomous aerial robots that have been reported are still at an early development stage [5]. Very few existing robotic systems with automated functionalities have been validated in the factory in fully operational conditions, and in these systems the autonomous functionalities are constrained to simple tasks, limiting their range of potential applications.

This paper presents the design, development, and validation of a fully autonomous aerial robotic system performing complex tasks in manufacturing industries. The proposed robotic system adopts an efficient modular architecture in which all the required sensors, electronics, and processing are on board the aerial robot. It was adapted to two different use cases, namely *light object logistics* and *missing tool search*, that are relevant in a wide variety of manufacturing industries, and have been extensively validated in the Centro Bahía de Cádiz (CBC) Airbus D&S factory in fully working conditions, see Figure 1.

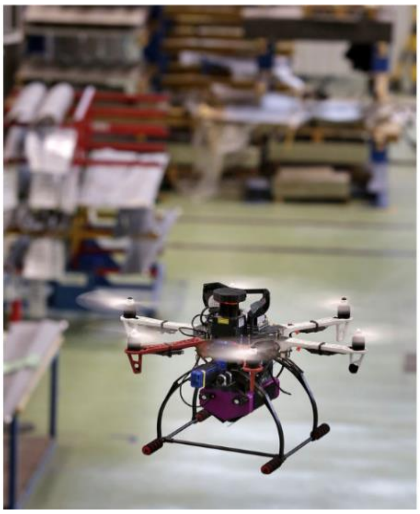

Figure 1: Aerial robot performing autonomous light object delivery in the Centro Bahía de Cádiz (CBC) Airbus D&S factory.

In this work, the University of Seville, FADA-CATEC and Airbus D&S joined complementary efforts to develop and demonstrate how aerial robots can be introduced in industrial processes while keeping or even improving manufacturing quality. To the best of the authors' knowledge, it is one of the first times that a fully autonomous aerial robotic system performing complex tasks has been validated with Technology Readiness Level 7 (TRL-7) in a manufacturing plant in full working conditions. The presented system was selected as one of the two aerial robotics projects finalists (out of more than 50 participants) in the European Robotics Challenges (the EU-funded EuRoC project) and received the "Best Drone-based Solution" award in the 1st EU Drone Awards [6]. A video showing the operation of the proposed robotic system in the factory is available at https://youtu.be/tmxPpsGN5ts.

The rest of the paper is structured as follows. Section 2 summarizes the main related work. The main objectives and requirements are briefly described in Section 3. Section 4 presents the robotic system and its main functional modules. Sections 5 and 6 describe respectively its implementation and validation with TRL7 in the CBC Airbus D&S factory. The main lessons learnt are summarized in Section 7. The conclusions and future work are summarized in Section 8.

* Corresponding author
fjperez@catec.aero (F.J. Perez-Grau)

2. Related Work

A very wide variety of approaches and techniques have been developed to improve the degree of autonomy of aerial robots in complex scenarios. We first summarize the techniques more related to our approach focusing on robot pose estimation and robot navigation, and then, present the main existing works dealing with the use of aerial robots in factory and warehousing applications.

Arguably, the most extended approaches for indoor autonomous navigation of aerial robots are based on visual cameras due to their low cost and weight. A wide variety of successful techniques for visual-inertial odometry and visual Simultaneous Localization And Mapping (SLAM) have been proposed using monocular systems [7, 8], stereo-vision [9, 10], or RGB-D sensors [11, 12]. However, visual-based localization has limitations that are relevant in factory and warehouse environments. First, they are sensitive to lighting conditions, and factories usually have high densities of objects that can cause strong local lighting changes. Solving complex illumination conditions is an active research topic especially for ground robots [13], but aerial robots pose additional limitations in terms of vibrations and motion blur. Also, factories can have large empty spaces, where the objects are seen at large distances, and visual-based techniques do not always behave well in cases where, under small robot displacements, there is little change in the gathered images. Besides, the range of stereo-vision is determined by its baseline, and aerial robots have strong size and payload limitations. Depth measurements from RGB-D and stereo sensors are usually limited to 8-10 m, which could be insufficient for large indoor scenarios such as aircraft manufacturing plants.

Other works have proposed the use of visual markers for robot global self-localization, see e.g. [14, 15]. These methods are efficient and simple but dependent on the image quality. Similar approaches using radio tags around the operating environment and on board the aerial robot have also been proposed, see e.g. [16]. However, in some environments, they can suffer from strong radio signal attenuation and high measurement outlier levels. Besides, both cases require the installation of specific infrastructure in the environment, which hampers the solution scalability and incurs deployment and maintenance issues.

LiDARs have been largely employed for obtaining robust and accurate measurements in large environments. Although they are usually heavier and more expensive than most cameras, their performance is not affected by lighting conditions, allowing for 24/7 operations and also in environments with fog or steam. LiDAR data is provided with high resolution and at sufficiently high sampling rates for pose estimation. A good number of LiDAR-based odometry and SLAM techniques have been proposed, see survey in [17]. Standard 3D LiDARs are generally big and heavy for moderately-sized aerial robots.

Many localization techniques rely on a pre-built map to obtain higher levels of accuracy. This is well suited for factories since the aerial robot will be operating in the same area recurrently. One of the most commonly used approaches is Monte-Carlo Localization (MCL) [18], which makes use of a Particle Filter to estimate the robot pose analyzing the matching of the sensor measurements with the map. MCLs have been often proposed to obtain long-term aerial robot localization, see e.g. [19], which presented an MCL that combined ultra-wideband and RGB-D measurements, or [20], in which 3D LIDAR and visual measurements were fused using an MCL. Other approaches are based on pure registration of point clouds. The Iterative Closest Point (ICP) algorithm [21] is widely used to efficiently match between two consecutive geometric point clouds (local robot pose estimation) or between the received point cloud and a previously registered map to obtain global pose estimation or to reduce drift accumulation.

Regarding robot navigation, many different approaches have been proposed for online and offline trajectory planning targeting aerial robots, from relative or local navigation to global navigation based on predefined maps. The most efficient solutions usually combine both approaches, such as [22]. Sampling-based algorithms, such as Rapidly-exploring Random Trees (RRT) [23, 24], have been employed to navigate autonomously through large spaces. For instance, different probabilistic methods are used in [25] to solve the motion planning problem in aerial robots. However, the trajectories planned by probabilistic methods can vary significantly from one execution to another, resulting in potentially inconsistent navigation. In contrast, graph-based algorithms, such as Lazy Theta* [26], provide deterministic paths and can reuse previous computations, such as in [27], even though this work was only validated in a laboratory environment. This brings important benefits for navigating inside a factory in everyday conditions since we are very interested in robustness and accuracy. Besides, other works focus not only on collision avoidance for safety, but also on maintaining the efficiency of manufacturing processes [28]; while focused on human-robot collaboration, a similar approach could be explored in case more than one aerial robot is used in a factory. The adopted approach should have a very low computational cost since all the modules of the proposed robotic architecture are executed onboard. In our problem, a navigation system based on pre-defined waypoints combined with reactive obstacle avoidance methods is a robust and efficient alternative. Of course, it requires selecting appropriate methods, along with robust and efficient algorithms, as well as exhaustive development and validation methodologies.

Over the past decade, aerial robot technologies have brought a revolution in many fields, cost reductions and robustness improvements have made aerial robots an economically viable option for a broad range of industrial sectors. Aerial robots are currently making a real impact on applications that take advantage of their ability to fly quickly and safely at high altitudes or reach difficult or hazardous places, in a wide variety of sectors. Their interest is not limited to just using an aerial camera, but are also starting to involve more sophisticated procedures including physical interaction, such as [29], which presented a semi-autonomous aerial robotic manipulator for contact inspection in industrial settings. Most of the above applications take place outdoors, where Global Navigation Satellite Systems (GNSS) are available for localization or navigation. Manufacturing operations, on the contrary, are almost exclusively in populated indoor spaces, and this involves relevant challenges to provide the required accuracy and robustness to perform autonomous missions. Safety, noise, and privacy are common concerns, and the maneuverability of the aerial robots should be adapted to the specific scenario. On the other hand, indoor operations are not subject to Remotely Piloted Aircraft System (RPAS) regulations, and weather conditions are not an issue.

There is an increasing interest in using drones in warehousing operations such as inventory checks, intra-logistics applications, or inspection and surveillance, among others. For instance, work [5] presents a summary of the main reported drone-based systems for warehouse operations. These systems are still at an early prototype testing phase, and in most cases, the drones are manually operated within line of sight by skilled pilots [30]. Eyesee [31] is one of the first drone-based solutions for automated inventory in warehouses. It performs automated barcode reading and flight navigation using vision-based localization. In this system, the accuracy constraints of vision-based localization in factory settings are compensated by performing simple robot tasks and trajectories, which although sufficient for the envisioned inventory application, can constrain its use in other applications.

Recently, the advantages of radio-based sensors in robotized warehouses and industrial scenarios have motivated intense research and commercial effort. Available commercial solutions are mostly based on ground robots that are able to navigate and locate RFID tags in 2D, see e.g. [32, 33]. To the best of our knowledge, there is not any commercial product based on an autonomous aerial robot yet. The potentialities of such robotic systems have motivated strong research interest in the last years. Work [34] presented a system to track the position of an aerial robot while it navigates through the environment by using static UWB anchors at previously known positions. Other works detect and/or locate radio tags assuming that the robot position is known or can be estimated. The work in [35] presents a blockchain-based architecture to manage a warehousing application based on RFID tags installed on the stored items. This work focuses on the communication architecture and only detects the tags, but does not estimate their position. The work in [36] fuses measurements from a LiDAR and a camera to generate a map of a warehouse and navigate through it, and uses RFID signals and visual markers to detect the presence of placed stock – each item has its own RFID tag–. However, they only validate presence detection and do not give tag localization errors. Besides, RFID transmissions can cover low distances since the size of receivers increases significantly with the transmission range. The work in [37] uses a large high-range static RFID system that emits signals at a distance of 50 m, which are responded to by the RFID tags. These responses are read by a small RFID receiver on board an aerial robot. It avoids the need for a long-range RFID emitter on board the robot, but requires the robot to fly near the RFID tags to receive their (weak) RFID responses. Our work uses UWB tags instead of RFID to alleviate these issues and be able to accurately locate tools in a complex industrial environment.

The problem of searching for missing tools is of high importance to the aerospace industry. A few commercial solutions to keep track of them are starting to arise. One example is [38], which uses a combination of RFID tags installed at the tools and UWB tags installed at the workers' badges, belt-clips, or helmets. Each time a worker draws a tool from the cabinet, its UWB tag is paired with the tool, and its location is tracked using UWB anchors installed in the factory. If the tool is not returned, the tracking history of the worker is used to search for the missing tool. This approach presents three main drawbacks for our problem: first, an ad-hoc infrastructure is required to keep track of the worker's position; second, the system provides a full tracking history instead of a specific 2D or 3D location to search at, which leads to longer search times; and third, if another worker (or robot coworker) displaces the tool, it separates from its assigned worker's track. Our system uses UWB tags at the tools and a UWB system at the aerial robot to avoid these issues: it needs no UWB infrastructure, it provides the current location of the tool, and it is robust to manipulation by ~~several~~ workers.

3. Problem Formulation and Solution Design

Although ground robots have been largely used for decades [39], aerial robots provide interesting advantages to increase the level of automation in manufacturing industries. The upper part of factories is rather sparse and static, whereas the lower part is often very dense and dynamic, constraining the motion of ground robots. As shown in Figure 2, this is the case in the CBC Airbus D&S factory. Also, safety constraints impose low-speed motions for ground robots, while this restriction is not present for aerial robots. Moreover, notice that RPAS regulations do not apply in indoor environments.

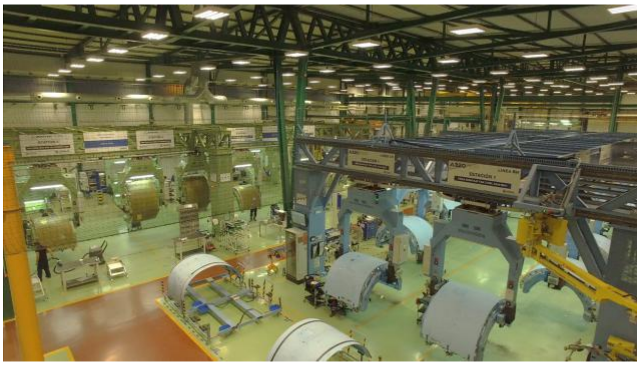

Figure 2: Aerial view of the CBC Airbus D&S factory.

The introduction of aerial robots would increase the useful factory production space, while at the same time could indirectly reduce the equipment needed on the floor. Introducing aerial robots in a manufacturing factory poses challenging issues. From a technical perspective, the system should include precise GNSS-denied long-term robot pose estimation, as well as efficient planning and navigation systems suitable for dynamic and tight environments. From implementation and operational perspectives, the system should ensure safe and robust aerial robot navigation in crowded environments. Safety and robustness for everyday operations in the factory were fundamental requirements that drove the design, development, and implementation of our system.

3.1. Envisioned Use Cases

Our objective is to increase the level of automation in industrial processes by integrating fully autonomous aerial robots collaborating with humans. Airbus D&S identified two potential use cases where aerial robots could improve productivity by reducing operation times and manufacturing costs: light object logistics and missing tool search. In both cases, the aerial robot velocity and flexibility are key issues to improve the currently implemented processes, and also, two main advantageous features of aerial robots over other robotic solutions.

The assembly of aerostructure parts requires many different materials and small and light components (rivets, sealing materials, glues, among others). For instance, in the CBC factory, more than 500 different small components and materials are necessary for the assembly of every single fan cowl. The transportation of light components (lower than 500 g) constitutes around 98% of the internal logistics operations in this factory, which yields the importance of reducing costs in these operations. Moreover, a number of these components such as sealants are kept refrigerated before use, and they must be heated just before using them and, once heated, have a very limited lifetime. These constraints, together with the high manufacturing quality requirements, and the wide variety of small components, make light object logistics very relevant in the assembly of aerostructure parts. All these light components are located at specific storage centers, at different places in the factory. In the current Airbus D&S operation, when a worker finds that he has run out of a component, he informs his supervisor, which starts a procedure to ask for it. This procedure ensures traceability and quality requirements, which are fundamental in the aeronautic industry. When the small component is ready, the supervisor brings the component to the worker.

Our first goal is to use aerial robots for light object logistics. The aerial robot transports and provides light components to the workers that demand them. For transportation, the light components are inserted in capsules of a suitable size. The aerial robot is equipped with a cargo mechanism for the transportation and delivery of capsules. When a worker needs a specific component, he informs the system using a visual interface that is executed e.g. in a tablet. This request is sent to the storage center, where the component is prepared. Another worker at the storage center places the light object in the capsule, puts the capsule in the robot cargo mechanism, and commands the aerial robot to perform a transportation and delivery mission. Finally, when the component has been delivered, the worker who made the request acknowledges reception through the same interface. The proposed system also ensures Airbus D&S traceability and quality requirements.

Many manufacturing processes require a high number of different and specialized tools that are used by workers. In the CBC Airbus D&S factory, more than 100 different tools are involved in the assembly of every fan cowl. Missing tools occur occasionally originating delays and reducing productivity. Moreover, missing items are particularly harmful in aeronautic industries since they may cause Foreign Object Damage (FOD) events. Most factories that require many different specialized tools are endowed with intelligent tool cabinets that can detect when a tool is missing, but cannot locate the missing tool. When a tool is considered missing, it is searched in the factory causing delays and requiring significant human labor.

In the second use case, a missing tool search and localization functionality was implemented. The objective is to compute the location of the missing tools with sufficient accuracy such that they can be easily found by human workers. Missing tool search can be performed on-request (e.g. at the end of every work shift) or automatically (e.g. executed while the robot is navigating performing light object logistics tasks). Each tool is equipped with an Ultra-Wide Band (UWB) tag with a unique identifier. The robot is equipped with a UWB emitter that can send requests and measure the range to the UWB tags in its radio coverage. The gathered range measurements from each tag are processed to estimate the tool's 3D location.

3.2. Requirements for the Industry

RPAS regulations do not apply indoors. The main limitations for the operation of RPAS outdoors are not present inside the factory, such as the restriction to visual line of sight flights, operations at night, or even the presence of a licensed pilot.

A safety analysis was carried out along with the Industrial Safety and Health Department at Airbus D&S, following the main industrial safety and health European directives:

- Directive 89/391/EEC: Measures to Encourage Improvements in the Safety and Health of Workers at Work;
- Directive 92/58/EEC: Minimum Provisions for Safety and/or Health Signs at Work; and
- Directive 89/654/EEC: Minimum Health and Safety Provisions in Workplaces.

This analysis led to the design of several required actions and modifications in the factory prior to allowing the use of aerial robots during working hours with the main goal of reducing the risk of accidents with workers. These actions are compiled in Section 5.

Another fundamental factor that was raised by Airbus D&S during the initial design stage is the envisioned use of this technology in larger areas, and/or different factories across Europe. Scalability and implementation flexibility required a minimum set-up in the factory. To achieve this, we focused on providing a fully onboard solution, where all the sensors, electronics, and processing required for the full system operation are on board the aerial robot.

Also, there were requirements in terms of productivity to be accomplished by the developed solution. In high added-value human-dependent tasks, such as aircraft assembly processes with highly demanding lead times, the reduction of person-hours and operation times is key for the success of this technology. Table 1 summarizes the main operational requirements for each use case. Each requirement includes a minimum acceptable (soft) value and a highly desirable (hard) value. A final critical requirement was that the factory's productive activities can never be stopped or disturbed. This involved a careful design of the implementation strategy, which is described in Section 5.

Table 1
Summary of the main productivity-related operational requirements.

Req.	Description	Soft value	Hard value
RL1	Weight of the cargo	300 g	500 g
RL2	Light object delivery time	5 min	3 min
RT1	Time to find all the missing tools	5 min	3 min
RT2	Tool search accuracy: maximum radius of the elliptic confidence interval (3σ)	3 m	0.5 m
RT3	Number of missing tools that can be located simultaneously	3	10

4. The Developed System

The robust operation of autonomous aerial robots in indoor manufacturing industry in fully working conditions with the presence of workers and high density of metallic structures is a challenging task. Our aim is to develop a system that works reliably and safely in the factory with the minimum possible setup, discarding solutions that require the installation of infrastructure in the factory, and hence also avoiding the need for procedures for their calibration and maintenance.

4.1. Hardware Description

The aerial platform, see Figure 3, is based on the hexacopter *DJI F550 Flame Wheel*. It provides a suitable trade-off between size, payload capacity, and flight autonomy while offering plenty of space for all the onboard subsystems. The landing gear is a modified version that offers more space underneath the main platform, which is devoted to the installation of a cargo hold for the transportation of light objects. The selected low-level control autopilot is the general-purpose flight controller *Pixhawk 1* [40] with PX4 [41], an open-source flight control software for aerial robots that provides a flexible set of tools. All the processing tasks are performed on board in a standard *UP*-board, configured as the companion computer connected to *Pixhawk 1* using the *MAVLink* protocol [42] over a serial connection. The software was developed on ROS (Robot Operating System) [43] on top of Ubuntu Linux.

The adopted main localization and navigation sensor is a LiDAR. We used a *Hokuyo UTM-30LX* 2D LiDAR, which has a range of 30 meters and a scan rate of 40 Hz, sufficient for the type of scenarios considered. This sensor acquires scans lying only in the horizontal plane, but similarly to [44], we installed small mirrors to deflect a narrow portion of the laser beams to obtain range measurements in the vertical plane. With this configuration, we lose a fraction of the horizontal LiDAR field of view in exchange for having a precise altimeter with the same sensor, avoiding the need for a separate altimeter. Moreover, since manufacturing plants often have dense floors whereas their ceilings contain few static elements, the adopted solution was to place the mirrors to deflect the laser beams upwards, not downwards. The LiDAR sensor was mounted on top of the aerial frame, and the mirrors were placed in a specifically designed 3D-printed mount.

To deal with the light object logistics use case, the aerial robot is equipped with a cargo hold designed to accommodate the items to be transported. This small compartment is automatically opened to release its contents using two small *Futaba* servo motors, controlled by the onboard computer. For the missing tool search use case, a UWB emitter is also integrated on the aerial robot connected to the onboard computer via USB. The selected sensor is an *In-Circuit Radino 32 DW1000* tag module. Based on the *Decawave DW1000* chip, it encapsulates an antenna, a UWB processing chip, a USB interface, and a battery. It has a maximum detection range of 300 m (outdoors) and a range error standard deviation of 10 cm (30 cm indoors). This module is also attached to the tools to be searched in the factory.

Figure 3 shows all the physical system components, describing where each one of them is located in the aerial platform.

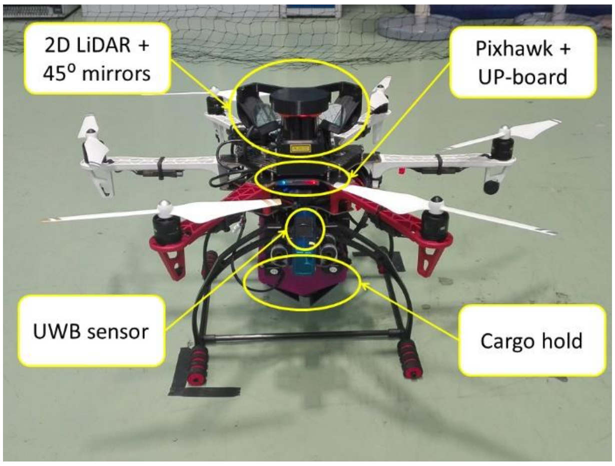

Figure 3: Closeup picture of the developed aerial system.

4.2. Architecture Design

Figure 4 shows a scheme of the developed architecture. All the modules are executed in real-time on board the robot. The autopilot is in charge of controlling the aerial robot flight following the received references. It also participates in the robot pose estimation providing the readings from the Inertial Measurement Unit (IMU), and integrates such pose estimation to close the control loop. The *6DoF Pose Estimation* modules provide estimations of the robot position and orientation integrating the measurements provided by the 2D LiDAR and the IMU as detailed in Section 4.3. Finally, the *Autonomous Navigation* system implements the robot navigation, and provides velocity commands as references to the autopilot, see Section 4.4.

The architecture also includes the visual interface module *GUI*, that the human worker would use to select the desired mission: *Light Object Logistics* or *Missing Tool Search*, which are responsible for performing the considered use cases. Both send to *Autonomous Navigation* the waypoints for the accomplishing of each type of mission. *Light Object Logistics* also triggers the actuators to open and close the cargo hold compartment for dropping the payload. *Missing Tool Search* receives the robot pose and the range measurements from the UWB attached to the tools being searched and estimates their 3D position.

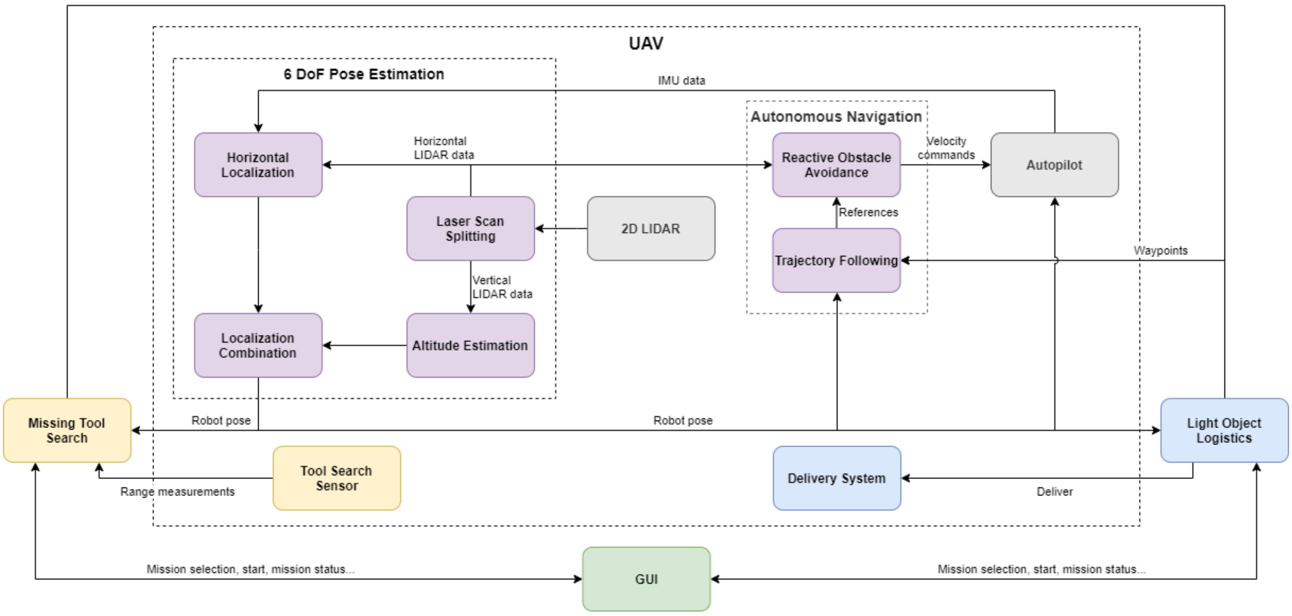

Figure 4: Functional architecture of the proposed system.

4.3. 6DOF Pose Estimation in the Factory

The scheme adopted for the aerial robot 6-DOF pose estimation is shown in Figure 4. The involved modules are described in the following.

4.3.1. Laser Scan Splitting

This module splits the raw 2D LiDAR scan into its horizontal and vertical measurement segments. Each scan is composed of 1080 points (270º range at 0.25º angular resolution), arranged in a single array. The mirrors deflect the laser scan for 22.5º at each end of the sensor range, which corresponds to the first and the last 90 points from the raw LiDAR scan array. To avoid mirror imperfections at the mirror edges, 20 points at the interface between the vertical and horizontal segments are removed. Therefore, the vertical scan segment is created with the measurements with indices [0-79] and [1000-1079], and the horizontal scan corresponds to those in the range [100-979].

4.3.2. Altitude Estimation

This module filters the vertical LiDAR scans to provide estimates of the robot's altitude. There are several difficulties to be solved. First, not all factory ceilings feature a constant height. Height variations in different parts of the factory should be compensated. Also, many factories have static structural elements, such as beams, or objects hanging from the ceiling, e.g. lamps. These objects may appear as sudden discontinuities in the vertical LiDAR measurements that need to be filtered. In our case at the CBC factory, the ceiling height is locally uniform, but contains many lamps and beams, as can be seen in Figures 2 and 6. These artifacts are filtered by exploiting the fact that the LiDAR measures actual ceiling points most of the time. Hence, the maximum measured distance from the incoming vertical scan segments is taken as the robot-ceiling measurement. A lamp or a beam could affect one of the mirrors, but it is very unlikely that all measurements of both separate mirrors are affected simultaneously, as can be observed in Figure 5-right.

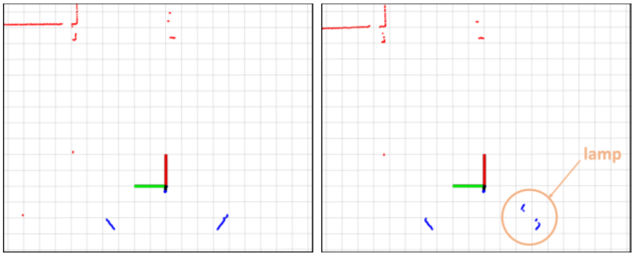

Figure 5: Layout of the split 2D LiDAR scan. The red points represent LiDAR measurements in the horizontal plane (caused by walls and beams), while the blue points correspond to LiDAR measurements in the vertical plane. Two cases are shown: left) the vertical segments of both mirrors measure robot-ceiling distances, and right) one vertical segment measures robot-ceiling distance while the other contains points partially reflected by a lamp.

Altitude Estimation operates in two main steps. First, the average of the N greatest measurements in the vertical segments received at time k is computed. Next, temporal discontinuities in the estimated altitude are identified: if the difference between average measurements at time k and k-1 is higher than a threshold, it is assigned as caused by a vertical discontinuity in the measurements. In this case, an offset is added to the average measurement to keep the aerial robot at the same estimated height, preventing that sudden changes in altitude estimation affect the robot control. The LiDAR high scan rate (40Hz) helps to avoid this effect while the robot is deliberately ascending/descending, e.g. during take-off/landing.

4.3.3. Horizontal Localization

High accuracy in horizontal localization is a strong requirement for safety and mission accomplishment. In the CBC Airbus D&S factory, the flight corridors for the aerial robot have a width of 3 m. Also, in the light object logistics use case, the top openings of each hopper at each delivery point is an 80x80 cm square (see Figure 6), which requires a maximum horizontal localization error of 40 cm, which should be fulfilled after the robot has traversed almost 40 m from take-off.

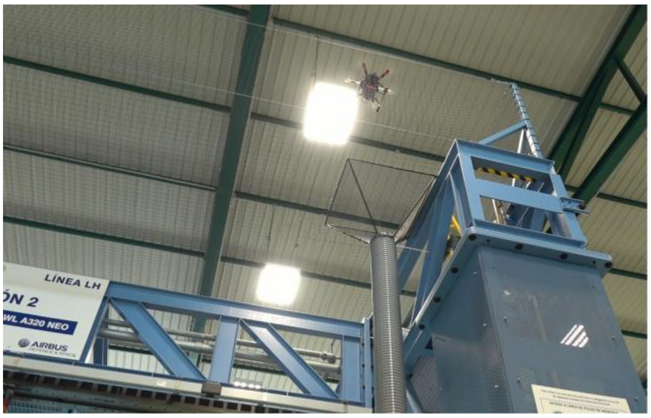

Figure 6: Light object logistics delivery point, with the aerial robot positioned right before releasing the requested component.

The adopted approach differentiates between global and local pose estimation. The global pose estimation provides absolute estimates of the robot pose w.r.t. a coordinate system attached to the factory map, and is based on a map alignment and correction strategy. The robot waypoint locations are specified within this global map, thus global pose estimates are used to guide the aerial robot. In contrast, the local pose estimation provides ~~position and orientation~~ estimates w.r.t. the robot pose where it started processing consecutive ~~when the previous~~ LiDAR scans ~~was received~~, and operates at higher rates due to the lower amount of information to be processed. Local pose estimation is mainly based on laser-based odometry. Both methods are based on registration (measurement alignment), one of the most well-known approaches for pose estimation [45], which objective is to find the rigid transformation that best aligns data so that they can be placed in a common coordinate system.

The local pose estimation method uses the ICP algorithm for aligning the laser scans received at consecutive times, providing LiDAR-based odometry. Aligning consecutive laser scans is prone to drift (error accumulation over time), hence our approach builds an overall map as the scans are being registered. Then, subsequent scans are compared back to this ’local’ map. This persistent comparison prevents error accumulation but increases memory usage as the map grows. To prevent this, downsampling configurations for this map are adopted consequently. ICP requires a good initial estimate of the transformation between the compared scans to prevent being stuck at local minima. In our case, no initial transformation is needed because the scan rate is sufficiently high (40 Hz). Nevertheless, ICP convergence can fail occasionally. In these cases, the method first recognizes and then corrects such failures. Scan misalignment is detected by monitoring consecutive pose estimations. If the difference exceeds a given threshold (in both position and orientation), the new estimated pose is discarded, and the ’local’ map is reset using the current laser scan. This strategy does not affect flight performance, preventing undesired effects at the cost of introducing some drift when this effect occurs.

Although this method is suitable for flight stability, it requires an additional global matching for accurate pose estimation.

The global pose estimation method makes use of the same ICP algorithm to align the current laser scan with a static map of the factory. This map is pre-computed during prior flights using the local pose estimation approach, but ensuring that the flight speed and overall motion are convenient for not inducing any map reset. ~~Needless to say,~~ This static factory map only requires to be computed once. The alignment of the current laser scan with the global static map helps to find out the global robot pose. By comparing this pose to the current local pose, we are able to correct the drift introduced by the local pose estimation method; hence, such adjustment allows the aerial robot to safely arrive at the correct destination with the required accuracy.

Finally, module *Localization Combination* integrates the above estimations to assemble the full 6-DoF localization of the robot before feeding it to the autopilot control. The robot position is obtained by combining the *X* and *Y* coordinates provided by the local pose estimator (which has been corrected with the global pose estimation) with the *Z* coordinate provided by the altitude estimator. The orientation is obtained by combining the yaw angle estimated by the local pose estimator with the roll and pitch angles from the IMU.

4.4. Autonomous Navigation

In the design of the autonomous navigation system, we preferred safety and robustness for everyday operation in the factory rather than efficiency. We adopted, see Figure 4, a robust robot navigation scheme based on predefined waypoints that were selected considering aspects such as the expected trajectories for each use case, the geometry of the factory and established flight corridors, the safety of workers, and also practical considerations such as the positions of the air vents and large doors, which could create wind turbulence that might hamper the aerial robot control. Figure 7 shows a layout of the factory with one example of the robot trajectories for each type of mission and the involved waypoints. All the waypoints (eleven in total) were set at the same altitude: 7.5 m above the ground level. Two types of autonomous missions were defined, one for each use case, both expressed as concatenations of elemental tasks such as take-off, go to a waypoint, or open the cargo hold to release the light object.

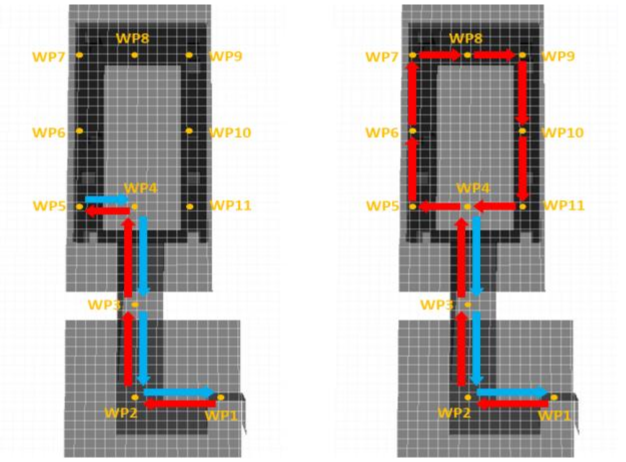

Figure 7: Layout of the factory with the selected navigation waypoints (in orange) and sample trajectories for the two missions: light object logistics (left) and missing tool search (right).

Table 2 shows the tasks considered along with the accepted parameters for each task. In light object logistics missions, the robot takes off, flies from the storage center to the delivery point of the worker that requested the object (delivery point *A* at waypoint *WP*5 in the example shown in Figure 7-left), delivers the object by consecutively commanding the opening and closing of the cargo hold, returns to the storage center, and lands. In missing tool search missions, the robot takes off, describes the tool search mission trajectory defined to cover the required factory zone (the *Fan Cowl Assembly Area* in our case), returns to the storage center, and lands.

Table 2
Elemental tasks in the proposed system.

Task	Parameters	Description
Take-off	altitude	Commands a vertical take-off to a specific altitude
Land	altitude	Commands a vertical land from a specific altitude
GoToWP	waypoint 3D coordinates, yaw angle, wait time, delivery flag	Commands new waypoint, velocities are computed to fulfill the time constraints
OpenCargoHold	n/a	Triggers actuators to open the cargo hold
CloseCargoHold	n/a	Triggers actuators to close the cargo hold

The waypoints for each mission are provided by modules *Light Object Logistics* and *Missing Tool Search*, see Figure 4. *Trajectory Following* ensures that the robot accurately follows the expected trajectory to accomplish the mission. The predefined safe trajectories are composed of a collision-free path between waypoints, each with an associated time reference defined according to the robot kinematics and scenario constraints. Waypoints are defined w.r.t. the reference frame of the static global factory map. However, the aerial robot autopilot accepts references expressed in its local reference frame. *Trajectory Following* receives both the global and the local pose estimations and calculates the current transformation matrix between them to provide the appropriate references to the autopilot control system. *Trajectory Following* is also in charge of transitioning between waypoints during the mission. When the aerial robot pose reaches a sphere of radius *wpR* around the last commanded waypoint, *Trajectory Following* sends the autopilot the coordinates of the next waypoint.

For additional safety, the autonomous navigation system includes one functionality for detecting and avoiding unexpected dynamic obstacles (e.g. other aerial robots) or static obstacles not included in the static global map. The *Reactive Obstacle Avoidance* module implements a basic reactive obstacle avoidance method. It receives the estimated robot pose, the 2D LiDAR scan, and the autopilot references calculated by *Trajectory Following*. It defines a safety spherical area with a radius of 1.2 m around the aerial robot. If it detects an object within this area, it generates a reference to the autopilot in the opposite direction. The LiDAR high scan rate (40 Hz) ensures good performance with obstacles that are static or move with moderate velocities. If the presence of the obstacle persists for a given time compromising the mission accomplishment, the mission is aborted for safety. This mechanism indirectly provides additional protection against potential robot localization drift.

4.5. Tool UWB-based Localization

The objective of this module is to estimate the 3D location of missing tools. All tools are equipped with a UWB tag with a unique identifier (ID). While the robot flies, it periodically (every 2 s) emits UWB signals with those IDs. Every time a UWB tag attached to a tool receives a signal with its ID, it sends back a response message, from which the onboard UWB device extracts a range measurement. Robot motion is slow when compared to range measurement gathering, simplifying measurement synchronization.

This module uses the robot location estimates and the range measurements gathered from the missing tools. Manufacturing plants often have multiple metallic structures that make UWB measurements highly prone to outliers (due mainly to multi-path propagation phenomena).

After outlier filtering, a Bayesian method integrates the filtered UWB measurements using two sequential stages executed independently for each tool. First, a Particle Filter (PF) approximates the localization of the tool, which helps to deal with multi-modality that appears in range-only localization. Second, after the PF converges to a uni-modal solution, the PF is terminated and the resulting tool estimation is used to initialize an Extended Kalman Filter (EKF) that keeps refining the tool location estimation. It is known that EKFs are more computationally efficient than PFs and deliver similar results when the problem has already reached uni-modality [46]. Every tool has different radio IDs, which makes the association of measurements with their correspondent PF-EKF pair straightforward. Independent PF-EKF schemes for each tool were preferred to the more complex multi-tool filters for computational efficiency. The adopted approach enabled real-time onboard execution simultaneously with the rest of the modules.

4.5.1. Outlier Filtering

Outlier rejection works as follows. Let z_k be the range measurement received from a UWB tag at time *k*. Let X_k^r be the estimated position of the robot at time *k*. Assume that the robot will never get closer than $3\sigma^{UWB}$ to any tool, where σ^{UWB} is the standard deviation (assumed constant) of the UWB range measurement error. σ^{UWB} is usually low for current UWB technology (0.2 m in our case), hence this assumption is not a constraint in practical terms. Let σ_k^r be the square root of the maximum eigenvalue of Σ_k^r (the robot location covariance matrix) so that the robot location uncertainty is represented by its most restrictive 1-dimensional representation. With the above assumptions, if the robot moves between times $k1$ and $k2$ away from the tool, the resulting UWB measurements can be analyzed in Figure 8. It can be seen in Figure 8-left that the minimum feasible difference between the two inlier measurements z_{k1} and z_{k2} is: $z_{k2}^{min} - z_{k1}^{max}$. On the other hand, Figure 8-right gives the maximum feasible difference: $z_{k2}^{max} - z_{k1}^{min}$. Equivalently, if the robot moves straight to the tool, the maximum and minimum differences remain the same but negated in sign. Since these are the boundary cases of all 3D movements of the robot, we can define the maximum and minimum feasible differences between two inlier measurements z_{k1} and z_{k2} as:

$$\|z_{k2} - z_{k1}\| \geq d(X_{k2}^r, X_{k1}^r) - 6\sigma^{UWB} - 3\sigma_{k2}^r - 3\sigma_{k1}^r \quad (1)$$

$$\|z_{k2} - z_{k1}\| \leq d(X_{k2}^r, X_{k1}^r) + 6\sigma^{UWB} + 3\sigma_{k2}^r + 3\sigma_{k1}^r \quad (2)$$

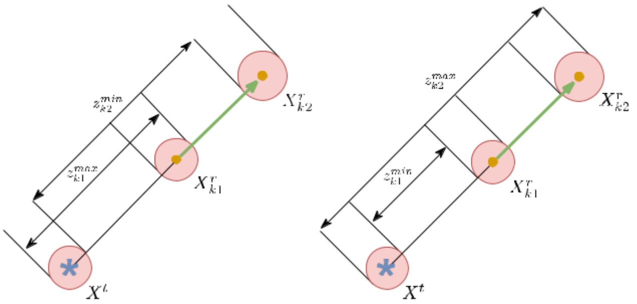

Figure 8: Robot moving straight away from the tool (from X_{k1}^r to X_{k2}^r). The red circles represent the $3\sigma_k^r$ uncertainty of the robot's positions (orange dots) and the radio range measurements (depicted as uncertainty around the real tool position X^t at the blue asterisk): left) case of minimum difference between inlier measurements; and right) case of maximum difference between inlier measurements.

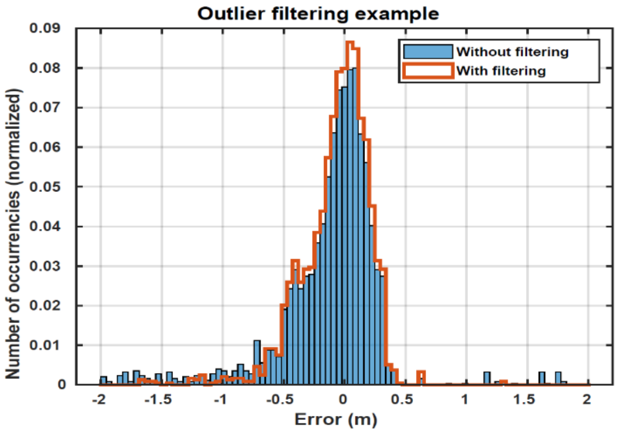

Figure 9: Histogram of UWB measurement errors in one experiment in the factory before (in blue) and after (in red) outlier filtering.

This expression serves as a check for outlier detection since it will be true (with high probability) if the measurement is not an outlier, i.e. if all the measurement and localization errors lie in their own ±3σ range, which occurs with probability 98.8% = 0.997^4 (notice that 99.7% is the probability that one error lies in its ±3σ boundary assuming Gaussian distributions). Based on expressions (1) and (2), a median filter was adopted to deal with possible consecutive outliers. First, the filter discards all the UWB measurements taken when the robot height in X_k^r is lower than 0.5, since the measurements near the floor have a high outlier probability. The filter maintains 3 vectors with the last *N* robot pose estimations, robot pose uncertainties, and UWB measurements for that tool. Each new measurement and robot pose (X_k^r, σ_k^r, z_k) is verified to satisfy expressions (1) and (2), but comparing not with the previous measurement but against the median of the measurement vector and its corresponding robot pose. Hence, if the evaluation of (1) and (2) is true, the measurement z_k is considered an inlier. If not, it is considered an outlier. This simple and efficient method provided satisfactory results. Figure 9 shows that most of the outliers in one experiment were removed, except for a few remaining outliers that are caused by a tool that was confined between metallic structures, which resulted in a significantly higher σ^{UWB} than that used.

4.5.2. The PF Stage

Once the UWB measurements have been filtered, they are introduced in the PF-EKF localization scheme. The first stage is the PF, a Bayesian recursive filter suitable for multi-modal estimation problems such as range-based localization. The adopted

PF follows the structure described in [46]: each particle $\tau^{[j]}$ of the PF represents a hypothesis of the position of the tool, and τ is the full probability distribution of the tool's position, which is given by the set of all particles. The PF is initialized when the first measurement of a missing tool UWB tag, z_0, is received. The initial particles are drawn from a 3D annular distribution with mean z_k and width $6\sigma^{UWB}$. This distribution takes into account all the possible 3D locations of the tool that may have led to the received measurement. All particles located higher than 3 m are rejected since tools are always at human reach altitudes. The particles with negative heights (located under the floor) are also removed.

Once the PF has been initialized, its particles are recursively updated integrating each new measurement z_k, making them condensate towards the tool's real position over time. Notice that the tools are assumed static, i.e. the predicted position of the tool is the same as in the previous instant. Hence, the particle prediction model adopted was the static model $\tau^{[j]} = \tau^{[j-1]}$. In the PF update stage, with each new measurement z_k, the weight ω_j of each particle (the probability that particle $\tau^{[j]}$ reflects the actual position of the missing tool) is updated using the following Gaussian likelihood function:

$$\omega_j = \omega_j \frac{1}{\sigma^{UWB}\sqrt{2\pi}} exp\left(\frac{\left(d\left(\tau^{[j]}, X_k^r\right) - z_k\right)^2}{2(\sigma^{UWB})^2}\right) \quad (3)$$

where $d\left(\tau^{[j]}, X_k^r\right)$ is the distance between the particle $\tau^{[j]}$ and the current robot's position X_k^r. This distance is also the expected measurement at instant *k* for particle $\tau^{[j]}$, and thus the likelihood function relates to the similarity of this expected measurement with the actual measurement z_k. The weights ω_j of all particles are initialized with the same value at the first initialization, and are normalized after each update.

The particle importance resampling is performed to provide a faster PF convergence and account for possible errors in particle removal. When the new weights at time *k* are computed, a total of *M* particles are removed from the filter with a random probability that is inversely proportional to their weights (lower weight implies a higher removal probability). Each removed particle is substituted by a particle copy of a highly probable particle (higher weight implies a higher probability of being copied). The particles copied are moved randomly a few centimeters to better analyze the area surrounding highly probable particles. In our case, 10% of the total particles are removed at each resampling.

PFs enable multi-modal estimation but require a significantly higher computational burden than uni-modal estimation tools such as EKFs. The PF convergence to a uni-modal solution is periodically checked. If σ^{τ}, the square root of the maximum eigenvalue of Σ^{τ}, is below a certain value (we used $3\sigma^{\tau}$ = 3 m), the PF is considered as converged. Then, the estimated location of the tool (μ^{τ}) is given as the weighted mean of all particles. The covariance matrix of the PF particles is used, alongside μ^{τ}, to initialize the EKF of the next step. Finally, the PF is terminated.

4.5.3. EKF-based Refinement

The EKF represents the tool position by a Gaussian with mean μ^{τ} and covariance Σ^{τ}. It relies on the assumption that the tool is static, so: $\hat{\mu}^{\tau} = \mu_{k-1}^{\tau}$, i.e. the predicted position of the tool is the same as in the previous instant. This is used as the prediction model adopted for the EKF. The observation model is the Euclidean distance between the robot position and the tool position, such that: $\hat{z}_k = d(\mu^{\tau}, X_k^r)$. Given the prediction and observation models, and the initial values of μ^{τ} and Σ^{τ}, the filter can be easily implemented as in [38]. This EKF was able to accurately localize the missing tools in the factory, without needing any complex modeling of signal reflections, as can be seen in the experiments in Section 6. Moreover, the adopted PF-EKF scheme has very low computational consumption and can be executed in real-time on board the robot together with the rest of the modules. For example, 4 PFs (e.g. localizing 4 tools using PFs) running in parallel only use 10% of one core in the UP-board processor. Also, running 4 EKFs use <1% of one core, involving a computational cost saving of >90% w.r.t. PFs, which justifies the adopted PF-EKF approach.

5. From the Lab to the Factory

5.1. Adaptation of the Factory

Figure 10 summarizes the deployment of the proposed system in the CBC Airbus D&S factory. Following the safety analysis described in Section 3.2, the first decision was to determine the robot flight zone, which had to be at least 5 m above the ground to ensure enough physical separation with human workers. Imposed by the factory Industrial Safety and Health Department as an unavoidable requirement, a safety net was designed and installed below the flying level to prevent physical contact with workers or manufacturing elements.

The safety net has a U-shaped design and surrounded the flight zone except for the upper part, to enable the safety pilot to react and recover the robot if necessary. Also, in the light object logistics use case, hoppers were designed to connect the flying zone to a tray at human reach where objects are delivered. Two hoppers were installed in the factory at delivery points *A* and *B*. The location for the robot's take-off and landing was chosen in such a way that it would not pose major worker mobility constraints in the factory working area. It was placed in free space within the logistics and storage center. The aerial robot takes off and lands from the ground, hence a safety vertical corridor –surrounded by the safety net– was installed from the ground to the flying altitude. To avoid constraining the mobility of workers while not in use, the bottom part of the netting is free to be rolled up.

5.2. Safety and Robustness Validation

Implementing an autonomous aerial robot in a factory in full working conditions requires careful safety and robustness analyses and validation procedures. A dual complementary approach was adopted. On one hand, we established safety analyses and procedures by collecting all possible risks and defining mitigation measures. On the other hand, we adopted an incremental robustness validation approach with increasing levels of presence of workers and complexity.

Table 3 summarizes the main risks and proposed mitigation measures adopted. Most risks were addressed with several complementary passive and active mitigation measures. The safety net is the main mitigation means for many of the risks. The modular and robust design of the system functionalities is very relevant against software failures. Installing the safety net reduced the available flying area to the space covered by the net. This restriction suggested the adoption of a navigation approach based on predefined waypoints, which simplifies robot autonomous navigation and enhances robustness. We also implemented modules that supervise that hardware and software components are operating under nominal parameters and that safely finish the mission in case of unexpected behaviors.

Table 3
Main risks identified and implemented passive and active mitigation measures.

Risk	Passive Mitigation	Active Mitigation
Aerial robot hardware failure	Safety net	Pre-flight and in-flight hardware status supervision modules
Software failure	Modular software design; Safety net	Pre-flight and in-flight software status supervision modules
Robot localization / navigation error	Safety net	6-DoF pose estimation supervision
Physical contact with human workers	Safety net; Robot flights above 7 m	Reactive obstacle avoidance
Presence of unexpected obstacles	Safety net	Reactive obstacle avoidance
Inaccurate dropping of light objects	Hopper to increase the dropping area	Accurate robot positioning modules
Inaccurate missing tool localization		UWB outlier filter; Tool localization covariance monitoring

The 6DoF localization system was extensively tested in a mock-up scenario installed at the indoor flying arena of FADA-CATEC, as it can be seen at https://youtu.be/O9pD1dPnTn8. Missing tool search modules were tested directly at the CBC factory. The final stage was to test the fully autonomous system in a factory that could not stop production for such experiments. This stage was divided into four steps with an incremental presence of workers:

1. **Experiments out of regular working time (>20:00 h)**. To reduce risks, there were very few workers in the factory. This step required around 3 months of experiments with one workday in the factory every one or two weeks. Experiments in the factory were interleaved with method refinement in the lab and indoor flying arena with the collected data.
2. **Experiments in the afternoon work shift without involving workers**. Workers got used to the aerial robot flying in the factory but still did not interact with it. This step required 2 months of experiments with one or two workdays in the factory every week. All refinements were tested directly in the factory.
3. **Experiments in the afternoon work shift involving workers.** The workers started interacting with the robot and asked for a specific small object using the light object logistics system. This step required around 1 month of experiments with one or two workdays in the factory every week.
4. **Experiments in the morning work shift in fully working conditions**. The factory was in full working conditions, with tens of workers around and some of them interacting with the system. The system was safe and robust. This step required around 4 weeks of experiments with between two and four workdays in the factory every week.

6. Validation in the Factory

This section summarizes the validation experiments of the robotic system operating fully autonomously in the CBC Airbus D&S factory (Cádiz, Spain), which were performed during the final months of 2017 and the beginning of 2018. In these experiments, the robot traversed a total distance of ~19 km, with a total flight time of ~27 hours. The description of the partial module and functionality experiments, mainly performed in the laboratory and the FADA-CATEC flying arena, has been omitted for brevity. A video of the experiments is available at https://youtu.be/tmxPpsGN5ts.

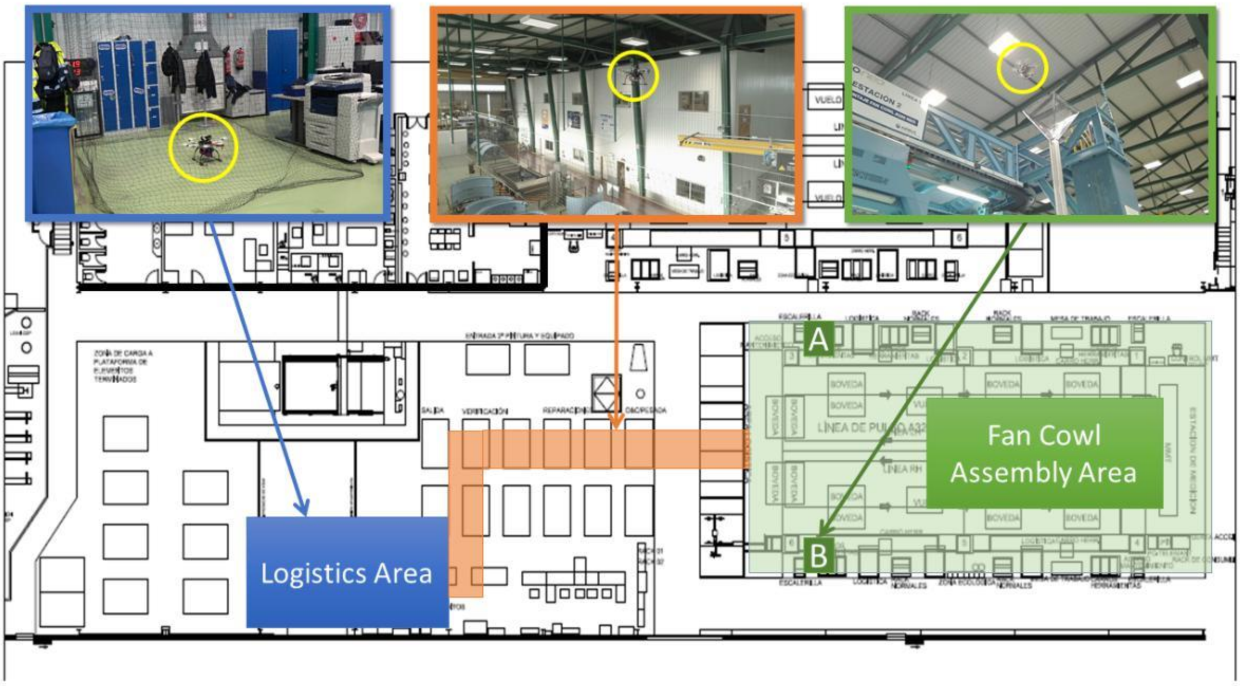

Figure 10: Layout of the CBC Airbus D&S factory showing the storage center at the logistics area with the take-off/landing location (in blue), the flying corridor (in orange) towards the fan cowl assembly working area (in light green), which includes hoppers A and B for light object delivery (in dark green).

6.1. Light Object Logistics

First, an experiment of the light object logistics use case mission is presented. It starts when a component request reaches the logistic center from a worker located near one of the delivery points at the *Fan Cowl Assembly Area* (delivery point *A* in this experiment). Once the component is placed in the robot's cargo hold, the mission starts. The robot takes off and performs a pure ascent up to waypoint *WP1* (see Figure 7), located at an altitude of 7.5 m. Next, it navigates to the following waypoints *WP2* (see Figure 11-a) and *WP3* through the horizontal corridor, until it reaches *WP4* at the *Fan Cowl Assembly Area* (Figure 11-b). Then, it navigates to delivery point *A*, which is defined by waypoint *WP5*. Approximately 100 s after taking off, the robot reaches *WP5* with the required accuracy during at least 5 s. At that time, the robot is ready for cargo delivery (it is represented in Figure 11-c by displaying the delivery waypoint marker with green color). At that time, the cargo hold is opened: the capsule with the requested component is dropped through the hopper. After delivery, the aerial robot starts navigating back following the same waypoints in reverse order, see Figure 11-d. Meanwhile, the worker retrieves the component from the capsule and acknowledges reception. The aerial robot finally lands at the logistics area after having traversed more than 70 m. The whole flight takes an average of 200 s.

More than 150 light object logistics missions were performed in the factory in full working conditions. Unfortunately, there was no infrastructure to record any ground truth in the factory. For validation assessment, we considered that: robot navigation in each experiment is successful if it is capable of safely reaching all the mission waypoints; object delivery is successful if the robot releases the cargo accurately in the delivery hopper; and the full mission is considered successful if the robot completes the mission autonomously without any external assistance fulfilling the delivery time requirements, from take-off to landing. Table 4 shows the obtained results considering all the experiments performed in the factory.

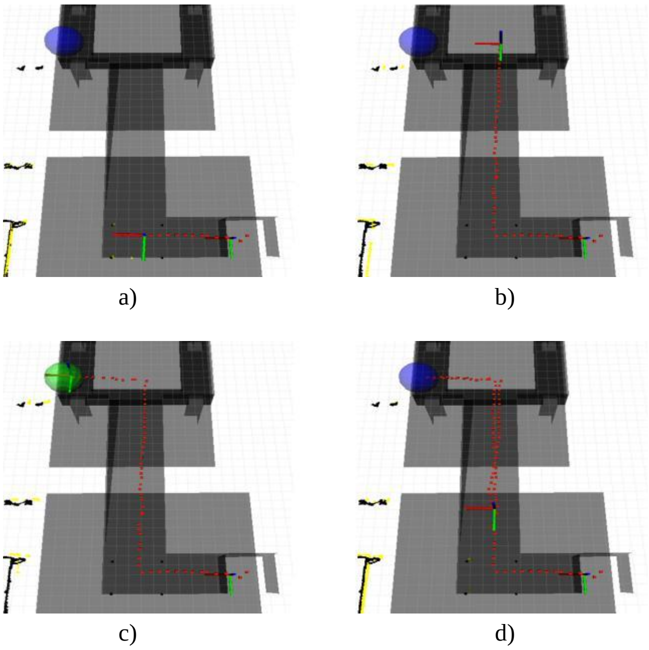
a) b) c) d)

Figure 11: Light object delivery experiment: a) the robot takes off and navigates through the vertical and horizontal ducts following waypoints WP1 and WP2; b) the robot continues to waypoints WP3 and WP4; c) the robot reaches WP5 (location of delivery point *A*) and opens the cargo hold to release the requested component, and; d) the robot returns following the same path towards the logistics area.

The navigation success rate was 96%. The 6DoF pose estimation in the factory was accurate enough to allow the aerial robot to traverse a distance of around 70 m per mission with a positioning error of 0.4 m. The system performed as expected except in only a low percentage of the cases in which the flight was interrupted due to pose estimation errors.

Table 4
Success ratio of the light object logistics validation experiments in the CBC Airbus D&S factory.

	Hopper A	Hopper B	Total
Navigation success rate	95.9%	96.2%	96.1%
Object delivery success rate	97.3%	97.4%	97.4%
Mission success rate	95.9%	96.2%	96.1%
Object delivery Mean time	97 s	99 s	98 s
Object delivery Max time	100 s	104 s	104 s
Mission Mean time	196 s	197 s	196 s
Mission Max time	201 s	206 s	206 s

The light objects were correctly delivered in 97.4% of the tests. On some occasions, the navigation system failed while the robot returned to the logistics area after accomplishing object delivery: hence, a navigation failure does not necessarily imply a delivery failure. It should be taken into account that the results shown in Table 4 correspond to all the experiments performed in the factory. The navigation failures occurred during Stage 1 and Stage 2 of the implementation in the factory, according to Section 5.2. In Stage 3, after adjusting in the factory, no navigation failure was registered, obtaining success rates of virtually 100% in the >65 missions that were performed in Stage 3. These adjustments mainly applied to the 6DoF pose estimation modules, whose parameters were refined during the first experiments in the factory. Particularly important was the threshold for discontinuity detection in the altitude estimation (see Section 4.3.2), which needed to be increased according to the higher factory size compared to the laboratory testbed. Also, we increased the position and orientation thresholds for detecting failures in the local pose estimator (see Section 4.3.3), as well as the point cloud downsampling configuration for the map, due to the higher distances observed at the factory.

Table 4 also includes the object delivery and mission accomplishment times; for both of them, the average and the worst-case times are shown. The delivery time, i.e. the time between the worker request and the delivery, is useful to assess the productivity improvement when compared to current manual-based procedures in the factory. The average delivery time was ~98 s with a maximum (worst-case) time of 104 s. Both times largely fulfilled the hard (more restrictive) time requirement for this mission (**RL2** in Table 1).

6.2. Missing Tool Localization

The tool localization system was validated by testing different places for hiding tools and analyzing the system performance in finding them. Several tools were searched for in every experiment. Below, one missing tool localization experiment is described, which was executed while the robot navigated performing light object logistics tasks.

Table 5
Success ratio of the missing tool localization system.

Tool placement	Tables		Ground		Fan cowls/Structures		Under fan cowl		Total
Outlier Filter	**No**	**Yes**	**No**	**Yes**	**No**	**Yes**	**No**	**Yes**	**Yes**
Found with $3\sigma^{\tau} < 3m$	97.1%	99.0%	96.3%	98.1%	96.3%	97.1%	71.8%	88.4%	97.5%
Found with $3\sigma^{\tau} < 0.5m$	93.3%	98.1%	70.4%	97.6%	58.5%	95.2%	31.6%	56.9%	94.1%
PF convergence Mean time	61 s	54 s	60 s	53 s	78 s	65 s	105 s	89 s	61 s
PF convergence Max time	69 s	60 s	72 s	59 s	96 s	82 s	125 s	113 s	72 s
Tool search Mean time	97 s	81 s	101 s	84 s	121 s	98 s	160 s	121 s	91 s
Tool search Max time	102 s	92 s	106 s	89 s	129 s	115 s	164 s	138 s	103 s

First, an intelligent tool cabinet reports three tools as missing: their IDs are given to the missing tool search module. The PF of each missing tool is initialized after the reception of the first range measurement from its UWB, which occurs when the robot gets closer to them. The PF particles start evolving as new range measurements are integrated. Figure 12-a shows that two PFs have been initialized, and the robot trajectory up to that time is shown in red. The third PF is initialized between Figure 12-a and Figure 12-b. Later, when the robot reaches *WP*4, the value $3\sigma^{\tau}$ of one PF becomes lower than 3 m, hence it transitions to an EKF using its current mean and covariance matrix, see Figure 12-b. At *t*=70 s, the other two PFs also converge to EKFs, and the first EKF already has a $3\sigma^{\tau}$ lower than 0.5 m, see Figure 12-c. When the robot returns to the logistics area, the delivery mission finishes and all the missing tools have been localized. The EKFs of the three tools have converged to a $3\sigma^{\tau}$ lower than 0.5 m, and their location and uncertainty regions are shown in the factory map so that workers can easily find them, see Figure 12-d. The resulting missing tool localization errors (Euclidean distance between the estimated and actual tool positions) in this specific mission were respectively 0.10 m, 0.05 m, and 0.23 m.

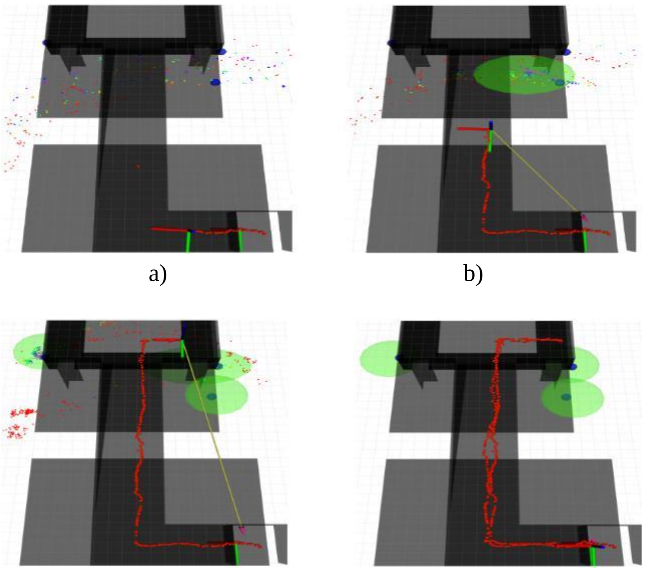
a)
b)
c)
d)

Figure 12: Missing tool search experiment while the robot performs light object logistics tasks. The tool ground-truth locations are depicted as blue spheres. The PFs particles are depicted as colored dots, scaled from blue (higher particle weights) to red (lower weights). The uncertainty regions with radii lower than 3 m are shown as green ellipsoids.

More than 200 validation experiments of missing tool searches were conducted, most of them while also testing light object logistics missions. The missing tools were randomly placed in the *Fan Cowl Assembly Area* at different spots where it was probable to forget a tool: on the floor, over the working tables, and near the fan cowls or the factory structures. Table 5 summarizes the results obtained. The table also accounts for the missions in which the robot navigation system failed. Notice that a navigation failure does not always preclude successful missing tool search accomplishment. In many cases, navigation failure due to high localization errors did not affect significantly the missing tool search and, in others, the tools were found before the mission was interrupted due to the navigation failure. The overall success rate with 3 m accuracy was 97.5%, and with 0.5 m accuracy was 94.1%, meaning that almost all objects were successfully found.

Table 5 shows the percentage of times that the tools were correctly localized inside the $3\sigma^{\tau}$ <3 m confidence area and the $3\sigma^{\tau}$ <0.5 m area for the different spots, with and without the outlier rejection filter. The table also shows the average and worst-case times for PF convergence, and for accomplishing missing tool search missions (tool correctly found inside the $3\sigma^{\tau}$ <0.5 m area) . Although these times depend on the spot where the missing tools were located, both tool search times were lower than the duration of light object logistics missions and also lower than the hard (more restrictive) time requirement for this mission (**RT1** in Table 1). The worst performance was obtained with missing tools located under the fan cowls: they become shielded by the metallic structure, hampering UWB reception, and increasing the outlier measurement ~~outlier~~ levels.

Table 5 summarizes all the experiments performed in the factory. After adjusting the navigation system in the factory in Stage 3, no navigation failure was registered, obtaining missing tool success rates of virtually 100% except for tools under fan cowls. Tools near metallic structures caused outliers that were alleviated by integrating more measurements in the PF-EKF scheme and using the outlier filter. The method had similar performance for flights while performing logistics tasks and for on-request missing tool missions.

7. Lessons Learnt

The safety of the personnel and the factory elements is crucial. Robot fall damages can be overcome by adding active security measurements (such as the reactive obstacle avoidance in Section 4.4) or passive security measures (such as propeller guards). During all validation stages at the factory, the aerial robot fell on the safety net only on one occasion; this took place during Stage 1 one of the first days of testing the 6DoF pose estimation modules. The safety net is a cost-effective installation that successfully eliminates fall damages and does not alter the functioning of the factory.

It is also of high interest to analyze the reaction of the workers to the presence of aerial robots. During the first days, workers were interested in the system and its validation, but after several successful experiments, they tend to ignore the robot and continue working without distraction. The noise generated by the robots' propellers was one of our concerns. However, it was neglected by the noise emitted from other manufacturing machines. Moreover, the workers usually wear noise-canceling headphones during their full shift. The interaction of the workers with the robot was satisfactory and required almost no instructions. To achieve that, it was important to design a graphical user interface (GUI) that allowed the workers to perceive the robot as another tool, and not as a complicated system with undefined functionalities.

The upper parts of factories are suitable for aerial robot operations with almost no risk of collision. One particular issue to account for is that aerial robots can raise dust when flying. The take-off/landing locations (in the logistics area) should be clear of dust, but there is no need of eliminating the dust in all the areas since robots fly several meters above the ground. The air vents located at the ceiling can perturb the robot flight control and can alter the measurements of barometer-based altitude sensors. The first issue was solved by properly tuning the robot controller, and the second, by using the 2D LiDAR to measure altitude.

CBC factory exhibits a slanted roof with beams and lamps, but the same altitude estimation module could be reused in other factory environments (e.g. saw-tooth roof) since it already takes into account slants and discontinuities; nevertheless, specific parameter tuning and testing steps are recommended before implementing our system in other factory environments.

UWB technology has proven to be a very cost-effective solution for missing tool search. Even though it is necessary to have outlier filtering mechanisms, the advancements in low power consumption and accuracy make UWB a very suitable technology.

Although simulations and experiments with mock-ups are very good solutions for initial development, it has been crucial to gather as much real data as possible. These datasets allow for fine-tuning without conducting specific experiments and enable parallel development with copies of the robot's software.

The robot motors, ESCs and propellers were chosen for allowing a flight autonomy of 10-12 minutes with the total estimated platform weight, which we considered adequate for the validated use cases. The energy consumption of all the sensors, computers and electronics on board the robot was 16 W. The current prototype is powered by a single battery, which implies a full system restart whenever a battery change is needed. Redesign efforts are currently taking place in order to improve the acceptance of the system. First, separate batteries for electronics and rotors will allow subsequent flights without powering off the onboard computer and sensors. Second, a system for automated battery change will minimize the need for system supervision by factory personnel.

8. Conclusions

This paper presents a fully autonomous aerial robotic system performing complex tasks in manufacturing industries. The system was designed, developed, and implemented with two fundamental requirements: safety to workers and robustness for everyday operation. It includes modules for accurate GNSS-denied self-localization, autonomous navigation, radio-based localization, and obstacle avoidance, among others, providing a fully onboard solution where all involved sensors, electronics, and processing are embarked on the robot, enabling implementation flexibility and scalability.

Two main use cases are considered: transportation and delivery of light objects in the factory and localization of missing tools, which are of interest in –and could be easily extended to– a wide variety of manufacturing industries. The system validation with TRL-7 in the Centro Bahía de Cádiz (CBC) Airbus D&S factory in fully working conditions provided satisfactory results that fulfilled all the system and use case functional requirements. To the best of our knowledge, it is one of the first times a fully autonomous aerial robotic system performing complex tasks has been validated in an indoor manufacturing plant in fully working conditions.

We believe that the presented work represents a good example of the adoption of robots in industrial applications that will open indoor manufacturing applications for aerial robotics. Wide lines for future research, development, and innovation are opened after this work. The use of other robotic platforms more suitable than multi-rotors to densely populated environments is a natural extension. Soft aerial robots or coaxial platforms with a protective structure around the blades are promising, but in the current state of technology, their limited payload capacity is a critical issue that constrains their feasibility. Although the safety net does not perturb the system missions or the factory operation, removing it is very interesting for opening the solution to a wider range of applications and scenarios. In our case, the safety net was imposed as an unavoidable requirement by Industrial Safety and Health Department at Airbus D&S. Also, the full integration of the proposed system within the factory manufacturing procedures and the extension to other use cases or other manufacturing processes are object of current research.

Acknowledgements

This work has been performed within the context of EuRoC-ARCOW project funded by the European Commission under contract 608849. Partial funding has been also obtained from the EU-funded H2020 AERIAL-CORE project (contract 871479), and also the ARM-EXTEND (DPI2017-8979-R) project funded by the Spanish National R&D Plan.

References

[1] Fratini, L., Ragaiand, I., & Wang, L. (2020). New trends in manufacturing systems research 2020. Journal of manufacturing systems.

[2] Esmaeilian, B., Behdad, S., & Wang, B. (2016). The evolution and future of manufacturing: A review. Journal of Manufacturing Systems, 39, 79-100.

[3] Lu, Y., Xu, X., & Wang, L. (2020). Smart manufacturing process and system automation–a critical review of the

standards and envisioned scenarios. Journal of Manufacturing Systems, 56, 312-325.

[4] Bonnin-Pascual, F., Ortiz, A., Garcia-Fidalgo, E., & Company-Corcoles, J. P. (2019). A reconfigurable framework to turn a MAV into an effective tool for vessel inspection. Robotics and Computer-Integrated Manufacturing, 56, 191-211.

[5] Wawrla, L., Maghazei, O., & Netland, T. (2019). Applications of drones in warehouse operations. Whitepaper. ETH Zurich, D-MTEC.

[6] EYIF, EU Drone Awards (2017), https://younginnovator.eu/eyif/industry-year-4-0/eu-drone-awards-2017, accessed: 2021-02-11.

[7] Forster, C., Zhang, Z., Gassner, M., Werlberger, M., & Scaramuzza, D. (2016). SVO: Semidirect visual odometry for monocular and multicamera systems. IEEE Transactions on Robotics, 33(2), 249-265.

[8] Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics, 34(4), 1004-1020.

[9] Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2015). Keyframe-based visual-inertial odometry using nonlinear optimization. The International Journal of Robotics Research, 34(3), 314-334.

[10] Pire, T., Fischer, T., Castro, G., De Cristóforis, P., Civera, J., & Berlles, J. J. (2017). S-ptam: Stereo parallel tracking and mapping. Robotics and Autonomous Systems, 93, 27-42.

[11] Mur-Artal, R., & Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics, 33(5), 1255-1262.

[12] Labbé, M., & Michaud, F. (2019). RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. Journal of Field Robotics, 36(2), 416-446.

[13] Wu, X., Sun, C., Zou, T., Li, L., Wang, L., & Liu, H. (2020). SVM-based image partitioning for vision recognition of AGV guide paths under complex illumination conditions. Robotics and Computer-Integrated Manufacturing, 61, 101856.

[14] Kwon, W., Park, J. H., Lee, M., Her, J., Kim, S. H., & Seo, J. W. (2019). Robust autonomous navigation of unmanned aerial vehicles (UAVs) for warehouses' inventory application. IEEE Robotics and Automation Letters, 5(1), 243-249.

[15] Zhong, Y., Wang, Z., Yalamanchili, A. V., Yadav, A., Srivatsa, B. R., Saripalli, S., & Bukkapatnam, S. T. (2020). Image-based flight control of unmanned aerial vehicles (UAVs) for material handling in custom manufacturing. Journal of Manufacturing Systems, 56, 615-621.

[16] Torres-González, A., Martinez-de Dios, J. R., & Ollero, A. (2018). Range-only SLAM for robot-sensor network cooperation. Autonomous Robots, 42(3), 649-663.

[17] Yang, J., Li, Y., Cao, L., Jiang, Y., Sun, L., & Xie, Q. (2019). A Survey of SLAM Research based on LiDAR Sensors. Int J Sens.; 1 (1), 1003.

[18] Hornung, A., Oßwald, S., Maier, D., & Bennewitz, M. (2014). Monte Carlo localization for humanoid robot navigation in complex indoor environments. International Journal of Humanoid Robotics, 11(02), 1441002.

[19] Perez-Grau, F. J., Caballero, F., Merino, L., & Viguria, A. (2017). Multi-modal mapping and localization of unmanned aerial robots based on ultra-wideband and RGB-D sensing. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 3495-3502). IEEE.

[20] Paneque, J. L., Martínez-de Dios, J. R., & Ollero, A. (2019). Multi-sensor 6-DoF localization for aerial robots in complex GNSS-denied environments. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1978-1984). IEEE.

[21] Pomerleau, F., Colas, F., Siegwart, R. & Magnenat, S. (2013). Comparing ICP variants on real-world data sets. Autonomous Robots, 34(3), 133-148.

[22] Kümmerle, R., Ruhnke, M., Steder, B., Stachniss, C., & Burgard, W. (2015). Autonomous robot navigation in highly populated pedestrian zones. Journal of Field Robotics, 32(4), 565-589.

[23] Noreen, I., Khan, A., & Habib, Z. (2016). Optimal path planning using RRT* based approaches: a survey and future directions. Int. J. Adv. Comput. Sci. Appl, 7(11), 97-107.

[24] Zammit, C., & Van Kampen, E. J. (2018). Comparison between A* and RRT algorithms for UAV path planning. In 2018 AIAA guidance, navigation, and control conference (p. 1846).

[25] Oleynikova, H., Burri, M., Taylor, Z., Nieto, J., Siegwart, R., & Galceran, E. (2016). Continuous-time trajectory optimization for online UAV replanning. In 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 5332-5339). IEEE.

[26] Nash, A., Koenig, S., & Tovey, C. (2010). Lazy Theta*: Any-angle path planning and path length analysis in 3D. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 24, No. 1).

[27] Perez-Grau, F. J., Ragel, R., Caballero, F., Viguria, A., & Ollero, A. (2018). An architecture for robust UAV navigation in GPS-denied areas. Journal of Field Robotics, 35(1), 121-145.

[28] Liu, H., & Wang, L. (2021). Collision-free human-robot collaboration based on context awareness. Robotics and Computer-Integrated Manufacturing, 67, 101997.

[29] Trujillo, M. Á., Martínez-de Dios, J. R., Martín, C., Viguria, A., & Ollero, A. (2019). Novel aerial manipulator for accurate and robust industrial ndt contact inspection: A new tool for the oil and gas inspection industry. Sensors, 19(6), 1305.

[30] Maghazei, O., & Netland, T. (2019). Drones in manufacturing: exploring opportunities for research and practice. Journal of Manufacturing Technology Management.

[31] Hardis-Group, Eyesee, the inventory drone solution (2019), https://eyesee-drone.com/, accessed: 2021-02-11.

[32] Keonn, Robin: Rfid robot for inventory and location (2020), https://keonn.com/systems-product/robin/, accessed: 2021-02-11.

[33] Fetch-Robotics, Tagsurveyor (2020), https://fetchrobotics.com/products-technology/datasurvey/tagsurveyor/, accessed: 2021-02-11.

[34] Macoir, N., Bauwens, J., Jooris, B., Van Herbruggen, B., Rossey, J., Hoebeke, J., & De Poorter, E. (2019). Uwb localization with battery-powered wireless backbone for drone-based inventory management. Sensors, 19(3), 467.

[35] Fernández-Caramés, T. M., Blanco-Novoa, O., Froiz-Míguez, I., & Fraga-Lamas, P. (2019). Towards an autonomous industry 4.0 warehouse: A UAV and blockchain-based system for inventory and traceability applications in big data-driven supply chain management. Sensors, 19(10), 2394.

[36] Beul, M., Droeschel, D., Nieuwenhuisen, M., Quenzel, J., Houben, S., & Behnke, S. (2018). Fast autonomous flight in warehouses for inventory applications. IEEE Robotics and Automation Letters, 3(4), 3121-3128.

[37] Ma, Y., Selby, N., & Adib, F. (2017, August). Drone relays for battery-free networks. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication (pp. 335-347).

[38] W. Systems (2020), Autonomous tool tracking with uwb + rfid, https://www.wisersystems.com/applications/aerospace-tool-tracking, accessed: 2021-02-11.

[39] De Ryck, M., Versteyhe, M., & Debrouwere, F. (2020). Automated guided vehicle systems, state-of-the-art control algorithms and techniques. Journal of Manufacturing Systems, 54, 152-173.

[40] Meier, L., Tanskanen, P., Fraundorfer, F., & Pollefeys, M. (2011, May). Pixhawk: A system for autonomous flight using onboard computer vision. In 2011 IEEE International Conference on Robotics and Automation (pp. 2992-2997). IEEE.

[41] Meier, L., Honegger, D., & Pollefeys, M. (2015, May). PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In 2015 IEEE international conference on robotics and automation (ICRA) (pp. 6235-6240). IEEE.

[42] MAVLink (2021), Micro Air Vehicle communication protocol, https://mavlink.io/en/, accessed: 2021-02-11.

[43] Quigley, M., Gerkey, B., & Smart, W. D. (2015). Programming Robots with ROS: a practical introduction to the Robot Operating System. " O'Reilly Media, Inc.".

[44] Grzonka, S., Grisetti, G., & Burgard, W. (2009, May). Towards a navigation system for autonomous indoor flying. In 2009 IEEE International Conference on Robotics and Automation (pp. 2878-2883). IEEE.

[45] Pomerleau, F., Colas, F., & Siegwart, R. (2015). A review of point cloud registration algorithms for mobile robotics. Foundations and Trends in Robotics, 4(1), 1-104.

[46] Thrun, S. (2002). Probabilistic robotics. Communications of the ACM, 45(3), 52-57.