Simulation in Real Conditions of Navigation and Obstacle Avoidance with PX4/Gazebo Platform

Jesús García
Applied Artificial Intelligence Group
Universidad Carlos III,
Madrid, Spain
jgherrer@inf.uc3m.es

Jose M. Molina
Applied Artificial Intelligence Group
Universidad Carlos III,
Madrid, Spain
molina@ia.uc3m.es

***Abstract*—Design of UAV systems requires from advanced tools to analyse the system components and their interaction in real operational conditions. In this work, the authors present an approach to evaluate a LIDAR sensor and the capacity for navigation and obstacle avoidance in simulated situations using a real UAV platform. It uses available software for mission definition and execution in UAVs based on PixHawk flight controller and peripherals. The solution shows physical integration of the main types of sensors in UAV domain both for navigation and collision avoidance, and at the same time the use of powerful simulation models developed with Gazebo. Some illustrative results show the performance of this navigation and obstacle avoidance function using the simulated sensors and the control of the real UAV in realistic conditions.**

Keywords*— *UAVs sensor processing, Data Analysis, System Design, Gazebo simulation.

I. Introduction

UAVs flying autonomous mission must be controlled based on data streams from electro-mechanical sensors and local or global positioning systems. The controller usually is an embedded microcontroller with appropriate interfaces to all vehicle components. In this work, PixHawk Px4 has been selected as hardware/software platform able to integrate navigation and detection sensor data together with software modules to process the stream of data with fusion algorithms [1].

Fusion of complementary sensors is essential for navigation, a very extended area due to the ubiquity of GPS and the availability of inertial sensors based on inexpensive MEMS components [2], [3], [4]. Other options for non-dependence on the GPS signal involve the deployment of autonomous localization systems such as the recognition of the environment by artificial vision [5] or location by means of electromagnetic beacons [6], with the associated cost of developing a complementary infrastructure.

Complementary to navigation technologies, the use of lasers in combination with other range detection sensors (sonar, radar, video), allows to extend the navigation conditions and obstacle avoidance**.** In the air vehicles (UAVs), the integration requirements (consume, weight, dimensions) are much more restrictive but, even so, it is a line in continuous development [7],[8],[9].

Distance sensors are very useful for a wide variety of UAV applications, such as the measurement of altitude, obstacle detection and surface mapping. There are several types of distance sensors, and each of them has its advantages, limitations and restrictions:

- LiDAR (Light Detection and Ranging) determines the distance to an object or surface with a pulsed laser beam.
- SoNAR (Sound Navigation and Ranging) is a technique that uses the propagation of sound to navigate, communicate or detect objects. It uses sound impulses.
- Radar based on radio waves with frequency modulation. Detect objects based on emitting signals that are reflected.

These distance detection technologies studied have their own characteristics, restrictions and limitations. Choosing the right sensor depends on measurement purpose of the mission. Google and its autonomous vehicle use a LiDAR sensor as the primary sensor to "see" the area around the car when it is in autonomous driving mode. The company Tesla uses RaDAR sensors in their cars equipped with the Autopilot system, because these sensors can detect objects in hostile environments such as snow, rain and dust, while a LiDAR could have problems in these conditions. The ultrasound systems have been proposed in addition for short-range detection and indoor navigation, avoiding collision with walls and near objects in the environment.

Obstacle avoidance algorithms is one of the main goals of vehicle autonomy research and a good methodology requires software simulation to carry out tests without incurring risks for people and devices before having an operational solution. The hardware platform used in this work is PixHawk Px4 and, in this developed environment Gazebo is a powerful 3D simulation environment for autonomous vehicles that is particularly suitable for testing object-avoidance. Gazebo can be used with Software in Loop (SIL) and Hardware in Loop (HIL) design.

This work is based on real UAV that is used to perform predefined flight missions. This UAV platform is optimized following the methodology proposed in [10] for selection of parameters and quality metrics, and has been extended to analyse the integration of obstacle avoidance function. The paper presents in section II the selected platform, design tools and environment for real and simulation experimentation. Section III explains how LIDAR is modelled in Gazebo simulation environment, where real platform and environment is modelled. Section IV explains how simulation is used to provide data input to the controller of the real UAV. Section IV also illustrates the extension of navigation capability with an implementation of obstacle avoidance function. The integration of available and developed components makes possible this function, integrated as an embedded application to process the detection sensor to control the navigation, which illustrated in realistic conditions. Finally, section V shows the simulated test of this function.

II. THE PIXHAWK AND PX4 UAV SYSTEM

A. Architecture

The PixHawk flight controller is open-hardware microprocessor designed to implement autopilot solutions, and integrates two boards, PX4FMU (flight management unit) and PX4IO (input/output). Among the integrated sensors, it contains:

- L3GD20H gyroscope at 760 Hz
- LSM303D magnetometer at 100 Hz
- MPU-6000 accelerometer at 1000 Hz
- MS5611 barometer at 1000 Hz

The control software of PixHawk is PX4, a real-time operating system based on NuttX and consists of two main layers: PX4 Flight Stack, with the embedded applications for UAV control, and PX4 Middleware. This last layer is the interface that allows the flow of data from more sensors to applications through a publish/subscribe system called uORB, which makes the data coming from the sensors available to the applications of the Flight Stack. The outstanding modules are flight controller and sensor data processing [11]. This work proposes an obstacle avoidance function implemented as a parallel application running together with Fligth Stack.

PX4 implements data processing methods for navigation to estimate the vehicle attitude and cinematics:

a) Direction Cosine Matrix

Takes as input the triaxial accelerometers and gyroscopes data to obtain the attitude, usually represented as Direction Cosine Matrix (DCM) or the orientation parameters of the vehicle (roll, pitch, yaw angles)

b) Inertial Navigation System (INS)

This algorithm calculates the trajectories and corrections that allows the vehicle to move between single points using the DCM data, integrated to compute the vehicle attitude with high frequency.

c) Extended Kalman Filter (EKF)

The Px4 system counts with several Extended Kalman Filter algorithms to fuse sensor data in a function that exploits the specific noise and accuracy characterization of each sensor to estimate the vehicle navigation parameters. There are three different modes of EKF operation.

- EKF1; Only use the DCM for attitude control and the Inertial navigation for position prediction (dead reckoning)
- EKF2; Use the GPS for 3D velocity and position. The GPS altitude can be used if barometer data is very noisy.
- EKF3; If there is no GPS, it can use optical flow to estimate 3D velocity and position.

As mentioned, this work explores the possibility of extending data processing by including data from lidar sensor in order to implement obstacle avoidance function as a function working in parallel with the flight management unit, sending appropriate commands when a hazardous situation is detected.

B. Possibility of SIL and HIL design.

Pixhawk supports SIL and HIL simulation using Gazebo [11]. Gazebo makes possible to debug navigation and object-avoidance algorithms in PixHawk fligh controller, without using any real device. In addition, Gazebo offers various models of real autonomous vehicles, saving modelling time. In case of not being offered by Gazebo, it will be necessary to use time and resources in a correct physical modelling [12], so that the conditions of simulation could be as close as possible to reality.

Specifically, in the case of Pixhawk, a HIL configuration (Fig.1) allows running the code written in the flight controller without using any real sensor. That fact allows a first contact in the study of the effect of parameters of navigation and obstacle avoidance algorithms.

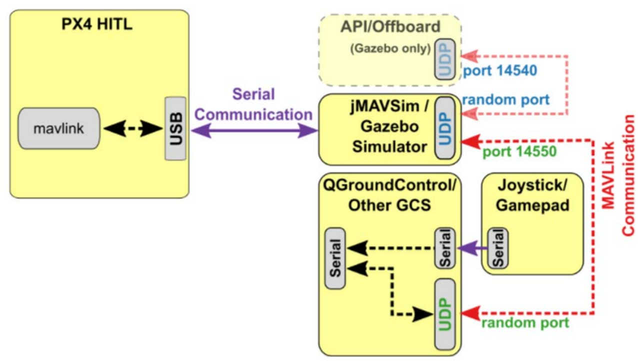


Fig 1. Gazebo HITL Environment [11]

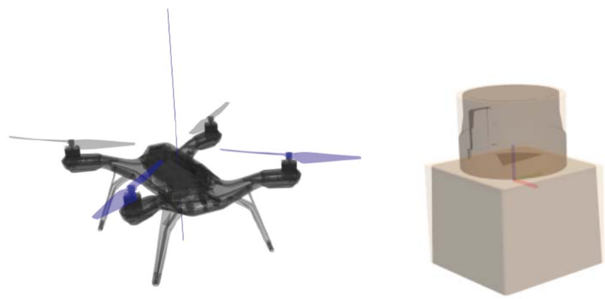
III. LIDAR MODEL ON GAZEBO SIMULATOR

The role of simulation is important to fast and effective verification of the entire system of communication and integration of sensors with the PX4, within appropriate parameters and make tests of validation and integration of the components.

Among some available tools integrated with PX4 software, Gazebo simulator is highly recommended because it provides a powerful 3D simulation ecosystem, versatility and pre-existing available libraries of models. For instance, it supports a variety of models of vehicles such as quadrotors (Iris and Solo models), airplanes, Rover, etc. These two features and the easy interaction with elements such as ROS, MAVLink and QGroundControl make it a competitive candidate to be used as the main simulation model in this work. Other existing alternatives are:

- jMAVSim, an exclusive multirotor simulator characterized by its lightness, easy to configure and allows to check if the aircraft takes off, turns and lands properly.
- AirSim, a simulator that provides simulations of physics and a visualization very close to real world, it consumes a large amount of resources intensively.

The system developed in this word integrates the 3RD Iris model of the real UAV previously use to execute missions (Fig. 2). From this point, the proposal develops a 2D LiDAR sensor simulation that is mounted on the top to evaluate its capacity for detecting obstacles.



a) b)

Fig 2. Models implemented in Gazebo. A) Iris quadrotor b) lidar 2D scanner

In order to validate the 2D LIDAR model, some static trials have been performed with the vehicle in a fixed position on ground and one or two obstacles. (Figs. 3,4).

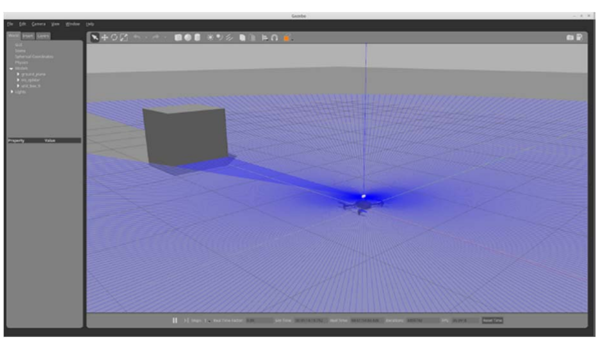


Fig 3. 2D LIDAR simulation on Gazebo with one obstacle

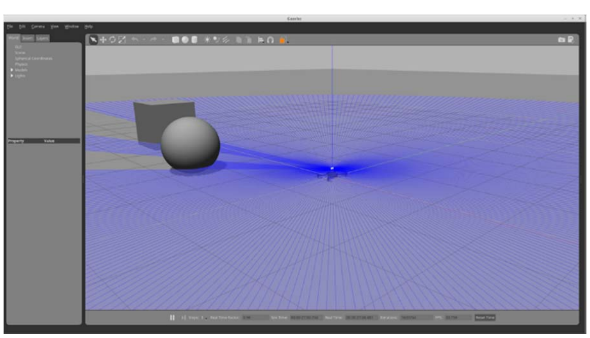


Fig 4. 2D LIDAR simulation on Gazebo with two obstacles

IV. Simulation

This section represents the synthesis of the developed elements, in order to reach the final goal: to create a system that, through the data obtained by the sensor, is able to detect the presence of obstacles that interpose in the flight mission and react before them by sending a series of MAVLink commands that allow us to manage the trajectory of the mission to avoid this obstacle. Once the drone encounters an unknown element in the execution of the mission, the obstacle detection application will be activated.

A. Architecture

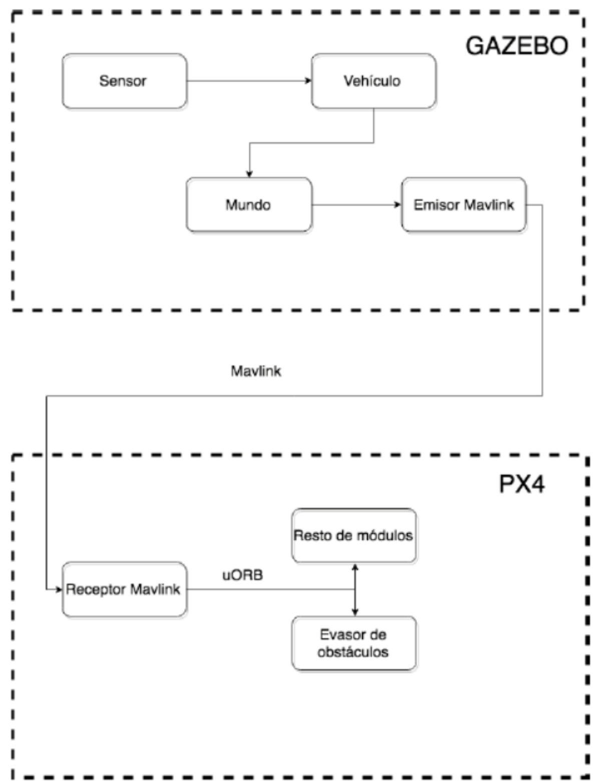


Fig 5. Integration of Gazebo simulator Gazebo with PX4 and obstacle avoidance

Figure 5 shows how the sensor module should be added within simulator. A part of the module is on the vehicle, and in turn, the vehicle is part of a world in which we have all the modules of the simulation scenario and other modules such as the communication through Mavlink. As the communication module and the sensor module are in the same world, they can communicate with each other to send the information to PX4 via Mavlink with the sensor message already predefined.

B. Communications con PX4 via "MAVLink messages"

As indicated, in PX4 there are two main communication modes, Mavlink and uORB. uORB is responsible for the internal communication of PX4, within the same execution space, while Mavlink is responsible for external communications between PX4 and other systems such as the ground control station. In this paper, the execution of the obstacle detection process is a secondary task in PX4 Flight Stack, using uORB within the system itself to communicate processors. In the case of having needed more resources for the execution of the module, it could have been executed in an external hardware and send the commands through Mavlink. Thus, an alternative would be to process the data in another microprocessor, for example, a Raspberry Pi and communicate with PX4 through Mavlink.

Therefore, the objective is establishing a communication with the flight controller to modify the mission accordingly to the conditions through a dedicated process running in parallel to the navigation. The command messages allow to greatly simplify the programming of the application. The general process is to select the flight mode and the specific parameters for that mode, as indicated in the following tables (Tables I, II).

TABLE I. Messages for flight mode selection and mission parameters

176	MAV_CMD_DO_SET_MODE	Set system mode.
	Mission Param #1	Mode, as defined by ENUM MAV_MODE
	Mission Param #2	Custom mode - this is system specific, please refer to the individual autopilot specifications for details.
	Mission Param #3	Custom sub mode - this is system specific, please refer to the individual autopilot specifications for details.
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty

TABLE II. Available modes with PX4

Value	Field Name	Description
0	MAV_MODE_PREFLIGHT	System is not ready to fly, booting, calibrating, etc. No flag is set.
80	MAV_MODE_STABILIZE_DISARMED	System is allowed to be active, under assisted RC control.
208	MAV_MODE_STABILIZE_ARMED	System is allowed to be active, under assisted RC control.
64	MAV_MODE_MANUAL_DISARMED	System is allowed to be active, under manual (RC) control, no stabilization
192	MAV_MODE_MANUAL_ARMED	System is allowed to be active, under manual (RC) control, no stabilization
88	MAV_MODE_GUIDED_DISARMED	System is allowed to be active, under autonomous control, manual setpoint
216	MAV_MODE_GUIDED_ARMED	System is allowed to be active, under autonomous control, manual setpoint
92	MAV_MODE_AUTO_DISARMED	System is allowed to be active, under autonomous control and navigation (the trajectory is decided onboard and not pre-programmed by waypoints)
220	MAV_MODE_AUTO_ARMED	System is allowed to be active, under autonomous control and navigation (the trajectory is decided onboard and not pre-programmed by waypoints)

This capacity of selection flight modes allows to use this communication together with the data obtained from the sensor, to make a system that modifies the flight path when a mission is running in autonomous mode, providing commands that insert new waypoints into the mission to evade obstacles

C. Obstacle avoidance

Although it is not the main objective of this project, to evaluate the 2D LIDAR sensor, a functional detection system and obstacle avoidance [13] has been made consisting of three main phases:

- **No Danger**. In this phase, the measured distance is checked and compared with the safety distance. If the distance is smaller, then the system will enter into danger phase.
- **Danger**. This phase is created to avoid false readings or when reading is the distance to the ground. In the last case, if the projected distance is similar to the estimated height, with a margin of error, the obstacle is discarded, using basic geometry to estimate height as □ = cos α * d, where h is the height and α is the pitch angle.
- **Avoidance**. To avoid possible obstacles, the command "insertion of a new waypoint" will be used, taking as parameters coordinates and height to inserts in the mission.

New coordinates are computed with direct geometric calculations. A normal vector to velocity is used to define the inserted waypoint (Fig. 6).

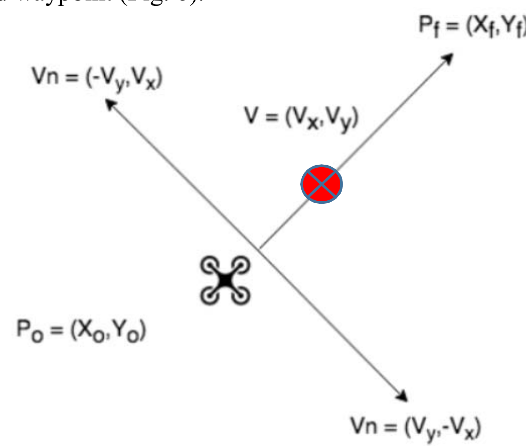


Fig 6. Geometry of obstacle avoidance solution

$\overline{P_n} = \overline{P_0} \pm \mathrm{SD} * \bar{v}_n$, being SD (Safety Distance), a configuration parameter mentioned above. To decide whether to turn to the left or right, before starting the avoidance manoeuvre, left and right sectors of sensor readings are analysed. If there is no obstacle at the left, the drone will turn to the left, if there is an obstacle the drone will look at possible dangers on the right, as on the left, if there is no danger turn on the right, but if there is danger turn 180º to the back part of the drone, and it would repeat the process. If for some reason the drone is in danger in all directions, it will automatically land.

V. Evaluation in Realistic Conditions

The methodology proposed in [10] was designed to assess the system performance under realisitic illustrative conditions. The first step of this methodology is the definition of the UAV platform, the type, cinematic characteristics, set of sensors and the navigation algorithms..

A. GPS and INS data fusion

The data fusion process of Pixhawk (EKF2 filter) provides the attitude integrating data from magnetometer, gyroscope and accelerometer, and then fuse with accelerometers and GPS data to estimate position and velocity. So, the sensor fusion system is based on a loosely coupled architecture which uses GPS position and velocity measurements to aid the INS, typically used in most of navigation solutions based on sensor fusion [14],[15]. Some illustrative results of this function and logged data were explained in [10].

B. Obstacle avoidance validation

The 2D LIDAR model implemented in Gazebo is connected with PX4 and sends simulated data (distances and angles) which are processed as described in section IV. Tests are defined as a mission without obstacle and then the same mission with obstacles, putting the drone in several situations where it must turn in one or more objects, in this case a house as shown in Fig. 7. The simplest mission is a simple single-point mission: the drone will have to take off and go to point 1 without stopping and waiting for new orders there. In Fig, 8 we can see the mission to be carried out without obstacle, and Fig. 9 with the obstacle in the way.

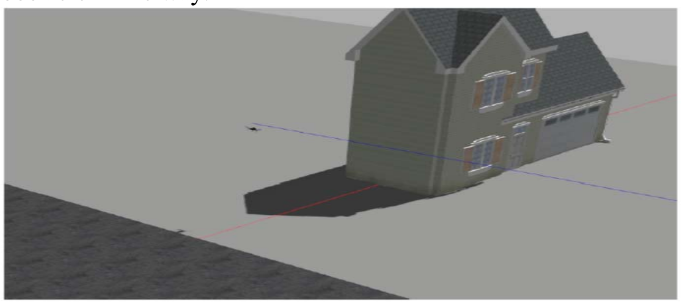


Fig 7. Simulation of a house as obstacle in Gazebo

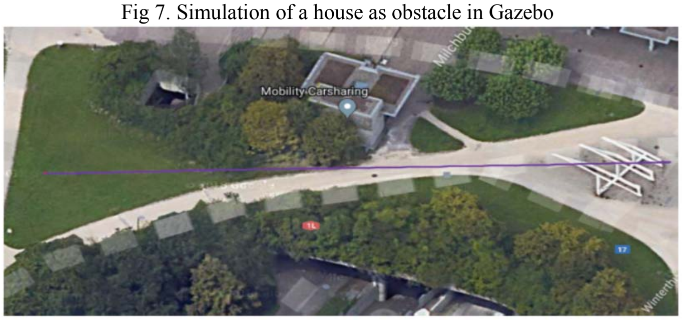


Fig 8. Free mission

Fig 9. Obstacle in trajectory

Figs. 10,11 shown velocities of flight in Cartesian coordinates. In the case with no obstacle the velocity is constant since the mission starts with take off until the end. The vertical component of velocity changes at the beginning of the mission while the drone is ascending, and the rest of the mission remains constant. As shown in Fig. 11, when the drone detects the object both components of velocity go down to 0 or even are negative to counteract the braking, and after that velocity takes the indicated value to carry out the modified mission shown in Fig. 9.

Regarding the attitude of drone, as shown in Figs. 12, 13, mission starts in both figures and there are a few seconds where the pitch is constant while the drone is taking off. After that, as can be seen in Fig. 13, it leans forward to move and when the evasion manoeuvre begins the pitch increases to positive values to stop the drone and stabilizes at 0, then increases again to avoid the object, once the object is avoided, returns to 0, to return to the mission target and lean forward again to advance and continue the mission until the end. Conversely, Fig. 12 shows a pitch with no changes practically until mission ends.

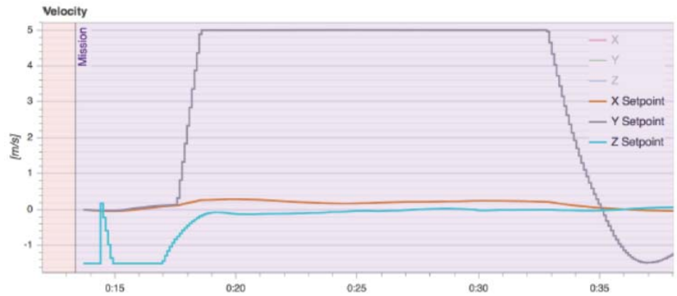


Fig 10. Velocity without obstacles

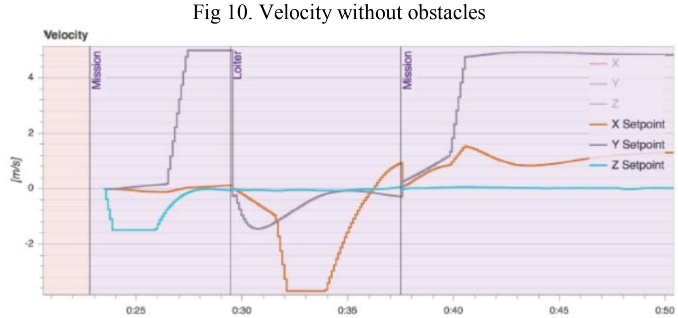


Fig 11. Velocity with obstacles

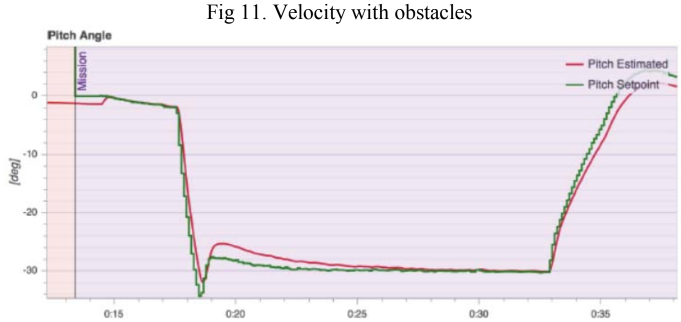


Fig 12. Pitch without obstacle

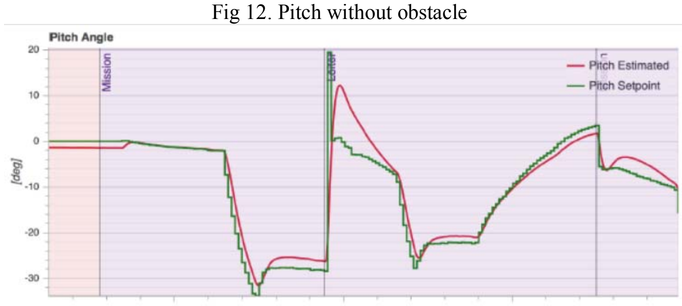


Fig 13. Pitch with obstacle

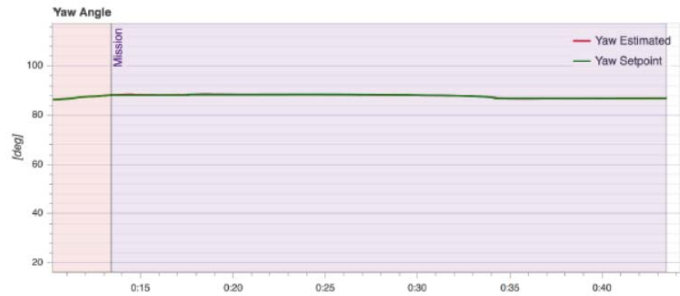


Fig 14. Yaw without obstacle

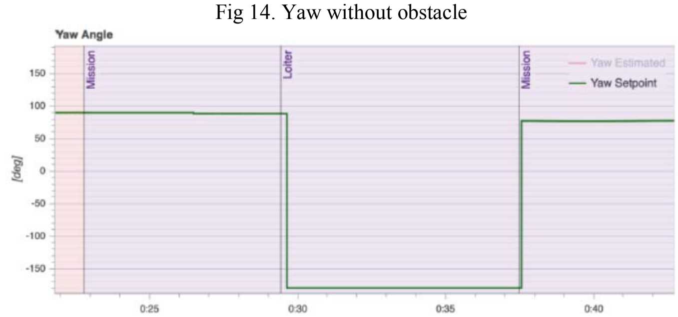


Fig 15. Yaw with obstacle

Regarding heading, Fig. 15 shows the yaw of drone, it performs a horizontal turn of 90º to avoid the obstacle. The trajectory starts with heading of 90º, it turns to increase it in 90º until 180º, although this is indicated in the graph as -180º. As reference, Fig. 14 shows the trajectory without obstacle and the yaw remains stable throughout the mission.

VI. Conclusions

This paper presented a platform (Pixhawk PX4) and methodology to experiment with realistic simulated conditions for UAV autonomous navigation. Based on data analysis and characterization the algorithms can take advantage of available sources. In addition, a modelling and software simulation methodology has been added using Gazebo to model 2D lidar/ultrasound sensors. Results shown successful integration of the sensor and a dedicated data processing task running in the platform. This tool is especially interesting to test the obstacle avoidance function, as previous analysis before operation in real scenarios.

So, the work developed to improve the current autopilot system implemented in PX4 has addressed the integration of a LIDAR sensor to implement an illustrative obstacle avoidance functionality. The possible solutions to interact with the PX4 system were analysed, including the communication between processes with uORB and how to create a secondary task that runs in the background and performs the task in real time.

As future work it is necessary to add a driver to obtain the actual laser distance sensor data and include it in the communication message in order to test the system with real data. The same methodology could be used to integrate another type of distance sensor. A more precise evasion system could be developed and evaluate the performance compared with other alternatives in the state of the art. Finally, this system may need a computing capacity greater than that provided by Pixhawk, so it could be implemented in a parallel computer that communicates with the main system through Mavlink but with the same structural scheme proposed in this work, comparing the computational load of both alternatives.

Acknowlegments

This work was funded by public research projects of Spanish Ministry of Economy and Competitivity (MINECO), references TEC2017-88048-C2-2-R, RTC-2016-5595-2, RTC-2016-5191-8 and RTC-2016-5059-8. Authors also are grateful to Enrique Martínez Martínez and Ivan Serrano Ruiz, Graduate Engineers at UC3M in 2018, for their helpful collaboration in this work.

References

[1] Garcia J. Molina J.M., J. Trincado. “Analysis of real data with sensors and estimation outputs in configurable UAV platforms”. Sensor Data Fusion: trends, solutions and applications 2017. Bonn, Germany, 10-12 Oct. 2017.

[2] Groves P.D. “Navigation using inertial sensors”. IEEE AES Magazine (Vol.30, Iss. 2) 42 - 69 Feb. 2015.

[3] Britting, K.R. “Inertial navigation systems analysis”. Artech House, 2010.

[4] Farrel J.A., “Aided Navigation: GPS with High Rate Sensors”, McGraw-Hill, New York, 2008.

[5] Choi H., Kim Y.. “UAV guidance using a monocular-vision sensor for aerial target tracking”. Control Engineering Practice, 22, 10-19, 2014.

[6] Torres-González A., Martínez-de Dios J.R., Ollero A. “Robot-Beacon “Distributed Range-Only SLAM for Resource-Constrained Operation". Sensors 2017, 17(4).

[7] Ferrick A., Fish J., Venator E. and Lee G.S. “UAV Obstacle Avoidance Using Image Processing Techniques Technologies for Practical Robot Applications (TePRA”, 2012 IEEE International Conference on 23-24 April 2012

[8] Fasano G., Accado D., Moccia A., Moroney D., “Sense and avoid for unmanned aircraft systems”. IEEE AESS Magazine, vol. 31, nº 11, pp. 82-110, 2016.

[9] Rankin G., Tirkel A., Leukhin A. “Millimeter Wave Array for UAV Imaging”. MIMO Radar Radar Symposium (IRS), 2015 16th International 24-26 June 2015.

[10] J. García, J.M. Molina."Analysis of Sensor Fusion Solutions for UAVs", Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA). 2018. Granada, Spain October 23-26 2018.

[11] Meier L., PX4 Development Guide. [Online]. Available: https://dev.px4.io/en/

[12] Martí E., García J., Molina J.M. “A Simulation Framework for UAV Sensor Fusion”. Hybrid Artificial Intelligence Systems 5th International Conference, HAIS 2010, San Sebastián, Spain, June 23-25, 2010.

[13] Fasano G., Accado D., Moccia A., “Sense and avoid for unmanned aircraft systems”. IEEE Aerospace and Electronic Systems Magazine, pp. 82-110, 2016.

[14] Martí E., García J., Escalera A., Molina J.M., Armingol J.M. “Context-Aided Sensor Fusion for Enhanced Urban Navigation”. Sensors Vol. 2012 16802-16837. 2012

[15] Layh T., Demoz G.E.“Design for Graceful Degradation and Recovery from GNSS Interruptions”, IEEE AESS Magazine Vol. 32, N9, September 2017. 4-17.