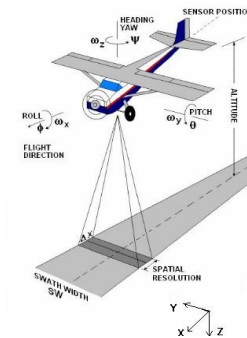


# **Airborne attitude estimation using a Kalman filter**

**Master thesis by  
Mathieu Marmion**



**UNIS – The University Centre of Svalbard  
(Longyearbyen, Norway)**

**NTNU – Norwegian University of Science and Technology  
(Trondheim, Norway)**

**ENSIEG – Superior National School of Electrical Engineering  
Part of the Polytechnic National Institute of Grenoble (INPG)  
(Grenoble, France)**

**February 15, 2006**



## SUMMARY

This project is ending my masters whose main studies directions are the signal processing, the systems of navigations and the remote sensing. I did it at UNIS, the university centre of Svalbard in Norway. This project was covered by the 3 universities I depend on: INPG, NTNU and UNIS.

The aim of my project was to create an attitude estimator for an airborne platform used for remote sensing application. The attitude is the angular orientation. The data have to be processed in real time with the compiler Delphi (pascal). I used several sensors: one GPS and one whole sensor bought to the firm Microstrain containing three accelerometers, three magnetometers and three turn rate sensors featuring a gyroscope. One estimator from Microstrain is sold as well with the sensors. It reacts strongly to the inertial forces which flaw the estimation. The estimator I had to build had to be more accurate, less sensitive to the inertial forces.

The theory of the quaternions is used. It permits to describe a 3D rotation with a four dimensional state vector. This theory is based on the Euler's theorem which stipules that "a 3D solid rotation is equivalent to only one, along an eigen axis, with an eigen angle".

It was possible to create 2 models of estimators regarding the previous statement:

- an integration model
- a Kalman filter.

The integration model is based on the Hughes equation. It links the quaternions state vector with its first time derivative and with the turn rate sensor's data. A simple integration is made to estimate the attitude.

The Kalman filter is also based on the Hughes equations. The essence of this Kalman filter is to adjust a prior estimation based on an attitude model, via measurements provided by the accelerometers and the GPS. Thus the estimation is updated and more precise. The measurement noise is taken into account in this method.

Both models were implemented.

After a short analysis, it was possible to be aware of the complementarities of these 2 models. In one hand, the Kalman filter is really precise in steady conditions; but reacts strongly to the inertial forces. In another hand, the gyro model is drifting in steady conditions, because of the gyro bias, but does not react strongly to the inertial forces. Thus, I had the idea to combine both models, regarding the motions conditions: steady or not. Two parameters were implemented on this purpose. One detecting the inertial forces and the second one the solid rotations. Both are based on the accelerometers data. If the motion conditions are steady, the Kalman filter provides the attitude estimation. Else, the integration model does it. The bias of the gyroscope is taken into account. The Kalman filter, which was improved, tracks this gyroscope bias in real time. Thus, the drift observed with the integration model disappeared.

A comparative analysis was performed between the 3 previous models: the Kalman filter, the integration model and the combined estimator. The Microstrain estimator is also taken into account, to provide a reference for the accuracy we want to. A low pass filter was added to improve the accelerometers data. Two main kinds of tests

were experimented: some laboratory ones, to validate the estimators regarding the problem statement, and some “realistic” ones, to estimate the relevance of the estimators’ behaviour for the final airborne application.

The laboratory tests shown clearly, that the switch between the 2 basic models is rather efficient. The combined estimator is the best one compared to the Kalman filter and the integration model, but is as well even better than the one of Microstrain.

The realistic tests were performed with a car, or pushing a trolley. Because we do not have planes data, we thought that a car could simulate quite well the noise of a plane, to provide realistic data. It appeared that the realistic environment noises the measurements significantly. The steady parameters had to be adjusted. It was obvious that the combined estimator was not accurate enough. The one of Microstrain was even better. Some improvements were necessary. Nevertheless, the results were not completely out of relevance. The inertial forces flawed the estimation.

To get rid of the wrong estimation caused by the inertial forces, 2 approaches were considered.

The first one consists to remove the inertial forces from the accelerometer data, before their processing. The GPS provides the speed of the plane. A time derivative of this speed, gives us an access to the linear acceleration. Considering the estimated Roll, we know how the plane is turning, and it gives an access to the centrifugal forces. We tested that in a car (only the linear acceleration). It appeared that the GPS provides some data only every 1-2 seconds. It is not fast enough. During this time, the estimation has enough time to become biased.

We thought about a second technique to improve the combined estimator. It consists to improve the switch between the 2 basic estimators. Both steady parameters are based on the noisy data of the accelerometer. The parameter detecting the inertial forces was improved, implemented an independent Kalman filter tracking the gravity constant in real time. This last method improved the switch, and as a consequence the estimation. As a result, the combined estimator was better than the one of Microstrain pushing a trolley, and nearly equivalent in a car. Because of these good results, we decided to test the estimator in a plane.

We flew with the Norwegian coast guards. The sensors were fixed on the plane floor. Before the take-off, the estimation was quite correct. During the flight, it was totally wrong. Why?

Before this flight, we did not have any plane data. It appeared that a car is really less noisy than a plane (even in Svalbard roads). The estimator was not configured for such noise. Especially the steady parameters. They did not detect if the flying conditions were steady or not. As a consequence, the gyro bias was badly estimated and the integration model was drifting. Because the steady conditions were not detected, the combined estimator provided a drifting estimation from the integration model. The Kalman filter alone, behaved well as soon as the flying conditions were steady. After an analysis of the raw plane data, I assume it is possible to adjust the sensibility threshold of the steady parameters. These thresholds are suggested in the report.

The perspectives are numerous. We now dispose of real plane data. As a consequence it will be possible to make some perfect realistic simulations to adjust the combined estimator. It should improve the estimation significantly.

Some other improvements could be also taken into account; like mechanical approaches, based on aeronautical theories, or using the GPS on a better way.

## TABLE OF CONTENTS

<b>1. Introduction</b>	<b><u>p. 6</u></b>
<b>2. Problem statement</b>	<b><u>p. 8</u></b>
<b>3. The different estimators</b>	<b><u>p. 12</u></b>
• 3.1. Kalman filtering	p. 12
• 3.2. Airborne attitude estimation	p. 15
• 3.3. The integrated quaternion model	p. 20
• 3.4. Prior comparison of these 2 approaches	p. 20
<b>4. The combined estimator</b>	<b><u>p. 22</u></b>
• 4.1. The enhanced Kalman filter	p. 22
• 4.2. The combined estimator	p. 28
<b>5. Analysis of the different solutions</b>	<b><u>p. 30</u></b>
• 5.1. Bench tests	p. 30
• 5.2. The real behaviour – Dynamic tests	p. 36
• 5.3. Summary of the tests	p. 39
<b>6. The improvements</b>	<b><u>p. 41</u></b>
• 6.1. The GPS	p. 41
• 6.2. The “gravity Kalman filter”	p. 42
• 6.3. Summary of the gravity tracking	p. 47
<b>7. The final application: the airborne test</b>	<b><u>p. 48</u></b>
• 7.1. The plane vibrations	p. 48
• 7.2. The attitude during the flight	p. 50
• 7.3. Summary of the flight	p. 51
<b>8. Perspectives – Conclusion</b>	<b><u>p. 52</u></b>

**Appendix:**

• A. Least square adjustment	p. 54
• B. Quaternion	p. 57
• C. Hughes equation	p. 61
• D. Matrix derivation	p. 64
• E. The semi-continuous prediction model	p. 65
• F. Equipments analysis	p. 66
• G. Configuration of the low pass filter	p. 73
• H. Optimisation of the deviation factor threshold	p. 77
• I. The steady parameters	p. 79
• J. Effects of the plane rotation on the accelerometer	p. 82

**References – Related links**

p. 85

# 1. Introduction

This project completes my master program. I started it in my French home institution, the “Grenoble Polytechnic National Institute” (INPG), where I obtained my bachelor degree, specializing in signal processing. Due to the Double Degree exchange agreement between INPG and NTNU (Norwegian University of Sciences and Technology), Trondheim, I spent one year abroad following the master program, in the Information Technology, Mathematics and Electrical Engineering faculty (ITME). My main study direction was “navigation and remote sensing”. In 2005, I accepted as a Norwegian Master student.

In order to finalize the double degree program, I did my master thesis at UNIS, the University Centre in Svalbard located in Longyearbyen, Norway. I was student in the department of Geophysics, for 6 months (from the 15<sup>th</sup> of Aug. 2005 to the 15<sup>th</sup> of Feb. 2006).

As a consequence of the above I had different supervisors to support my thesis.

1. Fred Sigernes from UNIS, professor at the Geophysics department.
2. Jocelyn Chanussot from INPG, working in the Signals and Images processing Laboratory of Grenoble.
3. Börje Forssell, professor at NTNU, at the IME faculty.

My subject deals with **attitude (angular orientation) estimation of an airborne platform**. Fig.1 shows our experimental setup.

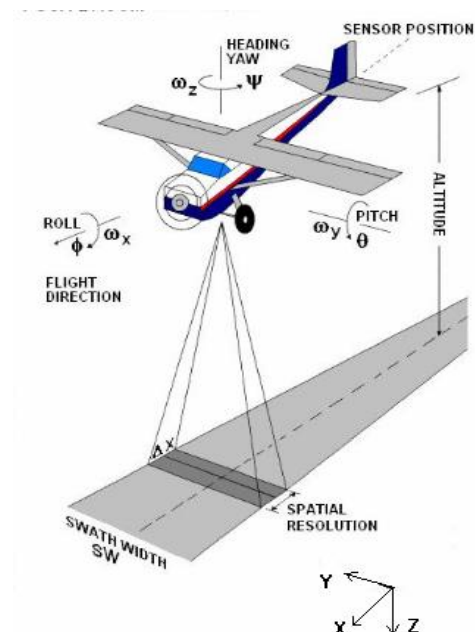


Figure 1. Orientation of airborne carrier.

The platform contains an optical system, a laptop and navigation equipment to determine the attitude.

We use a GPS to obtain the position, an accelerometer and a turn rate sensor to obtain the attitude.

In remote sensing, the knowledge of the optical system attitude is fundamental, for the creation of maps.

To localize an object at ground level, it is important to know where the plane is located in space, and also the angles (attitude) with respect to the vertical.

These angles are defined in the body frame coordinate system as Roll, Pitch and Yaw.

Another situation illustrates the importance of a good attitude estimation. In order to localize an object; we can imagine a fishing boat working close to a protected area. This situation is not seldom, since a great part of Svalbard is protected and the maritime area has a quite good fish abundance reputation. If the coast guard is checking the position of a boat, it has to be precise. Therefore, the Norwegian coast guard are interested in this project, and support it.

As we have illustrated, having a good estimation of attitude is fundamental in remote sensing.

My project is separated in 6 different phases which are summarized below.

Phase 1: To acquaint myself with the project. F. Sigernes spent a lot of time with me at the beginning explaining me the goal of the project and his expectations: to create a real time attitude estimator. After that, the problem statement was established.

Phase 2: Afterwards, it was possible to implement and to code 2 estimators regarding the statement. Testing as well to estimate the project progression. It permits to point out the limits of each solution: they complete each other regarding the flying conditions.

Phase 3: A third estimator was implemented: a hybrid of both previous estimators. This estimator has the advantages of each solution removing their disadvantages depending on the flying conditions. Some parameters were implemented on this purpose. Nevertheless, the Kalman filter had to be improved to be used efficiently.

Phase 4: Laboratory and realistic tests were performed. They compared the properties of each estimator and their behaviour in different environments.

Phase 5: Due to the results, some improvements of the hybrid estimator (the best one) were necessary, especially the parameters detecting the flying conditions. These improvements were quite efficient.

Phase 6: Then, it was the time to use our estimator on the final application: the airborne test.

This report shows the work I did during the 6 months I was in Svalbard, to estimate an airborne platforms attitude. First of all, it is necessary to describe precisely the project, to know exactly the expectations from it and the properties it should have. The description is in the next chapter: the problem statement.

## 2. Problem Statement

As it was introduced previously, to create a map via pictures taken from an airplane, it is necessary to know precisely the attitude of the camera. The platform contains a sensor including turn rate sensors, accelerometers and magnetometers acquired from the American firm Microstrain Inc. The GPS is from the company Garmin.



Figure 2. The GPS is linked to the laptop via a USB Huber. The turn rate sensors/accelerometers/magnetometers are connected to the laptop with the same USB Huber. The sensors are alimeted by an independent battery (DC 12 V) at the exception of the GPS.

A mechanical gyroscope can be described in an orthogonal coordinate system with 3 rotors turning on each axis. The aim of the device is to provide the attitude of the object on which it is fixed to. A turn rate sensor is fixed on the gyroscope. It measures how the gyroscope is rotating.

There are different types of gyroscopes. The one we use is a piezo-electric one. Like most of them, it is not perfect. There is an electrical noise and a drift (bias). As we mentioned previously, an accelerometer and a magnetometer are linked with it, forming the complete sensor.

Microstrain sells software with the gyroscope, which processing the data estimates the attitude. When we test it on the airborne platform, the results are not satisfying enough. At the launch of the plane or during the flight, the estimated attitude is wrong, because the system is too sensitive to inertial forces. For example, when we accelerate on the flat launching path, the estimated attitude gives us a positive Pitch as if we were going up. The same phenomenon appears with the Roll when we are turning; because of the centrifugal force. It reacts like a ball on a horizontal racquet. If this racket is moving the ball is rolling.

**Because of the flawed response to inertial forces, the goal of my project was to create an attitude estimator, not sensitive to the inertial forces, using the sensors I disposed, without additional hardware. The estimated attitude will be as accurate as possible, processing the data in real time.**



This project is subject to some restrictions:

***a) Real time processing:***

As we mentioned already, we want to estimate the attitude of a plane taking pictures. Thus, if we want to create a map, we need to know the location when we are taking the picture. Flying is expensive. Therefore, we can not afford to take pictures without being sure of their validity. Thus the processing of the raw data has to be done in real time.

Furthermore, we want to implement this system in an autopilot for a drone. To make it work, the data have to be processed in real time as well, to command the different motors with a negligible delay.

We will see later that this real time condition was a determining factor in deciding which solutions were available to implement.

Furthermore, we have to consider that the attitude has continuous variations. Thus the estimation has to be continuous as well, or if not, at least that such the sampling period is really small regarding the attitude's time properties. We want the displayed attitude to be smooth.

***b) Accuracy***

The attitude estimator has to be as accurate as possible. The noise from the sensor and from the platform itself affects the accuracy of the estimator. The plane is vibrating, thus it creates some noise as well. The noises have to be filtered out to provide an accurate estimation.

***c) Hardware***

The budget is limited. The price of the equipment is in the order of 1200€ for the Microstrain sensor head and 400€ for the GPS. Taking into account the laptop used to process the data, the project starts to become quite expensive. Furthermore, it is only one part of the total platform. The optical part of the system is not taken into account. In addition, we have to add the cost of the flights.

We have to be aware about the economical factor, if we want to use some other filter in our system, like low pass filters. The filter should be coded.

The coding and the implementation of the estimator was an important feature of the project. A large part of the program code existed already. Thus for questions of adaptation, the coding language had to be the same one as the existing one. This program language is "Pascal". We use the compiler Delphi 5.

#### d) The access to the data

It was not easy to get any realistic raw data, because of the limited number of flights. The cost was the main limiting factor.

The system has to be robust as well. Primarily, we want to use it on a plane, but the attitude estimator has to be adaptable for different platforms like in helicopters or in drones (UAVIS).

Furthermore, the created interface has to be legible and flexible, to be easily adjusted regarding the utilisation. We use the existing one created by F. Sigernes. The one shown on Fig.3 above is the final one with all the necessary options we need. The first one was simpler, without the bias tracking and without the low pass filter neither.

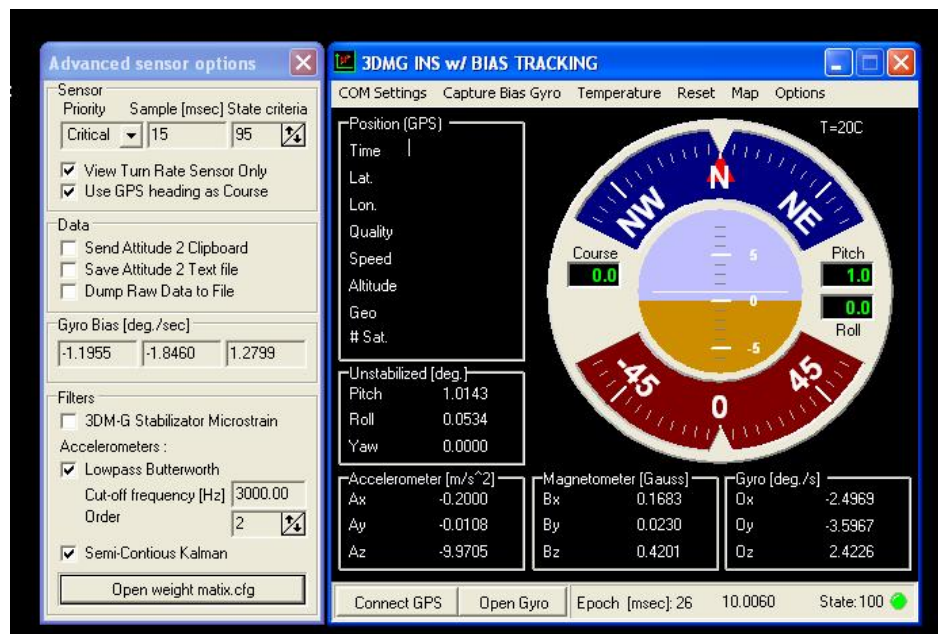
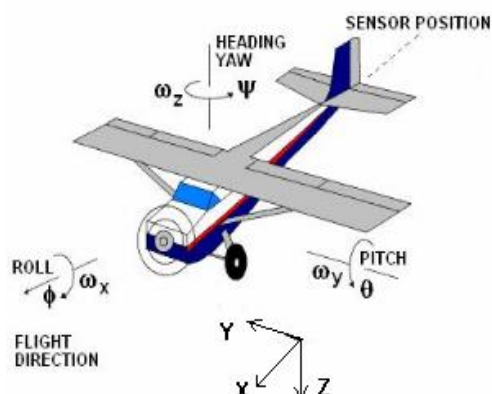


Figure 3. A snapshot of the program interface.

The interface window contains numerous options which were necessary in order to implement the project.

#### e) A theoretical difficulty: the “Gimbal lock”



We introduced previously that the reference coordinate systems in which the attitude is defined, is the body frame coordinate system. The attitude is described by 3 angles, the Pitch, the Roll and the Yaw. The Yaw is the course or the direction we are heading towards. Pitch is nose up or down. Pitch is wing up or down. These three angles are called the Euler angles.

Figure 4. The 3 Euler angles (Roll, Pitch, Yaw) define the attitude of the plane.

As a consequence, it is necessary to use a technical model regarding these angle coordinates.

A simple model that would describe the plane's attitude is simple rotations around three axis. But we have to consider that a rotation in 3 Dimensions is not commutative.

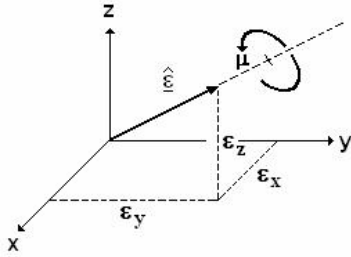
For example, to illustrate this non-commutability, we can consider a 3 orthogonal direct space coordinates system (x, y, z) and 2 rotation matrices,  $\underline{A}$  and  $\underline{B}$  such that

$$\underline{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \text{ and } \underline{B} = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.1)$$

$\underline{A}$  corresponds to an x axis rotation with an angle  $\theta$  and  $\underline{B}$  is a rotation around the z-axis with an angle  $\varphi$ . If we multiply these 2 matrices together, we obtained the following results:

$$\underline{A} \cdot \underline{B} = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi \cos \theta & \cos \varphi \cos \theta & \sin \theta \\ -\sin \varphi \sin \theta & -\cos \varphi \sin \theta & \cos \theta \end{bmatrix} \text{ and } \underline{B} \cdot \underline{A} = \begin{bmatrix} \cos \varphi & \sin \varphi \cos \theta & \sin \varphi \sin \theta \\ -\sin \varphi & \cos \varphi \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (2.2)$$

We don't have the equality between the two products. It illustrates the non-commutability. Thus, if we consider a simple rotation model, we have a problem about the order of the considered rotations, which is called the "Gimbal lock".



The Gimbal lock can be solved using a mathematical object, called "quaternion", based on the Euler theorem, which established that "a solid rotation is equivalent to a rotation along one fixed axis  $\vec{\varepsilon}$  (the eigen axis) with a magnitude  $\mu$  (the eigen angle)".

The quaternion describes a 3D rotation with a 4 dimensional state vector.

We define it as  $q = [\cos \mu, \vec{\varepsilon} \sin \mu]$ . More information about the quaternions are found in appendix B.

A quaternion is a four dimensional state vector. It is defined as

$$\underline{q}_k^T = [q_s \ q_x \ q_y \ q_z] \quad \begin{cases} q_s = \cos \mu \\ q_x = \vec{\varepsilon}_x \sin \mu \\ q_y = \vec{\varepsilon}_y \sin \mu \\ q_z = \vec{\varepsilon}_z \sin \mu \end{cases} \text{ and } \vec{\varepsilon} = \vec{\varepsilon}_x + \vec{\varepsilon}_y + \vec{\varepsilon}_z \quad (2.3)$$

All these restrictions have to be taken into account for the estimator's design. Thus the number of solutions is limited and three will be presented: two basic ones in the next chapter and a hybrid one later. We will see that the hybrid estimator is a combination of the 2 basic ones.

### 3. The different estimators

Based on the quaternion theory, it is possible to build 2 models of estimators: a model deducing directly from the first derivative of quaternion equation called the Hughes equation (appendix C), and another model using a Kalman filter.

In contrast to a static frame of reference, a body in a dynamic or moving system is influenced by random accelerations / forces from nearly all directions. The body's attitude obtained by a static sensor will as a consequence produce errors. A pendulum mounted inside a car can be used as an example to illustrate this. If the car is at rest or moving with constant speed, the pendulum's angle with the vertical represents a good measure of the car's tilt angle. On the other hand, when the car is accelerated or decelerated the pendulum will start to swing as a response to the forces acted up on it. As a result, large tilt errors occur.

In order to obtain the correct attitude, it is necessary to measure both the acceleration and the turn rate. That is why we will use a typical sensor, composed as a 3 axis solid state accelerometer combined with turn rates gyros – often called Inertial Measurements Units (IMU). Usually the sensors are quite noisy. Thus it is necessary to use filters, such as Kalman filters.

#### 3.1. KALMAN FILTERING

##### *Introduction*

The Kalman filter may be explained by the following analogy. A man sleeping in a known dark room wakes up in the middle of the night and wants to go out of the room. The room lights are out, so he has to find the exit door in total darkness. He has *a priori* information on where the door is located. He makes an *estimation* based on this information and starts to walk in one direction. After a short while, he walks straight onto a wall. This is regarded as a *measurement*. He now knows a little bit more where he is located in the room. In other words, he *updates* his initial estimate with the measurement. At this point a *prediction* of a new direction is possible, based on his knowledge of the room (*room model*). The process is repeated with measurements and will consequently improve his chances of finding the door.

The previous is the main essence of Kalman filtering. It is based on sequential adjustment in the static case (appendix A). All observations up to time  $t$  are used to obtain optimal estimations of the unknowns. Fortunately, we do not need to store any data as we proceed to the next estimate.

Firstly, a discrete approach of the filter is developed.

##### *The system model*

In a dynamic system such as an airplane, the unknown parameters are the attitude (roll, pitch and yaw), the coordinates and the velocity. These form the elements of the state vector. The state vector is time dependent, and it may be predicted for any instant  $k$  by means of system equations. The predicted values are then updated by the use of observations, which contain information about the true state vector.

In a discrete form the state vector is defined as  $\underline{x}_k^T = [x_1 \cdots x_n]_k$ . It consists of  $n$  unknown variables. The model for prediction is assumed to be a linear as a function of the previous state

$$\underline{x}_{k+1} = \underline{T}_k \underline{x}_k + \underline{w}_k \quad (3.1.1)$$

$\underline{w}_k^T = [w_1 \cdots w_n]_k$  is the noise of the system. The noise of the state vector could for example in an airplane be motor vibrations. Eq. (3.1.1) is often called the Markov process where  $\underline{T}_k$  is the transition matrix.

On the observational side, the *measurement* model is given by

$$\underline{l}_k = \underline{A}_k \underline{x}_k + \underline{n}_k \quad (3.1.2)$$

$\underline{l}_k$  is the measurement vector.

The measured noise  $\underline{n}_k$  is due to the sensors themselves (sensor noise).  $\underline{A}_k$  is called the measurement matrix. This measurement will improve the estimated state vector, as shown in the appendix A.

### ***The Kalman equations***

The updated estimated state vector  $\hat{\underline{x}}_k$  is a linear combination with the *a priori* state  $\hat{\underline{x}}_k^-$ , estimated regarding only on the previous state vector via the Markov process, and the error of measured prediction. The relation was established previously with Eq. (A.20) in the appendix A.

$$\hat{\underline{x}}_k = \hat{\underline{x}}_k^- + \underline{K}_k (\underline{l}_k - \underline{A}_k \hat{\underline{x}}_k^-) \quad (3.1.3)$$

$\underline{A}_k \hat{\underline{x}}_k^-$  is taken into account as a prediction of  $\underline{l}_k$ .

The main problem in Eq. (3.1.3) is to find the optimal  $\underline{K}_k$ . This is done by minimizing the trace of the error covariance matrix,  $\underline{Q}_k$  with respect to  $\underline{K}_k$ . This can be done since the trace of  $\underline{Q}_k$  is a sum of the mean square errors in the estimates of all the elements of  $\underline{x}_k$

$$\frac{\partial \text{trace}(\underline{Q}_k)}{\partial \underline{K}_k} = 0 \quad (3.1.4)$$

By definition the error covariance matrix of the updated state vector is

$$\underline{Q}_k \cong E[(\underline{x}_k - \hat{\underline{x}}_k)(\underline{x}_k - \hat{\underline{x}}_k)^T] \quad (3.1.5)$$

Correspondingly, the error covariance matrix of the estimated state vector is

$$\underline{Q}_k^- \cong E[(\underline{x}_k - \hat{\underline{x}}_k^-)(\underline{x}_k - \hat{\underline{x}}_k^-)^T] \quad (3.1.6)$$

$\underline{Q}_k$  can be obtained using Eq. (3.1.3)

$$\underline{x}_k - \hat{\underline{x}}_k = \underline{x}_k - \hat{\underline{x}}_k^- - \underline{K}_k (\underline{A}_k \underline{x}_k + \underline{n}_k - \underline{A}_k \hat{\underline{x}}_k^-) \quad (3.1.7)$$

or simply

$$\underline{x}_k - \hat{\underline{x}}_k = (\underline{x}_k - \hat{\underline{x}}_k^-) - \underline{K}_k \underline{A}_k (\underline{x}_k - \hat{\underline{x}}_k^-) - \underline{K}_k \underline{n}_k \quad (3.1.8)$$

Eqs. (3.1.7) and (3.1.8) are inserted into Eq. (3.1.5) and yields

$$\underline{Q}_k = \underline{Q}_k^- - \underline{Q}_k^- \underline{A}_k^T \underline{K}_k^T - \underline{K}_k \underline{A}_k \underline{Q}_k^- + \underline{K}_k \underline{A}_k \underline{Q}_k^- + \underline{K}_k \underline{A}_k \underline{Q}_k^- \underline{A}_k^T \underline{K}_k^T + \underline{K}_k \underline{R}_k \underline{K}_k^T \quad (3.1.9)$$

Or simply

$$\underline{Q}_k = (\underline{Q}_k^- - \underline{K}_k \underline{A}_k \underline{Q}_k^-) (\underline{I} - \underline{A}_k^T \underline{K}_k^T) + \underline{K}_k \underline{R}_k \underline{K}_k^T \quad (3.1.10)$$

From the above expression is possible to compute Eq. (3.1.4). The derivation rules are listed in appendix D.

$$\frac{\partial [\text{trace}(\underline{Q}_k)]}{\partial \underline{K}_k} = -2 \underline{Q}_k^{-T} \underline{A}_k^T + 2 \underline{K}_k \underline{A}_k \underline{Q}_k^{-T} \underline{A}_k^T + 2 \underline{K}_k \underline{R}_k = 0 \quad (3.1.11)$$

The optimal  $\underline{K}_k$  used in the Eq. (3.1.3) is then

$$\underline{K}_k = \underline{Q}_k^{-T} \underline{A}_k^T (\underline{A}_k \underline{Q}_k^{-T} \underline{A}_k^T + \underline{R}_k)^{-1} \quad (3.1.12)$$

This is the same gain factor that we found in the appendix A, Eq. (A.20).

Furthermore, the updated error covariance matrix is calculated by inserting Eq. (3.1.12) in (3.1.9)

$$\underline{Q}_k = (\underline{I} - \underline{K}_k \underline{A}_k) \underline{Q}_k^{-} \quad (3.1.13)$$

The error covariance matrix has decreased. At this point we are finally able to calculate the updated state vector with measurements by Eq. (3.1.3). The predicted state vector is computed by the model equation

$$\hat{\underline{x}}_{k+1}^{-} = \underline{T}_k \hat{\underline{x}}_k \quad (3.1.14)$$

based on the Markov process (3.1.1), neglecting the model noise. The predicted error covariance matrix before the measurement  $\underline{L}_{k+1}$  then becomes

$$\underline{Q}_{k+1}^{-} = E[(\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^{-})(\underline{x}_{k+1} - \hat{\underline{x}}_{k+1}^{-})^T] = \underline{T}_k \underline{Q}_k \underline{T}_k^T + \underline{W}_k \quad (3.1.15)$$

### Kalman algorithm – filter sequence

To summarize, the filter sequence is based on 4 steps: Estimation, Measurements, Updates and Prediction.

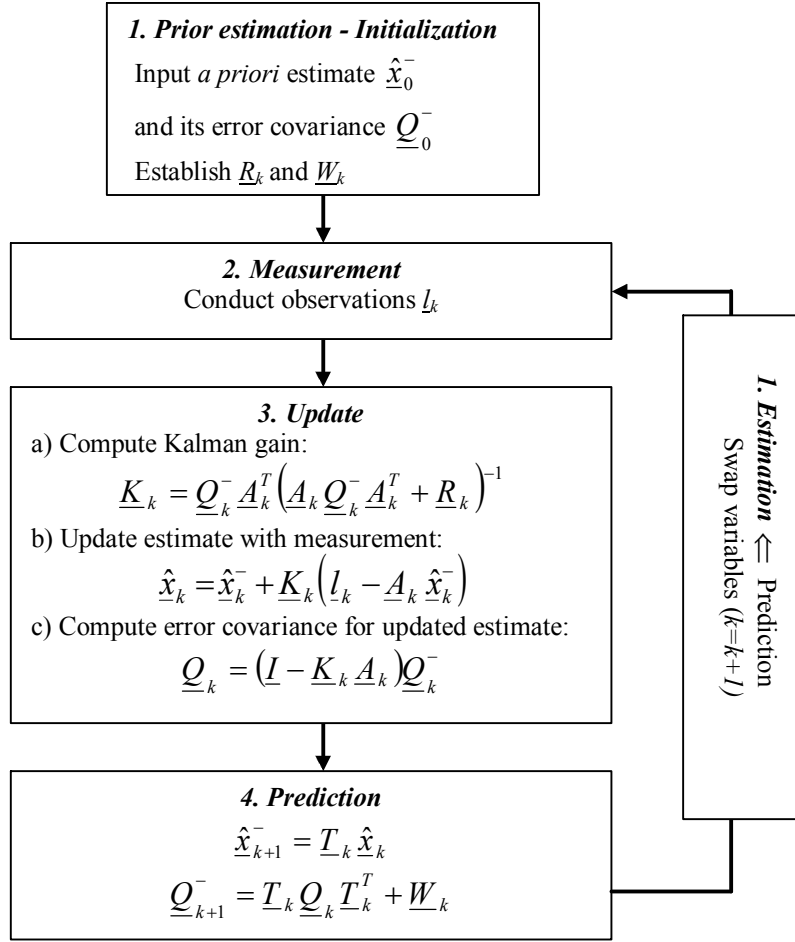


Figure 5. The Kalman filter float diagram

### 3.2. AIRBORNE ATTITUDE ESTIMATION

#### Axis configuration

The aim of this study is to obtain an estimate of the attitude of our airborne instruments. The measurements need to include both the acceleration and the turn rate in 3 dimensions (3D).

Fig.6 shows the fixed frame coordinate system of the aircraft. The sensors are 3 orthogonal mounted rotational rate sensors (gyros) and 3 accelerometers. The x-axis is out of the nose of the aircraft, the y-axis points out the right wing, and z-axis is down through the center of mass. The roll angle  $\phi$  is defined positive (+) with right wing down. The pitch angle  $\theta$  is defined similarly with nose up as positive direction.  $\varphi$ , yaw or

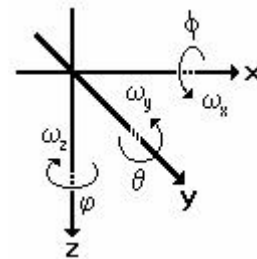


Figure 6. Axis.  $\phi$  is roll,  $\theta$  pitch and  $\varphi$  yaw.  $\omega$  is the angular turn rate [deg./s].

heading, is positive in the clockwise direction when viewed from above. The angular body rates are as follows.  $\omega_x$  is the roll rate (angular velocity) about the x-axis with positive direction as right wing down.  $\omega_y$  is the pitch rate about the y-axis with positive direction defined as nose up. Correspondingly,  $\omega_z$  is the yaw rate about the z-axis, positive in the clockwise direction when viewed from above.

### Measurements

The measured attitude or the pitch, roll and yaw angles of the aircraft are calculated from the accelerometer and the GPS

$$\underline{l}_k = \begin{bmatrix} \phi \\ \theta \\ \varphi \end{bmatrix}_k = \begin{bmatrix} \text{ArcTan}(-a_y / a_z) \\ \text{ArcSin}(-a_x / \sqrt{a_x^2 + a_y^2 + a_z^2}) \\ \text{GPS heading} \end{bmatrix}_k, \quad (3.2.1)$$

where  $a_i \forall i \in [x, y, z]$  represents the three axis's accelerometer readings. These angles are also known as the Euler angles.

### The state vector

In order to obtain smooth and continuous attitude estimates as a function of time, quaternion mathematics is used.

$$\underline{q}_k^T = [q_s q_x q_y q_z]_k. \quad (3.2.2)$$

A short summary of the nature and behaviour of quaternions are found in appendix B.

### The measurement model

The measurements are related to the observational model as follows

$$\underline{l}_k = \begin{bmatrix} \phi \\ \theta \\ \varphi \end{bmatrix}_k = \underline{A}_k \cdot \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix}_k + \underline{n}_k = \underline{A}_k \cdot \underline{q}_k + \underline{n}_k. \quad (3.2.3)$$

The measurement matrix  $\underline{A}_k$  is defined as

$$\underline{A}_k \cong \begin{bmatrix} \frac{\partial \phi}{\partial q_s} & \frac{\partial \phi}{\partial q_x} & \frac{\partial \phi}{\partial q_y} & \frac{\partial \phi}{\partial q_z} \\ \frac{\partial \theta}{\partial q_s} & \frac{\partial \theta}{\partial q_x} & \frac{\partial \theta}{\partial q_y} & \frac{\partial \theta}{\partial q_z} \\ \frac{\partial \varphi}{\partial q_s} & \frac{\partial \varphi}{\partial q_x} & \frac{\partial \varphi}{\partial q_y} & \frac{\partial \varphi}{\partial q_z} \end{bmatrix}_k \quad (3.2.4)$$

This matrix is also called a transition matrix since it permits us to convert from quaternion- to Euler frames of reference. The mathematics behind the construction of  $\underline{A}_k$  is non- trivial, but doable. It is Jacobian matrix (see appendix A, Eq. (A.19)).



### ***The system model - prediction***

The instantaneous rate change of an airborne quaternion is given by Hughes (1986)

$$\dot{\underline{q}}_k = \frac{1}{2} \underline{\Omega}_k \cdot \underline{q}_k, \quad (3.2.5)$$

where  $\underline{\Omega}_k$  is the quaternion omega matrix for the current rotational sensors

$$\underline{\Omega}_k = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}_k. \quad (3.2.6)$$

$(\omega_x, \omega_y, \omega_z)_k$  represent the rolling, pitching and yawing turn rates in the strap down inertial reference frame, as measured by the rotational rate sensors at the moment  $t_k$ .

The transition matrix  $\underline{T}_k$  is then given as

$$\underline{T}_k = \underline{I} + \frac{1}{2} \underline{\Omega}_k \cdot \Delta t_k, \quad (3.2.7)$$

where  $\Delta t_k = t_{k+1} - t_k$  is the epoch or sampling period. The transition matrix satisfies the relation

$$\underline{q}_{k+1} = \underline{T}_k \underline{q}_k + w_k, \quad (3.2.8)$$

which is the Markov process or the predictive state of the Kalman filter. Note that the transition matrix is updated at every epoch by the use of measurements from the gyros.

The fundamental Kalman relations are now established by Eqs. (3.2.3) and (3.2.8). The final filter sequence is shown in Fig. 7. The explanation on time evolution of quaternions (Hughes equation) is formulated in appendix C. The transition matrix Eq. (3.2.7) is based on a semi-continuous predictive approach - described in appendix E.

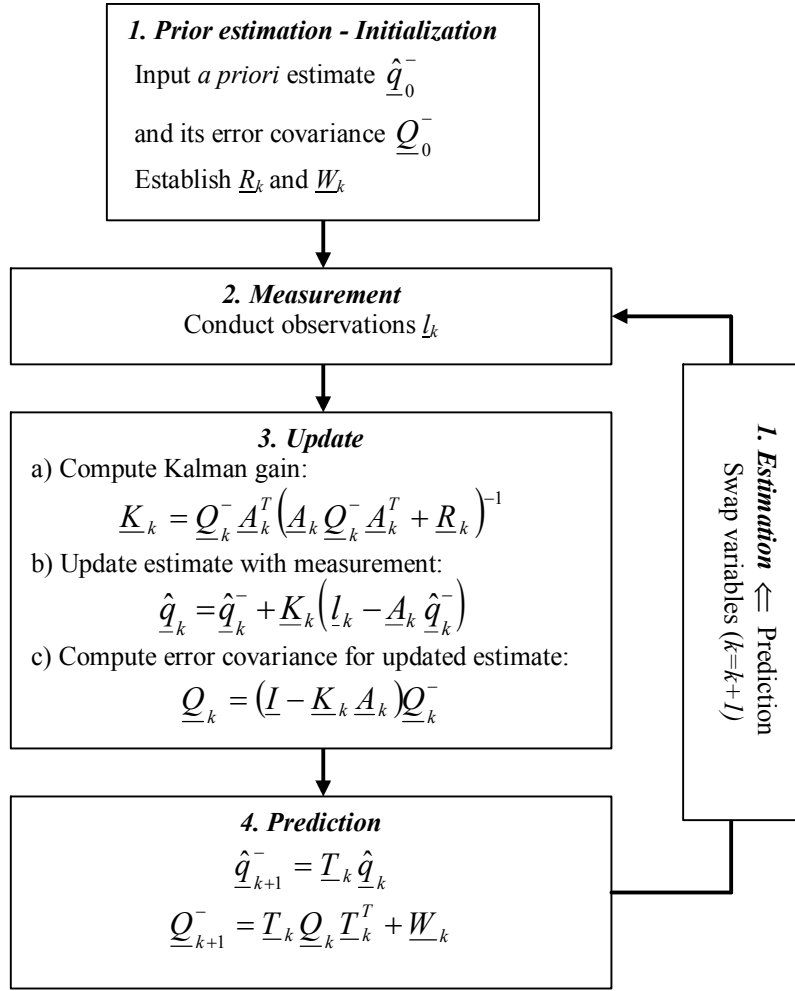


Figure 7. The Kalman filter float diagram for airborne attitude control.

$\underline{W}_k$	Noise model covariance matrix
$\underline{R}_k$	Noise measurement covariance matrix
$\underline{l}_k$	Measurements (Euler angles)
$\underline{K}_k$	Kalman gain
$\hat{\underline{q}}_k$	Updated estimation with measurements $\underline{l}_k$
$\underline{Q}_k$	Updated error covariance matrix
$\hat{\underline{q}}_k^-$	Estimation before measurement $\underline{l}_k$
$\underline{Q}_k^-$	Estimation error covariance matrix before measurement $\underline{l}_k$
$\hat{\underline{q}}_{k+1}^-$	Prediction before measurement $\underline{l}_{k+1}$
$\underline{Q}_{k+1}^-$	Prediction error covariance matrix before measurement $\underline{l}_{k+1}$
$\underline{T}_k$	Markov matrix
$\underline{A}_k$	Measurement matrix

This is a discrete/semi-continuous approach of the Kalman filter. It can be considered as a solution to estimate the attitude of the plane. The data are processed by a computer whose sampling period is 25 ms. It is really small compared to the reaction time of the plane; considering the Nyquist-Shanon theorem, it corresponds to a maximal detectable frequency of rolling/pitching/yawing of 20Hz. An airplane acting normally, should not move so quickly. The Yawing, Pitching and Rolling should have slow and smooth variations. Thus a semi-continuous Kalman filter with such properties can be a solution.

Furthermore, to run a Kalman filter, it is necessary to initialise it; we need  $\underline{W}_k$  (in this case, there is no model noise, thus this matrix is nul),  $\underline{R}_k$ ,  $\underline{Q}_0^-$  and  $\hat{\underline{q}}_0^-$ . To determine these matrices, an equipments analysis is necessary. One is presented in the appendix F.

### ***Why not use the continuous Kalman filter?***

A continuous Kalman filter exists. The attitude of a plane is a continuous variable, thus the continuous filter might be a good alternative.

It is possible to make equivalence with the discrete one as shown on Fig. 8. The notations are the same ones, without any indexes, because of continuity reasons.

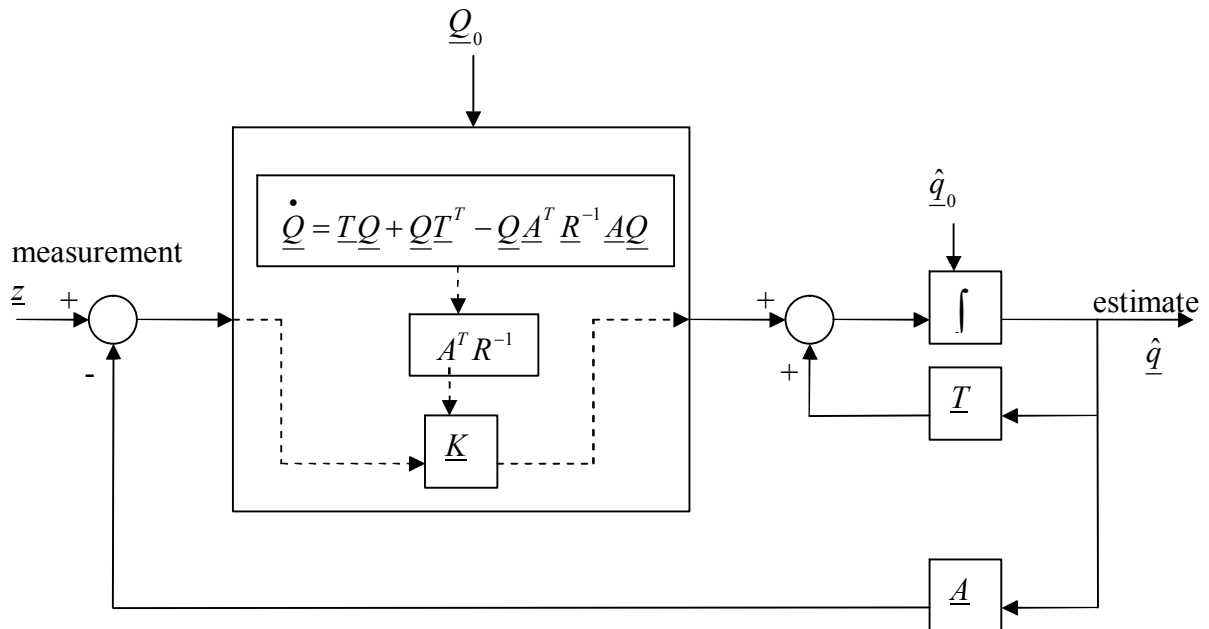


Figure 8. Simplified diagram of a continuous Kalman filter process.

The point of this figure is not to demonstrate the formulas of the continuous Kalman filter, but to underline the equation concerning the covariance propagation, used to compute the Kalman gain. Because of the last term of this equation, the differentiate equation is not linear. It is possible to solve it, with a Riccati process. **But it has to be done off-line.**

The estimator has to process the data in real time. Thus **the continuous Kalman filter is not an adequate solution.**

### 3.3. THE INTEGRATED QUATERNION MODEL.

#### *Introduction:*

There is another kind of estimators based on a different approach from the Kalman filter's one. The basic model is the same one, but the computation of the attitude parameters is even faster, because of its simplicity.

In this model only 2 sensors are used; the GPS and the turn rate sensors. We assume the turn rate sensor as nearly perfect, it means without any bias, or at least this bias has to be constant during the experiment time. (c.f. appendix F)

#### *The integration model:*

To explain this method, the quaternion is still used as a state vector. This model is based on the Hughes equation defined previously in Eq. (3.2.5). Thus we have

$$\dot{\underline{q}} = \frac{1}{2} \cdot \underline{\Omega} \cdot \underline{q}$$

with the same matrix  $\underline{\Omega}$ . A Taylor development to the first order provides us with

$$\underline{q}(t) = (\underline{I}_4 + \frac{1}{2} \cdot \underline{\Omega} \cdot (t - t_0)) \underline{q}(t_0)$$

where  $\underline{I}_4$  is the four dimensional identity matrix and  $\underline{q}(t_0)$  the initial quaternion. For this method we still use a computer, thus it is necessary to introduce a sampling. The turn rate data are transmitted to the computer every 25 ms, which is a quite good sampling period (we underlined it previously). We obtain

$$\underline{q}_{k+1} = (\underline{I}_4 + \frac{1}{2} \cdot \underline{\Omega}_k \cdot \Delta t_k) \underline{q}_k$$

in which  $\Delta t_k = t_{k+1} - t_k$ , and the state vector defined at the corresponding epoch regarding on the index.

To determine the yaw, we could use the turn rate sensor, but for the initialisation anyway, we need a compass, the GPS one. Thus we continue to use the compass to estimate it. The initial quaternion is  $\underline{q}_0 = [1 \ 0 \ 0 \ 0]^T$ . It means no initial input rotation and the system has to be initialised on a perfectly flat attitude.

### 3.4. PRIOR COMPARISON OF THESE 2 APPROACHES:

To attest of the validity of these 2 estimators, one simple experiment is done. We put the sensor on a table, with a known attitude and we shake it. It tests the accuracy of the displayed attitude and the reaction of the system in dynamic conditions. The true attitude is constant (Roll=0 deg; Pitch=0 deg).

Regarding the integration model, we took into account the bias of the gyroscope. We approximated it as a constant over time regardless of the fact that the equipments analysis shows it is not a good assumption.

The yaw is not represented because we performed the experiment indoor, thus it was impossible to use the GPS.

The results are shown on Fig. 9 next page.

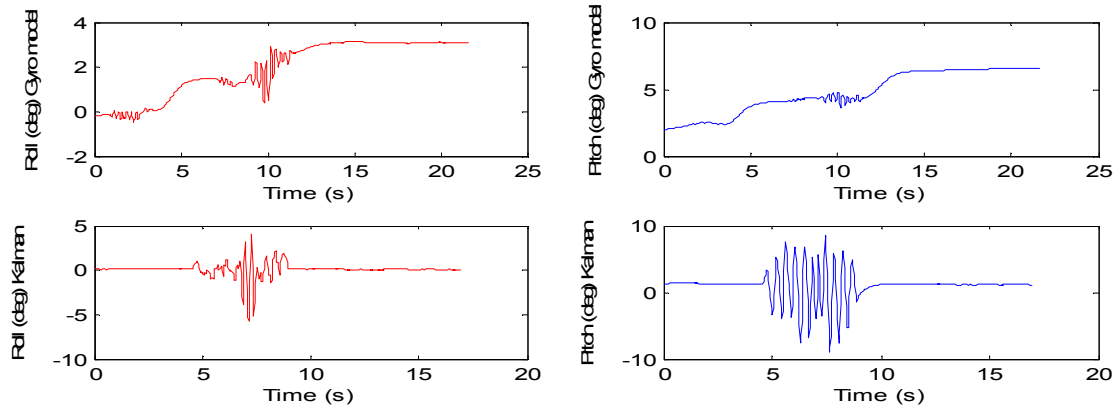


Figure 9..Estimated attitude regarding on the 2 models shaking the sensor on a flat attitude. Top: integration model, bottom: Kalman filter. (red: Roll, blue: Pitch).

Many different properties appear on this figure:

- **The inertial forces:** the Kalman filter reacts strongly to the shaking. The fact we use the accelerometer to update the estimation makes it sensitive to these inertial forces.
- **The drift:** the integration model is drifting; it is due to the gyroscope bias. Apparently the turn rate sensor reacts to the shaking drifting. Nevertheless, the reaction is quite limited. The bias approximation is not perfect either.
- **The estimation bias:** at the beginning of the experiment, the estimated attitudes are perfect. The couple (Roll, Pitch)  $\approx (0, 1)$  is well estimated by both solutions. As soon as no drift was observed the integration model is not biased, but it keep in memory the mistakes of past estimations and thus become biased as soon as an error occurs. The Kalman filter is auto corrected and not biased, because of the updating process occurring for each sample.
- **The real time:** both solutions act in real time. There is no delay between the shaking and the estimation.
- **Continuity:** the estimated attitude is rather smooth. The sampling is small enough to describe the plane attitude. No steps are visible.

We can compose the following comparative table:

	Integration model	Kalman filter
<b>Inertial reaction</b>	no (limited)	yes
<b>Drift</b>	yes	no
<b>Robustness</b>	+	+++
<b>Complexity</b>	+	+++
<b>Continuity</b>	yes	yes
<b>Real time</b>	yes	yes
<b>Initialisation</b>	Flat attitude	No request.

Table 1: qualitative comparative analysis of the 2 models.

This simple experiment is sufficient to be aware of the limits of each estimator. At this stage in the project, it is not necessary to go deeper into the comparison. Neither solution is good enough. Nevertheless, it appears clearly that these 2 models complete each other perfectly. Thus we have to consider a third estimator which is a combination of these both ones. Then a more through analysis will be performed.

## 4. The Combined Estimator

This estimator is a combination of the ones described previously. This combined approach permits to get the different advantages of each previous solution, and at the same time, to get rid of their defaults. It should provide us a better estimation.

Nevertheless, to use the integration model, it is necessary to know the bias of the gyroscope to get rid of the drift. On the appendix F, it is underlined that this bias is temperature dependent. The temperature varies over the time. As a consequence the gyro bias is also time dependent. Thus it is necessary to track the bias during the experiment or the flight. A flight can be long, therefore the temperature may change while, by consequence the bias also.

Thus, first of all the bias has to be tracked in real time, before combining the previous estimators.

This was done using an enhanced version of the Kalman filter.

### 4.1. THE ENHANCED KALMAN FILTER

The essence of this filter is to track the gyroscope bias. It is really simple to implement it, because the measurements are the same ones used by the first Kalman filter estimating the attitude. However, it is necessary to create a new model.

#### *The Markov process:*

We defined the following state vector at the time  $t_k$  by

$$\underline{X}_k = \begin{bmatrix} \underline{W}_{bias} \\ \underline{q} \end{bmatrix}_k = \begin{bmatrix} w_{x\_bias} \\ w_{y\_bias} \\ w_{z\_bias} \\ q_s \\ q_x \\ q_y \\ q_z \end{bmatrix}_k = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}_k \quad (4.1.1)$$

with  $\underline{q}$  the previous quaternion vector and  $\underline{W}_{bias}$  represents the bias of the gyroscope.

We build the model in dynamic conditions. We have the following equations for the bias

$$\dot{\underline{W}}_{bias} = \begin{bmatrix} \dot{w}_{x\_bias} \\ \dot{w}_{y\_bias} \\ \dot{w}_{z\_bias} \end{bmatrix}_k = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_k \quad (4.1.2)$$

We consider that the gyroscope bias is not time dependent between two consecutive epochs. This hypothesis seems to be quite accurate regarding the results obtained from the equipments analysis. As soon as the temperature does not change, the gyroscope bias is quite constant. Thus between 2 epochs ( $\approx 25 \text{ ms}$ ) the gyro bias is assumed to be constant.

The quaternion's model is still based on the Hughes equation, in which the bias is inserted:

$$\dot{\underline{q}}_k = \frac{1}{2} \underline{\Omega}_k \cdot \underline{q}_k, \quad (4.1.3)$$

and  $\underline{\Omega}_k$  becomes

$$\underline{\Omega}_k = \begin{bmatrix} 0 & -(\omega_x - w_{x\_bias}) & -(\omega_y - w_{y\_bias}) & -(\omega_z - w_{z\_bias}) \\ (\omega_x - w_{x\_bias}) & 0 & (\omega_z - w_{z\_bias}) & -(\omega_y - w_{y\_bias}) \\ (\omega_y - w_{y\_bias}) & -(\omega_z - w_{z\_bias}) & 0 & (\omega_x - w_{x\_bias}) \\ (\omega_z - w_{z\_bias}) & (\omega_y - w_{y\_bias}) & -(\omega_x - w_{x\_bias}) & 0 \end{bmatrix}_k. \quad (4.1.4)$$

Using this Eq. (4.1.4), the differential system follows

$$\dot{\underline{q}}_k = \begin{bmatrix} \dot{q}_s \\ \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \end{bmatrix}_k = \begin{bmatrix} -\frac{1}{2} \cdot (\omega_x - w_{x\_bias}) \cdot q_x - \frac{1}{2} \cdot (\omega_y - w_{y\_bias}) \cdot q_y - \frac{1}{2} \cdot (\omega_z - w_{z\_bias}) \cdot q_z \\ + \frac{1}{2} \cdot (\omega_x - w_{x\_bias}) \cdot q_s + \frac{1}{2} \cdot (\omega_z - w_{z\_bias}) \cdot q_y - \frac{1}{2} \cdot (\omega_y - w_{y\_bias}) \cdot q_z \\ + \frac{1}{2} \cdot (\omega_y - w_{y\_bias}) \cdot q_s - \frac{1}{2} \cdot (\omega_z - w_{z\_bias}) \cdot q_x + \frac{1}{2} \cdot (\omega_x - w_{x\_bias}) \cdot q_z \\ + \frac{1}{2} \cdot (\omega_z - w_{z\_bias}) \cdot q_s + \frac{1}{2} \cdot (\omega_y - w_{y\_bias}) \cdot q_x - \frac{1}{2} \cdot (\omega_x - w_{x\_bias}) \cdot q_y \end{bmatrix}_k \quad (4.1.5)$$

Thus, according to the Eq.(4.1.2) and (4.1.5) it yields

$$\dot{\underline{X}}_k = \begin{bmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix}_k = \frac{1}{2} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ -(\omega_x - x_0)x_4 - (\omega_y - x_1)x_5 - (\omega_z - x_2)x_6 \\ + (\omega_x - x_0)x_3 + (\omega_z - x_2)x_5 - (\omega_y - x_1)x_6 \\ + (\omega_y - x_1)x_3 - (\omega_z - x_2)x_4 + (\omega_x - x_0)x_6 \\ + (\omega_z - x_2)x_3 + (\omega_y - x_1)x_4 - (\omega_x - x_0)x_5 \end{bmatrix}_k = f(\underline{X}, t_k) \quad (4.1.6)$$

This equation system is not linear, thus we have to linearise it, doing a Taylor development at the first order. It gives

$$\underline{X}_{k+1} = \underline{T}_k \underline{X}_k \text{ with } \underline{T}_k = \underline{I}_7 + \Delta t \cdot \left[ \frac{\partial f}{\partial \underline{X}} \right]_k \quad (4.1.7)$$

The above equation provides us the fundamental model equation for the extended Kalman filter. In Eq.(4.1.7), the 7 dimensions identity matrix appears,  $\underline{I}_7$ , and the epoch  $\Delta t$  as well.

Firstly  $\frac{\partial f}{\partial \underline{X}}$  may be computed.

$$\left[ \frac{\partial \underline{f}}{\partial \underline{X}} \right]_k = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ +x_4 & +x_5 & +x_6 & 0 & -(\omega_x - x_0) & -(\omega_y - x_1) & -(\omega_z - x_2) \\ -x_3 & +x_6 & -x_5 & (\omega_x - x_0) & 0 & (\omega_z - x_2) & -(\omega_y - x_1) \\ -x_6 & -x_3 & +x_4 & (\omega_y - x_1) & -(\omega_z - x_2) & 0 & (\omega_x - x_0) \\ +x_5 & -x_4 & -x_3 & (\omega_z - x_2) & (\omega_y - x_1) & -(\omega_x - x_0) & 0 \end{bmatrix}_k \quad (4.1.8)$$

We should underline that the matrix  $\underline{\Omega}_k$  appears, which is coherent: the quaternion vector is still present in the global state vector. The Markov matrix can be found from Eq. (4.1.7)

$$T_k = \frac{1}{2} \cdot \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ +x_4 \cdot \Delta t & +x_5 \cdot \Delta t & +x_6 \cdot \Delta t & 2 & -(\omega_x - x_0) \cdot \Delta t & -(\omega_y - x_1) \cdot \Delta t & -(\omega_z - x_2) \cdot \Delta t \\ -x_3 \cdot \Delta t & +x_6 \cdot \Delta t & -x_5 \cdot \Delta t & (\omega_x - x_0) \cdot \Delta t & 2 & (\omega_z - x_2) \cdot \Delta t & -(\omega_y - x_1) \cdot \Delta t \\ -x_6 \cdot \Delta t & -x_3 \cdot \Delta t & +x_4 \cdot \Delta t & (\omega_y - x_1) \cdot \Delta t & -(\omega_z - x_2) \cdot \Delta t & 2 & (\omega_x - x_0) \cdot \Delta t \\ +x_5 \cdot \Delta t & -x_4 \cdot \Delta t & -x_3 \cdot \Delta t & (\omega_z - x_2) \cdot \Delta t & (\omega_y - x_1) \cdot \Delta t & -(\omega_x - x_0) \cdot \Delta t & 2 \end{bmatrix}_k \quad (4.1.9)$$

The Markov matrix used in the first version of the Kalman filter appears inside the above matrix, Eq.(4.1.9). Tracking the bias does not change the dynamic evolution of the quaternion vector. Only that the bias is removed from the data read by the turn rate sensors.

### ***The measurement model:***

We still measure the Euler angles via the accelerometer; thus, the measurements are

$$\underline{l}_k = \begin{bmatrix} \phi \\ \theta \\ \varphi \end{bmatrix}_k \quad (4.1.10)$$

Then the new measurement matrix can be deduced from Eq.(4.1.10), giving

$$\underline{l}_k = \begin{bmatrix} \phi \\ \theta \\ \varphi \end{bmatrix}_k = \begin{bmatrix} 0_{3 \times 3} & \underline{A}_k \end{bmatrix} \cdot \begin{bmatrix} \underline{W}_{bias} \\ \underline{q} \end{bmatrix}_k + \underline{b}_k \quad (4.1.11)$$

with  $\underline{b}_k$  the noise measurement and  $\underline{A}_k$  the measurement matrix defined in the previous filter. The Euler angles are read via the accelerometer, as before. Thus  $\underline{b}_k = \underline{n}_k$  (measurement noise of the first Kalman filter).

Both fundamental models (Markov process and measurement model) used to build a Kalman filter are now established. The noise need to be determined in order to initialize the Kalman filter.



### Initialisation:

The noise measurement is the same as the first Kalman filter's one. As a consequence the noise measurement covariance matrix is the one established on the appendix F, hence

$$\underline{R}_k = \begin{bmatrix} \sigma_\phi^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\varphi^2 \end{bmatrix}_k = \begin{bmatrix} 1.0510 \cdot 10^{-5} & 0 & 0 \\ 0 & 1.3556 \cdot 10^{-5} & 0 \\ 0 & 0 & 3.74 \cdot 10^{-4} \end{bmatrix} \quad (4.1.12)$$

$\sigma^2$  is the variance.

Three more variables are processed in the enhanced filter compared with the first version. All the variables are assumed to be uncorrelated.

We consider as quaternion's initialisation the one established during the equipments analysis, we just have to add an initialisation for the bias.

We would have to know the temperature. For a given temperature, we choose a good approximation of the bias.

By example, referring to the equipments analysis, if the temperature is 32°C, the initial bias becomes

$$\underline{\hat{W}}_{bias0}^- = [-0.0046 \quad 0.0049 \quad 0.0216]^T \text{ rad / s} \quad (4.1.13)$$

Thus, our initial  $\underline{\hat{X}}_0^-$  is then

$$\underline{\hat{X}}_0^- = [-0.0046 \quad 0.0049 \quad 0.0216 \quad 1 \quad 0 \quad 0 \quad 0]^T \quad (4.1.14)$$

But it is really seldom that we start the program at exactly 32°C. Thus the used initial state vector is

$$\underline{\hat{X}}_0^- = [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^T$$

The bias does not have a proper initialization. Thus, the Kalman filter needs several loops to estimate it. On the dial of the displayed attitude, a jump of the attitude appears when we start the estimation. After 2 or 3 s, the bias is tracked.

To compute the prior error covariance matrix, we have to consider how the initial input estimation is established.

For the bias, the turn rate sensor is used, and for the quaternion the accelerometer

$$\underline{Q}_0^- = \begin{bmatrix} \sigma_{\omega_x}^2 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \sigma_{\omega_y}^2 & \ddots & & & & \vdots \\ \vdots & \ddots & \sigma_{\omega_z}^2 & \ddots & & & \vdots \\ \vdots & & \ddots & \sigma_{q_s}^2 & \ddots & & \vdots \\ \vdots & & & \ddots & \sigma_{q_x}^2 & \ddots & \vdots \\ \vdots & & & & \ddots & \sigma_{q_y}^2 & 0 \\ \vdots & & & & & \ddots & \sigma_{q_z}^2 \\ 0 & \dots & \dots & \dots & \dots & 0 & \sigma_{q_z}^2 \end{bmatrix} = \begin{bmatrix} 10^{-4} & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 10^{-4} & \ddots & & & & \vdots \\ \vdots & \ddots & 10^{-4} & \ddots & & & \vdots \\ \vdots & & \ddots & 5 \cdot 10^{-5} & \ddots & & \vdots \\ \vdots & & & \ddots & 5 \cdot 10^{-5} & \ddots & \vdots \\ \vdots & & & & \ddots & 5 \cdot 10^{-5} & 0 \\ \vdots & & & & & \ddots & 5 \cdot 10^{-5} \\ 0 & \dots & \dots & \dots & \dots & 0 & 5 \cdot 10^{-5} \end{bmatrix} \quad (4.1.15)$$

All the fundamental equations of the extended Kalman filter are now established.

This new improved version of the Kalman filter is quite relevant because no other measurements are necessary. The same ones as before are used which is an advantage in practice.

It underlines an aspect of the Kalman filter: it can improve itself. Regarding the measurement model equations, the measurements are only used to estimate the attitude, but not the bias. The quaternion vector itself is used to compute the gyroscope bias. It is a kind of auto regulation since the measurements correct the flawed estimation.

## Results

To control the efficiency of the bias tracking, we compare the output values of the turn rate sensor, when this one is not shacked, with the estimated bias.

If there is no movement, the turn rate sensor data correspond to the gyro bias. Fig.10 shows the results.

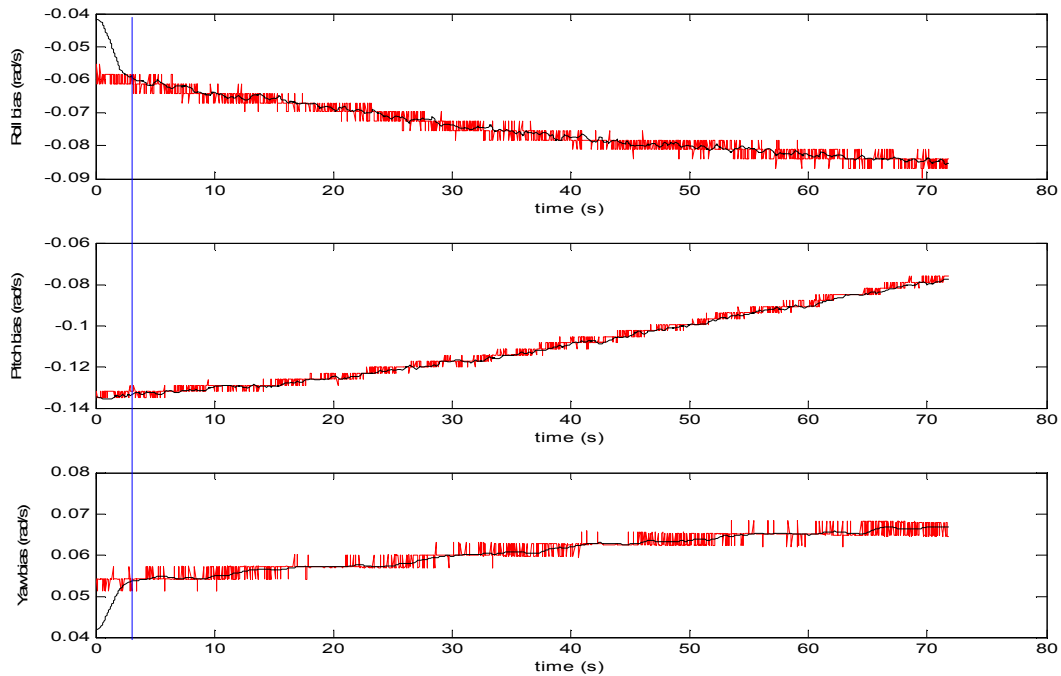


Figure 10. Estimated gyro bias via the filter (black) and measured bias via the turn rate sensor (red).

In Fig.10 it appears clearly that the estimated bias follows almost exactly the bias measurement. It means that the filter tracks perfectly the default of the gyroscope at least on static conditions.

The bias is not perfectly constant either. During the 70 s of the experiment (long time), the values changed significantly. We have to consider the units: rad/s.

A jump of values is present at the beginning of the bias tracking due to the initial values input to the Kalman filter for the bias state vectors components.

However in dynamic conditions, the bias is not perfectly estimated. This is obvious if only the left wing is moving up. Normally only the Roll should change, but a Pitch

drift is observed. This is due to a bad estimation of the bias in these conditions. The Fig.11 below illustrates this on the top graph.

When the right wing is going up, the Kalman filter processes the turn rate sensor variation as an increasing of the bias and not like an input. This is probably due to the accelerometer reaction which is not as fast as the turn rate sensor's one. It means that the measurements made via the accelerometer is not be taken into account when the rotation starts. Thus the gyro bias estimation is biased and there is an influence on the attitude via the Hughes equation defined previously.

To correct this, it is necessary to make a hypothesis! It is assumed **that the bias is constant during the wings motion** (short time). Thus the 2 waves observed on the top graph Fig.11 resulting from a wrong estimation should be removed.

Regarding this assumption, when the wing is rotating, we force the bias to stay constant using its previous value, computed during the steady state (static).

Steady condition means here a constant speed vector. To detect it, 2 criteria are necessary. The first one is detecting the inertial forces and the second one the solid rotations. They are describes in the appendix I.

After implementation of these conditions, we obtained the results, on the lower panel of Fig.11.

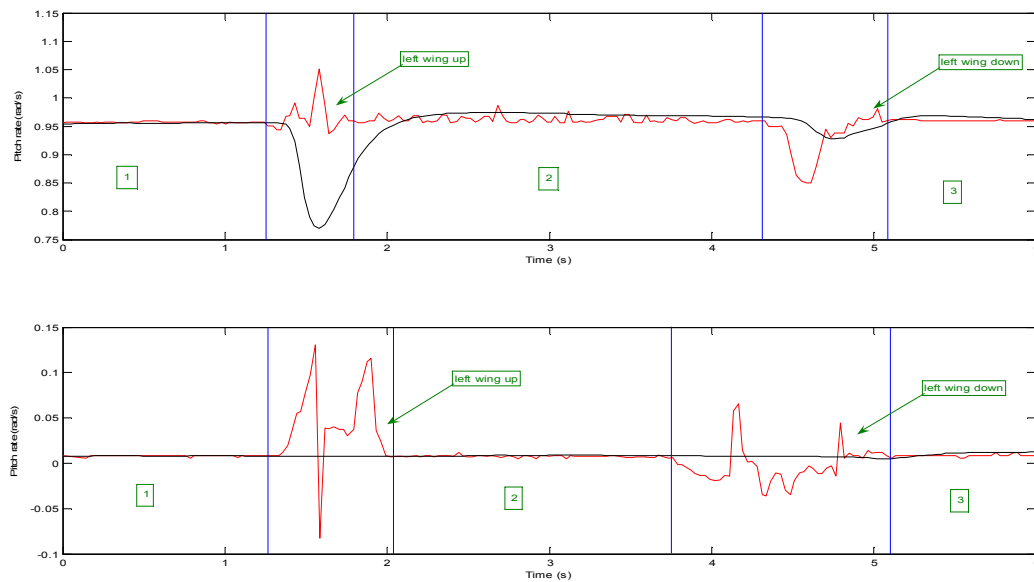


Figure 11. Pitch rate variations (red) and estimated bias (black). Top shows without any correction. Bottom is with differentiation between steady and non-steady conditions.

The bias estimation is perfectly smooth. Moreover the jump observed on the estimated attitude disappeared (10 times lower). Thus the hypothesis “the bias is constant during the wing motion” is validated (it means during a short time). The zones labelled 1,2 and 3 in Fig.11 corresponds to movements with steady conditions. The turn rate sensor provides the gyro bias because of absence of rotations and inertial forces.

With this enhanced Kalman filter, it is possible to track the bias of the gyroscope in real time. The access to this parameter is really important, because all the essence of the combined solution can make sense. The tracked bias can now be returned into the integration model of the quaternion, and the effect of external forces cancelled.

#### **4.2. THE COMBINED ESTIMATOR:**

This estimator consists to switch from the integration model to the Kalman filter, based on steady conditions or not.

It appeared on the first short analysis we made about the both models that the Kalman filter does not react very well to the shaking, but on steady conditions, it is the best model. On the other hand, the integration model behaves quite well during dynamic conditions. In addition to this, the fact that we have access to the gyroscope bias, the integration model improves accuracy significantly. The drift is removed.

Thus the main idea of the combined solution is to switch between these 2 models.

##### ***The main idea: to switch***

To define which model should be used to estimate the plane's attitude, both factors defined on the appendix I are used. The first factor makes us enable to get rid of the inertial acceleration problems and the second one detect if we are doing a solid rotation.

The most relevant is to use the integration model as the main one, parallel to the Kalman filter, providing the bias of the gyroscope. When the conditions are steady, the attitude is estimated via the Kalman filter, else via the integration model. Furthermore to update the integration model with the Kalman filter removes the error memory effect of the integration model. The Kalman's auto correction is by consequence applied on both models. By consequence, the estimation is not biased.

We had to solve a problem of continuity. When the switch occurs from one model to another one, both have independent evolutions, thus the obtained attitude is discontinuous, which is not correct.

To solve this problem, one solution is to force the integration quaternion model to be equal to the Kalman one when the movement is steady, and to force the Kalman state vector (quaternion, which means the 4 last components of the state vector) to be equal to the integration quaternion model in the opposite case (un-steady conditions). Thus the estimation is perfectly continuous.

To make it smoother, a low pass filter is used on the accelerometer. Thus the system is less noisy and perfectly smooth. The calibration of the low pass filter is described on the appendix G.

Fig.12 summarizes the process.

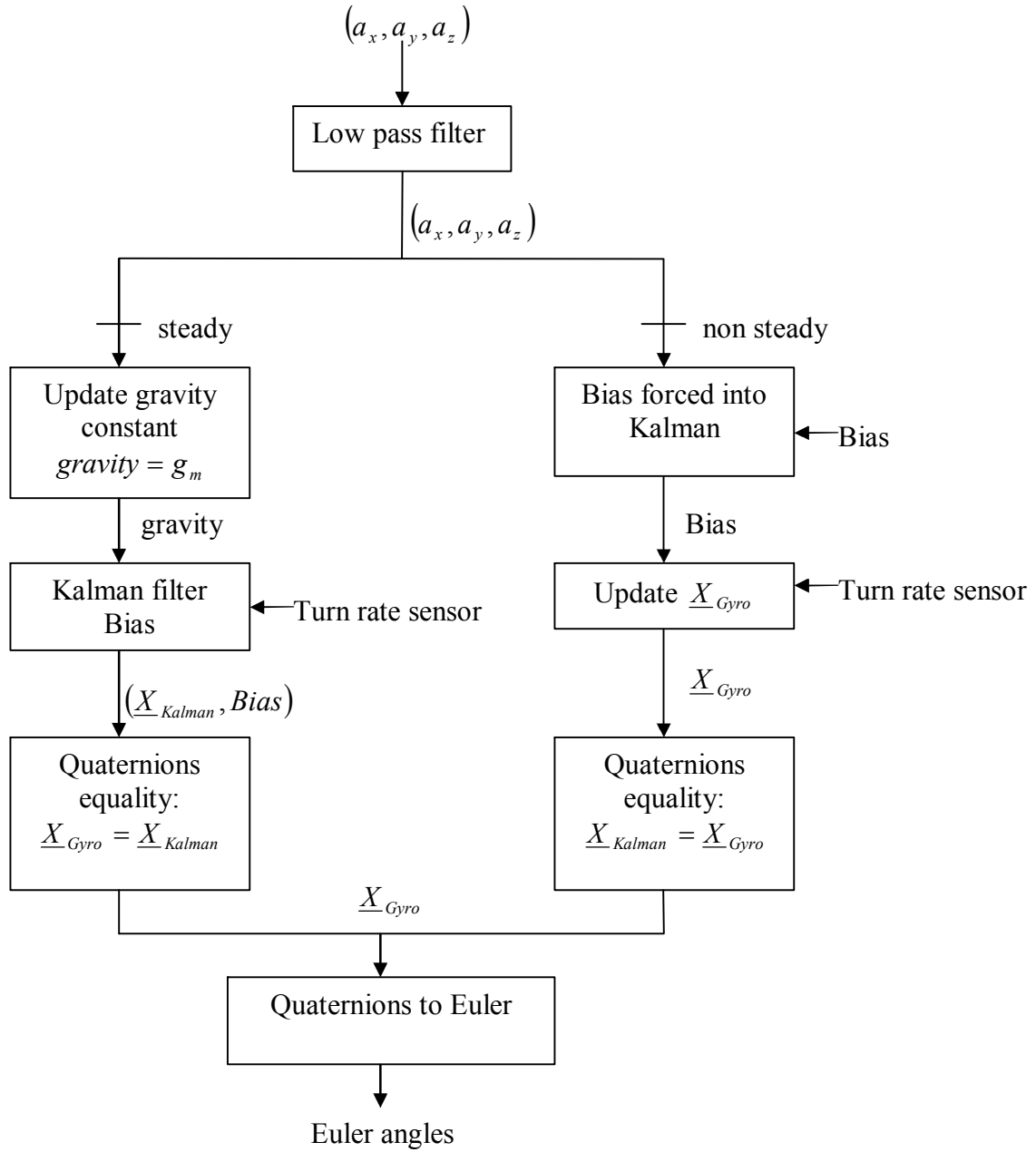


Figure 12. Code architecture used to obtain the best attitude estimation as possible. The steady and non steady conditions are described in the appendix I.

This is the model we choose to estimate as best as possible the attitude of the plane. The fact to combine 2 models with different characteristics permits us to take the advantages of each, removing their disadvantages.

## 5. Analysis of the different solutions

To compare the estimators, it was necessary to perform numerous tests with different concepts. The access to plane's data is quite difficult because of the cost. Thus some tests were made in a car or simply pushing a shopping cart. Nevertheless, first of all it was necessary to test the 3 estimators in the laboratory to compare their properties. The tests also include the Microstrain estimator, to give an idea of the relevance of our estimators.

### 5.1. BENCH TESTS:

A comparative analysis of these estimators was done regarding three different criteria:

- the stability and the accuracy
- the capacity to get rid of the inertial components
- the response time to a step

We underline that the integration model and the Kalman filter did not have perfect results regarding all these criteria with independent processing. We would like to control how the combined solution reacts.

#### *The noise: steady conditions*

To estimate the accuracy of the system and its noise, we processed to different experiments.

The first one was to determine whether or not the displayed attitude is correct. We put the system on different plans whose inclination varied. In Fig. 13, the sensor is Rolled with a magnitude of -8 deg.

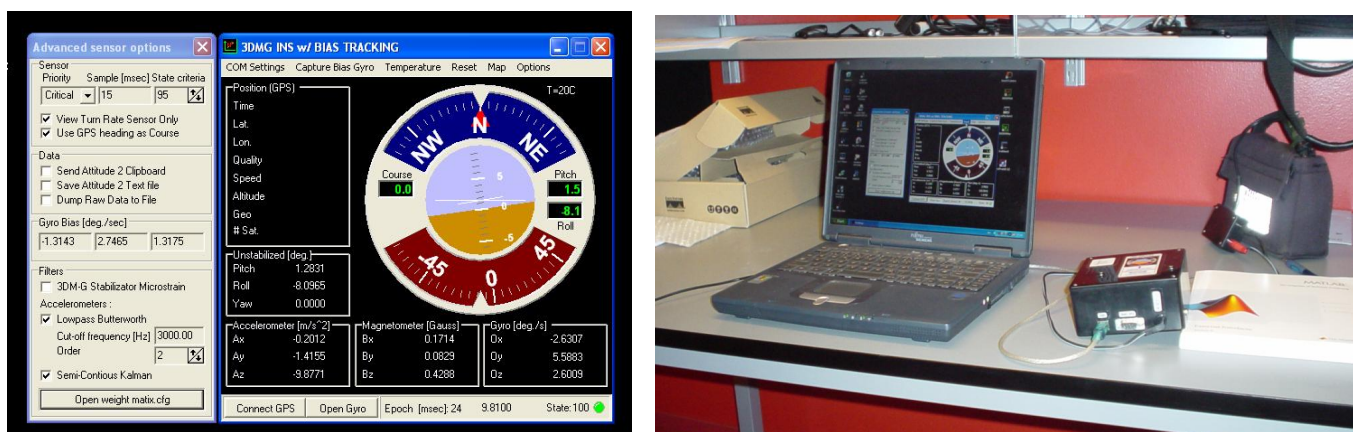


Figure 13. First experiment to test the precision of the solutions: the sensor has one edge on a book. After measurements of the angle via trigonometric formulas we assume that the angle should be -8.0 deg (measurement with geometric instruments). The combined solution provides us the displayed attitude (left part of the picture).

We had the following results:

Gyroscope model	Kalman filter	Combined solution	Microstrain
-8.3 deg.	-8.1 deg.	-8.1 deg.	-8.1 deg

Table 2: mean value of the measurements made by the different solution of a known angle: -8 deg.

We can consider that the plan does not have a perfectly known inclination, maybe -8.1 deg, because of the measurements. The results are quite relevant for the 3 solutions. We should stress that the measurement made by the Kalman filter is better than the integration model one (gyroscope model). It is coherent; it is due to the initialization default which affects the measurement; on the desk, the attitude is not totally flat. Thus the initialisation was not perfect. Furthermore it is important to underline that to make an estimation with this model, it is important to take into account the gyro bias. Otherwise the estimation is wrong. A constant bias was input, which is not precise. The enhanced solution provides the same measurement as the Kalman filter, which is coherent because in steady state situation, the combined estimator is switched on the Kalman filter.

For the 3 solutions, in steady conditions, the estimated attitude is not biased. It means that the mean value of the measurement is the true attitude.

The second experiment was to evaluate the magnitude of the noise and the stability of each solution. The sensor is put on the table without any shaking.

The results are shown on the table 3.

	Gyro model	Kal. – l.p.f	Kal. + l.p.f.	Comb. est.	Microstrain
Variance Roll (deg <sup>2</sup> )	0.0030	0.0013	0.0009	0.0010	0.0028
Variance Pitch (deg <sup>2</sup> )	0.3690	0.0049	0.0017	0.0009	0.0058
Variance Yaw (deg <sup>2</sup> )	0.0038	31829	32410	0.00006	0.00005
$\sigma_{Roll}^2 + \sigma_{Pitch}^2$ (deg <sup>2</sup> )	0.372	0.0062	0.0026	0.0019	0.0086

Table 3: comparison of the measurements variance in steady conditions, for the 3 estimators and the attitude provided by microstrain. Blue: best value ; red: worst value. (l.p.f. means low pass filter).

Regarding on the table 3, **in steady conditions**, the best model is the “enhanced one”. The worst one is the integration model (gyro model). This is due to the drift of the gyroscope. The stabilization is not perfect. The Fig.14 illustrates this as well.

Fig.14 also shows the influence of the low pass filter (l.p.f.) on the attitude. The noise magnitude is less important with this filter; the variance is almost divided by 3.

Moreover, the combined solution behaves like the Kalman filter with low pass filter in steady situation. This is coherent with the code architecture.

Nevertheless, we do not consider the yaw variance. The yaw is not continuous. Since there is a jump of 360 deg. between the value +0 and -0 deg., the obtained yaw variance is not relevant at all. We use a GPS to determine the Yaw. The experiment is made indoor, thus no measurements. The yaw value oscillated between +0 and 360. Thus the accuracy for it, is the same for all solutions, that of the GPS. That is why it does not make sense to display it in the Fig.14, in which the estimated attitude of each solution is displayed.

The estimation provided by Microstrain is quite smooth, but not as precise as the one provided by the combined estimator. This last one seems to be the best one.

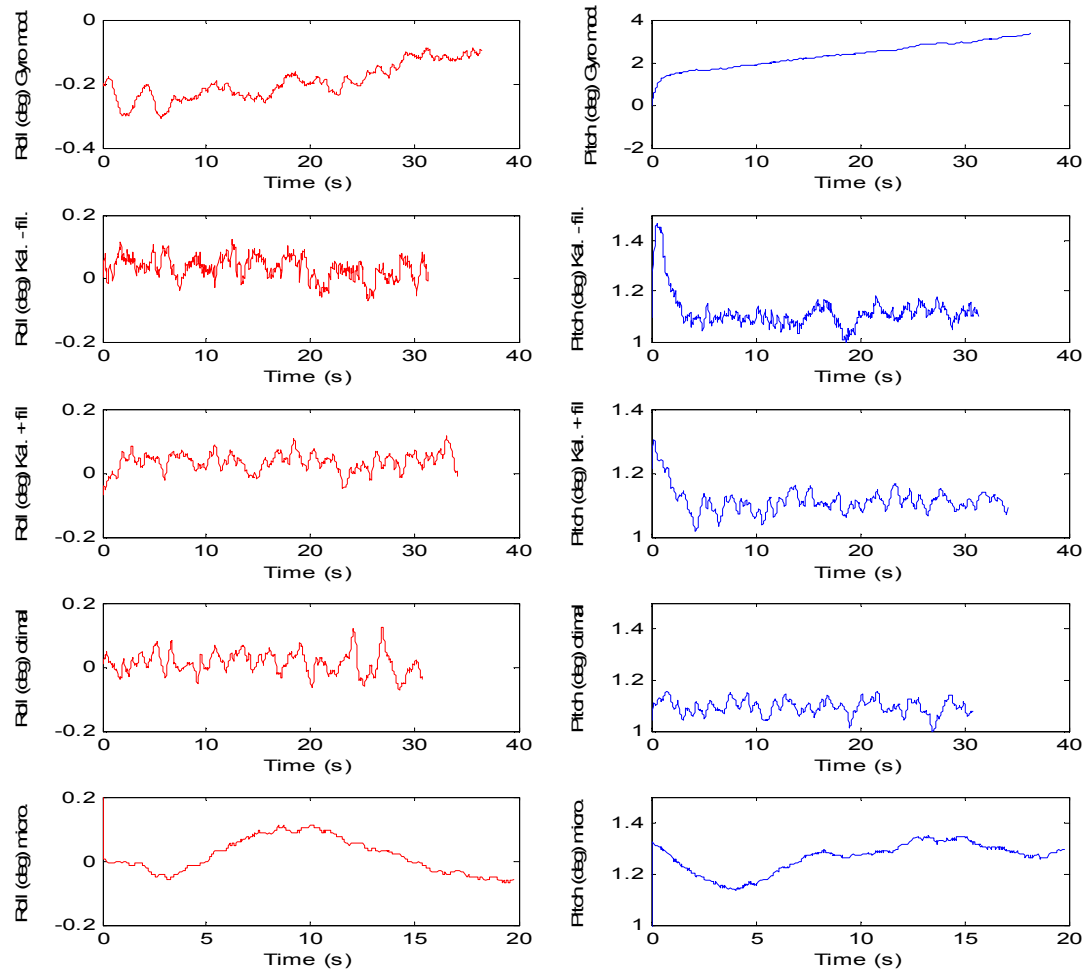


Figure 14. Comparison of the noise for the different models; from the top to the bottom: gyro model, Kalman without the low pass filter, Kalman with the low pass filter, combined estimator, microstrain (Red: Roll, blue: Pitch). The tracked gyro bias is not input into the gyro model on this figure; it explained the drift.

### *The inertial force:*

When the plane turns we have a centrifugal force, and when it is taking off/landing, we have acceleration. For more details on the accelerations, refer to the appendix J. We would like to get rid of these inertial components for the following reasons.

- If we want to take a picture when the plane is turning, we have to get rid of the centrifugal force to be precisely determine the attitude.
- If we have an inertial sensitive estimator on a drone to control its attitude (like we plan to do), during the launch the estimated Pitch would be higher than it should be. Thus the plane will never be launched because the wings motors will receive the instructions to decrease the lift. Thus, the commands sent to the motor are wrong.

To evaluate the robustness of the solutions, the system is shackled on a horizontal table with an almost constant true attitude: (Roll, Pitch)  $\approx$  (0, 1).



The low pass filter is used with the Kalman filter, to have less noisy results. Fig.15 illustrates the results of this last experiment.

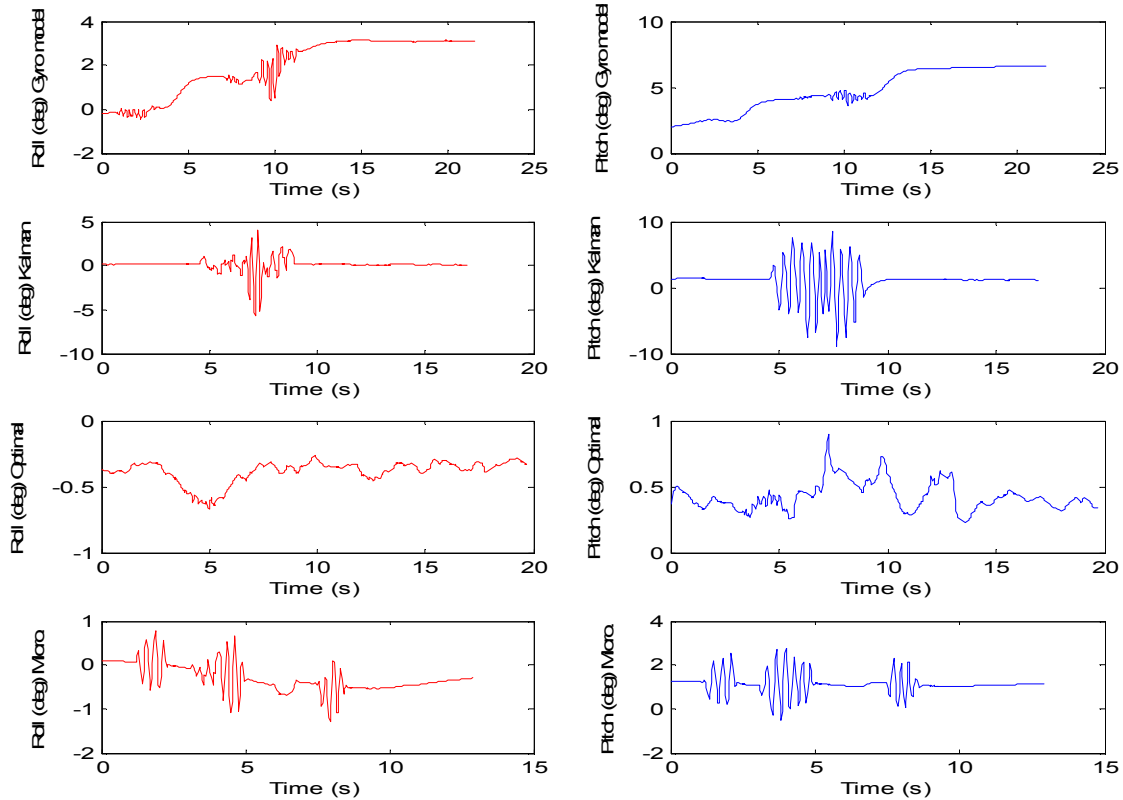


Figure 15. Comparison of each model, regarding on the inertial factor: We shake horizontally the system on a flat support. The roll and the pitch should be constant quite close to 0. (Red: Roll ; Blue: Pitch)  
From the top to the bottom: Gyro model, Kalman filter, combined solution, microstrain.

Several different interesting aspects appear on this Fig.15 :

- The drift of the gyroscope model is still present. The vibrations don't affect too much the displayed attitude during the shaking. The Kalman filter reacts normally, we explained these results before in section 3.4.
- The combined estimator has a really good behaviour regarding on these dynamic conditions. The switch between the 2 basic models seems to be quite efficient. The estimation is not biased and not really affected by the shaking. The past does not affect the current estimation.

If we compare this combined estimator with the one from Microstrain, this last one is almost two times as sensitive to the shaking (the window of value for Microstrain has a 2 magnitude, whereas 1 for our third estimator).

The combined solution appears to be the optimal one in shaking conditions.

### *A real time application: response time to a step*

One of the main properties the system should feat is to react to an excitation with the shortest delay (few ms). When the system is moved slowly, the estimation is smooth and follows the movement quite well. To ensure the estimation does not have any delay we can test the limits of the present estimators.

To test it, we could set up the following experiment. We put the gyroscope in a known original position (Roll, Pitch, Yaw)  $\approx (-8, 1, 0)$  and we move it quickly (like a jump) to the known final position  $(0, 1, 0)$ .

(NB: the Pitch is not null: it is inclination of the table).

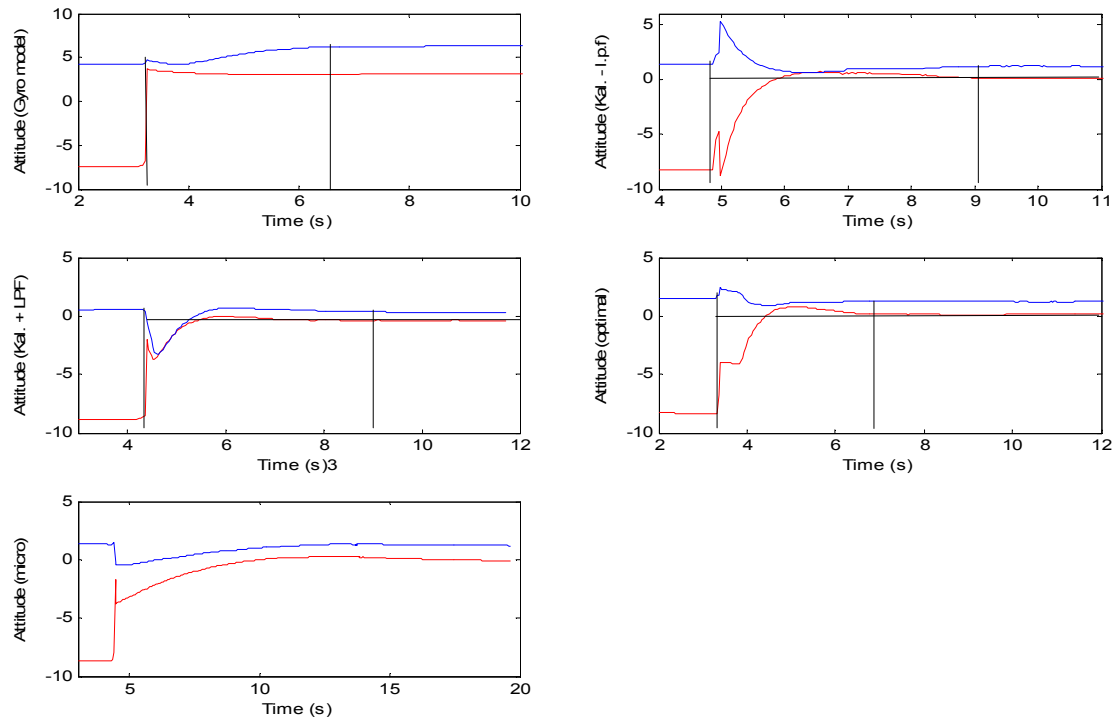


Figure 16. Response time to a step (-8 to 0 deg). Top left: gyro model; top right: Kalman without low pass filter  
Middle left: Kalman with low pass filter; right: combined solution. Bottom: Microstrain.  
(Blue: Pitch; red: Roll). The scale is in degrees

On Fig.16 the different responses appear. It is not easy to define a response time for the Microstrain's solution, because even after 15s of experiment, the system did not reach a stable equilibrium. Nevertheless after  $t = 13s$ , the attitude is quite stable. These graphs can be summarized into the Table 4.

	Gyro model	Kal. - l.p.f.	Kal. + l.p.f.	Comb. est.	Microstrain
Response time (s)	3.4	4.3	4.8	3.7	>5

Table 4: Response times for the different models. (Fastest: blue ; Slowest: Red).

The Gyro model is the fastest one. The combined solution is quite fast as well; this is because the Gyro model's properties are used when the conditions are not steady. We can also see the influence of the low pass filter on the Kalman estimator. The low pass filter slows down slightly the Kalman filter. However we have a better estimation.

The graph also displays an interesting property of the combined solution. When we move the system, we use the Gyro model properties (c.f. code architecture), until we detect another steady state. It happened at  $t = 5s$ . The combined solution detects a new steady state and use the Kalman filter model to estimate the attitude. It found an error of estimation and corrected the estimated attitude. Thus, we underline the auto correction property avoiding any estimation bias.

Some oscillations are observed on the Kalman solution and on the combined one exactly after the start of the step. It might be due to the accelerometer default. The accelerometer is made with springs, thus it might correspond to the influence of the springs oscillations via the measurements.

This experiment permits us to have an idea of the time response of the system. In reality, the attitude should be continuous, without any step like this, but it permits to detect if some non-relevant values are present, testifying of any defaults. For example, if a response time were higher than 10s, this estimator could not be really adequate for a real time application.

Moreover, the fact that the implemented estimators detect instantaneously the change of position is a relevant property.

We compared the estimators on different situations: dynamic and static ones. The switch between the both solutions (Kalman + l.p.f. and gyro model) seems to be a quite good combination.

Table 5 shows a comparison between the 3 described solutions:

	<b>Integration model</b>	<b>Kalman filter</b>	<b>Comb. Est.</b>
<b>Inertial reaction</b>	no	yes	limited
<b>Drift</b>	yes	no	no
<b>Robustness</b>	+	+++	+++
<b>Real time</b>	++	+	++
<b>Continuous</b>	yes	yes	yes
<b>Complexity</b>	+	++	+++
<b>Initialisation</b>	Flat attitude	No request	No request

Table 5: Advantages and disadvantages of the 3 solutions.

As we expected, the combined estimator has the advantages of both single ones, circumventing the disadvantages. The combined solution is the optimal one regarding on the whole tests.

Regarding the Microstain's estimator, the combined estimator seems to be better as well.

The previous tests were made in the laboratory. The estimator is created to be used in a plane. This means in a different environment. Some more realistic experiments were made, to be closer to the normal behaviour during a realistic application. These experiments were performed placing the sensors in a moving trolley or in a driving car.

The fact that the combined estimator is the best one compared to the gyro model and to the Kalman filter, only this one will be thoroughly tested, to assess its limits.

## 5.2. THE REAL BEHAVIOUR – DYNAMIC TESTS:

The realistic tests were separated in 2 different sessions: with the trolley and afterwards by a car. It is evident that these “realistic” conditions are quite different compared to the flying conditions. The trolley is not vibrating like a plane, but for a car, the accelerations and the vibrations might be stronger. Thus it can not substitute a test in a plane, but, in my first opinion, it should give an idea of realistic conditions.

### *The trolley tests*

The tests were made at UNIS, inside the building. Thus the yaw was not tracked because GPS does not work inside. It means only its variations were tracked via the turn rate sensor into the transition matrix for the Kalman filter and into the Hughes equation for the integration model. Thus, if the trolley turns, the yaw rate reacts as well. However the displayed attitude comes back to the position 0 because of the measurements made via the GPS: indoor the measured yaw is null. This pushes the system to display 0 after some epochs.



Figure 17. Inertial platform inside a trolley to test the attitude estimation in more realistic conditions.

The utilisation of a trolley provides a better simulation of the inertial forces which can be experimented by the system during a flight. Most of us have been at least once in a trolley. Even if it is a long time ago we keep in memory that the accelerations can be quite strong. It can be also really soft. Thus, these different conditions were tested.

It is also necessary to make numerous of preliminary tests to find the best deviation threshold (appendix I) used to detect if the conditions are steady or not. For a trolley, regarding the appendix H, the optimum threshold is 97 for the criteria deviation and we choose 1.5 for the gravitational criteria. This one is fixed, because the gravity is measured with a relative accuracy of 1-2% in steady conditions. Thus it is useless to choose less.

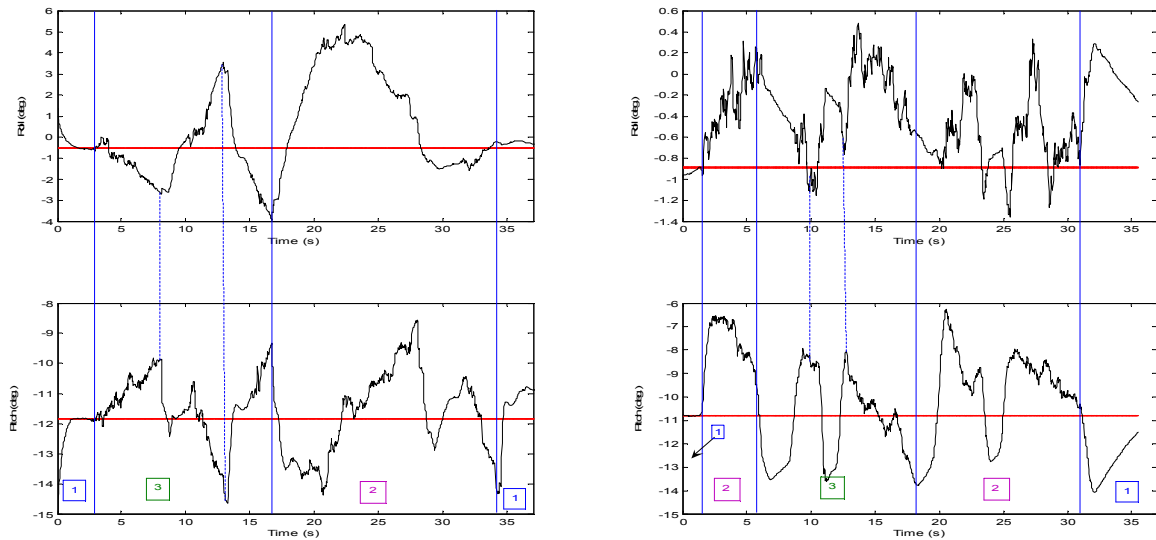


Figure 18. Estimated attitude pushing a trolley. Left : combined estimator; right: microstrain.  
Top: Roll; Bottom: Pitch both in degree. Red: true attitude

In Fig.18, the red attitude corresponds to the true attitude. Nevertheless it is only an indication, it can not be considered as really precise, because during the experiment the sensor can move a little bit on the trolley ( $\pm 1$  deg). Furthermore, the corridor does not have a perfectly constant attitude. The Roll and the Pitch can have some small variations. Nevertheless it is a relevant approximation.

On these graph, different conditions were tested. It is possible to separate them in 3 distinctive parts:

1. Part 1: initialisation and end of the experiment. The attitude is smooth and well estimated.
2. Part 2: steady conditions. The trolley is moving with a constant speed magnitude.
3. Part 3: dynamic conditions: strong shaking of the trolley. Both estimators are sensitive to the inertial forces

First of all, regarding this test, both solutions seems to react as the same way.

The Pitch of the combined solution seems to be better. Its extremes values are -14.5 and -9 degrees, thus the range of the displayed Pitch is almost  $\pm 2.5$  degrees.

With the Microstrain software, the extreme values are -14 and -7 degrees, thus the range is almost  $\pm 3.5$  degrees.

The Pitch of the combined solution looks better and more accurate.

The Roll estimated by the combined estimator seems to be smoother than the one estimated by the Microstrain. Nevertheless, this last one looks quite precise.

The Roll's variance of Microstrain is better ( $0.15 \text{ deg}^2$  and  $1.43 \text{ deg}^2$ ).

Regarding the Pitch's variances, the one of the combined estimator is better ( $4.32 \text{ deg}^2$  and  $4.56 \text{ deg}^2$ ).

Considering on the Trolley test, both models have similar properties. Nevertheless, in real conditions the results are less accurate as expected regarding on the bench tests. It was quite interesting to compare the combined estimator to the Microstrain's one on a trolley, but it could be even more interesting in a car.

## The car test

- the noise

First of all, we should describe the environment. The car is driven on a snowy road and the attitude is estimated. The sensors are fixed on the car.



The car is vibrating during the measurements. It could be quite relevant to have an idea about its noise. Thus the first thing was to estimate it.

The results are shown on Fig. 19.

The experiment takes place outside. The GPS is used.

Picture: me on the car during the test. The gyroscope is on the floor, the GPS below the windshield and the laptop on my knees.

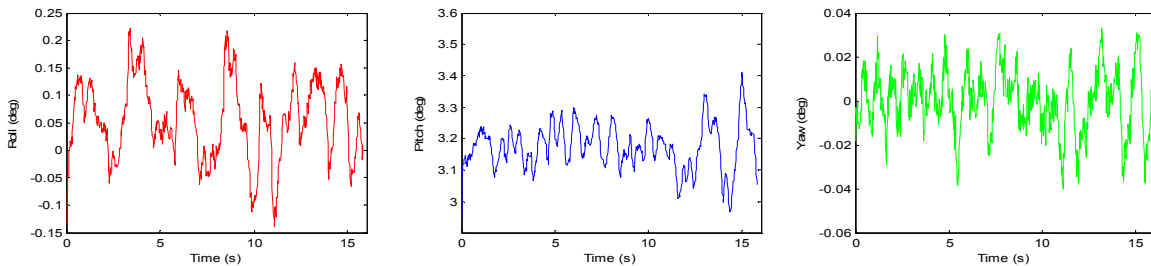


Figure 19. Influence of the car's noise on the attitude because of its own vibrations. From the left to the right: Roll, Pitch, Yaw.

The above measurements were made without any movements, thus it corresponds to the influence of the car vibrations to the attitude. To compare it with the noise measured on the laboratory desk, it is possible to make the table 6:

	Roll covariance	Pitch covariance	Yaw covariance
<b>On a bench</b>	0.0010 deg <sup>2</sup>	0.0009 deg <sup>2</sup>	----
<b>In a car</b>	0.0051 deg <sup>2</sup>	0.0052 deg <sup>2</sup>	0.0002 deg <sup>2</sup>

Table 6: Covariance of the attitude parameter influenced by the car noise; the chosen values for the first row were established previously on the table 3, column "combined solution".

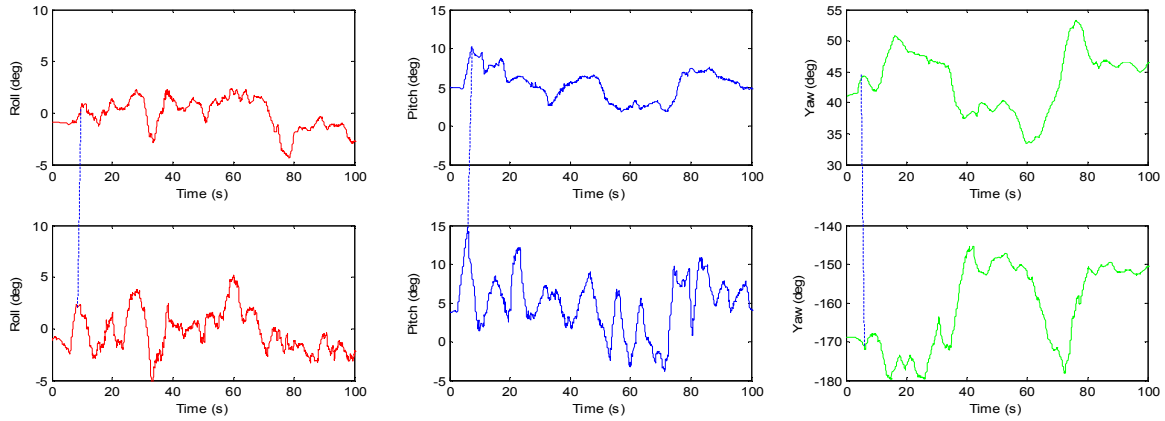
It is obvious that the car increases the noise significantly. The fact we used a merge for the noise measurement matrix and initial covariance matrix is quite relevant. It means that it is not necessary to change them.

- the test

It is also necessary to find the relevant threshold for the 2 steady parameters. We obtain the criteria 93 for the derivation criteria and still 1.5 for the gravity threshold.

We estimate the attitude via the combined estimator and via the Microstain's one driving on the same road, at nearly the same speed

The estimated attitudes are illustrated on Fig. 20.



**Figure 20. Estimated attitude driving a car. Top row: Microstrain; bottom row: combined estimator.**  
**From the left to the right: Roll, Pitch, Yaw. The microstrain's yaw is obtained via the magnetometer and for the combined estimator via the GPS.**

Both attitudes are close to each other. Nevertheless, the Microstrain's estimation seems to be slightly better. The driving conditions were nearly the same ones, exception for the acceleration magnitude which can not be controlled so easily.

Both systems are sensitive to the first acceleration. This moment is pointed out by the dashed line in Fig.20. The combined estimator felt the acceleration because the Kalman filter is used at the beginning, when the conditions are perfectly steady; the switch of the combined estimator is not accurate enough.

For the first time it was possible to have any access to the yaw data. The GPS seems to work quite well and provides us a relevant attitude. The orientation does not seem to be the same one between the 2 measurements: this because the GPS was on the dash board. We put it here in another direction that the driving one, for questions of stability on the dash board. The Microstrain's estimator uses the magnetometers.

The scale of the Yaw axis is interesting; it is almost the same one for both estimator and the estimated yaw is smooth.

### 5.3. SUMMARY OF THE TESTS

Regarding this comparative analysis, we can underline that the combined estimator is the optimal solution, compared to the single ones we described (the Kalman filter and the integration model).

It is as well important to mention that the estimations provided by this estimator is not biased and continuous (no visible steps).

Furthermore the restrictions underlined in the problem specification are respected; this estimation is perfectly in real time, there is no delay. We did not buy any more Hardware. There is no "Gimbal lock". The quaternion theory seems to be suitable to estimate the attitude.

If we compare the combined solution and that of Microstrain, the “laboratory” test seems to be better for the former solution.

Nevertheless this combined estimator and the Microstrain’s one seem to be equivalent to describe the trolley attitude.

However, the Microstrain’s one is still a little bit better for the car test. Thus it is necessary to think about some improvements, especially for the accuracy in real conditions.

Table 7 summarizes these comparative tests.

	Gyro mod.	Kal. + l.p.f.	Comb. Sol.	Micro.
Bias (estimated attitude)	yes	no	no	no
Inertial reaction	no	yes	limited	yes
Response time (s)	3.4	4.8	3.7	>5
Real time	yes	yes	yes	yes
Initialisation	flat attitude	-	-	-
Auto correction	-	+	+	+
Complexity	+	++	+++	???
Trolley test			++	++
Car test			+	++

Table 7: Comparative analysis of the all test.

**PS:** The tests performed in the car can not be quantitative for different raisons:

- a) I do not have a perfect attitude estimator which is providing me the true attitude in real time. Thus I do not have any reference to compare. That is why I used the Microstrain estimation as guidance for the relevance of my results.
- b) The attitude of the road where we performed the car test is not constant, thus it does not make sense to compute any average neither variance. If I could know the true attitude, an analysis of the error of estimation would have been relevant. It would have provided the accuracy. But I do not have any access to these data.



## 6. The improvements

We underlined previously that the combined estimator is the best one compared to the Kalman filter and the integration model separately.

Laboratory tests inside the school showed that the combined solution is better than the Microstrain's one. But on more realistic condition, it is the opposite. Thus, it is necessary to find a way to improve our estimator. It was actually an important part of the project.

Different methods were considered to improve our system. The first one involved use of the GPS. The second one consisted of improving the detection of the steady and non steady conditions, by adding another variable into the Kalman filter state vector, the gravity. We tried both these solutions.

### 6.1. THE GPS:

To improve the estimated attitude, we need to get rid of the inertial forces. These forces affect the displayed attitude, especially because of the Kalman filter. The inertial forces bias the estimator via the measurements. Thus the main idea is to remove these noisy forces using the GPS.

With the GPS, it is possible to have an access to the velocity of the plane or of the car. Thus, with a simple derivation, it is possible to have the linear acceleration. This linear acceleration, which direction is in the direction motion, is noted  $a_{px}$ .

It is also possible to have an access to the centrifugal forces due to the plane rotation (see appendix J).

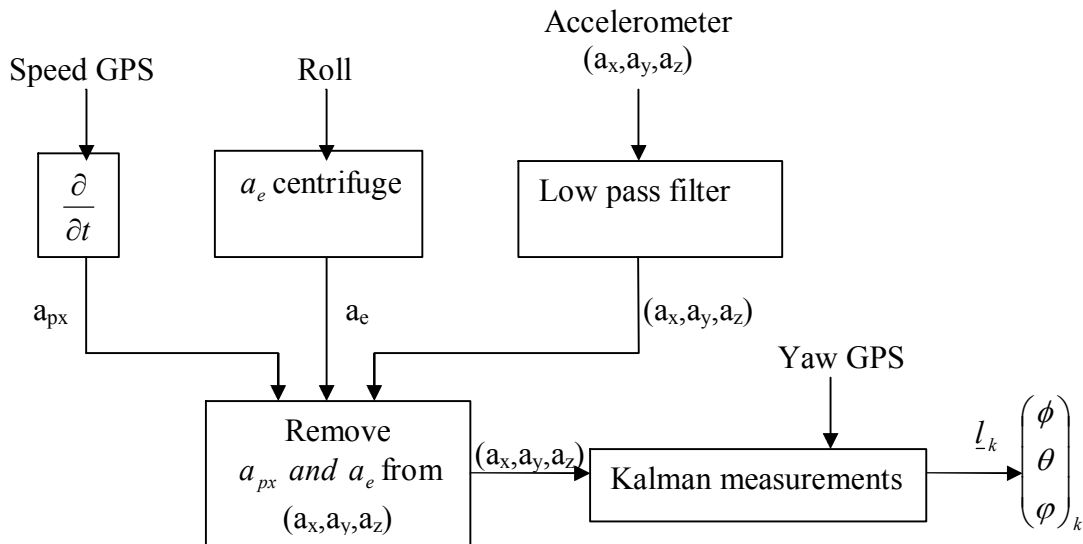


Figure 21. Process diagram of the GPS utilisation to improve the measurements of the Kalman filter.

We tried the solution outlined in Fig.21 in a car, taking off the part processing the Roll, because a car does not roll like a plane does. But the GPS provides us information only every 1s. That is fast enough to provide the yaw, but not for the acceleration. In 1 second the plane can have a strong acceleration and the estimated attitude is biased.

Thus another solution was tried with the gravity tracking. We took into consideration this factor because it is fundamental for the detection of steady movement (c.f. appendix I). The gravity we get, even if we use a low pass filter is really noisy. The consequence of this noisy parameter is that the switch which occurs for the combined estimator is not precise. Thus, to improve this parameter and to take off a large part of the noise, we use an extended version of the Kalman filter.

## 6.2. THE “GRAVITY KALMAN FILTER”:

We will start to describe the gravity Kalman filter, which is followed by an evaluation of its performance.

### *The filter*

We have to find the Markov matrix of the system. It is quite simple since we can consider the earth gravity as a constant for our application. Even if it changes with the latitude and the longitude. In Longyearbyen the gravity is  $9.83 \text{ m/s}^2$ , in Paris  $9.82 \text{ m/s}^2$  and at the equator  $9.81 \text{ m/s}^2$ . If we consider the noise of our sensor and the small area covered during the flight,  $g$  is approximated to be a constant.

Thus, the transition matrix between 2 epochs is the identity matrix for this variable.

For this filter, the state vector becomes

$$\underline{X}_{k+1} = [x_0 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7]_{k+1}^T = \underline{T}_k \cdot \underline{X}_k \quad \text{with} \quad x_7 = \text{gravity}.$$

The transition matrix is

$$\underline{T}_k = \frac{1}{2} \cdot \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ +x_4 \cdot \Delta t & +x_5 \cdot \Delta t & +x_6 \cdot \Delta t & 2 & -(\omega_x - x_0) \cdot \Delta t & -(\omega_y - x_1) \cdot \Delta t & -(\omega_z - x_2) \cdot \Delta t & 0 \\ -x_3 \cdot \Delta t & +x_6 \cdot \Delta t & -x_5 \cdot \Delta t & (\omega_x - x_0) \cdot \Delta t & 2 & (\omega_z - x_2) \cdot \Delta t & -(\omega_y - x_1) \cdot \Delta t & 0 \\ -x_6 \cdot \Delta t & -x_3 \cdot \Delta t & +x_4 \cdot \Delta t & (\omega_y - x_1) \cdot \Delta t & -(\omega_z - x_2) \cdot \Delta t & 2 & (\omega_x - x_0) \cdot \Delta t & 0 \\ +x_5 \cdot \Delta t & -x_4 \cdot \Delta t & -x_3 \cdot \Delta t & (\omega_z - x_2) \cdot \Delta t & -(\omega_y - x_1) \cdot \Delta t & -(\omega_x - x_0) \cdot \Delta t & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}_k$$

Regarding the measurement model, we measured the gravity via the accelerometer.

The measurements matrix becomes

$$\underline{l}_k = [\phi \ \theta \ \varphi \ g \ m]_k^T$$

With the Roll, the Pitch and the Yaw still obtained via the accelerometer and  $g\_m$  defined as

$$\begin{cases} g\_m_k = \sqrt{a_{x_k}^2 + a_{y_k}^2 + a_{z_k}^2} & \text{if} \quad \left| \frac{g\_m_k - gravity_{k-1}}{gravity_{k-1}} \right| \cdot 100 \leq 0.2 \\ g\_m_k = g\_m_{k-1} & \text{else} \end{cases}$$

This permits to remove the inertial forces.

The measurement model is

$$l_k = \begin{bmatrix} \underline{0}_{3 \times 3} & \underline{A}_k & \underline{0}_{3 \times 1} \\ \underline{0}_{1 \times 3} & \underline{0}_{1 \times 4} & 1 \end{bmatrix}_k \cdot \begin{bmatrix} \underline{W}_{bias} \\ \underline{q} \\ \underline{gravity} \end{bmatrix}_k + \begin{bmatrix} \underline{b} \\ \underline{b\_gravity} \end{bmatrix}_k.$$

The noise  $\underline{b}_k$  is the same one as in the previous Kalman filter.  $\underline{g\_m}$  is obtained via the same accelerometer. After an analysis of the noise  $\underline{b\_gravity}$ , we underlined that its characteristics are the same ones as the noise  $\underline{b}_k$ . Thus we obtain (also after some experiments) the following noise measurement covariance matrix:

$$\underline{R}_k = \begin{bmatrix} \sigma_\phi^2 & 0 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 & 0 \\ 0 & 0 & \sigma_\varphi^2 & 0 \\ 0 & 0 & 0 & \sigma_{g\_m}^2 \end{bmatrix}_k \approx \begin{bmatrix} 1.0510 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 1.2556 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 3.74 \cdot 10^{-4} & 0 \\ 0 & 0 & 0 & 3 \cdot 10^{-4} \end{bmatrix}_k$$

To have a good initialisation, we consider  $x_{7_0} = 9.83 \text{ m/s}^2$  which is quite accurate, but the theoretical value is not so interesting for us. Our interest is in the value that the sensor provides us at the initialisation time. We consider

$\sigma_{gravity}^2 = 5 \cdot 10^{-5} \text{ m}^2 / \text{s}^4$  which corresponds to the accuracy of the system at the initialisation.

Then we have the following initialisation vector:

$$\underline{\hat{X}}_0^- = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 9.83]^T$$

We defined the initial covariance matrix such as

$$\underline{Q}_0^- = \begin{bmatrix} 10^{-4} & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 10^{-4} & \ddots & & & & & \vdots \\ \vdots & \ddots & 10^{-4} & \ddots & & & & \vdots \\ \vdots & & \ddots & 5 \cdot 10^{-5} & \ddots & & & \vdots \\ \vdots & & & \ddots & 5 \cdot 10^{-5} & \ddots & & \vdots \\ \vdots & & & & \ddots & 5 \cdot 10^{-5} & \ddots & \vdots \\ \vdots & & & & & \ddots & 5 \cdot 10^{-5} & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & 5 \cdot 10^{-5} \end{bmatrix}$$

All the equations are established. It is now possible to run this filter.

Nevertheless, we did not run this filter more than once or twice, because to compute the Kalman gain we need to inverse a matrix 4\*4 whereas 3\*3 as previously. It takes a long time, also to make computations with 8\*8 matrices. For a real time application, this is not relevant (the processing time is multiplied by 1.5). Thus, we made a simple filter tracking only the gravity, running in parallel of the attitude estimator.

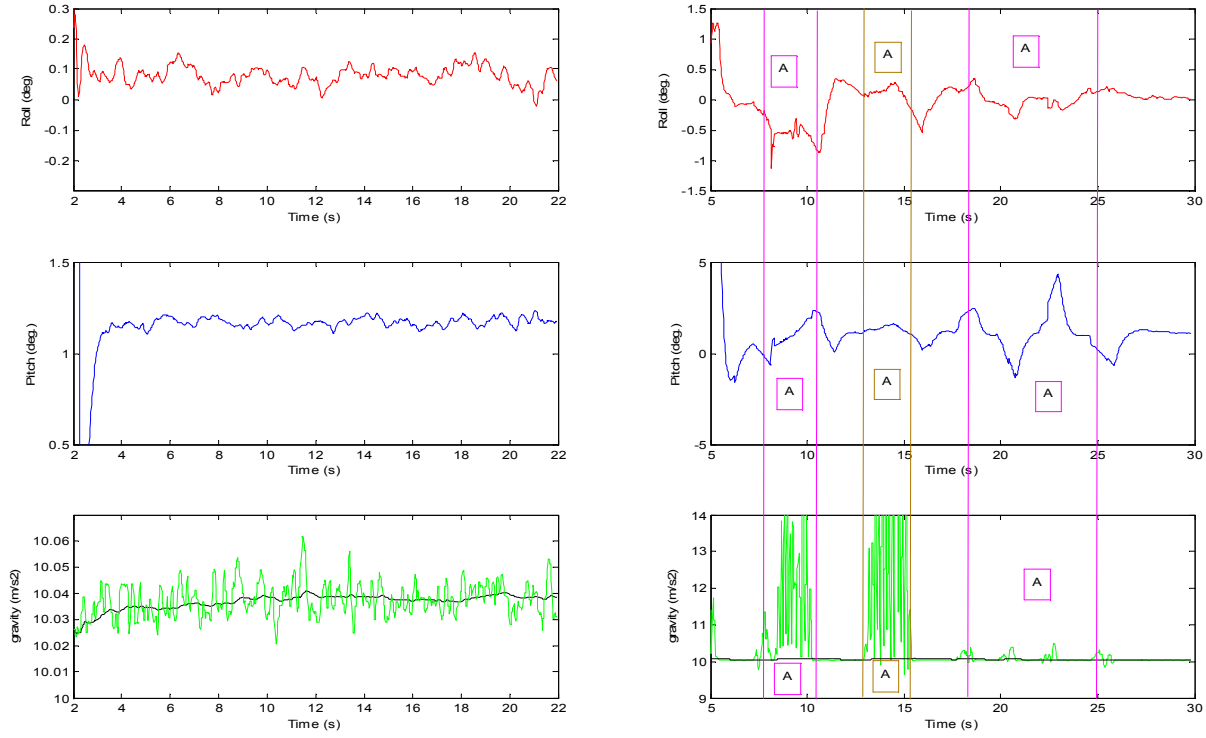
### ***The results:***

#### ***- On the gravity factor:***

This filter was tested in steady conditions and in dynamic ones.

At first, we put the sensor on the desk, and we measured the gravity with and without the filter. The results are shown on Fig.22 on the left column.

Secondly, we still put the sensor on the desk, but we shake it strongly. This permits us to see how the filter reacts in dynamic conditions. The result is shown in Fig.22, right column.



**Figure 22. Attitude and gravity estimation.**

**Right column: steady conditions; bottom: filtered gravity (black), gravity without Kalman filter (green)**

**Left column: dynamic condition; bottom: filtered gravity (black), forces applied to the system: gravity + inertial forces (green). A zones: shaking zones.**

The results are clear. We can see that under steady conditions, the filtered attitude is smoother than without the Kalman filter. A large part of the noise is gone and the estimated gravity is quite accurate ( $\pm 0.005 \text{ m/s}^2$ ).

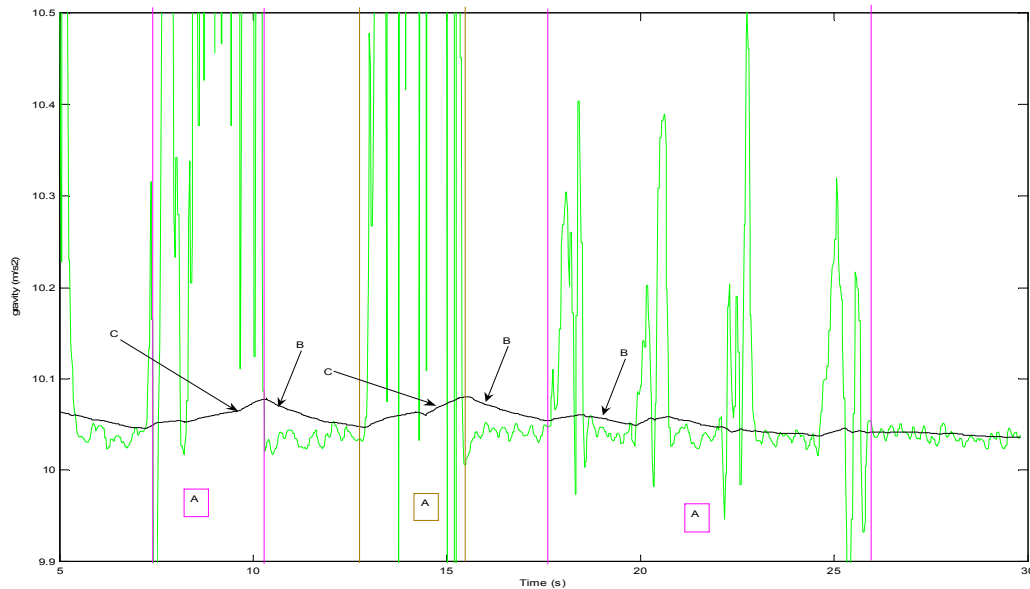
In dynamic conditions, the Kalman filter behaves well. For reasons of scale, we should enlarge the last graph, regarding the estimated gravity. This is shown in Fig. 23, on the next page.

We can easily recognize the shaking zones and the fact that the Kalman filter takes into account the measurements. The B and C arrows indicate the zones in where the filter reacts to the shaking and also to the measurements.

The Kalman filter takes into account the measurements but not strongly because of the Kalman gain  $K$ . If this gain is small, the prediction, before the measurements, is more taken into account than the difference between the prediction and the measurements.

By consequence, the initial input values are of significant importance ( $\underline{R}_k$  as well).

Thus the filter is affected but not strongly to the shaking.



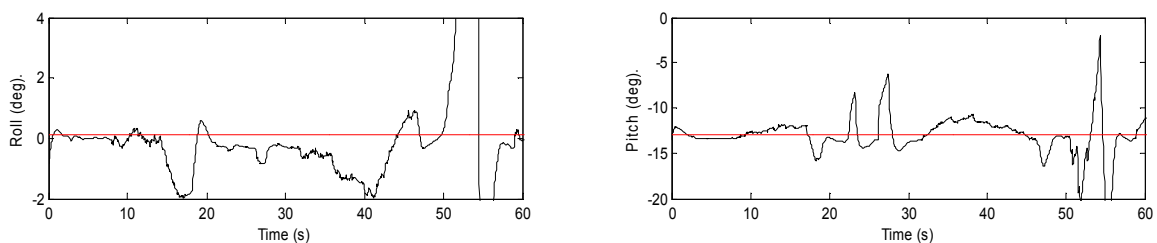
**Figure 23. Reaction of the Klamen filter regarding on its environment.**  
**Green: measurements, forces applies on the system (gravity and inertial forces)**  
**Black: estimated gravity.**

The “B arrows”, outside of the A parts, point out how the Kalman filter reacts at the end of the excitations. The main excitations are indicated by “C arrows”.

#### - *On the estimated attitude:*

To underline the efficiency of this gravity tracking, we push a trolley and we try to find the optimal deviation factor threshold.

If we use this last version of the Kalman filter, we obtain the best results for a threshold of 93. For this test, the gravity threshold is 0.5%.



**Figure 24. Estimated attitude via the improved combined estimator (left: Roll; right: Pitch) for a deviation factors threshold of 93 (red: mean value, thus true value).**

The estimated attitude seems to be quite smooth and the accelerations at the beginning and at the end of a movement appear but not as strongly as in the previous test, which was done without the gravity tracking.

The optimal estimation we had, was for the deviation factor treshold 98/97 as shown on the Fig.25. The gravity threshold was 1.5%.

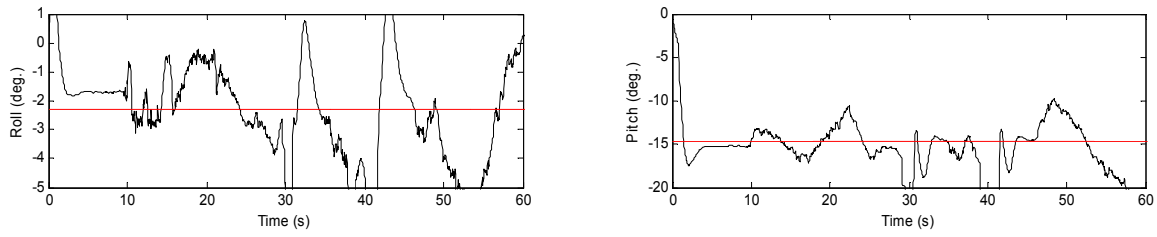


Figure 25. Estimated attitude pushing a trolley without the gravity tracking, for an optimal deviation factor threshold of 98. (left Roll, right: Pitch). Red: average, which is this time a bit different of the true value: a few tenths of degrees.

Tracking the gravity seems to improve the system.

In the following table we can compare the accuracy regarding each factor for both estimators.

	Grav. Track.	No Track.
<b>Dev. Threshold</b>	93	98
<b>Mean Roll (deg.)</b>	0.11	-2.30
<b>Mean Pitch (deg.)</b>	-12.98	-14.67
<b>Variance Roll (deg<sup>2</sup>.)</b>	5.14	10.33
<b>Variance Pitch (deg<sup>2</sup>.)</b>	4.33	7.91
<b><math>\sigma_{Roll}^2 + \sigma_{pitch}^2</math> (deg<sup>2</sup>.)</b>	9.47	18.24

Table 8: Statistical comparison of the attitude provided by the optimal solution when we track the gravity and the one obtained without the gravity tracking . Blue: best values; red: worst values.

The results are quite obvious. The gravity tracking improved the estimation significantly. Now for a factor equal to 93, we have the following accuracy:

$$\begin{cases} \phi = 0.11 \pm 2.26 \text{ deg.} \\ \theta = -12.98 \pm 2.08 \text{ deg.} \end{cases}$$

Without the gravity tracking, we had the Roll with a precision  $\pm 3.21 \text{ deg.}$  and the pitch accuracy was  $\pm 2.81 \text{ deg.}$  These results are obtained with really strong accelerations.

With nearly the same way of shaking, the Microstrain's estimator provides an attitude with an accuracy which is  $\sigma_{\phi} = 1.19 \text{ deg}$  and  $\sigma_{\theta} = 2.44 \text{ deg}$  ( $\sigma_{Roll}^2 + \sigma_{pitch}^2 = 7.38 \text{ deg}^2$ ).

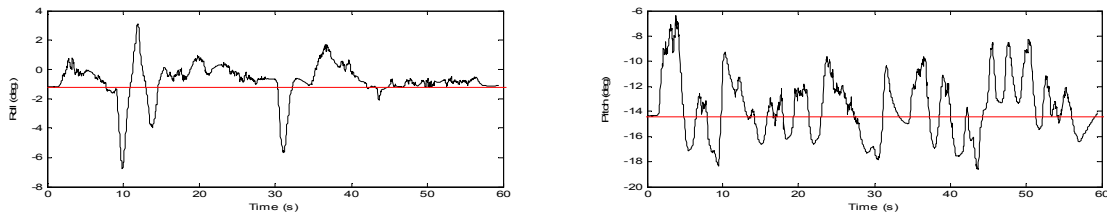


Figure 26. Estimated attitude via the Microstrain software with a strong shaking trolley test. Left: Roll, right: Pitch. Red: true attitude.

Thus in the trolley test with a strong shaking, the two solutions seem to be still nearly equally effective.

Nevertheless it is interesting to underline that the Microstrain's estimator seems to be better because of the Roll accuracy. The microstrain's yaw is established via the magnetometer, thus the urge Roll's variations resulting of a U-turn of the trolley does not appear (at  $t=50s$  on Fig. 24). It appears for the combined estimator because inside, the GPS is not used (it does not appear for outdoor tests as explained on the appendix H). If we neglect it, the Roll's variance of the combined estimator is as well almost  $\sigma_\phi = 0.80 \text{ deg}$ . And by consequence the combined estimator is the best one.

### 6.3. SUMMARY OF THE GRAVITY TRACKING:

We decided to improve the estimator because the comparative analysis underlined the un-sufficient accuracy of the estimator, especially compared to the one of Microstrain. To describe the trolley attitude, both estimators behaved equally, and for the car test, the hybrid estimator was not satisfying enough.

Among the two solutions we implemented, only one was neglected quickly: the one dealing with the GPS. The data are transmitted with a too low rate.

The second implemented possibility seemed to be more relevant. It consists to track the sensed gravity via a Kalman filter to have a better detection of the flying conditions.

The results are clear. The gravity is well tracked and the attitude estimate is more accurate. The one provided by the Microstrain's software is not so precise.

With the trolley test, it is possible to have any access to numerical data, like the standard deviation.

Microstrain	Combined estimator
$\begin{cases} \sigma_\phi \approx 1.19 \text{ deg.} \\ \sigma_\theta \approx 2.44 \text{ deg.} \end{cases}$	$\begin{cases} \sigma_\phi \approx 0.80 \text{ deg.} \\ \sigma_\theta \approx 2.08 \text{ deg.} \end{cases}$

Table 9: Accuracy of the estimated attitude via the combined estimator and the one of Microstrain. This accuracy was obtained pushing a trolley.

We can underline that even if it is not described on the report a car test was also performed. It appeared that the combined estimate and the one of Microstrain react as a same way.

It was now time to use our estimator in a real airborne application.

## 7. The final application: the airborne test

The combined estimator looks ready to be used in a real airborne application. We flew with the coast guards around Svalbard at low altitude to detect the presence of fishing boat in the darkness. F. Sigernes created a camera with a light amplifier. My estimator is used to localize fishing boats.



Figure 27: on the left we can see the camera located behind the second window of the plane. On the right, the airborne platform is located at the back of the plane.

The coast guard company supports this experiment because it can improve their working conditions. The polar night is quite long: almost 3 months. Thus, they have to work often in the total darkness. Our optical platform can help them.

The sensor is located on the floor of the plane. Its sensed attitude is (0,0,Yaw) when the plane is on the garage. The computer is located at the back of the plane. This test contains a lot of unknown. It is the first time that we can have access to some real raw data. It is also possible to estimate if the tests performed previously were realistic enough to simulate the attitude. We will see that this test was quite surprising.

### 7.1. THE PLANE VIBRATIONS:

On Fig.28 is shown all the parameters which are necessary to describe the flying conditions. Three different zones appear on this figure in four domains.

- 1) zone 1: steady conditions. The plane is not flying but slowly running on the tarmac to go to the beginning of the launching way. There are two zones 1: the first one.

The attitude is well estimated by the combined estimator. The tarmac is flat like it should be and the yaw is constant. Nevertheless, the steady parameters are alarming:  $g_m$  corresponding to  $\sqrt{a_x^2 + a_y^2 + a_z^2}$  and deviation. For more information about these parameters, refer to the appendix "Steady parameters". In steady conditions, the plane is so vibrating that  $g_m$  takes some values from 9 to 11  $m/s^2$  (standard deviation of 0.77  $m/s^2$ ). It corresponds to 8% of the gravity. In our implemented code and with the previous "realistic" tests, this rate was only 0.5%. The deviation factor reacts as well to these vibrations: regarding the obtained values, the deviation threshold should be almost 70. With the trolley and with the car it was higher than 90. We can conclude that the plane's



vibrations mask the steady conditions. Thus, the steady flight conditions are not detected because the chosen thresholds are not well adjusted.

- 2) zone 2: planes rotation. Both parameters  $a_y$  and Yaw show clearly that the plane is turning on the tarmac. It is located at the beginning of the glide path, so the true attitude is almost (0,0,Yaw). The estimation is quite exact.
- 3) zone 1: the second steady zone. The estimate is wrong. Because of these noisy conditions, the gyro bias might be bad estimated. Thus the integration model is drifting because of this bad estimation. This zone 1 is located after a zone 2. Afterwards the steady conditions are not detected. Thus the estimation is not updated by the Kalman filter. It implies a wrong estimation of the attitude via the combined estimator.
- 4) zone 3: the plane is taking off. The  $g_m$  parameter is clear: an external acceleration is applied on the system. The accelerometers components are coherent with this. It is simply the acceleration produced by the plane engines to take off. The accelerometer component which is the most varying is  $a_x$ , along the flight direction.

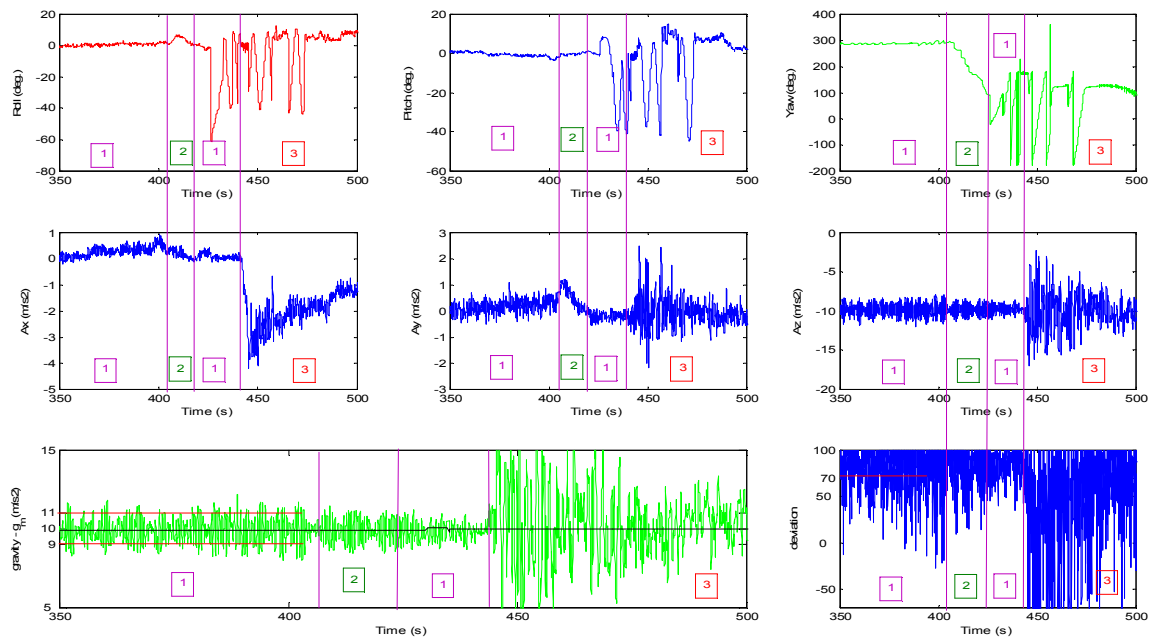


Figure 28. Necessary parameters describing the flight conditions.

Top row: Roll (red), Pitch (blue) and Yaw (yaw) in degrees.

Middle row:  $A_x$ ,  $A_y$  and  $A_z$  in  $m/s^2$  (respectively from the left to the right).

Bottom row: Steady parameters.  $g_m$  (green) and gravity (black): left. Deviation factor.

About the yaw it is important to notice that the displayed steps are normal. We have to be aware that an oriented angle is defined with a 360 deg. factor. By example both yaws, +90 and -270 deg. are equivalent.

During the flight, we were not aware that the estimate problem came from the steady parameter, or at least I did not expect that the thresholds would be so different compared to the car test.

By consequence I decided to continue to register the data and to estimate the attitude via the Kalman filter only. It shows the attitude which would be displayed in steady conditions with the combined estimator.

Besides I decided to continue to register, to get some real raw data. It is interesting for the perspective of the project (next and last chapter).

## 7.2. THE ATTITUDE DURING THE FLIGHT:

The used estimator is the Kalman filter with the low pass filter. On Fig.29 a short period of the flight is represented. It corresponds to some steady conditions (zones 1) but also curves (zones 2).

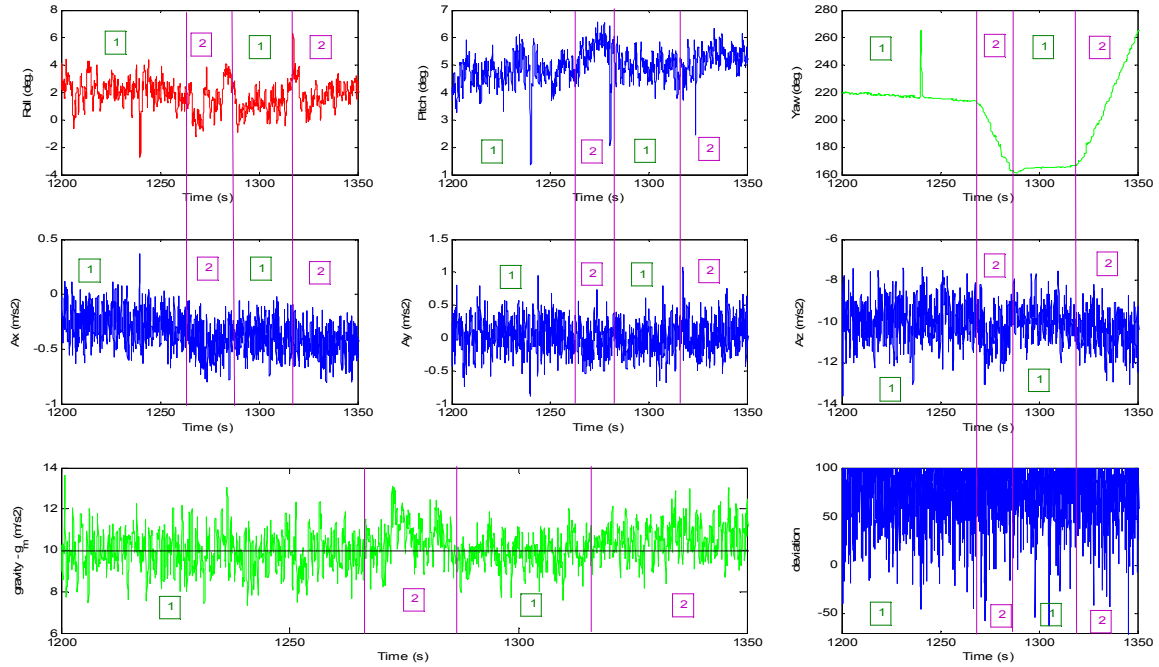


Figure 29. Steady flying conditions and rotation  $t=1275s$ . The estimation is provided by the Kalman filter.

Top row: Roll (red), Pitch (blue) and yaw (green) in degrees.

Middle row: Ax, Ay, Az from the left to the right.

Bottom row: steady parameters:  $g_m$  and gravity ( $m/s^2$ ; left) and deviation factor (right).

On this figure, it is really obvious that the Kalman filter provides a really noisy attitude because of the plane's vibrations. During the period corresponding to the first zone 1, the conditions being steady, the attitude should be constant. Thus it is possible to have an access to the accuracy. The obtained standard deviations are  $\sigma_{roll} = 0.94 \text{ deg}$  and  $\sigma_{pitch} = 0.58 \text{ deg}$ . The estimation is quite precise.

On the tarmac, the mean Roll was null. The mean of the Roll during the flight is not null. We have to consider a navigation factor to explain this: the wind. The day we performed the test was a bit windy. I asked the pilot how they correct the trajectory when the weather is windy. They do it with the Roll if it is blowing in any side of the plane. It can explain the Roll average of +2 deg.

On the zones 2, represented on Fig.29, regarding the yaw, the plane is turning. We will comment the first turn. The yaw is decreasing. It means the plane is turning to the right. Thus, the right wings is going down which corresponds to a decreasing of the Roll. The Roll and the yaw parameters are coherent. **But the values are wrong.**

Regarding the yaw, the plane is turning of 50 deg. in almost 20s. It corresponds to an angular speed of 0.04 rad/s. The pilot told me that a maximal regular turn rate is 0.05 rad/s. Referring to the appendix J, the curvature radius is 1832 m and the Roll should be -16 deg. without the wind, thus -14 deg. with. The plane could turn even if the Roll is null (uncoordinated), but it is seldom that the pilots do it. Thus we can consider that the estimated roll is wrong.

The centrifugal force appears on the gravity parameter.  $g_m$  is bigger, testifying an external force. At the beginning of the rotation, the deviation factor is smaller, during the time that the plane takes its turning position. As soon as it is made, the centrifugal force is constant. The accelerometer components stay nearly constant and the

deviation factor does not detect the rotation. Only the gravity parameter detects it. It illustrates why the combination of these both parameters is necessary.

### 7.3. SUMMARY OF THE FLIGHT:

The plane test was the main application of my project. We flew with the coast guards to control the positions of fishing boats.



The picture on the left is one of the pictures we provided. At the top of the picture is written the date and the time. At the bottom, the geographical information are displayed: latitude, longitude, velocity, altitude, roll, pitch and yaw.

On this picture we can see that the plane is rolling with an angle quite close to the estimation: almost 2 deg. which seems quite correct.

This plane test provided us much useful information:

- Regarding the results, the other tests we performed can not substitute a real plane test. However, now we have some realistic raw data (accelerometer components, turn rates and GPS). It is now possible to have realistic simulations.
- The combined estimator did not work because as soon as an estimation mistake occurred the estimation was biased. The steady conditions were not detected to update the estimator because of the plane's vibrations.
- The Kalman filter behaves well in steady conditions as expected. When the flying conditions are not steady, it is less precise as expected as well.
- The steady parameters work well. It is just necessary to adjust correctly the threshold to detect properly if the flying conditions are steady or not. The planes vibrations are really important. The table 10 illustrates it.

	Roll covariance	Pitch covariance	Yaw covariance
<b>On a desk</b>	0.0010 deg <sup>2</sup>	0.0009 deg <sup>2</sup>	----
<b>In a car</b>	0.0051 deg <sup>2</sup>	0.0052 deg <sup>2</sup>	0.0002 deg <sup>2</sup>
<b>In a plane</b>	0.0329 deg <sup>2</sup>	0.0256 deg <sup>2</sup>	0.0015 deg <sup>2</sup>

Table 10: Attitude variance due to the plane's engines, when this one does not move.

The vibrations of the plane are clearly much more intensive than the ones of the car. This combined with the air perturbations it explains partly why cars test can not substitute a plane test.

This final application underlined the fact that even if the combined estimator is ready to estimate a trolley attitude, or a car attitude, it is not for a plane attitude.

Nevertheless, some perspectives are possible to improve it.

## 8. Perspectives – Conclusion

My project consisted to create a real time airborne attitude estimator. After I have been aware of the problem statements, it was possible to create it, in respect of the expectations. During the project, my strategy changed significantly. At the beginning, I thought that the 2 first models I implemented, the Kalman filter and the integration model, would be efficient enough to estimate perfectly the attitude.

Firstly, I did not think at all that the inertial forces could make the problem so difficult. Secondly, the bias of the gyroscope had some variations out of my expectations.

The inertial forces biased completely the Kalman estimate. The gyro drift biased the integration model. Because of this, I had to think about a better solution. The combined estimator which I had to improve. The results are compiled in the following table.

	Gyro mod.	Kal. + l.p.f.	Comb. Sol.	Micro.
Bias (estimated attitude)	yes	no	no	no
Inertial reaction	no	yes	limited	yes
Response time (s)	3.4	4.8	3.7	>5
Real time	yes	yes	yes	yes
Yaw measurement	turn rate	GPS	GPS	magnetometer
Initialisation	flat attitude	-	-	-
Auto correction	-	+	+	+
Complexity	+	++	+++	???
Trolley test			++	++
Car test			++	++
Plane test		+	-	-

Table 11: Summary results table.

Each estimator respects a large part of the problem statement. Nevertheless, the optimal one, the hybrid model, has to be improved.

- The estimation is not biased. Regarding the tests and the results the gyro model is biased as soon as an error occurs. The Kalman filter is biased in dynamic conditions. On the trolley test and on the car test, the combined estimator is not biased either, because of a relevant switch between the 2 models. This switch is based on the flying conditions (steady or not).
- All the estimators processed the data in real time.
- We did not buy any other sensors. We just bought a coded low pass filter.
- The estimator is auto corrected, thus, robust. It means when an estimation error occurs, the estimator corrects it itself via measurements.
- The combined estimator is as accurate as the one of Microstrain at least for the trolley and car tests. It is even better for the laboratory tests. Nevertheless, the Kalman filter behaves quite well in steady conditions in a plane.
- The estimator can be adapted on systems vibrating (=noise) as much as a car or a trolley. The one I implemented is as accurate as the one of Microstrain for these conditions. But for a plane, it has to be improved.

Nevertheless, there were, in my opinion, 2 big difficulties to reach the goal of my project, create a plane attitude estimator processing the data in real time.

- 1) The first difficulty was the absence of precise reference. I did not have any access to reference values to compare the efficiency of my estimators. It made the quantitative analysis difficult, especially in realistic environment. I used the Microstrain estimation as reference, even if I knew that it was not perfect. As a consequence, it was hard for me to know if my estimation was biased or not. The accuracy is even harder to determine.
- 2) The second difficulty was the absence of real plane data. The final test proves that a flight can not be simulated by a driving car. Nevertheless, for the future, I register all the raw data of the flight. This means that it will be possible to make some simulations with realistic data.

As it is mentioned previously, it is necessary to improve the attitude estimation. To manage it, it might be interesting to perform many simulations with the flight raw data, optimizing as much as possible the combined estimator estimate. Regarding these tests, it might be possible to find some relevant thresholds for the state criteria, improving the estimation (like almost 70 for the deviation threshold and 8% for the gravity factor).

It might be quite relevant to enlarge the used techniques. I mean, to use different concepts, like mechanical ones. After the flight I talked to a pilot, and he told me, if I know the aircraft speed (obtained via the GPS), the wind speed, its direction and the Roll of the aircraft, I could really improve my estimator. Furthermore, some models including mechanical concepts exist, but they are intrinsic to the plane itself: the size of the wing, their lift and some other geometric parameters. If we measure the dynamic moment of the plane, it opens some other perspectives to estimate the attitude. But the estimator will not be so flexible and adaptable. It will be adjusted for only one plane.

Regarding the future applications of this attitude estimator, it is already planned to use it in a drone taking pictures of the ground.

It was actually a big challenge and very interesting to do this project. It permits me to create a product, and to see its final application. Furthermore it fits perfectly with my education in signal processor, specialized in remote sensing and navigation systems.

I also would like to express my gratitude to Fred Sigernes who offered me the opportunity to solve such nice challenge.

## APPENDIX A:

### Least square adjustment

The least square adjustment is a method used to improve the estimation of parameters. This is made via sequential improvements adjusted by measurements. It means the more we have measurements about parameters, the more accurate we know these parameters.

Consider a chemist searching the ion concentration of a solution as an illustration. The chemist has a glass full of salty water. He wants to estimate the salt concentration. Thus he starts with one measurement. The equipment is not very accurate, and the range is quite large due to the noise. Thus, he performs another measurement, to improve the first estimation.

The second measurement will provide additional information about the concentration. Thus, the combination of both measurements will be statistically more precise and the chemist's incertitude will decrease; his human incertitude can be an analogy with the mathematical covariance. The chemist can repeat this process until his certitude is big enough.

This is the essence of the least squares adjustment. First of all, the estimation of parameters via noisy measurements is explained in the following section. This is based on the least square methods, used to solve noisy equations. Secondly, the sequential adjustment is demonstrated algebraically.

#### *Standard adjustment*

Least square adjustment is based on equations where the observations are expressed as a function of unknown parameters. In matrix notation, the linear observation model is expressed as

$$\underline{l} = \underline{A} \underline{x}, \quad (\text{A.1})$$

where  $\underline{l}$  is the vector of observations (measurements),  $\underline{A}$  the design matrix and  $\underline{x}$  the vector of unknowns. In case of nonlinear functions, a Taylor series expansion is usually performed. The series is then truncated after the second term in order to obtain a linear function. Furthermore, we introduce the covariance matrix of the measurements

$$\underline{Q}_l \equiv E[\underline{l} \cdot \underline{l}^T] = \begin{bmatrix} \sigma_1^2 & \rho_{12} \cdot \sigma_1 \cdot \sigma_2 & \cdots & \cdots & \rho_{1n} \cdot \sigma_1 \cdot \sigma_n \\ \rho_{12} \cdot \sigma_2 \cdot \sigma_1 & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ \rho_{1n} \cdot \sigma_n \cdot \sigma_1 & \cdots & \cdots & \cdots & \sigma_n^2 \end{bmatrix}. \quad (\text{A.2})$$

Here  $\sigma_i^2$  is the squared variance of the measured  $l_i$  variable, and  $\rho_{ij}$  is the correlation factor between  $l_i$  and  $l_j$ . Moreover we have  $\rho_{ij} = \rho_{ji}$ . In our cases there is no correlation between the sensors, thus  $\rho_{ij} = 0$  if  $i \neq j$ .  $E[.]$  is the mathematic hope.

The inverse of the covariance matrix

$$\underline{P} = \underline{Q}_l^{-1} \quad (\text{A.3})$$

is called the weight matrix. If we assume  $n$  observations and  $u$  unknown parameters, the design matrix  $\underline{A}$  consists of  $n$  rows and  $u$  columns. The system is redundant when

$n > u$ . In most of the cases,  $n = u$ . The inconsistency is due to observational errors or noise. In order to solve the model and assure consistency, the noise vector  $\underline{n}$  is added to the vector of observations

$$\underline{l} = \underline{A} \underline{x} + \underline{n}. \quad (\text{A.4})$$

The solution of (A.4) becomes unique and consistent if we apply the least square principle, which state that the weighted sum of the squared residuals is minimized (Teunissen, 1995).

In terms of our notation the least square principle is given by

$$\frac{\partial [\underline{n}^T \underline{P} \underline{n}]}{\partial \underline{x}} = 0 \quad (\text{A.5})$$

If we apply this minimum principle to the above observational model, we obtain first the following equation

$$\underline{n}^T \underline{P} \underline{n} = (\underline{l} - \underline{A} \underline{x})^T \underline{P} (\underline{l} - \underline{A} \underline{x}) \quad (\text{A.6})$$

or

$$\underline{n}^T \underline{P} \underline{n} = (\underline{A} \underline{x})^T \underline{P} \underline{A} \underline{x} - \underline{x}^T \underline{A}^T \underline{P} \underline{l} - \underline{l}^T \underline{P} \underline{A} \underline{x} + \underline{l}^T \underline{P} \underline{l}. \quad (\text{A.7})$$

Secondly after derivation (see appendix “Matrix derivation”)

$$\frac{\partial [\underline{n}^T \underline{P} \underline{n}]}{\partial \underline{x}} = \underline{A}^T \underline{P} \underline{A} \underline{x} + \underline{A}^T \underline{P}^T \underline{A} \underline{x} - \underline{A}^T \underline{P} \underline{l} - \underline{A}^T \underline{P}^T \underline{l} = 0. \quad (\text{A.8})$$

Since  $\underline{P}^T = \underline{P}$ , we get

$$2(\underline{A}^T \underline{P} \underline{A} \underline{x} - \underline{A}^T \underline{P} \underline{l}) = 0. \quad (\text{A.9})$$

Hence

$$\underline{A}^T \underline{P} \underline{A} \underline{x} = \underline{A}^T \underline{P} \underline{l}. \quad (\text{A.10})$$

The solution is then simply

$$\underline{x} = (\underline{A}^T \underline{P} \underline{A})^{-1} \underline{A}^T \underline{P} \underline{l}. \quad (\text{A.11})$$

In a more general form

$$\underline{x} = \underline{G}^{-1} \underline{g}, \quad (\text{A.12})$$

where  $\underline{G} = \underline{A}^T \underline{P} \underline{A}$  and  $\underline{g} = \underline{A}^T \underline{P} \underline{l}$ .

The above method gives us a practical solution to noisy measurements. The solution is optimized according to the influence of noise.

### ***Sequential adjustment***

The goal of this method is to improve the solution of Eq. (A.4) by conducting sequential measurements. Each measurement leads to a new estimation of  $\underline{x}$  which will be more and more accurate as we continue. To illustrate this, let us divide the observation model into two sets

$$\begin{cases} \underline{l}_k = \underline{A} \underline{x}_k + \underline{n}_k \\ \underline{l}_{k+1} = \underline{A} \underline{x}_{k+1} + \underline{n}_{k+1} \end{cases}. \quad (\text{A.13})$$

Focusing on the first set, a preliminary solution is given by the Eqs (A.11) and (A.12) as

$$\underline{x}_k = (\underline{A}^T \underline{P} \underline{A})^{-1} \underline{A}^T \underline{P} \underline{l}_k = \underline{G}^{-1} \underline{g}_k. \quad (\text{A.14})$$

The solution  $\underline{x}_k$  is computed to limit to influence of the noise  $\underline{n}_k$  ( $\underline{l}_k \approx \underline{A} \underline{x}_k$ ). Next we consider the second set of measurements, and obtain similarly

$$\underline{x}_{k+1} = (\underline{A}^T \underline{P} \underline{A})^{-1} \underline{A}^T \underline{P} \underline{l}_{k+1} = \underline{G}^{-1} \underline{g}_{k+1}. \quad (\text{A.15})$$

Let us assume that the second measurement is a little bit different to the first one

$$\underline{l}_{k+1} = \underline{l}_k + \Delta \underline{l}. \quad (\text{A.16})$$

Eq. (A.15) becomes

$$\underline{x}_{k+1} = \underline{G}^{-1} \underline{A}^T \underline{P} (\underline{l}_k + \Delta \underline{l}). \quad (\text{A.17})$$

Furthermore, using Eq. (A.14)

$$\underline{x}_{k+1} = \underline{x}_k + \underline{G}^{-1} \underline{A}^T \underline{P} \Delta \underline{l}. \quad (\text{A.18})$$

Since  $\Delta \underline{l} = \underline{l}_{k+1} - \underline{l}_k$ ,

$$\underline{x}_{k+1} = \underline{x}_k + \underline{G}^{-1} \underline{A}^T \underline{P} (\underline{l}_{k+1} - \underline{l}_k). \quad (\text{A.19})$$

Finally  $\underline{l}_k \approx \underline{A} \underline{x}_k$ , and the updated state equation takes the form

$$\underline{x}_{k+1} = \underline{x}_k + \underline{K} (\underline{l}_{k+1} - \underline{A} \underline{x}_k), \quad (\text{A.20})$$

where  $\underline{K} = (\underline{A}^T \underline{P} \underline{A})^{-1} \underline{A}^T \underline{P}$ . The above equation is a linear combination of  $\underline{x}_k$ .

Consequently,  $\underline{A} \underline{x}_k$  may be considered as a prediction of  $\underline{l}_{k+1}$  before any measurements are conducted. **This result is used in Kalman filtering.**

This estimation takes care about the last measurement. The covariance matrix change as well with this last measurement.

### ***Covariance propagation***

The covariance matrix of  $\underline{x}_k = \underline{G}^{-1} \underline{A}^T \underline{P} \underline{l}_k$  is

$$\underline{Q}_{x_k} = (\underline{G}^{-1} \underline{A}^T \underline{P}) \underline{Q}_l (\underline{G}^{-1} \underline{A}^T \underline{P})^T \quad (\text{A.21})$$

or simply

$$\underline{Q}_{x_k} = \underline{G}^{-1} (\underline{A}^T \underline{P} \underline{A})^{-1} \quad (\text{A.22})$$

since  $\underline{Q}_l = \underline{P}^{-1}$ .

It is possible to demonstrate that the new obtained covariance matrix is

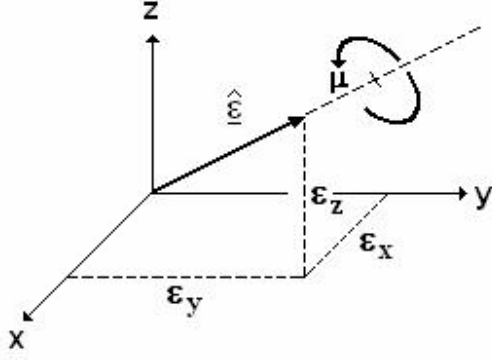
$$\underline{Q}_{x_{k+1}} = (\underline{I} - \underline{K} \underline{A}) \underline{Q}_{x_k} \quad (\text{A.23})$$

The demonstration is the same one made in the Kalman filter equation demonstration.

Thus the chemist is decreasing his incertitude about the ion concentration of the solution.



## APPENDIX B: Quaternion mathematics



Any fixed point rotation of a rigid body can be described as a single rotation of a certain angle (the eigenangle,  $\mu$ ) about a single axis (the eigenaxis of Euler axis,  $\hat{e}$ ). This is known as Euler's theorem.

Based on the Euler's theorem, Hamilton (1866) constructed the quaternion. For our work, only a subset of Hamilton's quaternion calculus is required to do rotations. See also appendix B.

In the following we write a quaternion as a four element vector

$$\underline{q} = [q_s, q_x, q_y, q_z], \quad (\text{B.1})$$

where  $q_s = \cos(\mu/2)$ ,  $q_x = \epsilon_x \sin(\mu/2)$ ,  $q_y = \epsilon_y \sin(\mu/2)$  and  $q_z = \epsilon_z \sin(\mu/2)$ .

It can also be viewed as a scalar coupled to a three-dimensional vector

$$\underline{q} = (q_s, [q_x, q_y, q_z]) = (q_s, \vec{q}). \quad (\text{B.2})$$

The magnitude of the quaternion is then defined as

$$\|\underline{q}\| = \sqrt{q_s^2 + q_x^2 + q_y^2 + q_z^2}. \quad (\text{B.3})$$

The dot product between the quaternions  $\underline{p}$  and  $\underline{q}$  is

$$\underline{p} \cdot \underline{q} = p_s q_s + p_x q_x + p_y q_y + p_z q_z. \quad (\text{B.4})$$

From (B.2) and (B.3) we see that

$$\|\underline{q}\|^2 = \underline{q} \cdot \underline{q}. \quad (\text{B.5})$$

The quaternion is normalized if the magnitude equals 1. Therefore, normalization of a quaternion is conducted by dividing each component with its magnitude.

$\underline{q}_N$  is the unit quaternion of  $\underline{q}$  given by the expression:

$$\underline{q}_N = \frac{\underline{q}}{\|\underline{q}\|} = \frac{[q_s, q_x, q_y, q_z]}{\sqrt{q_s^2 + q_x^2 + q_y^2 + q_z^2}}. \quad (\text{B.6})$$

Furthermore, the conjugate of a quaternion is defined as

$$\bar{\underline{q}} = [q_s, -q_x, -q_y, -q_z]. \quad (\text{B.7})$$

The multiplication of two quaternions under the above rules leads to the definition

$$\underline{p} \circ \underline{q} = [p_s q_s - \vec{p} \cdot \vec{q}, p_s \vec{q} + q_s \vec{p} + \vec{p} \times \vec{q}]. \quad (\text{B.8})$$

A vector  $\underline{v}$  can be rotated rather elegant by the use of a quaternion. We assume that  $\underline{q}$  is a rotation unit quaternion.

$$\underline{v}' = \underline{q} \circ \underline{v} \circ \underline{q}^{-1} \quad (\text{B.9})$$

The above expression is much faster to implement than the transformation using rotational matrices. Note that the inverse is equivalent to calculate the conjugate of the quaternion, if the quaternion is normalized (unit quaternion).

The eigenaxis and the eigenangle can be computed by the following procedure known as Hall's method

$$\begin{aligned}
q_m &= (q_x^2 + q_y^2 + q_z^2) \\
\text{sign}(x) &= \begin{cases} +1 & x > 0 \\ +0 & x = 0 \\ -1 & x < 0 \end{cases} \\
\mu &= 2 \text{ ArcCos}(q_m) \\
\hat{\underline{e}} &= \frac{\text{sign}(q_s)}{q_m} [q_x, q_y, q_z].
\end{aligned} \tag{B.10}$$

The above method will fail if  $q_s = 0$ .

Traditionally, the angles representing the pitch, roll and yaw ( $\phi, \theta, \varphi$ ) are the most used parameters to describe rotations of an aircraft. We call them the Euler angles. A set of Euler angles can be converted to a quaternion by the following equations

$$\begin{aligned}
q_s &= \text{Cos}(\phi/2)\text{Cos}(\theta/2)\text{Cos}(\varphi/2) + \text{Sin}(\phi/2)\text{Sin}(\theta/2)\text{Sin}(\varphi/2) \\
q_x &= \text{Sin}(\phi/2)\text{Cos}(\theta/2)\text{Cos}(\varphi/2) - \text{Cos}(\phi/2)\text{Sin}(\theta/2)\text{Sin}(\varphi/2) \\
q_y &= \text{Cos}(\phi/2)\text{Sin}(\theta/2)\text{Cos}(\varphi/2) + \text{Sin}(\phi/2)\text{Cos}(\theta/2)\text{Sin}(\varphi/2) \\
q_z &= \text{Cos}(\phi/2)\text{Cos}(\theta/2)\text{Sin}(\varphi/2) - \text{Sin}(\phi/2)\text{Sin}(\theta/2)\text{Cos}(\varphi/2).
\end{aligned} \tag{B.11}$$

For small angles we obtain

$$\begin{aligned}
q_s &\approx 1 \\
q_x &\approx \phi/2 = \text{roll}/2 \\
q_y &\approx \theta/2 = \text{pitch}/2 \\
q_z &\approx \varphi/2 = \text{yaw}/2
\end{aligned} \tag{B.12}$$

Similarly, a quaternion is converted back to Euler angles by

$$\begin{aligned}
\phi &= \text{ArcTan}(2(q_y q_z + q_s q_x)/(1 - 2(q_x^2 + q_y^2))) \\
\theta &= -\text{ArcSin}(2(q_x q_z - q_s q_y)) \\
\varphi &= \text{ArcTan}(2(q_x q_y + q_s q_z)/(1 - 2(q_y^2 + q_z^2)))
\end{aligned} \tag{B.13}$$

A quaternion  $\underline{q}$  is called rotational if its normalized and can be converted into a  $3 \times 3$  rotational matrix DCM. This matrix is often called the Directional Cosine Matrix (Klumpp, 1976).

$$\underline{DCM} = \underline{R} = \begin{bmatrix} 1 - 2 \times (q_y^2 + q_z^2) & 2 \times (q_x q_y + q_s q_z) & 2 \times (q_x q_z - q_s q_y) \\ 2 \times (q_x q_y - q_s q_z) & 1 - 2 \times (q_x^2 + q_z^2) & 2 \times (q_y q_z + q_s q_x) \\ 2 \times (q_x q_z + q_s q_y) & 2 \times (q_y q_z - q_s q_x) & 1 - 2 \times (q_x^2 + q_y^2) \end{bmatrix}. \tag{B.14}$$

The inverse procedure, to calculate the quaternion from a rotational matrix, is performed in several stages. If the trace  $T = 1 + R_{00} + R_{11} + R_{22}$  of the matrix is greater than zero then

$$\begin{aligned}
 S &= 2\sqrt{T} \\
 q_s &= S/4 \\
 q_x &= (R_{12} - R_{21})/S \\
 q_y &= (R_{20} - R_{02})/S \\
 q_z &= (R_{01} - R_{10})/S.
 \end{aligned} \tag{B.15}$$

On the other hand, if  $T$  is zero, then we have to identify which major diagonal element that has the greatest value. In the case where  $R_{00} > R_{11}$  and  $R_{00} > R_{22}$  then

$$\begin{aligned}
 S &= 2\sqrt{(1 + R_{00} - R_{11} - R_{22})} \\
 q_s &= (R_{12} - R_{21})/S \\
 q_x &= S/4 \\
 q_y &= (R_{01} + R_{10})/S \\
 q_z &= (R_{20} + R_{02})/S.
 \end{aligned} \tag{B.16}$$

Else if  $R_{11} > R_{22}$  then

$$\begin{aligned}
 S &= 2\sqrt{(1 + R_{11} - R_{00} - R_{22})} \\
 q_s &= (R_{20} - R_{02})/S \\
 q_x &= (R_{01} + R_{10})/S \\
 q_y &= S/4 \\
 q_z &= (R_{12} + R_{21})/S.
 \end{aligned} \tag{B.17}$$

Finally, if the above fails

$$\begin{aligned}
 S &= 2\sqrt{(1 + R_{22} - R_{00} - R_{11})} \\
 q_s &= (R_{01} - R_{10})/S \\
 q_x &= (R_{20} + R_{02})/S \\
 q_y &= (R_{12} + R_{21})/S \\
 q_z &= S/4.
 \end{aligned} \tag{B.18}$$

The translation matrix  $\underline{A}$  that translate the quaternion frame to Euler angles is calculated from the estimated state vector  $\underline{q}$  and the Directional Cosine Matrix  $\underline{R=DCM}$ .

$$\underline{A} = \begin{bmatrix} \frac{\partial \phi}{\partial q_s} & \frac{\partial \phi}{\partial q_x} & \frac{\partial \phi}{\partial q_y} & \frac{\partial \phi}{\partial q_z} \\ \frac{\partial \theta}{\partial q_s} & \frac{\partial \theta}{\partial q_x} & \frac{\partial \theta}{\partial q_y} & \frac{\partial \theta}{\partial q_z} \\ \frac{\partial \varphi}{\partial q_s} & \frac{\partial \varphi}{\partial q_x} & \frac{\partial \varphi}{\partial q_y} & \frac{\partial \varphi}{\partial q_z} \end{bmatrix} = \begin{bmatrix} \frac{2R_{22}q_x}{R_{22}^2 + R_{12}^2} & \frac{2(R_{22}q_s + 2R_{12}q_x)}{R_{22}^2 + R_{12}^2} & \frac{2(R_{22}q_z + 2R_{12}q_y)}{R_{22}^2 + R_{12}^2} & \frac{2R_{22}q_y}{R_{22}^2 + R_{12}^2} \\ \frac{2q_y}{\sqrt{1 - R_{02}^2}} & \frac{-2q_z}{\sqrt{1 - R_{02}^2}} & \frac{2q_s}{\sqrt{1 - R_{02}^2}} & \frac{-2q_x}{\sqrt{1 - R_{02}^2}} \\ \frac{2R_{00}q_z}{R_{00}^2 + R_{01}^2} & \frac{2R_{00}q_y}{R_{00}^2 + R_{01}^2} & \frac{2(R_{00}q_x + 2R_{01}q_y)}{R_{00}^2 + R_{01}^2} & \frac{2(R_{00}q_s + 2R_{01}q_z)}{R_{00}^2 + R_{01}^2} \end{bmatrix}$$

(B.19)

## APPENDIX C:

### Hughes equation:

We have to consider some properties of the quaternion to find this equation. We consider first a rotating system. The solid orientation can be represented as a vector  $\vec{x}$ . This can be noted as a quaternion  $\underline{x}=(0, \vec{x})$ . We can consider the unit rotational quaternion  $\underline{q}$ , such as  $\underline{q}=\left[\cos\left(\frac{\mu}{2}\right), \vec{\varepsilon}_\mu \sin\left(\frac{\mu}{2}\right)\right]$ .

A rotation can be modeled by

$$\underline{x}' = \underline{q} \circ \underline{x} \circ \underline{q}^{-1} = \underline{q} \circ \underline{x} \circ \underline{\bar{q}} \quad (\text{C.1})$$

It transforms the vector  $\vec{x}$  to  $\vec{x}'$ , following a rotation of  $\mu$  around  $\vec{\varepsilon}_\mu$ . Thus we have

$$\underline{x} = \underline{\bar{q}} \circ \underline{x}' \circ \underline{q} \quad (\text{C.2})$$

To establish the Hughes equation, we should derivate (C.2).

$$\dot{\underline{x}} = \dot{\underline{\bar{q}}} \circ \underline{x}' \circ \underline{q} + \underline{\bar{q}} \circ \underline{x}' \circ \dot{\underline{q}} \quad (\text{C.3})$$

using the fact that  $\underline{q}$  is a unit quaternion we have  $\underline{q}^{-1} = \underline{\bar{q}}$ . The Eq. (C.3) and (C.1) yield

$$\dot{\underline{x}} = \dot{\underline{\bar{q}}} \circ \underline{q} \circ \underline{x} + \underline{x} \circ \underline{\bar{q}} \circ \dot{\underline{q}} \quad (\text{C.4})$$

$\underline{q}$  is a unit quaternion, we have  $\underline{q}^{-1} = \underline{\bar{q}}$ , thus  $\underline{\bar{q}} \circ \underline{q} = [1, \vec{0}]$  which is the unit element for the quaternion. We derivate this property and the next relation follows

$$\dot{\underline{\bar{q}}} \circ \underline{q} + \underline{\bar{q}} \circ \dot{\underline{q}} = \underline{0} \quad (\text{C.5})$$

If we assume  $\underline{\bar{q}} \circ \dot{\underline{q}} = [\gamma; \vec{W}]$ , we necessarily have  $\dot{\underline{\bar{q}}} \circ \underline{q} = [-\gamma; -\vec{W}]$ . What is  $\gamma$ ?

We have by definition:

$$\underline{q} \circ \underline{p} = [q_s p_s - \vec{q} \cdot \vec{p}; q_s \vec{p} + p_s \vec{q} + \vec{q} \times \vec{p}] \quad (\text{C.6})$$

thus

$$\begin{aligned} \dot{\underline{\bar{q}}} \circ \underline{q} &= \left[ \cos \frac{\mu}{2}; -\sin \frac{\mu}{2} \cdot \vec{\varepsilon} \right] \circ \left[ \cos \frac{\mu}{2}; \sin \frac{\mu}{2} \cdot \vec{\varepsilon} \right] \\ &= -\frac{\mu}{2} \cdot \left[ \sin \frac{\mu}{2}; \cos \frac{\mu}{2} \cdot \vec{\varepsilon} \right] \circ \left[ \cos \frac{\mu}{2}; \sin \frac{\mu}{2} \cdot \vec{\varepsilon} \right] \end{aligned} \quad (\text{C.7})$$

or simply

$$\dot{\underline{\bar{q}}} \circ \underline{q} = -\frac{\mu}{2} \cdot \left[ \cos \frac{\mu}{2} \sin \frac{\mu}{2} - \sin \frac{\mu}{2} \cos \frac{\mu}{2} \cdot \vec{\varepsilon} \cdot \vec{\varepsilon}; \sin^2 \frac{\mu}{2} \cdot \vec{\varepsilon} + \cos^2 \frac{\mu}{2} \cdot \vec{\varepsilon} + \sin \frac{\mu}{2} \cos \frac{\mu}{2} \cdot \vec{\varepsilon} \times \vec{\varepsilon} \right] \quad (\text{C.8})$$

Then

$$\dot{\underline{\bar{q}}} \circ \underline{q} = -\frac{\mu}{2} \cdot [0; \vec{\varepsilon}] \quad (\text{C.9})$$

Thus  $\gamma=0$ . From this equation we can see the angular velocity appear. We want to have a matrix representation of these above equations. Thus we should express them differently. We have the following system:

$$\begin{cases} \underline{\vec{q}} \circ \dot{\underline{\vec{q}}} = \begin{bmatrix} 0; \frac{\dot{\vec{\mu}}}{2} \vec{\varepsilon} \end{bmatrix} \\ \dot{\underline{\vec{q}}} \circ \underline{\vec{q}} = \begin{bmatrix} 0; -\frac{\dot{\vec{\mu}}}{2} \vec{\varepsilon} \end{bmatrix} \end{cases} \quad (C.10)$$

If we insert Eq.(C.10) into Eq. (C.4), it gives

$$\dot{\underline{\vec{x}}} = \begin{bmatrix} 0; -\frac{\dot{\vec{\mu}}}{2} \vec{\varepsilon} \end{bmatrix} \circ \underline{\vec{x}} + \underline{\vec{x}} \circ \begin{bmatrix} 0; \frac{\dot{\vec{\mu}}}{2} \vec{\varepsilon} \end{bmatrix} = \begin{bmatrix} 0; -\frac{\dot{\vec{\mu}}}{2} \vec{\varepsilon} \times \vec{x} + \vec{x} \times \frac{\dot{\vec{\mu}}}{2} \vec{\varepsilon} \end{bmatrix} \quad (C.11)$$

And then

$$\dot{\underline{\vec{x}}} = \begin{bmatrix} 0; \vec{x} \times \dot{\vec{\mu}} \vec{\varepsilon} \end{bmatrix} = \underline{\vec{x}} \circ \begin{bmatrix} 0; \dot{\vec{\mu}} \vec{\varepsilon} \end{bmatrix} \quad (C.12)$$

We can write the rotation rate vector as

$$\underline{\vec{\omega}} = \begin{bmatrix} 0; \dot{\vec{\mu}} \vec{\varepsilon} \end{bmatrix} = 2 \cdot \underline{\vec{q}} \circ \dot{\underline{\vec{q}}} \quad (C.13)$$

As a consequence it follows

$$\dot{\underline{\vec{q}}} = \frac{1}{2} \underline{\vec{q}} \circ \underline{\vec{\omega}} \quad (C.14)$$

Now we are able to use a matrix representation. We have to consider the following equations:

$$\begin{aligned} \underline{\vec{q}} \circ \underline{\vec{p}} &= [q_s p_s - \vec{q} \cdot \vec{p}; q_s \vec{p} + p_s \vec{q} + \vec{q} \times \vec{p}] \\ &= \begin{bmatrix} q_s & -\vec{q}^T \\ \vec{q} & q_s I_3 + \tilde{\underline{\vec{q}}} \end{bmatrix} \cdot \begin{bmatrix} p_s \\ \vec{p} \end{bmatrix} = \underline{\underline{Q}} \cdot \begin{bmatrix} p_s \\ \vec{p} \end{bmatrix} \end{aligned} \quad (C.15)$$

with  $\tilde{\underline{\vec{q}}}$  defined such as  $\vec{q} \times \vec{x} = \tilde{\underline{\vec{q}}} \cdot \vec{x}$ ; we obtain

$$\tilde{\underline{\vec{q}}} = \begin{bmatrix} 0 & -q_z & q_y \\ q_z & 0 & -q_x \\ -q_y & q_x & 0 \end{bmatrix} \quad (C.16)$$

The introduction of eq. (C.16) into eq. (C.15) gives us

$$\underline{\underline{Q}} = \begin{bmatrix} q_s & -q_x & -q_y & -q_z \\ q_x & q_s & -q_z & q_y \\ q_y & q_z & q_s & -q_x \\ q_z & -q_y & q_x & q_s \end{bmatrix} \quad (C.17)$$

We can write as well, because of anti-symmetric quaternion product properties,

$$\underline{p} \circ \underline{q} = \underline{\overline{Q}} \cdot \begin{bmatrix} p_s \\ \underline{p} \end{bmatrix} \quad \text{with} \quad \underline{\overline{Q}} = \begin{bmatrix} q_s & -\vec{q}^T \\ \vec{q} & q_s \underline{I}_3 - \underline{\tilde{q}} \end{bmatrix} \quad (\text{C.18})$$

We consider the equation (C.14), then

$$\dot{\underline{q}} = \frac{1}{2} \underline{\Omega}(\underline{\varpi}) \cdot \underline{q} \quad , \quad (\text{C.19})$$

where

$$\underline{\Omega} = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} . \quad (\text{C.20})$$

Considering that

$$\dot{\underline{\mu}} \vec{\varepsilon} = \varpi_x \vec{\varepsilon}_x + \varpi_y \vec{\varepsilon}_y + \varpi_z \vec{\varepsilon}_z \quad (\text{C.21})$$

we obtain the Hughes formula, equation (C.19).

**APPENDIX D:**  
**Matrix derivation**

It is forbidden to do

$$\frac{\partial \underline{x}}{\partial \underline{x}} \quad \text{nor} \quad \frac{\partial \underline{x}^T}{\partial \underline{x}^T} . \quad (\text{D.1})$$

On the other hand, it is allowed to

$$\frac{\partial \underline{x}^T}{\partial \underline{x}} = \frac{\partial \underline{x}}{\partial \underline{x}^T} = \underline{I} \quad (\text{D.2})$$

$$\frac{\partial [\underline{P}\underline{x}]}{\partial \underline{x}} = \frac{\partial [\underline{P}\underline{x}]^T}{\partial \underline{x}} = \underline{P}^T \quad (\text{D.3})$$

$$\frac{\partial [\underline{P}\underline{x}]}{\partial \underline{x}^T} = \underline{P} \quad . \quad (\text{D.4})$$

In addition if  $\underline{C} = \underline{C}^T$  then

$$\frac{\partial [\underline{A}\underline{C}\underline{A}^T]}{\partial \underline{A}} = [\underline{C}\underline{A}^T]^T + \underline{A}\underline{C} = 2 \cdot \underline{A}\underline{C} \quad (\text{D.5})$$

If  $\underline{A}\underline{B}$  is a square and  $\underline{C}$  is a symmetric matrix then

$$\frac{\partial [\text{trace}(\underline{A}\underline{B})]}{\partial \underline{A}} = \underline{B}^T \quad (\text{D.6})$$

$$\frac{\partial [\text{trace}(\underline{A}\underline{C}\underline{A}^T)]}{\partial \underline{A}} = 2\underline{A}\underline{C} \quad (\text{D.7})$$

Derivation of a scalar with a matrix:

$$\frac{\partial \alpha}{\partial \underline{A}} = \left[ \frac{\partial \alpha}{\partial a_{ij}} \right] \quad (\text{D.8})$$



## APPENDIX E: The semi-continuous prediction model

In most of cases the time dependent state vector  $\underline{x}(t)$  of a dynamic system may be modeled by a set of differential equations of the first order

$$\dot{\underline{x}}(t) = \underline{F}(t) \underline{x}(t) + \underline{w}(t) \quad (\text{E.1})$$

$\dot{\underline{x}}(t)$  is the time derivative of the state vector,  $\underline{F}(t)$  the dynamic matrix, and  $\underline{w}(t)$  is the driving noise. If we assume that the state vector  $\underline{x}(t_0)$  is known at initial epoch  $t_0$  and that the dynamic matrix  $\underline{F}(t)$  contains only constant coefficients, then the solution for the equations system (E.1) can be written as

$$\begin{aligned} \underline{x}(t) &= \underline{T}(t, t_0) \underline{x}(t_0) + \int_{t_0}^t \underline{T}(t, \tau) \underline{w}(\tau) d\tau \\ &= \underline{T}(t, t_0) \underline{x}(t_0) + \underline{e}(t). \end{aligned} \quad (\text{E.2})$$

The matrix  $\underline{T}$  is called the transition matrix. This matrix can be expressed as a function of the dynamics matrix  $\underline{F}$  through a Taylor expansion of  $\underline{x}(t)$

$$\underline{x}(t) = \underline{x}(t_0) + \dot{\underline{x}}(t_0)(t - t_0) + \frac{1}{2} \ddot{\underline{x}}(t_0)(t - t_0)^2 + \dots \quad (\text{E.3})$$

Using equation (E.1) and neglecting the driving noise  $\underline{w}(t)$

$$\underline{x}(t) = \underline{x}(t_0) + \underline{F}(t_0) \underline{x}(t_0) (t - t_0) + \frac{1}{2} \underline{F}(t_0)^2 \underline{x}(t_0) (t - t_0)^2 + \dots \quad (\text{E.4})$$

In addition, if we introduce  $\Delta t = t - t_0$ , the transition matrix is given as an infinite series with respect to  $\underline{F}$  as

$$\begin{aligned} \underline{T}(t, t_0) &= \underline{I} + \underline{F}(t_0) \Delta t + \frac{1}{2} \underline{F}(t_0)^2 \Delta t^2 + \dots \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \underline{F}(t_0)^n \Delta t^n. \end{aligned} \quad (\text{E.5})$$

In a discrete form Eq. (E.2) may be written as

$$\underline{x}_{k+1} = \underline{T}_k \underline{x}_k + \underline{w}_k, \quad (\text{E.6})$$

where  $t_k = k \cdot \Delta t$  and from Eq. (E.5)

$$\underline{T}_k = \underline{I} + \underline{F}_k \cdot \Delta t \quad (\text{E.7})$$

for  $n=1$ . Eq. (E.6) is known as the Markov process and it is the fundamental system model of prediction in the discrete Kalman filter.

## APPENDIX F:

### Equipments analysis

We want to test the equipments of the Inertial Measurements Units (IMU) to know their properties. It is important to know their accuracy to optimize the Kalman filter, into the platform.

To measure the bias of the gyroscope, the knowledge of the accelerometer's accuracy and the magnetometer's one will permit us to adjust the Kalman filter's parameters. Thus, it is possible to calibrate the Kalman estimator regarding its environment. This would improve its accuracy.

The main sensor we use is an orientation sensor produced by Microstrain. This sensor combines 3 angular rates gyros with 3 orthogonal accelerometers, 3 orthogonal magnetometers, multiplexer and embedded microcontroller to output its orientation in dynamic and static environment.



It utilizes the triaxial gyros to track dynamic orientation and the triaxial DC accelerometers along with the triaxial magnetometers to track static orientation. It provides orientation in matrix, quaternion and Euler formats. The digital serial output can also provide temperature.

We observed that even if there is not any input into the system, the equipment measures a drift. It is due to the gyroscope bias. To evaluate it, and the accuracy of this equipment, we established the following protocol:

#### Protocol:

We measure the magnetic field, the acceleration and the rotation rate of the system in free conditions. That means without any motion. The sensor is on the table. Thus all the output should be constant. The experiment period is 1 minute, because statistically that is enough to observe the system behaviour. It corresponds to almost 1900 sets of measurement.

For each set a large amount of data is recorded; one signal about the Epoch (the sampling of the data and their processing), 3 signals about the accelerometer ( $A_x$ ,  $A_y$ ,  $A_z$ ), 3 from the Magnetometer ( $M_x$ ,  $M_y$ ,  $M_z$ ) and 3 from the turn rate sensor ( $W_x$ ,  $W_y$ ,  $W_z$ ).

Furthermore we assume that the system temperature can affect the equipment accuracy. All the measurement instruments are built with a lot of electronic components which are quite sensitive to the temperature.

That is why we repeat the previous protocol for different temperatures: 32, 38, 42, 44, 47 and 50 Celsius. However, we will not describe all of these experiments. We will only treat the most relevant ones here: at  $T=32$  C,  $T=38$  C,  $T=42$  C and  $T=50$  C.

## Analysis of measurements:

### A. First experiment: Determination of $\underline{Q}_0^-$ and $\underline{R}_k$ .

Date: 29/09/2005

Temperature: 32.18 C

Time of the experiment: 60.51 sec.

1948 sets of data

Sampling time: 25 ms

No movement

The different values which are interesting for each sensors parameter are the mean and the covariance factor (because of the absence of excitations). The variance of the value gives us the spread of the value:

Variance =  $E[(X - \text{mean}(X))^2]$ . The larger the variance is, the less accurate the sensor is.

We can establish the 2 followings tables:

Epoch	Mx	My	Mz	Ax	Ay	Az	Wx	Wy	Wz
0.0312	0.0023	0.0341	0.4747	0.1170	0.0473	-9.9196	-0.0046	0.0049	0.0216

Table F.1: mean value of the 3 sensors in free conditions at T=32.18 C (international unit system).

Epoch	Mx	My	Mz	Ax	Ay	Az	Wx	Wy	Wz
1.26*10 <sup>-5</sup>	4.32*10 <sup>-7</sup>	1.11*10 <sup>-6</sup>	1.17*10 <sup>-4</sup>	0.0013	0.0010	0.0515	5.66*10 <sup>-5</sup>	7.90*10 <sup>-5</sup>	5.02*10 <sup>-5</sup>

Table F.2: variance value of the 3 sensors in free conditions at T=32.18 C (international unit system)

The Epoch is quite constant. Its value is 31.2 ms with a very small variance, inferior to 0.0001 (rad/s)<sup>2</sup>. It corresponds to the sampling time with the processing time. Thus we can consider it as a constant. (The processing time is almost 7 ms).

#### About the magnetometer:

The 3 fields components of the magnetometer are (Mx, My, Mz). It measures the magnetic field of the environment. The magnetometer gives us the yaw angle if we do not use the GPS. Thus, only the Mx and My components are interesting for us since

$$\varphi = \arctan\left(\frac{My}{Mx}\right) - 4.5 \text{ with a flat attitude (the 4.5 deg. corresponds to the correction}$$

due to the difference between the North Pole and the magnetic pole in Svalbard).

Using this formula, we obtain after analysis of the data

$$\sigma_{\varphi}^2 = 3.75 \cdot 10^{-4} \text{ rad}^2 \Leftrightarrow \sigma_{\varphi} = 1.11 \text{ deg.}$$

The compass of the GPS is a slightly more accurate, which is quite practical. We do not have to change either initialisation matrix when we use it. Furthermore we prefer to use the GPS which is located outside, because the optical system is inside the plane; thus the cabin could affect the magnetometer measurements. Furthermore Svalbard is located quite close to the North Pole. Thus an important deviation can affect the Compass and the magnetic field curve is not parallel to the earth curve. It would have been easier along the equator. Thus the magnetic field is not so easy to analyse.

*About the accelerometer:*

We observe from the measurements that all values are not null. For  $A_z$ , the accelerometer measures the gravity constant (which is almost  $9.83 \text{ m/s}^2$ ), thus it is normal to measure something with the same size ( $-10 \text{ m/s}^2$ ). Nevertheless it is not really precise. The  $A_x$  and  $A_y$  components should be null. The fact they are not, comes from the default of the sensor.

Nevertheless, we use this sensor for the measurements of  $\phi$  (roll) and  $\theta$  (pitch).

We use the formula 
$$\begin{cases} \phi = \text{ArcTan}(-A_y / A_z) \\ \theta = \text{ArcSin}(-A_x / \sqrt{A_x^2 + A_y^2 + A_z^2}) \end{cases}$$

If we compute the couple (Roll, Pitch) for each set of measurement, we can deduce the spreading of the attitude measurements, due to the spreading of the accelerometer.

We obtain 
$$\begin{cases} \sigma_\phi^2 = 1.0510 \cdot 10^{-5} \text{ rad}^2 \\ \sigma_\theta^2 = 1.3556 \cdot 10^{-5} \text{ rad}^2 \end{cases} \Leftrightarrow \begin{cases} \sigma_\phi = 0,19 \text{ deg} \\ \sigma_\theta = 0,21 \text{ deg} \end{cases}$$

Thus the following measurement covariance matrix used in the Kalman filter, in these experimental conditions is

$$\underline{R}_k = \begin{bmatrix} \sigma_\phi^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\varphi^2 \end{bmatrix}_k = \begin{bmatrix} 1.0510 \cdot 10^{-5} & 0 & 0 \\ 0 & 1.3556 \cdot 10^{-5} & 0 \\ 0 & 0 & 3.74 \cdot 10^{-4} \end{bmatrix}$$

We assume that this Matrix is constant during the experiments time.

*About the gyroscope/turn rate sensor:*

We have to consider Table F.1 and F.2, but also Fig. F.1.

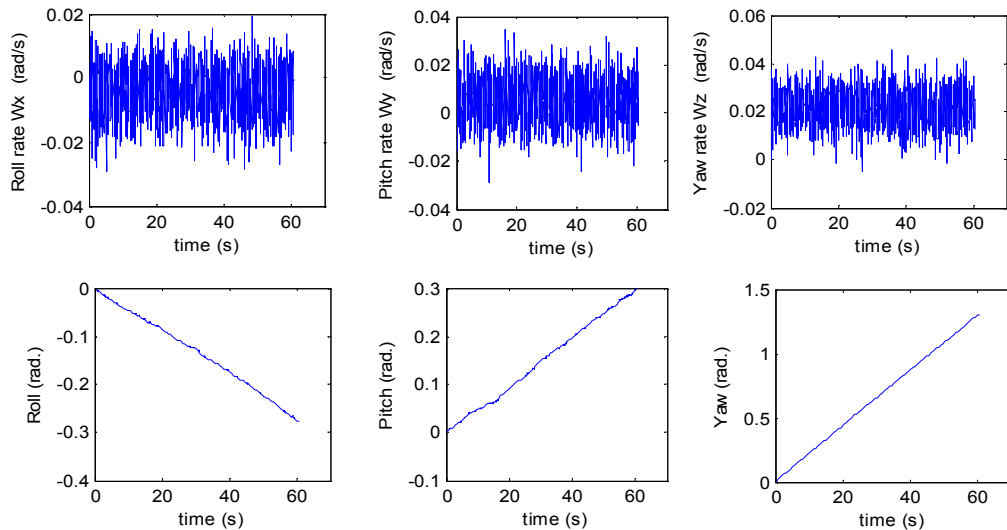


Figure F.1: bias of the gyro in free conditions at  $T=32.18 \text{ C}$  and the resulting attitude bias.

We can see that the Roll, the Pitch and the Yaw computed via the turn rate sensor have quite linear variations. Thus the most relevant factor to know is the derivative of the attitude angles, and one property is fundamental: this derivative (the rate or the

angular velocity) is almost constant. It means that the bias is not really time dependent, at a constant temperature.

We can approximate this bias by its mean value, hence regarding Table 1,

$$(W_r\_bias, W_p\_bias, W_y\_bias) \approx (-0.0046, 0.0049, 0.0216) rad/s$$

$$or (-0.26, 0.28, 1.24) deg/s$$

The noise of the turn rate sensor has noticeable amplitude (variance of  $0.0001 rad^2/s^2$ ); furthermore after analysis of the spectrum, we can affirm that it is not a white noise. Thus to approximate this bias at a constant temperature to its mean value is quite relevant but not perfect.

Thus we can consider an improvement for our Filter. The Euler angle rate we use in the matrix  $\underline{\Omega}_k$  can be written as  $\omega_i = \omega_i - W_i\_bias$  for  $i = x, y, z$ .

It is important to consider the bias of the gyroscope because it affects the control of the attitude strongly. The sensor sends some wrong information to the filter. This information has to be filtered.

We have to determine  $\underline{Q}_0^-$  as well. We want to make the initialisation with the

quaternion  $\hat{q}_0^{-T} = [1 \ 0 \ 0 \ 0]$ . This means we do not input any rotations into the system at the beginning.

For the integer solution, we have to do this initialisation with a perfectly flat attitude, it means when  $(A_x, A_y) \approx (0, 0)$ . Thus it is necessary to consider as well the mean value of  $A_x$  and  $A_y$ . The plane is not perfectly horizontal at  $(0, 0)$  but at  $(0.1170, -0.0473) m/s^2$  (regarding on table 1). This is due to the accelerometer noise.

For small angles (like at the initialization) we have

$$\begin{cases} q_s \approx 1 \\ q_x \approx \phi / 2 = roll / 2 \\ q_y \approx \theta / 2 = pitch / 2 \\ q_z \approx \varphi / 2 = yaw / 2 \end{cases}$$

Regarding these formulas, we could have these relations:

$$\begin{cases} \sigma_{q_x}^2 = \frac{\sigma_\phi^2}{4} \\ \sigma_{q_y}^2 = \frac{\sigma_\theta^2}{4} \\ \sigma_{q_z}^2 = \frac{\sigma_\varphi^2}{4} \end{cases}$$

But we should also consider the covariance of the gyro which is drifting; thus a noisy gyro reminiscence could be measured. We have  $\sigma_{gyro}^2 \approx 10^{-4} rad^2/s^2$  (with a tolerance).

Thus the covariance matrix becomes approximately

$$\underline{Q}_0^- = \begin{bmatrix} \sigma_{q_s}^2 & 0 & 0 & 0 \\ 0 & \sigma_{q_s}^2 & 0 & 0 \\ 0 & 0 & \sigma_{q_y}^2 & 0 \\ 0 & 0 & 0 & \sigma_{q_z}^2 \end{bmatrix} \approx \begin{bmatrix} 5 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 5 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 5 \cdot 10^{-5} & 0 \\ 0 & 0 & 0 & 5 \cdot 10^{-5} \end{bmatrix}$$

All of these values are very small. Thus we know quite accurately the initial position.

It is important to notice that these results are available at the temperature  $T=32.18$  C.

NB: We can consider  $\sigma_{q_s}^2 \approx 5 \cdot 10^{-5}$  because if we consider the Gainer's formula of  $q_s$  in the general case (appendix "Quaternion"), we should obtain  $\sigma_{q_s}^2$  as the same order as the other components of the quaternion.

## B. The Other Experiments: The influence of the temperature on the sensors.

We will consider the other experiments with different temperature conditions to assess the influence of this factor on the measurement equipment.

*The temperature dependence of the gyro bias:*

As we mentioned already, it is important to estimate this "wrong" information. We noticed that it is possible to estimate the bias as the average of itself. But this average has different values depending on temperature. Thus we will consider 4 experiments, including the first one: at  $T=32.18$ C, at  $T=38.31$ C,  $T=44.21$ C and  $49.87$ C.

We obtain the following graph:

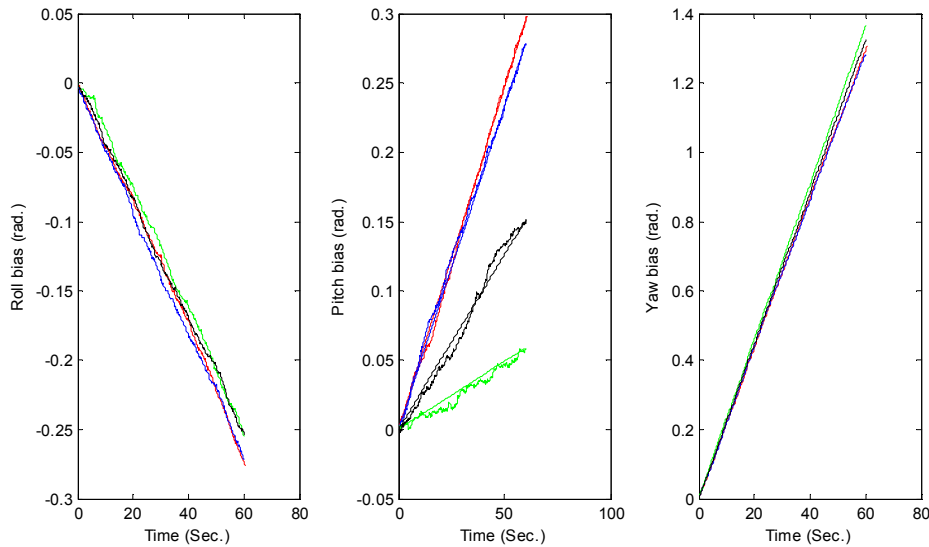


Figure F.2: Bias evolution at  $T=32.18$ C (red),  $38.31$ C (blue),  $44.21$ C (black) and  $49.87$ C (green).

Many different properties appear on this figure.

- Firstly concerning the Roll and the Yaw bias: the bias always follows the same linear function. Thus we can deduce that the turn rate bias is nearly the same for all temperatures. Furthermore, we can approximate it, regarding the fact there is no spreading of these values, hence

$$(W_{roll\_bias}, W_{yaw\_bias}) \approx (-0.0046, 0.0217) \text{ rad/s} \\ \text{or } (-0.26, 1.24) \text{ deg/s}$$

- Secondly, regarding the Pitch: for T=32 C and 38 C, the bias is nearly the same. But when the temperature is increased, 2 phenomena appear:
  - The turn rate bias has less amplitude
  - The spread of the turn rate bias increases. We can still approximate it regarding on a linear function, but this is less accurate than at low temperature.

The decrease of the angular rate angle is not linear with respect to the temperature. The temperature is not constant over the time, as a consequence the turn rate bias is not either. Thus we have to find another way to estimate the gyro turn rate bias.

*The accelerometer:*

We consider only Ax and Ay for the average, because only these values are interesting to determine the initial position.

	T=32C	T=38C	T=44C	T=50C
<b>Ax (m/s<sup>2</sup>)</b>	0.1170	0.1561	0.1911	0.2343
<b>Ay (m/s<sup>2</sup>)</b>	0.0473	-0.0228	-0.0991	-0.1667

Table F.3: average of the accelerometer component at different temperatures

For the accuracy we should as well consider the Az component:

	T=32C	T=38C	T=44C	T=50C
<b>Ax (m<sup>2</sup>/s<sup>4</sup>)</b>	0.0013	0.0023	0.0019	0.0014
<b>Ay (m<sup>2</sup>/s<sup>4</sup>)</b>	0.0010	0.0010	0.0011	0.0011
<b>Az (m<sup>2</sup>/s<sup>4</sup>)</b>	0.0515	0.0009	0.0009	0.0475

Table F.4: variance of the accelerometer components at different temperatures

Regarding the tables F.3 and F.4, the Ax and Ay accuracies are hardly affected by the temperature. Only the mean values show “important” variations, which could be a problem to establish the initial position of the integration model. Regarding these 2 tables, it occurs with the couple  $(Ax, Ay) \approx (0.1900, -0.0600) \text{ m/s}^2$ .

The Az accuracy is temperature dependant. That could be a problem to establish the matrix  $\underline{R}_k$ . This matrix is also temperature dependent and thus time dependant. Our assumption “ $\underline{R}_k$  is not time dependant is wrong” but Table 4 shows that the largest variance occurs at T=32C. Thus, if we use the matrix established previously at T=32C, it will not be really a problem. Only the accuracy will be underestimated (that is better than to be overestimated).

*The magnetometer:*

This sensor is not so much used because of the experimental conditions. We can compile the following table:

	<b>T=32C</b>	<b>T=38C</b>	<b>T=44C</b>	<b>T=50C</b>
<b>Mean Yaw</b>	1.50 rad. (or 85.9 deg.)	1.49 rad. (or 85.4 deg.)	1.48 rad. (or 84.8 deg.)	1.38 rad. (or 79.3 deg.)
$\sigma_{\phi}^2$ (rad <sup>2</sup> )	$3.73 \cdot 10^{-4}$	$3.97 \cdot 10^{-4}$	$3.97 \cdot 10^{-4}$	$4.53 \cdot 10^{-4}$
<b>Yaw (rad.)</b> using mean Mx and My	1.503 (=86.1 deg.)	1.491 (=85.4 deg.)	1.474 (=84.4 deg.)	1.385 (=85.1 deg.)

Table F.5: influence of the temperature on the yaw measurement, regarding on the magnetometer

The information given by the magnetometer can be affected a bit by the temperature. The warmer the environment is, the less accurate the magnetometer is. This is another good argument to prefer the GPS to the magnetometer.

## CONCLUSION:

To use the Kalman filter in optimal conditions, we have to adjust it taking into account its environment. That is why the knowledge of the sensor properties is fundamental to run it.

This analysis was also necessary to establish the initial matrices of the filter. However, the temperature dependence of the accelerometer's accuracy forces us to take some merge.

We underlined that the gyro bias is temperature dependent and time dependant, which is quite annoying. Its evolution depends on the environment. An approximation to its mean value is not so relevant. We need to track it.



## APPENDIX G: Configuration of the low pass filter

The equipments analysis showed that the accelerometer is rather noisy. It may have an influence on the system accuracy. The wanted estimation has to be quite accurate, thus adding a low pass filter on the accelerometer could improve the estimation.

We use a Butterworth low pass filter. We bought it (for \$ 24). This filter is coded and processes the data in real time.

It is necessary to estimate the cut off frequency. Thus we made a spectral analysis of the accelerometer with and without any movements.

### *The noise analysis*

We actually have two different accelerometers. The study will only deal with the one I used. The second one, less noisy, is already on the platform.

To estimate the noise of the accelerometer, we put it on the desk, with no shaking (stable). Measurements are obtained and after processing of the raw data, we obtain the power spectrum of the 3 accelerations noise components.

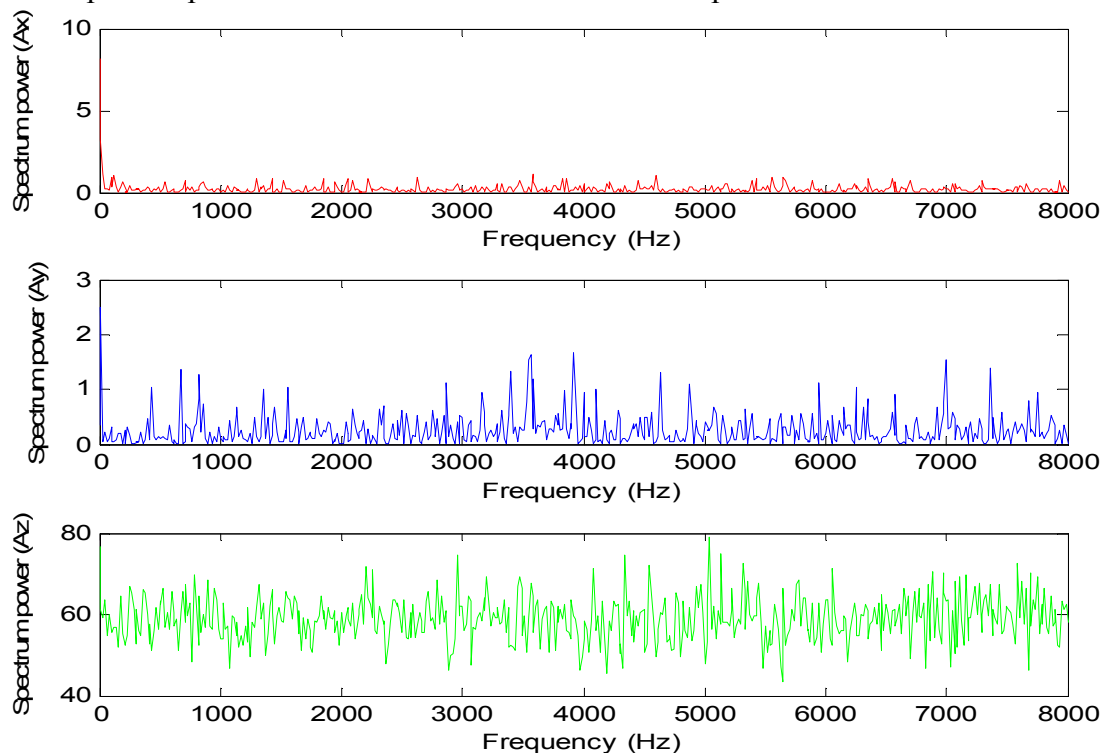


Figure G.1: power spectrum of the noise from the accelerometers without the low pass filter. From the top to the bottom row: Ax, Ay and Az noise components.

The most noisy component is clearly Az. That is a problem because when we track the gravity constant, the main component is always on Az. The 2 other ones also are quite noisy, but on a really limited scale.

We want to use a low pass filter on the accelerometer signal for 2 reasons:

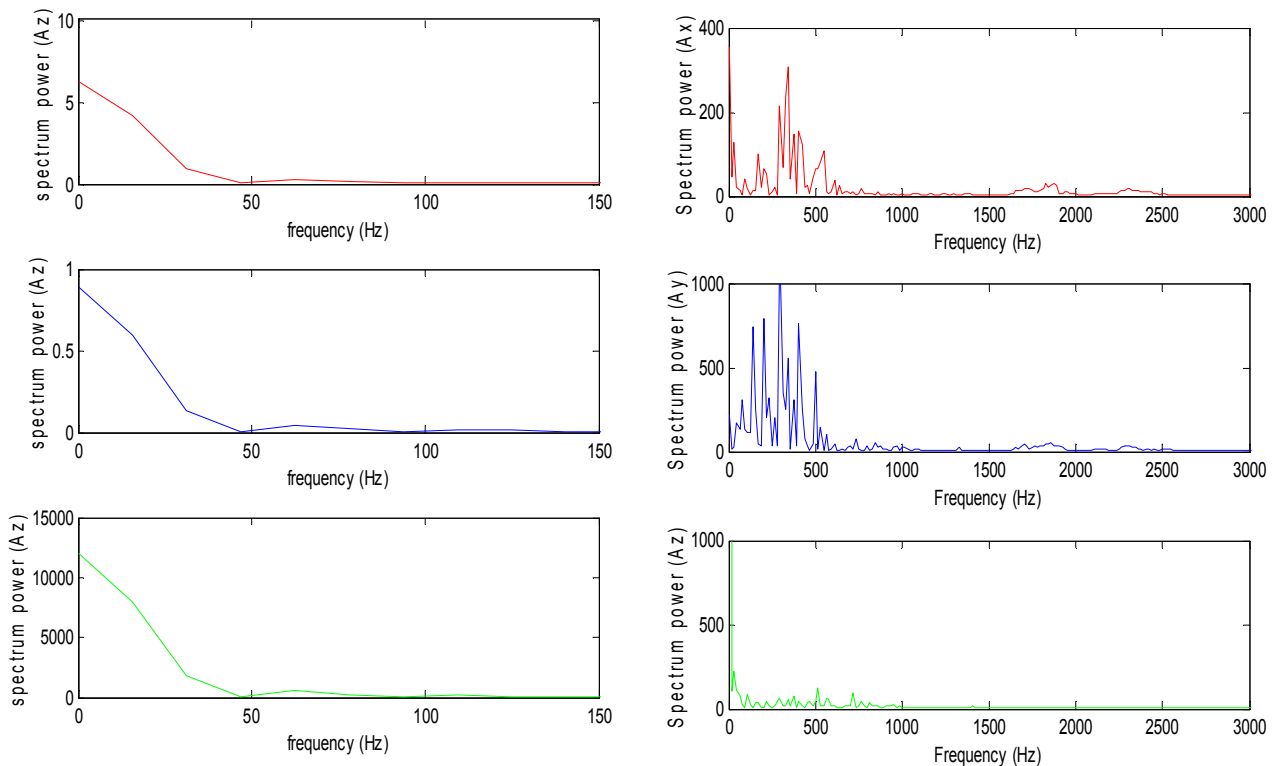
- to remove big sudden inertial accelerations. On a flight it could happen on bad weather conditions.
- to remove the small oscillations of the sensed acceleration (noise of the accelerometer and also the noise of the plane itself). To manage it we should choose an appropriate cut off frequency. Too low, and the filter will remove the normal movements we want to describe, and too high, no benefits.

The application of a low pass filter should improve the estimated attitude.

### ***Determination of the cut off frequency:***

Protocol: we want to determine the frequencies corresponding to the normal motion of the system, resulting from the plane's motion. The experiment is divided in 2 parts:

- we measure the 3 components of the accelerometer (acceleration + noise and not only noise as previously) without any movements and we estimate the spectrum power.
- We repeat this first experiment, shaking the system, as if we were in a plane.



**Figure G.2: spectrum power of the accelerometer in steady state (left), and in shaky conditions (right). From the top to the bottom: Ax, Ay and and Az.**

We can see that without any shaking, no motion, we have only low frequencies on the spectrum. The displayed spectrum power is narrow (only 150 Hz) because at higher frequencies it is totally flat. That is normal because we measure a constant.

Afterwards we shake the system in all directions, and we can see appear some other frequencies. These frequencies are due to the shaking and should not be filtered (we display only until 3000 Hz for the same reason as before).

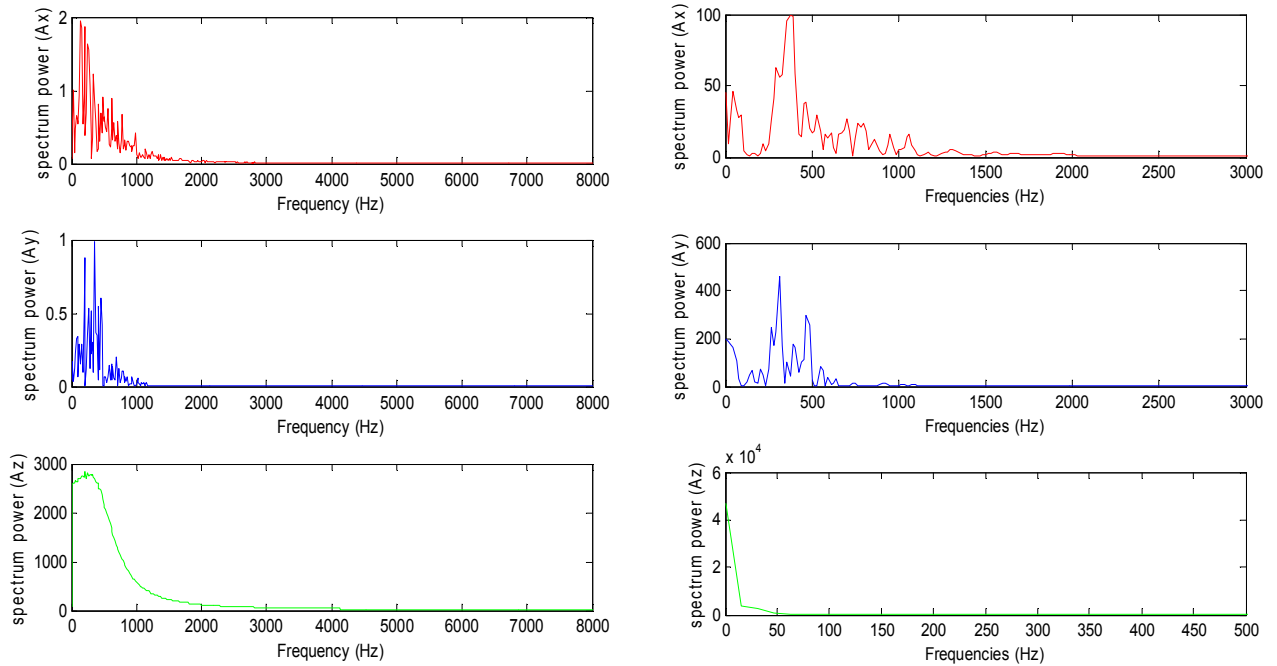
Thus we can deduce from these 2 graphs that the movement data resulting from the motion correspond to a range of frequencies from 0 to almost 2500 Hz. Thus the cut off frequency should not be under this threshold of 2500 Hz (which is quite high).

Thus we want to take a merge choosing a cut off frequency of 3000Hz.

Nevertheless we would like to remark concerning the sampling period. To digitalise a signal with such frequencies, the sampling time should be lower than 0.2ms (Nyquist-Shanon). Regarding on the processing time, 7ms, this is not possible. That is why the system can not respond instantaneously to sharp variations. Besides, we have to be aware that a plane does not roll, neither pitch, neither yaw with such fast movements. Thus 25ms is a short enough sampling period, even if I agree, it is not perfect.

### *Effects of the low pass filter:*

The chosen low pass filter is a 2<sup>nd</sup> order Butterworth filter, because it is possible to use it in real time. This filter is coded.

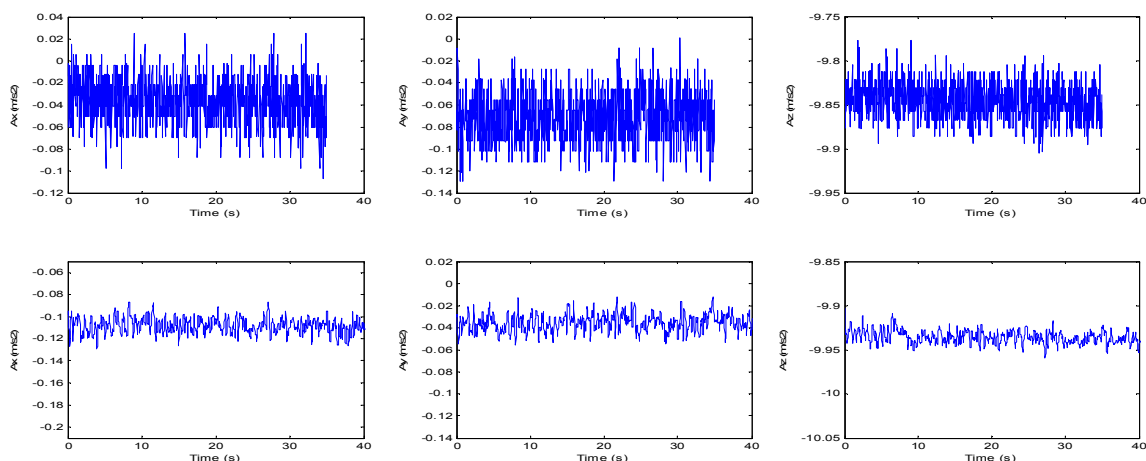


**Figure G.3: Effect of the low pass filter on the accelerometer's noise components for 2 different cut off frequencies: 1500Hz (left) and 3000Hz (right). The sensor was shaken.**

We can see that the filter works perfectly. In Fig.G.3 it is possible to observe its effects on the accelerometers noise. To test it, the chosen cut off frequency was 1500 Hz. It is clear that the noise is almost completely removed (nevertheless with such low cut off frequency, the filter slows down the estimator; it does not react in real time).

Regarding the shaking, the spectrum power is almost the same as the one shown in Fig. G.2. It means the movements are preserved (the cut off frequency was then 3000Hz and the estimator reacts in real time; thus the information corresponding to the movement is not removed).

On Fig.G.4, the effect of the filter in the time domain is obvious:



**Figure G.4: Effect of the low pass filter (3000Hz) on the accelerometers components (top: without filter; bottom: with it).**

The filter removed a large part of the noise. The 2 measurements were made at almost the same temperature. During the experiment (short time) the sensor should measure some constants. Thus the variance of these components provides us the accuracy:

	$\sigma_{a_x}^2 (m^2 / s^4)$	$\sigma_{a_y}^2 (m^2 / s^4)$	$\sigma_{a_z}^2 (m^2 / s^4)$
<b>Without l.p.f.</b>	0.0004	0.0005	0.0004
<b>With l.p.f.</b>	0.00005	0.00006	0.00005

Table G.1: variance of the accelerometers components: with and without the l.p.f.

The accelerometers components are really less noisy and by consequence, the estimated attitude can be more precise as shown on the next figure, Fig. G.5.

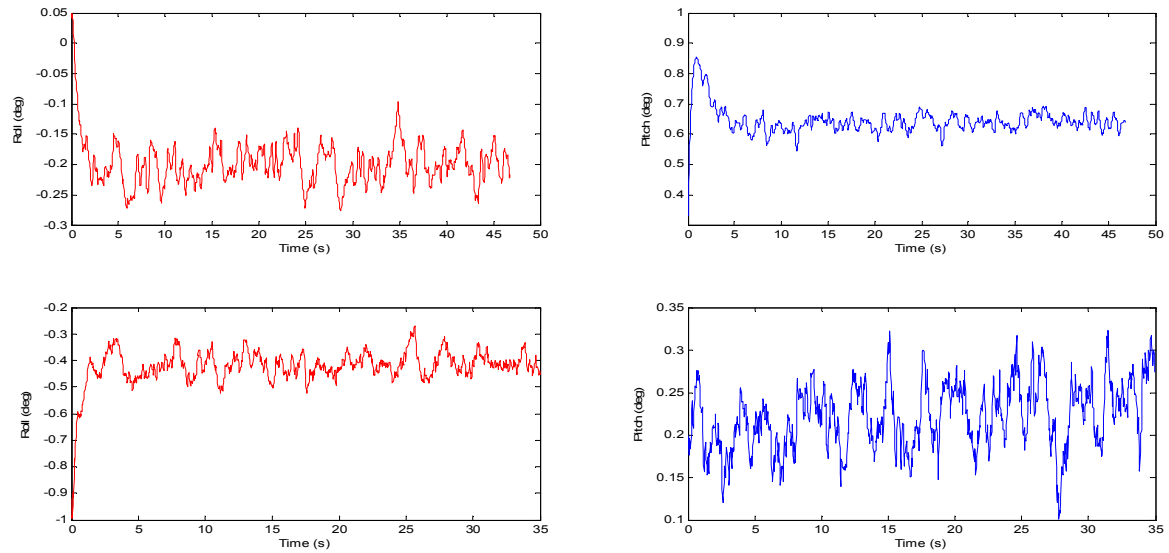


Figure G.5: estimated attitude with l.p.f. (Top row) and without (bottom row). Red: Roll; Blue: Pitch.

It is useless to display the yaw because it is provided by the GPS and not via the accelerometer.

The following table summarize the results:

	$\sigma_{roll}^2$ in $deg^2$	$\sigma_{pitch}^2$ in $deg^2$
With l.p.f	0.0014	0.0020
Without l.p.f.	0.0043	0.0015

Table G.2: effect of the low pass filter on the estimated attitude.

The measurements are processed to obtain the attitude by the following formulas:

$$\begin{cases} roll = \arctan(-a_y / a_z) \\ pitch = \arcsin(-a_x / \sqrt{a_x^2 + a_y^2 + a_z^2}) \end{cases}$$

The effect of the l.p.f. is more noticeable on the Roll than on the Pitch.

The low pass filter improved a little bit the estimated attitude. The last test is done under static conditions. The influence of the filter is more important for the detection of steady conditions (c.f. appendix “The steady parameters”). Both are directly based on accelerometer data. Thus it is possible to detect the movement conditions better.

## APPENDIX H: Optimisation of the deviation factor threshold

To use the combined estimator in the best way, we should calibrate the threshold values for switching from the Kalman filter to the gyro model and the vice versa. However, only the deviation factor threshold can really be adjusted.

To manage it we put the system in a trolley and we moved it with different movements, with always a constant attitude (Roll, Pitch)  $\approx (-1, -15)$ : shaking, strong accelerations, slow ones as well as steady conditions.

For four deviation factor thresholds, which are 94, 95, 97 and 98, we obtain the following results, shown in Fig.H.1.

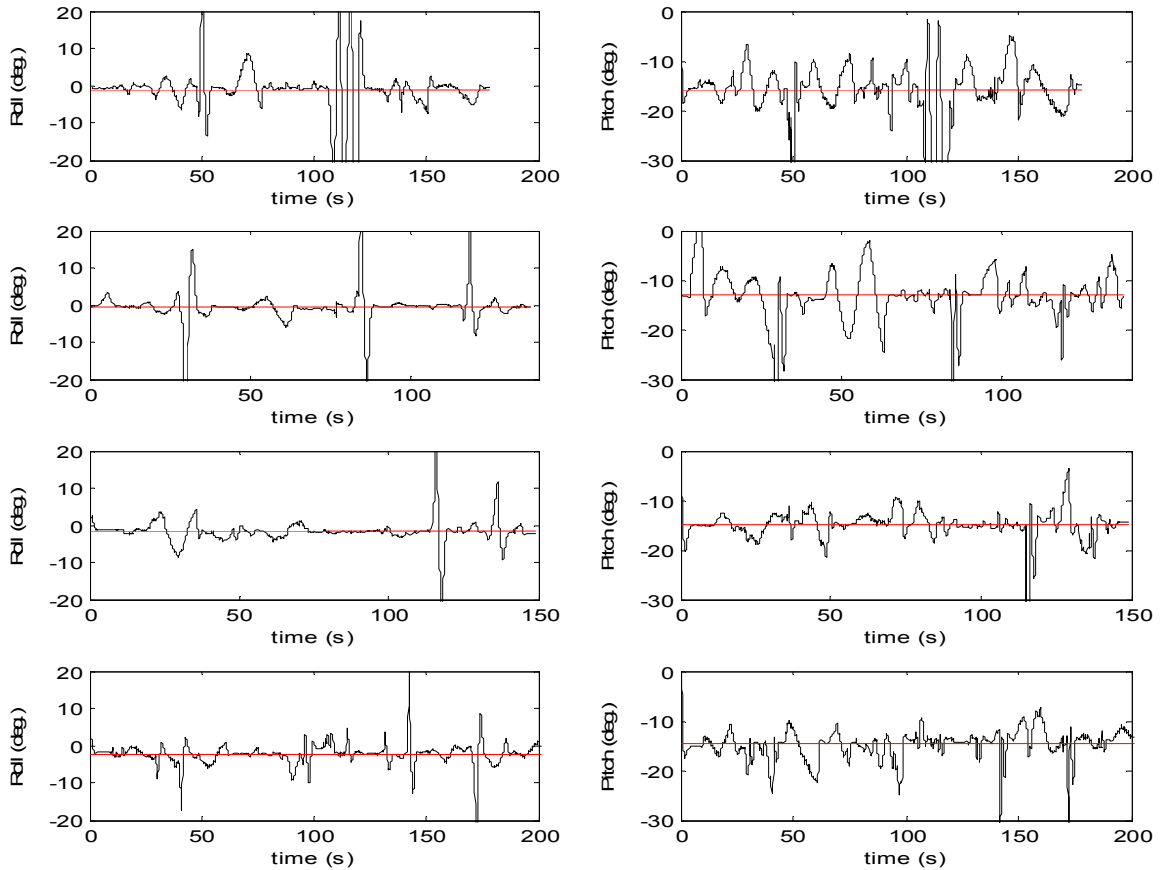


Figure H.1: Observed attitude in a moving trolley for different “deviation” factor thresholds.

Red: mean value (true value); Black: estimated attitude.

From the top row to the bottom one: the threshold is 94, 95, 97 and 98.

We can observe some very sharp variations, which appear more strongly for the Roll, for the 4 tests. These variations appear as well on the Pitch. They correspond to a 180 deg rotation of the Yaw. It means the U-turn of the trolley. We made these tests indoor, thus it was impossible to use a GPS. That is why we observe this behaviour (for example at  $t=50s$  and  $t=110s$  for the test “94”). For a full rotation of the sensor, the measurement stays null, even if the Markov matrix receives from the turn rate sensor some information testifying of a change of course direction. These 2 information contradict each other, thus the estimator becomes unstable and correct itself quickly.

Nevertheless, we can see that the optimal factor seems to be 97. This judgement is qualitative. The estimated attitude is not perfect, but quite close to the true attitude and closer compared to the other threshold values.  
Consider the following table H.1, which shows quantitatively the comparison criteria.

Deviation threshold	94	95	97	98
Mean Roll (deg.)	-1.24	-0.62	-1.62	-2.30
Mean Pitch (deg.)	-15.98	-13.14	-14.84	-14.67
Variance Roll (deg <sup>2</sup> .)	61.64	37.32	15.68	10.33
Variance Pitch (deg <sup>2</sup> .)	44.60	27.75	9.63	7.91
$\sigma_{Roll}^2 + \sigma_{pitch}^2$ (deg <sup>2</sup> .)	106.24	65.07	25.31	18.24

Table H.1: statistical comparison of the attitude provided by the optimal solution for 4 different thresholds.  
Blue: best values; red: worst values.

When I pushed the trolley in the corridor, as shown on the picture bellow, we consider that the corridor is quite flat and the measured Pitch is the one of the trolley board. Thus we assume that it should stay constant even if this is not perfectly true. The Roll can have some small variations. That is why we consider that the variance is a quite good indicator, to assess the accuracy of the system.



Figure H.2: Me pushing the trolley to test the accuracy of the estimator in realistic conditions.

Regarding this table, the most adequate deviation threshold to estimate the attitude of the trolley is 98. With a Roll variance of 10.33 deg<sup>2</sup> and a pitch variance of 7.91 deg<sup>2</sup>, we can have a quite good estimation such as

$$\begin{cases} \phi = -2.30 \pm 3.21 \text{ deg.} \\ \theta = -14.67 \pm 2.81 \text{ deg.} \end{cases}$$

We can notice that the Roll variance exceeds the Pitch one, because of the U-turns we underlined previously. If we use the GPS as planned, this Roll variance should strongly decrease.

As conclusion, the estimator is optimized for a threshold of 97 either 98. It depends if we want to have a qualitative or a quantitative estimation.

NB: In any cases, the estimation is not biased (the maximum bias is almost 0.2 deg. which can be neglected).

## APPENDIX I: The steady parameters

For the combined estimator, a switch between the Kalman filter and the integration solution occurs, regarding the environmental conditions.

If the motion is steady, it means with a constant velocity vector (same magnitude, same direction), the Kalman filter is used. It permits an auto correction, and a perfect initialisation.

Otherwise, if the conditions are not steady, the integration model is used.

Thus it is really necessary to detect both these conditions. Two parameters are used on purpose, which will be discussed below.

### *First parameter: the gravity factor*

We considered  $g_m = \sqrt{a_x^2 + a_y^2 + a_z^2}$  via measurements provided by the accelerometer.

If the plane has a constant velocity vector, there is no external force imposed on the system. Thus the measurement  $g_m$  is quite close to the earth gravity constant which is almost  $9.83 \text{ m/s}^2$  in Svalbard.

If the plane is turning or simply having a linear acceleration (or deceleration), everything inside the plane will experience the pseudo-force of inertia. Thus,  $g_m$  is even larger than 9.83. It can be smaller if the plane is in free fall. In this case, the conditions are not steady either.

The inertial forces resulting from a plane rotation are described in the appendix “Effects of a plane rotation on the accelerometer”.

By consequence the used criteria is the following one:

$$\begin{cases} \text{if } \frac{|g_m - \text{gravity}|}{\text{gravity}} \cdot 100 \leq \text{threshold} : & \text{steady} \\ & \text{else no steady conditions} \end{cases}$$

It is really necessary to use a relative criteria because it is based on measurements. It means because of the accuracy of the accelerometer, even if the sensor is not moving, it could measure  $10.02 \text{ m/s}^2$  instead of 9.83. Actually the range of sensed gravity is from  $+9.60$  to  $10.20 \text{ m/s}^2$  after analysis, depending on the temperature. Thus the gravity value has to be updated quite often.

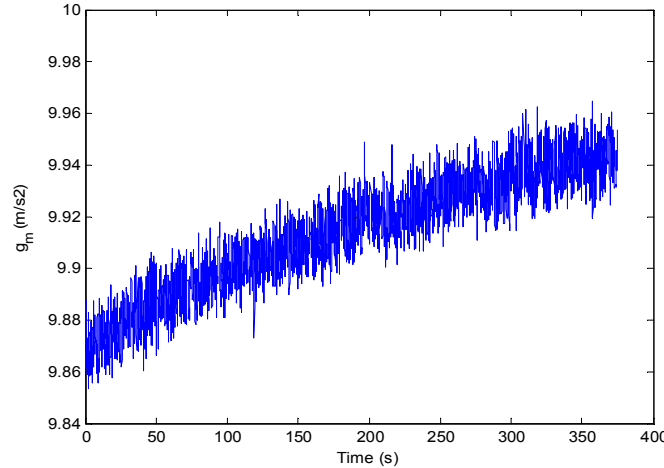


Figure I.1: evolution of the sensed gravity over the time in steady conditions

On the previous graph, it is obvious that the accelerometer is not really precise and that the sensed gravity increases a lot and quite quickly over the time.

About the threshold, it can not be lower than 1.5% because of the accelerometer precision.

Regarding on the equipment analysis, the most noisy component is Az. Its variance is almost  $0.05 \text{ m}^2/\text{s}^4$ , which corresponds to  $0.23 \text{ m/s}^2$ . This corresponds to nearly 2% of gravity; but the low pass filter was not used. With it, this rate can become 1.5%.

On Fig.I.1, the low pass filter is used.

NB: on the appendix “Equipments analysis” we consider the variance to estimate the accuracy of the accelerometer. The experiment was quite short, thus we can approximate the measured gravity without the noise sensor to a constant during a short time. Thus the variance can provide the standard deviation. On Fig.I.1, the experiment is quite long and this last remark is not true anymore.

### ***Second parameter: the deviation***

We considered another state factor: the deviation of the accelerometer.

The 3 accelerometers components are stored at a given epoch  $t_k$ . We obtain the measurement triplet  $(a_{x\_k}, a_{y\_k}, a_{z\_k})$ . We continue the process and we obtain at the next epoch the triplet  $(a_{x\_k+1}, a_{y\_k+1}, a_{z\_k+1})$ .

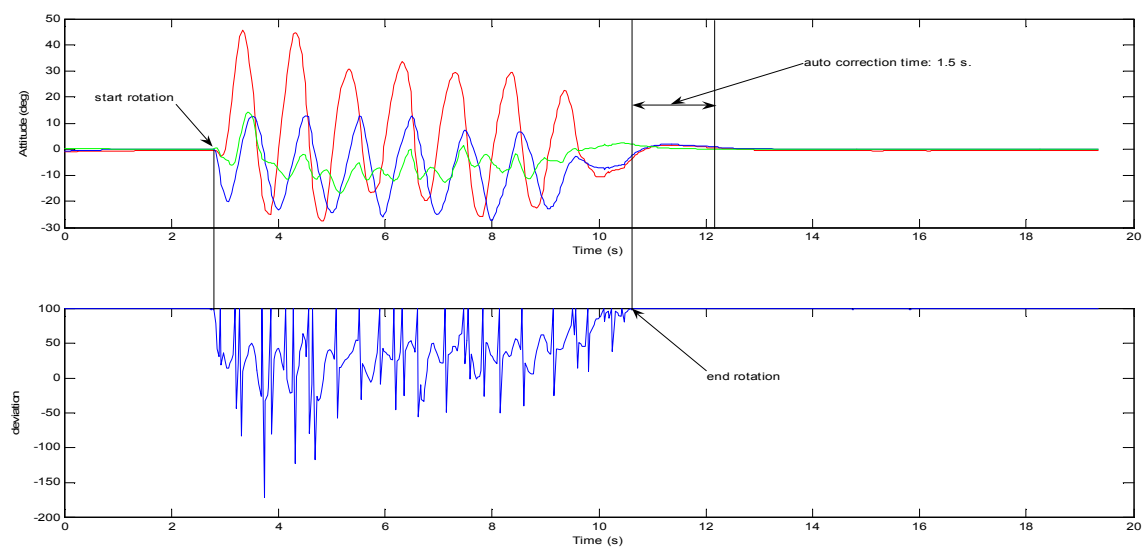
We defined the deviation factor as

$$\text{deviation}_{k+1} = \left( 1 - \sqrt{(d_{x\_k+1})^2 + (d_{y\_k+1})^2 + (d_{z\_k+1})^2} \right) \cdot 100$$

With

$$d_{i\_k+1} = a_{i\_k+1} - a_{i\_k} \text{ for } i = x, y, z.$$

This factor gives us information about the movement of the system. If the system is rotating, the factor will decrease. We just have to find a limit threshold depending on the application.



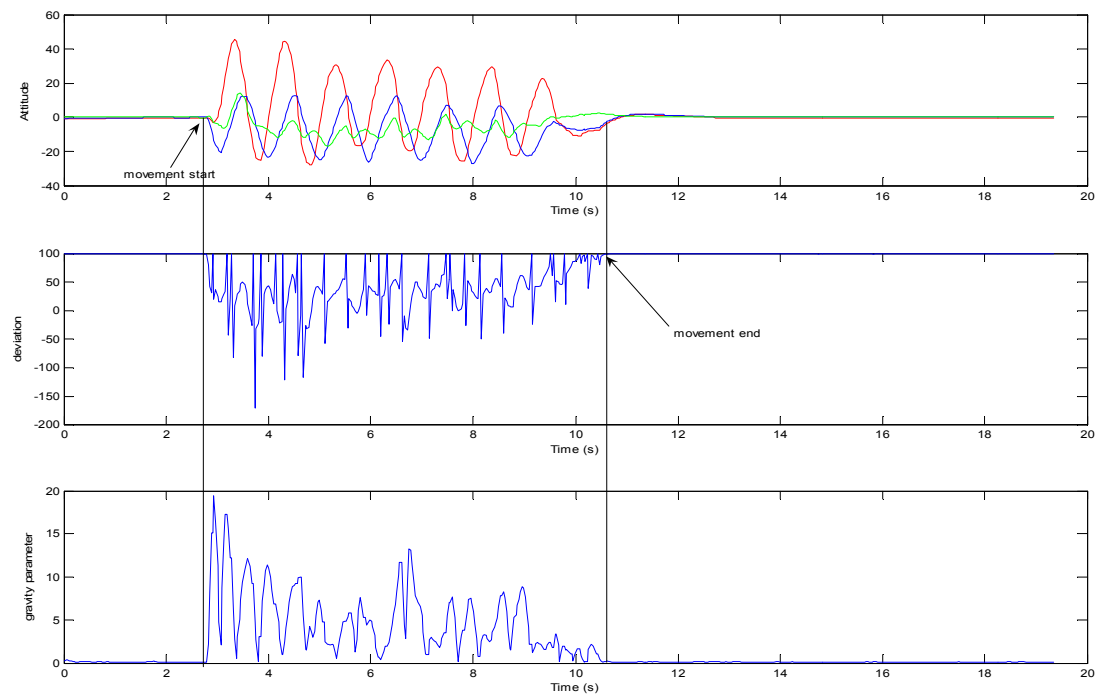
**Figure I.2: the deviation criteria (bottom) when the system is doing a solid rotation. Top: estimated attitude using the combinative solution; red: roll; blue: pitch; green: yaw.**



The second factor reacts strongly to the solid rotation as shown in Fig.2. In this figure, the combinative solution is used to estimate the attitude. Moreover, still in this graph, it is possible to see the auto correction time of the system which is almost 1.5s.

### ***Conclusion***

The Fig.I.3 illustrates quite well the reaction of both factors to non-steady condition movement:



**Figure I.3: reaction of the both parameter regarding on random movements.**

**Top row: Attitude; Red: Roll; Blue: Pitch; Green: Yaw. (in degrees).**

**Medium row: Deviation parameter.**

**Bottom row: Gravity parameter.**

Both parameters detect perfectly if the conditions are steady or not. However it is necessary to use both:

- If the movement has a constant acceleration, the deviation factor will not detect it, only the gravity one.
- If the movement is a perfect solid rotation, without any inertial forces, the first criteria will not detect it, only the second one.

Thus the combination of both is necessary.

## APPENDIX J: Effects of the plane rotation on the accelerometer

To run the Kalman filter, an accelerometer is used and by consequence the inertial forces affects the attitude measurements.

To remove these inertial forces, it is necessary to know them exactly. This Appendix describes the effects of a plane rotation on the accelerometer. We have to mention an important hypothesis. The plane does not drift on the air because of the centrifugal force. It means the plane is doing a perfect circle in the orthogonal plan of the rotation vector. Otherwise some other forces can appear: The Coriolis forces.

*In the earth referential system:*

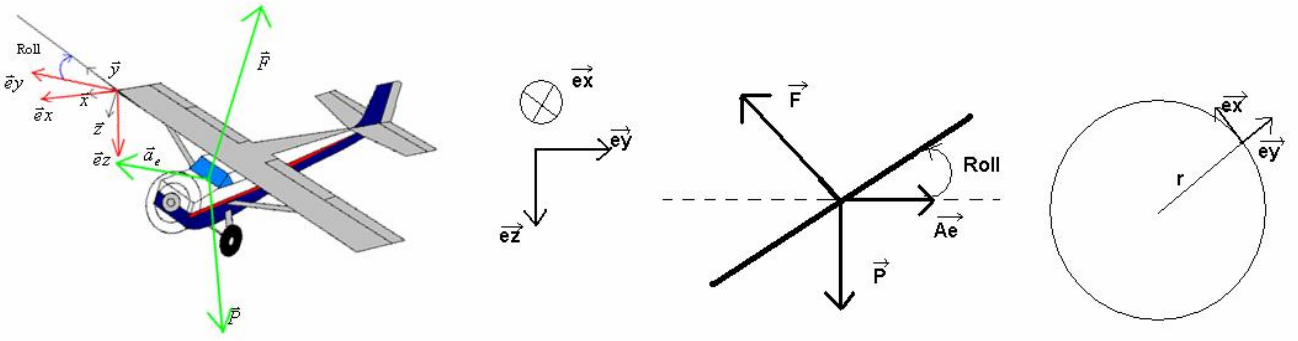


Figure J.1: Earth referential coordinate system. From the left: the system, the earth axis, model from behind, the plane trajectory

We consider the system plane {mass  $M_p$ } in the referential coordinate system established in Fig J.1. The plane is moving in the x direction and the z direction is down.

We apply the dynamic fundamental theorem to this system:

$$M_p \cdot \vec{a}_p = -M_p \cdot \vec{a}_e = \vec{F} + M_p \cdot \vec{g} \quad (\text{J.1})$$

We project this equation on the both  $(\vec{e}_y, \vec{e}_z)$  axis:

$$\begin{cases} \text{on } \vec{e}_z : & 0 = -F \cdot \cos \phi + M_p \cdot g \\ \text{on } \vec{e}_y : & M_p \cdot a_e = F \cdot \sin \phi = M_p \cdot \frac{V^2}{r} \end{cases} \quad (\text{J.2) and (J.3})$$

Including Eq.(J.2) in Eq.(J.3) yields

$$\begin{cases} a_e = g \cdot \tan \phi \\ r = \frac{V^2}{g \cdot \tan \phi} \end{cases} \quad (\text{J.4})$$

The inertial relations are established in the earth coordinate system. Then by geometric transformations, it is possible to establish these formulas in the gyro\_planes coordinate system.

***In the gyro\_plane referential coordinate system:***

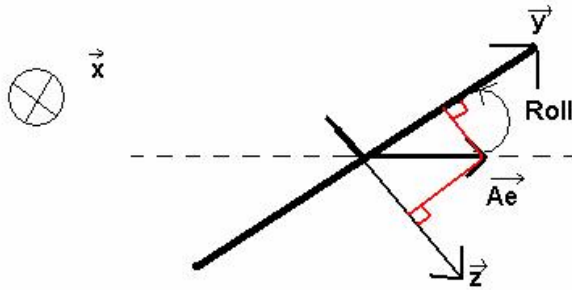


Figure J.2: the gyro\_plane coordinate system from behind the plane

In this referential coordinate system, described in Fig.J.2, we consider the projection of  $\vec{a}_e$  on the  $(\vec{y}, \vec{z})$  axis. Thus we know how the accelerometers measurements are affected when the plane is turning. The sensor will sense

$$\begin{cases} a_y = a_e \cdot \cos \phi = g \cdot \sin \phi \\ a_z = a_e \cdot \sin \phi = g \cdot \frac{\sin^2 \phi}{\cos \phi} \end{cases} \quad (\text{J.5})$$

Regarding on Eq.(J.5), we should have  $\phi \neq \pm \frac{\pi}{2}$ . This is not a problem because we consider a validity range for the angles; they should be smaller than 70 deg.

***NB: coherence of these relations:***

If the Roll is null, the plane is going along a straight line. Eq.(J.4) gives us an infinite curve radius which is coherent and Eq.(J.5) demonstrates that we don't have any inertial forces, which is coherent as well.

The sign of the Eq.(J.5) is also coherent with the Roll sign.

**NB: if the plane is drifting**

A relative velocity orthogonal to the rotation vector appears.

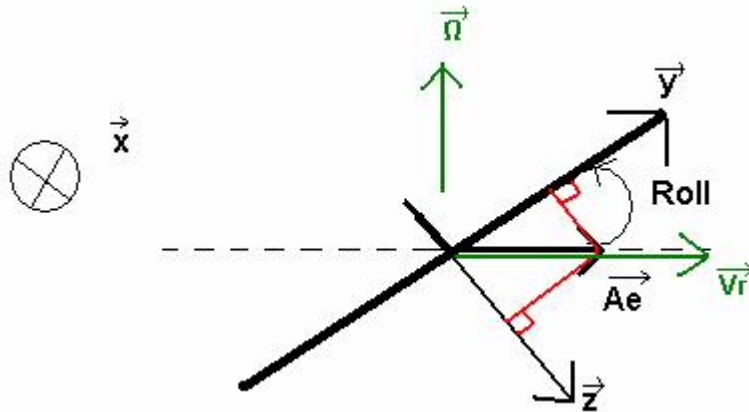


Figure J.3: the relative velocity  $V_r$  in the gyro\_plane coordinate system. The plane is seen from behind.

The Coriolis acceleration is orthogonal to the plan defined by  $\vec{\Omega}$  and  $\vec{V}_r$ ; it is defined by

$$\vec{a}_c = 2 \cdot \vec{\Omega} \times \vec{V}_r \quad (\text{J.6})$$

Thus  $\vec{a}_c$  is along the x axis, on the flight direction. The force resulting from this acceleration is

$$\vec{F}_c = -M_p \cdot \vec{a}_c \quad (\text{J.7})$$

Which is oriented against the plane flight directions.

We don't use this because it is really hard to estimate  $\vec{V}_r$ . It depends on many factors we don't know, like for example the size of the planes wings.

## REFERENCES

Klumpp, A.R.: Singularity-Free Extraction of a Quaternion from a Direction-Cosine Matrix, *Journal of Spacecraft*, Vol. 13, No 12, pp. 754-755, Dec. 1976.

Hamilton, W. R., *Elements of Quaternions*, Longmans, Green & Co., London, 1866.

Hofmann-Wellenhof, B., and Lichtenegger, H., *GPS Theory and Practice*, 5th ed. Springer-Verlag Wien New York, ISBN 3-211-83534-2, p.248 – 254, 1992.

Hughes, P.C., *Spacecraft Attitude Dynamics*, New York: John Wiley & Sons, 1986.

Teunissen, PJG, The invertible GPS ambiguity transformations, *Manuscripta geodaetica*, **20**, 489-497, 1995.

Brown, R.G. and Hwang, P.Y.C., *Introduction to random signals and applied Kalman filtering*, third edition, New York: John Wiley & Sons, ISBN 0-471-12839-2, p.214 – 219, p.289-293, 1997.

## RELATED LINKS

Schwab, A.L., Quaternions, finite rotation and Euler parameters, 2002, <http://tam.cornell.edu/~als93/quaternion.pdf>

Bonnet Torres, O., Filtrage de Kalman appliqué à la navigation inertielle, Onera-DCSD, 2003, <http://www.cert.fr/dcsd/THESES/obonnet/refdoc/kalmanNav.pdf>

Slater, D., Quaternions, 1991, <http://www.resonancepub.com/quaterni.htm>

Aron D. Kahn, The Autopilot project: <http://autopilot.sourceforge.net/index.html>.