Automatic Landing without GPS

Thomas LEONARD

LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6006
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2016

Automatic Landing without GPS

Thomas LEONARD

LUNDS UNIVERSITET

Department of Automatic Control

Msc Thesis
ISRN LUTFD2/TFRT--6006--SE
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Media-Tryck
Lund 2016

Abstract

Sagem Défense et Sécurité (now Safran Electronics & Defense), a French space and defense company of the SAFRAN group, is working on the next generation of Unmanned Aerial System (UAS). This UAS features a fully automatic Unmanned Aerial Vehicle (UAV) equipped with a state-of-the-art navigation system. This navigation system relies mainly on a high-accuracy Inertial Measurement Unit (IMU) coupled with a GPS receiver. But the GPS is known to be easy to jam, either naturally (solar flare for example) or intentionally. In the event of a loss of GPS signal, the navigation system is not able anymore to provide accurate position and speed information to the Flight Controller (FC). Deprived of reliable position and speed information the FC is not able to guide the UAV safely to the ground.

So the goal of the project detailed in this report is to add to the existing UAS the ability to land safely in case of a GPS loss. At the core of the solution described in this report is a sensor fusion algorithm taking as input inertial, vision based, barometric, laser and azimuthal measurements. The filter is using all these measurements to establish reliable position and speed estimates.

Even if very reliable systems enabling automatic landing without GPS exist today; they all require heavy and expensive ground equipment. This is why SAGEM decided to develop its own solution using more embedded sensors and less ground equipment. This is a first step toward a fully embedded automatic landing system nondependent on GPS availability, a very active field of research today. All the tests done during the thesis and presented in this report shows the efficiency and robustness of this solution.

Acknowledgements

During my entire Master's Thesis I have received great help and support from my colleagues at SAGEM. I would like to thank them for the time they spent with me, I have learnt a lot from them. It has been a privilege to work with them on this very interesting project.
I would also like to thank Prof. Rolf Johansson for his strong involvement in my thesis supervision and for his help, along with Prof. Karl-Erik Årzén and my opponents Amanda Eriksson, Markus Malmros, Farid Alijani and Jacob Mejvik for helping me to improve this report.
Finally I would like to thank the previous student who worked on this subject and my supervisor at SAGEM as my work is based on what they had done and I owe many things presented in this report to their work.

Acronyms

Above Ground Level	**AGL**
Automatic Landing System	**ALS**
Above Mean Sea Level	**AMSL**
All Weather Camera	**AWC**
Body Reference Frame	**BRF**
Calibrated Air Speed	**CAS**
Earth-Centered Earth-Fixed	**ECEF**
Extended Kalman Filter	**EKF**
Flight Controller	**FC**
Global Positioning System	**GPS**
Ground Station	**GS**
Inertial Measurement Unit	**IMU**
Inertial Navigation System	**INS**
North East Down	**NED**
Position Velocity Attitude Time	**PVAT**
Touch Down Point	**TDP**

Unmanned Aerial System	**UAS**
Unmanned Aerial Vehicle	**UAV**
World Geodetic System 1984	**WGS84**

Contents

1. Introduction

This master thesis is part of the Master in Engineering Physics of Lund University. It is attached to the Automatic Control Department. Prof. Rolf Johansson, Lund University, has been supervising this thesis and provided me great help during the different stages of the project and Prof. Karl-Erik Arzén is the examiner. This thesis started on January 2016 at SAGEM facilities near Paris, France, and lasted 6 months, until the end of June 2016.

This section starts with an introduction on the context of the project that will lead to the presentation of the main goals of the project along with the expected results. The section ends by the outline of the report.

1.1 Context of the Thesis

Safran Electronics & Defense

This master's thesis has been conducted at the Unmanned Aerial System (UAS) department of SAGEM in France. In this department SAGEM has been developing UASs for more than 20 years.

Part of the French group SAFRAN, SAGEM is also known worldwide as the European leader in Inertial Navigation System (INS) for aeronautical, nautical and terrestrial applications, as the world leader in helicopter flight control and as the European leader in optronic and tactical UAS.

It has been a great opportunity to work at SAGEM and I received help from people with a deep knowledge of inertial navigation and navigation filters.

UAS Project

This thesis took part in a UAS development project. SAGEM is developing an Unmanned Aerial Vehicle (UAV), fully automatic, able to perform a broad panel

of missions thanks to various payloads and long flight endurance. It is a fixed wing aircraft, looking like a conventional general aviation aircraft as it can be seen on Fig. 1. More information can be found about this project in documents [1] [2] available on SAGEM's website.

Fig. 1 - SAGEM's UAV

This UAV is always connected to a ground station, as visible in Fig. 2, from which it is operated. Inside a ground station there is a UAV operator who can monitor the aircraft and send high level commands while a payload operator is controlling the embedded payload.

Fig. 2 - Ground Station, on the left the UAV operator and on the right the payload operator

The UAV is navigating using high technology inertial sensors developed by SAGEM but the GPS is a key part of the navigation system. It is well known that a GPS receiver is very easy to jam, either intentionally or even naturally by an intense solar activity, for example, as warned in research paper [3]. Such an event resulting in a loss of GPS signal for a long period of time could have catastrophic consequences for the UAV that would not be able to land safely on the runway. So in order to make the product more robust it must be able to land safely without using the GPS, in a degraded mode operation. The development and test of this feature is the subject of this thesis.

Thesis's Scope and Organization

Scope

Before the beginning of this thesis another student, also doing a master's thesis, worked on this subject entitled "automatic landing without GPS" here at SAGEM. As the report made at this time is not publicly available there will not be any reference to it.

The scope of the thesis presented in this report extends from design of the solution to simulation tests and performance assessment. The design itself was already in a quite advanced stage when this thesis started and some simulations had already been done in order to measure the performances. That means that the tasks associated with the thesis were to understand the solution proposed for the landing without GPS problem, then to test it more widely than it had previously been tested and finally to improve it in order to reach the expected level of performance.

The Automatic Landing System (ALS), understood automatic landing system without GPS, has to be implemented on the UAS in the coming years. Therefore it was a truly operational thesis. That means that along with the research and testing work came a few operational problems which were a bit out of the scope of the thesis; only a few of them will be briefly mentioned in this report. This was a great opportunity since flight test data were available in order to improve the models and perform open loop testing.

The integration of the solution on the real system and the flight tests are not in the scope of the thesis, but a test plan should be handed out at the end.

Organization
So as suggested by the scope of the thesis, the first step was to go through what had previously been done on the subject. The study was focused on the previous master's thesis done on the subject and also on the literature on inertial and vision based navigation and landing. After this first step it was possible to start doing simulations with the code already available in order to reproduce the results of the previous thesis. By doing so it started to appear where the models had to be improved, where the solution could be improved and what had not been tested yet. So the models and the simulator were improved in order to test the solution in conditions closer to reality, this led to a few improvements in the filter which will be detailed in this report.

Tools
The main tool used during this thesis was Matlab [rev 2015a] with the Simulink environment. Please refer to [4] if an introduction to this software is needed. For some rendering task the computer vision toolbox was used but it is not necessary to reproduce the algorithms presented in this report.

1.2 Goals and Specifications

Goals
The main goal of the thesis is to test and improve a solution to safely bring the UAV back on the ground in case of GPS loss. As the project did not start from scratch and because the feature will be implemented on a UAS already able to perform automatic landings when the GPS is active, this goal has been redefined as: establish a reliable position and speed estimate without using the GPS. This position and speed information is then sent to the flight controller (FC) so it can control an automatic landing. That means that for the flight control system there will not be any significant differences between the normal operation mode and the degraded mode as it will be provided with a position and speed information in both modes.

In order to land safely, the position provided to the flight control systems has to respect accuracy constraints. As those data are confidential they will be mentioned as $LatAccuracy$ and $LongAccuracy$. The accuracy that has to be achieved in the position estimate is $LatAccuracy$ in the lateral axis (perpendicular to the runway), and $LongAccuracy$ in the longitudinal axis. Those specifications

had been determined prior to the beginning of the thesis and they result from the flight control system, aircraft dynamics and runway size specifications. It is important to understand that it is not the accuracy of the landing but only the accuracy of the position estimate, to which the guidance error will be added to give the landing accuracy. Fig. 3 shows how these accuracy constraints look like.

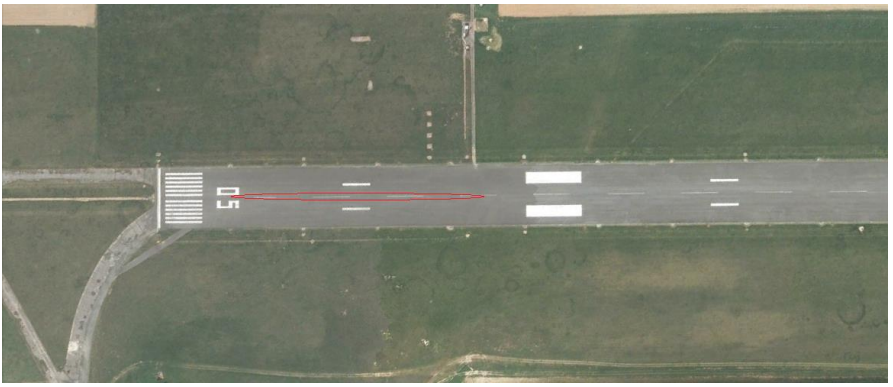


Fig. 3 - An illustration of what the position estimate looks like: more accuracy has to be achieved on the lateral axis than on the longitudinal axis in order to stay on the runway

Constraints

As the solution designed for the automatic landing without GPS is made to be integrated on an existing UAS, there are a few constraints inherent to this existing system. The goal is to implement the solution with the least modifications of the existing system. That means that:

- No kind of ground equipment not already used for normal operations can be added
- Only a limited computing power is available
- No sensor or equipment can be added to the UAV

Those constraints are due to the fact that it is a commercial product and the cost or the complexity of the product cannot increase because of this safety feature. How those constraints bounded and guided the thesis will be stressed through the report.

1.3 Outline

In this first section the general context of the thesis has been defined along with an overview of the goals and constraints defining the project. In the next section the state of the art of inertial and vision based navigation will be presented in order to introduce the solution studied by this master's thesis. Then the mathematical details will be introduced, first with a section about the mathematical models and then with a section presenting the sensor fusion filter at the core of the proposed solution. Then the simulation environment and test protocols will be defined before going into a few tests and performance assessments. Finally, I will give my point on view on what needs to be done before having a fully operational and integrated solution.

Many of the results and data presented in this report will be given in percentage or arbitrary unit as they are confidential and cannot be made publicly available.

2. State of the Art

2.1 Navigation

Navigation is the science of "knowing where you are relative to where you want to be" as explained in the book Global Positioning Systems, Inertial Navigation, and Integration [5]. This book is giving a good overview on the state-of-the-art of inertial navigation and it divides navigation into 5 basic types:

1. Pilotage, knowing where you are by recognizing landmarks
2. Dead Reckoning, knowing where you are by knowing initial position and, at any time, heading and speed
3. Celestial Navigation, knowing where you are by using angles between celestial bodies and local verticals
4. Radio Navigation, knowing where you are by using radio signals sent by known stations
5. Inertial Navigation, knowing where you are by knowing initial Position Velocity Attitude Time (PVAT) vector and, at any time, attitude rates and accelerations

Type n°5 is the only method fully free of external references or devices. The second one could also be free of external references but not for aerial navigation because an external reference, such as GPS, is needed to measure the ground speed. But, as detailed further in the report, inertial navigation has its limitations and therefore a mix of several types of navigation is often used to get an accurate and reliable navigation system.

Inertial Navigation

A detailed explanation of inertial navigation can be found in [5]. Just a few characteristics and typical behaviors of inertial navigation necessary to understand

the rest of the report are described here. This description is only valid for a strapdown inertial measurement unit (IMU) using accelerometers and gyrometers.

Principle

An IMU is keeping track of the PVAT vector of a body in an inertial reference frame, i.e., a coordinate frame in which Newton's law of motion is valid [5]. Measuring the acceleration of the body in the inertial reference frame and integrating it once gives the speed and twice gives the position. To be able to do this integration the initial position and velocity need to be known.

In a strapdown IMU, the acceleration of the body, given by the sensors in the body reference frame, is computed in the inertial reference frame by using the attitude of the body. The attitude of the body is obtained by measuring and integrating the angular rotation speed of the body in its reference. To be able to do this integration the initial attitude needs to be known.

So the IMU is using two kinds of sensors: accelerometers and gyrometers. A flow chart summarizing the architecture of a strapdown IMU is available in Fig. 4.

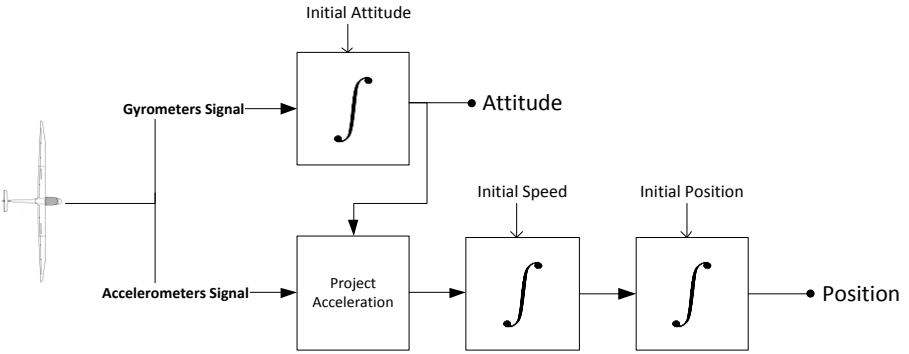


Fig. 4 - Strapdown IMU flowchart

Sensors

There are 3 orthogonally mounted accelerometers in an IMU. They measure the specific acceleration, each on one axis of the body frame, and as they are orthogonally mounted the specific acceleration vector of the body is obtained by summing the measurement of each accelerometer.

Moreover, there are three gyrometers orthogonally mounted and each of them gives the rotation rate around one axis of the body frame.

Error and Performances

An IMU is using multiple integrations and is therefore very sensitive to measurement noise. In the type of IMU used nowadays for aircraft navigation, laser gyros are used. They are very accurate but in an IMU the error is coming principally from gyrometer bias as explained in the technical report [6].

Description of the errors can be found in [5] and a model is derived in Section 3 of this report, but what is important here is that the velocity estimation error is bounded thanks to the Schuler oscillation phenomenon explained in [7]. For a good aircraft IMU a typical shape for the estimation error is a sinus oscillating at the Schuler frequency with a 2 m/s amplitude resulting in a drift in position estimation of around 2 nm/h as illustrated below.

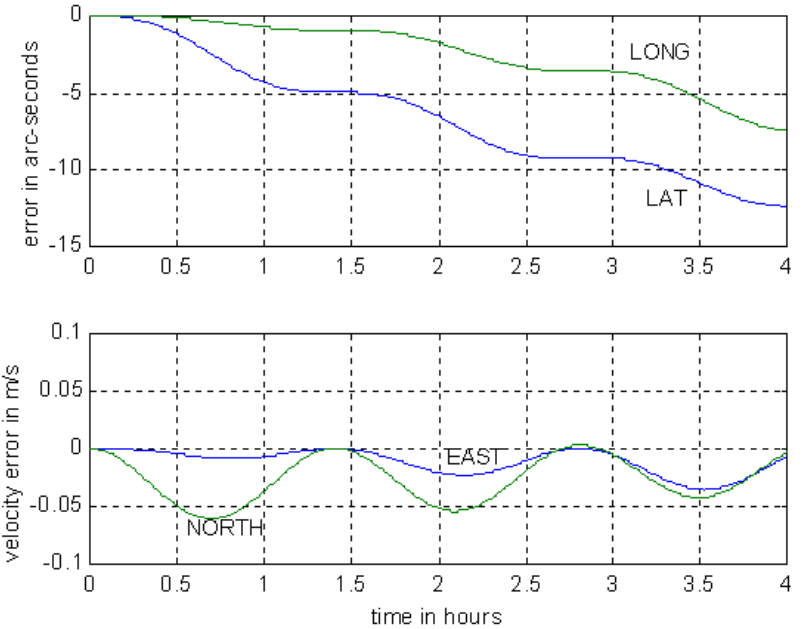


Fig. 5 - Drift in position (Longitude and Latitude) and error in speed (on North/East coordinates) of an IMU. Simulation done by GPSoftNav.com

As said before, a way to improve this accuracy is to combine IMU with another type of navigation. This combination gives a navigation system, mainly based on inertial measurements, called Inertial Navigation System (INS).

Sensor Fusion for Navigation

The most commonly used combination is the GPS/IMU, the high accuracy of the IMU during short period compensates the noisy GPS measurements and the drift of the IMU is compensated by the high accuracy of GPS over long period. The fusion of the IMU measurements with the GPS measurements is often done with a Kalman filter or one of its derivatives [8] [5]. But in the thesis's case, combination with the GPS is not possible since the UAV is in a scenario where GPS signal is not available. But another source of radio navigation or even landmarks can be considered and combined with the inertial measurements.

Directional Antenna

A directional antenna can give the azimuth and elevation of the aircraft to the antenna. By knowing the position of the antenna this gives information on the position of the aircraft. This information is not complete and only the azimuth and elevation are known but not the distance from the antenna to the drone. As illustrated in Fig. 6, the directional antenna can give the same measurement while the aircraft is at two different positions. So it is not very accurate, maybe sufficient to bring the aircraft close to the runway but clearly not enough to perform a landing. A way to improve accuracy and use this technique for landing is described in [8], a Thales patent, but it requires more equipment than available as it uses a locating device able to perform distance calculation.

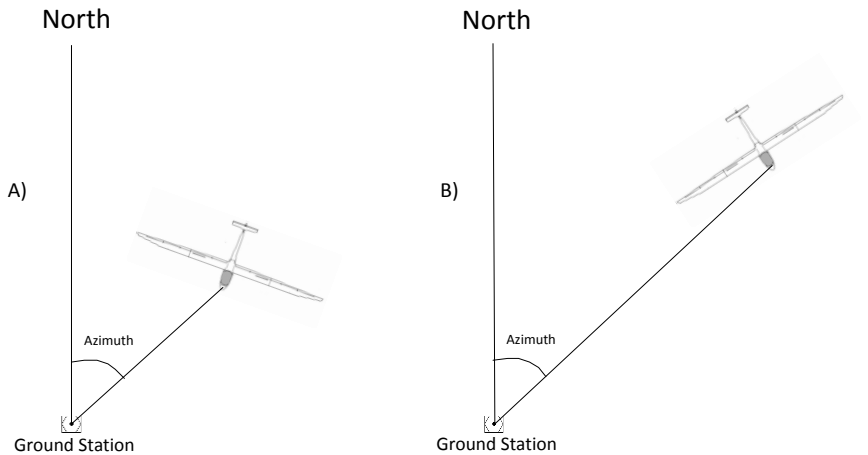


Fig. 6 - In situations A and B the directional antenna is giving the same information while the aircraft is clearly not at the same position

Image Aided Navigation

[9] is a very complete book giving an overview on all the methods studied so far to perform vision aided navigation and landing. Those methods fall into a few categories but many of them are only suitable for navigation in a known environment because they require a data base of images of the environment that the computer uses to find the position of the aircraft. In unknown environments it is possible to use image analysis in order to recognize the runway by comparing it to a stack of runway images for example as done in the research [10]. It requires a lot of computational resources and, it is therefore not suitable for our application since a very limited computational resource is available. So a new way of using vision, or a tradeoff, is to compensate the lack of information and computational resources with small and sparse inputs from a ground operator.

Terrain Matching

Terrain matching consists of using a map of an area and try to match what the aircraft is sensing with this map in order to know its position. An example of this approach using an elevation map and a height sensor is described in [11]. This is very efficient and widely used for high precision low height flying, but it requires a map in the data base of the UAV and is therefore not suitable since there is no such data base onboard SAGEM's UAV.

ILS and MLS

The Instrument Landing System (ILS) and the Microwave Landing System (MLS) are widely used (ILS mainly) to guide aircraft to the runway. Since the end of the '60s Airliners can perform auto landing using an ILS [12]. But the main drawback of such technique is that it requires equipment on the airport which is not compatible with the expressed need. VOR and other radio navigations techniques commonly used in aviation fall in the same category.

After studying all those methods it seems that the most suitable for the intended application are the hybridization using vision and using a directional antenna. Before giving an overview of the considered solution an overview of how the UAV is working when GPS is available will be presented.

Technique	Phase	Remark
Directional Antenna	Navigation	Can be considered
Directional Antenna + Ranging	Navigation & Landing	Requires non available equipment
Image aided without Operator help	Navigation & Landing	Requires high computation power, or complex data base and learning process
Image aided with Operator help	Navigation & Landing	Can be considered
Terrain Matching	Navigation & Landing	Requires terrain data base
ILS/MLS/VOR…	Navigation & Landing	Requires non available ground equipment

Table 1 - Summary of all the techniques found in the literature

2.2 Normal Operation Mode

Control Chain Architecture

The architecture of the UAV is typical and can be found in many UAVs. As any automatic vehicle and as an aerial vehicle, an UAV has many sensors. They are all connected to a flight controller (FC) which is responsible for calculating and sending the control commands to all the actuators. As said in the first section, one constraint of the project is to modify the FC as little as possible. Therefore the solution will be interfaced between the sensors and the FC as shown in Fig. 7. That is why it is necessary to first have a look at how the UAV is working when GPS is available.

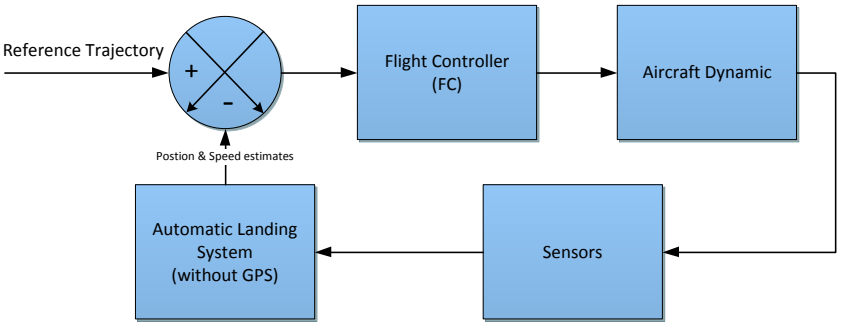


Fig. 7 – Simplified vision of the Control Chain Architecture, the Automatic Landing System is introduced between the sensors and the flight controller.

Sensors

The UAV is equipped with a complete set of sensors. All the sensor measurements are sent to the FC which is performing a sensor fusion in order to know the Position Velocity Attitude Time (PVAT) vector of the drone which is the base used by any FC to calculate the control commands. In this section each sensor is briefly described and more details will be found in the next section while deriving a mathematical model for each of them.

Air Data

As any aircraft, the UAV is equipped with air data sensors giving static and dynamic pressure. Air speed (CAS) and barometric altitude (AMSL) are

calculated from that. Air data are also sent to the navigation system which is using them in a sensor fusion algorithm.

Magnetic Heading

Again as any aircraft, the UAV is equipped with a compass. An electric compass in this case, which is giving the heading of the drone, i.e., the angle between the axis of the drone and the magnetic north.

Navigation system

The navigation system, as explained in Section 2.1, is using different measurements and performs sensor fusion in order to calculate the PVAT vector of the UAV. In normal operation mode the navigation system is using the IMU, the GPS and the altitude given by air data.

Laser Height Sensor

The laser height sensor is mounted under the body of the drone, facing toward the ground. It calculates the height, the altitude above the ground (AGL), of the drone using the laser ranging and the attitude information.

Communication with the Ground Station

In normal operation mode, the UAV is always connected to a ground station thanks to a high bandwidth wireless connection. The ground station is continuously receiving data from the UAV and displays an intuitive representation of them to the drone and payload operators.

The operators can send commands to the drone using controls available in the ground station. But none of the operators in the station know how to fly the drone: they are only able to provide very high-level orders. This has to be taken into account in the design of the algorithms: if an operator input is required it must be very basic and high level. The communication with the ground station is way more robust than the GPS signal and it is therefore considered that in case of GPS loss there is still a connection between the UAV and the GS.

Trajectories

It is very interesting to look at the kind of trajectories the UAV is following during different parts of the flight. The phases which are interesting for this thesis are the cruise phase, the approach phase, the landing and braking phases since they are the four phases in which the UAV will have to fly during a landing without GPS scenario.

En-Route Trajectory

An en-route trajectory is defined using waypoints. A waypoint is composed of :

- A geographical point, defined by its latitude and longitude
- An altitude, defined AMSL
- A speed
- A set of actions
- An order number

The flight controller (FC) is guiding the UAV straight from waypoint to waypoint respecting the target velocity and climbing/descending linearly between two waypoints in order to be at the required altitude when passing a waypoint. This is a very simplified description but there is no need to go into the details for this report.

Approach Trajectory

The goal of the approach trajectory is to bring the UAV aligned with the runway at a certain distance of the planned touch down point (TDP). It is still using the waypoint concept but those waypoints are respecting constraints to be sure that they will lead the UAV to the desired position at the desired speed and aligned with the runway.

Landing Trajectory

The landing trajectory is the one followed by the drone from the end of the approach trajectory to the TDP. This trajectory is not composed of waypoints. From an horizontal point of view it is a straight line aligned with the middle of the runway. And from the vertical point of view it has a vertical profile which is compatible with a landing: if the slope is too steep the UAV will gain too much speed to be able to land and if it is too shallow there is a high risk of collision with obstacles. Moreover, it cannot be a straight trajectory as the UAV must decrease its vertical speed before touching down in order to preserve the landing gear and avoid bouncing. An illustration of a landing trajectory can be seen in the figure below.

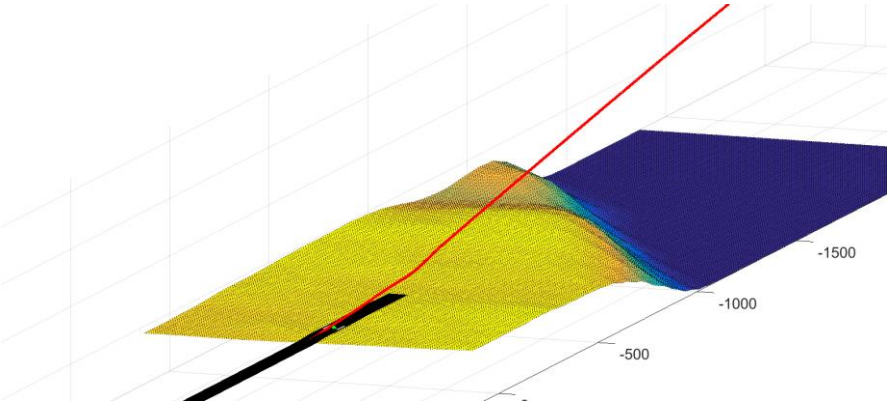


Fig. 8 - 3D plot of a typical landing trajectory at Le-Havre airport, the explanation on how the terrain was modeled can be found in Section 3.2.

Braking Trajectory

The braking phase starts once the UAV is on the runway and ends once the UAV is rolling at a suitable speed for taxiing. During this phase the UAV must follow the center line of the runway.

GPS Loss

In the scenario studied during the thesis, the GPS loss occurs while following an en-route trajectory. In order to land, the UAV has to fly an en-route trajectory to join an approach trajectory followed by a landing trajectory and the landing will be considered as successful when the UAV will stop, still on the runway, after the braking phase.

But, due to the GPS loss, the navigation system is not able anymore to perform sensor fusion and is therefore sending drifting position and speed information to the FC. So the FC is fed with very unreliable information and cannot guide the drone along the expected trajectories. That is why a system, which is estimating, without GPS signal, the drone's position and velocity and sending it to the FC, is suggested in order to land safely in case of GPS loss. The next subsection provides more details about the suggested solution.

2.3 Suggested Solution

As already mentioned the global solution for the automatic landing without GPS problem is based on the existing control algorithm and existing sensors which already enable the UAV to land automatically when the GPS signal is available.

Only the navigation algorithms are changed. Instead of using the inertial/GPS sensor fusion presented before, the navigation algorithm is using an inertial/radio measurement sensor fusion and an inertial/vision sensor fusion depending on the phase. Indeed, the landing has been decomposed in four phases.

Four Phases

During the different four phases the sensors available are different. The four phases are:

- Return
- Final
- Short Final
- Braking

This decomposition is very close to the trajectory decomposition seen in Section 2.2. The only difference is that the landing trajectory is decomposed here in two phases: the final and the short final phases. This is due to vision sensor considerations. Each phase will be described: when it starts, when it stops and what the navigation strategy is.

Return

The return phase is starting when the Automatic Landing System (ALS) engages. It is not clear yet what will be the conditions required to enter this mode, but the GPS loss will be at the origin. The goal of this phase is to navigate along the return trajectory. That means that it ends when the UAV is at a certain distance of the runway, aligned with the runway. This phase is the longest of the four phases, thus the one during which the IMU drift will have the largest value, but it is also the one with the lowest requirement in positioning accuracy.

During this phase the navigation algorithm will use the azimuth from the UAV to the ground station (GS) as described in Section 2.1. This information is sent to the UAV by the GS itself thanks to the directional antenna mounted on it and used for data transmission. As seen in Section 2.1 this radio equipment/IMU sensor fusion is not complete and permits only a good estimation of the position along the axis perpendicular to the GS/UAV axis. So in case of GPS loss the return trajectory is modified to do a complete turn around the GS, this enables to drastically improve the positon and speed estimate as explained later on. An example of a return trajectory is shown in Fig. 9.

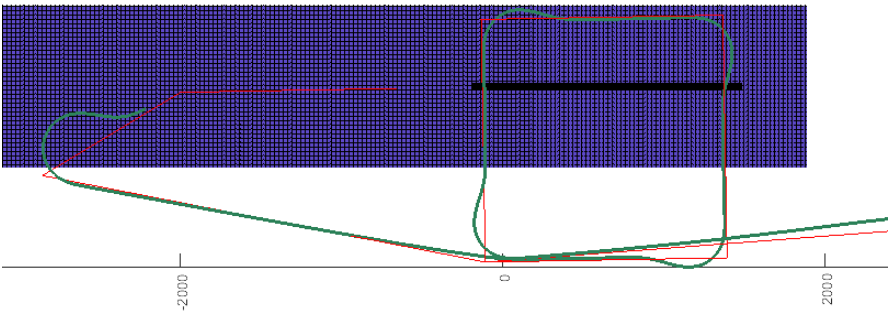


Fig. 9 - In red the reference trajectory, the turn around the GS (a square) can be seen. In green the actual trajectory followed by the UAV during a closed loop simulation, the green track stops when the UAV switches to the final phase

Once the turn around the GS is completed, the UAV is joining the axis of the runway at the desired distance in order to start the next phase.

Final

During final and short final the UAV is following the landing trajectory which is exactly the same in the GPS loss degraded mode and in the normal operation mode. The final phase is ending a few meters before the planned touch down point (TDP). During this phase the navigation is done using vision/IMU/laser fusion.

A computer vision algorithm is tracking the TDP in the image of an "all weather camera" (AWC). But as stated in Section 2.1, an input from the operator is required in order to compensate the lack of data and computing power. The operator has to point on his screen where the desired touch down point is in the camera view. It must be repeated two or three times because as the UAV is getting closer to the runway a better accuracy can be achieved. This action is very simple thanks to an ergonomic user interface. The view of the operator can be seen in Fig. 10. The vision algorithm is keeping track of this designated point. This algorithm is not in the scope of this thesis and has been provided by another SAGEM department.

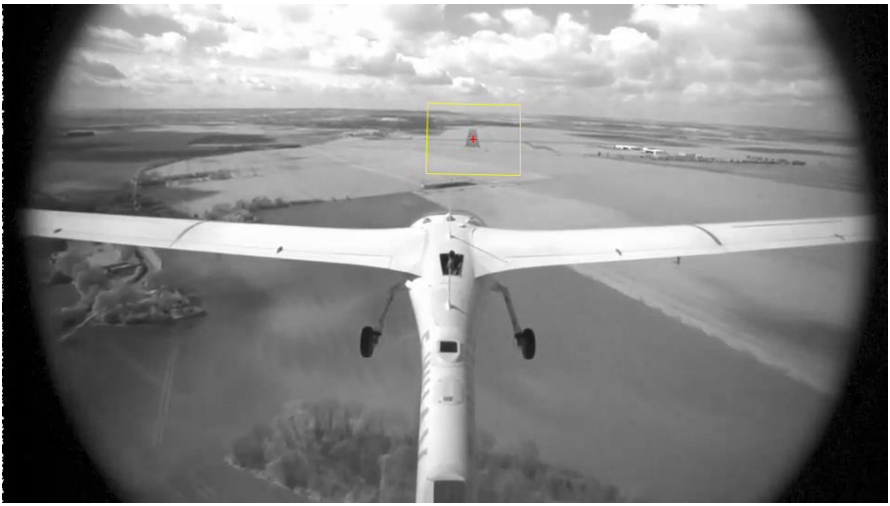


Fig. 10 - Tracking Algorithm view, the red cross is the tracked point, the TDP, and the yellow square delimits the image analysis area

Once the UAV gets very close to the runway this vision algorithm is not able to keep track of the TDP anymore so the final phase stops and the UAV is switching to the short final phase which is just using IMU and will drift freely as described in Section 2.1. So the final phase is critical and requires a high accuracy in both position and ground speed (GS) estimates as it will determine how accurately the TDP will be reached.

Short Final

As already said, during this phase, navigation is done only by using inertial measurements that means that it is subject to the drift of the IMU. This phase is very short, it lasts only a few seconds, it ends when the landing gears touch the ground. The accuracy of this phase is only dependent on the accuracy of the position and speed estimate at the end of the final if the drift of the IMU over a few seconds is considered as negligible.

Braking

Finally, once the UAV is on the ground it must brake and stop, staying on the runway. This phase is still unclear but it might use wheel encoder/IMU/magnetic heading sensor fusion to follow the center line of the runway.

Sensor Fusion

As just seen, during all the phases, sensor fusion is used. The sensor fusion is done in all the cases using an extended Kalman filter [19] [20]. The extended Kalman filter has been selected due to the non-linearity of the system. The filter will be described in Section 4. The sensors used for each phase are summarized in Table 2.

Phase	Sensors
Return	IMU Barometer Directional Antenna
Final	IMU Barometer Laser Camera
Short Final	IMU Laser
Braking	IMU Compass Wheel Encoder

Table 2 - Phases and sensors available

The context, the problem and the solution being globally defined, the next section will give more details and start with the mathematical model of the problem.

3. Mathematical Model

3.1 Aircraft and Environment

Many coordinate systems are used in this report. All of them are introduced here along with the relationship between them. A more complete description and some calculation details can be found in [13].

Earth coordinate system

Four kinds of coordinate system linked to the earth have been considered. They are used for navigation, control, flight dynamic simulation and data analysis.

ECEF

The first one is named ECEF, for earth-centered earth-fixed. This classical Cartesian coordinate system is spinning with the earth. Its axis can be seen in the figure below.

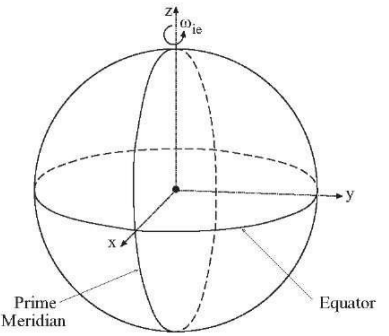


Fig. 11 - ECEF axes

A vector expressed in this coordinate system will have the subscript $_e$.

WGS 84

The second one is the World Geodetic System 1984 (WGS84) [14]. It is the coordinate system used by the GPS. This coordinate system is earth-centered and uses earth-fixed axes, the same as the ECEF. It is based on parameters used to model Earth's size, shape, and gravity [16]. More information on how WGS84's axes are defined can be found in [16] [15].

The Navigation System of the UAV is using this coordinate system. This is the reference frame as defined in Section 2.1. But as it is earth-fixed it is not really an inertial reference frame, therefore the IMU is performing some corrections, which will not be described here, to compensate for rotation of the earth and gravity. It is not a classical Cartesian coordinate system [13], a point is described with two angles (latitude, longitude) and one distance (altitude). Coordinates expressed in this coordinate system will have the subscript $_g$.

For a point P with the coordinates $P_e = (X, Y, Z)_e$ in ECEF and $P_g = (\lambda, \varphi, h)_g$ in WGS84 the relationships are:

ECEF→WGS84

$$\lambda = \arctan\left(\frac{Y}{X}\right)$$
$$\varphi = \arctan\left(\frac{Z + e'^2 b sin^3(\theta)}{p - e^2 a \cdot cos^3(\theta)}\right) \quad (1)$$
$$h = \frac{p}{\cos(\varphi)} - N$$

WGS84→ECEF

$$X = (N + h)\cos(\varphi)\cos(\lambda)$$
$$Y = (N + h)\cos(\varphi)\sin(\lambda) \quad (2)$$
$$Z = \left(\frac{a^2}{b^2}N + h\right)\sin(\varphi)$$

with:

$$p = \sqrt{X^2 + Y^2}$$
$$\theta = \arctan\left(\frac{Z\ a}{p\ b}\right)$$
$$a = 6378137\ m$$
$$b = 6356752.31424518\ m$$
$$e = \sqrt{\frac{a^2 - b^2}{a^2}}$$
$$e' = \sqrt{\frac{a^2 - b^2}{b^2}}$$
$$N = \frac{a}{\sqrt{1 - e^2 sin^2(\varphi)}}$$

NED

A North East Down (NED) coordinate system is centered on a point at the earth's surface, it is earth-fixed. It uses the WGS84 ellipsoid model as the WGS84 coordinate system. A vector expressed in this coordinate system will have the subscript $_n$. This coordinate system is very convenient for landing and navigation in local area. The TDP can be used as the center for example.

Considering a TPD with the coordinates P_e^{ref} in ECEF and $P_g^{ref} = \left(\lambda_{ref}, \varphi_{ref}, h_{ref}\right)_g$ in WGS84. For a point P with the coordinates $P_e = (X, Y, Z)_e$ in ECEF and $P_n = (X, Y, Z)_n$ in NED, the relationships are:

ECEF→NED

$$P_n = R_{n/e}\left(P_e - P_e^{ref}\right) \quad (3)$$

NED→ECEF

$$P_e = R_{n/e}^{-1} P_n + P_e^{ref} \quad (4)$$

with:

$$R_{n/e} = \begin{bmatrix} -\sin(\varphi_{ref})\cos(\lambda_{ref}) & -\sin(\varphi_{ref})\sin(\lambda_{ref}) & \cos() \\ -\sin(\lambda_{ref}) & \cos(\lambda_{ref}) & 0 \\ -\cos(\varphi_{ref})\cos(\lambda_{ref}) & -\cos(\varphi_{ref})\sin(\lambda_{ref}) & -\sin() \end{bmatrix}$$

The ECEF, WGS84 and NED coordinate systems are summarized in the picture below.

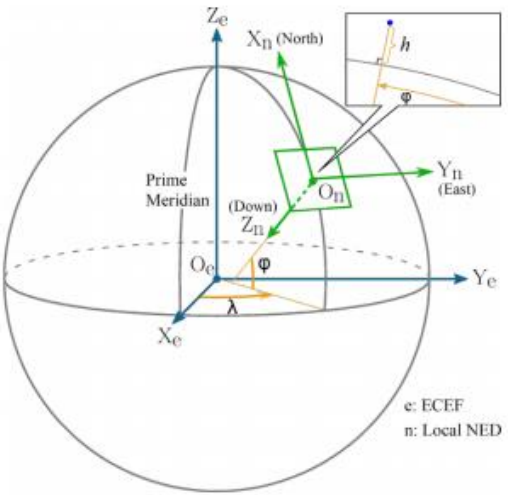


Fig. 12 - ECEF, WGS84 and NED coordinate [14]

Runway

The last coordinate system, which will be mainly used for results analysis, is obtained by rotation, around Z_n axis, of a NED coordinate system: it is centered at a TDP and has Y axis aligned with the runway as illustrated by Fig. 13. A vector expressed in this coordinate system will have the subscript r.

Considering a runway with the orientation θ as illustrated in Fig. 13. For a point P with the coordinates $P_r = (X, Y, Z)_r$ in runway coordinate and $P_n = (X, Y, Z)_n$ in NED, the relationships are:

Runway→NED

$$P_n = R_{n/r} P_r \quad (5)$$

NED→Runway

$$P_r = R_{n/r}^{-1} P_n \quad (6)$$

with:

$$R_{n/r} = \begin{bmatrix} \sin(\theta) & \cos(\theta) & 0 \\ \cos(\theta) & -sin(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

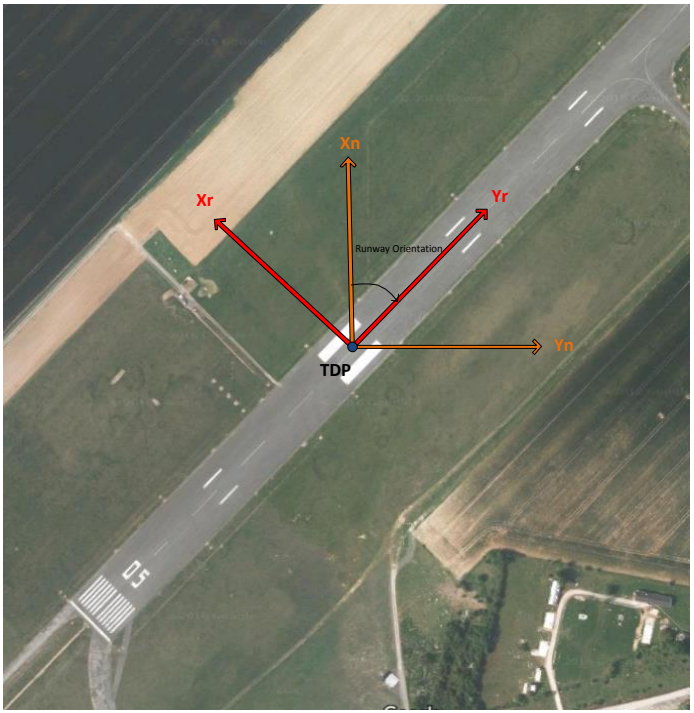


Fig. 13 - NED and Runway coordinate systems

Aircraft coordinate system

In the aircraft each sensor has its own reference frame, but for this report only two are of interest: the one used by the navigation system and the one used by the camera.

Body Reference Frame

There is a reference coordinate system named the body reference frame (BRF). The center is the center of gravity of the aircraft and the axes of this reference frame are the one used by the navigation system to describe the attitude of the plane, they are defined relative to geometric considerations [13]. This is the body reference frame defined in Section 2.1.

The axes of the body coordinate system can be obtained from the ECEF or NED axes by three successive rotations. The three rotation angles, known as Euler angles, are the pitch, roll and yaw angles and are illustrated by Fig. 14. More information about Euler angles can be found in [14]. One thing important is that those three rotations are not commutative so an order which will be used every time had to be defined:

$$\text{Yaw} \rightarrow \text{Pitch} \rightarrow \text{Roll}$$
$$(\psi, \theta, \varphi) = (Yaw, Pitch, Roll)$$

A vector expressed in this coordinate system will have the subscript $_b$.

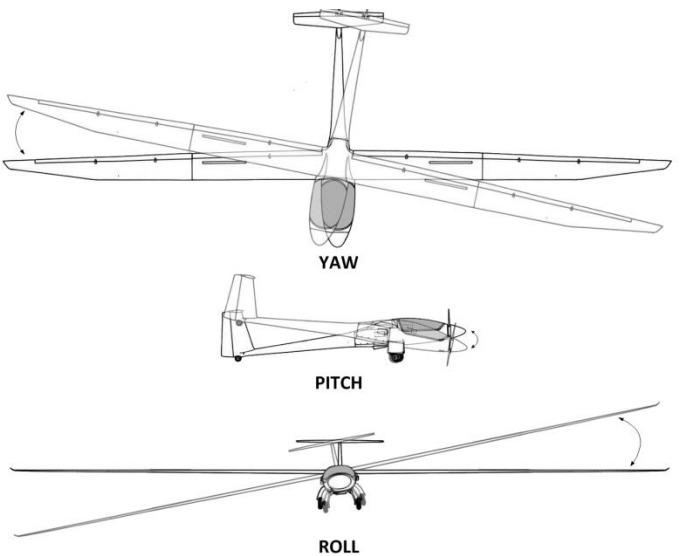


Fig. 14 - Attitude angles illustration

Considering an UAV with the position P_n^{body} in NED and the attitude (ψ, θ, φ). For a point P with the coordinates $P_b = (X, Y, Z)_b$ in the body coordinates and $P_n = (X, Y, Z)_n$ in NED, the relationships are:

NED→BRF

$$P_b = R_{b/n}\left(P_n - P_n^{body}\right) \tag{7}$$

BRF→NED

$$P_n = R_{b/e}^{-1} P_b + P_n^{body} \tag{8}$$

with:

$$R_{b/n} = R_\psi R_\theta R_\varphi$$

$$R_\psi = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$R_\varphi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{bmatrix}$$

Camera

The camera is mounted on the plane and has its own coordinate system. It can be obtained by translation and rotation of the body reference. A vector expressed in this coordinate system will have the subscript c.

Considering a camera with the position P_b^{camera} in the body coodrinate and the mounting angles (ψ, θ, φ). For a point P with the coordinates $P_b = (X, Y, Z)_b$ in the body coordinates and $P_c = (X, Y, Z)_c$ in the camera coordinates, the relationships are:

BRF→Camera

$$P_c = R_{c/b}\left(P_b - P_b^{camera}\right) \quad (9)$$

Camera→BRF

$$P_b = R_{c/b}^{-1} P_c + P_b^{camera} \quad (10)$$

with:

$$R_{c/b} = R_\psi R_\theta R_\varphi$$

$$R_\psi = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$R_\varphi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{bmatrix}$$

3.2 Sensors

In this subsection the model used for each sensor are described. These models are the one used by the simulator that will be described in Section 5. All these models are parametric.

IMU

The IMU has been described in Section 2.1, and as said in this section the main source of error is the bias of the gyro. So a discrete-time model for a time step dt and considering the gyro drift has been derived:

$$\epsilon\varphi_{k+1} = \epsilon\varphi_k + \frac{\epsilon V_k}{R_T}.dt + D \cdot dt$$

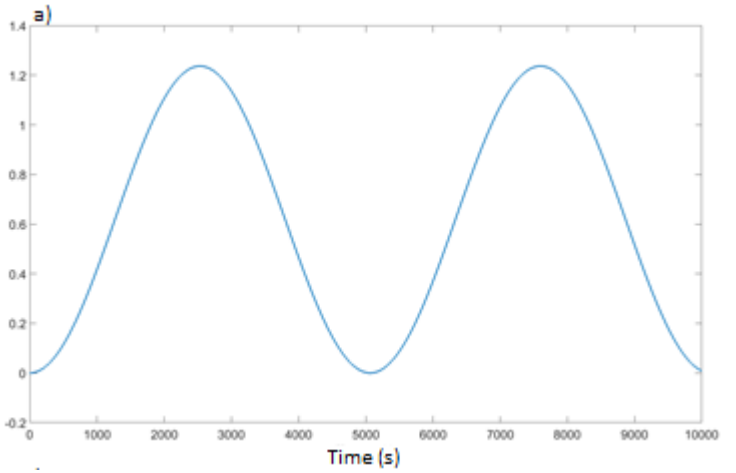
$$\epsilon V_{k+1} = \epsilon V_k - \epsilon\varphi_{k+1} g \cdot dt \quad (11)$$

$$\epsilon P_{k+1} = \epsilon P_k + \epsilon V_{k+1} \cdot dt$$

with:

$\epsilon\varphi_k$ The attitude error at step k
ϵV_k The speed error at step k in NE
ϵP_k The position error at step k in NED
R_T Earth's radius
$D = \begin{pmatrix} \cos(\psi) & -sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix} \cdot \begin{pmatrix} D_X \\ D_Y \end{pmatrix}$
ψ UAV heading
D_X Gyro drift on X_b
D_Y Gyro drift on Y_b

As shown in Fig. 15, this model gives the expected behavior, i.e., a drift in position and an oscillation in speed. The simulation was done for a standstill UAV, so a perfect IMU would have given a speed equal to 0 m/s and a still position.



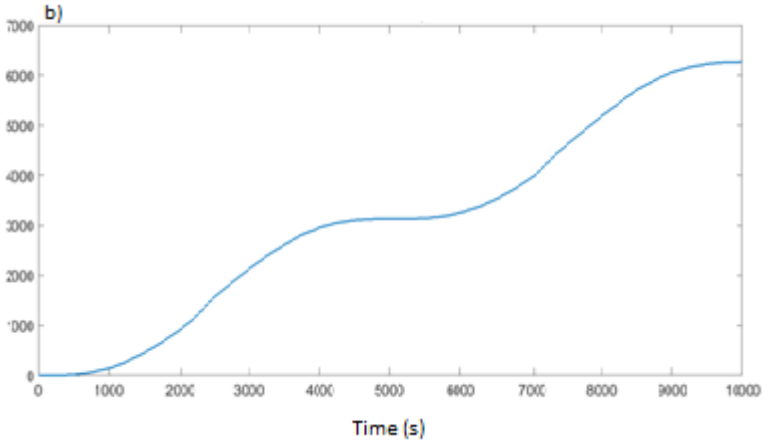


Fig. 15 - a) Speed error and b) Position drift along X_b axis with the IMU model

Laser Height Sensor

The model for the laser height sensor is very basic and soon-coming data from flight tests will help to derive a more realistic model. For now the laser height sensor measurement H_k^{Laser} is calculated from the real height of the aircraft with a bias (b) and a white noise (n). The bias is very small and is given by the manufacturer as being of a few centimeters. Moreover the laser height sensor features a low pass filter filtering obstacles (trees, houses…), therefore the model implements a time delay (τ) and the terrain model will be considered as free of obstacles.

$$H_k^{Laser} = H_{k-\tau}^{AGL} + b + \mathrm{n}_k \tag{12}$$

Height and altitude are not to be mistaken. The difference between height and altitude is the ground elevation as illustrated by Fig. 16. How the ground elevation is represented in the simulator will be explained in Section 5.

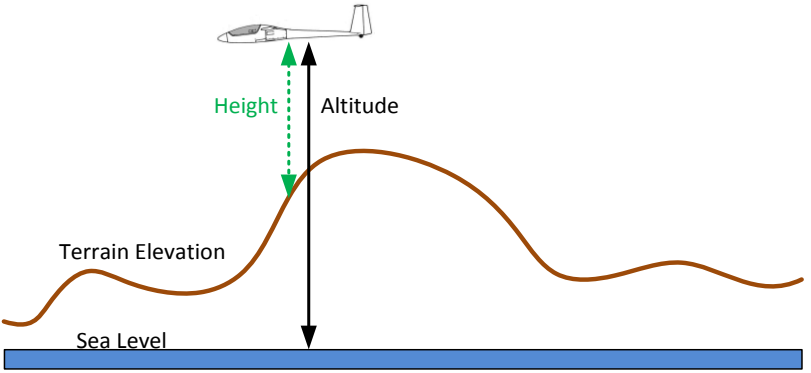


Fig. 16 - Altitude vs Height

Barometer

A very basic model has been selected for the barometer, returning the measured altitude at time step k H_k^{Baro}. Actually it is a model for the baro-inertial altitude measurement, i.e., the altitude measurement established by a sensor fusion between the barometer and the IMU measurements. The altimeter is simply considered as returning the true altitude of the UAV affected by a scale factor (sf) and a bias (b). The bias represents an error in the altimeter reference (QNH) [16]. The scale factor models the error due to the difference between the real atmosphere and the model used to calibrate the barometer (ISA model) [16]. In a real barometer there is also an important time delay but this time delay is almost canceled thanks to the baro-inertial sensor fusion. Only a fraction of this time delay remains (τ) and is taken into account in the model.

$$H_k^{Baro} = Z_{k-\tau}^{NED} + b + sf \cdot Z_{k-\tau}^{NED} \quad (13)$$

Camera

The classic pinhole model is used for the camera. Education paper [17] describes this model and how the position of a real object in the camera image can be calculated.

A point P with the coordinate $P_c = (X, Y, Z)_c$ in the camera coordinate has for coordinates in the picture (with top left corner as (0,0)):

$$\left(\frac{Bearing + \frac{FOV_H}{2}}{IFOV_H}, \frac{Elevation + \frac{FOV_v}{2}}{IFOV_v} \right) \quad (14)$$

with:

$D = |(\mathrm{X}, \mathrm{Y}, \mathrm{Z})\mathrm{c}|$
$Bearing = atan2(Y, X)$
$Elevation = \mathrm{asin}\left(\frac{Z}{D}\right)$
FOV The field of view
$IFOV$ The field of view per pixel

Harmonization error and time delay introduced by the camera are also taken into account.

Computer Vision

The computer vision algorithm is not a sensor in itself but it will be presented here as a sensor returning the position $P^{TDP}_{Tracking\ k}$of the tracker in the picture. The model for the computer vision takes into account the time delay (τ) introduced by image compression and image processing, the error (n) of the tracking algorithm considered as a white noise and the error in the tracked position considered as a bias (b) on the real position of the tracked point.

$$P^{TDP}_{Tracking\ k} = \mathrm{P}^{TDP}_{Picture\ k-\tau} + \mathrm{b}_k + \mathrm{n}_k \quad (15)$$

Directional Antenna

The model of the directional antenna is calculating the measured azimuth α_k from the real azimuth α^{real}considering the error in the harmonization of the antenna,

i.e., a bias (b), the noise (n) in the tracking modeled as a white noise and a delay (τ).

$$\alpha_k = \alpha_{k-\tau}^{real} + b + n_k \quad (16)$$

Sensors Frequency

Every sensor is working at its own frequency. These different frequencies are part of the models and all sensors are considered to run at a fixed frequency.

Sensor	Frequency
IMU Attitude	100 Hz
IMU Other	50 Hz
Baro	50 Hz
Laser	10 Hz
Directional Antenna	2 Hz
Camera	25 Hz

Table 3 - Range of order of the different frequencies

3.3 Aircraft Dynamics

This model describes how the aircraft is moving. It includes both the dynamics of the aircraft and the guidance done by the Flight Controller (FC). The hypotheses made are that:

- The aircraft is following the roll and the vertical speed commands sent by the FC
- The ground speed is constant
- The plane is at equilibrium on the pitch axis

This is a very coarse approximation but it will not have any impact on the behavior of the filter. Nevertheless, in order to assess the global performance of the system, in closed loop simulation, a better model will be required. The equations of this model can be found in Appendix A.

The real algorithm of the FC will be used when the aircraft's dynamic model will be improved but so far a basic autopilot is implemented in the simulator.

4. The Filter

4.1 Mathematical Formulation

The filter is used to estimate the position and velocity errors of the IMU. Then the IMU measurement is corrected of this error and sent to the FC. That means that the filter is not working on states but on state errors, which is typical for navigation filter as explained in the book [6]. The selected filter is an Extended Kalman Filter (EKF) because of the non-linearity of the system [19] [20].

State-Space Representation

The states considered are the error made by the IMU in the position $(X, Y, Z)_n$ in NED and the error made by the IMU in the horizontal speed $(VN, VE)_n$ also in the NED. As the Z value calculated by the IMU is coming from a sensor fusion between the barometric and the inertial measurements, there is no need of considering the error on the vertical velocity.

$$x = \begin{bmatrix} \epsilon X \\ \epsilon Y \\ \epsilon Z \\ \epsilon VN \\ \epsilon VE \end{bmatrix} = \begin{bmatrix} X^{IMU} \\ Y^{IMU} \\ Z^{IMU} \\ VN^{IMU} \\ VE^{IMU} \end{bmatrix} - \begin{bmatrix} X^{real} \\ Y^{real} \\ Z^{real} \\ VN^{real} \\ VE^{real} \end{bmatrix}$$

$$\begin{bmatrix} \epsilon X \\ \epsilon Y \\ \epsilon Z \\ \epsilon VN \\ \epsilon VE \end{bmatrix} = \begin{bmatrix} IMU\ Position\ Measurement\ Error\ on\ X_n\ axis \\ IMU\ Position\ Measurement\ Error\ on\ Y_n\ axis \\ IMU\ Position\ Measurement\ Error\ on\ Z_n\ axis \\ IMU\ Speed\ Measurement\ Error\ on\ X_n\ axis \\ IMU\ Speed\ Measurement\ Error\ on\ Y_n\ axis \end{bmatrix} \quad (17)$$

A discrete-time nonlinear stochastic model has been derived:

$$x_{k+1|k} = F x_k + v_k$$
$$y_k = h(x_k) + e_k$$
$$F = \begin{bmatrix} 1 & 0 & 0 & dt & 0 \\ 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

with:

x_k State Vector at step k
$x_{k+1|k}$ State Vector at step $k+1$
v_k Process Noise at step k
dt The time step

In this model the speed estimation error is considered as constant, which is not the case as seen in Section 2.1. Therefore the speed estimation error variation will be included as a noise, but it is not a Gaussian noise, so the EKF filter will not be optimal and this will be a limit of the solution.

The propagation of the filter is done as explained in book [19]

$$\hat{x}_{k+1|k} = F \hat{x}_{k|k}$$
$$\hat{x}_{k|k} = x_{k|k-1} + K_k \left(y_k - h\left(\hat{x}_{k|k-1}\right) \right)$$
$$K_k = P_{k|k-1} H_k^T (H_k . P_{k|k-1} H_k^T + R)^{-1}$$
$$P_{k|k} = (I - K_k H_k) P_{k|\,k-1}$$
$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k \quad (19)$$
$$H_k = \left(\frac{\delta h(x)}{\delta x} \right)_{x = \hat{x}_{k|k-1}}$$

with R and h_k being different for each phase, they are given in the next paragraphs.

Process Noise Matrix

The process noise is considered uncorrelated and therefore the process noise matrix is diagonal:

$$Q_k = \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 \\ 0 & 0 & 0 & 0 & q_5 \end{bmatrix}$$

Return

During the return phase, only the azimuth from the directional antenna is available.

$$h_k(x) = \frac{1}{D_{Est_k}} \begin{bmatrix} -X^{est} \\ -Y^{est} \end{bmatrix}_k$$

$$\begin{bmatrix} X^{est} \\ Y^{est} \end{bmatrix}_k = \begin{bmatrix} X^{IMU} \\ Y^{IMU} \end{bmatrix}_k - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} x - \begin{pmatrix} X^{GS} \\ Y^{GS} \end{pmatrix}_n \quad (20)$$

$$D_{Est_k} = \sqrt{X^{est^2} + Y^{est^2}}$$

with $X^{GS}$and Y^{GS} the coordinates of the ground station in the NED coordinate system centered at the TDP. And the sensor noise matrix associated to this measurement is:

$$R^{return} = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}$$

Final

During the final the vision is always available and the laser height measurement is available once the UAV gets close enough to the ground.

Without Laser

$$h_k(x) = f\left(\frac{1}{D_{Est_k}} - \begin{bmatrix} X^{est} \\ Y^{est} \\ Z^{est} \end{bmatrix}_k\right)$$

$$\begin{pmatrix} X^{est} \\ Y^{est} \\ Z^{est} \end{pmatrix} = \begin{pmatrix} X^{INS} \\ Y^{INS} \\ Z^{INS} \end{pmatrix}_k - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} x \tag{21}$$

$$D_{Est} = \sqrt{X^{est\,2} + Y^{est\,2} + Z^{est\,2}}$$

with f the relationship between NED and camera coordinates as seen in Section 3.2.

The sensor noise matrix associated with this measurement is:

$$R^{Final} = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}$$

With Laser

$$h_k(x) = \begin{bmatrix} f\left(\frac{1}{D_{Est_k}} - \begin{bmatrix} X^{est} \\ Y^{est} \\ Z^{est} \end{bmatrix}_k\right) \\ Z^{est} \end{bmatrix}$$

$$\begin{bmatrix} X^{est} \\ Y^{est} \\ Z^{est} \end{bmatrix} = \begin{bmatrix} X^{INS} \\ Y^{INS} \\ Z^{INS} \end{bmatrix}_k - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} x \tag{22}$$

$$D_{Est} = \sqrt{X^{est\,2} + Y^{est\,2} + Z^{est\,2}}$$

with $f$the relationship between NED and camera coordinates as seen in Section 3.2.

The sensor noise matrix associated with this measurement is:

$$R^{Final/Laser} = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix}$$

Short Final

On short final the filter runs without any measurement so:

$$h_k(x) = 0 \tag{23}$$

The sensor noise matrix associated with this measurement is:

$$R^{Short\ Final} = [0]$$

Braking

The behavior of the filter during this phase is still to be determined.

Standby Mode

The standby mode has not been mentioned yet, it is the mode in which the filter is running when the UAV is working in normal operation mode. As the GPS is available it is used to estimate in real time the state of the filter so when the GPS is lost the filter starts with a good estimate and initial covariance matrix. In this mode:

$$H = I(5,5) \tag{24}$$

The sensor noise matrix R associated with this measurement is given by the GPS module.

4.2 Implementation

MatLab & Real System

The filter has been implemented under MatLab Simulink for testing purpose. It is implemented in Simulink using a MatLab function.

On the real system, the code of the filter will have to run on the FC. One way of generating it could have been to use MatLab code generation feature, but it is not very efficient when a MatLab function is used in a Simulink model. Another possibility is to write the code again, being careful to the real time constraints inherent to this filter. The code is quite short so this solution is conceivable. This is out of the scope of this thesis and will be mentioned in the future work section.

Sensors Frequency

As exposed in Section 3.2, each sensor is working at its own frequency. The filter is running at the frequency of the IMU Position and Speed data and every time another measurement arrives it is taken into account by the filter with the next IMU attitude data. There is no synchronization between the sensors. Except for the IMU attitude data and computer vision data, they are synchronized with timestamps. The impact of the desynchronization is exposed in Section 6.4.

Transition between Phases

The transition between two phases must be smooth and must occur at the right moment. The first transition is triggered when the GPS loss is confirmed. The filter is leaving the standby mode and starts to take into account the measurements from the directional antenna. At the beginning of the return phase the estimation of the error of the IMU is very good since it was done using the GPS signal. And the initial covariance for this phase is taken as the last covariance matrix calculated in standby mode.

The return phase stops when the UAV passes the last waypoint of the return trajectory. The filter starts to take into account the vision measurement and does not use the directional measurement any more. This transition is done without any change in the covariance matrix.

The next transition is when the laser height sensor is activated. At this point either an increase of the covariance on the altitude axis or a reset of the altitude estimate is necessary. Both have been tested and in case of steep terrain the

increase of the covariance matrix works better but it is the opposite when the terrain is almost flat. A mix between the two techniques is a good compromise.

The criteria for the last two transitions are still to be determined.

4.3 Tuning

Calculation of initial parameters

A first estimate of a suitable set of parameters for the filter was calculated using the specification of the sensors. As they are confidential they are given arbitrarily scaled. The hand tuned parameters will be scaled the same way in order to make comparison possible but unfortunately, due to the scaling, a R and Q value comparison cannot be done.

$$Q_{theo} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2E^{-11} & 0 \\ 0 & 0 & 0 & 0 & 2E^{-11} \end{bmatrix}$$

$$R_{theo}^{return} = \begin{bmatrix} 5E^{-4} & 0 \\ 0 & 5E^{-4} \end{bmatrix}$$

$$R_{theo}^{Final} = \begin{bmatrix} 5.8E^{-9} & 0 \\ 0 & 5.8E^{-9} \end{bmatrix}$$

$$R_{theo}^{Final/Laser} = \begin{bmatrix} 29E^{-9} & 0 & 0 \\ 0 & 29E^{-9} & 0 \\ 0 & 0 & 2E^{-5} \end{bmatrix}$$

Hand Tuning

From this initial set of parameters the parameters were tuned by doing many simulations. It led to a new set of parameters reported here using the same scaling as for the previous values:

$$Q_{tuned} = 2.25 Q_{theo}$$

$$R_{tuned}^{Return} = 400 R_{theo}^{return}$$

$$R_{tuned}^{Final} = 100 R_{theo}^{Final}$$

$$R_{tuned}^{Final/Laser} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 400 \end{bmatrix} R_{theo}^{Final/Laser}$$

One can notice the increase in the noise on the sensor process matrices. This was expected as the noise models in the EKF are Gaussian which is not the case of the noise in the system. But the increase is way larger than expected. For the laser sensor it can be explained by the fact that the theoretical calculation was done taking into account the sensor accuracy only, and not the error introduced by the terrain elevation.

Another step of tuning will be necessary once the code is implemented on the real process.

5. Simulation Environment

5.1 Architecture

The simulator is implemented in MatLab using the Simulink environment. It has 4 main parts as shown in Fig. 17:

- Aircraft Dynamics
- Sensor Models
- Filter
- Data Log

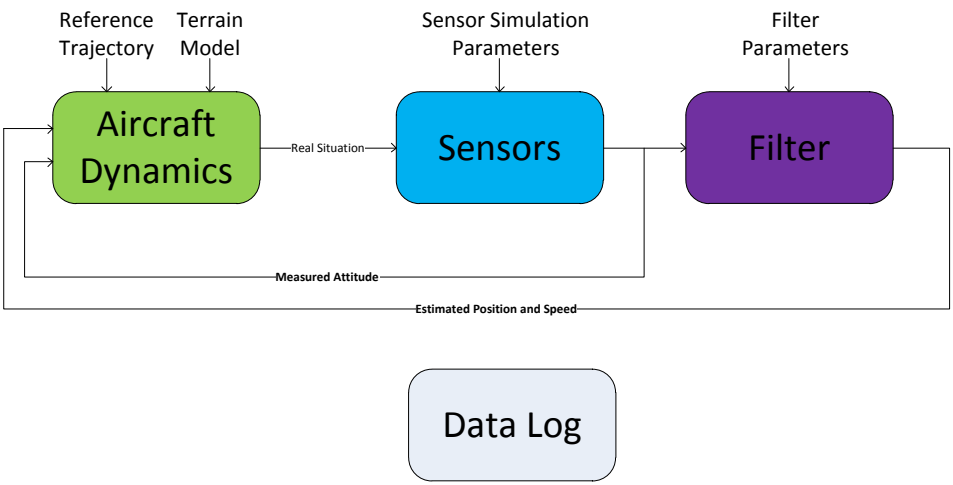


Fig. 17 - Simulator Architecture

Aircraft Dynamics

The aircraft dynamics block computes the control commands according to Section 3.3 and it simulates the flight dynamics of the aircraft according to the equations in Section 3.3.

The terrain model taken as input by this block is used to calculate the real height of the aircraft. The terrain is modeled by a grid of points at which the ground elevation is known, a linear interpolation is used to calculate the ground elevation at the UAV's position. The grid of elevation can be made either using real data from Google API, or by using formulas to generate a particular terrain. It will be used in Section 6.1 to analyze the impact of the terrain on the behavior of the filter.

Sensor Models

The sensor models block includes all the sensor models presented in Section 3.2 except the computer vision part which is included in the filter block because the filter block is the part containing everything that will have to be implemented on the real UAV.

Filter

The filter is the part estimating the IMU error. Then it corrects the IMU measurement with the estimated error and sends the measurements to the Aircraft Dynamic block where the FC will compare it to the reference trajectory in order to calculate the control commands.

Data Log

The data log block is simply gathering and saving information for post simulation analysis.

5.2 Open-Loop Simulation

It is also possible to do open-loop simulation. Data from real flights can be sent to the simulator and used instead of the models. The simulator needs at least the real position/speed of the aircraft to run in open loop. It is possible to add data from the computer vision algorithm run on a real video, the directional antenna data, the IMU data, the laser height and barometric measurements.

5.3 Monte Carlo

As the simulator has many parameters, it is possible to run Monte Carlo simulations in order to assess the filter's performance. For each simulation a set of parameters is randomly chosen according to a normal or uniform law. This permits to distribute all the parameters associated with the sensor models seen in Section 3.2.

6. Performance Study

In this section are presented the results of all the performances studies done during this thesis. This section ends by a global conclusion on the performances of the filter.

The different studies are:

- Impact of the terrain elevation
- Impact of the angular harmonization of the camera
- Impact of the position harmonization of the camera
- Impact of latencies and desynchronization
- Performance of the "Fly Back"

6.1 Terrain Elevation

The previous thesis done on the "landing without GPS" subject only considered a flat terrain. This particular case showed the interest of using the laser height sensor in order to compensate for the bias of the altimeter. The purpose of this subsection is to study if it is still the case for non-flat terrains and what the impact of the terrain elevation on the filter is.

Data, Parameters and Protocol

For this study, 4 different terrains, plotted in Fig. 18 and Fig. 19, were used:

- A flat terrain
- A terrain with a 3% slope until 300m before the TDP and then flat (yellow plot)
- A terrain with a 20m cliff 300m before the TDP (blue plot)
- A terrain with a 15m x 10m obstacle 300m before the TDP (orange plot).

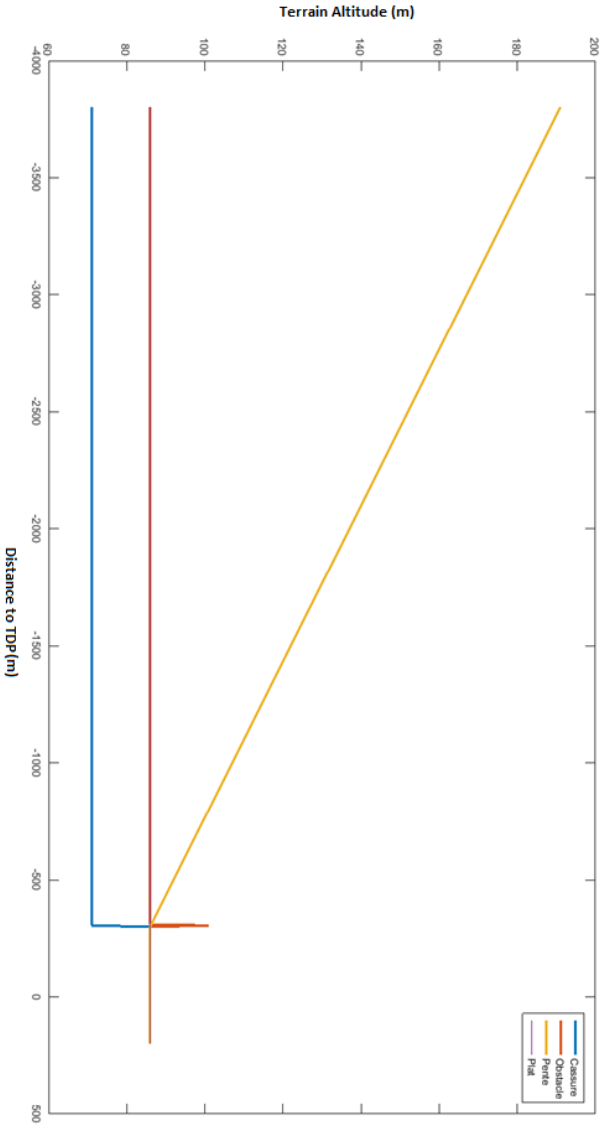


Fig. 18 - Vertical cut along runway's axis. The TDP is at 89m AMSL

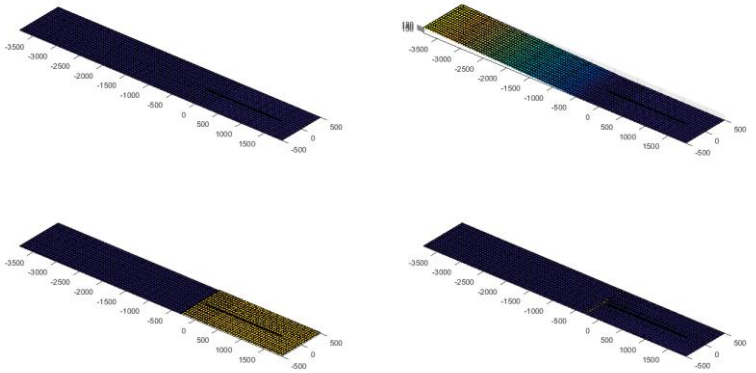


Fig. 19 - 3D plot of the 4 terrains

Flight data are issued from a real flight for which a video recording on which the tracking algorithm can be run is available. For each terrain the laser measurements are simulated according to the model presented in Section 3.2 and the four terrain models.

For each terrain 1500 Monte-Carlo simulations were launched, randomly choosing the following parameters:

IMU Parameters	
Parameter	**Law**
Gyro Bias X (°/h)	Normal
Gyro Bias Y (°/h)	Normal
Time since last reset	Uniform
Roll Bias (°)	Normal
Pitch Bias (°)	Normal
Heading Bias (°)	Normal
Altimeters Parameters	
Parameter	**Law**
Baro Bias (m)	Normal
Laser Bias (m)	Normal
Baro Scale Factor (m)	Normal
Laser Noise (m)	Normal
Laser Activation Height (m)	Normal

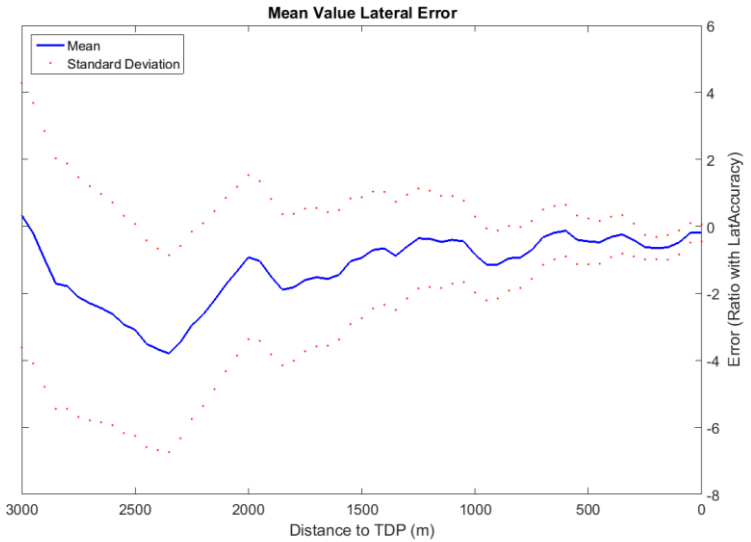
Vision Parameters (Harmonization Error)	
Parameter	**Law**
Rotation X Sensor (°)	Normal
Rotation Y Sensor (°)	Normal
Rotation Z Sensor (°)	Normal
Filter's Initial Parameters*	
Parameter	**Law**
Erreur en X NED (m)	Normal
Erreur en Y NED (m)	Normal

*The initial estimate errors are a bit optimistic, especially on the speed but that will not affect the qualitative conclusion of this section.

The mean value and standard deviation used cannot be given here as they would give confidential information on the sensors used by SAGEM on its UAV.

Filter Reference Performance

The reference for the study is the performance of the filter without using the laser height sensor.



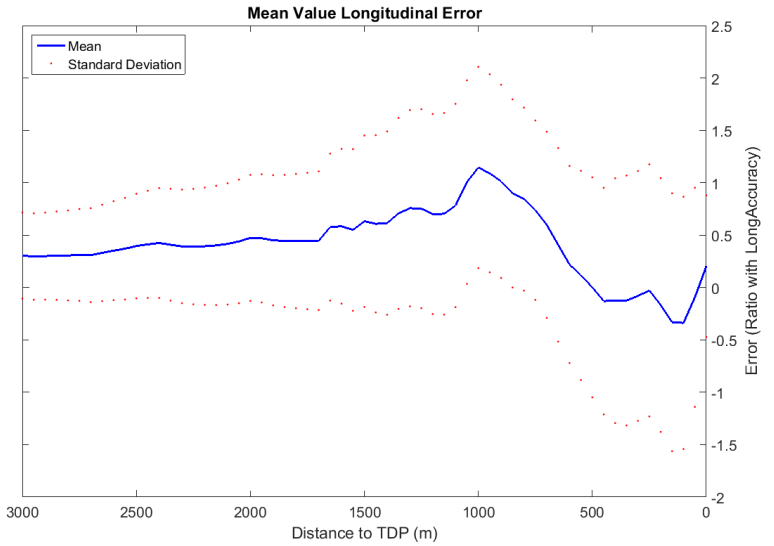


Fig. 20 - Result of the Monte Carlo Simulations for the reference scenario

Diagrams in Fig. 20 are to be read from left to right: the simulations start 3 000m away from the TDP and end at the TDP. The first plot shows the mean value, over 1500 simulations, of the error between the lateral position of the UAV (axis X_r) and the actual lateral position of the UAV. The second plot shows the same error along the runway axis (Y_r). So this is the plot of the mean error of the filter made on the position estimation projected on the runway coordinates.

This Monte Carlo simulation is the reference and the performance will be compared using Table 4, looking only at the longitudinal axis because the impact of the terrain elevation is focused on this axis:

Distance to TDP	Mean Error	Standard Deviation	% Valid Simulations
300m	-0.12	1.19	63 %
150m	-0.17	1.21	56 %
100m	-0.33	1.23	47 %

Table 4 - This table summarizes the mean error, the standard deviation of the error and the percentage of simulations within the limit $LongAccuracy$, at different distances from the TDP, along the runway axis.

Measurement after 100m are not taken into account because it is approximately the distance at which the system will switch to the short final mode, so only the performance up to 100m before the TDP are interesting here.

It is also interesting to look at the histogram of the errors, for example in Fig. 21 with the initial distribution of the longitudinal error.

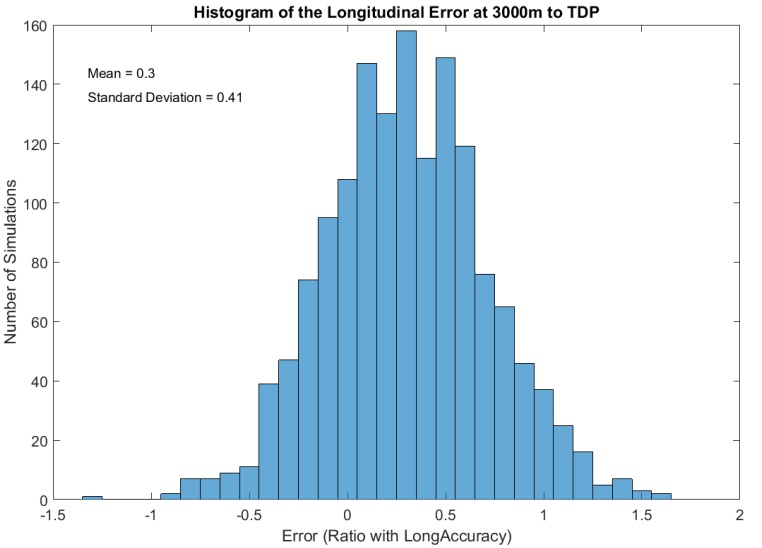


Fig. 21 - Histogram of the longitudinal error at 3000m to the TDP

In order to improve the performance of the filter, it is first of all necessary to understand the impact of an altitude estimation error.

Impact of an Altitude Estimation Error

On the longitudinal axis (Y_r), the filter is using the height above the TDP and the elevation with which the TDP is viewed by the camera. In the case of an error in the height estimation (barometer error, non-flat terrain…) the filter will look for the best way to see the TDP with the measured elevation at the measured height as illustrated by Fig. 22.

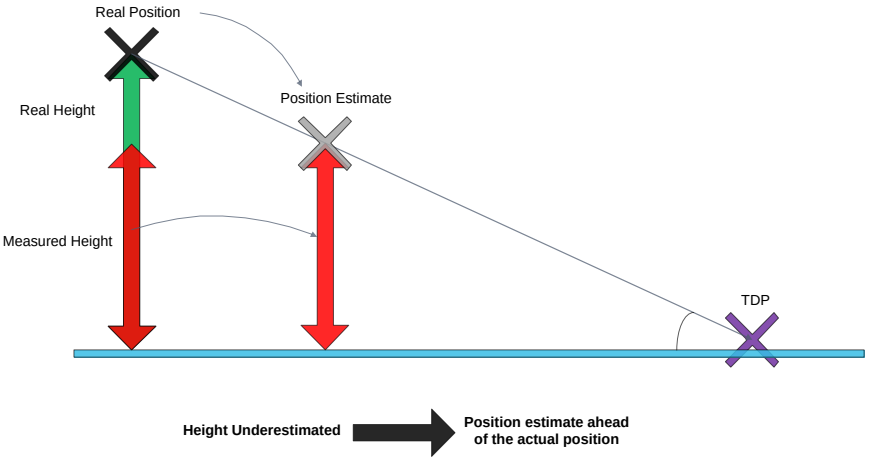


Fig. 22 - Illustration of height underestimation consequence

So according to Fig. 22, when the height of the UAV is underestimated, the position estimate is ahead of the actual position along the longitudinal axis (Y_r). The same way, if the height of the UAV is overestimated then the position estimate is behind the actual position.

So a good height estimate is mandatory for a good behavior of the filter. Therefore the bias of the barometer (actually baro-inertial) must be compensated and that is what the laser height sensor is used for.

One must also look at what happens when both the height and the elevation measurements are correct but the estimated position is false. This can happen if the filter previously received measurements with too much noise to make a good estimation. The analysis is always made for the actual position of the UAV at a fixed distance to the TDP. Open and closed loop must be distinguished because in open loop the real position of the UAV is on the approach slope while in closed loop the estimated position is on the approach slope (open-loop data come from a real flight when real position, acquired with GPS, were provided to the FC while in closed loop the estimated position is provided to the FC).

.

Open Loop

In the open-loop case, Fig. 23 illustrates what happens when the position estimate is ahead of the actual position.

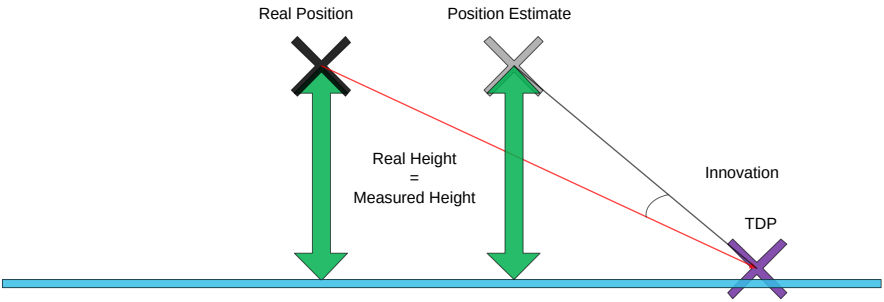


Fig. 23 - In open loop the real position is on the glide slope (red)

And Fig. 24 illustrates what happens when the position estimate is behind the actual position, considering that it is behind of the same distance as it was ahead in the previous case:

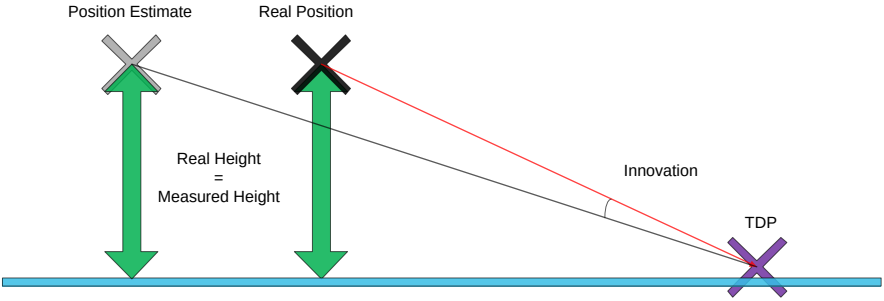


Fig. 24 - In open loop the real position is on glide slope (red). The distance between the real and actual position is the same as in the previous case and the real position is at the same distance to TDP than in the previous case

So for the same estimation error in norm, if the estimate is ahead then the innovation is larger. This can be confirmed by the plot in Fig. 25 showing the innovation vs the estimation error (calculated geometrically in the case of a perfect guidance along a 6° slope).

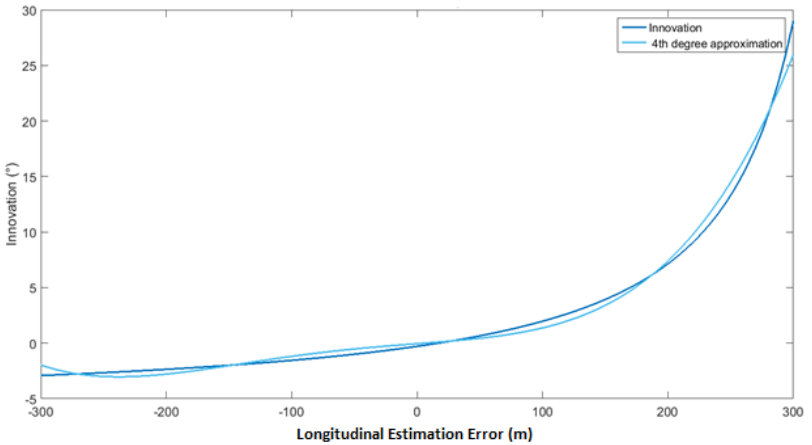


Fig. 25 - Innovation as a function of the longitudinal estimation error at 350 to the TDP

It is clearly not linear. Even if an EKF is used, this non-linearity is too strong to disappear. So in open loop an estimate that is ahead of the actual position will be easier to correct for the filter. This phenomenon will be stressed in the simulations results.

Closed Loop

In closed loop, the situation when the estimate is ahead of the actual position is illustrated by Fig. 26.

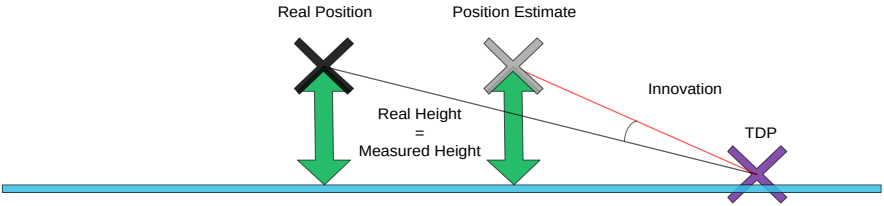


Fig. 26 - In closed loop the position estimate is on the glide slope (red)

And Fig. 27 illustrates the situation when the estimate is behind the actual position, considering the real position as being at the same position as previously (because the plots in this report are showing in abscise the actual distance to the TDP and not the estimated distance to the TDP) and considering that the estimation error has the same norm as for the previous case:

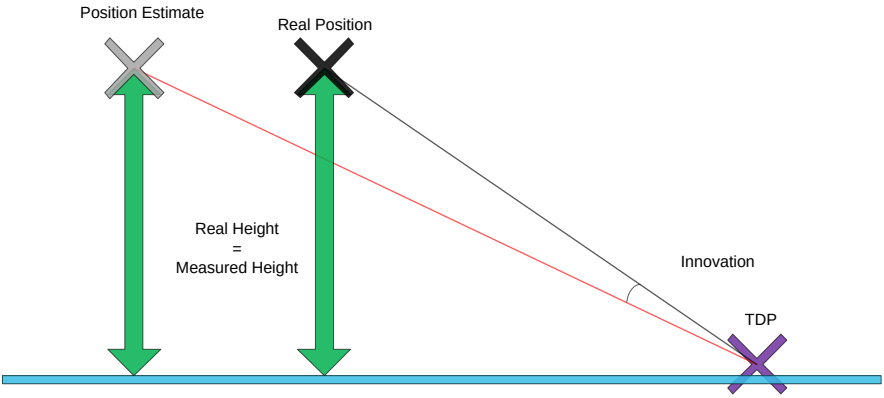


Fig. 27 - In closed loop the position estimate is on the glide slope (red). The real position is at the same distance to the TDP than in the previous case and the distance between the position estimate and the real position is also the same

So the phenomenon seen in open loop seems absent in closed loop, the innovation has the same norm in both cases. This can be confirmed by the same plot as before, Fig. 28, which is now linear:

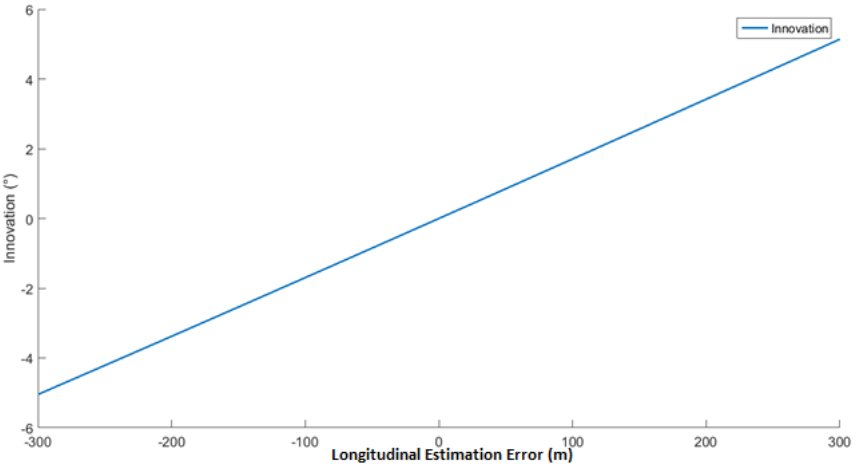


Fig. 28 - Innovation as function of the longitudinal estimation error at 350 to the TDP

Adding Laser Height Measurement

The conclusion of the previous Master's thesis was that adding the laser height measurement, in the flat terrain case, enables the filter to compensate for the barometer bias and therefore improves performances. It must first of all be verified. Result from the Monte Carlo simulations can be seen below and Table 5 does the comparison with the reference case.

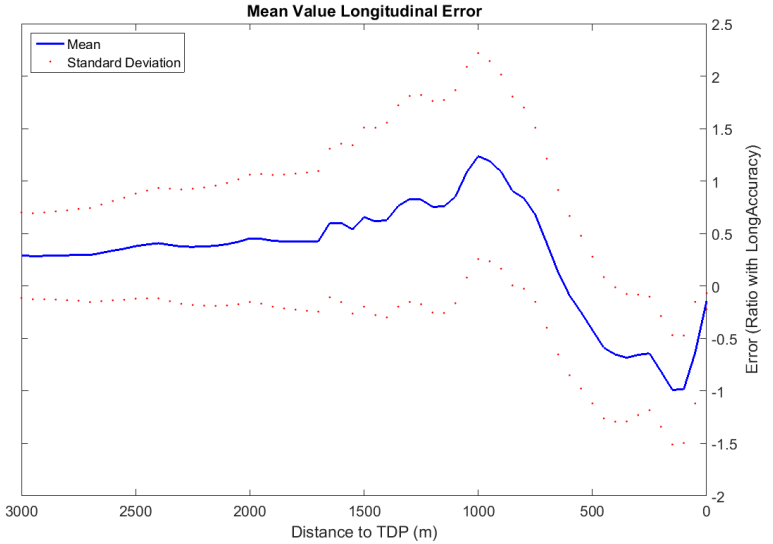


Fig. 29 - Result of the Monte Carlo Simulations

Distance to TDP	Mean Error	Standard Deviation	% Valid Simulations |var.
300m	-0.68	0.61	73 % |+ 10 %
150m	-0.81	0.53	70 % |+ 14 %
100m	-0.99	0.52	59 % |+ 12 %

Table 5 - This table summarizes the mean error, the standard deviation of the error and the percentage of simulations within the *LongAccuracy* limit (with the evolution from the reference case), at different distances from the TDP, along the runway axis.

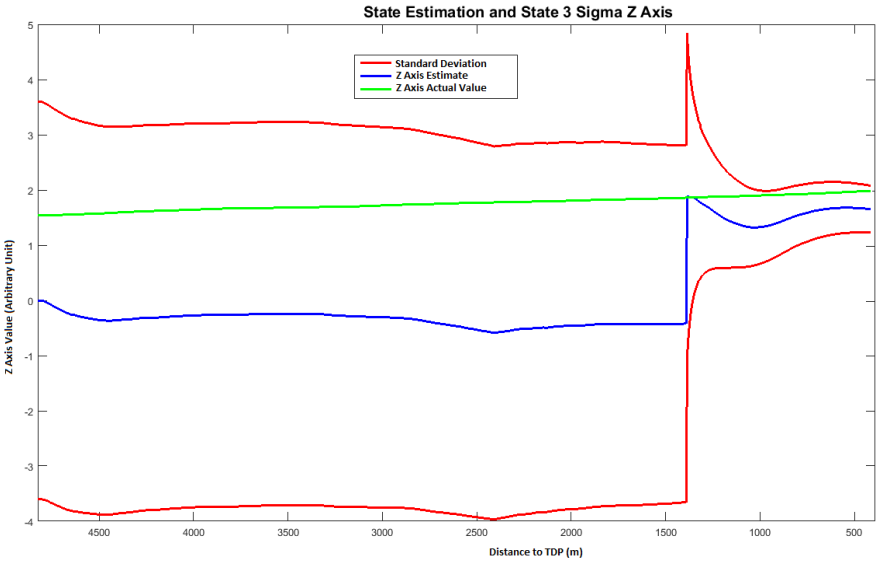


Fig. 30 - Estimation on Z axis for one simulation. The laser activation occurs around 1000m to the TDP and we can see the effect it has on the state estimation

So the laser height sensor activation is, as expected, improving by more than 10% the performance in the flat terrain scenario. This can be explained by the fact that it greatly improves the height estimation as it can be seen in Fig. 30 where the state estimation, the covariance of the filter for this state and the actual state are plotted.

The negative mean value of the error in Table 5 can be explained by the fact explained in the beginning of this subsection: estimates ahead of the actual position are easier to correct than estimates behind. This can be seen on the histogram Fig. 31: 100m away from the TDP the estimates in advance (positive error) were corrected by the filter.

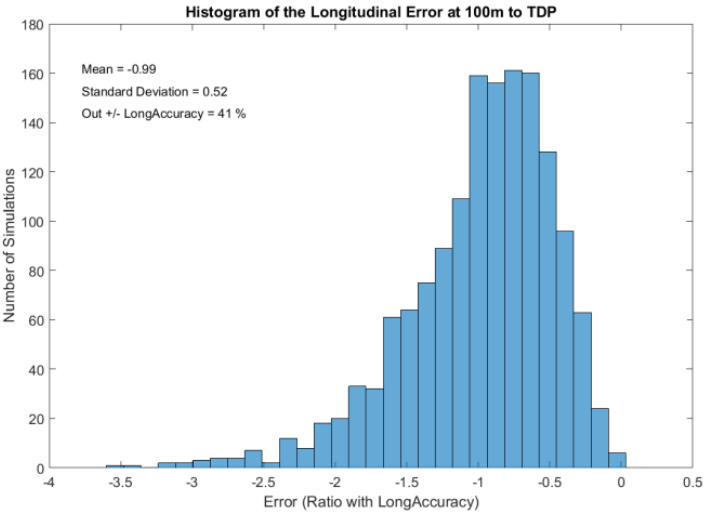


Fig. 31 – The right part of the histogram is cut because the simulations with a positive error have been corrected by the filter.

But if this phenomenon was the only one, an increase in the rate of simulations in the +/-*LongAccuracy* limit is expected as the UAV gets closer to the TDP. But it is not the case because as seen in Table 5, it goes from 73% at 300m to 59% at 100m. So added to the phenomenon described in the beginning of this subsection, there is something else, which already appeared in the reference case, but will not appear in closed loop. One way of explaining it is that the open-loop data are not very accurate, especially the altitude data. There is probably a bias in the altitude considered as the "real altitude" that introduces an error with what is seen in the video by the computer vision algorithm. The estimate seems to be behind the actual position. So as seen in the beginning of this subsection, this may be caused by an overestimation of the altitude.

To check this hypothesis some simulations applying a correction of a few meters to the altitude data were run. This was done later in the project and at this point different parameters were used for the filter. The results are reported in the table below.

Distance to TDP	% Valid Simulations Initial/Corrected
300m	83 % / 80 %
150m	80 % / 95 %
100m	74 % / 96 %

Table 6 - Table comparing the performance with the initial altitude data and with a 4m correction

Thanks to the small correction the expected behavior is obtained: the percentage of simulations within the limits increases as the UAV gets closer to the TDP. Moreover the new parameters suggested for the filter greatly improve the performances.

So it is confirmed that in the case of a flat terrain the laser height sensor greatly improves the performance of the filter, but is it still the case for non-flat terrains?

Obstacle

The results of the simulations done on the obstacle terrain are presented in Fig. 32 and Table 7.

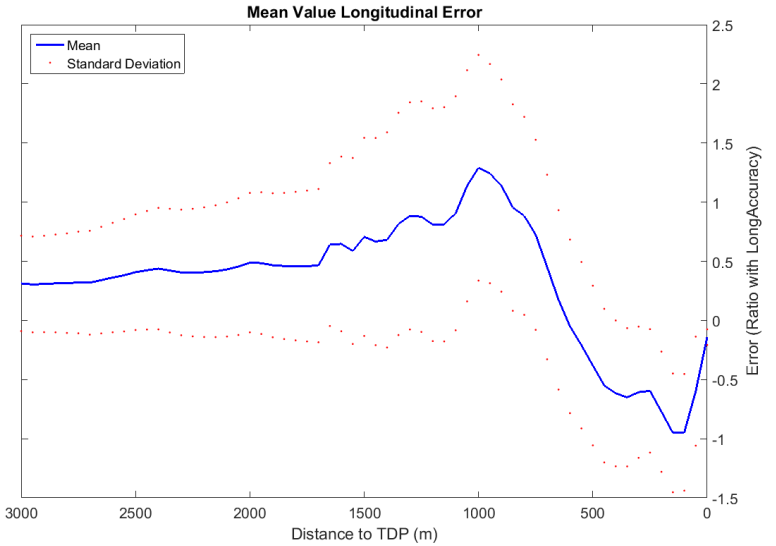


Fig. 32 - Result of 1500 Monte Carlo Simulations

Distance to TDP	Mean Error	Standard Deviation	% Valid Simulations |var.
300m	-0.65	0.58	77 %| + 14 %
150m	-0.77	0.51	72 %| + 16 %
100m	-0.95	0.50	60 %| + 13%

Table 7 - This table summarizes the mean error, the standard deviation of the error and the percentage of simulations within the *LongAccuracy* limit (with the evolution from the reference case), at different distances from the TDP, along the runway axis

This obstacle leads to a very small improvement in the performance compared to the flat terrain with laser sensor case (~2%). This can be explained by the fact that above the obstacle the laser height sensor is underestimating the height of the UAV as it can be seen on Fig. 33. This will, as seen previously in this subsection, lead to an estimate ahead of the real position and that is easier to correct for the filter.

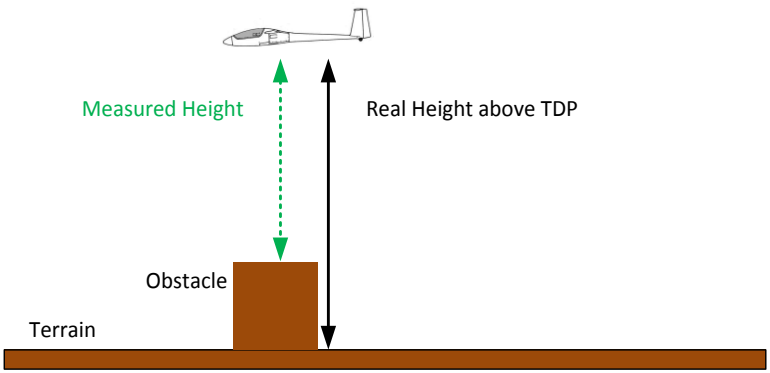


Fig. 33 - The obstacle leads to an underestimated height

The conclusion is that the impact of the obstacle is negligible.

Slope

The results of the simulations done on the slope terrain are presented in Fig. 34 and Table 8.

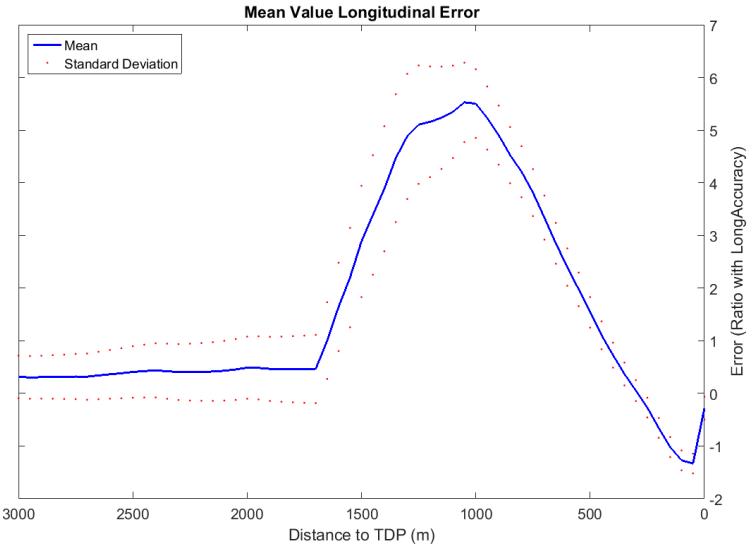


Fig. 34 - Results of 1500 Monte Carlo Simulations

Distance to TDP	Mean Error	Standard Deviation	% Valid Simulations |var.
300m	0.37	0.22	100 % | + 37 %
150m	-0.66	0.19	95 % |+ 39 %
100m	-1.02	0.19	49 % | + 2 %

Table 8 - This table summarizes the mean error, the standard deviation of the error and the percentage of simulations within the *LongAccuracy* limit (with the evolution from the reference case), at different distances from the TDP, along the runway axis

The slope has a huge impact on the longitudinal axis. The increase in the number of valid simulation is only due to the phenomenon explained in the beginning of this subsection. Moreover, the maximum value of the error is close to 5.5 (that means 5.5 times the expected accuracy). This will imply large trajectory corrections and that is not acceptable. In view of this result the relevance of the laser height sensor is in doubt. It may be possible to activate the laser sensor later during the approach once the UAV gets closer to the TDP because normally close to the TDP the terrain is flat and free of obstacles.

Cliff

The results of the simulations done on the cliff terrain are presented in Fig. 34 and Table 9.

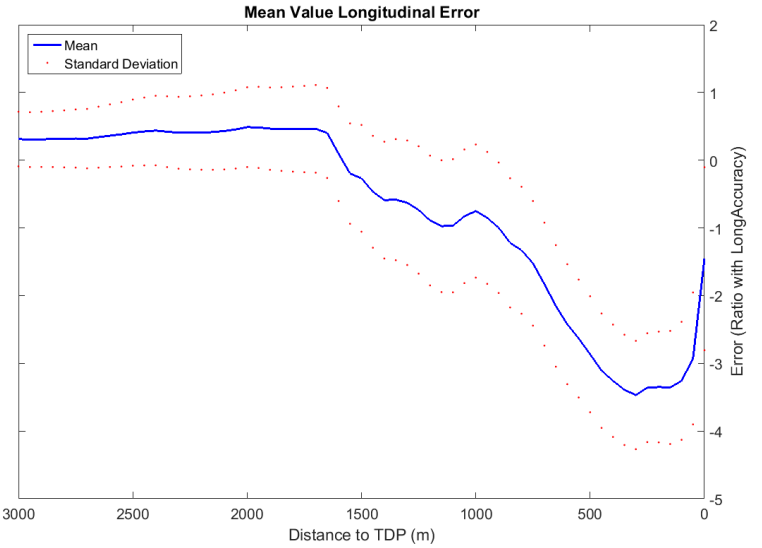


Fig. 35 - Results of 1500 Monte Carlo Simulations

Distance to TDP	Mean Error	Standard Deviation	% Valid Simulations |var.
300m	-3.39	0.81	0 %| - 63 %
150m	-3.35	0.82	0 %| - 56 %
100m	-3.36	0.84	0 %| - 47 %

Table 9 - This table summarizes the mean error, the standard deviation of the error and the percentage of simulations within the *LongAccuracy* limit (with the evolution from the reference case), at different distances from the TDP, along the runway axis

The cliff scenario is the one with the largest impact on the filter, more than the slope. The relevance of the laser sensor activation is even more in doubt.

The effect of the cliff is that the laser height sensor overestimates the height of the UAV as it can be seen on Fig. 36. As explained in the beginning of this subsection, this leads to an estimate behind the actual position and that is harder to correct for the filter. So a reversed cliff should give better results. It has been verified through simulations.

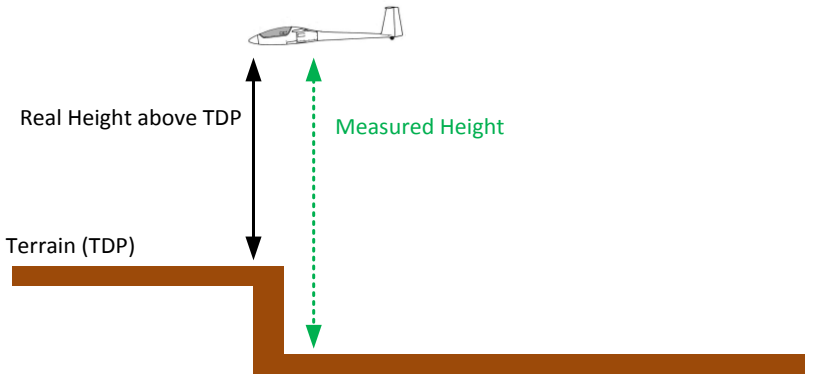


Fig. 36 - The cliff leads to an overestimated height

In Table 10, as in Table 7 the comparison was done later on with the new filter parameters.

Distance to TDP	% Valid Simulations Case 1/2
300m	13 % / 1 %
150m	2 % / 34 %
100m	0 % / 87 %

Table 10 - Table comparing the performances in the case of a cliff (case 1) and in the case of a plateau (case 2)

In Table 10, comparing the results in the case of a cliff and the case of a plateau, the expected results are obtained. That confirms the hypothesis and those simulations show one more time that the new parameters of the filter give better performances than the previous one.

Delayed Laser Activation

The idea of delaying the laser activation, in order to obtain better performances when the terrain is not flat, had to be tested. It had been tested in the flat terrain and slope scenarios, keeping the same protocol as before. The laser was set to activate at 50m or 20m (AGL) while previously it was at 100m (AGL).

Flat Terrain

The results of the simulations at 100m to the TDP in the flat terrain scenario are reported in Table 11.

Activation Height	Mean Error	Standard Deviation	%Valid Simulations	Error max 1σ
100m	-0.99	0.52	59 %	2.2
50m	-0.80	0.54	70 %	1.7
25m	-0.54	0.44	86 %	2.1
Without LASER	-0.33	1.23	47 %	2.2

Table 11 - Table comparing the performances for different height of laser activation

Even if the mean value is worse in the 25m case than without the laser, the standard deviation is way smaller so the performance is way better. This is again explained by the phenomenon stressed in the first part of this subsection.

Slope

The results of the simulations at 100m to the TDP in the slope terrain scenario are reported in Table 12.

Activation Height	Mean Error	Standard Deviation	% Valid Simulations	Error max 1σ
100m	-1.02	0.19	49 %	6.5
50m	-0.39	0.33	95 %	3.1
25m	-0.36	0.41	94 %	2.2
Without LASER	-0.33	1.23	47 %	2.2

Table 12 - Table comparing the performances for different height of laser activation

The best performance is achieved with an activation at 50m even if the maximal value of the error is a bit worse in this case. Anyway, with 25m or 50m the performance is greatly improved.

It can also be noticed that the slope gives better performances than the flat terrain. This is also due to the phenomenon explained in the beginning of this subsection as the laser height sensor underestimates the height of the UAV because of the slope.

So the late activation of the laser brings better performances but it has some limits. If the laser activation occurs too late the UAV and the filter will not have time to correct previous error. This error must be corrected before switching to the

short final phase. An optimal height for the laser activation will have to be determined during more realistic simulations and test flights. A good first guess, could be, say, 50m.

Conclusion

The results of those simulations are only qualitative, since at this point the filter did not have its definitive design and tuning. The conclusion is that the terrain will not be a problem for the filter. It will have an impact on it but nothing that will prevent a good landing to happen if the laser is not activated too early. The impact of the terrain is mainly on the longitudinal axis, it will add to all the other factors which have an impact on this axis such as the camera harmonization as exposed in the next subsection.

6.2 Angular Harmonization of the Camera

Protocol

For every scenario 1000 Monte-Carlo simulations were run in open loop randomly choosing the following parameters:

IMU Parameters	
Parameter	**Law**
Gyro Bias X (°/h)	Normal
Gyro Bias Y (°/h)	Normal
Time since last reset	Uniform
Roll Bias (°)	Normal
Pitch Bias (°)	Normal
Heading Bias (°)	Normal
Altimeters Parameters	
Parameter	**Law**
Baro Bias (m)	Normal
Laser Bias (m)	Normal
Baro Scale Factor (m)	Normal
Laser Noise (m)	Normal
Laser Activation Height (m)	Normal

Vision Parameters (Harmonization Error)	
Parameter	**Law**
Rotation X Sensor (°)	Normal
Rotation Y Sensor (°)	Normal
Rotation Z Sensor (°)	Normal
Filter's Initial Parameters*	
Parameter	**Law**
Erreur en X NED (m)	Normal
Erreur en Y NED (m)	Normal
Erreur en Z NED (m)	Normal
Erreur en VX NED (m/s)	Normal
Erreur en VY NED (m/s)	Normal

The computer vision is supposed perfect and the terrain is flat in order to avoid any error compensation. For the yaw and pitch axis 8 scenario were used, 4 for each axis, with harmonization errors of 0.3° 0.5° 0.7° and 0.9°. And for the roll axis 4 scenario were used with harmonization errors of 0.5° 1.0° 1.5° and 2.0°. An harmonization error means that the camera is not mounted at the theoretical angle and this will introduce an error when comparing the position of the TDP in the picture with the theoretical position of the TDP in the picture (innovation of the filter in the final phase).

Pitch

The results of the simulations done for the pitch harmonization error are reported in Table 13 and Table 14.

Distance TDP | Harmonization Error	0°	0.3 °	0.5°	0.7°	0.9°
300m	68%	60%	55%	49%	49%
150m	75%	66%	64%	62%	59%
100m	-	69%	71%	63%	52%

Table 13 - Table of results showing the percentage of simulation within the limit $LongAccuracy$ at different distance to the TDP

Distance TDP | Harmonization Error	0°	0.3 °	0.5°	0.7°	0.9°
300m	1.02	1.22	1.35	1.46	1.53
150m	0.91	1.10	1.17	1.20	1.21
100m	-	-	-	-	-

Table 14 - Table of results showing the standard deviation of the estimation error (scaled with $LongAccuracy$) on the longitudinal axis at different distance to the TDP

The pitch harmonization error has the expected impact on the performances but it remains reasonable for small harmonization errors. The impact on the lateral axis is not reported since during the landing the UAV has its wings leveled most of the time so the impact of a pitch harmonization error is very small.

Yaw

The results of the simulations done for the yaw harmonization error are reported in Table 15 and Table 16.

Distance TDP | Harmonization Error	0°	0.3 °	0.5°	0.7°	0.9°
300m	94%	85%	75%	64%	53%
150m	77%	91%	85%	79%	73%

Table 15 - Table of results showing the percentage of simulation within the limit $LatAccuracy$ at different distances to the TDP

Distance TDP | Harmonization Error	0°	0.3 °	0.5°	0.7°	0.9°
300m	0.56	0.64	0.72	0.78	0.86
150m	0.46	0.54	0.60	0.68	0.74

Table 16 - Table of results showing the standard deviation (scaled with $LatAccuracy$) of the lateral estimation error at different distances to the TDP

Same as for the pitch error, for the yaw harmonization error only the impact on the lateral axis is reported since it has a very limited impact on the longitudinal axis, even null when the wings are leveled. Again the impact is limited for a reasonable harmonization error.

Roll

The results of the simulations done for the roll harmonization error are reported in Table 17, Table 18, Table 19 and Table 20.

Distance TDP | Harmonization Error	0°	0.5 °	1°	1.5°	2°
300m	94%	94%	94%	94%	94%
150m	77%	77%	77%	77%	77%

Table 17 - Table of results showing the percentage of simulation within the $LatAccuracy$ limit at different distance to the TDP

Distance TDP | Harmonization Error	0°	0.5 °	1°	1.5°	2°
300m	0.56	0.54	0.54	0.54	0.54
150m	0.46	0.44	0.44	0.46	0.46

Table 18 - Table of results showing the standard deviation of the Lateral estimation error at different distance to the TDP

Distance TDP | Harmonization Error	0°	0.5 °	1°	1.5°	2°
300m	68%	70%	70%	70%	69%
150m	75%	76%	77%	77%	77%

Table 19 - Table of results showing the percentage of simulation out of the $LongAccuracy$ limit at different distance to the TDP

Distance TDP | Harmonization Error	0°	0.5 °	1°	1.5°	2°
300m	1.02	1.01	1.01	1.00	1.00
150m	0.91	0.90	0.90	0.90	0.89

Table 20 - Table of results showing the standard deviation of the Longitudinal estimation error at different distance to the TDP

Unlike for the pitch and yaw harmonization error, the roll harmonization has an impact on both the lateral and longitudinal estimations. But this impact is very limited and this can be explained by the fact that the tracked point, the TDP, is almost in the center of the picture.

Conclusions

The axis on which the impact is the greatest is the pitch. The harmonization on the roll axis is not critical since the TDP is in the center region of the picture.

It is important to know that those simulations were done in the ideal scenario of a flat terrain and perfect computer vision tracking. A non-flat terrain would add to the harmonization error on the longitudinal axis and therefore reduce even more the tolerance on the pitch harmonization.

It is still not possible to look at the absolute performances of the filter since it still does not have a definitive architecture and tuning. But by looking at the relative decrease in performances due to harmonization error it seems that an error of 0.5° on the yaw axis reduces the rate of success by 15%, it is a reasonable limit according to the persons responsible for the harmonization.

6.3 Position Harmonization of the Camera

The position of the camera in the body reference frame cannot be perfectly known. So the impact of an error in the position harmonization had to be studied. Only a

geometric study, illustrated in Fig. 38, has been conducted in order to convert the position error in an angular error, and the angular error impact has already been studied in the previous subsection.

Impact at 100m to the TDP

Yaw and Pitch harmonization errors are studied, Fig. 37, at 100m to the TDP in the worst case scenario in which the position error add along each axis adds.

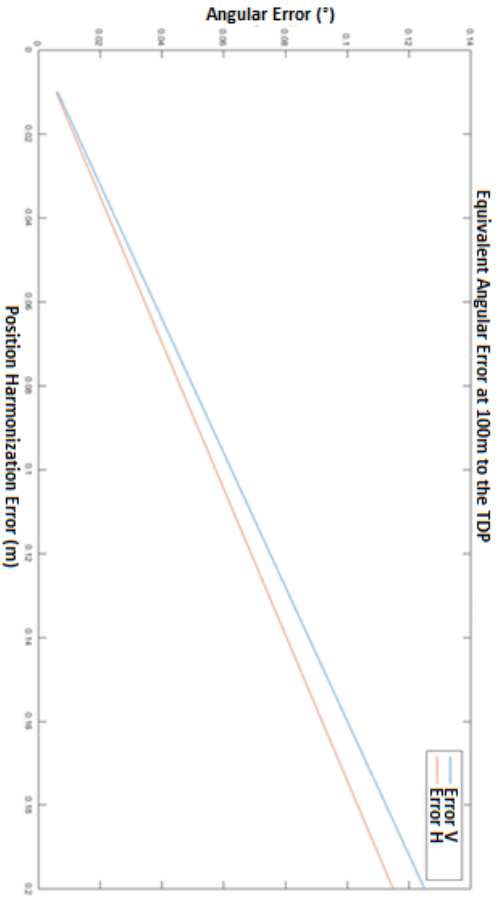


Fig. 37 - Equivalent angular error for different position errors at 100m to the TDP

As at 100m to the TDP the equivalent angular harmonization error due to the position error must be less than to 0.1° then, according to Fig. 37, the error in position harmonization must be below 16cm. It seems an achievable goal and this constraint has been transmitted to the person responsible for the harmonization procedure.

Impact during Landing

In Fig. 39, the equivalent angular harmonization error for a position harmonization error of 16cm is calculated and plotted, still in the worst case scenario. For the previous plot in Fig. 37, the distance to the TDP was fixed and the position harmonization error was the variable. But now the harmonization error is set to 16cm and the distance to the TDP is the variable. This shows what would be the impact of a 16cm error during the whole landing phase.

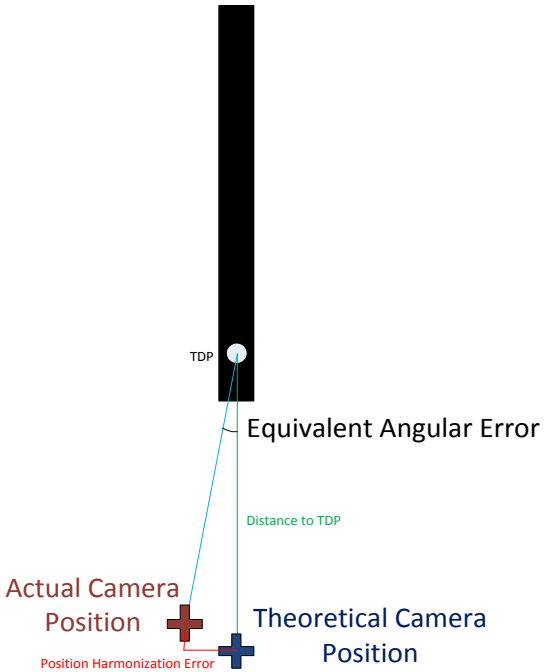


Fig. 38 - The equivalent angular error depends on the position harmonisation error and distance to TDP, that is why two analysis were conducted. One by fixing the distance to the TDP and one by fixing the position error

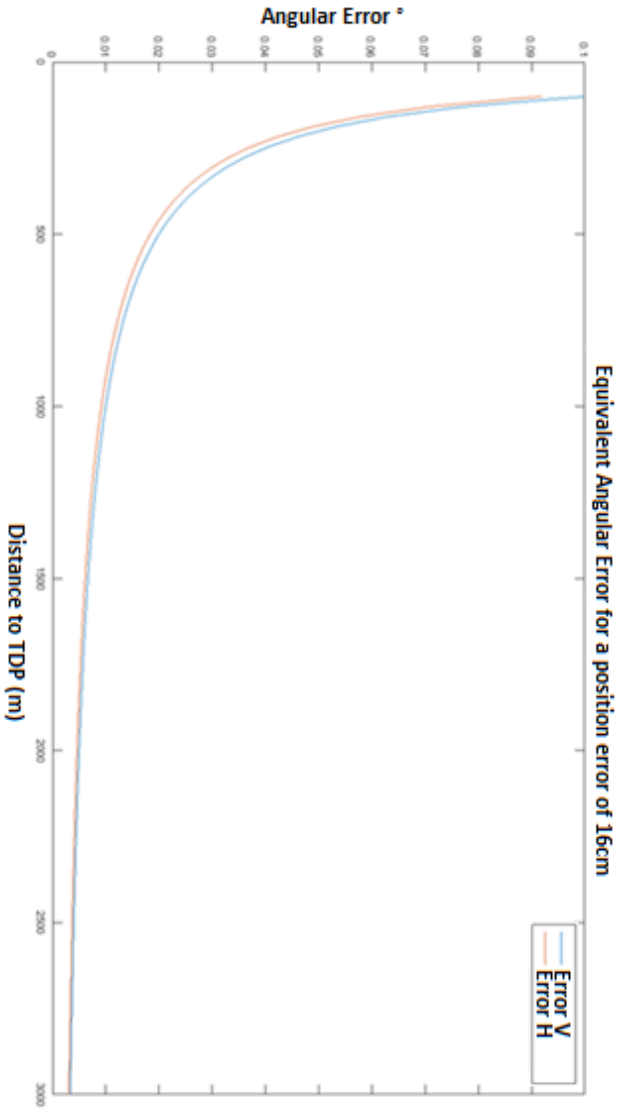


Fig. 39 - Equivalent angular error at different distance to the TDP for a 16cm position error

The impact of the error position becomes significant only in the last 500m.

6.4 Latencies

So far the simulator was not implementing the different frequencies and latencies of the sensors. So once added to the simulator it was possible to study the impact of the latencies in order to determine if synchronization is needed. The reference scenario is without desynchronization but with each sensor working at its own frequency as given in Table 21.

Sensor	Frequency
IMU Attitude	100 Hz
IMU Other	50 Hz
Baro	50 Hz
Laser	10 Hz
Directionnal Antenna	1 Hz
Camera	25 Hz

Table 21 - Frequencies used for simulation

As explained in Section 4.2 the filter is running at the IMU Other data rate and any other measurement is sent to the filter with the next incoming IMU attitude measurement. This study is again done using Monte Carlo simulations, in open and closed loops.

Reference Scenario

Performance, reported in Table 22 and Table 23, has assessed by looking at the percentage of simulation within the limits (both $LatAccuracy$ *and* $LongAccuracy$) over 1000 Monte-Carlo simulations.

Closed Loop

Distance to TDP	300 m	150 m	100 m
Longitudinal axis	71%	72%	78%
Lateral axis	74%	82%	84%

Table 22 - Percentage of simulations within the limits in closed loop for the reference case

Open Loop

Distance to TDP	300 m	150 m	100 m
Longitudinal axis	90%	90%	90%
Lateral axis	86%	91%	94%

Table 23 - Percentage of simulations within the limits in open loop for the reference case

Even if the Monte Carlo parameters are very pessimistic the performance are very good when there is no delay. Table 22 and Table 23 will be used as the reference for the next simulations.

AWC Camera

The first latency studied was the one introduced by the camera. The camera is introducing a delay L_{ti} between the real world and its output. And this delay cannot be corrected as the timestamp on the camera data is put at its output. A typical camera in the considered category has a latency smaller than 100ms. The impact of this delay has been studied using the same protocol as for the other studies: Monte Carlo simulations in open and closed loop with 1000 simulations for every delay: 0ms, 50 ms, 100ms and 500ms. The results are reported along with a comparison with the reference scenario in Table 24-27.

Closed Loop

L_{ti} / Distance to TDP	300 m	150 m	100 m
0 ms	71%	72%	78%
50 ms	72% (+1%)	72% (+0%)	77% (-1%)
100 ms	75% (+4%)	74% (+2%)	79% (+1%)
500 ms	61% (-10%)	59% (-13%)	66% (-12%)

Table 24 - Percentage of simulations within the $LongAccuracy$ limit in closed loop and comparison with the reference case

L_{ti} / Distance to TDP	300 m	150 m	100 m
0 ms	74%	82%	84%
50 ms	76% (+2%)	82% (+0%)	84% (+0%)
100 ms	76% (+2%)	82% (+0%)	84% (+0%)
500 ms	70% (-4%)	74% (-8%)	74% (-10%)

Table 25 - Percentage of simulations within the $LatAccuracy$ limit in closed loop and comparison with the reference case

In closed loop there is no real impact on the performance if the delay is smaller than 100ms. But the result is not the same in open loop.

Open Loop

L_{ti} / Distance to TDP	300 m	150 m	100 m
0 ms	90%	90%	90%
40 ms	73% (-7%)	78% (-12%)	78% (-12%)
100 ms	74% (-6%)	77% (-13%)	77% (-13%)

Table 26 - Percentage of simulations within the $LongAccuracy$ limit in open loop and comparison with the reference case

L_{ti} / Distance to TDP	300 m	150 m	100 m
0 ms	86%	91%	94%
40 ms	79% (-7%)	85% (-6%)	90% (-4%)
100 ms	81% (-5%)	86% (-5%)	91% (-3%)

Table 27 - Percentage of simulations within the $LatAccuracy$ limit in open loop and comparison with the reference case.

There is a larger impact in open loop than in closed loop. The decrease in success rate is in this case around 10%, this is still acceptable if the global performance is increased in order to reach a 90% success rate. This latency affects more the longitudinal axis than the lateral axis and this can be explained by looking at the pitch and roll dynamic: the pitch dynamic is stronger than the roll dynamic as shown in Fig. 40.

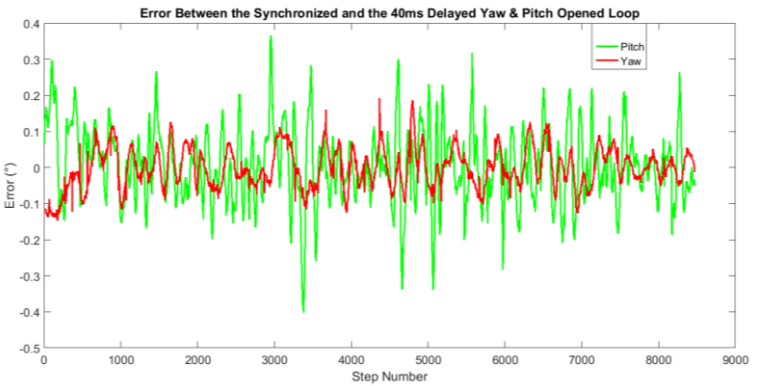


Fig. 40- Comparison between the error introduced by the delay in the pitch and yaw axes

Moreover, in order to explain the difference between the close and open loop results the pitch dynamic has been plotted in both cases in the figure below.

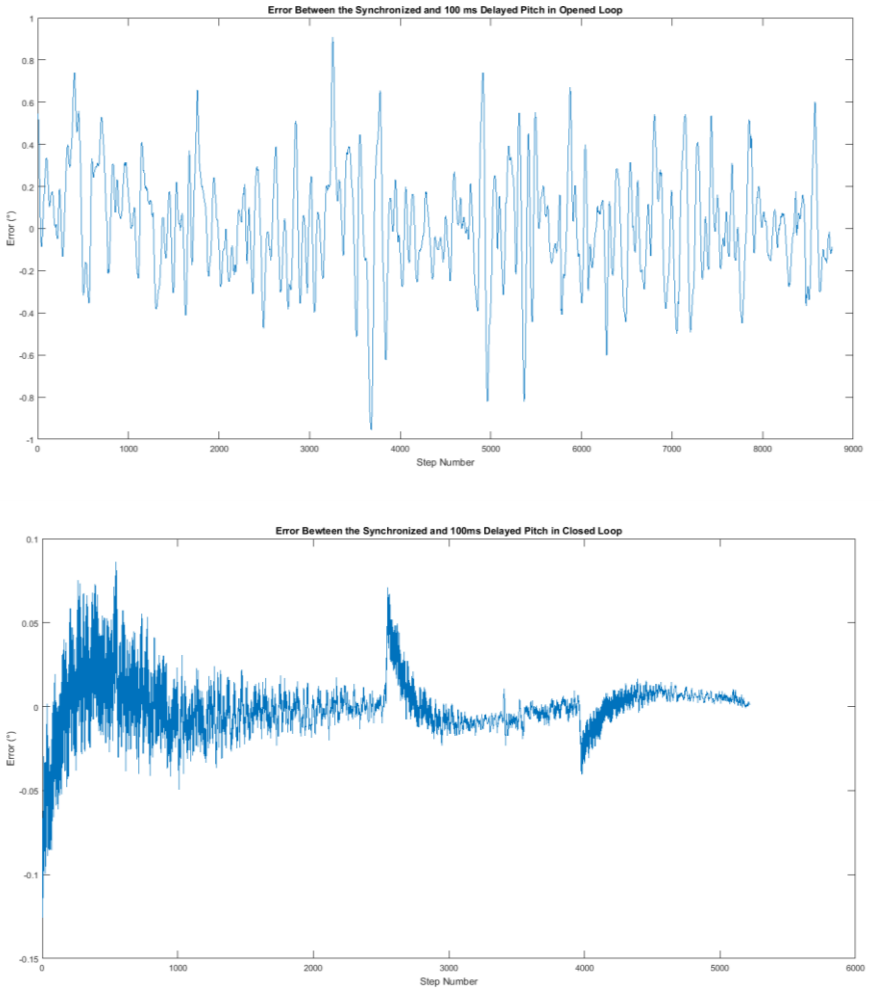


Fig. 41 - a) Open loop pitch dynamic b) Closed loop pitch dynamic

It is clear that it is not the same dynamic and that the pitch dynamic in open loop is 10 times larger (in norm) than in closed loop. This explains why the delay more affects the open loop than the closed loop. This also shows that a better model of the dynamic of the UAV is needed to go further in performances

assessment in closed loop. The hypothesizes made in Section 3.3 are too strong for this kind of analysis.

So even if the impact of the delay introduced by the camera is not negligible, it seems reasonable if the AWC latency is kept below 100ms. An analysis of this delay must be conducted to see if it is possible to model it and account for it in the filter.

Computer Vision

After studying the effect of the latency introduced by the camera the effect of the latency introduced by the whole computer vision chain has been studied. The hypothesis is that thanks to timestamping the attitude data and video data are synchronized at 40ms (only the delay introduced by the camera cannot be corrected by timestamping). The delay of the computer vision chain is estimated to be around 200ms, this includes compression and image analysis.

Open Loop

In Table 28 and Table 29, the results of the Monte Carlo simulations are reported along with a comparison with the reference performance.

L_{ci} / Distance to TDP	300 m	150 m	100 m
0 ms	73%	78%	78%
150 ms	72% (-1%)	78% (+0%)	77% (-1%)
200 ms	73% (+0%)	79% (+1%)	78% (+0%)
250 ms	76% (+3%)	82% (+4%)	82% (+4%)

Table 28 - Percentage of simulations within the $LongAccuracy$ limit in opened loop and comparison with the reference case

L_{ci} / Distance to TDP	300 m	150 m	100 m
0 ms	79%	85%	90%
150 ms	78% (-1%)	86% (+1%)	89% (-1%)
200 ms	79% (+0%)	84% (-1%)	89% (-1%)
250 ms	79% (+0%)	87% (+2%)	93% (+3%)

Table 29 - Percentage of simulations within the $LatAccuracy$ limit in opened loop and comparison with the reference case.

The delay seems to improve the performance; it might come from the poor synchronization of the open loop data. Nevertheless, the conclusion that this delay will have a negligible impact can be drawn. This is compatible with the fact that

the UAV has a relatively slow dynamics. The effect of this delay has not been studied in closed loop since the previous part showed that a better model of the UAV is needed for this kind of closed-loop simulation.

6.5 Fly Back Performance

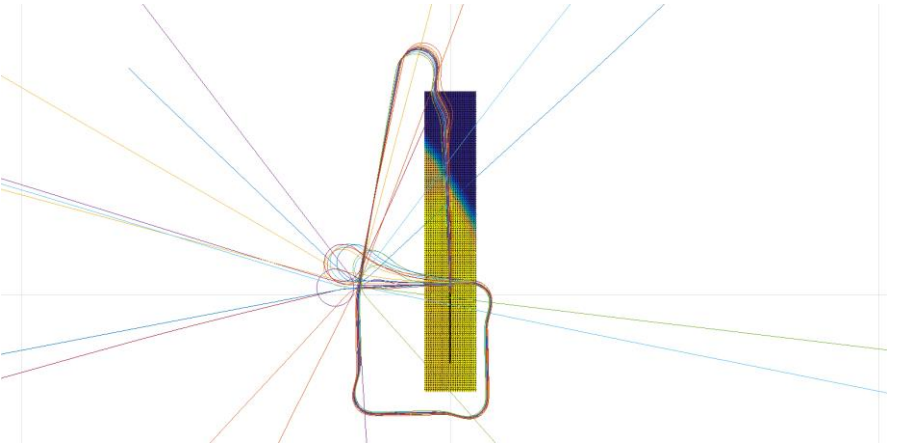


Fig. 42 – Plot of the trajectories followed by the UAV during Monte Carlo simulations. The runway is oriented vertically, on the bottom of the picture

Many simulations were done for the fly back, showing that the proposed solution is working. In Fig. 42 a few simulations from a Monte Carlo simulation have been plotted. It is a closed loop simulation with a real terrain model. The airfield is in north of France on a cliff, the part in blue is actually water and another view is available in Fig. 8. This terrain was chosen because of this cliff and its impact on the laser height sensor. The simulations were full procedures, it means that they were starting from the loss of GPS signal and ended at the touch down. The runway is in the bottom of the picture in black and oriented vertically. The ground station is situated to the left of the runway.

For each simulation the UAV starts with different initial conditions and sensors parameters. It can be seen that every simulation is following the planned trajectory describing a large square around the GS before joining the final. The turns before reaching the GS pattern are due to a problem in the autopilot model. On the top of the picture, after the 180° to head back toward the runway, the UAV is following a track doing a 30° interception angle with the axis of the runway. The UAV turns onto final as soon as it intercepts the runway axis, this is why it seems to zig-zag a bit.

It is clear that all the simulations are converging and the UAV is nicely following the planned trajectory. There is no data available yet to do open-loop simulation for the flight back but the closed loop validated the solution.

6.6 Global Performance

Scenario

In this subsection a few more details on the filter and its behavior are given. An open loop simulation using as much real data as possible has been done. Using a flight recording for which the video, the PVAT and the laser sensor data are available. As there is not any data available for open loop analysis of the fly back this subsection will just focus on the landing phase. The data start 5km away from the TDP, aligned with the runway. The approach lasts around 2min30s.

Estimation

Lateral Position Estimation

The lateral position estimation made by the filter during this approach is plotted in Fig. 43 and the error made by the filter in Fig. 44.

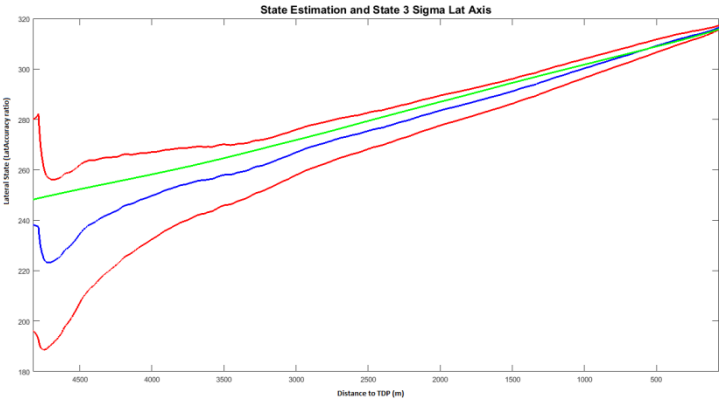


Fig. 43 - Position estimation on the lateral axis (X_r). In green the actual state, in blue the estimate and in red the 3σ envelope

It can be seen that the filter is consistent and it seems to converge quite well.

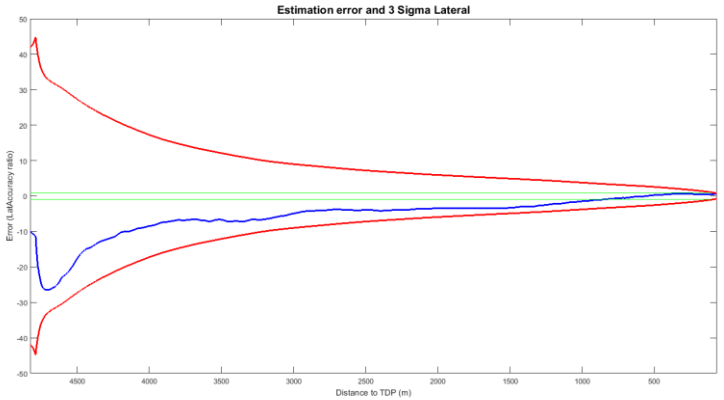


Fig. 44 - Estimation error (*LatAccuracy* ratio) on the lateral axis. In blue the estimation error, in red the 3 σ envelope and in green the +/- *LatAccuracy* limit

The good behavior of the filter is confirmed and the +/- $LatAccuracy$ is reached and not left at 850m to the TDP, more than 20s before switching to the short final phase. At the beginning the filter seems to diverge, this is because at this point the tracking algorithm was not locked on the TDP yet.

Longitudinal Position Estimation

The longitudinal position estimation made by the filter during this approach is plotted in Fig. 43 and the error made by the filter in Fig. 46.

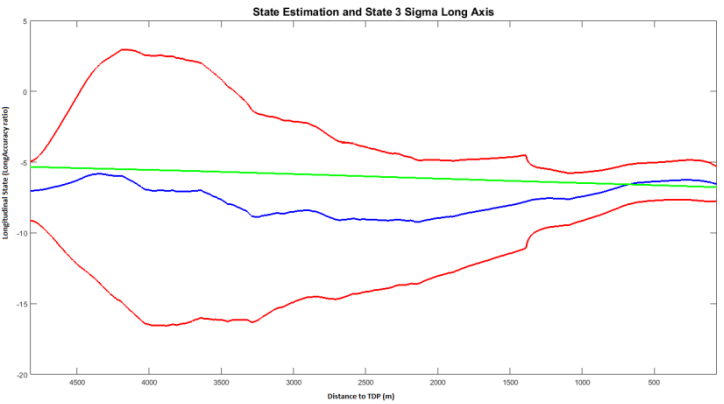


Fig. 45 - Position estimation on the lateral axis (Y_r). In green the actual state, in blue the estimate and in red the 3σ envelope

The behavior of the filter seems to be almost as good on the lateral axis. During the first hundreds meters the covariance is increasing, this is probably due to the fact that the UAV is very far from the TDP and a very small error in the measurement would lead in a huge error in longitudinal position estimation. This is carried by the Jacobian of the function h(k) in the filter.

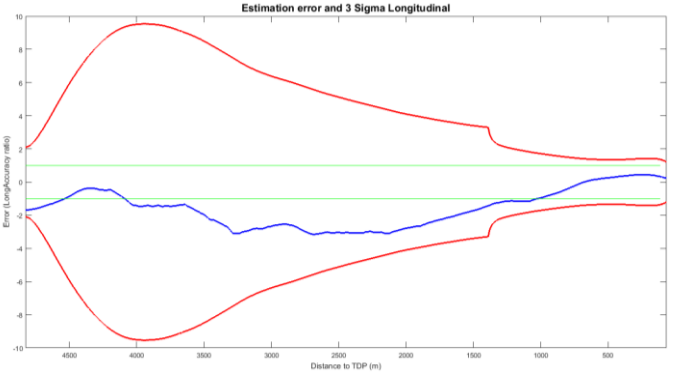


Fig. 46 - Estimation error on the longitudinal axis. In blue the estimation error, in red the 3 σ envelope and in green the +/- *LongAccuracy* limit

The +/-$LongAccuracy$ limit is reached 1km before the TDP, which is also very good. The effect of the laser activation can be seen around 1km to the TDP in Fig. 45 and Fig. 46.

Lateral Speed Estimation

The lateral speed estimation made by the filter during this approach is plotted in Fig. 47.

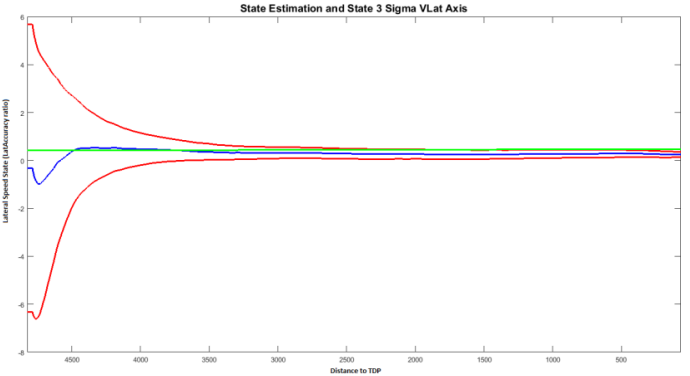


Fig. 47 - Speed estimation on the lateral axis (Xr). In green the actual state, in blue the estimate and in red the 3σ envelope

The behavior of the filter is correct but has to be improved as it is not consistent at the end.

Longitudinal Speed Estimation

The lateral speed estimation made by the filter during this approach is plotted in Fig. 48.

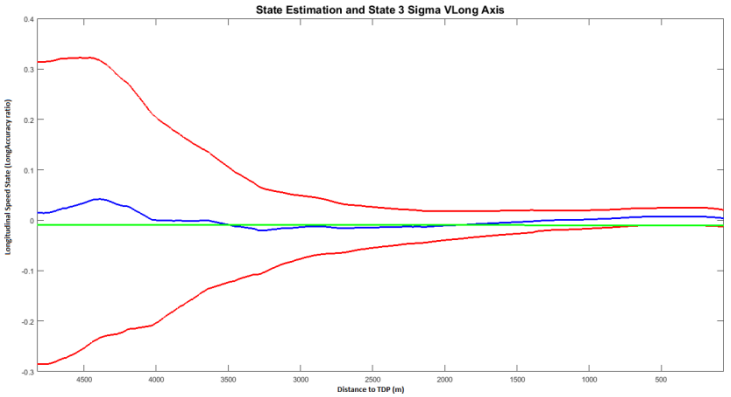


Fig. 48 - Speed estimation on the longitudinal axis (Yr). In green the actual state, in blue the estimate and in red the 3σ envelope

The filter is again not consistent at the end, and the estimate is not as good as it should be so it has to be improved, but the behavior of the filter is correct.

It is very important to get a good speed estimate because in the short final mode there will not be any way to improve the estimate and the plane will just drift at the speed estimation error.

Innovation

It is also very interesting to look at the innovation of the filter, i.e., the difference between the measurement and the predicted measurement made by the filter. It gives hints on the behavior of the filter and it is an easy way to detect errors like faulty measurement. A faulty measurement detection could even be added to the filter using the innovation and its covariance.

Camera Lateral Axis

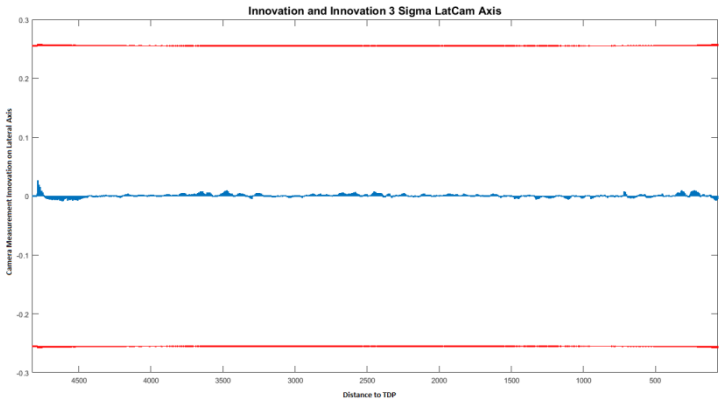


Fig. 49 - Vision measurement innovation on the lateral axis (blue) with the 3σ envelope (red)

The innovation plotted in Fig. 49 is quite small compared to the 3σ envelope. This is compatible with what was said in Section 4.3, when tuning the parameters, about the increase of the R coefficient for the vision based measurements.

Camera Longitudinal Axis

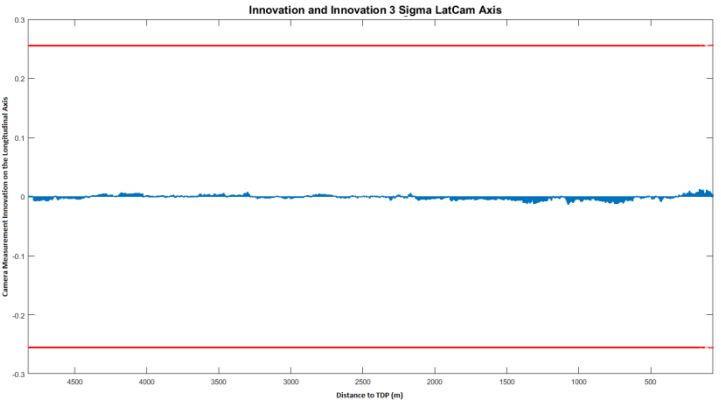


Fig. 50 - Vision measurement innovation on the longitudinal axis (blue) with the 3σ envelope (red)

The innovation plotted in Fig. 50 is again quite small compared to the 3σ envelope, as for the lateral axis.

Laser

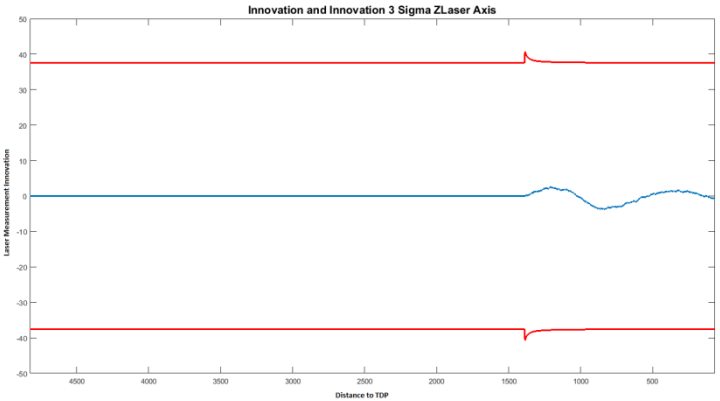


Fig. 51 - Laser measurement innovation (blue) with the 3σ envelope (red)

It can be seen in Fig. 51 that the laser sensor is not active until around 1000m to the TDP. Once it is active the innovation seems in a good range of order compared with the covariance. The terrain is not flat and this contributes a lot to the innovation.

Regarding this simulation the filter seems to be well tuned but the tuning could probably still be improved a bit in order to improve the estimation of the velocity errors. This will be mentioned in the next section about future work.

7. Future Work and Conclusion

First of all, the filter algorithm must be improved to include faulty measurement detection. It is necessary to have an even more robust solution.

Moreover, the wind and the magnetic declination have to be taken into account. The wind should not be a problem regarding the way the algorithm works but the magnetic declination has not been taken into account yet which would probably lead to huge errors is certain part of the world. This has to be thought through before implementing the algorithm on the UAV.

Some tests remains to be done, especially in closed loop. But for that there is first of all the need to improve the models as stated in Section 6.4.

The ground phase, from the touch down until the UAV stops, has to be developed. A suggested solution has been drawn during this thesis but has not been tested yet. This is why there is no result or explanation about the ground phase in this report.

Too late during this project, it appeared that the directional antenna of the ground station is not measuring only the azimuth but also the elevation to the UAV. This elevation measurement has to be added to the filter and it will improve the performance of the fly back.

Finally, the code will be generated for test and tuning on the real system and the test flights will validate the solution studied during this thesis.

The solution studied during this thesis is not fully automatic as it requires a sparse input from the UAV operator. Of course it could be automated by adding a more advanced computer vision algorithm able to recognize the runway and the touch down point in the picture. This was not the goal of the thesis and was not in the scope either, but everything developed and tested during this thesis can be

used along with a more advanced computer vision algorithm. This would lead to a fully automatic landing without GPS solution which is robust and easy to implement on SAGEM's UAV. It could also be implemented on other UAVs but the main drawback of this solution is that it uses a ground station equipped with a directional antenna able to track the UAV. This was not a problem for this project since this antenna was already a part of the UAS.

Many tests were done during this thesis and the tens of thousands simulations done proved that the solution is working and is robust. State-of-the-art technologies free of ground equipment, presented in Section 2, are using more complex computer vision algorithm and/or large databases of the overflown areas. This is also very promising and, with all those solutions, fixed-wing UAVs will be able to perform landing without GPS without ground equipment in the near future.

.

Appendix A

UAV Dynamic Model

As stated in Section 3.3, the model used for the dynamic of the UAV is quite basic. It is based on 4 hypotheses:

- The ground speed is constant → $V_n(k).(1,1,0)^t = V_{ground}$
- The vertical speed follows the consign → $V_Z(k) = V_Z^{cmd}(k)$
- The roll follows the roll consign → $\varphi(k) = \varphi^{cmd}(k)$
- The UAV is at equilibrium on the pitch axis → $\theta(k) = \frac{A}{V_n(k)^2} - B + \tan\left(-\frac{V_Z(k)}{V_{ground}}\right)$ with A and B some parameters characterizing our UAV

And we also have:

$$P_n(k) = P_n(k-1) + V_n(k).dt$$
$$\psi(k) = \psi(k) + \frac{g}{V_{Sol}}.\tan(\theta(k)).dt$$

With:

$$(Yaw, Pitch, Roll) = (\psi, \theta, \varphi)$$
$$P_n\ Position\ of\ the\ UAV\ in\ NED$$
$$V_n\ Speed\ if\ the\ UAV\ in\ NED$$

Flight Controller Model

As stated in Section 3.3, the model used for the Flight Controller is not based on the real algorithm of the real FC. But it use simplified algorithm to reproduce almost the same behavior. There is no need of improving this part while the UAV dynamic model has not been improved.

The FC is computing a roll and a vertical speed commands.

The roll command is established using helmsman's law as illustrated in Fig. 52. As for a boat autopilot the command is calculated from the cross track and heading errors, a saturation is applied to avoid rolling the UAV.

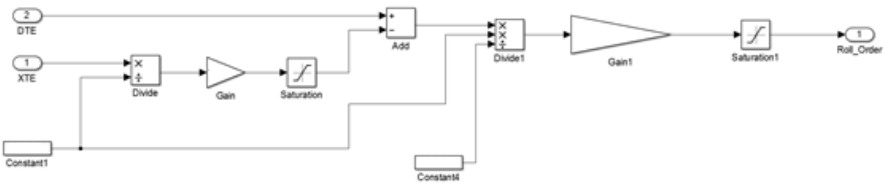


Fig. 52 - Roll control

For the vertical speed control a simple PI controller with ant windup is used. There is some saturation to constrain the pitch angle. With the appropriate tuning, which will not be described here, it gives the expected behavior on the altitude control as it can be seen on the figure below. But as seen in Section 6.4 it gives a very bad pitch dynamics.

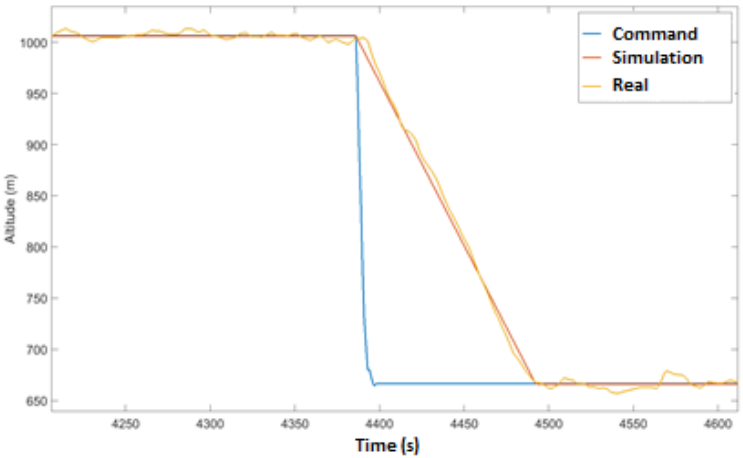


Fig. 53 - Altitude step command, simulation compared with real data

References

[1] SAGEM, "Systeme Patroller," [Online]. Available: http://www.sagem.com/fr/aeronautique-et-espace/drones/systeme-patroller. [Accessed 09 05 2016].

[2] SAGEM, "Patroller Homeland Security," [Online]. Available: http://www.sagem.com/sites/sagem/files/patroller_homeland_security.pdf. [Accessed 19 05 2016].

[3] T. Oberst, “Solar flares cause GPS failures, possibly devastating for jets and distress calls, Cornell researchers warn", Cornell Chronicle, 26 09 2006. [Online]. Available: http://www.news.cornell.edu/stories/2006/09/solar-flares-could-seriously-disrupt-gps-receivers. [Accessed 09 05 2016].

[4] MathWorks, "Matlab," 2016. [Online]. Available: http://fr.mathworks.com/products/matlab. [Accessed 09 05 2016].

[5] M. Grewal, L. Weill and A. Andrews, Global Positioning Systems, Inertial Navigation, and Integration, John Wiley & Sons, Hoboken, NJ, Inc, 2001, pp. 11,18,133,148.

[6] O. J. Woodman, "An introduction to inertial navigation", University of Cambridge, Cambridge, UK, 2007, p.24.

[7] P.G. Savage, "Schuler Oscillations", Strapdown Associates, Inc., 27 06 2014. [Online]. Available: http://strapdownassociates.com/Schuler%20Oscillations.pdf. [Accessed 19 05 2016].

[8] P. Garrec and P. Cornic, "Autonomous And Automatic Landing for Drone". United States Patent US 8,265,808 B2, 2012.

[9] J. Vezinet, A. Escher, A. Guillet and C. Macabiau, "State of the art of image-aided navigation techniques for aircraft approach and landing," ION ITM

2013, International Technical Meeting of The Institute of Navigation, Jan 2013, San Diego, United States, pp 473-607, 2013.

[10] A. Miller, S. Mubarak and H. Don, "Landing a UAV on a Runway Using Image Registration", University of Central Florida, 400 Central Florida Blvd, Orlando, FL.

[11] R. Zelenka, Integration of Radar Altimeter, Precision Navigation, and Digital Terrain Data for Low-Altitude Flight, NASA Technical Memorandum 103958, Ames Research Center, Molfet Field, California, 1992.

[12] "Automatic Landing," *Flight International*, pp. 670-672, 30 10 1969.

[13] G. Cai, B. Chen and T. Lee, Unmanned Rotorcraft Systems, 2011.

[14] National Geospatial-Intelligence Agency, "World Geodetic System 1984," 2000.

[15] "World Geodetic System 1984," 2012. [Online]. Available: http://www.unoosa.org/pdf/icg/2012/template/WGS_84.pdf. [Accessed 10 05 2016].

[16] Oxford Aviation Academy, ATPL Aircraft General Knowledge, 2008, pp. 53-75.

[17] "The Pinhole Camera," [Online]. Available: https://www.ics.uci.edu/~majumder/vispercep/cameracalib.pdf. [Accessed 10 05 2016].

[18] M. Giebner, Tightly-Coupled Image-Aided Inertial Navigation System via a Kalman Filter, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2003 pp. 37, 51-62.

[19] R. Johansson, Predictive and Adaptive Control, Lund University, Dept. Automatic Control, 2015, Lund, Sweden p. 129.

[20] H. Jia, Data Fusion Methodologies for Multisensor Aircraft Navigation Systems, Bedford United Kingdom: Cranfield University, 2004, p51.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER´S THESIS

Date of issue
July 2016

Document Number
ISRN LUTFD2/TFRT--6006--SE

Author(s)
Thomas Leonard

Supervisor
Rolf Johansson, Dept. of Automatic Control, Lund University, Sweden
Karl-Erik Årzén, Dept. of Automatic Control, Lund University, Sweden (examiner)

Sponsoring organization

Title and subtitle
Automatic Landing without GPS

Abstract
Sagem Défense et Sécurité (now Safran Electronics & Defense), a French space and defense company of the SAFRAN group, is working on the next generation of Unmanned Aerial System (UAS). This UAS features a fully automatic Unmanned Aerial Vehicle (UAV) equipped with a state-of-the-art navigation system. This navigation system relies mainly on a high-accuracy Inertial Measurement Unit (IMU) coupled with a GPS receiver. But the GPS is known to be easy to jam, either naturally (solar flare for example) or intentionally. In the event of a loss of GPS signal, the navigation system is not able anymore to provide accurate position and speed information to the Flight Controller (FC). Deprived of reliable position and speed information the FC is not able to guide the UAV safely to the ground.

So the goal of the project detailed in this report is to add to the existing UAS the ability to land safely in case of a GPS loss. At the core of the solution described in this report is a sensor fusion algorithm taking as input inertial, vision based, barometric, laser and azimuthal measurements. The filter is using all these measurements to establish reliable position and speed estimates.

Even if very reliable systems enabling automatic landing without GPS exist today; they all require heavy and expensive ground equipment. This is why SAGEM decided to develop its own solution using more embedded sensors and less ground equipment. This is a first step toward a fully embedded automatic landing system nondependent on GPS availability, a very active field of research today. All the tests done during the thesis and presented in this report shows the efficiency and robustness of this solution.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-98

Recipient's notes

Security classification