Radiofrequency Positioning System Aided With Sensor Fusion

Miguel Ángel Vicens Oviedo

Directed by:

Prof. Jordi Mallorquí

MASTER THESIS

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER OF RESEARCH IN INFORMATION AND COMMUNICATION TECHNOLOGIES

Abstract

A lo largo de los últimos años han ido apareciendo diversos sistemas de radionavegación con distintos grados de precisión, entre otros: LORAN (Long Range Navigation), Decca, GNSS (Global Navigation Satellite System), siendo este último el más utilizado en la actualidad.

El sistema GNSS, que engloba los sistemas GPS (Global Positioning System) y GLONASS (GLObal NAvigation Satellite System), está conformado por una constelación de satélites, que se utilizan para determinar el posicionamiento y localización en cualquier parte del globo terrestre. Estos sistemas fueron ideados con la finalidad de posicionar vehículos y aeronaves en un ámbito militar. Con la generalización de dichos sistemas para usos civiles, los sistemas GPS han extendido su uso, evolucionando y aportando una precisión de hasta unos pocos metros. Pero el GPS podría no ser la mejor opción para aplicaciones de mayor precisión.

En nuestro ámbito de investigación, se requiere conocer de forma muy precisa la posición en cada momento de una plataforma en movimiento. Entre los sistemas disponibles a nivel comercial, ninguno se ajusta a nuestras especificaciones, por lo que se ha implementado un sistema a medida para mejorar el posicionamiento.

Los sistemas utilizados en este proyecto son: la técnica de estimación TDOA (Time Difference Of Arrival), que combinada con el algoritmo de multilateralización (MLAT) nos permite medir la diferencia de fase de la señal chirp (que varía su frecuencia en función del tiempo) y convertirla a diferencias de posición. También usaremos unos sensores inerciales (acelerómetro y giroscopio), que a través del Filtro Extendido de Kalman (EKF) y la fusión de sensores se encargarán de mejorar la precisión de esta estimación.

En esta tesis, evaluaremos dos escenarios para probar el EKF con diferentes ecuaciónes dinámicas de movimiento. El primer escenario es la simulación de cuatro antenas transmisoras (slaves) dispuestas en forma de cuadrado, distribuidas en un área geográfica concreta, en la cual se halla el vehiculo (antena receptora o master), que debe calcular su posicion dentro de esa area con la ayuda del EKF y los sensores inerciales.

El segundo escenario será un caso práctico en el cual disponemos de tres antenas transmisoras colocadas en forma de triángulo. En este caso, la antena receptora se situará dentro del área del triángulo y se irá moviendo milimétricamente en una única dimensión para posteriormente evaluar los resultados con el EKF.

En este contexto, la intención de esta tesis ha sido implementar mejoras en el sistema de posicionamiento, introduciendo elementos que proporcionan información adicional o redundante y procesarla con este fin.

Abstract

During the last few years, different navigation systems with varying degrees of accuracy have appeared, such as: LORAN (Long Range Navigation), Decca, GNSS (GLObal NAvigation Satellite System), the latter being the most widely used nowadays.

GNSS, which includes GPS (Global Positioning System) and GLONASS (Global Navigation Satellite System) , consists of a constellation of satellites which are used for positioning and location anywhere on the globe. These systems were designed in order to position vehicles and aircraft for military purposes. With the widespread use of such systems for civilian use, GPS have extended their use, evolving and providing accuracy to a few meters. But the GPS might not be the best choice for higher accuracy applications.

In our research field , we need to know very precisely the current position of a moving platform. Among the commercially available systems, none fits our specifications, therefore a system to improve positioning has been implemented.

The systems used in this project are: the estimation technique TDOA (Time Difference of Arrival), which combined with the multilateration algorithm (MLAT) allows us to measure the phase difference of the chirp signal (frequency varies depending on the time) and transform to positions difference. We will also use some inertial sensors (accelerometer and gyroscope), which through the Extended Kalman Filter (EKF) and sensor fusion will be responsible for improving the accuracy of this estimate.

In this thesis, we will consider two scenarios to test the EKF with different dynamic equations of motion. The first stage is the simulation of four transmiting transmit antennas (slaves) arranged in a square, distributed in a particular geographical area in which the vehicle (or master receiving antenna) must calculate its position within that area with EKF and inertial sensors support.

The second stage will be a case in which we have three transmitting antennas placed in a triangle shape. In this case the receiving antenna will be located within the area of the triangle and it will be really moving in one dimension to further evaluate the results with EKF.

In this context, the intention of this thesis has been to implement improvements in the positioning system, introducing elements that provide additional or redundant information and process it for this purpose.

Abstract

Al llarg dels darrers anys han anat apareixent diversos sistemes de radionavegació amb diferents graus de precisió, entre d'altres: LORAN (LOng RAnge Navigation), Decca, GNSS (Global Navigation Satellite System), sent aquest últim el més utilitzat en l'actualitat.

El sistema GNSS, que engloba els sistemes GPS (Global Positioning System) i GLONASS (GLObal NAvigation Satellite System), està conformat per una constel·lació de satèl·lits, que s'utilitzen per determinar el posicionament i localització en qualsevol part del globus terrestre. Aquests sistemes van ser ideats amb la finalitat de posicionar vehicles i aeronaus en un àmbit militar. Amb la generalització d'aquests sistemes per a usos civils, els GPS han estès el seu ús, evolucionant i aportant una precisió de fins a uns pocs metres. Però el GPS podria no ser la millor opció per a aplicacions de major precisió.

En el nostre àmbit de recerca, es requereix conèixer de forma molt precisa la posició en cada moment d'una plataforma en moviment. Entre els sistemes disponibles a nivell comercial, cap s'ajusta a les nostres especificacions, pel que s'ha implementat un sistema a mida per millorar el posicionament.

Els sistemes utilitzats en aquest projecte són: la tècnica d'estimació TDOA (Time Difference Of Arrival), que combinada amb l'algoritme de multilateralització (MLAT) ens permet mesurar la diferència de fase del senyal chirp (que varia la seva freqüència en funció del temps) i convertir-la a diferències de posició. També farem servir uns sensors inercials (acceleròmetre i giroscopi), que a través del Filtre Estès de Kalman (EKF) i la fusió de sensors s'encarregaran de millorar la precisió d'aquesta estimació.

En aquesta tesi, avaluarem dos escenaris per provar el EKF amb diferents equacions dinàmiques de moviment. El primer escenari és la simulació de quatre antenes transmissores (slaves) disposades en forma de quadrat, distribuïdes en una àrea geogràfica concreta, en la qual es troba el vehicle (antena receptora o màster), que ha de calcular la seva posició dins d'aquesta àrea amb l'ajuda del EKF i els sensors inercials.

El segon escenari serà un cas pràctic en el qual disposem de tres antenes transmissores col·locades en forma de triangle. En aquest cas, l'antena receptora es situarà dins l'àrea del triangle i s'anirà movent mil·limètricament en una única dimensió per posteriorment avaluar els resultats amb el EKF.

En aquest context, la intenció d'aquesta tesi ha estat implementar millores en el sistema de posicionament, introduint elements que proporcionen informació addicional o redundant i processar-la amb aquesta finalitat.

Index

INDEX OF FIGURES

INDEX OF TABLES

ABREVIATIONS

AOA	*Angle Of Arrival*
DF	*Direction Finding*
DR	*Dead reckoning*
RSS	*Received Signal Strength*
POA	*Carrier signal Phase Of Arrival*
RF	*Radiofrequency*
TOA	*Time of Arrival*
TDOA	*Time Difference Of Arrival*
FDOA	*Frequency Difference of Arrival*
DD	*Doppler Difference*
INS	*Inertial Navigation System*
MEMS	*Micro-Electro-Mechanical System*
EKF	*Extended Kalman Filter*
KF	*Kalman Filter*
PRF	*Pulse Repetition Frequency*
PRI	*Pulse Repetition Interval*
GPS	*Global Positioning System*

1. Motivation and objectives

1.1. Motivation

Accurate real-time positioning of an object and its dynamic behaviour estimation has many applications in various fields of engineering. Research about this topic has been driven from fields such as aeronautics industry. When working with navigation and positioning systems in real time, we have to take into account the influence of various factors that difficult measurements, such as noise, signal strength, weather phenomena, topography... these fluctuations of measured variables can be minimized by compensating filters. Getting increasingly accurate systems and impair their fields of application, was what motivated me to look for a proposal to develop research on Kalman filter and sensor fusion.

1.2. Goals

Commercial available systems to determine object positioning have different degrees of precision, providing an estimated position with a margin of error of a few meters. This error is not acceptable for the running of our application. There are more elaborated and precisely trading systems, but their size and weight make them unsuitable to handle in practice. Therefore, we have implemented a customized positioning system, which provides the desired resolution, introduces improvements in positioning, it has reduced size and weight of the entire system to suit small-scale vehicles.

In our scope, we have a mobile platform, which is required to know exactly the position where it is in every moment. Within this project my intention has been to introduce improvements through elements that provide additional or redundant information in order to process it and improve the position.

Specifically, the positioning system presented in this thesis, uses phase difference measurement. The system consists of some antennas installed in known fixed positions (slaves) distributed in a geographic area and an antenna (master) located on the mobile platform of unknown position, which has to be located.

The aim of this project is to obtain accurate estimates of the master position and other navigation states. The method we will use is Kalman filtering, based on the fusion of information coming from the inertial sensors, dead reckoning and distance measurement of RF, which will be developed and tested with simulated data and data obtained from experimental tests.

2. Background - State of the art

This chapter is an introduction to different methods of estimating the position as well as various algorithms to get the distance.

In a positioning system the elements can be divided in two categories:

- Physical Elements: those devices that make possible the implementation (hardware). Among the physical elements we have the Anchor nodes (fixed position; also known as slave antennas) and User nodes (variable, unknown location; called master antenna).

- Schematic Elements: Procedures or additional tools necessary for the Physical Elements to work (software).

When we talk about an Anchor or a User node, we are making reference to a transmitter, a receiver or a combination of both. Anchor nodes are static and its locations are known; whereas User nodes are the targets whose positions are unknown and are moving inside the scene. The use of certain devices (transmitter, receiver or transceiver) depends on the implemented architecture and it is intimately bounded to the Schematic elements. In most of the cases you need more than one Anchor node, depending on how many dimension or coordinates the location it has to be specified.

The Schematic Elements are methods, procedures or any kind of reasoning applied to signals involved in the Positioning System. They include the information we extract from the signals and the processing that we apply to that information. Among the means used to extract information from the arriving signal we have *Angle Of Arrival* (AOA) or *Direction Finding* (DF) used mainly in direction-based systems, whereas

when finding the distance we use *Received Signal Strength* (RSS), carrier signal *Phase Of Arrival* (POA), *Time of Arrival* (TOA), *Time Difference Of Arrival* (TDOA), *Frequency Difference of Arrival* (FDOA) and *Doppler Difference* (DD) [1].

Depending on the chosen method, the algorithm that will ultimately deliver the location will be different.

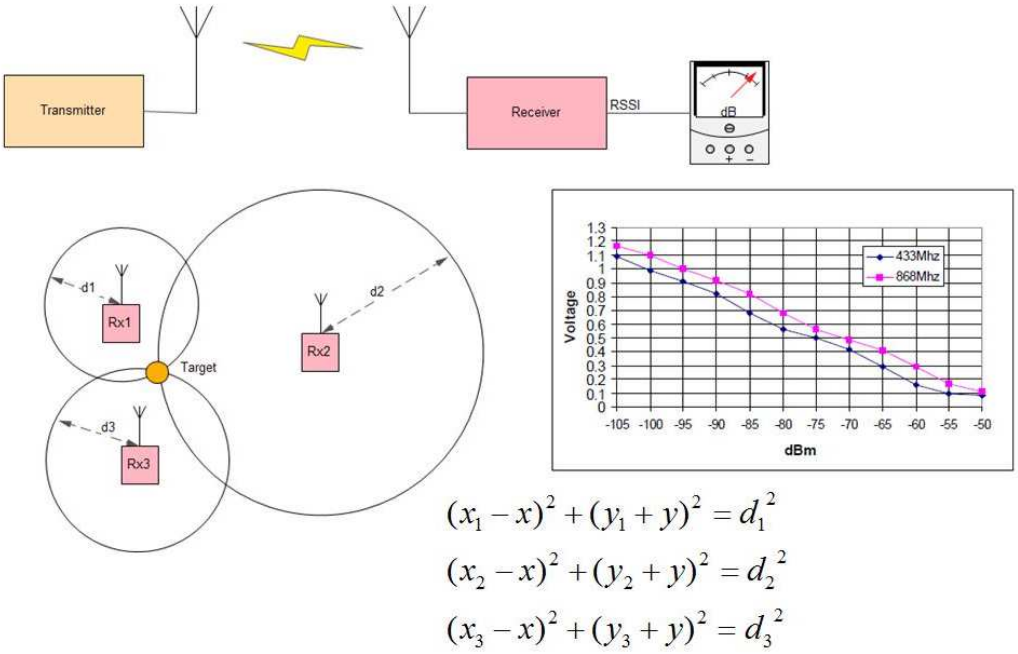
2.1. Location techniques

The parameters that have been most commonly used for positioning are the *Received Signal Strength* (RSS), *Angle of Arrival* (AOA), *Time of Arrival* (TOA) and a variant of the latter, *Time Difference of Arrival* (TDOA) [1][2].

2.1.1. RSS Estimation

The signal strength of the transmitting source is measured at several static receivers. The main idea is that, if the relation between distance and power loss is known, the received signal strength measured at a node can be used to estimate the distance between that node and the transmitting node, assuming that the transmit power is known.

Ideally, each measurement will generate a circle centered at the corresponding receiver with a radius that represents the distance between the object and that receiver. By intersecting several circles, the object position can be obtained. At least three receivers are required in two-dimensional positioning in order to resolve the intersection ambiguities. An example is shown in *Figure 1.*



$$(x_1 - x)^2 + (y_1 + y)^2 = d_1^2$$
$$(x_2 - x)^2 + (y_2 + y)^2 = d_2^2$$
$$(x_3 - x)^2 + (y_3 + y)^2 = d_3^2$$

Figure 1. Received Signal Strength system

2.1.2. Time of Arrival (TOA)

The TOA uses the time which wave travels from transmitter to receiver, also called *Time of Flight* (TOF), to measure the distance between both.

Since the propagation time of the signal is known, the measured time can provide a radius of a circle centered at the receiver. In two-dimensional localization, we need at least three circles to solve the intersection, which means three receivers. Moreover, synchronization between transmitter and each receiver is mandatory, either by having common clocks or changing timing information via certain protocols. An example is shown in *Figure 2.*

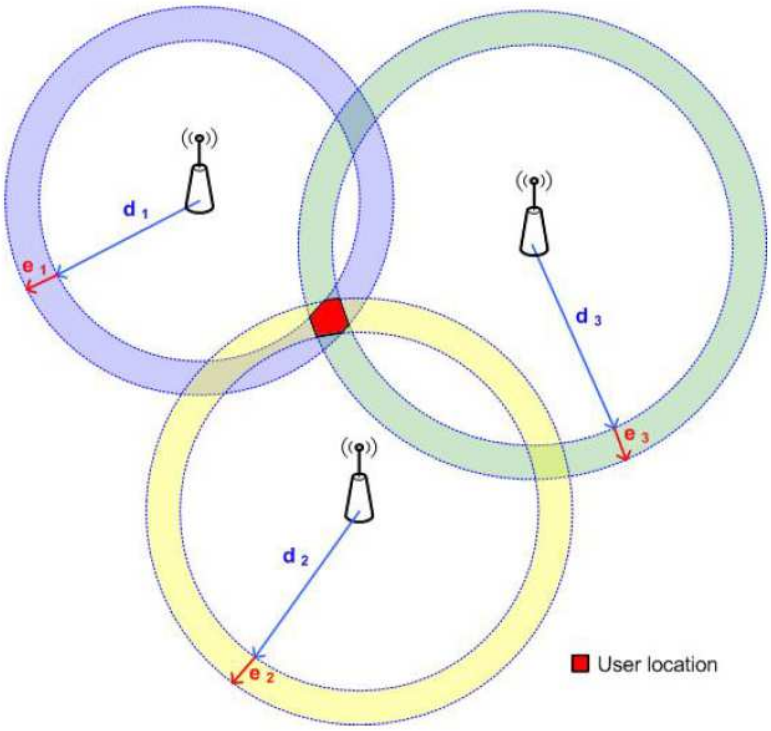


Figure 2. TOA example

2.1.3. Time Difference of Arrival (TDOA)

In this technique we measure the time difference at which the signal from the transmitter arrives at two different receivers. Given that the difference is measured between transmitters, this method only requires synchronization between receivers, which is easier to achieve compared to synchronization between transmitters and receivers.

Each time difference is converted into a hyperboloid with the constant distance between both receivers. The estimated position is the intersection of a number of hyperboloids. In two-dimensional positioning, at least two hyperboloid are required. This means that at least two pairs of receivers are required and this is achievable with three receivers. We can obtain a TDOA measurement by estimating relative TOA at each reference node and then computing difference between the two estimated TOA. An example is shown in *Figure 3.*

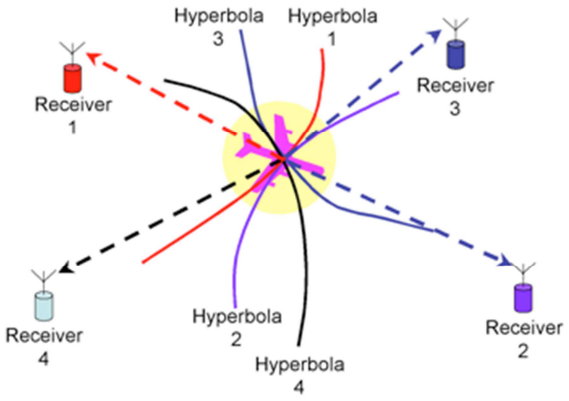


Figure 3. TDOA example

2.1.4. AOA Estimation

This estimation (also called *Direction Of Arrival* or *Direct Finding*) measures the angle of arrival of the signal sent by the object to be positioned at several receivers. The receivers obtain the estimation by directing the main lobe of a directional antenna or an adaptive antenna array.

Each measurement forms a radial line from the receiver to the object to be positioned. In two-dimensional positioning, the location of the object is defined at the intersection of two directional lines of bearing (*Figure 4*). In practice, more than two receivers may be employed to combat inaccuracies introduced by multipath propagation effects.

Moreover, this method has the advantage of not requiring synchronization of the receivers nor an accurate timing reference. However, due to temperature variations and antenna mismatches, this estimation requires multiple antennas or a beamforming antenna and their regular calibration.

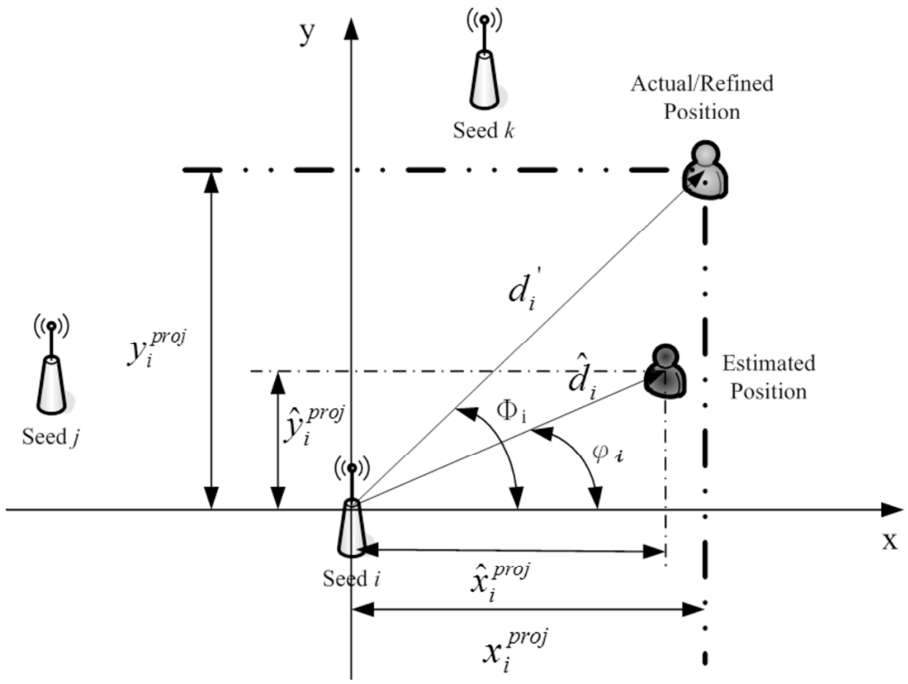


Figure 4. AOA scheme

2.2. Positioning methods

After the estimation is done, it is necessary to implement the correct algorithm to extract the distance information. Among the available algorithms we can find:

- Trilateration
- Multilateration

In the next chapter we will explain the trilateration and multilateration algorithms, the latter being used in our project, due to its capability to improve the system.

2.2.1. Trilateration Algorithm

Trilateration is the process of determining absolute or relative locations of points by measurement of distances, using the geometry of circles, spheres or triangles. Trilateration has practical applications in navigation, including Global Positioning Systems (GPS). In contrast to triangulation, it does not involve the measurement of angles.

In this algorithm, the position of the User node (in two dimension) is obtained from the TOA/RSS measurements to three anchor nodes. The problem is that, each measurement establishes a circle of possible solutions; when having three measurements, the ambiguity disappears, as we can see in *Figure 5* and the next equation. It is important to have in mind that if the solution is wanted in three dimensions, the required theoretical measurements are four.

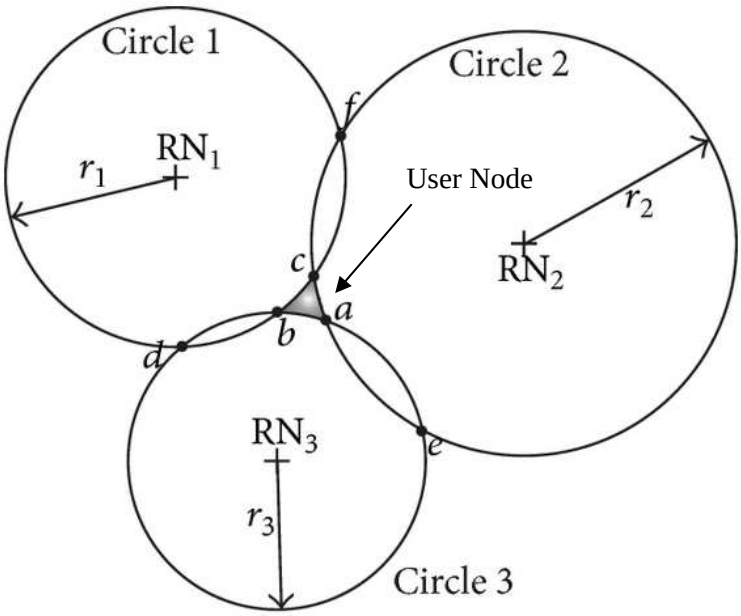


Figure 5. Trilateration. User node is located in the middle of the intersection of circles

$$r_i = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2}, \quad i = 1, 2, \ldots, m. \qquad (1)$$

2.2.2. Multilateration Algorithm

Multilateration (MLAT) is a common technique in radionavigation systems, based on the measurement of the difference in distance to two antenna stations at known locations that broadcast signals at known times. Unlike measurements of absolute distance or angle, measuring the difference in distance between two stations results in an infinite number of locations that satisfy the measurement. When these possible locations are plotted, they form a hyperbolic curve. To get the exact location along that curve, multilateration relies on multiple measurements: a second measurement taken

to a different pair of stations will produce a second curve, which intersects with the first. When the two curves are compared, a small number of possible locations are revealed.

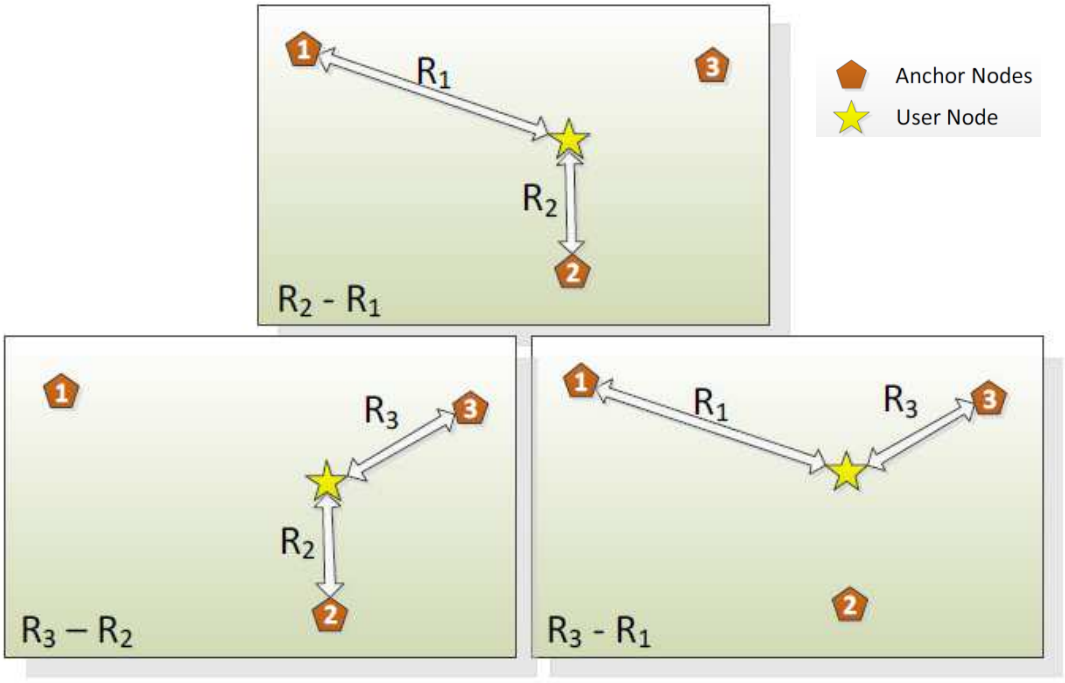


Figure 6. Multilateration

The most popular hyperbolic navigation system was LORAN-C, which was used around the world until the system was shut down in 2010. Other systems remain used nowadays, but use of satellite navigation systems has been imposed.

2.3. Examples of Positioning Systems

2.3.1. GPS

The GPS is a space-based navigation system that provides location and time information in all weather conditions, anywhere on the Earth, where there is an unobstructed line of sight to four or more GPS satellites. The system was originally developed for military purpose, but currently is being used for civil and commercial users around the world and it is freely accessible to anyone with a GPS receiver. The U.S. Department of Defense (DoD) developed the system, which originally used 24 satellites and it became fully operational in 1995.

The GPS concept is based on time. The satellites carry very stable atomic clocks that are synchronized to each other and to ground clocks. Any drift from true time maintained on the ground is corrected daily. Likewise, the satellite locations are monitored precisely. GPS satellites continuously transmit their current time and position. A GPS receiver monitors multiple satellites and solves equations to determine the exact position of the receiver and its deviation from true time. At a minimum, four satellites must be in view of the receiver for it to compute four unknown quantities (three position coordinates and clock deviation from satellite time).

2.3.2. LORAN-C

Loran-C is a hyperbolic radionavigation system which allows a receiver to determine its position by listening to low frequency radio signals transmitted by fixed land-based radio beacons; initially it had a big expense with the equipment needed to interpret the signals, which was improved over time with the use of newest electronic devices. The development of more accurate radionavigation systems, made that most of LORAN-C stations were shut down on 2010, although several other stations remain active.

The navigational method provided by LORAN is based on measuring the time difference between the reception of signals from a pair of radio transmitters. A given constant time difference between the signals from the two stations can be represented by a hyperbolic line of position. If the positions of the two synchronized stations are known, then the position of the receiver can be determined as being somewhere on a particular hyperbolic curve where the time difference between the received signals is constant. In ideal conditions, this is proportionally equivalent to the difference of the distances from the receiver to each of the two stations. So a LORAN receiver which only receives two LORAN stations cannot fully fix its position; therefore the receiver must receive and calculate the time difference between a second pair of stations. This allows a second hyperbolic line to be calculated, on which the receiver is located. Where these two lines cross is the location of the receiver, as shown before in *Figure 6*.

2.3.3. CHIRP-BASED SYSTEM

Chirp signals have been extensively used in radar and sonar systems to determine distance, velocity and angular position of objects and in wireless communications as a spread spectrum technique to provide robustness and high processing gain. Recently,

several standards have adopted chirp spread spectrum (CSS) as a low-power and low-complexity real-time localization.

A chirp is a signal in which the frequency increases ('up-chirp') or decreases ('down-chirp') with time. It is able to retrieve the distance information from the frequency shift produced by the difference of paths which the signal travels when arriving to one or another Anchor Node. The method is based in a Multilateration algorithm that yields the following relationship:

$$\Delta R = \frac{ck_r \cdot \overbrace{(t_i - t_1)}^{TDOA} \cdot T}{B_w}; i = 2,3 \qquad (2)$$

The advantages of this system are that it has completely avoided time synchronization by using a TDOA Estimation and the circuits are quite simple. Probabilities of detection of more than 75% were presented in experimental tests run in cables as well as over the air.

So, taking into account all the previous considerations, our positioning system should take the benefits of TDOA and Chirp system, we work in the middle ground. Using a chirp, we obtain a signal with a bandwidth wide enough to achieve a good resolution. With TDOA we don't need synchronization between the receiver and transmitter antennas, only transmitters must be synchronized with a central unit and simplifies the process.

3. System description

The complete configuration of the positioning system is described in this chapter: radiofrequency systems, INS, DR systems, navigation equations, and also reference frames and rotation matrices between them are presented. For more information about notation used in this chapter, go to Appendix A.

3.1. Radiofrequency system positioning

Radio positioning or radio location is understood as the exact placement of a certain individual or object with respect to a known reference point, using the radio electric

spectrum. This reference point is often assumed to be the center of the Earth coordinate system (the coordinate systems will be described in the chapter 3.4). In general, the reference point can be any point on the Earth that is known to the system and which coordinates can be related to. In some radio positioning systems it is not only important to obtain the position of an object in the space, but also obtain temporal evolution of the position of this object.

A radionavigation system is a special case of radio positioning which performs tracking functions in order to help users to move to a desired destination. In particular, a tracking system is a special case of radio positioning working under a precise time reference and on several occasions in real time. In those applications, it is important to know the time information in order to obtain the position of a certain object in an specific instant and also speed and acceleration. Thus, future positions of the device, as a function of time, can be predicted.

One of the most important applications of our positioning system is the measurement of distances. The implementation of a positioning system phase-synchronized is based on the measurement of the phase deviation associated with a transmission medium which is used to transmit a local oscillator signal from a central point to a remote location. It is possible to obtain the distance from the central point to the remote location using this phase offset if two parameters are well known: the exact frequency of the carrier and the speed of propagation of that wave in the medium. The measured phase offset depends on the speed of propagation of the electromagnetic wave and has an ambiguity given by the carrier wavelength.Thereby, if the distance *d*, to be measured, is longer than one wavelength λ, then the phase measure will contain a 2π ambiguity:

$$\phi_{\text{measured}} = \phi_{\text{medium}} \cdot (\text{mod}\ 2\pi) \qquad (3)$$

where ϕ_{medium} is the total phase offset introduced by the transmission medium. This ambiguity cannot be resolved with a single value of the carrier wavelength. Then the distance will be equal to:

$$d = \left(\frac{\phi_{\text{measured}}}{2\pi} + N\right) \cdot \lambda \quad [m] \qquad (4)$$

being *N* an unknown positive integer, that reflects the number of times we have passed through the ambiguity range [6][7][8].

The measurement may be profitable if a previous approximated value of the distance is known. Therefore it will be possible to measure the differential changes over a known value with a resolution much more precise than one wavelength. In the synchronization system the phase offset of the medium is measured and compensated at the central point in order to maintain a constant phase (for changes in the distance) in the remote location. Thus, the resolution of the differential measurement is given by the phase error of the compensated loop. Hence, the standard deviation of the phase error of the compensated system, $\sigma_{\phi\mathrm{error}}$, will determine the standard deviation of the differential longitude measurement σ_{d}, with the next equation:

$$\sigma_{\mathrm{d}} = \left(\frac{\sigma_{\phi\mathrm{error}}}{2\pi}\right) \cdot \lambda \quad [\mathrm{m}] \qquad (5)$$

This value will be used in simulations to characterize the noise standard deviation of distance measurements.

There are other possible applications for phase synchronization in distance measurement in addition to the one-dimensional case. For example, it is possible to transmit microwave signals from two (or more) synchronized units to an additional common receiver in order to measure displacements in other axes. The transmitted signals can be modulated, for example, with complex frequency sweep (chirp signal) patterns in order to improve the characteristics of the signal acquired by the common receiver. The common receiver will monitor the displacements of the remote unit if the central unit is attached to a fixed reference position. One example of this configuration is shown in *Figure 7*. Here it can be seen that the common receiver can monitor the displacements of the remote unit in the vertical axis, Y , making use of the reference signal sent from the central unit.

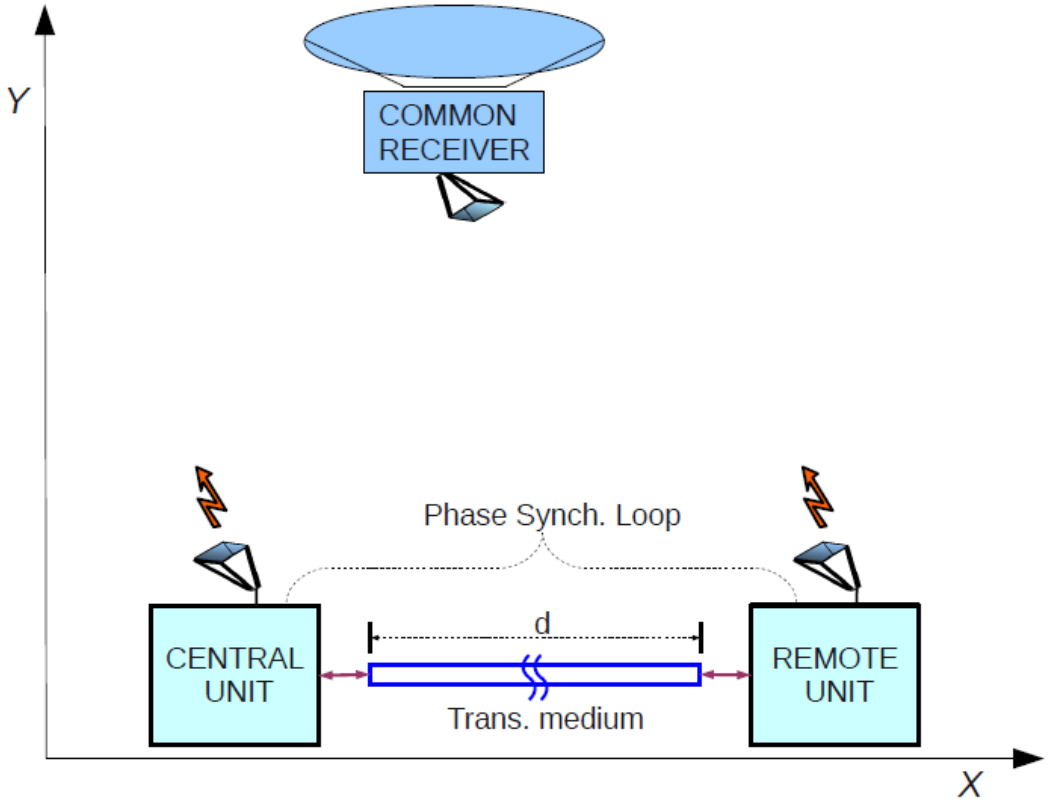


Figure 7. Simplified diagram of the high precision displacement measurement,

using a common receiver and two synchronized transmitters.The small displacements in the Y axis in the remote unit can be detected with high precision in the common receiver.

This configuration can be extended to a set of remote units synchronized in phase, using the central unit as a reference in a star topology. All the synchronized units of the star topology are attached to a fixed position then this set can be used to provide a precise location of the receiver. Hence a radio localization system can be implemented.

In our case, we set four units: one central unit and three remote units (from now on called slave antennas), disposed as a squared topology and they will be used in our simulation. In experimental case, we will dispose them on a triangular shape. In *Figure 21* the simulation scheme will be better detailed and in *Figure 33* we can see the real case.

The theoretical path that the signal has to cross is described in Diagram 1.

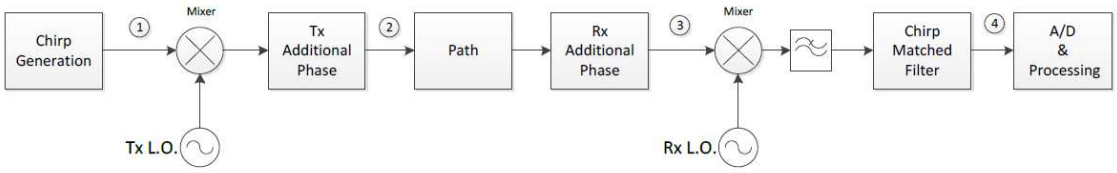


Diagram 1. Transmiter and Receiver

The transmitters' local oscillators have a frequency and an initial phase. The receiver is supposed to have the exact same frequency due to hardware mounting. It will also be assumed that is the same for all the transmitters because of the use of the phase synchronization system between them.

The signal will propagate a variable distance *r* to our mobile platform, with receiver antenna (master antenna). Every time it is transmitted, it will experience a different change of phase. Measuring this phase will add information that can also be translated to the space domain. In this case, the retrieved information describes small variations in the movement of the platform. The variations can be sampled as much as it is wanted by varying the pulse repetition frequency (PRF), so the time sampling frequency is low. The goal is not to allow a variation larger than half a wavelength in order to avoid ambiguities in the phase measurements.

Due to the fact that all the transmitters will broadcast the same chirp with the same parameters and with the same carrier, the only way to differentiate one from the other in the receiver is by making a time shift between transmissions, as seen on next *Figure 8.*

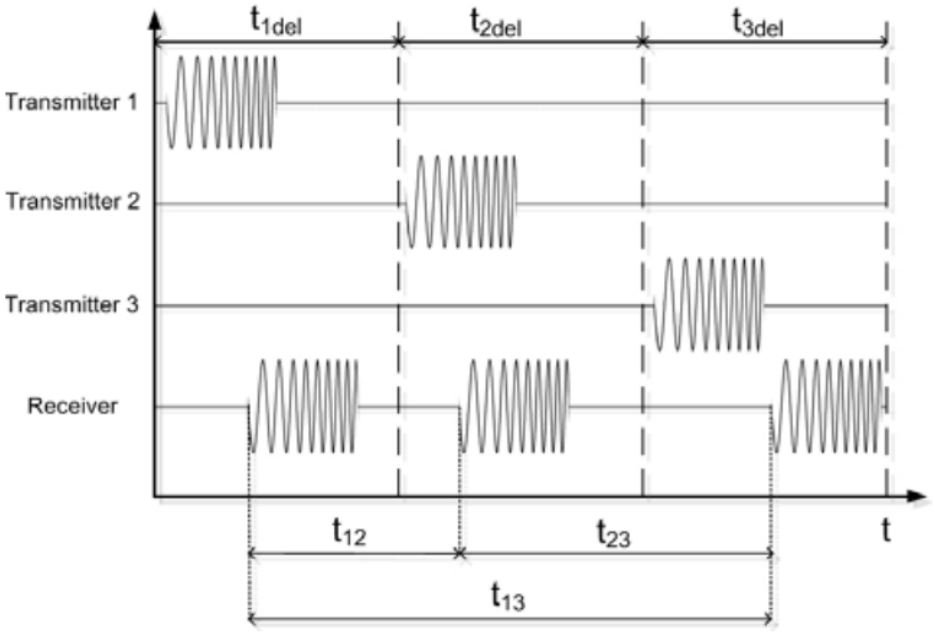


Figure 8. Illustration of the signals emitted and received in the location system

This separation cannot be arbitrary; it should be adjusted according to the Scene dimensions and the Pulse Repetition Interval (PRI). For instance, if the Scene included the UPC and its surroundings, the approximate diameter of extension would be 2.5 Km .Thus, the signal from one transmitter must be separated from the other one at least $\frac{2500}{c} seconds$, where *c* is the speed of light. For *c* = 299792458 m/s, the separation $d_{tx} = 8.33\ \mu s$. If this delay is not enough or a safer interval is required, the separation can be increased but it must be considered that there is an upper limit. Because there is a continuous emission of chirps, all the transmitted signals must fit inside one period of repetition. The equation that summarizes this situation, assuming a uniformly distributed emission of signals is:

$$\frac{\mathrm{D_{UPC}}}{\mathrm{c}} \leq \mathrm{d_{tx}} \leq \frac{\mathrm{PRI}}{\mathrm{N_{tx}}} \qquad (6)$$

Where $\mathrm{N_{tx}}$ is the number of Anchor transmitting nodes of the architecture, in the most simple case (two dimensions), would be three.

3.2. Inertial Navigation System (INS)

The *Inertial Navigation System* (INS) is divided into: Inertial Measurement Unit (IMU) and a computer part responsible for estimating states of navigation equations.

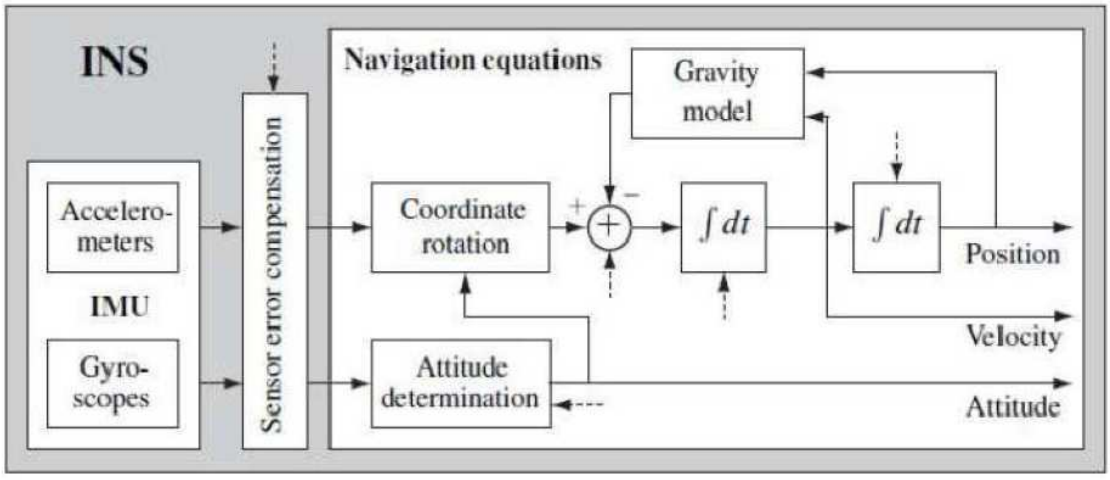


Figure 9. Diagram of INS

3.2.1. Inertial Measurement Unit (IMU)

An IMU is a device able to measure the acceleration force and angular velocity, consisting in an accelerometer and a gyroscope. To obtain angles through the IMU trigonometric calculations are required.

In this project we use InvenSense MPU-6050. It is an 6DOF IMU (6 Degrees Of Freedom) chip communicating through the I2C bus. This means that the sensor gives us the information taken from the accelerometer and the gyroscope, in 3 axes each of them (3 + 3 = 6DOF). There are a lot of microcontrollers that have as a peripheral the I2C bus, allowing us to communicate with the sensor and get data, like Arduino. It is a serial communication bus widely used in industry for communicating microcontrollers and peripherals.

The InvenSense MPU-6050 sensor contains a MEMS (Micro-Electro-Mechanical System) accelerometer and a MEMS gyroscope in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefore it captures the X, Y, and Z channel at the same time.

When comparing traditional inertial sensors with MEMS technology, the latter offers us cheaper and more accurate devices, making possible to construct low-cost global navigation satellite system aided INS for monitoring vehicle behavior.

Low cost sensors often produce dirty measurements, implying the need for digital signal post processing of the raw sensor data. An increasing amount of sensor information requires advanced information fusion. With Kalman filter, we can improve the dirty measurements proceding from the sensors, by integrating them with other informations and getting increasingly accurate measurements.

This IMU is widely used in drones, as it provides high precision. The MPU-6050 incorporates an accelerometer with a variable sensitivity of ± 2 g ± 4 g ± 8 g ± 16 g and the gyroscope $\pm 250\ ^{\circ}/_{s}\ \pm 500\ ^{\circ}/_{s}\ \pm 1000\ ^{\circ}/_{s}\ \pm 2000\ ^{\circ}/_{s}$

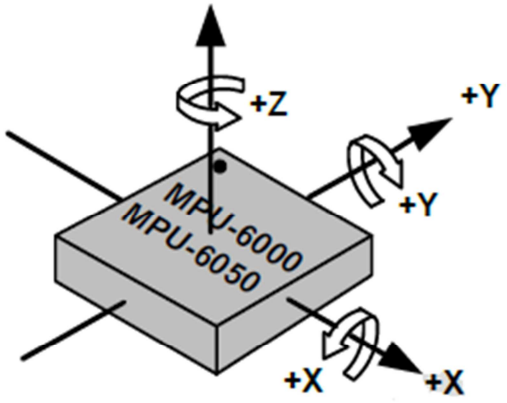


Figure 10. MPU-6050 axis

The sensor is oriented to the vehicle coordinates, so measurements are always related to the axes of the IMU (as shown in *Figure 10*). In aeronautical and aerospace engineering intrinsic rotations around IMU axes are often called Euler angles.

An aircraft in flight is free to rotate in three dimensions:

- Roll, rotation about an axis running from nose to tail.
- Pitch, nose up or down about an axis running from wing to wing

- Yaw, nose left or right about an axis running up and down

Normally, these axes are represented by the letters X, Y and Z and this is made in such a way that the X is used for the longitudinal axis. In *Figure 11* we see a representation of Roll, Pitch and Yaw.

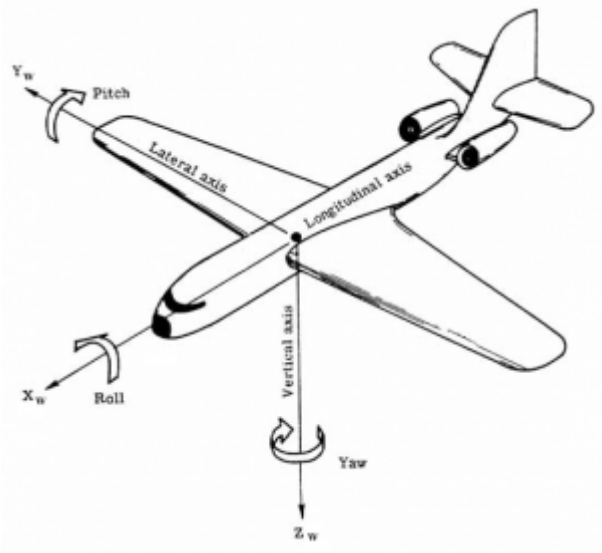


Figure 11. Coordinate system: Roll, Pitch and Yaw

3.2.1.1. Accelerometer

An accelerometer is a device that allows us to get accurate acceleration measures. It means that the acceleration is associated with the mass of the body or platform where it is contained, so it is related to its local inertial frame and accelerometer measures the acceleration relative to that frame. There are different devices measuring acceleration in a coordinate axis or more, according to the device we are using it contains an accelerometer when only measuring for an axis, or three, for three axis.

The accelerometer detects the acceleration of Earth's gravity, that is, an acceleration of approximately 9,8 m/s² perpendicular to the ground. The values that we obtain from the accelerometer are raw, so we have to apply them a division of 16384 [25] to convert measures to force unit *G,* knowing that 1*G*= 9.8 $^{m}/_{s^2}$

An accelerometer at rest relative to the Earth's surface will indicate approximately 1 *G* downwards, because at any point on the Earth's surface acceleration downwards is

noted to the local inertial frame. To obtain the acceleration due to motion with respect to the Earth, we need to compensate this gravity force, being subtracted and making corrections for effects caused by the Earth's rotation relative to the inertial frame.

Let's suppose that the IMU is perfectly aligned with the ground. Then, as shown in the *Figure 12*, the Z axis marks 9.8, and the other two axes mark 0. Now suppose the IMU is turned 90 degrees. Now X axis is perpendicular to the ground, thus X will set the acceleration of gravity.

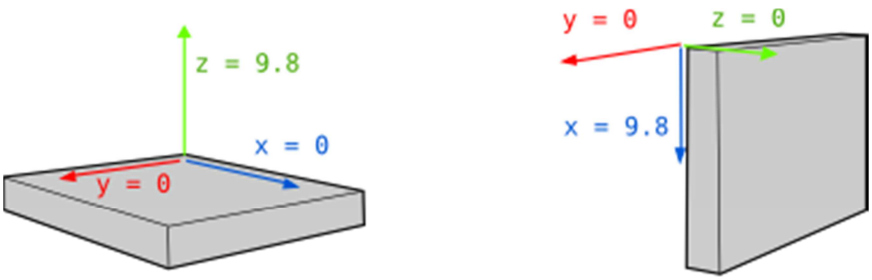


Figure 12. IMU moving exemple

With the information of acceleration in the three axis (X, Y, Z) we can get by trigonometric calculations the orientation in degrees of the platform, where it is attached. The calculation of the angle of our platform regarding to the center of the accelerometer is based on the direction cosines. In Apendix B we can find more information about them.

3.2.1.2. Gyroscope

The gyroscope is a device that allows us to get accurate angular velocity measures, expressed in $^{\circ}/_{s}$ (degrees/second). We can know the vehicle rotation angle on itself (Yaw) by measuring the angular velocity in the Z axis.

The rotation angle of the vehicle is faster or slower as time progresses. Travel angle *θ(t)* in a time interval *t* is calculated by the next formula:

$$\theta(t) = \omega_0 t + \frac{1}{2}\alpha t^2 \qquad (7)$$

Kinematics basic formula

$$Z\ angle = Z\ angle_{\ anterior} + w_z \cdot \Delta t + \frac{1}{2} \cdot a \cdot At^2 \quad (8)$$

Basic formula applied to IMU

Where Δt is the elapsed time whenever the formula is calculated and w_z is the angular velocity in the Z axis.

3.2.2. Error on the measures

The IMU measurements are seriously affected by several error sources, as they are subject to bias and drifts in measurements. Sensors can be sensitive to temperature, vibrations and other variables, so this must be taken into account when calculating the measurements. There is another trouble, noise, that is all interference affecting electronic devices. The accelerometer can measure any angle, but their readings are noisy and have a certain range of error.

In *Figure 13* we can see the signal collected by the accelerometer at rest. As shown, angle keeps about 88,5º but due to noise and non-linearity of the sensor we have these variations.

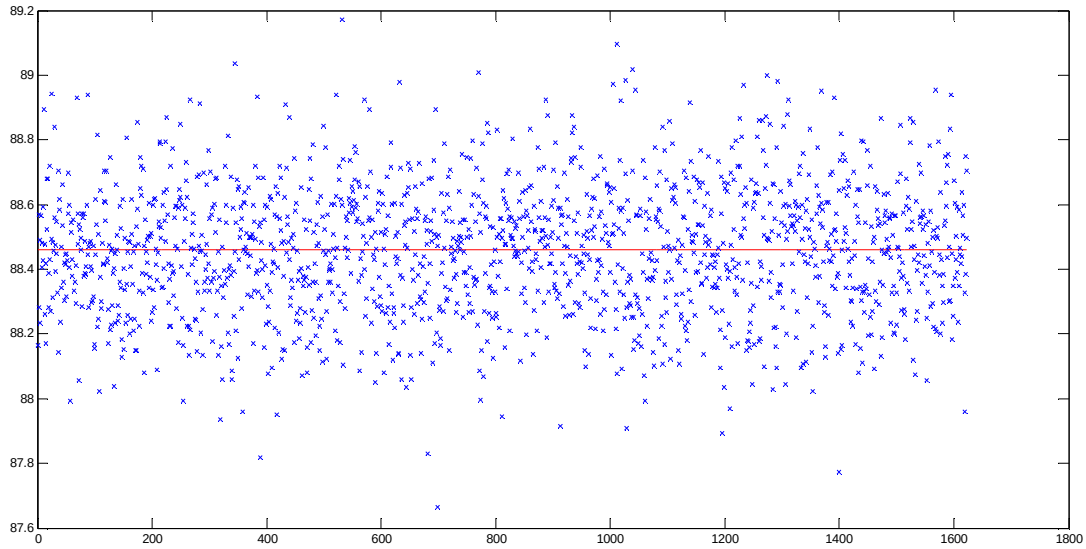


Figure 13. Real value of the accelerometer.

Red line represents real angle. Each blue x represents real measure from accelerometer

The accelerometer can also detect any acceleration other than gravity. Therefore, if we move the IMU without twisting, this acceleration will be detected by the sensor.

Unlike the accelerometer, the gyroscope gives very accurately measures. But calculating the angular rate is unavoidable that a small error occurs. With time, error eventually accumulates and real measurements may differ from estimated measurements (drift).

There are several ways to combine data from the accelerometer and gyroscope so we can obtain accurate measurements. Different algorithms are used to minimize sources of error (noise and drift), as the complementary filter, particle filter, Kalman filter, etc. One of the best is Kalman filter, used in aircraft, rockets and satellites. The Kalman filter is able to calculate the error of each measure from the previous measures, compensate it and give the real value of the measurements.

Error	Accelerometer	Gyroscope
Bias	80 mg	0.05 º/s
No linearity	0.5%	0.2 %

Table 1. Errors from MPU-6050

3.3. Dead reckoning

The dead reckoning is a mathematical method using simple trigonometric formulas to estimate the current location of a vehicle by calculations based on the direction and speed of navigation over a period of time.

The vast majority of land mobile robots used today, have dead reckoning technology as the main method of its navigation strategy. They need to remove the accumulated errors at continuous settings with various aid navigation systems.

The simplest implementation technique of dead reckoning is called odometry. This term implies that the movement of a vehicle along the path is derived directly from an odometer or tachometer on board. An example of common technology to implement odometry is optical encoder coupled directly to motors or wheel axes.

There are many sensors to quantify angular displacement and speed, such as:

- Potentiometers.
- Optical encoders.
- Magnetic encoders.
- Inductive encoders.
- Capacity encoders.

By knowing wheels' angle of rotation $\phi_R\ (t)$, their radius R_R and the sampling time *T* (see *Figure 14*), the speed v can easily be calculated, as follows:

$$v = \frac{S_R(t)}{T} = \frac{R_R \phi_R(t)}{T} \qquad (9)$$

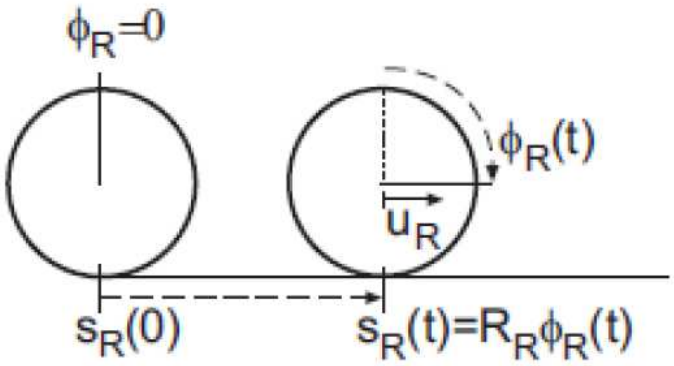


Figure 14. Encoder's principle of operation

$S_R(t)$ is the distance travelled in a sampling time. Speed is the result of dividing the distance by the sampling time.

The changes of orientation are calculated from the differences in speeds of each motor, measured by the two encoders on the two wheels. We obtain the angle from the following formula:

$$\theta_k = \theta_{k-1} + \Delta\theta \qquad (10)$$

Knowing the angle, speed of each of the components in the coordinate system of the platform can be calculated:

$$v^p = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \end{bmatrix} \quad (11)$$

Dead reckoning system has to implement the following navigation equation to estimate the position as a stand-alone system:

$$x_{k+1}^p = x_k^p + dt_k \cdot \tilde{v}^p \quad (12)$$

Where dt_k is the time step, x_k^p is the previous position of the vehicle and $\tilde{v}^p$ the estimated speed of the vehicle. For more information go to Appendices E and F

3.4. Frames and Rotations

There are many coordinate systems, as we are disposing different sensors and various reference frames, we have to select the most appropriate. Each sensor provides measurements with respect to a given reference frame. In this project, distance measures are provided by RF system in a local coordinate system. Accelerometer and gyroscope provide inertial measurements expressed relative to their instrument axes or body frame; this aspect will be explained in the next section and further information about reference frames can be read in [12] and Appendix C. Given that different sensors provide measurements relative to different frames, the measurements in different frames are only comparable if there are convenient means to transfer the measurements between the coordinate systems.

As an example, consider the vehicle and Earth reference frames. Sensors onboard the vehicle may resolve their data in the vehicle reference frame; however, the user of the vehicle is interested in knowing the vehicle state (i.e., position and velocity) in the Earth reference frame, as seen in *Figure 15*. The vehicle frame is attached to and rotates with the vehicle. The Earth frame is attached to and rotates with the Earth. Because the vehicle is free to translate and rotate with respect to the Earth, the vehicle and Earth frames can arbitrarily translate and rotate with respect to each other. Therefore, the

navigation system must maintain accurate estimates of the quantities necessary to allow transformation of variables between these reference frames.

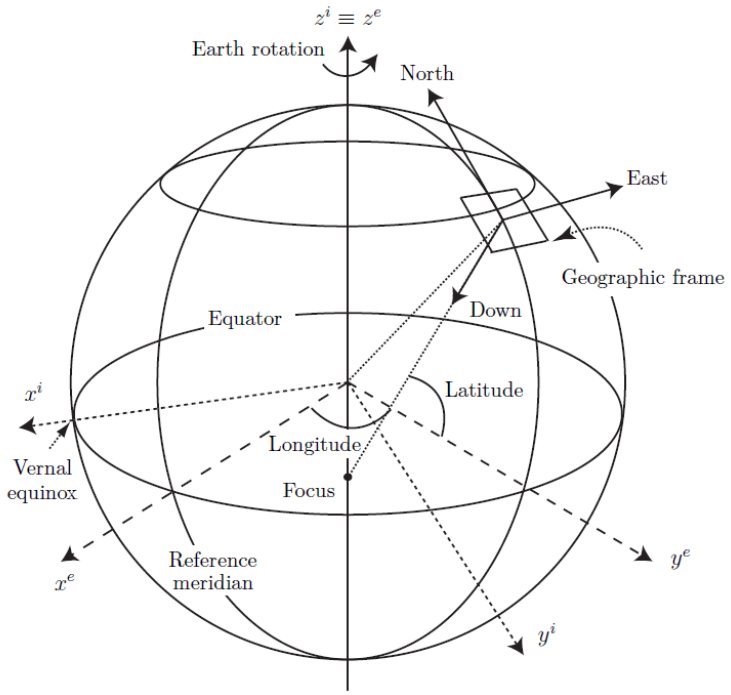


Figure 15. Illustration of the Earth centered Earth frame

3.4.1. Reference frame definitions

This section defines various frames ofreference that are commonly used in navigation system applications.

3.4.1.1. Inertial Frame

An inertial frame is a reference frame in which Newton's motion laws apply. The origin of the inertial coordinate system is arbitrary, and the coordinate axis may point in any three perpendicular directions. All inertial sensors produce measurements relative to an inertial frame.

3.4.1.2. Body Frame

In navigation applications, the objective is to determine the position and velocity of a vehicle based on measurements from various sensors attached to a platform on the

vehicle. This determines the definition of vehicle and instrument frames of reference and their associated coordinate systems. The body frame is rigidly attached to the vehicle of interest, usually at a fixed point such as the center of gravity. Picking the center of gravity as the location of the body frame origin simplifies the derivation of the kinematic equations and is usually convenient for control system design.

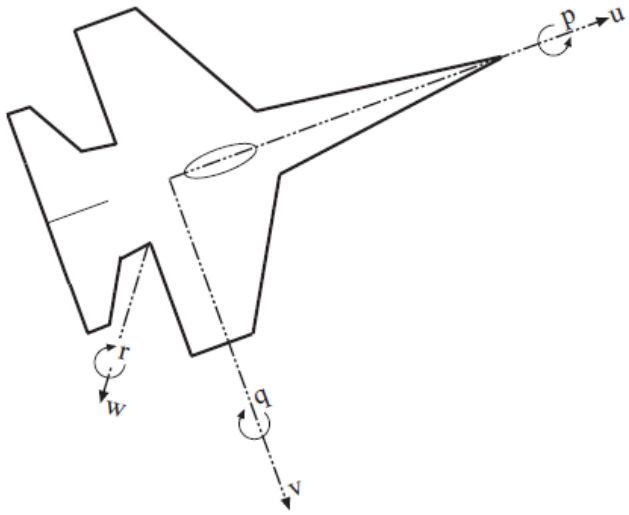


Figure 16. Top view of vehicle (body) coordinate system

3.4.1.3. Platform Frame

The sensor platform is rigidly attached to the vehicle, for various reasons the origin of the platform frame may be offset or rotated with respect to the origin of the body frame. The origin of the platform coordinate frame is at an arbitrary point on the platform.

3.4.1.4. Instrument Frames

Typically an instrument frame of reference is defined by markings on the case of the instrument. Sensors within the instrument resolve their measurements along the sensitive axes of the instruments. Ideally, the instrument sensitive axes align with the instrument frame of reference; however, perfect alignment will not be possible under realistic conditions. In addition, the sensitive axes may not be orthogonal. Instrument manufacturers may put considerable effort in factory calibration and orthogonalization routines in addition to temperature, linearity, and scale factor compensation.

Such manufacturer defined compensation algorithms that will be factory preset and programmed into the instrument. Depending on the desired level of performance, the navigation system designed may have to consider additional instrument calibration either at system initialization or during operation.

3.4.2. Rotations

The angular orientation of the mobile platform (master antenna) with respect to the navigational frame is expressed using the Euler angles: φ, θ, ψ (roll, pitch, and yaw, respectively). For more information about this topic see Apendices A, B and C.

The local frame of reference (or navigation frame) of this application is the surface in which positioning has to be performed. So it can be considered a two dimension application as the position will be given in X and Y components. But the model can easily be adapted for three dimension applications, only by adding the Z component corresponding to the perpendicular axes to the surface pointing up.

Inertial measurements are given in the instrument frame of reference, as sensors resolve their measurements along the sensitive axes of the instruments, the MPU-6050 in this case. The IMU and the master antenna (User Node) are installed on the platform frame. For simplicity, a perfect alignment with the instrument frame will be supposed, so measurements will be given in the body frame.

Since the only misalignment can be between the body frame of reference and the local frame in which the applications works, a rotation must be used so that the measurements in body coordinates can be applied to resolve the navigation states in the local system of coordinates.

The transformation is a simplification of the general three dimension rotation matrix:

$$R_l^b = \begin{bmatrix} \cos\theta\cos\psi & -\cos\theta\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\theta\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (13)$$

This transformation involves the ϕ, θ, ψ (roll, pitch and yaw) angles, corresponding to counterclockwise rotations in the X, Y and Z axes, respectively. As the application

works in two dimensions, in X-Y plane, roll and pitch angles are supposed to be zero, and the orientation of the sensor with respect to the local frame is indicated by the yaw angle (obtainable using the navigation equations and the angular rate in the Z component of IMU).

With these considerations, the rotation matrix is simplified as follows:

$$R_l^p = T_l^p = \begin{vmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (14)$$

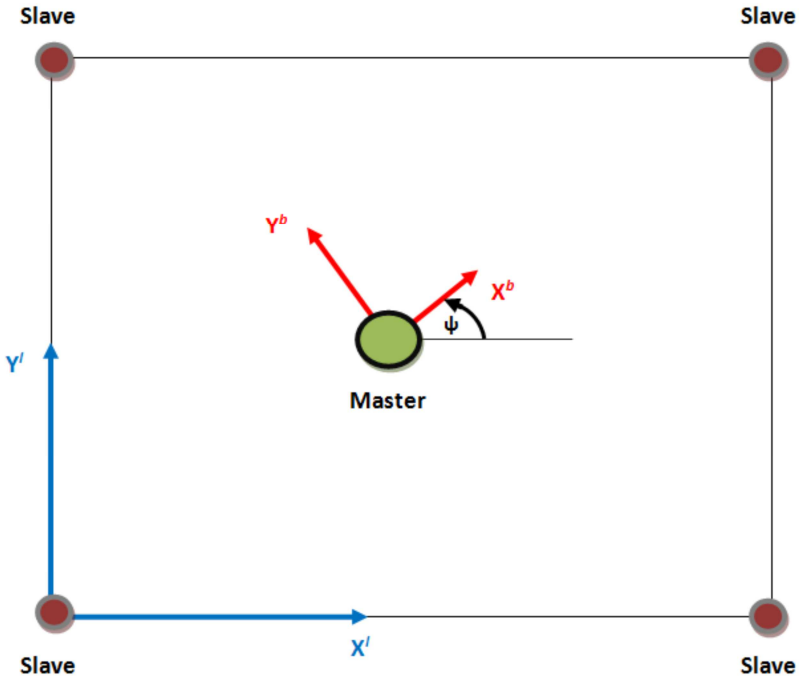


Figure 17. Diagram of the coordinate system.

Platform coordinate system (red) and global coordinate system (blue)

T_l^p is the rotation matrix that goes from the local coordinate system (the coordinate system origin carries the subscript *l*) to the coordinate system of the platform (the end coordinate system carries the index p). We can also obtain the reverse case T_p^l inverting the matrix. Since the matrix is orthogonal, the inverse matrix is equal to the transposed matrix. This is the matrix we will use in our simulation:

$$T_p^l = \begin{vmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (15)$$

3.5. Navigation Equations

Navigation equations must be used to estimate position, velocity, orientation, etc. They take into account the values of the previous states and IMU measurements related to time.

An INS is able to estimate, as a stand-alone system, the master's navigation states by using these navigation equations. But integration implies that errors grow with time, so the error will be unbounded. To see more information about derivation of the navigation equations go to references [9] [10]. A simpler version will be considered here, as shown in *Figure 18*.

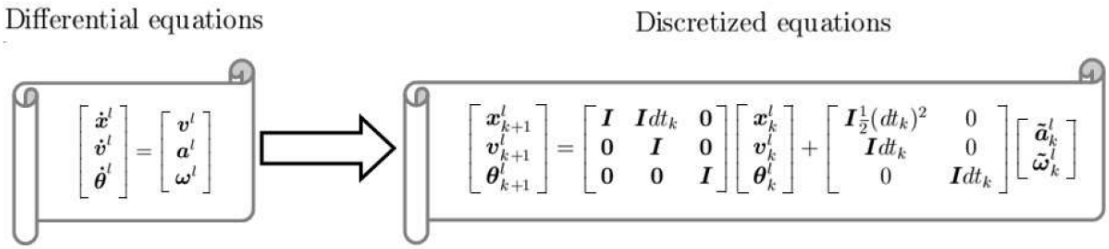


Figure 18. Navigation equations in differential and discretized mode

In the equations, master's navigation states are designated as x for the position, v for the velocity and θ as the orientation (angle). The acceleration is written as $\tilde{a}$ and the angular rate as $\tilde{\omega}$ because this measurements are in the inertial frame.

The superscript l indicates that the value is in the local frame. The subscript k reports that the value is from the current time step, and the subscript $k+1$ is for values of the next time step. The time difference between these two steps is dt_k

Navigation equations regarding to position:

$$x_{k+1} = x_k + v_{xk} \cdot dt_k + \frac{1}{2} \cdot dt_k \cdot \tilde{a}_{xk}^l \quad (16)$$

$$x_{k+1} = y_k + v_{yk} \cdot dt_k + \frac{1}{2} \cdot dt_k \cdot \tilde{a}_{yk}^{l} \quad (17)$$

Navigation equations regarding to velocity:

$$v_{k+1} = v_{xk} \cdot dt_k + dt_k \cdot \tilde{a}_{xk}^{l} \quad (18)$$

$$v_{k+1} = v_{yk} \cdot dt_k + dt_k \cdot \tilde{a}_{yk}^{l} \quad (19)$$

Navigation equations regarding to angle:

$$\theta_{k+1} = \theta_k + dt_k \cdot \tilde{\omega}_k^{l} \quad (20)$$

3.6. Sensor data fusion

The objective of sensor fusion is to obtain more accurate information than is present in any individual sensor source by combining information from different sources, enhancing the reliability of the system; the information fusion tries to reduce ambiguities in the measures and we get more robustness against spurious information from whatever of the sources.

The main use of the information fusion in this project is in the field of navigation, to produce estimates of the vehicle's position, velocity, etc. from the sensor measurements.

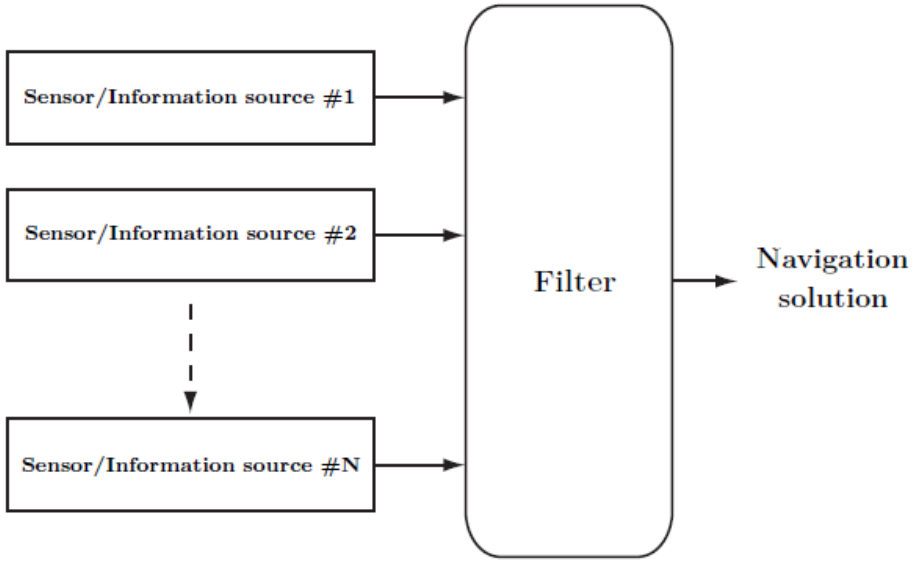


Figure 19. Filtering process.

There are many filters we can use to fusion the information coming from sensors, as seen in *Figure 19.* The most widely used nonlinear filtering approach is the extended Kalman filter (EKF). The idea behind the EKF is to linearize the navigation and observation equations around the current navigation solution and turn the nonlinear filtering problem into a linear problem. Assuming Gaussian distributed noise sources, the Minimum Mean Square Error (MMSE) solution to the linear problem is then provided by the Kalman filter.

For non-Gaussian distributed noise sources, the EKF provides the linear MMSE solution to the filtering problem. Unfortunately, the linearization in the EKF means that the original problem is transformed into an approximated problem which is solved optimally. This can seriously affect the accuracy of the obtained solution and the filter may lead to divergence of the system.

Therefore, in systems of nonlinear nature and non-Gaussian noise sources, more refined nonlinear filtering approaches such as Sigma-Point filters (Unscented Kalman filters), particle filters (sequential Monte Carlo methods) and exact recursive nonlinear filters, which keep the nonlinear structure of the problem and may significantly improve system performance. Due to its simplicity, the most widely used filter is EKF, and so it will be developed in this project.

The RF positioning system takes measurements of the three or more slaves from time to time, as seen before on *Figure 8*. These measurements are computed by the system, which takes a long time to process data, so we have to wait until new data are coming. This slows the process and the information comes late. The RF positioning system is unable to provide orientation information, it only gives the position.

IMU and dead reckoning are able to provide acceleration, angular rate and velocity information and this information is passed to the navigation equations. Sampling rate of measurements is high, so they can handle with high dynamics. Due to slight errors in the sensors and imperfect sensor calibrations, after integrating the position solution, the calculated position after some time of measurement position away from the sensors' actual position.

The fusion of these two systems compensates deficiencies of both and produces a better performance than systems working individually [13] [14]. In *Figure 20* you can see an outline of information fusion, that is, to take the strengths and weaknesses of both systems, put them together and make them stronger.

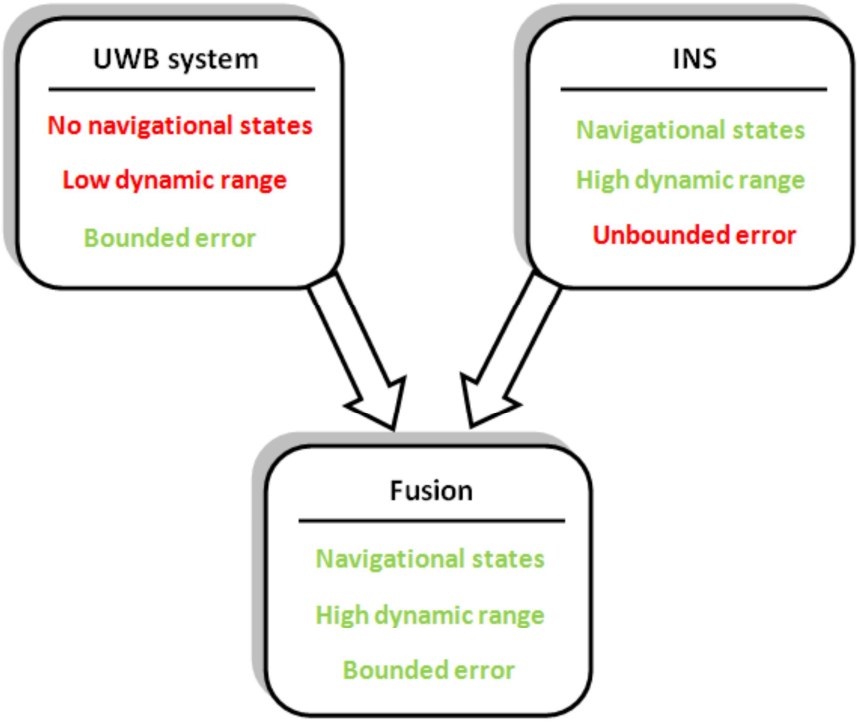


Figure 20. Summary of properties in the three systems

4. Introduction to Kalman filter

The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector estimator that is optimal in the sense that it minimizes the estimated error covariance, when some presumed conditions are given. Since the time of its introduction, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. This is likely due in large part to advances in digital computing that made the use of the filter practical, but also to the relative simplicity and robust nature of the filter itself. Rarely do the conditions necessary for optimality actually exist, and yet the filter apparently works well for many applications. The Kalman filter has been used extensively for tracking in interactive computer graphics.

The theory of the discrete Kalman filter is based on a process model and the measurement equation. The process model expresses the physical characteristics of the system. It predicts how the system status changes from one instant to the next. The state transition equation is:

$$X_{k+1} = A \cdot X_k + W_k \qquad (21)$$

Where X_{k+1} and X_n are state vectors expressing the system state in k and $k + 1$ times, respectively. A is the state transition matrix expressing the physical equations that govern the system state transitions. W_k is the process noise vector. The noise sources have known covariance and assume the system inaccuracies. The measurement equation of the process is as follows:

$$Z_{k+1} = H_k \cdot X_k + V_k \qquad (22)$$

Where H_k is the observation matrix and V_k represents the noise of measurements. This equation expresses how measured data are related to the system status. From a set of measures, this equation defines what should be the expected state of the system. In the next section each of the terms are explained separately.

4.1. Kalman filter

The Kalman filter, also known as Linear Quadratic Estimation (LQE), is an algorithm that uses series of measurements observed over time, containing noise (random

variations) and other inaccuracies and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. [15] [16] [17].

The algorithm works in two steps. In the prediction step, Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement is observed, these estimates are updated using a weighted average, with more weight being given to more certain estimates. Because of the algorithm's recursive nature, it can run in real time using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required.

4.1.1. Algorithm and variables

Kalman filter estimates a process using feedback control. First, the filter estimates the process state at a given time, each time *k*, and then obtains feedback in the form of noisy measurements.

Taking this into account, the equations can be divided into two groups:

- Time update equations
- Measurement update equations

To inicialize the filter, some initial parameters have to be given. So initial estimates for $\hat{x}_{k-1}^{-}$ and P_{k-1}^{-} we can tune. The first phase of the filter will be the Time Update ("Predict")

- Project the state ahead

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (23)$$

- Project the error covariance ahead

$$P_k^- = AP_{k-1}A^T + Q \quad (24)$$

Time update equations project forward in time the state and error covariance estimates to obtain the *a priori* estimates for the next time step.

After that, measurement update equations introduce feedback, specifically they use a new observed measurement to obtain the *a posteriori* estimate, correcting the previous information.

In time update phase, the first equation calculates the previous estimate $\hat{x}_k^-$ with the matrix A, where the state equations are, the control matrix B and the vector u_{k-1} . The second equation calculates the prediction covariance.

The second phase would be the Measurement Update ("Correct") that comprises the next steps:

- Compute the Kalman gain

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (25)$$

- Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \quad (26)$$

- Update the error covariance

$$P_k = (I - K_k H) P_k^- \quad (27)$$

In measurement update phase, the first thing to do is calculate the Kalman gain K_k. Then it is necessary to measure the process to obtain z_k. With z_k measurement and estimation of measurement $H\hat{x}_k^-$ innovation is obtained. Innovation refers to comparation between reality and prediction $(z_k - H\hat{x}_k^-)$.

Now, using innovation and Kalman gain, it is possible to generate an estimate of *a posteriori* state, as shown in the second equation for updating.

The last equation generates an estimated *a posteriori* error covariance.

After the time and measurement update (or prediction and correction phase) the procedure is repeated from time to time, projecting previous estimates *a posteriori* to obtain the new *a priori* estimates.

A brief description of Kalman variables:

- Q: process noise covariance.
- R: noise covariance measures

Both two are assumed to be constant at each iteration in this project. Normally Q and R change with each iteration k.

- A matrix: the state transition matrix expressing the physical equations that govern the system.
- B matrix: control input model. Relates the control input to the state.
- u_k vector: The control vector that indicates the magnitude of the control input.
- H matrix: is the observation model, indicating the relationship between measurements and the state vector at time k.
- P_k^- matrix: covariance prediction. Measure of the estimated accuracy of the state estimate
- z_k vector: measurement vector. It contains the real measurement we received in this time step.
- $\hat{x}_k^-$: The previous estimated state at step k, given prior knowledge of the process to step k.
- $\hat{x}_k$: *a posteriori* estimated state in step k given the z_k measurement.

4.1.2. Process and estimations

The Kalman filter is used to estimate the state of linear dynamic systems governed by a model as it follows:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (28)$$

The estimation also needs measures, as in the following equation:

$$z_k = Hx_k + v_k \quad (29)$$

The random variable w_k represents the noise process, which is the error between the process characterization and the actual process. Another random variable v_k represents the noise measures, which can be considered as the error between the estimated measure (Hx_k) and the actual measure.

We assume that:

1. v_k and w_k are white processes:

$$E\left[v_k v_l^H\right] = 0 \ \ for\ k \neq l \quad (30)$$

$$E\left[w_k w_l^H\right] = 0 \ \ for\ k \neq l \quad (31)$$

2. v_k and w_k have zero mean, they are Gaussian processes with known covariance:

Mean

$$\mu_k = E[v_k] = E[w_k] = 0 \quad (32)$$

Covariance

$$R_k = E\left[v_k v_l^H\right] = R_k \delta \quad (33)$$

$$Q_k = E\left[w_k w_l^H\right] = Q_k \delta \quad (34)$$

3. v_k and w_k are independent processes:

$$E\left[v_k w_l^H\right] = E[v_k] E\left[w_l^H\right] = 0 \quad (35)$$

When deriving the Kalman filter equations, the first step is to find an equation that calculates the *a posteriori* estimate $\hat{x}_k$ as a linear combination of the *a priori* estimate

$\hat{x}_k^-$ and a weighted difference between the current measurement z_k and measures prediction $H\hat{x}_k^-$ calculated using the *a priori* state estimate.

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad (36)$$

As explained above, the difference $(z_k - H\hat{x}_k^-)$ is called innovation and measures the discrepancies between predicted measurement and observed measurement. If innovation is zero, they totally agree.

The matrix K is the Kalman gain that minimizes the *a posteriori* error covariance P_k. Details of the calculation are presented in [18]. The general expression for K in the current time step k is:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (37)$$

Observing this expression we may obtain information about the behaviour of Kalman filter. When the *a priori* error covariance P_k^- approaches zero, the Kalman gain also tends to zero, so the innovation is not trusted. If P_k^- tends to zero is because the *a priori* state estimate is very similar to the real value of the state and then it is better to trust in the estimation rather than measurement. On the other hand, if the measurement noise covariance R approaches zero, it means that measurements have little error, so it should be better to rely on measuring more than priori state estimate.

4.1.3. Filter parameters and tuning

Initial estimates for the state $\hat{x}_0$ and the estimated error covariance P_0 have to be provided for the filter's initialization. If the initial calculation model is wrong, we should use an initial value from a higher estimate of covariance error, as this indicates that the initial estimate of the state is wrong and is not trusted. In addition, a high value of P_0 allows a faster convergence of the filter. On the other hand, if we use a high value in P_0, estimates will not be smooth and error therein will be higher in the initial phase of the trajectory. So there is a middle ground in choosing the initial estimate error covariance.

Another parameter related to the convergence and smoothness of the estimates is the process noise covariance Q. It is difficult to determine its value because of the complexity of direct observation of the process that we estimate. So if knowledge of the process model is low, we have a high degree of uncertainty, which must choose a high Q value, in this case the z_k measures will be more reliable than the x_k. In contrast, if the process model is well known, a low value of Q allows the filter strength against outliers in measurements, but with a slow convergence due to milder changes in estimates.

In conclusion, the values of P_0 and Q have to be chosen taking into account the initial knowledge of the state and the process model, and will affect the speed of convergence, softness in estimating changes and robustness in front of faulty measurements.

The measurement noise covariance R can be measured before operating the filter. Since it is possible to measure the process, several sample values can be taken in order to determine the variance of the measurement noise. If values of R are high, you cannot rely on the measurements and the filter will continue the process model.

4.2. Extended Kalman Filter

The Extended Kalman Filter (EKF) is the nonlinear version of the Kalman filter, which linearizes about an estimate of the current mean and covariance. The state transition and observation models don't need to be linear functions of the state, but may instead be differentiable functions.

4.2.1. Process and estimation

As explained before, the Kalman filter estimates the state x of a discrete process governed by a linear stochastic differential equation. Very frequently, the process and/or the ratio of measurement for the process is not linear. When this happens, the solution is a linearized Kalman filter on an estimate of the current state: the extended Kalman filter (EKF).

As explained before on Kalman algorithm, in this situation equations can be divided into two groups:

- *Time update* equations
- *Measurement update* equations

To inicialize the filter, as seen before, some initial parameters have to be given. So initial estimates for $\hat{x}_{k-1}^{-}$ and P_{k-1}^{-} we can tune. The first phase of the filter will be the Time Update ("Predict"), constituted by the next processes:

- Project the state ahead

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (38)$$

- Project the error covariance ahead

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \quad (39)$$

The second phase would be the Measurement Update ("Correct") that comprises the next steps:

- Compute the Kalman gain

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + V_k R V_k^T \right)^{-1} \quad (40)$$

- Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \quad (41)$$

- Update the error covariance

$$P_k = (I - K_k H_k) P_k^- \quad (42)$$

The estimate is linearized around the current estimate using the partial derivatives of functions and process measurements.

Some previous definitions of the Kalman filter are now different. The state equation is a nonlinear function:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \quad (43)$$

That is related to the prior state to time step k - 1, an optional input u_{k-1} and the noise process w_{k-1}.

The z_k measures also have a nonlinear relationship with the state, as follows:

$$z_k = h(x_k, v_k) \quad (44)$$

The process and measurement noise stay as w_k and v_k, respectively. The non-linear function h is a function of actual state of x_k and v_k.

The state vector and measures have to be approximated without using the noise, because you can't know their values:

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (45)$$

$$\tilde{z}_k = h(\tilde{x}_k, 0) \quad (46)$$

The ~ sign indicates that it is an approximation and $\hat{x}_{k-1}$ estimates *a posteriori* the previous step. It is important to note that after the non-linear transformations the various random variables don't retain the normal distribution. Next, as we have to linearize the function, we use the Jacobian to do this, the Jacobian consists in partial derivates of functions. For more information about it, go to [19] [20] [21].

A is the Jacobian matrix of partial derivatives of f with respect to x, where each element has the following form. The Jacobian matrices A, W, H, and V change each time instant k.

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0) \quad (47)$$

W is the Jacobian matrix of partial derivatives of f with respect to w:

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0) \quad (48)$$

H is the Jacobian matrix of partial derivatives of f with respect to x:

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, 0) \quad (49)$$

V is the Jacobian matrix of partial derivatives of f with respect to v:

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{x}_k, 0) \quad (50)$$

4.3. Dynamic models and algorithms designed

4.3.1. Dynamic models

In applications such as positioning, tracking and navigation, a good model to describe the object motion is essential [12] [22].

There are two types of models for state estimation. The first one are kinematic models considering unknown inputs as process noise and the other one includes known inputs.

The name of the dynamic model depends on the navigation parameters that are included in the state vector. The reason to include a navigation parameter in the state vector is to get an estimation of it.

-If only position is included in the state, the name of the model is constant position (CP).
-If speed is also included, the name of the model is constant velocity (CV).

-If the state vector also contains the acceleration, the model name is constant acceleration (CA).

As stated before, movement may be considered in one dimension (X), two dimensions (X, Y) or three dimensions (X, Y, Z). This project is intended to work in only two dimensions.

The equations of position, velocity and acceleration can be expressed as a polynomial function of time. In *Table 2*, kinematic equations are represented.

	Derived Form	Integral Form
Position	$x(t) = x_o + v_o \cdot t + b \cdot \frac{t^2}{2} + c \cdot \frac{t^3}{6} + d \cdot \frac{t^4}{12}$	$x_o + \int_0^t v \cdot dt$
Velocity	$\frac{dx}{dt} = v(t) = v_o + b \cdot t + c \cdot \frac{t^2}{2} + d \cdot \frac{t^3}{3}$	$v_o + \int_0^t a \cdot dt$
Acceleration	$\frac{d^2x}{dt^2} = a(t) = b + c \cdot t + d \cdot t^2$	

Table 2. Kinematic equations

State vectors and dynamic equations to calculate the state at each time step are shown in *Table 3*. As stated before, the noise w_k can be interpreted as a process noise for pure kinematic model, where B is a control matrix and u_k is a control vector. We can use them as an input in case of detected movement. Thus, each dynamic model has two possibilities, taking into account whether a navigation parameter measured as an input or not. If speed measurement is taken, it can be used as an input to a model constant position, and then the measurement noise has the role of process noise.

On the other hand, it might not be of interest to use the speed measurement as an input (for example, if used in the measurement update of the Kalman filter) and is preferred to estimate its value. Then the dynamic model chosen has to be a constant speed. The same applies if a measure of acceleration is taken. It can be used as a constant speed model input or can be estimated as a model of constant acceleration.

Model name	State	Dynamic equation
Constant position	$x_k = \begin{bmatrix} x \\ y \end{bmatrix}$	$x_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} w_k$
Constant Speed	$x_k = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}$	$x_{k+1} = \begin{bmatrix} I_{2\times2} & TI_{2\times2} \\ 0_{2\times2} & I_{2\times2} \end{bmatrix} x_k + \begin{bmatrix} \frac{T^2}{2} I_{2\times2} \\ TI_{2\times2} \end{bmatrix} w_k$
Constant acceleration	$x_k = \begin{bmatrix} x \\ y \\ v_x \\ v_y \\ a_x \\ a_y \end{bmatrix}$	$x_{k+1} = \begin{bmatrix} I_{2\times2} & TI_{2\times2} & \frac{T^2}{2} I_{2\times2} \\ 0_{2\times2} & I_{2\times2} & TI_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & I_{2\times2} \end{bmatrix} x_k + \begin{bmatrix} \frac{T^3}{6} I_{2\times2} \\ \frac{T^2}{2} I_{2\times2} \\ TI_{2\times2} \end{bmatrix} w_k$

Table 3. Two dimension kinematic models.

$I_{2\times2}$ Is the diagonal matrix of two dimensions and $0_{2\times2}$ is a matrix of 2 × 2 of zeros.

4.4. Designed algorithms

This section shows different algorithms for our application. Algorithms will be divided into 2 sections:

- The first group uses the Extended Kalman Filter (EKF) to fuse the RF positioning system measurements with dead reckoning, which gives us the speed and direction of the master.

- The second group uses the Extended Kalman Filter (EKF) to fuse the RF positioning system measurements and INS, which provides a measure of acceleration and angular velocity.

4.5. Extended Kalman Filter with navigation system estimation

In this section, the fusion algorithm using the EKF, information coming from the RF positioning system and dead reckoning system are presented. With the speed and orientation measures coming from the DR system, dynamic models that can be used are CP and CV.

In the first case, the measurements of speed and direction are used as an input to the EKF. In the second case, the measures are introduced into the filter during the upgrade, so their functions are corrected on the *a priori* estimated state. In both algorithms, the RF positioning system is used in the correction phase.

4.5.1. Constant Position model (CP)

This algorithm uses the EKF algorithm and measurement range of RF positioning system, then the velocity and orientation are not needed. As it follows, the dynamic constant position model of the state vector only contains the position in two dimensions that has to be estimated.

$$x_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad (51)$$

The equation that characterizes the dynamic CP model, as shown in *Table 3*, is:

$$x_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} w_k \quad (52)$$

And the measurements are represented by the following expression:

$$z_k = h(\hat{x}_k^-) + v_k \quad (53)$$

There are two groups of equations used in EKF. First of all, *time update* equations, also known as prediction equations. The initial estimates, are obtained from:

$$\hat{x}_k^- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \hat{x}_{k-1} \quad (54)$$

And the *a priori* covariance error is:

$$P_k^- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} P_{k-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T + T^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} Q_k \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T \quad (55)$$

As the only used measures are distance measurements from RF system (distance from the master to four slaves), then $h(\hat{x}_k^-)$ has to calculate the distance from the previous position of the master $\hat{x}_k^- = \begin{bmatrix} \hat{x}_k^- \\ \hat{y}_k^- \end{bmatrix}$ to the known slaves, expressed as $(S_{x,n}, S_{y,n})$, for n slaves position. The Euclidean distance is a nonlinear function relating *a priori* estimation of the state with the measurement:

$$h(\hat{x}_k^-) = \begin{bmatrix} \sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2} \\ \vdots \\ \sqrt{\left(\hat{x}_k^- - S_{x,4}\right)^2 + \left(\hat{y}_k^- - S_{y,4}\right)^2} \end{bmatrix} \quad (56)$$

Now, the Jacobian H_k may arise directly as:

$$H_k = \frac{\partial\left(h(\hat{x}_k^-)\right)}{\partial x_k} = \begin{bmatrix} \frac{\left(\hat{x}_k^- - S_{x,1}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2}} & \frac{\left(\hat{y}_k^- - S_{y,1}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2}} \\ \vdots & \vdots \\ \frac{\left(\hat{x}_k^- - S_{x,4}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,4}\right)^2 + \left(\hat{y}_k^- - S_{y,4}\right)^2}} & \frac{\left(\hat{y}_k^- - S_{y,1}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2}} \end{bmatrix} \quad (57)$$

Finally, the three equations for the measurement update step are the same as those used in the EKF algorithm section.

$$K_k = P_k^- H_k^T \left[H_k P_k^- H_k^T + R_k\right]^{-1} \quad (58)$$

$$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - h(\hat{x}_k^-)\right) \quad (59)$$

$$P_k = [I_{2\times 2} - K_k H_k] P_k^- \quad (60)$$

R_k and Q_k noise matrices may change in each iteration, and its value will depend on the situation in which the application runs. They will have to be tuned to improve the performance of the algorithm and for greater precision in the results.

4.5.2. Constant Position as an input velocity model (CPCV)

This algorithm is very similar to that explained in the previous section (CP). It uses the EKF equations and measurements of RF positioning system, also speed and direction are provided by the dead reckoning system. Position is the only navigation state included in the state vector; therefore, the dynamic model used is CP.

$$x_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \tag{61}$$

Once the state vector has been defined and knowing that the characterization of the model is the same as in the algorithm CP, the time update equations are presented.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{62}$$

This is the equation that gives the preliminary estimation:

$$A = I_{2\times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{63}$$

$$B = TI_{2\times 2} = \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} \tag{64}$$

$$u_k = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \end{bmatrix} \tag{65}$$

The input vector u_k, incorporates the velocity v and direction θ measurements estimated *a priori* state. Due to this fact, the process noise covariance Q_k contains measurement noise dead reckoning system. Then the *a priori* covariance error is given by:

$$P_k^- = AP_{k-1}A^T + BQ_kB^T \tag{66}$$

Measurement estimation $h(\hat{x}_k^-)$ and the Jacobian H_k are the same as in the algorithm of the CP, since the RF distance measurements are the only used in the correction phase of EKF.

$$h(\hat{x}_k^-) = \begin{bmatrix} \sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2} \\ \vdots \\ \sqrt{\left(\hat{x}_k^- - S_{x,4}\right)^2 + \left(\hat{y}_k^- - S_{y,4}\right)^2} \end{bmatrix} \quad (67)$$

$$H_k = \frac{\partial\left(h(\hat{x}_k^-)\right)}{\partial x_k} = \begin{bmatrix} \frac{\left(\hat{x}_k^- - S_{x,1}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2}} & \frac{\left(\hat{y}_k^- - S_{y,1}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2}} \\ \vdots & \vdots \\ \frac{\left(\hat{x}_k^- - S_{x,4}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,4}\right)^2 + \left(\hat{y}_k^- - S_{y,4}\right)^2}} & \frac{\left(\hat{y}_k^- - S_{y,1}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2}} \end{bmatrix} \quad (68)$$

Finally, the measurement update equation is the same as in the general case on EKF algorithm.

4.5.3. Constant Velocity model (CV)

This is the first algorithm designed to estimate the speed and position. Thus, corresponding to the CV model, the state vector is:

$$x_k = \begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \end{bmatrix} \quad (69)$$

As seen in *Table 3*, the dynamic model is governed by the following equation:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (70)$$

Where:

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} I_{2\times2} & TI_{2\times2} \\ 0_{2\times2} & I_{2\times2} \end{bmatrix} \quad (71)$$

$$B = \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} = \begin{bmatrix} \frac{T^2}{2} I_{2\times2} \\ TI_{2\times2} \end{bmatrix} \quad (72)$$

Once we have the model of state, error covariance can be defined as:

$$\hat{x}_k^- = A\hat{x}_{k-1} \quad (73)$$

$$P_k^- = AP_{k-1}A^T + BQ_kB^T \quad (74)$$

Since this algorithm is not using the measurements as inputs, the matrix Q_k is the process noise covariance, $h(\hat{x}_k^-)$ is the measure function, that integrates the values of range (distance from master to the four slaves) and also measures velocity and orientation:

$$h(\hat{x}_k^-) = \begin{bmatrix} \sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2} \\ \sqrt{\left(\hat{x}_k^- - S_{x,2}\right)^2 + \left(\hat{y}_k^- - S_{y,2}\right)^2} \\ \sqrt{\left(\hat{x}_k^- - S_{x,3}\right)^2 + \left(\hat{y}_k^- - S_{y,3}\right)^2} \\ \sqrt{\left(\hat{x}_k^- - S_{x,4}\right)^2 + \left(\hat{y}_k^- - S_{y,4}\right)^2} \\ \sqrt{\left(\hat{v}_{x,k}^-\right)^2 + \left(\hat{v}_{y,k}^-\right)^2} \\ arctan\left(\frac{\hat{v}_{y,k}^-}{\hat{v}_{x,k}^-}\right) \end{bmatrix} \quad (75)$$

Then the Jacobian $H_k = \frac{\partial\left(h(\hat{x}_k^-)\right)}{\partial x_k}$ is the following expression:

$$\mathbf{H}_k = \begin{bmatrix} \frac{(\hat{x}_k^- - S_{x,1})}{\sqrt{(\hat{x}_k^- - S_{x,1})^2 + (\hat{y}_k^- - S_{y,1})^2}} & \frac{(\hat{y}_k^- - S_{y,1})}{\sqrt{(\hat{x}_k^- - S_{x,1})^2 + (\hat{y}_k^- - S_{y,1})^2}} & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{(\hat{x}_k^- - S_{x,4})}{\sqrt{(\hat{x}_k^- - S_{x,4})^2 + (\hat{y}_k^- - S_{y,4})^2}} & \frac{(\hat{y}_k^- - S_{y,4})}{\sqrt{(\hat{x}_k^- - S_{x,4})^2 + (\hat{y}_k^- - S_{y,4})^2}} & 0 & 0 \\ 0 & 0 & \frac{\hat{v}_{x,k}^-}{\sqrt{(\hat{v}_{x,k}^-)^2 + (\hat{v}_{y,k}^-)^2}} & \frac{\hat{v}_{y,k}^-}{\sqrt{(\hat{v}_{x,k}^-)^2 + (\hat{v}_{y,k}^-)^2}} \\ 0 & 0 & \frac{-\frac{\hat{v}_{y,k}^-}{\hat{v}_{x,k}^-}}{1+(\frac{\hat{v}_{y,k}^-}{\hat{v}_{x,k}^-})^2} & \frac{\frac{1}{\hat{v}_{x,k}^-}}{1+(\frac{\hat{v}_{y,k}^-}{\hat{v}_{x,k}^-})^2} \end{bmatrix} \quad (76)$$

The measurement update equations are those of the general EKF, noting that innovation is the difference between the observed measurement z_k (including RF measurements of distance, speed and direction) and estimation of $h(\hat{x}_k^-)$, as explained before.

4.6. Extended Kalman Filter to Inertial Navigation System

Two algorithms are developed in this section. Both equations use the EKF to fuse information from RF system with acceleration and angular velocity measurements provided by the INS. Acceleration measurements from INS are the platform coordinates, so we must apply the rotation matrix for the acceleration in local coordinates, as described in Appendix C and D.

The first method uses the CV model, so the estimated navigation states are position and velocity. The INS measurements are used as an input in the EKF, and the RF position values are used in the correction phase.

The second algorithm includes acceleration in the state vector, so it can be estimated. Due to that fact, the model is characterized by constant acceleration. RF system measurements and the INS are used to correct the EKF.

4.6.1. Constant velocity with input acceleration model (CVCA)

As stated before, the state vector includes position and speed of the master, being the general state vector for CV model.

$$x_k = \begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ x_{y,k} \end{bmatrix} \quad (77)$$

The dynamic equation which characterizes the model is the same as the CV algorithm seen before, but considering that the noise process w_k is replaced by measurement of acceleration in the local coordinate system and the matrix W is now B (referring to the input matrix relating to state), so the time update equation for the state vector is:

$$x_k = Ax_{k-1} + Bu_{k-1} \quad (78)$$

Where:

$$A = \begin{bmatrix} I_{2\times2} & TI_{2\times2} \\ 0_{2\times2} & I_{2\times2} \end{bmatrix} \quad (79)$$

$$B = \begin{bmatrix} \frac{T^2}{2} I_{2\times2} \\ TI_{2\times2} \end{bmatrix} \quad (80)$$

$$u_k = a^l = \begin{bmatrix} a_x \\ a_y \end{bmatrix} = T_b^l a^b \quad (81)$$

u_k input vector incorporates acceleration a and angular rate ω in measurements. One consideration is taken: the two components of the acceleration a_x and a_y are calculated using the master orientation obtained by propagating the navigation equation related to orientation θ with angular rate ω, as shown in *Figure 18*.

The covariance error *a priori* is given by:

$$P_k^- = AP_{k-1}A^T + BQ_kB^T \quad (82)$$

INS measurements are used as inputs, so the RF positioning system is used only in the correction phase of the EKF, including z_k measurement vector. So measurement estimation $h(\hat{x}_k^-)$ works in the same way as in CP or CPCV algorithm.

$$h(\hat{x}_k^-) = \begin{bmatrix} \sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2} \\ \vdots \\ \sqrt{\left(\hat{x}_k^- - S_{x,4}\right)^2 + \left(\hat{y}_k^- - S_{y,4}\right)^2} \end{bmatrix} \quad (83)$$

Then the Jacobian H_k will be:

$$H_k = \frac{\partial\left(h(\hat{x}_k^-)\right)}{\partial x_k} = \begin{bmatrix} \frac{\left(\hat{x}_k^- - S_{x,1}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2}} & \frac{\left(\hat{y}_k^- - S_{y,1}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2}} \\ \vdots & \vdots \\ \frac{\left(\hat{x}_k^- - S_{x,4}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,4}\right)^2 + \left(\hat{y}_k^- - S_{y,4}\right)^2}} & \frac{\left(\hat{y}_k^- - S_{y,1}\right)}{\sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2}} \end{bmatrix} \quad (84)$$

The measurement update phase follows the general case for EKF, using the definition of $h(\hat{x}_k^-)$ and H_k.

4.6.2. Constant Acceleration model (CA)

In this algorithm the position, velocity and acceleration of the master are estimated, as indicated in this state vector:

$$x_k = \begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \\ a_{x,k} \\ a_{y,k} \end{bmatrix} \quad (85)$$

RF positioning system measures and the INS measurements are used in the EKF correction phase, so the dynamics equation for CA model are shown in *Table 3*. w_k vector is the process noise.

$$x_k = Ax_{k-1} + Ww_{k-1} \quad (86)$$

Where:

$$A = \begin{bmatrix} I_{2\times2} & TI_{2\times2} & \frac{T^2}{T}I_{2\times2} \\ 0_{2\times2} & I_{2\times2} & I_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & I_{2\times2} \end{bmatrix} \quad (87)$$

$$W = \begin{bmatrix} \frac{T^3}{6}I_{2\times2} \\ \frac{T^2}{2}I_{2\times2} \\ TI_{2\times2} \end{bmatrix} \quad (88)$$

The a *priori* state estimation error and covariance equations are obtained from time update:

$$\hat{x}_k^- = A\hat{x}_{k-1}^- \quad (89)$$

$$P_k^- = AP_{k-1}A^T + WQ_kW^T \quad (90)$$

The estimated measurements in $h(\hat{x}_k^-)$ are the distances from master to each slave and local frame accelerations. ω is not estimated, as it is used to calculate the orientation of the master.

$$h(\hat{x}_k^-) = \begin{bmatrix} \sqrt{\left(\hat{x}_k^- - S_{x,1}\right)^2 + \left(\hat{y}_k^- - S_{y,1}\right)^2} \\ \vdots \\ \sqrt{\left(\hat{x}_k^- - S_{x,4}\right)^2 + \left(\hat{y}_k^- - S_{y,4}\right)^2} \\ \hat{a}_{x,k}^- \\ \hat{a}_{y,k}^- \end{bmatrix} \quad (91)$$

$$\mathbf{H}_k = \frac{\delta(h(\hat{\boldsymbol{x}}_k^-))}{\delta \boldsymbol{x}_k} = \begin{bmatrix} \frac{(\hat{x}_k^- - S_{x,1})}{\sqrt{(\hat{x}_k^- - S_{x,1})^2 + (\hat{y}_k^- - S_{y,1})^2}} & \frac{(\hat{y}_k^- - S_{y,1})}{\sqrt{(\hat{x}_k^- - S_{x,1})^2 + (\hat{y}_k^- - S_{y,1})^2}} & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \frac{(\hat{x}_k^- - S_{x,4})}{\sqrt{(\hat{x}_k^- - S_{x,4})^2 + (\hat{y}_k^- - S_{y,4})^2}} & \frac{(\hat{y}_k^- - S_{y,4})}{\sqrt{(\hat{x}_k^- - S_{x,4})^2 + (\hat{y}_k^- - S_{y,4})^2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (92)$$

5. Simulation and comparison algorithms

5.1. Introduction

This chapter shows the test results of the algorithms described in the previous chapters with simulated trajectories using Matlab.

In this section we will carry out simulations of the algorithms we have seen before to check its performance in different paths. We will make two graphs, in which we show a measurement error of all the algorithms presented before and the time it takes to process the algorithm. In F*igure 21* we can see the main simulation scheme, comprising the following:

- Four slave antennas located at these coordinates (X, Y), expressed in meters: (0,0) (10,0) (10,10) (0,10)

- The mobile platform carrying the master and the IMU, will move into the area of the slave antennas and by the EKF and using sensor fusion method we will estimate the position of the mobile platform.

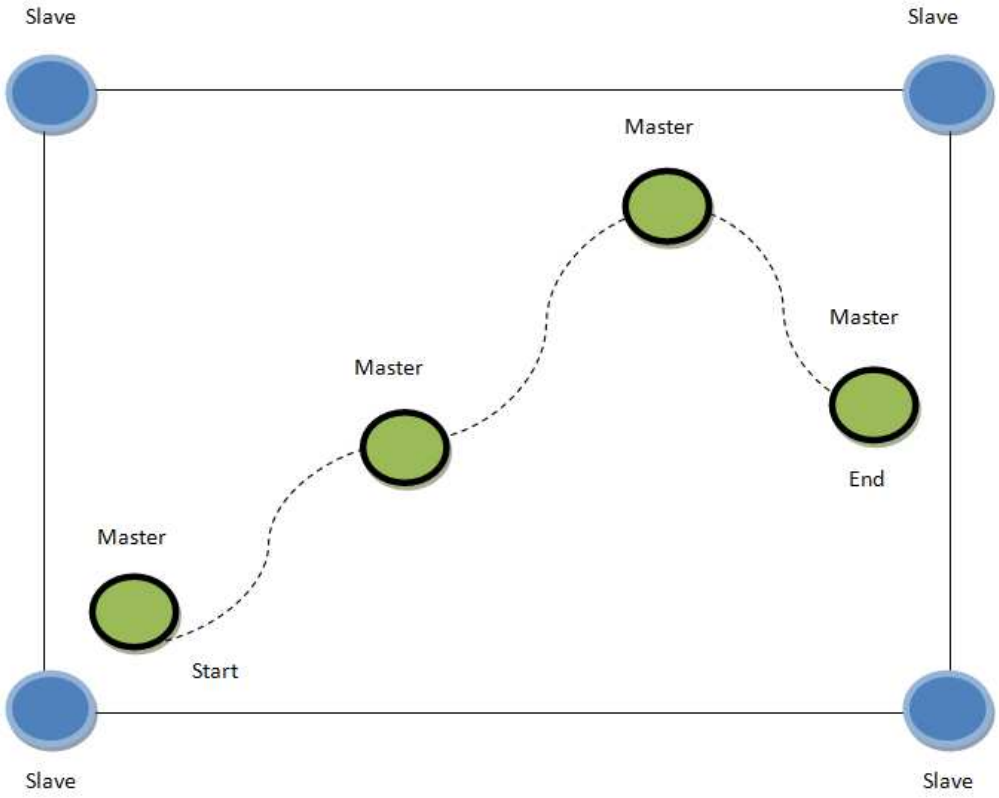


Figure 21. The simulation scheme

5.2. Simulation Setup

The trajectories are generated following the translational kinematic models shown in *Table 3*. Thus, a trajectory will follow the CP, CV, CPCV, CVCA and CA models. Also, spline trajectory is proved. It is created with the "spline" function of Matlab, which connects the points we have defined joining them and building a trajectory. As this trajectory is not following any of the dynamic models used in developed algorithms, it will be a good test for them.

The procedure for obtaining the true trajectory and noisy measurements begins with the propagation of two dimension kinematic equations of dynamic models for some time steps. The time period between two samples is T and is set to 100 ms. In each iteration, a sample of the state vector is stored.

Once all samples are obtained, the calculation of measurements is performed. The true trajectory is constituted by the position vector in the X and Y components. Velocity is obtained by the difference between two consecutive samples of position and dividing by T, first for a component and then the other. The acceleration is calculated by the difference of two consecutive velocity samples and dividing by T, again for the two components.

Range is calculated by computing the Euclidean distance between the master position in each iteration and the positions of the four slaves. The slaves are located at the vertices of a square of 10 m x 10 m. Specifically, their coordinates are (0,0), (10,0), (10,10) and (0,10).

To obtain the orientation we need to calculate the arctangent from the velocity components. Finally, the angular velocity is the difference between two consecutive samples of orientation divided by the sampling time T. Thus, the true trajectory of the master and all possible real measurements can be obtained. In order to acquire the noise measurements, white Gaussian noise is added.

The last thing to do is provide the state vector matrix and initial covariance error. Q process noise matrix and R noise measurement matrix have to be given and they could be modified for better performance.

Once the simulation is over, comparison between algorithms is performed based in two parameters. First, the most important, is Root-Mean-Square Error (RMSE) of the difference between the positions of real and estimated trajectory. The expression is written as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N^o\,Samples}(\|x_k - \hat{x}_k\|)^2}{N^o\,Samples}} \quad (93)$$

Where $\|x_k - \hat{x}_k\|$ is:

$$\|x_k - \hat{x}_k\| = \sqrt{(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2} \quad (94)$$

This error can also be evaluated in other navigation states, such as velocity or orientation. But in this project, only position error is evaluated, as it is intended to develop position algorithms.

The other parameter that will be compared between algorithms is the computation time required to run the simulations.The less computing time, the better the algorithm. For real-time operation, must be less than:

$$MaxComputingTime = Number\ of\ Sample \cdot 100ms \quad (95)$$

Finally, the results of the simulation are the average of 50 different simulations for each of the algorithms.

5.3. Simulation with a Constant Position trajectory

The translational kinematic model followed to create the trajectory is:

$$x_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot x_k + \begin{bmatrix} T \\ T \end{bmatrix} \cdot w_k \quad (96)$$

The starting position is in $x_0 = (1,5)$. w_k noise process has velocity units and its effects are changing the master position. Its variance for each component is $\sigma_x = \sigma_y = 0.02\,{}^{m}/_{s}$.

Q , R and P parameters have been adjusted to obtain the best precision in positioning. The resulting trajectories for this algorithm are shown in *Figure 22*.

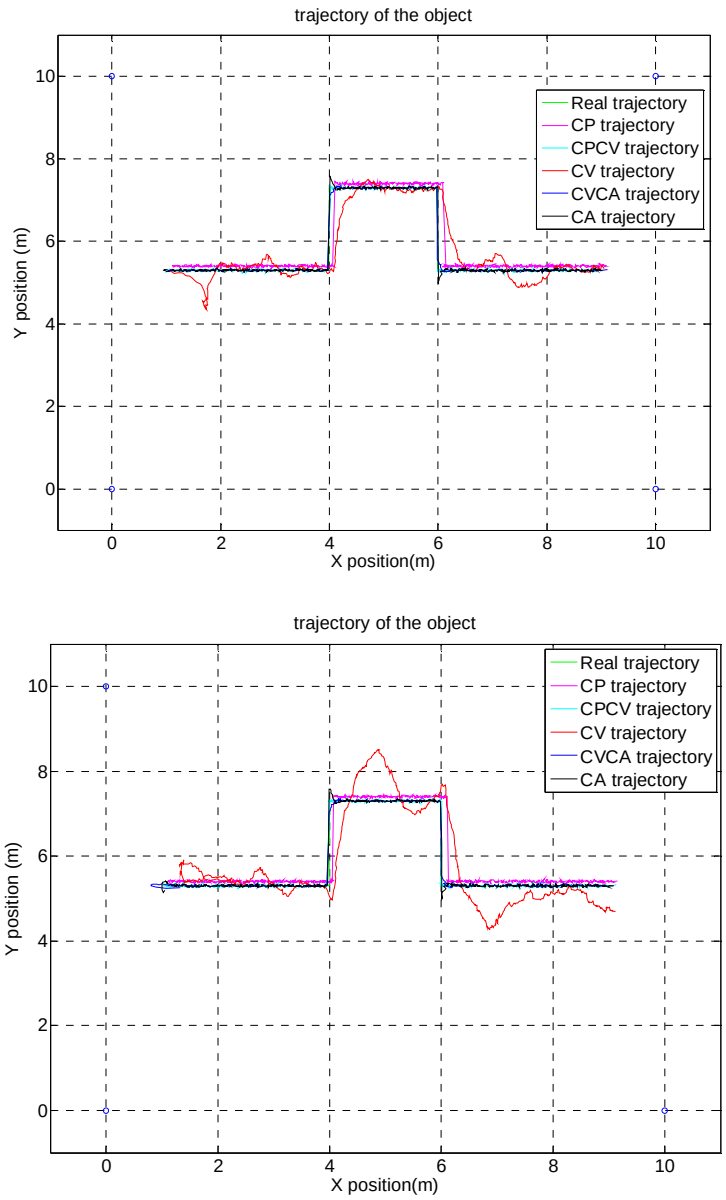


Figure 22. Trajectory obtained by applying the CP algorithm.

a) Above, proper initialization. b) At the bottom, random initialization.

All algorithms seem to have a good performance, as none of them are diverging. Graphics with the RMSE in position and time used in the simulation will help us to decide which algorithm is better. They can be seen in *Figure 23*.

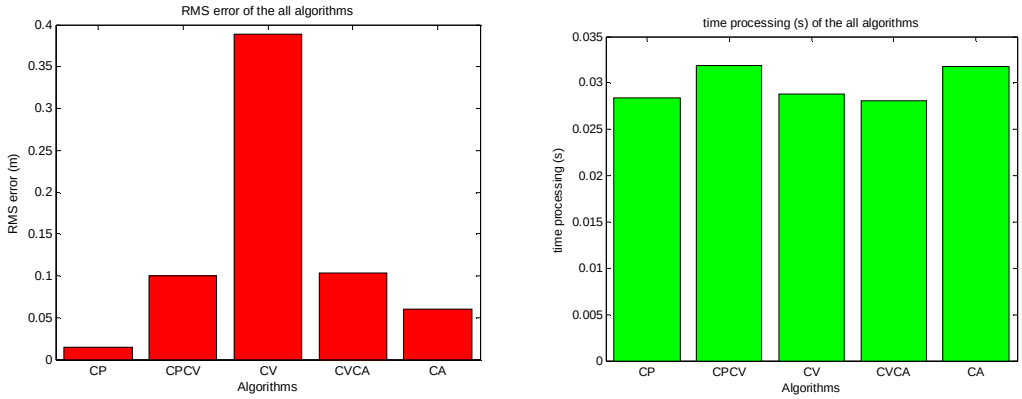


Figure 23. a) Left, RMSE graphic b) Right, time calculation

At first glance, CPCV, CA and CVCA algorithms seem to get the best accuracies, in the order of 10 cm. CP algorithm has the best, of about 1cm, while CV has a RMSE error of about 40 cm. Considering computation time, they all have similar results, approximately 30 ms. Fortunately, they don't exceed the maximum computing time allowed (100 ms).

In *Figure 23* we can observe that CV algorithm has some problems of trajectory tracking, probably due to incorrect adjustment of the parameters Q, R and P. Regarding to RMSE, they are similar to the values obtained with the correct initialization. Computation times are similar also.

In conclusion, if master is moving describing a trajectory of CP, the best algorithms are CP, CPCV, CA and CVCA, with accuracies in the order of 1-10 cm. Computation time is not a problem for them.

5.4. Simulation with a Constant Velocity trajectory

The trajectory is generated, following the translational kinematic equation:

$$x_k = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot x_{k-1} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \cdot w_k \qquad (97)$$

w_k noise process has units of acceleration. The variance for the noise process in each component is the same and equal to $\sigma_x = \sigma_y = 0{,}001\,{}^{m}/_{s^2}$

Two different simulations are performed. In the first one, algorithms are initialized with the correct initial state vector. The second simulation is initialized to a random position. Both simulations are generated with an initial velocity module equal to 0.2091 ${}^{m}/_{s}$, and the master oriented approximately 51.8236º. The resulting trajectories, the position RMSE and computation time are presented in *Figures 24 and 25*.

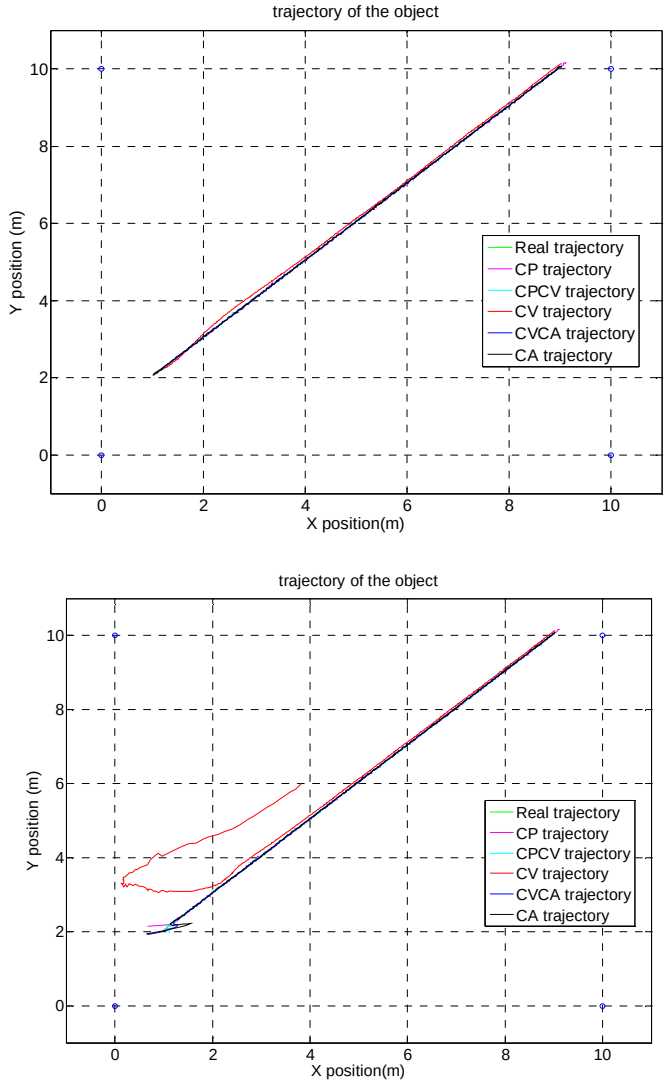


Figure 24. Trajectory obtained by applying the CV algorithm

a) Above, proper initialization b) At the bottom, random initialization.

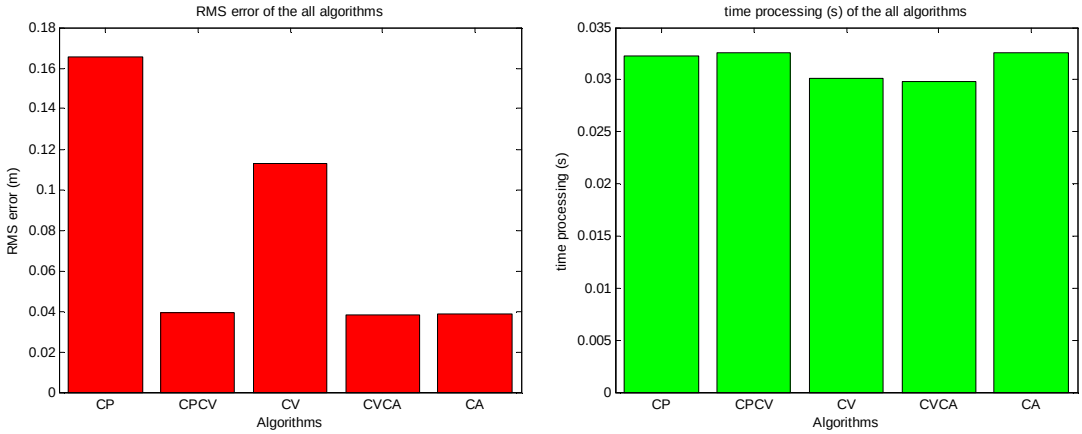


Figure 25. a) Left, RMSE graphic b) Right, time calculation

All algorithms reach convergence in a few iterations. In terms of positioning accuracy, the best algorithms are CPCV, CVCA and CA, with only 4 cm of error. The other algorithms also converge but their error is bigger, between 11 ~ 17 cm. It is important to note that the computation times are similar to those obtained in previous simulations.

5.5. Simulation with a constant acceleration trajectory

The trajectory is generated by the following dynamic equation:

$$x_{k+1} = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot x_k + \begin{bmatrix} \frac{T^3}{6} & 0 \\ 0 & \frac{T^3}{6} \\ \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \cdot w_k \qquad (98)$$

In this case, w_k noise process has units of jerk (derivative of acceleration), and their effects modify the acceleration values. The variance of noise process in each component is $\sigma_x = \sigma_y = 0.00001\ {}^{m}/_{s^3}$

Again two simulations are performed: one with the correct initialization in the state vector and the other with random initialization. The initial speed of the master is 0.33 m/s, orientated 21.61º, the acceleration in the X component is 0.03 m/s^2 in the first simulation and 0.05 m/s^2 in the second. The acceleration in the Y component is 0.0106 m/s^2 and 0.003 m/s^2respectively. *Figure 26* shows the results obtained in Matlab.

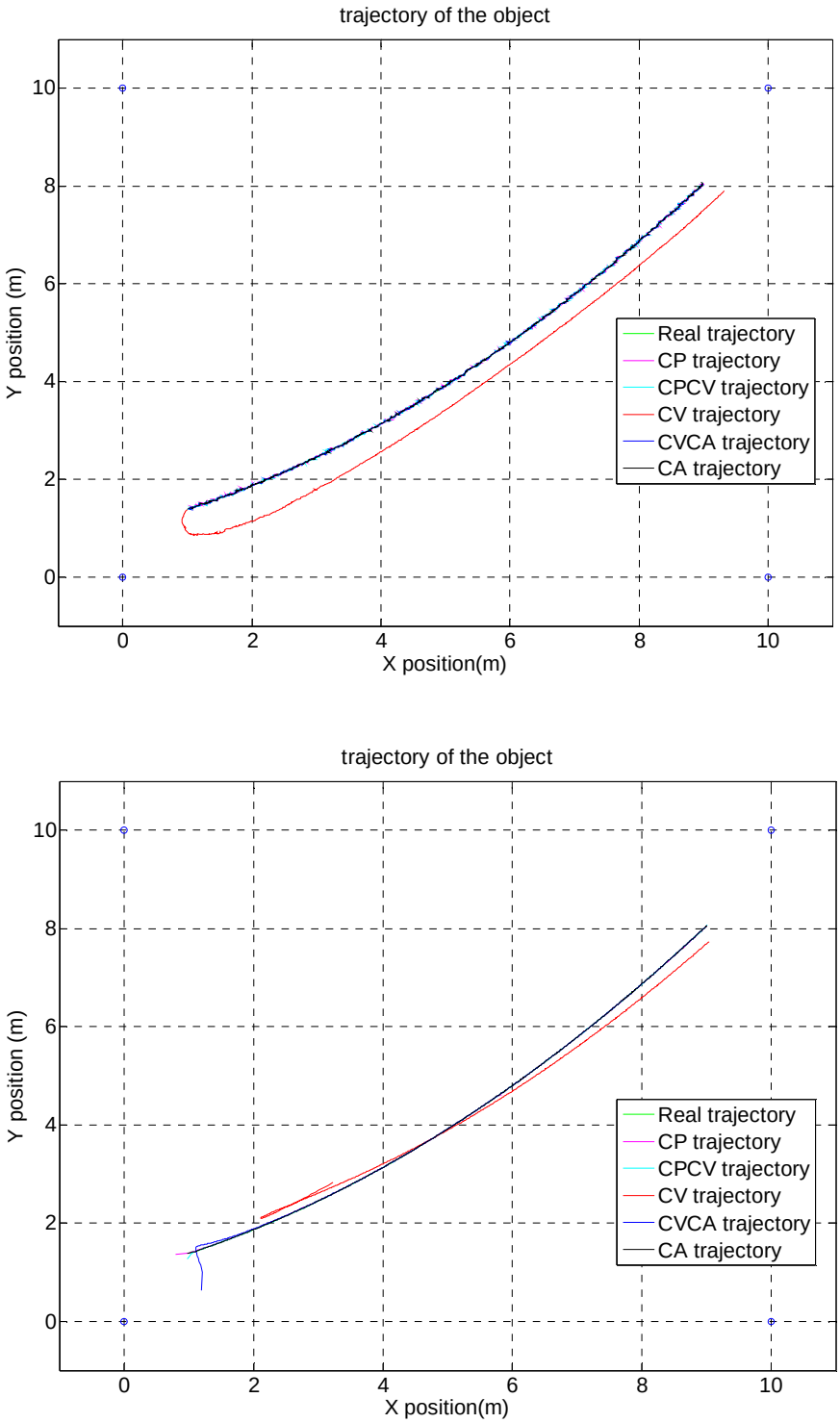


Figure 26. Trajectory obtained by applying the CA algorithm.

a) Above, proper initialization. b) Bottom, random initialization.

In both cases the state vector with random and correct initialization of the position, convergence is reached in a few iterations. Errors obtained are slightly smaller than in previous simulations. The best algorithms are CP, CPCV, CVCA and CA, with errors between 1.5 ~ 2.5 cm.

The time taken to run simulations appears to be independent of the analyzed trajectory, and the results are very similar to the values previously obtained.

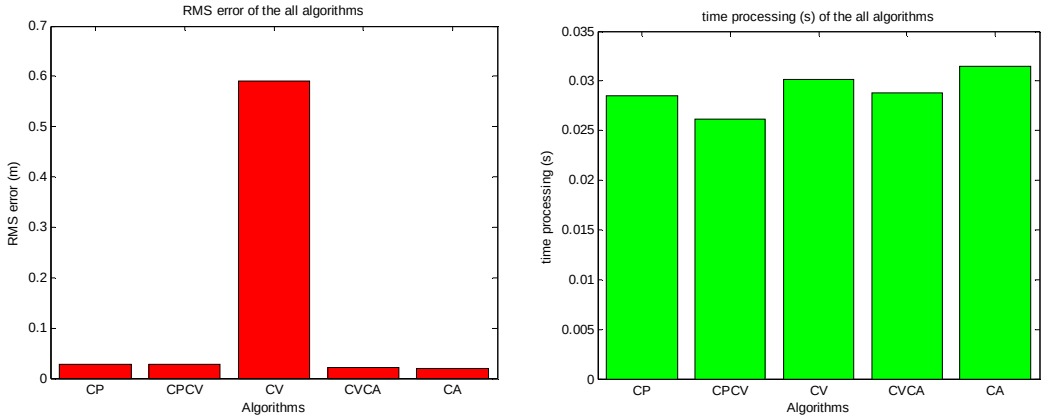


Figure 27. a) Left, RMSE graphic b) Right, time calculation

5.6. Simulation with a trajectory with Spline Matlab function

The generation of this trajectory is carried out using the Matlab "spline" function. This function uses a cubic interpolation between points, it generates paths between them, considered as inputs for the chosen trajectory.

Once the trajectory is established and the noise measures calculated, two simulations can be carried out in the same way as previously seen: one with a successful initialization and the other at random. Results are presented in *Figure 28*.

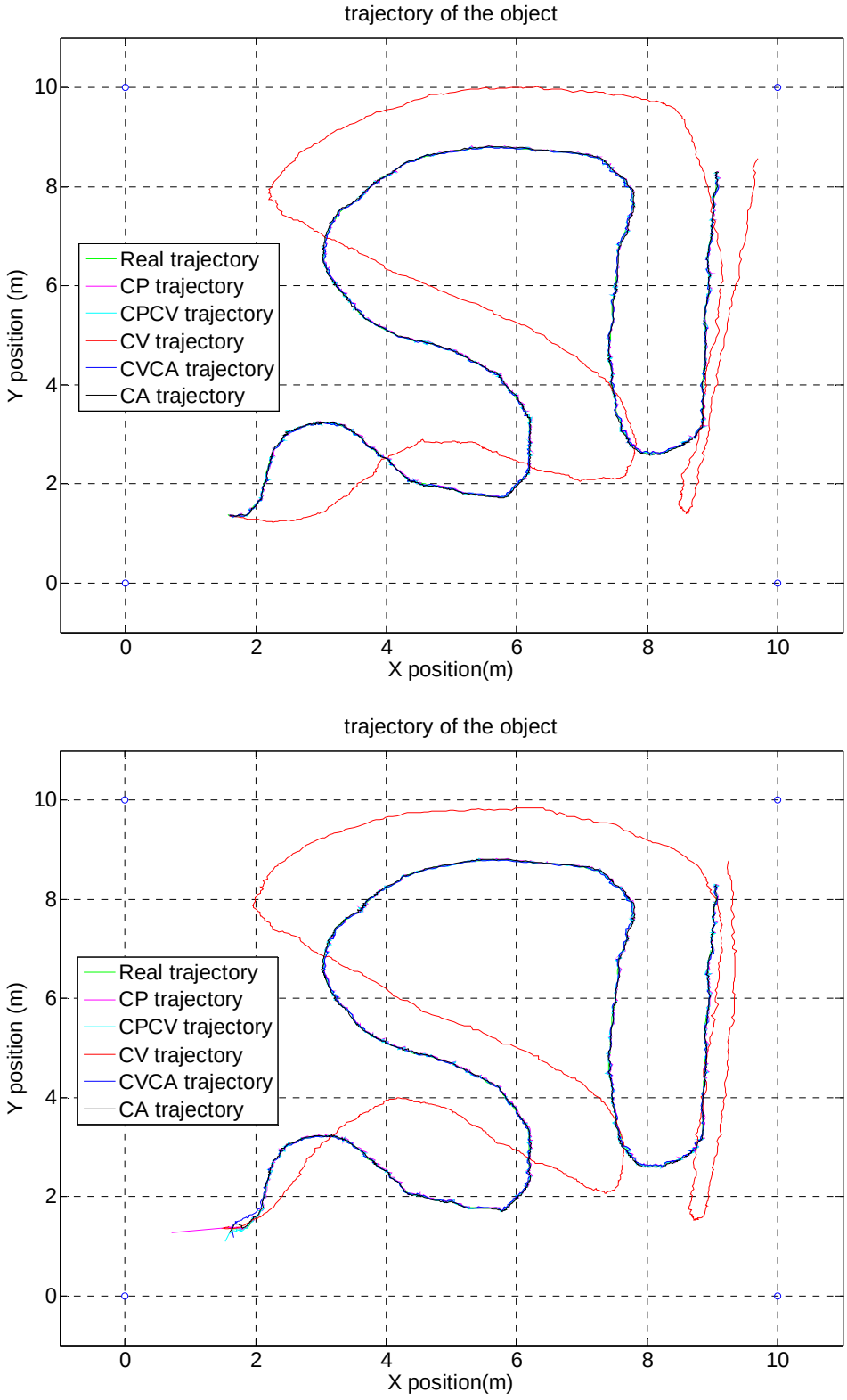


Figure 28. Trajectory obtained by applying the equations of Spline.

a) On top, proper initialization. b) On the bottom, random initialization.

Simulations show the algorithms' convergence. The best algorithms are CP, CPCV, CVCA and CA; errors are small, on the order of 3 ~ 4 cm, so it seems that the algorithms have enough power to correctly estimate the master position even if the dynamic model is none of the models used by the algorithms. Computing time graphic confirmed that algorithms are not dependent on the trajectory model, and they spend similar time in all simulations.

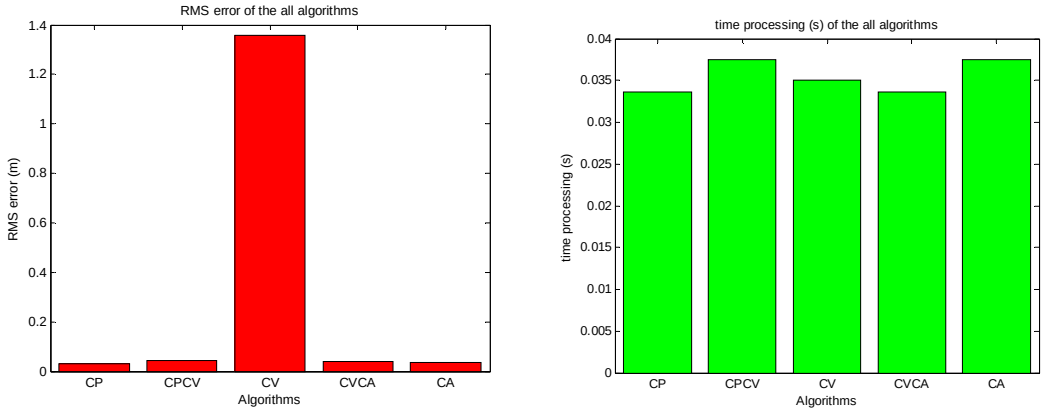


Figure 29. a) Left, RMSE graphic b) Right, time calculation

5.7. Conclusion

The trajectory CV, in all algorithms, is the one giving the worst results. In the other simulated trajectories, all algorithms are working quite well. The most consistent among them are CP, CPCV, CVCA and CA, maintaining low error in all cases. The RMSE and computing time obtained in the simulations are summarized in *Table 4*. Computation times are below the maximum allowed time (they can do all the calculations needed before the next action is taken).

It is important to realize that all the algorithms developed are improving the accuracies obtained with RF, DR and INS systems working on a stand-alone configuration. This is one of the benefits of information fusion.

In conclusion, in a simulated scenario with controlled and known errors, all the algorithms have good performance. Now, the interesting thing is to test the algorithms on a real and uncontrolled scenario, where we have many more sources of error such as vibration or interference, possible areas of shadow, outliers in the measurements and other unexpected circumstances that may worsen results.

6. Simulation with stand-alone systems

In this section, we will simulate the positioning systems RF, DR and INS with a stand-alone configuration. These cases are analyzed with the spline trajectory. In the next *Figure 30*, we can see the real measure and the RF trajectory.

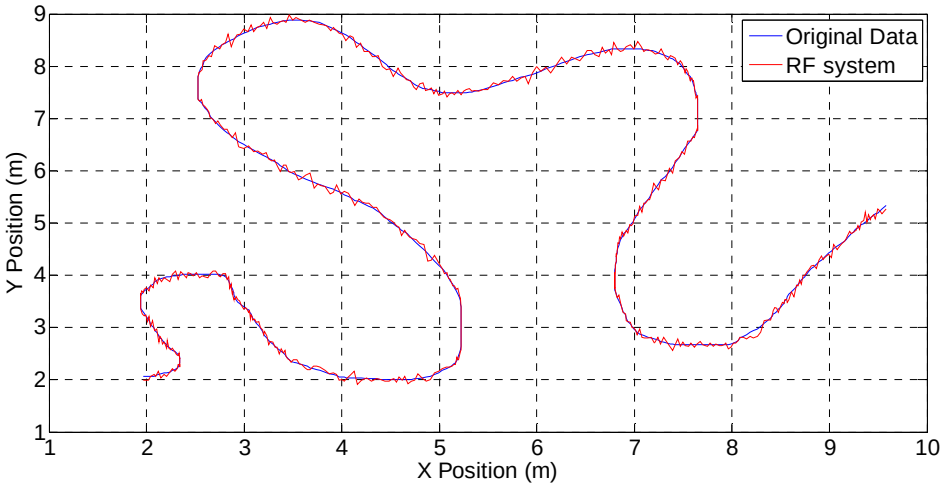


Figure 30. RF system stand-alone

The RMSE obtained is 3,264 cm. It's a low value because the scenario of simulations is only taking into account the general errors, but in real scenarios, other sources of error such as interferences can be present. However, in general, the algorithms have less error than the stand-alone RF system.

INS and DR system can work as autonomous positioning systems. The trajectories resulting from the navigation equations of the DR system are shown in *Figure 31*.

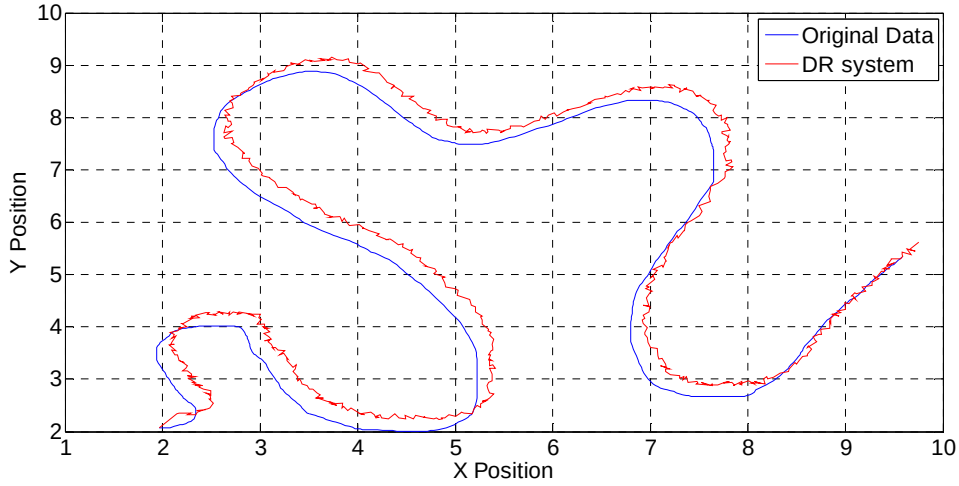


Figure 31. DR system stand-alone

It is easy to see that the results are not good. The DR system is following the true trajectory, but is starting to diverge. It is logical since the error grows with time. The RMSE is 1,18 m.

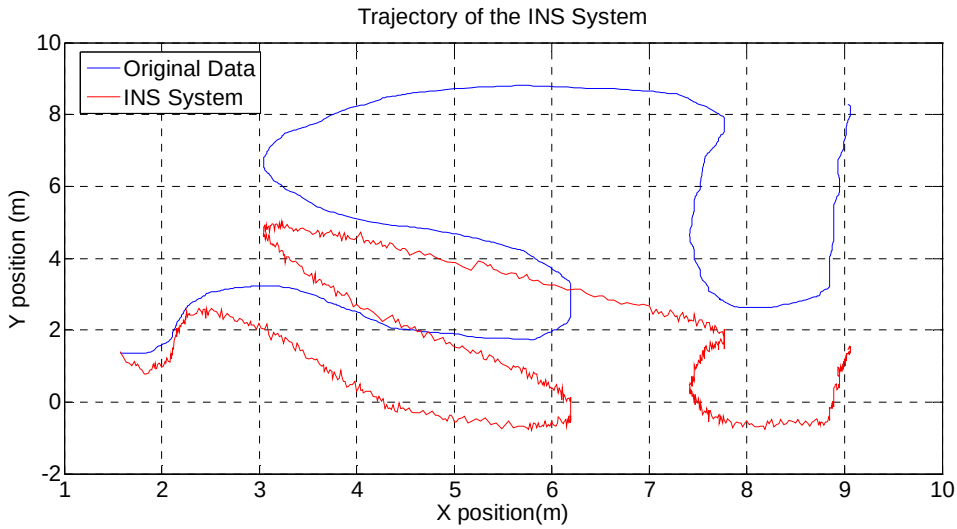


Figure 32. Inertial navigation system as a stand-alone

On the other hand, the INS trajectory is diverging from the beginning, due to the biases and the increasingly accumulated error. RMSE is 3.27 m.

In conclusion, the developed algorithms are improving the results that we obtain with the stand-alone systems. These results are confirming the advantages of the information fusion.

6.1. Results discussion

After all the possible simulations with the stand-alone or the sensor fusion, we have created a table where the error measurements (RMSE) and the computational times are summarized:

Algorithms	CP model		CV model		CA model		Spline	
	Error	Time	Error	Time	Error	Time	Error	Time
CP	1 cm	28 ms	16 cm	31ms	2,71 cm	28 ms	1,44 cm	34 ms
CPCV	10 cm	32 ms	4 cm	30 ms	2,67 cm	26 ms	4,49 cm	37 ms
CV	40 cm	29 ms	12 cm	29 ms	59,1 cm	30 ms	138 cm	35 ms
CVCA	10 cm	27ms	3,9 cm	28 ms	2,09 cm	29 ms	4,21 cm	34 ms
CA	4,9 cm	31ms	3,8 cm	32 ms	1,86 cm	32 ms	4,18 cm	36 ms
RF system	–		–		–		3,26 cm	
DR	–		–		–		118 cm	
INS	–		–		–		327 cm	

Table 4. Summary of the RMSE

Regarding to RMSE, the CV algorithm is giving the worst results. In the others simulated trajectories, all the algorithms are working well. The most constant of them are the CP, CPCV, CVCA and CA, maintaining a low error in all the cases. The computing times are below the maximum time allowed (they can make all the necessary calculation before new measurements are taken).

We can conclude that systems working independently have bigger error than with EKF information fusion. Sensor fusion allows us to recollect information from other systems and fuse them to get the best system position with minimum error. We can estimate other variables, as velocity or acceleration, as stated before. In this case, we have only focused on the position.

7. Test with experimental datasets

In this chapter, previously developed algorithms will be tested with a real experiment. These experimental results were obtained with 3 antennas placed in a triangular shape and an object in the centre of the triangle (master). We were moving the master from 0 up to 10 mm in steps of 0,5 mm and obtaining results. In the next figure 33, we see the configuration of the experiment.

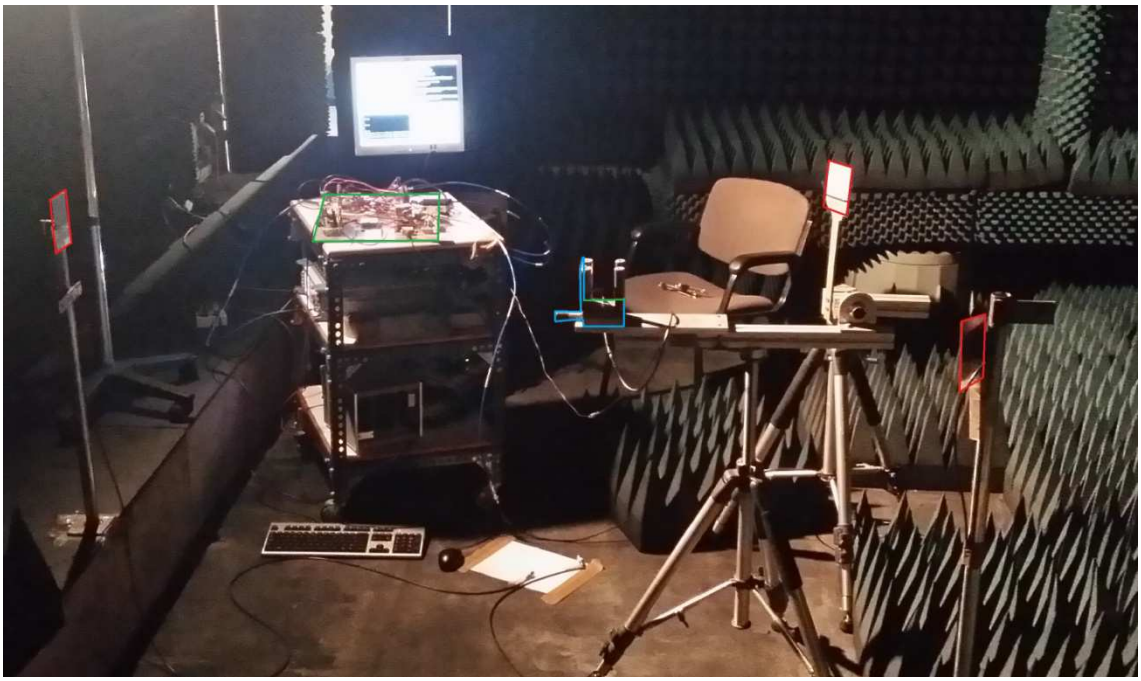


Figure 33. Vision of the real experiment

In red the slave antennas are represented, in blue there is the master antenna and RF system is painted green.

We performed the experiment in the anechoic chamber because we needed that the RF positioning system to avoid the noise from outside, being in a controlled scenario. In the anechoic chamber we had isolation in front of outside interference. Anechoic chamber is a camera with special isolation, which external noise cannot penetrate inside the chamber.

The master is located in a micrometer screw in the center of the triangle (blue line in Figure 33) and the three slave antennas are in the vertexs of the triangle (red line). In our experiment, we move the micrometer screw in steps of 0.5 mm to prove if the system is receiving and detecting changes, to a distance of 10 mm.

Once we have obtained these measurements, we processed the received signals and we were able to calculate the distance at which the master is. When we have the differencies of distances, we apply EKF to them to improve the measurement results. In this case, the movement has been a linear motion.

We conducted three types of movement, as we can see in *Figure 34, 35* and *36,* respectively:

-In the first experiment we have moved the micrometer screw in 0.5 mm steps

-In the second experiment we have kept the master quiet during all the measurements

-In the last experiment we have moved in steps of -0.5 mm.

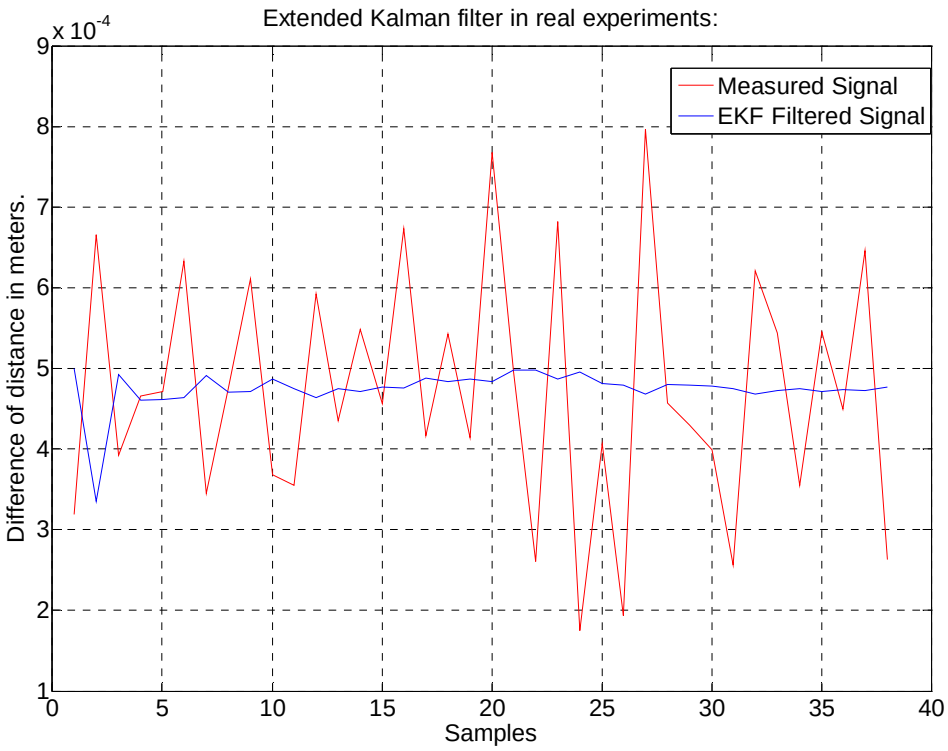


Figure 34. Measured signal (red) and filtered signal(blue) with 0.5 mm steps

As we can see, EKF converges quickly, tends to 0.5 mm and , although we have some errors in our signal due to noise, the filter is really working well.

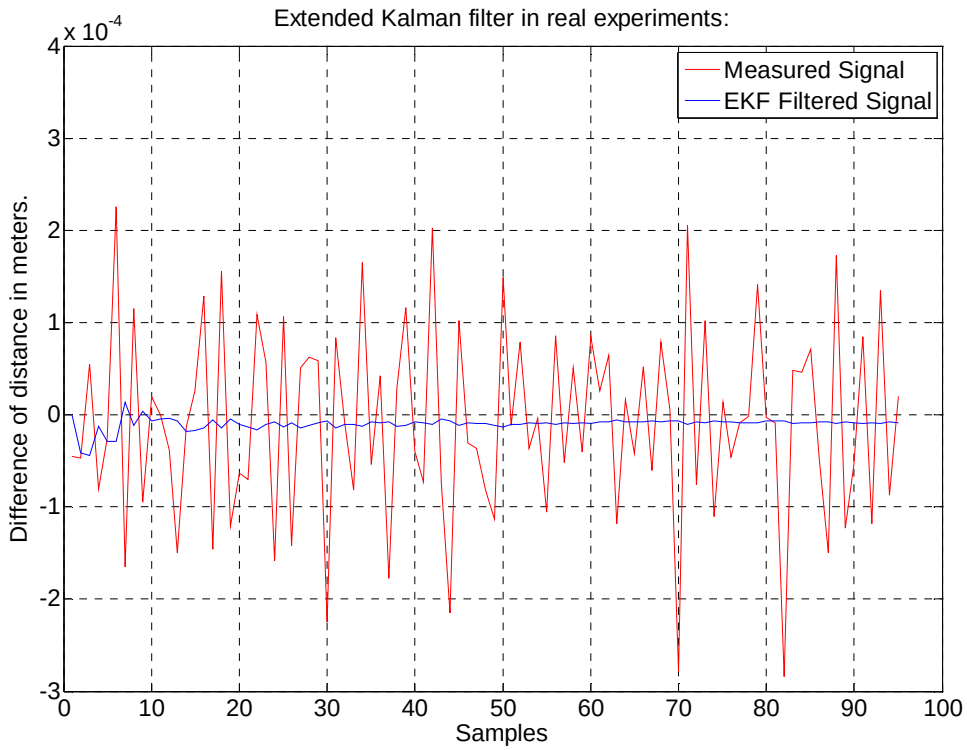


Figure 35. Measured signal (red) and filtered signal(blue) with still master

In the second experiment, which consisted of leaving the master without any movement, the EKF converges quickly and stays on 0 mm. We can see the noise in the measurements, but nonetheless, the filter is able to avoid the noise and estimate the real position accurately.

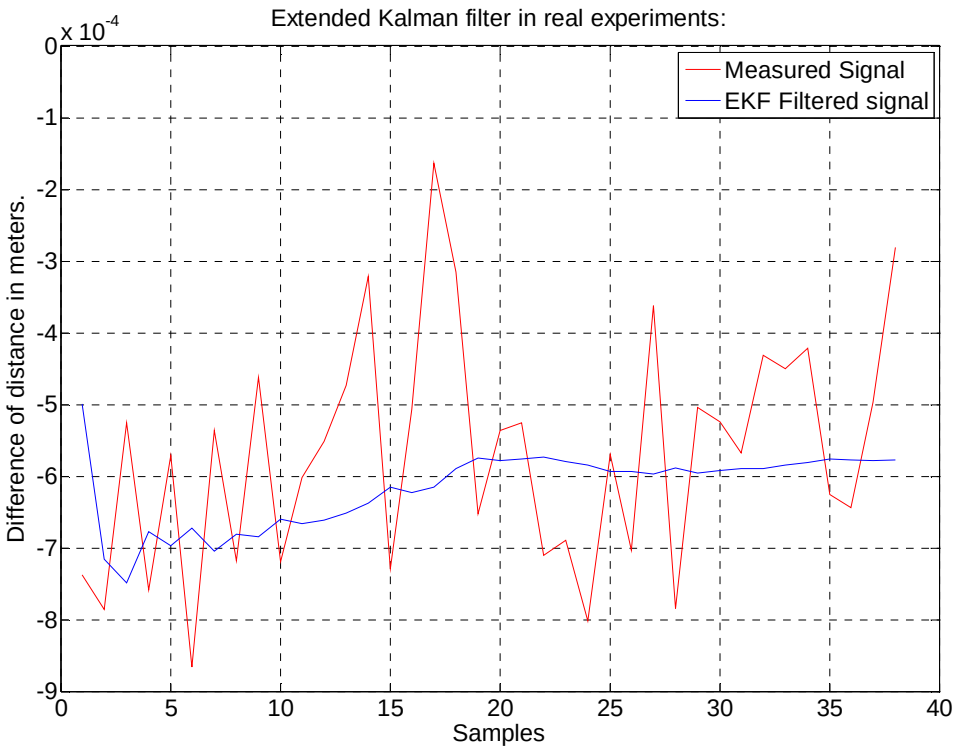


Figure 36. Measured signal (red) and filtered signal(blue) with negative 0,5 mm steps

In the last case, we can see the experiment in which the master moves in steps of -0.5 mm. As we observe, Kalman filter tries to converge to -0.5 mm but it is not so accurate as required. This may be due to a poor fit in P, Q or R, as these parameters are important for the filter to follow our signal.

In our experiment, we have used a micrometric screw to prove if the distance measurements were accurate enough; moreover, we designed a mobile platform to prove the system with different characteristics, as seen in *Figure 37*.

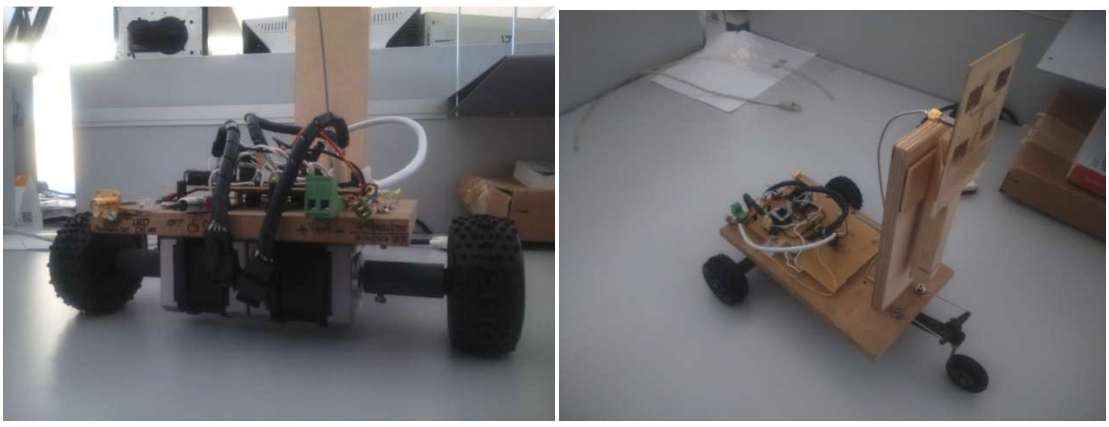


Figure 37. The prototype of the mobile platform

7.1. Real experiment conclusions

In the experiments we have done in the anechoic chamber and in view of the results we have obtained, we can conclude that applying the EKF to our measurements is a good choice for keeping track of the position, given its precision and speed of computation. The EKF can be used in real time; as we have seen in the simulations, the processing time is very low, in the order of about 30 ms.

As we have seen in the experiment in which the master moved in steps of 0.5 mm, the result of applying Kalman to these measurements has been successful. The same happens applying the Kalman filter to measurements when we are motionless. The only result being more or less satisfactory is when we move the master in negative steps. It may be because the movements we did in the micrometer screw were not accurate; however, Kalman algorithm attempts to converge towards the position of -0.5 mm.

8. General conclusions

The goal of this project was to provide positioning algorithms for navigation applications. In our case, there was no commercial system giving us the high accurate degree we needed for the measurements, so a RF positioning system was designed. Through the use of EKF and the sensor fusion, we could obtain satisfactory results by combining of the three systems (RF, DR, INS), rather than with stand-alone systems.

The solution is focused on the development, implementation and evaluation of different sensor fusion algorithms. Such algorithms are able to take the benefits of individual sensors that get better accuracy in the measurements, through the fusion of the information acquired by each sensor.

A total of five algorithms have been implemented. Three of them (CP, CPCV and CV) use RF positioning system measurements and the velocity and orientation which are given by a dead reckoning system. The two others (CVCA and CA) use RF positioning system measurements and they work with acceleration and angular rate measurements of an INS.

We have adapted the trajectories to dynamic models; we have created also several paths based on different kinematic equations and we have tested them in different algorithms. As we have seen in the RMSE graphic before (Table 4), EKF is a good filter to track the position.

The results have shown that all the dynamic models could elaborate all the calculations required in the time step between two consecutive measurements. These results confirm that the use of algorithms developed in real-time applications is feasible.

After that, we tested the algorithms with real data sets taken from an experimental setup, in which the sources of errors could not be known and algorithms had to be tuned in order to obtain the best performance. These data sets were acquired from a doctoral thesis based on a RF positioning system of synchronized phase [3] [4]. The test results have shown that the algorithms that fuse measurements from the RF system, DR system and INS, were able to improve accuracies obtained with the individual systems.

However, as stated before, algorithms have been able to match the results of the RF system, which was the best of the individual systems. This was the confirmation of the robustness of the fusion sensor algorithms developed, which are able to discard unreliable measurements and estimate the position and other navigation states based on the useful information.

8.1. Future work

Although in this project satisfactory results have been obtained, the application of all this systems to the real environment is a challenge. These algorithms have been developed to use them in two dimension positioning applications. However, increasing to three dimensions would provide a solution to all kinds of positioning applications. This would be another important improve to the work done in this project.

Another improvement it would be to implement *Unscented Kalman Filter* algorithm (UKF), since when the dynamic model and measurements are highly nonlinear, EKF loses efficiency. The result is a filter which captures the true mean and covariance

more accurately. In addition, this technique removes the requirement to calculate Jacobians and the inversion of the innovation covariance matrix in EKF filter, which for complex functions can be a difficult task itself.

A. Appendix A

In *Table 5* notational convention used in this project is described.

x	Scalar number
$\bar{x}$	Vector number
A	Matrix
$\hat{x}$	Estimated value of x
$\hat{x}^-$	A priori estimated value
$\tilde{x}$	Measured value x
δx	Error $x - \hat{x}$
x_k	x value at the moment k
T_a^b	Transformation/rotation matrix from reference frames a to b

Table 5. Notation

As algorithms are implemented in discrete time, estimates will evolve using a discrete time T step. When a variable in a certain time interval is indicated, k subscript is used.

B. Appendix B

Director cosine

The director cosines of R vector are the cosins of angles that each of the coordinate axis conform. They represent in a three-dimensional plane like this:

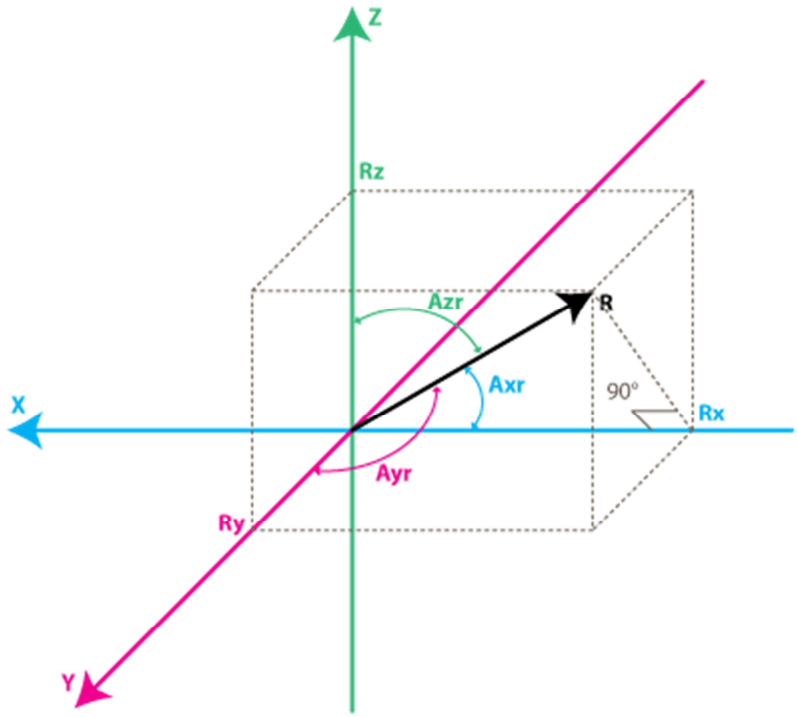


Figure 38. Coordinate system

Three angles are identified in the image (A_{xr} , A_{yr} , A_{zr}). Their formulas to find the angle are:

$$\cos A_{xr} = \frac{R_x}{|R|} \quad ; \quad \cos A_{yr} = \frac{R_y}{|R|} \quad ; \quad \cos A_{zr} = \frac{R_z}{|R|} \qquad (99)$$

Where the module of the vector R formula is used:

$$|R| = \sqrt{{R_x}^2 + {R_y}^2 + {R_z}^2} \qquad (100)$$

If we want to know the angle, we can apply the arcosine or inverse cosine:

$$A_{xr} = \cos^{-1}\left(\frac{R_x}{|R|}\right) \quad ; \quad A_{yr} = \cos^{-1}\left(\frac{R_y}{|R|}\right) \quad ; \quad A_{zr} = \cos^{-1}\left(\frac{R_z}{|R|}\right) \qquad (101)$$

In *Figure 39* we can see a sample of real values of the accelerometer, the angle depending on the orientation of the sensor.

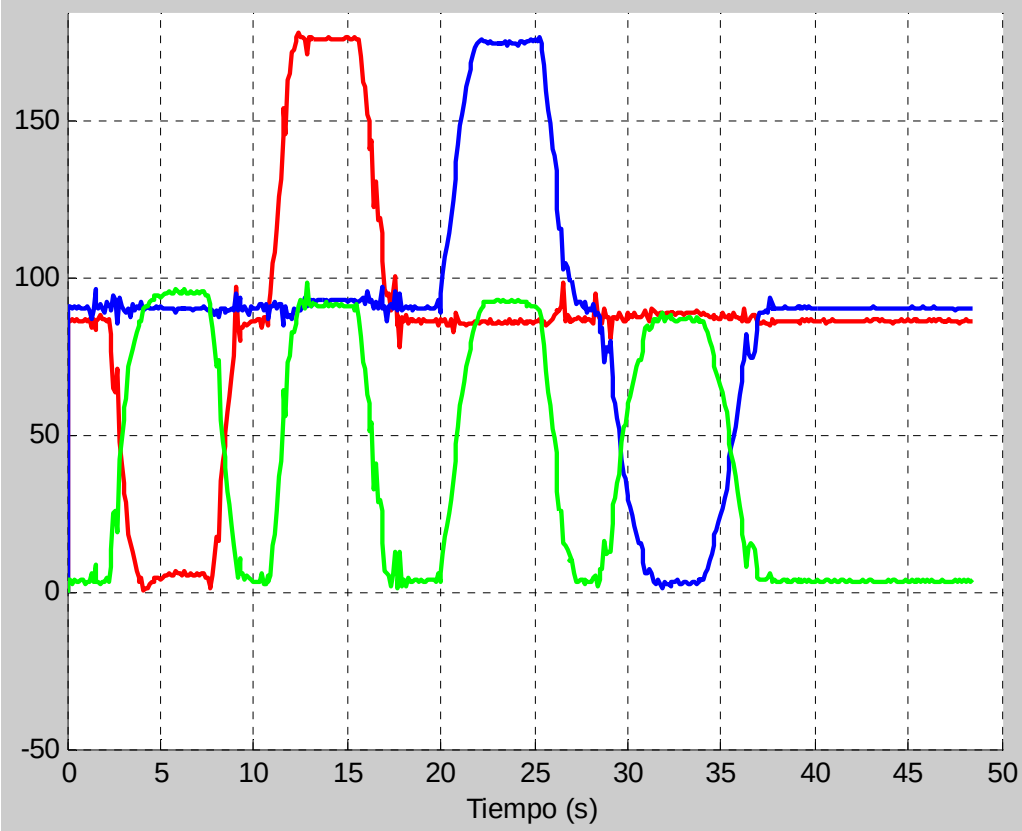


Figure 39. Real measurement applying cosine director

In this case all angles are measured with respect to gravity, which is vertical downwards.

The red line represents roll angle. If under 90º means that we are tilted left, and if above 90º right. The blue line represents pitch angle. We can see when we turn upwards, the signal dials 180º and when we are downwards, the dial is 0º. The green line shows the Z angle of the accelerometer.

We can use other equations that give us the angle between -90 ° and 90 °, as shown below:

$$Angle\ X = atan\left(\frac{y}{\sqrt{x^2 + z^2}}\right) \quad (102)$$

$$Angle\ Y = atan\left(\frac{x}{\sqrt{y^2 + z^2}}\right) \quad (103)$$

$$Angle\ Z = atan\left(\frac{\sqrt{x^2 + y^2}}{z}\right) \quad (104)$$

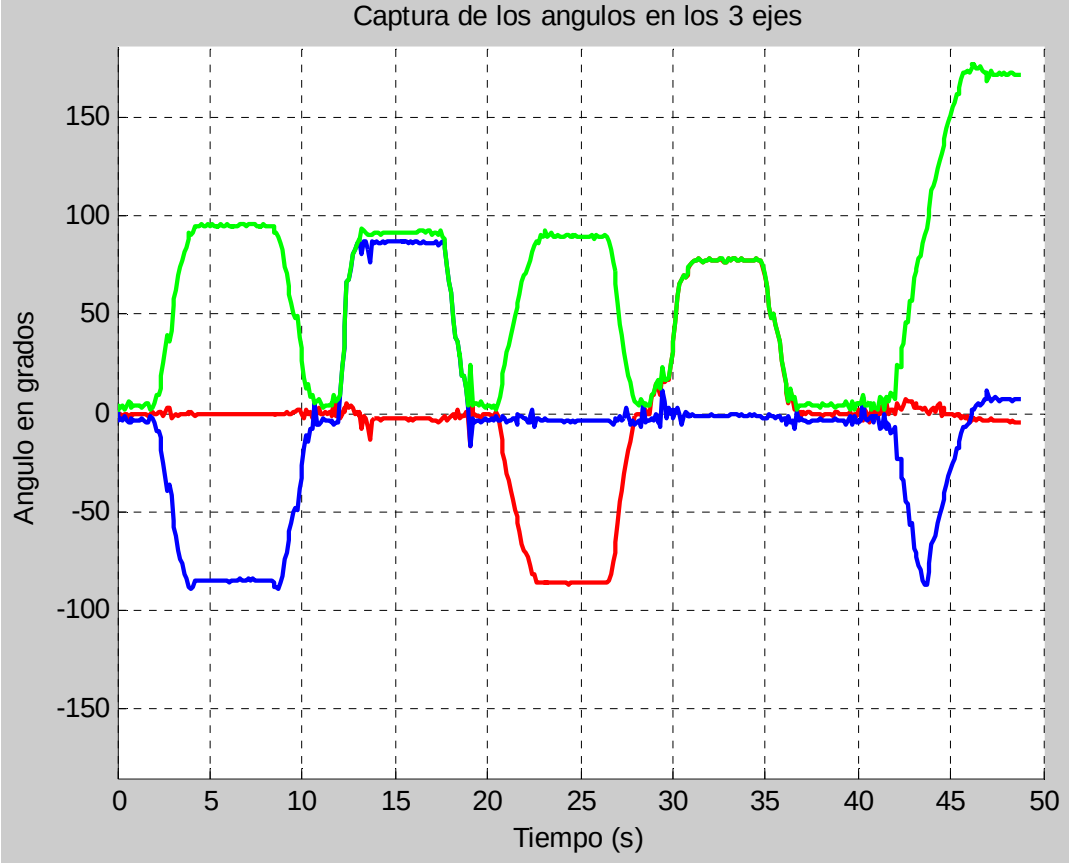


Figure 40. Real measure applying arctan

At the end of the graph we can see that the green line goes from 0º to 180º, this means that when we pass 90º, the platform is inverted.

C. Appendix C

Platform rotation

In navigation applications, the goal is to determine the position and velocity measurements based on several sensors connected to a vehicle. In our study we only consider two dimensions, measures will be given in the components *X* and *Y*. But the model can be easily adapted to three dimensions, adding the *Z* component corresponding to the perpendicular axis.

The inertial measurements are obtained in the reference axis of the sensor (local coordinate system). The IMU will be installed in the center of the platform with the master antenna. The sensor is perfectly aligned and oriented with the platform.

To know the orientation or rotation of the master antenna in the reference coordinate system (local coordinate system) and the body coordinate system (slave antennas origin position), we need a rotation matrix where measures of rotation platform coordinate system can be applied to navigation equations in the global coordinate system. *Figure 41* shows coordinate system.

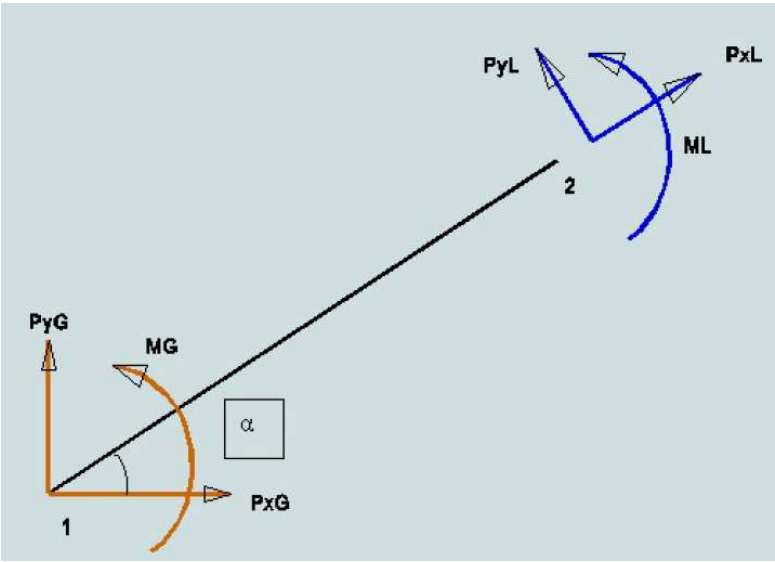


Figure 41. Local coordinate system (blue) and a global coordinate system (red)

$$P^G = T \cdot P^L \qquad (105)$$

Where P^G are the coordinates in the global reference system (transmitting antennas), T is the rotation matrix and P^L are the coordinates in the reference system of the body (master).

$$\begin{bmatrix} P_x^G \\ P_y^G \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -sen\,\alpha & 0 \\ sen\,\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_x^L \\ P_y^L \\ 1 \end{bmatrix} \quad (106)$$

D. Appendix D

Rotation matrix

Rotation matrices define algebraically which is the rotation in a three dimension space, considering a rotation angle. Rotation matrices have qualities that are important to note:

- Its coordinate axes are orthogonal vectors (angle of 90 degrees between them)
- The determinant is 1
- If the normal of any vector belonging to the matrix outputs, the result is 1, so it is a unitary matrix.
- As an orthogonal matrix its transpose and inverse are equal

Figure 42 shows a coordinate axis and matrix.

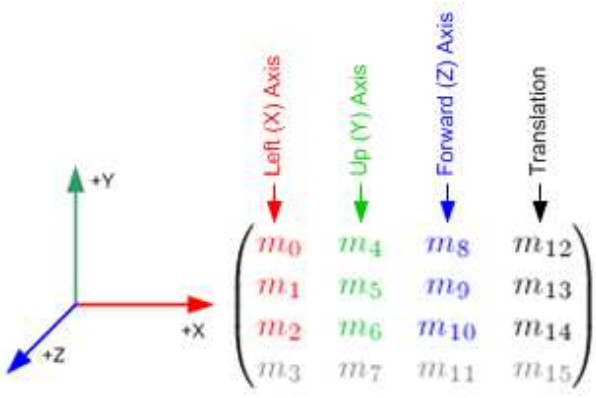
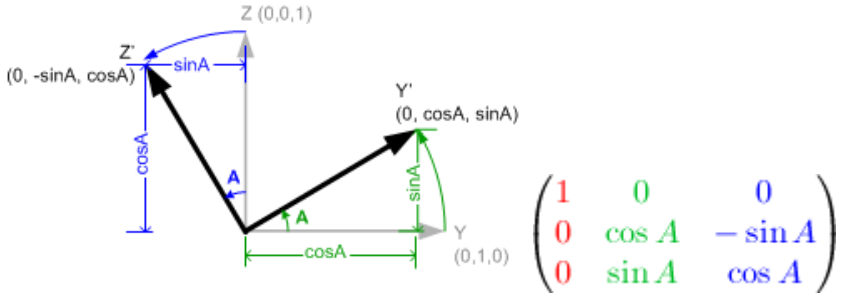


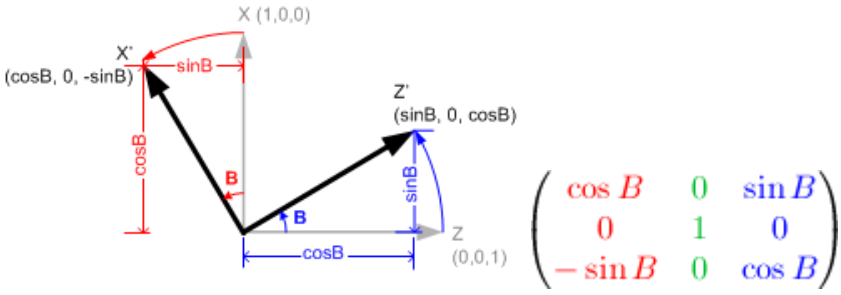
Figure 42. Rotation matrix

First, we look at a rotation around each axis; X, Y, and Z. We project the three axes on a plane in 3 different ways, so the axis we want to rotate is directed toward us. The positive direction of rotation is in the counterclockwise direction.



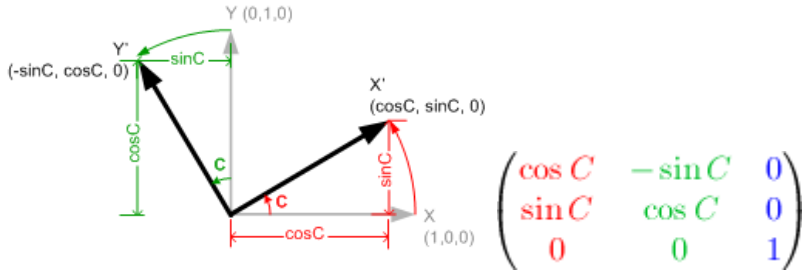
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix}$$

Figure 43. Rotation about Left(X) Axis (Pitch) and Rotation Matrix



$$\begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix}$$

Figure 44. Rotation about Up(Y) Axis (Yaw, Heading) and Rotation Matrix



$$\begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure 45. Rotation about Forward(Z) Axis (Roll) and Rotation Matrix

We can combine these rotations of independent axes in a matrix by multiplying the three matrices together. As matrix multiplication is not commutative, the order is important. There are 6 different possible combinations; RxRyRz, RxRzRy, RyRxRz, RyRzRx, RzRxRy and RzRyRx. The combination we will use in this project is RzRyRx.

$$
\begin{aligned}
&R_Z R_Y R_X \\
&= \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \\
&= \begin{pmatrix} \cos C \cos B & -\sin C & \cos C \sin B \\ \sin C \cos B & \cos C & \sin C \sin B \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \\
&= \begin{pmatrix} \cos C \cos B & -\sin C \cos A + \cos C \sin B \sin A & \sin C \sin A + \cos C \sin B \cos A \\ \sin C \cos B & \cos C \cos A + \sin C \sin B \sin A & -\cos C \sin A + \sin C \sin B \cos A \\ -\sin B & \cos B \sin A & \cos B \cos A \end{pmatrix}
\end{aligned}
$$

Figure 46. Three dimension rotation matrix

$$
R_l^b = \begin{vmatrix} cos\theta cos\psi & -cos\theta sin\psi + sin\phi sin\theta cos\psi & sin\phi sin\psi + cos\phi sin\theta cos\psi \\ cos\theta sin\psi & cos\theta cos\psi + sin\phi sin\theta sin\psi & -sin\phi cos\psi + cos\phi sin\theta sin\psi \\ -sin\theta & sin\phi cos\theta & cos\phi cos\theta \end{vmatrix} \qquad (107)
$$

In this rotation matrix, the angles ϕ, θ, ψ, correspond to the counterclockwise rotation in the X, Y and Z respectively. As our system works in 2 dimensions in the XY plane, the roll and pitch angles are zero and orientation sensor will depend on the angle ψ with respect to the local coordinate system.

E. Appendix E

Stepper motor

In our case, the vehicle will carry the IMU and master antenna. To move the vehicle we will use two step by step engines. With them, we can be very accurate determining speed and orientation of the vehicle.

We will use stepper motors NEMA size 17; the stepping motor can be used as a unipolar or bipolar and has a step angle of 1.8 °(200 steps / rev). Our engine has the characteristic that each phase can support 1.2 A to 4 V, allowing a torque of holding 3.2 kg-cm (44 oz-in).

Figure 47. Stepper Motor

As we can control the step of each motor independently and accurately, in this case we do not need any encoder to know the position and velocity. To know how many mm correspond to one step, we can find out with the following formula:

$$step\ for\ mm\ = \frac{step\ of\ one\ revolution \cdot microsteps}{mm\ for\ revolution} \quad (108)$$

The motor speed can be calculated as follows:

$$v = \frac{steps\ for\ mm}{T}\ [^{mm}/_{s}] \quad (109)$$

T is the sampling rate. Keep in mind that the motor is connected to a wheel. It may be the case that when we advance a step, the wheel doesn't progress; this can be due to various factors such as friction with the ground or the engine speed.

F. Appendix F

Speed controller for stepper motors

To control the stepper motor will use Pololu A4988 board. In *Figure 49* we can see the image of the board and the operation scheme.

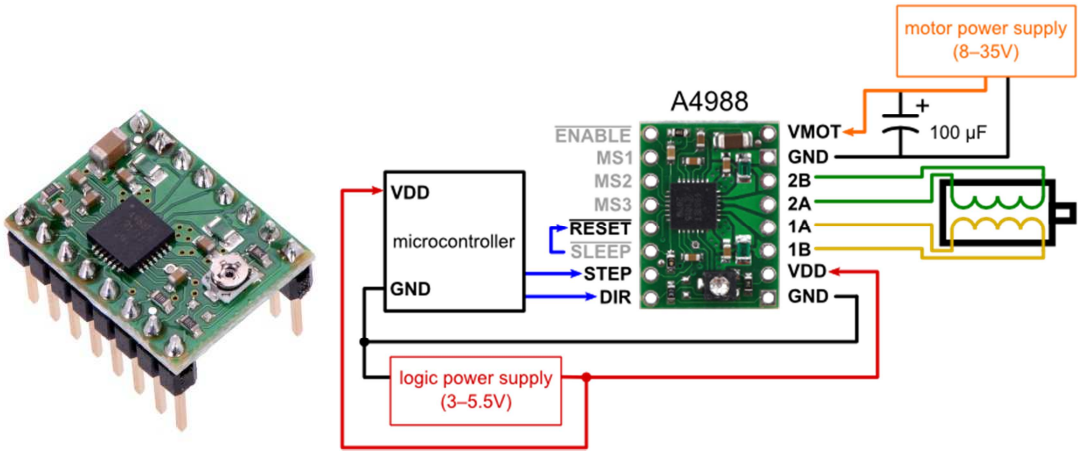


Figure 48. Pololu A4988 board and diagram of conexion

The circuit operation is very simple. The A4988 board has two pins called *STEP* and *DIR*. If we want the motor to move a step, we will send a pulse to *STEP* pin; if otherwise we want to change the direction, we will send a 0 or 1 logic state in the *DIR* pin to change the motor rotation.

The motor speed will depend on the rate at which we send pulses, at least we can send pulses of 1 microsecond.

Bibliography

[1] Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking: Monitoring and Surveillance Techniques for Target Tracking, **Guoqiang Mao** and **Baris Fidan**, 2009.

[2] *Mobile positioning based on relaying capability of mobile stations in hybrid wireless networks.* **Zhao, L., Yao, G. and Mark, J.W.** 5, 2006, IEE Proceedings on Communications, Vol. 153, pp. 762-770.

[3] Phase synchronization scheme for very long baseline coherent arrays, **Juan Carlos Merlano Duncan**, 2012, Master Tesis.

[4] Theoretical Study of an RF Positioning System Using Phase Synchronized Anchor Nodes, **Juan Carlos Rodriguez Silva**, 2011, Master Tesis.

[5] *Distance measurement with phase-stable CW radio link using the Chinese remainder theorem.* **Belostotski, L., Landecker, T.L. and Routledge, D.** 8, April 2001, Electronics Letters, Vol. 37, pp. 521-522.

[6] Phase unwrapping and a robust chinese remainder theorem. **Xiang-Gen Xia** and **Genyuan Wang**. Signal Processing Letters, IEEE, 14(4):247 –250, april 2007.

[7] Distance measurement with phase-stable CW radio link using the Chinese remainder theorem. **L. Belostotski, T.L. Landecker, and D. Routledge**. Electronics Letters, 37(8):521 –522, apr 2001.

[8] A technique for microwave ranging and remote phase synchronization. **Leonid Belostotski, T. L. Landecker, and D. Routledge.** IEEE Trans. on Instrumentation and Measurement., 51(3):551–559, June 2002.

[9] Navigation System for a Micro-UAV. **J. O. Nilsson**, Master thesis, KTH, Stockholm, Sweden, 2008.

[10] **M.S. Grewal, L.R. Weill**, and **A.P. Andrews**. Global Posi-tioning Systems, Inertial Navigation and Integration, Second Edition. Wiley, 2007.

[11] **Y. Zhao**. Vehicle Location and Navigation Systems. Artech House, 1997.

[12] **J. A. Farrell**, Aided Navigation. GPS with High Rate Sensors. McGraw-Hill, 2008.

[13] **A. De Angelis, J. Nilsson, I. Skog, P. H¨andel,** and **P. Carbone**, "Indoor Positioning by Ultrawide Band Radio Aided Inertial Navigation," Metrology and Measurement Systems, vol. 17, no. 3, pp. 447–460, 2010.

[14] Teaching Sensor Fusion and Kalman Filtering using a Smartphone, **Gustaf Hendeby, Fredrik Gustafsson and Niklas Wahlström,** Linköping University, 2014.

[15] **G. Welch and G. Bishop**, "An introduction to the kalman filter," UNC-Chapel Hill, Technical Report 95-041, July 24 2006.

[16] Optimal Filtering with Kalman Filters and Smoothers a Manual for the Matlab toolbox EKF/UKF, **Jouni Hartikainen, Arno Solin,** and **Simo Särkkä,** Aalto University School of Science, August 16, 2011.

[17] **Brian D. O. Anderson, John B. Moore**, (1979). Optimal Filtering. New York: Prentice Hall. pp. 129–133.

[18] **R. G. Brown and P. Y. C. Hwang**, Introduction to Random Signals and Applied Kalman Filtering, Second Edition. John Wiley & Sons, 1992.

[19] Structure from Motion using the Extended Kalman Filter, **Javier Civera, Andrew J. Davison, José María Martínez Montiel**,2012.

[20] The Extended Kalman Filter: An Interactive Tutorial for Non-Experts, http://home.wlu.edu/~levys/kalman_tutorial/

[21] SLAM-Course - 04 - Extended Kalman Filter, **Cyrill Stachniss**, https://youtu.be/DE6Jn2cB4J4

[22] Tracking and Sensor Data Fusion : Methodological Framework and Selected Applications, **Wolfgang Koch**, 2014.

[23] SLAM Course - 03 - Kalman Filter - **Cyrill Stachniss,** https://youtu.be/ELyQMGF_WIg

[24] SLAM Course - 06 - Unscented Kalman Filter, **Cyrill Stachniss,** https://youtu.be/DWDzmweTKsQ

[25] InvenSense – MPU6050, http://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/

[26] Elliott D. Kaplan, **Understanding GPS: Principles and Applications, Second Edition,** 2005

[27] Branko Ristic, Sanjeev Arulampalam, Neil Gordon, **Beyond the Kalman Filter: Particle Filters for Tracking Applications,** 2004

[28] Eli Brookner,**Tracking and Kalman Filtering Made Easy,** 1998