UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE UPV INGENIEROS DE TELECOMUNICACIÓN

ESTUDIO E IMPLEMENTACIÓN DE ALGORITMOS PARA LA ESTIMACIÓN DE LA POSICIÓN MEDIANTE SISTEMAS INERCIALES CON ARDUINO

Daniel Olivares Garcés

Tutor: Jorge Gosalbez Castillo

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2017-18

Valencia, 24 de junio de 2018

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
www.etsit.upv.es

VLC/ CAMPUS VALENCIA, INTERNATIONAL CAMPUS OF EXCELLENCE

EMAS Gestión ambiental verificada REG.NO. ES-CV-000030

Resumen

Este proyecto tiene como objetivo el estudio e implementación de un sistema de posicionamiento mediante sensores inerciales. El sistema, estará compuesto por una unidad de medición inercial (IMU), una plataforma Arduino para recoger, procesar y enviar las mediciones de la IMU y un ordenador para analizar y visualizar los resultados sobre Matlab.

A lo largo de este documento, separado en capítulos, serán abordadas las diferentes partes de este proyecto. En primer lugar, se plantearán los objetivos, a continuación, se explicará en que consiste un sistema de navegación inercial, así como algunos de sus conceptos fundamentales. Además, el cuarto capítulo estará dedicado a todo lo relacionado con el hardware (funcionamiento, elección, características etc.).

Una vez establecidos los objetivos y los elementos a utilizar, se explicará la implementación de las ecuaciones y el desarrollo experimental (calibración, algoritmos de filtrado, elección de pruebas etc.). Finalmente, a partir de los resultados obtenidos y el trabajo realizado se expondrá una conclusión y una propuesta de trabajo futuro.

Resum

Aquest projecte té com objectiu l'estudi i implementació d'un sistema de posicionament mitjançant sensors inercials. El sistema estarà compost per una unitat de medició inercial (IMU), una plataforma Arduino per a arreplegar, processar i enviar les mesures de la IMU i un ordinador per a analitzar i visualitzar els resultats sobre Matlab.

En aquest document, organitzat en capítols, es tractaran les diverses parts d'aquest projecte. Primerament, es plantejaran els objectius. A continuació, s'explicarà en que consisteix un sistema de navegació inercial, així com alguns dels seus conceptes fonamentals. Amés, el quart capítol estarà dedicat a tot el relacionat amb el hardware (funcionament, elecció, característiques, etc).

Una vegada establits els objectius i els elements a utilitzar, s'explicarà la implementació de les equacions i el desenvolupament experimental (calibratge, algoritmes de filtrat, elecció de proves, etc). Finalment, a partir dels resultats obtinguts i el treball realitzar s'exposarà una conclusió i una proposta de treball futur.

Abstract

This project has as objective the study and implementation of a positioning system using Inertial sensors. The system will consist in an Inertial Measurement Unit (IMU), an Arduino platform to obtain, process and send the measurements from the IMU and a computer to analyse and visualize the results by Matlab.

Along this document, separated in chapters, the different parts of the project will be exposed. Firstly, the objectives will be posed, after that, the idea of inertial navigation will be explained beside other fundamental concepts. Moreover, the fourth chapter contains a description of the hardware used.

After establish the objectives and elements to use, the implementation of the equations and the experimental development (calibration, filer algorithms, test etc.) will be explained. Finally, starting from the obtained results and work, a conclusion and a future work proposal will be exposed.

Agradecimientos

En primer lugar, a mi tutor Jorge Gosálbez Castillo por su dedicación y esfuerzo al darme la oportunidad de realizar este trabajo a distancia con las dificultades esto que conllevaba.

También a mi familia, por su apoyo incondicional durante toda mi etapa como estudiante, sin ellos habría sido imposible llegar hasta aquí.

Finalmente, al resto de profesores con los que he coincidido en esta escuela, por sus enseñanzas y por su disposición a ayudar en todo momento.

Índice

Capítulo 1. Introducción y objetivos.

Este proyecto, surge como necesidad de desarrollar un sistema de posicionamiento alternativo para un georradar que permita la medición y estudio de techos, bóvedas y otras estructuras pertenecientes al patrimonio construido. Teniendo en cuenta que se pretende medir elementos del patrimonio construido, el sistema ha de cumplir tres condiciones que limitan el uso de los sistemas de posicionamiento habituales (GPS y encoders mecánicos). Dichas condiciones son, precisión (<1 cm), posibilidad de uso en interiores y evitar contacto físico con la superficie bajo estudio. En la Tabla 1 se sintetiza las capacidades de los sistemas de posicionamiento tradicionales frente a las condiciones establecidas.

	GPS	**Encoder mecánico**
1. Precisión de centímetros	No	Si
2. Uso en interiores	No	Si
3. Evitar contacto físico	Si	No

Tabla 1. Condiciones que debe cumplir el sistema vs sistemas de posicionamiento tradicionales

Un georradar o radar de penetración, es un dispositivo que utiliza técnicas no destructivas para el análisis de materiales. El mecanismo está basado en la transmisión de ondas electromagnéticas de banda ultra ancha. Emitiendo una señal, esta se ve reflejada parcialmente cada vez que se alcanza la frontera entre dos materiales. Registrando la señal reflejada y analizándola, se puede obtener información sobre la composición y estado de la superficie a estudiar. Para ello, será imprescindible que el georradar cuente con un sistema de posicionamiento que registre la localización de cada medida.

Figura 1. A) Ejemplo de medida del nervio izquierdo del luneto 2 de la iglesia de San Nicolás de Valencia. B) Radargrama

Existen diversas aplicaciones para esta tecnología, estudio de suelos, estudio de hormigones, aplicaciones arqueológicas etc. En la mayoría de los casos la localización del dispositivo se realiza por GPS (espacios abiertos y grandes) o mediante dispositivos de contacto como encoders mecánicos (espacios más reducidos, Figura 1). Sin embargo, en este caso es necesario buscar una alternativa que cumpla los requisitos expuestos anteriormente.

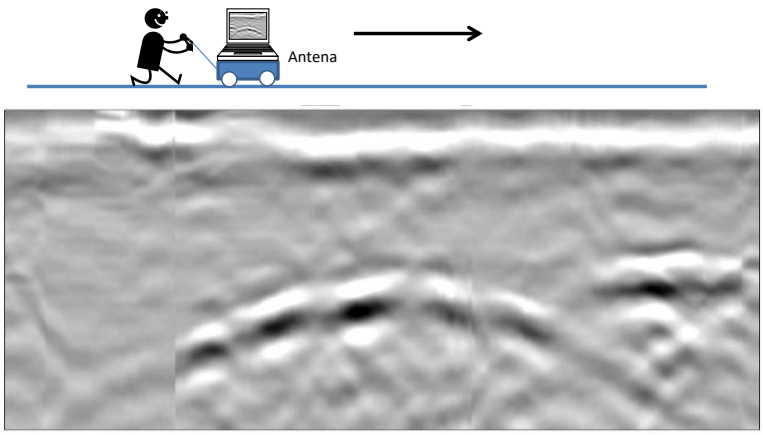

Figura 2. Representación de la toma de medida con un georradar con encoder de posición

Como solución al problema anterior, se propone la implementación de un sistema de navegación inercial de bajo coste mediante una IMU que cumpla las tres condiciones mostradas en Tabla 1 y proporcione las señales de desplazamiento que necesita el georradar para realizar la medida conforme la antena se desplaza. Una IMU (Inertial Measurement Unit) es un dispositivo electrónico que mediante la combinación de sensores inerciales (giroscopio y acelerómetro principalmente) informa de la velocidad, orientación y fuerzas gravitacionales experimentadas por un móvil. Mediante estos datos, se establecerá el sistema de navegación inercial.

Este trabajo se estructura de la siguiente forma. En este primer capítulo, se introduce el proyecto y se describen sus objetivos. En el siguiente capítulo, se realiza una introducción a la navegación inercial explicando los conceptos básicos necesarios para el desarrollo del proyecto. Los sistemas de coordenadas (navegación y body), ángulos de Euler y matriz de cambio de coordenadas serán los principales temas a abordar en este apartado. Acto seguido, en el Capítulo 3, se muestra el desarrollo teórico e implementación de las ecuaciones necesarias para la navegación, mientras que en el Capítulo 4, se explica todo lo referido al hardware utilizado, elección, funcionamiento, características etc. Como resultado de todo lo anterior, en el capítulo 5 se explica el desarrollo experimental, así como de las calibraciones, pruebas y métodos de filtrado utilizados. Finalmente, se muestran los resultados obtenidos, una conclusión de ellos y una propuesta de trabajo futuro.

Capítulo 2. Navegación inercial

En este capítulo son descritos los principios y conceptos básicos sobre navegación inercial necesarios para el desarrollo de este proyecto. Se comienza explicando los sistemas de coordenadas utilizados, así como varios conceptos necesarios para trabajar con ellos. Estos conceptos serán la matriz de cambio de coordenadas DCM y los ángulos de Euler. Finalmente, se indican las transformaciones necesarias para poder trabajar en un único sistema de referencia.

2.1 Conceptos básicos

Un sistema de navegación inercial o INS (Inertial Navigation System), es un sistema de ayuda a la navegación. Basa su funcionamiento en una IMU (Inertial Measurement Unit), una IMU, es una unidad electrónica que combina varios sensores, al menos un acelerómetro y un giroscopio (6 ejes). Mediante un procesado de las mediciones de los sensores, el sistema es capaz de calcular de forma continua una estimación de la posición, velocidad, actitud y orientación del móvil sin necesidad de una referencia externa. También existe la posibilidad de incorporar un magnetómetro, en ese caso se podría usar el norte magnético como orientación de referencia y el sistema pasaría a estar considerado de 9 ejes.

Para establecer un sistema de navegación inercial, es necesario además un sistema de referencia. Este puede ser definido como aquel sistema de coordenadas en el que las Leyes de Newton del movimiento se cumplen. Cabe destacar que los sistemas de referencia ni rotan ni aceleran.

A la hora de incorporar la IMU en el móvil existen dos formas posibles. En primer lugar, existe el sistema *gimbaled*, en el cual, la IMU se monta sobre una plataforma móvil con libertad de rotación en los tres ejes, dicha plataforma tiene la capacidad de rotar para aislar al sensor de las rotaciones externas.

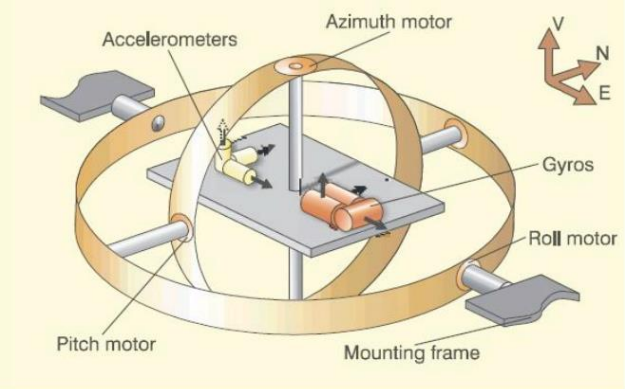

Figura 3. Plataforma gimbaled [8]

En segundo lugar, existe el sistema *strapdown*, en el cual los ejes de los sensores están alineados con los ejes del móvil. En este caso la capacidad de cálculo necesaria es mayor puesto que hay que compensar las rotaciones externas de forma matemática.

En este proyecto, se opta por el sistema *strapdown*. En primer lugar, por su implementación mucho más sencilla en lo relativo a espacio y a hardware, además, esta aplicación (posicionamiento de un georradar en superficies pequeñas) requiere tiempos y distancias de medida mucho menores que los habituales en un sistema de navegación (normalmente navegación de aviones), por lo que el posible error acumulado por rotaciones externas sería insignificante y fácilmente compensable.

2.2 Sistemas de coordenadas

Como ya se ha mencionado anteriormente, un sistema de coordenadas, no es más que un conjunto de valores que sirven para definir la posición de un punto en el espacio. Existen numerosos sistemas de coordenadas, sin embargo, para este proyecto, solo será necesario explicar dos.

2.2.1 Sistema de coordenadas de cuerpo (body)

El sistema de coordenadas de cuerpo, se define directamente sobre el vehículo, su origen, corresponde con el centro de masas del móvil, el eje x apunta hacia adelante, el eje y hacia la izquierda, y el eje z hacia arriba, siguiendo la regla de la mano derecha. Este sistema es el típico en plataformas strapdown, ya que los ejes se mueven al mismo tiempo y de igual forma que los sensores.

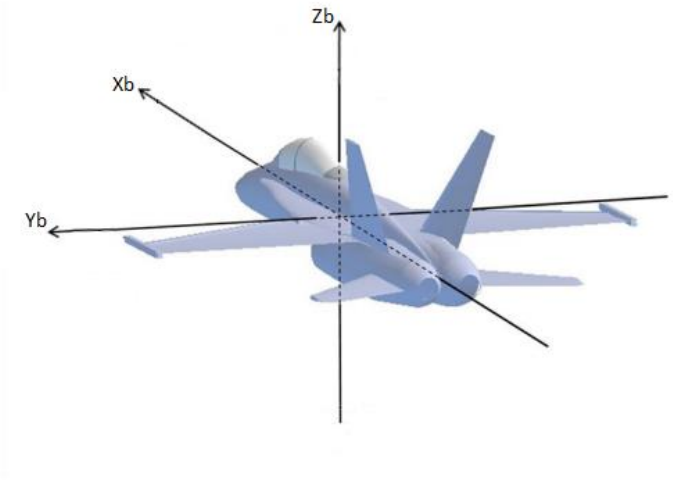

Figura 4. Coordenadas de body [11]

2.2.2 Sistema de coordenadas de Navegación

Este sistema tiene su origen en la posición inicial del móvil. Es un sistema local con los ejes XY colocados de tal manera que forman un plano tangente a la tierra en el origen. Típicamente, el eje X apuntará al norte, el Y hacia el Este y el Z hacia abajo, esta combinación se conoce como NED (North, East, Down) aunque no es necesariamente la única. Otra de las configuraciones es la conocida como ENU (East, North, Up), con el eje z apuntando hacia arriba. La principal diferencia respecto al sistema de body es que en este caso los ejes son fijos e independientes del desplazamiento del móvil.

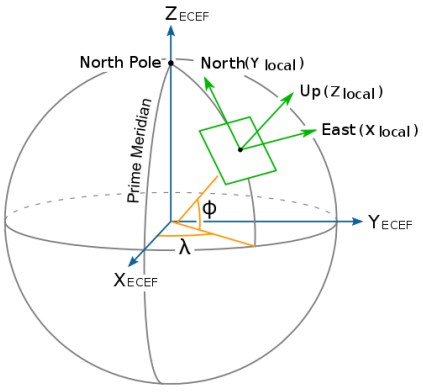

Figura 5. Coordenadas de navegación (ENU) [12]

2.3 Ángulos de Euler

Cualquier rotación de un cuerpo puede ser descrita mediante tres ángulos. Si estos ángulos corresponden al giro en cada uno de los ejes de un sistema de referencia ortogonal, se conocen como ángulos de Euler. Es decir, los ángulos de Euler pueden describirse como una sucesión ordenada de tres giros.

Los ángulos de Euler se identifican con los ángulos típicos en navegación de Roll (∅), Pitch(θ) y Yaw (ψ). Como se ha mencionado, los ángulos de Euler no están únicamente definidos, por tanto, es imprescindible aplicarlos siempre en el mismo orden.

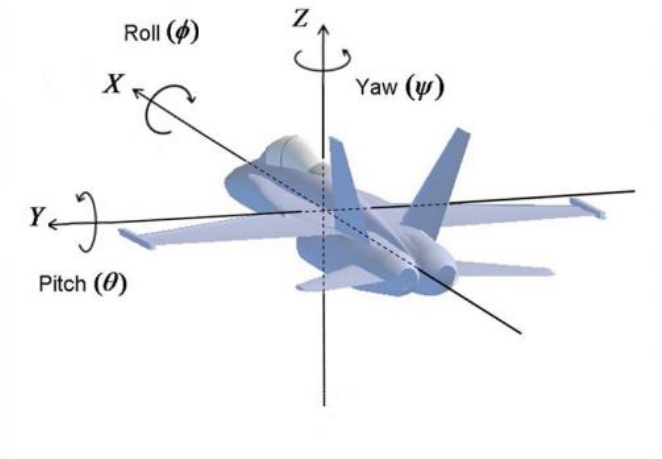

Figura 6. Ángulos de Euler [11]

2.4 Matriz de direcciones coseno

Hasta ahora, han sido descritos los ángulos de Euler y los dos sistemas de coordenadas a utilizar. Por tanto, la siguiente necesidad será un método que permita cambiar de un sistema de coordenadas a otro. Para ello es imprescindible explicar qué es una matriz de rotación.

Una matriz de rotación, en nuestro caso matriz de direcciones coseno (DCM), es un sistema de cambio de coordenadas capaz de trasladar un vector de un sistema de referencia a otro.

Pongamos por ejemplo el paso de un sistema *a*, a un sistema *b*, en primer lugar, se calcula la matriz de cambio de coordenadas C_a^b

$$C_a^b = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad \textbf{(2.1)}$$

Cada una de las componentes de la matriz corresponde a cada uno de los cosenos de los ángulos existentes entre los ejes de ambos sistemas de coordenadas. Una vez tengamos la matriz de rotación, para pasar de un sistema a otro, simplemente tendremos que multiplicar el vector por la propia matriz.

Cabe destacar un par de propiedades de la matriz de rotación, respecto a su determinante y a su transpuesta:

$$\left|C_a^b\right| = 1 \quad \textbf{(2.2)}$$

$$\left(C_a^b\right)^T = C_b^a \tag{2.3}$$

2.5 Cambio de coordenadas de cuerpo a navegación mediante ángulos de Euler

2.5.1 Giros de Yaw, Pich y Roll

Como se ha mencionado anteriormente, el orden de los giros es determinante en el resultado y por tanto ha de ser fijo. En este caso, el primer giro establecido, será el giro de Yaw, queda descrito de la siguiente manera:

$$C_\psi = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

A su vez, el giro de Pitch:

$$C_\theta = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \tag{2.5}$$

Y finalmente el giro de Roll:

$$C_\emptyset = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\emptyset & sin\emptyset \\ 0 & -sin\emptyset & cos\emptyset \end{bmatrix} \tag{2.6}$$

2.5.2 Matriz de cambio de coordenadas

Una vez definidos los tres giros y multiplicándolos entre sí, se obtiene la matriz de cambio de coordenadas de navegación a body.

$$C_n^b = C_\psi\, C_\theta\, C_\emptyset \tag{2.7}$$

$$C_n^b = \begin{bmatrix} cos\theta\, cos\psi & cos\theta\, sin\psi & -sin\theta \\ cos\psi\, sin\theta\, sin\emptyset - sin\psi\, cos\emptyset & cos\psi\, cos\emptyset + sin\psi\, sin\theta\, sin\emptyset & cos\theta \\ cos\psi\, sin\theta\, cos\emptyset + sin\psi\, sin\emptyset & sin\psi\, sin\theta\, cos\emptyset - cos\psi\, sin\emptyset & cos\theta \end{bmatrix} \tag{2.8}$$

Esta última ecuación se corresponde con la matriz de cambio de coordenadas de navegación a cuerpo. Sin embargo, la transformación necesitada en justo la inversa, por tanto, tal y como se ha mencionado antes, la matriz ha de ser traspuesta.

$$C_b^n = \left(C_n^b\right)^T \quad \textbf{(2.9)}$$

Nota: a partir de este momento, los subíndices *b* y *n* serán usados para referirse a coordenadas de body y navegación respectivamente.

Capítulo 3. Implementación de las ecuaciones

3.1 Estimación de Actitud

La actitud es el dato usado para determinar la orientación del vehículo respecto a los ejes lateral y longitudinal del sistema de referencia, permitiendo así conocer el rumbo de la navegación. Sus tres componentes son los ángulos de Euler:

$$E = [\phi\ \theta\ \psi]^T \tag{3.1}$$

3.1.1 Estimación de actitud en reposo

En primer lugar, es necesario calcular la actitud inicial, para ello se supone un estado de reposo donde se tienen en cuenta únicamente las mediciones del acelerómetro. Estudiando la descomposición de la fuerza de gravedad en cada uno de los ejes, son estimados los valores iniciales de pitch ($\emptyset_0$) y roll (θ_0) en grados, asumiendo que el yaw inicial Ψ_0 es igual a 0.

Sea a_0 la medición inicial del acelerómetro (en estado de reposo)

$$a_0 = [a_x\ a_y\ a_z] \tag{3.2}$$

Se normaliza respecto al valor de gravedad:

$$a_{0g} = \frac{a_0}{\sqrt{{a_x}^2 + {a_y}^2 + {a_z}^2}} \cdot 9.81 \tag{3.3}$$

Obteniendo:

$$\emptyset_0 = \text{atan}\left(\frac{a_{0gx}}{\sqrt{{a_{0gy}}^2 + {a_{0gz}}^2}}\right) \cdot \frac{180}{\pi} \tag{3.4}$$

$$\theta_0 = \text{atan}\left(\frac{a_{0gy}}{\sqrt{{a_{0gx}}^2 + {a_{0gz}}^2}}\right) \cdot \frac{180}{\pi} \tag{3.5}$$

3.1.2 Cambios de actitud a partir de los giroscopios

Una vez calculada la actitud inicial, se procede a actualizar la actitud en cada iteración, para ello se parte de la ecuación diferencial de los ángulos de Euler tal y como se explica en [3]:

$$E_{dif} = \begin{bmatrix} \emptyset' \\ \theta' \\ \psi' \end{bmatrix} = \begin{bmatrix} 1 & sin\emptyset \tan\theta & cos\emptyset \tan\theta \\ 0 & cos\emptyset & -sin\emptyset \\ 0 & sin\emptyset \sec\theta & cos\emptyset \sec\theta \end{bmatrix} \cdot \omega \quad \textbf{(3.6)}$$

E_{dif} corresponde a la variación de giro en coordenadas de body en cada una de las iteraciones , $\omega = [p\ q\ r]^T$ corresponde a las mediciones del giroscopio (velocidades angulares) y $[\emptyset'\ \theta'\ \psi']^T$es el vector de derivadas de los ángulos Euler.

A continuación, se procede a integrar los diferenciales de los ángulos de Euler, obteniendo así el vector de actitud E, para ello, se toma el valor de E en la iteración anterior y se le suma el diferencial calculado en la actual multiplicado por el incremento de tiempo entre iteraciones.

$$E[n] = E[n-1] + E_{dif}[n]\Delta t \quad \textbf{(3.7)}$$

3.2 Trayectoria

En este apartado, se establece un sistema de medición de desplazamiento empleando las medidas de los acelerómetros, sin embargo, estas medidas están afectadas por la gravedad y por la orientación del móvil, por tanto, hay que suprimir sus efectos.

Sea el vector de gravedad g, donde $g_x = g_y = 0$ y $g_z = g = 9.81\ m/s^2$:

$$g_n = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad \textbf{(3.8)}$$

Aplicando la matriz de cambio de coordenadas C_n^b:

$$g_b = C_n^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} -g\ sin\theta \\ g\ cos\theta \sin\emptyset \\ g\ cos\theta \cos\emptyset \end{bmatrix} \quad \textbf{(3.9)}$$

Y a_{Ib} la ecuación de aceleración teniendo en cuenta la gravedad

$$a_{Ib} = a_b - \omega \times v_b - g_b \quad \textbf{(3.10)}$$

Combinando ambas ecuaciones obtenemos:

$$\begin{aligned} a_{Ibx} &= a_{bx} + v_{by} \cdot r - v_{bz} \cdot q + g\ sin\theta \\ a_{Iby} &= a_{by} - v_{bx} \cdot r - v_{bz} \cdot p - g\ cos\theta \sin\emptyset \\ a_{Ibz} &= a_{bz} + v_{bx} \cdot q - v_{by} \cdot p - g\ cos\theta \sin\emptyset \end{aligned} \quad \textbf{(3.11)}$$

Dónde:

$a_{Ibx/y/z}$: Aceleración en coordenadas de body una vez suprimidos los efectos de la gravedad.

$v_{bx/y/z}$: Velocidad en coordenadas de body en cada uno de los ejes.

p,q,r: Velocidad angular en cada uno de los ejes (vector ω).

$\emptyset$: Ángulo de roll.

θ: Ángulo de pitch.

g: Módulo del vector de gravedad.

Mediante estas ecuaciones se obtiene el vector de aceleraciones en coordenadas de body, por lo que será necesario un cambio de coordenadas. Una vez aplicado el cambio, se obtiene el vector de aceleraciones en coordenadas de navegación.

$$a_n = C_b^n a_{Ib} \tag{3.12}$$

Tras los pasos anteriores y con el fin de calcular el desplazamiento del cuerpo, se realiza una doble integración, de la primera, se obtiene la velocidad, mientras que de la segunda el desplazamiento. Ambos valores en coordenadas de navegación.

$$v_n[n] = v_n[n-1] + a_n[n]\Delta t \tag{3.13}$$

$$p_n[n] = p_n[n-1] + v_n[n]\Delta t \tag{3.14}$$

Capítulo 4. Hardware utilizado

En este capítulo se realiza una descripción de los elementos de hardware usados. El conjunto consta de una IMU para las mediciones de aceleración y velocidad angular y de un microcontrolador para la lectura de los datos, procesamiento y su transmisión al ordenador mediante puerto serie. En este caso se ha seleccionado la IMU MPU6050 GY-521 y Arduino DUE como microcontrolador.

4.1 Arduino due

En este proyecto, el microcontrolador elegido es Arduino Due, esta placa de desarrollo está basada en un procesador de 32 bit CortexM3 ARM con gran poder de procesamiento respecto a otros arduinos y placas de prototipado.

A nivel técnico, Arduino Due consta de 54 pines digitales de entrada y salida (12 utilizables como salidas PWM), 12 entradas analógicas, 4 UARTs (puertas seriales por hardware), reloj de 84 MHz, conexión comptainle con USB-OTG, 2 TWI, Jack de alimentación, 2 TWI, conexión JTAG, botón de reset y botón de borrado. Cabe destacar que a diferencia de otras placas arduino, Arduino Due trabaja con 3,3 V en lugar de los 5 habituales, lo que puede suponer algún problema de incompatibilidad.

La elección de esta placa en concreto se debe a dos motivos, el primero de ellos la capacidad de procesamiento, otros formatos de Arduino tienen procesadores menos potentes y podrían ser insuficientes para este proyecto. El otro motivo es la capacidad de comunicación mediante protocolo I2C, lo que simplifica la conexión con la IMU.

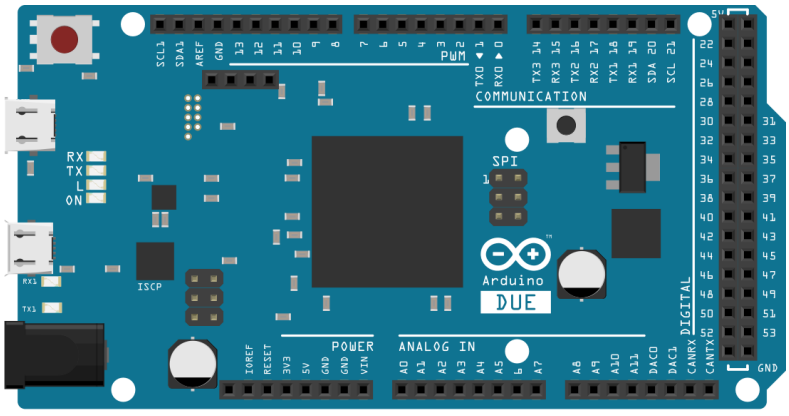

Figura 7. Arduino Due.

Características	
Microcontrolador	AT91SAM3X8E
Voltaje de operación	3,3 V.
Voltaje recomendado de entrada (pin Vin)	7-12 V.

Pines de entrada y salida digitales	54 pines I/O (12 PWM)
Pines de entrada análogos	12
Pines de salida análogos	2
Corriente de salida total en los pines I/O	130 mA.
Corriente DC máxima en el pin de 3.3V	800 mA.
Corriente DC máxima en el pin de 5V	800 mA.
Memoria Flash	512 KB
SRAM	96 KB
Velocidad de reloj	84 MHz.

Tabla 2. Resumen características técnicas

4.2 MPU6050 GY-521

La IMU elegida es la GY-521, basada en el chip MPU-6050. Esta IMU está compuesta por un acelerómetro y un giroscopio de tres ejes cada uno junto a una unidad de procesado digital interna. Los motivos de la elección de este dispositivo fueron su bajo precio, en torno a 2€ y su fácil implementación.

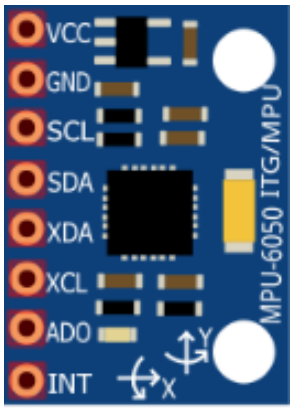

Figura 8. MPU-6050

Algunas de las características principales del dispositivo son 3,3 V de alimentación y unos rangos por defecto de +/- 250 º/s para el giroscopio y +/- 2g para el acelerómetro. Sin embargo, dichos rangos pueden ser también configurados a +/- 500, 1000 y 2000 en el caso del giroscopio y +/- 4, 8 y 16 g para el acelerómetro. En este proyecto, se usarán los rangos por defecto debido a que proporcionan una mayor sensibilidad, en concreto 131 LSB/(º/s) y 16.384 LSB/g respectivamente.

Finalmente cabe destacar que la conectividad del módulo se realiza mediante protocolo I2C, lo que simplifica bastante la conexión con el Arduino.

4.2.1 Acelerómetro

Dentro de la IMU, la función de este instrumento, es medir la aceleración absoluta en cada uno de los ejes. Para ello, consta de un sistema MEMS (Microelectromachanical System) cuyo funcionamiento queda muy bien resumido en la siguiente figura:

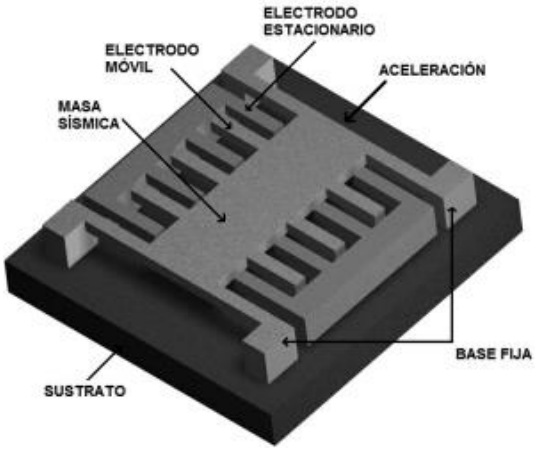

Figura 9. Funcionamiento acelerómetro [9]

Se pueden distinguir dos partes principales, ambas compuestas por placas metálicas, una de ellas se encuentra fijada al sustrato, mientras que la otra se encuentra suspendida. Debido a que no existe contacto físico entre las placas metálicas, se crea una capacidad. Al sufrir algún tipo de aceleración, la placa móvil se moverá en dirección opuesta respecto al sustrato, produciendo un cambio en la capacidad proporcional a la aceleración. Midiendo ese cambio y haciendo los cálculos pertinentes se obtiene la aceleración.

El problema que presentan este tipo de dispositivos es la precisión, a pesar de ser medidas absolutas y no acumular error, son muy sensibles al ruido, por lo que será necesario un filtrado de la señal antes de poder utilizarla.

4.2.2 Giroscopio

Un giroscopio o giróscopo, es un instrumento que mide la cantidad de rotación en cada uno de los ejes.

Existen diversos tipos de giroscopios (mecánicos, de anillo láser, de fibra óptica, etc.) en este caso, al igual que el acelerómetro se trata de un giroscopio MEMS también denominado como giroscopio vibratorio de efecto Coriolis (CVG).

El efecto Coriolis, se puede considerar como una fuerza ficticia que aparece sobre un cuerpo en movimiento cuando este se encuentra en un sistema en rotación. El funcionamiento del CVG se basa en eso, un objeto vibrante, tiende a vibrar en el mismo a plano, aunque este esté rotando. Sin embargo, por el efecto Coriolis, el objeto vibratorio al girar, efectuará una fuerza sobre su soporte, y midiendo esa fuerza, será posible determinar la cantidad de giro en ese plano.

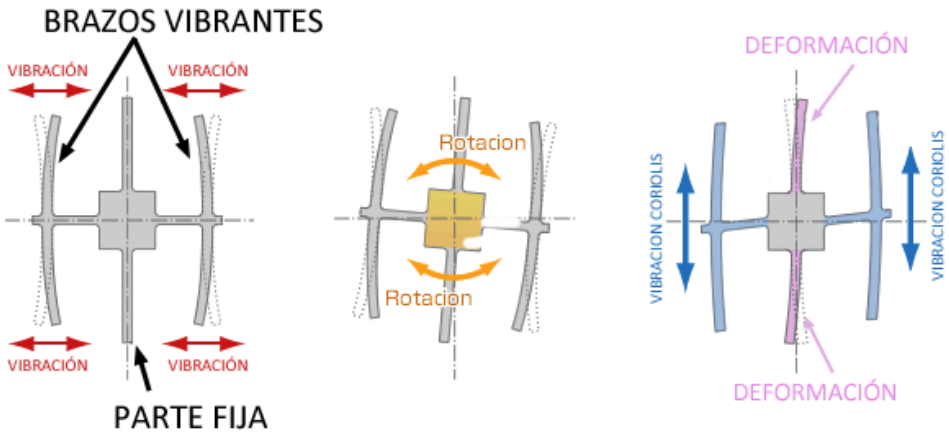

Figura 10. Funcionamiento del giroscopio [10]

A la hora de la medición, el mecanismo es parecido al del acelerómetro, ciertas partes de la estructura son sometidas a vibración, y debido al efecto de la fuerza de Coriolis son deformadas, produciendo así una variación de la capacidad que puede ser medida y traducida en un valor de velocidad angular. En este caso, al tratarse de un giroscopio de 3 ejes, la rotación será medida por separado en cada uno de ellos, siendo necesarias, por tanto, 3 estructuras como las mencionadas.

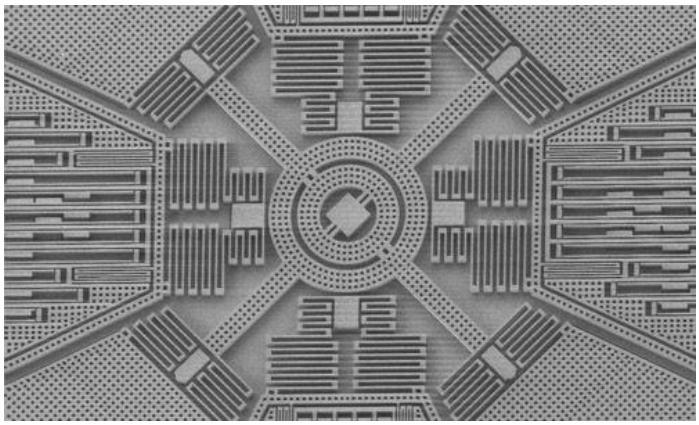

Figura 11. Detalle de la estructura de un giroscopio [10]

Cabe destacar, que este dispositivo no proporciona el ángulo total de giro, sino la velocidad angular instantánea, por lo que será necesario una integración para obtener la cantidad de giro.

El principal inconveniente que presenta este tipo de dispositivos es diferente al del acelerómetro. A corto plazo, son dispositivos muy precisos e inmunes al ruido, sin embargo, a medio y largo plazo presentan deriva en la medición, la cual deberá ser compensada.

4.3 Conexionado y montaje

El conexionado del conjunto será sencillo, en primer lugar, el pin Vcc del sensor se conecta al pin de alimentación 3,3 V del Arduino, el pin GND del sensor se conecta al GND del arduino, al igual que el pin AD0 del sensor, que se conecta a GND para referenciarlo a 0. Los pines de conexión SDA y SCL del sensor y del Arduino se conectan respectivamente. Finalmente, el pin INT del sensor se conecta con el PW2 del Arduino. Arduino y CPU estarán conectados por puerto serie mediante cable USB.

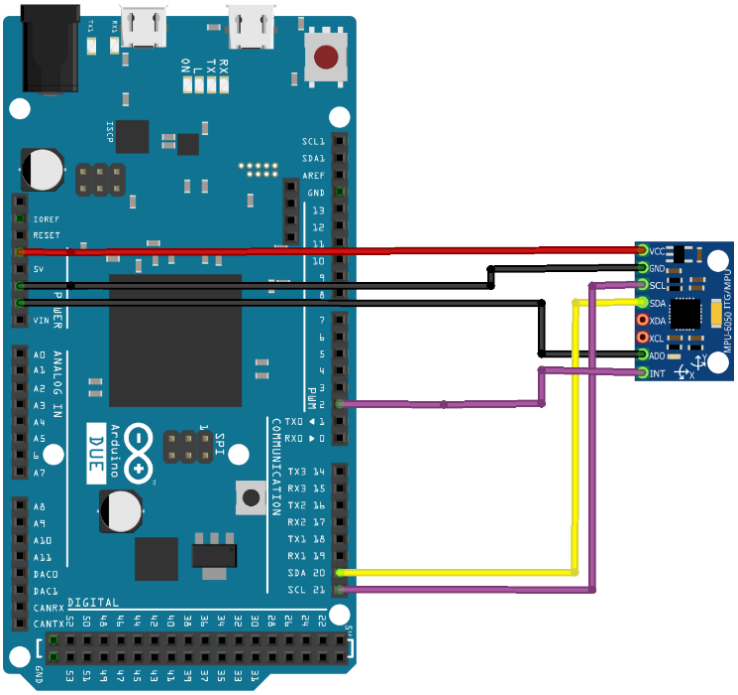

Figura 12. Conexionado Arduino Due-MPU 6050

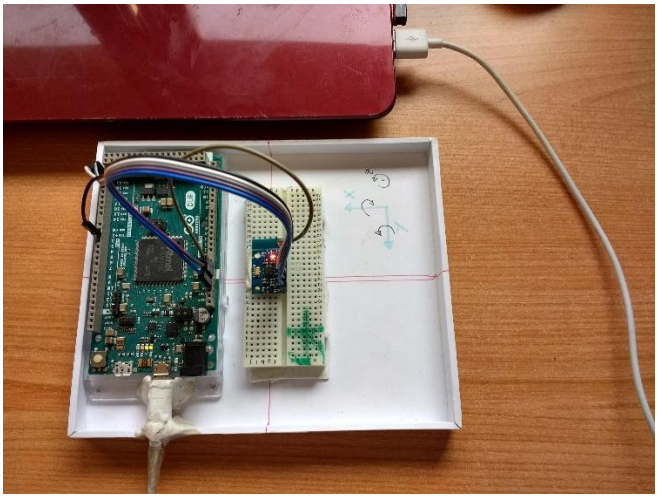

Figura 13. Montaje IMU-Arduino-CPU

Capítulo 5. Desarrollo experimental

5.1 Envío de datos y calibración en plataforma Arduino

Tal y como se ha explicado en el capítulo anterior, la lectura y envío de datos al ordenador es realizada mediante Arduino. Todo el código se implementa usando la propia aplicación de Arduino, a continuación, se muestran y explican las partes más significativas de este.

```
void loop()
{
   ta=millis();
   mpu.getAcceleration(&ax, &ay, &az);
   mpu.getRotation(&gx, &gy, &gz);
   // Coonversión a m/s2 y deg/s
   axl = ax*(9.81/16384);
   ayl = ay*(9.81/16384);
   azl = az*(9.81/16384);
   gxl = gx*((250.0/32768.0)/57.29578);
   gyl = gy*((250.0/32768.0)/57.29578);
   gzl = gz*((250.0/32768.0)/57.29578);

   tb=millis();
   dt = tb-ta;
   delay(50-(dt)); //Envio de datos cada 20 ms.
   printRAW();
}
```

Figura 14. Lectura y conversión de los datos del sensor

En primer lugar, se obtienen los valores del sensor utilizando las funciones .getAcceleration y .getRotation. de la librería MPU6050. A continuación, teniendo en cuenta los fondos de escala y los rangos por defecto del acelerómetro y giroscopio, se realiza la conversión de las mediciones a unidades internacionales (m/s^2 y º/s), en el caso de los º/s, estos serán a su vez transformados en rad/s. Finalmente, y para que el envío de datos sea constante, se establece un retardo de 50 ms menos el tiempo de procesado y se llama al método printRaw() para enviar los datos.

Aunque el envío de datos es sencillo y no plantea un gran problema, el sensor está afectado por unos errores de offset, estos pueden deberse a pequeños fallos de fabricación o a la dificultad para colocar el sensor completamente horizontal. Por tanto, será necesaria una calibración.

Para realizar dicha calibración, se utiliza un sketch de Arduino desarrollado por Luis Ródenas basado en la librería I2Cdev y un trabajo previo de Jeff Rowberg [13]. En este sketch se realizan una sucesión de medidas con el sensor colocado horizontalmente, al mismo tiempo se actualizan periódicamente los offset hasta obtener unas mediciones próximas a las ideales (0 en todas las componentes excepto la aceleración en el eje z que debe coincidir el valor de gravedad 9.81). Una vez finalizada la ejecución del sketch, este devuelve unos offset que serán introducidos en el sketch principal como se muestra en la siguiente figura.

```
void printRAW()
{
   Serial.print(axl); Serial.print(",");
   Serial.print(ayl); Serial.print(",");
   Serial.print(azl); Serial.print(",");
   Serial.print(gxl); Serial.print(",");
   Serial.print(gyl); Serial.print(",");
   Serial.println(gzl);
}

void setup()
{

   Wire.begin();
   Serial.begin(38400);
   mpu.initialize();
   // configuración de offset
  mpu.setXAccelOffset(-6138);
  mpu.setYAccelOffset(-387);
  mpu.setZAccelOffset(1625);
  mpu.setXGyroOffset(58);
  mpu.setYGyroOffset(57);
  mpu.setZGyroOffset(87);
```

Figura 15. Método printRaw() y configuración de offset

En esta figura, ax1, ay1, y az1, se corresponden con los valores de aceleración en m/s^2 en cada uno de los ejes, mientras que gx1, gy1 y gz1 con cada una de las velocidades angulares en rad/s. El método .setAcceloffset es el usado para la calibración de offset utilizando los valores obtenidos en el sketch de Luis Ródenas.

5.2 Estimación de actitud

Una vez enviados los datos mediante puerto serie, estos son leídos por Matlab para ser procesados e interpretados.

5.2.1 Estimación inicial de actitud

Como ya se explicó en el capítulo 3, debido a la aplicación del georradar, la posición inicial de este no es necesariamente horizontal. Por tanto, es necesario un método que permita calcular la actitud inicial del dispositivo. Para ello, se utilizan únicamente las medidas del acelerómetro en reposo.

En estado de reposo, se obtienen y almacenan 80 mediciones en un periodo de 4 s, a continuación, se realiza un promediado de la aceleración en cada uno de los ejes y aplicando (3.3) y (3.4), son obtenidos los ángulos de pitch y roll respecto al suelo, suponiendo un yaw de 0º. El código en Matlab es el siguiente:

```
while d<80
    nothing = fscanf(arduino,'%f,%f,%f,%f,%f,%f')';
    estim(1,d)=nothing(1);
    estim(2,d)=nothing(2);
    estim(3,d)=nothing(3);
    d=d+1;
   % disp('Estimando posicion')
end
%Estimación de la actitud inicial
    a_estim(1,1)=mean(estim(1,:));
    a_estim(2,1)=mean(estim(2,:));
    a_estim(3,1)=mean(estim(3,:));
    a_estim = a_estim/sqrt(sum(a_estim.^2))*9.81; %Normalizar respecto a 9.81
    auxX = a_estim(1);
    auxY = a_estim(2);
    auxZ = a_estim(3);

    theta = -atan(auxX/sqrt(auxY^2+auxZ^2));%Pitch
    phi   = atan(auxY/sqrt(auxX^2+auxZ^2));%Roll
    E(:,n) = [phi; theta; 0];
    E_acum(:,n)=E(:,n);
```

Figura 16. Estimación uncial de actitud

En esta parte del script, el vector *nothing* es la lectura del sensor en cada una de las iteraciones. Los valores de aceleración, que se corresponden con los tres primeros elementos del vector son almacenados en *estim*, a continuación, se realizan el promediado y los cálculos mencionados, obteniendo así los ángulos buscados de pitch y roll.

5.2.2 Estimación de actitud

La estimación de actitud una vez comenzada la medición, se realiza de forma diferente. En primer lugar, se crea el método matrizRotacion que implementa la matriz de cambio de coordenadas (2.8).

```
function R = matrizRotacion(thetaX,thetaY,thetaZ)
if nargin == 1
    thetaZ = thetaX(3);
    thetaY = thetaX(2);
    thetaX = thetaX(1);
end
% Matrices de rotación Roll,pich y yaw
Rx = [1             0               0;
      0             cos(thetaX)     -sin(thetaX);
      0             sin(thetaX)     cos(thetaX)];

Ry = [cos(thetaY)    0              sin(thetaY);
      0             1               0;
      -sin(thetaY)  0               cos(thetaY)];

Rz = [cos(thetaZ)    -sin(thetaZ)     0;
      sin(thetaZ)     cos(thetaZ)     0;
      0               0               1];
 R = Rx*Ry*Rz;
```

Figura 17. Matriz de rotación

En el siguiente paso, se declaran las funciones C y C2. C corresponde a un cambio de coordenadas del vector de actitud E, aplicando la matriz de rotación (2.8). C2 equivale a la implementación de la ecuación diferencial de los ángulos de Euler (3.5). Tras esto, se procede a realizar el resto de cálculos, tal y como se muestra en la siguiente figura:

```
% Calculamos la derivada
E_dif = C2(E(1,n),E(2,n),E(3,n))*w(:,n);
% Calculamos los nuevos angulos
E(:,n) = E(:,n) + E_dif*dt;
E_acum(:,n+1)= E_acum(:,n)+E_dif*dt;
% Actualizamos la matriz de rotación con los nuevos angulos
C = matrizRotacion(E(1,n),E(2,n),E(3,n));
Cabs = matrizRotacion(E_acum(1,n),E_acum(2,n),E_acum(3,n));
```

Figura 18. Estimación de actitud

El primer paso es la obtención el incremento de velocidad angular en la iteración actual, para ello, se multiplica el vector de actitud transformado por la medición del giroscopio (w). A partir de ese incremento de velocidad angular se calcula el incremento de cada ángulo, finalmente, se almacena y se actualiza la matriz de rotación con los nuevos ángulos.

5.2.3 Pruebas y Filtro complementario

Para comprobar el correcto funcionamiento a tiempo real, se estableció una gráfica tridimensional en la que la actitud del móvil (representado como un triángulo) era actualizada cada iteración. Los resultados obtenidos fueron los siguientes:

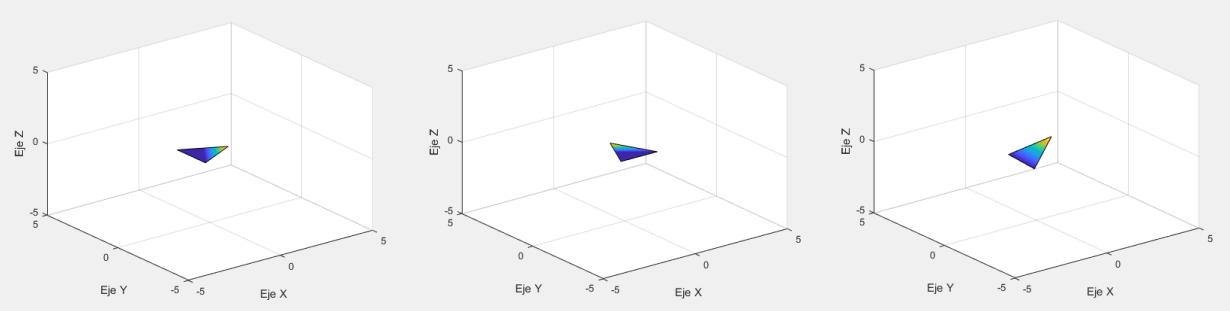

Figura 19. Pruebas de rotación

De izquierda a derecha pueden observarse posición inicial, giro en yaw de 90º y giro en pich de 45º. Aunque la imagen muestra solo una pequeña muestra de las pruebas realizadas, observando el resto de pruebas y los valores numéricos de cada una ellas, pudo comprobarse que el método era bastante preciso.

Sin embargo, a pesar de la precisión a corto plazo, al realizar mediciones más largas se puedo apreciar una deriva o *drift* en las medidas. Para eliminar este error, se decidió implementar un filtro complementario. Este filtro, calcula el ángulo de giro mediante dos formas diferentes para luego combinarlos. Aprovechando así la precisión a corto plazo del giroscopio y la ventaja de no acumular error del acelerómetro.

```
%Angulos calculados a partir del giroscopio
        gx(n)=gx(n-1)+g0x(n)*dt;
        gy(n)=gy(n-1)+g0y(n)*dt;
%Angulos de roll y pitch estimados mediante acelerometro
        cgx(n)= atan(a0y(n)/(sqrt(a0x(n).^2+a0z(n).^2)));
      cgy(n)= atan(-a0x(n)/(sqrt(a0y(n).^2+a0z(n).^2)));
%Filtro Complementario
        fgx(n)  = 0.98*(fgx(n-1)+g0x(n)*dt)  + 0.02*cgx(n);
       fgy(n)  = 0.98*(fgy(n-1)+g0y(n)*dt)  + 0.02*cgy(n);
    t2= toc;
    pause(0.05-t2);
    n=n+1;
end
```

Figura 20. Implementación del filtro complementario

En primer lugar, se estiman los ángulos (gx, gy) integrando las mediciones del giroscopio (g0x, g0y), a continuación, se calculan los mismos ángulos mediante el acelerómetro usando (3.3) y (3.4). Finalmente, según se muestra en (5.1), se combinan ambas medidas. El coeficiente *a* puede variar entre 0 y 1 según la confianza en cada uno de los sensores, en este caso, se ha optado por 0,98 para el giroscopio y 0,02 para el acelerómetro, dichos valores serán justificados más adelante.

$$Angulo(n) = a * \left(angulo(n-1) + angulo_{giro}(n) * dt\right) + (1-a) * angulo\_acel(n) \tag{5.1}$$

Una vez implementado, se realizaron diversas pruebas que verificaron su correcto funcionamiento, a continuación, se muestra una de ellas:

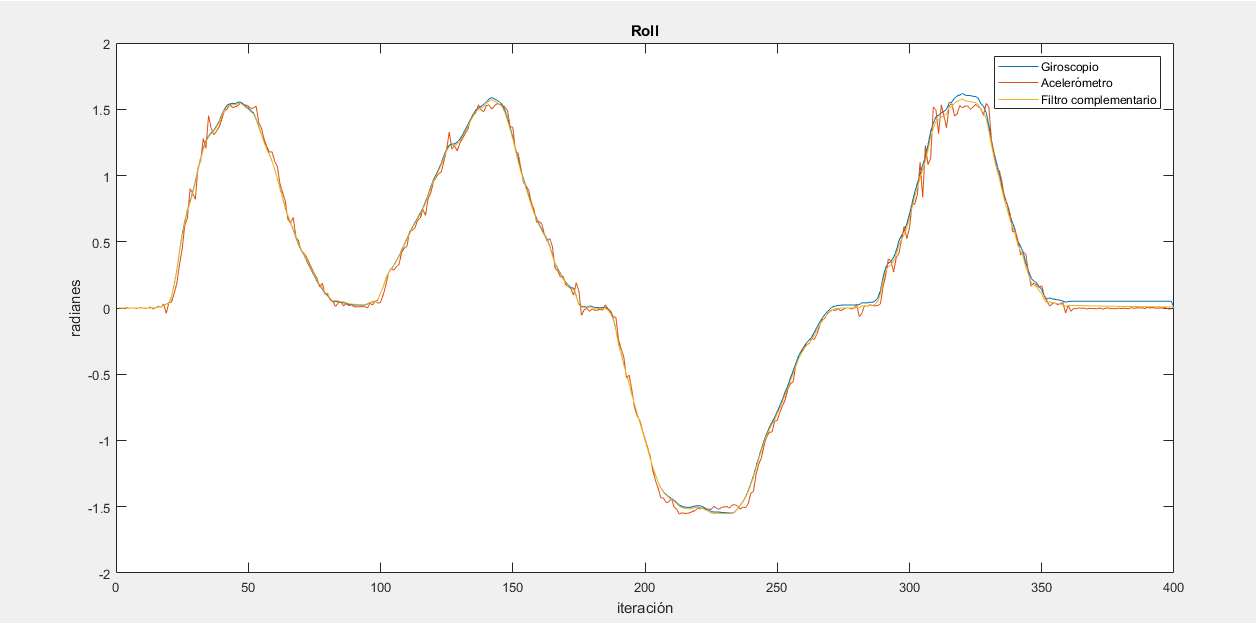

Figura 21. Ángulo de roll calculado por diferentes métodos

En la figura 19, puede apreciarse la gráfica que representa el ángulo de roll tras varios giros de 90º. En azul se muestra la estimación a partir del giroscopio, a priori, mucho más precisa que la obtenida con el acelerómetro, mostrada en color naranja. Sin embargo, es necesario fijarse en las últimas medidas para apreciar el efecto del filtro complementario.

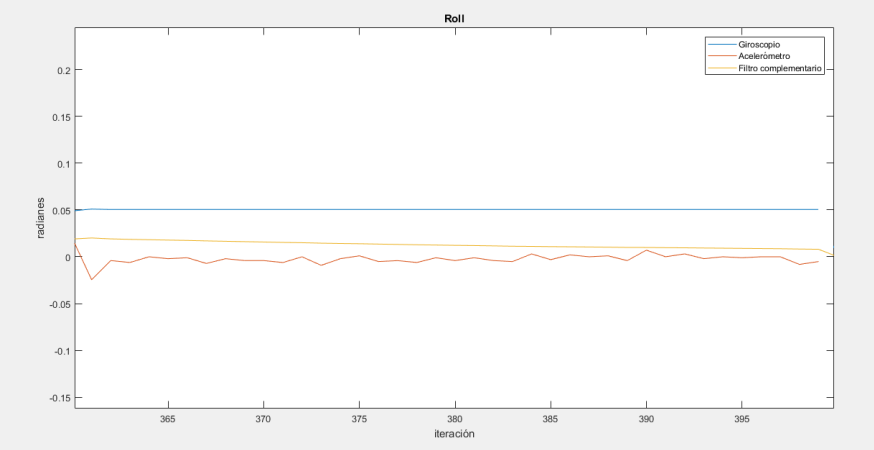

Figura 22. Detalle del ángulo de roll

Mostrando en pantalla únicamente las últimas iteraciones, en las cuales el sensor ya ha vuelto a su posición inicial de 0º, se observa que la estimación del giroscopio no vuelve a 0, a diferencia de la del acelerómetro o de la señal filtrada (en color amarillo). Como puede comprobarse, la señal filtrada mantiene la forma de la estimación del giroscopio en todo en el recorrido, pero sin presentar deriva a largo plazo.

Además, la figura 22 puede ser utilizada para explicar el valor del coeficiente a. Como puede comprobarse, una vez el sensor vuelve a estado de reposo, la señal filtrada tarda un poco en volver a converger a 0, esto se debe a que la medición del acelerómetro tiene muy poco peso frente a la del giroscopio 0.02 a 0.98. Aumentando el peso del acelerómetro la convergencia sería más rápida, sin embargo, también aumentaría el ruido en la señal debido al ruido del acelerómetro. Por tanto, tras varias pruebas se determinó que 0.98 y 0.02 eran unos valores equilibrados para un buen filtrado.

Las únicas limitaciones que presenta este filtro son la incapacidad para calcular la rotación del eje z (yaw) así como ángulos más allá de +-90º en los ejes x(roll) e y(pitch).

5.3 Estimación de posición

5.3.1 Implementación del modelo de navegación

Partiendo del cálculo de actitud anterior, el modelo de navegación se implementa de la siguiente manera:

```
%Componentes del Vector de acelarciones en body
        abi(1,n) = ab(1,n) + vb(2,n)*w(3,n) - vb(3,n)*w(2,n) + g*sin(E_acum(2,n));
        abi(2,n) = ab(2,n) - vb(1,n)*w(3,n) + vb(3,n)*w(1,n) - g*cos(E_acum(2,n))*sin(E_acum(1,n));
        abi(3,n) = ab(3,n) + vb(1,n)*w(2,n) - vb(2,n)*w(1,n) - g*cos(E_acum(2,n))*cos(E_acum(1,n));
        %Velocidad body
        vb(:,(n+1))=vb(:,n)+abi(:,n)*dt;
        %Paso de Aceleraciones de body a navegación
        an=(Cabs)*abi;
        %Velocidad de navegación
        if abs(norm(ab(:,n))-9.81)<0.1 %Mantener velocidad en caso de no aceleración
            vn(:,n+1)=vn(:,n);
        else
        vn(:,n+1) = vn(:,n) + an(:,n)*dt;
        end
        %Posición en coordenadas de navegación
        pn(:,n+1) = pn(:,n) + vn(:,n)*dt;
    t2= toc;
    pause(0.05-t2);
    n=n+1;
end
```

Figura 23. Estimación de posición

Dónde:

abi: Aceleración absoluta en coordenadas de body una vez compensados los efectos de la gravedad.

ab: Aceleración medida por el acelerómetro.

An: Aceleración del móvil en coordenadas de navegación.

vb: Velocidad del móvil en coordenadas de body.

vn: Velocidad del móvil en coordenadas de navegación.

Pn: Posición del móvil en coordenadas de navegación.

w(x,n): Velocidad medida por el giroscopio.

E_acum: Vector de actitud acumulada (ángulos de Euler).

Cabs: Matriz de cambio de coordenadas.

g: Módulo del vector de gravedad.

Las tres primeras líneas corresponden a (3.10), de donde se obtiene la aceleración del cuerpo en coordenadas de body. Derivando esas aceleraciones se consigue la velocidad del móvil en el mismo sistema de coordenadas, necesario a su vez para calcular *abi* en la siguiente iteración.

Una vez se tiene la aceleración compensada en coordenadas de body *abi*, se multiplica por la matriz de cambio de coordenadas obteniendo las aceleraciones en coordenadas de navegación *an*.

Finalmente, mediante una doble integral de *an* se obtiene la posición *pn* en cada una de las iteraciones. Como paso intermedio, se comprueba si el módulo del vector de aceleraciones medidas es próximo a 9.81, en ese caso, se entiende que no ha habido un cambio significativo en la aceleración y por tanto no se actualiza la velocidad.

5.3.2 Pruebas y resultados

Antes de probar el sistema completo con los datos del sensor, se estableció un banco de pruebas en Matlab. Introduciendo valores simulados de aceleración y giro, pudo comprobarse que el sistema funcionaba perfectamente.

Después de la verificación con mediciones simuladas, se procedió a realizar diversas pruebas con el sensor. Una de las pruebas efectuadas consistió en recorrer con el sensor en posición horizontal una línea de 1,5 m en dirección *x-y* positiva, por tanto, al final de la medición, debería haberse recorrido aproximadamente 1.06 m en *x* y en *y*, sin embargo, tras otras mediciones y al igual que en el resto de pruebas los resultados no fueron los deseados.

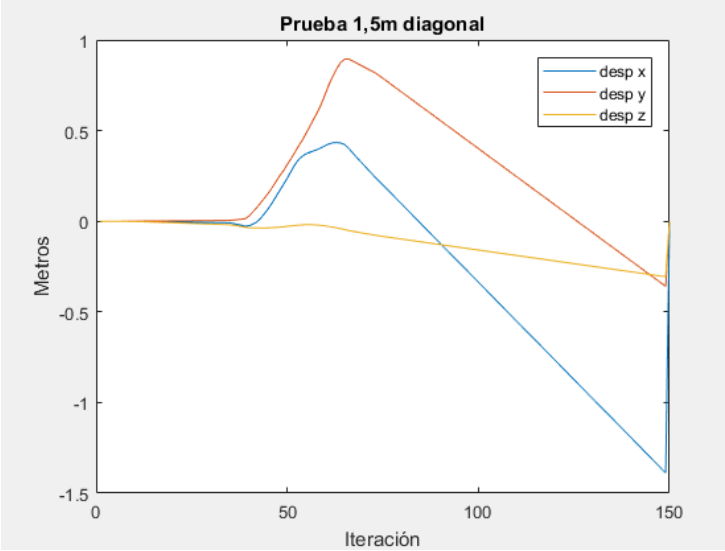

Figura 24. Prueba de desplazamiento

En la figura anterior, se muestra el desplazamiento en cada uno de los ejes tras un movimiento de 1.5 m en diagonal y sin giros. En el caso ideal, tanto la línea azul como la naranja deberían haber alcanzado un valor de 1.06 m y mantenerse, sin embargo, el resultado final es bien diferente.

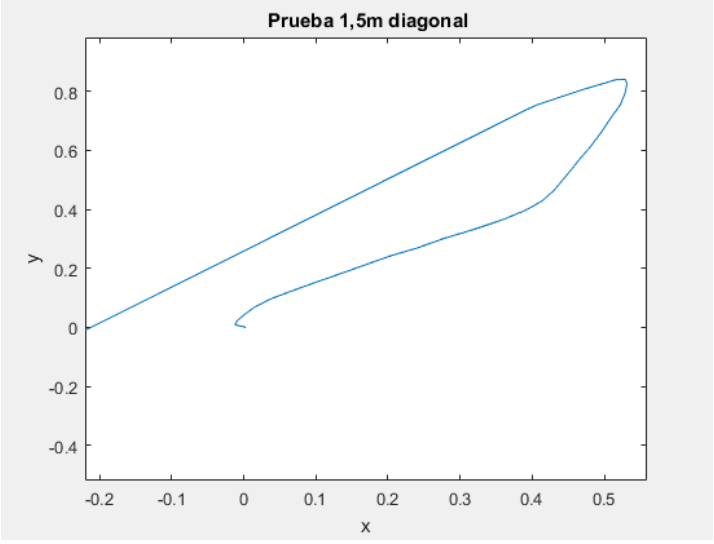

Figura 25. Prueba de desplazamiento en plano x-y

Si se observa ese mismo desplazamiento en el plano en el plano x-y se aprecia que al principio es más o menos correcto, sin embargo, el alto ruido estropea rápidamente la medición. Además, como los valores de velocidad y posición dependen del valor anterior, el error se va acumulando invalidando por completo la medida tras unas pocas iteraciones.

Tras los malos resultados, se volvió a calibrar el dispositivo y se revisó todo el proceso, no obstante, no hubo ninguna mejora. Analizando cada dato por separado, se encontró que el error provenía de la medida del acelerómetro.

Con la intención de simplificar el problema, se estableció una nueva prueba en la que se recorría un metro en línea recta, teniendo en cuenta únicamente el desplazamiento en x. Además, para tratar de eliminar el ruido, se repitió la misma prueba usando diferentes filtros. Los resultados fueron los siguientes:

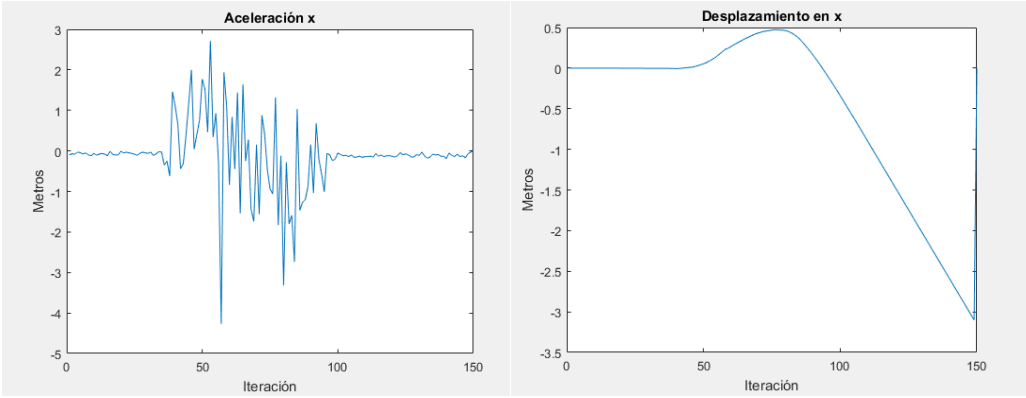

Figura 26. A) Medición de aceleración sin filtrar B) Desplazamiento correspondiente

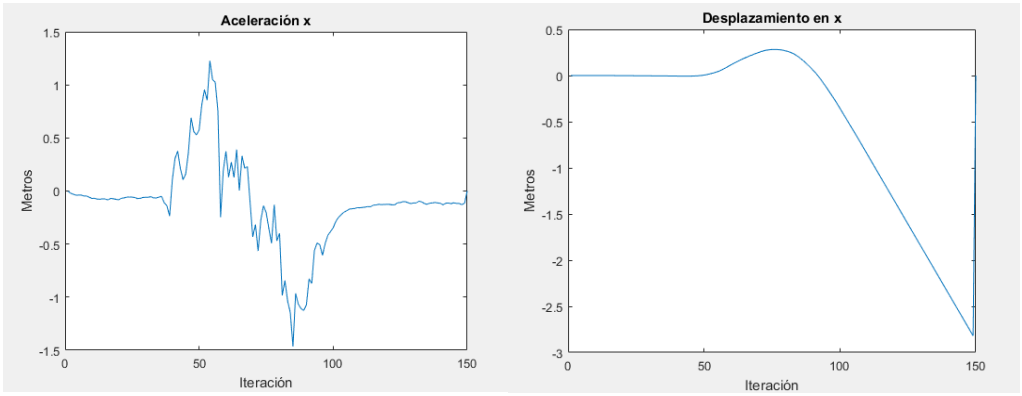

Figura 27. A) Medición de aceleración con filtro paso bajo B) Desplazamiento correspondiente

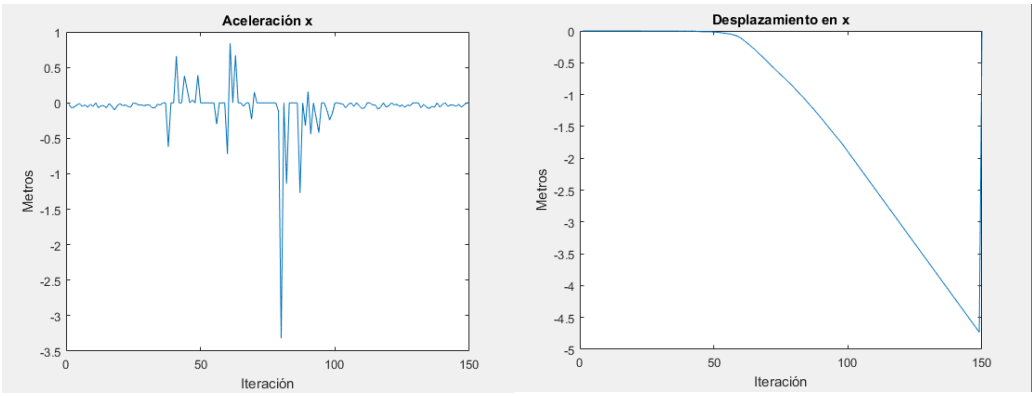

Figura 28. A) Medición de aceleración con filtro de mediana B) Desplazamiento correspondiente

Las tres gráficas situadas a la izquierda corresponden a la misma medición de aceleración, la primera de ellas sin filtrar, la segunda aplicando un filtro paso bajo y por último un filtro de mediana. Al lado derecho, se muestran los desplazamientos correspondientes a cada aceleración.

Tras diversas pruebas, el único filtro que mejoraba algo era el paso bajo, no obstante, a pesar de optimizar sus coeficientes la mejora era pequeña e incluso en ocasiones empeoraba el resultado, como se comprueba en la figura 24.

A modo de prueba final y con el fin de demostrar que la precisión del sensor era insuficiente para mediciones pequeñas pero el sistema funcionaria perfectamente en otro tipo de aplicación, se colocó el sensor en un coche, el cuál recorrería 25m hacia adelante y posteriormente hacia atrás, el resultado fue el siguiente:

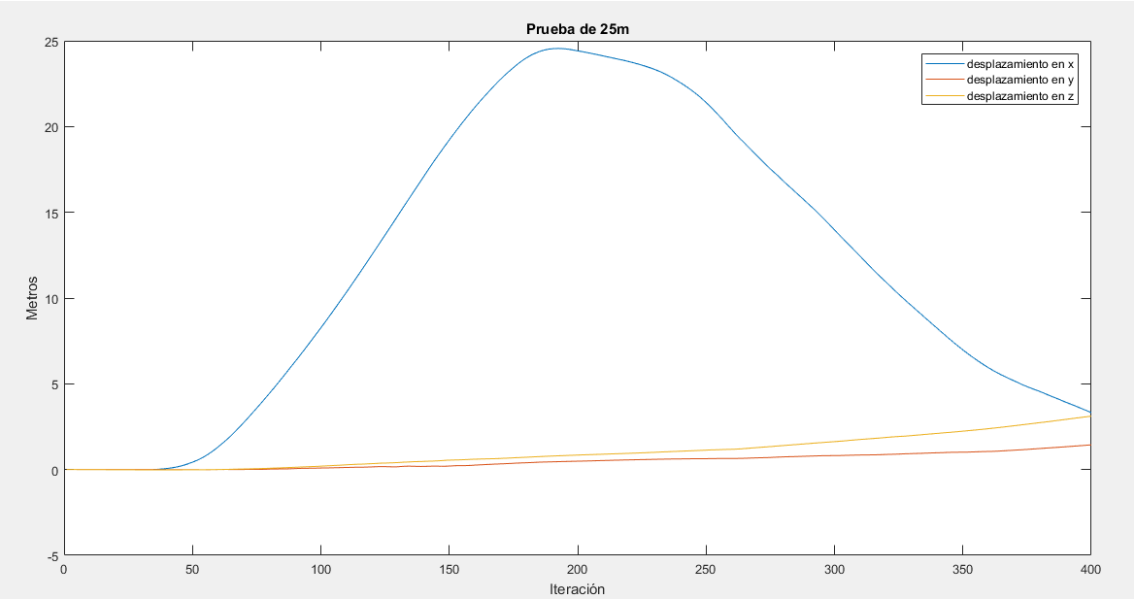

Figura 29. Desplazamiento en cada eje tras prueba de 25 m.

Como puede comprobarse, aunque el resultado no es perfecto si es muy similar a la distancia recorrida real. Además, si se tienen en cuenta la ausencia de filtrado y las imperfecciones de la superficie donde se realizó la prueba, se demuestra que el sistema funciona correctamente para distancias grandes. No obstante, al no ser útil para el objetivo del proyecto no se trabajó más este tipo de mediciones.

Capítulo 6. Conclusiones y propuesta de trabajo futuro

6.1 Conclusiones

Tras el desarrollo de este proyecto se alcanzaron las siguientes conclusiones:

- El sistema de navegación inercial implementado funciona correctamente aplicando mediciones simuladas. Además, realizando mediciones más largas, de al menos 5 metros, el sistema también muestra un funcionamiento correcto.
- En distancias cortas y por tanto aceleraciones bajas, la precisión del acelerómetro es insuficiente para obtener medidas precisas, por tanto, el sistema no es válido para la aplicación inicial de georradar buscada.

Como conclusión final se podría decir que se ha implementado un sistema de navegación inercial que podría ser usado en otras aplicaciones. Sin embargo, para una aplicación de georradar como la de este proyecto no es una alternativa válida.

6.2 Propuesta de trabajo futuro

A modo de continuar este proyecto existen varias opciones posibles. Una de ellas podría ser la sustitución del acelerómetro por uno o incluso varios de mayor precisión. Obteniendo así una señal lo suficientemente limpia como para poder eliminar el ruido y obtener mediciones precisas.

Otra forma podría ser la implementación de algún filtro más complejo como por ejemplo el filtro de Kalman. No obstante, debido a la gran imprecisión de las medidas es posible que el ruido de este sensor sea demasiado grande para poder ser compensado. Además, como en este proyecto se buscaba una implementación de bajo coste, no se contempló el uso de sensores de mayor precisión y, por tanto, precio.

Finalmente, en caso de obtener unos resultados suficientemente precisos con las propuestas anteriores, se debería proceder a lo que inicialmente era el objetivo principal de este proyecto.

El sistema de posicionamiento más usado en georradares como el de este proyecto es el encoder mecánico. Aunque hay diferentes tipos, a modo general se puede definir como un dispositivo electromecánico que define la cantidad de giro en su eje mediante varias señales de pulsos. Lo más habitual es que el encoder sea de tipo incremental, que proporciona un número determinado de valores de contaje por giro, así como un impulso de puesta a 0 una vez por giro.

Dentro de los encoder incrementales existen varios tipos, uno de los más frecuentes es el de dos canales, en el que el sentido de giro se indica en función del orden de desfase que existe entre los canales A y B, como se muestra en la figura siguiente.

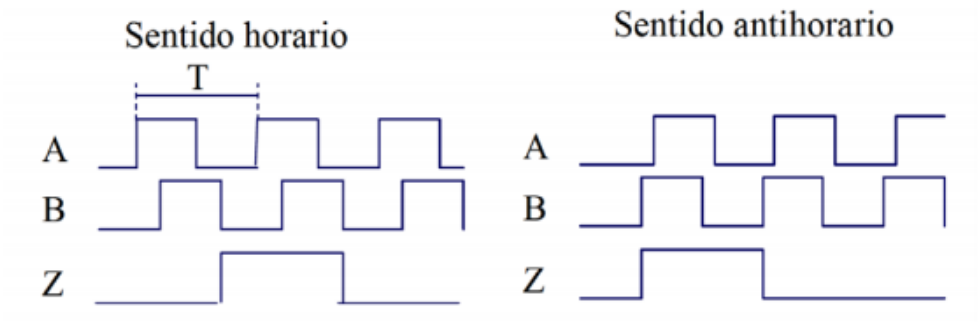

Figura 30. Encoder incremental de dos canales [14]

Por tanto, y aunque dependería del tipo de encoder a sustituir, el objetivo final del proyecto consistiría en trasladar todo el código de Matlab a Arduino y mediante los valores de posición y actitud, generar las señales de pulsos equivalentes. Una vez generadas las señales estas serían enviadas mediante los pines digitales o analógicos (según el caso) al georradar, dando así por finalizado este proyecto.

Bibliografía

[1] Naylamp Mechatronics, *"Tutorial MPU6050, Acelerómetro y Giroscopio"*, https://naylampmechatronics.com/blog/45_Tutorial-MPU6050-Aceler%C3%B3metro-y-Giroscopio.html

[2] María José Pardilla Márquez, trabajo fin de grado "*Estudio de la aceleración y posición en sistemas inerciales por medio de Arduino*".

[3] Gonzalo Ferrer Mínguez, trabajo fin de grado, "*Integración Kalman de sensores inerciales INS con GPS en un UAV"*

[4] Utreras A., Guallichico J., Rosales A., Ávalos E., Escuela Politécnica Nacional, Departamento de Automatización y Control Industrial, Quito, Ecuador "*Diseño e Implementación de un Sistema de Navegación Inercial Tipo Strapdown para estimar la Posición de un Robot Móvil*" Revista Politécnica - Julio 2013, Vol. 32, No. 2, Páginas: 112–119.

[5] Alejandro Vázquez Fraga, Universidad da Coruña, trabajo fin de grado "*Integración mediante filtro de kalman de sensores inerciales y gps para la estimación de la posición de un vehículo*"

[6] Luis Llamas, "*Cómo usar un giroscopio en nuestros proyectos de Arduino*" https://www.luisllamas.es/como-usar-un-giroscopio-arduino/.

[7] Wikipedia, acelerómetro, https://es.wikipedia.org/wiki/Aceler%C3%B3metro.

[8] Plataformas gimbaled, "*Inertial navigation system information",* https://www.globalspec.com/learnmore/sensors_transducers_detectors/tilt_sensing/inertial_gyros

[9] MPU 6050 datasheet, https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/

[10] Luis Ródenas, "*Arduino script for MPU-6050 auto-calibration*", https://42bots.com/tutorials/arduino-script-for-mpu-6050-auto-calibration/.

[7] MPU 6050 datasheet, https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/

[8] Inertial Navigation Systems Information _https://www.globalspec.com/learnmore/sensors_transducers_detectors/tilt_sensing/inertial_gyros

[9] Como funciona un acelerómetro, http://www.sensoricx.com/2018/02/como-funciona-un-acelerometro.html

[10] Ander Gracia Moisés, trabajo fin de grado, Diseño y construcción de un robot autobalanceado mediante Arduino,_https://docplayer.es/53704637-Diseno-y-construccion-de-un-robot-auto-balanceado-mediante-arduino.html

[11] https://theopenacademy.com/content/mathematics

[12] ECEF ENU Longitude Latitude Upwardness relationships, _https://commons.wikimedia.org/wiki/File:ECEF_ENU_Longitude_Latitude_Upwardness_relationships.svg

[13] Arduino script for MPU-6050 auto-calibration, https://42bots.com/tutorials/arduino-script-for-mpu-6050-auto-calibration/

[14] Universidad Miguel Hernández, Práctica de automatización industrial sobre contadores rápidos. http://umh1772.edu.umh.es/wp-content/uploads/sites/799/2013/02/Practica-5.pdf