

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Visual-Inertial Indoor Navigation Systems and Algorithms for UAV Inspection Vehicles

*Lorenzo Galtarossa, Luca Francesco Navilli
and Marcello Chiaberge*

Abstract

In UAV navigation, one of the challenges in which considerable efforts are being focused is to be able to move indoors. Completing this challenge would imply being able to respond to a series of industrial market needs such as the inspection of internal environments for safety purpose or the inventory of stored material. Usually GPS is used for navigation, but in a closed or underground environment, its signal is almost never available. As a consequence, to achieve the goal and ensure that the UAV is able to accurately estimate its position and orientation without the usage of GPS, an alternative navigation system based on visual-inertial algorithms and the SLAM will be proposed using data fusion techniques. In addition to the navigation system, we propose an obstacle avoidance method based on a Lidar sensor that allows navigation even in the absence of light.

Keywords: unmanned aerial vehicle (UAV), GPS denied, indoor navigation, Lidar, inertial measurement unit (IMU), visual-inertial odometry (VIO), 3D reconstruction

1. Introduction

Nowadays indoor navigation is one of the most popular topics in the field of scientific research; this is because it is itself a very interesting challenge from the scientific point of view and with a potentially huge impact on the current market with multiple applications ranging from the industrial sector to the relief sector in emergency situations.

As well known, while in outdoor navigation, UAVs use Global Position System (GPS) signal to easily understand, with good accuracy, their position in space, in indoor navigation the possibility to use this technology for positioning and localization decays [1–3]. In fact, the Global Navigation Satellite System (GNSS) in indoor environments is almost blocked or made too weak by buildings, walls or several potential sources of interference.

The first approaches to indoor UAV navigation used technologies such as Wi-Fi, Bluetooth or Ultra-Wide Band [4]. The disadvantage and the great limitation of these technologies is that they need to structure or prepare the environment in order to be able to navigate inside with the UAVs.

In recent years, the miniaturization of both visual sensors and computers that guarantee good computing power and at the same time less weight has made possible a new different approach to the topic of UAV indoor navigation.

This approach is based on inertial and visual systems, for example, see [5–7], with enormous advantage of being free from any kind of need to structure the environment and therefore potentially flexible and universal. The position of the UAV is estimated using inertial measurements provided by gyroscopes and accelerometers that are now available in every smartphone and with small dimensions and weights. The accuracy of this type of inertial measurements is good but not sufficient in order to guarantee a precise indoor positioning. In fact the estimate of the UAV position based only on inertial systems tends to diverge and drift over time due to the fact that inertial measurement unit (IMU) measurements are corrupted by noise and bias with the results in pose estimates unreliable for long-term navigation. To avoid the effects of this phenomenon, the inertial system is combined with a visual one that uses a camera to collect information and extract features from the surrounding environment and track them over time in order to estimate the trajectory of the camera. This approach is usually referred to as visual-inertial odometry (VIO). Information from the camera can also be used to build a map of the environment and then perform what is called simultaneous localization and mapping (SLAM).

In this chapter we propose a system architecture that allows a UAV to inspect a tunnel, which is a closed environment, navigating autonomously. To estimate the position of the UAV in the absence of GPS, we used Robust Visual Inertial Odometry (ROVIO) which is a predictor of inertial visual states based on an extended Kalman filter (EKF) that combines the visual information of a monocular camera with the measurements derived from the IMU inertial platform. At the same time, a navigation and obstacle avoidance algorithm based purely on a Lidar sensor is proposed.

The UAV is equipped with a companion computer in which Robotic Operating System (ROS) is installed and allows the processing of information coming from the monocular camera and the IMU as well as those coming from the Lidar for the navigation.

Furthermore, a scheduling system has been implemented and embedded on the computer companion that allows to set different strategies to approach the inspection of the tunnel before starting the mission. Defined safety patterns that are activated in case of dangerous situations for UAV and humans are also into the scheduling system.

The chapter is organized as follows. In Section 2, we briefly analyze and describe the sensors used and their characteristics, and we go into detail on how the architecture of the system is defined. In Section 3, we explain which criteria characterize the navigation system and what is the logic behind it. In conclusion we present the results achieved, outlining the performance of the proposed system for indoor navigation evaluating possible improvements for future research.

2. System architecture

This chapter describes the overall system architecture under different points of view. We start with a short description of ROS, that is, the framework that allows to manage different UAV's operation. Then we move to analyze the hardware and payload of the UAV, we describe all the crucial characteristics and we explain why those characteristics and components are crucial for the project. Afterwards we explain why between the several VIO algorithms implemented in

the past years, we select and use ROVIO and how we design the visual-inertial sensor. Moreover, we propose a scheduling system based on some setting parameters that are crucial to well set up at the beginning of the mission in order to define the positioning of the UAV inside the tunnel. These parameters allow to define both inspection and navigation settings, the last ones are useful to change based on the geometrical characteristic of the tunnel. Instead, the parameters define the type of inspection that is needed to be performed in order to collect data or achieve the inspection's objectives.

2.1 Robot operating system

The heart of the whole system is robot operating system (ROS); it is an open source framework to manage robot's operations, tasks and motions. Among the several features that ROS has, the most relevant is the availability of code, packages and open source projects. This is a key element in the development of complex systems which often encompass different skills and concepts [8, 9].

A set of processes can be represented in a graph as a node that can receive, send and process messages, called topics, coming from other sensors, actuators and nodes.

In this system the two main topics for the construction of the algorithm are those of the Lidar and the odometry that give to the system the information about the obstacles around the drone (coming from the Lidar) and the pose outgoing from ROVIO which defines the position and the orientations of the UAV along the six DOF.

The information on these two messages is fed to the navigation algorithm which returns the topic of the speed to be assigned to the drone during the inspection.

2.2 UAV's payload

Referring to **Figures 1** and **2**, the main components of the UAV are:

1. Custom frame with a propulsion system designed for a total payload of about 4 kg
2. LED lighting system for navigation and acquisition of frames even in complete darkness and absence of light
3. Cameras for the acquisition of photograms that allow the construction of a three-dimensional model of the inspected tunnel and environment
4. Visual-inertial sensor used for positioning, control and as the main source of odometry
5. Laser sensors for detecting distance from the ground
6. Voltage and current distribution system, mainly 12 and 5 V
7. LiDAR 2D laser scanner for detecting obstacles and relative distances

As a payload there is also a mini computer companion that has the necessary power to perform, record, process and analyze data from all the sensors and to move the UAV accordingly.

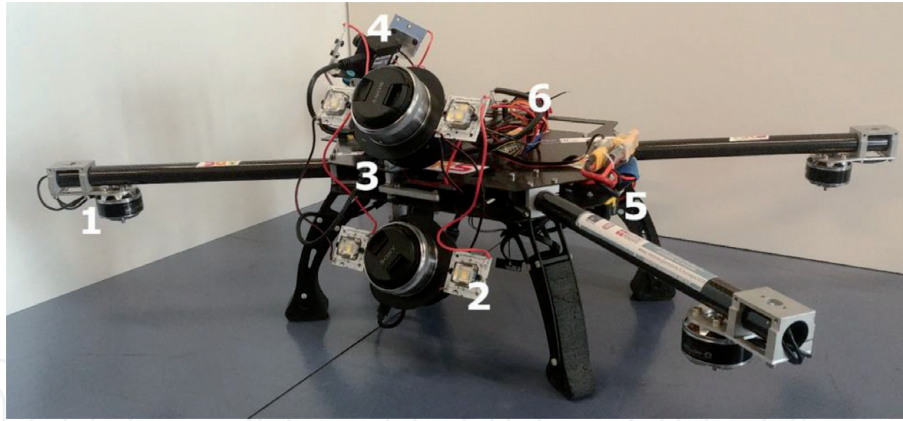


Figure 1.
Assembly UAV payload, first perspective.

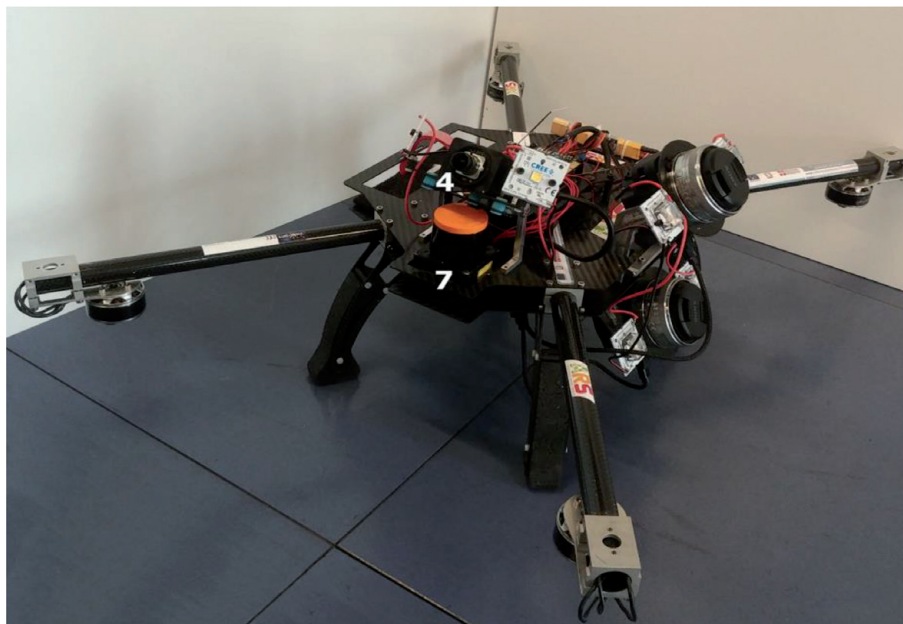


Figure 2.
Assembly UAV payload, second perspective.

2.3 Visual-inertial sensor

As mentioned in the introduction, the lack of a GPS requires the use of a sensor that can guarantee the correct positioning inside a closed space. In particular, follow the research trend in the field of computer vision; the sensor is composed by a monocular camera and an IMU inertial measurement sensor. These two sensors are connected to each other by a mechanism called hardware trigger. This choice was made to ensure maximum precision in the acquisition of data from both sensors since it is a crucial point in order to obtain a precise positioning of the UAV. The kind of sensors described above is preferable to purely visual-based techniques or any other sensor configurations for large number of advantages:

- Unlike monocular simultaneous localization and mapping (SLAM) based only on visual sensor, the generated maps have an absolute scale.
- Status estimation and feature tracking, which allow to understand how the UAV is moving in the space, are more robust to the motion blur and fast rotations than exclusive visual-based system.

- IMU data can be used to provide instantaneous estimates at over 100 Hz.
- The installation of this hardware is cheaper, smaller, lighter and lower in power consumption with respect to the three-dimensional laser scanner or even stereo configurations.

However, this type of approach has two main problems: the first one is related to the IMU camera timestamp synchronization that can cause large errors and drift in the state estimate of the UAV. The second one is that the system has to be able to continuously estimate and compensate the drift and the distortions of the IMU data. These problems are mainly related to the VIO algorithm chosen for this project, ROVIO [10]. ROVIO is a visual-inertial state estimator based on EKF which proposed several novelties. In addition to FAST corner features, whose 3D positions are parameterized with robot-centric bearing vectors and distances, multi-level patches are extracted from image stream around these features. These patch features are tracked and warped based on IMU predicted motion, and the photometric errors are used in the update step as innovation terms. The choice to use ROVIO is made based on the average CPU load of the visual-inertial algorithms proposed by [6]; in fact, the CPU usage—considering the limited CPU resources of the computer companion and the amount of all the operations to be performed during the UAV mission—was considered the main aspect on which to base the overall system design.

ROVIO, unlike other odometry systems (e.g. mono VINS) that attempt to compensate the time's errors, requires that all timestamps be accurate in order to work properly. Considering this aspect, several manual experiments have been done in order to investigate the incidence of the time synchronization and timestamp acquisition on ROVIO. From our experiments we can see that the temporal accuracy depends on both application and the state estimator, but more generally we can say that the range of time acquisition must be between 2 and 5 milliseconds. Besides this threshold, it is no longer possible to follow rapid movements that cause the divergence and drift of the overall system. On the other side, below two milliseconds, we do not perceive huge improvements from the operational point of view.

Most of the camera sensors acquire their timestamp when the image is sent to the computer companion. However, there are many potential sources of delay that can affect the accuracy of the timestamp related to an image like the exposure time of the camera, the internal data processing, internal filter (from IMU point of view), data transfer and also the OS scheduler of the camera. For most camera sensors on the market, these delays are generally including between 5 and 30 milliseconds. While some delays related to the exposure or some other parameters of the camera are constant or can be expected, unknown delays prevent, to the computer companion point of view, from providing accurate timing information to any visual-inertial estimator.

For this reason, we decide to use a custom-made sensor directly linked to a microcontroller that receives data from the IMU and use a trigger line to check when the camera captures images. When an image is taken, as consequence, the microcontroller transmits information about the timestamp and IMU to the computer companion that links it to the image coming from the visual sensor. **Figure 3** shows the schematic of the circuit between microcontroller, IMU, camera and computer companion, while **Figure 4** the two visual and inertial sensors.

2.4 Scheduling system

The overall system, **Figure 5**, is designed as follow: through a web application linked to a web server, the user can select and set the parameters of the mission.

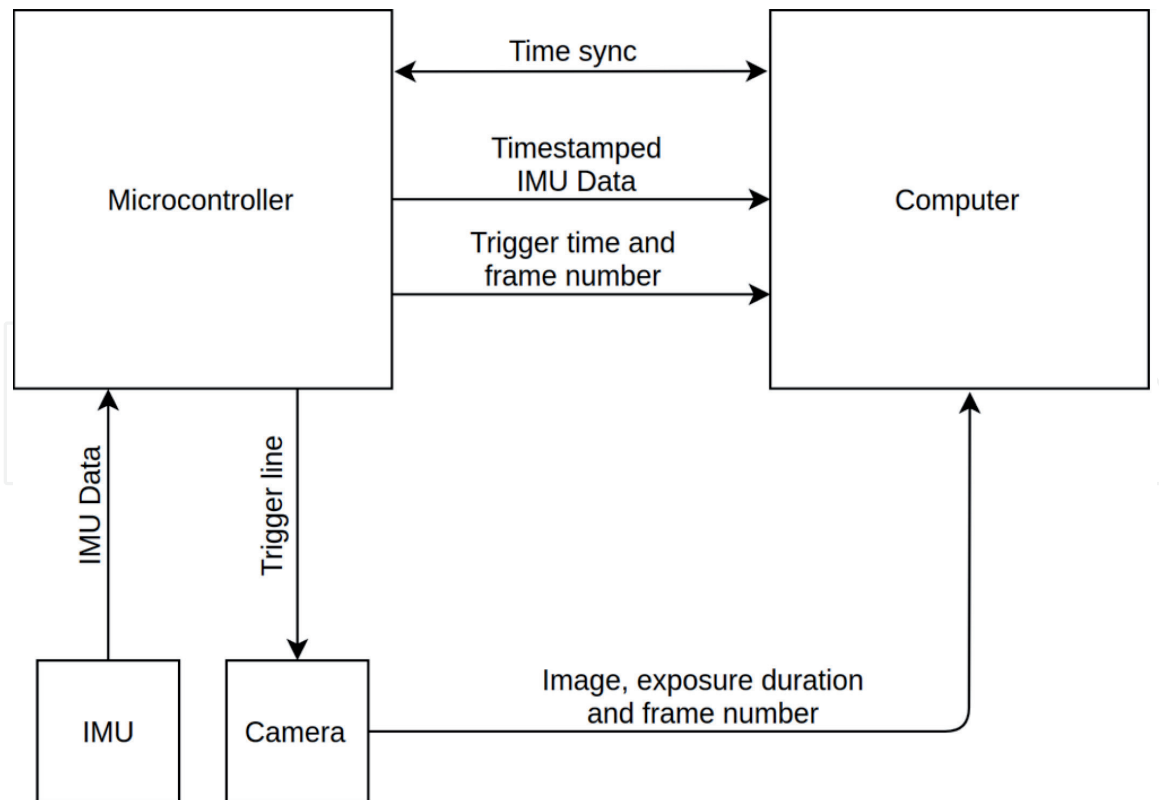


Figure 3.
How computer companion, microcontroller and sensors are linked.

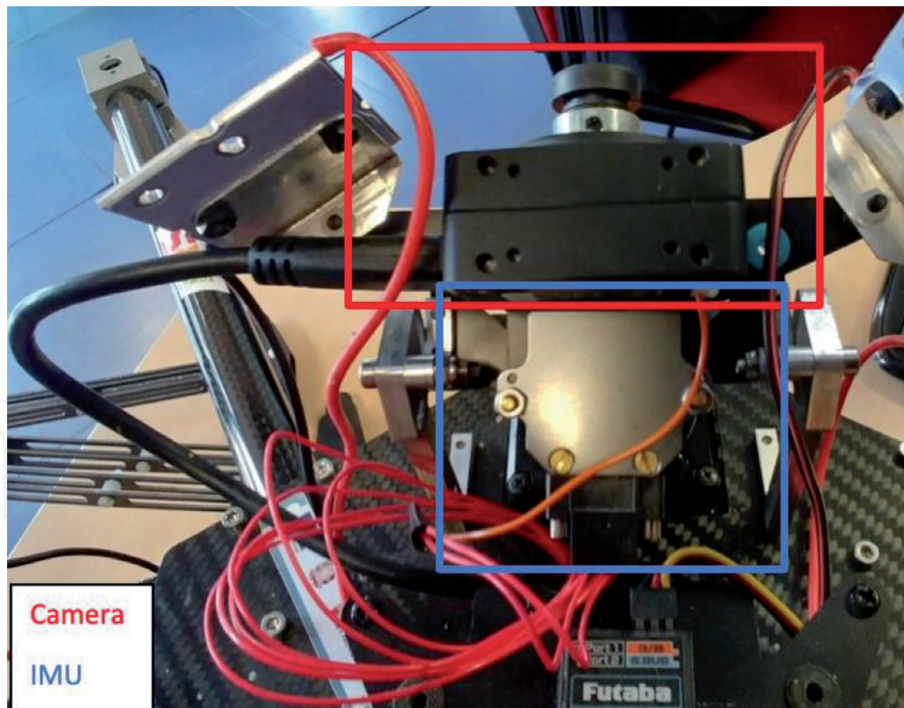


Figure 4.
Camera and IMU sensors.

These parameters are loaded through the scheduling system in which some patterns related to the overall status check of the UAV system (e.g. battery status, sensors status, LED status) are implemented.

At the lower level, there are the ROS nodes responsible for navigation, VIO and flight controller manager that execute the commands translated by the scheduling system.

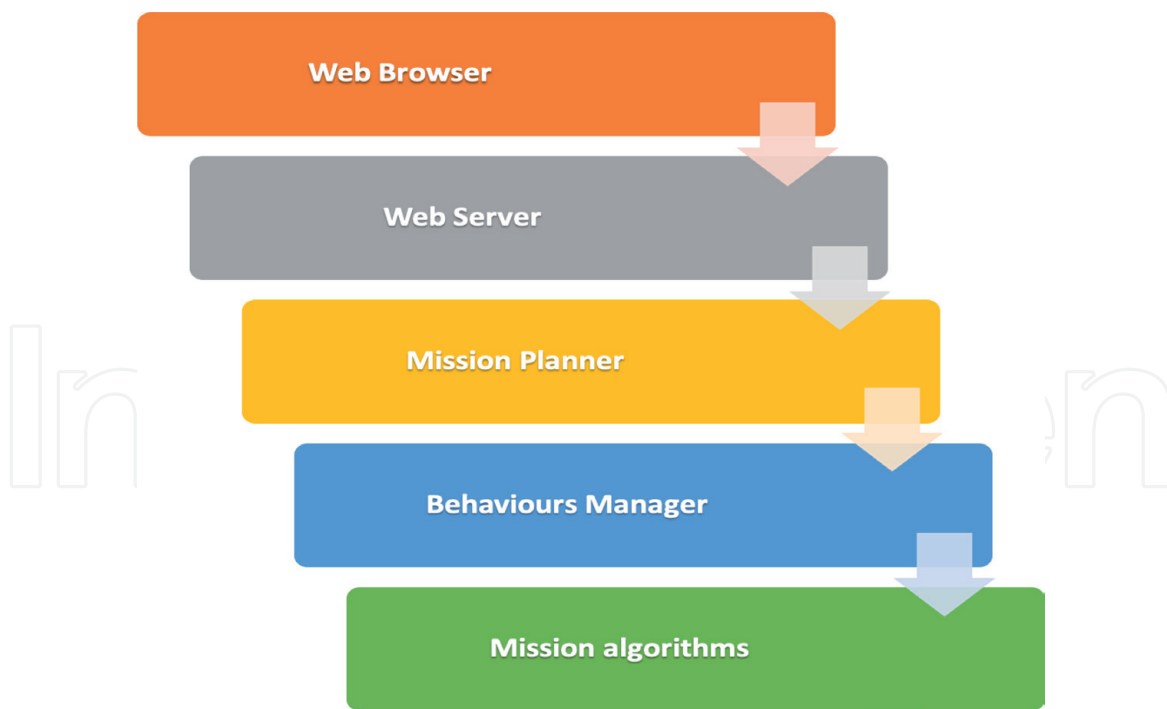


Figure 5.
 Logic behind the system.

Parameter	Type of mission
Tunnel diameter (m)	1/2
Distance to travel (m)	2
Position altitude (m)	1/2
Data record: camera and Lidar (on/off)	1/2
Cruise speed (m/s)	1/2
K positioning (K)	1/2
Come back (on/off)	1/2
Maximum distance (m)	1/2
Minimum distance (m)	1/2

Table 1.
 Setting parameters for each type of mission.

The scheduling system is based on SMACH (<http://wiki.ros.org/smach>) that is a task-level architecture based on ROS for rapidly creating complex robot behaviour. In this application the possible behaviors are two and depend from the type of mission that the user selected through the web GUI:

- Mission 1: Complete exploration of the tunnel
- Mission 2: Partial exploration of the tunnel (fixed distance chosen from the user)

For both missions it is possible to specify if the UAV must return to the home position or land at the end of the tunnel once the exploration is completed. Moreover, there are some specifications that the user can select by GUI. These parameters are related to the geometry of the tunnel and some working condition and are obviously related to the type of mission selected (**Table 1**).

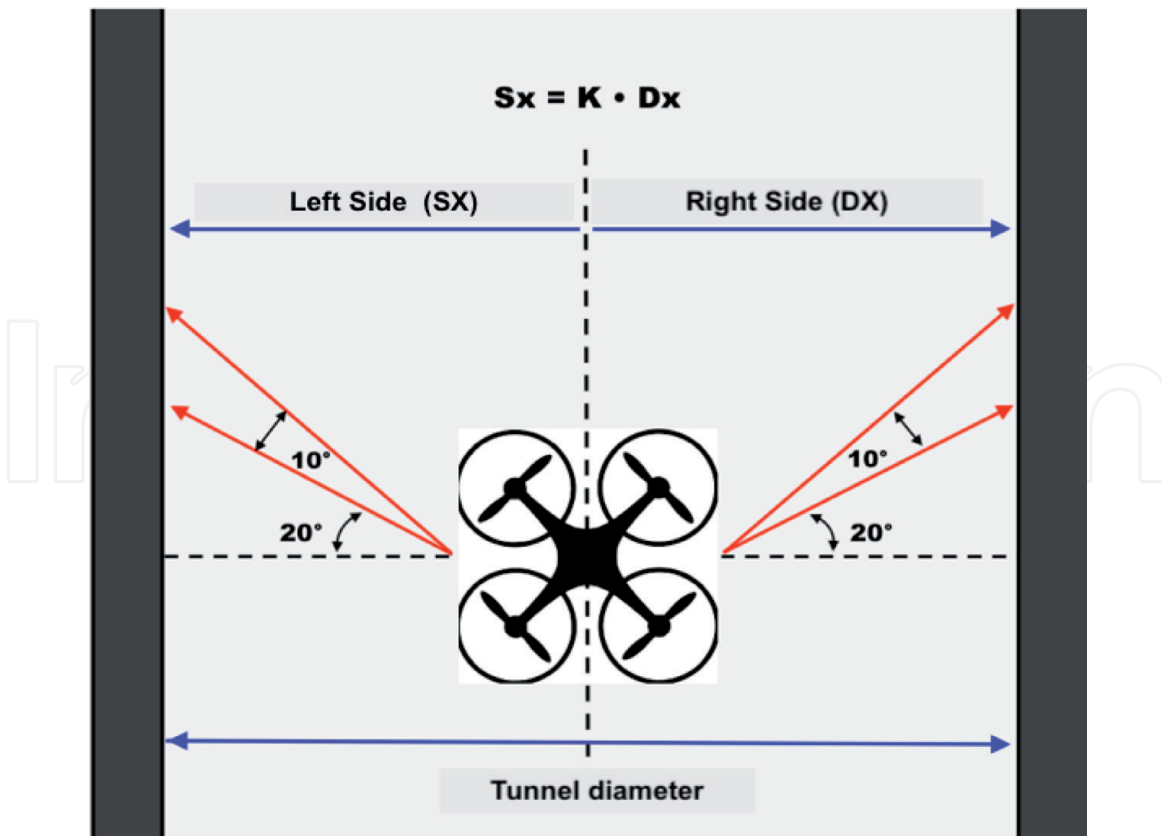


Figure 6.
K parameter logic.

The K parameter indicates the position that the UAV must maintain, while the tunnel inspection is defined as the ratio between the distance of the UAV with the right and left walls of the tunnel (**Figure 6**).

In the hypothesis in which K has been defined equal to 1, the drone will carry out the mission remaining in a central position with respect to the left and right walls. In the same way, with $K = 2$, the distance held to the left wall by the UAV will be doubled compared to the right distance.

The same ratio is maintained even during return to home navigation, when the reference system of the drone will be rotated 180° on the xy plane.

This positioning system was thus implemented to allow a 3D reconstruction of the tunnel inspected by using a single camera.

3. Flight system

Navigation and obstacle avoidance are one of the fundamental problems in mobile robotics, which are being already studied and analyzed by the researchers in the past 40 years. The goal of navigation is to find an optimal path from a starting to the goal point with obstacle avoidance competence. In order to guarantee an autonomous navigation, the robot must be able to safeguard a certain reliability in terms of position (IMU, GPS or other sensors) and ensure a map sufficiently precise to generate a path without collisions and faithful to the real one.

When the robot is in a complete unknown area and does not have information about the surrounding area, the global motion planning fails and does not produce any solution [11]. For this kind of situations, the local motion planning is more suitable.

The objective of the obstacle avoidance is to move a robot towards an area that is free of collisions thanks to the information handled by the sensors during the motion execution, which are steadily updated [12].

In this chapter the autonomous flight system will be defined. In particular, all the aspects concerning the navigation and the related intrinsic logic will be explained.

3.1 Navigation algorithm

Given the application context, a blind tunnel of semi-circular or circular cross-section with a diameter ranging from 2 to 5 metres, it was necessary to develop a specific navigation algorithm that would allow the UAV to explore the surrounding environment avoiding obstacles that could arise during the investigation of the tunnel. The environment taken into consideration for the definition of the algorithm was structured in a tunnel with an entrance and an exit, where there were no bifurcations of the channel.

Within a dark and unknown environment, the use of a Lidar is crucial to carry out navigation in an appropriate manner and for the implementation of the obstacle avoidance algorithm.

Light detection and ranging (Lidar) is a remote sensing technique that allows to determine the distance of an object or a surface using a laser pulse. The distance of the object is determined by measuring the time elapsed between the pulse emission and the reception of the retro-diffused signal. In the same way, to define the height from the ground, the height sensor is necessary. It allows stabilization of the UAV and its navigation to a predefined altitude with the possibility, thanks to the autopilot, of enabling terrain following or the technology that in an automatic way maintains a constant relative distance with respect to the ground.

The main task of the Lidar sensor is to monitor three distances during the navigations. The three distances are one front to the drone navigation and the two laterals, considering a 20° of inclination with respect to the perpendicular drone (**Figure 7**).

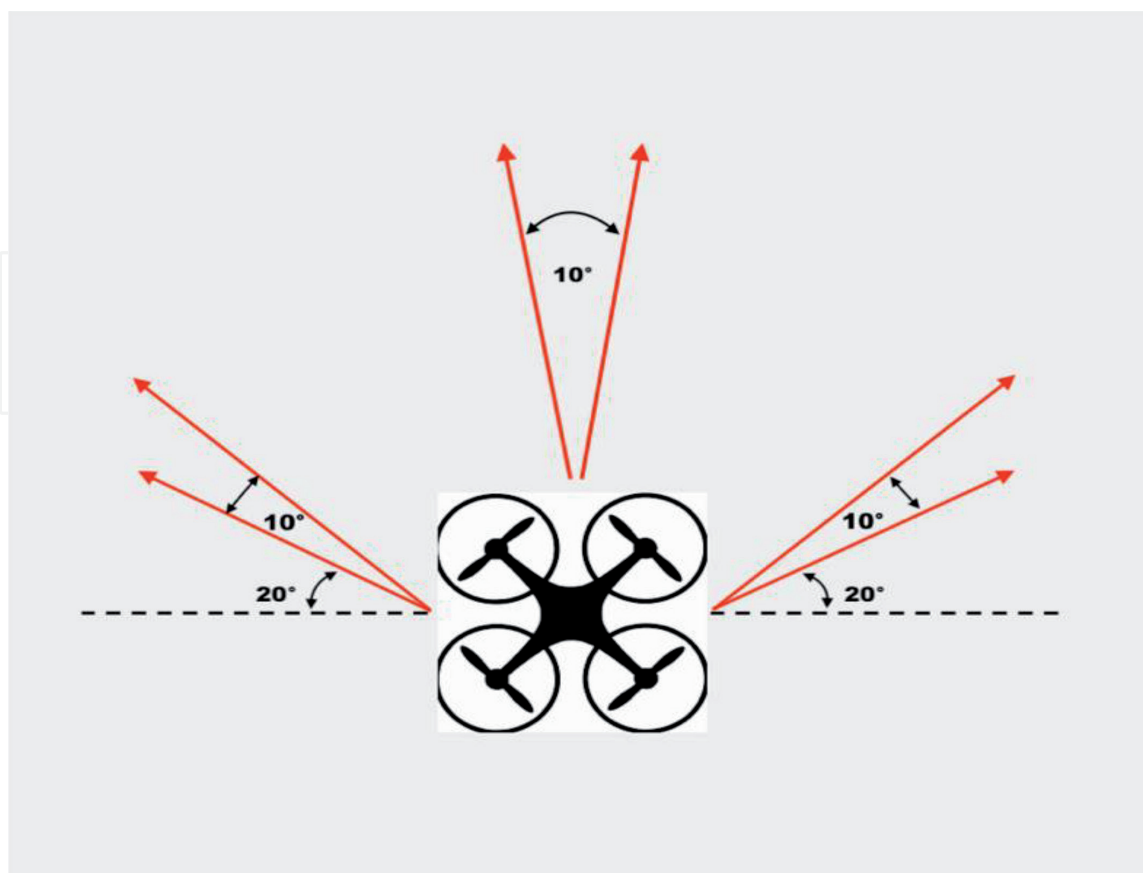


Figure 7.
Monitoring of the three distances.

This solution allows to monitor the frontal space to make sure that the path is free from obstacle, while the two lateral distances serve to guarantee the correct positioning within the UAV tunnel. Being that a single acquisition in any direction may not return completely valid information (optical sensor readings may be subject to disturbance and error depending on the type of surface, color and material on which the signal bounces), it is thought to acquire more data for each direction in a range of 10° in order to achieve a satisfactory level of consistency of the data.

The acquisition of the front distances is necessary to avoid hitting an obstacle present inside the tunnel and more importantly, once the tunnel is investigated in its entire length, recognize the end and be able to start the landing operation. The threshold set for the frontal control has been limited to 5 metres (maximum distance). This implies that until no obstacle is identified in this radius, the UAV will proceed to a predefined cruising speed (1 m/s); on the contrary, if an obstacle is detected, the speed will begin to decrease directly proportional to the distance between the UAV and the above obstacle.

At the minimum threshold value, 2 m from the obstruction, the drone resets its speed by stopping and remaining in hovering condition.

Recognizing the impossibility of advancing the UAV has two possible strategies to pursue: the first strategy involves the initialization of the landing operation, whereas the second includes first a 180° of rotation and then proceeding to the home positioning. Which of the two operations carried out is decided by the operator during the planning of the mission?

Another crucial point of the project was the planning of the rotations that had to be carried out when the anti-collision system recognized the end of the tunnel.

This phase was managed with the aid of the rotation matrices, with the aim of maintaining, during the rotation phase of 180° , the position saved immediately before the start of the rotation. This system had to be studied due to the problem brought by the vibrations and the imperfect balance of the payload installed on the UAV which meant that, in the hovering phase, considering only the rotation along the z axis, the system results are unstable and difficult to control.

With the use of this mechanism during the rotation phase, in addition to the angular speed, there is a continuous contribution of the linear speed along the x and y axis whose goal is to bring and keep the drone in the initial position (x_0, y_0) of rotation.

Considering this aspect, the 180° rotation is managed in two steps:

- Phase 1: The drone makes a 90° counterclockwise rotation and makes a shift on the roll axis to bring the ratio between the two walls to the predefined K value.
- Phase 2: The drone makes a further rotation of 90° , positioning itself with the head towards the direction of round trip.

This positioning system defined using a constant K as a function of the ratio between the distance from the two right and left walls and the other parameters mentioned above, **Table 1**, has thus been implemented to meet the future need of performing a 3D reconstruction of the tunnel inspected (**Figure 8**).

3.2 Test in simulative environment

In this section, various tests will be presented to validate the operation of the entire system and the obtained results, which are evaluated in different unknown indoor environments such as tunnels, to describe the advantages and limitations of this project.

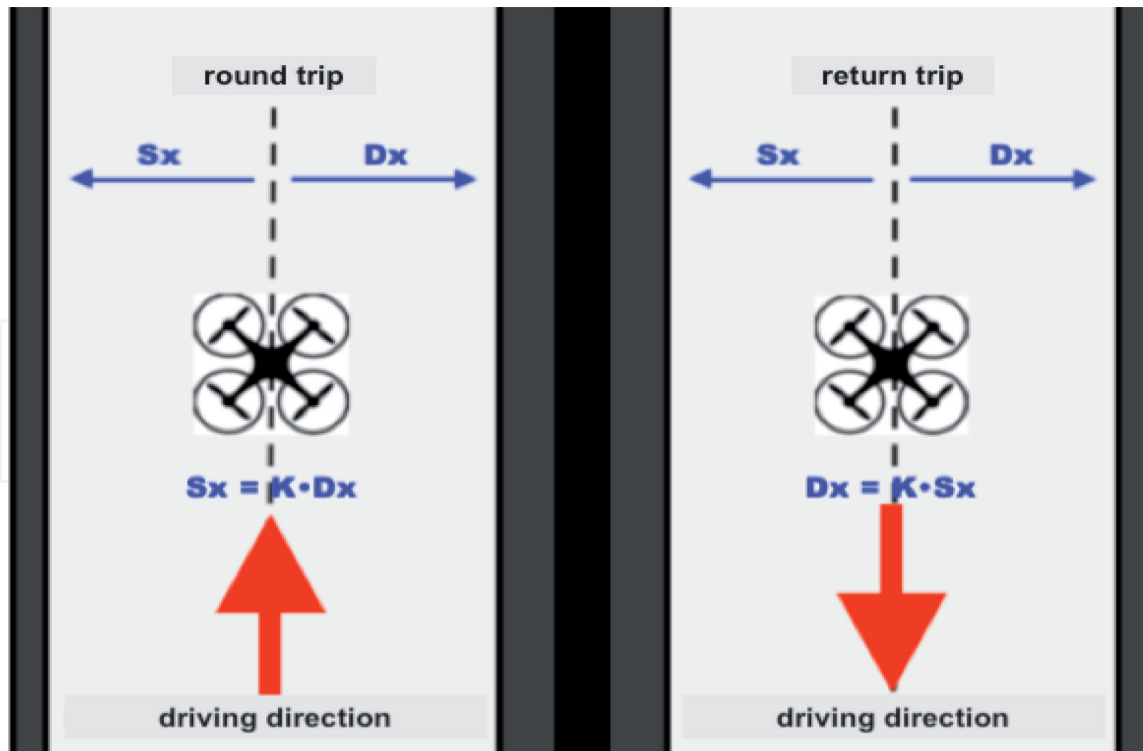


Figure 8.
Tunnel exploration.

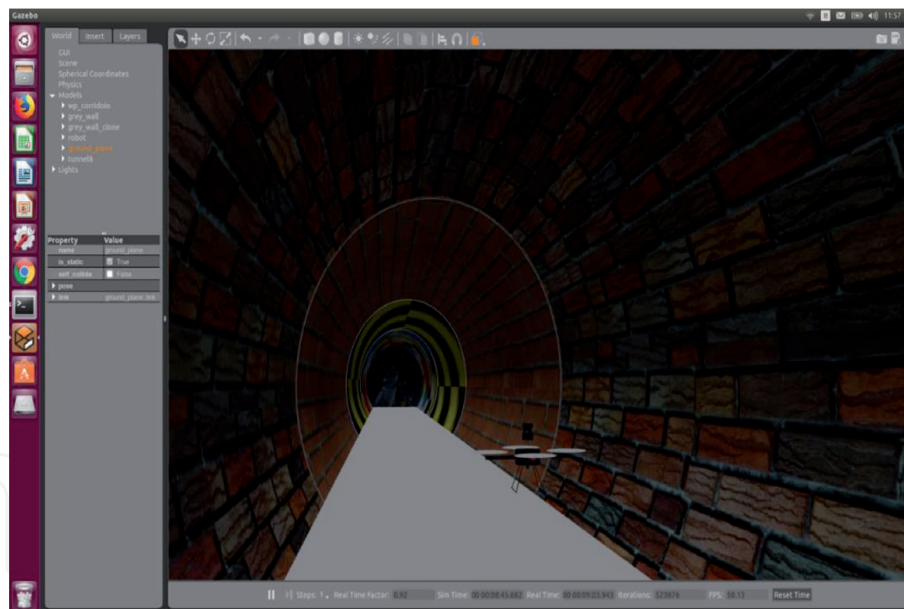


Figure 9.
UAV test in simulative environment.

In order to validate the navigation algorithm presented in this document, before performing real field tests, it was preferred to apply a more precautionary approach by testing the logic of the software in the simulation field.

This kind of approach is preferred for UAV since the failure of navigation frequently involves serious damage to the hardware and therefore, in cascade, a strong impact on the cost of the project.

To assess the quality of the software developed, the first tests were performed in a simulative environment using a UAV model (Figure 9). The simulative environment was defined using Gazebo, while Rviz was used to display the results (both tools are provided by ROS).

The first, Gazebo, is a 3D simulator for rigid bodies and robots, which offers the possibility to simulate precisely and efficiently robots in complex indoor and outdoor environments, with the ability to faithfully reproduce the real situation. The advantage of this tool is the presence of an easy programmable interface, but even more the fact of being an open source software with a strong active community of developers in the world.

Rviz is a suitable tool to view the 3D status of the robot and the performance of the algorithms, to debug faulty behaviors and to record sensor data.

The main purpose was to evaluate the functioning ability of the navigation algorithm. To do this, various simulations were carried out with different parameters, to test the obstacle avoidance algorithm in every aspect.

3.3 Test in real scenario

Once the algorithm and its procedure were validated in all virtual scenarios, the behaviour of the system was tested in a real environment.

The first test carried out using the drone in real scenario was operated in a facility with technical characteristics described in **Table 2** and **Figure 10**.

During the test a precise routine has been followed:

1. UAV positioning at the beginning of the tunnel.
2. System power-on and lipo-battery connection on UAV.
3. Check communication link between UAV and ground station.
4. Execution of ROS launch file.
5. Set up mission parameters.
6. Start mission.

The types of tests that have been performed are divided into two categories:

- Type A: tests conducted in a lighting environment
- Type B: tests with on board LED lighting, in a dark environment

Table 3 shows the results obtained for type B condition and the relative absolute error calculated as the difference in Euclidean distance traveled by the UAV between the point of take-off and point of landing. The distances over which the tests were performed are respectively 10, 20 and 30 m, iterated 10 times in order to compare the error related to the odometry data. **Table 4** displays the average minimum and maximum error for each different test.

Stretch	Height	Width	Length
a	2.15	2.40	11
b	2.10	2.35	30
c	1.90	2.40	5

Table 2.
Characteristics of the tunnel.

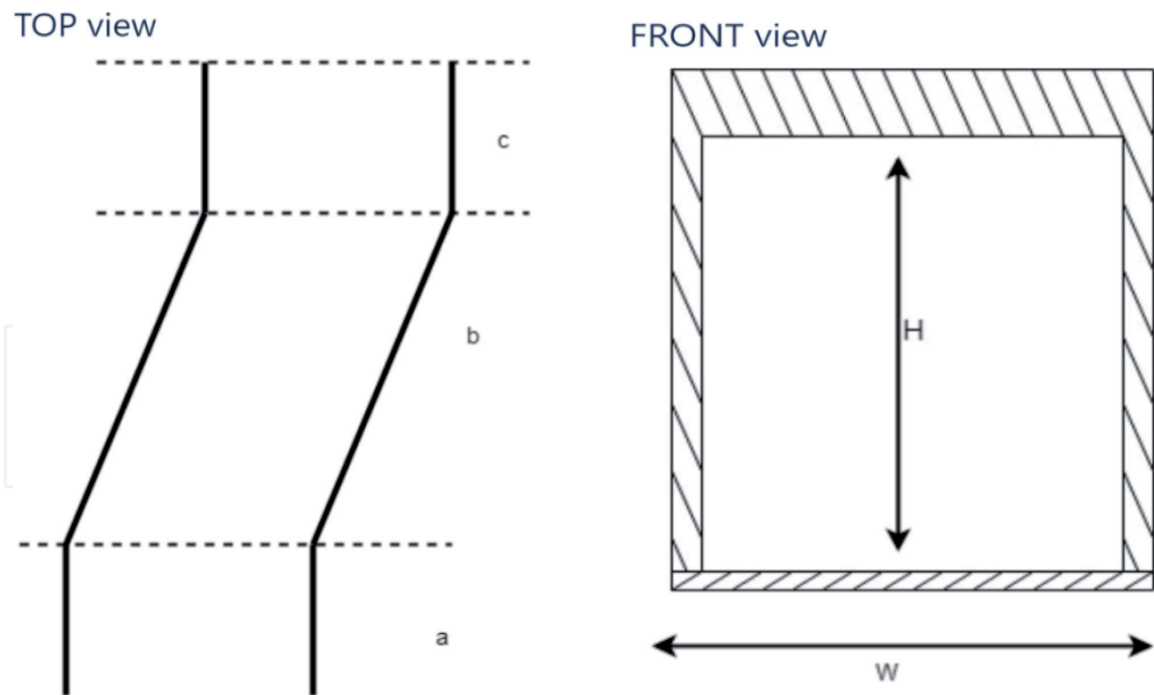


Figure 10.
View of the tunnel.

Test (m)	1°	2°	3°	4°	5°	6°	7°	8°	9°	10°
10 m	0.701	0.867	0.383	1.197	0.280	0.981	0.840	1.111	1.269	0.311
20 m	1.369	1.150	0.511	0.731	0.403	0.732	1.208	0.820	1.242	0.335
30 m	1.35	0.610	0.113	0.689	0.223	0.134	1.383	1.572	1.175	0.301

Table 3.
Absolute error in metres for each different test lengths.

	10 m	20 m	30 m
Minimum	0.280	0.335	0.113
Maximum	1.269	1.369	1.572
Average	0.794	0.850	0.755

Table 4.
Minimum, maximum and average error.

The minimum error obtained for the various ranges of distance tested is consistent with the results obtained in other recent works of the literature [13, 14]. At the same time, if we analyse the average error obtained by performing multiple consecutive tests for each range of distance, it can be seen that an improvement in the visual-inertial system is possible, although the system already guarantees great robustness in operation. An improvement could be obtained by using a different hardware, more performing IMU, and at the same time deepening the aspect related to the calibration of the camera in order to further succeed in decreasing the odometric error during navigation.

After conducting a series of test in facility, we definitively validate the result of the project and the system design in a different real tunnel (**Figure 11**). In this situation it was confirmed that the precision and reliability of the algorithms were enough to allow the system to navigate in total autonomy for at least a stretch of 100 metres.



Figure 11.
Tunnel inspection performed by the UAV during the test of system validation.

4. Discussion and conclusion

The inspection of tunnels and infrastructure for water and hydroelectric resources and, more generally, for any underground work, is essential for the efficient maintenance of the infrastructure itself and provides significant benefits for the rational use of resources. This type of activity collects important information about the current state of consistency of the structure. By highlighting potential or actual failure conditions such as cracks, deformations or other types of problems, it is possible to plan any safety maintenance operations in a timely way. These inspections are now carried out mainly by human operators, with considerable risks to their safety and health at work: claustrophobic, dark and dirty environment.

The idea of this project is to apply innovative techniques, to overcome these problems with the future purpose to ensure greater safety, avoid the inconvenience and risks arising from these activities for the human operator and meet the market needs. As a consequence, a scheduling system has been presented and allows to set different strategies to approach the inspection of the tunnel before starting the mission. Autonomous driving techniques in the six degrees of freedom are developed to ensure the obstacle avoidance in confined space using a simple Lidar sensor. By applying visual-inertial odometry and its fusion with the aid of a Kalman filter, it has been possible the realization of a UAV system able to perform an autonomous inspection of indoor environment like tunnels or conduits.

Although the results shown in this work in terms of robustness and consistency are encouraging, in the future there will be a need to develop advanced techniques considering different scenarios and environments. One possible improvement could be brought developing navigation algorithm based on other types of sensor and using alternative approach. In conclusion, it is central to continue to investigate visual-inertial algorithm since its contribution has proven essential for the robustness, reliability and efficiency of the overall system.

Acknowledgements

The authors want to thank D. Miccone and P. De Petris from WP Web and the Politecnico di Torino Inter-departmental Centre for Service Robotics PIC4SeR, (<https://pic4ser.polito.it>) for the support and knowledge shared.

IntechOpen

IntechOpen

Author details

Lorenzo Galtarossa, Luca Francesco Navilli and Marcello Chiaberge*
Department of Electronics and Telecommunications (DET), Politecnico di Torino,
Turin, Italy

*Address all correspondence to: marcello.chiaberge@polito.it

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Hardy J et al. Unmanned aerial vehicle relative navigation in GPS denied environments. In: 2016 IEEE/ eION Position, Location and Navigation Symposium (PLANS). Savannah, GA; 2016. pp. 344-352. DOI: 10.1109/PLANS.2016.7479719
- [2] Balamurugan G, Valarmathi J, Naidu VPS. Survey on UAV navigation in GPS denied environments. In: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs). Paralakhemundi; 2016. pp. 198-204. DOI: 10.1109/SCOPEs.2016.7955787
- [3] Bachrach A, Prentice S, He R, Roy N. Range-robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics*. 2011;28(5):644-666. DOI: 10.1002/rob.20400
- [4] Ramezani M, Acharya D, Gu F, Khoshelham K. Indoor positioning by visual-inertial odometry. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2017;IV-2/W4:371-376. DOI: 10.5194/isprs-annals-IV-2-W4-371-2017
- [5] Jones ES, Soatto S. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*. 2011;30(4):407-430. DOI: 10.1177/0278364910388963
- [6] Sun K et al. Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*. 2018;3(2):965-972. DOI: 10.1109/LRA.2018.2793349
- [7] Huang G. Visual-inertial navigation: A concise review. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2019
- [8] YoonSeok P, HanCheol C, RyuWoon J, TaeHoon L. ROS Robot Programming. Seoul, Republic of Korea: ROBOTIS Co., Ltd; 2017
- [9] Quigley M, Gerkey B, Smart WD. *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. Sebastopol, CA, USA: O'Reilly Media; 2015
- [10] Bloesch M, Omari S, Jaeger A. ROVIO. 2015. Available from: <https://github.com/ethz-asl/rovio>
- [11] Siciliano B, Khatib O, editors. *Springer Handbook of Robotics*. Berlin Heidelberg: Springer-Verlag; 2016
- [12] Djekoune O, Achour K, Toumi R. A sensor based navigation algorithm for a mobile robot using the DVFF approach. *International Journal of Advanced Robotic Systems*. 2009;6(2)
- [13] Zhang Z, Scaramuzza D. A tutorial on quantitative trajectory evaluation for visual-inertial odometry. 2018:7244-7251. DOI: 10.1109/IROS.2018.8593941
- [14] Cappellaro R. Comparison of Stereo Visual Inertial Odometry Algorithms for Unmanned Ground Vehicles [Master Thesis]. Politecnico di Torino & PIC4SeR; 2019