Politecnico di Torino

FACOLTÀ DI INGEGNERIA
Corso di Laurea Magistrale in Mechatronic Engineering

Master's degree Final Thesis

Multi-Sensor Data Fusion Techniques for Autonomous Vehicles Navigation

Candidate:

Alessio D'Andrea

Thesis advisor:

Prof. Massimo Violante

Academic Year 2017-18

Nature shows us only the tail of the lion.
But there is no doubt in my mind that the lion belongs with it even if he cannot reveal himself to the eye all at once because of his huge dimension.
We see him only the way a louse sitting upon him would.

Albert Einstein

Acknowledgements

It is not easy to quote and thank, in a few lines, all the people who contributed to the birth and development of this work: those with a constant collaboration, those with moral or material support, those with advice and suggestions or just with words of encouragement.
First of all, I would like to thank Prof. Massimo Violante, as a supervisor of my thesis, for giving me explanations and above all for his advice and encouragement in the many moments of discouragement and anxiety that have seized me. But above all for proposing this project that turned out to be very interesting, both from an educational and practical point of view.
A big thank you also to Jacopo, for giving me suggestions both during the implementation of the project and during the writing of the thesis.
Another big thank you goes to my classmates whom I met during their studies, each of them it was of great importance in giving me some advice for an exam, or even just accompanying me in the study.
Last but not the least, I would like to thank my family, and my brother, for all the support given to me during these years, for giving me the necessary stimuli, first to start this academic "adventure", and then to continue it, stand me close to the moments when I thought not to come more at the head.

Summary

In this thesis will be illustrated an implementation of an inertial navigation system for autonomous vehicles, also called Inertial Navigation Systems (INS).
In autonomous vehicles, the localization is important, with the loss of the GPS signal, very frequent in urban centers, or in tunnels, it is necessary to use the measurements of other sensors to continue the localization.
The Inertial Navigation System compute localizations using two different measurement methods, the fist method uses vehicular sensors, like odometer and steering angle, and calculate the location using the bicycle model.
The second measurement method uses IMU sensors, like accelerometer, gyroscope, and magnetometer, will be illustrated an algorithm that approximates localization, using these sensors.
The two different measurements are analyzed by the Multisensor Data Fusion, which increases accuracy, generating a third estimated measurement.
The measurement of the vehicle sensors is not very reliable, because the wheel slippage can lead to incorrect measurements, so it is also necessary to use also IMU sensors.
The IMU sensors used are of the MEMS type (Micro Electro-Mechanical Systems), they are constantly evolving sensors, and available on the market at low cost, however they also have some disadvantages related to the sensitivity that bring the system to instability after about a minute.
The measurements provided by the IMU sensors are filtered through the use of Kalman Filter, appropriately calibrated, and supplied to the algorithm that rototranslates the vehicle on the x, y and z axes. The vehicle sensors, unlike can provide very high accuracy measurements, so it is preferable to give a greater weight.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, demand has grown, both in the civil and industrial sectors of devices with autonomous guidance, an example would be the robot vacuum cleaner, the robot lawn-mower, or even a car.
The technological evolution and the lower prices of microcontrollers and sensors, have allowed the development of these new autonomous guidance devices.
Already in the last century, much progress has been made as in astronomy field, and in aviation, developing autonomous rover, and plane with autopilot, that embedded inertial navigation system (INS).

Figure 1.1: Autonomous Car

In Figure 1.1 is shown an Autonomous Car, it is possible to notice the presence of GPS,

odometer, and body computers that implement an Inertial Navigationm System. The current state of the INS for terrestrial use at present is not well known, since they are commercial solutions, it is difficult to analyze public data on this product.
"*An inertial navigation system is a navigation aid that uses a computer, motion sensors (accelerometers), rotation sensors (gyroscopes) and occasionally magnetic sensors (magnetometers), to continuously calculate via dead reckoning the position, orientation and velocity (direction and speed of movement) of a moving object without the need for external references.*
It is used on vehicles such as ships, aircraft, submarines, guided missiles and spacecraft. Other terms used to refer to inertial navigation systems or closely related devices include inertial guidance system, inertial instrument.
Older INS systems generally used an inertial platform as their mounting point to the vehicle and the terms are sometimes considered synonymous" [**5**].

1.1 Losing GPS signal

Autonomous vehicles, such a cars or surveillance vehicles, need GPS position for their features,but GPS signal can be obstructed by buildings, we can see an example in Figure 1.2.
In terrestrial vehicles losing GPS signal is very common, we can remember for personal experiences, using GPS navigator signal loss is very common, due to different conditions. For a car typical cases are tunnels and the buildings that interfere communication with the satellites.

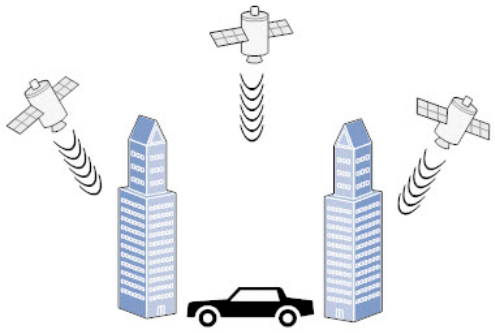

Figure 1.2: Building obstructions

Buildings, specially in city, are the first reason of losing GPS signal.
In another scenario, we can have it with the use of a surveillance device inside a building, the losing GPS signal is even more frequently.
Even more and more growing is the use of surveillance robots, both for civilian and military use, they are even more exposed to problems related to the loss of GPS signal, especially inside buildings, in the Figure 1.3 we can see one of these robots:

Figure 1.3: Surveillance Robot

1.2 Accurancy of GPS signal

Nowadays, the number of satellites are increasing, and also satellite navigation systems: GLONASS and GALILEO (not already available).
But the results are not always are accurate, we can consider a bad weather scenario, due to lightning, the GPS receiver is still able to receive the satellite signal, but the location is wrong.

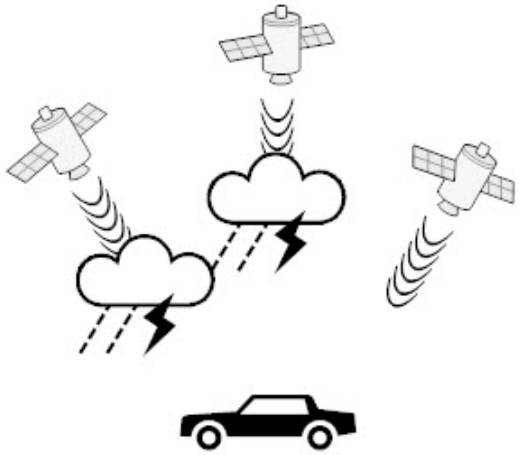

Figure 1.4: Rain scenario

Another example, that we can consider is represented by reflections, we always have GPS communication, due to buildings, or other obstacles that produce reflections.

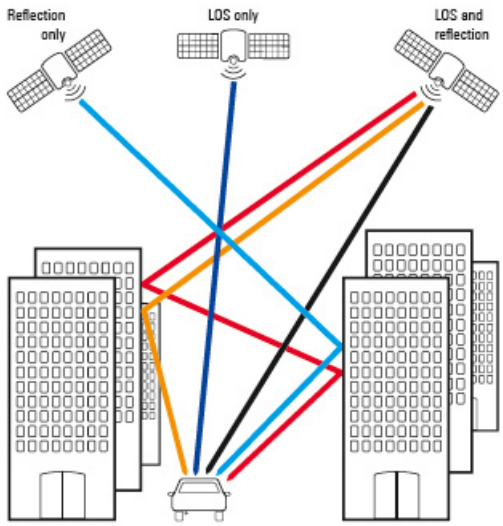

Figure 1.5: GPS reflection

1.3 Thesis goal

There are many cases in which we would have data provided by the GPS wrong or absent. The evolution of INS systems has not had a rapid growth as for the INSs used in the aerospace field, both for problems related to magnetic disturbances, and for the oscillations due to the roughness of the ground.
In this thesis we will present a study case in which using other sensors, we could estimate the position, speed, and orientation normally provided by GPS sensors.
In the presented study case, Multisensorial Datafusion will be illustrated, implemented using various Kalman Filters, and the Kinematics adopted to generate the models.
Will be explained the various problems related to sensors, data processing, and the limits of presented solution.

Chapter 2

Background

The main objective of this thesis is to estimate the various data that the GPS sensor should provide, in absence, deficiency or corrupted signal conditions. To implement our model, we will develop two models, the first generate a path using IMU data, the second generate another path using vehicular sensor. Vehicle signals such as steering angle and wheel encoder, provide values with very high accuracy, that we will consider ideal. Instead the measurements from the IMU are very noise affected. To solve this problem, we will use the Kalman Filters, appropriately calibrated for each sensor. Despite the model of the vehicular sensors, in our case the "Bicycle model" generates very ideal data, does not take into account the slipping of wheels and the axial geometry of the vehicle. In our case model implemented by IMU will solve slipping problem, and magnetometer will give us initial vehicle orientation. Vehicular sensors are considered ideal, and in our case study do not require filtering. Unlike, IMU sensors taken into account, are not high-performance, but very cheap, and in some cases make the system really unstable.Great care is taken to filter the signals, and a calibration is also performed for the Magnetometer and the Accelerometer sensors. Gyroscope sensor unlike, does not need calibration, as it produces an absolute quantity. Once the two models have been generated, the data is merged into a data fusion appropriately calibrated in order to give greater weight to the data considered better. In the following paragraphs the theory of Kalman Filters, Data-Fusion and Kinematics will be explained.

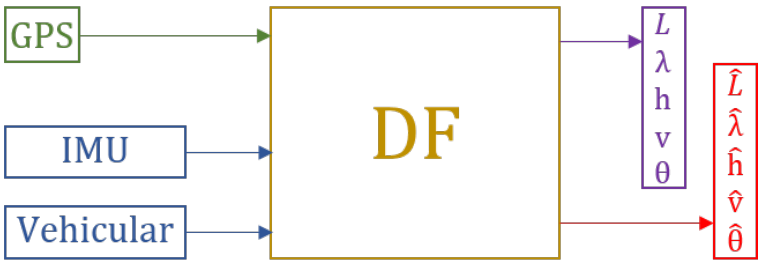

Figure 2.1: DF Model

2.1 Kalman Filter

"*All sensors are subject to noise, very often, as in our study there is an addition of white noise, where the spectral power density is the same for each frequency.*"[**12**]
Sensor resolution must be sufficient for necessary measurement. Filter the noise is necessary, in order to have stable calculation. For filtering the noise, there are different techniques, our choice was on Kalman Filter, for excellent results on linear systems with gaussian error with zero mean.
Before talking about the Kalman Filter, we will give a brief introduction to the standard deviation. The standard deviation, or even the average square deviation, is an index of dispersion of a distribution of values, in our case it is related to the noise that our sensors add to the measurements.
The Standard Deviation can be expressed using the following formula[**9**]:

$$\sigma = s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{2.1}$$

And can be represented using a Gaussian, as a distribution around a Mean Value:

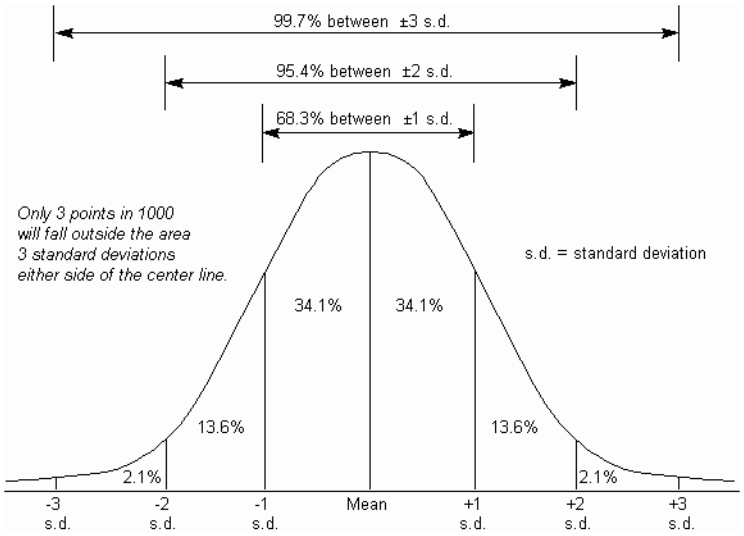

Figure 2.2: Standard Deviation. Figure from [**9**]

A distribution equal to $\pm 1\sigma$ has a probability equal to 68.2%, while a distribution with standard deviation equal to $\pm 3\sigma$, has a probability of 99.7%.

Kalman Filter works in a loop of two phase process: prediction and estimation.
In prediction phase, the filter generate estimate states, for initial state, initial condition are necessary. In estimation phase, the estimate state are updated using a weighted average of the new measurements, increases the weight of the estimated variable, and consequently

the estimated state becomes more accurate.
"*The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown.*"[**6**]
"*The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups:* ***time update*** *equations and* ***measurement update*** *equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the* ***a priori*** *estimates for the next time step.*
The measurement update equations are responsible for the feedback—i.e. for incorporating a new measurement into the ***a priori*** *estimate to obtain an improved a posteriori estimate. The time update equations can also be thought of* ***as predictor*** *equations, while the measurement update equations can be thought of* ***as corrector*** *equations.*
Indeed the final estimation algorithm resembles that of a ***predictor-corrector*** *algorithm for solving numerical problems.*"[**6**]

Discrete Kalman Filter time *update equations*, are the following:

$$\hat{x}_k^- = A\hat{x}_{k-1} + B\hat{u}_{k-1} \tag{2.2}$$

$$P_k^- = AP_{k-1} + A^T + Q \tag{2.3}$$

Discrete Kalman filter *measurement update equations*, are the following:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{2.4}$$

$$\hat{x}_k = \hat{x}_k^- K_k(z_k - H\hat{x}_k^-) \tag{2.5}$$

$$P_k^- = (I - K_k H)P_k^- \tag{2.6}$$

The first step, is to compute Kalman gain K_k, and next step will be to measure the process to obtain z_k, and than to generate an *a posteriori* state estimate by incorporing the measurements in equation of $\hat{x}_k$, and final step is to have *a posteriori* covariance estimate, P_k.
The steps will be repeated with the previous a posteriori estimates used to predict the new a priori estimates.

Discrete Kalman Filter algorithm can be represented by the follow diagram: Deploying

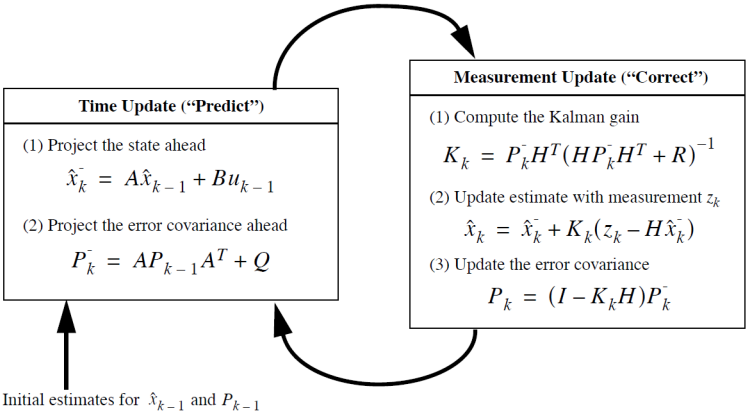

Figure 2.3: Kalman Filter operations

the filter, the measurement noise covariance is usually measured a prior to operation of the filter. Measurement error covariance is generally practical, because we need to be able to measure the process anyway, so we should generally be able to take some off-line sample measurements in order to determine the variance of the measurement noise.
Process noise covariance Q is more difficult to determine, not ever we are able to observe the process that we are estimating. Often, it is possible that a poor process model can produce an acceptable result with much uncertainty in the process, due to the choice of Q.
In other cases, we can tune the parameters R and Q, with good results, introducing less noise possible.[**6**]
We note that under condition where Q and R are constant, both estimation error covariance P_k and Kalman gain K_k will stabilize quickly and then remain constant. If this is the case, these parameters can be pre-computed by either running the filter off-line, or for example by determining the steady-state value of P_k[**7**].

2.2 Multi-sensor Data Fusion

Purpose of Multisensor Data Fusion is to fuse data coming from different sensors, in order to have direct or indirect measurements, increasing the accuracy of the system.
Multisensorial Data Fusion, can be represented by the following diagram[**9**]:

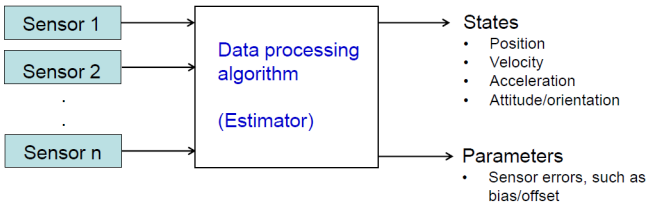

Figure 2.4: Multi-sensor Data Fusion diagram. Figure from [**9**]

Combine observations from a number of different sensors to have a robust and complete description of an environment or process of interest.
Using a Recursive Bayesian Filtering techniques, is possible to integrate measurements from inertial sensors with measurements from different kind of sensors [**8**].
The main objective of Data Fusion utilization is reduction uncertainty, allowing to makes the system more robust. Data Fusion is very similar to the Kalman Filter, to introduce

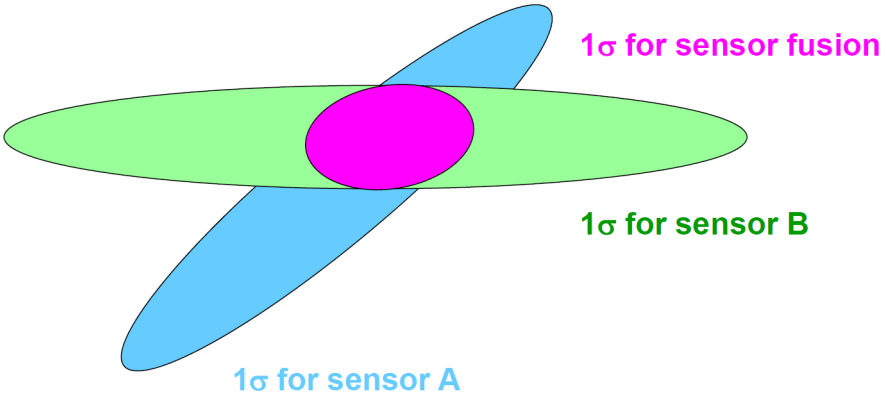

Figure 2.5: Multi-sensor Data Fusion uncertainty. Figure from [**9**]

Data Fusion, we must consider a linear system as:

$$x_k = Ax_{k-1} + Bu_k \tag{2.7}$$

Adding to the equations, the process noise w, we will have:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \tag{2.8}$$

x_k is the state of system, x_{k-1} the state of system at the previous time step, A an B the matrix relates the state at the previous time step to the state at the current step, and

the output matrix that relates the optional control input u to the state x. Now we must introduce z_k, as follow:

$$z_k = Hx_k + v_k \tag{2.9}$$

where v is th measurement noise introduced by the sensor.[**8**]
Two sensor Data Fusion can be described by two measurement equations:

$$z_1 = x + v_1 \tag{2.10}$$

$$z_2 = x + v_2 \tag{2.11}$$

considering that v is the noise with standard deviation σ, we can combine the two measurements as follow[**9**]:

$$\hat{x} = k_1 z_1 + (1 - k_1) z_2 \tag{2.12}$$

$$\hat{x} = (\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}) z_1 + (\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}) z_2 \tag{2.13}$$

2.3 Kinematics

In the case study presented in this thesis, the use of roto-translations was necessary. In our case, we will use roto-traslation on rigid body, in 3D space, with right orthonormal reference frame.

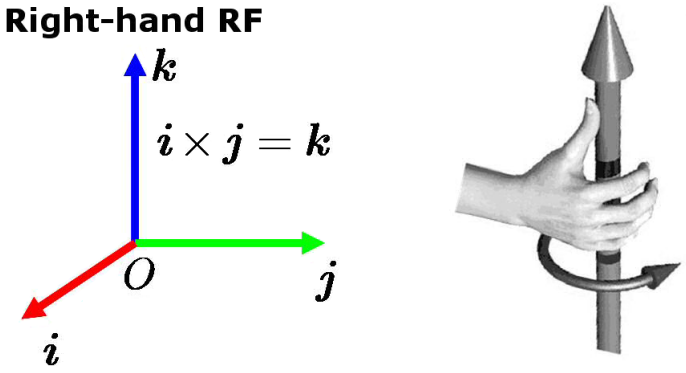

Figure 2.6: Right-hand Reference Frame. Figure from [**28**]

A rigid translation is the operator Trasl(v, t) that displaces v parallel to itself of an amount and along a direction specified by the vector $t = [t_1 t_2 t_3]^T$ [**28**]:

$$Trasl(v,t) \equiv \boldsymbol{v}^t \doteq \begin{bmatrix} v_1 + t_1 \\ v_2 + t_2 \\ v_3 + t_3 \end{bmatrix} = v + t \tag{2.14}$$

[**28**]

A rotation of a rigid body, whose columns are the representations of the basis unit vectors of Ref. Frame(n) in Ref. Frame(m), as from the Euler Theorem, can be expressed as[**28**]:

$$\boldsymbol{R}_n^m = [i_n^m j_n^m k_n^m] = \begin{bmatrix} i_{n1}^m & j_{n1}^m & k_{n1}^m \\ i_{n2}^m & j_{n2}^m & k_{n2}^m \\ i_{n3}^m & j_{n3}^m & k_{n3}^m \end{bmatrix} \tag{2.15}$$

[**28**] Elementary rotations, for example on z axis of an angle γ, can be expressed as follow:

$$Rot(z,\gamma) \doteq Rot(\boldsymbol{k},\gamma) \doteq R_{k,\gamma} \doteq \begin{bmatrix} cos_\gamma & -sin_\gamma & 0 \\ sin_\gamma & cos_\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

[**28**]

Roto-translation can be of four different type:

- The first case (a) is when we have an initial rotation of the reference frame on common origin, and a translation respect to the fixed frame.

- The second case (b), we have for first rotation, and after a translation on mobile reference frame.
- The third case (c), we perform for first translation of reference frame, and after a rotation on the mobile reference frame.
- The fourth case (d), we have for first translation, and after a rotation on fixed frame[**28**].

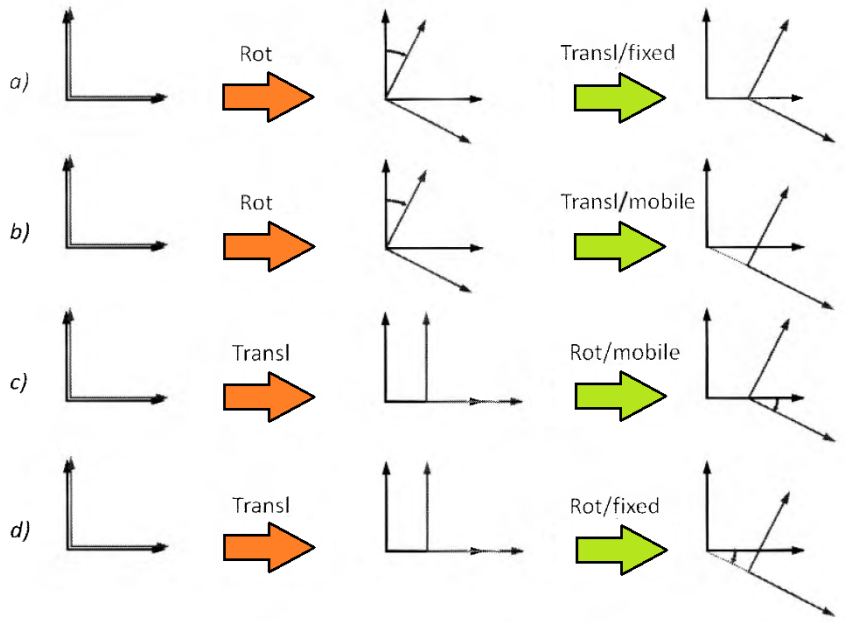

Figure 2.7: Different ways to perform a planar rototranslation. Figure from [**28**]

The final results will be different, our study case is implemented like "d" case, for this reason, we will use the following formula:

$$v_P^0 = \boldsymbol{R}_m^0(v_P^m + t') = \boldsymbol{R}_m^0 v_P^m + \boldsymbol{R}_m^0 t' \tag{2.17}$$

[**28**] and $v_{AB}^0 = \boldsymbol{R}_m^0 v_{AB}^m$ t' is the translation vector from Ref. Frame(0) to Ref. Frame(m), represented in Ref. Frame(m). A roto-traslation can be represented in Homogeneous Coordinate:

$$T_m^0 \doteq \begin{bmatrix} \boldsymbol{R}_m^0 & \boldsymbol{t}_m^0 \\ O^T & 1 \end{bmatrix} \tag{2.18}$$

where $\boldsymbol{R}_m^0$ is the rotation, and $\boldsymbol{t}_m^0$ is the translation[**28**].

Chapter 3

Inertial Navigation System

Inertial navigation is a self-contained navigation technique in which measurements provided by accelerometers and gyroscopes are used to track the position and orientation of an object relative to a known starting point, orientation and velocity. Inertial measurement units (IMUs) typically contain three orthogonal rate-gyroscopes and three orthogonal accelerometers, measuring angular velocity and linear acceleration respectively.Recent advances in the construction of MEMS devices have made it possible to manufacture small and light inertial navigation systems.[**10**]

3.1 Strapdown Inertial Navigation

The strapdown navigation algorithm is shown in the following figure: Analyzing the algo-

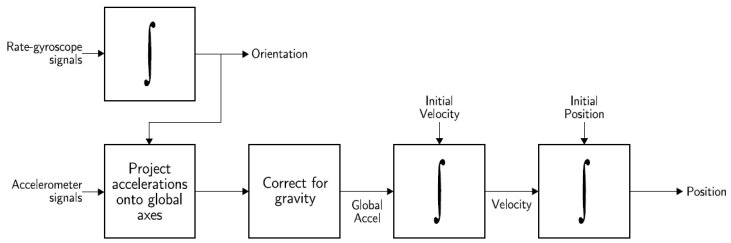

Figure 3.1: Starpdown Inertial Navigation. Figure from [**10**].

rithm in detail, most important is how errors which arise in the individual accelerometers and gyroscopes propagate. In orientation, angular velocity signals obtained from the gyroscopes are 'integrated' by the standard INS attitude algorithm, therefore errors in the gyroscope signals propagate through to the calculated orientation. For most MEMS devices white noise and uncorrected bias errors are the main causes of an error in the orientation. White noise causes an angle random walk whose standard deviation grows proportionally to the square root of time. An uncorrected bias causes an error in orientation which grows linearly with time. Quantisation errors also arise in the calculated attitude due to the

quantisation of the angular velocity samples and due to the integration scheme used to update C. [**10**]

$$C(t+\delta t) = C(t) \cdot \exp(\int_{t}^{t+\delta t} \Omega(t)\, dt) \tag{3.1}$$

We can say that in tracking position, error propagate through double integration, and in angular velocity, errors are combined together during roto-traslation algotithm. This causes several problems. Firstly, the accelerations of the device are integrated in the wrong direction. Secondly, acceleration due to gravity can no longer be correctly removed[**10**].

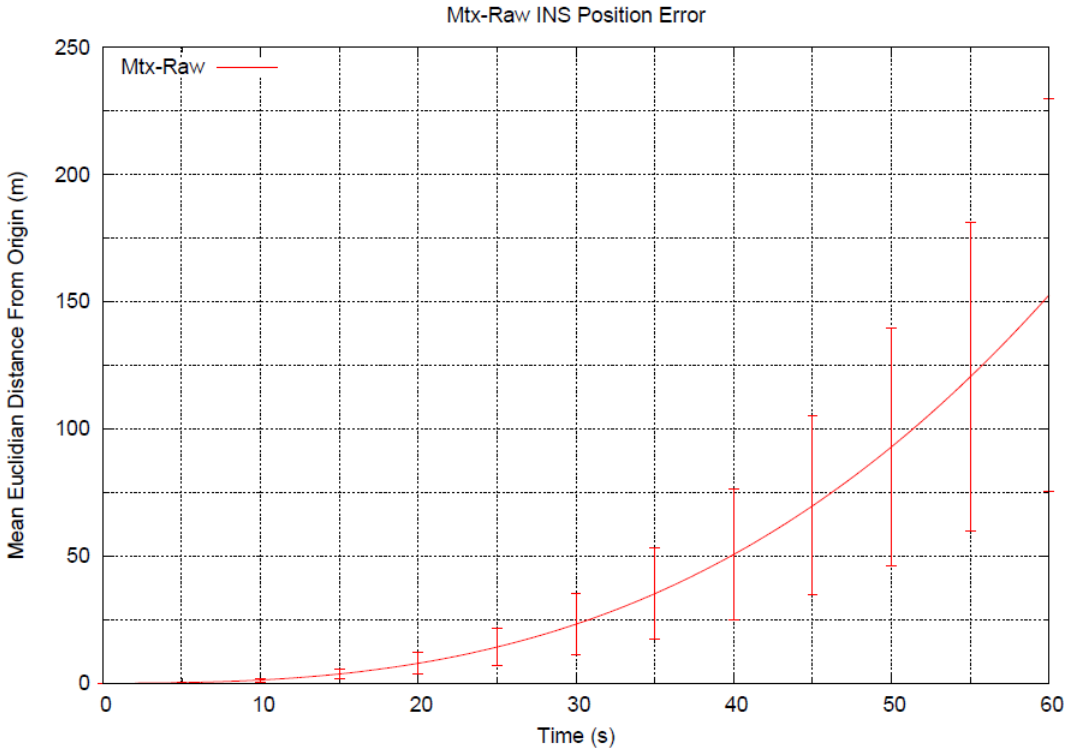

Figure 3.2: Increases of average drift. Figure from [**10**].

"*In a plot above, is depicted how the average drift in position incurred the system increases over time.*"[**10**]

3.2 Reducing Drift

As errors propagate due to algorithms, one solution is to use Sensor Fusion, drifting can be reduced or corrected . A sensor fusion algorithm maintains this state using IMU accelerometer and gyroscope signals together with signals from additional sensors or sensor systems. There are many techniques for performing sensor fusion, the most popular of which are Kalman and particle filters. [**10**] We will speak about the implementation of this algorithms on the following chapters. One common approach is to periodically correct drift using position data from an absolute positioning system, such as GPS, or using Magnetometers. GPS data is commonly used, however GPS positions can usually only be obtained outdoors and will typically provide absolute positions which are only accurate to around 15 m. This makes fusion with GPS data unsuitable for indoor use and for applications such as human motion capture where high accuracy is required. Another type of sensor commonly used to reduce drift is the vector magnetometer, which measures magnetic field strength in a given direction. Inertial measurement units often contain three orthogonal magnetometers in addition to the orthogonal gyroscopes and accelerometers. The magnetometers measure the strength and direction of the local magnetic field, allowing the north direction to be found. Magnetometers are not accurate enough to replace gyroscopes in INSs.[**10**]

Chapter 4

Bicycle Model

In robotics there are many different configuration of wheeled robot, quadcopter, train, omni-wheeled robot, with different degrees of freedom. Our study is focused on vehicle with four wheels, of which the front steering wheels, so we would have two degrees of freedom. To obtain a good modeling of the vehicle, it is necessary to analyze the steering

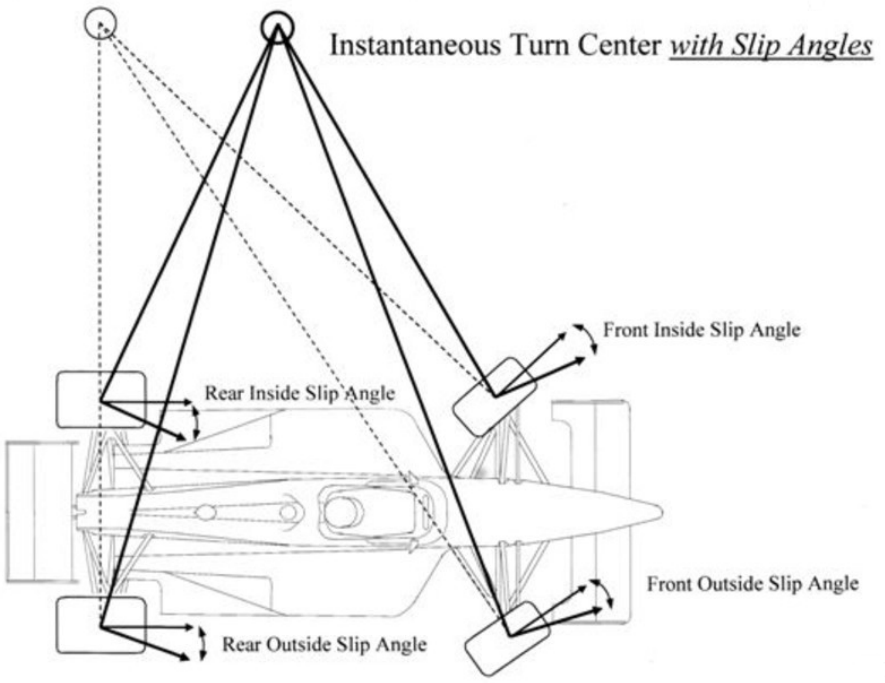

Figure 4.1: Ackerman model[**27**].

system adopted, in the event that the Ackerman model is chosen, we would have the steering wheels with different angles, which generate friction in steering.

In following figure is possible to see the particular geometry of Ackerman model in case of steering angle is huge. In order to use this model, the speeds of both wheels are

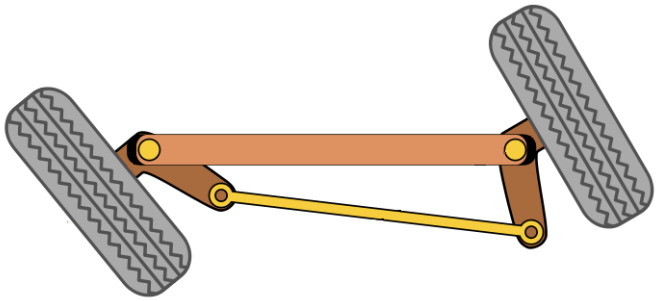

Figure 4.2: Ackerman Geometry

necessary, and it is more difficult to configure. Instead a widely used model is that of the bicycle, easily adaptable to different types of configuration. It has always 2 degree of freedom, but only one odometer on rear wheel, and one steering angle at the front side. It

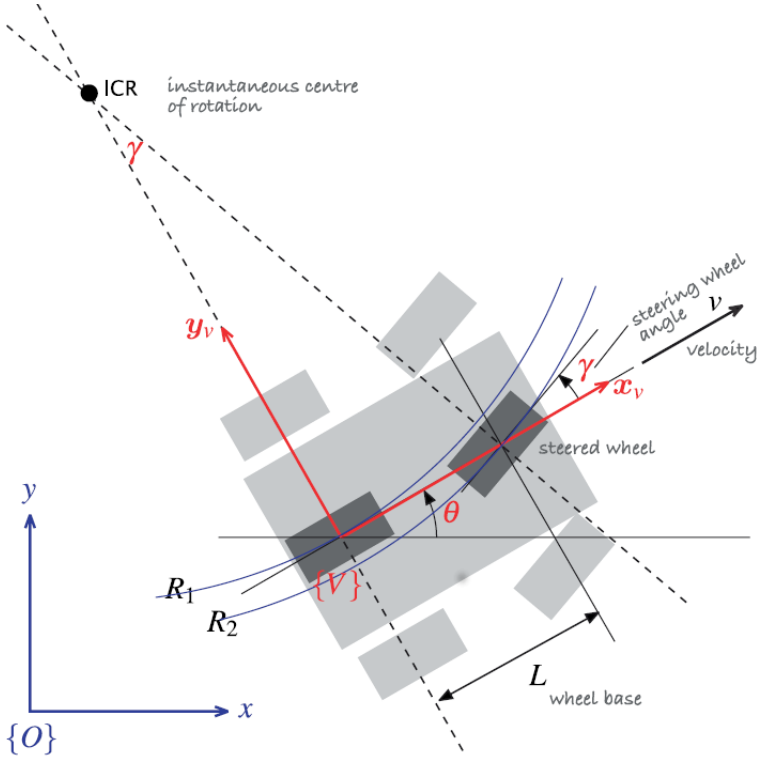

Figure 4.3: Bicycle model. Figure From [**26**].

is represented by three generalized coordinates q=(x,y,θ), instead ICR is the instantaneous centre of rotation, thanks to it is possible to calculate the direction of vehicle.

4.1 Simulink Bicycle Model

The velocity of vehicle in the Instantaneous Reference Frame is computed using the following formulas:

$$\dot{x} = v\cos\theta \tag{4.1}$$

$$\dot{y} = v\sin\theta \tag{4.2}$$

$$\dot{\theta} = \frac{v}{L}v\tan\gamma \tag{4.3}$$

Where L is the distance between the front wheels and the rear wheels, and γ the steering angle. Discrete time integrator is used to integrate $\hat{x}$ and $\hat{y}$. In following figure is depicted

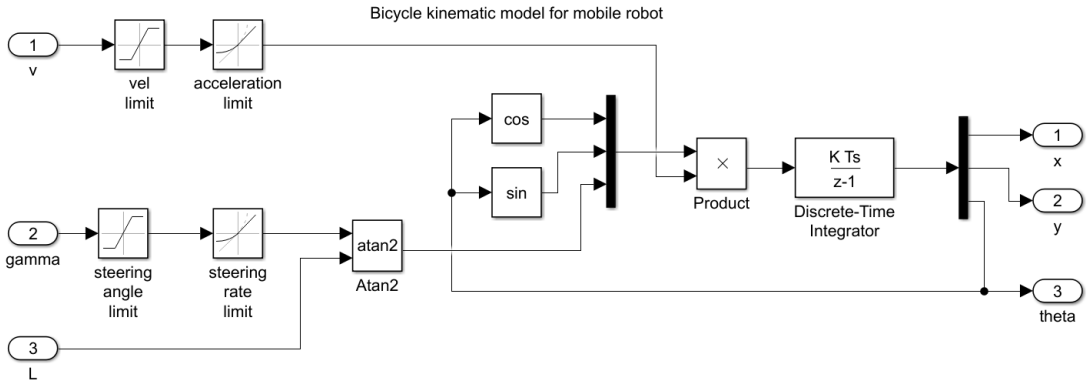

Figure 4.4: Bicycle Simulink Model[**26**].

the behaviour of vehicle, when the steering angle is positive, θ increases linearly. In the absence of steering, the y position remains constant.

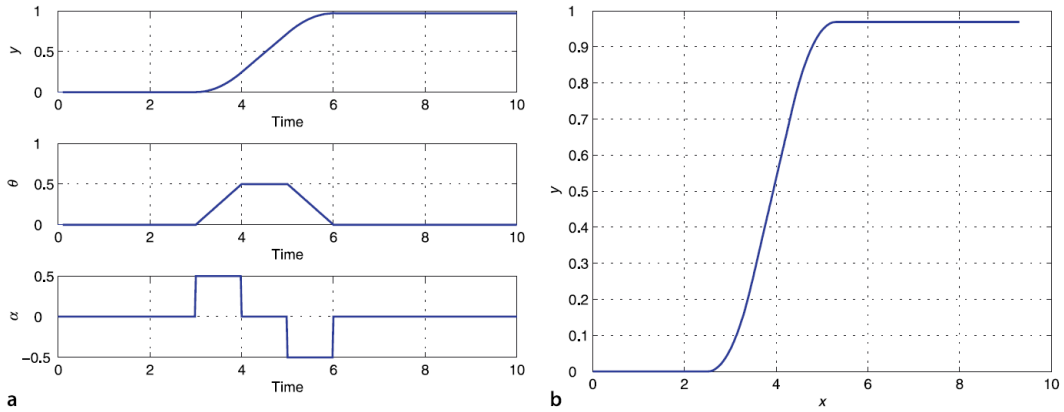

Figure 4.5: Vehicle response as a function of time[**26**].

Chapter 5

Global Positioning System

5.1 Introduction on GPS

Nowadays Global Positioning System (GPS) is widespread also on commonly used devices like satellite navigator, satellite anti-theft devices, smartphones, notebooks, and others. GPS is a system able to determine geographic coordinate, in particular latitude and longitude position, velocity, altitude, time and orientation respect to geographical north. The

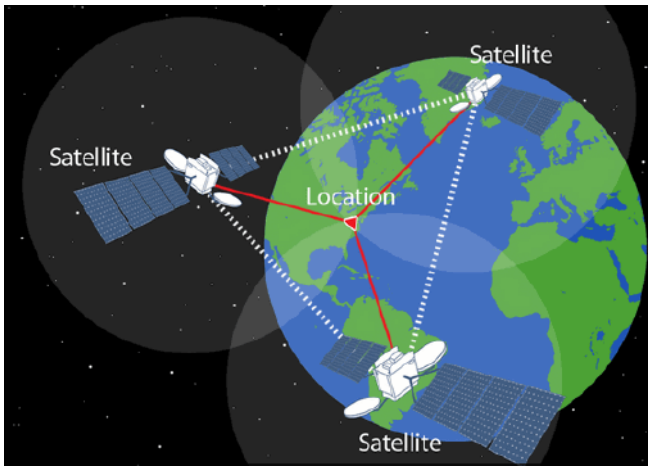

Figure 5.1: Satellite's triangulation[**23**].

principle of operation of GPS uses three components, witch are ground station, satellites and GPS devices. "*The GPS concept of operation is based upon satellite ranging. Users determine their position by measuring their distance from the group of satellites in space. The satellites send act as precise reference points*"[**25**]. Satellites send to the hearth a "time signal", GPS receiver is able to receive and interpret according to various protocols called "Qk", occasionally the satellites also send the ephemeris which are the positions of the constellation. GPS receiver based on the propagation time, they are able to calculate the distance between the various satellites. It is not possible to use GPS with less than 4

satellites, instead, if more satellites are available, more precise is the localization. "*GPS satellites fly in Medium Earth Orbit (MEO) at an altitude of approximately 20.200 km (12.550 miles). Each satellite circles the Earth twice a day*"[**24**]. GPS sensor calculate

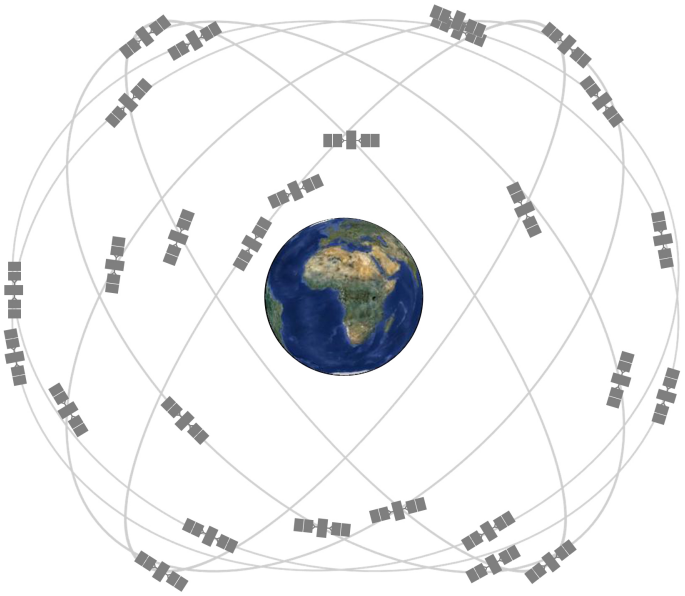

Figure 5.2: Satellites in orbits[**24**].

distances from satellites using a simple formula:

$$distance(i)_{sat} = speed_{sat} * timedelay(i) \tag{5.1}$$

Satellites travel at light speed, 299.792 km/s, they have on board an atomic clock, considering that satellites are far many kilometers, the a delay of communication it's quantified in time, it follows the calculation of distance. To compute the triangulation, at least 4 satellites are required. The connection to a few satellites, is also the cause of a low sensitivity, but the sensitivity is also linked to other factors, the noise of the signal received, the position of the satellites, the weather conditions, and the various obstacles that affect the connection.
Where the speed is the satellite speed, that travels at the light speed, around . The time, is calculated considering that satellite send time based on atomic clock, considering that satellites are far many kilometers, the time is a delay of communication. Having synchronization with at least 4 satellites, GPS device compute the triangulation. Each satellite works for civil and for militar use, on two different frequencies, even if it is constantly evolving. There are also other positioning systems, GLONASS that born in the cold war period, Galileo which will be available in 2019. There are also hybrids, A-GPS (Assisted GPS), it add the benefits of mobile communication, to make localization more precise.

5.2 Simulink Model of loss of GPS signal

The main purpose of this case study is to estimate GPS signal when, GPS device lose communication with satellites. In Simulink model, is implemented using if-else statement, normally the model transmits GPS signals that come from GPS sensor. When are available less than 4 satellites, are transmitted estimated coordinate. The model as in input GPS sensor data, IMU and vehicular data obtained form sensors. Latitude and Longitude are expressed in degree, minute of arc, and separated by "." decimal of minute of arc.

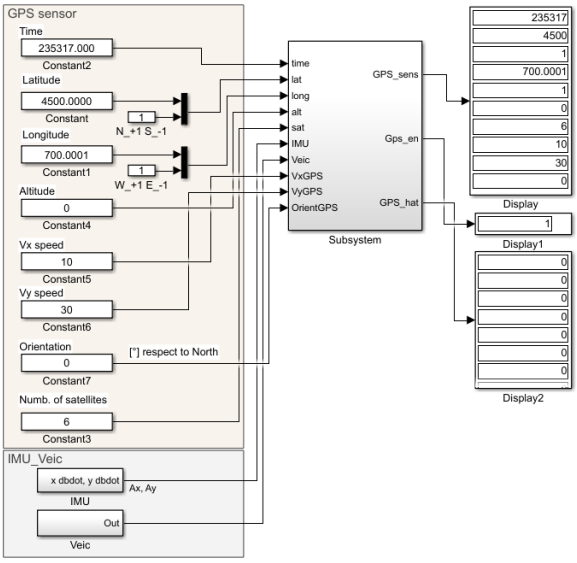

Figure 5.3: Simulink GPS model

The if-else statement is implemented in Simulink:

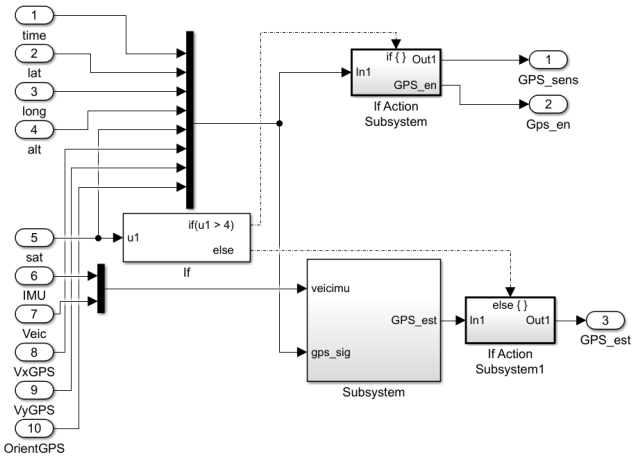

Figure 5.4: If-Else Simulink statement

The algorithm provides for use in macro-areas, that depend by latitude. Our case study is focused on the area along the 45°N of longitude, which is also valid for the 45°S of longitude. Along the various latitude the calculations are always performed in the same way, because the polar circumference is always the same. Before analyzing the algorithm, we must take into account that for calculation are considered the following circumferences:

- Polar circumference: 39940650 m
- Equator circumference: 40075017 m
- 45°N circumference: 28284271 m

In the algorithm is performed an addition of amount of displacement in meters on the angles of latitude and longitude. After that, there is a reconversion of latitude and longitude, which are returned as estimated.

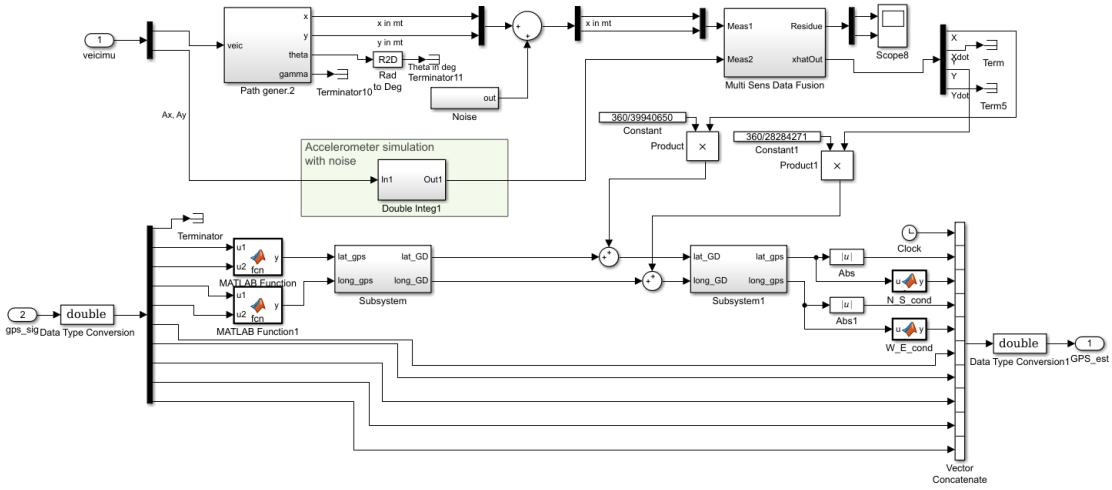

Figure 5.5: Estimation GPS Coordinate Algorithm

Chapter 6

Implementation

This chapter discusses step by step implementation, starting from basic model, up to complete model, highlighting the various algorithms adopted. There will be references to the development of Kalman filters, and to datafusion.

6.1 Model with two axis accelerometer

Assuming that we have an accelerometer with only two axis, neglecting gravity acceleration, we'll have Ax and Ay. Integrating two times the two variables we achieve px and py position. The model use accelerations in m/s^2, and positions are used in m. In the

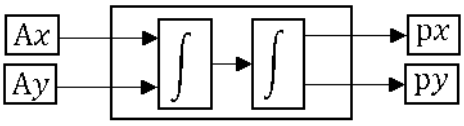

Figure 6.1: Schematic representation of double integration.

scheme of figure 6.1 is considered only translation on the plane X-Y, the model doesn't allow rotation. Real sensors, as well as accelerations sensors, are affected by noise. A way to filter noise, is to consider the sensors with Gaussian error, from Literature: The static and kinematic tests have shown a better performance using the 4th order AR model. [**1**] In the following chapter, the use of Kalman Filter is best explained.

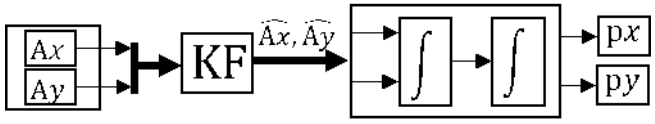

Figure 6.2: Addition of Kalman Filter

In Simulink the model is implemented as follow:

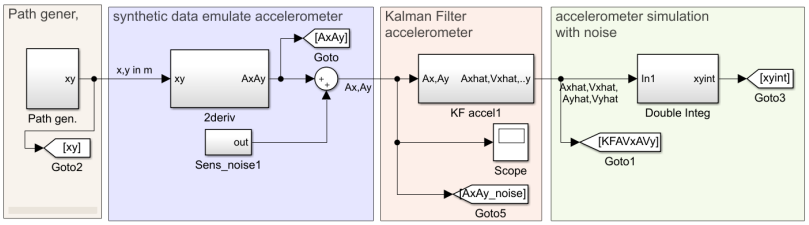

Figure 6.3: Simulink Model of pure translation on 2D plane

In reality it is not possible to use an accelerometer on the X-Y plane only, there is the inclination factor, absolutely not to be neglected. In order to simulate an accelerometer with two axis, we defined a path in x-y plane, that subsequently is derived 2 times, and is add a noise, to have a pseudo-accelerometer. After the signals are processed by Kalman Filter, and finally doubly integrated to have estimated position. In graph, we can see that path is followed not very good, the noise adds up exponentially, due to the use of the two integrators. In the best case, after 50 seconds, the measurement on x axes is 3 times far from real position.

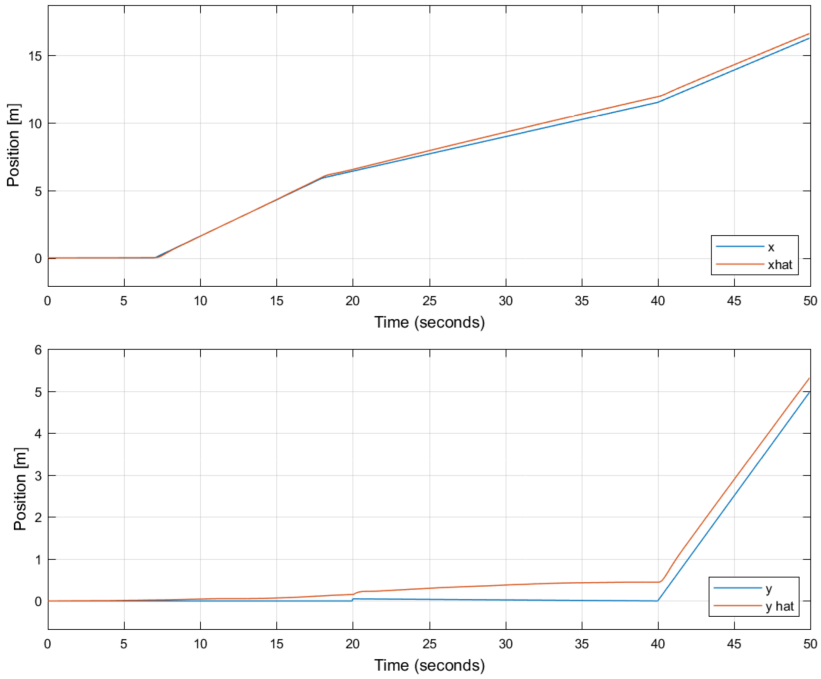

Figure 6.4: Real Generalized Coordinates vs Estimated Generalized coordinate

In following figure is depicted how filtering process, of x and y acceleration, data are not easily understandable, but let us analyze the operation of the Kalman Filter.

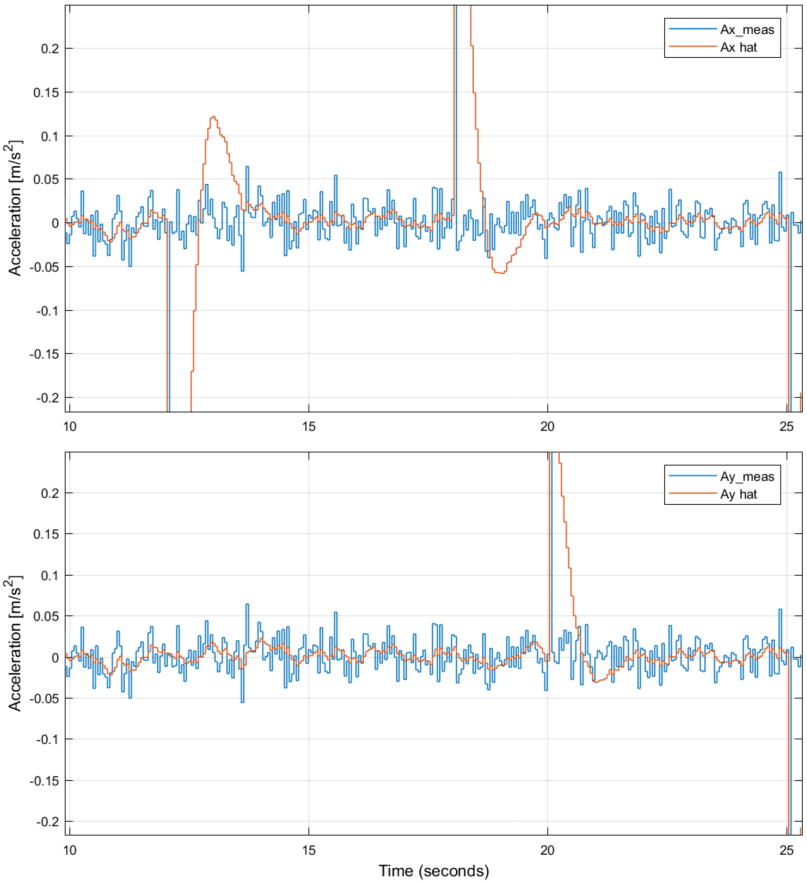

Figure 6.5: Filtering process of pure translation model

6.2 Model with two axis accelerometer and one axes rotation

In this model we consider a translation with on 2D plane, but also the rotation contribution. Rotation contribution is measured using only z axes, since roll and pitch angles on a land vehicle are small. Roto-translation is implemented using Homogeneus Transformation, as described in chapter 2, now we have 3 generalized coordinate: x, y and ψ. In Simulink the model implemented basing on the previous model, and adding simulated

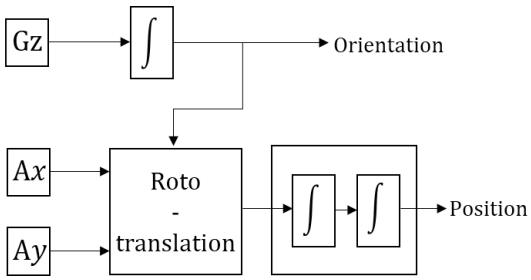

Figure 6.6: Schematic representation of 2D rototranslation.

gyroscope, with an appropriate Kalman Filter. Roto-translation is performed using a multiplication between Yaw rotation Matrix and xy acceleration vector. A comparison is then

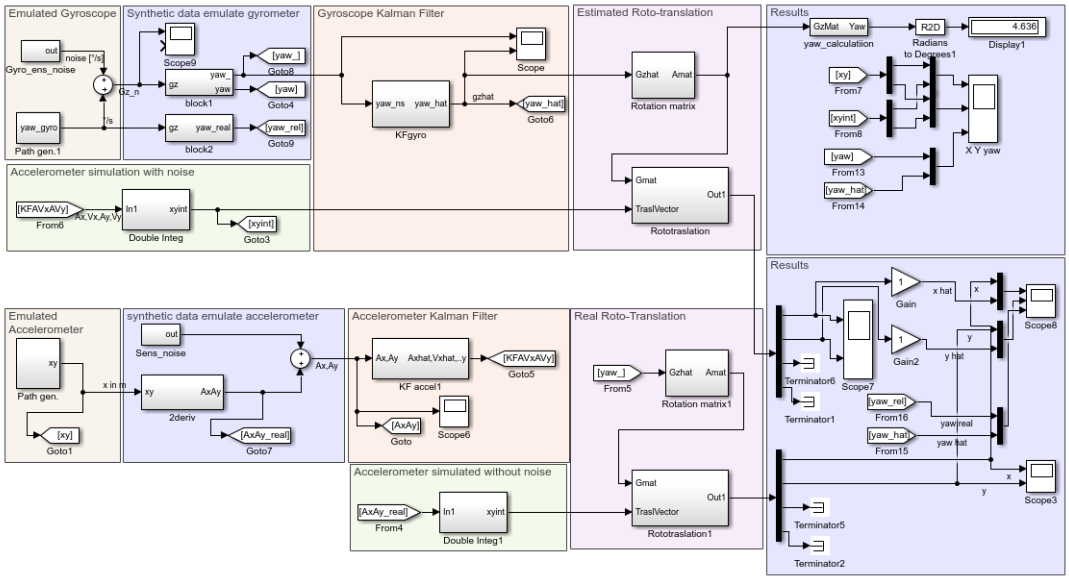

Figure 6.7: Simulink Model of 2D plane rototranslation.

made between the theoretical data, in the absence of noise, and the estimated data.

In following figure is also added the vehicle orientation indicated with ψ angle. Up to 20 seconds the results are acceptable, after many errors are added, and the position on xy it is no longer usable.

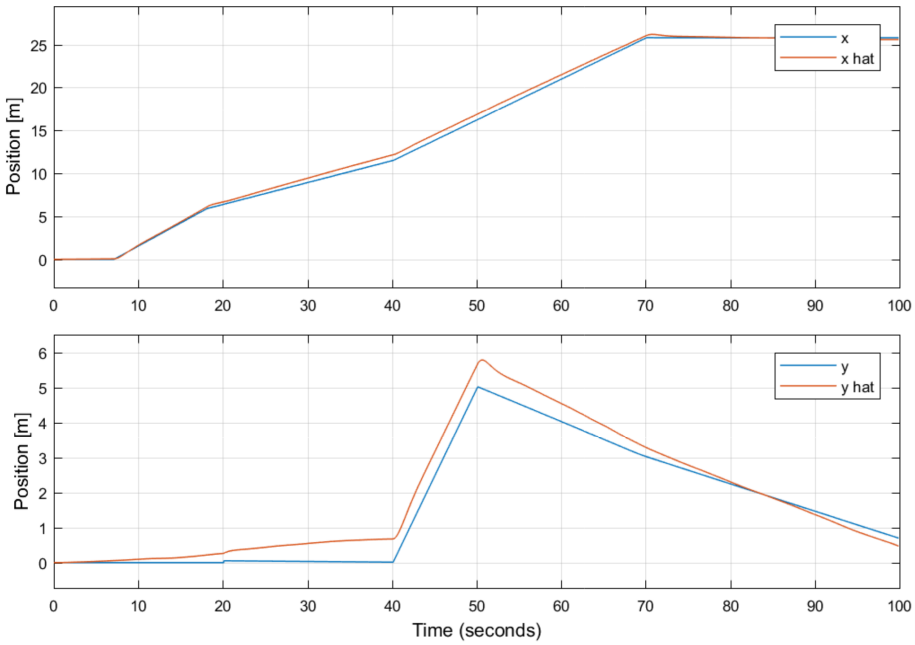

Figure 6.8: Theorical position vs estimated position

The orientation instead is acceptable, in this case even after 100 seconds.

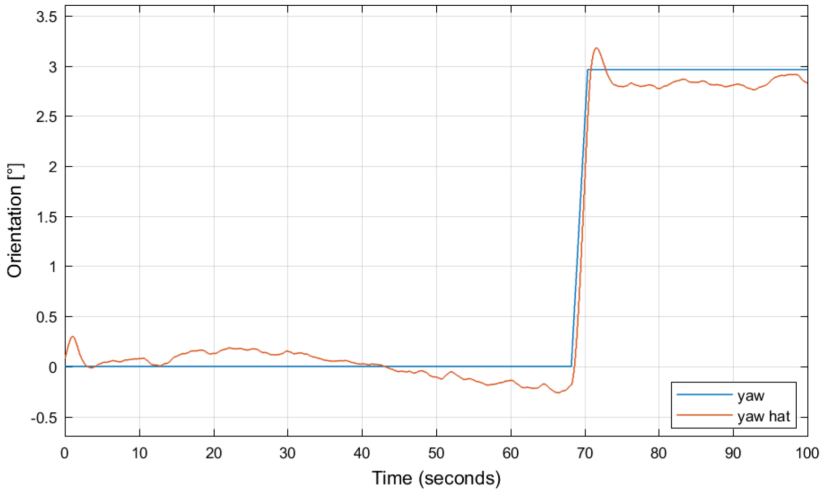

Figure 6.9: Theorical orientation vs estimated orientation

6.3 Model with two axis accelerometer, one axes rotation, and magnetometer

The previous model, don't take into account Earth orientation, it could be possible a manual insertion of the initial orientation may be possible when the vehicle is started. A better solution is to use a compass, in our model we will add a magnetometer. Even if MEMS Magnetometers have some sensitivity problems, reported in chapter 10, we will use a simulated magnetometer, in order to have our model more performing. As shown in previous model, Filter Performance about position and orientation, here we have +45° respect to North, that appear after 5 seconds (initialization phase). In this case, the results are similar to previous model.

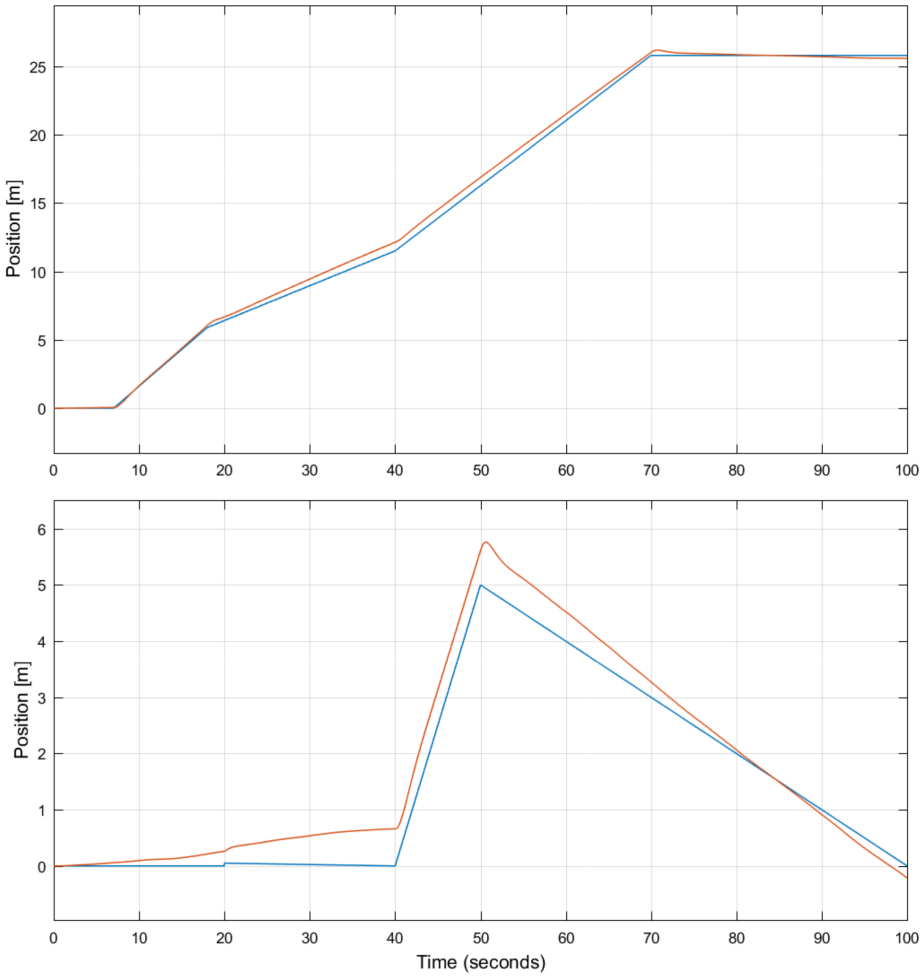

Figure 6.10: Theoretical position vs estimated position using Compass

In the orientation instead we have an addition of 45°.

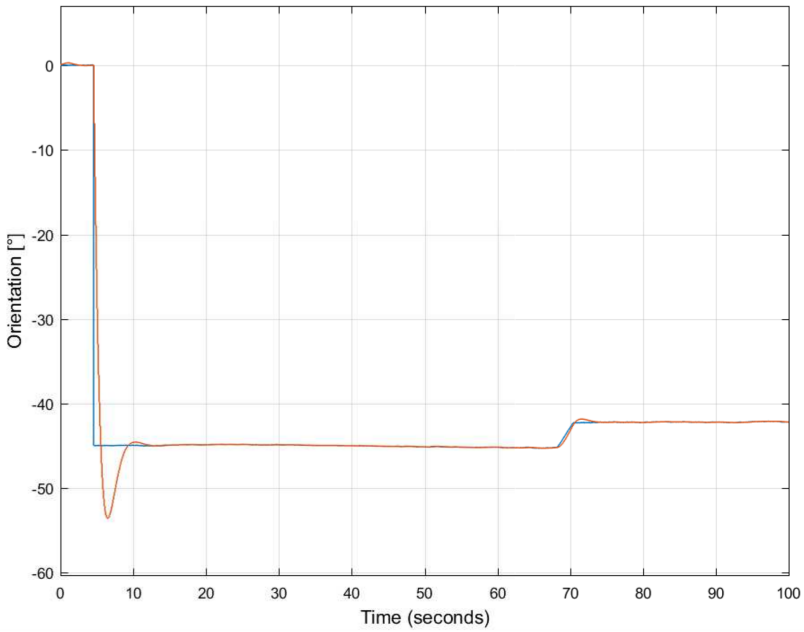

Figure 6.11: Theoretical orientation vs estimated orientation using Compass

6.4 Data Fusion Model with two Vehicular Sensor and Accelerometer Sensor

In this paragraph is reported the Data Fusion Model that fuse two measurements, the first measurement is an indirect measurement obtained from the vehicular sensor, the second measurement is obtained from accelerometer, also indirect. The Simulink Model is schematized in Figure 6.12.

In the graph of Figure 6.13 it's shown the comparison of two indirect measurements, and

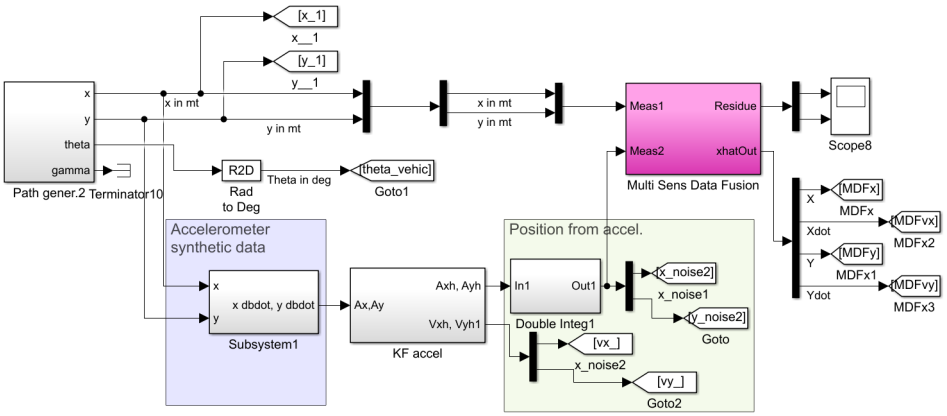

Figure 6.12: Simulink Model of Data Fusion with Vehicular and Accelerometer Sensor

the fusing of them. The first graph represent the position in of vehicle along x axis, the second graph, the position along y axis. The model is simplified, does not consider roto-translations, but only the movements on the x and y axes, so you can have a simulation that uses data fusion, with comparable data. In red is represented di displacement calculated coming from vehicular sensor measurement, in yellow the displacement calculated coming from accelerometer sensor measurement. Fused data is represented by blue line, here is between the two measurement line. In the two graphs it's quite evident that measurements coming from accelerometer sensors increase along the time, in this case exponentially, because the errors are integrated two times. The fusing, and also vehicle movement start after 5 seconds, due to the time necessary to initialize the Kalman Filter, and that of the Data Fusion. Measurements are considered acceptable for up to 60 seconds, after which the GPS sensor must provide the current position to the system. Or that the measurements coming from the IMU will be neglected, in order to have measurements coming only from the vehicle sensors.

In Figure 6.14, is possible to see also graphs about velocities on x and y axes, it should be noted that the speed obtained from the data fusion has delays, due to iterations in the algorithm itself.

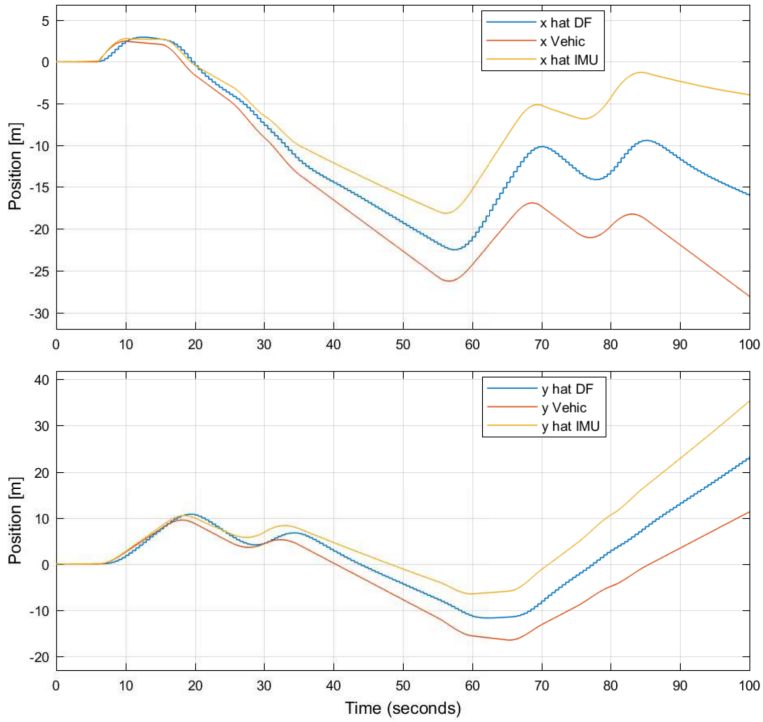

Figure 6.13: Comparison of positions: from vehicular sensor, from accelerometer sensor, and fused.

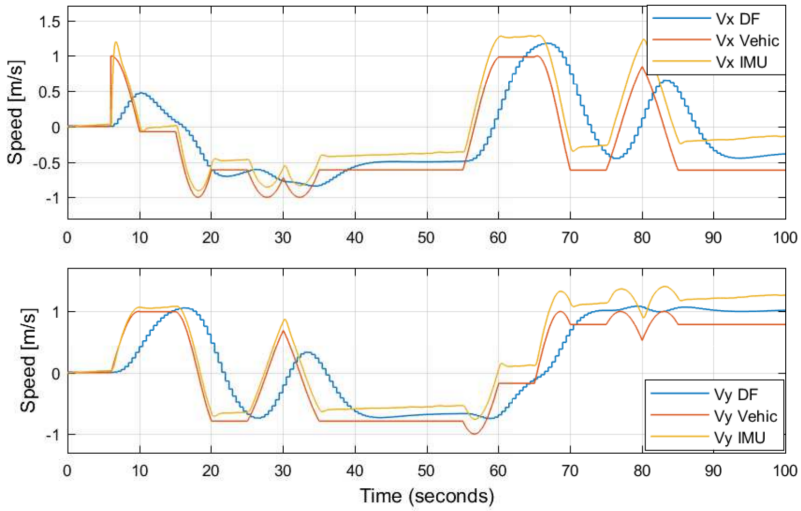

Figure 6.14: Comparison of speeds: from vehicular sensor, from accelerometer sensor, and fused.

As already occurred in the first paragraphs, the noise of the IMU sensors, in this case of the accelerometer, in Figure 6.15 we can notice an exponential growth of the error along the time.

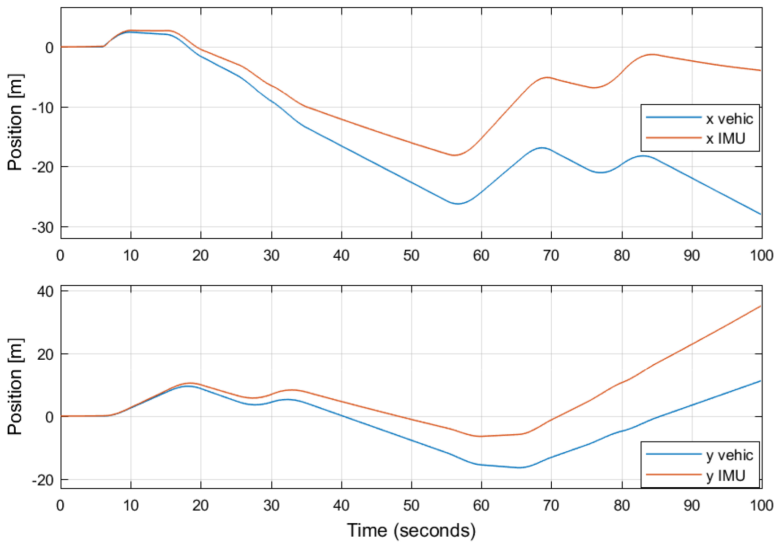

Figure 6.15: Real displacement vs Filtered displacement

Chapter 7

Kalman Filter

7.1 Tuning KF for Accellerometer Sensor

In our study, for design IMU, it has been taken into account accelerometer sensor MPU-6050 produced by InvenSense (very cheap MEMS). From technical data, using it at ±2g of full scale range, the sensitivity is 0.610·mg/LSB, and Zero-G level can change of ±35mg on X and Y axis, and ±60mg on Z axis. The model consider has two input for signal measurement, in this case are the accelerations Ax and Ay. In KF for first we tune the Zero-Order Hold on 0.05s, that is the same sample time of system, to have a fast response. After, we introduce Initial condition for estimated state, and Initial condition for estimated error covariance, as follow:

$$\hat{x}_{0acc} = [0.001; 0.01; 0.001; 0.01] \tag{7.1}$$

$$\hat{P}_{0acc} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.2}$$

Subsequently, we set parameters for Process noise covariance, that is:

$$Q_{acc} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{MAXacc} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{MAXacc} \end{bmatrix} \tag{7.3}$$

where Var(X)=1.1975*10^{-3}m/s^2/LSB, considering a noise with uniform density in the range $\pm 3\sigma$, with range of sensor set on $\pm 4g$, and State transition matrix:

$$A_{acc} = \begin{bmatrix} 1 & \sigma_t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \sigma_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.4}$$

where δ_t=0.05s as sample time set on the ZOH, and in the system.

7.2 Tuning KF for Gyroscope Sensor

"*The measurement errors of MEMS gyroscope is usually composed of the known errors and random errors. The known errors can be eliminated by the testing procedure, thus only the random errors are discussed for the gyroscope. A common model for the MEMS gyroscope is widely used in many application which mainly includes the white noise denoted as angular random walk (ARW) and bias drift due to rate random walk (RRW).*"[**4**][**1**],[**2**],[**3**]
The output rate signal of such gyroscopes can be described as a simplified model:

$$y(t) = \omega(t) + n(t) \tag{7.5}$$

where y(t) is the output rate signal of the gyroscope, ω(t) is the true rate signal, and n(t) is the white noise corrupting the gyroscope rate measurement. [**4**] The covariance matrix R for the noises v(t) can be expressed as:

$$R = \sigma_n^2 \cdot \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \vdots & \vdots & \vdots & \ddots \\ \rho & \rho & \cdots & 1 \end{bmatrix} \tag{7.6}$$

where σ_n^2 c and ρ is the constant cross-correlation factor. [**4**] An additional Kalman Filter has been implemented for measurements obtained from the gyroscope, as was previously done for the accelerometer. The tuning is as follow: In KF for first we tune the Zero-Order Hold on 0.05s, that is the same sample time of system, to have a fast response. After, we introduce Initial condition for estimated state, and Initial condition for estimated error covariance, as follow: Zero-Order Hold is set on 0.05 s, as in previous KF. Initial condition for estimated state, and Initial condition for estimated error covariance:

$$\hat{x}_{0gyro} = [0.001; 0.01; 0.001; 0.01] \tag{7.7}$$

$$\hat{P}_{0gyro} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.8}$$

Parameter for Process noise covariance, is:

$$Q_{gyro} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{MAXgyro} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{MAXgyro} \end{bmatrix} \tag{7.9}$$

where Var(X)=7.633*10^{-3}°/s/LSB, considering a noise with uniform density in the range

$\pm 3\sigma$, and range of sensor set on $\pm 250°/s$, and State transition matrix:

$$A_{gyro} = \begin{bmatrix} 1 & \sigma_t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \sigma_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.10}$$

where δ_t=0.05 s.

Chapter 8

Multisensorial Data Fusion

Chapter 9

Experimental Results

Chapter 10

Sensor Calibration

"*A sensor is a device that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument.*"[**11**] The operating principle is to transform a physical quantity into an electrical or mechanical one. Nowadays there are different types of sensors for several functionalities, here below we will deal with the sensors taken into consideration for our case study. The sensors we use produce an electronic signal, which are send in SPI protocol to microcontroller. Our software is implemented using Simulink Embedded Coder using Model Based Software Design approach on reference board with a processor NXP K64F. The sensors can be of an absolute type, or of a relative type. Those of a relative type measure a quantity with respect to a fixed part, or to a mobile system. Unlike those of the absolute type, they measure an amount with respect to conventional zero. [**12**] In our two model discussed previously, tests were carried out with accelerometers, gyroscopes and magnetometers. Instead, the wheel encoders and steering encoder were then simulated, they are not similar between them, the first is absolute and the second is relative.

10.1 Accelerometer Sensor

The sensor chosen to perform accelerometer functionality is InvenSense MPU6050, measuring the acces- sories on 9 axes, X, Y, and Z, the same sensor also arranges a gyroscope on board.

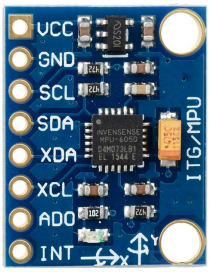

Figure 10.1: MPU-6050 Sensor[**21**]

"*The MPU devices provide the world's first integrated 6-axis motion processor solution that eliminates the package-level gyroscope and accelerometer cross-axis misalignment associated with discrete solutions. The devices combine a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die together with an onboard Digital Motion Processor™(DMP™) capable of processing complex 9-axis sensor fusion algorithms using the field-proven and proprietary MotionFusion™engine.*"[**21**] The sensor has the following sensitivity specification,under the following conditions: VDD = $2.375V - 3.46V$, VLOGIC (MPU-6050 only) = 1.8Vpm5% or VDD, T_A =25°C[**21**].

PARAMETER	**CONDITIONS**	**MIN**	**TYP**	**MAX**	**UNITS**
Full-Scale Range	AFS_SEL=0		±2		g
	AFS_SEL=1		±4		g
	AFS_SEL=2		±8		g
	AFS_SEL=3		±16		g
ADC Word Length	Two's complement		16		bits
Sensitivity Scale Factor	AFS_SEL=0		16386		LSB/g
	AFS_SEL=1		8192		LSB/g
	AFS_SEL=2		4096		LSB/g
	AFS_SEL=3		2048		LSB/g
Initial Calibration Tolerance			±3		%
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C
Nonlinearity	Best Fit Straight Line		0.5		%
Cross-Axis Sensitivity			±2		%

Table 10.1: Accelerometer Sensitivity

Our setting is AFS_SEL=1, where the sensor works in the range ±4 g. The parameters for calibration called ZERO-G OUTPUT are the following[**21**]:

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Initial Calibration Tolerance	X an Y axes		±60		mg
	Z axis		±80		mg
Zero-G Level Change vs. Temperature	X an Y axes, 0°C to +70°C		±35		mg
	Z axis, 0°C to +70°C		±60		mg

Table 10.2: Accelerometer ZERO-G Output

Other parameters most important for our study are about noise performance:

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
NOISE PERFORMANCE					
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		$\mu g/\sqrt{Hz}$
LOW PASS FILTER RESPONSE					
	Programmable Range	5		260	Hz

Table 10.3: Accelerometer Noise performance and Filter response

The diagram below shows the orientation of the axes of sensitivity and the polarity of rotation. Note the pin 1, identifier (•) in figure.

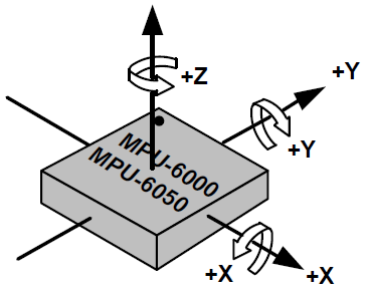

Figure 10.2: MPU-6050 Axes Orientation. Figure from [**21**]

Same axes are valid also for gyroscope sensor. Always from the manuals, we have cross-axis sensitivity as a percentage of the gyroscope or accelerometer's sensitivity for a given orientation error, respectively. As shown below:

Orientation Error (θ or ϕ)	Cross-Axis Sensitivity ($\sin\theta$ or $\sin\phi$)
0°	0%
0.5°	0.87%
1°	1.75%

Table 10.4: Sensor Orientation Error

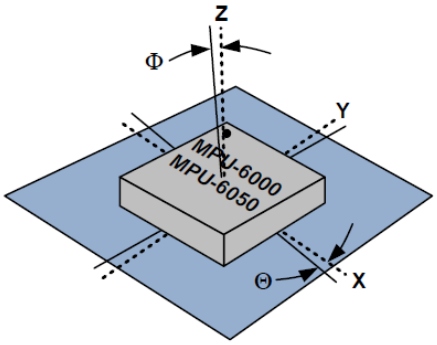

Figure 10.3: Cross-axis Sensitivity. Figure from [**21**].

To proceed with the calibration of the accelerometer, we must first distinguish the absolute and relative calibration. Absolute acceleration measurement could be measurement of earth's gravitation field or the acceleration forces experienced in an automobile. Unlike, un example of using relative acceleration is the calculation of orientation angle using ratios of the reading from the x, y, and z accelerometer channels.[**13**] "*Although the earth's gravitational field is often stated to be 9.81 ms^{-2}, in practice the apparent gravitational field measured by an accelerometer varies by 0.7% from minimum to maximum over the earth's surface as a consequence of the earth's rotation, the earth's equatorial bulge and the effects of altitude. The apparent gravitational field at sea level at the north pole is 9.832 ms^{-2} but is only 9.763 ms^{-2} at the 5895 m summit of Mount Kilimanjaro located almost on the equator.*"[**13**] To calibrate the sensor, we can use zero-g level and sensitivity parameters indicated above, and converting raw-measurements A_x, A_y, and A_z into normalized measurements A_{x1}, A_{y1}, and A_{z1}. In order to have an high compensating level calibration, is suggested to use tilt-compensated electronic compasses and level monitoring systems, for accelerometer calibration.[**14**] Accelerometer raw data and normalized measurements are linked by the following formula 10.1:

$$\begin{bmatrix} A_{x1} \\ A_{y1} \\ A_{z1} \end{bmatrix} = [A\_m]_{3X3} \cdot \begin{bmatrix} 1/A\_SC_x & 0 & 0 \\ 0 & 1/A\_SC_y & 0 \\ 0 & 0 & 1/A\_SC_z \end{bmatrix} \cdot \begin{bmatrix} A_x - A\_OS_x \\ A_y - A\_OS_y \\ A_z - A\_OS_z \end{bmatrix}$$
$$= \begin{bmatrix} ACC_{11} & ACC_{12} & ACC_{13} \\ ACC_{21} & ACC_{22} & ACC_{23} \\ ACC_{31} & ACC_{32} & ACC_{33} \end{bmatrix} \cdot \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} + \begin{bmatrix} ACC_{10} \\ ACC_{20} \\ ACC_{30} \end{bmatrix} \tag{10.1}$$

Where A_m is the 3 x 3 misalignment matrix between the accelerometer sensing axes and the sensor's axes, A_SC_x,A_SC_y and A_SC_z are sensitivity (or scale factor) and A_OS_x,A_OS_y and A_OS_z is the zero-g level called also Offset. The goal of accelerometer calibration is to determine 12 parameters from ACC_{10} to ACC_{33},so that with any given raw measurements at arbitrary positions, the normalized values A_{x1}, A_{y1} and A_{z1} can be obtained, using the follow equation 10.2[**14**]:

$$|A| = \sqrt{A_{x1}^2 + A_{y1}^2 + A_{z1}^2} = 1 \tag{10.2}$$

Calibration can be performed at raw data with ODR = 100 Hz at each position with known A_{x1}, A_{y1} and A_{z1}. Then apply the least square method to obtain the 12 accelerometer calibration parameters. This method requires professional equipment, unlikethe calibration performed by us consists in canceling the acceleration gravity effect. In our application accelerometer sensor is used as relative measurement, we measure acceleration on 3 axis. In order to cancel gravity acceleration, we consider that the vehicle is stationary for 5 second, during this time frame, we measure the three accelerations, and calculate the norm. We assume the norm as an acceleration of gravity, and calculate the projections on the axes, generally Ax and Ay, and we'll use the new reference frame.

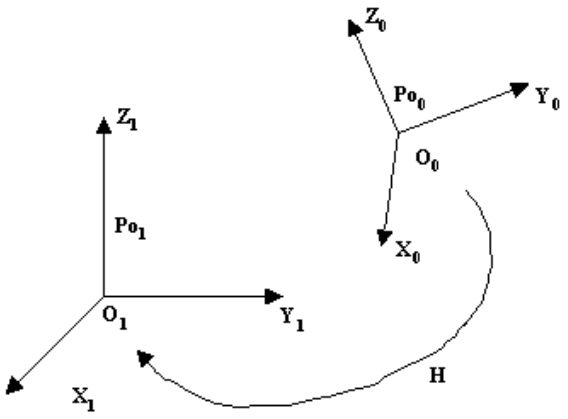

Figure 10.4: Gravity compensation Reference Frame

The proposed solution is implemented as a State Machine inside a Simulink Model,

normalized acceleration is computed as:

$$\|A_{nrm}\| = \sqrt{A_x^2 + A_y^2 + A_z^2} \tag{10.3}$$

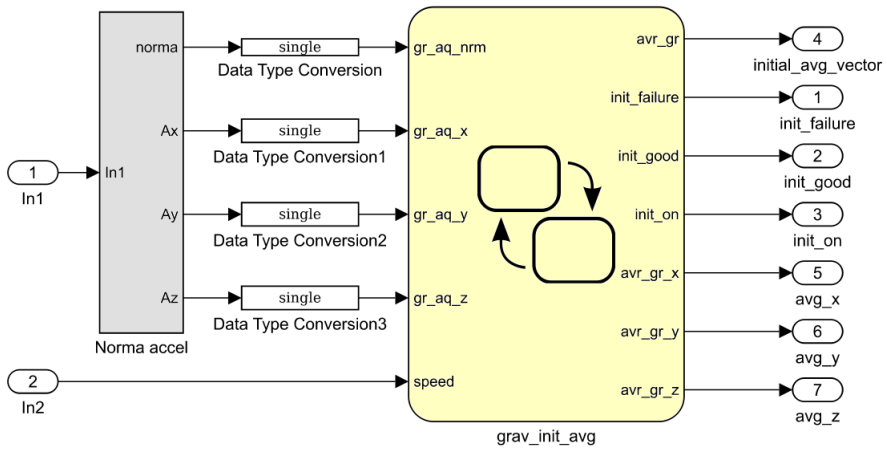

Figure 10.5: Gravity compensation Simulink Subsystem

Vehicle when it is stationary, has the following average acceleration: avr_gr_x, avr_gr_y and avr_gr_z, they are computed in State Flow, only when vehicle speed is equal 0, otherwise the chart goes in failure mode.

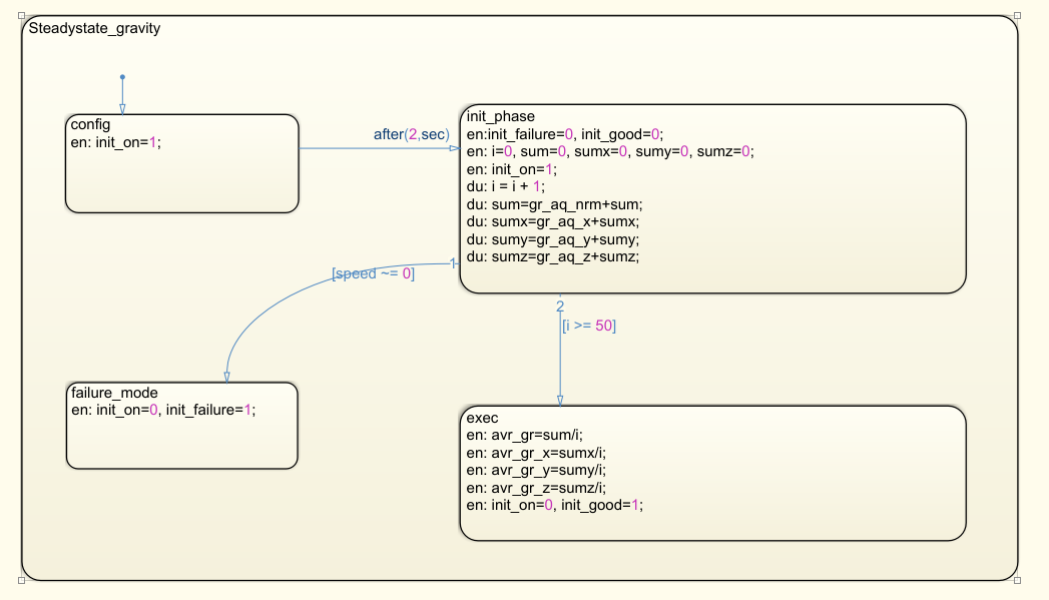

Figure 10.6: Gravity average State Flow

10.2 Gyroscope Sensor

Gyroscope sensor is also integrated in the package MPU6050, and it has the same orientation axis of accelerometer sensor. The sensor has the following sensitivity specification,under the following conditions: VDD = $2.375V - 3.46V$, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A =25°C[**21**].

PARAMETER	**CONDITIONS**	**MIN**	**TYP**	**MAX**	**UNITS**
Full-Scale Range	FS_SEL=0		±250		°/s
	FS_SEL=1		±500		°/s
	FS_SEL=2		±1000		°/s
	FS_SEL=0=3		±2000		°/s
ADC Word Length			16		bits
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(°/s)
	FS_SEL=1		65.5		LSB/(°/s)
	FS_SEL=2		32.8		LSB/(°/s)
	FS_SEL=3		16.4		LSB/(°/s)
Initial Calibration Tolerance			±3		%
Sensitivity Factor Tolerance	25°C	-3		+3	%
Sensitivity Factor Tolerance Variation Over Temperature			±2		%
Nonlinearity	Best Fit Straight Line; 25°C		0.2		%
Cross-Axis Sensitivity			±2		%

Table 10.5: Gyroscope's Sensitivity

Our setting is FS_SEL=0, where the sensor works in the range ±250 °/s. The parameters for calibration called ZERO-RATE OUTPUT are the following[**21**]:

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Initial ZRO Tolerance	25°C		±20		°/s
ZRO vs Temperature	-40°C to +85°C		±20		°/s
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mV$_{pp}$; V$_{DD}$=2.5V		±0.2		°/s
Power-Supply Sensitivity (10-250Hz)	Sine wave, 100mV$_{pp}$; V$_{DD}$=2.5V		±0.2		°/s
Power-Supply Sens. (250-100kHz)	Sine wave, 100mV$_{pp}$; V$_{DD}$=2.5V		±4		°/s
Linear Accel. Sens.	Static		0.1		°/s/g

Table 10.6: Gyroscope's Zero Rate Output

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
NOISE PERFORM.	**FS_SEL=0**				
Total RMS Noise	DLPFCFG=2 (100Hz)		0.05		°/s-rms
Low-frequency RMS Noise	Bandwith 1Hz to 10Hz		0.033		°/s-rms
Rate Noise Spectral Density	At 10Hz		0.005		°/s/$\sqrt{Hz}$
MECHANICAL FREQUENCIES					
X-Axis		30	33	36	kHz
Y-Axis		27	30	33	kHz
Z-Axis		24	27	30	kHz
LOW PASS FILTER RESPONSE	Programmable Range	5		256	Hz
OUT. DATA RATE	Programmable	4		8000	Hz
START-UP TIME	**DLPFCFG=0**				
ZRO Setting (from power-on)	to ±1°/s of Final		30		ms

Table 10.7: Gyroscope's Noise performance and Mechanical frequencies

For gyroscope calibration, we must take into account that oscillating behavior is different between the three axes: Gyroscope sensor measures an absolute size, but as in the tables above, there is a sensitivity factor tolerance that can be compensated using the offsets. For first we must consider that "*MEMS gyroscopes that are sensitive to acceleration*

and gravity. Therefore, orientation with respect to gravity is worth consideration."[**15**] In figure[**15**], we can see the exact application factor, true value in orange, raw value in blue:

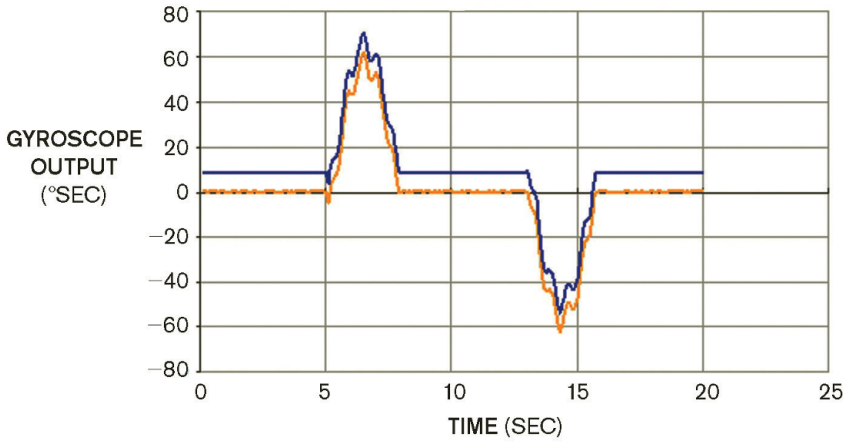

Figure 10.7: Gyroscope with correction factor[**15**]

An Analog Device application note [**15**] suggested us to follow this four steps to calibrate gyroscope sensor: 1. Calculate the bias-offset-correction factor by averaging the first 3 seconds of data. The bias correction will be the opposite polarity of this average. In this example, the bias-correction factor is -8.6°/sec. This correction yields an accuracy of greater than 0.1°/sec for the bias estimate. 2. Subtract the bias estimate from the time record. Then integrate output data from the 4-second time stamp to the 9-second time stamp. In this case, the measured angle displacement is 95.1°. The scale factor from this step is 90° divided by 95.1°, or 0.946. 3. Using the bias-corrected response from Step 2, integrate output data from the 12-second time stamp to the 16-second time stamp. In this case, the measured angle displacement is -95.3°. The scale factor from this step is 90° divided by 95.3°, or 0.944. 4. Average the results of steps 2 and 3 to calculate the scale-factor correction, which in this example is 0.945.[**15**]

10.3 Magnetometer Sensor

Magnetometer sensor used to test the implementation is NXP FXOS8700CQ provided on the reference board NXP K64F, it uses SPI communication, it has the axes arranged parallel to the other two MEMS sensors. FXOS8700CQ sensor include accelerometer and magnetometer, we consider use in the magnetometer mode only. The sensor has the following sensitivity specification,under the following conditions: VDD = $2.5V$, VDDIO = 1.8V, T_A =25°C[**22**].

PARAMETER	**CONDITIONS**	**MIN**	**TYP**	**MAX**	**UNITS**
Full-Scale Range		1200			μT
Sensitivity			0.1		μT/LSB
Sensitivity Change vs. Temperature			±0.1		%/°C
Zero-flux offset accuracy			±10		μT
Zero-flux offset Change with temperature			±0.8		μT/°C
Hysteresis			±0.5		%FS$_{MAG}$
Nonlinearity Deviation from best-fit straight line			±1		%FS$_{MAG}$
Magnetometer output noise	ODR=800Hz, OSR=2		1.5		μT-rms
	ODR=400Hz, OSR=4		1.2		μT-rms
	ODR=200Hz, OSR=8		0.85		μT-rms
	ODR=100Hz, OSR=16		0.6		μT-rms
	ODR=50Hz, OSR=32		0.5		μT-rms
	ODR=12.5Hz, OSR=128		0.35		μT-rms
	ODR=6.25Hz, OSR=256		0.3		μT-rms
	ODR=1.56Hz, OSR=1024		0.3		μT-rms

Table 10.8: Magnetometer Sensitivity[**22**]

Before discussing about calibration, will be illustrate problems of the magnetometers also

called e-compasses. "*Hard-iron effects are due to magnetic materials in the vicinity of the sensor. These materials result in an apparent offset to sensor readings when the source of interference is fixed spatially, relative to the sensor. For a given point in space, plotting magnetometer measurements at various sensor rotations results in the sphere shown on the right hand side of figure*"[**16**]:

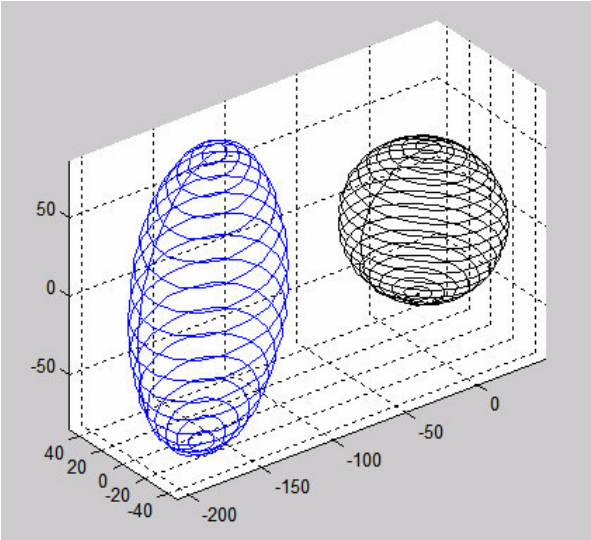

Figure 10.8: Magnetometer measurements in the sphere

"*This makes sense, as the magnitude of the 3D magnetic field should not change just because the sensor is rotated. Soft-iron effects result from the interaction of ferrous materials near the sensor interacting with the ambient magnetic field. If the source of soft iron interference is again fixed spatially with respect to the sensor and does not demonstrate magnetic hysteresis, then the sphere of plotted measurements is distorted into an ellipsoid. This is shown (along with a hard-iron offset) on the left side in blue.*"[**16**] To computing compensation, computation are similar to accelerometer sensor, we have the following variables: scale factors M_SCx, M_SCy, and M_SCz, the offsets M_OSx, M_OSy, M_OSz, caused by the hard-iron distortion and the M_si matrix caused by the soft-iron distortion were determined. The relation between the values of the standardized magnetometer and the raw data of the sensor itself is:

$$\begin{bmatrix} M_{x1} \\ M_{y1} \\ M_{z1} \end{bmatrix} = [M\_m]_{3X3} \cdot \begin{bmatrix} 1/M\_SC_x & 0 & 0 \\ 0 & 1/M\_SC_y & 0 \\ 0 & 0 & 1/M\_SC_z \end{bmatrix} \cdot [M\_si]_{3X3} \begin{bmatrix} M_x - M\_OS_x \\ M_y - M\_OS_y \\ M_z - M\_OS_z \end{bmatrix}$$
$$= \begin{bmatrix} MAG_{11} & MAG_{12} & MAG_{13} \\ MAG_{21} & MAG_{22} & MAG_{23} \\ MAG_{31} & MAG_{32} & MAG_{33} \end{bmatrix} \cdot \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} + \begin{bmatrix} MAG_{10} \\ MAG_{20} \\ MAG_{30} \end{bmatrix} \tag{10.4}$$

where the matrix M_m is the 3 × 3 matrix of misalignment between the magnetic axes of the sensor and the axes of the board where the sensor is mounted. Assuming the solder chip perfectly parallel to the board, this matrix can be assumed to be a 3 × 3 identity matrix [**17**] [**18**] [**19**]. To calculate the offset values of the sensor, also altered by presence of metallic materials, we proceeded to acquire the raw measurements from the magnetometer, with the sensor placed on a car, with horizontal x and y axes. Several turns have been made, around a fixed point, and the values obtained have been reported on the 2D and 3D charts.

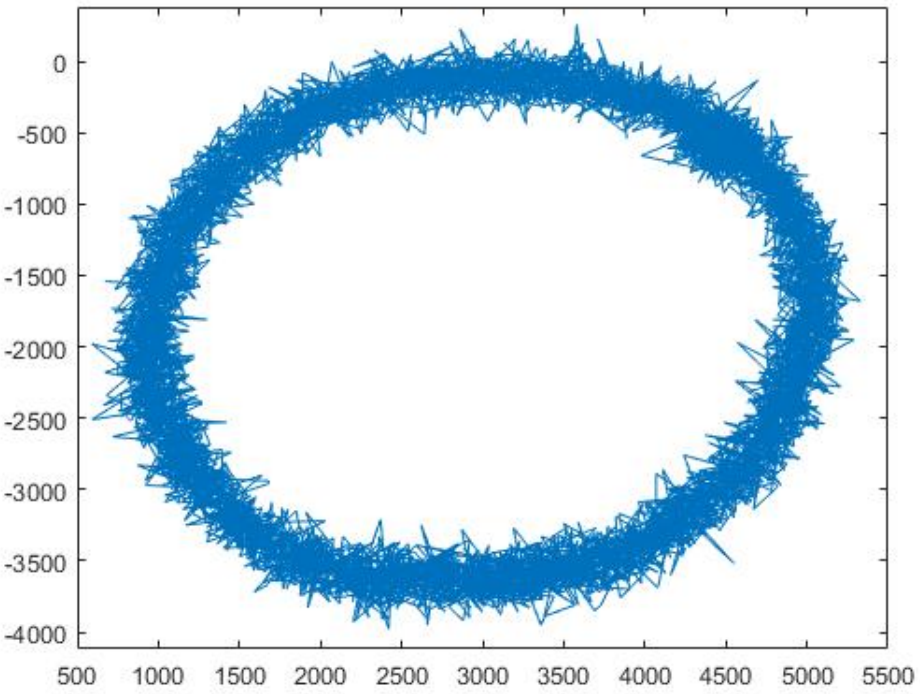

Figure 10.9: Magnetometer raw measurements on X and Y

The acquisition has produced very satisfactory results, the slight distortion is due to presence of iron material, but not excessive. We then evaluated along the Z axis, the variations far from the average, which usually indicate the presence of manholes.
The most important graph is 2D, the 3D graph is useful for verifying substantial deviations on the z axis, even if a direct observation can be made directly from changes of Mz over time. A further confirmation of the quality of measurements, that we can obtain from the analysis of this last graph, the values are almost constant. In case there is a presence of drains, or other iron objects, it is better to ignore the magnetometer in the calculation of the Yaw angle.
One time that we have calculate offset parameter, we add a subsystem to correct signals provided by the sensor, as follow:

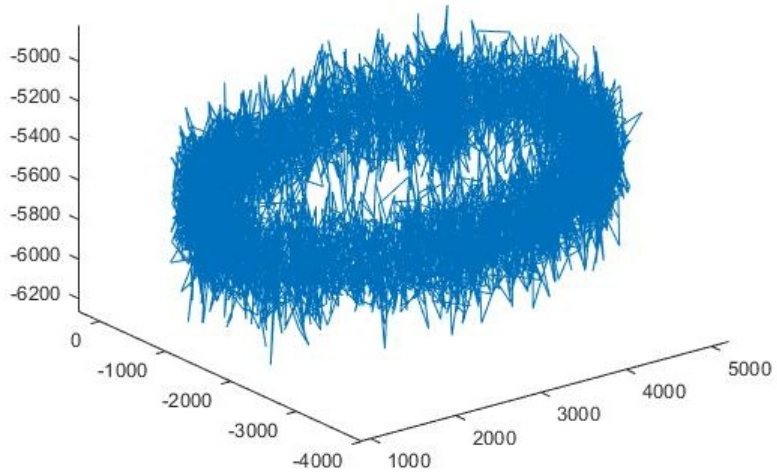

Figure 10.10: Magnetometer raw measurements, 3D graph

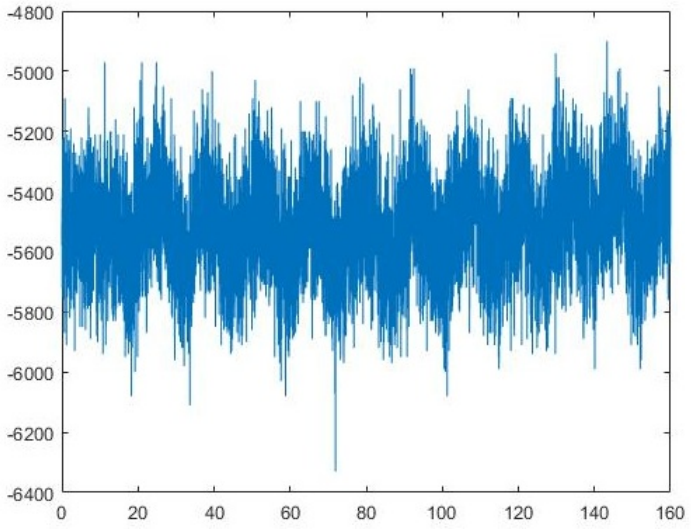

Figure 10.11: Magnetometer's raw measurements on Z axis

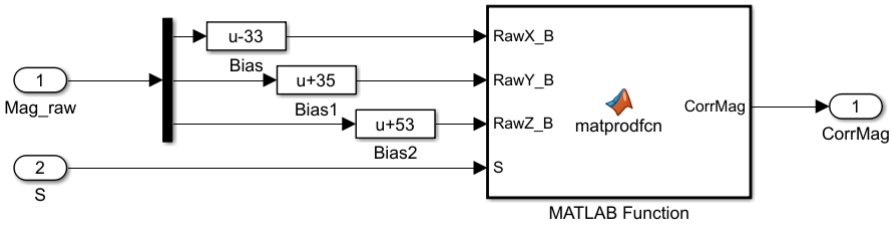

Figure 10.12: Subsystem with offset Magneto compensation

```
1 function CorrMag = matprodfcn(RawX_B,RawY_B,RawZ_B,S)
2 Raw_B=[RawX_B,RawY_B,RawZ_B];
3 CorrMag = Raw_B*S;
```

Listing 10.1: matprodfcn

10.4 Wheel and Steering encoder

Encoders can be divided into two sub-categories: absolute encoders and relative encoders. In the relative encoders, or in incremental encoders, the electrical output signals are proportional to the displacement of a rotating part (rotor) with respect to a solid part to the body (stator); In relative encoders, simple electronic circuits can count of notches or optical, magnetic, mechanical or other lines, uniformly distributed on the rotor or on the stator, producing signals proportional to the position. In our study vehicular sensors are: absolute encoder type for the measurement of the steering angle, and incremental encoder for the measurement of the speed wheel (odometer). Nowadays, the technology offers us encoders at very high resolutions, so much so that in our case study, the accuracy of the bicycle model is better than the IMU model, so we could overlook corrections related to the precision of the encoders.[**12**]

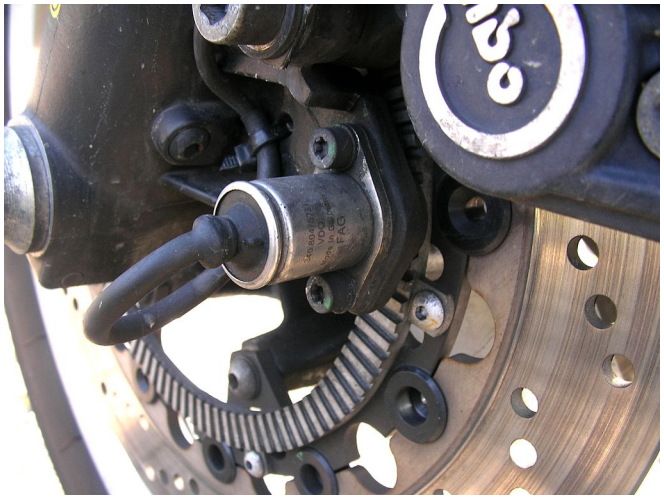

Figure 10.13: A relative encoder implemented by means of a phonic wheel, used as motorbike ABS sensor

Chapter 11

Quality of Achieved Results

Chapter 12

Conclusion

Bibliography

[1] Minha Park, Yang Gao - Error and Performance Analysis of MEMS-based Inertial Sensors with a Low-cost GPS Receiver *Sensors 2008, 8, 2240-2261, March 2008. MDPI. Full Research Paper* (pp. 2260).

[2] Xue, L., Yuan, W.; Chang, H.; Jiang, C. MEMS-Based Multi-Sensor Integrated Attitude Estimation Technology for Mav Applications. *In Proceedings of the IEEE NEMS, Shenzhen, China, 5–8 January 2009* (pp. 1031–1035).

[3] Lam, Q.M.; Stamatakos, N.; Woodruff, C.; Ashton, S. Gyro Modeling and Estimation of Its Random Noise Sources. *In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, TX, USA, 11–14 August 2003* (pp. 1–11).

[4] Chengyu Jiang, Liang Xue, Honglong Chang, Guangmin Yuan, Weizheng Yuan - Signal Processing of MEMS Gyroscope Arrays to Improve Accuracy Using a 1st Order Markov for Rate Signal Modeling *Sensors 2008, 8, 2240-2261, March 2008. MDPI. Full Research PaperSensors 2012, 12, 1720-137, February 2012. MDPI. Full Research Paper* (pp. 1720-137).

[5] Inertial Navigation System *https://en.wikipedia.org/wiki/Inertial_navigation_system*

[6] Greg Welch, Gary Bishop - An Introduction to the Kalman Filter *Department of Computer Science University of North Carolina at Chapel Hill. Welch & Bishop, TR 95-041, July 2006.* (pp. 1-16).

[7] Grewal, Mohinder S., and Angus P. Andrews. - Kalman Filterinf Theory and Practise. *Upper Saddle River, NJ USA, Prentice Hall, 1993*

[8] Davide Spina - Multi-Data Snensor Fusion for Robotics Applications. *System Engineering - Università degli studi di Catania Ph.D Thesis 2014* (pp. 1-141)

[9] Data Fusion, estimation and sensor calibration *PC- based instrumentation and microcontrollers - University of Oslo FYS3240, March 2015* (pp. 1-51)

[10] Oliver J. Woodman - An introductionm to inertial navigation *Computer Laboratory - University of Cambridge. Technical Report UCAM-CL-TR-696, ISSN 1476-2986, TR 95-041, August 2017.* (pp. 1-37)

[11] Inertial navigation system *http://www.juliantrubin.com/encyclopedia/electronics/sensor.html*

[12] Basilio Bona - Sensori per la Robotica *Department of Control Engineering Politecnico di Torino DAUIN/BB/2006/07.02, September 2006.*(pp. 1-38).

[13] Mark Pedley - High-Precision Calibration of a Three-Axis Accelerometer *Application Note - NXP Freescale Semiconductor Document Number: AN4399 Rev2.0, October*

2015.(pp1-35)

[14] Parameters and calibration of a low-g 3-axis accelerometer *AN4508 Application note - STMicroelectronics Document Number:DocID026444 Rev 1, 2014.*(pp 1-13).

[15] Mark-Looney - A simple calibration for MEMS gyroscope *Analog Devices EDN Europe, 2010.*(pp 1-4).

[16] Freescale Sensor Fusion Library for Kinetis MCUs *Freescale Sensor Fusion Release Document Number: FSFLK_DS, Rev. 0.7, September 2015.*(pp 1-37).

[17] Calibrazione magnetometro MEMS (e-Compass) *http://www.aepwebmaster.it/2016/07/16/calibrazione-magnetometro-mems-e-compass/*

[18] Christopher Konvalin - Compensating for Tilt, Hard-Iron, and Soft-Iron Effects *https://www.sensormag.com/components/compensating-for-tilt-hard-iron-and-soft-iron-effects December 2009.*

[19] Using LSM303DL H for a tilt compensated electronic compass *AN3192 Application note - STMicroelectronics, 2010.*

[20] Van Graas Frank - Workshop on GNSS Data Application to Low Latitde Ionospheric Research *School of Electrical Engineering and Computer Science - Ohio University, May 2013.*

[21] MPU-6000 and MPU-6050 Production Specification Revision 3.4 *InvenSense, Document Number: PS-MPU-6000A-00 Revision:3.4, August 2013*(pp 1-52)

[22] FXOS8700CQ 6-axis sensor with integrated linear accelerometer and magnetometer Rev. 8 *NXP Semiconductor, Document identifier: FXOS8700CQ, Aptil 2017*(pp 1-116)

[23] How function GPS *https://www.uniquevisitor.it/magazine/come-funziona-gps.php*

[24] GPS - Space Segment *https://www.gps.gov/systems/gps/space/*

[25] United States Department of Transportation - Federal Aviation Administartion *https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/faq/gps/#2*

[26] Peter Corke - Robotics, Vision and Control *ISBN 978-3-642-20143-1, Springer*(pp. 65-86)

[27] Ackerman model *www.racing-car-technology.com.au/SteeringAckerman4.doc*

[28] Basilio Bona - Dynamic Modelling of Mechatronic Systems *Celid ISBN 978-88-6789-011-8, September 2013*