Trabajo Fin de Grado
Grado en Tecnologías en Ingeniería Industriales

Descripción, actualización y ensayo de un vehículo radiocontrolado para la medida de irregularidades del terreno mediante sensores inerciales

Autor: José Cabello del Río
Tutor: José Luis Escalona Franco
Andrés Francisco Hidalgo

Dep. de Ingeniería Mecánica y de Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2015

Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Descripción, actualización y ensayo de un vehículo radiocontrolado para la medida de irregularidades del terreno mediante sensores inerciales

Autor:
José Cabello del Río

Tutor:
José Luis Escalona Franco
Profesor titular
Andrés Francisco Hidalgo
Profesor ponente

Dep. de Ingeniería Mecánica y de Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2015

Trabajo Fin de Grado: Descripción, actualización y ensayo de un vehículo radiocontrolado para la medida de irregularidades del terreno mediante sensores inerciales

Autor: José Cabello del Río

Tutor: José Luis Escalona Franco

Andrés Francisco Hidalgo

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2015

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

Gracias a mi hermana María, mi padre Juan Antonio, mi madre María A. y mi pareja María del Mar por sus consejos, su apoyo y su ánimo, que han sido el combustible que me ha dado la energía necesaria para llegar al final de mis estudios.

Gracias a mis compañeros por hacer amenos estos años de esfuerzo, sacrificio y estudio.

Gracias a mis profesores por su exigencia, sus clases y sus explicaciones.

Gracias al departamento de Ingeniería Gráfica por permitirnos el uso de una estación total y a Cristina Torrecillas por facilitarnos su utilización realizando la calibración del aparato.

Por todos ellos GRACIAS.

Resumen

Este Trabajo de Fin de Grado persigue el desarrollo de tecnologías de auscultación de menos coste que las usadas por las empresas ferroviarias. Con tal fin en estre Trabajo se trata de manera simplificada el problema de auscultación usando un modelo en dos dimensiones de un vehículo auscultador guiado por control remoto construido previamente en la Escuela Técnica Superior de Ingenieros de Sevilla. El trabajo realizado tiene como objetivo principal la realización de ensayos reales de auscultación de perfiles de terrenos mediante el uso de sensores inerciales. Para ello se lleva a cabo la actualización y descripción del software que controla el vehículo así como la utilización de un algoritmo de auscultación para predecir las irregularidades del perfil del terreno recorrido por el vehículo auscultador mediante los datos recogidos por los sensores que incorpora.

Índice

ÍNDICE DE FIGURAS

ÍNDICE DE CÓDIGOS

NOTACIÓN

TRV	Track Recording Vehicles
TRC	Track Recording Coaches
MEMS	Micro Electro-Mechanical Systems
IMU	Inertial Measurement Unit
PWM	pulse-width modulation
SCl	Serial Clock
SDA	Serial Data
MSB	Most Significant Bit
LSB	Less Significant Bit
PSD	Power Spectral Density
FFT	Fast Fourirer Transform

1 INTRODUCCIÓN

Este Trabajo de Fin de Grado se desarrolla con la ayuda del departamento de Ingeniería Mecánica de la Escuela Técnica Superior de Ingenieros de Sevilla. Se sitúa en el ámbito de la auscultación de vías ferroviarias, particularmente en la obtención de perfiles de superficies mediante un vehículo teledirigido que toma medidas de las aceleraciones y giros sufridos durante su recorrido con la ayuda de una IMU (Inertial Measurement Unit). El objetivo es conseguir un vehículo auscultador que reproduzca fielmente la geometría de la pista que atraviesa con el fin de conseguir desarrollar tecnologías de auscultación de menor coste que las usadas actualmente, basadas en modelos matemáticos.

1.1 Objetivo y estado del arte

La auscultación de vías consiste en evaluar las condiciones en las que se encuentra una vía midiendo diversos parámetros como pueden ser la geometría de los carriles o la catenaria. Estas mediciones se realizan a través de trenes o vagones auscultadores acoplados a trenes convencionales, que son capaces de medir los parámetros de la infraestructura mientras circulan por ella [1]. Existen dos tipos de medidas: geométricas y dinámicas. Las geométricas miden las variaciones de geometría respecto a una geometría de referencia mientras que las dinámicas miden la respuesta del vehículo que circula ante dichas irregularidades [2].

Los métodos de auscultación tradicionales estaban basados únicamente en la realización de medidas geométricas de la vía tales como calibre, superficie, alineación, nivelación transversal giro… y su comparación con los límites de los parámetros predefinidos con el fin de comprobar que la geometría de la vía se encontraba dentro de los márgenes que proporcionaban una marcha segura de vehículos. Estas técnicas se veían obligadas a un análisis off-line después de la recolección en la pista impidiendo así el conocimiento instantáneo de los resultados y por tanto del estado de la pista [3]. De modo que ha habido en la industria un movimiento por evaluar la calidad y rugosidad de la geometría de la vía basada en estimaciones en tiempo real. Algunos estudios como los realizados en las referencias [4] y [5] proponen además un estudio de la respuesta dinámica del vehículo ante dichas irregularidades mediante modelos de contacto entre rueda y carril.

Los medios actuales de auscultación ferroviaria como los TRV (Track Recording Vehicles) y los "hauled" TRC (Track Recording Coaches) conllevan costes de operación y mantenimeinto muy elevados [6]. Por el contrario, la auscultación geométrica mediante sensores inerciales (como los MEMS (Micro Electro-Mechanical Systems) de muy bajo coste comparados con otras tecnologías) colocados en distintas partes de vehículos en servicio es una alternativa de más bajo coste y que permite una inspección más regular de las características de la vía. Siendo estas características a su vez más beneficiosas para realizar mantenimiento predictivo, reduciendo así los costes generales de mantenimiento y mejorando las condiciones de seguridad [7].

Dada la relevancia de obtener sistemas auscultadores basados en la información proveniente de sensores inerciales colocados sobre vehículos en servicio, en el año 2013 se comenzó la construcción de un vehículo autónomo dirigido remotamente sobre el cual se colocaría una IMU (Inertial Measurement Unit) con el objetivo de poder estimar las irregularidades geometricas de terrenos con curvas suaves por donde el vehículo circulase.

Al inicio de este trabajo se contaba con una versión preliminar (sobre todo desde un punto de vista del software) del vehículo auscultador cuya construcción formó parte del Proyecto de Fin de Carrera “Desarrollo, instrumentación y construcción de un vehículo radio controlado para la medida de irregularidades del terreno mediante sensores inerciales” [2].

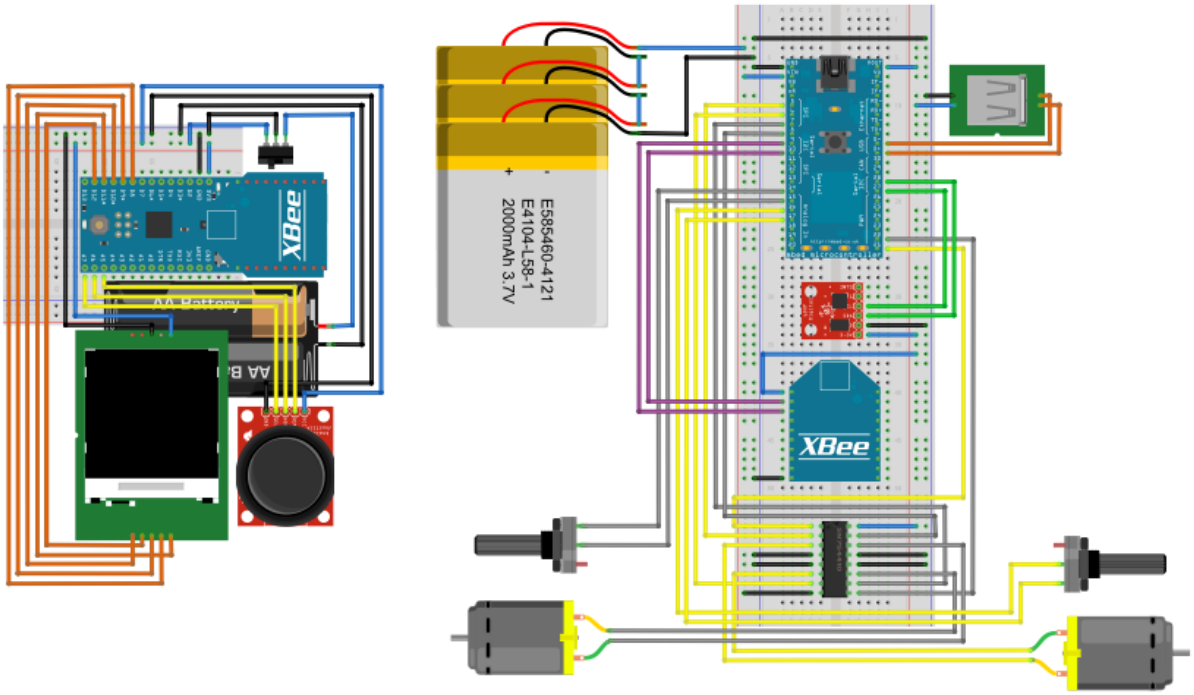


Figura 1: Elementos que componen el mando y el Rover

Como se observa en la Figura 1 el vehículo constaba de dos motores con sus respectivos decodificadores, un puente H, un microcontrolador mbed, dos antenas XBee (una para el mando y otra para el vehículo), un microcontrolador Arduino FIO y una pantalla de Adafruit para el control remoto, una IMU y un puerto USB para conectar una memoria flash donde almacenar los datos obtenidos dutante los ensayos. En esta versión inicial del vehículo autónomo se lograron conectar todas las partes del hardware interviniente a los microcontroladores y se estableció intercambio de señales entre el mando y el vehículo, pero el control mostraba ciertos problemas para mover el vehículo con una determinada precisión y fluidez, sin interrupciones debidas a diversos fallos y sin retardos entre el envío y la recepción de la señal de aceleración, frenado y giro. Estos problemas con el mando del vehículo a su vez impidieron analizar la lectura de aceleraciones y velocidades angulares desde la IMU, no llegándose a almacenar datos en la memoria flash y no ejecutándose ningún ensayo real con el vehículo. Por este motivo el principal objetivo de este Trabajo de Fin de Grado es el de realizar ensayos reales con el vehículo, obteniéndose así los datos de las mediciones de la IMU para estimar las irregularidades del perfil sobre el que se mueve el vehículo. Con este fin y teniendo en cuenta las mejoras que eran necesarias implementarse principalmente en el software de los microcontroladores del mando y del vehículo se obtendrá una versión mejorada del vehículo auscultador que se permita obtener la geometría de perfiles aleatorios mediante la utilización de sensores inerciales de bajo costo.

Si bien la construcción del vehículo se ha mantenido inalterado salvo el cambio del puente H por uno nuevo y la mejora de algunas conexiones puntuales que impedían el correcto funcionamiento de los elementos que componen el aparato.

La estructura de este documento consiste en siete secciones, incluyendo las conclusiones y referencias. En la Sección 2 se lleva a cabo una descripción general del funcionamiento del software que controla tanto el control remoto como el vehículo auscultador detallando con mayor profundidad los cambios efectuados en el mismo para mejorar la robustez de la ejecución del programa; en la Sección 3 se muestra la estimación de irregularidades en perfiles aleatorios mediante el Filtro de Kalman; a continuación en la Seccion 4 se describe la realización de los ensayos de auscultación usando el Rover para obtener los datos recogidos por los sensores de inercia de la IMU; las tres secciones restantes recogen los resultados y conclusiones . Finalmente se recoge la documentación usada para la realización de este trabajo.

1.2 Fundamento teórico

El movimiento del Rover sobre un perfil aleatorio se puede modelar, como se muestra en la Figura 2, como un sistema de dos grados de libertad, desplazamiento vertical x y giro θ, al que se le impone un desplazamiento en la base asociado al perfil.

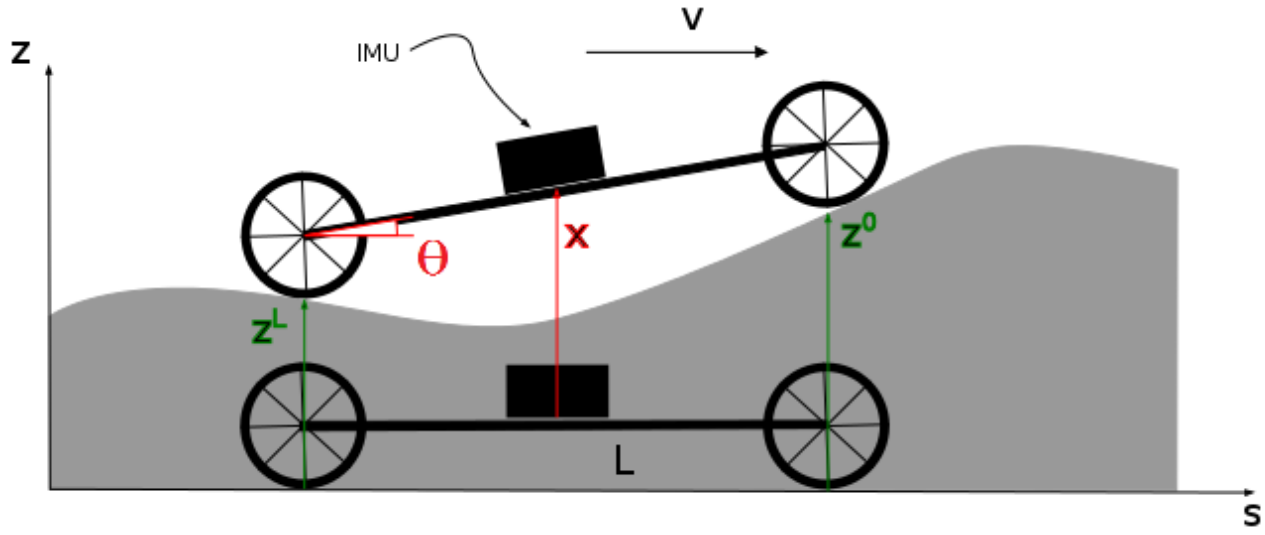


Figura 2: Modelo Rover - Perfil

El objetivo es calcular el perfil del terreno que recorre el vehículo, denotado por $z(s)$, a partir de los datos obtenidos del acelerómetro $\ddot{x}(t)$ y giróscopo $\dot{\theta}(t)$. Para ello se tienen en cuenta las siguientes hipótesis:

- La velocidad del vehículo a lo largo del recorrido es constante
- Los neumáticos se consideran rígidos
- El vehículo no tiene suspensión

De la Figura 2 se obtienen las siguientes relaciones:

$$x(t) = \frac{z^0 + z^L}{2} \tag{1}$$

$$\theta(t) = \frac{z^0 + z^L}{L} \tag{2}$$

Donde x(t) representa la posición de la IMU en función del tiempo, z^0 la coordenada vertical del perfil en la rueda delantera, z^L la coordenada vertical del perfil de la rueda trasera, θ(t) es el ángulo girado entre el vehículo con respecto a la horizontal y L la longitud de este. La aceleración vertical que sufre el sensor se conoce derivando dos veces respecto al tiempo la coordenada x como se muestra en las siguientes expresiones:

$$\dot{x} = \frac{dx}{ds}\frac{ds}{dt} = x'v \tag{3}$$

$$\ddot{x} = v\frac{dx'}{ds}\frac{ds}{dt} + \frac{dv}{dt}x' = x''v^2 + x'v \tag{4}$$

Donde x' y x'' son la derivada primera y segunda de la Ecuación (1) respecto de s y v es la velocidad de avance del vehículo, siendo z'^0, z'^L, z''^0 y z''^L las derivadas primeras y segundas de las coordenadas verticales del perfil en la rueda delantera y trasera del vehículo con respecto a s.

$$x' = \frac{z'^0 + z'^L}{2} \tag{5}$$

$$x'' = \frac{z''^0 +'' z^L}{2} \tag{6}$$

De igual forma se conoce la velocidad angular a la que se somete el sensor derivando con respecto al tiempo la coordenada θ:

$$\dot{\theta} = \frac{d\theta}{ds}\frac{ds}{dt} = \theta' v \tag{7}$$

Donde:

$$\theta' = \frac{z'^0 + z'^L}{L} \tag{8}$$

Las ecuaciones (4) y (7) representan la relación entre las señales IMU y la coordenada vertical del perfil por el que avanza el vehículo.

Suponiendo $z(s)$ como una distribución senoidal $z(s) = Ze^{\left(\frac{2\pi}{\lambda}\right)s} = Ze^{i\omega s}$ donde Z es la amplitud y ω la frecuencia, se tiene:

$$x''(s) = Ae^{i\omega s} \tag{9}$$

$$\theta' = \Omega e^{i\omega s} \tag{10}$$

Para el caso de la aceleración, haciendo uso de la Ecuación (6) y (9):

$$x''(s) = Ae^{i\omega s} = \frac{-\omega^2 Ze^{i\omega s} - \omega^2 Ze^{i\omega(s-L)}}{2} = \frac{-\omega^2 Ze^{i\omega s}\left(1+e^{i\omega L}\right)}{2} \tag{11}$$

De donde se obtiene:

$$A = -\frac{1}{2}\omega^2\left(1+e^{i\omega L}\right)Z \tag{12}$$

$$\frac{A}{Z} = -\frac{1}{2}\omega^2\left(1+e^{i\omega L}\right) = T_1(\omega) \tag{13}$$

Donde T_1 es la función de transferencia entre la aceleración y la irregularidad del perfil.

Para la velocidad angular, haciendo uso de la Ecuación (8) y (10):

$$\theta' = \Omega e^{i\omega s} = \frac{i\omega Ze^{i\omega s} - i\omega Ze^{i\omega(s-L)}}{L} = \frac{1}{L} i\omega Ze^{i\omega(s)}\left(1-e^{i\omega L}\right) \tag{14}$$

De donde se obtienen la amplitud Ω y la función de tranferencia entre la velocidad angular y la irregularidad T_2, que se muestra en las expresiones (15) y (16):

$$\Omega = \frac{1}{L} i\omega\left(1-e^{i\omega L}\right)Z \tag{15}$$

$$\frac{\Omega}{Z} = \frac{1}{L} i\omega\left(1-e^{i\omega L}\right) = T_2(\omega) \tag{16}$$

En la Figura 3 se observa que las funciones de transferencia (T_1 y T_2) de cada sensor se anulan para ciertas frecuecias. Esto significa que cada sensor es incapaz de detectar dichas frecuencias y debido a que $\omega = 2\pi/\lambda$ determinadas longitudes de onda. Por este motivo para obtener una descripción completa de la geometría del perfil se opta por la fusión de sensores mediante el uso del Filtro de Kalman, con el que se obtiene la estimación irregularidades de perfiles como se verá en la sección 3 Estimación de Irregularidades en Perfiles Aleatorios Mediante el Filtro de Kalman.

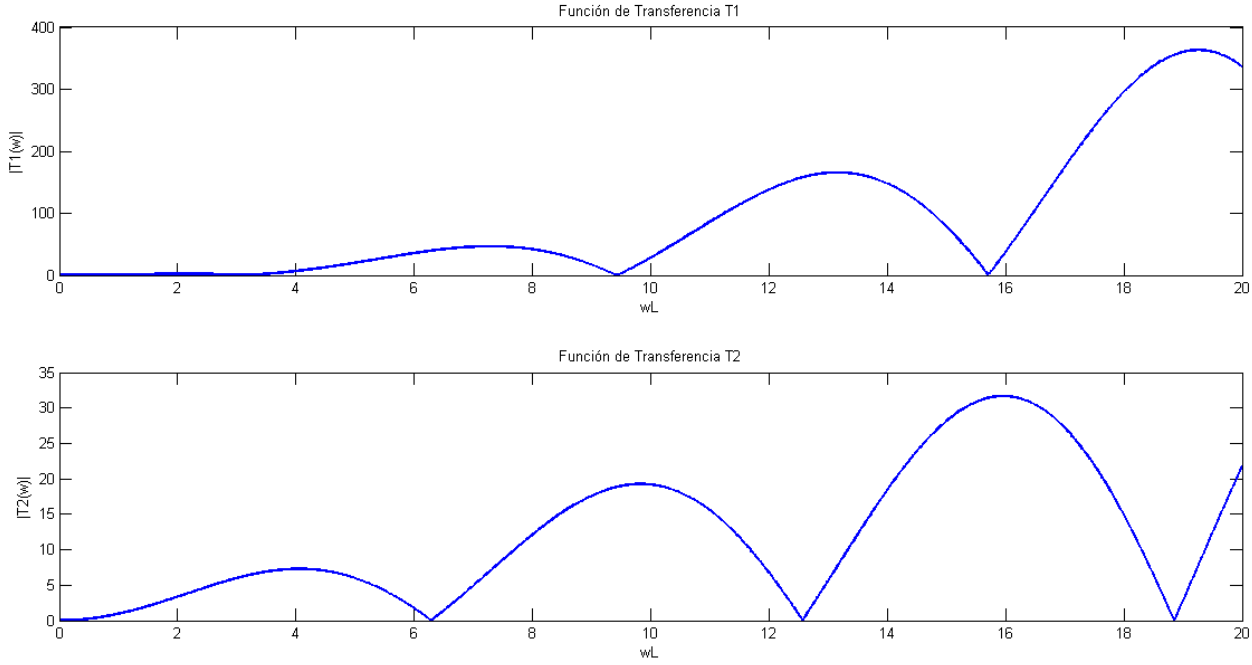


Figura 3: Funciones de Transferencia

2 DESCRIPCIÓN Y MEJORAS DEL ROVER

En esta Sección se describe el funcionamiento general del vehículo desde que se pone en marcha hasta que termina la ejecución del programa que lo controla. En primer lugar se describe de forma general la interacción entre los distintos elementos que componen tanto el Rover como el mando a distacia. En lo que sigue en esta sección se profundiza en la descripción de las modificaciones realizadas para mejorar la versión inicial de partida.

2.1 Descripción general del funcionamiento del Rover

El Rover es un vehículo autónomo que se controla de forma inalámbrica mediante un mando a distancia. Para controlar el mando a distancia se utiliza como procesador un Arduino FIO [8] al que se encuentran conectados un joistick mediante el cual el usuario actua sobre el vehículo y una pantalla Nokia LCD 5110 usada para mostrar información relativa al funcionamiento del aparato (apertura del fichero de texto, velocidad de referencia, velocidad real…). El intercambio de información se realiza mediante dos dispositivos Xbee [9] colocados uno en el Rover y otro en el control remoto. Para controlar el vehículo se utiliza como procesador una placa Mbed NXP LPC1768 [10] a la que se encuentran conectados, además de la Xbee, un Puente H que gestiona el funcionamiento de los motores Pololu [11] con sus respectivos decodificadores, una IMU de DroTek, compuesta por un aclelerometro BMA180 [12] y un giróscopo IGT3200 [13], para la recogida de datos que nos permitirá recoger las información necesaria para predecir la geometría del perfil recorrido y un puerto USB para el almacenamiento de los datos recogidos durante el funcionamiento del vehículo.

El usuario actua sobre el joystick del mando a distancia para ordenar al vehículo hacia donde debe dirigirse, inclinando el vástago hacia adelante para acelerar, hacia atrás para frenar y hacia los lados para girar. Hay que notar que la rueda trasera es libre y que el vehículo lo usaremos únicamente de forma que avanzance hacia adelante. El arduino recibe las señales del joystick y las envía mediante la Xbee hacia el Rover como se observa esquemáticamente en la Figura 4.

La programación realizada en la placa Mbed consiste en una programación por hilos, estos se encargan de la ejecución de forma paralela al programa principal de ciertas funciones. En la Figura 4 se observa además un diagrama de flujo que representa la dirección de ejecución del programa y hacia qué elementos se envía la información. El hilo 1 se encarga de la lectura de los datos de la IMU durante el proceso de auscultación y los almacena provisionalmente en un buffer de memoria, el hilo 2 se encarga del control de velocidad que gestiona los motores. El programa principal consite en un bucle que gestiona la información que proviene del mando a distancia, calcula la velocidad de referencia de los motores en función de la orden recibida y la envía al control de velocidad. También realiza las mediciones de la velocidad real de ambos motores a través de los decodificadores, enviándola de igual forma al control de velocidad para que este calcule la respuesta del vehículo para alcanzar la velocidad impuesta por el usuario, esta respuesta se envía al puente H que actua sobre los motores mediante una señal PWM moviendo el vehículo. Conocida la velocidad real, esta se envía de vuelta hacia el mando a distancia a través de las Xbee y se imprime su valor en la pantalla LCD con el objetivo de mantener al usuario informado. Finalmente se realiza la escritura en el fichero de texto de los datos almacenados en el buffer de memoria. Transcurrido un tiempo determinado el programa sale del bucle parando los motores y cerrando el fichero con la información recogida durante su recorrido.

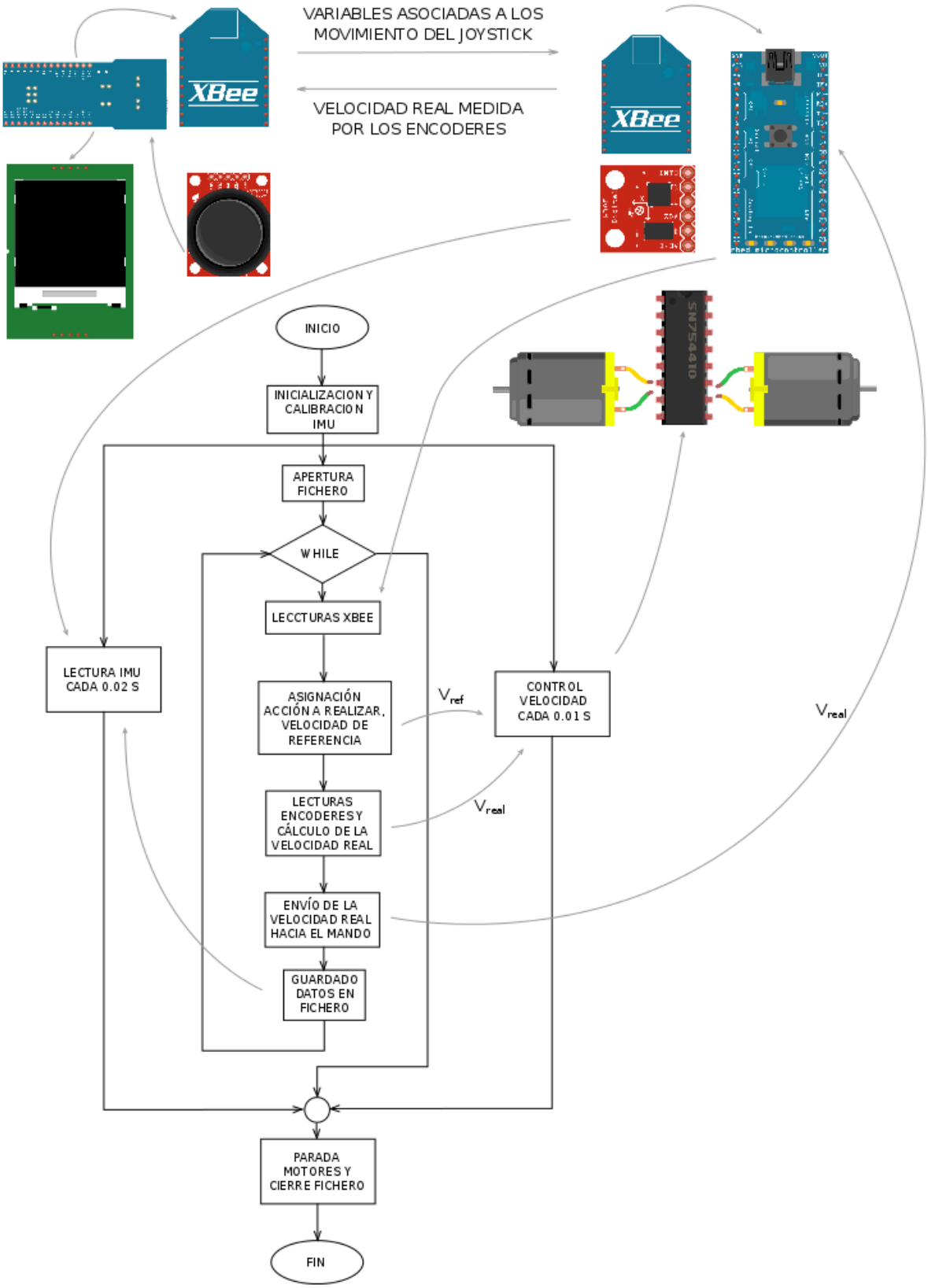


Figura 4: Esquema funcionamiento general del Rover

2.2 Decodificadores

Uno de los aspecto que se han mejorado con respecto a la version inical son los decodificadores, estos se encargan de medir la velocidad real de giro de los motores del vehículo y con ello podemos conocer la velocidad real con la que este avanza.

Inicialmente los canales A y B del decodificador derecho se encontraban conectados a los pines 16 y 17 de la Mbed y los del decodificador derecho a los 18 y 19 respectivamente. Estos pines estaban definidos como variables de tipo DigitalIn por lo que solo detectaban si en la entrada del pin había un cero o un uno. De este modo mediante bloques lógicos if se leía la señal que llegaba a cada pin para detectar cuando se producía un cambio de valor, de cero a uno o viceversa y se calculaba el tiempo transcurrido entre dos cambios de valor. De esta forma se obtenía el tiempo que se tardaba en recorrer los dientes de la señal del decodificador, conociendo los pasos por revolución del decodificador y la relación de transmisión entre la rueda y el eje del motor se calculaba la velocidad real de la rueda asociada a dicho motor.

Para realizar una medida de la velocidad real más precisa se definen los pines de entrada de las señales de los decodificadores como variables tipo InterruptIn. Todos los pines desde el 5 hasta el 30 de la Mbed excepto el 19 y el 20 pueden definirse como InterruptIn (ver Rob Toulson y Tim Wilmshurst [14]) de modo que los decodificadores se han reconectado usando los pines 17 y 16 para los canales A y B del decodificador derecho y usando los 14 y 15 para los pines A y B del decodificador izquierdo, tal y como se observa en la Figura 5. Definir estos pines como variables interrupt nos permite asociar la ejecución de una función a una subida o bajada en el valor de la señal de entrada a la Mbed. De modo que asociando una función que mida el incremento de tiempo entre cada subida y bajada podemos conocer la velocidad real de las ruedas del vehículo en cada instante. De esta forma nos aseguramos que obtenemos una medida en cada cambio de valor en las señeles de los decodificadores obteniéndose una programación mejor estructurada y valores más precisos de la posición y velocidad del vehículo.

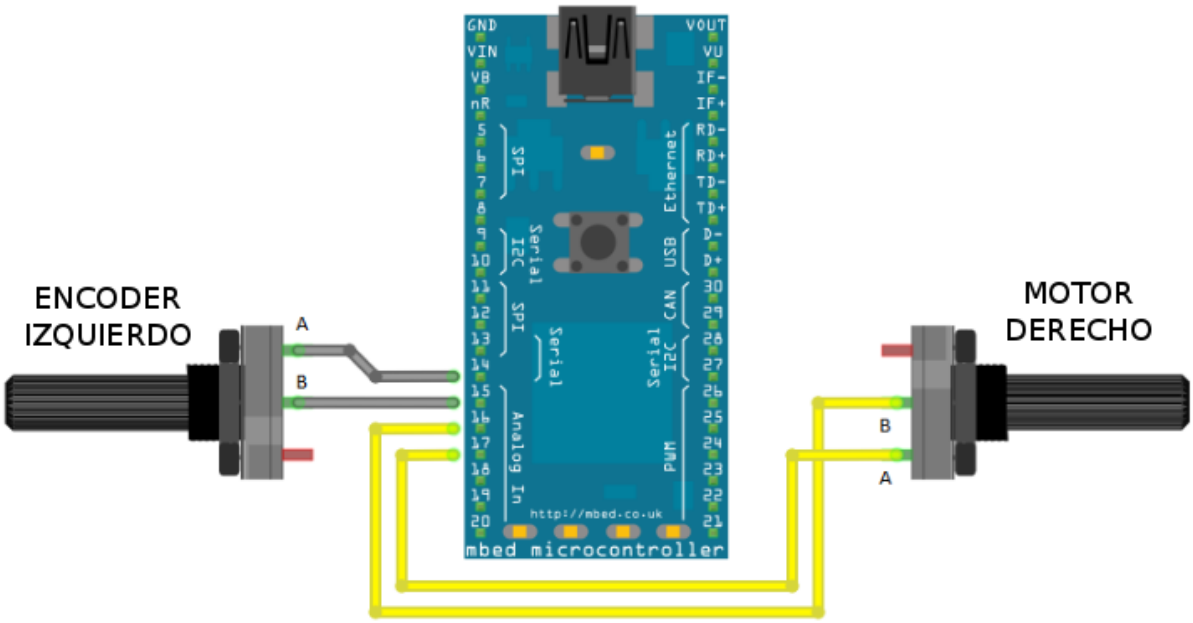


Figura 5: Conexiones encoderes - mbed

Para definir las variables de tipo InterruptIn y calcular la velocidad real de las ruedas del Rover se ha creado una clase llamada MyEcoder. Esta clase asocia un interrupt a cada subida y bajada de la señal del decodificador mediante la función Encoder que se muestra en el Código 1 y asocia a dichos interrupts la función enc mostrada en el Código 2 encargada de medir la velocidad de la rueda.

Código 1: Función Encoder

```
Encoder::Encoder(PinName chA, PinName chB, int Npulse, Timer t):
channelA(chA), channelB(chB), Np(Npulse){

    initTime_ = t.read_us();

    prevTime_ = 0;
    _SpeedTimer.start();
    pulses_ = 0;
    prevPulses_ = pulses_;

    chanA = channelA.read();
    chanB = channelB.read();

    channelA.rise(this, &Encoder::enc);
    channelA.fall(this, &Encoder::enc);
}
```

Código 2: Función enc

```
void Encoder::enc (void){
    int chanA  = channelA.read();
    int chanB  = channelB.read();

    currTime_ = _SpeedTimer.read_us();

    pulses_ = prevPulses_ + 1;
    deltaT_  = currTime_ - prevTime_;
    nrev = 1.0/32;
    rpm = (nrev/deltaT_)*1.0e6*60/30;

    prevTime_ = currTime_;
    prevPulses_ = pulses_;
}
```

Para obtener los valores de la posición y velocidad calculadas por la función enc, se utiliza la función miembro getrpm, perteneciente también a la clase MyEncoder y que se muestra en el Código 3 y Código 4.

Código 3: Función getrpm

```
double Encoder::getrpm (void){
     return rpm;
}
```

Código 4: Función getpulses

```
int Encoder::getpulses (void){
    return pulses_;
    }
```

2.3 Control PID

El control PID es el elemento encargado de controlar los cambios de velocidad que se efectúan durante el funcionamiento del vehículo. Para gestionar los controles PID se adjunta la función controlVelocity, que se muestra en el Código 5, al hilo 2 mencionado en la Sección 2.1 Descripción general del funcionamiento del Rover. Para generar el control se usa una clase llamada PID adquirida del cookbook de la página oficial de Mbed [15]. Con esta clase se definen dos objetos tipo PID asociados cada uno de ellos a cada motor rightController y leftController. Los parámetros de ajuste de los controles recomendados para esta aplicación son: Kc=1.05 para la parte proporcional, Ti=0.15 para la parte integral y Td=0 para la parte derivativa.

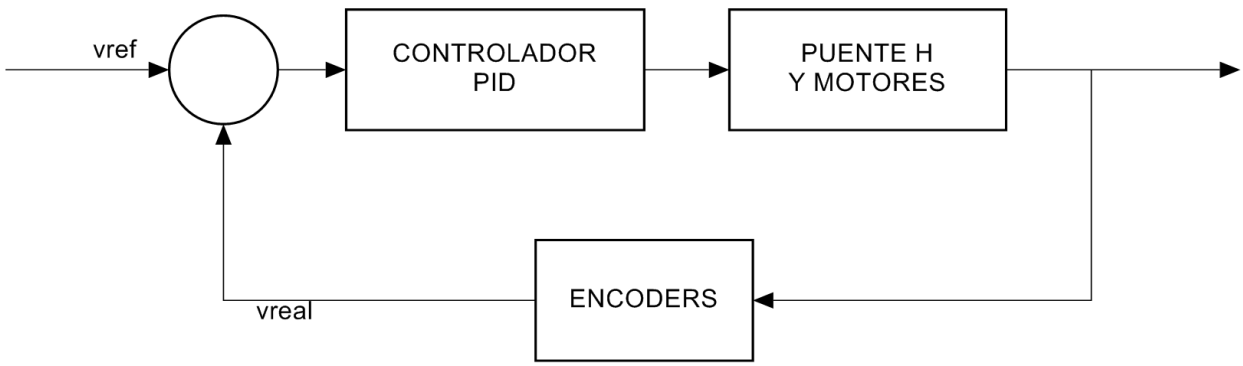
Código 5: Función controlVelocity

```
void controlVelocity(void const *n){

    rightController.setSetPoint(Vrd_);
    Va_mm = (rpm_R*2.0*PI*radio_mm)/60.0;
    Va_ = Va_mm/1650;
    rightController.setProcessValue(Va_);
    rightMotor.speed(rightController.compute());

    leftController.setSetPoint(Vri_);
    Vb_mm = (rpm_L*2.0*PI*radio_mm)/60.0;
    Vb_= Vb_mm/1650;
    leftController.setProcessValue(Vb_);
    leftMotor.speed(leftController.compute());
}
```

En la Figura 6 se observa esquemáticamente el funcionamiento de los controles PID. Mediante la función miembro setSetPoint se introduce el valor de velocidad real al control y mediante la función setProcessValue se introduce la velocidad real medida por los encoderes anteriormente y almacenada en las valiables rpm_R y rpm_L. Mediante la función compute se calcula la diferencia entre los dos valores generando así la respuesta del sistema para alcanzar la velocidad objetivo. Esta respuesta se envía al puente H que accionará los motores.



2.4 Puente H

El Puente H es el elemento encargado de controlar la velocidad y el sentido de giro de los motores. Con el fin de gestionar el funcionamiento del puente H se ha incluido en el programa contenido en la Mbed una clase llamada Motor también disponible en la página oficial de Mbed [16].

En la Figura 7 se muestran las conexiones realizadas entre el procesador, el puente H y los motores. Las conexiones de color amarillo son las asociadas al motor derecho mientras que las de color gris al izquierdo. Mediante los pines 5, 6, 7 y 8 se envían las señales que indican a cada motor el sentido de giro, las variables _fwd indican que el sentido de giro es hacia adelante mientras que las _rev lo indican hacia atrás. Mediante los pines 21 y 22 se envían la señal PWM que indica al motor la velocidad de giro.

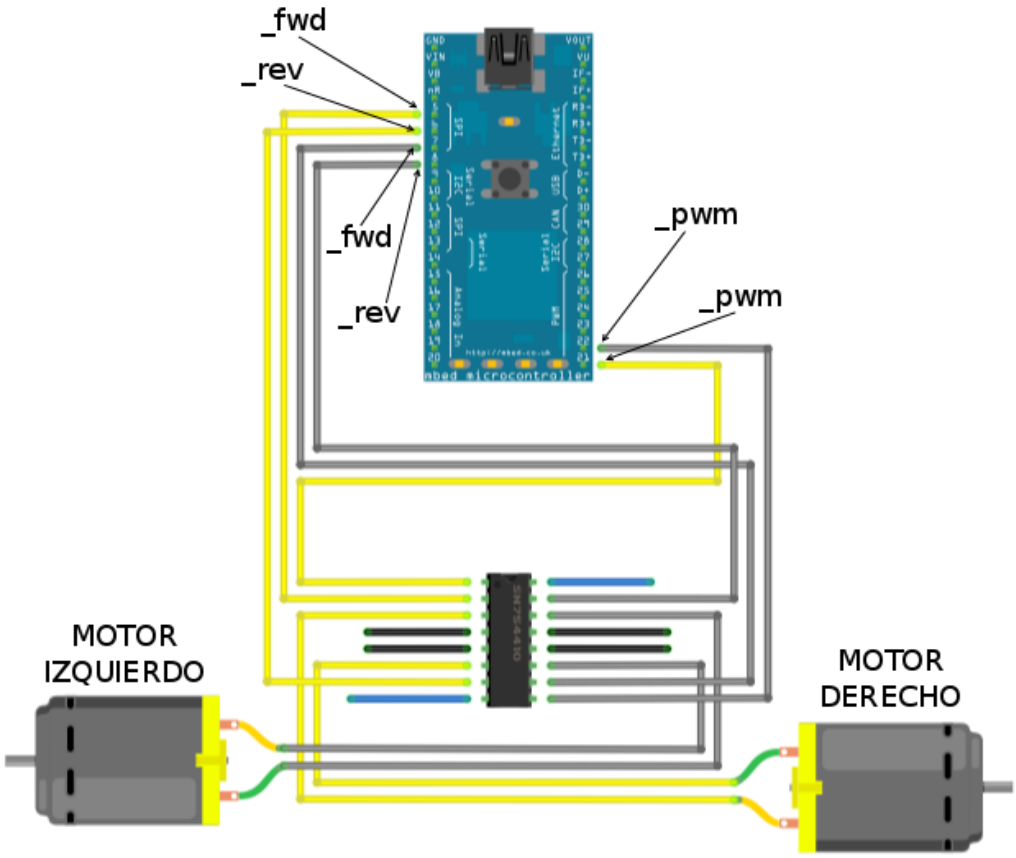


Figura 7: Conexiones motores - puente H - mbed

En el Código 6 se muestra la función speed encargada de la gestión del puente H que pertenece a la clase Motor. Esta recibe el valor de velocidad calculado por el control PID mediante la función compute como se observa en el Código 5. En función del signo del valor que recibe se determina el sentido de giro del motor enviando un uno o un cero a través de las variables DigitalOut _fwd o _rev. La variable PwmOut _pwm envía la señal PWM al motor haciéndolo girar a la velocidad deseada.

Código 6: Función speed

```
void Motor::speed(float speed) {
    _fwd = (speed > 0.0);
    _rev = (speed < 0.0);
    _pwm = abs(speed);
}
```

2.5 IMU

En la Figura 8 se observa la conexión de la IMU con el procesador, esta es una conexión puerto serie I^2C [12] realizada mediante una placa Droteck. La configuración I^2C se realiza mediante dos conexiones: SCL y SDA realizadas en los pines 27 y 28 de la Mbed denominando a este último como Maestro y al acelerómetro (BMA180) y al giróscopo (IGT3200) como Esclavos. Cada Esclavo tiene una dirección diferente, de modo que para leer la información recogida por la IMU el Maestro envía la dirección de esclavo del cual leerá y este envía la información solicitada.

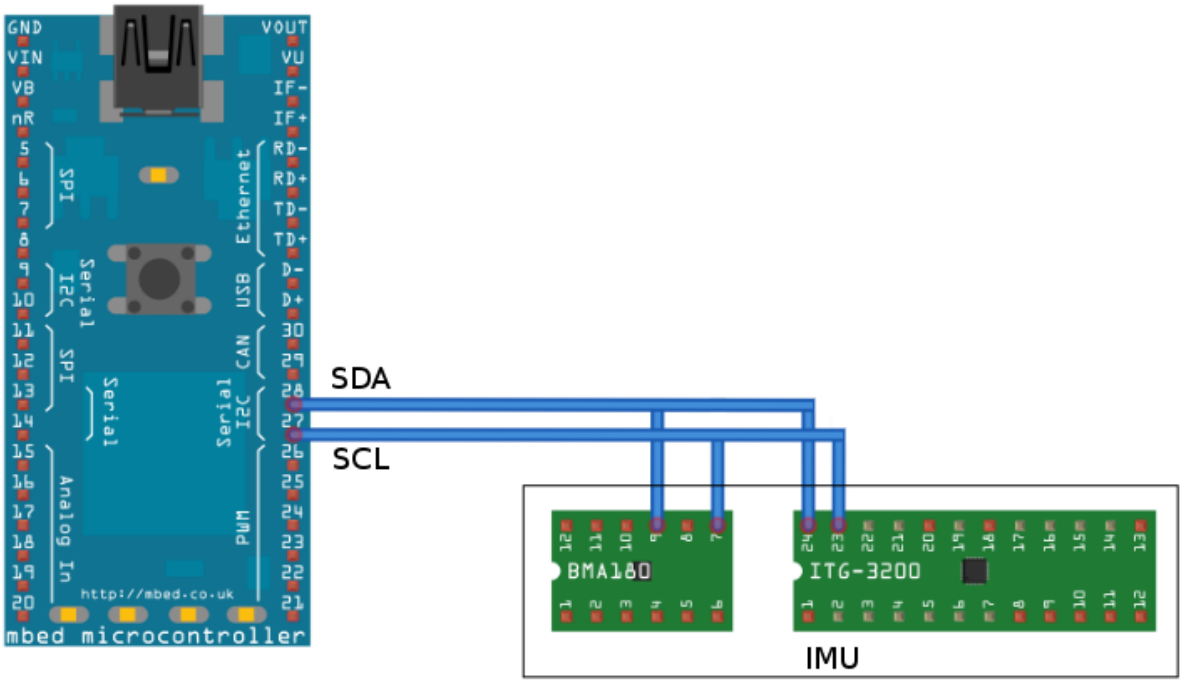


Figura 8: Conexión IMU – Mbed

Con el fin de gestionar la lectura de la IMU se usan unas librerías disponibles en la página de Mbed [16]: BMA180 e IGT3200, sin embargo la primera necesitó ser modificada para leer los datos de aceleración adecuadamente ya que se detectó un error. Las señales digitales del acelerómetro están descritas por 2's complement basado en 14 bits [12], de modo que el rango de medidas del acelerómetro de ± 2 g se obtiene de forma binaria:

-2.00000g	:	10 0000 0000 0000
-199975g	:	10 0000 0000 0001
		…
-0.00025g	:	11 1111 1111 1111
0.000000g	:	00 0000 0000 0000
+0.00025g	:	00 0000 0000 0001
		…
+1.99950g	:	01 1111 1111 1110
+1.99975g	:	01 1111 1111 1111

Cada valor esta compuesto por MSB (Most significant Bit) y LSB (Less significant Bit) tal como se observa en la Figura 9 [12].

Register Name	Register Address (hexadecimal)	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
acc_z_msb	07h	acc_z<13:6> (msb)							
acc_z_lsb	06h	acc_z<5:0> (lsb)						0	new_data_z
acc_y_msb	05h	acc_y<13:6> (msb)							
acc_y_lsb	04h	acc_y<5:0> (lsb)						0	new_data_y
acc_x_msb	03h	acc_x<13:6> (msb)							
acc_x_lsb	02h	acc_x<5:0> (lsb)						0	new_data_x

Figura 9: Mapa de memoria BMA180

La versión de la librería BMA180 descargada de página oficial de Mbed realizaba la lectura del LSB de cada dato de aceleración, debido a que los dos últimos bits de dicha lectura no son datos de aceleración estos eran eliminados y a continuación se leía el MSB almacenándose estos en una única variable que representaba el dato de aceleración. El hecho de eliminar los dos últimos bits antes de combinar el MSB y el LSB implicaba desplazar los bits no eliminados hacia la derecha colocando dos ceros en los bits 6 y 7 del LSB, de modo que al combinar estos para obtener el valor de aceleración se obtenía un dato no real. En la Figura 10 se representan las acciones descritas.

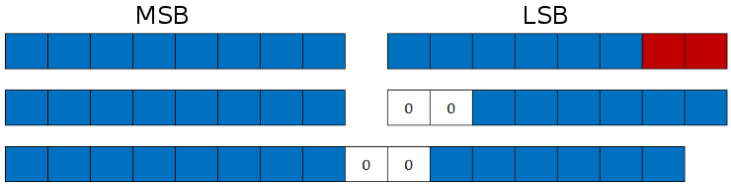


Figura 10: Lectura aceleración librería BMA180

Debido a esto, se modificó la librería descargada de modo que se realizaran en primer lugar las lecturas del MSB y LSB, a continuación su combinación y por último la eliminación de los dos últimos bits para obtener el dato real del valor de aceleración, como se representa en la Figura 11.

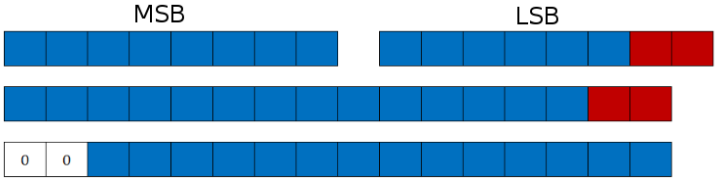


Figura 11: Lectura aceleración librería BMA180 modificada

2.6 Puerto USB

Con el objetivo de gestionar las lectuas de la IMU se decidió utilizar un buffer donde almacenar los datos recogidos por los sensores para después escribirlos en el fichero de texto. Para ello se utiliza una librería, llamada RingBuffer, que contiene una clase llamada Buffer con la que se generan los objetos que realizan la escritura y lectura de los datos en el buffer. Se define un buffer de 500 celdas, como el representado en la Figura 12, asociado a cada parámetro que se almacenará en el fichero: tiempo del ensayo en el que se realiza la lectura, velocidad de ambas ruedas y aceleraciones y velocidades angulares en las direcciones X, Y y Z.

Figura 12: RingBuffer

Como se recuerda del apartado 2.1 Descripción general del funcionamiento del Rover, el hilo uno de programación es el encargado de la lectura de los datos de la IMU mediante la función readIMU que se observa en Código 7. En ella se leen los datos de aceleración en g y las velocidades angulares en º/s de los sensores, además de las velocidades de los motores y a continuación se almacena esta información en el buffer correspondiente mediante la función put que pertene a la clase Buffer, esto se representa en la Figura 12 mediante el índice ESCRITURA.

Código 7: Función readIMU

```
void readIMU(void const *n){

    Acel3[1] = -(acc3.getX() - a_xBias3)*0.00024417043;
    Acel3[0] = -(acc3.getY() - a_yBias3)*0.00024417043;
    Acel3[2] = -(acc3.getZ() - a_zBias3)*0.00024417043;

    Gyro3[1] = -(gyro3.getGyroX()- w_xBias3) / 14.375;
    Gyro3[0] = -(gyro3.getGyroY()- w_yBias3) / 14.375;
    Gyro3[2] = -(gyro3.getGyroZ()- w_zBias3) / 14.375;

    rpm_R = R_encoder.getrpm();
    rpm_L = L_encoder.getrpm();

    buff_velocA.put(rpm_R);
    buff_velocB.put(rpm_L);

    timeIMU.put(t.read());

    buff_aX.put(Acel3[1]);
    buff_aY.put(Acel3[0]);
    buff_aZ.put(Acel3[2]);

    buff_wX.put(Gyro3[1]);
    buff_wY.put(Gyro3[0]);
    buff_wZ.put(Gyro3[2]);
}
```

Para almacenar la información del buffer en el fichero de texto se utiliza la función GuardaDatosBuffer que se presenta en el Código 8 por el índice LECTURA de la Figura 12. Esta función lee los datos almacenados en cada celda del buffer mediante la función get (perteneciente a la clase Buffer) y los escribe en el fichero creando una tabla como la de la Figura 13. El índice de LECTURA de desplaza siempre detrás del índice de ESCRITURA de modo que cuando llegan al final del buffer vuelven al inicio de este para continuar con el almacenamiento de datos. Teniendo en cuenta que los datos de las celdas en las que se sobresscribirá ya se encuentran almacenados en el fichero de texto.

Tiempo ensayo	Velocidad rueda derecha	Velocidad rueda izquierda	Aceleración eje X	Aceleración eje Y	Aceleracón eje Z	Velocidad angular eje X	Velocidad angular eje Y	Velocidad angular eje Z
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Figura 13: Tabla formato fichero de texto

Código 8: Funcion GuardaDatosBuffer

```
void GuardaDatosBuffer(){

    bool empty = buff_aX.isEmpty();
    float tiempo;

    if(empty == false){

        tiempo = ((float)ContadorDatos)*((float)PasoTiempo)/1.0e6;

        fprintf(fp,"%.2f %.3f %.3f %.3f %.3f %.3f %.3f %.3f %.3f \n",
timeIMU.get(), buff_velocA.get(), buff_velocB.get(), buff_aX.get(),
buff_aY.get(), buff_aZ.get(),buff_wX.get(), buff_wY.get(),
buff_wZ.get());

        myled2 = !myled2;
        ContadorDatos++;
    }
}
```

La memoria interna del Porcesador es de 512 KB [17], de modo que permite almacenar datos de ensayos de tan solo unos segundos de duración. Debido a la necesidad de realizar ensayos más largos el Rover incorpora un puerto USB para la conexión de un dispositivo de almacenamiento masivo externo para recoger la información del proceso de auscultación.

2.7 Mando a distancia

El mando a distancia se controla a través del Arduino FIO al que se le conectan una Xbee para la transmisión de datos entre el mando y el Rover, una pantalla LCD donde imprimir información relativa al funcionamiento del vehículo y un joystick con el que dirigir el aparato.

En la versión de partida con la que se comenzó a trabajar en este Trabajo de Fin de Grado la transmisión de datos entre el mando y el vehículo era muy lenta y poco fluida. Para mejorar este aspecto hubo que setear correctamente las direcciones de las dos Xbee con el programa X-CTU [18] de forma que cada una de ellas tuviera correctamente la dirección de la otra. Por otra parte el programa encargado de la transmisión de datos contenido en el arduino utilizaba muchos bloques lógicos if para detectar las señales y realizaba las lecturas de la Xbee cada cierto tiempo. Esto ha sido modificado sustituyendo el programa del arduino por el que se muestra en el Código 9 cuyo funcionamiento se describe a continuación. En este la lectura de las Xbee se realiza de forma libre, es decir con un baud rate de 4800, a diferencia del usado en la versión inicial de 9600. Con la introducción de estos cambios se mejoró notablemante la transferencia de datos entre el control remoto y el Rover.

En al Figura 14 se observa la secuencia de tareas que realiza el programa contenido en el Arduino además del intercambio de información entre los distintos elementos que componen el control remoto. El joystick genera dos señales analógicas en función del movimiento del vástago que se han llamado VERT y HORIZ y que son recibidas por el arduino que a continuación realiza la escritura en la pantalla LCD de la velocidad real y de si se ejecuta un giro. El dato de velocidad real que se envía desde el Rover se recibe en tres paquetes, esto permite realizar un checksum para comprobar que no se pierde información durante la transmisión de datos y que esta se realiza correctamente. Primero se realizan las lecturas de la Xbee que recibe el dato de velocidad real y a continuación se realiza el checksum. Por último se codifican las señales del joystick para enviarlas hacia el Rover. Las señales generadas por el joystick toman valores de 0 a1023 de modo que si la señal vertical toma valores entre 515 y 535 la variable VERT toma el valor de 70 indicando que se mantiene la misma velocidad de avance; si es mayor que 535, VERT toma el valor de 75 indicando que se reduce la velocidad; mientras que si es menor que 515, la variable VERT toma el valor de 80 indicando un aumento de velocidad.

De igual forma sucede con la varible HORIZ tomando los valores 24, 25 y 26 si el joystick se encuentra en su posición central, si se desplaza hacia la derecha o hacia la izquierda respectivamente. Estos valores se envía a través de la Xbee y con ellos se controla el movimiento del vehículo.

Código 9: Programa arduino

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>

Adafruit_PCD8544 display…
=Adafruit_PCD8544(8, 9, 10, 11, 12);
const int VERT = 5;
const int HORIZ = 6;
const int SEL = 13;

void setup() {
  pinMode(SEL,INPUT);
  digitalWrite(SEL,HIGH);

  Serial.begin(4800);

  display.begin();
  display.setContrast(50);
}
int vertical_ref, horizontal_ref;
int vertical_Accum, horizontal_Accum;
int horizontal_big;
int entero, rec, rec_aux, error,it, itchck;
int rec_accum;
int itmax = 100;
char chnum1, chnum2, chnum3,e;
char rounding, hznum1, hznum2;
char hznum3, hznum4;
int num1, num2, num3;

void loop() {
  int vertical, horizontal, select, Vel_real;
  int Vx, vert, check;
  double Vel_ref;
  calibrate_joystick();
  vertical = analogRead(VERT);
  horizontal = analogRead(HORIZ);
  select = digitalRead(SEL);

    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(BLACK);
    display.setCursor(0,0);
    display.println("Vreal: ");
    display.println(rec_accum);
    display.println("Rounding: ");
    display.println(horizontal_big);
    display.println("select: ");
if(select == 1){
  display.println(select);
}else{
  display.println("not ");
  digitalWrite(SEL,HIGH);
}
display.display();

it = 0;
rec_aux = 0;

if (Serial.available() > 0){
  num1 = Serial.read();
  num2 = Serial.read();
  num3 = Serial.read();

 itchck = 0;
  if(((num1 & num2) == num3)){
    rec_accum = num2;
    rec_accum = rec_accum << 8;
    rec_accum = rec_accum + num1;
    error = 0;
  } else{
    error = 1;
    rec_accum = e;
  }
}

if (515 <=vertical && vertical <= 535){
    vert = 70;
    check = 1;
} else if (535 < vertical){
    vert  = 75;
    check = 1;
} else if(vertical < 515){
    vert = 80;
    check = 1;
}

if (check ==1){
    entero = vert;
    chnum1 = entero;
    Serial.write(chnum1);
}
check = 0;
```

```
if (490 <=horizontal && horizontal…
 <= 530) {
      horizontal_big = 24;

   } else if (horizontal > 530) {
      horizontal_big = 25;

   } else if (horizontal < 490) {
      horizontal_big = 26;
   }

   entero = horizontal_big;
   hznum1 = entero;
   Serial.write (hznum1);
}
```

```
void calibrate_joystick (void) {
  int i;
  vertical_Accum = 0;  horizontal_Accum =
0;
  vertical_ref = 0; horizontal_ref = 0;

  for (i=0; i<100; i++) {
    vertical_Accum +=
analogRead (VERT);
    horizontal_Accum +=
analogRead (HORIZ);
  }

  vertical_Accum /= 100;
horizontal_Accum /= 100;
  vertical_ref = vertical_Accum;
horizontal_ref = horizontal_Accum;
}
```

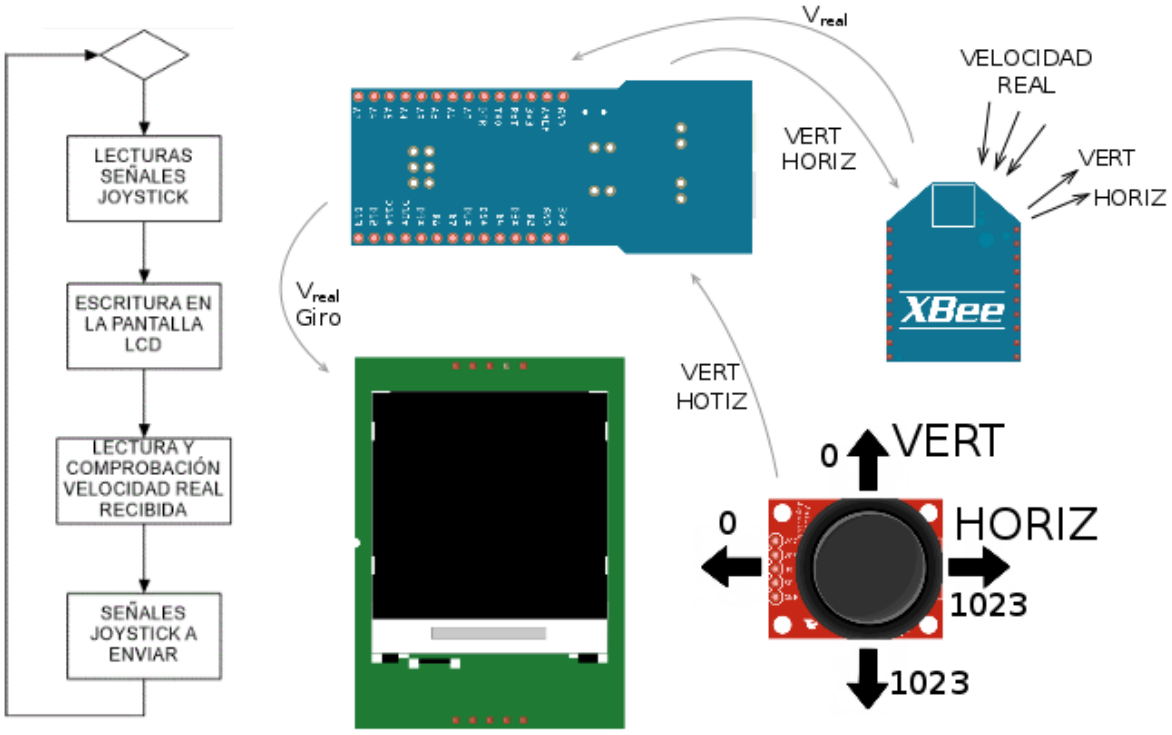


Figura 14: Esquema funcionamiento mando a distancia Rover

3 ESTIMACIÓN DE IRREGULARIDADES EN PERFILES ALEATORIOS MEDIANTE EL FILTRO DE KALMAN

En esta sección se explican los pasos seguidos para la obtención de un algoritmo que simula el proceso de auscultación de un vehículo auscultador de las características del Rover, como el de la Figura 15, que permita predecir la geometría de un perfil aleatorio a partir de señales de la IMU mediante el uso del filtro de Kalman. Para ello nos hemos basado en el algoritmo de auscultación propuesto por Hidalgo, Ros y Escalona en el documento "A Kalman filter-based algorithm for IMU signal fusion applied to track geometry estimation" [19].

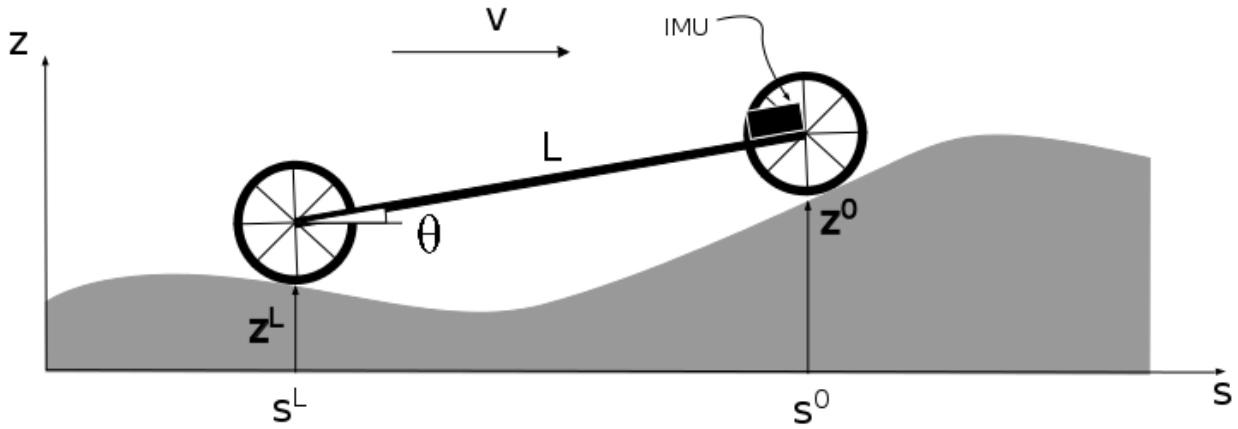


Figura 15: Modelo vehículo auscultador

De la Figura 15 se obtienen las ecuaciones (17) y (18) que representan la relación entre la geometría del perfil y las señales IMU:

$$\ddot{z}' = z''\dot{s}^2 + z'^0\ddot{s} \tag{17}$$

$$\dot{\theta} = \frac{z'^0 - z'^L}{L}\dot{s} \tag{18}$$

Adicionalmente con un decodificador situado en la rueda delantera podemos conocer en cada instante de tiempo el ángulo rotado por esta mediante la siguiente expresión, donde R es el radio de la rueda:

$$\varphi(t) = \frac{s^0(t)}{R} \tag{19}$$

De modo que usando la información procedente de los sensores, los parámetros de nuestro sistema y suponiendo los ruidos y errores de las medidas como variables estadísticas podemos estimar la geometría del perfil recorrido por el vehículo usando el filtro de Kalman discreto, descrito por las siguientes ecuaciones:

$$x_k = A_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \quad (20)$$

$$z_k = H_k x_k + v_k \quad (21)$$

Donde k es el instante de tiempo, x es el vector estado del sistema, u es el control input, z el vector de medida y w y v representan el ruido del proceso y de las medidas. A, H y G son matrices con las dimensiones apropiadas. El desarrollo en mayor detalle de las ecuaciones de estimación del algoritmo se encuentran en el documento anteriormente mencionado de la referencia [19].

El algoritmo se estructura como se muestra en la Figura 16. En primer lugar se genera un perfil aleatorio a partir del cual se calculan las señales teóricas de aceleración y velocidad angular que se obtendrían a través de la IMU para finalmente realizar la estimación del perfil aleatorio generado.

Figura 16: Algoritmo de auscultación

3.1 Generación de un perfil aleatorio

Para la generación de perfiles aleatorios se ha usado la teoría propuesta por Tyan, Hong, Tu y Jeng [20], en la que aproximan la geometría del perfil mediante una superposición de N funciones senoidales. La norma ISO sugiere determinados valores de longitudes de onda de las ondulaciones de la pista para la determinación de perfiles aleatorios, sin embargo para adaptar esta teoría a nuestro modelo se han tomado valores distintos, obtenidos en función de los rangos de trabajo de los sensores usados y de las dimensiones características de nuestro vehículo.

3.1.1 Definición de las irregularidades del camino aleatorio

Para calcular las longitudes de onda máximas y mínimas que permiten detectar las aceleraciones y velocidades angulares con los sensores que se usan en el vehículo se supone que este recorre un camino sinusoidal. El rango de trabajo del acelerómetro es +/-2 g y su sensibilidad de 0.25 mg. La mínima aceleración viene dada por:

$$\ddot{z}_{min} = A_{min}\left(\frac{2\pi v_{min}}{\lambda_{max}}\right)^2 \quad (22)$$

Donde A_{min} es la minima amplitud del perfil, v_{min} la velocidad minima de avance del vehículo y λ_{max} la máxima longitud de onda detectable por el acelerómetro. El motor Pololu del Rover gira a 350 rpm sin carga y la rueda tiene un radio de 0.045 m, luego la velocidad máxima que puede alcanzarse es de 1.65 m/s. Si se establece que la velocidad mínima de los ensayos sea 1 m/s y suponiéndose como λ_{max} 5 m, la amplitud mínima es A_{min} = 0.00155 m.

Para determinar la longitud de onda mínima de las ondulaciones del perfil aleatorio se debe considerar que la mayor rueda del vehículo atraviese sin problemas la parte del perfil con menor radio de curvatura. Considerando que el camino es sinusoidal en función de la coordenada s y teniendo en cuenta la curvatura de una curva definida por una función h se tienen las relaciones (23) y (24), donde A_h representa la amplitud del camino, C la curvatura K el radio de curvatura:

$$h(s) = A_h \sin\left(\frac{2\pi}{\lambda}s\right) \quad (23)$$

$$C = \frac{h''}{(1+h'^2)^{\frac{3}{2}}} = \frac{1}{K} \quad (24)$$

Debido a que el mayor radio de las ruedas del vehículo es de 0.045 m, se establece que el menor radio de curvatura del camino sea 0.05 m. De esta forma la máxima curvatura es:

$$C_{max} = A_h(\frac{2\pi}{\lambda})^2 = \frac{1}{0.05} \tag{25}$$

Despejando de esta expresión se obtiene el valor de longitud de onda mínima del camino aleatorio.

$$\lambda_{min} = 2\pi\sqrt{A_{min}0.05} = 0.055m \tag{26}$$

El giróscopo usado es un ITG 3200 con una resolución de 0.07(º/sec)/count (0.0012(rad/sec)/count). Teniendo en cuenta que la distancia entre ejes del vehículo es 0.16 m se expresa la velocidad angular mínima como:

$$\dot{\theta}_{min} = \frac{\sqrt{2}\,2\pi\,A_{min}\,v_{min}}{L\,\lambda_{max}} \tag{27}$$

Si se consideran las mismas condiciones que con el acelerómetro la amplitud mínima para el giróscopo es 1.08 e^{-4} m. Esta amplitud es menor que la obtenida con el acelerómetro, por lo que no se podrían detectar las aceleraciones generadas por ella dada la sensibilidad del acelerómetro que se está considerando. Por este motivo, la amplitud mínima usada para la generación de perfiles aleatorios es la obtenida con el acelerómetro y por tanto:

$$\lambda_{\min} = 0.055m \tag{28}$$

$$\lambda_{\max} = 5m \tag{29}$$

3.1.2 Función del camino aleatorio

Suponiendo que el vehículo viaja a velocidad constante, la coordenada vertical z del perfil aleatorio del camino de longitud L_{road} en función de la coordenada horizontal s puede aproximarse mediante la siguiente expresión.

$$z(s) = \sum_{i=1}^{N} A_i \sin(\Omega_i s - \phi_i) \tag{30}$$

Donde ϕ_i son tratados como variables aleatorias siguiendo una distribución uniforme en el intervalo [0, 2π) y A_i es la amplitud definida como:

$$A_i = \sqrt{\Phi(\Omega_i)\frac{\Delta\Omega}{\pi}},\ i = 1,2,\ldots N \tag{31}$$

Donde $\Phi(\Omega_i)$ es la PSD (Power Spectral Density) para Ω_i y $\Delta\Omega \triangleq \frac{\Omega_N - \Omega_1}{N-1}$ y donde las Ω_i se eligen como múltiplos de $\Delta\Omega$. La PSD tiene la siguiente expresión:

$$\Phi(\Omega) = \begin{cases} \Phi(\Omega_0)\Omega_0^{-2},\ 0 \le \Omega \le \Omega_1 \\ \Phi(\Omega_0)\left(\frac{\Omega}{\Omega_0}\right)^{-2},\ \Omega_1 \le \Omega \le \Omega_N \end{cases} \tag{32}$$

Donde $\Omega_0 = 1rad/m$ y $\Phi(\Omega_0)$ es el valor de la PSD para Ω_0, cuyo valor es función de la calidad de la geometría del perfil que se desea obtener. En nuestro caso se obtendrá una calidad pobre del perfil con el objetivo de que presente irregularidades notables para el vehículo, por lo que toma un valor de 64 e^{-6} m^3. Ω_1 y Ω_N se definen como:

$$\Omega_1 = \frac{2\pi}{\lambda_{máx}} = \frac{2\pi}{5} \tag{33}$$

$$\Omega_N = \frac{2\pi}{\lambda_{min}} = \frac{2\pi}{0.055} \tag{34}$$

Donde λ_{min} y $\lambda_{máx}$ se obtienen del apartado anterior, 3.1.1 Definición de las irregularidades del camino aleatorio, en función de las características del vehículo mientras que N se obtiene redondeando al entero inferior o superior la siguiente expresión, de modo que:

$$N = \frac{L_{road}}{\lambda_{min}} \tag{35}$$

Implementando estas ecuaciones en Matlab se ha desarrollado la función que se muestra en el Código 10. Conocidas tanto las irregularidades del terreno como la longitud del perfil se obtiene la geometría del perfil aleatorio.

Código 10: Generación perfiles aleatorios

```
function random_road(lm,lM,s,L)

N = ceil(s(length(s))/lm);
w_N = 2*pi/lm;
w_1 = 2*pi/lM;
dw = (w_N-w_1)/(N-1);
w = w_1:dw:w_N;
sigma = 0.016;
S0 = 64e-6;

for i=1:length(w)
    if (w(i)<= w_1)
        S(i) = S0*w_1^(-2);
    else
        S(i) = S0*(w(i))^(-2);
    end
    A(i) = (S(i)*(dw)/pi)^0.5;
    fi(i) = random('uniform',0,2*pi);
end

z = zeros(1,length(s));
for i=1:length(w)
    z = z + A(i)*sin(w(i)*s+fi(i));
end
```

3.2 Obtención señales modelo

Suponiendo que el Rover avanza con una determinada velocidad por la geometría obtenida mediante el procedimiento descrito en el apartado anterior definida por z, se obtiene la aceleración vertical sufrida por el acelerómetro ante las irregularidades del terreno como se indica en las siguientes expresiones:

$$\dot{z}(t) = \frac{dz}{dt} = \frac{dz}{ds}\frac{ds}{dt} = z'(s)\dot{s}(t) \tag{36}$$

$$\ddot{z}(t) = \frac{d^2z}{dt^2} = \frac{d^2z}{ds^2}\frac{ds}{dt}\dot{s}(t) + z'(s)\frac{d\dot{s}}{dt} = z''(s)\dot{s}^2(t) + z'(s)\ddot{s}(t) \tag{37}$$

Donde $\dot{s}$ y $\ddot{s}$ son la velocidad y aceleración de avance del vehículo y z' y z'' son la derivada primera y segunda de la coordenada vertical z de la geometría del perfil respecto a la coordenada longitudinal s, cuyas expresiones son:

$$z'(s) = \sum_{i=1}^{N} A_i \Omega_i \cos(\Omega_i s - \phi_i) \tag{38}$$

$$z''(s) = -\sum_{i=1}^{N} A_i \Omega_i^{\,2} \sin(\Omega_i s - \phi_i) \tag{39}$$

También podemos conocer las velocidad angular sufrida por el giróscopo ante las irregularidades del perfil mediante la siguiente expresión:

$$\dot{\theta}(t) = \frac{d\theta}{ds}\frac{ds}{dt} = \theta'(s)\dot{s}(t) \tag{40}$$

Donde θ' se relaciona con el perfil recorrido de la siguiente forma:

$$\theta'(s) = \frac{z'(s) - z'(s-L)}{L} \tag{41}$$

Donde $z'(s)$ es la derivada primera de la coordenada vertical del perfil respecto a s en la rueda delantera del vehículo (38) y $z'(s-L)$ la derivada primera de la coordenada vertical del perfil respecto a s en la rueda trasera (42), siendo L la longitud del mismo.

$$z'(s-L) = \sum_{i=1}^{N} A_i \Omega_i \cos(\Omega_i (s-L) - \phi_i) \tag{42}$$

El cálculo de las señales modelo se presenta en el Código 11 de modo que hpp es la aceleración que mediría el acelerómetro durante el funcionamiento del Rover mientras que ww sería la velocidad angular que mediría el giróscopo.

Código 11: Cálculo señales modelo

```
zl=z;
zlp=zeros(1,length(s));
zlpp=zeros(1,length(s));
zlpL=zeros(1,length(s));
for i=1:length(w)
    zlp=zlp+A(i)*w(i)*cos(w(i)*s+fi(i));
    zlpp=zlpp-A(i)*(w(i))^(2)*sin(w(i)*s+fi(i));
    zlpL=zlpL+A(i)*w(i)*cos(w(i)*(s-L)+fi(i));
end

hpp = zeros(length(s),1);
ww = zeros(length(s),1);
for i=1:length(s)
    hpp(i) = Vms(i)^2*zlpp(i) + Ams(i)*zlp(i);
    ww(i) = Vms(i)*(1/L)*(zlp(i)-zlpL(i));
end
```

Una vez obtenidas las señales del acelerómetro y giróscopo podemos predecir las irregularidades de la geometría del perfil recorrido por el Rover haciendo uso del Filtro de Kalman.

4 PRUEBAS EXPERIMENTALES

En esta sección se describen los pasos realizados para la ejecución de los ensayos experimentales usando el Rover para estimar la geometría del perfil del terreno sobre el que este avanza.

Los ensayos consisten en giar el vehículo auscultador por un terreno conocido de 5 m de longitud con recoger los datos de los sensores inerciales durante su avance del Rover con el objetivo de realizar, mediante el algoritmo de auscultación, la estimación del perfil del terreno en cuestión.

4.1 Medición del perfil de un tramo de pista

Con el objetivo de comparar el resultado de los ensayos de auscultación a realizar con el Rover se midió el perfil de un tramo de la pista de baloncesto situada en la parte trasera de los laboratorios de la Escuela Técnica Superior de Ingeniería de Sevilla.

Para conocer el perfil del terreno donde realizar los ensayos de auscultación se siguieron los siguientes pasos:

- Marcar los puntos a medir sobre el terreno que se desea conocer: Para ello se marcó en el terreno un rectángulo de 5 m de longitud por 0.34 m de ancho, en su interior se realizaron las marcas de los puntos a medir que forman dos carriles por los que avanzarán cada una de las dos ruedas delanteras del Rover. Cada uno de estos carriles, separados 0.10 m, está formado por cuatro líneas de puntos distanciadas 0.04 m entre sí y cada una de estas líneas está compuesta por 126 puntos separados por la misma distancia que la líneas. De este modo se reproduce sobre el terreno una malla de 1008 puntos que son medidos con una estación total. En la Figura 17 se representa esquemáticamente las marcas realizadas sobre la pista de baloncesto.

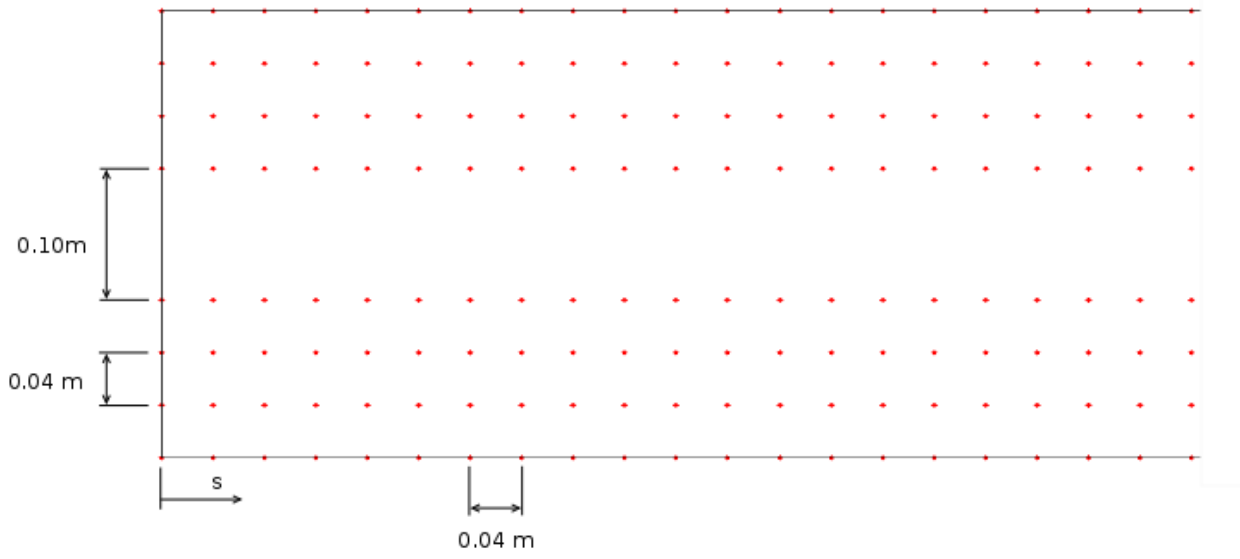


Figura 17: Esquema de las marcas realizadas en la pista de baloncesto

- Mediciones de los puntos marcados sobre el terreno: Para ello se utiliza una estación total GST101 de Leica Geosystems AG, como la que se observa en la Figura 18 (a), que mide respecto a su posición las coordenadas espaciales X, Y y Z de los puntos marcados en la pista. Para ello emite una onda electromagnética hacia un prisma, como el de la Figura 18 (b), colocado sobre el punto a medir, esta onda rebota en el prisma y regresa al equipo, calculándose la posición del punto. El equipo tiene una precisión de 7'' para la medición de angulos y se utilizó como modo de medición EDM el IR estándar [21]. En la Figura 18 (c) se observa la medición de un punto mientras que en la siguiente, Figura 19, se representan las coordenadas de los puntos medidos que conforman la supercicie de ensayo.

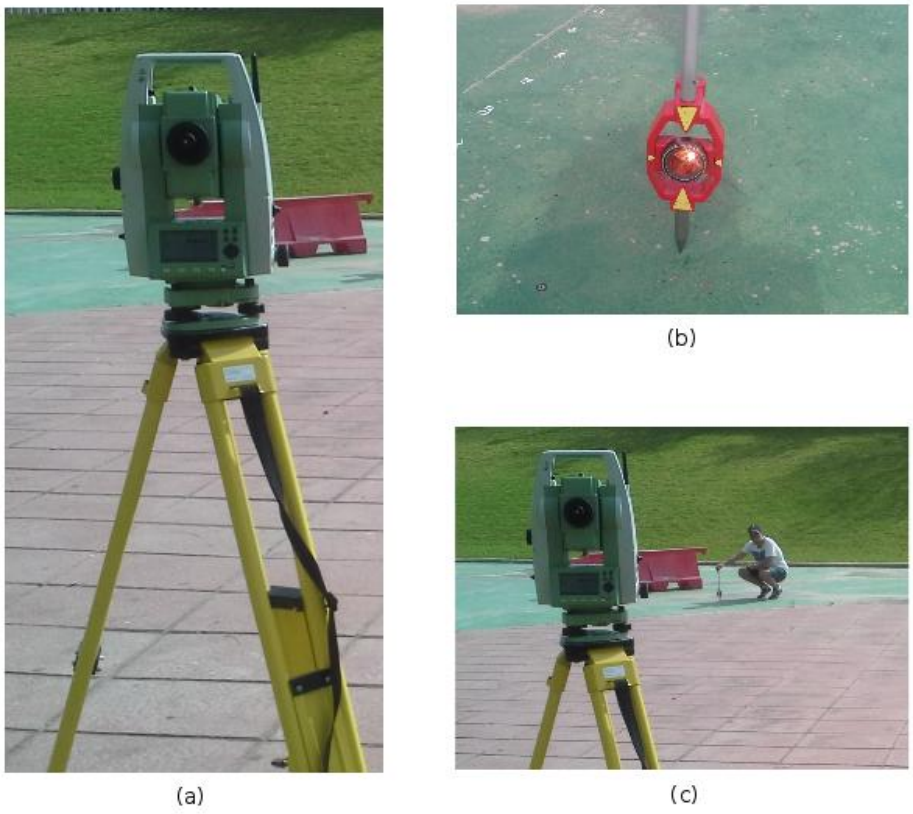


Figura 18: Medición estación total

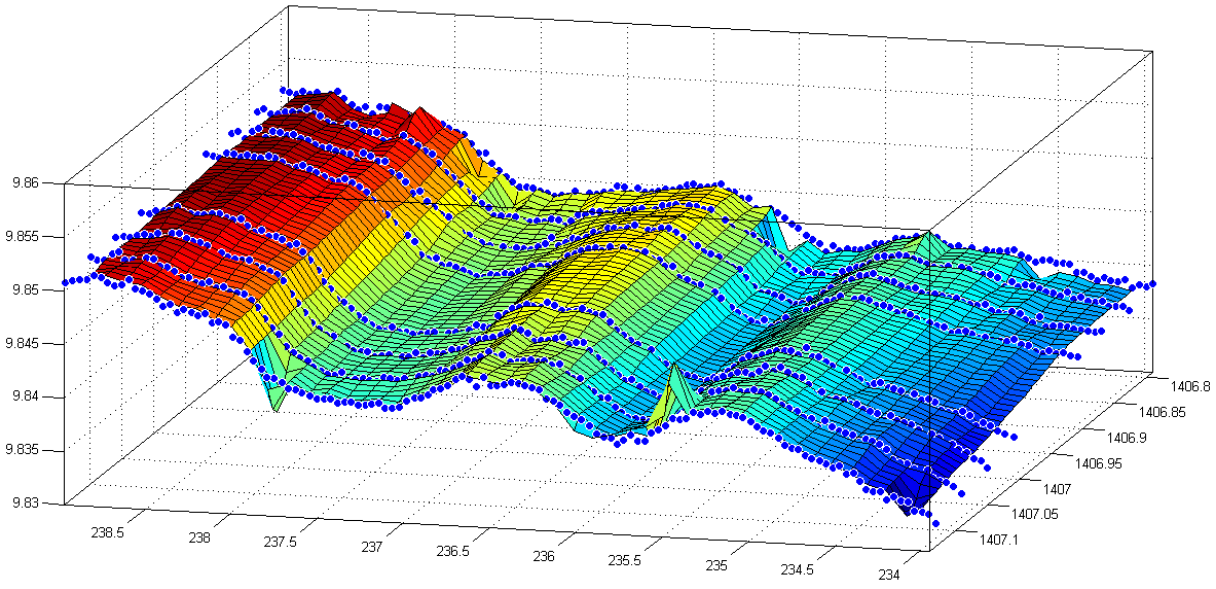


Figura 19: Representación de los puntos medidos en la pista

Con el fin de eliminar posibles errores en la medida, las coordenadas de los puntos representados en la Figura 19 fueron filtradas, representando del resultado en la Figura 20.

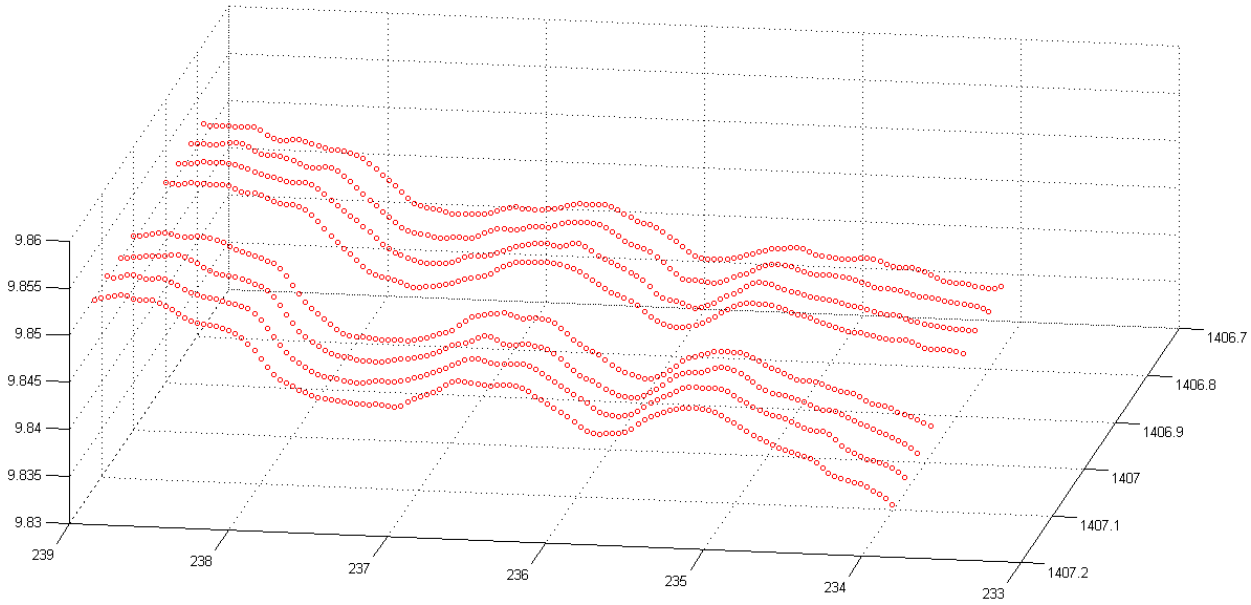


Figura 20: Representación de los puntos medidos en la pista filtrados

- Interpolación de los puntos obtenidos para generar una superficie conocida: Para ello se ha utilizado un programa de generación de superficies llamado SurGe Project Manager [22]. Con este programa se interpolan los datos medidos con la estación total (Figura 19) para obtener una superficie, representada en la Figura 21, con un número mayor de puntos cuyas coordenadas son conocidas.

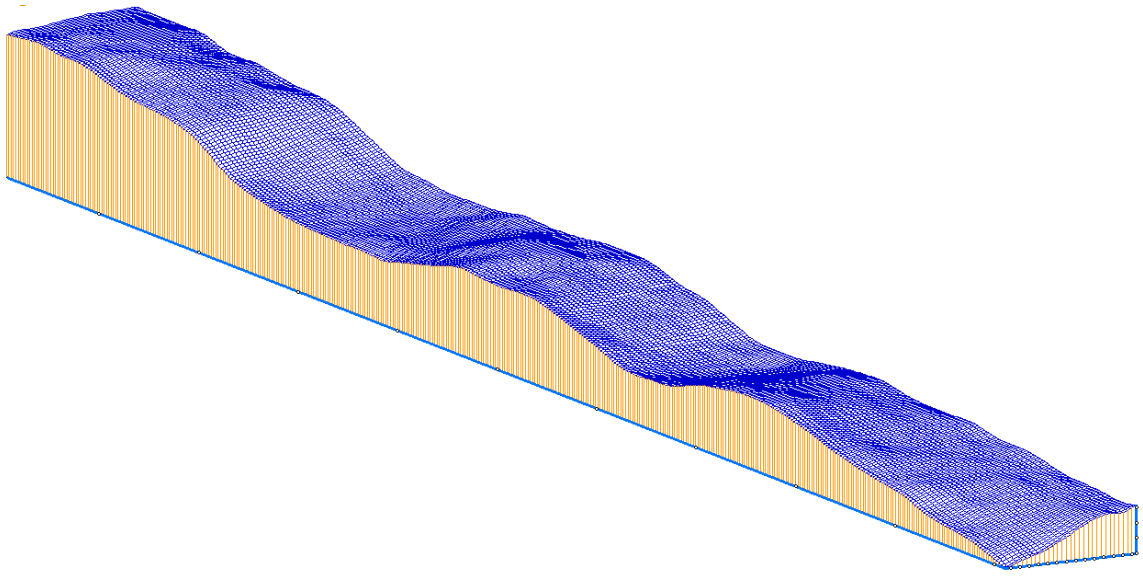


Figura 21: Vista 3D interpolación

- Representación de la línea central de perfil del terreno: Debido a que la IMU se encuentra colocada en el centro del vehículo auscultador durante los ensayos esta recorrerá la línea central del perfil del terreno, de modo que surge la necesidad de conocer las coordenadas verticales de los puntos de la interpolación que componen la líne central del perfil. De esta forma se obtiene la geometría del perfil del terreno donde se realizarán los ensayos de auscultación, que se representa en la Figura 22.

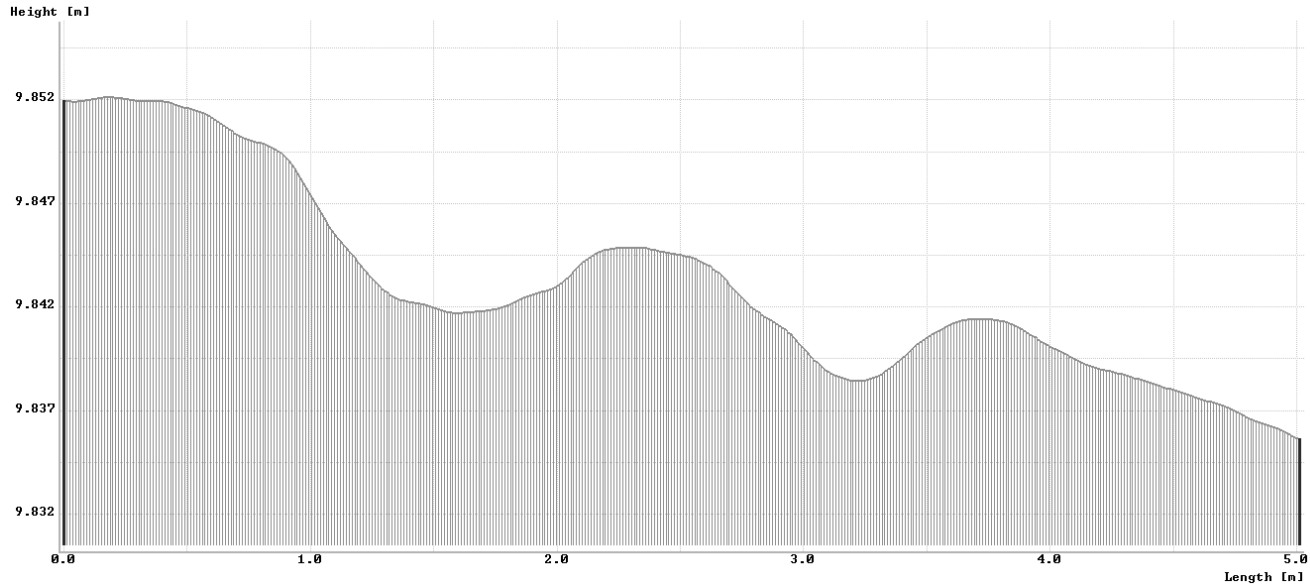


Figura 22: Perfil del terreno donde se realizan los ensayos de auscultación

4.2 Ensayos sobre el tramo medido

Los ensayos de auscultación consisten en recorrer con el Rover el tramo de pista de baloncesto de la Escuela medido como se describe en el apartado anterior 4.1. Con el fin de recorrer los 5 m de longitud del terreno a velocidad constante el vehículo arranca dos metros antes del punto inicial del tramo medido durante los cuales acelera hasta alcanzar la velocidad del ensayo y se detiene una vez recorrida la zona medida. En la Figura 23 se representa un esquema de las posiciones inicial y final del vehículo.

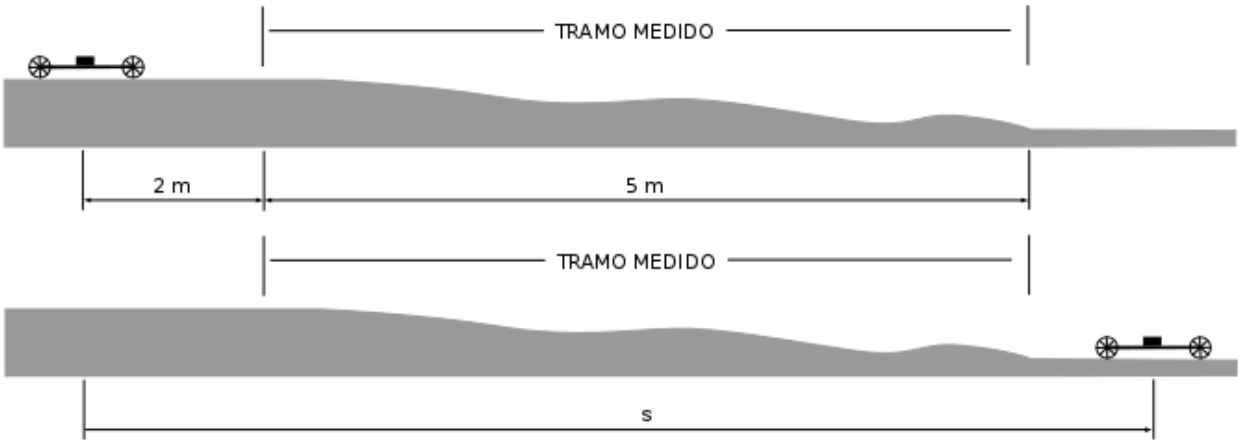


Figura 23: Posición inicial y final del Rover en los ensayos

Como resultado del proceso de auscultación el software que controla el Rover genera un fichero de texto que contiene, como se vió en la Figura 13, las velocidades de cada motor y las aceleraciones y velocidades angulares sufridas por el vehículo en cada instante de tiempo. Sabiendo que la IMU se encuentra orientada como se observa en la Figura 24 se concluye que los datos necesarios para estimar la geometría recorrida son la aceleración en dirección Z y la velocidad angular en dirección X, también son necesarias las velocidades de ambos motores ya que con ellas obtenemos el perfil de velocidad del vehículo durante el proceso de auscultación, la aceleración en dirección y nos da la aceleración en el sentido de avance del vehículo.

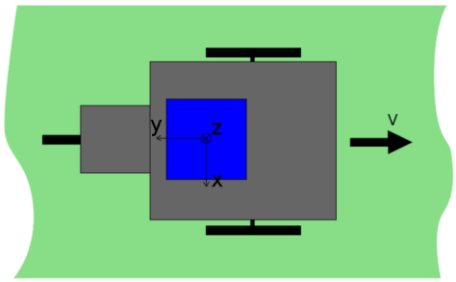


Figura 24: Orientación IMU

Debido a que no se disponía de tiempo suficiente para aplazar la realización de estos ensayos, durante los mismos se tuvo un problema con el buffer encargado del almacenamiento de los datos medidos por el vehículo. Este problema era que después de un tiempo determinado la frecuencia de adquisición de los datos que se almacenaban aumentaba, volviéndose irregular. Debido a la importancia de tener una adquisición de datos a intervalos de tiempo constante que nos permita el uso de la FFT, con el objetivo de evitar este problema con el buffer se estableció una frecuencia de adquisión de datos menor de modo que nos permitiera recorrer el camino medido en un tiempo menor al que la frecuencia de muestreo empezaba a ser irregular. Teniendo en cuenta el tamaño del buffer y que las velocidades que desarrolla el Rover se establecen en torno a 1 m/s se decidió establecer la toma de datos con un periodo de 0.02 s ya que establecer un periodo de adquisición más bajo nos obligaba a realizar los ensayos a velocidades mayores. Estas pruebas a mayores velocidades no pudieron realizarse con éxito debido a que el tramo de camino medido es muy estrecho y esto dificultaba el poder controlar el vehículo para mantenerlo en la trayectoria deseada. Posteriormente a la realización de los ensayos se solucionó parcialmente este problema simplificando al máximo las funciones de lectura y escritura de los datos en el buffer. No obstante, aún habiendo solucionado el problema del buffer, se ha observado que los incrementos de tiempo no siempre son regulares sino que aproximadamente cada 40 datos almacenados aparece aparece un intervalo de tiempo mayor que los establecidos. Esto implica que el valor que debía ser medido en ese instante no se almacena. Debido a este fenómeno en el apartado 4.3 se realiza un pequeño análisis para evaluar como afecta el hecho de la aparición de estos intervalos de tiempo mayores en la aplicación de la FFT.

De esta forma se realizaron dos ensayos para predecir la geometría del perfil medido que denominaremos como Ensayo 1 y Ensayo 2. El Código 12 representa el tratado de las señales del Ensayo 2 (siguiéndose una metodología similar para tratar las señales del Ensayo 1). El primer paso es eliminar tanto los datos incorrectos por el llenado del buffer como las primeras lecturas hasta que se empieza a medir el tiempo. A continuación se eliminan los datos no reales de velocidad. El siguiente paso es realizar un filtrado de la velocidad del Rover durante el ensayo con el objetivo de eliminar el ruido en la medida de los encoderes, una vez filtradas se eliminan los datos de reposo del vehículo ya que estos no aportan información al proceso de auscultación teniendo en cuenta que el origen de tiempo es t=0. A continuación se realiza el filtrado de las señales de aceleración y velocidad angular, a las primeras se les realiza un fitrado paso alto mientras que a las segundas uno paso bajo. A pesar de establecer un periodo de adquisición de datos constante incluso durante el correcto funcionamiento del buffer cada cierto numero de medidas se produce una con un periodo mayor siguiendo tras esta con el periodo establecido, de modo que se interpolan los datos para obtener señales con un periodo de muestreo constante, necesario para poder aplicar la FFT para comparar el contenido en frecuencia de los perfiles estimados y real. Una vez interpolados se calcula la longitud recorreida por el Rover durante el ensayo. Por último se obtienen las señales de la IMU en m/s y rad/s.

Una vez tratadas las señales procedentes de la IMU y los decodificadores, podemos predecir la geometría del perfil del terreno recorrido por el Rover usando, al igual que de forma teórica, el filtro de Kalman. En la siguiente sección 5 Resultados se muestran los resultados obtenidos de los dos ensayos.

Código 12: Tratado de señales de los ensayos para el uso del filtro de Kalman

```
% Cargado de datos
fileID=...
    fopen('MSCIMU_0p02_5_2m.txt','r');
formatSpec = '%f %f %f %f %f %f';
A = fscanf(fileID,formatSpec);
j = 1;
for i=1:9:length(A)
   tv(j) = A(i);
   va(j) = A(i+1);
   vb(j) = A(i+2);
   accy(j) = A(i+3);
   accz(j) = A(i+5);
   Gyrox(j) = A(i+7);
   j = j+1;
end

% Eliminación de datos incorrectos
tv(1:7)=[]; tv(654:end)=[];
va(1:7)=[]; va(654:end)=[];
vb(1:7)=[]; vb(654:end)=[];
accy(1:7)=[]; accy(654:end)=[];
accz(1:7)=[]; accz(654:end)=[];
Gyrox(1:7)=[];Gyrox(654:end)=[];

% Eliminación datos incorrectos de
velocidad
va(254-7) = (va(253-7)+...
                va(255-7))*0.5;
va(252-7) = (va(251-7)+...
                va(253-7))*0.5;
vb(254-7) = (vb(253-7)+...
                vb(255-7))*0.5;
vb(252-7) = (vb(251-7)+...
                 vb(253-7))*0.5;

radio_m = 0.045;
Va_m = (va*2.0*pi*radio_m)/60.0;
Vb_m = (vb*2.0*pi*radio_m)/60.0;
vms = (Va_m + Vb_m)*0.5;

% Filtrado de las velocidad
[b,a]=butter(9,0.1);
Vms = filter(b,a,vms);

% Eliminación de velocidades nulas
for i=1:length(tv)
   if Vms(i)< 0.0001
      ifirst = i;
      i = length(tv);
   end
end

tfinal = tv(end);
tini = tv(ifirst-1);
tv(1:ifirst-1)=[];
tv(1:end) = tv(1:end) - tini;
Vms(1:ifirst-1)=[];
accy(1:ifirst-1)=[];
accz(1:ifirst-1)=[];
Gyrox(1:ifirst-1)=[];

% Filtrado señales
fn = 25;
fc = 24.7;
[b,a]=butter(4,fc/fn,'high');
accz_filt = filter(b,a,accz);
accz = accz_filt;

[b,a]=butter(4,fc/fn);
Gyrox_filt = filter(b,a,Gyrox);
Gyrox = Gyrox_filt;

fc = 12;
[b,a]=butter(4,fc/fn,'high');
accy_filt = filter(b,a,accy);
accy = accy_filt;

% Interpolación datos
delts = 0.02;
tint=0:delts:tv(end);
Vms_int= spline(tv,Vms,tint);
accz_int= spline(tv,accz,tint);
Gyrox_int= spline(tv,Gyrox,tint);
accy_int= spline(tv,accy,tint);
Vms = Vms_int;
accz = accz_int;
Gyrox = Gyrox_int;
accy = accy_int;

% Cálculo longitud recorrida
s = zeros(1,length(tv));
s(1) = 0;
for i=1:length(tv)-1
   s(i+1) = s(i)+ Vms(i)*delts;
endsini(1)=0;
L=0.16;
LL = L + L*0.2;
Np_max=10;
dsini = LL/(Np_max-1);
for i=2:Np_max
   sini(i) = sini(i-1) + dsini;
end
s = s +sini(end);

% Señales para la estimación
hpp = (accz)*9.8;
ww = -Gyrox*pi/180;
Ams = accy*9.8;
```

4.3 Pérdida de datos

La aparición aproximadamente regular de intevalos de tiempo mayores a los establecidos implica que los valores que debían ser medido en esos instantes de tiempo no son medidos y por lo tanto implica una perdida de datos en las mediciones realizadas. Con el fin de estudiar el efecto que esta perdida de datos conlleva a la estimación de los perfiles mediante el filtro de Kalman y determinar la validez del uso de la transformada de Fourier frente a intervalos de tiempo no constantes, se simula en este apartado este efecto usando el algoritmo teórico de la sección 3. Para ello a las señales teóricas calculadas a partir del perfil aleatorio generado se le eliminan datos a una cierta frecuencia que simula esta perdida de datos debida a los incrementos de tiempo mayores para, posteriormente, realizar la estimación del perfil con estas señales modificadas.

A continuación se describe la metodología usada para modificar las señales teóricas. Dada una señal teórica cualquiera (aceleración o velocidad angular) con intervalos de tiempo constante como la que se muestra en la Figura 25, donde P1, P2, …, P10 son los datos medidos y t1, t2, …, t10 son los instantes de tiempo en los que se miden dichos puntos y definiendo Δt1=t2-t1, Δt2=t3-t2, …, Δt9=t10-t9 siendo todos estos iguales, cada cierto número de datos se elimina el valor en ese instante para generar un intervalo de tiempo mayor, donde ahora Δ't1=t2-t1, Δ't2=t3-t2, Δ't3=t5-t3, Δ't4=t5-t6 Δ't5=t8-t6, Δ't6=t9-t8 y Δ't7=t10-t9, donde Δ't3 y Δ't5 son mayores que el resto.

Debido al uso de Fourier que requiere intervalos de tiempo regulares, por lo que se reconstruye la señal haciendo que de nuevo sus intervalos de tiempo sean todos constantes por lo que ahora, como se observa en la Figura 26, P'1=P1, P'2=P2, P'3=P3, P'4=P5, P'5=P6, P'6=P8, P'7=P9 y P'8=P10.

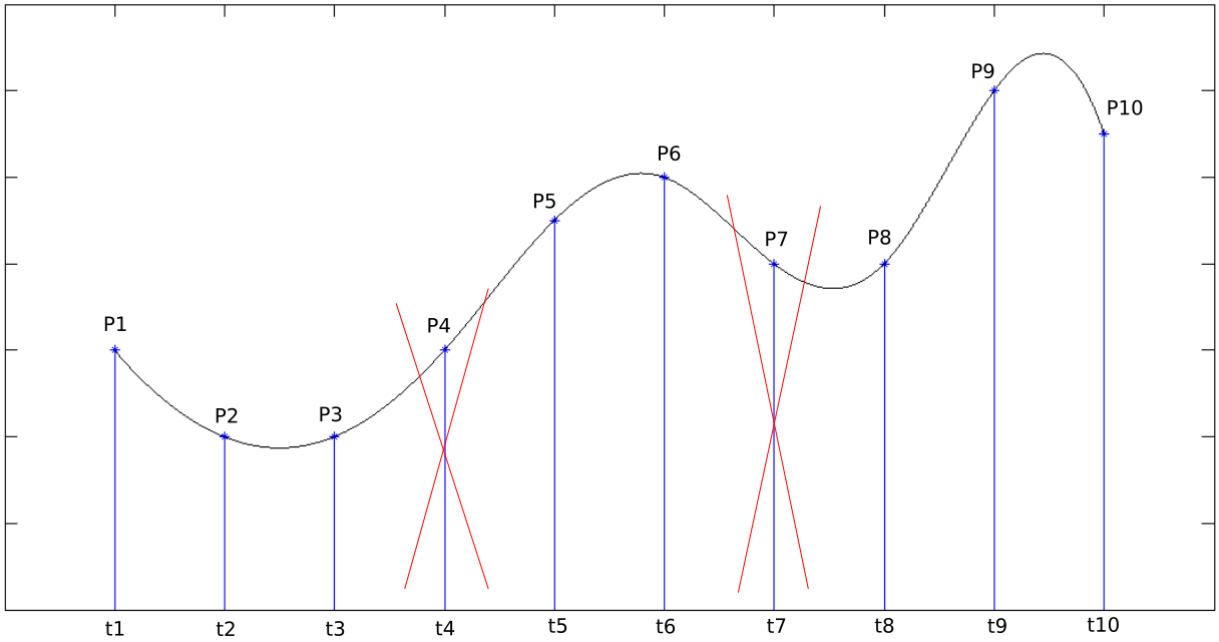


Figura 25: Señal teórica

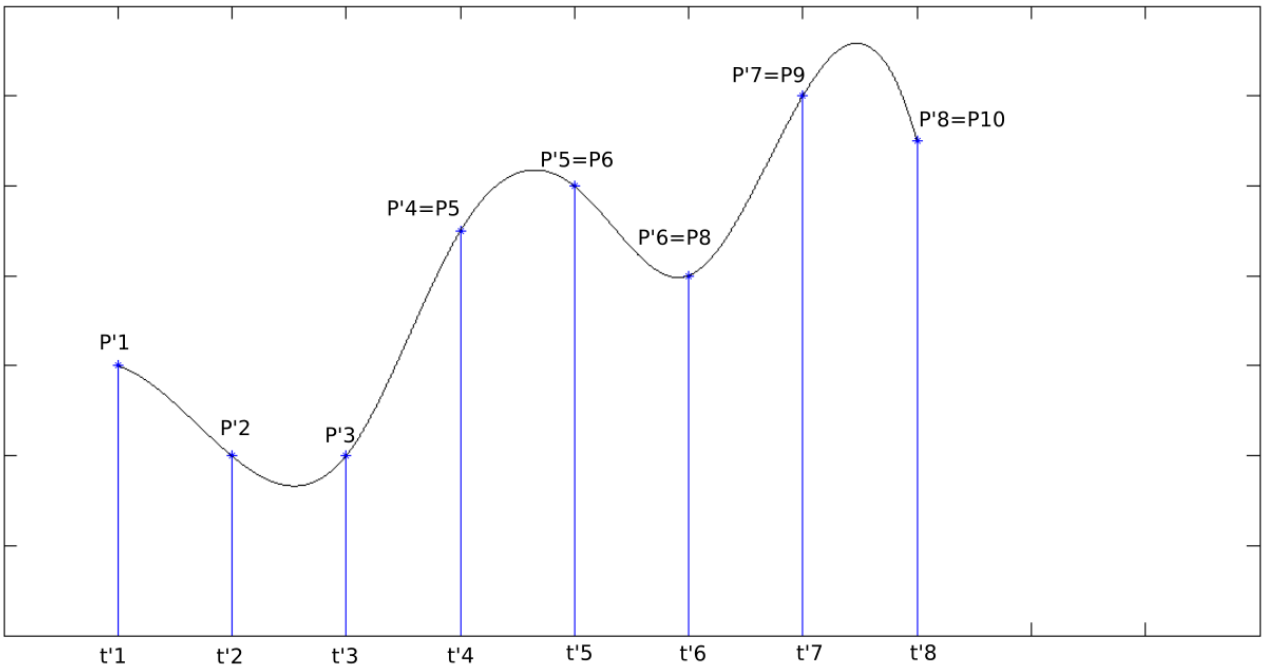


Figura 26: Señal con pérdida de datos

Para evaluar el efecto que esta perdida de datos tiene sobre la estimación del perfil y el uso de la FFT se realiza la estimación de perfiles aleatorios aplicando esta metodología a todas las señales que intervienen en la estimación mediante el filtro de Kalman (vector de tiempos, de aceleraciones, de velocidades angulares y la señal de los decodificadores).

5 RESULTADOS

En esta sección se presentan los resultados obtenidos aplicando el algoritmo de auscultación descrito en la sección 3 Estimación de Irregularidades en Perfiles Aleatorios Mediante el Filtro de Kalman para estimar las irregularidades de perfiles verticales. En primer lugar se realiza la estimación de un perfil aleatorio teórico. En segundo lugar se realiza la estimación del perfil real sobre el cual se han realizado los ensayos de auscultación con el Rover.

5.1 Estimación de perfiles teóricos

En esta sección se muestra el resultado de la simulación del proceso de auscultación teórica de un vehículo auscultador de las características del Rover que avanza a velocidad constante de 1 m/s sobre un perfil aleatorio, de 300 m de longitud. En color rojo se muestra el resultado de la estimación mientras que en azul se muestran los valores teóricos.

En la Figura 27 se muestra la evolución de la coordenada vertical de las irregularidades del perfil en función de la distancia recorrida por el vehículo y el perfil aleatorio representado por la expresión (30) de la sección 3.1.2. En la Figura 28 se observa el contenido en frecuencia de los perfiles mostrados en la Figura 27. Se ha supuesto para la aceleración un bias senoidal con una amplitud del 5% de la aceleración máxima con un periodo de 300 s, en la Figura 29 se observa la estimación del filtro de Kalman.

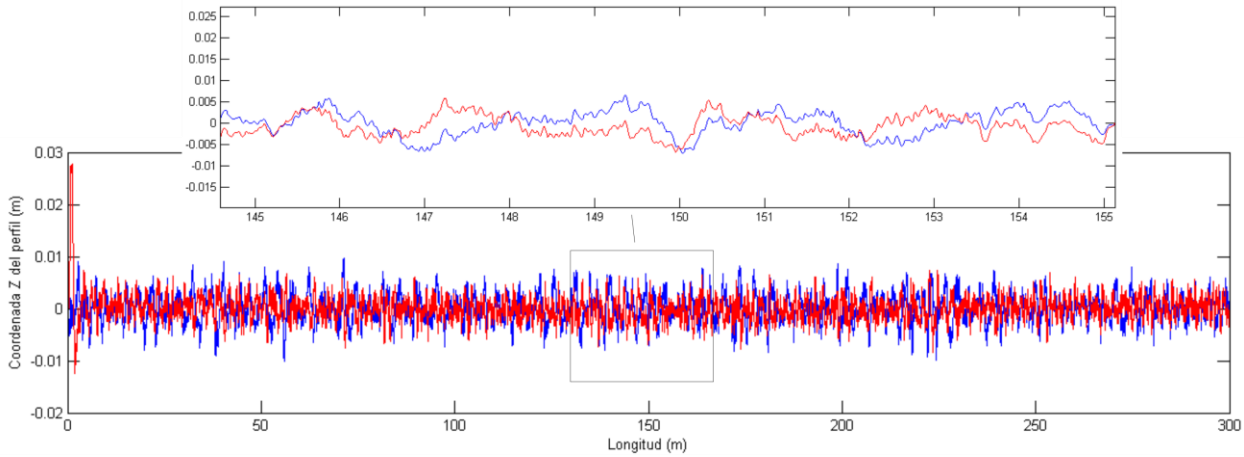


Figura 27: Estimación perfil aleatorio

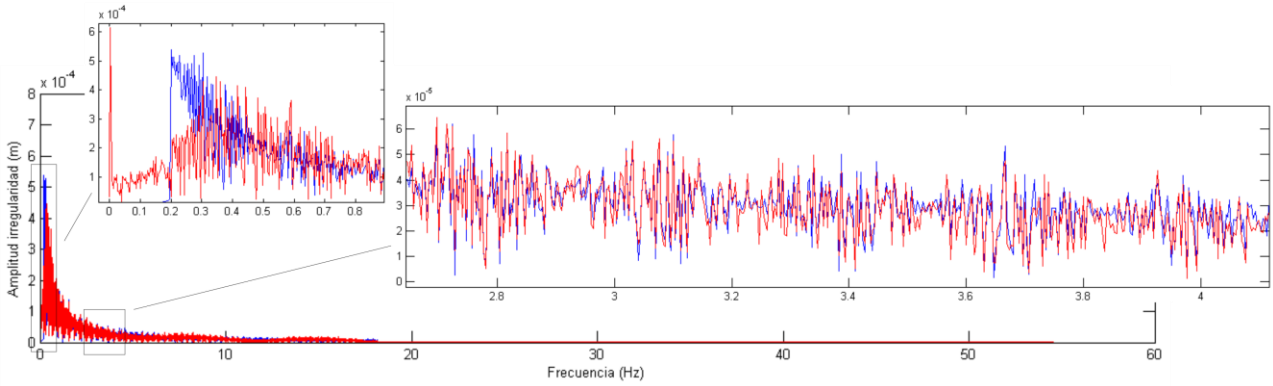


Figura 28: Contenido en frecuencia estimación perfil aleatorio

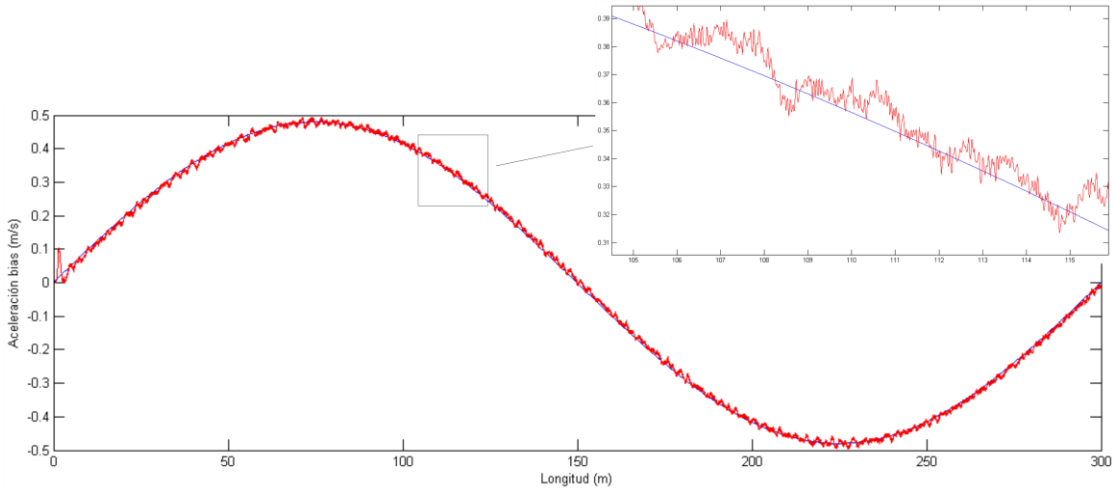


Figura 29: Bias acelerómetro

5.2 Estimación perfil real

Los resultados de los dos ensayos de auscultación de la geometría vertical de la pista de baloncesto medida en la Escuela usando el Rover como vehículo auscultador se presentan en este apartado. En azul se muestran los resultados obtenidos en el Ensayo 1 y los obtenidos en el Ensayo 2 en Rojo, mientras que en negro se representan los datos de los valores reales del perfil medido con la estación total.

En la Figura 30 se muestran en función del tiempo las aceleraciones verticales y las velocidades angulares obtenidas con el Rover al recorrer el camino medido, en la Figura 31 se muestra el contenido en frecuencia de las aceleraciones mostradas en la Figura 30. Se observa que los valores de aceleraciones son muy elevados también a baja frecuencias para las velocidades a las que avanza el vehículo, por lo que se sospecha que estos valores pueden no ser correctos, esto puede estar debido a una mala calibración del acelerómetro, que el acelerómetro tenga algún fallo o que en realidad el vehículo avance dando pequeños botes que no son apreciables a simple vista debido a que no tiene amortiguación y el suelo es relativamente rugoso. No obtante esto no son más que suposiciones y se está trabajando para solucionar este probema. Por este motivo se ha decicido filtrar las señales procedentes de los sensores con el fin de obtener una mejor concordancia entre los perfiles estimados y el perfil real, realizando un filtro paso alto a las aceleraciones y uno paso bajo a las velocidades angulares.

En la Figura 30 también se observa que a partir de un determinado tiempo los incrementos de tiempo en la adquisición de los datos de los sensores aunentan volviéndose irregulares, esto se debe al probema con el buffer que se tuvo durante la realización de los ensayos, que se menciona en la sección 4.2. Debido a la necesidad de que los incrementos de tiempo sean constantes para el uso de Fourier de decidió eliminar dichos datos. También se decidió interpolar las señales obtenidas con el fin de solucionar le problema de los puntuales intervalos de tiempo irregulares que introduce la Mbed durante las mediciones.

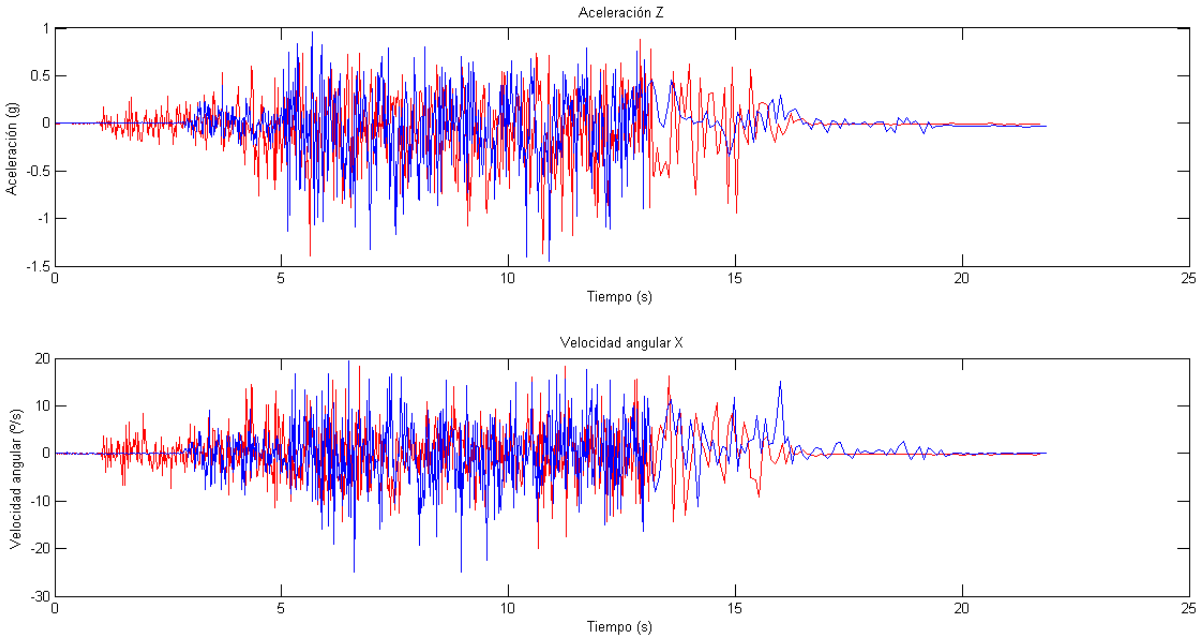


Figura 30: Aceleración verical y velocidad angular x obtenidas en los ensayos

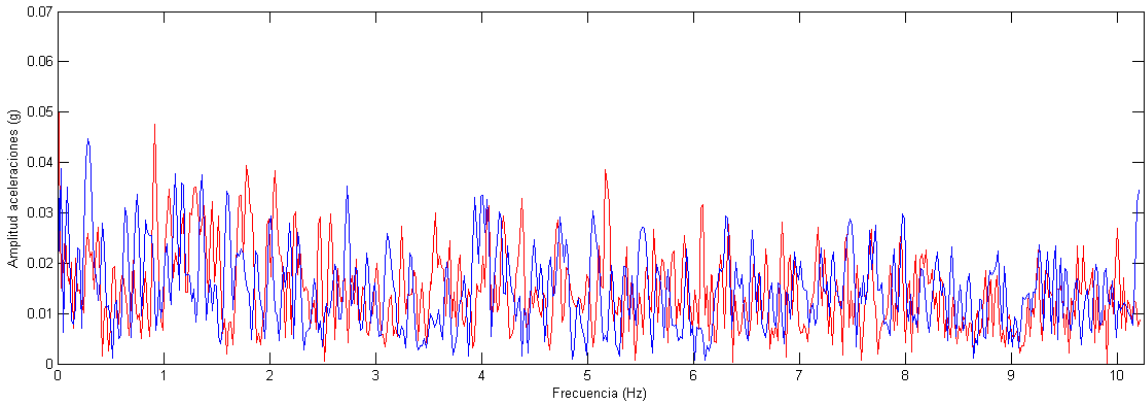


Figura 31: Contenido en frecuencia de las aceleraciones verticales de los ensayos

En la Figura 32 se muestran los perfiles de velocidad de ambos ensayos obtenidos como la media de las velocidades de cada rueda del vehículo. En ella se observa un pico de velocidad en el Ensayo 2 de unos 25 m/s. debido a que la velocidad máxima que desarrolla el vehiculo es de 1.65 m/s este pico no es un dato real de velocidad por lo que debe ser eliminado. Para ello se aproxima la velocidad en ese instante como la media del instante anterior y posterior.

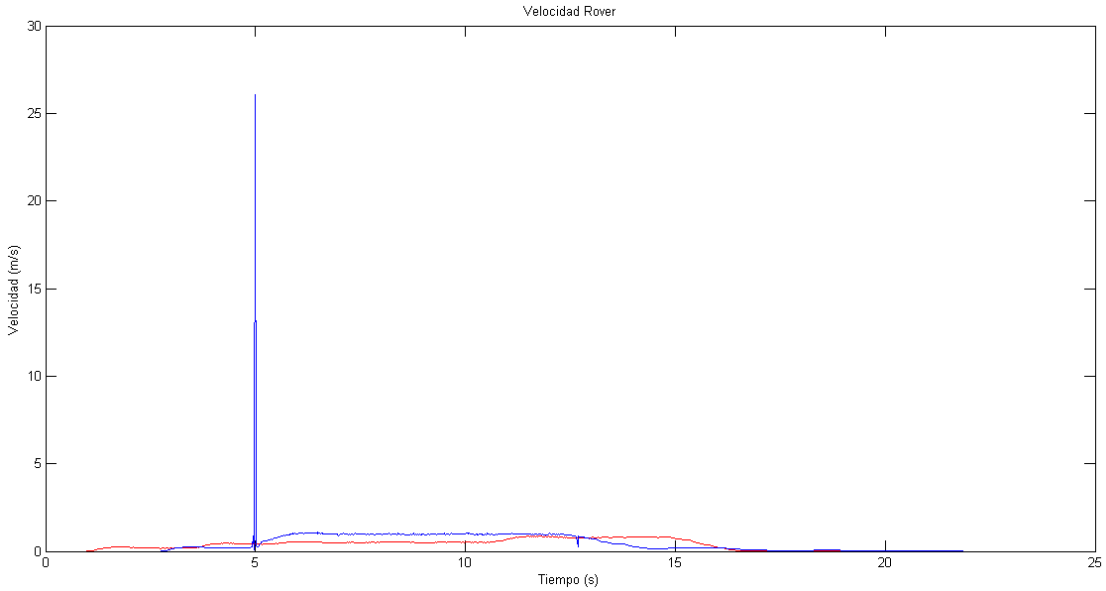


Figura 32: Perfiles de velocidad de los ensayos

En la Figura 33 se representan en función de la distancia recorrida por el Rover tanto la evolución de la coordenada z de las irregularidades de los perfiles estimados con el filtro de Kalman como el perfil real sobre el que se realizaron los ensayos, se observa que el perfil estimado en el Ensayo 1 no termina de recorrer el terremo marcado por el ensayo, esto es debido que no se alcanzó la velocidad necesaria para recorrer toda la longitud del terreno antes de que se llenara el buffer. Notese que aunque se ha realizado la estimación de la irregularidades del terreno en toda la longitud recorrida por el vehículo auscultador, el perfil conocido se encuentra entre los metros dos y siete de modo que tanto los dos primeros metros de los ensayos como el tramo final del Ensayo 2 no pueden ser comparados con los valores reales. Por este motivo se encuentra ampliada la zona conocida del perfil real. En la Figura 34 se muestra el contenido en frecuencia de los perfiles mostrados en la Figura 33.

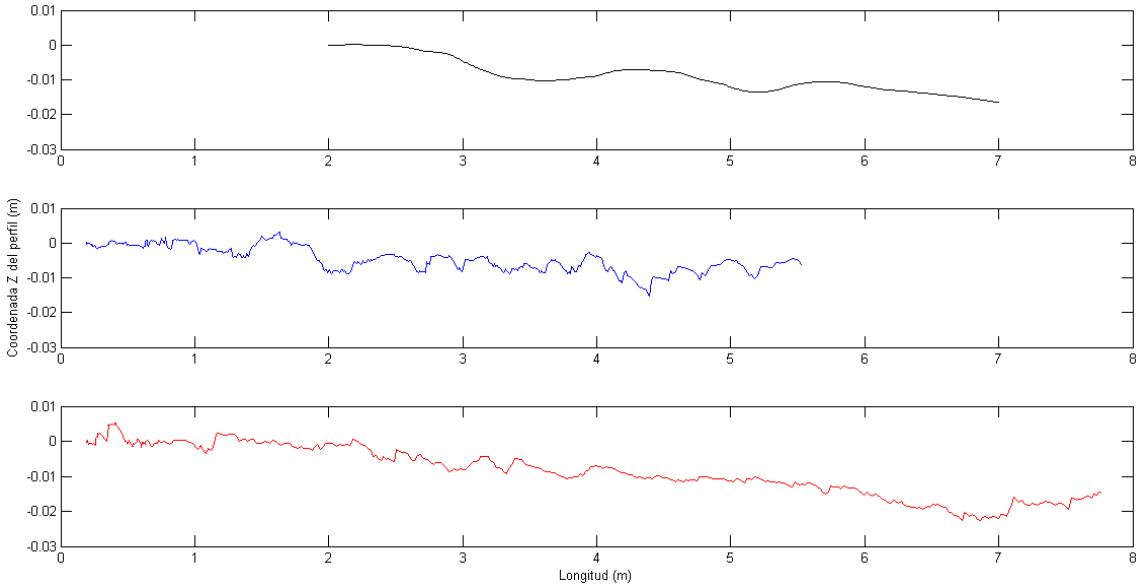


Figura 33: Estimación perfiles reales

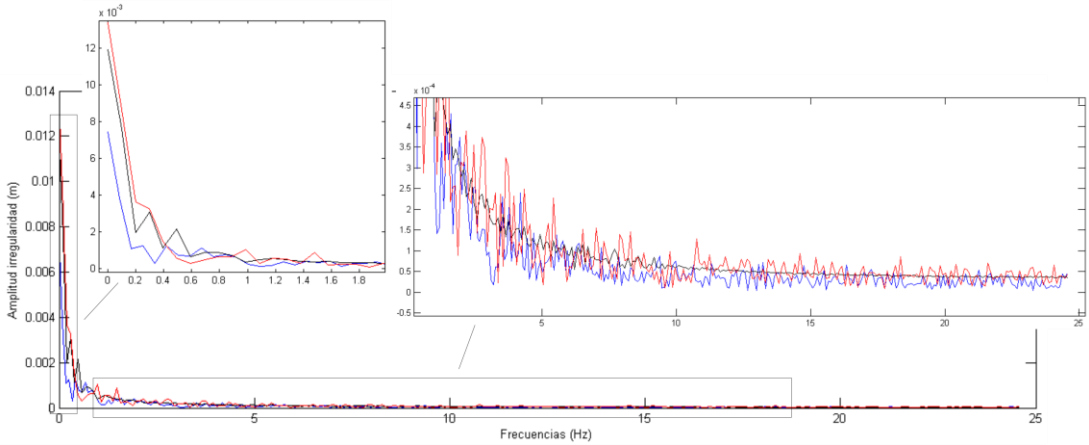


Figura 34: Contenido en frecuencia de los perfiles

Con el objetivo de mostrar una mejor representación de las irregularidades de los perfiles obtenidos en los ensayos se ha aplicado un filtro paso bajo a los perfiles de ambos ensayos para eliminar las altas frecuencias obteniendo un perfil más suave como el que se representa en la Figura 35.

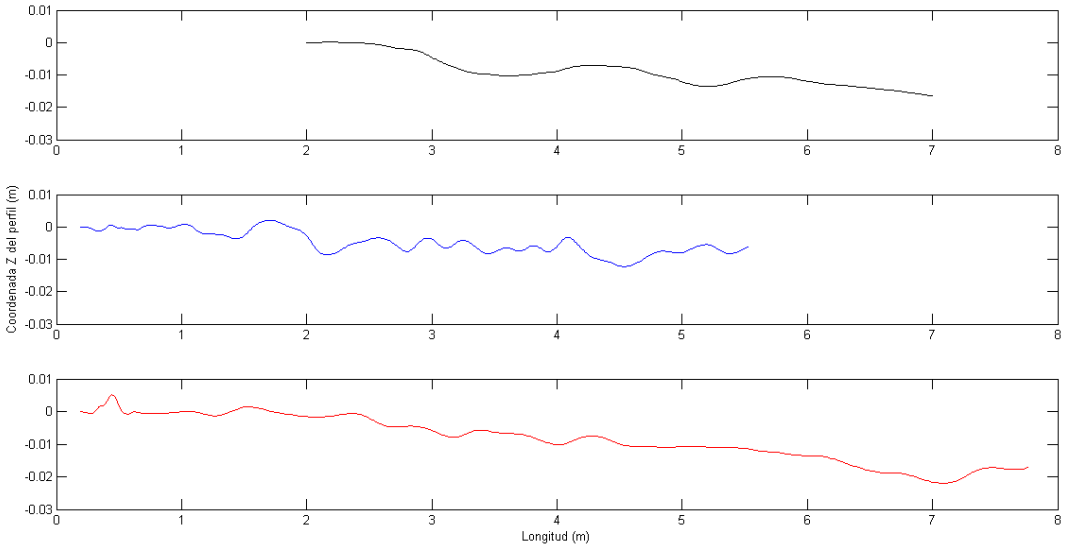


Figura 35: Estimación perfiles filtrados

5.3 Pérdida de datos

Con el objetivo de estudiar el efecto de la pérdida de datos se han realizado diversas simulaciones realizando la estimación de perfiles aleatorios a diferentes frecuencias de muestreo 100Hz y 1000 Hz. Con cada uno de estos perfiles aleatorios se realiza la estimación del perfil variando tanto la frecuencia a la que se pierde un solo dato como variando el número de datos que se pierden a una misma frecuencia de pérdida de datos.

Usando una frecuencia de muestreo de 100Hz, en la Figura 36 se muestra en negro la evolución del perfil teórico y en magenta y verde la evolución del perfil con una pérdida de un dato cada 1000 y 500 puntos, también se muestra su contenido en frecuencia que ha sido amplizado para mayor detalle. En la Figura 37 se muestra la misma información con una pérdida de dos y cuatro datos cara 1000 puntos. En las Figura 38 y Figura 39 se representan los mismo resultado usando en este caso una frecuencia de muestreo de 1000Hz.

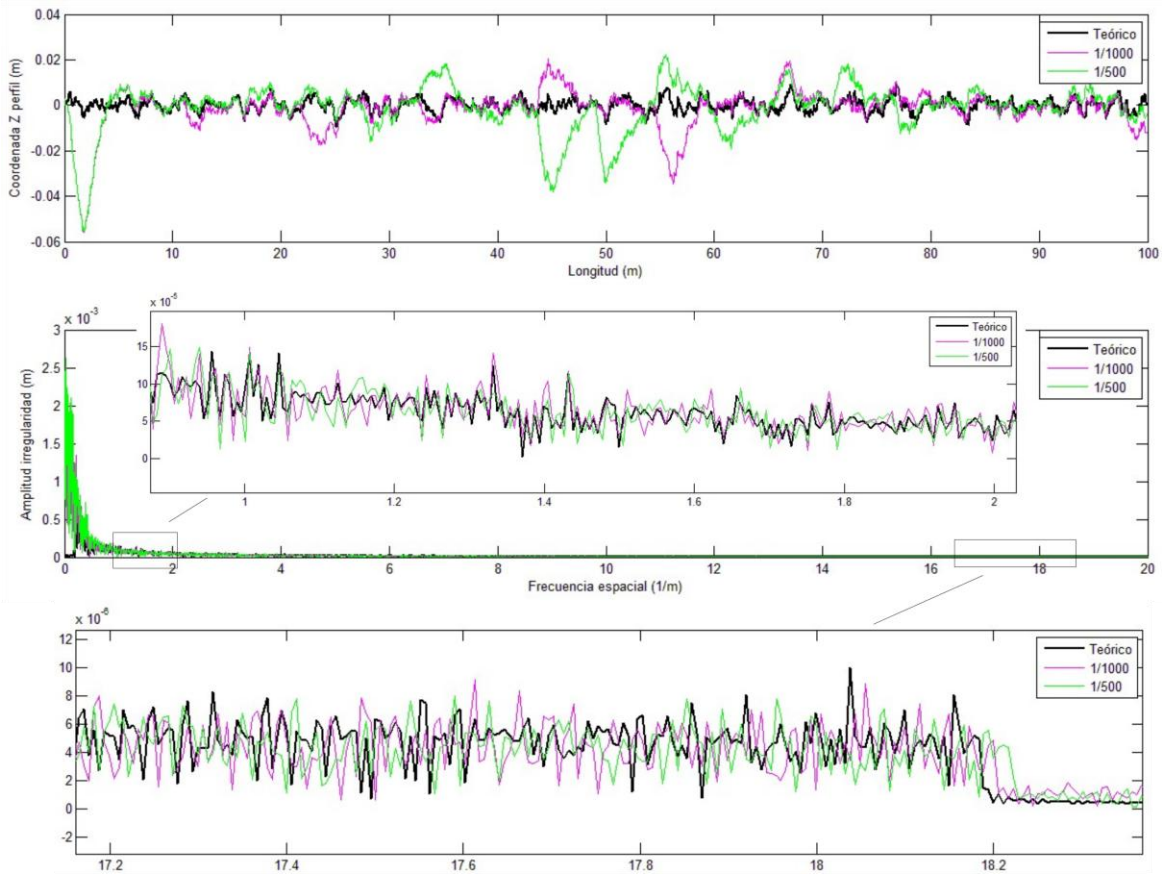


Figura 36: Ensayo a 100 Hz variando la frecuencia de pérdida de datos

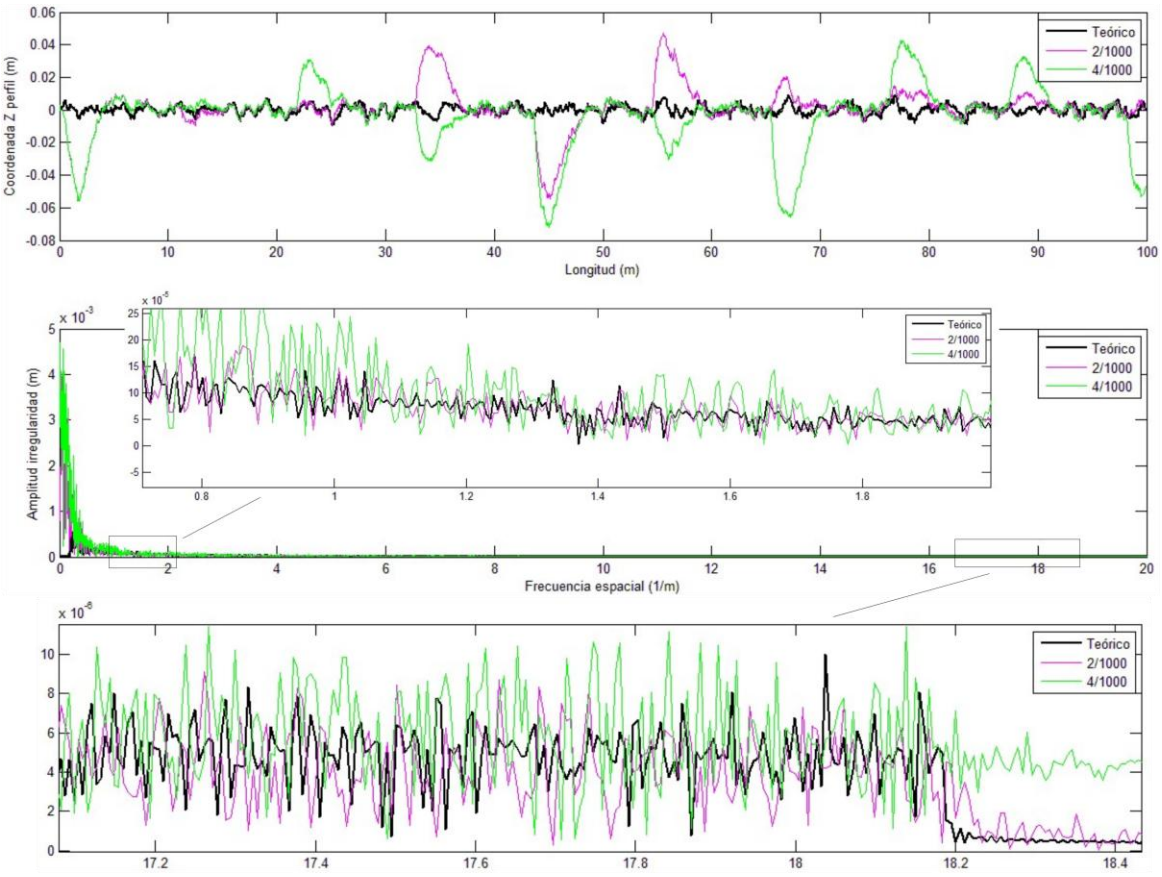


Figura 37: Ensayo a 100Hz variando el número de puntos de pérdida a una frecuencia

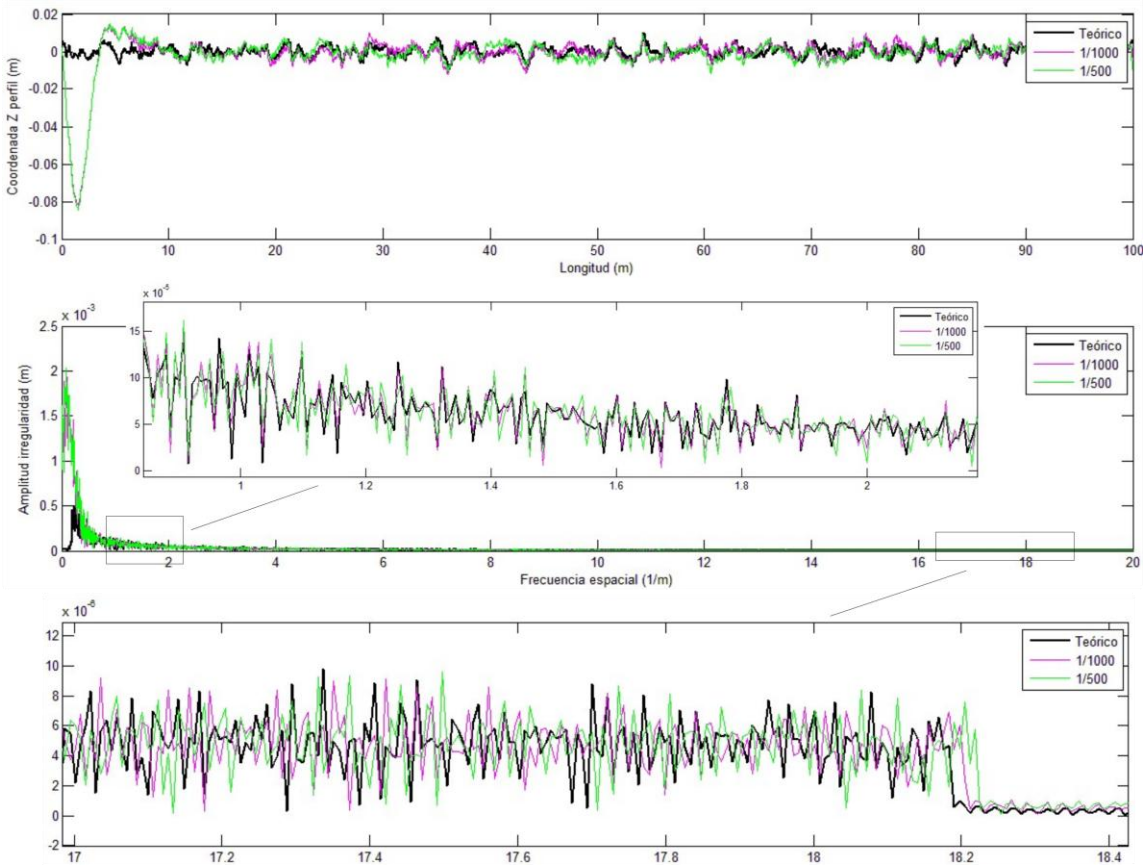


Figura 38: Ensayo a 1000 Hz variando la frecuencia de pérdida de datos

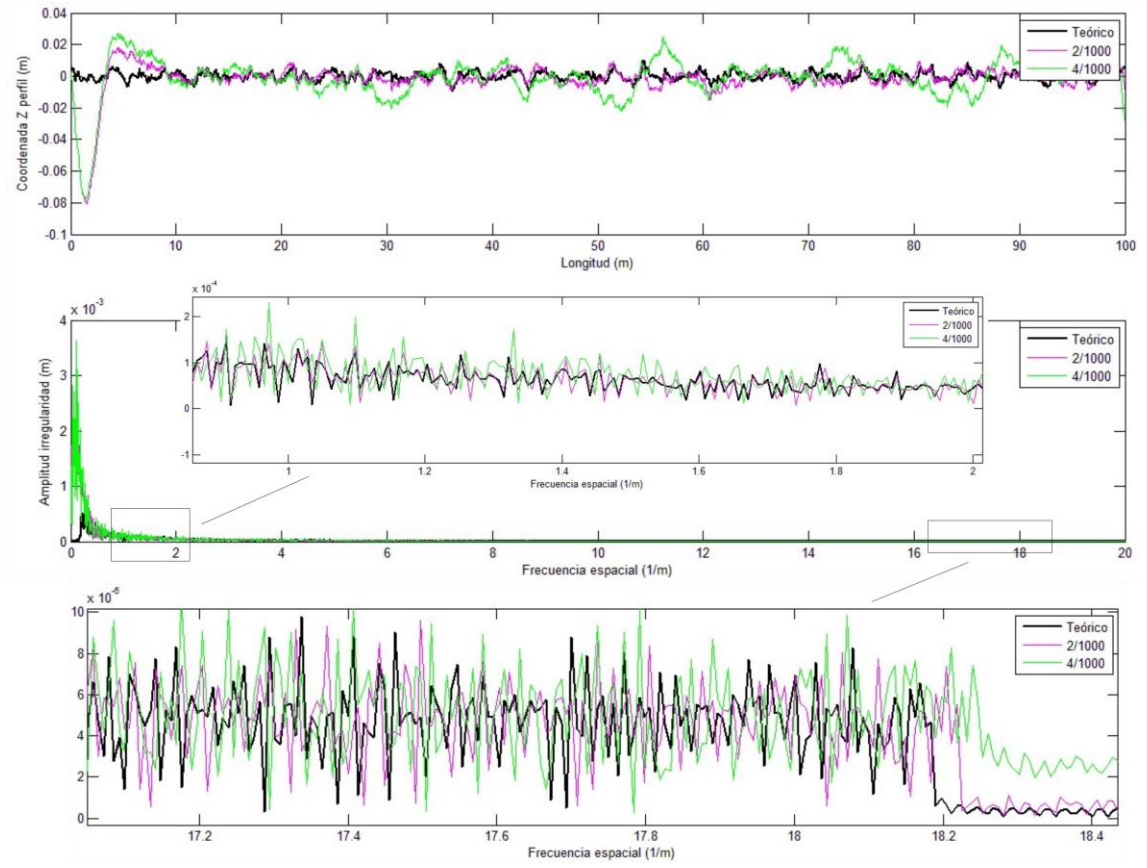


Figura 39: Ensayo a 1000Hz variando el número de puntos de pérdida a una frecuencia

6 CONCLUSIONES

Durante la realización de este Trabajo de Fin de Grado se ha conseguido poner en marcha la versión del Rover de la que partímos como vehículo auscultador de irregularidades verticales de perfiles mejorando principalmente la transmisión de datos entre el vehículo y el control remoto obteniendo una comunicación más fluida entre las antenas Xbee. Para ello se configuraron ambas antenas para asegurar que cada una tenía la dirección de la otra y se cambio el programa del arduino. Con la introducción de variables Interrput para medir las señales de los decodificadores para realizar el cálculo de velocidad real de avance del vehículo durante su marcha se obtuvo un cálculo más preciso de ésta. Con el fin de sustituir los bloques lógicos que controlaban el control de velocidad se realizó una programación usando ticker. Debido a una incompatividad con los Interrupt cuya solución requería trareas de priorización hacía que la programación fuese más costosa. De este modo se optó finalmente por una programación por hilos en paralelo, con los que se controla tanto el control de velocidad como la lectura de los datos de los sensores. Si bien hay que tener en cuenta que no se logró resolver a tiempo para los ensayos el problema del llenado del buffer. Esto nos limitó la frecuencia de adquisición de los datos en los ensayos, haciéndose uso únicamente de la memoria interna del microcontrolador. Por otro lado el hecho de realizar los ensayos sobre un terreno tan estrecho debido a la dura tarea de medición mediante el uso de la estación total ha dificultado manter el vehículo dentro de los límites del terreno medido.

Se ha utilizado el algoritmo de auscultación de vías de tren [19], con el que se realiza la estimación de irregularidades de perfiles aleatorios de carretera en mal estado. Estos perfiles tienen longitudes de onda que pueden ser detectadas con un vehículo de las dimensiones y características del Rover. Para ello se utilizó una función que genera las coordenadas verticales de perfiles aleatorios a partir de la cual se calculan las señales teóricas de la IMU empleadas para la estimación con el algoritmo.

También se ha de resaltar el hecho de haber podido realizar un ensayo de auscultación de las irregularidades verticales de un perfil real, utilizando el Rover como vehículo auscultador. Los resultados obtenidos obviamente no son satisfactorios, sin embargo esto nos permite ver los problemas reales que aparecen. Los datos de aceleración recogidos por el acelerómetro resultaron demasiado altos, esto nos llevó a pensar en un posible defecto del funcionamiento de este pero se descartó tras realizar una calibración del sensor. De momento no se ha detectado el origen de los valores mencionados, quizá la rugosidad de la superficie y la rigidez de la suspensión del vehículo tiene mayor influencia que la que se ha estimado inicialmente por lo que deben realizarse más ensayos y contrastar los resultados obtenidos con otras IMUs.

Debido a los valores de aceleración obtenidos y con el fin de mejorar la concordancia entre la estimación y el perfil real antes de la estimación mediante le filtro de Kalman del perfil real se realiza el filtrado de las señales recogidas durante el ensayo. Debido a la importancia de utilizar una frecuencia de muestreo regular para realizar el análisis en frecuencia mediante la transformada de Fourier y al poco tiempo disponible para la realización de los ensayos se realiza una interpolación para obtener los valores de las señales a intervalos de tiempo constantes. Este procedimiento puede que no arroje valores muy precisos y posiblemente sería necesario utilizar algoritmos para el análisis en frecuencia que contemplen intervalos de muestreo irregulares. Debido a esto se realiza un pequeño análisis de sensibilidad sobre el efecto de la perdida de datos en las señales adquiridas y su posterior tratamiento con la transformada de Fourier. A medida que aunmenta la perdida de datos los resultados obtenidos son menos fieles al perfil teórico, sin embargo este efecto es cada vez menos severo al aumentar la frecuencia de muestro.

7 REFERENCIAS

1. https://es.wikipedia.org/wiki/Auscultaci%C3%B3n_(ingenier%C3%ADa)#Ferrocarril.
2. B. Blanco "Desarrollo, instrumentación y construcción de un vehículo radio controlado para la medida de irregularidades del terreno mediante sensores inerciales", Universidad de Sevilla, 2014.
3. C. S. Bonaventura, A. M. Zarembski, J. W. Palese "Tracksafe: a track geometry car based real-time dyunamics simulator", Proceeding of JRC2005, Pueblo, Colorado, 2005.
4. E. G. Berggren, M. X. D. Li y J. Spännar "A new approach to the analysis ans presentation of vertical track geometry quality and rail roughness", Wear 265 1488-1496, 2008.
5. G. Charles, R. Goodall, R. Dixon "Model-based condition monitoring at the wheel-rail interface", Vehicle System Dynamics, 2008.
6. P. F. Weston, C. S. Ling, C. Roberts, C. J. Goodman, P. Li y R. M. Goodall "Monitoring vertical track irregularity from in-service railway vehicles" Universidad de Birmingham y Universidad de Loughborough, 2006.
7. P. Weston, C. Roberts, G. Yeo y E. Stewart "Perspectives on railway track geometry condition monitoring from in-service railway vehicles" Universidad de Birmingham, 2015.
8. https://www.arduino.cc/en/Main/ArduinoBoardFio.
9. http://www.digi.com/lp/xbee/.
10. https://developer.mbed.org/platforms/mbed-LPC1768/.
11. https://www.pololu.com/product/1443.
12. Bosch "BMA180 Digital, triaxial acceleration sensor", Data sheet, 2010.
13. Ivensense "ITG-3200 Product Specification", 2010.
14. R.Toulson y T. Wilmshurst "Fast and Effective Embedded Systems Design", 2012.
15. https://developer.mbed.org/users/aberk/code/PIDRover/.
16. https://developer.mbed.org/cookbook/Homepage.
17. https://developer.mbed.org/platforms/mbed-LPC1768/.
18. http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu.
19. A. F. Hidalgo, J. Ros y J. L. Escalona "A Kalman filter-based algorithm for IMU signal fusion applied to track geometry estimation" ECCOMAS Thematic Conference on Multibody Dynamics Barcelona, 2015.
20. F. Tyan, Y. F. Hong, S. H. Tu y W. S. Jeng "Generation of random road profiles" Universidad de Tamkang.
21. User manual 2.0 Leica Geosystems.
22. http://surgeweb.sweb.cz/.