Optimization of an Integrated GPS/INS Navigation System for Optimal Control of UAV Horizontal Motion

THESIS

MUHAMMAD KHALAF

SAINT PETERSBURG STATE UNIVERSITY OF AEROSPACE INSTRUMENTATION

June 2016

Saint Petersburg, Russian federation

Acknowledgments

“If we knew what it was we were doing,
It could not be called research, would it?“
Albert Einstein

I would like to express my deepest gratitude to my supervisor Prof. Dr. Alexander Nebylov for his guidance and insight. Additionally I would like to thank Dr. Vladimir Perliouk and Pilot Axel Schneider for their support during the experimental part of this thesis. A special thank goes to Prof. Dr.-Ing. Carsten Braun for his interest in this work and for agreeing to act as second examiner. The project could not have been done without these men.

I certify that the work submitted is my own and that any material derived or quoted from the published or unpublished work of other persons has been duly acknowledged

Student Full Name: MUHAMMAD KHALAF

Student Registration Number: 433629

Signed: ……………………………………………………

Table of Contents

Since the dawn of civilizations, man has looked into the sky searching for answers to his questions, some of these questions were "where am I?" And "How do I reach my desired destination?" the orientation on stars represents the oldest response to these questions.

The need of humanity to navigate according to precise and independent means, has led that over the past centuries various techniques and tools has been developed: Starting from Stars, passing through: Compass, Astrolabe, Sextant, Jacob's staff, Quadrant, Octant, Radar and down to GPS, and the goal is one: directing the man to his destination safely.

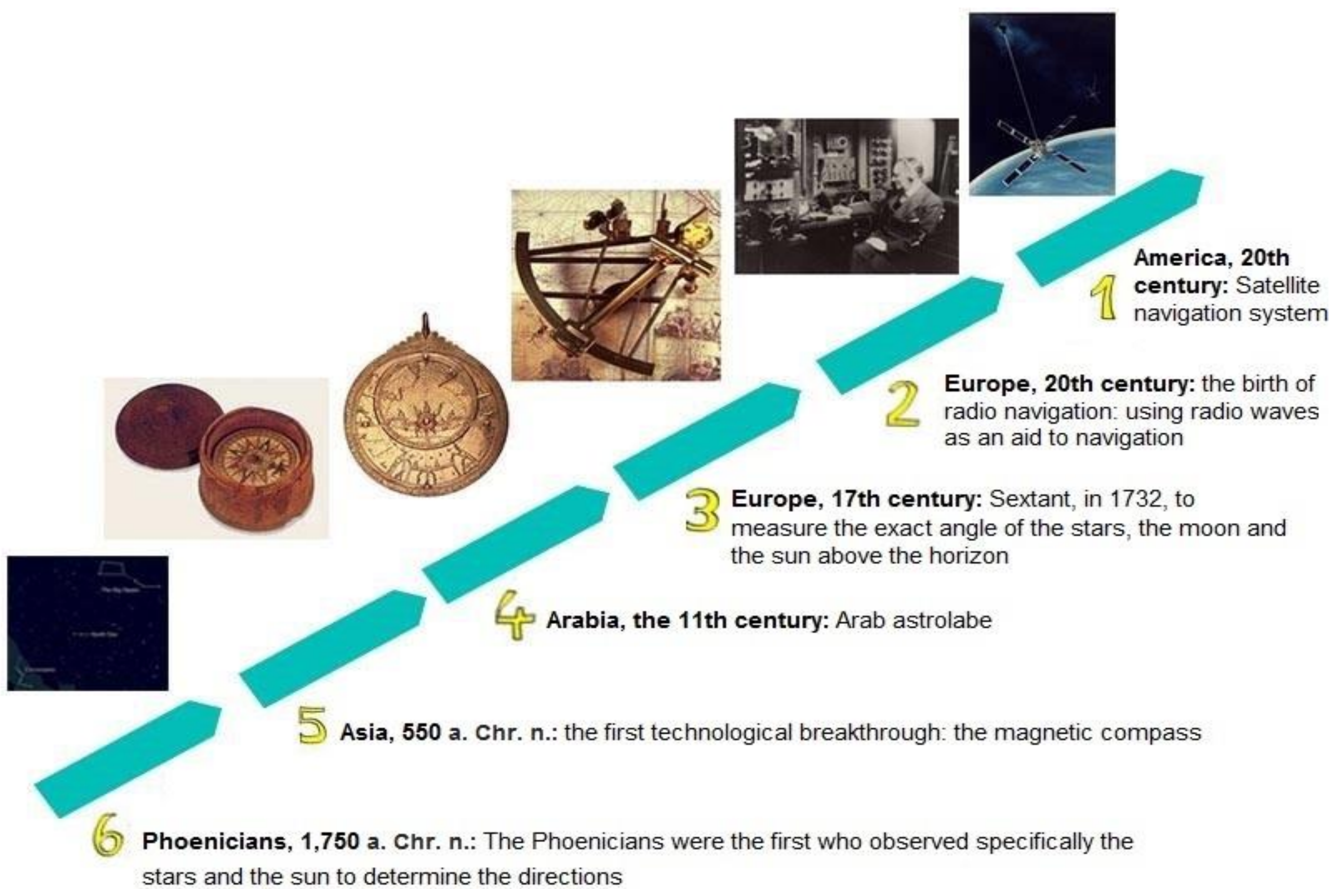


Fig. I Development chain of navigation tools

The aim of this thesis is to develop a precis, low cost and small navigation system for UAVs (Unmanned Air Vehicles). UAVs are nowadays used for a wide range of applications, one of this applications is related to monitoring missions. For this application it is essential to control the altitude of the UAV, for which unacceptably large deviations of this important navigation parameter from a nominal value can result an emergency situation and possibly a fatal accident.

The errors of measuring equipment lead to reduction of measurement accuracy, integration of several sensors within one system is one technique to overcome this problem. In this case estimation of errors becomes more complicated when there are a variety of errors components. The chosen method to control UAVs altitude is based on combination between accelerometer and GPS, the research objective deals with the estimation and minimization of measuring errors in this integrated navigation system, in order to design an automatic control system for UAVs horizontal motion, where the maximum error of measuring system should be not more than 3 till 5 meter.

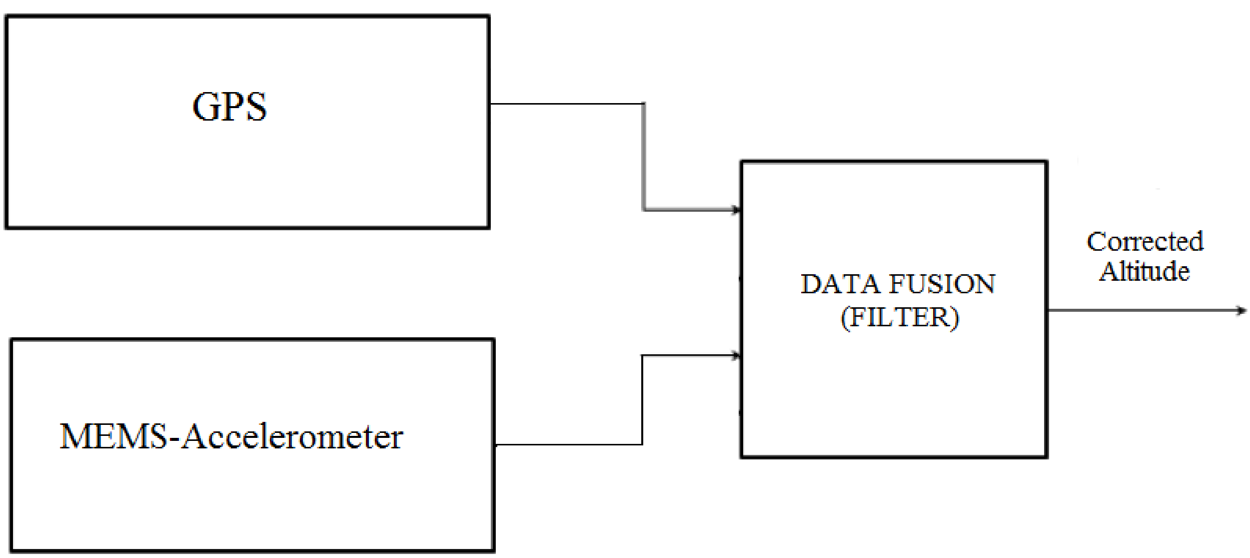


Fig. II Sensors data fusion

Chapter 1

INTRODUCTION

Objectives:

- Definitions, History and Design Technologies of UAV.

Definition

What is navigation? According to the Oxford Dictionary, navigation is defined as "the process or activity of accurately ascertaining one's position and planning and following a route". This definition can be divided into two concepts:

The first is the determination of the position, altitude and velocity of a moving body. The second is the planning and maintenance of a course from one point to another avoiding collisions and obstacles (optimal trajectory), this thesis focuses only on the first concept.

Fig. 1.1 the differences among the concepts of navigation, guidance, and control.

What is UAV?
The definition, according to US Department of Defense (DoD):
"A powered vehicle that does not carry a human operator, can be operated autonomously or remotely, can be expendable or recoverable, and can carry a lethal or nonlethal payload. Ballistic or semi-ballistic vehicles, cruise missiles, artillery projectiles, torpedoes, mines, satellites, and unattended sensors (with no form of propulsion) are not considered unmanned vehicles. Unmanned vehicles are the primary component of unmanned systems". U.S. Department of Defense.

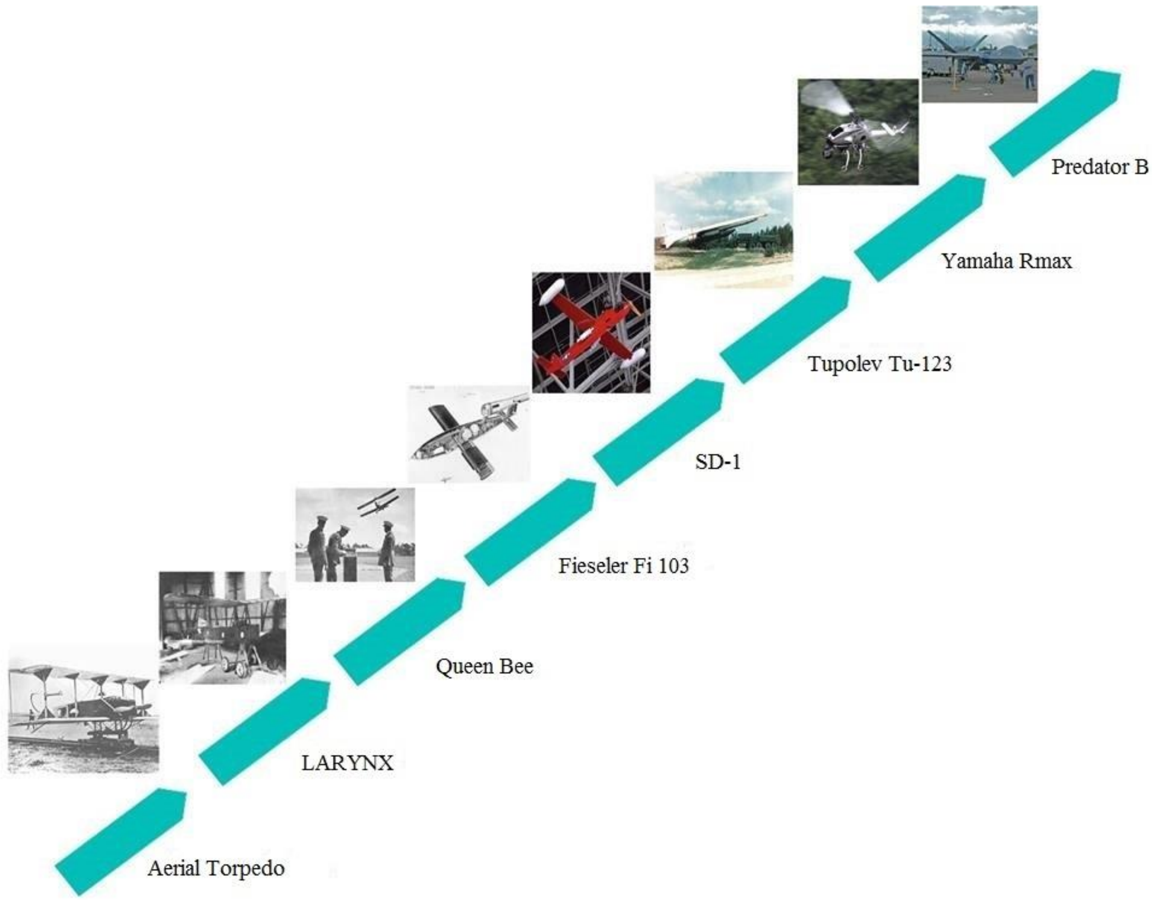
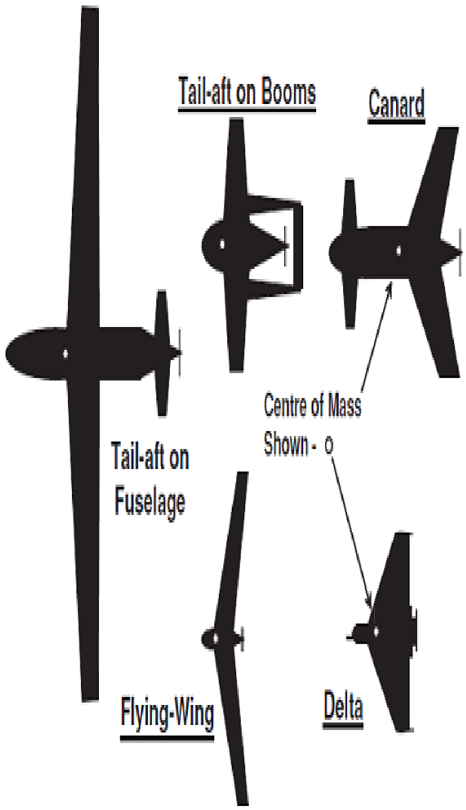


Fig. 1.2 The chain of UAVs developments from it beginning in 1916 to the present

- **UAVs Design:**

According to their take-off and landing methods, UAVs configurations can be group into three types:

a) horizontal take-off and landing,
b) vertical take-off and landing,
c) Hybrids.



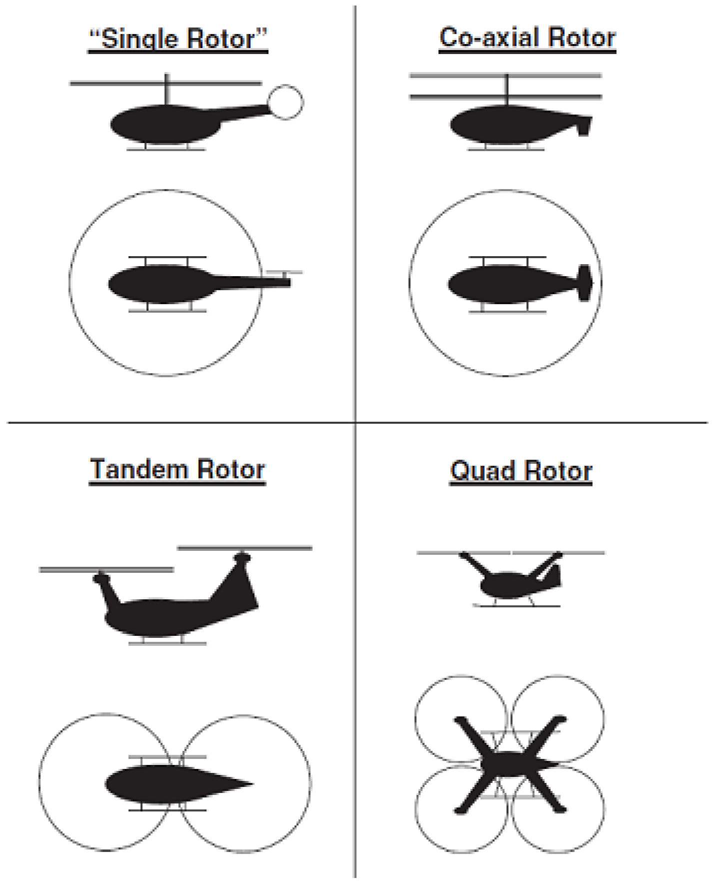


Fig. 1.3 HTOL aircraft configurations **Fig. 1.4** VTOL aircraft Configurations [1]

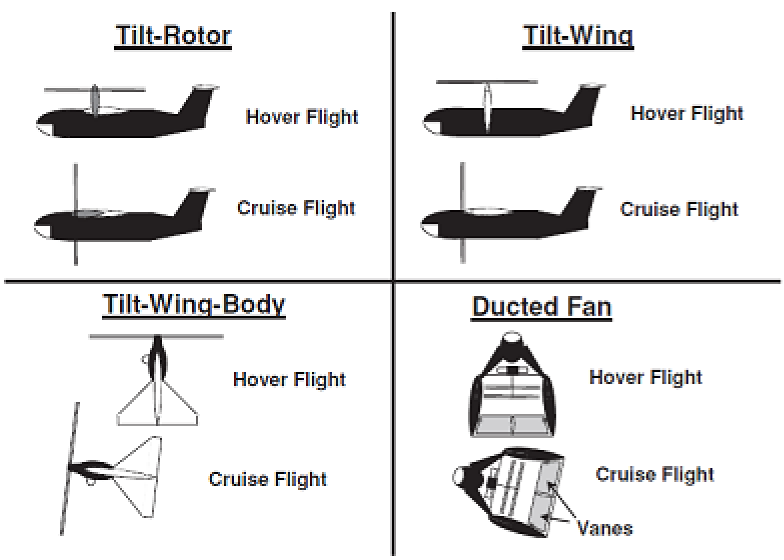
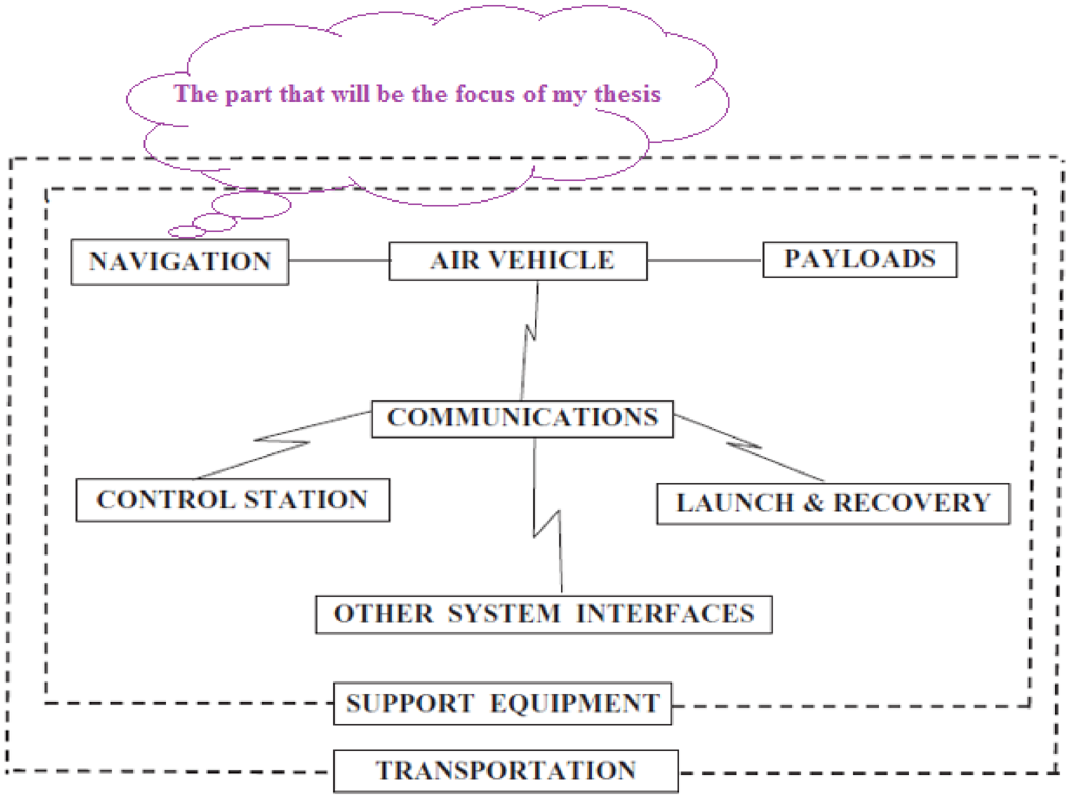


Fig. 1.5 Hybrid aircraft configurations [1]

- **System Composition**

The UAV is just one part of an entire system. The entire system comprising: A control station, the aircraft, the system of communication, and Support equipment.



[2]

Fig. 1.6 Desert Hawk system [3]

Chapter 2

REFERENCE FRAMES

Objectives:

- Description of The main coordinate frames used in navigation.

A reference frame is a coordinate system or set of axes within which the position of objects can be measured. In this section, the definition of important reference frames for the along this thesis used sensors are discussed.

- **Coordinate Frames**

➢ ECEF-Earth Centered Earth Fixed Frame.

➢ Body frame

2.1.1 Earth-centered, Earth-fixed reference frame

The Earth-centered, Earth-fixed reference frame F_E is fixed with respect to the Earth and follows the rotation of the Earth. Its origin O is located at the center of mass of the Earth, X-axis crosses the Greenwich meridian, the Z-axis is directed along the spin-axis of the Earth and the Y-axis is perpendicular to the previous, passing through the equatorial plane. The ECEF frame is used in the GPS system.

2.1.2 Body frame

The Body frame F_B is a vehicle-fixed reference frame. This is the most common reference frame used by the inertial sensors. Its origin is at the aircraft's center of mass. The X-axis points forward, along the nose of the airplane, in the symmetry plane of the aircraft; the Z-axis also lies in the symmetry plane but points downwards; and finally, the Y-axis is in the direction of the right-wing forming a right-handed orthogonal axis-system.

Chapter 3

CHOICE FOR A NAVIGATION METHOD

Objective:

- Which navigation method should be used to control the UAV's altitude?

The aim of this chapter is to select an appropriate method for determination of UAVs altitude. First available methods for determining the position will be described, next an appropriate method, according to several criteria, will be chosen.

- **Possible Position Determination Methods**

3.1 Global Positioning System

The US Department of Defense issued in 1973 a mandate to set up a satellite-based system for determination of position and velocity of any object as well as the time. The detailed definition from 1985 is:

"The Global Positioning System (GPS) is an all-weather, space- based navigation system under development by the Department of Defense (DoD) to satisfy the requirements for the military forces to accurately determine their position, velocity, and time in a common reference system, anywhere on or near the Earth on a continuous basis".

The position is determined via distance measurement method: a constellation of 24 satellites orbiting the Earth transmit encoded radio frequency signals, These transmitted signals move at light speed (c=300.000km/s) so now, everyone on Earth can receive these GPS signals and determine the duration "T" by calculates the difference between the transmission time and the arrival time. Distance "S" is then calculated by: $S = T \times c$. This requires, that the clock times of the satellites coincide with the clock time of the receiver. This is not generally fulfilled. Therefore, the measured signal propagation time contains a systematic error and the resulting calculated distance has a corresponding error component. The measured distances are therefore called pseudo-ranges. For the three-dimensional position determination, three distance measurements are required. If pseudo-ranges are used, so an additional measurement is necessary to compensate for the existing in all pseudo-ranges unknown receiver clock errors and to be able to calculate the user position.

GPS Structure:

The GPS consists of three segments:

1. Space Segment: The Space Segment of GPS system consists of all the available satellites
2. Control Segment: The GPS control segment consists of the existing on ground control stations.
3. User Segment: The user segment consists, as the name implies, from all end-users of GPS worldwide

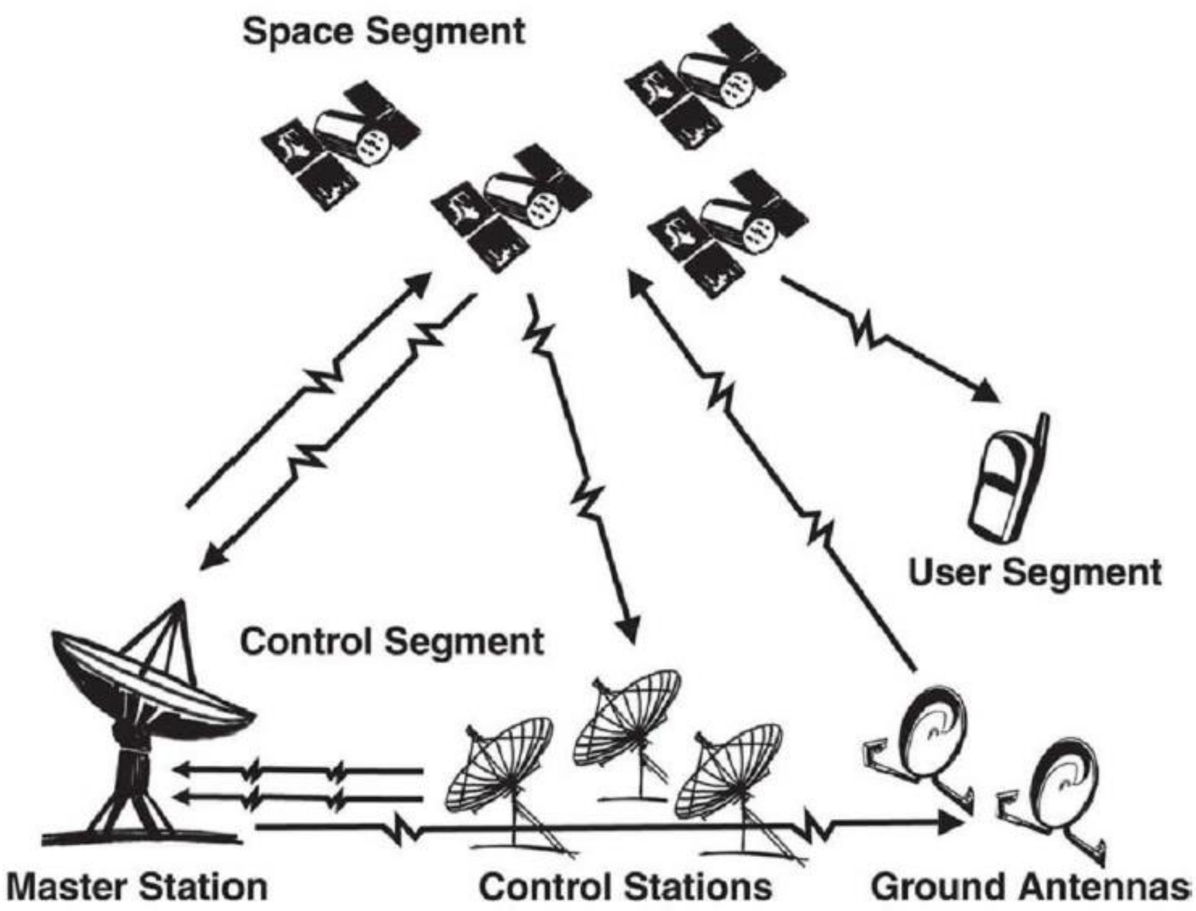


Fig. 3.1: A schematic overview of the GNSS architecture [4]

The highly accurate atomic clocks, being the heart of GPS satellites produce the fundamental frequency of 10.23 MHz. Coherently derived from this fundamental frequency are mainly two signals, the L1 and the L2 carrier waves generated by multiplying the fundamental frequency by 154 and 120, respectively, yielding

L1 = 1575.42MHz,
L2 = 1227.60MHz.

These dual frequencies are essential for eliminating the major source of error, i.e., the ionospheric refraction

The L1 frequency, contains a Coarse/Acquisition (C/A) ranging code and navigation data message, the ranging codes are to determine at what time the received signals were transmitted, while the data message consists of timing parameters and information about the satellite orbits. This code is omitted from L2 to deny full system accuracy to nonmilitary users, only The L1 band carrier is available for civil use .The second code is the precision code (P-Code), which is only for authorized users such as the U.S. military. The P-code is modulated on both carriers L1 and L2. Next, the receiver equipment converts the signals from the satellite into a position, velocity and time. [5]

GPS advantages and disadvantages are:

+ Long-time high accuracy level.

– The loss of signal due to blockage, multipath or jamming

3.2 Inertial Navigation System

Inertial navigation refers to the navigation with inertial sensors, inertial sensors are gyroscopes and accelerometers which respectively measure the rotation rates and the acceleration.

Principle of a gyroscope

Gyroscopes are based on Coriolis Effect: An object to which we apply a rotation and a translation is submitted to an acceleration due to the combination of these two movements. We can express this acceleration by the cross product of the angular speed $\vec{\Omega}$ of rotation and the velocity of translation $\vec{v}$.

The apparent force on the object is called Coriolis force: $\vec{F}_{Coriolis} = 2m_{object}(\vec{v}\times\vec{\Omega})$. By measuring the effect of this force on a proof mass which is moving at a known velocity, the rotational speed of the object the sensor is placed on, can be determined.

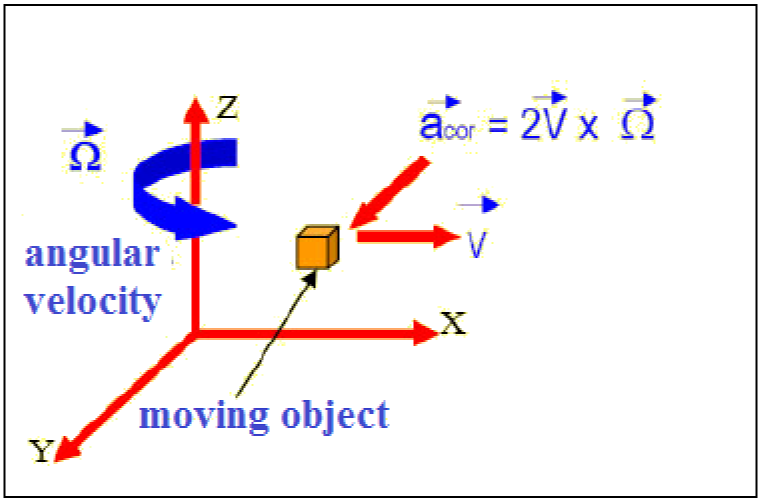


Fig. 3.2 working principle of gyroscope [6]

Principle of Accelerometer

The accelerometers are devices based on Newton's second law:
a proof mass is fixed to the sensor platform by a mechanical spring-damping system, an inertial force will deflect the proof mass; by measuring the mass displacement it is possible to calculate the acceleration using Newton's second law

$$\sum F = m\ddot{x}$$

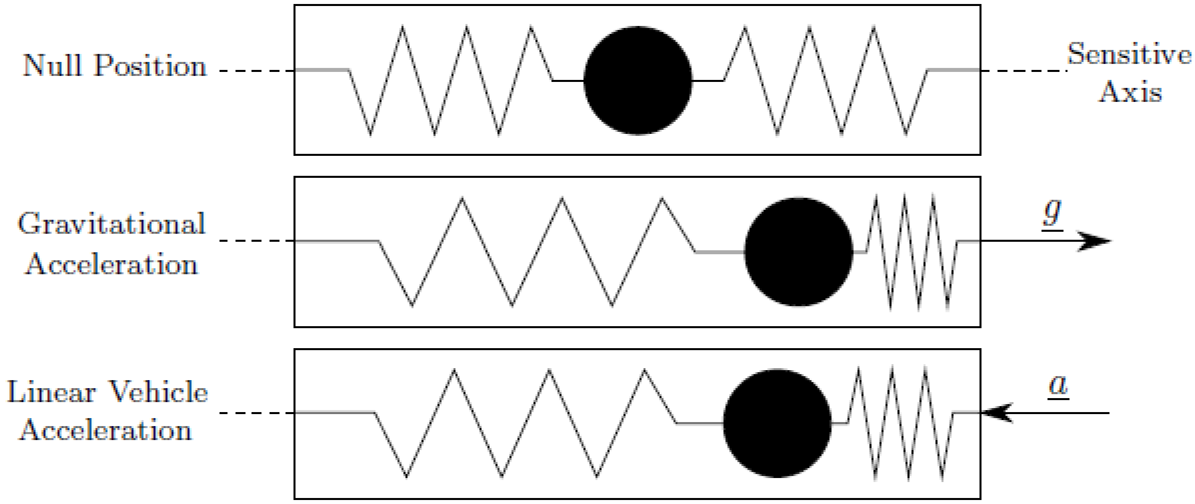


Fig. 3.3 working principle of accelerometer [7]

The output of an accelerometer is a measurement of the difference between the true vehicle acceleration “a” and the gravitational acceleration “g”, for the vertical acceleration can then be written:

$$\ddot{h} = a - g$$

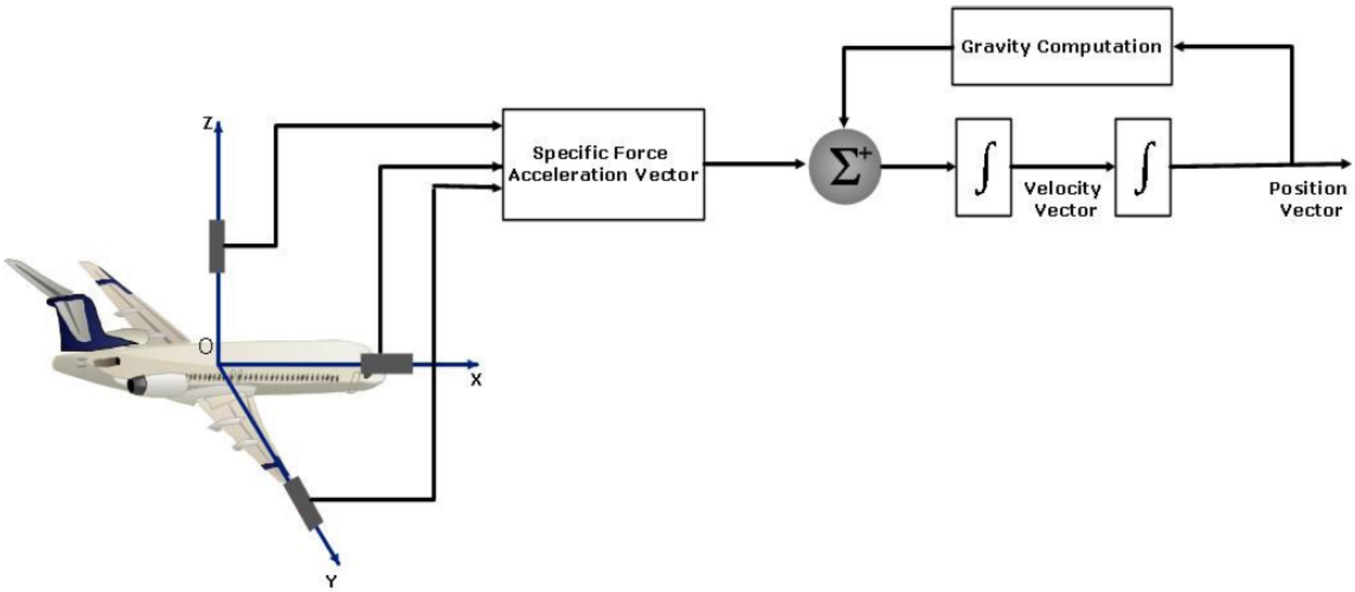


Fig. 3.4 The concept of Accelerometer [8]

INS advantages and disadvantages are:

+ It is self-contained, it does not need external references
+ short-time high accuracy level.

- The sensor errors (such as biases, and noise) are also integrated, and summed over time, so the system becomes less accurate over time.
- If gravity is accounted for in the accelerometer readings, then the system thinks it is moving when in fact it is not.

3.3 Integrated Navigation System

A GPS receiver can be combined with several other sensors to obtain more accurate results. For example the combination of a GPS sensor and an INS.

- GPS/INS

Based on their complementary properties, these systems are usually combined using some estimation technique (like Kalman filter) in order to obtain higher accuracy.

Advantages and Disadvantages of this system can be summarized as follows:

+ The integration of GPS/INS will provide much better accuracy than GPS or INS alone.
- For every integration a filter has to be designed. The use of this filter will also require computing power.

3.4 Map- Matching Positioning

In this approach, the moving platform uses its sensors to perceive its local environment, and this perception is then compared to a map previously stored in its memory. If a match is found, then the vehicle can calculate its position, altitude and orientation in this specific environment. Cameras and laser are examples of sensors that can be used with this type of positioning. The stored map of the environment can be an already available model, or it can be constructed from prior sensor data. [9]

Advantages and Disadvantages of this system are as follows:

+ Very high accuracy can be achieved.
+ The cameras can also be used for obstacles avoidance.
- The need to process all images.
- Weather conditions may influence or disable the system.
- Expensive

- **Criteria for select a navigation method:**

There are many criteria, which the positioning method should meet. These criteria are:

<u>Mission</u>

The nature of the UAV mission affects the requirement for onboard sensors, for monitoring missions, sensors with absolute positioning and high update rate are desirable.

<u>Accuracy</u>

Based on the design brief an accuracy of max. Error of 3-5m should be achieved.

<u>Weight and size</u>

Because there is little space in UAVs a small positioning system is preferred.

<u>Price</u>

The expenses should be kept reasonable in order to develop a low cost UAV

After considering all alternatives it was decided the combination of the GPS with the vertical channel of INS. This is the best choice in terms of accuracy for a reasonable price and matches modern concepts of complex measuring system construction.

In the next chapter, an integration architecture will be chosen and an optimal filter for data fusion will be designed.

Chapter 4

SENSORS INTEGRATION

Objectives:

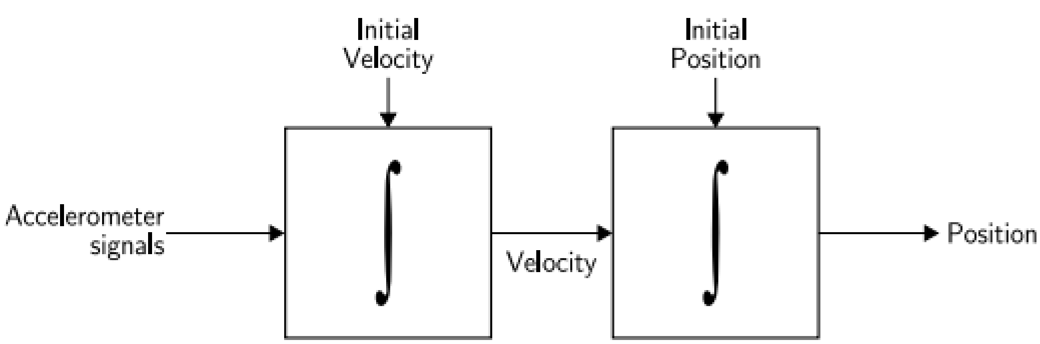
- Description of errors sources and models of each used sensors.
- Development and Optimization of integrated navigation system by choosing a filter capable to minimize the total errors.
- Analyze the potential accuracy of the developed navigation systems and calculate the root mean square error of the system to obtain the desired accuracy.

In practice, the accelerometers and GPS are affected by time-varying noises. These characteristics have to be modeled in order to design a suitable algorithm for data fusion.

4.1.1 The primary sources of error for accelerometer and GPS include:

• Bias, An accelerometer's bias is the amount of acceleration that is measured in the absence of any true acceleration.

• Integration errors:



The problem of inertial sensors is that the input signals are time integrated, as can be seen in the above Figure, a constant bias of ϵ therefore causes an error which grows quadratically for an accelerometer, i.e. $x_f(t) = \frac{1}{2}\epsilon t^2$. The error due to the accumulation of small bias over time is called bias drift. [10]

• Non-orthogonality errors, which result from a misalignment of the sensor axes caused by imperfections in the construction of the sensor assembly.

• Random noise, which is an additional signal resulting from the sensor itself or other electronic equipment that interfere with the output signal being measured. For example mechanical noise caused by the Brownian motion of the air.

Also the carrier signal of SNS is affected by noises such as:

• Satellite clock error: although the atomic clocks available in the satellites are very accurate, errors can be large enough to require correction due to the difficult synchronization between all the satellites.

• Receiver clock error: this is the largest source of error because usually they are inexpensive quartz crystal oscillators which are much less accurate than the satellite clocks.

• Ionospheric delay: Since there are free electron in the Ionosphere (around 40_1000 km), the GPS signal does not travel at the speed of light while in transit in this region. The delay is proportional to the number of free electrons encountered. The user can compensate for these errors by observing the time of arrival of two signals with different frequencies.

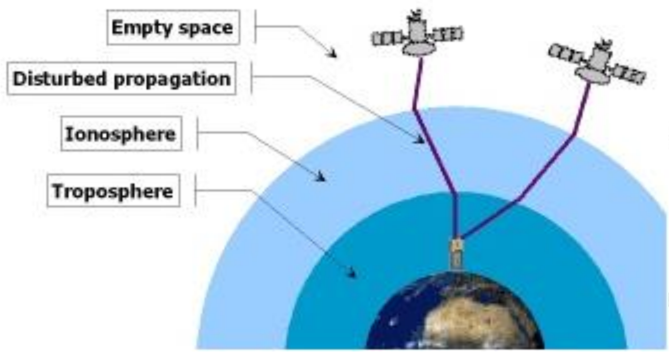
• Tropospheric delay: this low layer of the atmosphere (0~50 km) is rich in water vapor that refracts the GPS signal leading to a speed reduction (phase delay). It is also related to the satellite elevation.

• Multipath errors: besides the direct line of sight of the GPS signal, other reflected signals might also arrive at the receiver distorting the original one. A receiver with a cutoff angle usually helps to reduce this source of error.

• Satellite ephemeris errors: differences in the theoretical (estimated) and real satellite orbits results in ephemeris errors that compromise accuracy of the GPS receiver.

• Measurement error: the antenna, amplifiers and cables are not perfect and may introduce some noise. Furthermore, the signal quantization noise is commonly present.

[11]



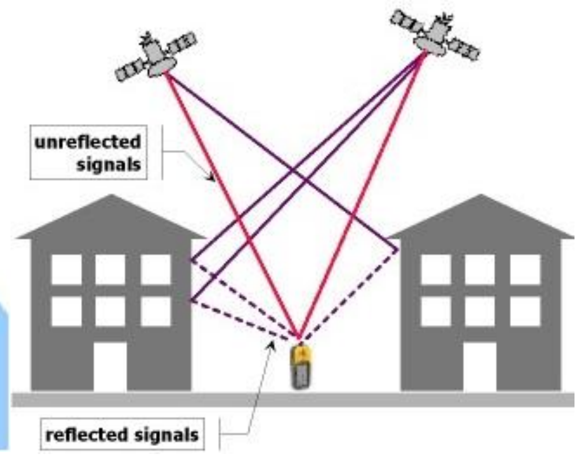


Fig. 4.1: Effect of atmosphere on GPS signals [12]

Fig. 4.2: Multipath[12]

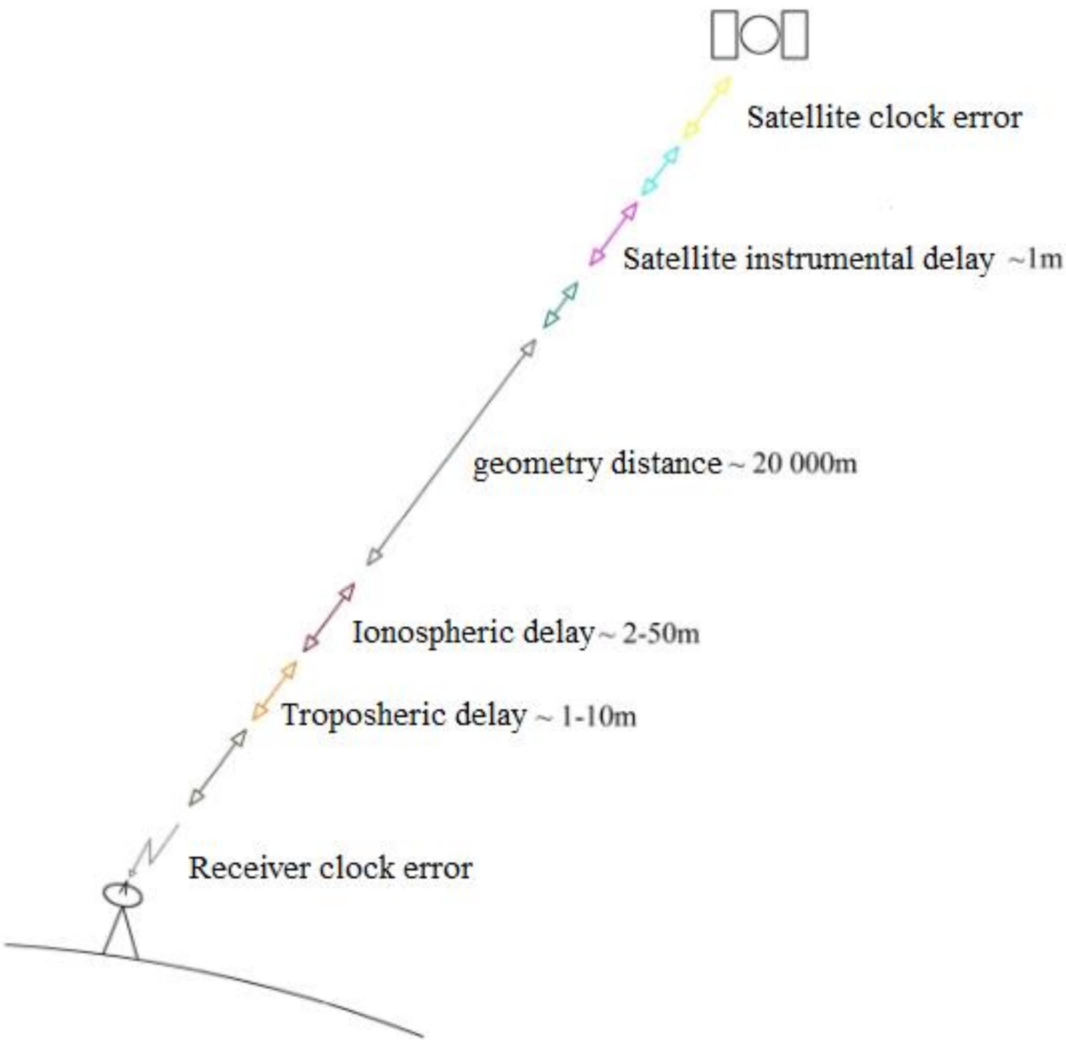


Fig. 4.3 Satellite's errors and their orders

4.1.2 Modeling of Errors:

To study the noise influence of a sensor, modeling techniques have to be used. The power spectral density (PSD) is the most common tool used to analyze data.

Characteristic PSD slopes of noise can be found in figures 4.4 and 4.6 for GPS and accelerometer respectively.

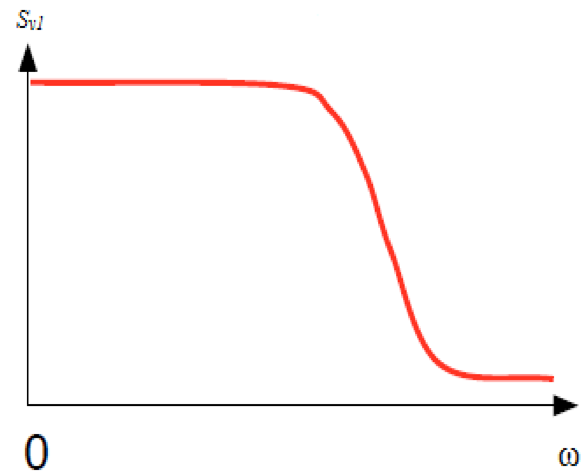


Fig. 4.4: Typical slope of PSD versus frequency for GPS data.

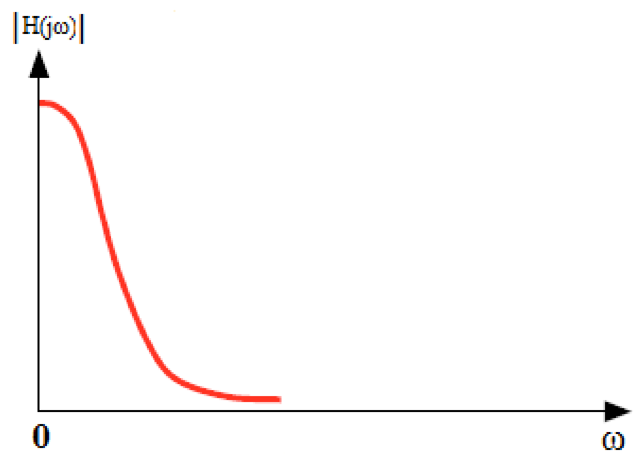


Fig. 4.5 Filter 1 to depress GPS's errors

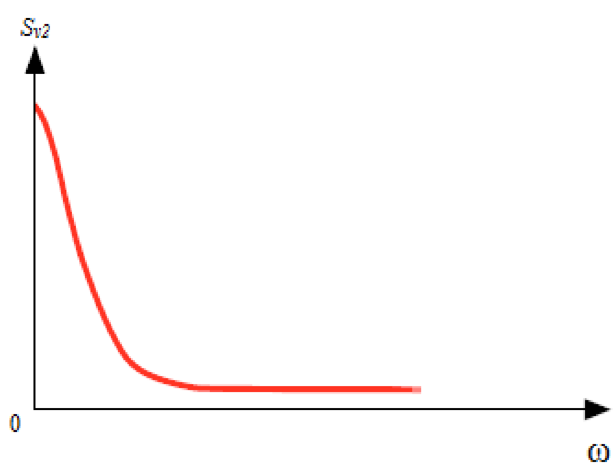


Fig. 4.6: Typical slope of PSD versus frequency for Accelerometer data.

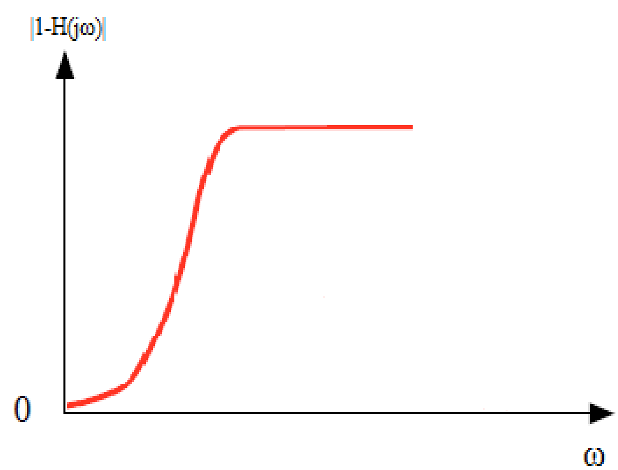


Fig. 4.7 Filter 2 to depress INS's errors

The errors in GPS are mainly of high frequency character, in order to depress them only the lowest-frequency components are taken from GPS. And the errors in accelerometers usually appear at the low-frequency part of the sensor output. In order to depress them, only the highest-frequency components are taken from the accelerometers. In this way, the spectral components of the useful signals least deformed by noise are taken from each of the sensors.

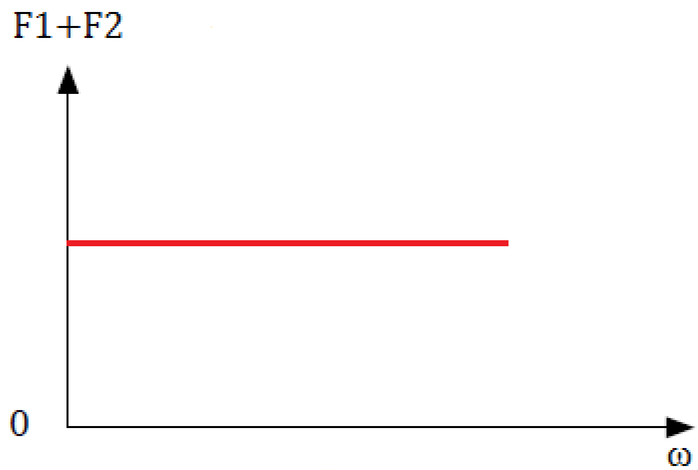


Fig. 4.8 normalization of the output signal after filtering (Filter1+Filter2)

The two sensors (GPS & Accelerometer) have different error properties: the accelerometer's errors grow without bound over time making the accelerometers reading useless over time, but unlike the accelerometers, the errors in GPS are bounded and stable. For accelerometer error δh, it is possible to indicate the maximum possible value of its dispersion second derivative and if I assume that the GPS errors contain only white-noise components v_{gps}, with this priori information the aim of the accuracy analysis in this research is to evaluate the maximum value of corresponding error components after their integration.

4.2 GPS/Accelerometer integration architecture

The generalized configuration of the integrated measuring system can be presented with the block-diagram in Figure 4.9 where *Si*, for $i = 1,2$, are the available onboard primary sensors (GPS & Accelerometer) whose output signals consists from a useful component $x_i(t)$ corresponding to the true value of the measured parameter (altitude) + a different values of noise $v_i(t)$.

$W_{si}(s)$ are the transfer functions of sensors, it is for GPS $W(s_1)$ equal to one, and for accelerometer $W(s_2)$ equal to S^2 , $H_{ci}(s)$ are transfer functions of computer, where the sensor signals should be filtered before their summation.

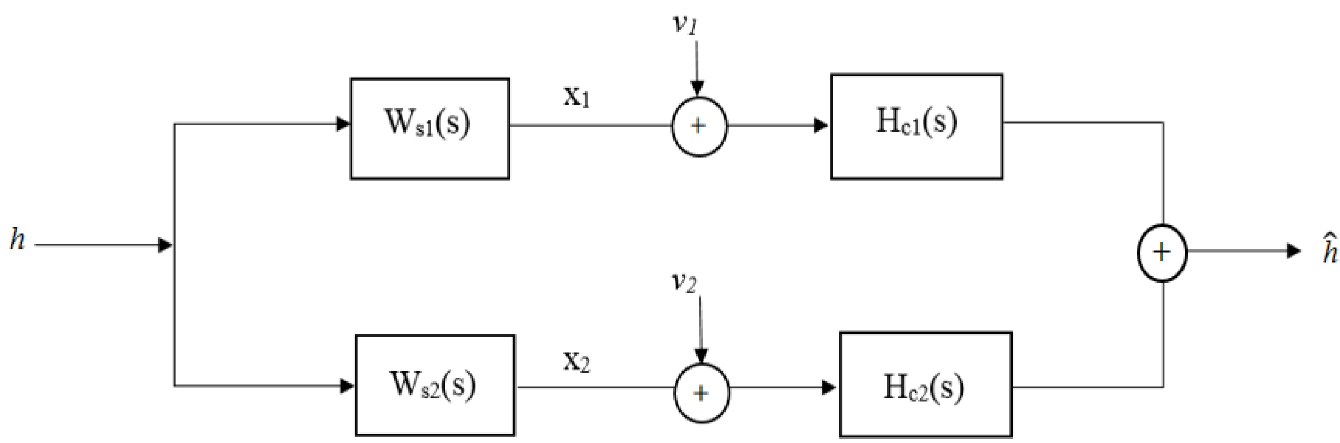


Fig. 4.9 Block diagram for the integrated measuring system.
Goal: finding the transfer functions of the filters $H_{c1}(s)$ & $H_{c2}(s)$

The filter is a mathematical algorithm providing an optimized output for a noisy input signal by using a predictive model. It estimates how the answer should be without the noise created by the used sensor.

The motivation for the research below is to design an optimal filtration based on the sensors errors equations and to examine the potential accuracy of the developed system. An accuracy analysis is necessary in order to find such a set of sensors of their use will provide the prescribed accuracy.

- **First Integration Architecture:**

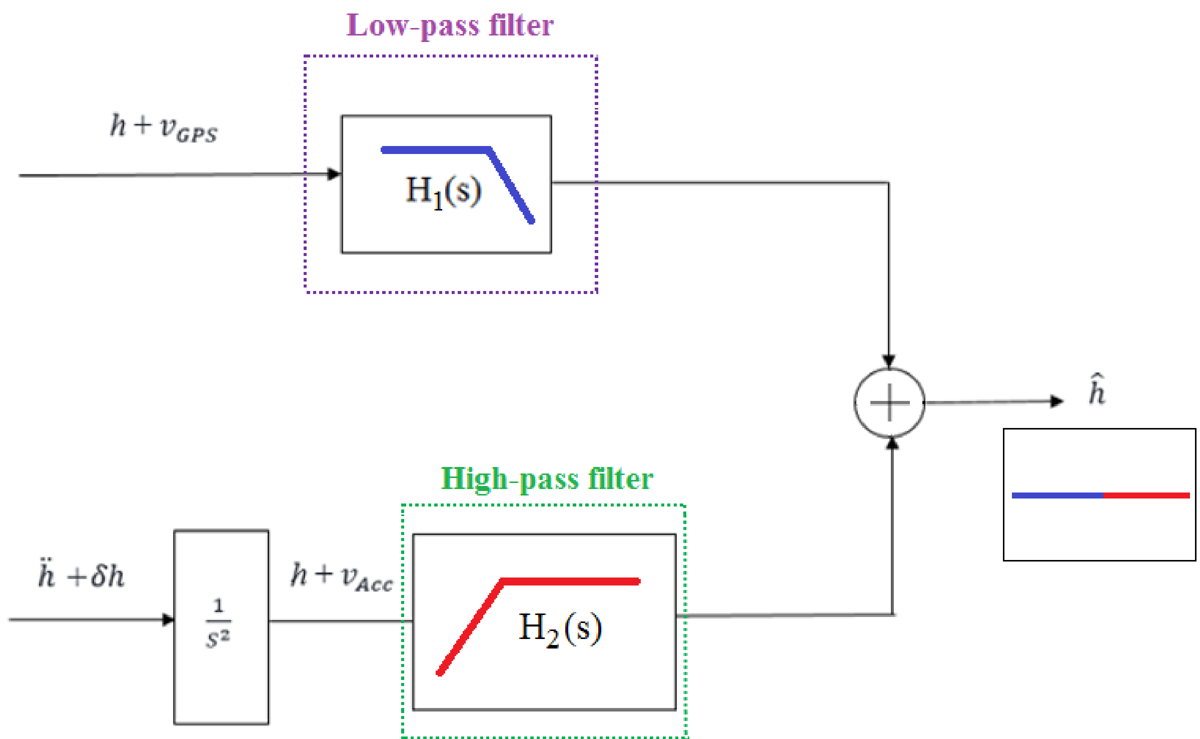


Fig. 4.10 Complementary filter configuration

The complementary filter is shown in Fig. 4.10, where v_{GPS} and v_{acc} are noisy measurements of altitude and $\hat{h}$ is the estimate of h produced by the filter.

The noise in GPS is mostly high frequency, and the noise in accelerometer is mostly low frequency. if $H_1(s)$ is a low pass filter to filter out the high-frequency noise in GPS, than $[1 - H_1(s)]$ is the complement, i.e., a high pass filter which filters out the low-frequency noise in accelerometer.

- **Second Integration Architecture:**

The accelerometer readings can be presented as a sum of two components: $h^{acc} = \delta h + \ddot{h}$ where δh is a accelerometer error, and $\ddot{h}$ is a vertical acceleration caused by any vertical motion of the aircraft.
The data from the GPS are $h^{GPS} = h + v_{gps}$, where h is the unknown altitude and v_{gps} is the error in the GPS measurements.

The differential measurements may be presented as follows:
$\hat{v} = (h + v_{gps}) - (h + v_{acc}) = v_{gps} - v_{acc}$
And $\hat{h}$ is the final estimated altitude.

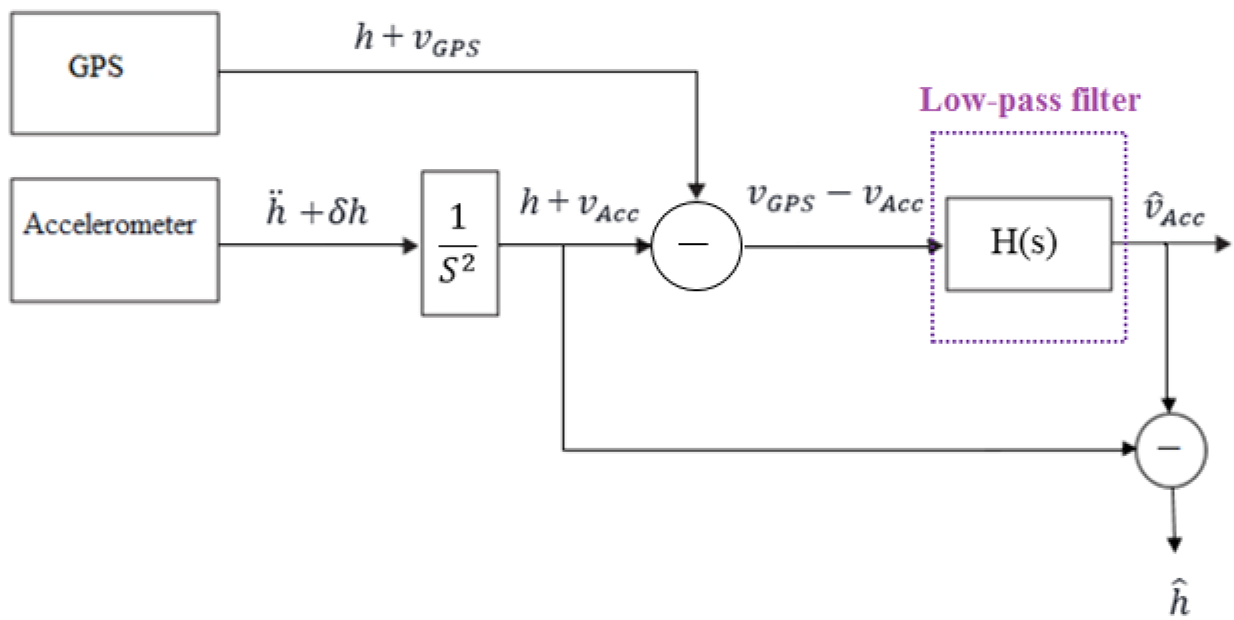


Fig. 4.11 Scheme of compensation
Data fusion synthesis: the output signal of GPS is injected into Accelerometer output signal

δh is the error in accelerometer having dimension of acceleration and v_{acc} is the error in accelerometer after integration (having dimension of position).

$\hat{v}_{acc}$: Estimation of accelerometer error.

Using a low-pass filter H(s) ensures that the powerful components of GPS error will not pass, in this case a good estimation of accelerometer's error will be taken.

4.3 Filtering System:

The main task in the synthesis of the integrated measuring system, consists of choosing the transfer functions $H_i(s)$, i =1, 2. of the filters and the transfer functions of complete channels within the system.

The order of the required transfer functions should be no less than a power of the limited derivative of error of the used sensors, as the accelerometer measures the second derivative instead of the signal itself, the order n should be also no less than two.

In order to keep the number of filter design parameters small, we have chosen a simple second order transfer function:

$$W(s) = \frac{K_2\ (1+\tau s)}{S^2}$$

It depends on only two parameters: K_2 is the gain coefficient and τ is the time constant.

There are two types of error feedback mechanisms used in integrated measuring system: open-loop and closed-loop. I have chosen the closed-loop configuration because without errors feedback to correct accelerometer measurements before their integration, the errors in accelerometer will grow larger with time, making the linearity assumption invalid. [13]

It follows for the unit Feedback configuration (the closed-loop channel) the following transfer function:

$$H_1(s) = \frac{W(s)}{1+W(s)} = \frac{K_2(1+\tau s)}{K_2+K_2\tau s+S^2}$$

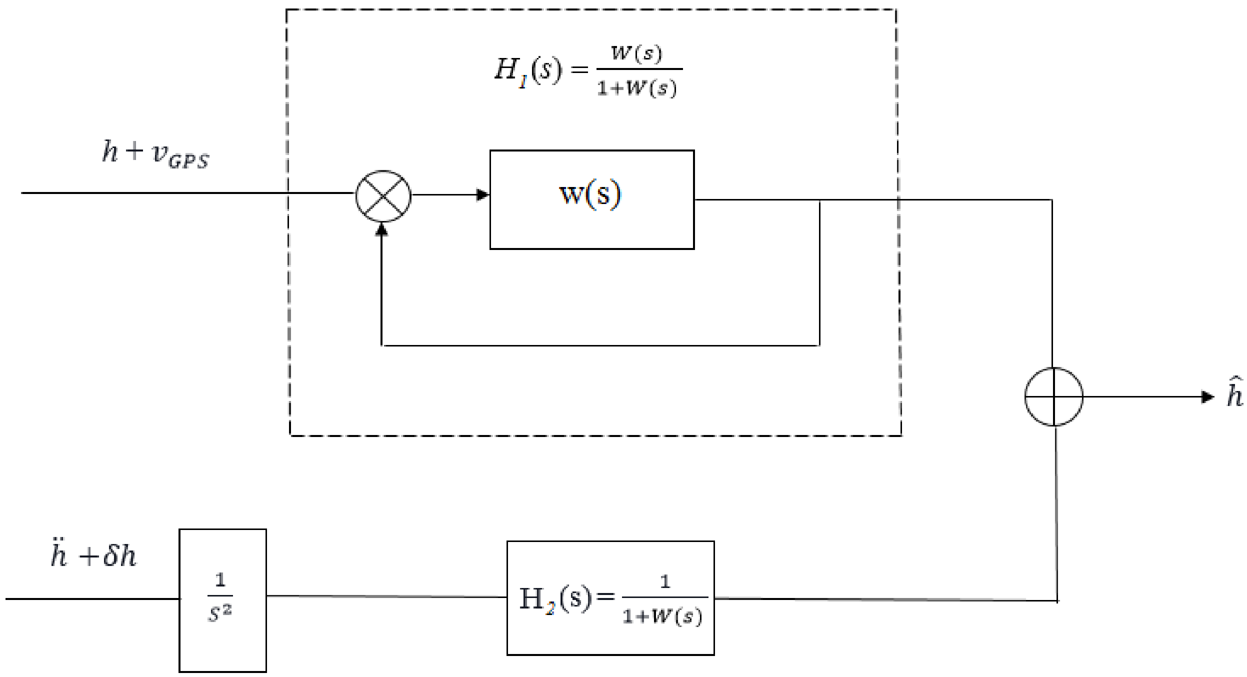


Fig. 4.12: feedback scheme.

In order to get an invariant system, the resulting transfer function should be equal to One: $H_1(s) + H_2(s) = 1$

It follows for the closed-loop channel such transfer Function:

$$H_1(s) = \frac{K_2(1+\tau s)}{K_2+K_2\tau s+s^2} \longrightarrow H_2(s) = 1 - H_1(s) = \frac{s^2}{K_2+K_2\tau s+s^2}$$

It follows the transfer functions of the whole system will be:

$$H_{c1}(s) = \frac{K_2(1+\tau s)}{K_2+K_2\tau s+s^2}.$$

$$H_{c2}(s) = \frac{1}{s^2} \times \frac{s^2}{K_2+K_2\tau s+s^2} = \frac{1}{K_2+K_2\tau s+s^2}$$

To be investigated function depends on 2 parameters K_2 & τ , which we need to optimize.

The chosen criterion of optimality will based on minimization the upper bound of total error of measurements.

Minimization the upper bound of total error of measurements:

Useful signal $h(t)$ and noises $v_i(t)$, are the centered components of the system input, the resultant error is represented as the sum of two components $e = e_G + e_A$. Where e_G the error from GPS and e_A is the error from accelerometer.

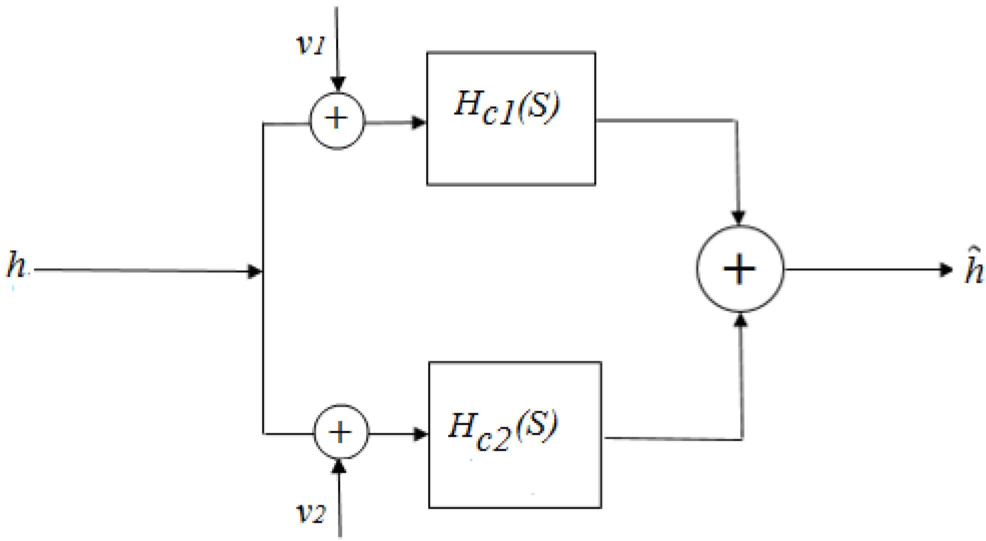


Fig. 4.13: overview of the designed integrated system

My objective here is to find the realizable *h*(t) by minimize the measurement error root mean square :

$$\overline{e^2(t)} = De = \overline{[h(t) - \hat{h}(t)]^2} \longrightarrow min \qquad [14]$$

Estimation of the error variance (dispersion) *De*:

The dispersion of resultant error is equal to the dispersion of GPS error added to the dispersion of accelerometer error: $De = D_{eG} + D_{eA}$

The dispersion is connected to the spectral density by integral relations:

$$D_{eG} = \frac{1}{\pi}\int_0^\infty |H_1(j\omega)|^2 \, S_G(\omega)d\omega \qquad [15]$$

$$D_{eA} = \frac{1}{\pi}\int_0^\infty |H_2(j\omega)|^2 \, S_A(\omega)d\omega \qquad [15]$$

$S_G(\omega)$ is the spectral density of noise in GPS and will be considered as constant (white noise) $S_G(\omega) = S_G$, Then the dispersion of error from GPS is:

$D_{eG}=S_G\,\Delta f_{eq}.$

Where: $\Delta f_{eq}= \frac{1}{\pi}\int_0^{+\infty} |H_1(j\omega)|^2\, d\omega = \frac{1}{\pi}\int_0^{+\infty} \frac{K_2^2+(K_2\tau\omega)^2}{(K_2-\omega^2)^2+K_2^2\tau^2\omega^2} = \frac{1+K_2\tau^2}{2\tau}$ [16]

→

$De = D_{eA} + D_{eG} = D_{eA} + S_G(\frac{1+K_2\tau^2}{2\tau})$

As the $S_A(\omega)$ is unknown (only the errors 2nd derivative dispersion is known but its spectrum is unknown) , so estimation of accelerometer's error becomes more complicated because the dispersion D_{eA} cannot be directly calculated . However it is possible to estimate its upper bound $\overline{D_{eA}}$ max, it is grounded on the approximation of the function $|H_2(j\omega)|^2$ by polynomial:

$$H_2(s) = \frac{1}{1+W(s)} = \frac{S^2}{K_2+K_2\tau s+S^2} = \frac{(j\omega)^2}{K_2+K_2\tau j\omega+(j\omega)^2} \;;$$

$$|H_2(j\omega)| = \frac{\omega^2}{\sqrt{(K_2-\omega^2)^2+K_2^2\tau^2\omega^2}} = \frac{\omega^2}{\sqrt{K_2^2+\left(K_2^2\tau^2-2K_2\right)\omega^2+\omega^4}}$$

$$|H_2(j\omega)|^2 = \frac{\omega^4}{(K_2-\omega^2)^2+K_2^2\tau^2\omega^2} \le C_2\omega^4$$

$$\frac{1}{(K_2-\omega^2)^2+K_2^2\tau^2\omega^2} \le C_2 \qquad ;\ \omega^2 = x$$

$$1= C_2\times[(K_2 - x)^2 + K_2^2\tau^2 x]$$

Solving this equation gives us the required polynomial coefficient C_{2}:

$$\rightarrow C_2 = \frac{1}{K_2^3\tau^2(1-\frac{K_2\tau^2}{4})}$$

After having determined the approximating polynomial coefficient from the condition $c_2 = \max |H_2(j\omega)|^2 / \omega^4$:

$$\overline{D}_{eA} = \frac{1}{\pi}\int_0^{\infty} H_2(j\omega)^2\, S_A(\omega)\, d\omega \leq \frac{1}{\pi}\int_0^{+\infty} c_2\omega^4 S_A(\omega) d\omega = C_2 D_2\,,$$

where $D_2 = \frac{1}{\pi}\int_0^{+\infty} \omega^4\, S_A(\omega) d\omega$ is the second derivative of error dispersion in accelerometer.

$$\overline{D}_{eA} = C_2 D_2 \rightarrow \overline{D}_{eA} = \frac{D_2}{K_2^3\tau^2(1-\frac{K_2\tau^2}{4})}$$

Hence:

$$\overline{De} = \overline{D}_{eG} + \overline{D}_{eA} \rightarrow$$

$$\overline{D_e}(K_2, \tau) = \frac{D_2}{K_2^3\tau^2(1-\frac{K_2\tau^2}{4})} + S_G\,\frac{1+k_2\tau^2}{2\tau}$$ **(my objective function)**

4.4 Analyze the potential accuracy of the navigation systems & calculate the root mean square error of the system

The objective function of optimization depends on particular parameters, whose optimal values are to be found:

$$\overline{D_e}(K_2, \tau) = \frac{D_2}{K_2^3 \tau^2 (1 - \frac{K_2 \tau^2}{4})} + S_G \frac{1 + k_2 \tau^2}{2\tau} \rightarrow \min.$$

Researching this function on extremum gives the optimum value for gain factor k_2 and time constant τ and the minimum upper bound of the total error in the form:

$$K_2^0 = 1.77 (\frac{D_2}{S_G})^{2/5}$$

$$\tau_0 = 0.92 (\frac{S_G}{D_2})^{1/5}$$

$$\overline{D}_{e,min} = \overline{D}_e(K_2^0, \tau^0) = 1.70 D_2^{1/5} S_G^{\frac{4}{5}}$$

[17]

Calculate the root mean square error of the system:

The idea of the use of the theoretical analysis was to determine the characteristics of input action:

The desired accuracy should lie in the range 3-5 m, if we consider this qualities of MEME- accelerometer & GPS:

D_2=1m^2/sec^4

S_G=10m^2/Hz

After optimization of the system (by using the optimal values of K_2 & τ) we will get an upper bound of error :

$$\overline{D}_{e,min} = 1.7\times (1)^{1/5}\times (10)^{4/5}= 10,71 \text{ m}^2$$

$$\sigma_{e,min}=\sqrt{\overline{D}_{e,min}} = 3,27 \text{ m}$$

Such an accuracy condition should be obtained as a result of optimizing the transfer functions $W(s)$ and using 2 sensors with these qualities:

D_2=1m^2/sec^4
S_G=10m^2/Hz

Chapter 5

ANALOG TO DIGITAL TRANSFORMATION

Objective:

- Getting the digital transfer function of analytical filters:
  $D_{c1}(z)=H_{c1}(s)$ & $D_{c2}(z)=H_{c2}(s)$

After the design of the desirable analog filters, they should be now transformed into digital filters.

Fig. 5.1 Chain of filter development

The transformation is supposed to:

1. faithfully approximate the frequency response of analog filter; and
2. Provide that the resulting digital filter is guaranteed to be stable.

One of most commonly used method of transforming analog is known as bilinear transformation. It is defined via expression:

$$S=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \quad [18]$$

Using this expression, the transformation of the analog filter transform function into a digital one can be expressed as:

$$H_{dig}(z)=H_{analog}\left(\frac{2}{T}\,\frac{1-z^{-1}}{1+z^{-1}}\right)$$

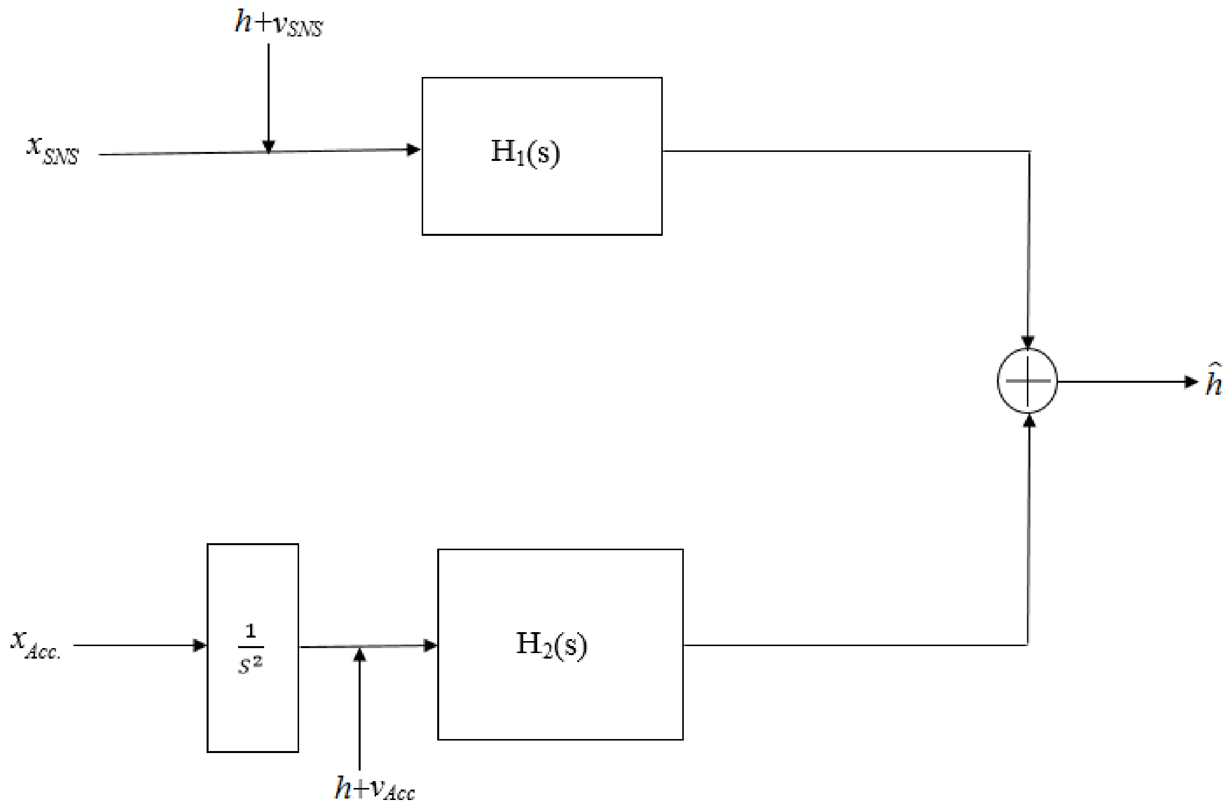


Fig. 5.2 two filters are needed to 1): process the output signal x_{SNS} of GPS : it consists of useful signal "*h*" + an error "v_{SNS}" , and 2): process the output signal of accelerometer x_{Acc} which consists of useful signal "*h*" + an error "v_{Acc}".

$$H_{c1}(s) = \frac{K_2(1+\tau s)}{K_2+K_2\tau s+S^2}$$

$$H_{c2}(s) = \frac{1}{K_2+K_2\tau s+S^2}$$

Then, by using the bilinear substitution for *s*, we can determine the transfer function in the digital domain.

$$D_{c1}(z)=\frac{K_2(1+\tau(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}))}{K_2+K_2\tau(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}})+(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}})^2}$$

&

$$D_{c2}(z)=\frac{1}{K_2+\tau(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}})+(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}})^2}$$

Select the discrete filters sampling period T:

Digital signal processing deals with discrete time signals. The signals are viewed as a stream of numbers, so it is necessary to determine first the discrete filters sampling period T:

$$W(j\omega)=\frac{K2\ (1+\tau j\omega)}{(j\omega)^2}$$

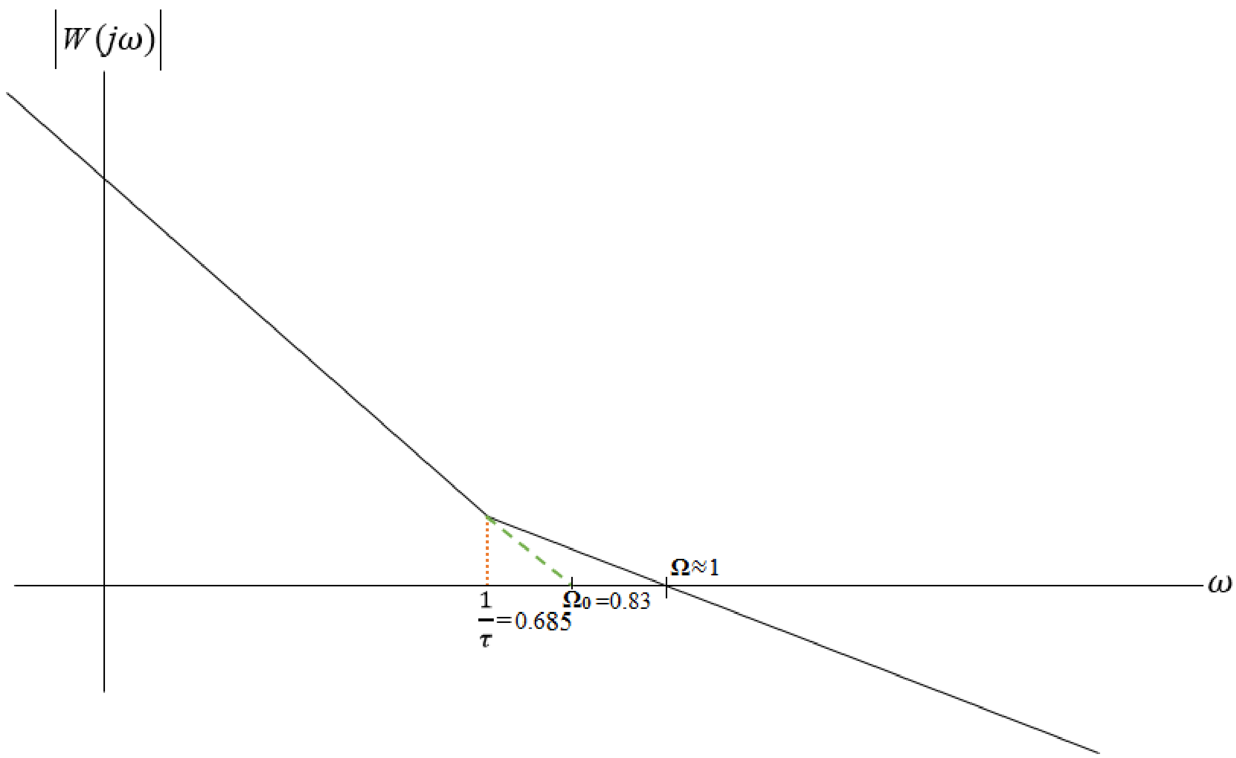


Fig. 5.3: the open-loop transfer function $\left|W(j\omega)\right|$ versus frequency ω.

In order that the digital transfer functions corresponds to the current transfer functions, T must lie 10 to 20 time further from Ω :

$\frac{2\pi}{T} \geq \Omega \times 10 \rightarrow T \leq \frac{2\pi}{10\Omega}$

$K_2 = 1.77(\frac{1}{10})^{\frac{2}{5}} \approx 0.7\ s^{-2}$

$\tau = 0.92(\frac{10}{1})^{\frac{1}{5}} \approx 1.46\ s$

$\frac{1}{\tau} = \frac{1}{1.46} \approx 0.685\ s^{-1}$

$\Omega_0 = \sqrt{K_2} = \sqrt{0.7} \approx 0.83\ s^{-1}$

$\rightarrow \Omega \approx 1\ s^{-1}$

$T \leq \frac{2\pi}{10\Omega} \approx 0.62\ s$

T selected as = 0.1 s

Hence:

In discrete time, the structure of the filters is described by the formulas:

$$D_{c1}(z) = \frac{K_2(1+\tau(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}))}{K_2+K_2\tau(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}})+(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}})^2} = \frac{c_0Z^2+c_1Z+c_2}{a_0Z^2+a_1Z+a_2}$$

Where:

$C_0 = K_2T^2 + 2K_2\tau T$; $a_0 = K_2T^2 + 2K_2\tau T + 4$;

$C_1 = 2K_2T^2$; $a_1 = 2K_2T^2 - 8$;

$C_2 = K_2(T^2 - 2\tau T)$; $a_2 = K_2T^2 - 2K_2\tau T + 4$;

and for D_{c2} (z):

$$D_{c2}(z) = \frac{1}{K_2+\tau(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}})+(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}})^2} = \frac{b_0Z^2+b_1Z+b_2}{a_0Z^2+a_1Z+a_2}$$

Where:

$b_0 = T^2$; $a_0 = K_2T^2 + 2K_2\tau T + 4$;

$b_1 = 2T^2$; $a_1 = 2K_2T^2 - 8$;

$b_2 = T^2$; $a_2 = K_2T^2 - 2K_2\tau T + 4$;

→ As a result, the differential equation of the filtering system will be:

$\hat{h}[n] = 0.050197h_{SNS}[n] + 0.003324h_{SNS}[n-1] - 0.046872h_{SNS}[n-2] + 0.002374\ddot{h}_{Acc}[n] + 0.004749\ddot{h}_{Acc}[n-1] + 0.002374\ddot{h}_{Acc}[n-2] + 1.896281\hat{h}[n-1] - 0.902930\hat{h}[n-2]$

Chapter 6

SIMULATION

Objectives:

- Study the stability and property of invariance of the designed system.
- Examine the chosen level of noise for each sensor.

In this chapter the system response to different inputs will be represented. The chosen platform for the real time implementation of the integrated GPS/Accelerometer algorithm is MATLAB/Simulink.

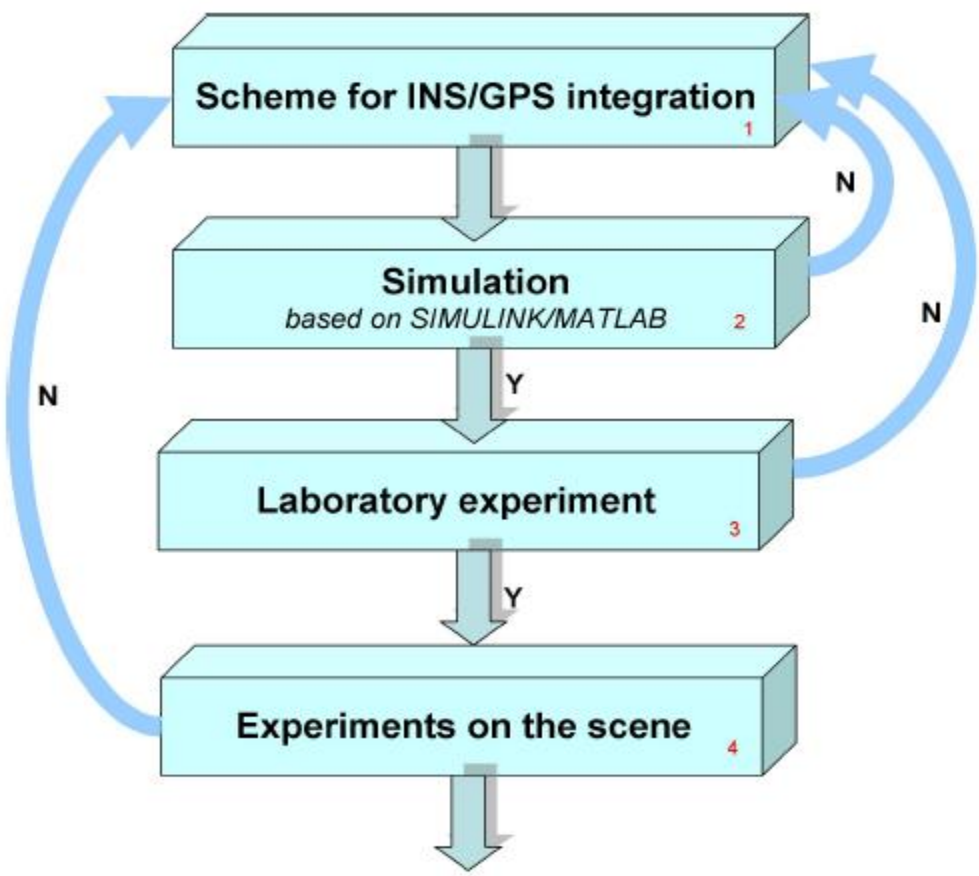


Fig. 6.1. Flow chart of the INS/GPS implementation [19]

6.1 Stability of the integrated measuring system:

The condition of invariance will be met if the useful signal comes out without disturbance after filtering, in this case the dynamic component of the error will be zero and the error *e* will not depend on the measured altitude *h*. —that is, the total error will have only two components, caused by the mutually independent disturbances S_G and D_2 .

Assume an input for useful signal has a form of unit step function, for fulfillment the condition of invariance an output of the same (or almost the same) function should be obtained in order to acquire a property of invariance

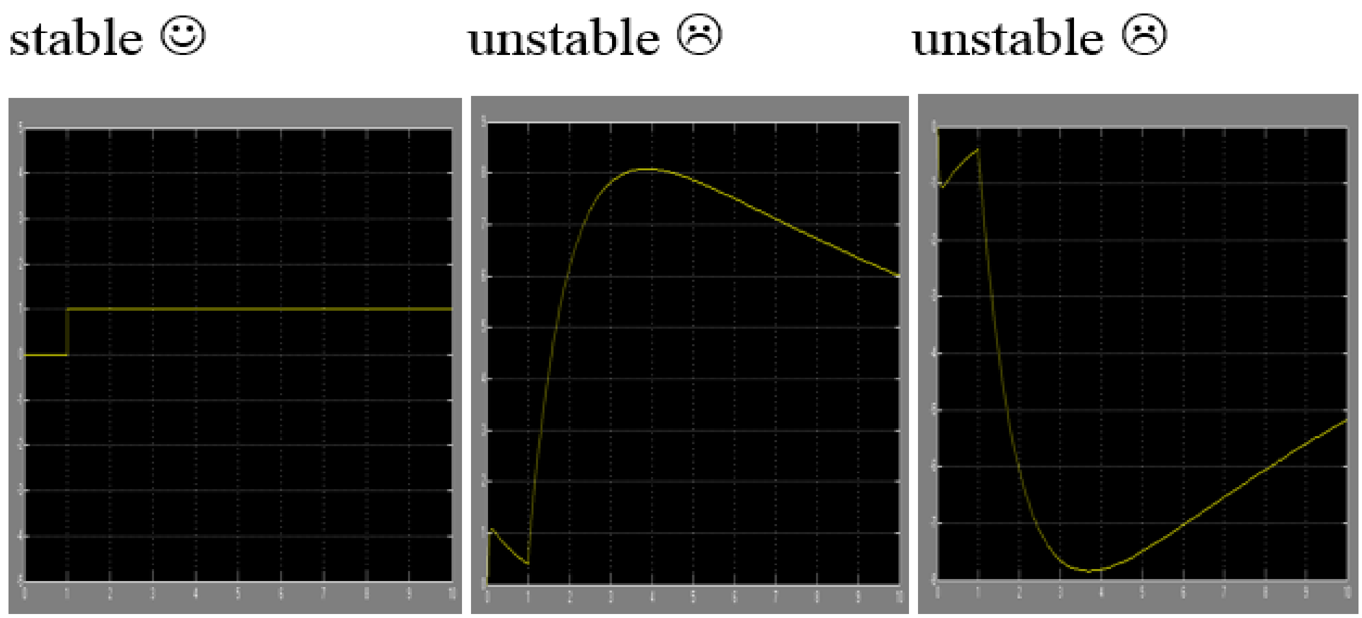


Fig. 6.2 Dynamic stability

In order to test the invariance property of the system two different inputs variation (harmonic and step function) are applied. The results are presented in 6.4 and 6.5 for harmonic and step function respectively.

The response of the system to the applied inputs will be also later used to determine the frequency-amplitude characteristics of the developed filtering system and to test the chosen values of error for each sensor.

Those needed parameters for the filters transfer functions, are known in the chapter 5 and are as follows:

$$D_{c1}(z)=\frac{0.050197Z^2+0.003324Z-0.046872}{1Z^2-1.896281Z+0.902930}$$

$$D_{c2}(z)=\frac{0.002374Z^2+0.004749Z+0.002374}{1Z^2-1.896281Z+0.902930}$$

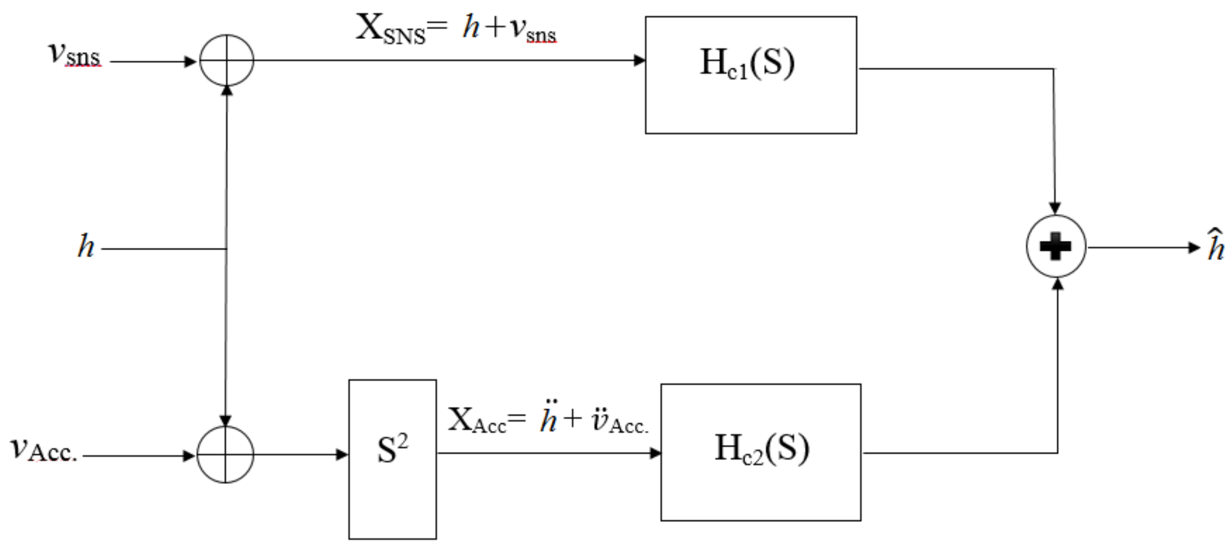


Fig. 6.3 Scheme of designed system

These scheme will be in Matlab/Simulink designed and its response to various inputs will be discussed, in order to be able to make a statement about system behavior.

I. System response to Harmonic function:

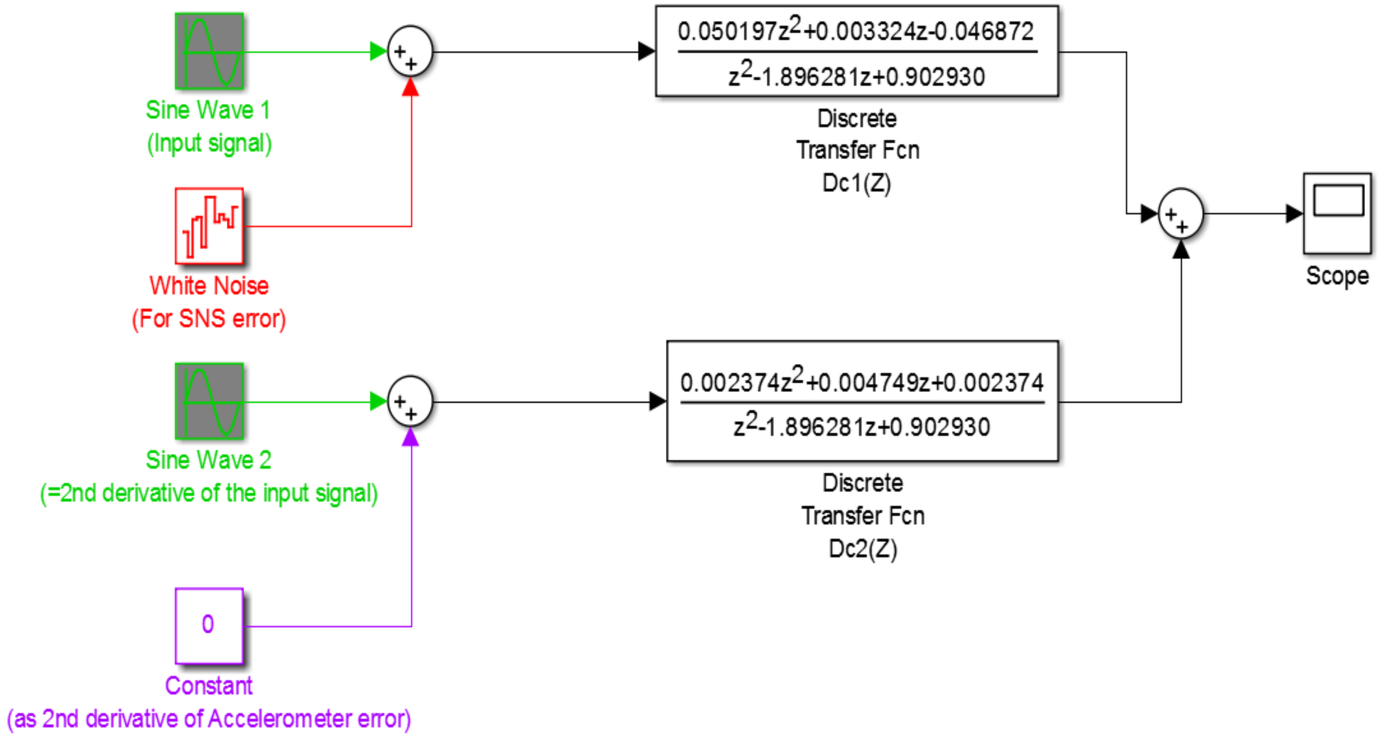
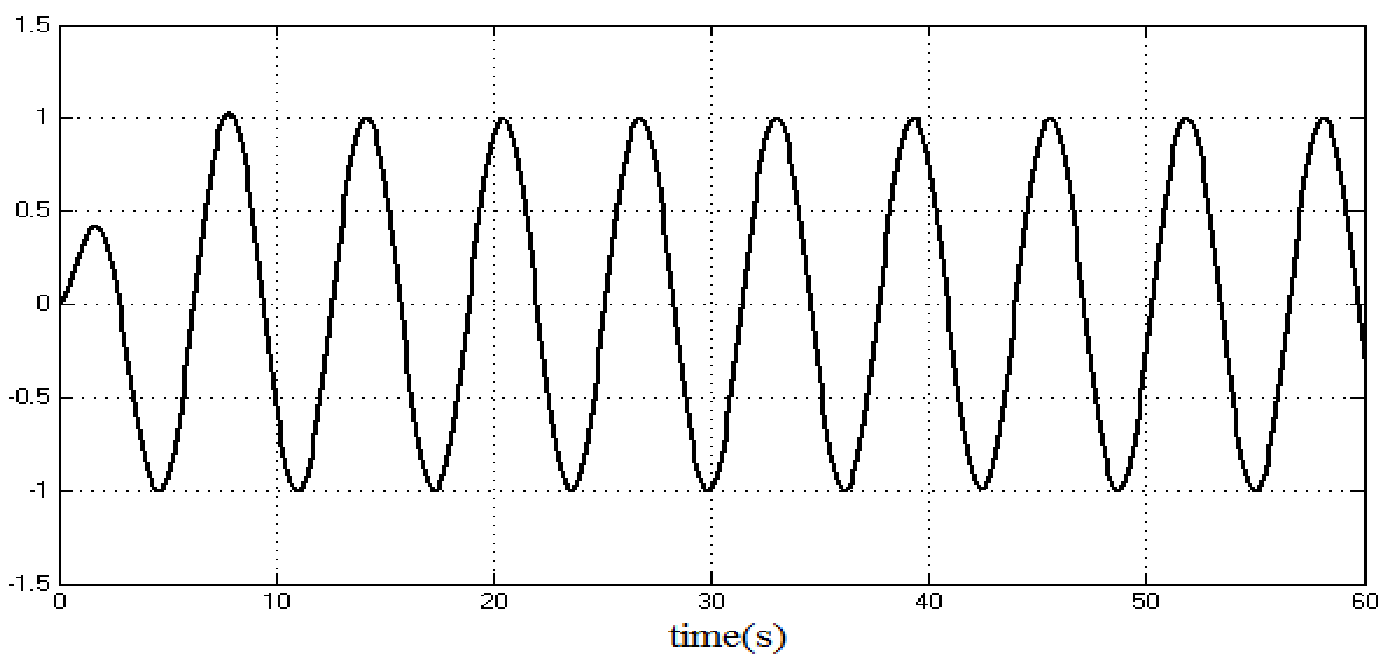


Fig. 6.4 Simulink model of filtration's schema under harmonic input = A*sin(ωt) , where A=1 & $\omega = 1$

<u>Output:</u>



II. System response to Step function:

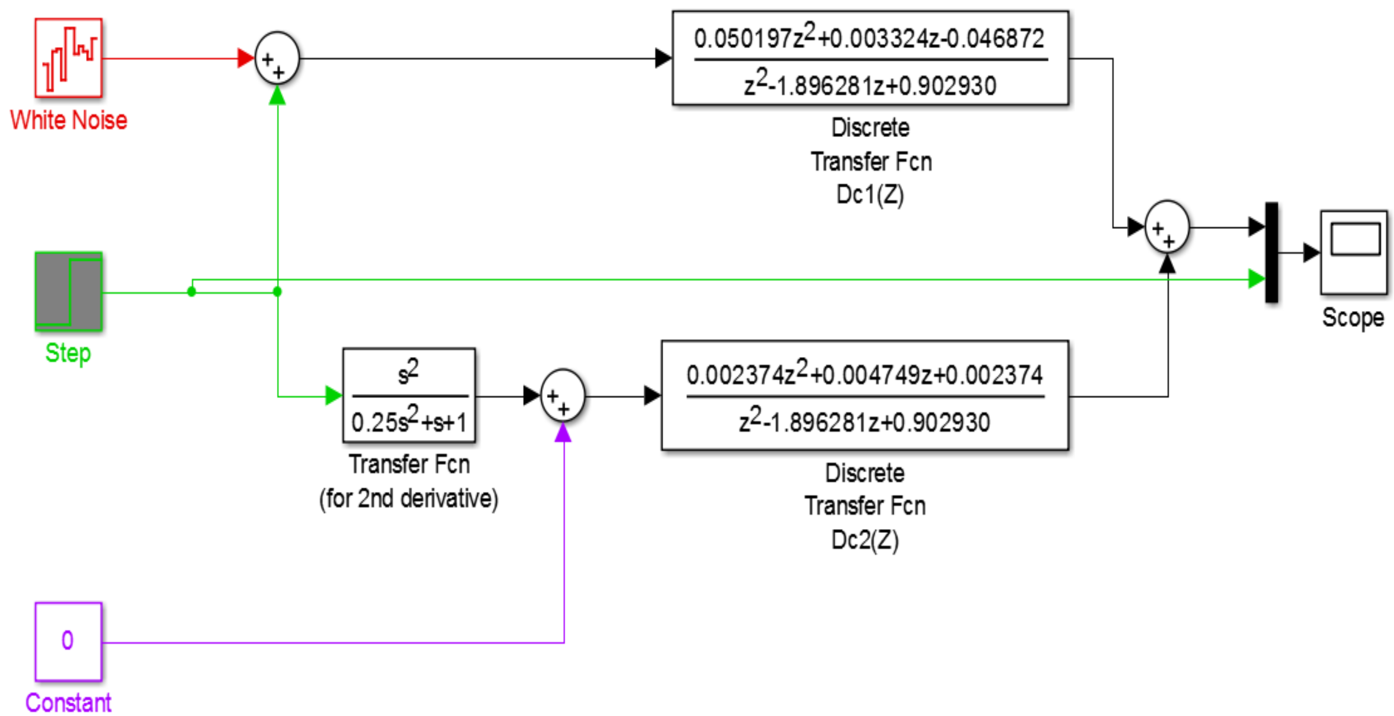
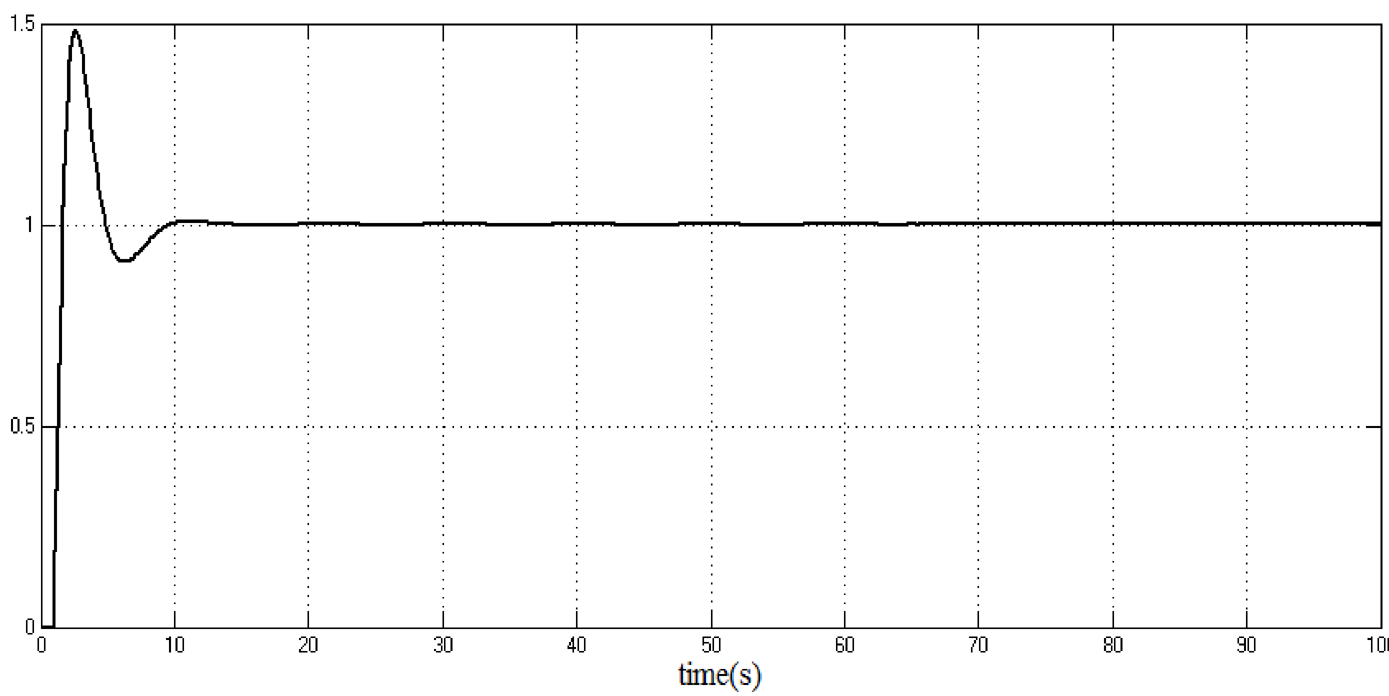


Fig. 6.5 Simulink model of filtration's schema under step input

<u>Output:</u>



6.2 check the values of errors for 2 sensors in SIMULINK:

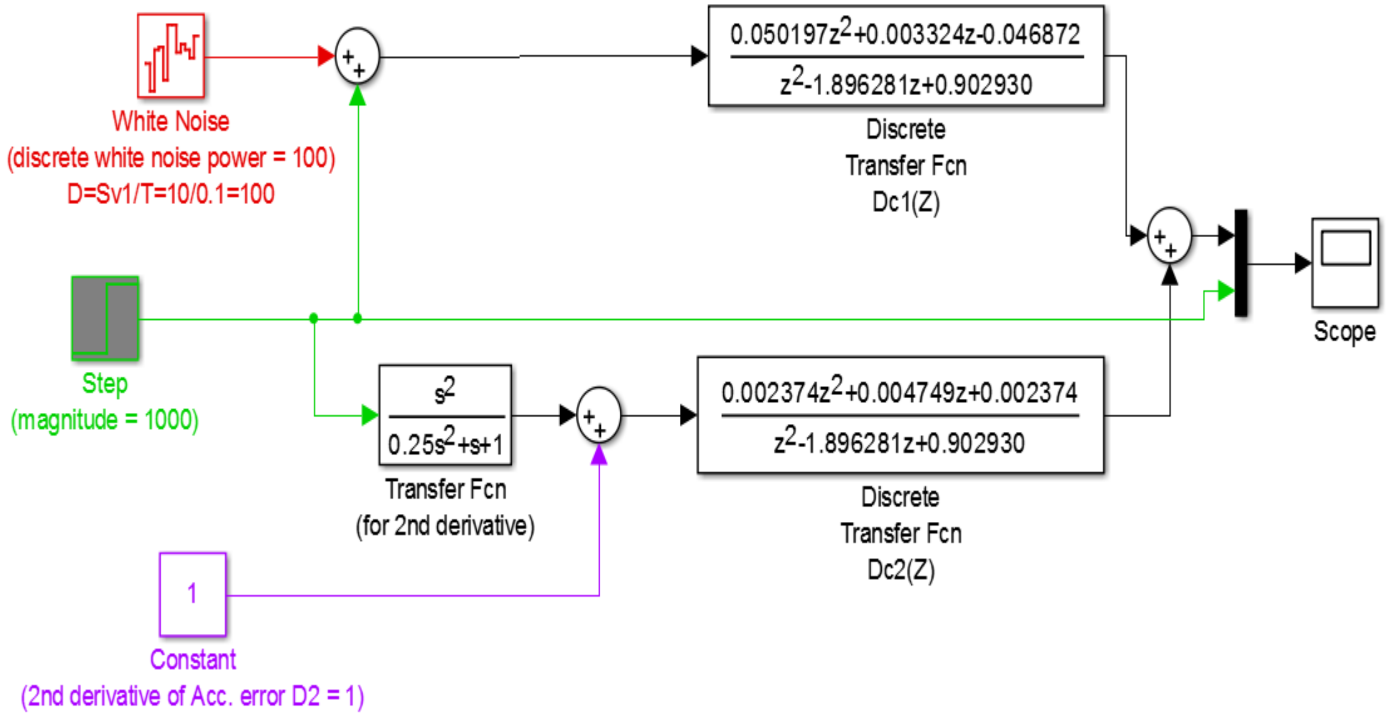
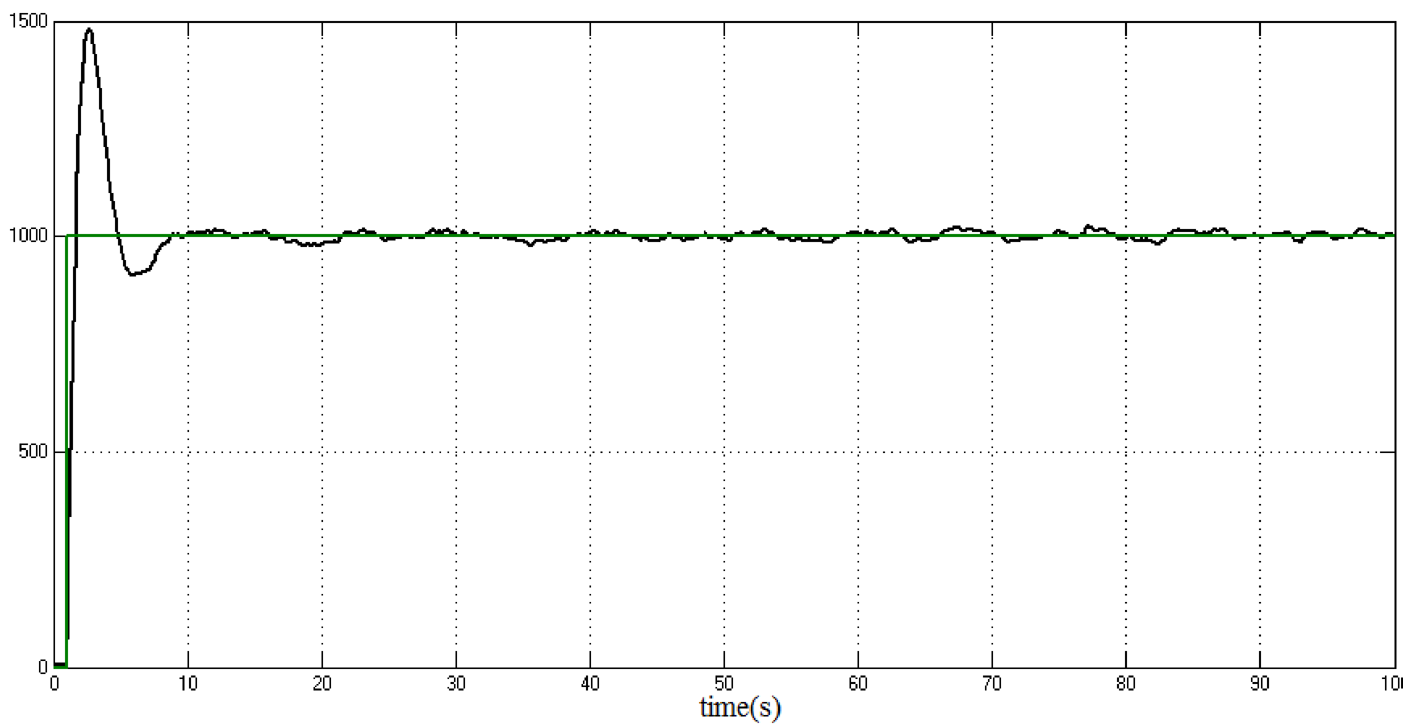


Fig. 6.6 Simulink model of filtration's schema under input: step=1000

Output:



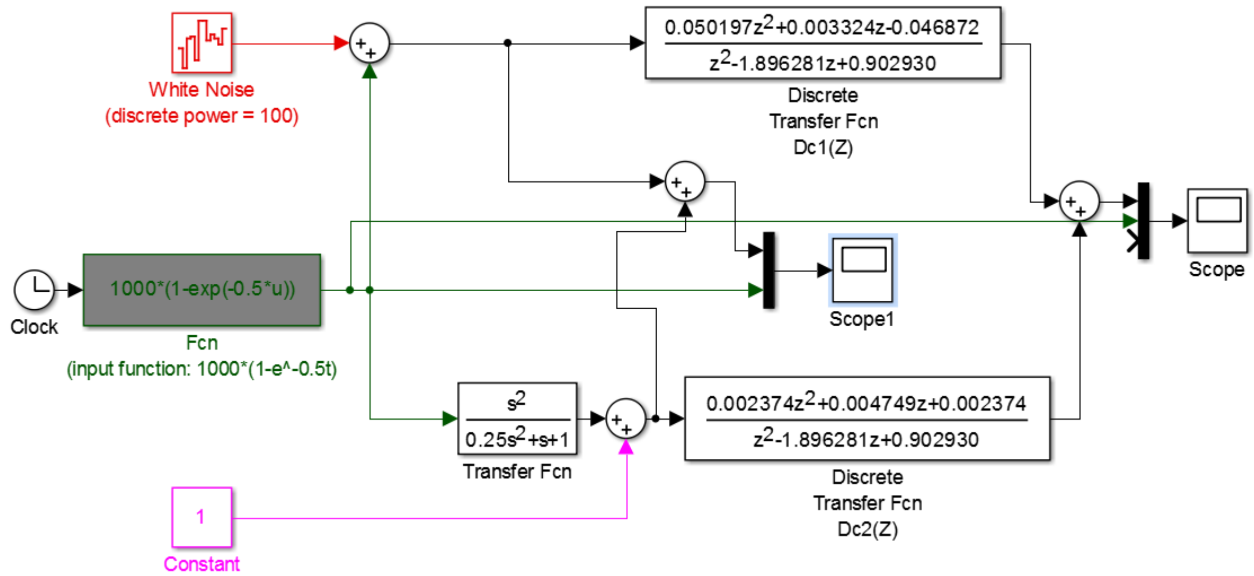


Fig. 6.7 Simulink model of filtration's schema under input: 1000×(1-e$^{-0.5t}$)

<u>Output:</u>

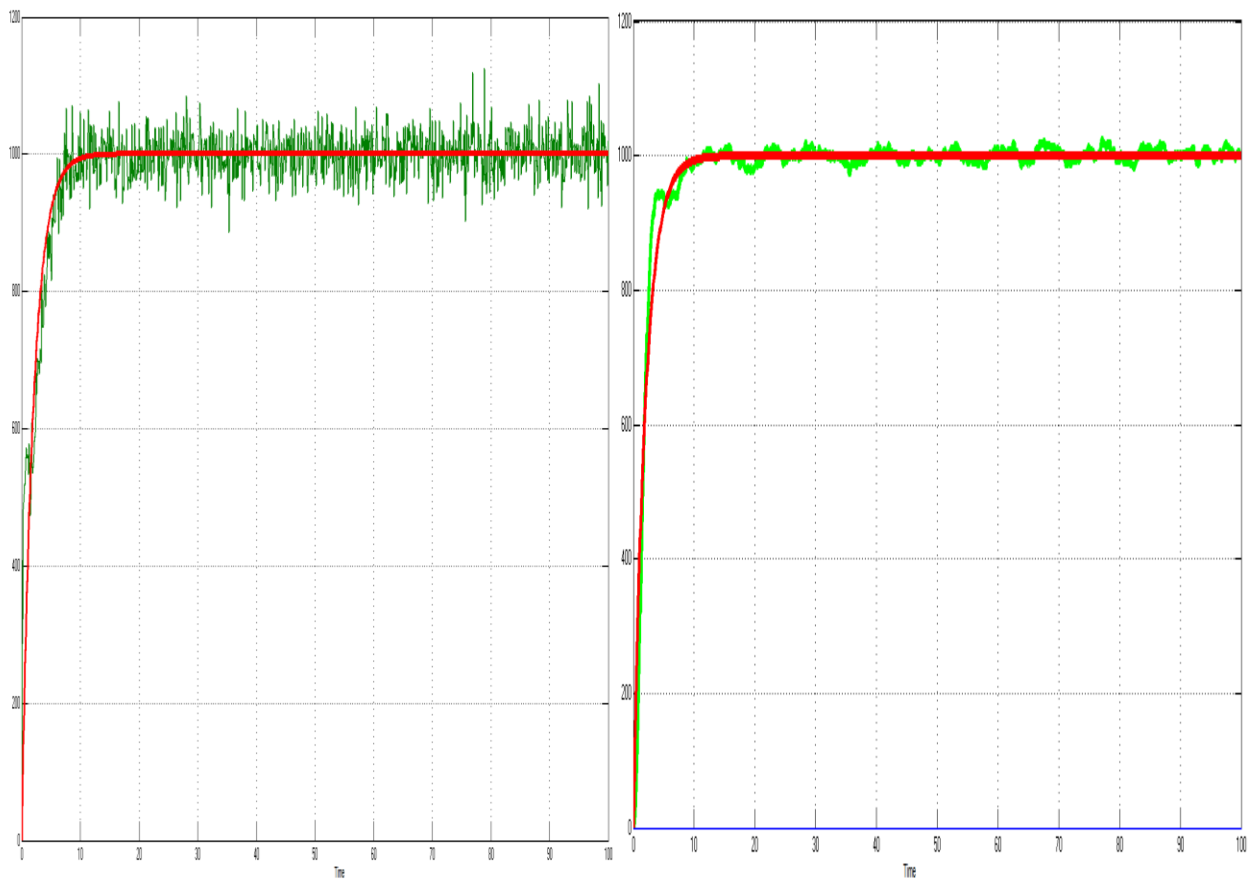


Fig. 6.8: Output Signal before and after Filtering

1. <u>Amplitude-frequency characteristic:</u>

Changing the frequency ω of the harmonic input function: $1\times\sin(\omega t)$ shows The amplitude dependence of the frequency, it shows the only acceptable range of frequencies for useful signal. This range from zero to approx. 10 rad/s.

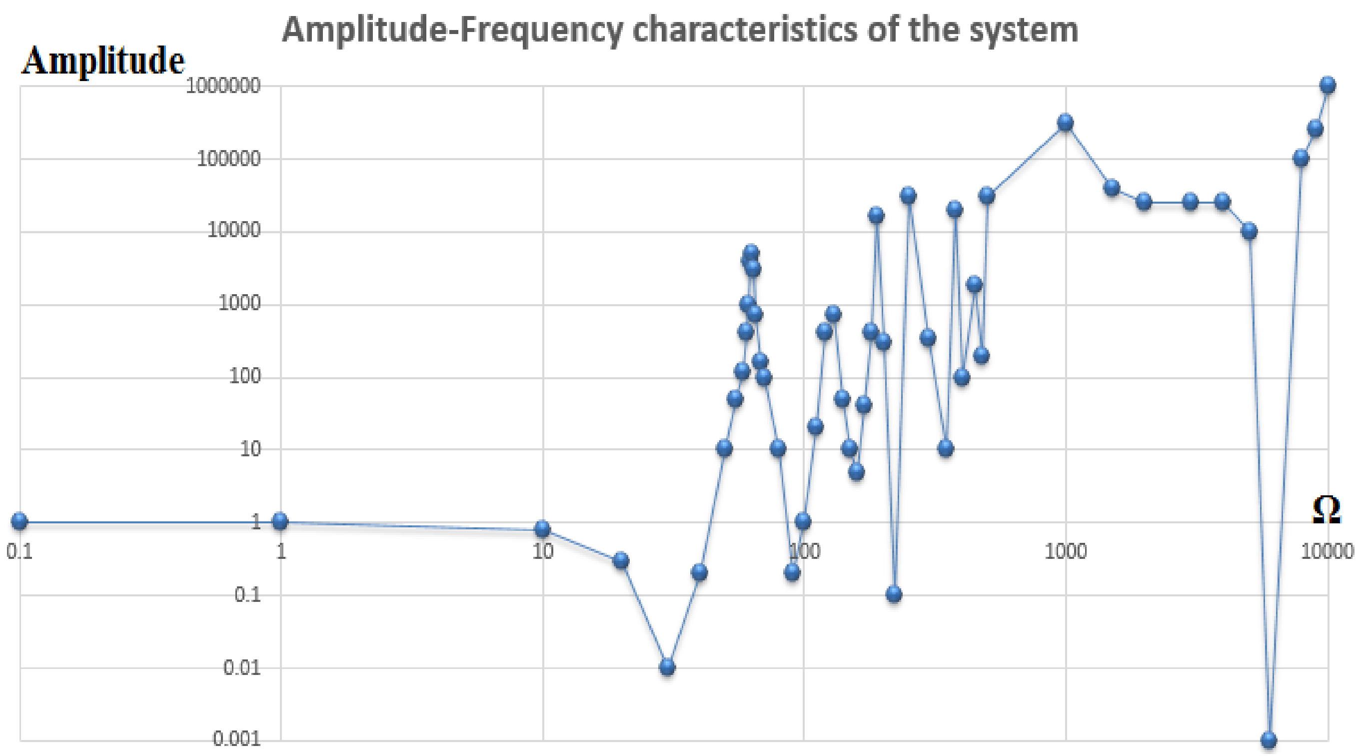


Fig. 6.8: Amplitude versus frequency

2. <u>Check the scheme of compensation:</u>

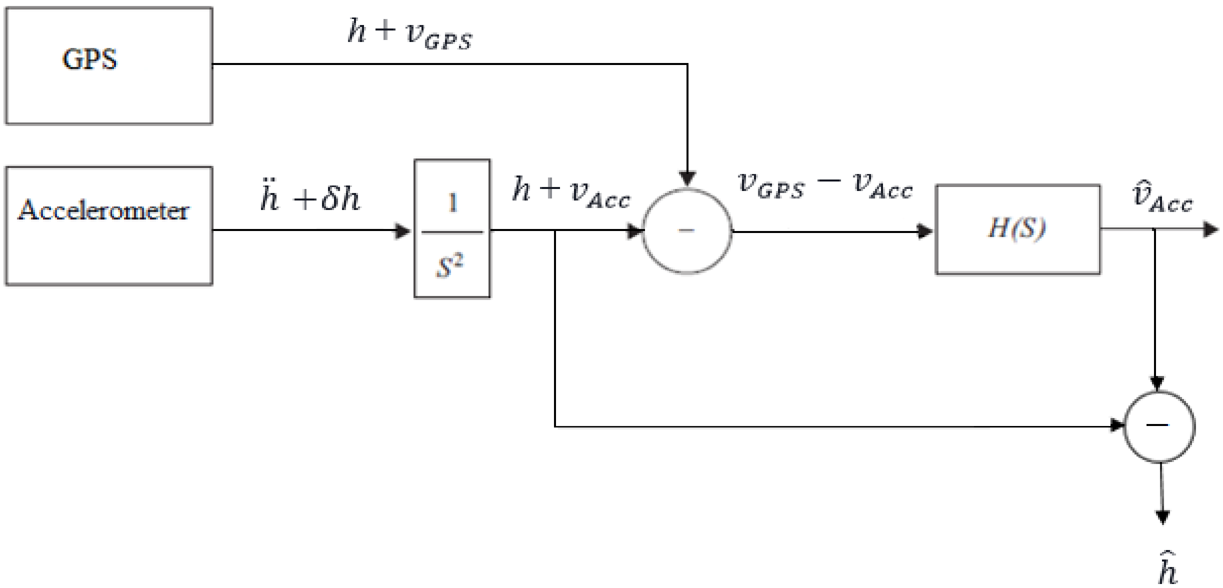


Fig. 6.9 Scheme of compensation

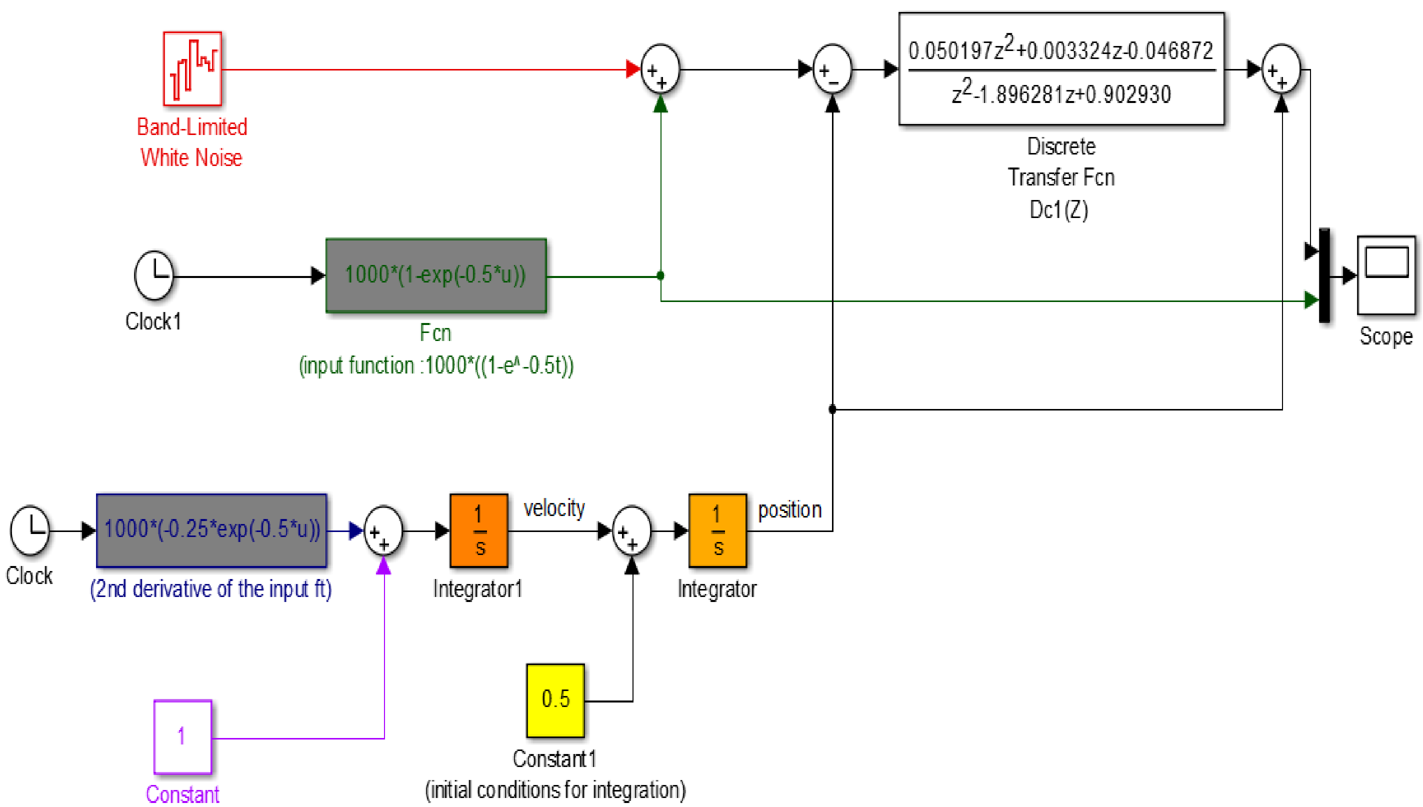
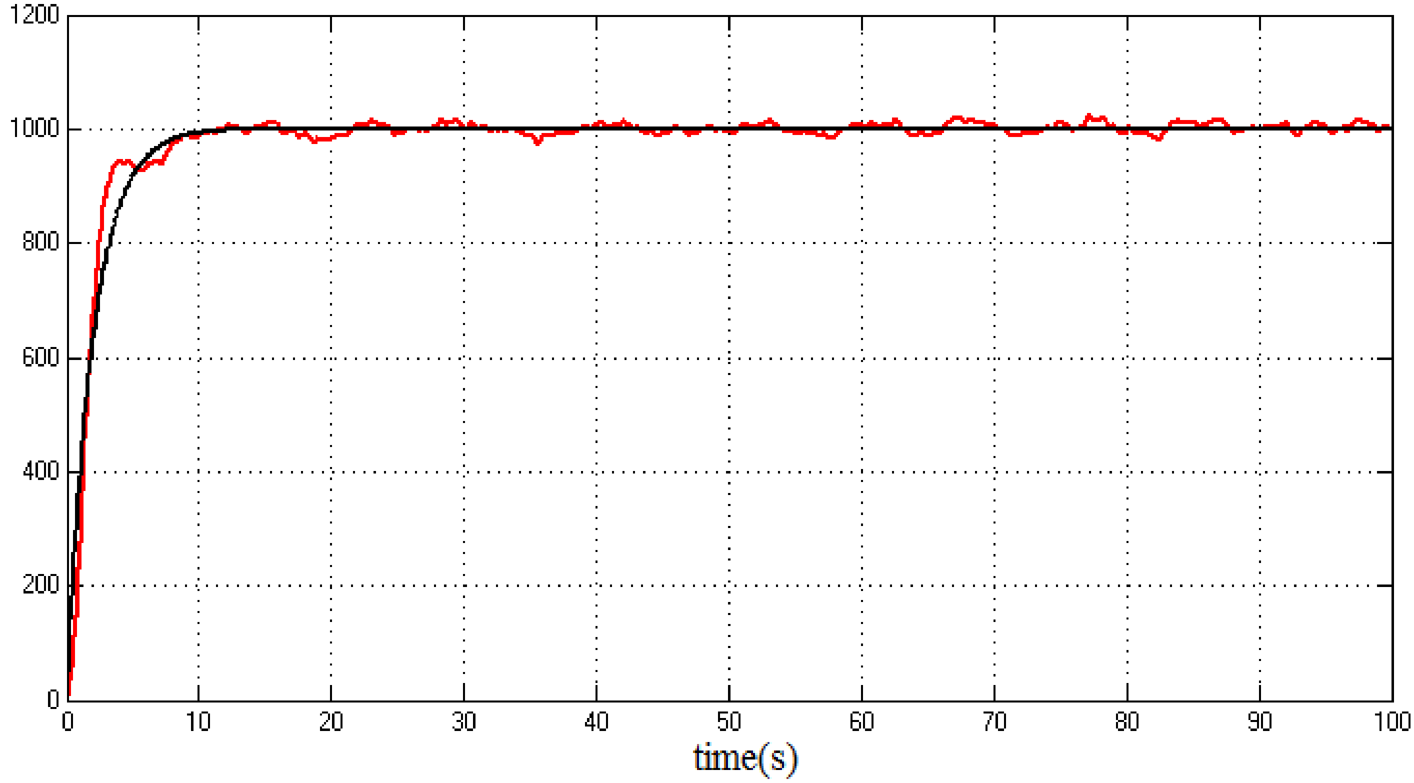


Fig. 6.10 Scheme of compensation in Simulink

Output:



3. **Results vs prediction:**

The main objectives of this chapter were to compare the calculated error $\sigma_{e, min}$ with the results from Matlab and the performance of the filters with different initial errors.

To test the stability and the invariance of the system an input of step function and harmonic function was chosen: It is seen from figures: 6.4 & 6.5 that the output is identical to the input, that's what I expected because the system has been so designated, that no dynamic error appears, this means that the system lets the whole useful signal pass without disturbance.

After checking the property of invariance, I checked the chosen values of errors for each sensor and the reliability of the designed filtering system, this can be achieved in Matlab by the use of the block "constant" for D_2 and block "band limited white noise" for S_G (figures 6.6 & 6.7). S_G corresponds to the level of current (not discrete) white noise, the power of the discrete white noise must be

$D = S_G/T \rightarrow = 10/0.1 = 100\ m^2/Hz^2$

After installing this data in Matlab and under an input of step function with magnitude =1000 and an exponential function: 1000×(1-$e^{-0.5t}$), I got the stochastic output. It is clear form comparing the two graphs (the mixed with errors signal before and after filtering) that as initial errors increase and without filtering, the useful signal disturbed drastically, after installing the two designed filters, I got a good estimation of the signal, that showed me the performance of the filters.

Furthermore it was searched for an excitation frequencies, which have the following property: optimum bandwidth of the excited frequency spectrum, it is shown in fig 6.8 that the signal's frequency should not exceed 10 rad/s.

In chapter 4 the Mathematical theory of optimal filtration gave:

$$\overline{D}_{e,min}=1.70 D_2^{1/5} S_G^{\frac{4}{5}}$$

The chosen values of error for each sensor were:

$D_2=1m^2/sec^4$
$S_G=10m^2/Hz$

It gave:

$$\overline{D}_{e,min} = 1.7\times (1)^{1/5}\times (10)^{4/5} = 10,71 \text{ m}^2$$

$$\sigma_{e,min}=\sqrt{\overline{D}_{e,min}} = 3,27 \text{ m}$$

The error $\sigma_{e,min}$ represents the difference between the output and the input signal

It is for example for t = 10 sec:

$\sigma_{e,\,min}$ = 1002 – 1000 = 2 m

And for t=100 sec: $\sigma_{e,\,min}$ = 1004 – 1000 = 4 m

The calculated error is almost similar to from Matlab displayed error.

Chapter 7

IMPLEMENTATION OF DEVELOPED GPS/INS ALGORITHM

Objectives:

- Verification of Developed Algorithms:

I. Dynamic Modeling

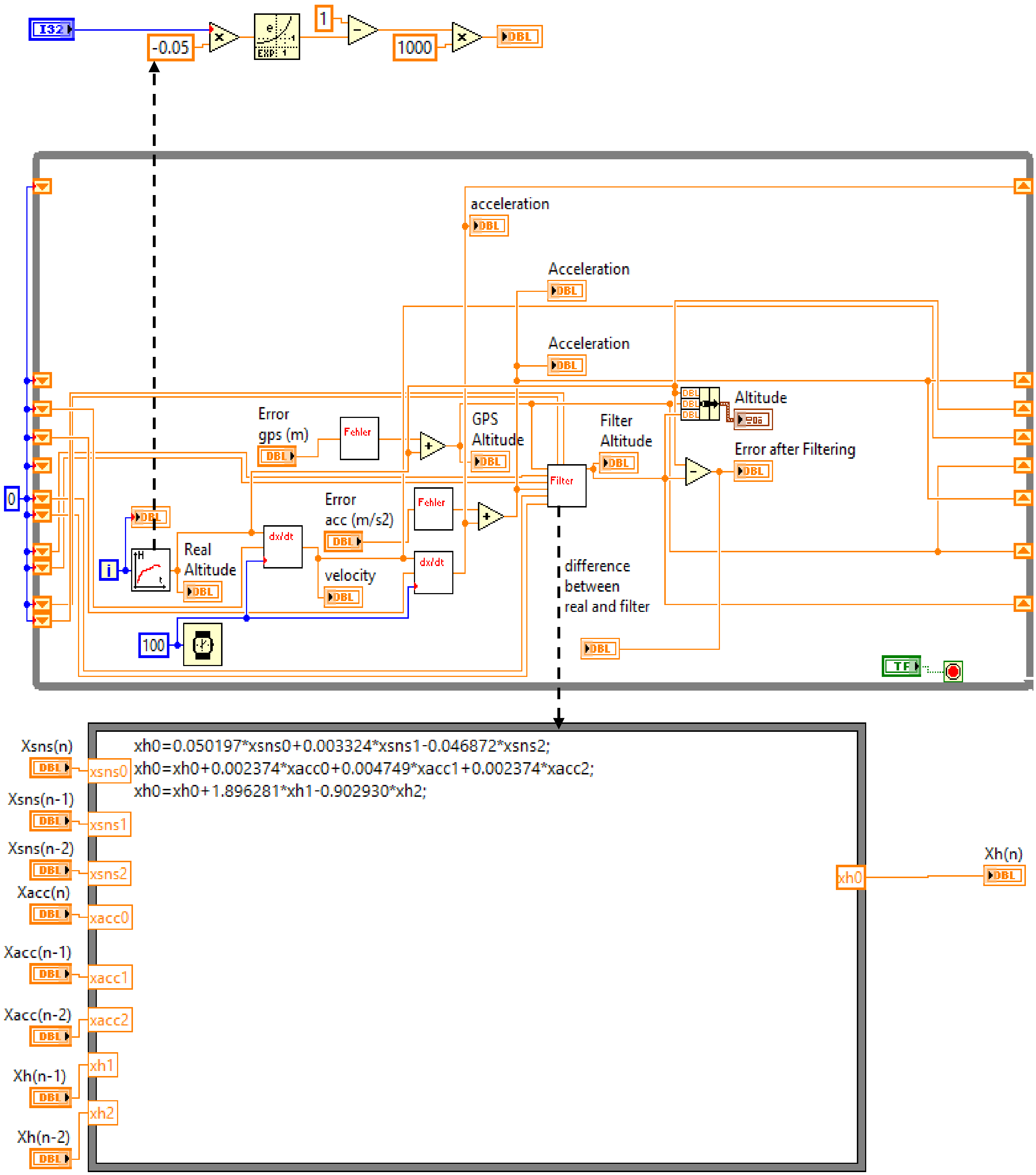
II. Lab and Flight Verification

After the developed navigation filter has been found in the simulation as a highly efficient, tests and flight tests have been carried out to verify the developed algorithm in the experiment. The focus of the analysis is the achievable accuracy, because this is essential for the considered application.

In order to see how the developed algorithm would work during real flight, first I modeled the System in LabVIEW, then I purchased the needed hardware and tested my system in Saint-Petersburg near Moyka River/Russia and then in an aircraft near airfield Linnich-Boslar/Germany

Compensation configuration and complementary configuration, both of them showed that they have almost the same dynamic features, the achieved accuracy during simulation was similar, so now for real flight verification I chose the complementary configuration because it is easier to program.

7.1 Dynamic Modelling:



The First chosen Scenario:

Assume that the desired flight path has the form of exponential function:

$$1000\times(1-e^{-0.05t})$$

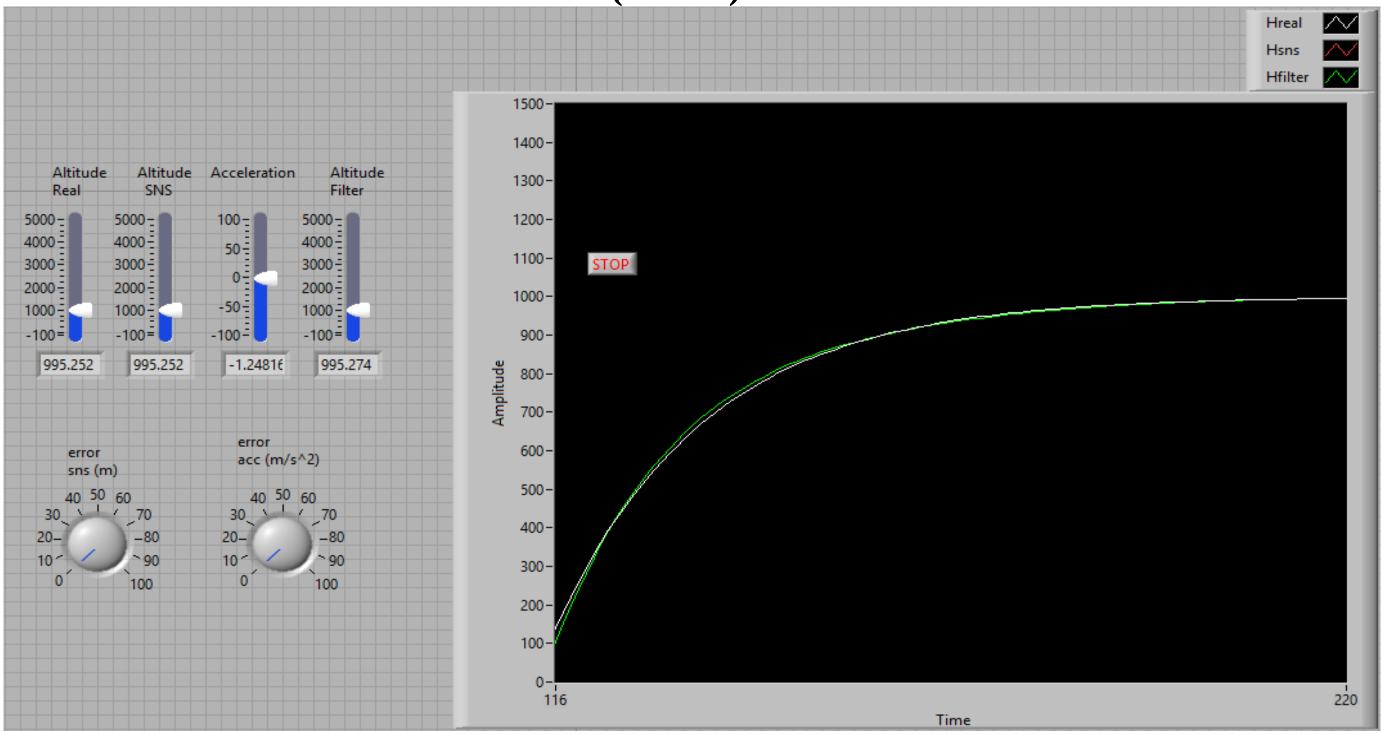


Fig. 7.1 Dynamic Modeling of the first Scenario

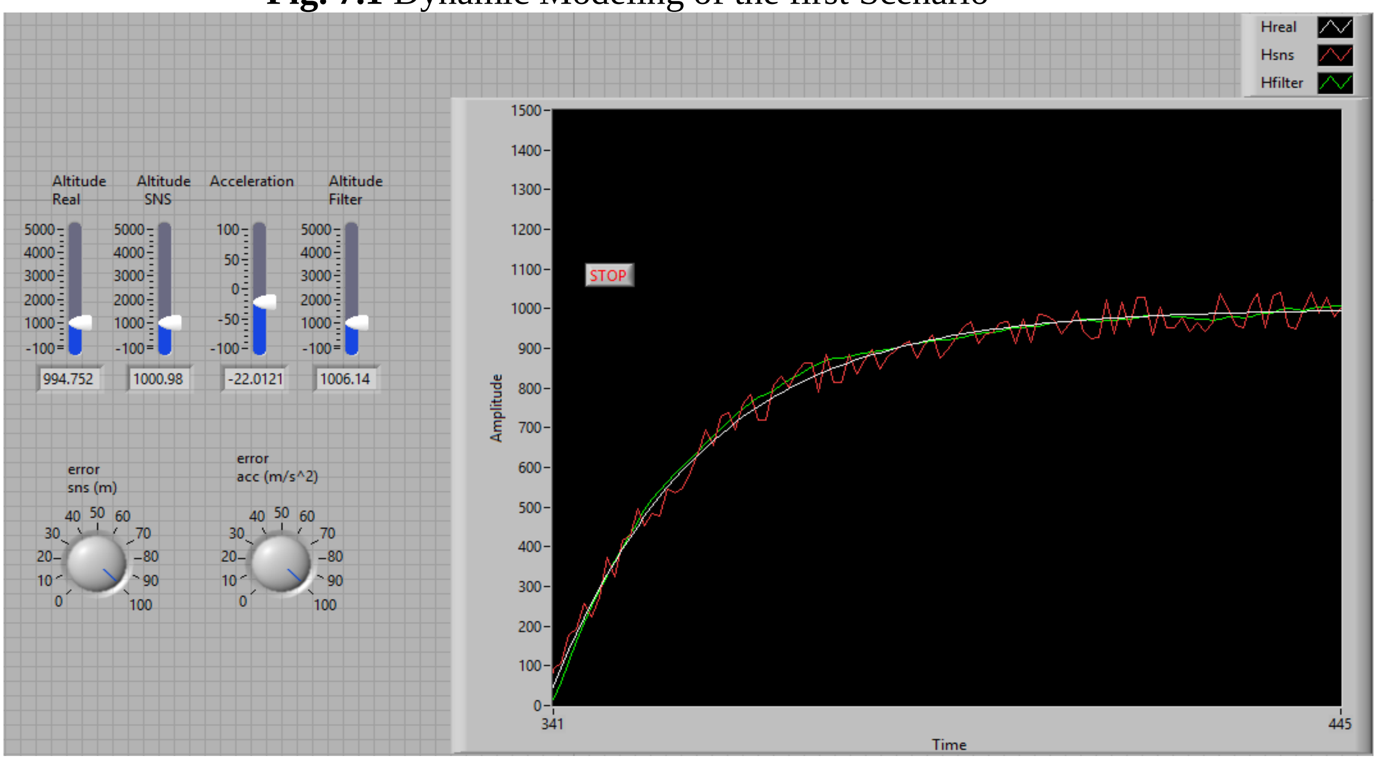


Fig. 7.2 Dynamic Modeling of the first Scenario after increasing the sensor's noises

The Second chosen Scenario:
Assume other desired flight path has the form of nonlinear function:

$$(0{,}2374t)^4 + 10t$$

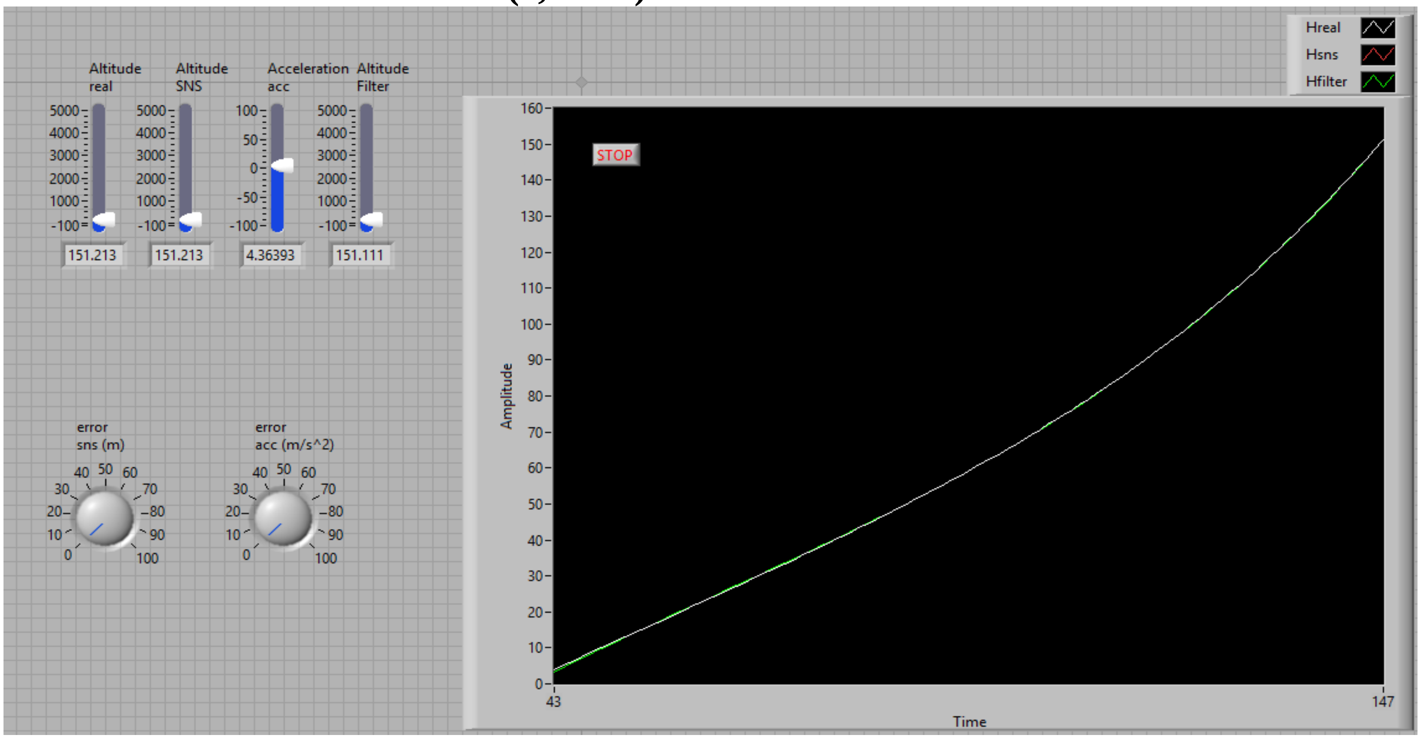


Fig. 7.3 Dynamic Modeling of the second Scenario

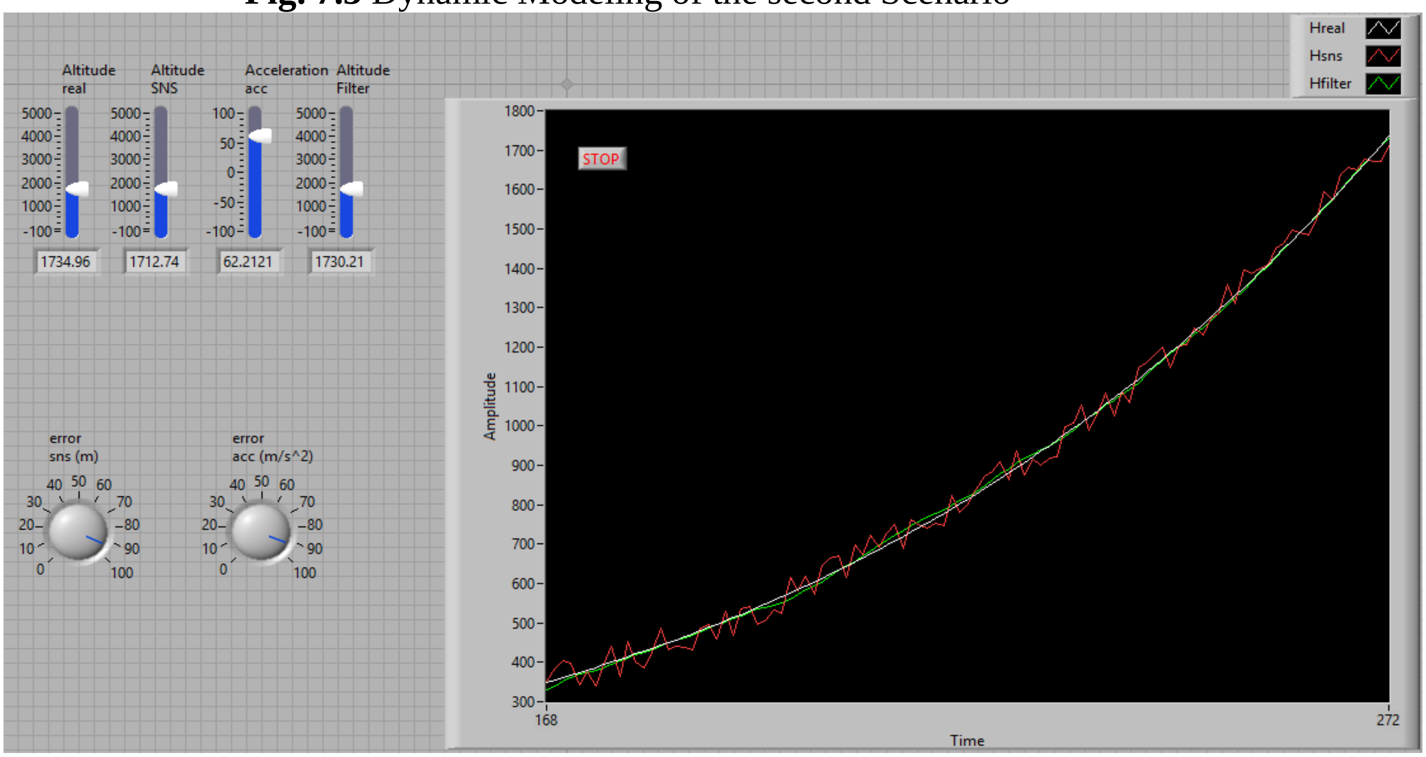


Fig. 7.4 Dynamic Modeling of the second Scenario after increasing the sensor's noises

As it can be seen from **Fig7.2** and **Fig7.4**, even with increasing the errors in each sensors, the readings of the developed filter stay compatible with the real desired trajectory. The average of difference between real desired altitude and altitude after error filtering is as calculated equal to 3,2 m .

Now, Lab and Flight Testing should be carried, the main problems to be solved here are:

- ❖ Purchase the necessary Hardware
- ❖ Writing the necessary code for GPS, INS, designed Filter, and other needed hardware
- ❖ Connect all the hardware in Arduino

7.2 Hardware purchase:

GPS:

Purchase

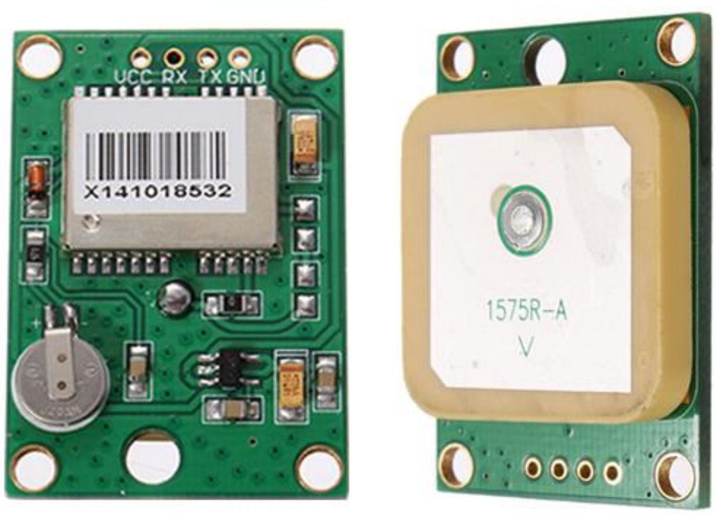


Fig. 7.5 GY-NEO6MV2

Features	
Accuracy	5 m
Dimensions	25mm x 35mm x 30mm
Weight	16 g
Price	11,00 €

Fig. 7.6 Features of GY-NEO6MV2

MEMS INS:

Purchase

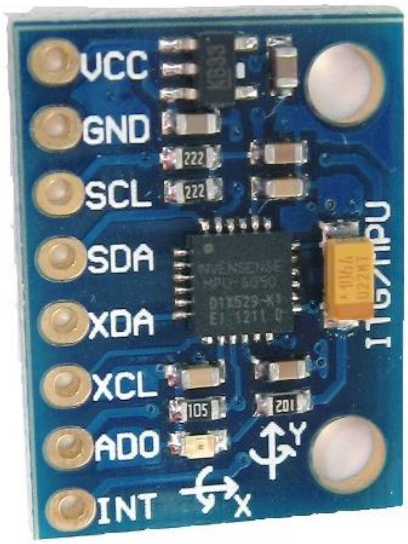


Fig. 7.7 MPU6050

Features	
Update rate	up to 40 Hz
Gyroscope full-scope range	±250, ±500, ±1000, ±2000°/sec
Accelerometer full-scale range	±2, ±4, ±8, ±16g
Dimensions	21mm x 16mm x 4mm
Weight	8 g
Price	4,00 €

Fig. 7.8 features of MPU6050

7.3 The system structure of altitude determination based on complex processing of accelerometer and satellite navigation system readings:

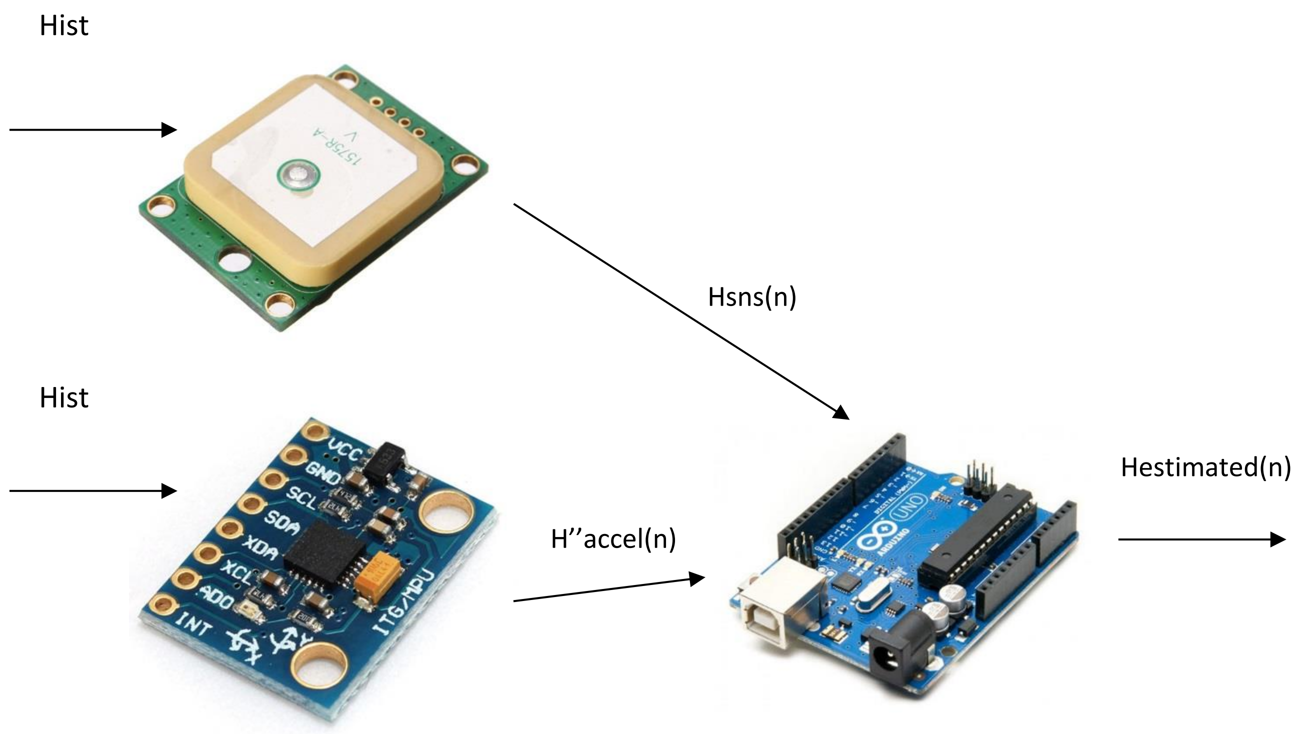


Fig. 7.9 sensors data fusion

H_{ist} - true altitude

H''_{accel} - acceleration measured by the accelerometer

H_{sns} - altitude measured by satellite navigation system

$H_{estimated}$- calculated flying altitude based on the following differential equation:

$H_{estimated}(n) = 0{,}050197 * Hsns(n) + 0{,}003324 * Hsns(n-1) - 0{,}046872 * Hsns(n-2)$
$+ 0{,}002374 * H''accel(n) + 0{,}004749 * H''accel(n-1) + 0{,}002374 * H''accel(n-2)$
$+ 1.896281 * H_{estimated}(n-1) - 0.902930 * H_{estimated}(n-2)$

Where:
n- number of the current measurement
n-1 - the number of previous measurements
n-2 - measuring the number two steps back

Next, a program for used hardware should be written in Arduino, here a code for GPS, MEMS INS, SD card (for saving the sensors readings), and designed filter should be written:

```
#include <Wire.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <SD.h>

TinyGPS gps;
SoftwareSerial ssGPS(5, 6);

static void print_int(unsigned long val, unsigned long invalid, int len);
static void print_date(TinyGPS &gps);
static void smartdelay(unsigned long ms);
static void smartdelay(unsigned long ms);
static void print_float(float val, float invalid, int len, int prec);
static void print_int(unsigned long val, unsigned long invalid, int len);
static void print_date(TinyGPS &gps,String DataGPS1=" ");
static void print_str(const char *str, int len);

String DataGPS = " ";

//SD card
int CS_PIN = 4;
bool read_time;
int time_0;
unsigned int LED = 2;
const int LED1 = 8;
const int Button = 7;
boolean writeCD = false;
boolean lastButton = HIGH;
boolean currentButton = HIGH;
boolean LedOn = false;

float Hightcode = 1000000.00;
uint8_t IMUAddress = 0x68;
int32_t accX;
int16_t WlenAcc = 8;
int cou_a = 0;
int flg_fi = 1;
int32_t Accl_S = 0;
int32_t Accl_wi[8];
int n = 2;
int k = 2;
int p = 2;
int32_t Hight[3];
int32_t XAcc[3];
int32_t XSNS[3];
int32_t Accel_out = 0;
```

```
int32_t Accel_OUT = 0;
float Acc = 0;
float HightOut = 0;
//uint32_t timer;

void setup() {

pinMode(LED1,OUTPUT);
pinMode(Button,INPUT);
pinMode(LED, OUTPUT);

  Serial.begin(9600);
  ssGPS.begin(9600);
  Wire.begin();
  i2cWrite(0x6B,0x00); // Disable sleep mode
  if(i2cRead(0x75,1)[0] != 0x68) {
    Serial.print(F("MPU-6050 with address 0x"));
    Serial.print(IMUAddress,HEX);
    Serial.println(F(" is not connected"));
    while(1);
  }
  pinMode(CS_PIN, OUTPUT);

    if (!SD.begin(CS_PIN))
    {
      Serial.println("NO CARD");
    } else  {Serial.println("CARD READY");
              LedOn = !LedOn;
              }

  // LED indicates GPS signal's valid

}

boolean vibro (boolean last)
 {
  boolean current = digitalRead(Button);
  if (last != current)
  {
    delay(5);
    current = digitalRead(Button);
    return current;
  }
}

void loop() {
```

```
  unsigned long age, date, time, chars = 0;
  currentButton = vibro(lastButton);
   if (lastButton == HIGH && currentButton == LOW)
    {
   //   LedOn = !LedOn;
      writeCD = !writeCD;
      }
     lastButton = currentButton;
digitalWrite(LED1,LedOn);
    /* Update all the values */
  uint8_t* data = i2cRead(0x3B,14);
  accX = ((data[0] << 8) | data[1]);
/**/  if (flg_fi == 1)
       {
        if (cou_a < WlenAcc)
        {
         Accl_wi[cou_a] = accX; cou_a++;
        }
        else {
              flg_fi = 0;
              for (cou_a = 0; cou_a < WlenAcc; cou_a++) {Accl_S += Accl_wi[cou_a];}
              cou_a = 0;
             }
       }
      else {
            if (cou_a < WlenAcc)
            {
             Accl_S -= Accl_wi[cou_a];
             Accl_wi[cou_a] = accX;
             Accl_S += Accl_wi[cou_a];
             Accel_OUT =((Accl_S>>3) - Accel_out); // Acceleromrter Zero correction
             cou_a++;
            }
            else cou_a = 0;
           }
 Acc=float(Accel_OUT);
 Acc=Acc/16384;
 Acc=(Acc-1)*9.728;

//DataGPS = print_date(sz);
// DataGPS = (month, day, year, hour, minute, second);
  // remember time_0 when GPS get signal
  if (!TinyGPS::GPS_INVALID_F_ALTITUDE && !read_time)
  time_0 = millis()/1000;
 if (!TinyGPS::GPS_INVALID_F_ALTITUDE) {
```

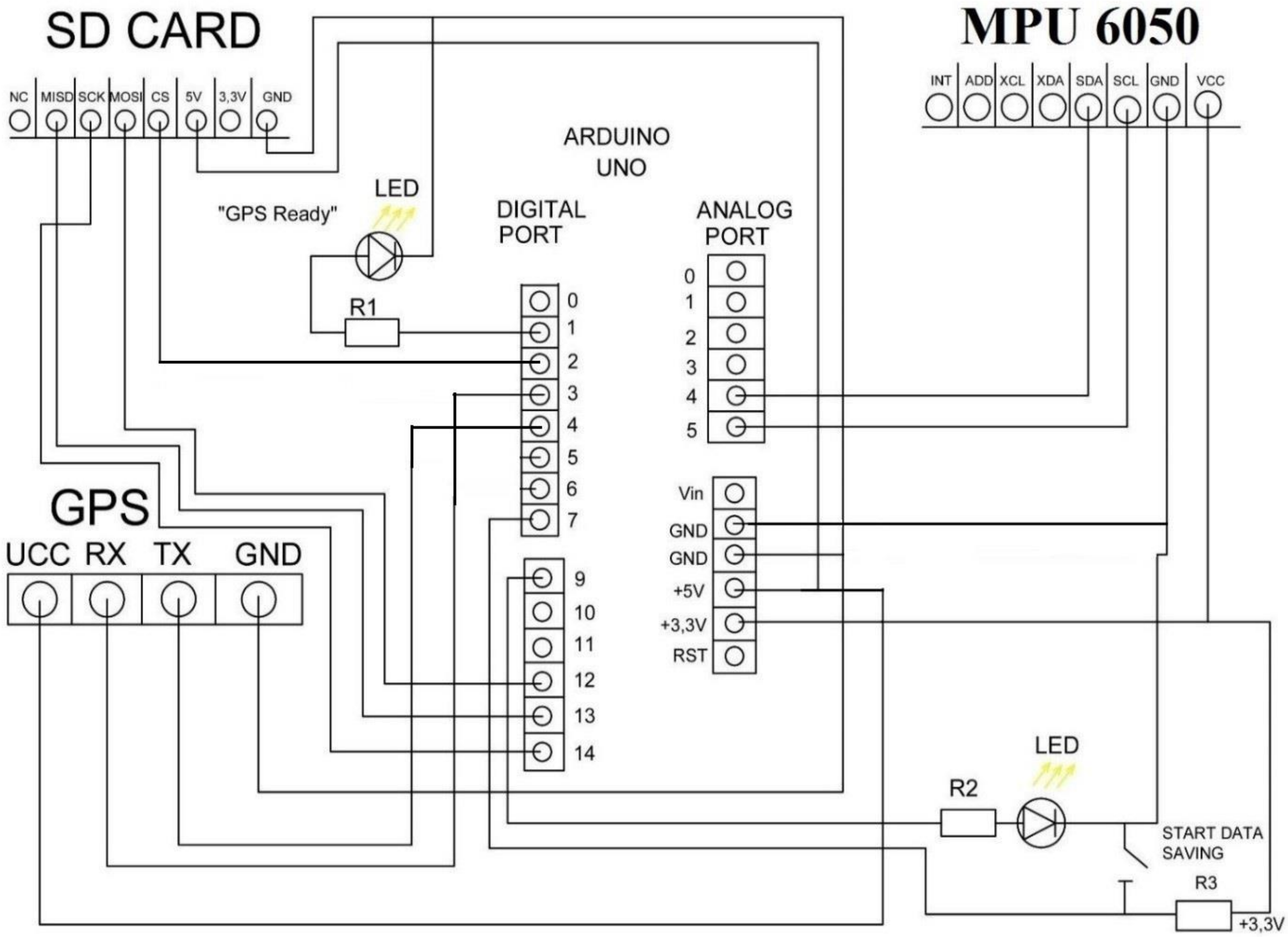
```
 Serial.print(" Time stamp from get GPS signal ");
 Serial.print(millis()/1000 - time_0);
 }
 else {
  Serial.print(" Time stamp from start Arduino ");
 Serial.print(millis()/1000);
 }

 Serial.print(" ----- Acceleration ");
 Serial.print(Acc);
 Serial.print(" ----- GPS altitude ");
 print_float(gps.f_altitude(), TinyGPS::GPS_INVALID_F_ALTITUDE, 7, 2);
 Serial.print(" ----- accurate acceleration ");
 Serial.print(HightOut);
 Serial.print(" ----- test    ");
// Serial.print(DataGPS);

// Serial.println(gps.date.value);

//  Serial.print("Date(ddmmyy): "); Serial.print(date); Serial.print(" Time(hhmmsscc): ");
 //   Serial.print(time/100);
 // Serial.print(" Fix age: "); Serial.print(age); Serial.println("ms.");
/**/
 Serial.println();
 delay(100); // The accelerometer's maximum samples rate is 1kHz

// Indicates working GPS
// Blink - YES data
// Continious light - NO data

    if  ( gps.f_altitude()== Hightcode)
    digitalWrite(LED, HIGH);   // turn the LED on (HIGH is the voltage level)
    else {
      digitalWrite(LED, HIGH);
      delay (50);
      digitalWrite(LED, LOW);
    }

  HightOut = 0.050197*XSNS[k]+0.003324*XSNS[k-1]-0.046872*XSNS[k-
2]+0.002374*XAcc[n]+0.004749*XAcc[n-1]+0.002374*XAcc[n-2]+1.896281*Hight[p-1]-
0.902930*Hight[p-2];

XAcc[n-2] = XAcc[n-1];
XSNS[k-2] = XSNS[k-1];
XAcc[n-1] = XAcc[n];
XSNS[k-1] = XSNS[k];
```

```
XAcc[n] = Acc;
XSNS[k] = gps.f_altitude();
Hight[p-2] = Hight[p-1];
Hight[p-1] = HightOut;

/* SD card*/
if(writeCD == true){
File dataFile = SD.open("log.csv", FILE_WRITE);
if (dataFile)
  {
 //   dataFile.printlngps.get_datetime(&date, &time, &age);
 //   dataFile.println("Date(ddmmyy): ");
 //   dataFile.println(date);
 //   dataFile.println(" Time(hhmmsscc): ");
 time_0 = millis()/1000;
    dataFile.print(millis()/1000 - time_0);
    dataFile.println("sec");
    dataFile.println("Accelerate");
    dataFile.println(Acc);
    dataFile.println(", Hight GPS ");
    dataFile.println(gps.f_altitude(), 3);
    dataFile.println(", correct Hight ");
    dataFile.println(HightOut);
    dataFile.println("\n");
    dataFile.close();
   digitalWrite(LED1,LOW);
   delay(50);
   digitalWrite(LED1,HIGH);
    }
   else Serial.println("Cant open log file");

}
}

 void i2cWrite(uint8_t registerAddress, uint8_t data){
  Wire.beginTransmission(IMUAddress);
  Wire.write(registerAddress);
  Wire.write(data);
  Wire.endTransmission();
}
 uint8_t* i2cRead(uint8_t registerAddress, uint8_t nbytes) {
  uint8_t data[nbytes];
  Wire.beginTransmission(IMUAddress);
  Wire.write(registerAddress);
  Wire.endTransmission(false);
  Wire.requestFrom(IMUAddress, nbytes);
```

```
  for(uint8_t i = 0; i < nbytes; i++)
    data[i] = Wire.read();
  return data;
}

static void smartdelay(unsigned long ms)
{
  unsigned long start = millis();
  do
  {
    while (ssGPS.available())
      gps.encode(ssGPS.read());
  } while (millis() - start < ms);
}

static void print_float(float val, float invalid, int len, int prec)
{
  if (val == invalid)
  {
    while (len-- > 1)
      Serial.print('*');
    Serial.print(' ');
  }
  else
  {
    Serial.print(val, prec);
    int vi = abs((int)val);
    int flen = prec + (val < 0.0 ? 2 : 1); // . and -
    flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;
    for (int i=flen; i<len; ++i)
      Serial.print(' ');
  }
  smartdelay(0);
}

static void print_int(unsigned long val, unsigned long invalid, int len)
{
  char sz[32];
  if (val == invalid)
    strcpy(sz, "*******");
  else
    sprintf(sz, "%ld", val);
  sz[len] = 0;
  for (int i=strlen(sz); i<len; ++i)
    sz[i] = ' ';
  if (len > 0)
```

```
    sz[len-1] = ' ';
  Serial.print(sz);
  smartdelay(0);
}

static void print_date(TinyGPS &gps)
{
  int year;
  byte month, day, hour, minute, second, hundredths;
  unsigned long age;
  unsigned long datego;
  String DataGPS1;
  gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);
  if (age == TinyGPS::GPS_INVALID_AGE)
    Serial.print("********** ******** ");
  else
  {
    char sz[32];
    sprintf(sz, "%02d/%02d/%02d %02d:%02d:%02d ",
        month, day, year, hour, minute, second);
    Serial.print(sz);
    DataGPS1 = sprintf(sz, "%02d/%02d/%02d %02d:%02d:%02d ",
        month, day, year, hour, minute, second);
  }
  print_int(age, TinyGPS::GPS_INVALID_AGE, 5);
  smartdelay(0);
}

static void print_str(const char *str, int len)
{
  int slen = strlen(str);
  for (int i=0; i<len; ++i)
    Serial.print(i<slen ? str[i] : ' ');
  smartdelay(0);
}
```

After testing the developed software, next all the hardware should be connected to Arduino:



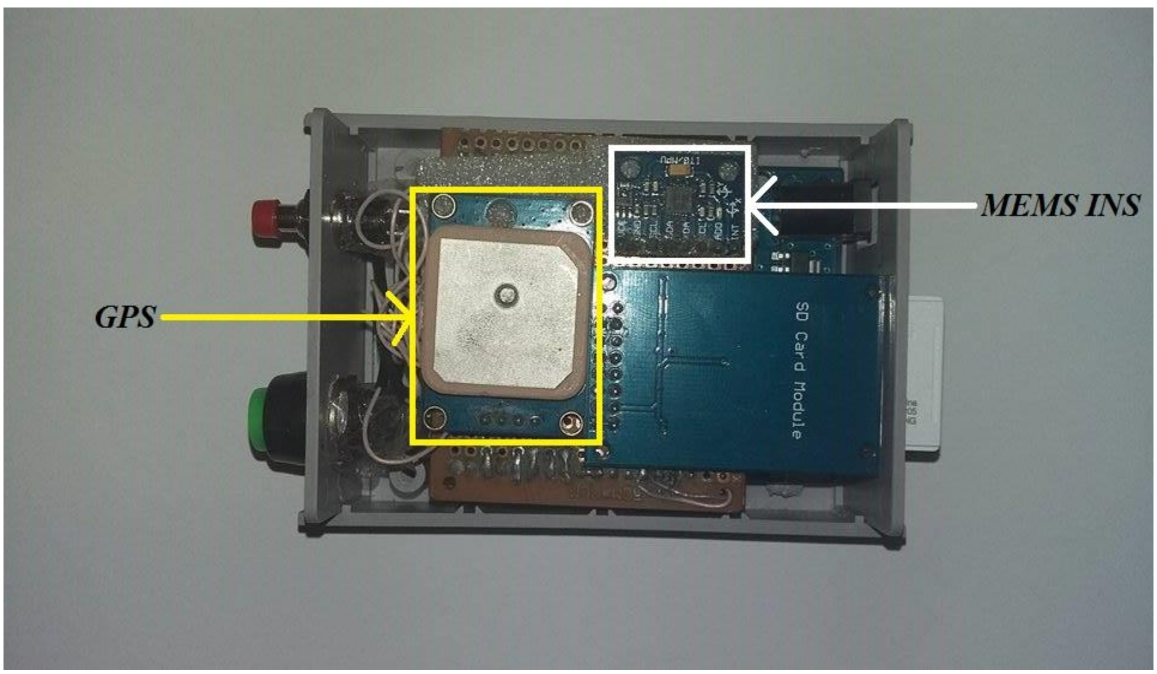


Fig. 7.10 Connecting of all hardware to Arduino

7.4 System Testing:

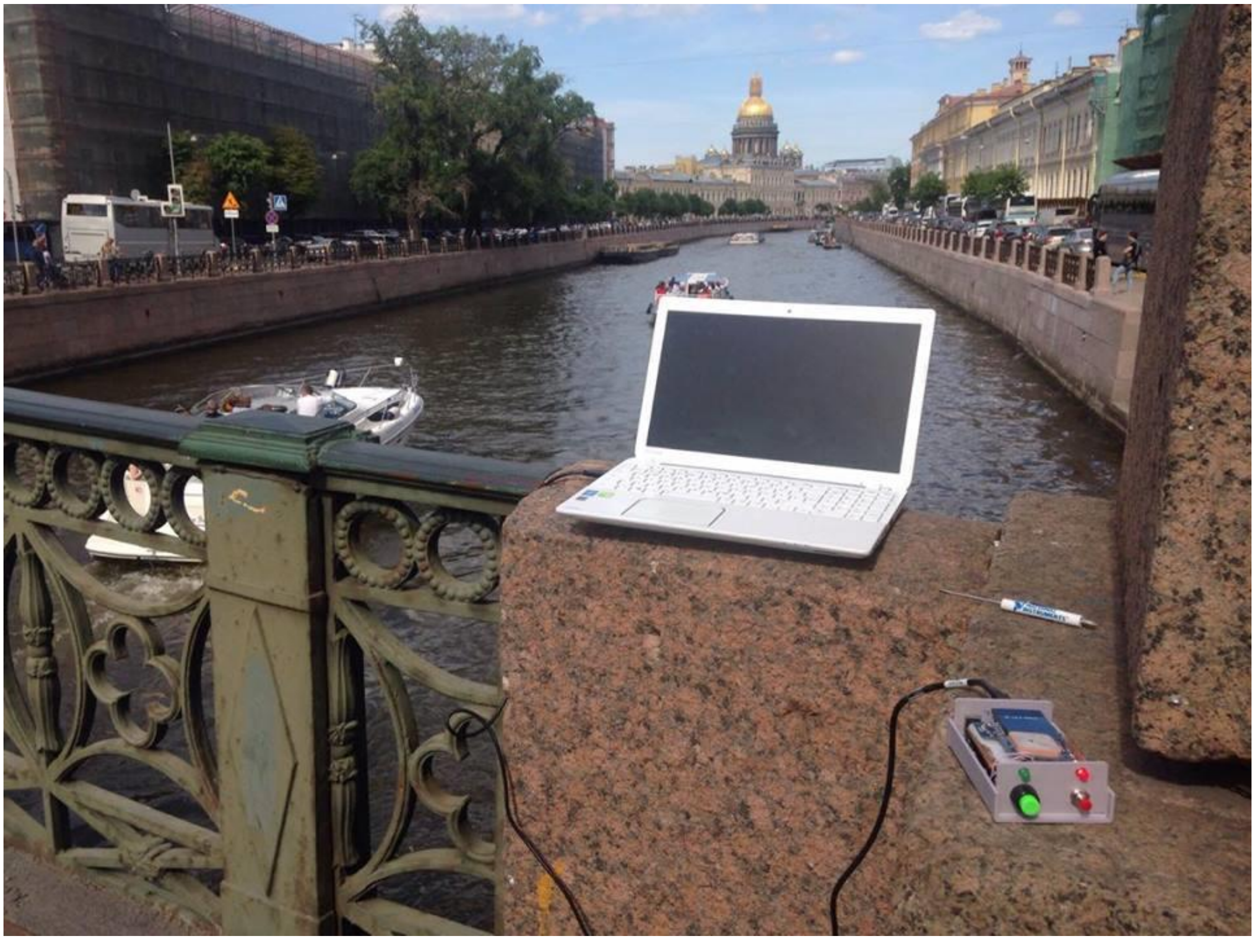


Fig. 7.11 Testing the system near Moyka River

I choose this place, because the altitude here is known, the area is equal to sea level.

Furthermore, I developed a program in LabVIEW to convert the saved data in SD card into graphic:

file (use dialog)

C:\Users\Администратор\Desktop\LOG6.TXT

Array

0	1	-0,27	1E+6	1,00062E+6
1	1	-0,26	1E+6	1,00055E+6
2	1	-0,27	1E+6	1,00048E+6
3	1	-0,28	1E+6	1,00041E+6
4	2	-0,25	1E+6	1,00035E+6
5	0	-0,35	1E+6	999968
6	0	-0,35	1E+6	999970
7	0	-0,36	1E+6	999972
8	0	-0,39	1E+6	999974
9	0	-0,35	1E+6	999974
10	0	-0,34	1E+6	999975
11	1	-0,32	1E+6	999979
12	1	-0,31	1E+6	999979
13	0	-0,32	1E+6	999979
14	0	-0,34	1E+6	999979
15	1	-0,42	1E+6	999979
16	1	-0,36	1E+6	999979
17	0	-0,33	1E+6	999979
18	0	-0,34	1E+6	999979
19	0	-0,35	1E+6	999979
20	1	-0,36	1E+6	999979
21	1	-0,36	1E+6	999979

length

13760

Fig. 7.12 transmission of SD card's readings into graphic

As expected, after filtering the sensors reading I got an altitude equal to zero meter:

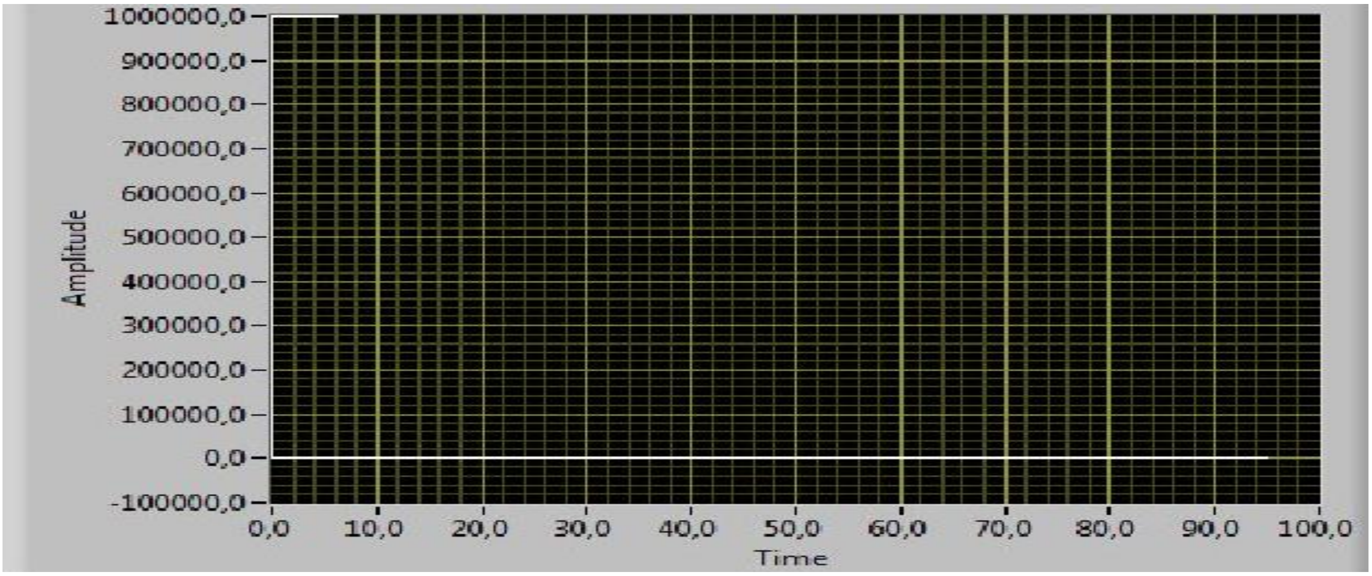


Fig. 7.13 GPS reading

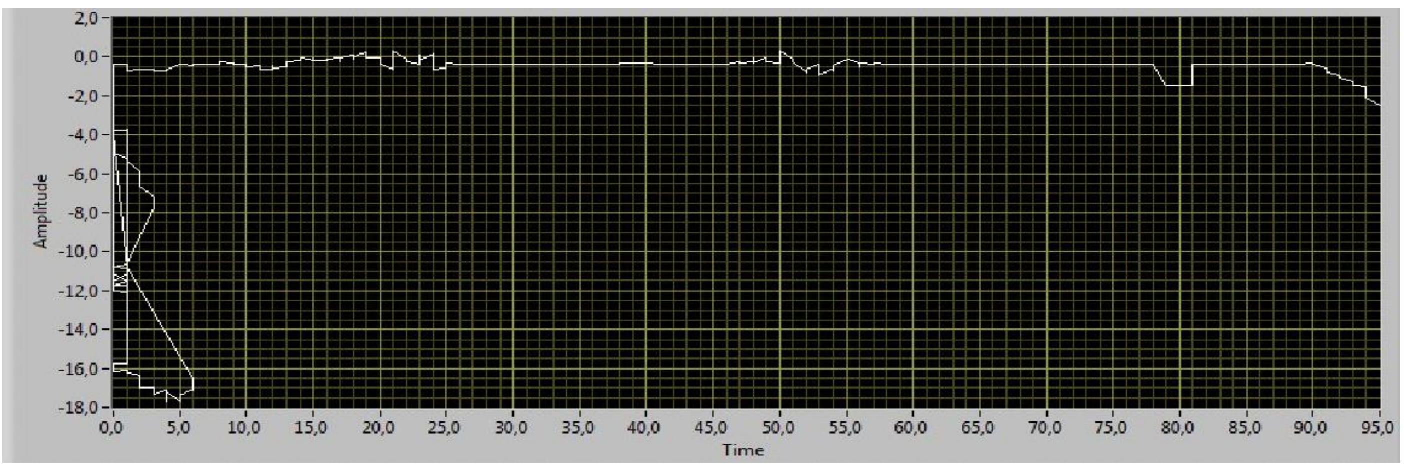


Fig. 7.14 Accelerometer reading

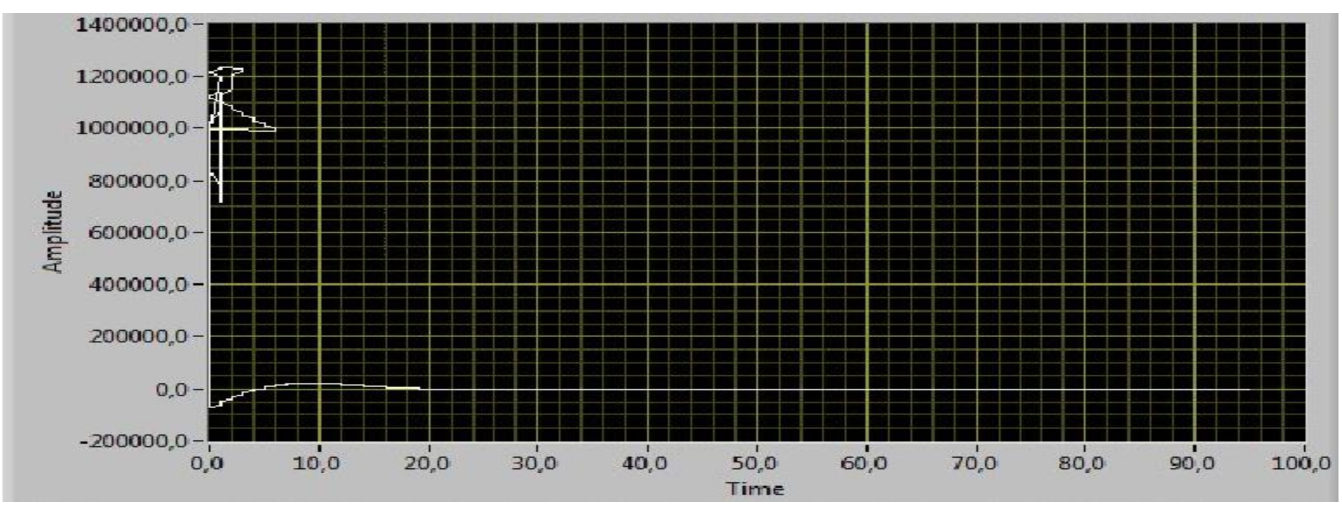


Fig. 7.15 Filter reading

7.5 After coming back to Germany, I tested my system in a FI-156 aircraft:

Fig. 7.16 Fieseler Fi 156

Fig. 7.17 installation in the aircraft

During the installation in the aircraft it must be ensured that the system needs to be bonded to a stable platform, otherwise in vibrating platform, the Z-axis of accelerometer will not stay 100% vertically and thus measure the gravitational acceleration which leads to fault of measurement.

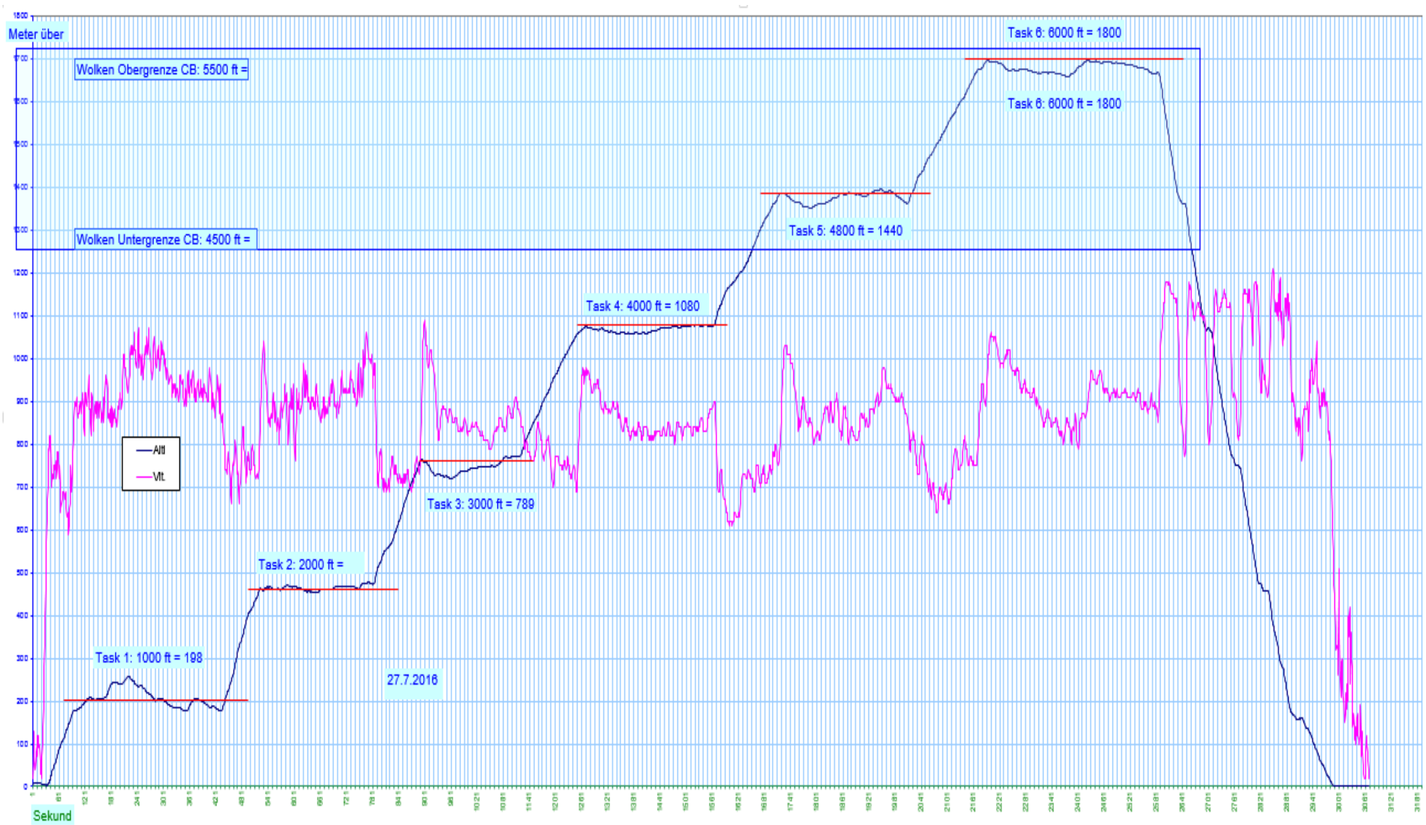


Fig. 7.18 flown altitude measured by Bräuniger IQ-Competition GPS

Fig. 7.19 Bräuniger IQ-Competition GPS

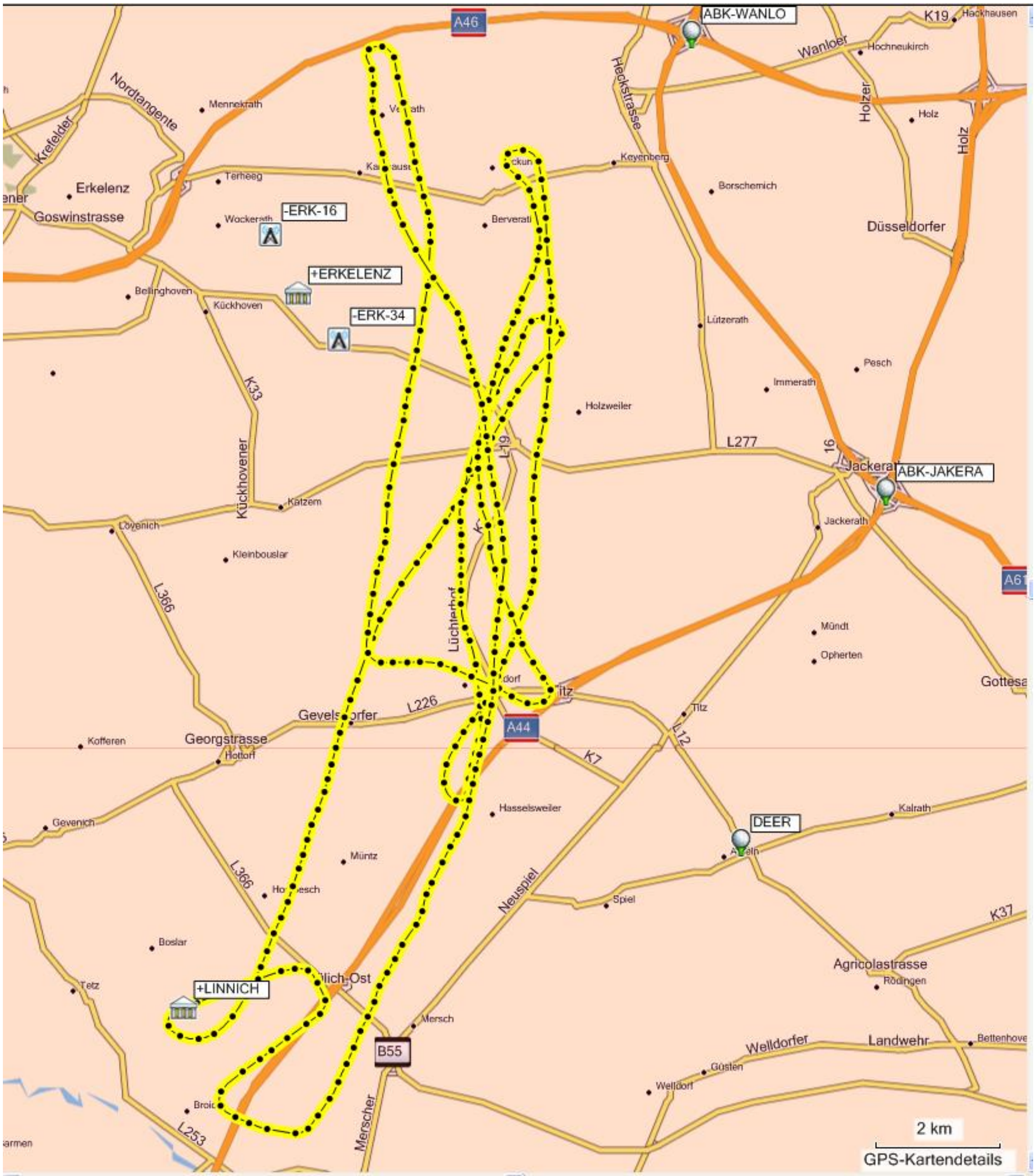


Fig. 7.20 flown route

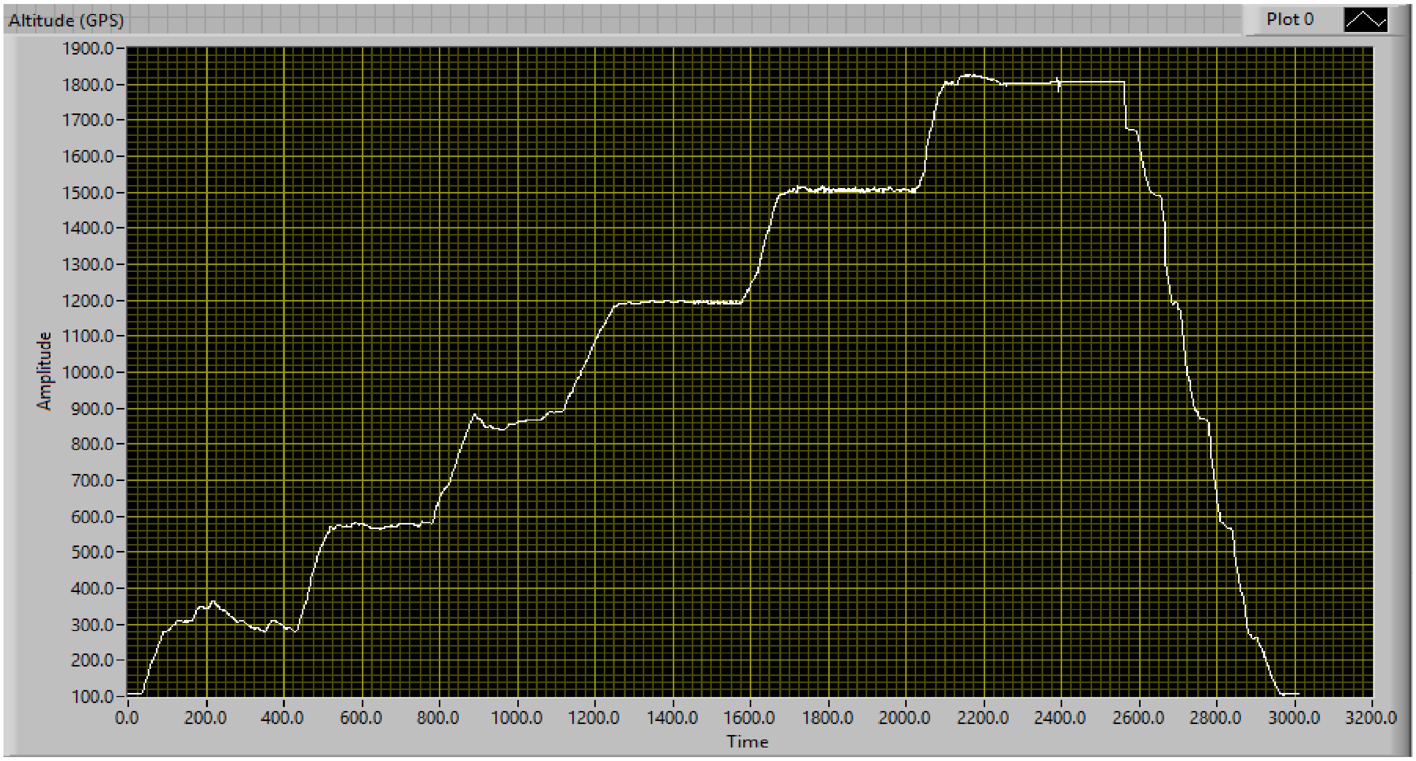


Fig. 7.21 flown altitude measured by GY-NEO6MV2 GPS

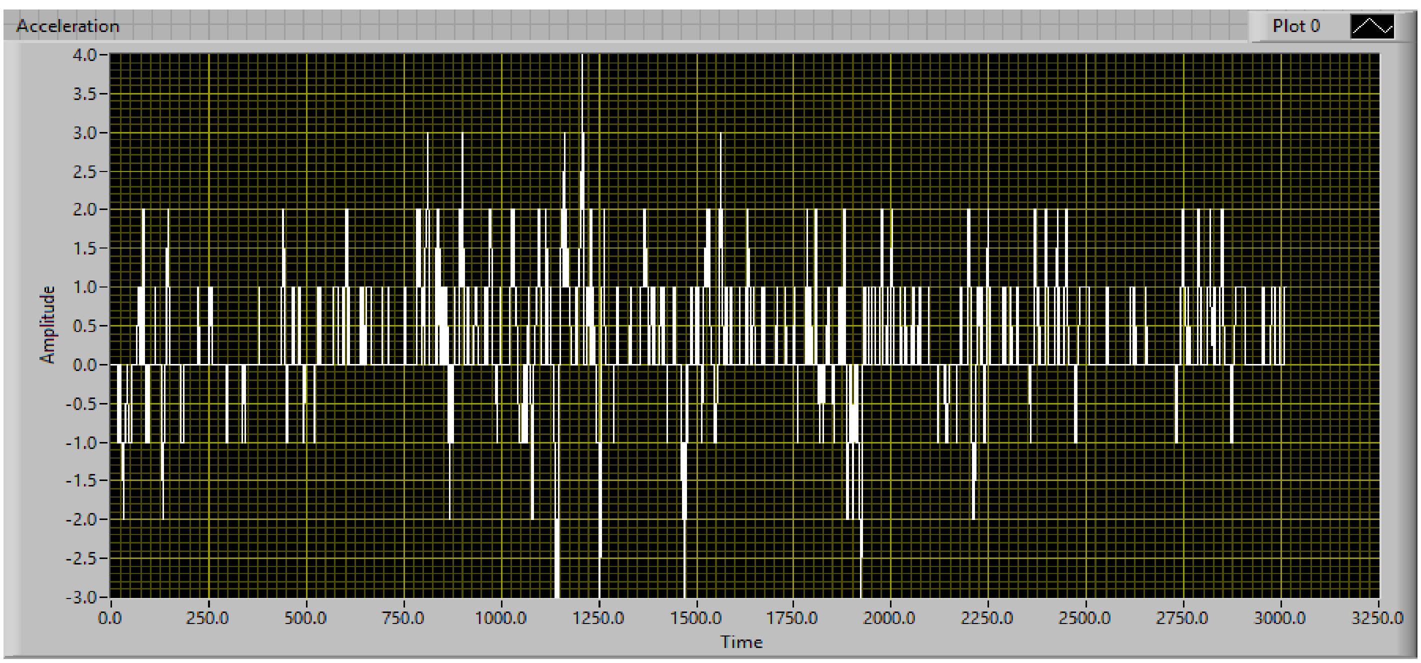


Fig. 7.22 Acceleration measured by MPU6050

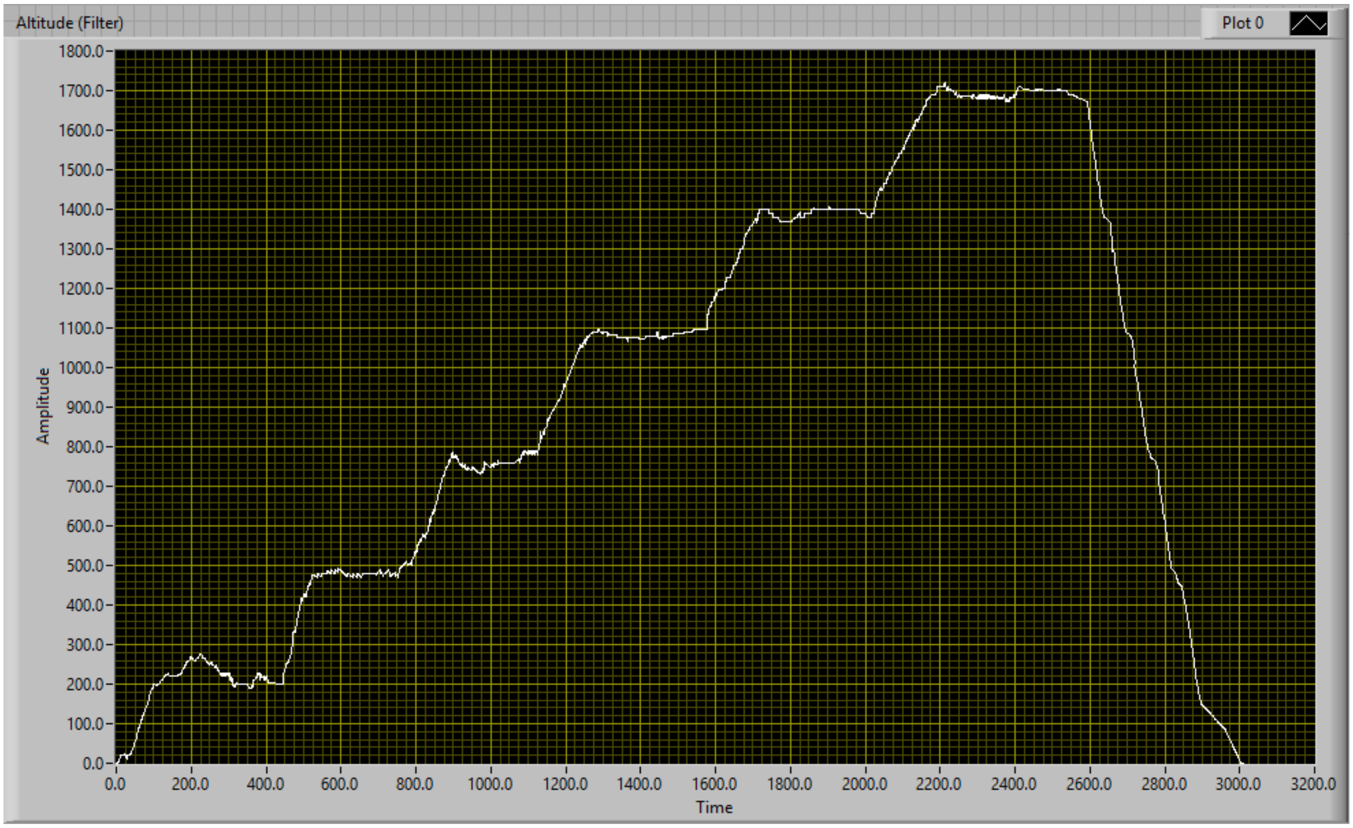


Fig. 7.23 flown altitude after filtering

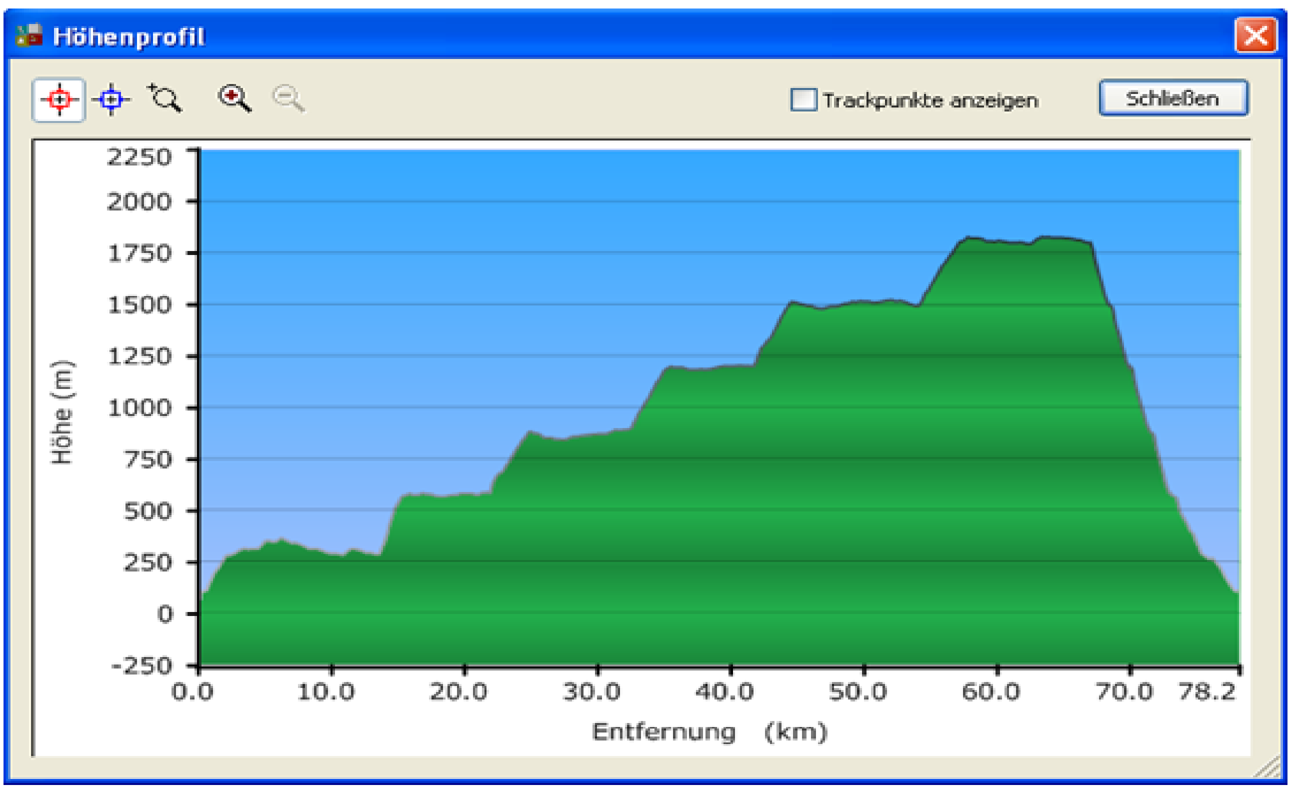


Fig. 7.24 flown altitude (m) vs distance (km) measured by ThinkNavi T7

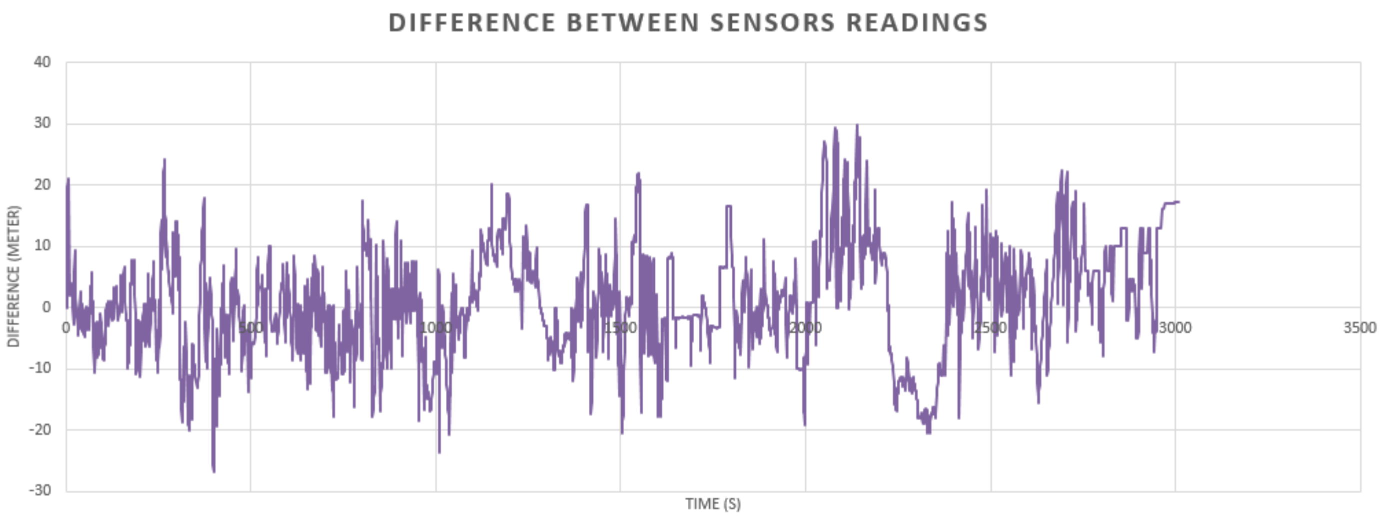


Fig. 7.25 difference between my sensor and Bräuniger's sensor

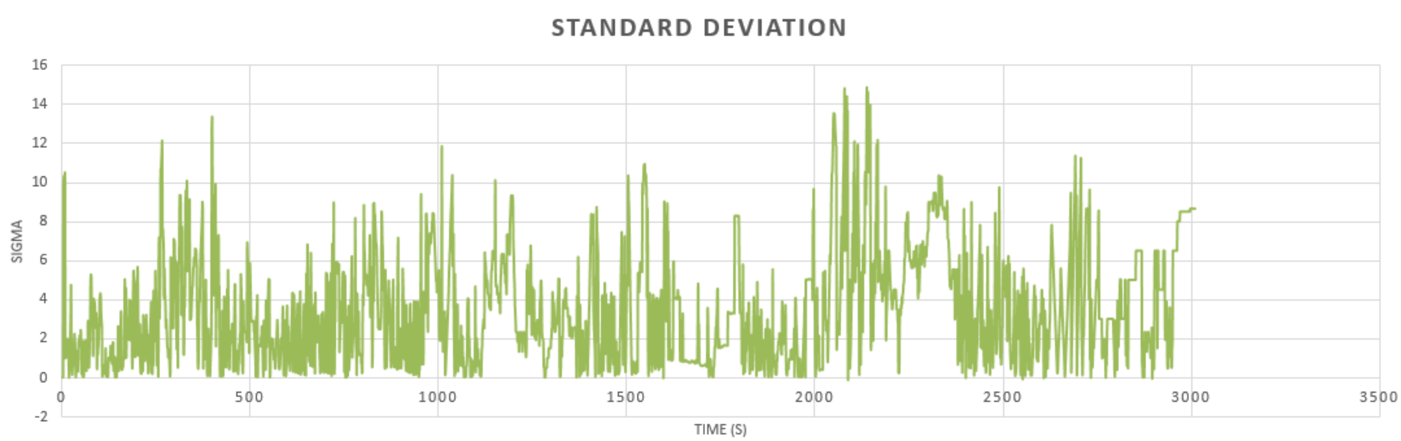


Fig. 7.26 standard deviation between two sensors

The maximum average of difference between the sensors data is ~ 15m, during Take-off and landing is my system even more accurate, it gave more reasonable results than Bräuniger's sensor.

In order to make a statement about the accuracy of a system, it requires a reference that is at least more accurate than the system to be investigated, thus no absolute indication of the accuracy of the developed GPS / Accelerometer system is possible within the available references. Also, for an optimization of the developed filter, the reference systems are insufficient. To this end, other methods must be used to verify if the goal (maximum error of 3 till 5 meter) is reached. Nevertheless, it can be deduced from the results that the system's algorithms process the real sensor data correctly and the accuracy of the reference systems is achieved.

Conclusion

Back to the first question that I asked, which method should be used to control the altitude of UAV?

The simulation results and experimental verification of the developed algorithms confirm the effectiveness of the proposed method to give a good estimation of UAV altitude even when the sensors noises increase.

Recommendation

In order to get a full Navigation, the use of INS can be extended to include all the axes (not only the vertical channel).

[1] Reg Austin, Aircraft Systems UAVs Design, Development and Deployment, 2010, P 37/41.

[2] Reg Austin, Aircraft Systems UAVs Design, Development and Deployment, 2010, P 9.

[3] Reg Austin, Aircraft Systems UAVs Design, Development and Deployment, 2010, P 202.

[4] Hugo José Dias Lopes, Attitude Determination of Highly Dynamic Fixed-wing UAVs, 2011, P11

[5] Bernhard, Lichtenegger, Herbert, Wasle, Elmar GNSS – Global Navigation Satellite Systems,Hofmann-Wellenhof, , 2008, P316

[6] Majd Alshawa, Contribution à la cartographie mobile: dévelopment et caractérisation d'un systeme basé sur un scanner laser terrestre, 2010, P 34

[7] Hugo José Dias Lopes, Attitude Determination of Highly Dynamic Fixed-wing UAVs, 2011, P17

[8] Maria de Fátima Alves Nunes Bento, Development and Validation of an IMU/GPS/Galileo Integration Navigation System for UAV, 2013. P 27

[9] Fundamentals of inertial navigation, satellite-based positioning and their integration, Aboelmagd Noueldin , Tashfeen Karamt , Jacques Georgy.-2013-P-4

[10] MEMS Inertial Sensors Technology, Felix Renaut, 2013, P.11

[11] Hugo José Dias Lopes, Attitude Determination of Highly Dynamic Fixed-wing UAVs, 2011, P13

[12] aboutcivil.org/sources-of-errors-in-gps.html

[13] Fundamentals of inertial navigation, satellite-based positioning and their integration, Aboelmagd Noueldin , Tashfeen Karamt , Jacques Georgy.-2013-P-249

[14] An engineering approach to optimal control and estimation theory, George Siouris , Air Force institute of technology , Wright-Patterson AFB, Ohio.-1995-P.38

[15] Alexander Nebylov. Ensuring Control Accuracy. Lecture notes in control and information sciences. Springer-Verlag, Heidelberg, Germany, 2004-P.98

[16] Alexander Nebylov. Ensuring Control Accuracy. Lecture notes in control and information sciences. Springer-Verlag, Heidelberg, Germany, 2004-P.129

[17] Alexander Nebylov. Ensuring Control Accuracy. Lecture notes in control and information sciences. Springer-Verlag, Heidelberg, Germany, 2004-P.204

[18] Signals and Systems Using MATLAB© Luis F. Chaparro Department of Electrical and Computer Engineering University of Pittsburgh Pittsburgh, PA, USA-2015-P785

[19] Tran Duc-Tan, Paul Fortier, Huu-Tue Huynh, Design, Simulation, and Performance Analysis of an INS/GPS System using Parallel Kalman Filters Structure, 2011, P. 91