TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Altitude Control System for Indoor Airship

Davide La Croce

Examiners: Dr.Sc.Tech. Jussi Collin
and Professor Robert Piché
Examiner and topic approved in the
Computing and Electrical Engineering
Faculty Council meeting on January
15th, 2014

ABSTRACT

Tampere University of Technology
Master's degree programme in Information Technology
La Croce, Davide: Altitude Control System for Indoor Airship
Master of Science Thesis August 2014
Major: Positioning and Navigation
Examiners: Dr.Sc.Tech. Jussi Collin and Professor Robert Piché
Keywords: airship, indoor, control system, real-time, embedded, altitude tracking, altitude estimation, altitude control, inertial measurement unit, Kalman filter, sensor fusion, fuzzy logic.

The indoor airship is designed to estimate direction and altitude using dead reckoning techniques. Sensors such as gyroscopes and accelerometers provide measurements affected by biases and additive noise. Degradation of accuracy produced by these errors is corrected using independent measurements from ultrasonic range finder, barometer and magnetometer that are supplied to the Kalman filter.

A real-time fuzzy logic controller uses the statistical best estimate to perform a feedback control loop. As result the airship is capable of maintaining the altitude in a range of 5cm from the set point.

PREFACE

This thesis is the result of a work started in the Pervasive Computing (PC) department and continued in the Automation Science and Engineering department (ASE). I proposed the topic and I want to thank my supervisors Jussi Collin and Robert Piché for their support and trust. They understood my enthusiasm for the project and they both allowed me to pursue my goal.

A special thank goes to Pavel Davidson for his guidance in the practical use of the Kalman Filter and inertial sensors.

I needed suggestions and discussions with many other people experienced in different fields. There is no space to list them all but a special thank goes to Juha Koljonen and Timo Pihlström that helped me respectively with the 3d print of the gondola and its electronics.

My parents, my brother, and my friends know how moody, foolish and obstinate I can be. I want to thank them all for their patience, support and motivation during my Master's degree period.

I'm available to discuss my results and to provide further details about my implementation. My email address is davide.lacroce89@gmail.com.

Tampere, August 2014

Davide La Croce

CONTENTS

Terms and definitions

UAV	Unmanned aerial vehicle
INS	Inertial navigation system
PVA	Position, velocity and attitude
IMU	Inertial measurement unit
COR	Center of rotation
COG	Center of gravity
MCU	Microcontroller unit
IEEE	Institute of electrical and electronics engineers
STP	Standard temperature and pressure (273.15 K and 100 kPa)
ISA	International standard atmosphere
MEMS	Micro electro-mechanical systems
SI	International system of units
PWM	Pulse width modulation
I2C	Inter-integrated circuit
UsRF	Ultrasonic range finder
AWGN	Additive white gaussian noise
SISO	Single-in-single-out
MIMO	Multiple-in-multiple-out
PCB	Printed circuit board
SMT	Surface mount technology
LSB	Least significant bit

Constant factors

$T_0 = 288.15\ \ [K]$	Sea level standard temperature
$p_0 = 101326\ [Pa]$	Sea level standard atmospheric pressure
$\rho_{air} = 1.2754\ [{}^{kg}/_{m^3}]$	Air density at STP
$\rho_h = 0.179\ [{}^{kg}/_{m^3}]$	Helium density at STP
$g_0 = 9.780326\ [{}^{m}/_{s^2}]$	Acceleration constant due to gravity
$c_1 = 44330.77$	Empirical coefficient for altitude calculation
$c_2 = 0.1902632$	Empirical coefficient for altitude calculation
$\Omega = 7.292115\ 10^{-5}\ [{}^{rad}/_{s}]$	Earth rotation rate
$\lambda_1 = 0.00193185138639$	Empirical coefficient for gravity calculation

$\lambda_2 = 0.00669437999013$	Empirical coefficient for gravity calculation

List of symbols

F	State transition matrix of a continuous linear dynamic system
Φ	State transition matrix of a discrete linear dynamic system
x	State vector of a linear dynamic system
z	Measurements vector
Q	Covariance matrix of process noise in the continuous system state dynamics
Γ	Covariance matrix of process noise in the discrete system state dynamics
R	Covariance matrix of measurements uncertainty
Q	Covariance matrix of process noise in the system state dynamics
P	Covariance matrix of state estimation uncertainty
K	Kalman gain matrix
H	Measurements sensitivity matrix
u	Control input state vector
B	Control input vector for continuous linear dynamic system
T	Control input vector for discrete linear dynamic system
i	Innovation vector
C_{from}^{to}	Rotation matrix from-to frame
l	Latitude
f	Specific force
b	Bias factor
s	Scale factor
k	Sensitivity
M	Misalignment error
w	Additive noise

1. Introduction

Centuries after the temerarious explorers, the navigation researches are focused on UAVs (Unmanned aerial vehicles). These are autonomous flying machines that can follow waypoints under the instruction of human operators.

Unfortunately, it is clear that these airplanes are meant to carry weapons. Autonomous vehicles can operate in many applications much more useful and toward the real needs of the people such as: surveillance, assistance and environment protection. Depending on the needs, a payload can be brought along in order to accomplish different tasks such as: collection of environmental data, capture of images and so on.

At the time this thesis is written, there is an extensive attention to the multi-copters drones. Science and engineering follow the fashion of the moment, so it seems that these particular drones can be adopted for every task.

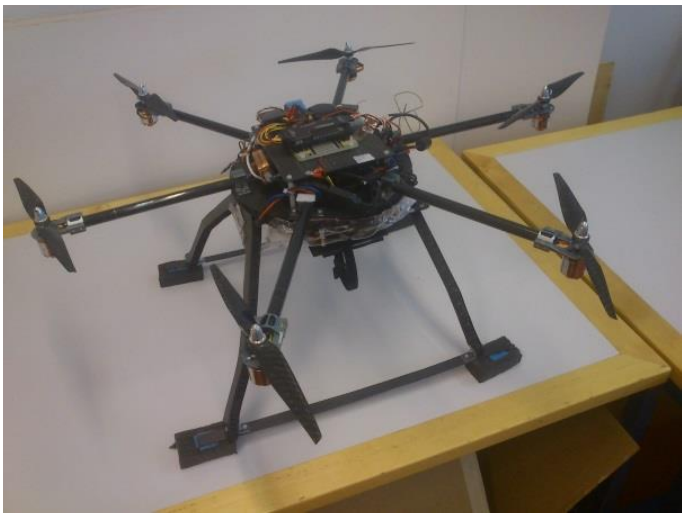


Figure 1.1. *TUT's remotely controlled hexa-copter*

However, these multi-copters have many disadvantages. Just to summarise some, they are: noisy, turbulent and unstable by nature. Despite the big capacity of the battery pack, their autonomy is very limited. Nobody would like to stay closer than a meter while they fly because the propellers are usually rotating at thousands rpm and they can cause injuries and accidents. The realization of an automatic robot for nobles purpose leads to some requirements. First of all the vehicle must never hurt or bring the people to a state of stress or be a threat to the individual safety. Secondary, many important applications for these robots take place indoors. The mobility and the insurmountable obstacles of these special robots must be taken into account. There is the need of a mean that is unlikely obstructed along his progress. It is important to answer all these practical questions. Is there a vehicle that reduces all these issues?

1.1. Airships

The word airship brings often memories back in time to the massive Hindenburg. The memories start focusing on the dramatic accident occurred in 1937, when the Zeppelin took fire and killed the entire crew and passengers.

The image of the fire convinced the mankind that flammable gasses can be really dangerous. Nowadays no engineer would consider the use of cheap and unstable hydrogen to fill a balloon. Instead, helium can generate a similar lifting force, having the advantage to be an inert gas. Inert elements are not subject to chemical reactions and cannot ignite.

For a long time airships have been used to carry passengers and cargo, but only from 1980' they became a really common mean to advertise products and companies. Even a light airship has to be filled with a big amount of gas, therefore the outer surface can be covered with logos and advertising messages. The smaller of them are called *indoor airships* because they are too light and delicate to fly outdoor. They have a balloon that contains up to 3 $m^3$of gas, for this reason they are reasonably small and quite agile.

The airships without a rigid frame are called *blimps*. The pressure of the helium gives them the oval shape.

It is important to underline the psychological reaction that people have during the sight of one of these vehicles. Everybody would agree that their movements are smooth, gentle and never aggressive. Sometimes people feel the need to touch the balloon in the attempt to remind their childhood.

Airships use a gas lighter than the air to generate a lifting force. The lifting force is proportional to the volume of the body and the density of the fluid. In this case it's the difference of density between the gasses that produced a lifting force. Large airship can generate more lift and carry a heavy payload.

On the other hand increasing the volume of the envelope implies an increment of the mass (weight of gas and envelope) and the frontal surface that produces a greater aerodynamic drag.

$$F_{lift} = (\rho_{air} - \rho_{gas})V \qquad (1.1)$$

The equation 1.1, derived from Archimede's principle, returns a lifting force (F) expressed in Kilograms. The result is supposed to be in Newton (SI), but it is a common procedure to keep it in unit of mass, because it shows directly the amount of payload that airships can carry. In the equation 1.1 the variable ρ_{air} refers to the air density, ρ_{gas} to the helium density and V to the volume.

For simplicity we are assuming a temperature of T_0 with a pressure at the sea level of p_0.

This is just a first approximation model that uses the main factors to provide a good estimate of the lift. All the parameters are function of temperature and atmospheric pressure that can affect the lifting force.

Assuming the pressure and density of the helium in the balloon as constant, the air density decreases with the altitude. It leads to a smaller difference of density between the gas and the air at high altitude, consequently a smaller lifting force is produced going on with the altitude. This is the reason why every balloon reaches the equilibrium at a certain altitude. The best meteorological balloons cannot go higher that 36km, where the air is replaced by a light vacuum.

1.2. Blimps

Nyton is the name chosen for our blimp and he takes inspiration from the design of some previous projects. There are mainly two projects called YARB (Yet another robotic blimp,) [1] and Blipduino (Blimp + Arduino) [2].

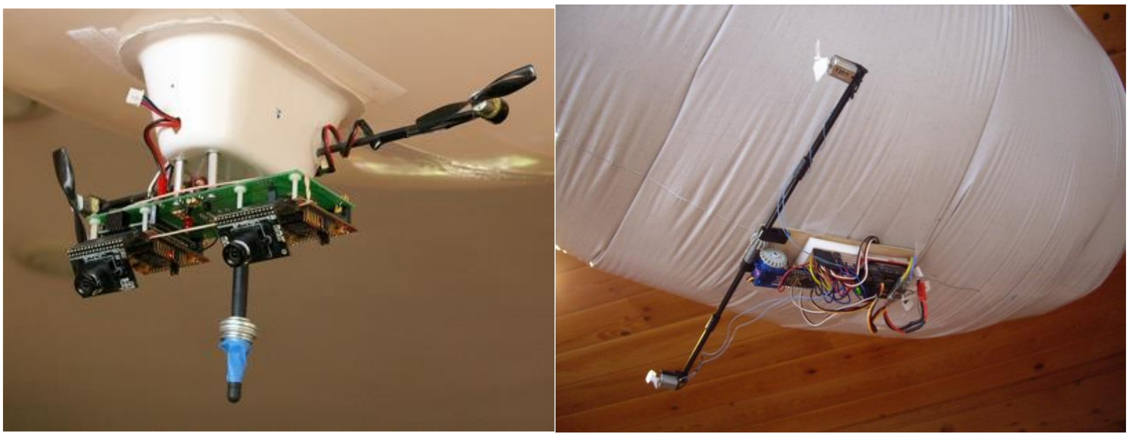


Figure 1.2. *YARB (left) and Blimpduino (right).*

They are both robotic blimps designed by hobbyists in their free time but with different features and hardware. YARB is fully remote controlled and it carries two small cameras that transmit a stereo image in real-time. It can be piloted from a mobile phone and the stereo images are the classical red-green that are visible using special glasses made of paper. Blimpduino has no cameras on-board but carries an ultrasonic range finder that is used to maintain the altitude. This is proposed as open platform for airships of every nature. Both projects do not have improvements from several years but they constitute a valuable example of what can be achieved.

By increasing the budget and the amount of work, it is possible to achieve more scientific and documented results. At the University of Freiburg (Germany) the researchers have been working on a very complex blimp. It aimed to be a platform for position tracking using IMU, odometer and range finders. The paper "Efficient Probabilistic Localization for Autonomous Indoor Airships using Sonar, Air Flow, and IMU Sensors" by Jörg Muller and Wolfram Burgard [3] describes in details their research.

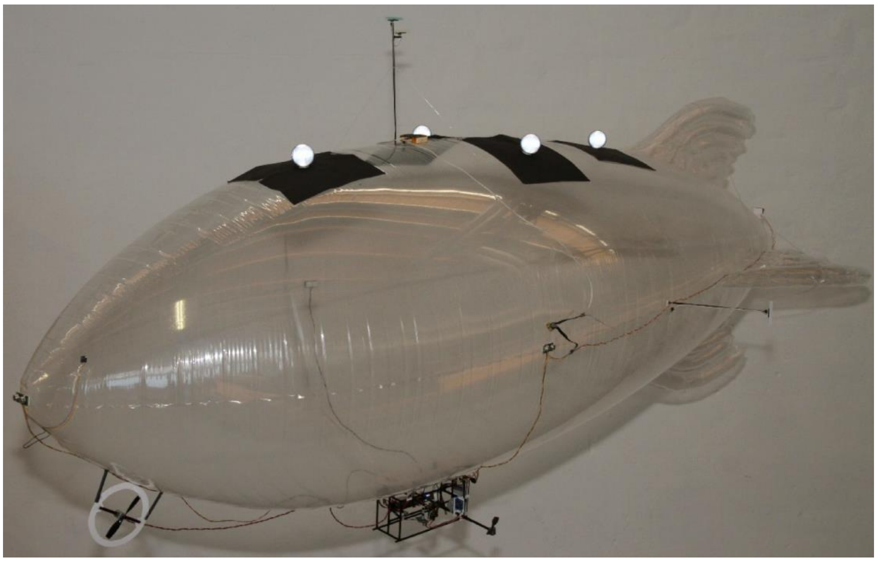


Figure 1.3. *Indoor airship by University of Freiburg*

All these blimps have a similar degign in terms of motors and tilting mechanism because all of them are meant to be manually controlled.

Our blimp can maintain a reference altitude in a range of 5 cms from the reference. The accelerometers are used to estimate the altitude and, as explained in the following chapters, the noise of the inertial sensors produces a massive error due to the random walk phenomena. This dramatic degradation of accuracy can be limited using measurements from ultrasonic range finder and barometer that are supplied to the Kalman filter. The heading is estimated with an accuracy of 5 degrees using of gyroscope and magnetometer. These measurements are fused through the use of an auxiliary Kalman filter. Magnetometers are dramatically influenced by presence of iron or ferromagnetic materials. As result the accuracy of the heading estimation may deteriorate in an unpredictable way.

The current version of the vehicle is able to perform in a quite accurate way, only few of the functions that are planned for further developments. The control loop for the heading angle has been simulated but not implemented yet. There are currently no aiding sensors related to the horizontal plane that can work together with the inertial measurements. Because of this, the tracking of the position on the horizontal plane is not possible yet.

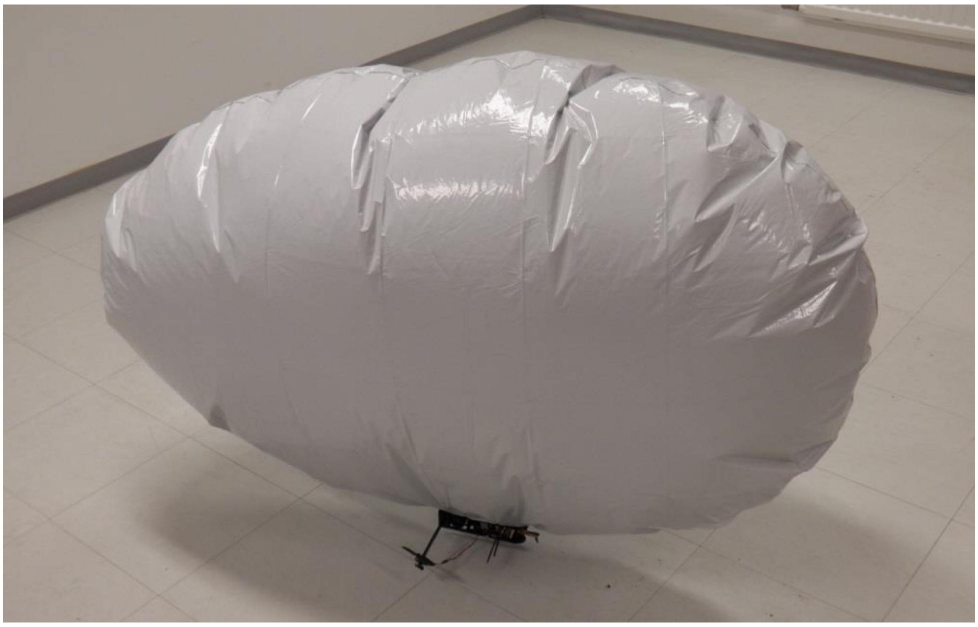


Figure 1.4. *Nyton airship during initialization phase*

In the following chapters, you will notice the use of a specific terminology as specified in the IEEE standard nomenclature for inertial measurement units (IEEE Standard for Inertial Sensor Terminology IEEE Std 528-2001) [4].

Every measurement will be reported with its absolute uncertainty and in every case in the SI units.

Considering the practical goal of the thesis, the following chapters are vastly dedicated to the correct readings of the sensors and about the implementation of the vehicle.

1.3. Thesis outline

The rest of the thesis is structured as follows:

Chapter 2

Theoretical background about the formulation of the discrete KF, tridimensional frames and standard models for gravity, atmosphere and other physical phenomena. Fuzzy logic is briefly explained. The principles behind the several sensors used in the project. The MEMS technology and its limitation. Error model and main factors.

Chapter 3

Inertial system for indirect estimation of position and attitude. The way it is initialized and how the KF can improve its performance. Reference frames and rotations.

Chapter 4

Model of kinetics and KFs for altitude and direction estimation.

Chapter 5

Introduction to fuzzy logic controllers and definition of the altitude controller.

Chapter 6

Hardware and software implementation of the blimp. The payload distribution and the real-time computations.

Chapter 7

Simulations and empirical results. Methods to ensure the coherency of data, practical issues and performance of the control loop.

Chapter 8

Conclusions and future improvements.

Chapter 9

Books, datasheets and other references.

2. Theoretical background

2.1. Earth magnetic field and magnetometers

The Earth magnetic field is generated because of the rotation of liquid metals contained in its core. The globe can be seen as a gigantic magnet with a North and South magnetic pole with intensity in the range from 0.25 to 0.65 Gauss. These poles are subject to a continuous movement that is currently unpredictable by the science. NOAA reports that in 2010 the magnetic north was at 84.97°N, 132.35° W and the South magnetic pole at 64.42°S, 137.34°E [5]. At an average of 450000 years the poles are subject to an inversion. Navigation systems yet use the compass indication to navigate and considering the continuous displacement of the magnetic poles, they have to be updated so to know the current relation between the geographical and magnetic North.

Magnetometers can measure the magnetic field produced by the Earth or any ferromagnet in the vicinity. For this reason, they can be used to determine the direction of the magnetic north or the presence of metals (metal detectors). In other words, they perform the same function of a classical compass but nowadays they are fit into an electronic chip that contains multiple magnetometers placed on three axes. If the two main axes are parallel to the local level, they measure the projection of the magnetic field and the angle of heading given by the equation 2.1 [6, p.7].

$$\psi_m = \frac{180}{\pi} atan2\left(B_x, B_y\right)[degs] \tag{2.1}$$

The heading angle ψ corresponds to the compass angle, while the argument B refers to the measured magnetic field on different axes. The tilt compensation is important to resolve the magnetometers reading to a fixed frame that does not depend on the tilt of the magnetometer.

MEMS based magnetometers are often tilt compensated or provide an accelerometer built in the same package to implement the compensation. In this way independently by the attitude, the compass will return the direction of the magnetic north. The readings of the magnetometers can be easily influenced by steady or moving metals and ferromagnets. In those cases the magnitude of the field grows above 1 Gauss and it is possible to notice the interference.

There are techniques to mitigate the effect of constant sources of interference in indoor environment. The main source of magnetometers error is due to magnetic disturbances (which are significantly different from the wide band noise) that differ from an environment to another.

2.2. Earth’s rotation, gyroscopes and Coriolis effect

The Earth makes a complete revolution around its axis in about 24h.
This defines the length of the day and, as stated in the WGS84 document [7], this value of the angular rate is $15.04\ [deg/h]$. Assuming to have the 3axis gyroscope (not affected by noise, defects or biases) that is aligned with the z-axis perpendicular to the ground, we would measure the projection of the earth rate only on two axes.

$$w = \begin{pmatrix} 0 \\ \Omega\cos(l) \\ \Omega\sin(l) \end{pmatrix} \qquad (2.2)$$

In the equation 2.2 [8] the variable w denotes the measurement vector while Ω and l are respectively the Earth’ rotation rate and the latitude. The Earth rate is assumed to be constant even though the earth is subject to ice forming, seasonal changes and other phenomena that continuously change this value.
The term gyroscope is used to indicate a machine that uses the gyroscopic effect or a device that is able to measure the angular rate experienced about an axis. The early mechanical rate gyros were gyroscopes made with a cylindrical spinning mass kept at a constant RPM with an electric motor.

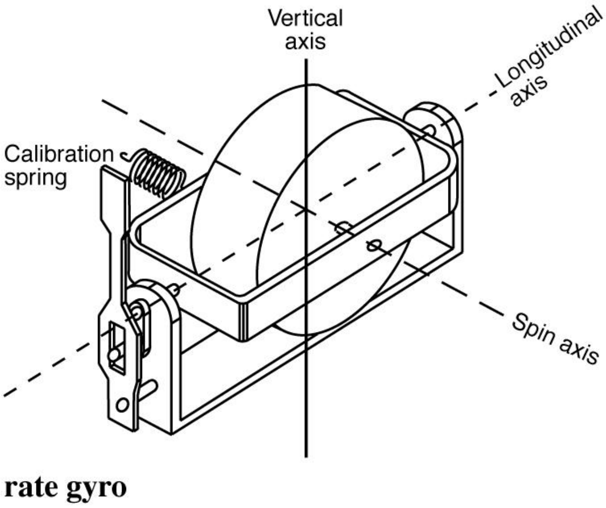


Figure 2.1. *Mechanical structure of a rate gyro* (*Groves, Paul D., 2008, Principles of GNSS, Inertial, and Multisensor Integrated Navigation System) [8].*

Because of the *conservation of angular momentum law*, the spinning mass tends to maintain its direction and therefore it becomes an intuitive mechanism to maintain a reference attitude. When a longitudinal torque is applied, the precession of the spinning mass is proportional to the angular rate on the longitudinal axis. In the early rate gyros the measurements were based on this effect. The precession was compensated with a magnetic field to maintain the pickoff. The current needed to maintain the pickoff was directly proportional to the angular rate.

There are many categories of rate gyros and different technologies but it is not purpose of this document to describe them all. Since we are using the gyroscopes on the planet Earth, it is important to consider the effect of its rotation. Very accurate gyroscopes (tactical and navigation grade) can easily measure the Earth rate. This effect is used for north finding and initial alignment of navigation systems. In case of low grade MEMS gyroscopes it is not possible to have such precise measurements because the Earth rate lies inside the measurement noise of every low grade MEMS gyroscope currently available.

When an object is moving with a velocity V on a straight line while it is on a rotating reference frame at a constant speed ω, then it has and acceleration a_c that is perpendicular to the velocity vector V [9].

$$\overrightarrow{a_c} = 2\omega \wedge \vec{V} = 2\omega V\vec{j} \qquad (2.3)$$

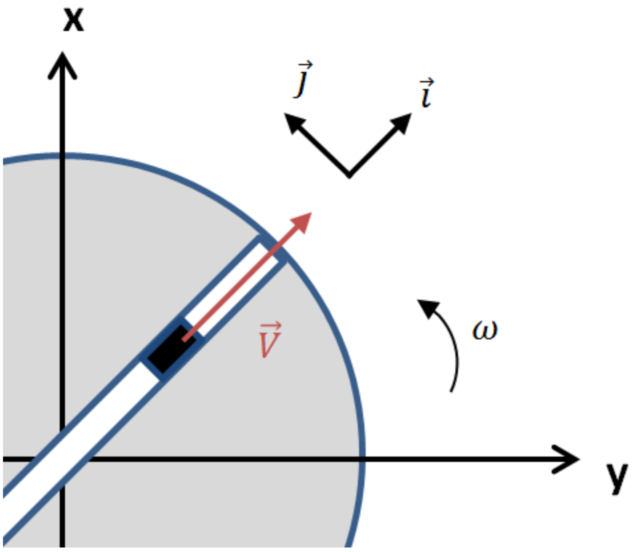


Figure 2.2. *Velocity on a rotating frame.*

Since the Earth is rotating and we are moving on it, it constitutes for us a rotating frame on which we try to keep a straight velocity.

$$\overrightarrow{a_c} = 2\Omega\vec{V} \qquad (2.4)$$

The Coriolis Effect is responsible for the deviation of the winds in a clockwise direction in the northern hemisphere. The effect is experienced in a counter clockwise direction in the southern hemisphere. This example reminds us that the Coriolis Effect takes a different direction depending on the hemisphere.

2.3. Earth atmospheric model and barometers

Measurements of the atmospheric pressure are an important clue to determine the altitude of a vehicle. By definition, an increment of pressure indicates the presence of a greater weight on a surface. At the sea level there are about 101 KPa of static pressure that is equivalent of $101000\ N/m^2$. This pressure produces the weight of several elephants on the surface of a desk. Fortunately the pressure is equally distributed around every object so to balance the load. The altitude is inversely proportional to the atmospheric pressure because the higher you go the less air is remaining above. The weight of the air is responsible for the atmospheric pressure; this is clear evidence that even gasses weight.

Nowadays the most accurate model for atmospheric pressure is the ISA (1975) [10]. This model has been made with several measurements from weather balloons in an outdoor environment. It is basically a lookup table of parameters like atmospheric pressure and air density, at different altitude, in standard conditions. The intermediate values are calculated with a linear interpolation from the lookup table. When we are estimating the altitude in the first layer of the atmosphere, called *Troposphere*, the formulation becomes slightly easier.

$$h(p) = c_1\left(1-\left(\frac{p}{p_0}\right)^{c_2}\right) \tag{2.5}$$

In the equation 2.5 [10] the altitude (h) is function of the pressure (p) where c_1 and c_2 are empirical coefficients. It should be noted that the pressure at sea level changes constantly with the weather conditions.

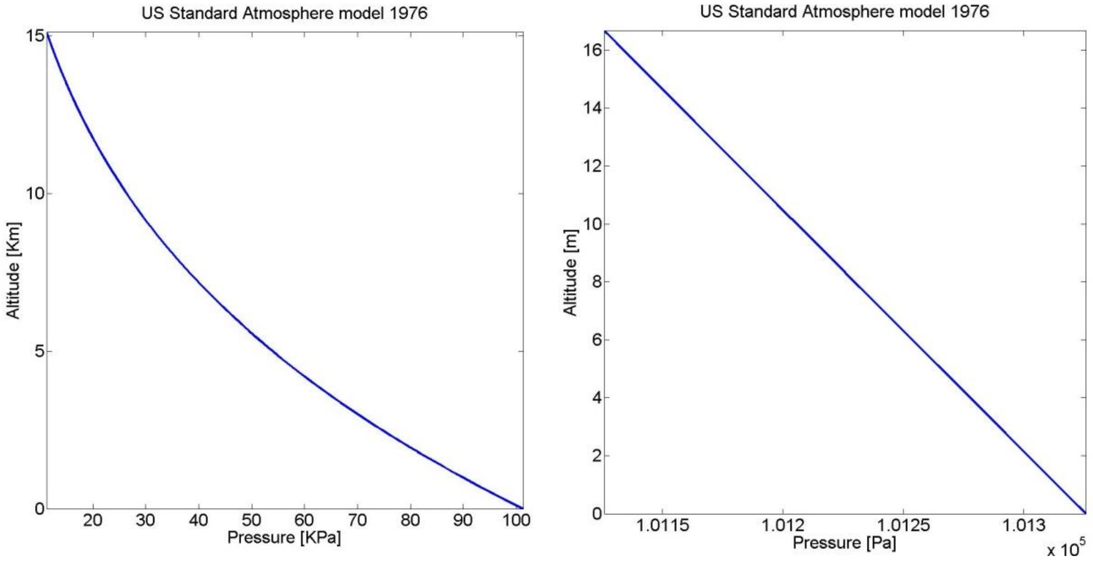


Figure 2.3. *Atmospheric pressure and altitude relation in a wide (left) and narrow (right) range.*

The barometer is a sensor that turns reading of the atmospheric pressure into a mechanical displacement and then into an electrical signal. The deformation of a thin membrane indicates the change of pressure that is experienced on its surface. This displacement moves the gauge of the classic barometers or produces an electrical signal. The sensor does not have a significant bias but the weather condition produce a similar effect. This can change from day to day or sometimes even faster. For example the sea level pressure is lower during a rainy day. Let´s assume to use this reference pressure from now on forever. Using the exponential model of the equation 2.5 we would expect tens of meters of error when the weather changes once again. The exponential model assumes reference pressure at sea level therefore it will provide the absolute altitude to the respect of the sea level. To obtain the altitude from the floor it's enough to subtract the result of the model when the barometer is lying on it (initialization).

2.4. Earth gravity model and accelerometers

Many gravity models were published in the past, from Potsdam (1930) to EGM2008 through WGS84 and EGM96. The most recent EGM2008 and EGM96 provide dramatic improvements in terms of accuracy if compared to the WGS84. They define the gravity as function of latitude, longitude, altitude and other minor parameters. On the other hand they grow in modelling complexity and they need a bigger computational power to be resolved. Despite WGS84 it is not the most accurate model of the gravitational field, it is the one we consider because simple but fairly accurate.

$$g(l) = g_0 \frac{1 + \lambda_1 \, sin^2(\, l)}{\sqrt{1 - \lambda_2 \, sin^2(\, l)}} [{}^{m}/{}_{s^2}] \tag{2.6}$$

In the equation 2.6 [7] the local gravity (g) is just function of the longitude (l) since the radius of the Earth changes with it. The parameter g_0 is the gravitation constant at sea level while λ_1 and λ_2 are empirical coefficients. For example, according to this model, the gravity experienced at 61.4 degs North (Tampere, FI) is about: $g_{61.4} = 9.8203012\ [m/s^2]$

The mass of the celestial bodies and the distance from it determine the magnitude of the gravitational field. On the other hand, it is not jet clear to science what is the mechanism that produces the gravitational field.

An accelerometer is an electromechanical device that measures the intensity of the acceleration experienced on one or multiple axis. Newton's principle of inertia claims that every mass tends to keep itself steady, when an external force is applied. This is the reason why we have to impress a greater force in order to lift a greater mass. Another important principle relates the force to the mass and the acceleration. In fact the force is proportional to the acceleration and the mass to which it is applied. The force and the acceleration applied to a mass will determine the variation of the quiet status and the movement of the object. At this point, the measurement of the acceleration can be just determined by the displacement of the proof mass from the place in which the force started to be applied.

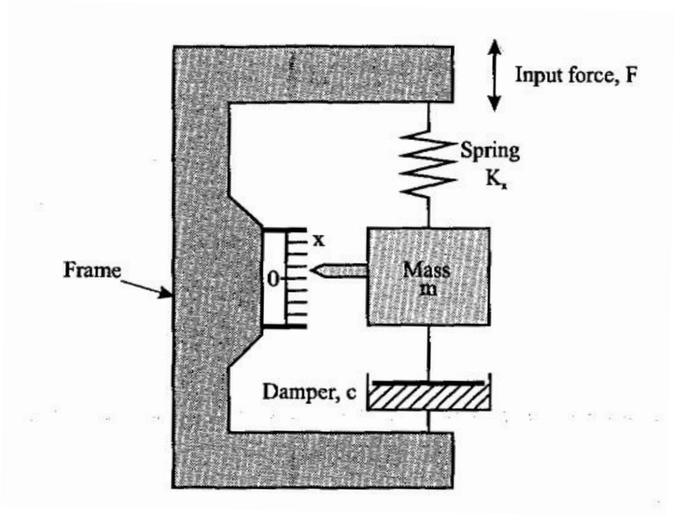


Figure 2.4. Spring-damper mechanism inside an accelerometer (*Groves, Paul D., 2008, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems) [8].*

$$f = m\frac{d^2x}{d^2t} + c\frac{dx}{dt} + kx \qquad (2.7)$$

The figure 2.4 shows the principle of the accelerometer measurement on one axis while the equation 2.7 is the corresponding differential equation (equilibrium derived from the figure) where the force (f) is related with mass (m), damping factor(c) and the elastic linear (k) of the Hooke's law.

There are many ways to implement the same configuration, for instance in the pendulum mechanism or using a vibrating structure.

It is important to highlight that accelerometers measure a specific force (f) as shown in the equation 2.8.

$$f = a + g \qquad (2.8)$$

The effect of the gravitational field (g) is added to the acceleration (a) due to other causes. Only during a complete parallel levelling to the surface of earth the gravity is not measured by the accelerometer.

2.5. MEMS technology

There are many classes of sensors based on different technologies. Depending on the principles there can have different performance and range of price. In our case the MEMS technology constitutes a very practical choice because it is low cost has a low power profile and it is small in size. At the current times the MEMS technology has yet a big limitation in terms of accuracy and measurement stability in time. Big steps have been done from their first realization in '90 and every decade we assist to big improvements. Considering the trend of improvement, it is likely that they will reach performance comparable to the current navigation grade within the next decade.

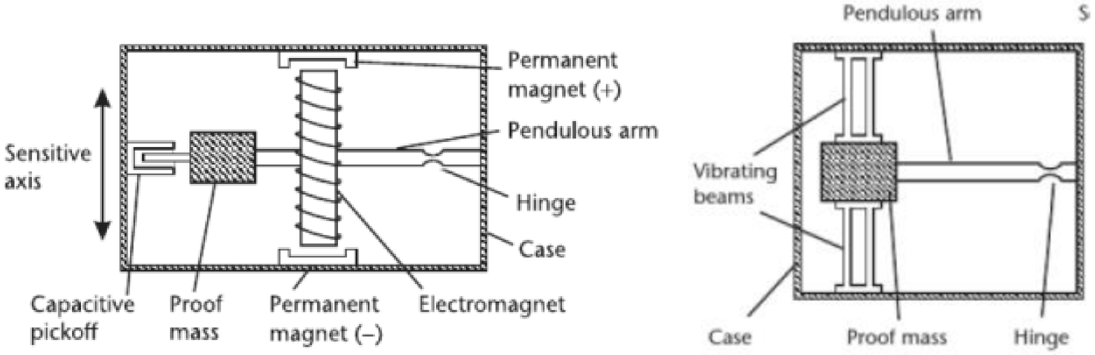


Figure 2.5. *Pickoff compensated MEMS (left) and vibrating mass technology (right) (Groves, Paul D., 2008, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems) [8].*

The figure 2.5 shows the working principles of two main MEMS technologies for accelerometers or gyroscopes: pendulous mass or vibrating beam. As we saw in the accelerometers, gyros and barometers there are very tiny moving parts that are free to shift or vibrate. The displacements are measured and compensated through the use of many techniques that produce very different results. Some producer uses piezoelectric materials or capacitive effects to determine the displacement even though the accuracy is very poor (cheap MEMS). Other companies use magnetic fields to compensate the pickoff or to levitate the proof mass. This technique is very complex but produces MEMS with high accuracy that fall in the tactical grade. Another technique based on a vibrating structure and the variation of the resonating frequency during the external action. The peak frequency will shift in the spectrum depending on the acceleration or the angular rate experienced by the electronic package and therefore by the body. In the book Groves, Paul D., 2008, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems [8], these configurations and techniques are vastly explained in the chapter 4.

2.6. Ultrasonic range finder

An UsRF is a compact and light (weight) sensor that generates an ultrasonic wave with a specific beam pattern. The wave travels straight and hits the objects producing a reflected wave and some multipath components. The same membrane that generates the sound is used to capture the reflected waves. The sensor measures the period between the transmission and the reception of the reflected waves. It is possible to estimate the distance from the body that reflected the beam because the wave travelled at the speed of sound in the air. The result will be affected by some uncertainty, but any case it will provide a useful clue about the distance from the main objects.

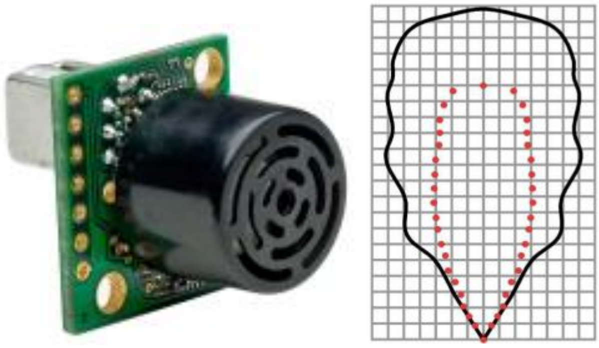


Figure 2.6. Compact *UrRF (right) and the corresponding sonic beam (right) (MaxBotix Inc., 2012, XL-MaxSonar®- EZ™ Series High Performance Sonar Range Finder) [11].*

The readings depend on the sensitivity and the design of the special sonic membrane. The beam can be narrow or wide, and it must be chosen depending on objects to detect.
A wide beam is good to identify human bodies or walls but it risks capturing objects that are not straight ahead the sight. A very wide beam may become a problem in the altitude estimation because sighting downward there is the risk to capture the walls to the side instead of the floor. In this case a narrow beam is a better choice.

2.7. Error model

Sensors measure physical quantities that are affected by errors. These errors can be only partly compensated and are due to many factors. For gyros and accelerometers, the typical error model is [8, p.117]:

$$\hat{f} = b_a + k_a(s_a + M_a)f + w_a \tag{2.9}$$

$$\widehat{w} = b_g + k_g\left(s_g + M_g\right)w + w_g \tag{2.10}$$

In the equations 2.9 and 2.10 the measured specific force $\hat{f}$ and rate $\widehat{w}$ are given by the addition of bias (b) and the composition of sensitivity factor (k) with scale factor (s) and misalignment factor (M) from the ideal force f and ideal rate w. w_a and w_g are the respective additive noise. Good performance in the design are achieved just taking into account the main error component like noise (reducible but never avoidable) and the run-to-run bias. Misalignment and scale factor are quite important in low grade MEMS.
An additive noise component is present in every sort of measurement and it can never be fully eliminated.
There are different kinds of noise modes but the most common is the AWGN (Additive White Gaussian Noise). It has the classical bell shaped normal distribution with zero mean value. In theory many models use the AWGN assumption in the stochastic process but many times other kind of noise are (naively or smartly) assumed to be distributed in this way.

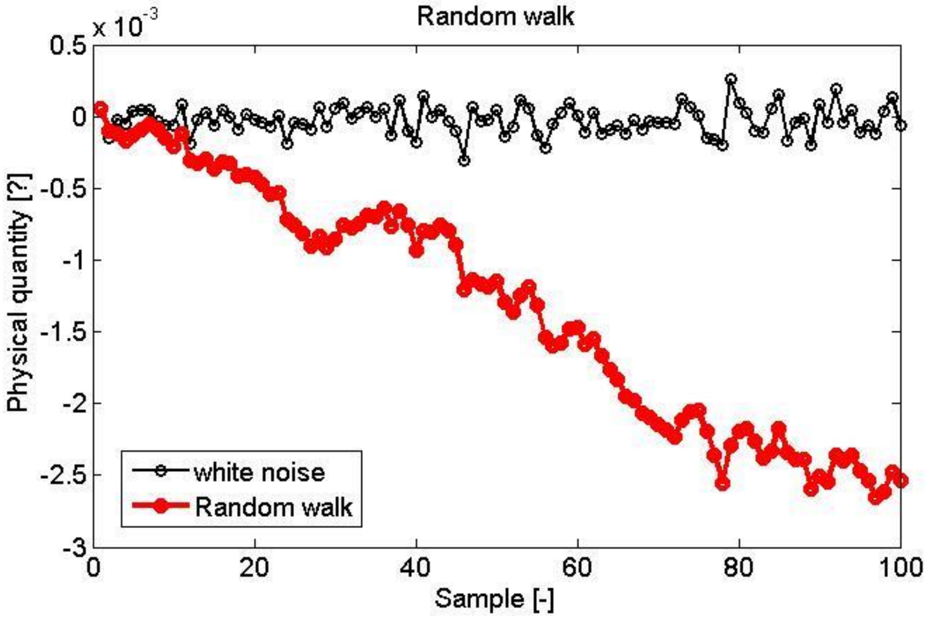


Figure 2.7. *Integration of noise produces the random walk.*

The standard deviation of the white noise defines the accuracy of the sensor. Measurements are provided in a range of confidence $\pm\sigma$ or $\pm 2\sigma$ that represents 68.2% or the 95.4% likelihood range for Gaussian noise.

In INSs there is the need to produce indirect measurements of the attitude, velocity and position using many stages of integration. Every reading has its central value and its noise level that will be integrated at every cycle. The accumulation of the white noise component is called *random walk* effect.

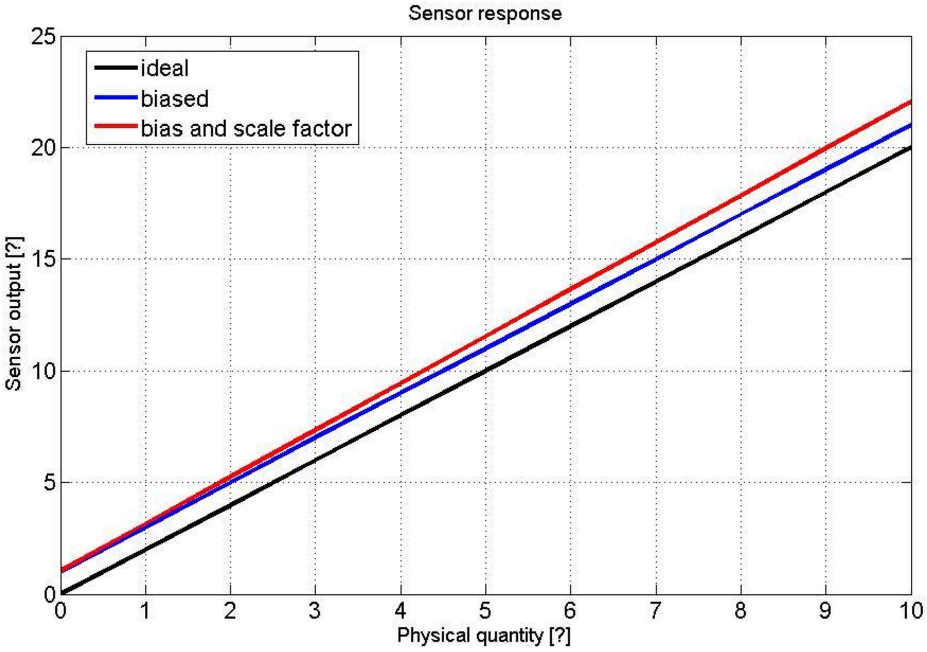


Figure 2.8. *Bias and scale factor from the ideal response.*

The ideal sensors in case of zero magnitude input produce a zero output.

Real sensors (excluding the AWGN) are actually producing a non-zero measurement in correspondence of a zero magnitude input. All the measurements are affected by an offset that shifts them to a different value. This is one of the main issues that must be compensated during the calibration phase. This is defined as “run-to-run” variation because, depending on the category of sensor, it changes every time that the device is turned on. Many sensors are produced in a package that contains an orthogonal triad of axes where the axes may be not completely perpendicular to each other. In a 3-axes accelerometer a small production defect can make the sensor measure a little projection of the acceleration applied on a perpendicular axis (or project part of the acceleration experienced on this axis to another of the triad). The *scale factor* is a gain factor that changes the slope of the expected correspondence between physical magnitude and output of the sensor. The measurements are expected to have the same linear sensitivity independently by their value. For example: if the physical quantity to be measured is unitary and the output is unitary, we would expect to have ten times the output for ten times the unitary quantity. The scale factor is responsible of a wrong reading that may produce only nine times the output at ten times the physical quantity.

This term is often confused with the *sensitivity factor* that instead shows the linear relation between the inputs the corresponding output. For example a thermic sensor produces a voltage depending on the temperature with a sensitivity of $s = 10[V/℃]$ while its typical the scale factor is $k = 0.003[V/℃]$

The figure 2.8 shows the ideal curve in black, the biased in-out in blue and the factor scaled curve in red. We discussed about the *non-zero-offset* even called *bias* and we assumed it to be constant. If the bias changes considerably after a period of time from the power up, it is considered to be instable. The term *bias instability* is used to indicate this phenomenon. This is called “in-run” variation because the parameter changes while the sensor is operating.

A sensor can be influenced during its measurement period from factors like temperature or other parameters. For example the additive noise is mainly generated by the thermic oscillation in the electronics components; consequently the noise and the accuracy of the instrumentation are temperature dependent. The most accurate sensors and oscillators are kept to a specific temperature and then calibrated because in presence of a warmer environment the sensitivity and the noise level can increase. A gyro might have a distortion in the readings caused by a linear acceleration or an accelerometer can be influenced by a rotation. In case of digital interface, the measurements are provided in a digital form. The sensor converted the physical quantity into an analogic electrical quantity. This analogic signal is converted with an ADC that has a noise figure and a limited amount of output levels. In genre the acquisition process is designed to provide enough significate bits to notice the measurement noise.

There are many other minor and rare sources of error. They are important to take into account only when that specific sensor technology is well known to be effected by it.

Two phenomena are often occurring: non linearity and hysteresis.

It is important to notice that all these modelled components are already known to the producers of sensors. They often provide sensors with a built-in compensation of non linarites or cross influences that are otherwise specified so to be compensated by the designer. All the relevant cross influences are in genre mapped by the producer of the sensor with a plot or some correction polynomial

It is possible that the different measurements we are trying to fuse have different bandwidths and therefore delays. We expect measurement of the current state while they provide an "old picture" of it. Gyros have a very high bandwidth and they can easily follow quick variations while magnetometers need a longer time to adapt to the projection of the magnetic field. For example: when you turn around with a compass in your hand, you can notice that your quick rotation does not give the time to the compass to provide the right heading. This delay is in terms of seconds and there is the risk to supply old measurements if the rate on the azimuthal axis is higher than this delay. Barometers have a similar attitude because they need time to experience a slight variation of pressure. Dead zone issue or limited bandwidths have to be taken into account in case of high dynamics systems. Fortunately this delay is not an issue considering the very slow vertical rate of the blimp. The same assumption is valid for azimuthal rate where the built-in delay of the compass does not constitute an issue.

3. Inertial system

The inertial navigation systems (INS) have a very complex structure and are used to determine the PVA solution (position, velocity and attitude estimation) of a vehicle during its navigation. INSs use measurements from on-board sensors such as accelerometers and gyroscopes. For this reason they are "self-contained" because they do not rely on any external infrastructure to compute the solution. This feature ensures a great reliability and resistance to external interferences. On the other hand the errors in the estimations are accumulated in time, and the senses must be really accurate to achieve high performances. Given the definition of speed as: instantaneous variation of position, and given the definition of acceleration as: instantaneous variation of speed. The basic idea of INSs is that, measuring the acceleration and processing it on different stages of integration, it is possible to indirectly measure the velocity and the displacement from the initial position. Knowledge of initial attitude, velocity and position constitutes the initial values of the integration stages.

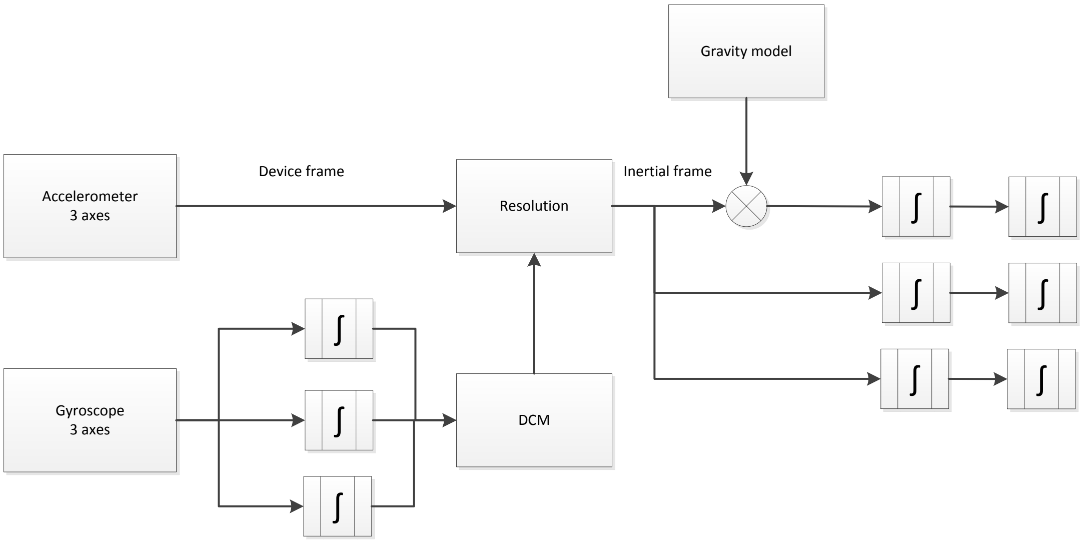


Figure 3.1. *General block diagram of an inertial system.*

The linear and angular velocities are assumed to be constant in the observation period. This is the key assumption of the *Dead Reckoning* technique. For example: if the gyro measures a constant angular rate of 20 deg/s it means that after a second the direction on that axis has changed of 20 degree. We are assuming that in that second the angular rate was constant and that there are no imperfections in the measurement and computation.

If explained in these terms it may look like an easy task. Unfortunately every sensor has some imperfection, sources of error, limited rate and bandwidth while aging and temperature change the characteristics of the measurements. Another issue is caused by the Earth gravitational field. Accelerometers cannot distinguish between the acceleration due to the applied force and the one generated by the gravitational field. In a steady situation the result of a measurement along the gravitational axis is indeed the local gravity value. It is called local acceleration because the magnitude changes from place to place on the Earth's surface. An analytic estimation of the gravitational field on that specific spot on Earth allows distinguishing the gravitational component from the actual force applied on the vertical axis. Rate gyros measure projection of the earth rate superimposed to the rate due to the rotation of the sensor.

The first historical implementations used a mechanical gimbaled platform. A gimbaled frame had many actuators (torquers) to keep the accelerometers parallel to the horizon. The rate gyros were used to sense the variation of the attitude of the vehicle and then used to adjust the orientation of the stabilized platform (capturing). In this way no matter the maneuvers, the inner part of the gimbal kept the direction of the initial calibration. This is called physical stabilized platform and may suffer many mechanical malfunctions and a limited capability to response in case of extreme maneuvers. Furthermore the theoretical *gimbal lock* may physically occur.

During the 80's the analytic stabilized platform substituted the mechanical solution. The idea is to keep the axes of the sensors fixed to the body of the vehicle (strapdown technique). The angular rate is integrated so to keep track of the instantaneous attitude. At this point the angles of the attitude are used to project the measurements to an analytic inertial frame. This technique produces the same result of a physical stabilized platform removing the mechanical issues and ensuring a greater reliability and effectiveness (even with extreme maneuvers). On the other hand the computational complexity and the accuracy of the instrumentation, makes the design of this kind of system really challenging.

Whether the stabilization technique is physical or analytic: gravitational field, Earth rate and Coriolis acceleration have to be compensated. The Earth rate depends on the latitude and the Coriolis acceleration is dependent (as in a chain) on the Earth rate.

3.1. Coordinate frames

Flying vehicles are free to rotate in 3 dimensions. We can define 3 axes that are fixed with it and that keep on follow the vehicle during its rotations. According to the aeronautical convention shown in figure 3.2, the triad of orthogonal right handed axes have the z-axis pointing down.

The angles ϕ (phi), θ (theta), (ψ) psi correspond to *pitch, roll, and yaw* in aeronautical terms. They grow following the right hand rule (right thread screw).

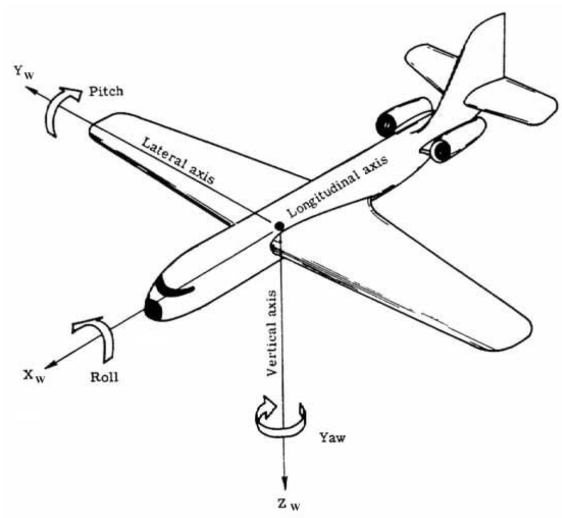


Figure 3.2. *Aeronautical axes and angles (NASA SP-367 Introduction to the Aerodynamics of Flight) [12].*

There is a slight difference between the *body frame* origin and orientation and the position of the *sensor frame*. In fact it happens quite commonly to have the sensors frame far from the centre of rotation.

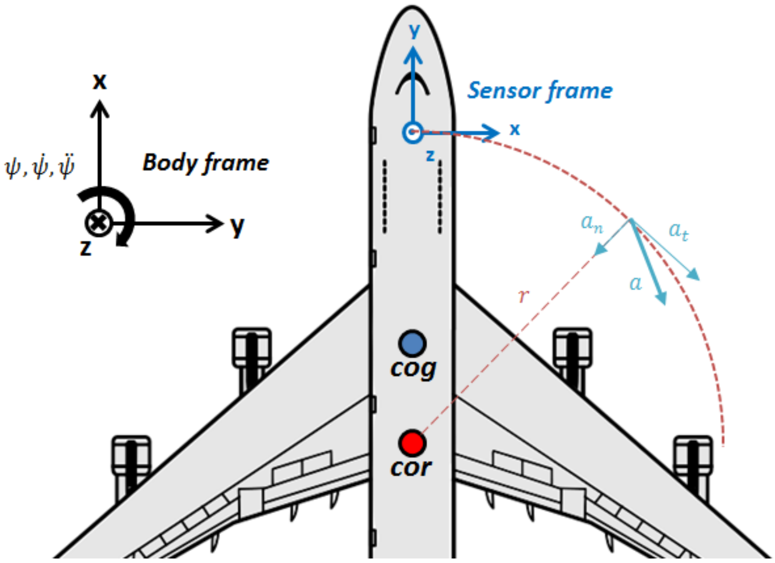


Figure 3.3. Example of sensor frame and body frame on a Boeing 747.

In the example of figure 3.3, the centre of rotation (cor) does not correspond to the centre of gravity (cog). The sensor frame is oriented z-axis up and originated in the cockpit with all the rest of the electronics. The accelerometers will measure a linear acceleration due to a rotation as expressed in the equation 3.1

$$a = a_t + a_n \tag{3.1}$$

$$a_t = r\ddot{\psi} \tag{3.2}$$

$$a_n = r\dot{\psi}^2 \tag{3.3}$$

In the equation 3.1 (derived from Newton's law), the acceleration consists of *tangential* (a_t) and *normal* (a_n) accelerations (equations 3.2 and 3.3). The first is due to the angular acceleration ($\ddot{\psi}$), while the second is due to the angular rate ($\dot{\psi}$). Both the components are proportional to the radius (r). In this case the radius is the distance from the sensor frame and the centre of rotation. In a Boing 747, this key factor that amplifies the effects of the acceleration is about 30 meters.

The resulting acceleration must be considered if its magnitude is greater than the measurement noise of the accelerometers. In case of very short distances from sensor frame and centre of rotations this phenomenon is so small that can be easily neglected. Furthermore low grade accelerometers (MEMS) cannot even notice these superimposed acceleration because it lies within the measurement noise. From the point of view of the gyroscope there is not such issue because it is strapped to the vehicle and it rotates with it.

The *inertial frame* is a fixed frame not subject to rotations. Usually it is aligned x, y to the local surface and with the z axis parallel to the effects of the gravity. The sensors frame measurements are resolved (rotation) to match this frame in order to have a reference that is independent by the attitude of the sensor and body frames. It is possible to supply the local gravity acceleration and compute the velocity and position estimation with multiple stages of integration. This frame is the ultimate in case of indoor navigation. Usually the position estimation on the x, y plane is proposed directly without conversions into latitude and longitude.

3.2. Rotation of axes

In order to obtain a rotation without changing the magnitude of the vector we can use many mathematical tools. The most intuitive and direct is the Rotation Matrix also called DCM (Direction Cosine Matrix) [8, p. 28].

$$C_\alpha^\beta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.4}$$

Using the DCM it is possible to trasform the components of a vector from a frame to another [8, p. 26]. The equation 3.4 shows an example of direction cosine matric (C) from frame α to frame β.

$$v_\beta = C_\alpha^\beta v_\alpha \tag{3.5}$$

In the equation 3.5 the vector v_α has been rotated in the order yaw, roll, and pitch to the new frame with components v_β. Notice that rotations performed in different order do not necessarily produce the same result.

DCMs may suffer a limitation called *gimbal lock*. It is due to the use of 3 coordinates to describe a rotation on 3 axes. These are not enough to univocally define all the possible configurations. In case of rotations of the angles so that all the coordinates are parallel, there is the loose of a degree of freedom. A solution to this issue is the use of quaternions that defining 4 components through the use of 3 complex vectors, allow a complete coverage of all the possible configurations. The complexity and the number of calculations to manage the quaternions are far beyond the needs of 90% of the cases. The *gimbal lock* is an extreme situation that occurs very rarely or never in case of low mobility systems. This possibility does not have to mystify the genuine and intuitive use of DCMs that are vastly used in many critical free applications.

3.3. Initialization

Calibration is the initial phase in which the vehicle is kept steady and aligned to many navigation references. In a mechanical IS the attitude alignment results in the 0g measurement on the horizontal axis and the local gravitational acceleration on the vertical axis. The heading is aligned to the geographical north through gyro compassing technique. With low grade gyros it is very difficult to perform the gyro compassing.

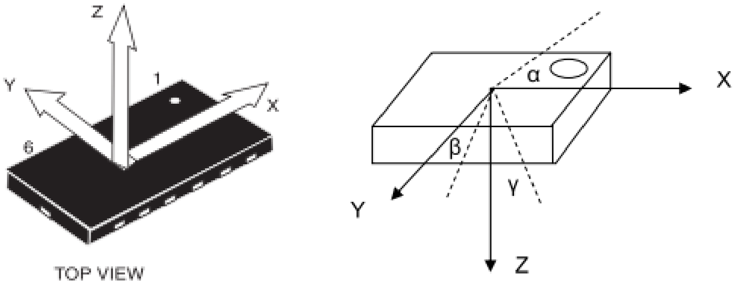


Figure 3.4. *Alignment of axes in the package (left) and corresponding tilt angles (right) (ST Microelectronics, LSM303DLHC Ultra compact high performance e-compass 3D accelerometer and 3D magnetometer module) [13].*

Using simple trigonometry on the figure 3.4, it is possible to obtain the following equations 3.6 and 3.7. The variables A_x and A_y are the measurements of the accelerometer on the x and y axes.

$$\alpha = pitch = \arcsin(\frac{A_x}{g})\ [rad] \qquad (3.6)$$

$$\beta = roll = \arcsin(\frac{A_y}{g})\ [rad] \qquad (3.7)$$

The corresponding angles with their error would be:

$$\alpha + \delta\alpha = pitch + \delta pitch = \arcsin(\frac{A_x + \delta_x}{g})\ [rad] \qquad (3.8)$$

$$\beta + \delta\beta = roll + \delta roll = \text{arcsin}(\frac{A_y + \delta_y}{g})\ [rad] \qquad (3.9)$$

In a practical case when the accelerometer is well aligned with the local gravity the perpendicular axis x and y have no projection of the gravity component. The measurements on these axes should be nonexistent but in reality they correspond to the biases. The other contribution is given by the noise level. The noise can be reduced averaging several samples. If the accelerometer has a typical bias of $\pm 60\ 10^{-3}[g]$ the corresponding alignment error would be of $\delta = \frac{360}{2\pi}\text{arcsin}\left(\frac{bias}{g}\right)[deg] = \frac{360}{2\pi}\text{arcsin}\left(\frac{60\ 10^{-3}}{g}\right)[deg] = 3.4398[deg]$

It is very common to have alignment errors of about 5 degree using low grade accelerometers. As shown in the example, the bias plays a very big role in the error budget. A smarter approach consists in the estimation and compensation of the bias before the leveling. The *bias nulling* is the process that tries to remove the bias when the sensor should measure only it (steady case). This bias is independent by the acceleration or rotation but it may change in time (bias instability).

There are techniques that imply the rotation of the axis so to compensate the common bias and discerning it from the earth rate (gyro) or the gravity (accelerometer). It implies the use of mechanical rotation of the sensors with a very accurate precision. It is not possible to place such heavy and expensive equipment on every vehicle. The only way to remove the bias is measuring for a short period the readings in a steady case when only the bias is measured and removes the average of it in the following (non-steady) samples. Any case of low grade gyro and accelerometer, the bias instability may be relevant. This process has to be repeated or automatized in a more complex filtering (e.g. KF or EKF).

3.4. Kalman filter

Kalman filter is a recursive algorithm that uses noisy measurements and knowledge of the system model to provide the optimal statistical estimate for a set of state variables. The state variables have a linear and dynamic relation expressed in terms of linear differential equations. Systems dynamics and measurements are both affected by noise that is assumed to be white Gaussian distributed. The Kalman filter has many applications, from economics to automatic controls but it has found its first application for the trajectory estimation of the Apollo Project. As explained in the paper by Mohinder S. Grewal and Angus P. Andrew Applications of Kalman Filtering in Aerospace 1960 to the Present [14], at that time the *data fusion problem* was yet open because many independent noisy measurements are providing clues about the aircraft trajectory. In other words the Space Age due very much to Rudolf Emil Kalman and the intuition that he had in 1958.

Groves D.P. uses the title “Correction of the Inertial Navigation Solution” [8, pg. 365], where the term *correction* explains briefly and accurately what is the purpose of the KF in IS system.

The calculations can be separated in two stages called *prediction* and *update.* The first propagates a priori the state vector using finite difference equations. When a new observation is available a gain (called *Kalman gain*) scales and fuses noisy clues of the state (measurements) with the prediction.

This is the original formulation in discrete time that shows the systems knowledge and asynchronous series of measurements (symbol at pg. 3).

$$\begin{gathered} x_k = \Phi x_{k-1} + \mathrm{T}u + w_k \\ w_n \sim N(0, \Gamma) \end{gathered} \tag{3.10}$$

$$\begin{gathered} z_k = Hx_k + n_k \\ n_k \sim N(0, R) \end{gathered} \tag{3.11}$$

Predict phase:

$$x_k(-) = \Phi x_{k-1}(+) + Bu \tag{3.12}$$

$$P_k = \Phi P_{k-1}(+)\Phi^T + \Gamma \tag{3.13}$$

Update phase:

$$S_k = HP_k(-)H^T + R \tag{3.14}$$

$$K_k = P_k(-)H^T inv(S_k) \tag{3.15}$$

$$\mathrm{i} = z_k - Hx_n(-) \tag{3.16}$$

$$x_k(+) = {x_k}^T + K_k \mathrm{i} \tag{3.17}$$

$$P_k(+) = P_k - KHP_k(-) \tag{3.18}$$

Note that this particular formulation has the innovation vector (i) well defined. This will allow the consistency check in the following chapter. Even though the measurements are always supplied quite regularly in a discrete domain, many systems have their natural formulation in continuous form.

$$\begin{gathered} \frac{\delta}{\delta t} x = Fx + Bu + w \\ w \sim N(0, Q) \end{gathered} \tag{3.19}$$

In this case the first order discrete equivalent is given by the equation 3.20, 3.21, 3.22.

$$\Phi = I_n + \Delta TF \tag{3.20}$$

$$\Gamma = \Delta TQ \tag{3.21}$$

$$T = \Delta TB \tag{3.22}$$

This is a first approximation that may not be legitimate in some case. The chapter 4.3 uses and explains further details of the KF.

4. Dynamic Models

The kinetic model does not need to be very accurate. In fact the real model has a multitude of variables and parameters that may be hard to estimate with high precision and that any case keeps on change in time. A rough model is a valid tool to experiment different control techniques and to simulate the response of the system.

The two motors can be steered (pitch up/down) and their thrust can be controlled separately. In this way the thrust can be directed vertically or on the horizontal plane.

The following equations are derived using the equilibrium of forces (Newton's law) from the free body diagrams of the figure 4.1 and 4.2.

4.1. Vertical axis

The motors are located on the center of gravity of the balloon. It is reasonable to neglect the forces that produce a pitch of the vehicle; consequently there are only translations on the vertical channel.

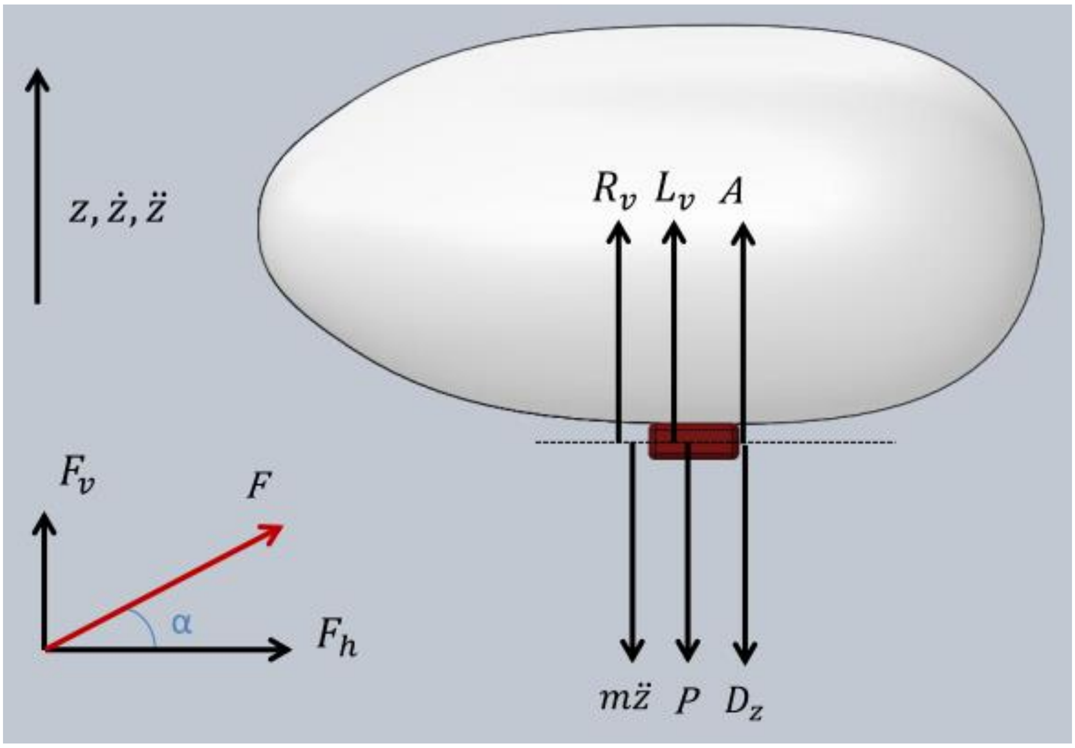


Figure 4.1. *Free body diagram of the vertical channel.*

The equilibrium of the forces at the COG (center of gravity) is the following:

$$\mathrm{R}_v + \mathrm{L}_v + A - \mathrm{D}_z - P - m\ddot{z} = 0 \tag{4.1}$$

$$\mathrm{D}_z = sign(\dot{z})\frac{1}{2}\rho\dot{z}^2 c_z S_z \tag{4.2}$$

$$\mathrm{P} = mg \tag{4.3}$$

$$A = \left(\rho_{air} - \rho_{gas}\right)V \tag{4.4}$$

The general force (F) is projected through the angle alpha on the vertical or horizontal plane. This is used to describe the thrust provided by the motors depending on the stearing angle.

The forces pointing up are due to the lift of the gas (A) and to the left (L) and right (R) motors (vertical projection). Gravity action (P), inertia and air drag (D) are acting against the lifting forces.

4.2. Horizontal plane

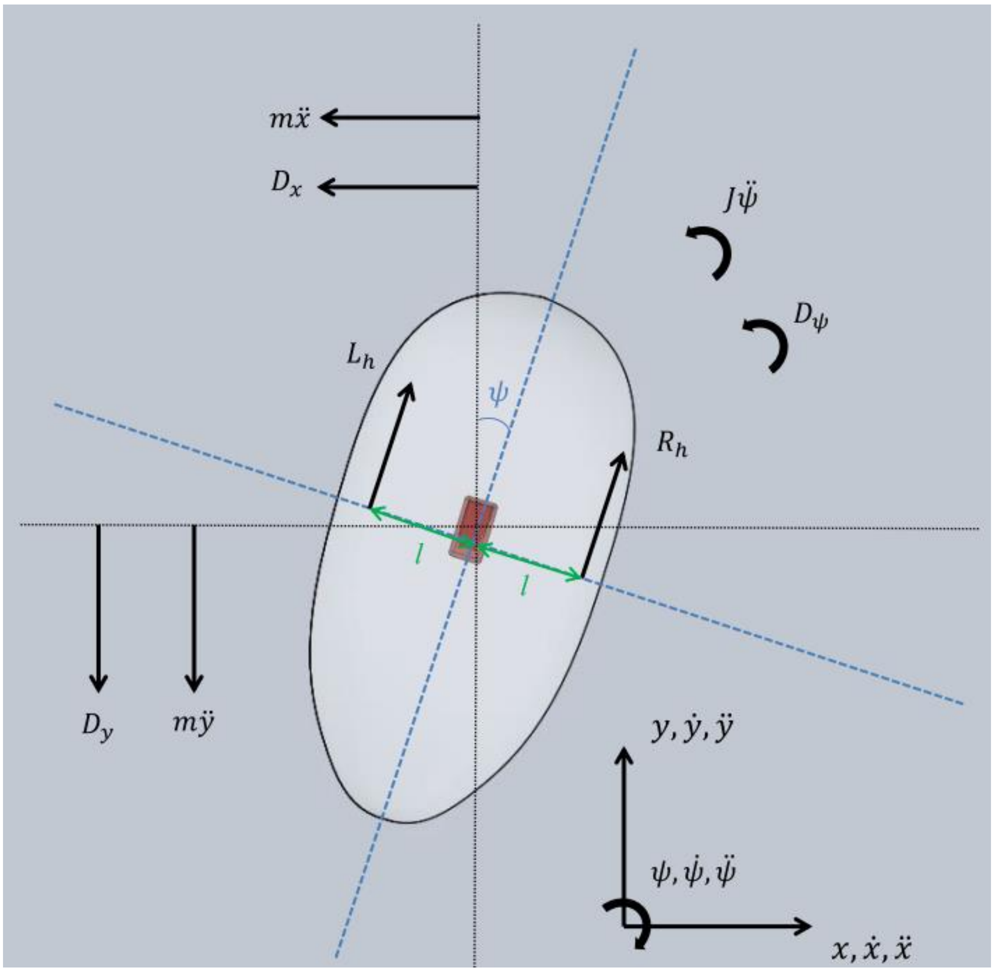


Figure 4.2. *Free body diagram for horizontal plane and heading.*

The horizontal equilibrium about the COG for translations and azimuth rotation gives:

$$(\mathrm{R}_h + \mathrm{L}_h)\sin\psi - \mathrm{D}_x - m\ddot{x} = 0 \tag{4.5}$$

$$(\mathrm{R}_h + \mathrm{L}_h)\cos\psi - \mathrm{D}_y - m\ddot{y} = 0 \tag{4.6}$$

$$(\mathrm{L}_h - \mathrm{R}_h)l - J\ddot{\psi} - \mathrm{D}_\psi = 0 \tag{4.7}$$

$$\mathrm{D}_x = sign(\dot{x})\frac{1}{2}\rho\dot{x}^2c_xS_x \tag{4.8}$$

$$\mathrm{D}_y = sign(\dot{y})\frac{1}{2}\rho\dot{y}^2c_yS_y \tag{4.9}$$

$$\mathrm{D}_\psi = sign(\dot{\psi})\frac{1}{2}\rho\dot{\psi}^2c_\psi S_\psi \tag{4.10}$$

The translations on the plane are modelled with the equations 4.5 and 4.6. The equilibrium of the forces uses the left (L) and right (R) motors (horizontal projection) together with the air drags (D) and the inertia. The variables S refers to the surface exposed on different axes while the corresponding aerodynamic coefficients are denoted with the letter c. The rotation about the center of gravity is modelled in the equation 4.7 where the left and right motors produce a torque. Once again the action of air drag (eq 4.8, 4.9, 4.10) and moment of inertia are taken into account. The aerodynamic coefficients are assumed to be constant because there are small variations of orientation.

Considering the dynamics and the speed of the blimp, the Reynolds number is very small. There are basically no turbulences and therefore the regime is laminar. The operator sign allows a drag even in case of negative speed since the square operator used in the dynamic pressure make it lose this property. Details about the dynamics of airplanes and airship are better explained in the reference book by Yechout, Thomas R., Introduction to Aircraft Flight Mechanics [15] in the chapter 1.3.

Coefficients and parameters are estimated in the chapter 7.1.

4.3. Altitude estimation

The quality of the inertial measurements is very important to track the altitude and it implies a big investment of money and a heavy load to carry on the balloon. Software filters may improve the estimation enough to control the vehicle with a reasonable accuracy. A KF is used to estimate the vertical state of the blimp and to fuse inertial measurements with independent series of measurement from the UsRF and the barometer.

In the reference book [16, pg. 402] there is an application of Kalman filtering that inspires our formulation. The book written by Mohinder S. Grewal and Angus P. Andrews, Kalman Filtering: Theory and Practice Using MATLAB, 3rd Edition [17], proposes a similar example at page 178. The error model of the equations 2.9 is our base for the KF propagation model. We are assuming a constant velocity model and no bias instability. The bias is a *state augmented variable* because we want the Kalman filter to estimate it. There are currently no formulations to cope with the colored or skewed distribution of the noise.

As stated in literature [8], at page 429, this formulation takes the name of *full state* or *total state.*

$$\frac{\delta x}{\delta t}\begin{bmatrix} z \\ \dot{z} \\ b \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} z \\ \dot{z} \\ b \end{bmatrix} + \begin{bmatrix} 0 \\ a_z \\ 0 \end{bmatrix} + w \tag{4.11}$$

$$z = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} z \\ \dot{z} \\ b \end{bmatrix} + v \tag{4.12}$$

$$Q = \begin{bmatrix} {\sigma_r}^2 & 0 & 0 \\ 0 & {\sigma_a}^2 & 0 \\ 0 & 0 & {\sigma_b}^2 \end{bmatrix} \tag{4.13}$$

$$R = \begin{bmatrix} {\sigma_u}^2 & 0 \\ 0 & {\sigma_p}^2 \end{bmatrix} \tag{4.14}$$

The variable z denotes the vertical coordinate (altitude), a_z is the measured acceleration in the vertical channel (inertial frame) while the variable b is the bias.

In the diagonal components of the Q and R matrices there are the standard deviation of the measurement noise of the accelerometer (a_z), UsRF (σ_u) and the altitude from barometer readings (σ_p). The remaining σ_r and σ_b are the noise that affect the rate and bias (null in theory).

Since the Kalman Filter is a discrete filter (original formulation), the model has to be converted to a discrete equivalent.

$$\begin{bmatrix} z_k \\ \dot{z_k} \\ b_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta T & 0 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} z_{k-1} \\ \dot{z}_{k-1} \\ b_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ a_z \Delta T \\ 0 \end{bmatrix} + w \tag{4.15}$$

$$z = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} z_k \\ \dot{z_k} \\ b_k \end{bmatrix} + v \tag{4.16}$$

$$\Gamma = \begin{bmatrix} (\sigma_r)^2 \Delta T & 0 & 0 \\ 0 & (\sigma_a)^2 \Delta T & 0 \\ 0 & 0 & (\sigma_b)^2 \Delta T \end{bmatrix} \tag{4.17}$$

This model will be proved to be light enough to run real-time and accurate enough to estimate the altitude. The variable ΔT is the sampling rate.

4.4. Direction estimation

It is easy to apply the same formulation for the heading estimation. The assumptions are the same but referred to gyroscope (equation 2.10) and magnetometer measurements.

$$\frac{\delta x}{\delta t}\begin{bmatrix}\psi\\ b\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\psi\\ b\end{bmatrix} + \begin{bmatrix}w_z\\ 0\end{bmatrix} + w \tag{4.18}$$

$$z = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\psi\\ b\end{bmatrix} + v \tag{4.19}$$

$$Q = \begin{bmatrix}{\sigma_g}^2 & 0\\ 0 & {\sigma_b}^2\end{bmatrix} \tag{4.20}$$

$$R = {\sigma_c}^2 \tag{4.21}$$

The discrete equivalent is:

$$\begin{bmatrix}\psi_k\\ b_k\end{bmatrix} = \begin{bmatrix}1 & \Delta T\\ 0 & 1\end{bmatrix}\begin{bmatrix}\psi_{k-1}\\ b_{k-1}\end{bmatrix} + \begin{bmatrix}w_z\Delta T\\ 0\end{bmatrix} + w \tag{4.22}$$

$$z = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\psi_k\\ b_k\end{bmatrix} + v \tag{4.23}$$

$$\Gamma = \begin{bmatrix}{\sigma_g}^2\Delta T & 0\\ 0 & {\sigma_b}^2\Delta T\end{bmatrix} \tag{4.24}$$

The variable ψ is the heading angle, while σ_g, σ_b and σ_c are respectively the noise levels of gyro, bias of the gyro (null in theory) and compass.

It is possible to merge the two formulations of altitude and direction into one. The separation makes the formulation easier and decouples the computation that becomes quick and efficient. All the parameters are identified in the chapter 7.7

5. Control Techniques

There are systems that are well known from the modelling point of view and for this reason they are reference case for generations of students. For instance: water tank, DC motor or the complex inverted pendulum. All of these have been analyzed with a Laplace or a Z-transform equivalent. In the same way several standard controllers or ad hoc techniques are expressed in these domains so to make a zero-poles system stability analysis. Using the multiplication properties of these domains it is possible to cancel the fastidious poles of the original system while an eye is kept on the phase margin. In theory if there are no variations of parameters the time domain (real) system will perform as expected.

Engineering prioritize the "Time to market" instead of accuracy and optimization. The effort of modelling and the quest of very high accuracy are often not needed for a final product. In genre from a first working proposal and a very accurate/efficient system there is the need of 3-4 times more effort.

Historically there are two techniques that tend to be used the most of the times: Standard controller and Fuzzy logic controller. This approach is trivial in case of SISO system but it becomes hard to manage in case of MIMO.

The tuning of the controllers is a matter of iteration and experience of the designer. Following a standard approach it would imply the use of a P, PD, PI or PID controller. The error signal (or parameters derived from it) is just multiplied by several gain factors in order to obtain a command. Just trying to tune the various gains it will make the control loop affective and easy to maintain. Fuzzy logic controllers

Fuzzy logic controllers answer the need of having a simple and quick method to design controllers. There are systems that are hard to model or that do not find a control solution with a standard approach (e.g. washing machine). They are especially useful when the system is complex or with a highly coupled dynamics. Membership functions and rules are defined in a linguistic and qualitative matter so to obtain the equivalent action of a human operator. The variables are expressed through the use of membership functions that define a gradual transition between false (0) and true (1). Instead of analysing the system and using its model, this approach uses the operators experience to define a set of rules. This is a Mamdani's inference based method (Mamdani, E.H. and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller) [18].

During 80's many Japanese companies used these controllers, even when the standard controllers were more intuitive. This allowed them to patent analogous systems of the one produced by European and American companies. As result of this fashion, nowadays fuzzy logic controllers are built in many washing machines, cameras and even complex systems like aircraft and submarines.

The classical example is about the definition of cold, warm and hot temperature.

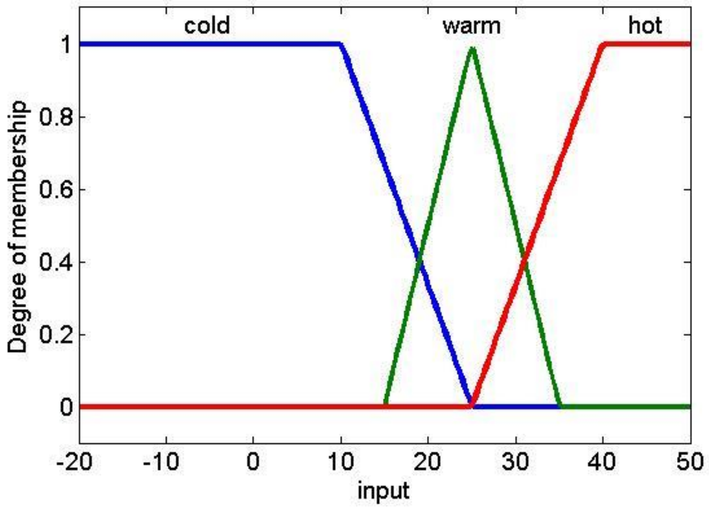


Figure 5.1. *Input membership function of the fuzzy logic controller (temperature).*

The definition of cold, warm and hot is clear to every human being. There is not a straight threshold but a gradual passage from the linguistic definition of one to another.

If we define the different clothes that men can wear depending on the temperature as in the figure 5.2:

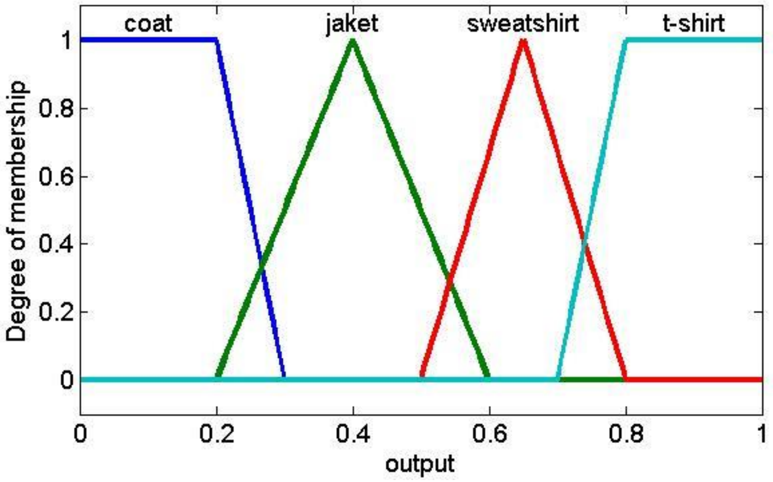


Figure 5.2. *Output membership function of the fuzzy logic controller (clothes).*

The fuzzy controller will provide us the answer on what to wear using rules like:

If (input *is* cold) *then* (output *is* coat)
If (input *is* hot) *then* (output *is* t-shirt)
………………..

Such rules are indeed interception (or union) of fuzzy sets that lead to an output.

As you can see, it would be very hard to design a standard controller that can provide such answers. This is a nice example of a system that does not have a proper model, but just a fuzzy definition of the input-output. The crisp value of temperature has to be converted in the fuzzy domain and interpreted in the rules. After this union of operands, the fuzzy output is converted to a *crisp* value. These procedures are called *fuzzyfication* and *defuzzification* and can be performed in different ways. The most commonly used defuzzification method is the Center of gravity (Mamdani, E.H. and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller) [18].

$$y = \frac{\int \mu_i(x)\, x\, \delta x}{\int \mu_i(x)\ \delta x} \qquad (5.1)$$

In the equation 5.2 the variable y is the crisp output while $\mu_i(x)$ is the aggregated membership function resulting of interception (or union) of fuzzy sets through the set of rules.

5.1. Altitude controller

The easiest way to keep the reference altitude and direction is to manage them separately with an actuator that acts only on that parameter. For instance the *rudder* of the airplane is used to change only the azimuthal direction of the vehicle. A variation of the rudder angle will generate a torque used to achieve the desired direction. The rudder does not affect the pitch or any case this variation will be compensated with another actuator specific for it. In our case the altitude is kept very easily when the propellers point up (vertically) and the motors act together to actuate the proper force to balance the weight. When the direction has to be kept the motors has to be directed horizontally with a counter spin so to generate a torque on the yaw axis. Unfortunately when you do so, the balance of the thrust and weight on the vertical channel is compromised and leads to a slight loss of altitude. This is a quite complex task even for a human operator, but with some practice he learns the procedure to follow and the margins required to move on the horizontal plane. The easiest way to achieve a direction is to gain some altitude and then produce a torque around the azimuthal that will make a slight change of direction. The operation is possible until the altitude gets to be too low and the control has to give priority to the vertical channel.

At the moment, it is performed only the altitude control using the KF’s altitude estimation while in a future release it will be the possibility to use the KF’s direction estimation in order to control the direction of the vehicle.

The reason for implement a fuzzy logic controller is influenced by the possibility to rule the direction in a future implementation. In other words, the current choice constitutes a complication but it will allow an easier and faster implementation of the MIMO controller needed to rule altitude and direction. Hopefully a long term view will provide a benefit in the future.

In the reference paper by Paul A., Debitetto, Fuzzy Logic for Depth Control of Unmanned Undersea Vehicles, IEEE journal of oceanic engineering, vol. 20, no. 3,[19] there is an application of fuzzy logic controller for that inspires our controller.

Their undersea vehicle fills tanks depending on the depth and depth rate while out blimp adjusts the motors depending on the altitude and altitude rate (in form of error and error rate).

The membership functions for the input are in the figures 5.3 and 5.4 and the output is defined in the figure 5.5

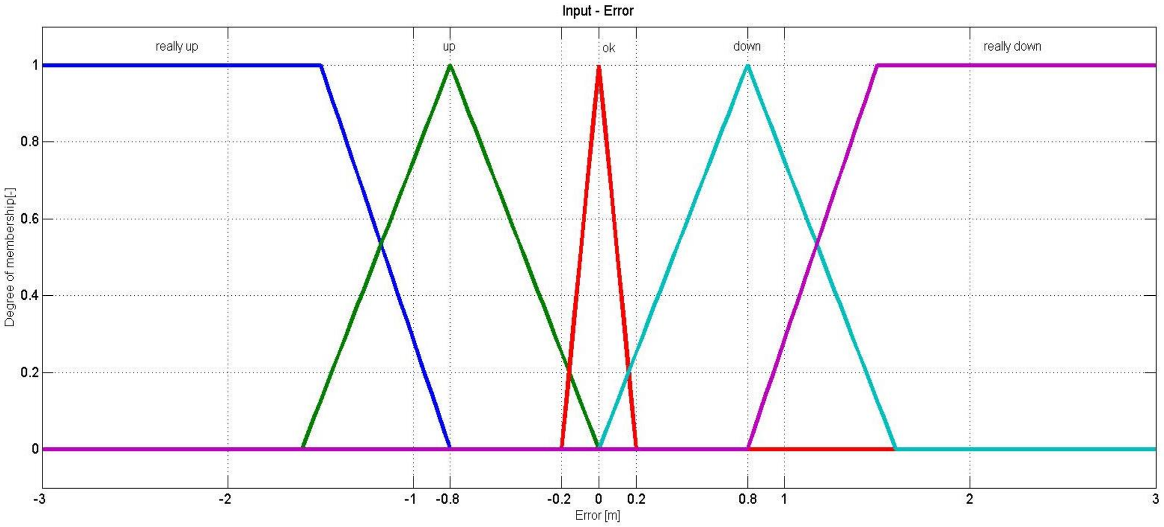


Figure 5.3. *Input error for the fuzzy logic controller.*

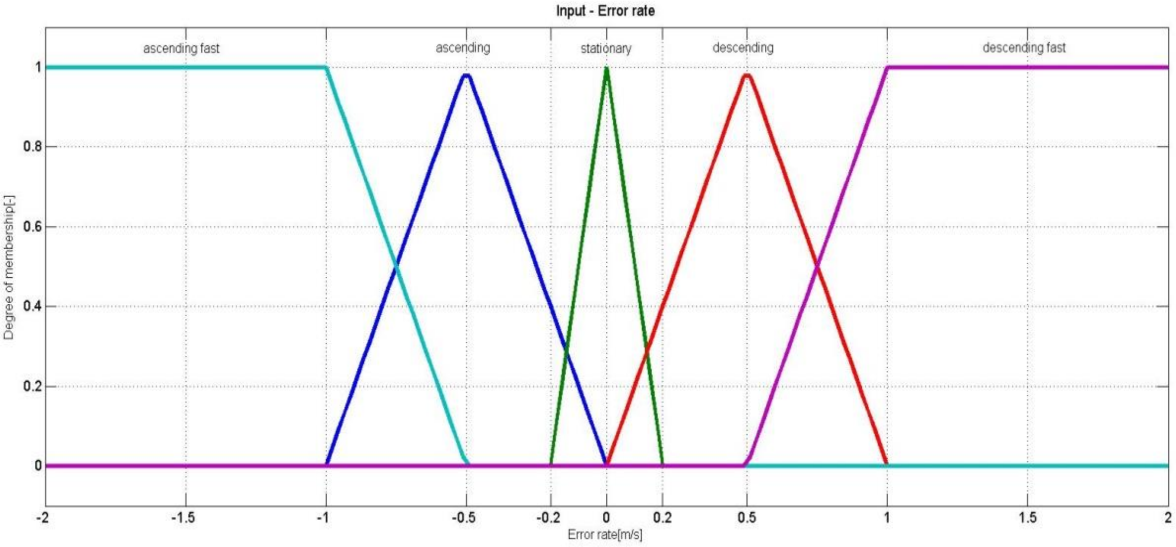


Figure 5.4. *Input error rate for the fuzzy logic controller.*

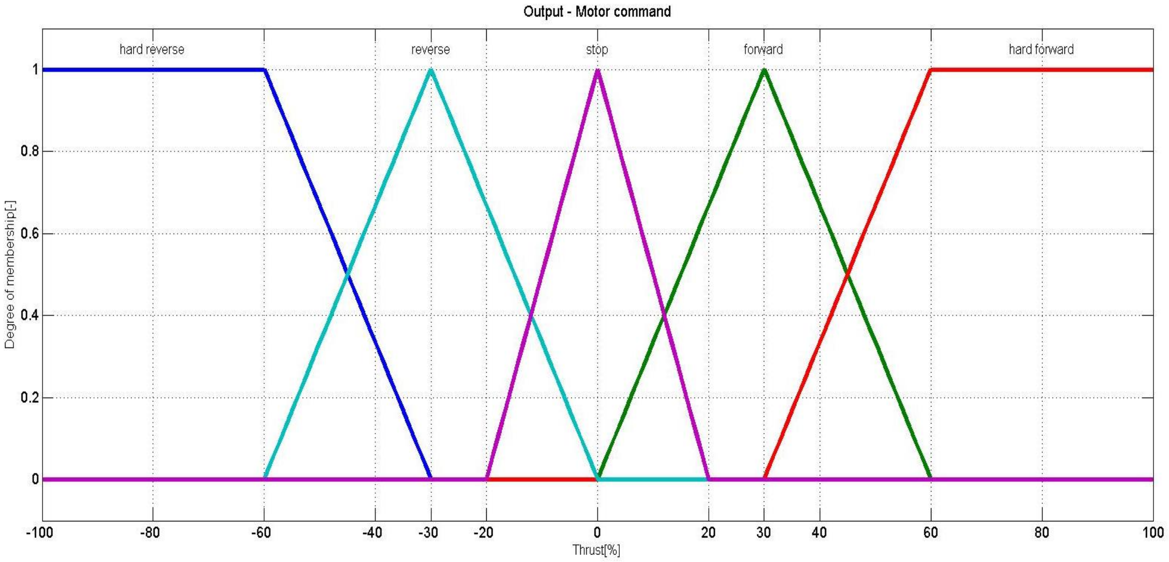


Figure 5.5. *Output from the fuzzy logic controller.*

As in the reference paper [17] we are defining set of rules for as all the possible combinations of error and error rate. Our implementation does not assume any symmetry in the motion so ascending and descending are modelled differently.

1. *If* (error *is* really_down) *and* (rate *is* descending_fast) *then* (output *is* hard_forward)
2. *If* (error *is* really_down) *and* (rate *is* descending) *then* (output *is* hard_forward)
3. *If* (error *is* really_down) *and* (rate *is* stationary) *then* (output *is* hard_forward)
4. *If* (error *is* really_down) *and* (rate *is* ascending) *then* (output *is* hard_forward)
5. *If* (error *is* really_down) *and* (rate *is* ascending_fast) *then* (output *is* forward)
6. *If* (error *is* down) *and* (rate *is* descending_fast) *then* (output *is* hard_forward)
7. *If* (error *is* down) *and* (rate *is* descending) *then* (output *is* forward)
8. *If* (error *is* down) *and* (rate *is* stationary) *then* (output *is* forward)
9. *If* (error *is* down) *and* (rate *is* ascending) *then* (output *is* forward)
10. *If* (error *is* down) *and* (rate *is* ascending_fast) *then* (output *is* reverse)
11. *If (error is ok) and (rate is descending_fast) then (output is forward)*
12. *If (error is ok) and (rate is descending) then (output is forward)*
13. *If (error is ok) and (rate is stationary) then (output is stop)*
14. *If (error is ok) and (rate is ascending) then (output is reverse)*
15. *If (error is ok) and (rate is ascending_fast) then (output is reverse)*
16. *If* (error *is* up) *and* (rate *is* descending_fast) *then* (output *is* stop)
17. *If* (error *is* up) *and* (rate *is* descending) *then* (output *is* reverse)
18. *If* (error *is* up) *and* (rate *is* stationary) *then* (output *is* reverse)
19. *If* (error *is* up) *and* (rate *is* ascending) *then* (output *is* reverse)
20. *If* (error *is* up) *and* (rate *is* ascending_fast) *then* (output *is* hard_reverse)
21. *If* (error *is* really_up) *and* (rate *is* descending_fast) *then* (output *is* reverse)
22. *If* (error *is* really_up) *and* (rate *is* descending) *then* (output *is* hard_reverse)
23. *If* (error *is* really_up) *and* (rate *is* stationary) *then* (output *is* hard_reverse)
24. *If* (error *is* really_up) *and* (rate *is* ascending) *then* (output *is* hard_reverse)
25. *If* (error *is* really_up) *and* (rate *is* ascending_fast) *then* (output *is* hard_reverse)

Error Error_rate	Really down	Down	Ok	Up	Really up
Descending fast	Hard forward	Hard forward	Forward	Stop	Reverse
Descending	Hard forward	Forward	Forward	Reverse	Hard reverse
Stationary	Hard forward	Forward	Stop	Reverse	Hard reverse
Ascending	Hard forward	Forward	Reverse	Reverse	Hard reverse
Ascending fast	Forward	Reverse	Reverse	Hard reverse	Hard reverse

Table 5.1. *Set of rules.*

A further partition of membership function could increase the accuracy of the control but on the other hand the controller would need more computational power.

This particular controller will be simulated in the chapter 7.8 and used for the real-time altitude control (chapter 7.10).

6. Implementation

The overall organization of the system is described in the figure 6.1.

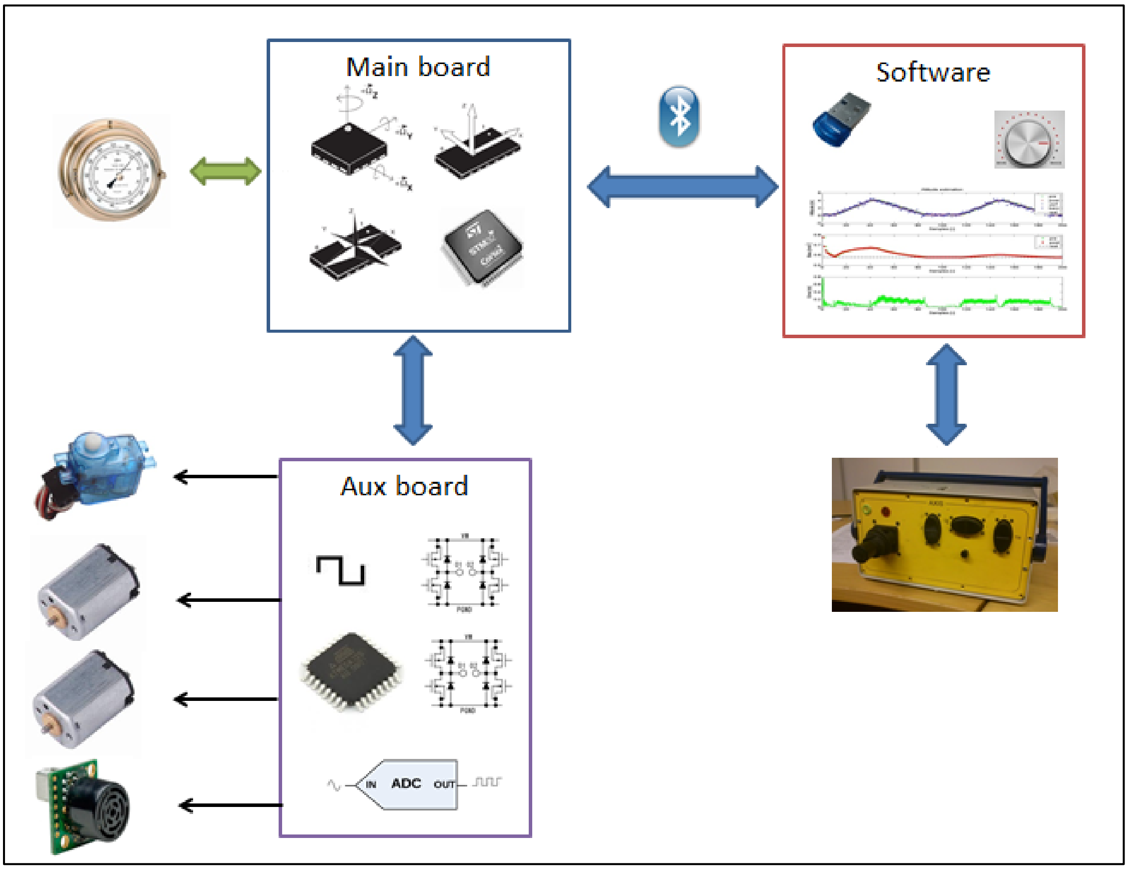


Figure 6.1. Functional block diagram of the system.

The components within the rectangles are physically implemented on the same PCB or software. The arrows represent the physical or radio connections between the blocks.

6.1. Gondola

The frame of the *gondola* contains the electronics and the batteries [20] [21] while the motors [22] are placed on a tillable unit. A tiny servo-motor [23] tilts the *arms* and the motors so to provide a thrust in different directions.

The frame of the *gondola* has been designed ad hoc (Solidworks CAD) so to match the requirements of the 3d printing process. This new technology ensures a custom and affordable design in terms of price, precision and equipment.

3d-printers melt and extrude a plastic filament so to create object with an inner honeycomb structure. The printed parts are very light but, despite the hollow structure, they are really tough.

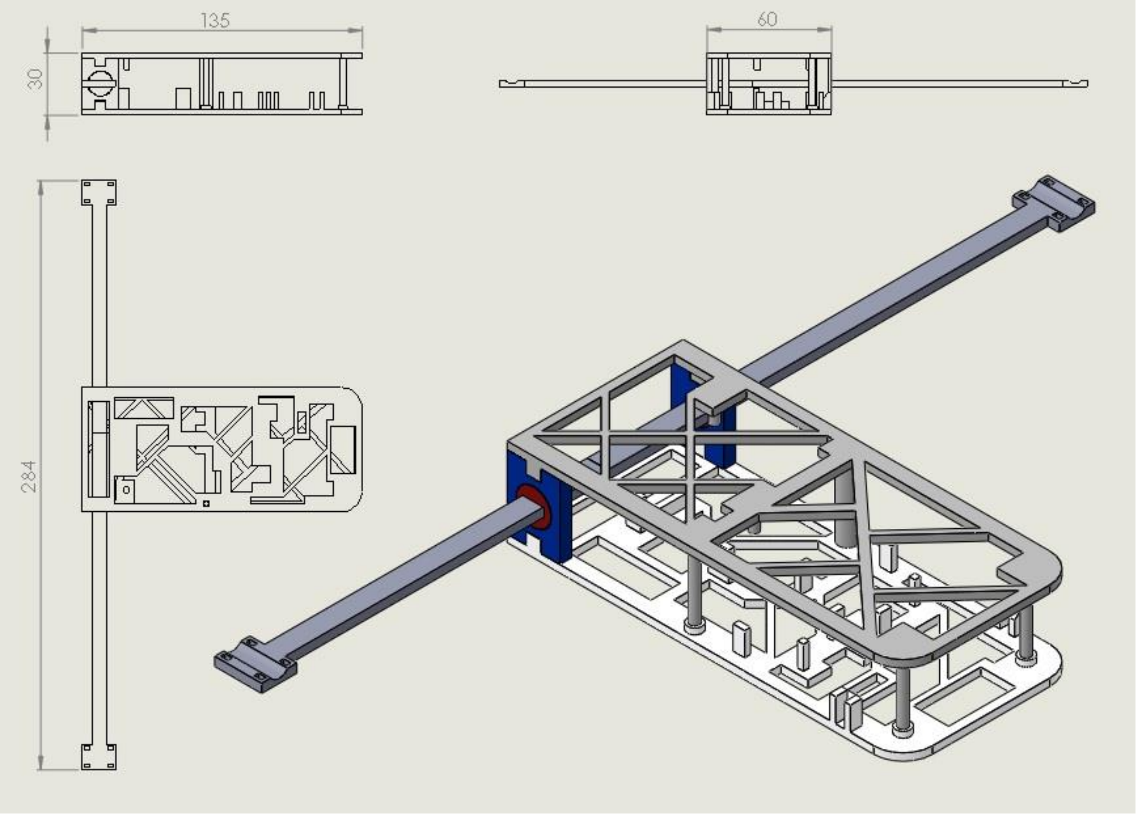


Figure 6.2. *Gondola's design.*

6.2. Electronics

The configuration has two parts: the main board and the auxiliary board.

The inertial sensors (gyro, accelerometer and magnetometer) [13, 24] are placed on the main board together with an ARM Cortex-M3 microcontroller [25]. This is a design made by Timo Pihlström and inspired by the ST Microelectronics INemo boards [26].

The communication with every sensor is performed using two I2C bus. The suite Keil uVision suite is used for the programming and debugging of the firmware. A barometer is soldered on a further PCB and connected on the I2C bus of the MCU using some external wiring. The auxiliary board has the purpose to extend the electronics that was not available on the main board. In fact the IMU board was meant to be just an acquisition and log device. There is no hardware dedicated to the management of the two motors, the servos and to read the ultrasonic range finder measurements.

The Pololu Baby Orangutan B-328 [28] (ATMega 328 MCU [29]) has been placed on a handmade board that directs the power supply partially to the microcontroller and partly to the servos[23], motor controller[30] and range finder[11].

A motor controller with an H-Bridge manages the inversion of the current that flows in the motors and allow the motors to spin independently in direction and speed. The MCU generates a PWM signal with adjustable duty cycle that modulates the speed of the propellers.

On the main board the serial line dedicated to the GPS module has been cut and redirected to the auxiliary board through an extension wire.
This solution is a “workaround“ that many engineers wouldn’t approve. The corresponding ad-hoc SMT design would weight 70 percent less that all this electronics and wires.

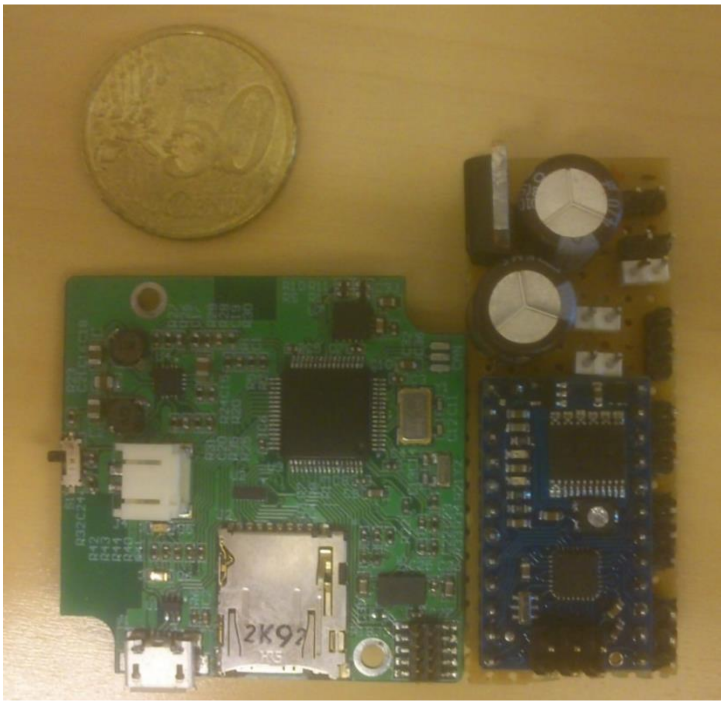


Figure 6.3. *Main and auxiliary board PCBs.*

6.3. Balloon

The envelope [31] is made of a composite of Mylar and nylon with a typical oval shape that can be contained in a box 1 x 0.7 x 0.4 meter. The dimension can change depending on the atmospheric and inner pressure.
Using the equation 1.1 with standard conditions for gasses and a balloon of about $0.12m^3$, we expect about 130g of lifting force.

$$F_{lift} = \left(1.2754\frac{kg}{m^3} - 0.1785\frac{kg}{m^3}\right)0.12m^3 = 0{,}131Kg$$

In case of lower payload than estimated, there is always the possibility to use a bigger balloon. If the frontal surface becomes too big, apart the air damping that it produces, the vehicle will result too wide to pass through the doors and invigilate in a small room.
To reach the balance a set of nails or some rice seed can be loaded until the resulting force is slightly pointing toward the floor. This prevents the blimp to fly freely to the ceiling.

One of the first tasks was to measure the effective load that the balloon is able to lift. This is done increasing the weight attached to it until a slight drop is noticed.

This test is done with a box that weights 98g (screws included) and some nails that weighs about 4,6g each. Eight of them make the airship move slightly toward the floor so they have been weighted to be 36.8g.Considering the scale accuracy of 2g we get:

$$Load = (98g) \pm 2g + (36.8) \pm 2g = 134.8g \pm 4g$$

Note that different temperature, air pressure or gas pressure can affect the amount of payload. The payload of the gondola is described in the table 9.1.

Component	**Weight**
LiPo 1S 3,6V 350 mAh 20C	12g
Main board	12g
USRF MB1220	5g
Aux board	24g
Frame	24g
LiPo 2S/1P 7,4V 350 mAh 25C	26g
Micro Servo Hitec HS 55	24g
Wires	7g
Sum	134g

Table 9.1. *Balance of payload.*

In the balance there is unexpectedly reported the weight of the wires. In the early phase it was not considered to be important but actually the weight of the copper is significant. The design of the gondola is the fourth to be produced. Every new release is very close to the previous, but had the aim to correct small alignments and especially to reduce the weight. A continuous removal of material from the design and a larger infill has reduced the weight of the frame from 36 g to about 24 g of the last version.

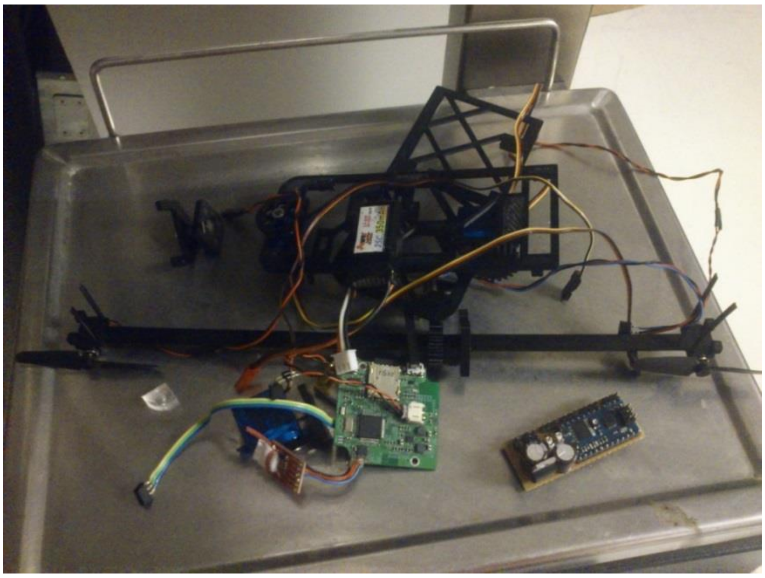


Figure 6.4. *Weight-scale and all the components that compose the payload.*

6.4. High level software

Several scripts have been implemented in Matlab. The functionalities are very focused on little tasks like acquiring measurements or processing the data.

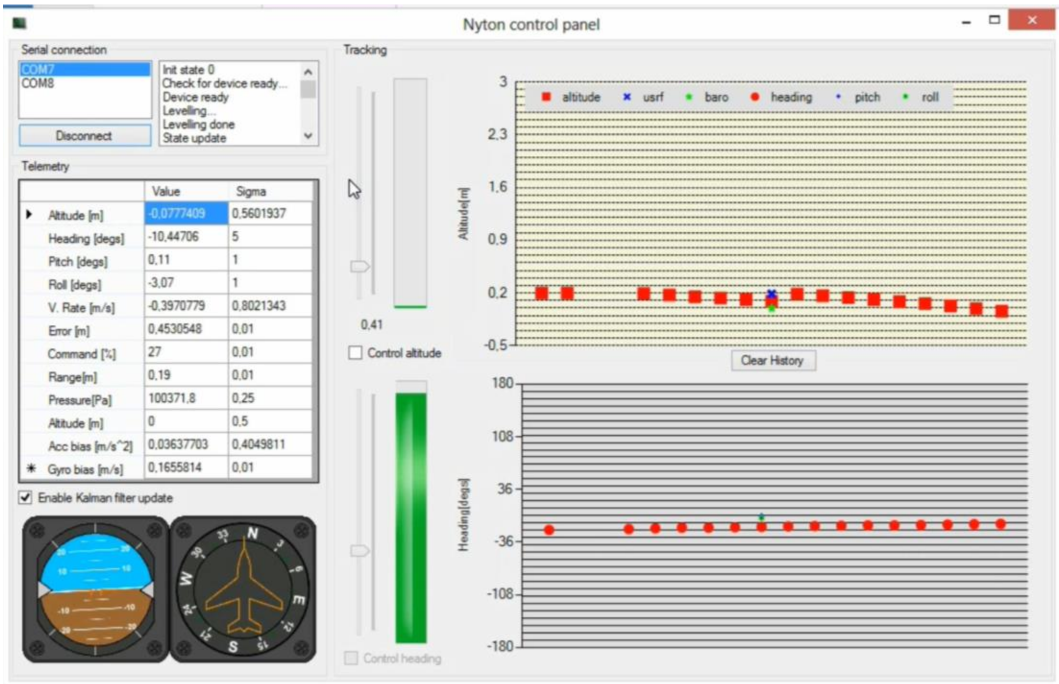
A Matlab's script originally computed the KF and the fully logic algorithm to perform real-time control. This is because the scripting capabilities allow a fast correction and implementation and there are several libraries that are ready to use. Using ready components speed up the realization of the prototypes. Especially the multiplication of matrices, the inversion and the fuzzy logic controller are built in and therefore they do not need further debug.

On the other hand the software is not optimized and quick in its execution. Several times, the plot of real-time samples despite the low rate (10 per seconds), has revealed to be a very heavy operation to perform in Matlab.

Because of these delays all the algorithms have been coded in Visual Basic.NET (interface in the figure 6.5). The real-time execution is very fluid if compared to Matlab.

The same high level software, allows the control setting of motor speed, servo angles and the manual control of the vehicle. In this mode, the commands are coming from a special control unit that has a joystick and several switches.

The vehicle can communicate with the control station through a Bluetooth link. A special Bluetooth adapter [32] is used to improve the radio coverage.



Init state 0
Check for device ready...
Device ready
Levelling...
Levelling done
State update

Telemetry

	Value	Sigma
Altitude [m]	-0,0777409	0,5601937
Heading [degs]	-10,44706	5
Pitch [degs]	0,11	1
Roll [degs]	-3,07	1
V. Rate [m/s]	-0,3970779	0,8021343
Error [m]	0,4530548	0,01
Command [%]	27	0,01
Range[m]	0,19	0,01
Pressure[Pa]	100371,8	0,25
Altitude [m]	0	0,5
Acc bias [m/s^2]	0,03637703	0,4049811
Gyro bias [m/s]	0,1655814	0,01

☑ Enable Kalman filter update

Figure 6.5. Real-time loops and communication.

6.5. Computation

The block diagrams in figures 6.6 and 6.7 show the main loops and the communication between the MCU on the gondola and the software at the control station. Notice that the radio link has to be opened on both sides and that the initial phase consists in a preliminary communication.

The propagation of the two KF is very light and in it is performed on the vehicle every time that the respective inertial data are produced. A hardware interrupt is raised and the new measurements are fetched from the registers. This happens at a constant rate of 95Hz for the gyro and 100Hz for the accelerometer.

The direction KF predict and update are implemented with this simple code:

```
void compute_heading_kalman_filter(){

     float T[2][2];
     float S2;
     float K2[2];
     float d;
     float C[2][2];

     //predict a priori state estimation(with command)
     x2[0]=x2[0]+w_res[2]*TS2+x2[1]*TS2;//xn=xn+B

     //predict a priori error covariance matrix estimation
     //Pn=F*P*F'+Q
     mult2x2(F2,P2,T);//Pn=F*P
     mult2x2_t(T,F2,P2);//P=Pn*F'
     P2[0][0]=P2[0][0]+Q2;//Pn=P+Q

     if(new_compass_meas==1){//update
```

```
                  new_compass_meas=0;
                  //compute kalman gain
                  S2=P2[0][0]+R2;
                  K2[0]=P2[0][0]/S2;
                  K2[1]=P2[1][0]/S2;

                  //compass-H2*x2
                  d=compass_yaw-x2[0];

                  //x2=x2+K*d
                  x2[0]=x2[0]+K2[0]*d;
                  x2[1]=x2[1]+K2[1]*d;

                  //K2*H2
                  C[0][0]=K2[0];
                  C[1][0]=K2[1];
                  C[0][1]=0;
                  C[1][1]=0;

                  //C*P2->T
                  mult2x2(C,P2,T);

                  //completes the P2=P2-K2*H2*P2
                  P2[0][0]=P2[0][0]-T[0][0];
                  P2[0][1]=P2[0][1]-T[0][1];
                  P2[1][0]=P2[1][0]-T[1][0];
                  P2[1][1]=P2[1][1]-T[1][1];
       }
  }
```

The formal KF multiplies matrixes that have null components. All the zero multiplications are not performed and in this specific case the inverse of the S matrix is just a division.

The direction KF is performed onboard while the altitude KF updates are computed remotely by the control unit because of its high floating point requirements.

Every time there is a new set of measurement available, state vector, covariance matrix and measurements are sent to the remote control unit. This performs the update computation (inverse of matrix) and sends it back to the MCU that will perform some other cycle of propagation on it. The predicted state vectors are sent every 0.1 seconds to the remote station that computes the fuzzy logic controller and sends directly the command to the motors. This may look like a very complicated communication structure, but considering the absence of a better hardware onboard, this is actually a very good performing cycle.

Every message (from both sides) has a special format with preamble, number of bytes message and an 8bit CRC [33].

#	@	N	Message	CRC

***Table 6.1.** Communication format.*

The figures 6.6 and 6.7 are flow charts of the computation performed on the gondola's electronics and on the high level software.

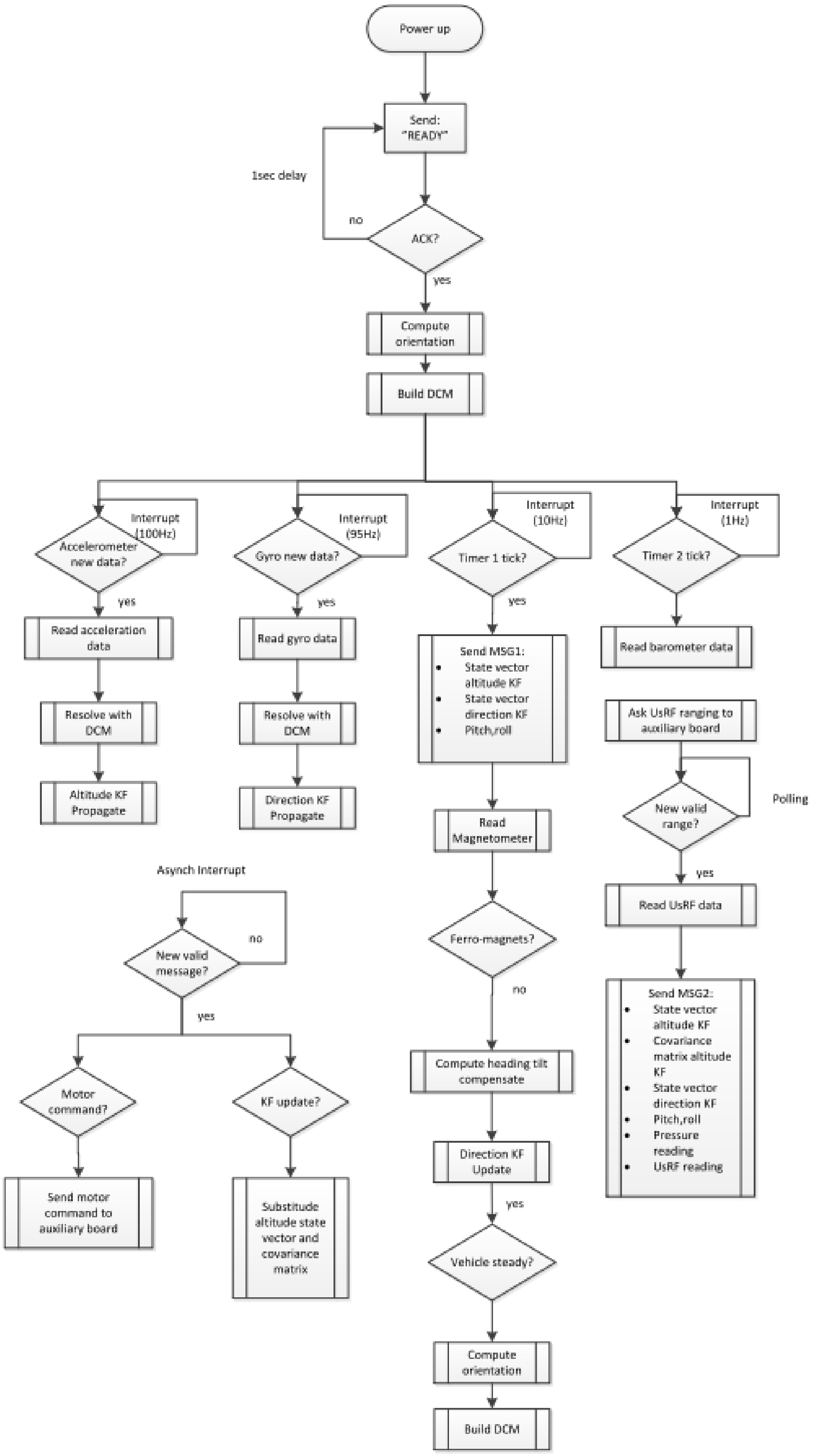


Figure 6.6. *Diagram of the firmware running on the vehicle.*

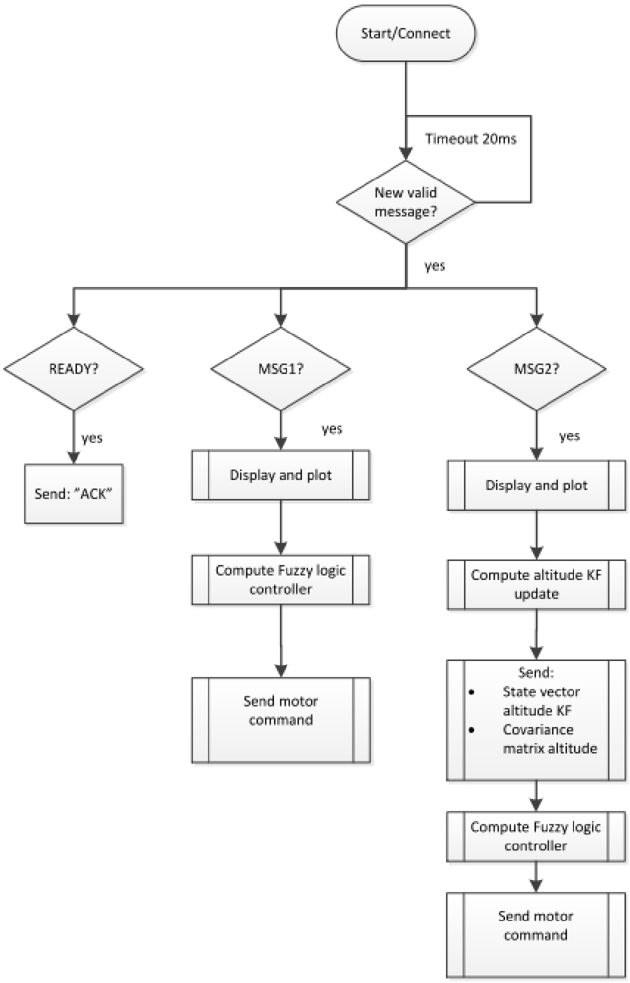


Figure 6.7. *Diagram of the software running on the control unit.*

7. Experimental results

According to Pietro Abelardo as cited in Ad Astralabium pg. 43: “Facta non verba”. During the evolution of the project, many issues rose up and many single easy solutions helped to reduce errors and cross influences in the measurements.

7.1. Validation of dynamic model

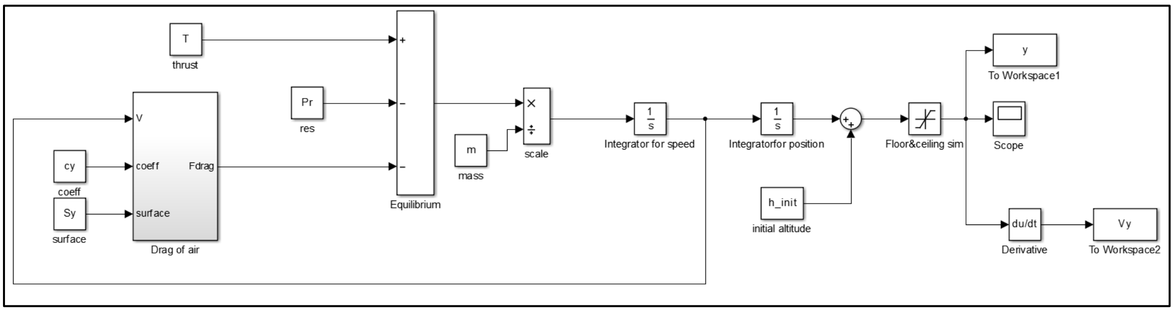


Figure 7.1. *Simulink block diagram used for the validation.*

As expanded already, the kinetic model of the blimp has not the purpose to be very accurate. Any case it is possible to check if it corresponds to reality with some simple test. The set of equations of the rigid body equilibrium (chapter 4) are used to implement a simulation on Matlab/Simulink. The terms of the differential equation (indicated with the dots like:$\dot{y}, \ddot{y}$) are scaled by coefficients and functions evolving in time.

This set of differential equations, once expressed in a first order set and then into a discrete time matrix form, can be solved with finite element integration method. Simulink provides another practical and faster way to proceed. The continuous time equations are converted, through minimum manipulations, to a block diagram that is similar to the electrical simulators used before the personal computer era. There are blocks to perform operations like signal generation, gain, integration and plot. The vertical translation from a fixed altitude will take a period of time to hit the ground during the free fall. The same period, up to some extent, has to be shown in the simulation (figure 7.2). In practice, the blimp was placed at some fixed altitude and a stopwatch was used to measure the period needed to hit the floor.

In the same way the blimp was set with the propellers at full speed from the floor toward the ceiling (figure 7.3). The stopwatch was keeping track of the period needed to ascend so to estimate the trust provided by the motors.

The batteries were fully charged so that the maximum speed can be reached. In theory these are the *free and forced response* of a nonlinear dynamic system.

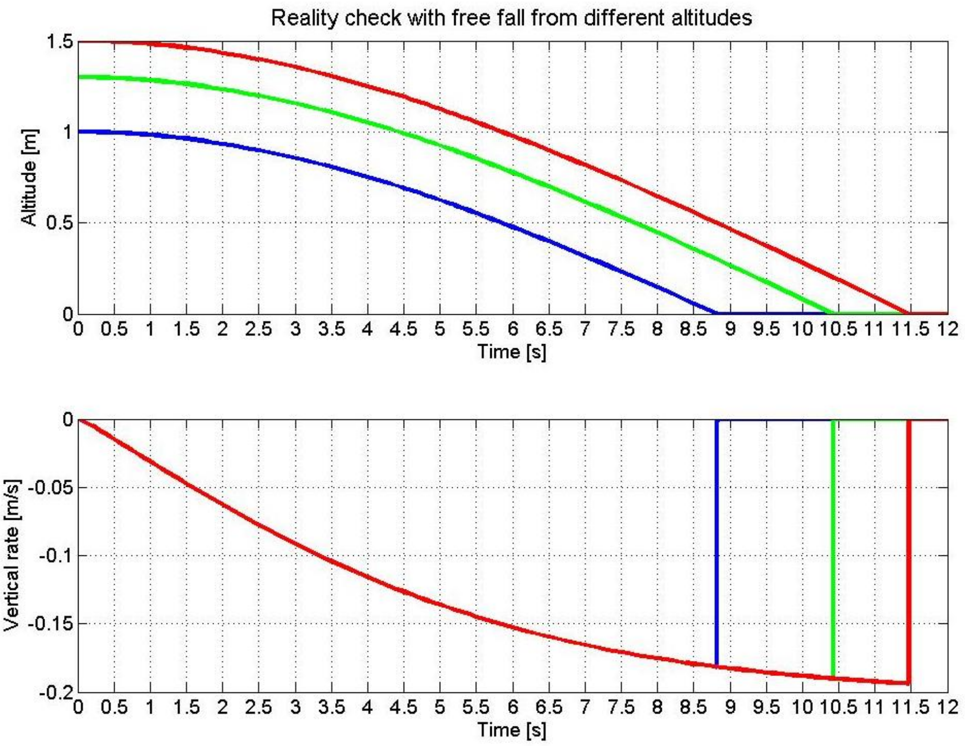


Figure 7.2. *Altitude variation starting from different altitudes.*

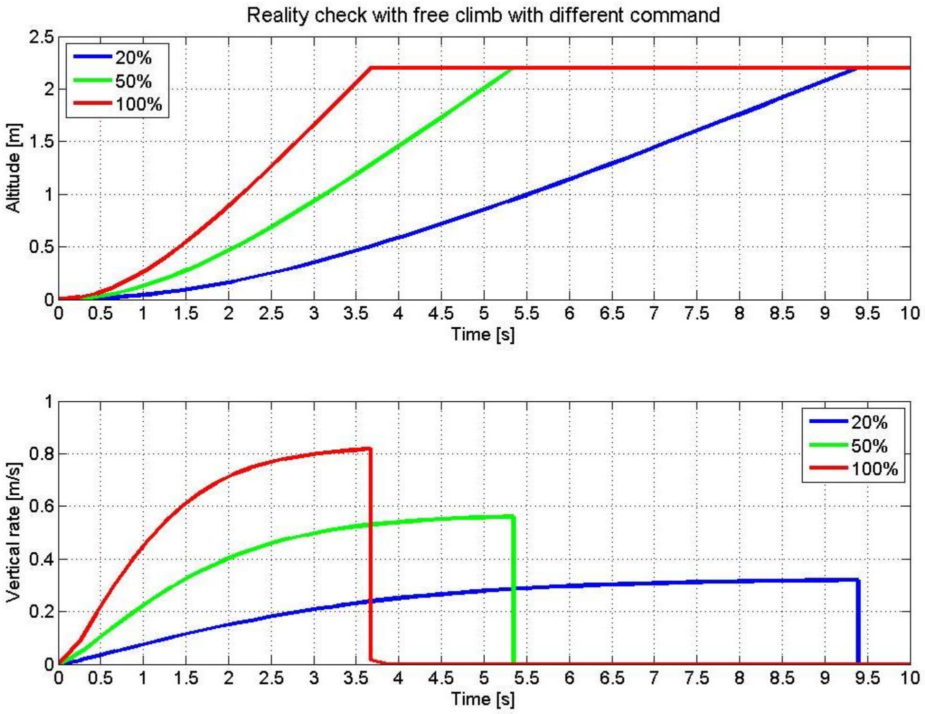


Figure 7.3. *Altitude variation with different motor commands.*

The simulation constants like aerodynamic coefficients, exposed surfaces and the relation between lift and weight have been changed iteratively so to match (as close as possible) the reality. The simulator goes actually beyond the vertical channel and the same check can be done for the translation on the horizontal plane and with the rotation around the azimuth.

At this stage, we have a quite accurate simulator of the blimp that can be used to experiment many different control techniques and having a clue about the control method to adopt.

7.2. Noise characterization

The accelerometer datasheet [12] reports a linear acceleration noise density of $ND_a = 220\mu G/\sqrt{Hz}$ at 1.344 KHz in the range $\pm 2g$. Our configuration uses a lower rate for which it is not provided a proper value, any case it is possible to use the noise density to have the order of magnitude of the expected noise.

$$\sigma_a = g\ ND_a \sqrt{rate} = 9.7803 \left[{}^{m}\!/_{s^2}\right] \left({}^{220\mu G}\!\Big/_{\sqrt{Hz}} \right) \sqrt{100[Hz]} = 0.0171 \left[{}^{m}\!/_{s^2}\right]$$

The same calculations are applied to the gyro using the datasheet [22] rate noise density of $ND_g = 0.03 \left[{}^{deg}\!/_{s}/\sqrt{Hz}\right]$.

$$\sigma_g = ND_g \sqrt{rate} = 0.03 \left[\frac{{}^{deg}\!/_{s}}{\sqrt{Hz}} \right] \sqrt{95[Hz]} = 0.2924 \left[{}^{deg}\!/_{s}\right]$$

The datasheets reports some typical noise distributions that are useful to understand the noise levels but in genre far from the real specific case.

It's a good practice to check if the same amount of noise is experienced on the raw data in a steady state. Steady measurements from accelerometer, gyroscope, magnetometer and barometer have been used since the early phases to compute a parametric statistical distribution of the readings.

The following figures 7.4, 7.5, 7.6 and 7.7 show the result a steady case measurements and the corresponding histogram.

It is possible to notice many effects described in theory like noise and bias. The raw data modulus of the three accelerometer axes reports about $10.6\,[m/s^2]$ that is much more than the gravity acceleration constant we would expect.This result is mainly due to run-to-run bias in each of the three channels.

We can notice that the measurements have mainly Gaussian distributed noise while from time to time some value is far from the distribution. These sporadic values are called *outliers* and there is a clear example is figure 7.6 where the 34th value of the y-channel is $0.3[deg/s]$ while the average is about $1[deg/s]$. This mean value is once again the effect of a bias because the IMU was kept steady during the measurements.

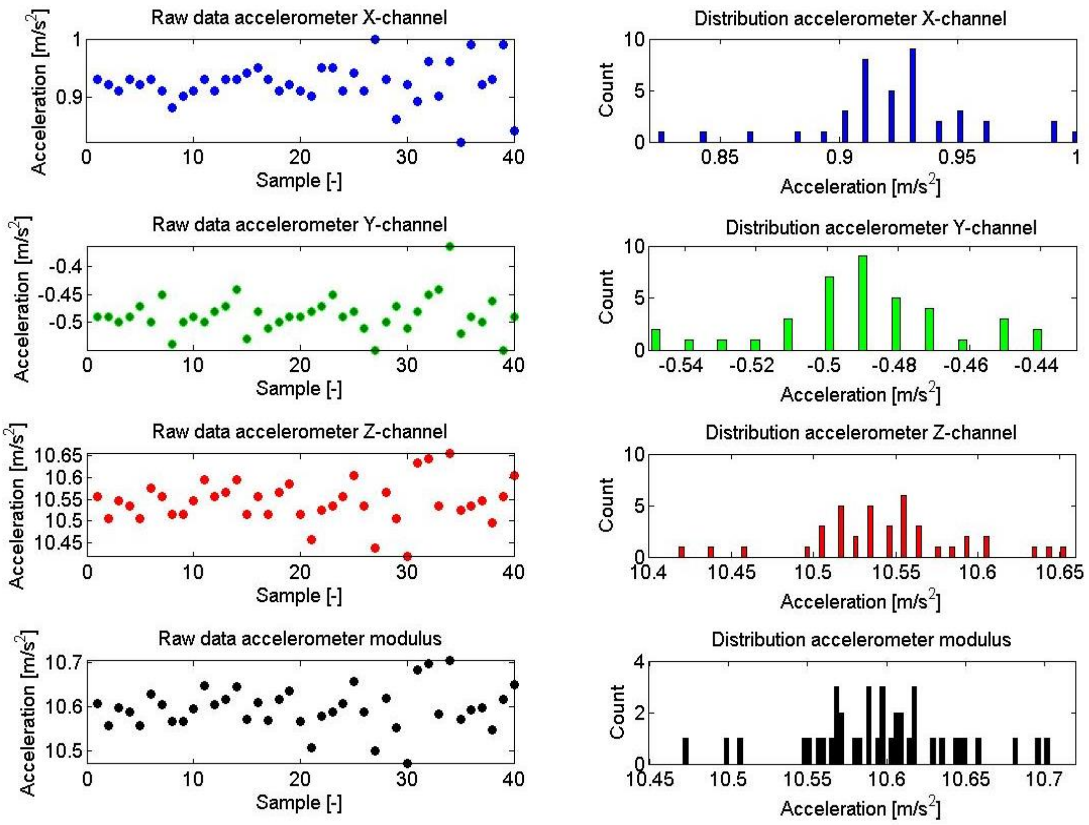


Figure 7.4. *Real raw data from LSM303DLHC 3-axes accelerometer [10] (100Hz).*

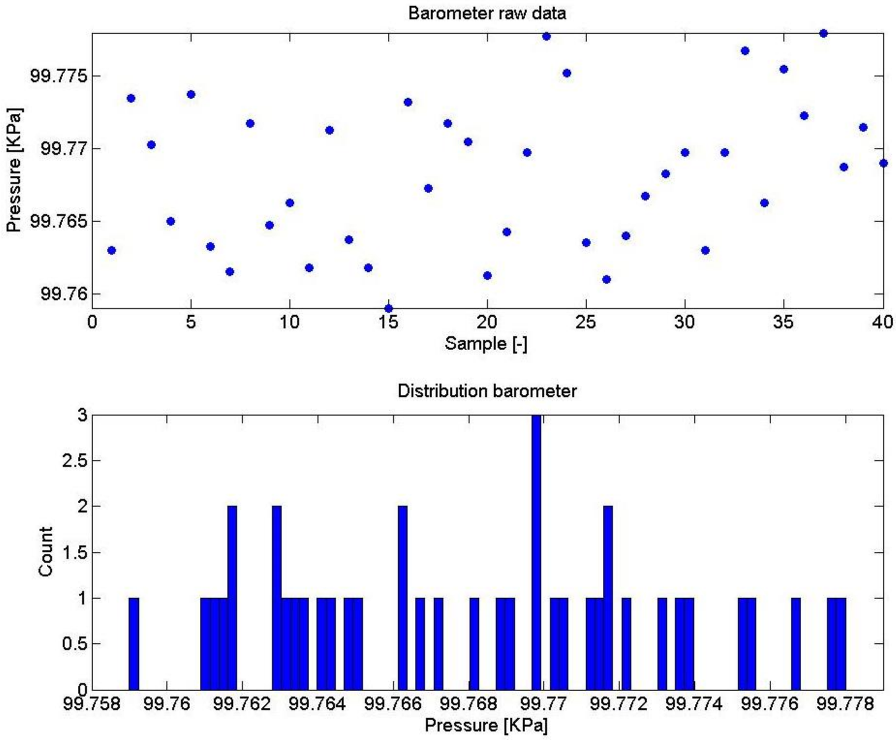


Figure 7.5. *Real raw data from MPL3115A2 barometer [23] (1Hz).*

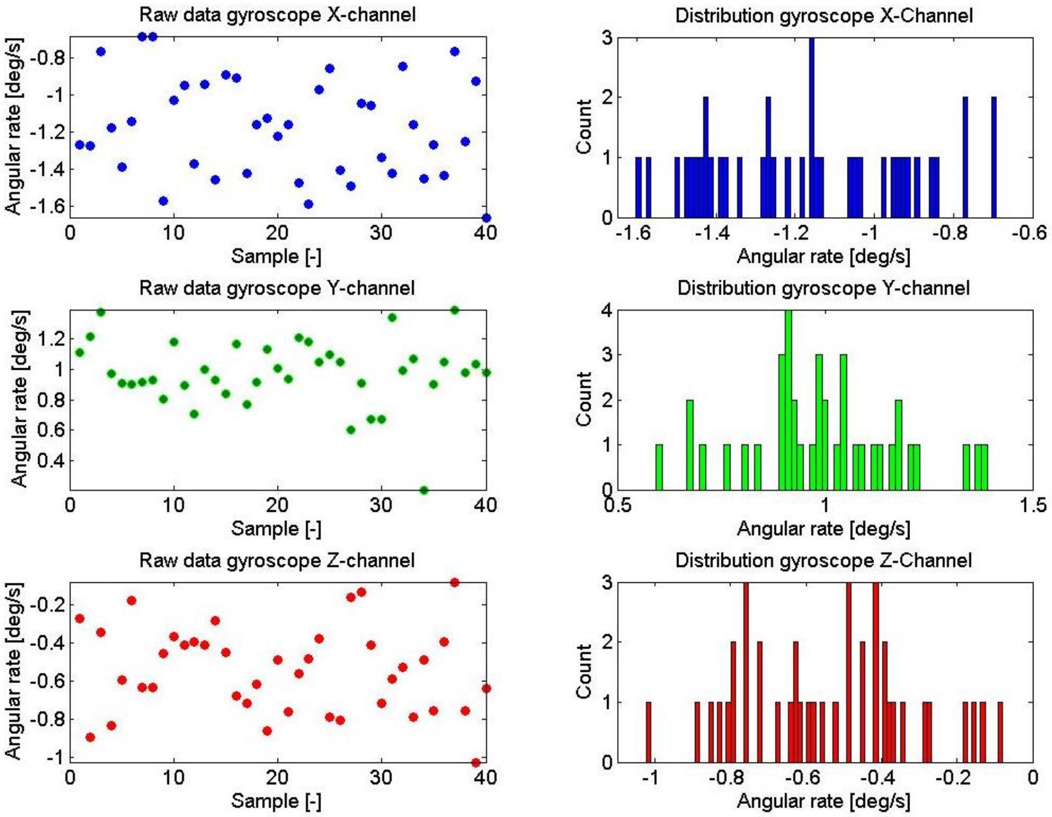


Figure 7.6. *Real raw data from L3GD20 3-axes gyroscope [22] (95Hz).*

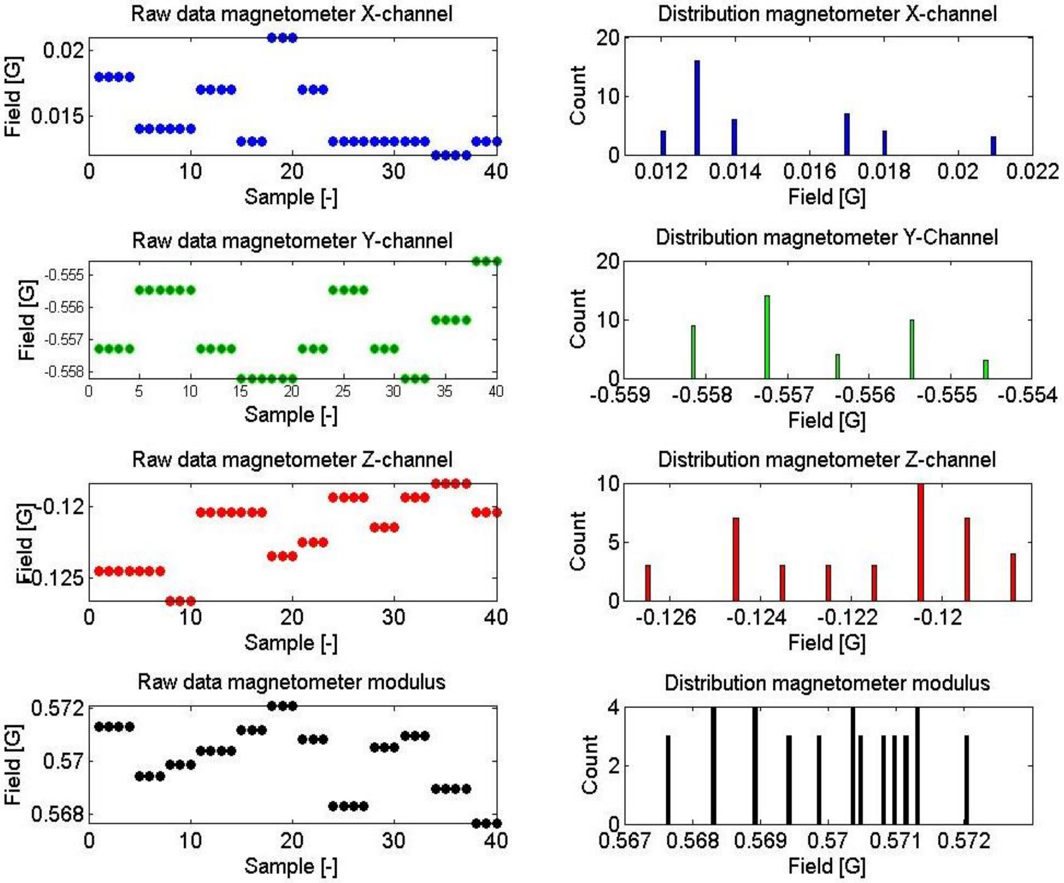


Figure 7.7. *Real raw data from LSM303DLHC 3-axes magnetometer [10] (100Hz).*

Sensor	Range	Measurement rate	Noise level/accuracy	Typical bias	Resolution
Accelerometer	$[-2;+2][g]$	$100[Hz]$	$0.0171\left[{}^{m}/_{s^2}\right]$	$\pm 60\ [mg]$	$1\ [mg]$
Rate gyroscope	$[-250;+250]\ [{}^{deg}/_{s}]$	$95[Hz]$	$0.2924\left[{}^{deg}/_{s}\right]$	$\pm 10\left[{}^{deg}/_{s}\right]$	$8.75\left[{}^{mdeg}/_{s}\right]$
Barometer	$[50;110]\ [KPa]$	$1[Hz]$	$1.5\ [Pa]$	-	$0.057\ [Pa]$
USRF	$[20;550]\ [cm]$	$1[Hz]$	$1\ [cm]$	-	$1\ [cm]$
Magnetometer	$[-1.3;+1.3]\ [G]$	$100[Hz]$	Environment dependant	-	$2\ [mG]$

Table 7.1. *Resume of the main parameters of the sensors*

In the table 7.1 are reported the values corresponding to the particular configuration. In case of different rate or range the accuracy may slightly vary. Even variation of voltage can change the range and the accuracy of some sensors.

7.3. Barometer test

The sensors characterization paragraph shows the accuracy of the barometer but we are interested in the altitude that results from the readings of pressure. Experimentation is the base of the scientific approach and we need to compare some reference altitude with the one resulting from the barometer and its altitude model (eq 2.5). There are sensors that can be tested in place while others need a wide space to show a slight difference in the measurements. The digital barometer in use needs about 0.3 meters of altitude span to trigger a LSB in the measurements.

The Festia building at TUT offers a very high ceiling and some connecting bridge to access the conference rooms. The measurement unit has been dropped from the bridge using a rope so to experiment a different altitude during the acquisition.

In the beginning it was not the gondola to be dropped but a basket that was containing it. This approach, even though safer that the direct mount to the gondola, didn’t allow the proper air flow to the barometer. In the measurements made with the sensors mounted on the gondola, the pressure reading resulted to be improved.

For the sake of an easier visualization in the figure 7.9 (first subplot) there is over imposed the plot of the moving average (10 samples of lag) of the raw measurements.

Figure 7.8. *Festia building (left) and sensors drop from the bridge (right).*

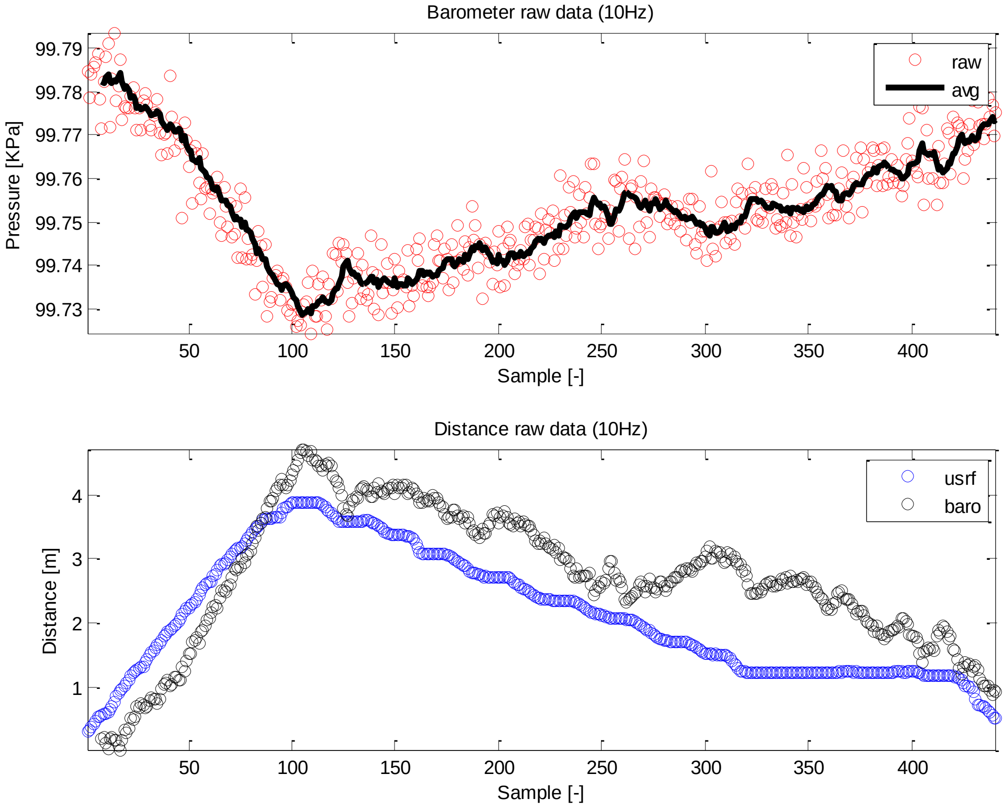


Figure 7.9. *Raw and averaged measurements of pressure (up) and corresponding altitude (down)*

Considering the UsRF reading as reference for the altitude, the altitude resulting from the pressure readings have a maximum error of about 2 meters. With such a slow vertical rate the errors due to limited bandwidth and delays are only in a small way. The pressure readings are at the best of the sensor possibilities and this maximum error is due slightly to the accuracy of the pressure measurements and vastly to the atmospheric model.

In the indoor environment this empirical model may not be accurate and an alternative approach consists in the punctual mapping of the pressure at different distances from a reference level (floor). The idea is to build a lookup table that at a given pressure returns a corresponding altitude. This solution provides a better resolution since the pressure is the one experienced that day with that precise temperature and with that airflow.

The best way to measure the pressure is to put the barometer with the membrane directed parallel to the main airflow. This avoids the PCB to shield the airflow or to have the membrane directed toward it. Both cases return a misreading.

It is common among the multi-copters community to cover the little hole of the barometers with a thin layer of cotton. This acts as low pass filter and avoids sudden variations of pressure.

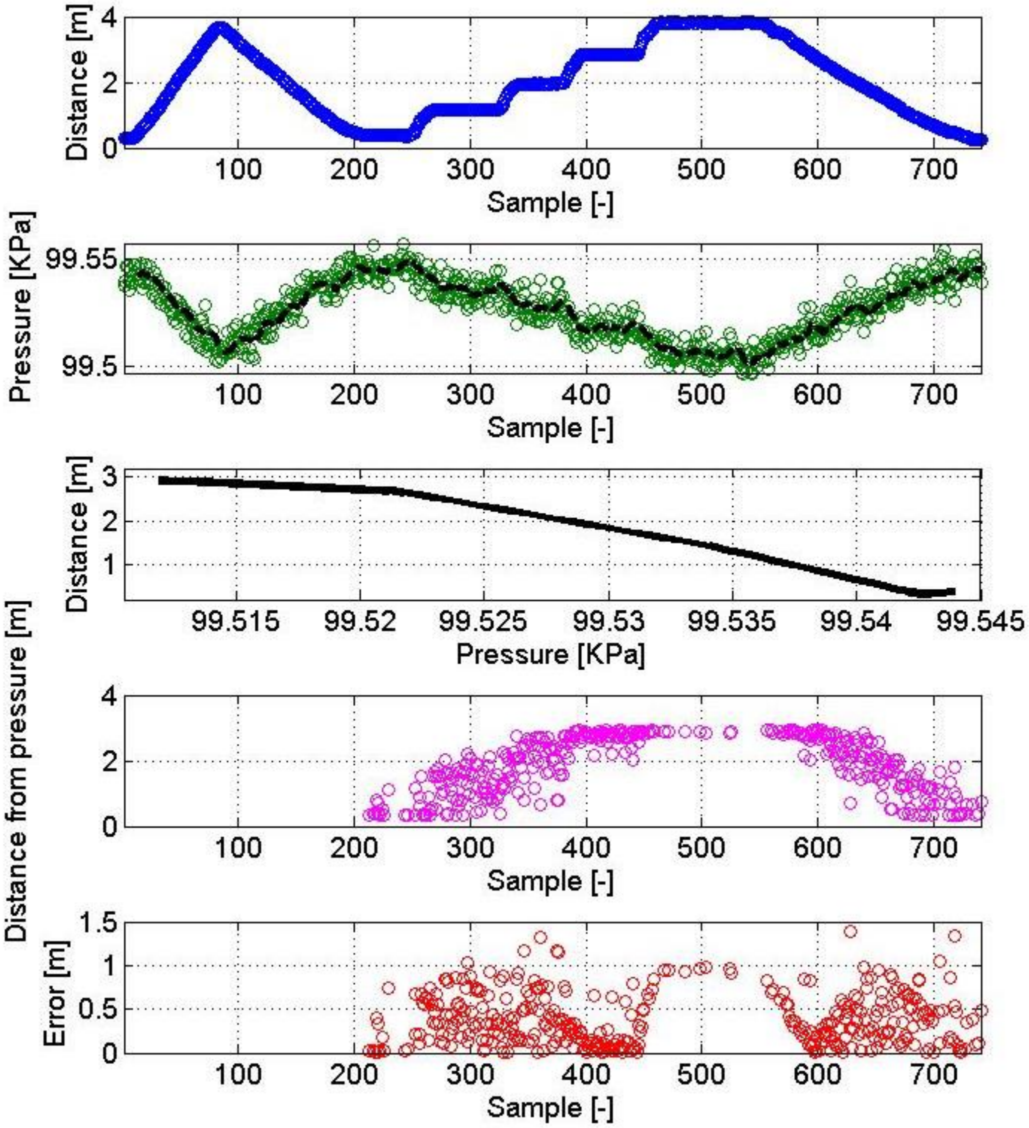


Figure 7.10. *Indoor lookup table for altitude corresponding to pressure*

The figure 7.10 shows how much we can reduce the error due to the altitude model. The first 200 samples of the acquisition averaged with a time lag of 10 are used to build a lookup table of pressure to corresponding altitude.

The following samples are used with the lookup table to compute the altitude from the measured pressure (not averaged). The maximum error is about 1.4 meters while the standard deviation of the altitude is about 0.8 meters. It should be noted that the airflow produced by a conditioning system or the opening of a window may influence the pressure and consequently the resulting altitude.

7.4. Barometer issues

The atmospheric pressure is affected by the airflow of the air conditioners but the influence is not known deterministically. Only a stochastic parameterization is possible.

During the late phases of the project, once that the main issues were analyzed and compensated, it has been possible to notice a cross influence of the barometer measurements. The two propellers receive a command from the control loop controller in order to maintain the altitude. Depending on the speed of the propellers the airflow coming from them influences very much the pressure experienced on the membrane of the barometer.

The solution would be to place the barometer farther from the interference source or to design a mechanic protection for it. The orientation of the barometer has been changed so to reduce this phenomena. Unfortunately, at this stage this is the maximum possible variation of the design.

The variance of the measurements increases proportionally with the RPM set at the motors. It appears that a faster rotation of the propellers produces an increment of pressure under the vehicle where is located the barometer. On the contrary, the inversion of rotation produces a depression. This deterministic influence can be managed with the increment of standard deviation of the pressure measurements (KF) when high RPM are set.

The vehicle has been placed at a fixed altitude and the propellers have been set to increase slowly their speed. Readings have been processed with the given RPM so to produce a first order polynomial.

$$\sigma_b(RPM) = \sigma_0 + \sigma_1 RPM \tag{7.1}$$

7.5. Coherency of data

In the figure 7.11 is possible to notice a very big problem related to the nature of the range finder.

During the acquisition performed in Festia a student was passing by chance under the gondola. The range finder was then measuring a distance from the tip of the sensor to the head of the student. Unfortunately we are not trying to estimate how tall are the students of the TUT’s campus. You can see from the plot how the barometer, despite its very bad accuracy, it is not effected by the passage of the moving object during the estimation of altitude.

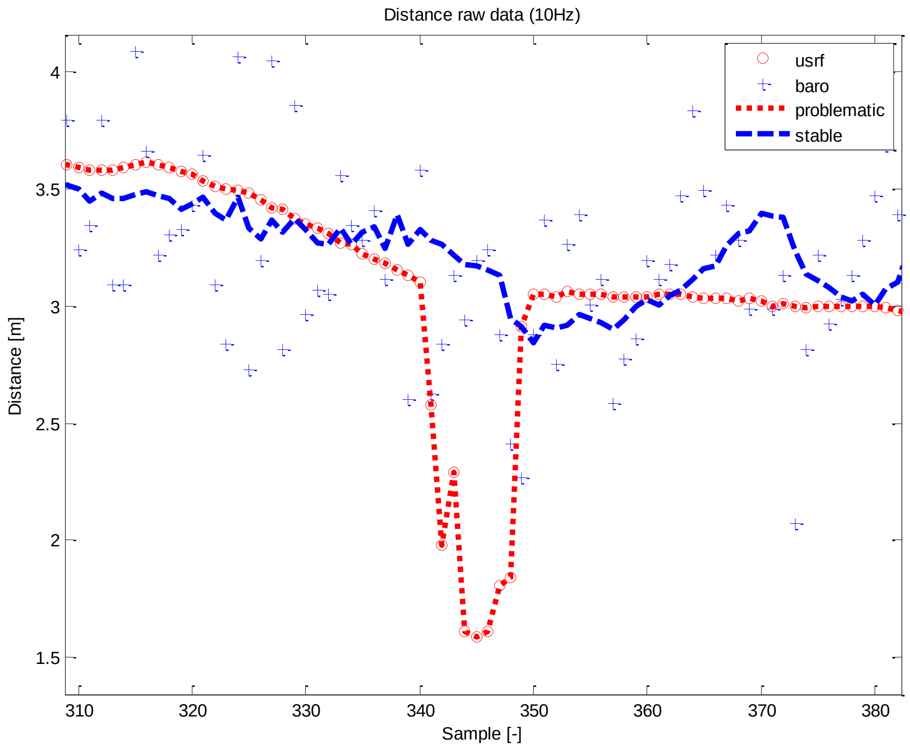


***Figure 7.11.** Incoherent data from the UsRF.*

Because of this reason it is necessary to handle with care the data provided by the range finder. Because of some outlier or change of mode of the series of measurements it is important to check that the innovation is consistent with its covariance by verifying that the magnitude is within the range of 95% likelihood.

$$i \in \pm 2\sqrt{S_k} \tag{7.2}$$

Only if the magnitude of the innovation (equation 7.2) [34, p. 18] is within this bound it can be considered as consistent and further used in the update.

In this application the consistency is checked before the update and if it violates the bounds only the barometer is considered to be consistent for the update. This avoids missing updates since the IS alone tends to produce dramatic drifts.

7.6. Simulations of altitude and direction estimation

The two KFs for altitude and direction estimation have been simulated in Matlab. The simulation in the figure 7.12 and 7.13 shows the computation of the propagation every 0.1 seconds (pre) while observations and measurements are provided every second (post). The convergence of the real and estimated position/heading converges after the stabilization of the bias estimate.

The purpose of the data fusion is to improve the accuracy that the single sensors would provide. After convergence the result of the Kalman filtering provides an estimate that is about 10-20% more accurate than the most accurate sensor used to feed it. In this case (low dynamics) the bandwidth is comparable with the one of the inertial sensors.

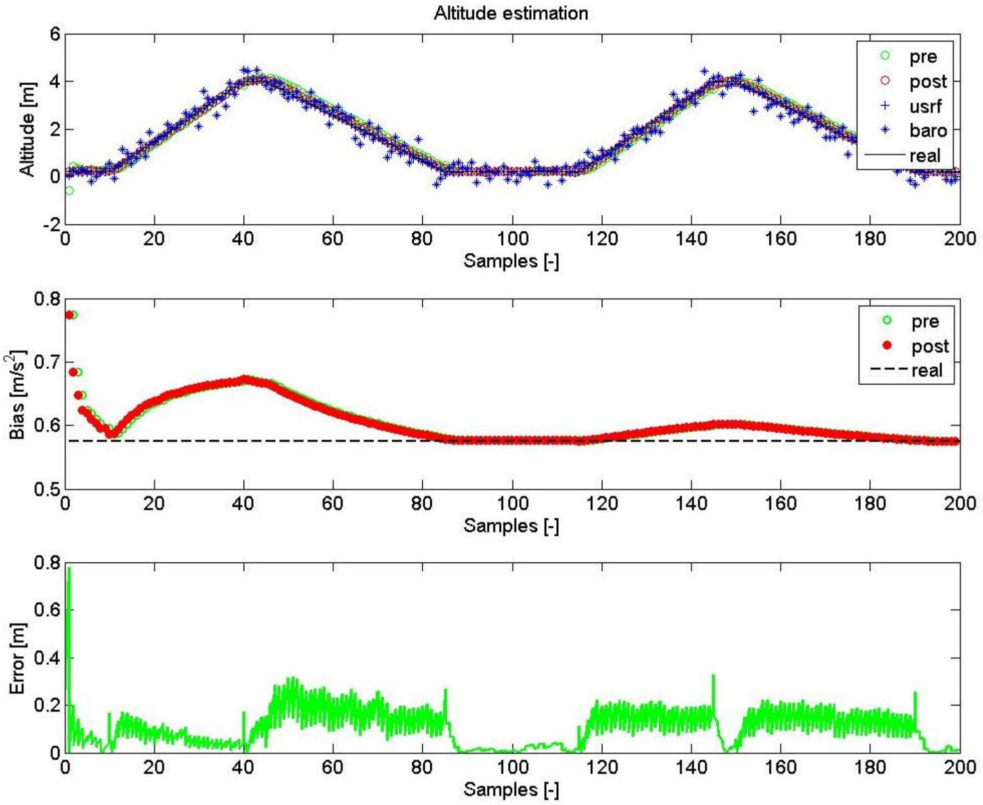


Figure 7.12. *KF simulation for altitude estimation.*

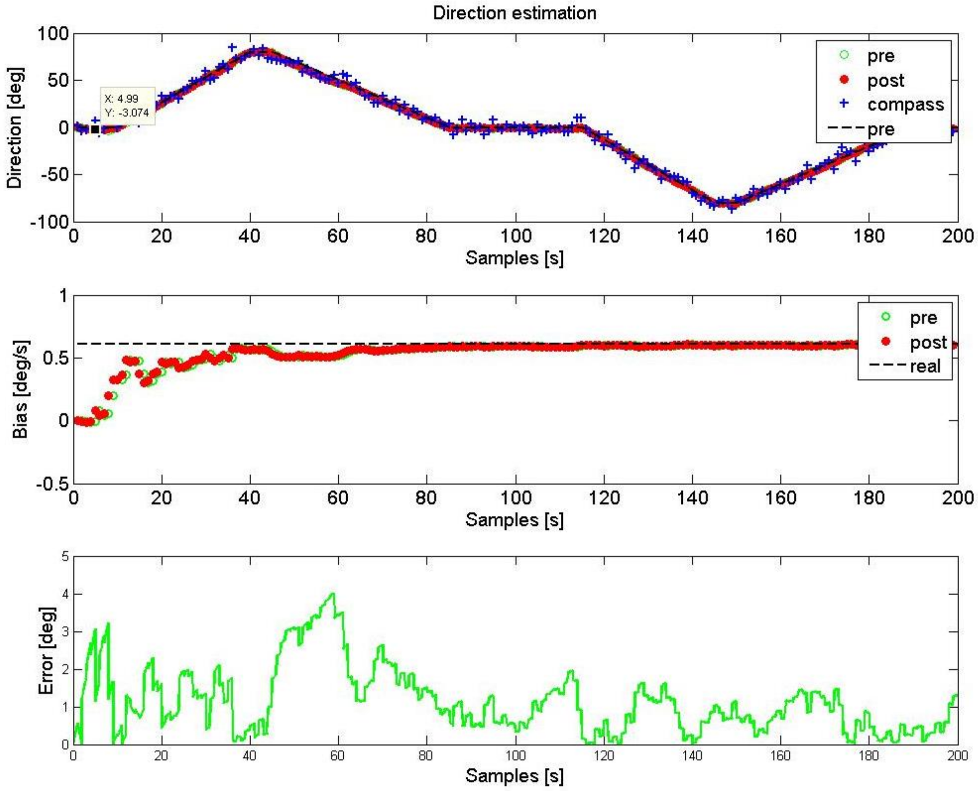


Figure 7.13. *KF simulation for direction estimation.*

7.7. Tuning by using the test platform

The Kalman filters used in this application are not complicated from the mathematics' point of view. In a controlled or simulated reality they converge quite quickly to the best statistical estimate every time that the parameters are exact and the assumptions are founded. In a true system with true dynamics and true sensors the KFs work very well only if the parameters correspond very accurately to reality.

The most of the times the working parameters are slightly different from the theoretical and the identification is done by slightly changing the parameters until the system has the expected response. The hardest task is the definition of the system noise covariance matrix (Q). The correct estimation of its components plays a very big role because it specifies how much the KF should trust the prediction of the state variables. In this case it expresses how big is the noise and error in the measurements of acceleration, the variation of bias and so on.

It is indeed a measure of how much to rely on the model of the system or on the series of correction measurements. Any case all the stochastic parameters play a role only during the independent measurements incoming and update step. The a priori phase does not use at all the covariance matrix to predict the state vector. It is a very bad idea to rely on elements of the covariance matrix of state estimation uncertainty even after and several updates.

Considering the grade of the accelerometer, it is important not to set directly the measurement noise in the Kalman filter. They are expected to work in the simulator but in reality a process of iterated attempts ensures the correctness of the parameters.

The same concept is applied to the parameters $\sigma_{p,}\sigma_b$ that are supposed to be zero because are both results of computations. In nature the noise is omnipresent and the floating point computation produces small errors. Because of these reasons they are arbitrary chosen to be very small in magnitude.

Kalman filters provide information about the accuracy of the estimate. As we discussed, the diagonal elements of the covariance matrix of state estimation uncertainty are just a theoretical clue about the accuracy of the estimate. How to check the effective errors in the altitude tracking?

For this purpose has been built a very simple platform using industrial dismissed components. As shown in the figure 7.17, the base has a rail that makes the white platform change altitude. On this rail there are 3 markers at different altitude.

It's enough to change the altitude and reach a specific reference point while the real-time tracking is running. For instance you can keep on change distance at a various rate (from test to test) and stop at 3 specific reference points. This would offer a comparison between the estimation and the real situation.

The only limit is the maximum altitude covered by the rail that is about 0.8m. At this altitude the variation of pressure are almost invisible therefore the barometer offers just a little clue about the altitude.

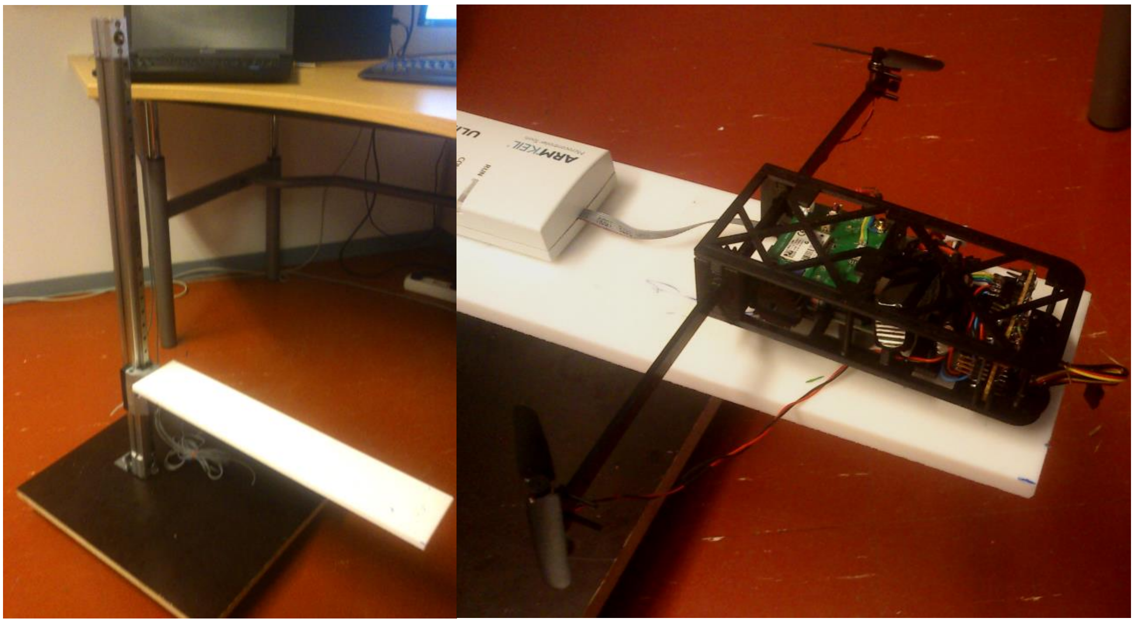


__Figure 7.14.__ Test platform for short range altitude estimation.

After the convergence of the bias estimate and at a fixed altitude, the covariance matrix reports a standard deviation in the altitude tracking of about 1cm. This corresponds to the range finder resolution and accuracy. Since the altitude is kept manually by an operator it is not possible to determine the level very accurately.

If we assume that the operator can reach and maintain the altitude with 5mm of accuracy, the estimate and the real accuracy seems to match. Industries use mechanized and very accurate platforms to test the tracking. These machines are far beyond our interest and possibility.

7.8. Simulations for altitude keeping

This simulation uses the results of the previous test platform and the kinetics and estimation simulations. The raw observations are fed to the KF that provides the best estimate fusing the measurements with the knowledge of the system dynamics. The output contains errors that can be modelled as a further virtual sensor with performance that are slightly better than the sensors that originated the observations.

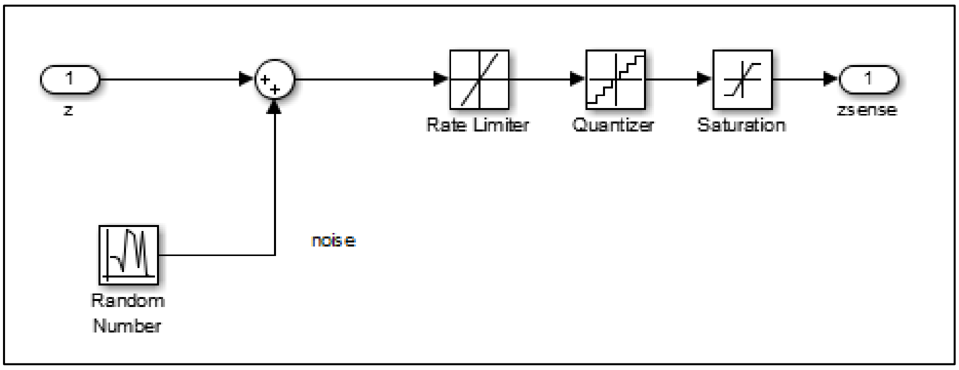


__Figure 7.15.__ Sensor simulator block diagram in Simulink.

Another advantage in the use of Simulink is the possibility to interrupt the line of feedback from the true simulated parameter and insert a block that simulates the noise and the imprecision of the sensors (figure 7.15). In this way the virtual sensor is modelled as the output of the KF that provides reading affected by white noise, limited bandwidth, digital quantization and a saturated range.

The fuzzy logic controller reported in the previous chapter is directly implemented and connected to the output of the virtual sensor in order to close the feedback loop. The settlement value will oscillate around the set point with a distance that is related to the accuracy of the feedback signal. A as rule of thumb, in order to have an accurate control you need to have a feedback signal that is 10 times more accurate than the span of settlement value required in the control. Even the command is quite aggressive and will never reduce toward the neutral. It's possible to notice in figure 7.16, the presence of the noise in the error and vertical rate and how this affects the performances of the control.

The noise therefore plays a very big role in the accuracy of the control. It is important to notice that this is just one of the possible tuning of the controller. There are no formal requirements for the settling time or the 10-90% period of the airship response. It is just important to minimize the command and the RPMs of the propellers because it related to the power consumption and autonomy of the blimp.

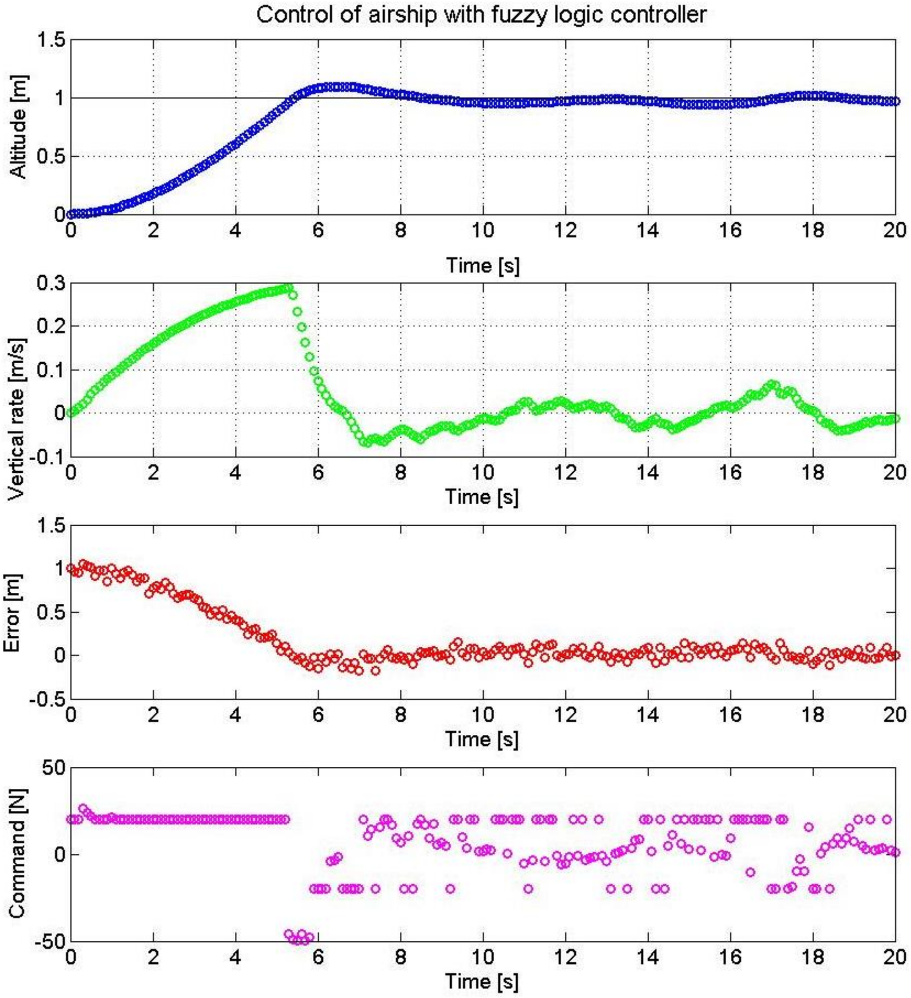


Figure 7.16. *Simulated control loop step response.*

7.9. Real-time direction estimation

The direction is estimated through a secondary real-time KF that merges measurements of the gyroscope and measurements from the magnetometer.

The heading is for simplicity referenced to the magnetic North and the tests have been done with the reference paper shown in the figure 7.17. It has been aligned with a standard compass toward the magnetic north.

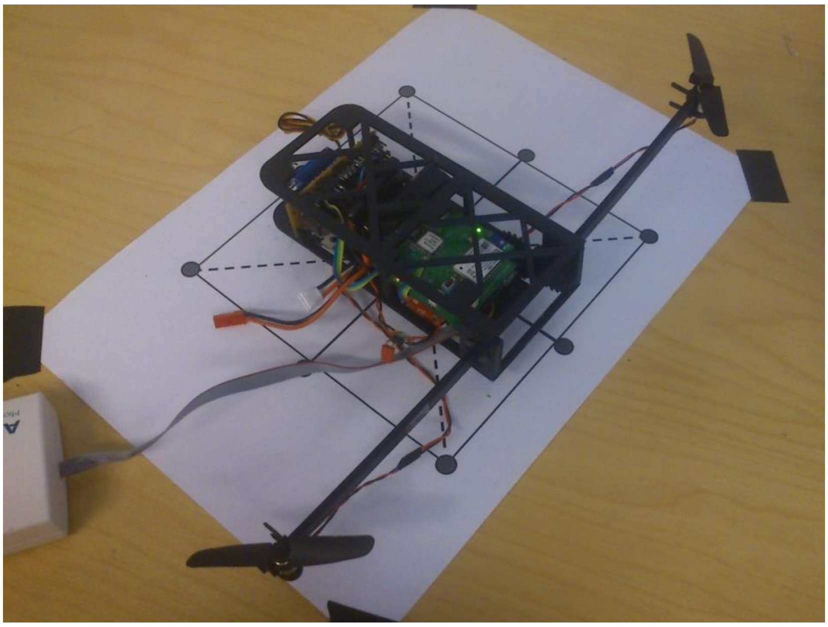
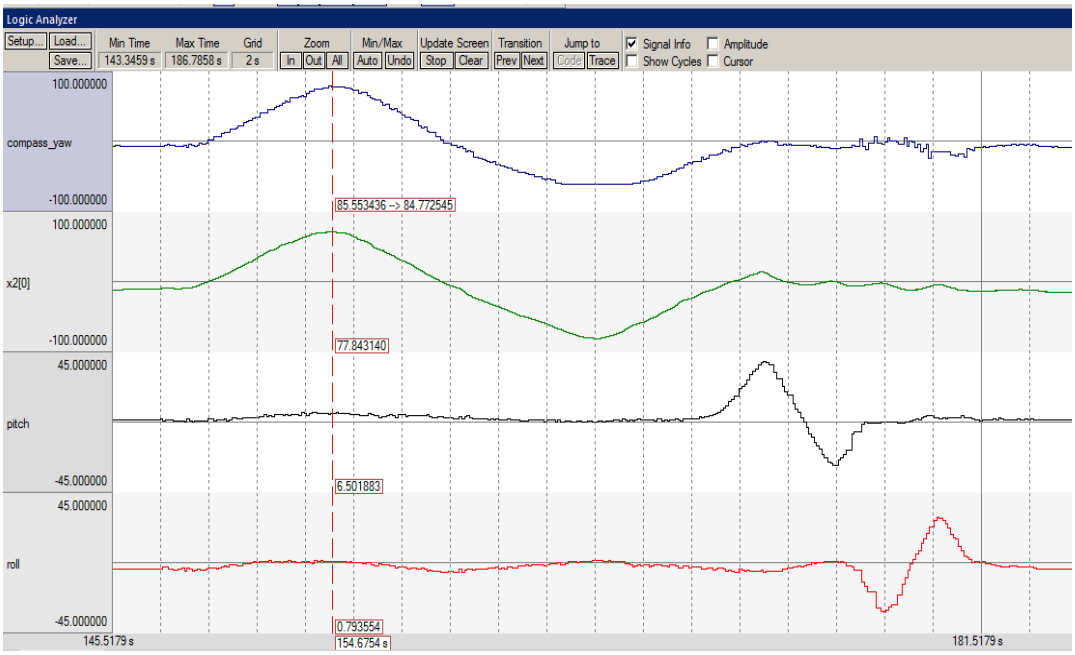


Figure 7.17. *Direction reference and real-time tracking of heading.*

The magnetic measurements are tilt compensated so that changing of attitude in the pitch and roll do not affect the heading estimation.

This is clearly visible in the real-time trace plot of the microcontroller debugging tool of the figure 7.18. It is not possible to estimate the real accuracy of the tracking because North reference and a mechanical test platform are not available. This kind of test goes beyond the budget of time and money available for this thesis. As rough estimate, the orientation estimated by the KF has an absolute error of about 5-10 degree.



Min Time 143.3459 s
Max Time 186.7858 s
Grid 2 s
☑ Signal Info
□ Amplitude
□ Show Cycles
□ Cursor

Figure 7.18. *Real-time trace with JTAG on the MCU state.*

7.10. Real-time altitude keeping

In the figure 6.6 of the implementation chapter it is visible the initialization and levelling procedure. This is done when the blimp lies on the floor and there are no movements. After this phase, the controller and the KF estimate run together and they work to maintain the reference altitude. The step response is a typical second order dynamics with an overshoot and a cisoidal oscillation toward the gradual settling.

In the figure 7.19 at about 8 seconds (80 samples), the set point (1.2 m) is reached and maintained with a span of about 5 cm. As expected the control performances are very much influenced by the accuracy of the altitude estimate. As we saw in the simulations, as rule of thumb you need a feedback signal 10 times more accurate than the span of the settlement value required.

After the settlement, some malevolent data are introduced covering the UsRF with the hand. At first it is coverage sporadically but from the 140th sample the coverage lasts longer and longer.

The coherency check understands that the data are incoherent and the range finder observations are not fed to the KF that uses only the series of measurements from the barometer.

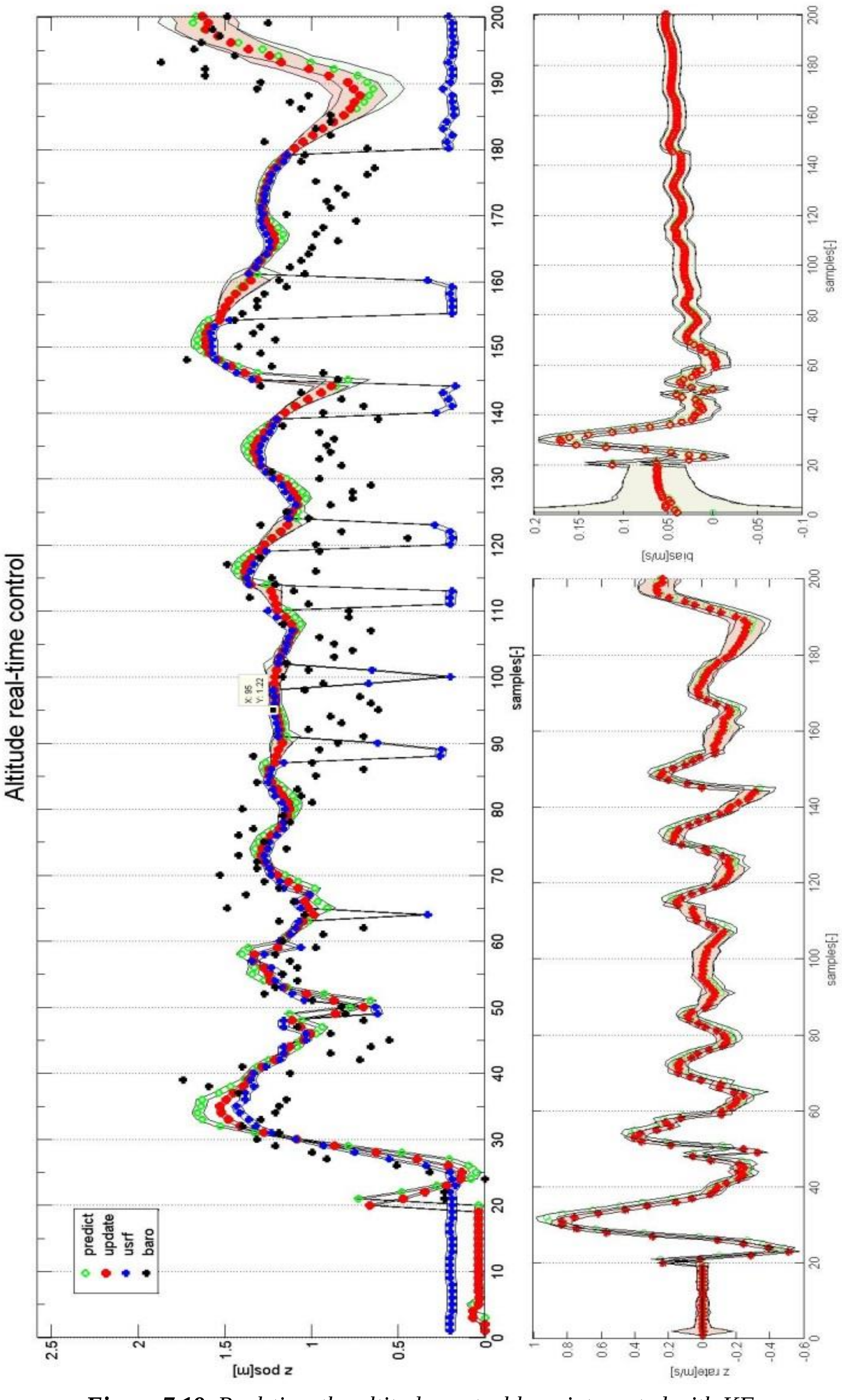


Figure 7.19. *Real-time the altitude control loop integrated with KF.*

For many reasons (explained in the previous paragraphs) the barometric readings are not in the same mode of the estimate but any case the innovation resulting from it is within the coherent range. Never the less they are the only series of measurements available (UsRF detected to be incoherent) and therefore they drive the estimate far from the altitude set point. The controller believes that there is a greater error with a high error rate. This leads the decision of increasing prominently the RPM of the propellers.

Between the samples 160 and 180 the UsRF is not covered and the altitude provided to the controller is again reliable. The last series in the plot shows the dramatic degradation of altitude estimation and control when the UsRF is covered for a long time.

The measurement noise of the feedback signal grows and the altitude tracking oscillate with a wider range.

Using the only barometer readings as altitude observations produces 1-2 meters of control accuracy that is not acceptable in the indoor environment. Conventional balloons operate at an altitude of the several thousand feet. Besides the indoor application, this control accuracy can be considered a great result.

Notice in the figure 7.19 how the set point is reached only when the bias (augmented state) converges to the right estimate.

The transparent area around the estimation and measurements represents the measurement confidence range of 95% likelihood (two sigma) from the diagonal element of the covariance matrix of state estimation uncertainty.

8. Conclusions

The indoor altitude estimation and control revealed to be a quite difficult task despite the low dynamic of the blimp. The set point is reached and maintained with a span of about 5 cm. It is possible to see that the blimp oscillates with a period of about 2 seconds but, considering the hardware limitations, this is a considerable result.

The time spent in the kinetics modelling and in the reality check has been good investment. There is a surprisingly good correspondence between the simulated and the real dynamics of the blimp. The choice of the fuzzy logic controller allows a nonlinear control and tuning of the gain. In this way the consumption and the bouncing due to the noise is limited. The methodology used in this application will become a useful tool to spend in the future career.

The interdisciplinarity of the topic and the need of an implementation took a lot of energy. In origin the project was meant to estimate and control the position even on the horizontal plane. The altitude tracking and control was just the first goal to achieve. It became clear, day after day, that the hardware and the time needed for the full implementation would be beyond the possibilities. The most of the things I have been doing for this thesis are completely new for me and have been clarified in progress.

Referring to Plutarch: “*Paupertas artis omnis perdocet*” (The poverty teaches all the arts). In absence of big means you have to learn to do the things are needed to accomplish your goal and sometimes you have to take the risk to fail if you want to learn. Whether you succeed or fail, you learnt something new and you open many possibilities for the future. In my case planning a schedule revealed to be a very complex task. Sometimes unexpected issues (firmware or electrical most of the times) took a long time to be solved because they depend on many variables and the identification of the source through variable elimination took a long time.

I’m sure that in the future the definition of hardware requirements will be more strict and based on *ad hoc* solution and not anymore based on modifications from available solutions.

8.1. Future Improvements

Limitations in the budget forced the use of inaccurate MEMS inertial sensors. These components are the main to be improved in the future version of the vehicle.

The use of higher grade accelerometers and gyros like the Analog Device ADIS16400/ADIS16405 [35] would allow a more reliable propagation using the dead reckoning technique, limiting the drift caused by the noise and the integration of it (random walk). The updates will be less frequent and possibly from a more coherent source.

The absence of a floating point unit on the main board, forced to the radical solution to send the state and covariance matrix to the control unit in order to be solved.

The next generation should carry an MCU equipped with a single/double precision floating point unit. For example the ARM Cortex-M4 [36] family has a built in floating point unit that is capable of image signal processing.

The motor electronics of the auxiliary board can be on the same main PCB and with SMT components. This implies a radical reduction of weight that will allow the loading of other measurement equipment.

The altitude can be measured with an ultrasonic range finder with a narrower beam that avoids the capture of objects to the sides.

A more accurate optical range finder could dramatically improve the tracking performance even though the coherency of the range readings will keep on be a limitation. In this sense, image based range reading can constitutes a breakthrough.

The image detection of landmark is the key to estimate the position on the horizontal plane.

The vertical channel can be ruled easier and in a more effective way, using another motor and propeller that keeps on point toward the floor. In this way the control is directly connected to the SISO altitude control loop.

The accuracy of the barometer can be improved using a higher grade sensor that could be placed far from sources of disturbance (propellers) and protected by a proper shield.

The motors have to be controlled in a loop to ensure the effective speed.

Servos have to get a better accuracy and an output signal for their effective position while they are moving.

References

[1] YARB 1.0 (Yet Another Robotic Blimp), http://www.surveyor.com/YARB.html

[2] Blimpduino project, http://diydrones.com/profiles/blogs/blimpduino-home-page

[3] Jörg Muller Wolfram Burgard,Efficient Probabilistic Localization for Autonomous Indoor Airships using Sonar, Air Flow, and IMU Sensors, University of Freiburg, Germany, 22pg

[4] IEEE Standard for Inertial Sensor Terminology IEEE Std 528-2001, 20 pg

[5] NOAA, Wandering of the Geomagnetic poles, http://www.ngdc.noaa.gov/geomag/GeomagneticPoles.shtml

[6] Freescale Semiconductor, 2012, AN4248 Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors, 21pg

[7] Department of Defense World Geodetic System 1984: Its definition and relationships with local geodetic systems, 2000, National Imagery and Mapping Agency Technical Report TR8350.2, 175pg.

[8] Groves, Paul D., 2008, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, 521 pg.

[9] Carlo Ferraresi, Terenziano Raparelli, 2007, Meccanica Applicata, Torino C.L.U.T. Editrice, 314p.

[10] NASA,1976,US Standard Atmosphere 1976,241pg

[11] MaxBotix inc., 2012, XL-MaxSonar®- EZ™ Series High Performance Sonar Range Finder, 15pg

[12] NASA SP-36,7 Introduction to the Aerodynamics of Flight

[13] ST Microelectronics, 2011, LSM303DLHC Ultra compact high performance e-compass 3D accelerometer and 3D magnetometer module,42pg

[14] Mohinder S. Grewal and Angus P. Andrews, IEEE Control Systems Magazine , June 2010, Applications of Kalman Filtering in Aerospace 1960 to the Present, 10 pg

[15] Yechout, Thomas R., 2003, Introduction to Aircraft Flight Mechanics, 650 pg

[16] David H. Titterton and John L. Weston, 2004, Strapdown Inertial Navigation Technology 2nd edition, ISBN 0 86341 358 7, The Institution of Electrical Engineers, 581 pg

[17] Mohinder S. Grewal and Angus P. Andrews, 2001 , Kalman Filtering: Theory and Practice Using MATLAB, 3rd Edition, 592pg

[18] Mamdani, E.H. and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13, 1975

[19] Paul A. Debitetto, Fuzzy Logic for Depth Control of Unmanned Undersea Vehicles, IEEE journal of oceanic engineering, vol. 20, no. 3, July 1995

[20] Power HD, LiPo Battery,20C 1S 250mAH datasheet, 3 pg

[21] Power HD, LiPo Battery,25C 2S 550mAH datasheet, 3 pg

[22] Motor-7mm 3.3 Ohm w/nanoconnector, http://www.microflight.com/Online-Catalog/Motors/7mm-3-3-Ohm-Coreless-Motor

[23] Hitec HS-55 servo motor, http://hitecrcd.com/products/servos/micro-and-mini-servos/analog-micro-and-mini-servos/hs-55-economy-feather-servo/product

[24] ST Microelectronics, 2013, L3GD20 MEMS motion sensor: three-axis digital output gyroscope rev 2

[25] Freescale Semiconductor, 2013, MPL3115A2 Xtrinsic MPL3115A2 I2C Precision Altimeter,42pg

[26] ST Microelectronics, 2013, STM32F103x6 Low-density performance line, ARM-based 32-bit MCU with 16 or32 KB Flash, USB, CAN, 6 timers, 2 ADCs, 6 com. Interfaces,90pg

[27] Pihlström Timo, 2014, Langaton inertiamittausyksikkö, Diplomityö - Master's thesis, Tampereen teknillinen yliopisto, 53p.

[28] Pololu, 2012 ,Pololu Baby Orangutan B-328 User's Guide,12pg

[29] Atmel, 8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash, ATmega48PA/ATmega88PA/ATmega168PA/ATmega328P datasheet, 448 pg

[30] Toshiba, TB6612FNG Driver IC for Dual DC motor, 11 pg

[31] Balloon (Blimp, Silver)- 52"x37" for Tri-Turbofan, http://www.microflight.com/Online-Catalog/R-C-Toys/Balloon-Blimp-shape-52x37-for-Tri-Turbofan_2

[32] Intermec, BU-2073-J (USB Bluetooth Adapter) Compliance Insert datasheet, 3pg

[33] Pololu, CRC8 bit calculation, http://www.pololu.com/docs/0J44/6.7.6

[34] Ian Reid,Hilary Term, 2001, Lecture Notes Estimation II, 44pg

[35] Analog Device, Triaxial Inertial Sensor with Magnetometer ADIS16400/ADIS16405 datasheet, 20 pg

[36] ARM, ARM Cortex M-4 reference page, http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php