

GRADO EN INGENIERÍA ELECTRÓNICA DE
COMUNICACIONES



Trabajo Fin de Grado

Sistemas inerciales para la ayuda a la evaluación del Parkinson

ESCUELA POLITECNICA

Autor: Cristina Fernández Baldovinos

Tutor: Ana Jiménez Martín

Cotutor: David Gualda Gómez

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

GRADO EN INGENIERÍA ELECTRÓNICA DE COMUNICACIONES

Trabajo Fin de Grado

Sistemas inerciales para la ayuda a la evaluación del
Parkinson

Autor: Cristina Fernández Baldovinos

Tutor: Ana Jiménez Martín

Cotutor: David Gualda Gómez

TRIBUNAL:

Presidente: Carlos Julián Martín Arguedas

Vocal 1º: Daniel Pizarro Pérez

Vocal 2º: Ana Jiménez Martín

FECHA:

CALIFICACIÓN:

A mis padres y hermanos...

Agradecimientos

Para empezar, me gustaría agradecer a mi tutora Ana y a mi cotutor David por ayudarme en gran manera, guiarme y apoyarme durante todo este tiempo. Gracias por enseñarme tanto y por hacer mucho más de lo que estáis obligados. Habéis sido unos tutores excelentes y estoy muy agradecida de haber compartido este camino con vosotros. También dar las gracias al grupo de investigación GEINTRA US&RF por toda la ayuda prestada.

Dar las gracias a toda mi familia, en especial a mi padre, porque nunca has dejado de confiar en mí y en mis capacidades, y siempre has estado dispuesto a ayudarme y animarme siempre que lo he necesitado. También gracias a mi madre y a mis dos hermanos, porque sin su apoyo no hubiera conseguido llegar hasta aquí.

Quiero agradecer en especial a Alberto, por acompañarme en todos los malos momentos e intentar hacerme sentir mejor, apoyarme y confiar en mí. Has sido un pilar muy grande estos últimos meses y te estoy muy agradecida.

Por último, dar las gracias a mis amigos y en general a todos los miembros de AOE, en especial a Siyu, Miriam, Ana, Ana T., Irene, Elena, Javier y Ernesto porque gracias a vosotros me habéis conseguido sacar una sonrisa cuando más lo he necesitado. Me habéis llenado de positivismo y me llenáis de energía en cada ensayo. Gracias por estar siempre ahí.

I - Resumen

Este Trabajo Fin de Grado presenta el desarrollo de un algoritmo que permite la detección de anomalías motoras en un enfermo de Parkinson utilizando una unidad de medición inercial (IMU) situada en el tobillo del paciente para medir la marcha. Las señales obtenidas de la IMU son procesadas mediante un Filtro de Kalman Extendido (EKF) con la intención de obtener la estimación de los ángulos de orientación del sensor (Roll, Pitch y Yaw). A partir de esta información se ha desarrollado el algoritmo que permite la detección de pasos normales y anómalos de una persona. También se ha diseñado una interfaz de usuario que permite la visualización de los resultados de las pruebas de forma cómoda y sencilla, de forma que ayude a los profesionales médicos a realizar el diagnóstico del paciente de una manera más objetiva. Los resultados obtenidos demuestran la validez de esta primera aproximación de procedimiento para obtener medidas objetivas del estado del paciente.

Palabras clave: Unidad de medida inercial (IMU), Parkinson, deterioro de la marcha, filtro de Kalman Extendido (EKF), ángulos de Euler.

II - Abstract

This Final Degree Project presents the development of an algorithm that allows the detection of motor impairment in Parkinson disease using an inertial measurement unit (IMU) placed on the patient's ankle to measure gait. The signals from the IMU are processed through an Extended Kalman Filter in order to obtain the estimation of the sensor's orientation angles (Roll, Pitch, Yaw). Based on this information, an algorithm to detect normal and abnormal gait has been developed. In addition, an user interface has been designed to visualize the test results in an easy way, in order to help physicians carry out the patient diagnosis as objective as possible. The obtained results prove the validity of this first approximation to get objective measures of the patient state.

Keywords: Inertial measurement unit (IMU), Parkinson, gait impairment, extended Kalman filter (EKF), Euler angles.

Índice

Agradecimientos	VI
I - Resumen	VIII
II - Abstract	X
Índice de figuras	XIV
Índice de tablas	XVI
III-MEMORIA	1
Capítulo 1: Introducción	3
1.1 Objetivos del proyecto.....	4
1.2 Estructura del documento	5
Capítulo 2: Conceptos teóricos.....	7
2.2 Unidad de medida inercial.....	9
2.3 Filtro de Kalman	11
2.4 Procesamiento de las señales	10
2.4.1 Algoritmo de detección de pasos	10
2.4.2 Análisis espectral	10
Capítulo 3: Algoritmo propuesto	19
3.1 Detección de pasos anómalos.....	10
3.1.1 Detección de pasos anómalos en el dominio del tiempo	10
3.1.2 Detección de pasos anómalos en el dominio de la frecuencia	22
3.1.3 Decisor	24
3.2 Obtención de características derivadas del paso	25
3.3 Diseño de la interfaz de usuario	26
3.3.1 Ventana principal.....	27
3.3.2 Ventana de estadísticas	28
3.3.3 Ventana de comparación	29
Capítulo 4: Resultados	31
4.1 Pruebas de optimización del algoritmo	31
4.2 Validación del algoritmo con pacientes reales	39
4.2.1 Rutina de los pacientes.....	39
4.2.2 Resultados del paciente P01-V1	40
4.2.3 Resultados del paciente P02-V1	42
4.2.4 Comparativa entre pacientes.....	44
Capítulo 5: Conclusiones y trabajos futuros.....	47
5.1 Conclusiones	47

5.2 Trabajos futuros	48
IV– Pliego de condiciones	51
V– Presupuesto	53
VI – Manual de usuario	55
VII – Planos	59

Índice de figuras

Figura 1: NGIMU de IO Technologies utilizada en el proyecto	9
Figura 2: Ángulos de Euler Z (Yaw); Y (Pitch); X (Roll).....	10
Figura 3: Diagrama del filtro de Kalman	11
Figura 4: Diagrama de bloques del filtro de Kalman extendido.....	15
Figura 5: Esquema de las piernas de una persona al caminar.....	16
Figura 6: Señal de estimación del ángulo Pitch obtenida de una persona al caminar	17
Figura 7: Diagrama de bloques del algoritmo propuesto	19
Figura 8: Diagrama de bloques del algoritmo de cruces por cero	20
Figura 9: Señal Pitch de una persona caminando de forma normal	21
Figura 10: Señal Pitch de una persona arrastrando los pies.....	21
Figura 11: Detección de pasos normales (rojo) y pasos arrastrando los pies (negro).....	22
Figura 12: Espectrograma de la señal Pitch	23
Figura 13: Diagrama de bloques de la clasificación de las ventanas de la señal Pitch.....	24
Figura 14: Detección de zonas de paso normal (azul), paso anómalo (magenta) y ausencia de paso (amarillo).....	24
Figura 15: Diagrama de bloques del algoritmo del decisor	25
Figura 16: Diseño de la ventana principal de la interfaz de usuario	27
Figura 17: Diseño de la página de estadísticas de la interfaz de usuario	28
Figura 18: Diseño de la página de comparación de pruebas de la interfaz de usuario.....	29
Figura 19: Parte de la prueba 1 de optimización del algoritmo.....	33
Figura 20: Parte de la prueba 2 de optimización del algoritmo.....	33
Figura 21: Parte de la prueba 3 de optimización del algoritmo.....	34
Figura 22: Parte de la prueba 4 de optimización del algoritmo.....	35
Figura 23: Parte de la prueba 5 de optimización del algoritmo.....	35
Figura 24: Parte de la prueba 6 de optimización del algoritmo.....	36
Figura 25: Parte de la prueba 7 de optimización del algoritmo.....	36
Figura 26: Parte de la prueba 8 de optimización del algoritmo.....	37
Figura 27: Parte de la prueba 9 de optimización del algoritmo.....	37
Figura 28: Parte de la prueba 10 de optimización del algoritmo.....	38
Figura 29: Detección de paso normal, paso anómalo y zona sin paso en el paciente P01-V1	40
Figura 30: Detección de picos de ruido como paso anómalo en el paciente P01-V1	41
Figura 31: Detección de zonas de paso anómalo en el paciente P01-V1.....	41
Figura 32: Detección de paso normal, paso anómalo y zona sin paso en el paciente P02-V1	43

Figura 33: Detección de zona de paso normal en el paciente P02-V1.....	43
Figura 34: Ventana principal de la aplicación.....	55
Figura 35: Menú desplegable de la página principal de la aplicación.....	55
Figura 36: Página principal de la aplicación cuando se ha cargado una prueba	56
Figura 37: Página principal de la aplicación mostrando los resultados de la detección.....	57
Figura 38: Página de visualización de estadísticas de la prueba.....	57
Figura 39: Página de comparación de pruebas	58

Índice de tablas

Tabla 1: Ground truth y resultados de la prueba 1	32
Tabla 2: Ground truth y resultados de la prueba 2	33
Tabla 3: Ground truth y resultados de la prueba 3	34
Tabla 4: Ground truth y resultados de la prueba 4	34
Tabla 5: Ground truth y resultados de la prueba 5	35
Tabla 6: Ground truth y resultados de la prueba 6	35
Tabla 7: Ground truth y resultados de la prueba 7	36
Tabla 8: Ground truth y resultados de la prueba 8	37
Tabla 9: Ground truth y resultados de la prueba 9	37
Tabla 10: Ground truth y resultados de la prueba 10	38
Tabla 11: Ground truth y resultados de todas las pruebas	38
Tabla 12: Estadísticas de las pruebas del paciente P01-V1	42
Tabla 13: Estadísticas de las pruebas del paciente P02-V1	44
Tabla 14: Estadísticas medias de las pruebas de ambos pacientes	44
Tabla 15: Coste del material utilizado	53
Tabla 16: Costes directos del proyecto	53
Tabla 17: Costes indirectos del proyecto	54
Tabla 18: Costes totales del proyecto	54

III-MEMORIA

Capítulo 1: Introducción

La enfermedad de Parkinson es actualmente una enfermedad neurodegenerativa cada vez más común en las personas de avanzada edad. Esta enfermedad se debe a la falta de dopamina en determinadas partes del cerebro, un elemento esencial en la transmisión neuronal. La carencia de este elemento produce un deterioro importante en las funciones motoras de las personas que lo padecen. Los síntomas que caracterizan esta enfermedad se pueden clasificar en motores y no motores, pero este trabajo se centrará exclusivamente en los motores. Uno de los principales síntomas es la ralentización del movimiento, esto es, la dificultad para caminar, el congelamiento del paso, la incapacidad de iniciar la marcha, y la falta de seguridad y firmeza al dar los pasos. También presentan lentitud en los movimientos, temblores, rigidez, movimientos involuntarios, caídas de forma frecuente y muchos otros síntomas. La magnitud de estos síntomas no es igual para todos los enfermos, ya que depende del grado de la enfermedad, su tipo y la medicación que estos reciben.

El seguimiento de la evolución de la enfermedad suele ser complicado puesto que los profesionales sanitarios realizan la evaluación de los pacientes, en gran medida, con la información que estos le transmiten a través de formularios. Dicha información no es totalmente objetiva debido a que el paciente puede olvidarse de ciertas cosas que le han ocurrido o dejarse guiar por sensaciones que son erróneas. Los médicos en muchas ocasiones tienen que confiar en lo que cuentan sus pacientes, sin ser esta información totalmente exacta. Además, la consulta médica es una situación no ordinaria para el enfermo y suele ser de corta duración, por lo que es posible que los síntomas no se manifiesten o no lo hagan de la misma forma que en situaciones normales. Dada esta situación, es de vital importancia lograr un dispositivo fiable que pueda medir de forma

precisa todos los parámetros necesarios para realizar una evaluación objetiva de los enfermos en su entorno diario. De esta forma, los profesionales de los centros de salud podrían desarrollar su trabajo con una mayor información, lo que repercutiría en una mejora en el tratamiento de estas personas.

Gracias a los rápidos avances tecnológicos de los últimos años, se puede encontrar en la literatura distintas alternativas para, a través de sistema inerciales, intentar evaluar los parámetros característicos de la enfermedad de Parkinson. Los sistemas inerciales se han introducido en diversas aplicaciones de la vida diaria, como por ejemplo en los teléfonos móviles o pulseras de actividad. Al igual que se pueden utilizar estos sistemas para estimar el número de pasos que da una persona a lo largo del día, también se está evaluando su utilización para estimar otras características del paso. En el caso de enfermos de Parkinson esto puede ser muy útil para detectar las diferentes anomalías que presentan al caminar y que ayude, como se ha mencionado anteriormente, a la evaluación de su enfermedad. Estos dispositivos, además, son pequeños y ligeros, lo que los hace sencillos de llevar sin ser una molestia para el usuario.

1.1 Objetivos del proyecto

El presente proyecto se centra en el diseño y evaluación de un algoritmo que permita estimar la magnitud de diferentes parámetros motores de una persona con la enfermedad de Parkinson. Para conseguirlo, se han fijado los siguientes objetivos parciales:

- Detección de pasos cortos y arrastrando los pies.
- Detección de los momentos en los que no hay paso.
- Obtención de diferentes parámetros del paso como la cadencia, velocidad y amplitud.
- Análisis de las diferencias entre un paso normal y el paso de un enfermo.

Además, se realizarán diferentes pruebas con pacientes reales que ayudarán a comprobar el correcto funcionamiento del algoritmo y si este pudiera ser utilizado como una herramienta de evaluación de enfermos de Parkinson en el sistema actual de salud.

1.2 Estructura del documento

La memoria de este Trabajo Fin de Grado se compone de 5 capítulos principales. En este primer capítulo se ha contextualizado el trabajo dando una visión general del proyecto. En el capítulo 2 se expone una introducción teórica a los diferentes conceptos utilizados para la realización del proyecto y que son esenciales para el desarrollo del algoritmo. En el capítulo 3 se explica en detalle todo lo referente al algoritmo diseñado. En el capítulo 4 se analizan los resultados obtenidos de las diferentes pruebas realizadas. Por último, en el capítulo 5, se muestran las conclusiones extraídas del proyecto y se comentan las diferentes líneas de trabajo posteriores.

Capítulo 2: Conceptos teóricos

Para comenzar a desarrollar el trabajo, en primer lugar, se ha realizado una búsqueda de información acerca de la temática del proyecto, a través de la lectura de diversos artículos científicos. Gracias a ellos se ha obtenido el conocimiento necesario para el desarrollo del proyecto. En este capítulo se expone toda esta información teórica que es la base fundamental del trabajo. En él se incluye información relativa a las posibles limitaciones motoras a detectar, el sensor utilizado y los algoritmos necesarios para el procesamiento de las señales del sensor. Los parámetros motores son medidos por un sensor inercial y se utilizarán dos algoritmos diferentes, uno para realizar la estimación de los ángulos de Euler a partir de las señales del sensor, y otro para la detección de pasos en dichas señales.

2.1 Estado del arte

Uno de los mecanismos más utilizados por los facultativos para conocer el grado de enfermedad del paciente es la escala UPDRS (Escala unificada para la evaluación de la Enfermedad de Parkinson) [1]. Esta escala se compone de diferentes apartados que se evalúan con una puntuación del 0 al 4, siendo 0 un nivel normal y 4 un nivel de alteración severa. Dicha puntuación es el resultado de unas pruebas determinadas que realiza el paciente y es el médico quien decide de forma subjetiva su valor dependiendo de lo que este observe. Por esta razón es de gran importancia desarrollar un sistema que permita evaluar a los pacientes de forma objetiva, y en consecuencia poder ajustar la medicación de manera más precisa.

En la evaluación de la enfermedad del Parkinson se puede distinguir entre síntomas motores y no motores. Este trabajo se centra en la detección de los parámetros motores siendo algunos de estos los siguientes:

- Congelamiento del paso (FOG): Es la imposibilidad de caminar, cuando el paciente se queda bloqueado sin poder dar un paso.
- Estados de On y Off: Estos estados son los cambios motores que se producen en el paciente a lo largo del día. Se llama estados de On al estado en el que los síntomas se encuentran bajo control, y estado de Off al estado en el que los síntomas aparecen o empeoran debido a ciclo activo de la medicación.

En la literatura se encuentran varios trabajos que se centran en conseguir esta evaluación objetiva. Algunos de estos trabajos buscan la relación entre esta escala UPDRS y las mediciones realizadas con sensores inerciales (IMU), bien mediante pruebas caminando para detectar parámetros del paso [2] o en reposo para medir los posibles temblores posturales [3]. En otros trabajos, como [4], se centran en diferenciar si el paciente se encuentra en una etapa temprana de la enfermedad o si se encuentra en un estado más avanzado, realizando varios tipos de pruebas controladas en un laboratorio, mediante una IMU colocada en el lateral del talón. Esto podría servir para realizar una detección precoz de la enfermedad.

El paso de los enfermos de Parkinson es más lento que el de un adulto sano como se demostró en [5]. Los autores utilizaron un acelerómetro triaxial colocado en la cintura y al analizar cada componente de la aceleración se pudo observar que este tipo de personas tienden a balancearse de forma lateral cuando caminan. Otros artículos, como [6], no solo examinan el paso en un trayecto recto sino también cuando se realizan giros. En este caso la IMU se posicionó en la parte baja de la espalda y la prueba consistió en realizar tareas domésticas. Otra característica importante para detectar es el FOG. En [7] se propone la detección de estos episodios por medio de un algoritmo de análisis frecuencial.

La postura encorvada que presentan la mayoría de los enfermos lleva asociado fuertes dolores y, además, incrementa el riesgo de sufrir una caída. En [8] se estudia este encorvamiento que no solo se produce en estático si no también caminando. Aparte de utilizarse como diagnóstico, esta información también puede servir para recordarle al paciente que debe corregir su mala postura. Sería de gran ayuda poder también predecir los movimientos para evitar esas malas posturas, por ello en [9] se busca la predictibilidad

de los movimientos del tronco y su evolución a lo largo del tiempo. Se comprobó que dicha predictibilidad era menor en los enfermos de Parkinson que en las personas sanas, y también se comprobó que había una ligera correlación entre el control postural y el nivel de deterioro cognitivo y motor.

Por otro lado, es fundamental poder monitorizar al paciente cuando este se encuentre en su propio domicilio, debido a que su comportamiento o síntomas pueden ser diferentes en comparación a realizar las pruebas en la consulta o entorno controlado. Por ello varios estudios, como [10], llevan a cabo los experimentos en casa del paciente, en los que este debe realizar las tareas normales de su día a día. En este caso la finalidad del estudio era la detección de los estados de Off y On, y su frecuencia.

2.2 Unidad de medida inercial

Una unidad de medida inercial (IMU) es un dispositivo electrónico compuesto, normalmente, por tres acelerómetros y tres giróscopos, que sirve para medir aceleración lineal y velocidad angular en cada uno de los ejes cartesianos. También pueden disponer de magnetómetros u otros sensores adicionales. Actualmente sus aplicaciones son muy diversas pero, en concreto, para este trabajo se ha utilizado para medir el movimiento de una persona al caminar. El dispositivo comercial NGIMU de IO Technologies, mostrado en la figura 1, ha sido el empleado a lo largo de este trabajo [11].



Figura 1. NGIMU de IO Technologies utilizada en el proyecto.

Para medir dicho movimiento se utilizarán los ángulos de Euler. Estos ángulos sirven para especificar la orientación de un cuerpo con respecto a un sistema de referencia fijo. Son 3 ángulos, llamados roll (Φ), pitch (θ) y yaw (Ψ), y representan en una secuencia determinada el ángulo de giro sobre los ejes X, Y y Z respectivamente. En este caso la convención utilizada es la de los ángulos de Euler Z-Y-X. Como se muestra en la figura 2, primero se obtiene el giro sobre el eje Z (Yaw), luego sobre el eje Y (Pitch) y, por último, sobre el eje X (Roll).

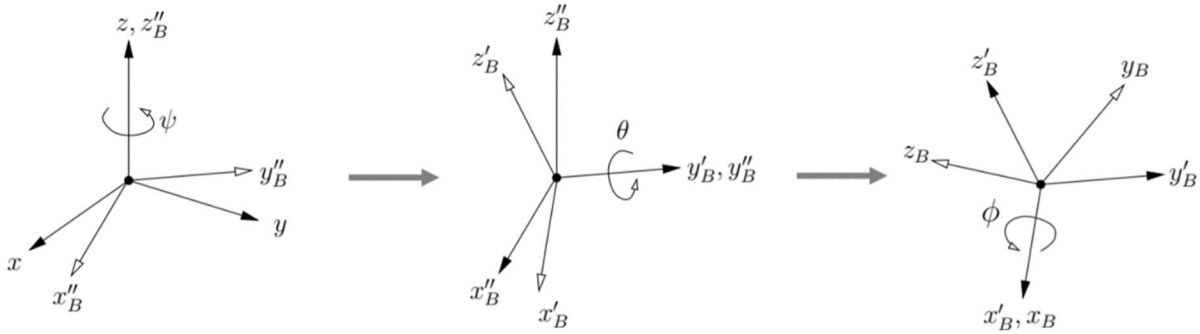


Figura 2. Ángulos de Euler Z (Yaw); Y (Pitch); X (Roll).

Los valores que proporcionan los giróscopos son respecto a un sistema de referencia que se mueve junto con la IMU, por lo tanto, al girar sobre un determinado eje del sistema de referencia, la posición de este varía con respecto al sistema de referencia fijo. Por ello, para expresar la posición final del cuerpo mediante los ángulos de Euler se tiene que tener en consideración la rotación en torno al resto de ejes, es decir, para obtener la posición final del eje Y hay que tener en cuenta la rotación hecha sobre el eje X, y para obtener la posición final del eje Z hay que tener en cuenta la rotación hecha sobre el eje X y también sobre el eje Y. Las ecuaciones (2.1) y (2.2) son las que relacionan los valores de los giróscopos con los ángulos de Euler, obteniéndose la matriz de rotación M (ecuación (2.3)) que se utilizará en el algoritmo de estimación de ángulos de Euler, siendo $R(\theta)$, $R(\Phi)$ y $R(\Phi)$ los ángulos de rotación sobre los ejes X-Y-Z.

$$\omega = R(\theta) \cdot R(\Phi) \cdot \begin{bmatrix} 0 \\ 0 \\ \dot{\Psi} \end{bmatrix} + R(\Phi) \cdot \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\Phi} \\ 0 \\ 0 \end{bmatrix} \quad (2.1)$$

$$\omega = M^{-1} \cdot \begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\Psi} \end{bmatrix} \quad (2.2)$$

$$M = \begin{bmatrix} 1 & \sin \Phi \tan \theta & \cos \Phi \tan \theta \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \frac{\sin}{\cos \theta} & \frac{\cos \Phi}{\cos \theta} \end{bmatrix} \quad (2.3)$$

Los sistemas inerciales son aparatos pequeños, económicos y fáciles de llevar, lo que lo hace perfecto para esta aplicación en la que una persona tiene que llevarlo puesto todo el día. En un futuro los podremos encontrar integrados en prendas o calzado gracias a los rápidos avances tecnológicos. Sin embargo, presentan una gran limitación debido al error acumulativo que presentan. Para ello se ha realizado un filtro de Kalman que permite estimar los ángulos de Euler evitando así un error grande en el sistema.

2.3 Filtro de Kalman

El filtro de Kalman es un algoritmo que permite estimar en tiempo real y de manera óptima, el valor de una o varias variables de un sistema lineal contaminado por ruido gaussiano, a partir de otras variables conocidas u observables. Es un algoritmo recursivo y gracias a él se pueden predecir estados pasados, presentes y futuros. Sin embargo, el sistema que se utiliza en este trabajo no es lineal, por lo es más preciso utilizar la variante del filtro de Kalman definido como filtro de Kalman Extendido (EKF).

El EKF es una variación del filtro de Kalman que permite resolver el problema de estimación de variables en sistemas no lineales. El fundamento es exactamente el mismo que el del filtro de Kalman original, pero se utiliza la aproximación de Taylor para la linealización del sistema. Esta linealización se realiza en cada estado, lo que hace que la aproximación del modelo del sistema sea cambiante. También se supone que la diferencia entre la predicción y la observación son pequeñas, ya que el algoritmo no funcionaría de forma correcta si esto no fuera así [12].

El proceso recursivo del filtro de Kalman consta de dos fases tal y como se representa en la figura 3.

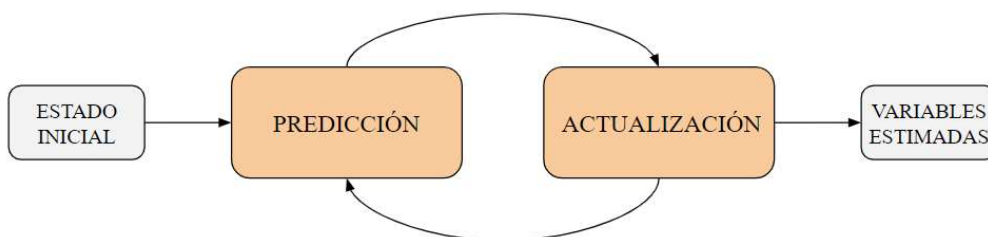


Figura 3. Diagrama del ciclo del filtro de Kalman.

La primera fase es la de predicción, en la que a partir del estado anterior se deduce el estado actual, conocido como estimación a priori. En la segunda fase, llamada fase de actualización, se corrige la predicción a partir de las observaciones del estado actual, dando lugar a la estimación a posteriori. Una vez acabada la fase de actualización, se realiza la predicción del siguiente estado a partir de la estimación a posteriori obtenida en la fase de actualización (estado anterior), de forma que todo el proceso se repite de forma sucesiva.

El modelo de un sistema no lineal se puede definir mediante las siguientes ecuaciones:

$$X_k = f(X_{k-1}, u_k) + W_k \quad (2.4)$$

$$Z_k = h(X_k) + V_k \quad (2.5)$$

Siendo:

- X_k : vector de estado, función dependiente del estado anterior y de la entrada al sistema, más ruido.
- Z_k : vector de observaciones, dependiente del estado actual del sistema más ruido.
- W_k : ruido del proceso, de distribución gaussiana, media cero y covarianza Q_k .
- V_k : ruido de la medida, de distribución gaussiana, media cero y covarianza R_k .

La primera fase del algoritmo, la fase de predicción, consiste en deducir el estado actual del sistema. Las ecuaciones que modelan esta fase son las siguientes:

$$\hat{X}_k^- = f(\hat{X}_{k-1}, u_k) \quad (2.6)$$

$$A_k = \frac{\partial f(\hat{X}_{k-1}, u_k)}{\partial \hat{X}_k} \quad (2.7)$$

$$P_k^- = A_k \cdot P_{k-1} \cdot A_k^T + Q \quad (2.8)$$

Siendo:

- \hat{X}_k^- : estimación del vector de estado a priori. Es función de la estimación del estado anterior.
- A_k : jacobiano de la ecuación de transición de estados que se utiliza para linealizar el sistema.

- P_k^- : covarianza del error a priori. Se calcula a partir de la covarianza del error del estado anterior, la matriz A y el ruido del proceso Q .

La fase de actualización tiene como objetivo la corrección de la predicción a partir de las mediciones del estado actual. Las ecuaciones necesarias para ello se muestran a continuación:

$$K = P_k^- \cdot H_k^T \cdot [H_k \cdot P_k^- \cdot H_k^T + R]^{-1} \quad (2.9)$$

$$\hat{X}_k = \hat{X}_k^- + K \cdot (Z_k - h(\hat{X}_k)) \quad (2.10)$$

$$P_k = (I - K \cdot H_k) \cdot P_k^- \quad (2.11)$$

$$H_k = \frac{\partial h(\hat{X}_k^-)}{\partial \hat{X}_k^-} \quad (2.12)$$

Siendo:

- K : ganancia de Kalman. Depende de la covarianza del error a priori, la matriz H y el ruido de medida R . Esta constante sirve para ajustar en qué medida afectan las observaciones a la estimación del estado.
- \hat{X}_k : estimación del estado a posteriori. Depende de la estimación del estado a priori, la ganancia de Kalman y de las observaciones.
- P_k : covarianza del error a posteriori. Es función de la covarianza a priori, la ganancia de Kalman y la matriz H.
- H_k : jacobiano de la ecuación de medida que se utiliza para linealizar el sistema.

Estas son las ecuaciones generales que conforman el filtro de Kalman extendido, sin embargo, se tienen que particularizar para el sistema utilizado. En este caso lo que se quiere obtener es la estimación de los ángulos de Euler.

Al calcular la orientación de los sensores se utiliza un método probabilístico, es decir, la orientación depende de los valores anteriores y de su varianza de error. Por lo tanto, en la primera iteración es necesario definir unas condiciones iniciales para los ángulos de Euler, ya que no se dispone de información previa. Las ecuaciones (2.13), (2.14) y (2.15) muestran la estimación de los ángulos Roll (Φ), Pitch (θ) y Yaw (Ψ) en la primera iteración.

$$\hat{\Phi}_0 = \tan\left(\frac{\alpha_{0,y}}{\alpha_{0,z}}\right)^{-1} \quad (2.13)$$

$$\hat{\theta}_0 = \tan\left(\frac{-\alpha_{0,x}}{\sqrt{\alpha_{0,y}^2 + \alpha_{0,z}^2}}\right)^{-1} \quad (2.14)$$

$$\hat{\Psi}_0 = 0 \quad (2.13)$$

Siendo:

- $\alpha_{0,x}$: aceleración inicial sobre el eje X.
- $\alpha_{0,y}$: aceleración inicial sobre el eje Y.
- $\alpha_{0,z}$: aceleración inicial sobre el eje Z.

La estimación del estado para los ángulos de Euler (ecuación (2.6)) es obtenida mediante la información de los giróscopos como se indica a continuación, siendo igual al valor de los ángulos en el estado anterior más el incremento de rotación (ecuación (2.3)):

$$\hat{X}_k^- = \begin{bmatrix} \hat{\Phi}_{k-1} + \Delta T(\omega_{k,x} + \omega_{k,y} \sin \Phi \tan \theta + \omega_{k,z} \cos \Phi \tan \theta) \\ \hat{\theta}_{k-1} + \Delta T(\omega_{k,y} \cos \Phi - \omega_{k,z} \sin \Phi) \\ \hat{\Psi}_{k-1} + \Delta T(\omega_{k,y} \frac{\sin \Phi}{\cos \theta} + \omega_{k,z} \frac{\cos \Phi}{\cos \theta}) \end{bmatrix} \quad (2.16)$$

La etapa de corrección solo se realiza cuando el pie se encuentra en contacto con el suelo. En ese momento, la norma de la aceleración es igual a $9,8 \text{ m/s}^2$ o $1g$ (Ecuación (2.17)), es decir, la aceleración de la gravedad. Se puede considerar que el sensor no está en movimiento en ese instante y, por lo tanto, se puede realizar una estimación de los ángulos Roll y Pitch [4]. El vector de estimación de las observaciones es el mostrado en la ecuación (2.18). Las iteraciones en las cuales no se realiza la corrección, el vector de estado será igual al vector de estado a priori.

$$|\alpha| = \sqrt{\alpha_{k,y}^2 + \alpha_{k,y}^2 + \alpha_{k,z}^2} \quad (2.17)$$

$$h(\hat{X}_k^-) = \begin{bmatrix} \hat{\Phi}_k^- \\ \hat{\theta}_k^- \end{bmatrix} \quad (2.18)$$

Con todo ello se pueden definir las siguientes matrices que se utilizan en el Filtro de Kalman:

- La matriz A, se corresponde con la ecuación (2.7) y es resultado de derivar la ecuación (2.16) respecto del estado actual.

$$A = \begin{bmatrix} 1 + \omega_{k,y} \cos \Phi \tan \theta - \omega_{k,z} \sin \Phi \tan \theta & \omega_{k,y} \sin \Phi \sec^2 \theta - \omega_{k,z} \cos \Phi \sec^2 \theta & 0 \\ -\omega_{k,y} \sin \Phi - \omega_{k,z} \cos \Phi & 1 & 0 \\ \omega_{k,y} \cos \Phi \sec \theta - \omega_{k,z} \sin \Phi \sec \theta & \omega_{k,y} \sin \Phi \sec \theta \tan \theta + \omega_{k,z} \cos \Phi \sec \theta \tan \theta & 1 \end{bmatrix} \quad (2.19)$$

- La matriz Q, se utiliza en la ecuación (2.8), y representa el ruido de proceso. El valor σ_ω^2 se obtiene de las varianzas de los giróscopos [12] siendo su valor 0.001^2 .

$$Q = \begin{bmatrix} \sigma_\omega^2 & 0 & 0 \\ 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & \sigma_\omega^2 \end{bmatrix} \quad (2.20)$$

- La matriz H, se corresponde con la ecuación (2.12). Al derivar la ecuación (2.18) respecto de la estimación a priori se obtiene la siguiente matriz.

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (2.21)$$

- La matriz R, se utiliza en la ecuación (2.9), y representa el ruido de la medida. Los valores σ_Φ^2 y σ_θ^2 se obtienen experimentalmente y son la desviación estándar de los ángulos Roll y Pitch. Ambos valores son 0.01^2 .

$$R = \begin{bmatrix} \sigma_\Phi^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \quad (2.22)$$

En la figura 4 se muestra el diagrama de bloques completo del algoritmo del filtro de Kalman extendido.

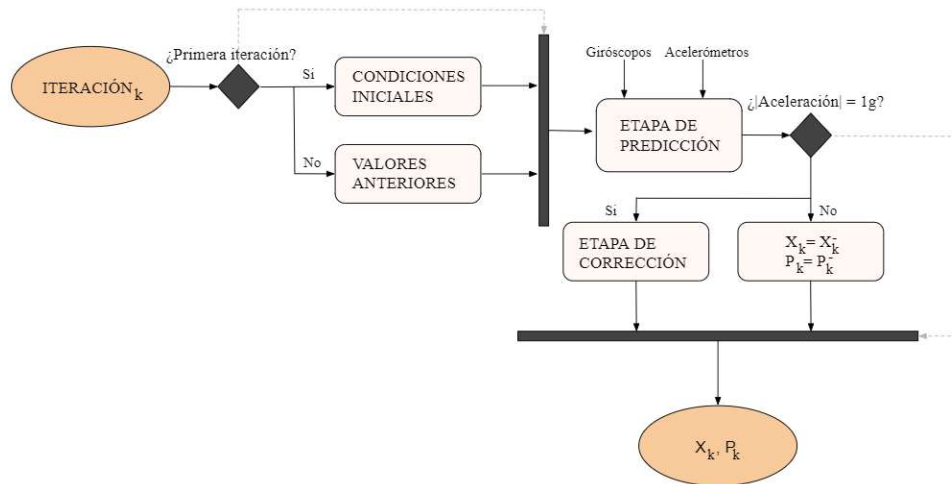


Figura 4. Diagrama de bloques del filtro de Kalman extendido [12].

2.4 Procesamiento de las señales

Los objetivos de este trabajo son la detección de pasos anómalos y otras características motoras, así que, para ello, se tiene que hacer un procesamiento de las señales obtenidas de los sensores. En este caso se ha realizado un análisis tanto en el dominio del tiempo como en el dominio de la frecuencia para detectar las diferentes características del paso.

2.4.1 Algoritmo de detección de pasos

Para la detección del paso se ha tomado como referencia un algoritmo que se basa en la detección de cruces por cero de la señal Pitch obtenida de la IMU [13].

El algoritmo de detección de pasos es diferente dependiendo de donde se coloque la IMU. En trabajos previos se ha posicionado el sensor en la parte alta de la pierna [12,13], sin embargo, se consigue una menor precisión que posicionándolo más abajo, por lo que en este trabajo se ha decidido colocar el sensor en el tobillo. De esta forma, al caminar, cuando se da un paso con la pierna que lleva la IMU, la señal pitch obtenida describe un determinado ángulo que se corresponde con el ángulo de apertura de la pierna con respecto a la vertical, y al dar un paso con la pierna que no lleva la IMU, se obtiene el ángulo en sentido contrario, tal y como se muestra en la figura 5.

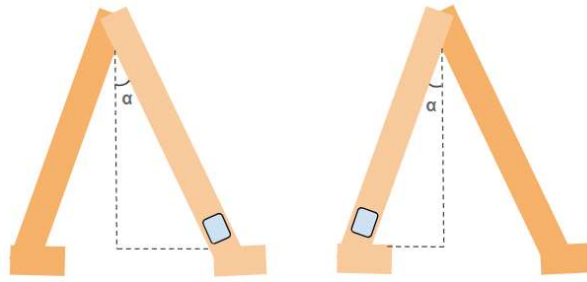


Figura 5. Esquema de las piernas de una persona al caminar.

En la figura 6 se muestra la señal de estimación del ángulo Pitch que se obtiene tras el procesamiento de las señales medidas por los sensores cuando una persona camina. Esta señal es una señal oscilante en la que cada máximo y mínimo se corresponden con un paso, es decir, cuando la apertura de las piernas es máxima. En definitiva, la detección de pasos consiste en la búsqueda de máximos y mínimos absolutos de la señal de estimación del ángulo Pitch.

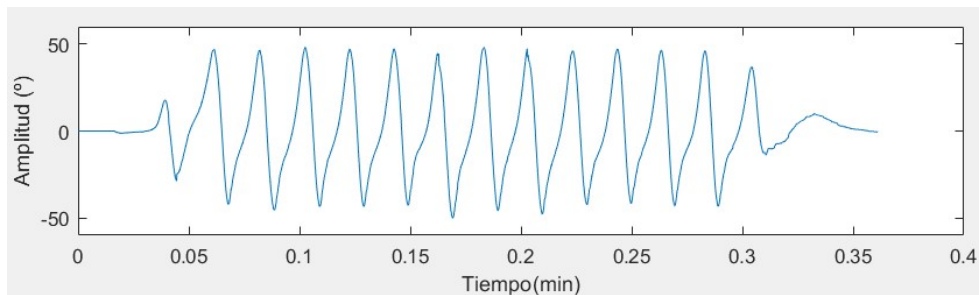


Figura 6. Señal de estimación del ángulo Pitch obtenida de una persona al caminar.

En primer lugar, para la detección de estos máximos y mínimos, se elimina el offset que pudiera encontrarse en la señal, es decir, se corrige y centra en cero, puesto que, debido a las derivas, ruido y diferentes errores, la señal se encuentra desplazada. Posteriormente se va analizando muestra por muestra buscando los cruces por cero, es decir, cuando la muestra anterior se encuentre por debajo de cero y la siguiente por encima, o viceversa. Al detectar dos cruces por cero, significa que en ese tramo habrá tanto un máximo como un mínimo, y de esta forma, se van encontrando todos los picos de la señal. Sin embargo, no todos los máximos y mínimos detectados corresponderían a un paso, puesto que se podrían detectar picos de ruido. Por ello, solo se tienen en cuenta aquellos máximos y mínimos que superen un determinado umbral.

2.4.2 Análisis espectral

Otra forma de conocer las características de la marcha es mediante un análisis frecuencial de la señal de estimación del ángulo Pitch. Se pretende realizar un análisis espectral de dicha señal para intentar diferenciar entre un paso normal y un paso anómalo, así como también el detectar los episodios de FOG (congelamiento del paso).

Para realizar este análisis se recurre a la transformada de Fourier de tiempo corto (Short-time Fourier transform). Esta herramienta matemática nos permite obtener la variación de la frecuencia de una señal a lo largo del tiempo. Esto es así ya que se divide la señal en diferentes ventanas de igual tamaño y se realiza la transformada de Fourier para cada tramo en concreto, obteniendo así, el espectrograma de la señal

Capítulo 3: Algoritmo propuesto

En este capítulo se explica cómo se ha diseñado e implementado el algoritmo que permite la consecución de los objetivos propuestos para este trabajo. En líneas generales, el algoritmo consiste en la detección de los diferentes pasos que da un individuo y conseguir diferenciarlos a partir de su amplitud y a su forma. También se extraen otras características derivadas del paso como son la cadencia, la amplitud, o la actividad de la persona, para obtener un perfil más completo del paciente. En la figura 7 se muestra el diagrama de bloques del algoritmo. A lo largo de este capítulo se profundizará en cada uno de estos bloques.

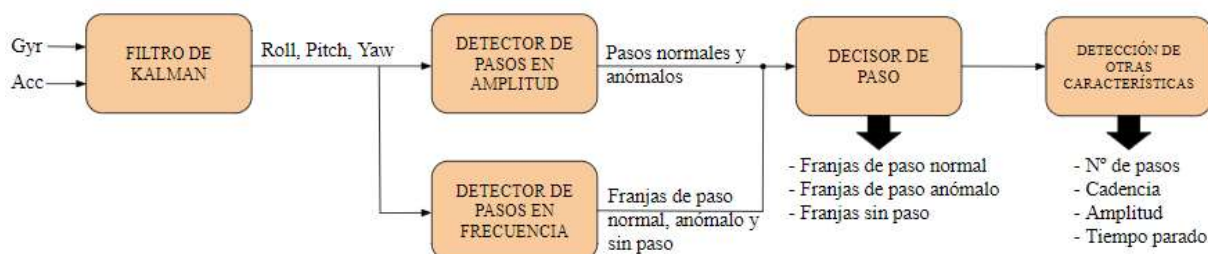


Figura 7. Diagrama de bloques del algoritmo propuesto.

3.1 Detección de pasos anómalos

Uno de los objetivos es la detección de los pasos normales y de los pasos que pueden presentar anomalías. Se ha realizado de dos formas diferentes: la primera forma de diferenciarlos es mediante la amplitud del paso, y la otra forma es mediante un análisis de la frecuencia. Posteriormente, con la información de ambas clasificaciones, se decide de forma definitiva cuáles son los pasos normales y cuáles son los pasos anómalos.

3.1.1 Detección de pasos anómalos en el dominio del tiempo

La razón para hacer una distinción entre los pasos normales y los pasos anómalos a partir de la amplitud de estos es porque los pasos de los enfermos de Parkinson suelen ser de amplitud menor que los que se observan en una persona normal. Tal y como se ha comentado en el capítulo anterior, el algoritmo realizado se basa en la detección de los cruces por cero de la señal pitch obtenida tras el procesamiento de las señales de la IMU. La figura 8 muestra el diagrama de bloques utilizado para la detección de los cruces por cero.

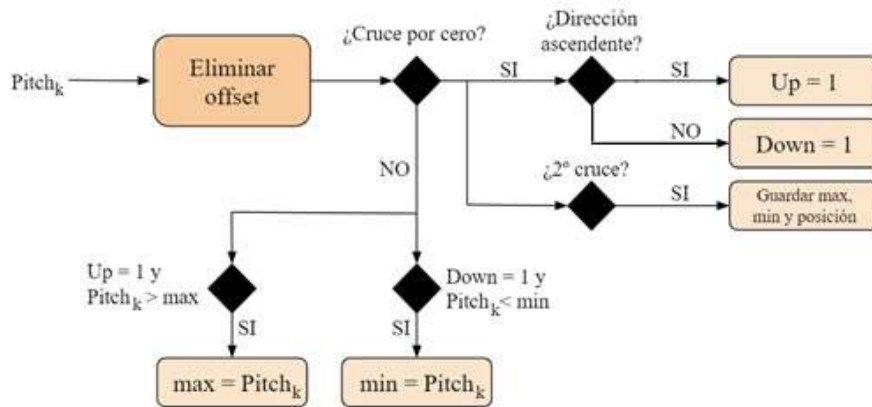


Figura 8. Diagrama de bloques del algoritmo de cruces por cero.

Para hacer este análisis se va estudiando muestra a muestra toda la señal de estimación del ángulo Pitch. En primer lugar, se realiza una media de los últimos 100 valores para poder eliminar el offset que pudiera tener la esta señal. Este valor obtenido se le resta a la amplitud de la muestra k del Pitch para colocarla en su sitio y poder centrar la señal en cero. Una vez hecho esto se buscan los cruces por cero. Habrá un cruce por cero siempre que el valor de la muestra anterior sea menor que cero y el valor de la muestra actual que estamos evaluando sea mayor que cero, o viceversa. Si no ha habido cruce por cero se comprueba si esa muestra se corresponde con un máximo o un mínimo local. Esto se hace comparando si el valor del Pitch en la iteración k es mayor que el valor de la variable máximo o menor que el valor de la variable mínimo. Inicialmente estas dos variables tienen el valor cero para que al recorrer la señal se puedan encontrar de forma correcta los máximos y mínimos. Cuando se encuentran dos cruces por cero se habrán detectado tanto un máximo como un mínimo, ya que al hacer un cruce en dirección ascendente se habrá encontrado un mínimo, y al hacer un cruce descendente se habrá

encontrado un máximo. Es entonces cuando se almacenan en un array el máximo, el mínimo y la posición en que ambos se encuentran.

De esta forma se pueden detectar todos los picos, máximos y mínimos, de la señal Pitch, pero no todos se corresponden con pasos. Para diferenciar entre pasos reales y picos de ruido de la señal, se aplica un umbral en amplitud. El valor de este umbral se ha definido tras la realización de diferentes pruebas con diferentes personas caminando en línea recta por un pasillo. De esta forma se han obtenido los valores de amplitud típicos de una persona normal. También se realizaron pruebas caminando en el mismo pasillo pero arrastrando los pies para obtener un umbral que distinga entre los pasos normales y los pasos arrastrando los pies. En la figura 9 se puede observar una de las pruebas hechas para encontrar el umbral de paso normal y en la figura 10 se muestra una de las pruebas arrastrando los pies. En ambas figuras se muestra la señal estimada del Pitch.

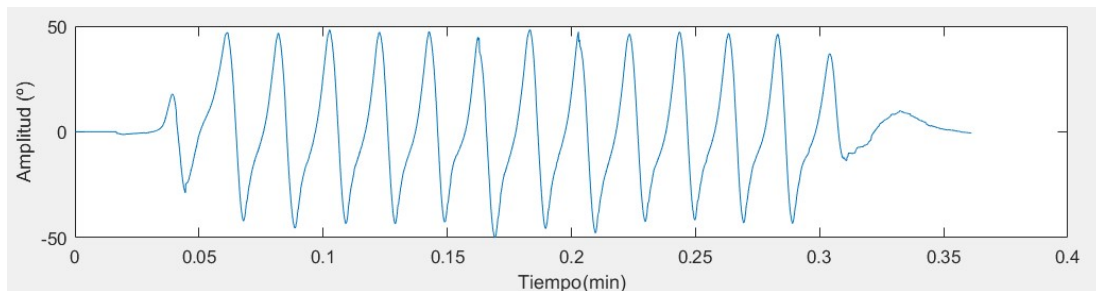


Figura 9. Señal pitch de una persona caminando de forma normal.

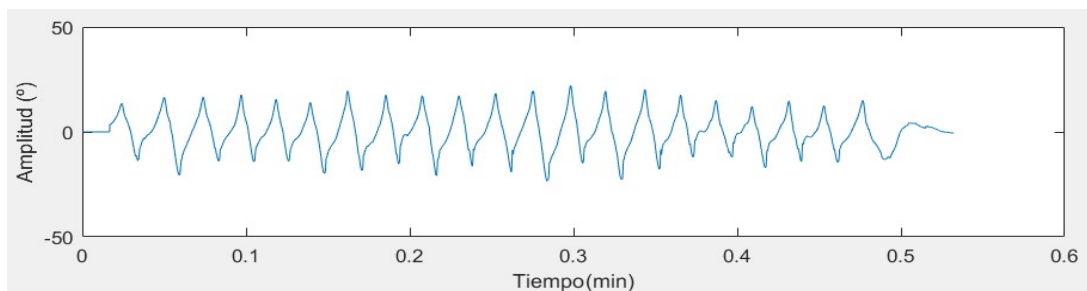


Figura 10. Señal pitch de una persona arrastrando los pies.

El valor elegido finalmente como umbral para el paso normal es de $\pm 30^\circ$ y $\pm 15^\circ$ para el paso arrastrando los pies. A partir de estos valores se clasifican todos los pasos que hemos guardado en el array en normales o anómalos, y se eliminan aquellos que no cumplen con ninguno de los dos umbrales, ya que se consideran ruido. Sin embargo, estos umbrales son muy genéricos y se busca que sean más personalizados, es decir, adaptados a cada persona, para lo que se realiza una reclasificación de los pasos. Para ello, se calcula la media de la amplitud de todos los pasos normales y de todos los pasos cortos

clasificados anteriormente. Después se evalúa uno a uno si el valor de la amplitud se asemeja más al valor de la media de pasos normales o al valor de la media de pasos cortos obtenidas y se asigna a la clase correspondiente. Este proceso se repite varias veces con el fin de asegurar la mejor clasificación posible. En la figura 11 se muestra una prueba realizada en el mismo pasillo mencionado anteriormente, en el que se ha mezclado pasos normales y pasos arrastrando los pies para comprobar el correcto funcionamiento del clasificador.

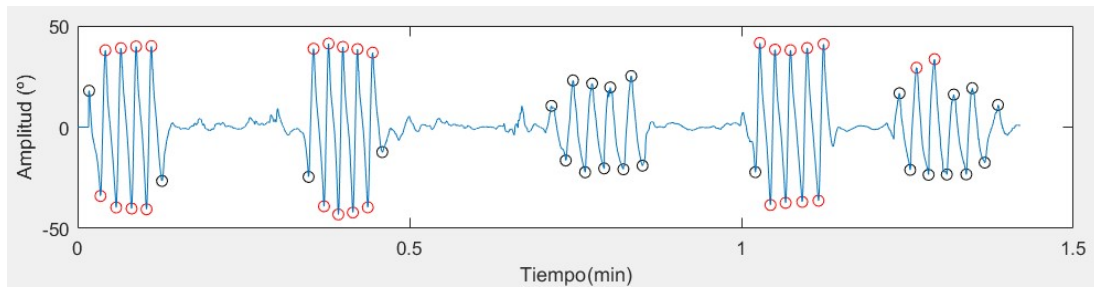


Figura 11. Detección de pasos normales(rojo) y pasos arrastrando los pies(negro).

3.1.2 Detección de pasos anómalos en el dominio de la frecuencia

Una amplitud pequeña del paso no siempre tiene que significar un deterioro motor, por ello es importante analizar la forma del paso también. A través de un análisis en el dominio de la frecuencia se puede extraer información adicional de la señal que ayude a entender mejor el estado del paciente. Además, se pueden detectar los momentos en los que la persona se encuentra parada y los episodios de FOG (congelamiento del paso) [7].

El primer paso para realizar este análisis consiste en realizar la transformada de Fourier de la señal Pitch completa para obtener la frecuencia dominante de la señal. Esta frecuencia se utilizará más adelante para encontrar en el espectrograma que se obtendrá después, el valor de la potencia a esa frecuencia. En el siguiente paso se realiza la transformada de Fourier de tiempo corto (Short-Fourier Transform) de la señal pitch. Como se ha explicado en el apartado teórico, esta herramienta permite obtener la variación de la frecuencia de una señal a lo largo del tiempo. De esta forma, se obtienen diferentes ventanas de igual tamaño a lo largo de todo el tiempo de la prueba en la que cada una tiene una frecuencia representativa. El espectrograma resultante se muestra en la figura 12.

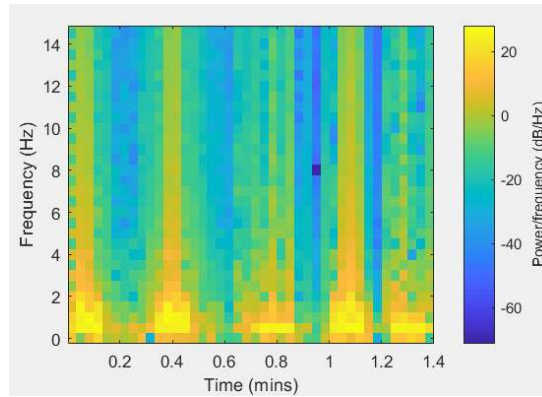


Figura 12. Espectrograma de la señal Pitch.

El tamaño de dichas ventanas se ha obtenido de forma experimental realizando diversas pruebas con diferentes señales de personas caminando evaluando desde un tamaño de ventana de 50 muestras hasta uno de 500 muestras. La elección del tamaño de la ventana es muy importante ya que si se realiza el análisis con un tamaño de ventana muy pequeño el tramo de señal analizado será muy corto y no se podrán observar las variaciones de la señal y, por lo tanto, no se encontrará el valor de frecuencia correcto. Por otro lado, si el tamaño de la ventana es muy grande no se obtiene una buena precisión y los resultados tampoco serán correctos. Finalmente se ha decidido un tamaño de ventana de 200 muestras (que correspondería a dos segundos) y además que no haya solapamiento entre las ventanas para diferenciar claramente los límites de cada franja. Estas características se utilizarán para todas las pruebas.

Para poder distinguir entre los pasos normales, pasos anómalos y los episodios de FOG se utilizan unos umbrales, que también han sido calculados de forma experimental al mismo tiempo que se hizo el cálculo del tamaño de la ventana. El valor umbral fijado para el paso normal es 25 dB/Hz y para el paso anómalo 10 dB/Hz. Aquellas ventanas que no cumplen dichos umbrales se considerarán zonas sin paso, en las que el individuo se encuentra parado o en reposo. Se recorre cada ventana del espectrograma y para la frecuencia dominante calculada anteriormente, se compara la potencia espectral de ese tramo con los umbrales obtenidos para hacer la clasificación, tal y como se muestra en el diagrama de bloques de la figura 13.

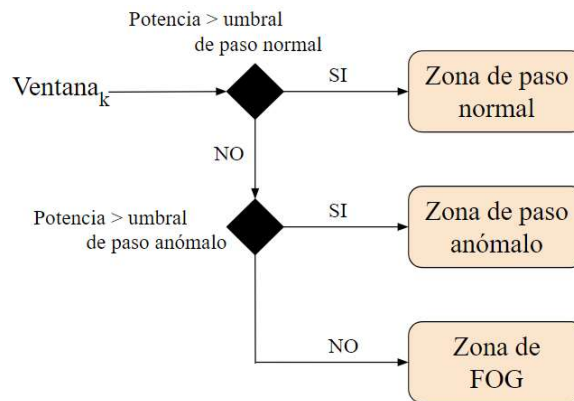


Figura 13. Diagrama de bloques de la clasificación de las ventanas de la señal pitch

De forma análoga al algoritmo de detección de pasos, los umbrales escogidos son genéricos y se quiere que sean más personalizados para adaptarse mejor a las características de cada persona. Para ello se realiza el promedio del valor de potencia espectral de todas las ventanas de cada clase y se procede a reclasificar cada ventana siendo los promedios calculados los nuevos umbrales. Este proceso se realiza varias veces para asegurar la mejor clasificación posible. Por último, se agrupan las franjas de una misma clase que sean consecutivas para hacer una franja de mayor tamaño. Un resultado ejemplo de esta clasificación se muestra en la figura 14.

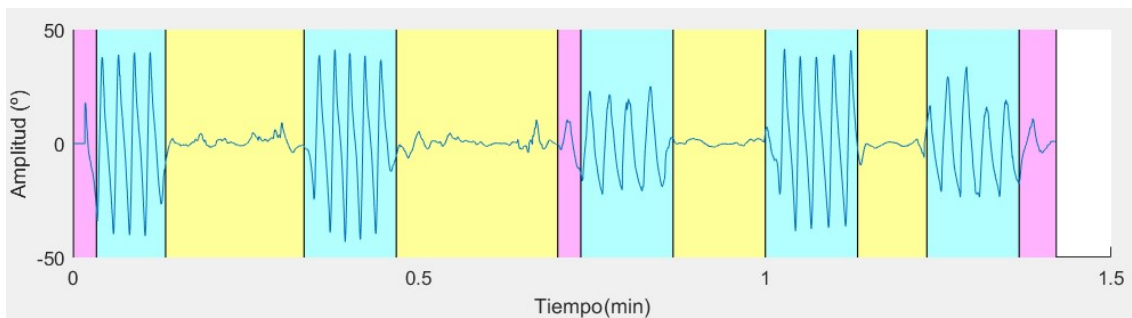


Figura 14. Detección de zonas de paso normal (azul), paso anómalo (magenta) y ausencia de paso (amarillo).

3.1.3 Decisor

Una vez clasificados los pasos tanto en el dominio del tiempo como en el dominio de la frecuencia, hay que decidir cuáles son las zonas de paso normal y las zonas de paso anómalo definitivas. Esto se hace porque un paso de amplitud pequeña no significa que sea un paso anómalo y, por otro lado, también puede haber errores en el análisis

frecuencial. Para ello se ha desarrollado un decisor que funciona tal y como se muestra en la figura 15.

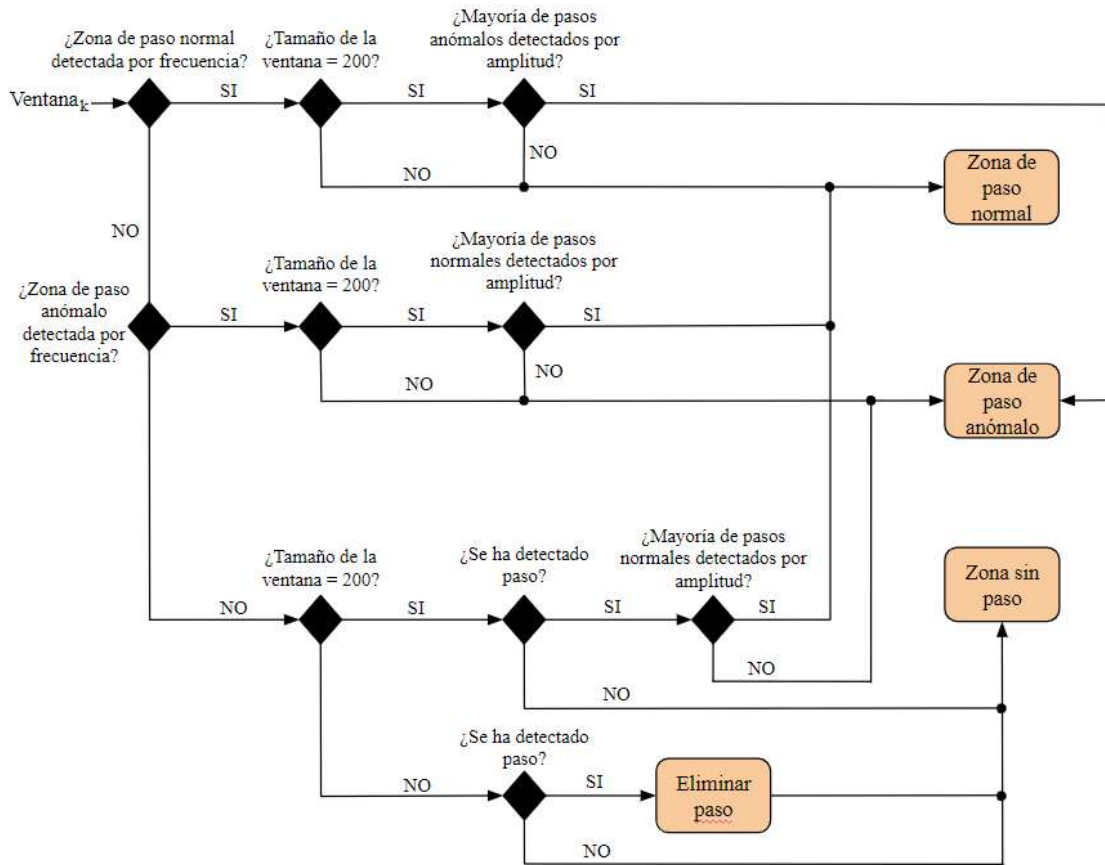


Figura 15. Diagrama de bloques del algoritmo del decisor.

3.2 Obtención de características derivadas del paso

Para tener un mejor conocimiento del estado y la evolución del paciente, no basta solo con detectar cuándo camina de forma normal y cuándo de forma anómala. Por ello también se han buscado otras características asociadas al paso que nos proporcionan más información. Estas características se han calculado para el total de la prueba y para cada hora, para poder analizar el estado del paciente durante el día. Las características que se han calculado son las siguientes:

- Número total de pasos: para calcularlo simplemente se suman todos los pasos que se han detectado. Se realiza tanto para pasos normales como para pasos anómalos. De esta forma se puede observar si la persona suele caminar más de forma normal o de forma extraña.

- Cadencia media del paso: es el tiempo que transcurre entre pasos. Para calcularlo se resta la posición de un paso menos la posición del anterior, y se divide entre cien (que es la frecuencia de muestreo) para obtener el tiempo entre esos dos pasos en segundos. Luego se hace una media con todos los valores obtenidos. Esta medida es útil porque normalmente los enfermos de Parkinson presentan una cadencia al caminar mayor que una persona normal, ya que caminan más lento.
- Actividad del paciente: esta característica representa el tiempo en la que la persona permanece parada, ya sea voluntaria o involuntariamente (FOG). Se calcula sumando el tamaño en muestras de todas las ventanas de FOG y dividiéndolo entre la frecuencia de muestreo de la IMU (100 muestras por segundo) para obtener el dato en segundos. Como las pruebas que se realizan son de larga duración, el dato obtenido se divide entre 60 para expresarlo en minutos y además se calcula también en porcentaje.
- Amplitud media del paso: se calcula realizando la media de la amplitud de los pasos detectados. Si bien se utiliza la amplitud para diferenciar entre un tipo de paso u otro y poder hacer la clasificación, también se quiere poder visualizar ese valor para que el médico tenga más información disponible. Como se ha explicado en apartados anteriores, normalmente el paso de los enfermos de Parkinson es de menor amplitud que el que da una persona normal, puesto que tienden a caminar arrastrando los pies.

3.3 Diseño de la interfaz de usuario

Para este trabajo se ha realizado una interfaz gráfica utilizando la herramienta GUIDE de Matlab. Gracias a esta interfaz se presentan de forma ordenada todos los resultados y todas las características detectadas en las pruebas realizadas a un paciente, de forma que sea de ayuda para que el médico tenga un mejor conocimiento del paciente y realice un mejor diagnóstico o seguimiento de la enfermedad. En este apartado se explica la funcionalidad y el desarrollo del mismo mientras que el funcionamiento se detalla en el apartado VI Manual de usuario.

La herramienta GUIDE se compone de un panel en el que se pueden arrastrar diferentes bloques, botones, tablas, gráficos y diferentes opciones de texto con el que

poder diseñar la aplicación. Las acciones que se realizan cuando el usuario de la aplicación interactúa con estos componentes se programan en un fichero de extensión .m de la misma forma en la que se ha realizado el resto del algoritmo. La interfaz realizada está dividida en tres bloques o ventanas que se explican en profundidad en este apartado, por lo que se tienen tres ficheros .m dedicados a la aplicación.

3.3.1 Ventana principal

Esta es la primera ventana que aparece al ejecutar el programa y en ella se puede visualizar los resultados de una prueba. Debido a que la mayoría de las pruebas tienden a ser de muy larga duración, se ha programado la interfaz para mostrar los resultados hora a hora para que resulte más sencillo el análisis. En la figura 16 se muestra el diseño de esta ventana principal realizado con la herramienta GUIDE.

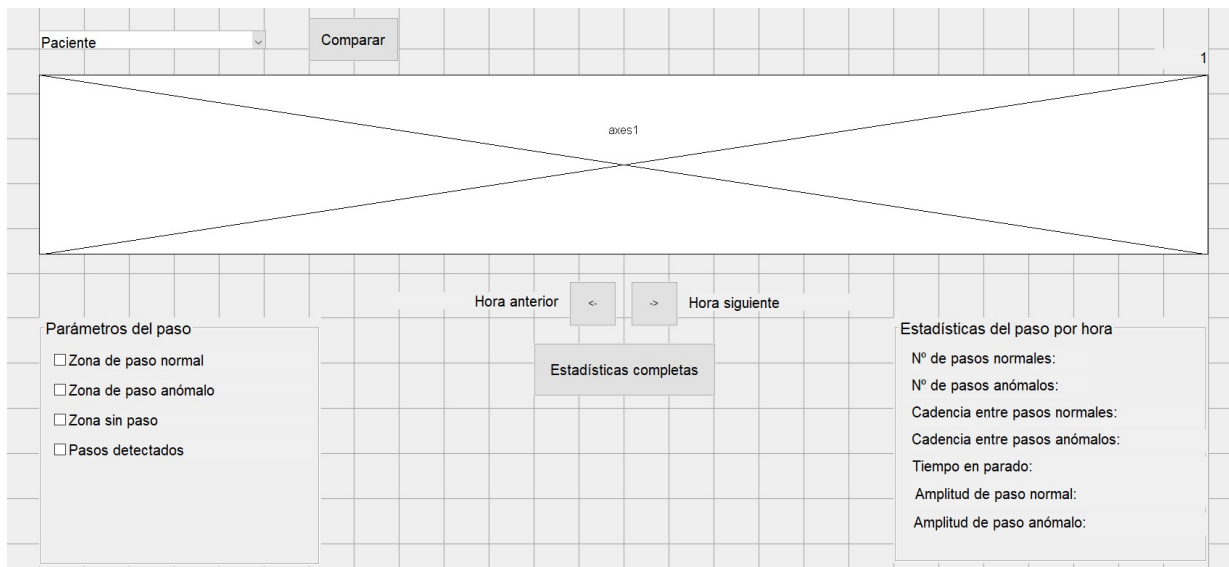


Figura 16. Diseño de la ventana principal de la interfaz de usuario.

Los elementos principales utilizados son:

- Menú pop-up: este menú se encuentra arriba a la izquierda y sirve para mostrar y elegir las pruebas disponibles para la visualización. Se ha programado para que muestre los títulos de los archivos de las pruebas que están guardadas en una determinada prueba. Cuando el usuario elige una de las pruebas, se llama a la función que contiene todo el algoritmo de detección de pasos explicado anteriormente.
- Ejes de visualización: cuando se ha elegido una prueba y se ha terminado de ejecutar el algoritmo, en ellos se visualizará la señal pitch de la prueba. Por

defecto se visualizará la primera hora, pero mediante otros botones de la interfaz se podrá desplazar a otras horas. También mediante otros botones se podrán visualizar los pasos detectados y las zonas que corresponden a los distintos pasos.

- Botones de hora anterior y hora siguiente: estos botones son los que permiten al usuario desplazarse hora a hora por la prueba.
- Texto de hora actual
- Texto de estadísticas: en él se escriben las estadísticas de cada hora.
- Checkbox: estos cuatro botones que se encuentran en el recuadro de “parámetros del paso” nos permiten visualizar en la señal los pasos detectados y las diferentes ventanas de paso.
- Botón de estadísticas completas: este botón abre una nueva página de la interfaz en la que se podrán consultar todas las estadísticas de la prueba elegida. Más adelante se explica con detalle.
- Botón de comparar: este botón abre otra página de la interfaz que permite comparar entre diferentes pruebas. También se explica su contenido más adelante.

3.3.2 Ventana de estadísticas

En la ventana de estadísticas se muestran todas las características del paso extraídas con el algoritmo. Se muestran las estadísticas para cada hora de la prueba y también las

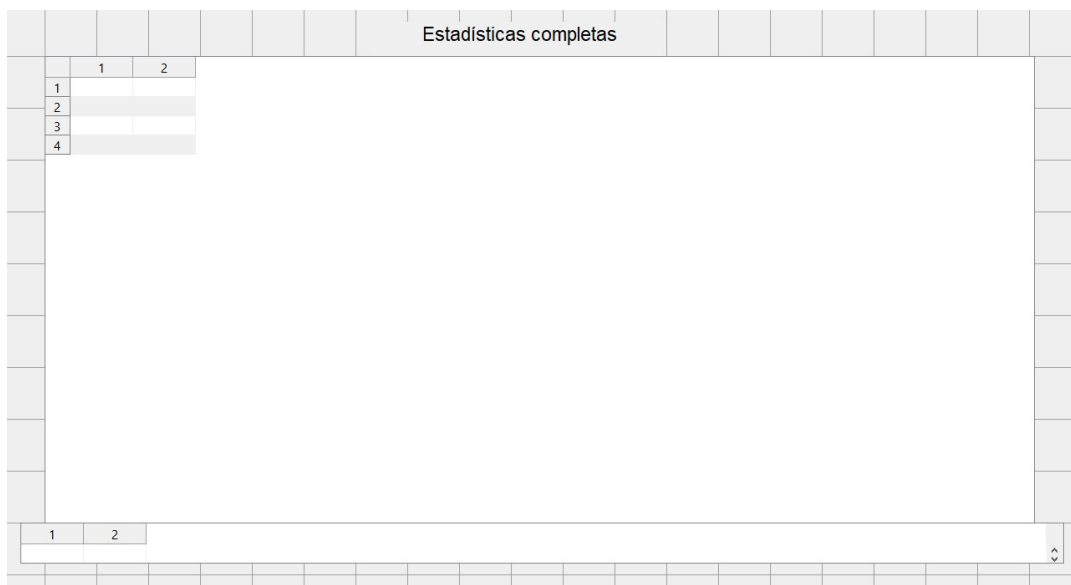


Figura 17. Diseño de la página de estadísticas de la interfaz de usuario.

estadísticas totales y medias en una tabla. En la figura 17 se muestra el diseño de la ventana principal realizado.

En este caso se ha utilizado un bloque de texto fijo para colocar el título de la página y dos tablas para exponer todos los valores. Las dimensiones de la primera tabla son dinámicas, ya que el número de filas dependerá del número de horas que dura la prueba, siendo este número de horas una variable obtenida del algoritmo. Como el algoritmo ya ha generado todos los datos de la prueba en la ventana principal, al presionar en dicha ventana el botón de ‘Estadísticas completas’, se pasan todas las variables necesarias, se abre la nueva interfaz y simplemente se representan en las tablas mediante la función “set”.

3.3.3 Ventana de comparación

Esta página es muy similar funcionalmente a la página principal, pero en vez de mostrar una prueba, se pueden mostrar hasta 3 pruebas para poder hacer una comparación. Se compone de tres menú pop-up para elegir las tres pruebas a comparar, tres ejes de visualización para representar la señal pitch de las tres pruebas, y tres recuadros de texto donde se imprimen las estadísticas totales de cada prueba. En la figura 18 se muestra el diseño para esta página.

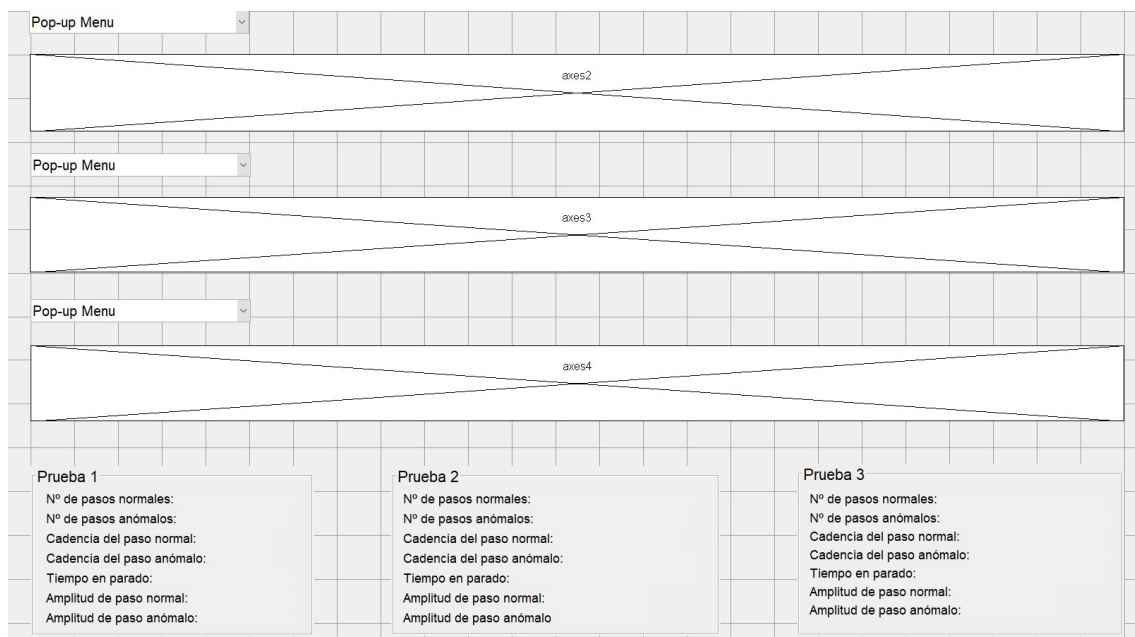


Figura 18. Diseño de la página de comparación de pruebas de la interfaz de usuario.

Capítulo 4: Resultados

En este capítulo se exponen los resultados de las pruebas realizadas. Además de realizar pruebas de simulación del comportamiento de un enfermo de Parkinson para optimizar el algoritmo diseñado, también se han podido efectuar pruebas con dos pacientes reales que tienen esta enfermedad. Se comentarán las características que presentan ambos pacientes para obtener las diferencias entre un estado temprano y otro más avanzado de la enfermedad.

4.1 Pruebas de optimización del algoritmo

En el capítulo anterior se ha explicado como se ha desarrollado el algoritmo para distinguir entre las diferentes características del paso de una persona, y se ha comentado que dicha distinción se realiza mediante diversos umbrales. El valor de dichos umbrales se había determinado tras realizar varias pruebas simulando diferentes características del paso. En este apartado se muestran los resultados obtenidos de la optimización de dichas pruebas. Para realizar la optimización se probó a realizar el análisis espectral variando el tamaño de ventana, desde 50 hasta 500 muestras, así como el valor de los umbrales hasta encontrar los valores que visualmente se percibía una buena clasificación.

La prueba realizada es en una secuencia que muestra diferentes formas de caminar. Dicha secuencia consiste, en primer lugar, en caminar de 15 a 20 pasos de forma normal, posteriormente recorrer de 15 a 20 pasos arrastrando los pies y luego alrededor de 10 pasos normales otra vez. A continuación, se realizan 4 series de bloqueo del paso y, por último, 10 pasos normales. Esta secuencia se repite un total de dos veces. En cuanto a las

series de bloqueo del paso, estas consisten en bloquearse al dar el primer paso e imitar que se ha quedado el pie pegado al suelo, como si no se pudiera caminar, y luego con la inercia recorrer 3-4 pasos muy rápidos y cortos para finalmente terminar con 5 pasos normales. En todas las pruebas se tomó nota del número exacto de pasos que dio cada participante para tener el *ground truth* y poder comparar con los resultados. Esta prueba se sometió a 5 individuos y cada participante realizó la prueba dos veces para poder tener un mayor número de datos con los que trabajar; por lo que cada prueba se ha tratado de forma independiente. Para obtener los resultados se ha analizado la señal Pitch estimada.

En las siguientes imágenes se visualiza solamente una parte de la prueba para poder observar mejor los resultados. Los resultados que se muestran son el resultado de aplicar el algoritmo a las pruebas una vez determinados los umbrales y se ha utilizado el color amarillo para marcar las zonas en las que no hay paso, el color azul para representar las zonas de paso normal, y el color magenta para mostrar el paso anómalo, que incluye los pasos arrastrando los pies y los bloqueos del paso. Visualmente es fácil detectar estas zonas pues una zona sin paso es aquella parte de la señal que no tiene una variación grande de amplitud y se encuentra en torno al cero, el paso normal es aquel que oscila de forma regular, y el paso anómalo es aquel que oscila presentando alguna irregularidad además de una menor amplitud que el paso normal. De todas las pruebas se ha obtenido el número de pasos totales detectados de cada tipo, así como los falsos positivos (FP) y falsos negativos (FN) del paso anómalo, como se mostrará en las respectivas tablas. Los falsos positivos son aquellos pasos que el algoritmo detecta como anómalos pero en realidad no lo son, mientras que los falsos negativos son aquellos pasos que deberían haber sido detectados como pasos anómalos pero que el algoritmo no los ha detectado así.

- Prueba 1

Ground truth		Resultados del algoritmo		Estadística	
Normal	Anómalo	Normal	Anómalo	FP	FN
133	68	125	93	15	4

Tabla 1. Ground truth y resultados de la prueba 1.

En este caso, como se muestra en la figura 19, se reconocen claramente las zonas de paso normal, de paso arrastrando los pies y las zonas sin paso. Los falsos negativos son

pocos en comparación a los 68 pasos anómalos que se dieron según el *ground truth*, sin embargo, hay bastantes falsos positivos (15). Esto es debido a que, normalmente el primer paso o el último de una serie de pasos normales suele ser más pequeño, detectándose así como un paso arrastrado. También se han detectado algunos pasos de más. Este problema se observa en la mayoría de las pruebas realizadas.

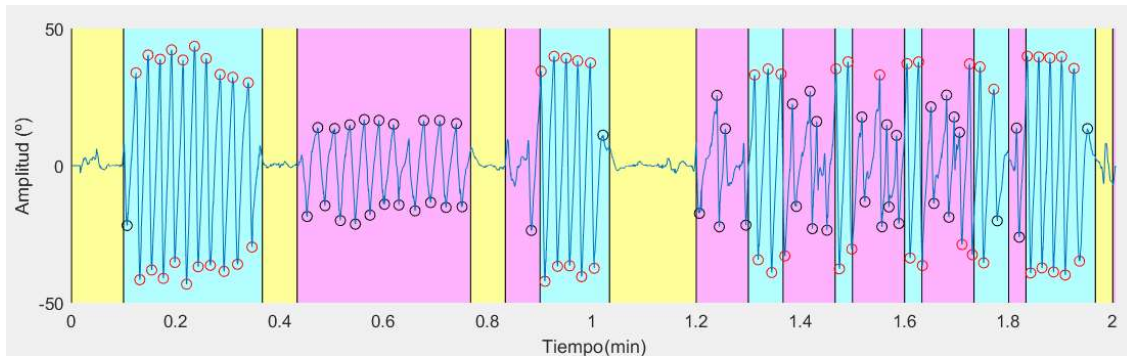


Figura 19. Parte de la prueba 1 de optimización del algoritmo.

- Prueba 2

Ground truth		Resultados del algoritmo		Estadística	
Normal	Anómalo	Normal	Anómalo	FP	FN
132	74	103	100	31	3

Tabla 2. Ground truth y resultados de la prueba 2.

En esta prueba, mostrada en la figura 20, se puede observar que la detección de las diferentes características se realiza de forma correcta, aunque presenta un error en el primer tramo de paso normal que produce un aumento en el número de falsos positivos, pues en el caso anterior era del 22.05% y en esta prueba es del 41.9%.

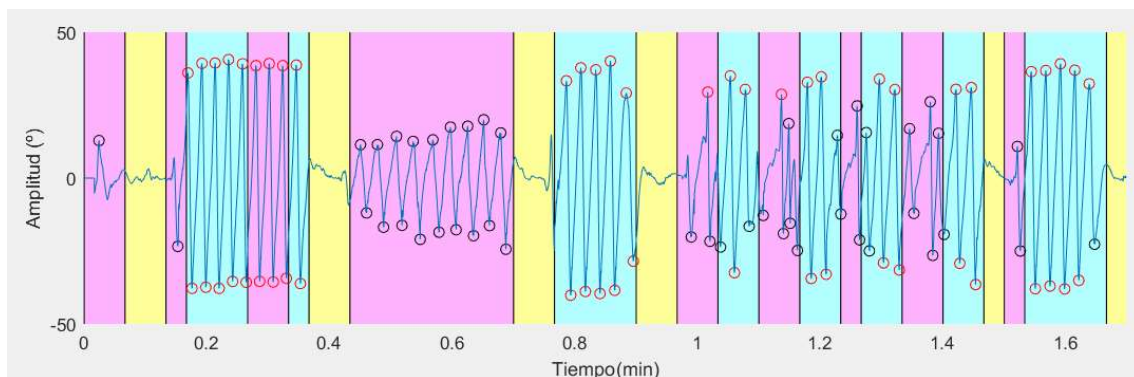


Figura 20. Parte de la prueba 2 de optimización del algoritmo.

- Prueba 3

Ground truth		Resultados del algoritmo		Estadística	
Normal	Anómalo	Normal	Anómalo	FP	FN
127	65	136	63	16	19

Tabla 3. Ground truth y resultados de la prueba 3.

En la siguiente prueba, figura 21, se puede observar que la detección es correcta, sin embargo, el número de falsos negativos es mayor, un 29.23% frente al 5.88% obtenido en la primera prueba. Esto es debido a que en los bloqueos el algoritmo solo detecta el principio de estos, dejándose algunos de los pasos pequeños iniciales sin detectar.

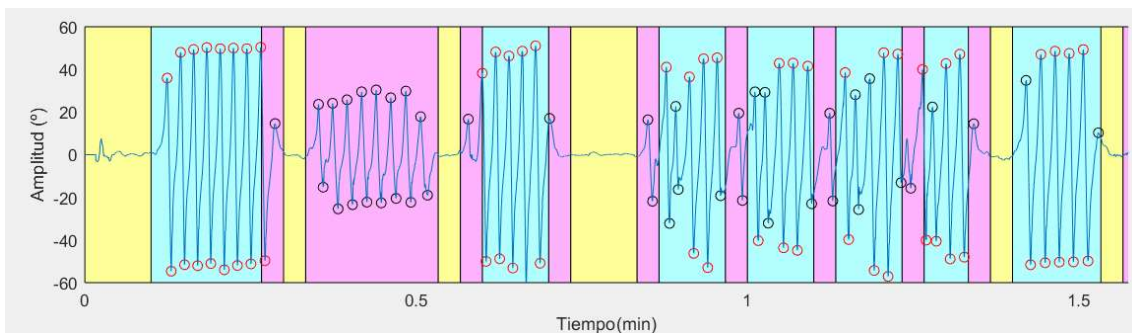


Figura 21. Parte de la prueba 3 de optimización del algoritmo.

- Prueba 4

Ground truth		Resultados del algoritmo		Estadística	
Normal	Anómalo	Normal	Anómalo	FP	FN
130	56	137	53	14	12

Tabla 4. Ground truth y resultados de la prueba 4.

En la siguiente prueba, mostrada en la figura 22, se observa lo mismo que en la anterior prueba. Solo se detecta el principio del bloqueo y, además, en este caso, uno de ellos no ha sido detectado debido a que presenta una forma similar a la de un paso normal, por lo que el porcentaje de falsos positivos y negativos es de un 10.76% y 21.43% respectivamente.

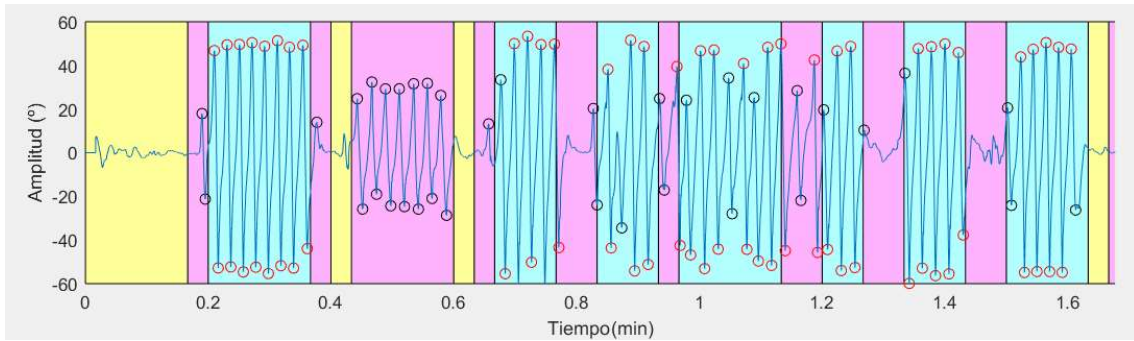


Figura 22. Parte de la prueba 4 de optimización del algoritmo.

- Prueba 5

Ground truth		Resultados del algoritmo		Estadística	
Normal	Anómalo	Normal	Anómalo	FP	FN
117	46	111	54	11	8

Tabla 5. Ground truth y resultados de la prueba 5.

En este caso (figura 23) ocurre lo mismo que en el caso anterior, pero los bloqueos no detectados son más. Esto ratifica la limitación para detectar el principio del bloqueo que se viene observando.

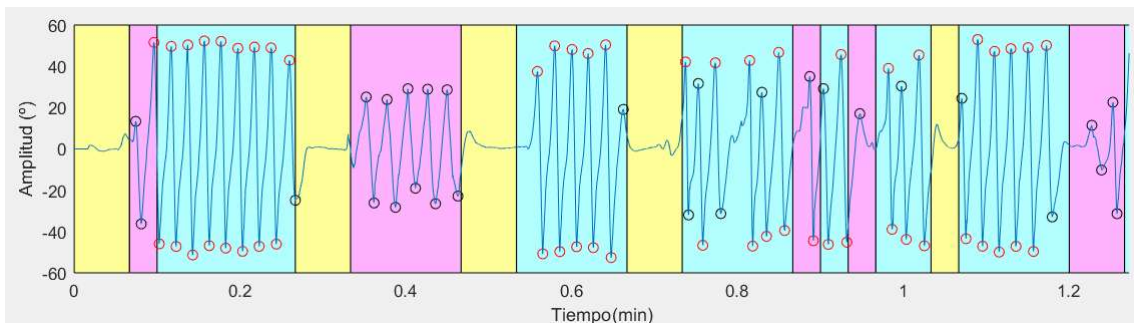


Figura 23. Parte de la prueba 5 de optimización del algoritmo.

- Prueba 6

Ground truth		Resultados del algoritmo		Estadística	
Normal	Anómalo	Normal	Anómalo	FP	FN
115	52	140	31	10	24

Tabla 6. Ground truth y resultados de la prueba 6.

En esta prueba, mostrada en la figura 24, se observa cómo los falsos negativos son mayores debido a que no se detecta la zona de paso arrastrado aunque se hayan detectado los bloqueos de forma correcta. Esto es así porque son pruebas donde se simulan los movimientos y en ocasiones se exageran. De hecho, se puede observar cómo la mitad de los pasos de esa zona presentan una amplitud que entra dentro de los límites de un paso normal.

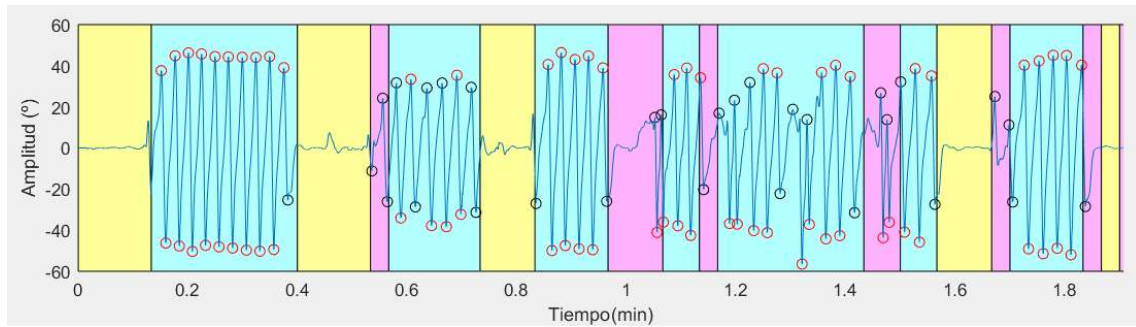


Figura 24. Parte de la prueba 6 de optimización del algoritmo.

- Prueba 7

Ground truth		Resultados del algoritmo		Estadística	
Normal	Anómalo	Normal	Anómalo	FP	FN
126	68	130	43	8	32

Tabla 7. Ground truth y resultados de la prueba 7.

En esta prueba (figura 25) tampoco se detecta el tramo de paso arrastrando los pies por lo que el porcentaje de falsos positivos es similar a la prueba anterior (24.61%). Sin embargo, los bloqueos se detectan mejor.

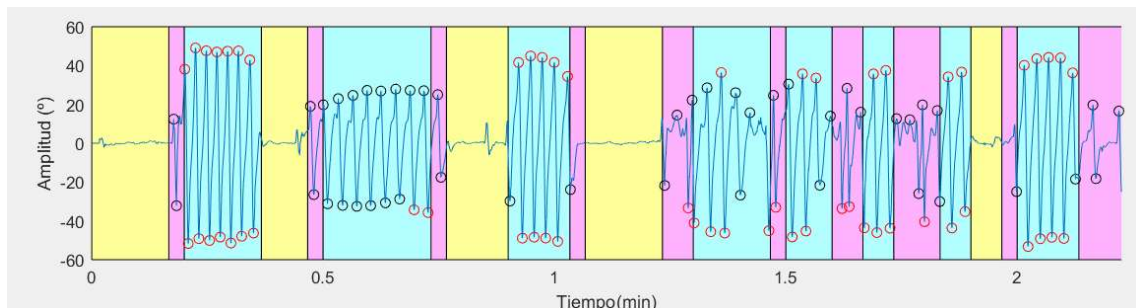


Figura 25. Parte de la prueba 7 de optimización del algoritmo.

- Prueba 8

Ground truth		Resultados del algoritmo		Estadística	
Normal	Anómalo	Normal	Anómalo	FP	FN
129	71	186	22	6	50

Tabla 8. Ground truth y resultados de la prueba 8.

Para la siguiente prueba, mostrada en la figura 26, también se observa lo mismo que en los dos casos previos y, además, aunque detecta cuándo se producen los bloqueos no detecta todos los pasos que lo componen.

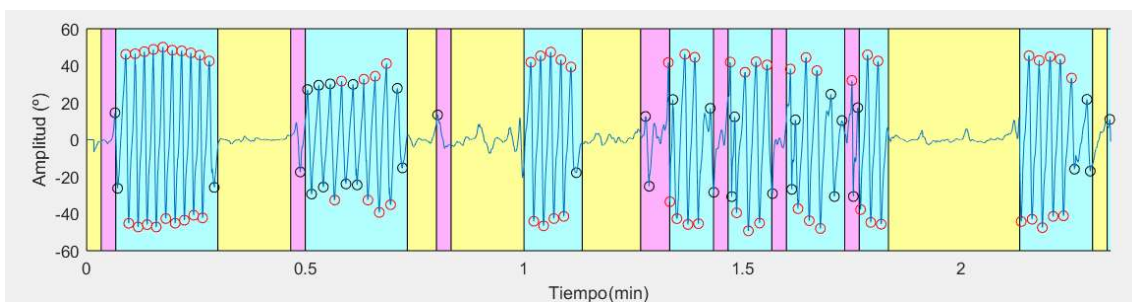


Figura 26. Parte de la prueba 8 de optimización del algoritmo.

- Prueba 9

Ground truth		Resultados del algoritmo		Estadística	
Normal	Anómalo	Normal	Anómalo	FP	FN
142	36	146	32	12	17

Tabla 9. Ground truth y resultados de la prueba 9.

El error observado en esta prueba (figura 27) es diferente a los de pruebas anteriores ya que el algoritmo detecta los bloqueos como momentos sin paso lo que hace que haya un mayor número de falsos negativos (11.64%). También se aprecia cómo se detecta un pico de ruido como paso anómalo debido a su gran amplitud.

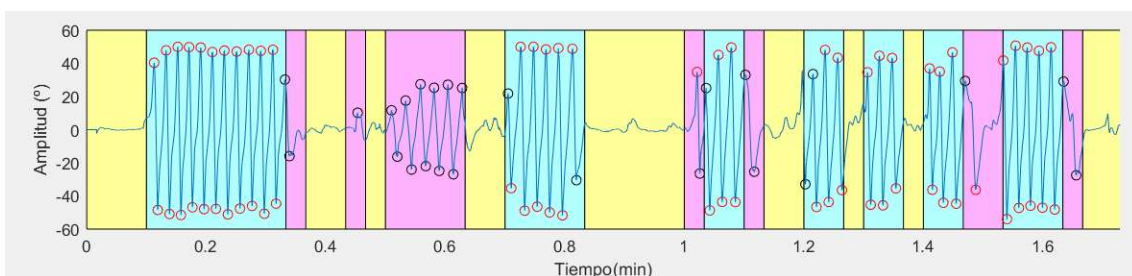


Figura 27. Parte de la prueba 9 de optimización del algoritmo.

- Prueba 10

Ground truth		Resultados del algoritmo		Estadística	
Normal	Anómalo	Normal	Anómalo	FP	FN
133	61	156	45	11	45

Tabla 10. Ground truth y resultados de la prueba 10.

En esta última prueba (figura 28) los resultados de falsos negativos y falsos positivos son grandes, 24.4% y 28.84% respectivamente, pues no se detecta de forma correcta la zona de paso anómalo ni los bloqueos del paso.

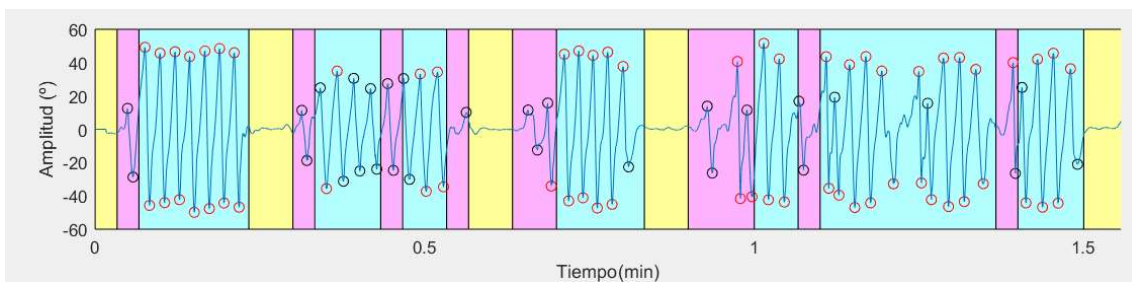


Figura 28. Parte de la prueba 10 de optimización del algoritmo.

En la tabla 11 se presenta un resumen de los resultados de las diferentes pruebas realizadas tras aplicar el algoritmo ya optimizado.

Prueba	Ground truth		Resultados del algoritmo		Estadística	
	Normal	Anómalo	Normal	Anómalo	FP	FN
1	133	68	125	93	15	4
2	132	74	103	100	31	3
3	127	65	136	63	16	19
4	130	56	137	53	14	12
5	117	46	111	54	11	8
6	115	52	140	33	10	24
7	126	68	130	43	8	32
8	129	71	186	22	6	50
9	142	36	146	32	12	17
10	133	61	156	45	11	45

Tabla 11. Ground truth y resultados de todas las pruebas.

En conclusión, los resultados obtenidos de las pruebas de simulación tras la aplicación del algoritmo optimizado muestran que en general, la detección de pasos es correcta pero presenta alguna limitación. En primer lugar destacar que las pruebas son simulaciones del paso de un enfermo de Parkinson y en ocasiones no ha sido bien ejecutado o ha sido exagerado de forma que ha afectado a la detección. También se ha comprobado que cada persona realiza los bloqueos del paso de forma diferente, obteniéndose diferentes casos, no detección, detección como zona sin paso o la detección incompleta de los mismos. Por último, mencionar que se detectan más pasos anómalos de los que realmente hay. Por una parte, es debido a que en ocasiones el primer y último paso de la marcha es de menor amplitud que el resto por lo que se clasifica como anómalo cuando no lo es. Por otra parte, el algoritmo detecta algunos picos de ruido aislados que son de gran amplitud y que en realidad no se corresponden con ningún paso.

4.2 Validación del algoritmo con pacientes reales

Una vez optimizado el algoritmo para que pueda detectar las características deseadas, se comprueba el funcionamiento del mismo con pruebas de pacientes reales. Para la realización de las pruebas se utilizó una IMU ubicada en el tobillo que los pacientes se colocaban tras el aseo diario y ponían a cargar durante la noche cuando se la quitaban por lo que se monitorizó a los pacientes durante el día de forma continuada en las que realizaban tareas cotidianas. En total se dispone de 16 días de monitorización correspondientes a dos pacientes con distinto grado de la enfermedad de Parkinson (7 de un paciente y 9 de otro). Cada prueba se corresponde con uno de estos días de monitorización y tienen como duración entre 10 y 15 horas. En este caso es imposible tener un *ground truth* por lo que para poder entender un poco mejor los resultados se recopilaban sus rutinas diarias.

4.2.1 Rutina de los pacientes

Conocer la rutina diaria de los pacientes puede ser de ayuda a la hora de evaluar las pruebas realizadas, pues estas consistieron en llevar colocado el sensor en el tobillo desde que el paciente se levantaba por la mañana hasta que se acostaba por la noche. La rutina de los dos pacientes que han realizado las pruebas son las siguientes:

- Paciente P01-V1:

El primer paciente se suele levantar a las 7 de la mañana y en torno a las 10 sale de su domicilio para ir a la biblioteca. Vuelve a casa a comer sobre las 13 o 13:30 horas. Luego se echa una siesta de una hora aproximadamente y alrededor de las 16:30 ve la televisión. Más tarde, se sienta en el ordenador. Este paciente utiliza bastón para caminar.

- Paciente P02-V1:

El segundo paciente se suele levantar en torno a las 7 u 8 de la mañana y entre las 8:30 y las 9:30 suele ir a caminar. Vuelve a su casa a las 13:30 aproximadamente para comer y luego se echa la siesta. Por la tarde suele salir a hacer varios recados y suele acostarse entre las 11 y media o 12 de la noche. El paciente dice que no se bloquea al caminar. Entre las pruebas que se disponen de este paciente hay una en la que se fue de excursión, por lo que durante esta prueba caminó más de lo normal. Este paciente se encuentra en una etapa más temprana de la enfermedad que el anterior paciente y aparentemente no se detectan síntomas motores de la enfermedad.

4.2.2 Resultados del paciente P01-V1

A continuación, se muestran los resultados de las pruebas realizadas con el primer paciente (P01-V1), que es el que se encuentra en un estado más avanzado de la enfermedad.

En las figuras 29 se observa cómo el algoritmo distingue entre las zonas de paso normal, paso anómalo y reposo para esta persona. A simple vista se puede comprobar que se distinguen de forma correcta aunque se puede apreciar la detección de algunos picos de ruido que no deberían ser detectados al igual que pasaba con las pruebas de optimización del algoritmo.

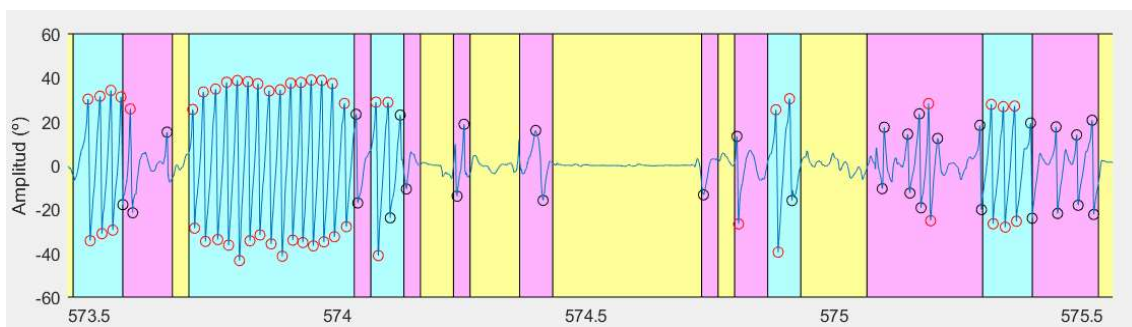


Figura 29. Detección de paso normal, paso anómalo y zona sin paso en el paciente P01-V1.

En la figura 30, se muestra una secuencia de alrededor de 6 minutos en la que el individuo se encuentra aparentemente en reposo y se detecta varios picos de ruido debido a la gran amplitud de estos. Hay que tener en cuenta que una de las limitaciones del algoritmo es que el número total de pasos anómalos es mayor que el real.

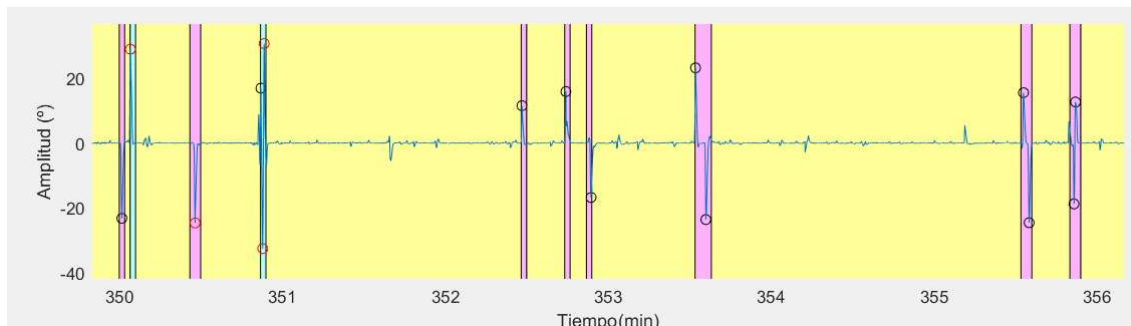


Figura 30. Detección de picos de ruido como paso anómalo en el paciente P01-V1.

Aun así, tal y como muestran la figura 31, lo normal en este paciente es encontrar periodos de tiempo en los que domina el paso anómalo, dando a entender que esta persona presenta una dificultad para moverse. Esta forma de caminar podría tomarse como referencia para compararlo en las siguientes evaluaciones y estudiar la evolución de la enfermedad.

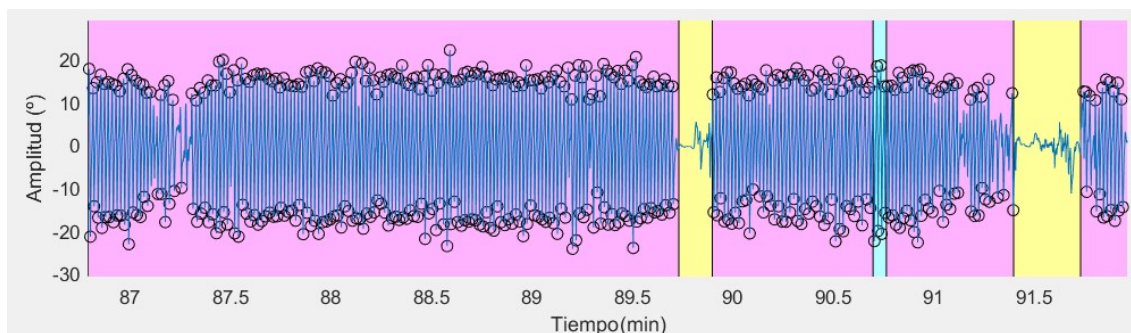


Figura 31. Detección de zonas de paso anómalo en el paciente P01-V1.

En la tabla 12 se muestran los resultados totales de todas las pruebas analizadas. Como se puede observar, el predominio de los pasos anómalos es evidente. La cadencia del paso anómalo es bastante alta, llegando en ocasiones a superar el segundo entre pasos. En conclusión, se podría decir que el paciente no camina de una forma segura y que duda al efectuar el paso. En cuanto a la amplitud del paso, se observa que tanto en la primera como la tercera prueba el valor es más pequeño del esperado. Por último, hay que destacar que esta persona no es muy activa. Normalmente permanece más del 90% del tiempo

parada. Esto puede deberse a la dificultad que le supone caminar y al encontrarse inseguro o incomodo opta por permanecer sentado o en reposo gran parte del día. Además, el tiempo agravará su situación, puesto que no solo perderá facultades motoras debido a la enfermedad de Parkinson, si no que también lo hará por la escasa movilidad que presenta.

Prueba	Número de pasos normales	Número de pasos anómalos	Cadencia del paso normal (s)	Cadencia del paso anómalo (s)	Amplitud del paso normal (°)	Amplitud del paso anómalo (°)	Tiempo parado (%)
1	465	3380	0.6377	0.8895	19.38	13.85	90.23
2	1050	1815	0.5507	0.7337	27.68	16.65	94.23
3	102	1130	0.8102	1.1435	13.74	8.65	95.64
4	1403	1873	0.7739	1.0150	28.69	16.33	91.21
5	652	1165	0.5256	0.7975	24.22	15.58	95.90
6	1060	2864	0.5775	0.6725	28.87	13.63	93.10
7	3834	1322	0.5030	0.5536	26.96	15.21	67.54

Tabla 12. Estadísticas de las pruebas del paciente P01-V1.

La última prueba ofrece unos resultados un tanto diferentes. Esto es porque dicha prueba es de corta duración y se realizó en el centro médico. Se observa como el predominio en este caso es del paso normal y además la cadencia entre pasos es menor que las obtenidas en el resto de las pruebas. Esto confirma la hipótesis de que en la consulta médica los síntomas no se presentan de igual forma que en un entorno cotidiano, y por eso es importante el desarrollo de esta aplicación que permite obtener las características motoras del paciente cuando se encuentra en su entorno diario.

4.2.3 Resultados del paciente P02-V1

En este apartado se muestran los resultados de las pruebas realizadas al segundo paciente. Esta persona se encuentra en un mejor estado que la anterior, así que se espera que los resultados sean mejores.

En las figuras 32 se muestran parte de una de las pruebas realizadas a este paciente en las que se comprueba a simple vista que el algoritmo realiza de forma correcta la distinción entre los diferentes tipos de paso. Al igual que ocurre con el otro paciente, se puede apreciar como el algoritmo también detecta como paso anómalo picos de ruido que no se corresponden con pasos.

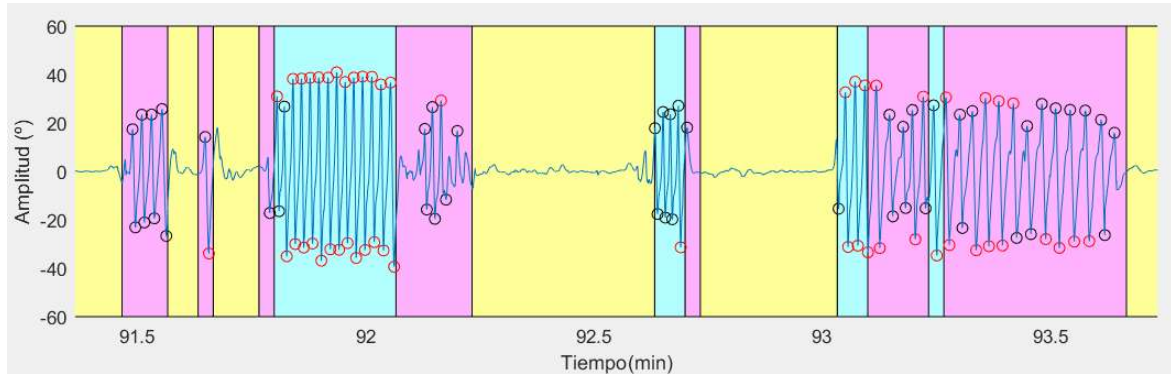


Figura 32. Detección de paso normal, paso anómalo y zona sin paso en el paciente P02-V1.

En este paciente lo que se observa principalmente son franjas de larga duración en las que predomina un paso normal. Por ejemplo, en la figura 33 se muestra un periodo de aproximadamente 20 minutos que, además, corresponde a las primeras horas del día que es cuando el paciente sale a caminar.

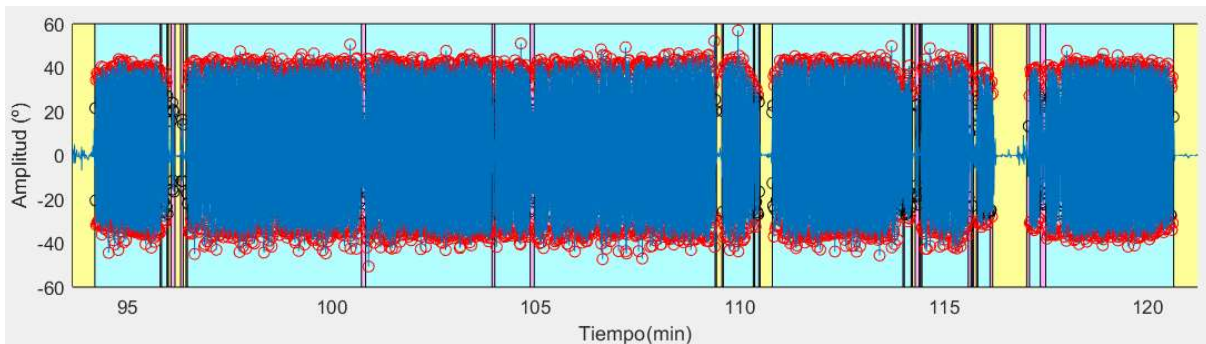


Figura 33. Detección de zona de paso normal en el paciente P02-V1.

En la tabla 2 se muestran las características completas de todas las pruebas realizadas a este paciente. Al igual que hemos visto en las imágenes anteriores se puede comprobar que el paso predominante es el paso normal, y además el número de pasos que da esta persona es mucho mayor que el obtenido para el otro paciente. Esto quiere decir que es una persona más activa que la anterior, y también se comprueba por el porcentaje de tiempo que se encuentra en reposo, que en este caso es menor que en el anterior. En el caso de la cadencia y de la amplitud, el resultado son valores bastante normales por lo

que en general se puede decir que este paciente tiene una buena movilidad y no presenta grandes síntomas.

Prueba	Número de pasos normales	Número de pasos anómalos	Cadencia del paso normal (s)	Cadencia del paso anómalo (s)	Amplitud del paso normal (°)	Amplitud del paso anómalo (°)	Tiempo parado (%)
1	2802	1749	0.5292	0.7326	28.66	16.86	81.07
2	5870	2766	0.5418	0.6672	27.17	16.35	74.85
3	4848	3825	0.5318	0.6946	28.21	15.46	88.19
4	19362	8723	0.5250	0.6332	34.04	16.67	68.35
5	8209	5452	0.5341	0.7096	30.05	15.68	81.00
6	8693	4439	0.5431	0.6809	29.22	17.16	83.50
7	14142	6308	0.5387	0.6885	27.85	16.17	75.89
8	9512	5896	0.5254	0.6567	29.49	16.12	81.75
9	7677	6084	0.5425	0.6626	27.95	15.62	82.71

Tabla 13. Estadísticas de las pruebas del paciente P02-V1.

4.2.4 Comparativa entre pacientes

Tal y como se ha mencionado anteriormente, los dos pacientes que han sido sometidos a las pruebas se encuentran en un estado diferente de enfermedad, por ello, es interesante comparar los resultados obtenidos de ambos y observar las diferencias que se encuentran. Para ello se ha calculado la media de todos los parámetros obtenidos de todas las pruebas, mostrándose a continuación los resultados en la tabla 14.

Paciente	Cadencia del paso normal (s)	Cadencia del paso anómalo (s)	Amplitud del paso normal (°)	Amplitud del paso anormal (°)	Tiempo parado (%)
P01-V1	0.6255	0.829	24.22	14.27	89.69
P02-V1	0.5345	0.6806	29.18	16.23	79.70

Tabla 14. Estadísticas medias de las pruebas de ambos pacientes.

En primer lugar, se puede observar cómo en el primer paciente predominan los pasos anómalos frente a los normales, siendo lo contrario en el segundo paciente. La cadencia, tanto para paso normal como para paso anómalo, es mayor en el primer paciente, que es el paciente que se encuentra en un estado más evolucionado de la enfermedad. Esto confirma que el primer paciente tiene más dificultad para caminar y una mayor lentitud. En cuanto a la amplitud del paso, esta es mayor en el segundo paciente y, además, la diferencia entre la amplitud del paso normal y la de un paso anómalo también es mayor, de forma que se puede reconocer mejor cada tipo de paso que en el caso del primer paciente. Otra característica importante de evaluar es la actividad de la persona. En el caso del primer paciente, su actividad es mucho menor que la del segundo. Al tener una peor movilidad y una gran dificultad para caminar lo normal es que la persona cada vez reduzca más su actividad debido a los problemas que enfrenta al intentar caminar.

Independientemente de los resultados obtenidos para cada uno de los pacientes, se ha demostrado la viabilidad de utilizar un sistema inercial para el seguimiento de la enfermedad del Parkinson. Estos resultados demuestran que el algoritmo, aun estando en fase inicial, ya puede dar información objetiva útil para la evaluación de la enfermedad por parte de un facultativo.

Capítulo 5: Conclusiones y trabajos futuros

Para la realización de este Trabajo Fin de Grado se han marcado una serie de objetivos a cumplir. Se pretendía conseguir una aplicación que permitiera la detección de anomalías en enfermos de Parkinson mediante el estudio del paso, de forma que se pueda utilizar en un entorno cotidiano. Para medir el paso se ha utilizado un sensor inercial colocado en el tobillo del paciente que gracias a su pequeño tamaño es fácil y cómodo de llevar, lo que lo hace perfecto para ser utilizado en el día a día.

A continuación se recopilan las conclusiones alcanzadas. El objetivo anteriormente mencionado es muy ambicioso por lo que este trabajo se puede definir como una aportación inicial para demostrar su viabilidad pero todavía queda un gran margen de mejora. En el último apartado de este capítulo se proponen algunas de las posibles mejoras.

5.1 Conclusiones

Para conseguir el principal objetivo del trabajo se han desarrollado una serie de objetivos parciales que se desarrollan a continuación.

La principal aportación a este Trabajo Fin de Grado ha sido el desarrollo del algoritmo de detección de paso que consigue diferenciar cuándo una persona camina de forma normal o presenta alguna anomalía como, por ejemplo, caminar arrastrando los pies o bloquearse al caminar. Este algoritmo se basa en el análisis de la señal del Pitch obtenida tanto en el tiempo como en la frecuencia. La clasificación del paso se ha

realizado en base a unos umbrales definidos empíricamente en base a los resultados simulados de distintos tipos de pasos ejecutados por diferentes voluntarios. También se han obtenido parámetros característicos del paso como la amplitud, la cadencia e incluso el tiempo en el que la persona se encuentra en reposo.

En general el algoritmo presenta buenos resultados. Se ha probado en dos pacientes con enfermedad de Parkinson que se encuentran en diferente estado. Esto ha permitido poder comparar entre ambos observando claramente las diferencias entre ellos. Estos valores podrían considerarse sus parámetros basales y si se repitiera la medida en unos meses se podría ver la evolución de una forma objetiva. Por lo tanto, estos datos recogidos a lo largo del tiempo podrían ser una ayuda objetiva al facultativo en el seguimiento de la enfermedad.

Por último, se ha realizado una interfaz de usuario dirigida al uso por parte de un profesional médico que sea fácil de utilizar y que permita visualizar todas las características obtenidas del paciente de forma que se pueda realizar un buen diagnóstico objetivo.

En conclusión, se ha obtenido un sistema que cumple con el objetivo principal del proyecto que es la evaluación objetiva de los síntomas motores de un enfermo de Parkinson en su entorno cotidiano.

5.2 Trabajos futuros

Una vez finalizado este Trabajo Fin de Grado se puede continuar en esta línea de trabajo mejorando el algoritmo desarrollado o introduciendo nuevas ideas que complementen el proyecto. En general se proponen las siguientes mejoras:

- Eliminar los picos de ruido

Como se ha comprobado en el capítulo de resultados, en ocasiones el algoritmo detecta picos de ruido como pasos anómalos. Lo ideal sería mejorar la detección de pasos para evitar detectar ese ruido y no falsear el total número de pasos anómalos.

- Detección de los bloqueos

Actualmente, el algoritmo detecta el bloqueo del paso como una zona de paso anómalo. Es cierto que estos episodios no son normales en las personas pero, aún así, sería conveniente detectarlos de forma independiente a los pasos anómalos ya que saber

la cantidad de bloqueos que ocurren es una información de gran utilidad para definir los estados de On y Off.

- Pruebas iniciales de cada paciente

Los umbrales que se utilizan para la detección del paso son unos umbrales calculados de forma experimental. Estos umbrales se intentan particularizar para el paciente en concreto pero esto se realiza con la misma prueba que se pretende evaluar, lo que no deja de ser poco fiable. Se propone que a la hora de realizar las pruebas con un paciente se obtengan los valores mediante una prueba anterior de más corta duración y llevada a cabo en un entorno controlado para poder obtener los valores de los umbrales correctos de cada persona. De esta forma, se podrán evaluar las siguientes pruebas que se realizan en el entorno del paciente de mejor forma.

Gracias a estas pruebas iniciales también se podrían obtener los parámetros necesarios para poder medir la anchura del paso en desplazamiento en vez de en grado de apertura de las piernas.

- Posicionar la IMU en otras partes del cuerpo

Durante todo este trabajo, el sensor ha sido colocado en el tobillo para la detección de características del paso. Aunque finalmente no se realizó, también se barajó la posibilidad de colocar el sensor en la muñeca para la detección de temblores o movimientos involuntarios. Colocar la IMU en la muñeca o en otras partes del cuerpo podría ser de utilidad para detectar más características que complementen el diagnóstico del paciente.

IV– Pliego de condiciones

A continuación, se exponen las condiciones que se deben cumplir para el desarrollo y correcto funcionamiento de la aplicación.

Pliego de condiciones generales

El objetivo principal de este documento es el realizar un estudio para:

- Comprender el funcionamiento de los sistemas inerciales y las posibles aplicaciones de los mismos en la vida cotidiana
- Comprobar la aplicación de los sistemas inerciales para detección de anomalías en enfermos de Parkinson.

El trabajo se compone de la siguiente documentación:

- Memoria
- Pliego de condiciones
- Presupuesto
- Manual de usuario
- Planos

Pliego de condiciones particulares

- Se debe disponer de un ordenador con procesador Intel i7 y 8GB de memoria RAM para poder utilizar la aplicación y guardar los datos del sensor.
- NGIMU: Dispositivo formado por un conjunto de sensores fabricado por IO Technologies, que incorpora un giróscopo, acelerómetro y magnetómetro triaxial, sensor de presión, humedad, y temperatura.
- El sensor NGIMU debe ser colocado en el lateral del tobillo de la pierna derecha y con la flecha que marca el eje X situada hacia arriba.
- NGIMU GUI: este entorno nos permite transferir del sensor al ordenador los ficheros de las pruebas realizadas.
- Los ficheros obtenidos del sensor tienen que ser guardados en una carpeta y la ruta en la que se encuentra dicha carpeta tiene que ser definida en el código del algoritmo.

- Matlab 2017b: en este entorno se ha implementado el algoritmo y la interfaz de usuario, por lo que es necesario para poder utilizar la aplicación realizada y visualizar los resultados de las pruebas.

V– Presupuesto

En este apartado se detallan las diferentes partidas de gasto necesarias para la realización del proyecto. En él se encuentran los gastos debidos al coste de los materiales, así como los gastos directos e indirectos debido al proyecto. Este presupuesto es orientativo puesto que no se pretende comercializar el producto final.

- **Costes del material utilizado**

En la tabla 15 se muestran los gastos debidos a los equipos utilizados en el proyecto.

Equipo	Precio (euros)	Periodo de amortización (años)	Uso (meses)	Coste de amortización (euros)
NGIMU	370	3	12	123.33
PC Intel i7 8GB RAM	800	3	12	266.67
TOTAL	1170	-	-	390

Tabla 15. Costes del material utilizado.

- **Costes directos del proyecto**

En la tabla 16 se muestran los gastos debidos al trabajo realizado. Se ha dividido el trabajo en diferentes bloques y se ha considerado que el coste del investigador es de 50 €/hora y el del trabajador que ha realizado las pruebas es de 30 €/hora.

Actividad	Tiempo empleado (horas)	Coste (euros)
Documentación	100	5000
Desarrollo del algoritmo	250	12500
Desarrollo de la aplicación	50	2500
Realización de pruebas y verificación	50	1500
TOTAL	450	21500

Tabla 16. Costes directos del proyecto.

- **Costes indirectos del proyecto**

En la tabla 17 se muestran los costes debidos a las licencias de los programas utilizados para el desarrollo del trabajo.

Licencia	Coste (euros)	Periodo de amortización (años)	Uso (meses)	Coste de amortización (euros)
Matlab 2017b	2000	2	12	1000
NGUIMU GUI	0	2	12	0
Microsoft office 365	450	2	12	225
TOTAL	2450	-	-	1225

Tabla 17. Costes indirectos del proyecto.

- **Costes totales del proyecto**

Los costes totales del trabajo se muestran en la tabla 6.

Concepto	Coste íntegro (euros)	Coste de amortización (euros)
Costes del material utilizado	1170	390
Costes directos del proyecto	21500	21500
Costes indirectos del proyecto	2450	1225
TOTAL	25120	23115

Tabla 18. Costes totales del proyecto.

Por lo tanto, el importe total estimado del proyecto presente en este trabajo asciende a la cantidad de:

Veintitrés mil ciento quince euros.

Alcalá de Henares, 6 de diciembre de 2019.

Firmado: Cristina Fernández Baldovinos

Graduada en Ingeniería Electrónica de Comunicaciones.

VI – Manual de usuario

Los resultados de los algoritmos desarrollados en este trabajo se pueden visualizar a través de una interfaz de usuario que se explica a continuación. La aplicación se ejecuta en un ordenador a través del entorno Matlab. Al ejecutar, la primera pantalla que se muestra es la mostrada en la figura 34.

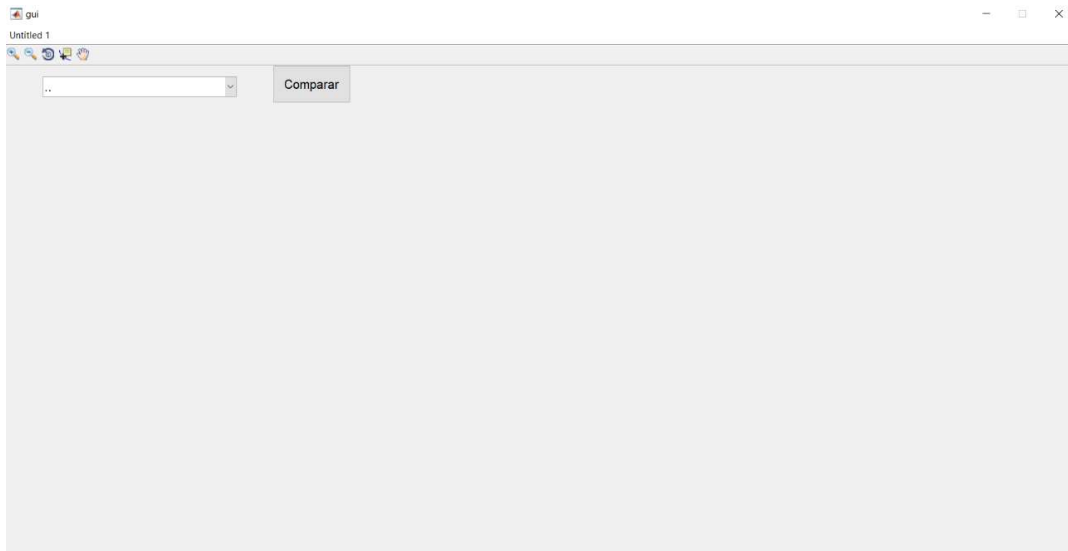


Figura 34. Ventana principal de la aplicación.

En esta primera pantalla se observa un botón que tiene escrito “Comparar”. Al hacer click en ese botón se abre otra pestaña para poder comparar varias pruebas. Más adelante se explicará en detalle esa pantalla. También se observa en esta página de inicio un menú desplegable que muestra todas las pruebas disponibles para visualizar (figura 35).

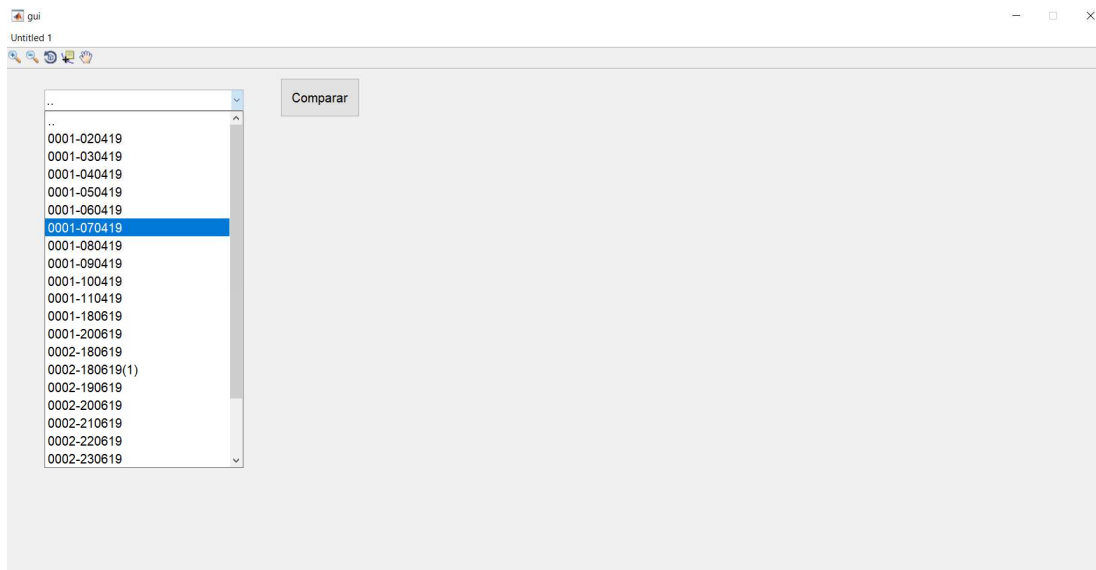


Figura 35. Menú desplegable de la página principal de la aplicación.

Al seleccionar una de las pruebas, tras esperar unos segundos, aparecerá una gráfica que muestra la primera hora de la señal del paso en la mitad superior de la pantalla (Figura 36). Arriba a la derecha de la gráfica se encuentra el número total de horas de la prueba y la hora que se encuentra visualizando en ese momento, 1/15 en el ejemplo mostrado. En la mitad inferior aparecen tres secciones. En la parte central se muestran dos flechas que permiten mostrar las horas siguientes o las horas anteriores de la prueba al presionarlas (“hora anterior” y “hora siguiente”).

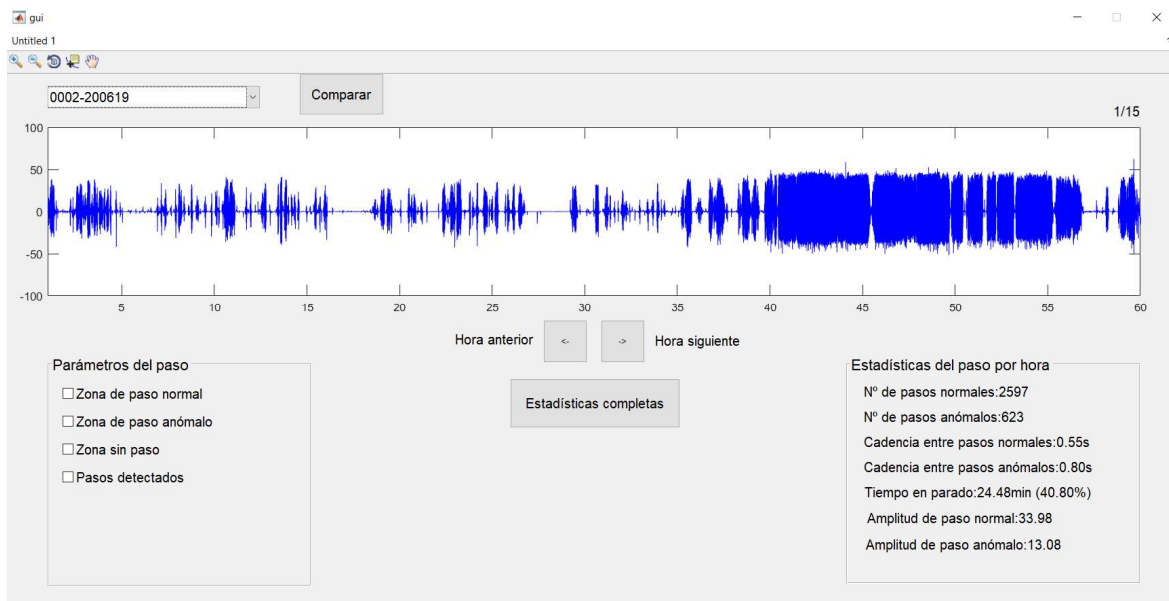


Figura 36. Página principal de la aplicación cuando se ha cargado una prueba.

En el recuadro de la derecha se pueden visualizar resultados generales tales como el número de pasos normales y anómalos, cadencia, tiempo en reposo y amplitud correspondientes a los datos de la hora mostrada en la gráfica. En el recuadro de la izquierda se encuentran cuatro botones que permiten mostrar en la figura la detección de los pasos y las diferentes zonas de paso en diferentes colores. El amarillo muestra las zonas sin paso, el azul las zonas de paso normal y el magenta las zonas de paso anómalo (Figura 37).

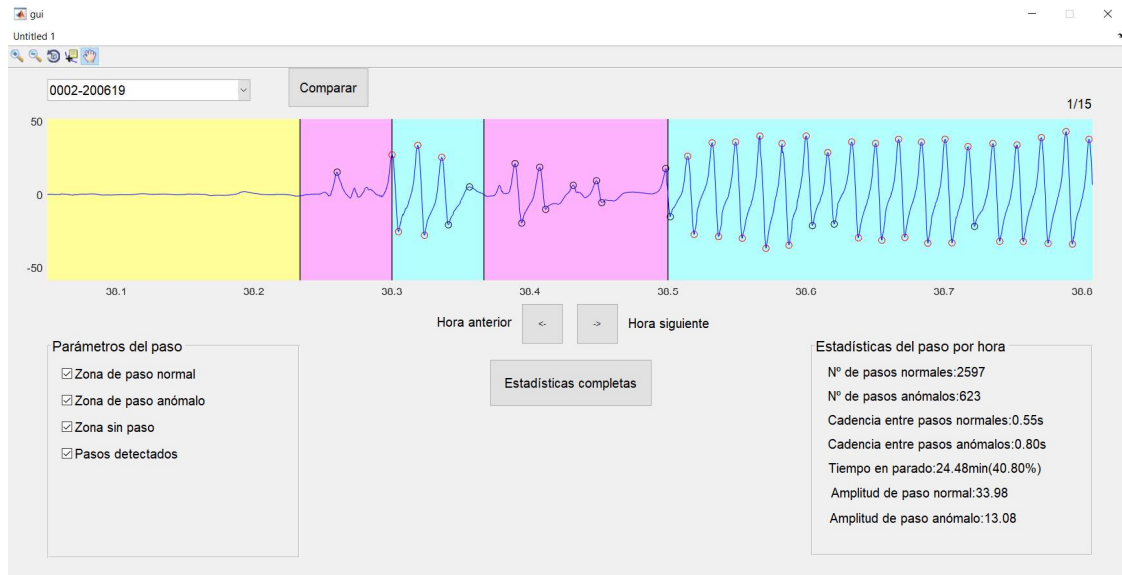


Figura 37. Página principal de la aplicación mostrando los resultados de la detección.

También se puede hacer ampliar una parte de la prueba y moverse por ella con los botones que se encuentran arriba en la barra de herramientas.

Si se hace click en el botón que hay en el centro de la página, que tiene escrito “Estadísticas completas”, se abrirá una nueva ventana que muestra una tabla con todas las estadísticas globales y por horas de la prueba, como se muestra en la figura 38.

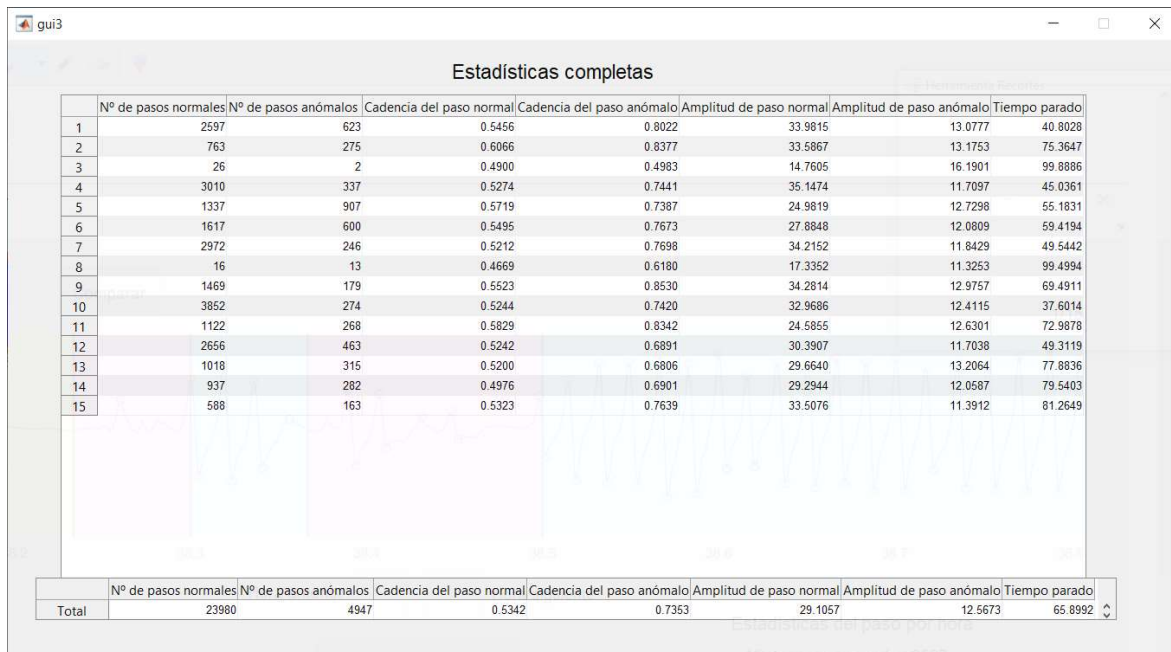


Figura 38. Página de visualización de estadísticas de la prueba.

Desde la pantalla de inicio se puede acceder al botón de “Comparación” y al seleccionarlo se abre una nueva pantalla donde se pueden encontrar tres menús desplegables que permiten comparar hasta tres pruebas diferentes. Una vez elegidas las pruebas a comparar se visualizarán las tres gráficas correspondientes a cada una de las pruebas y además un recuadro con las estadísticas globales de cada una de ellas, tal y como se indica en la figura 39. En esta ventana también es posible realizar zoom o desplazarse por las gráficas utilizando los mismos iconos que aparecen en la parte superior de la barra de herramientas

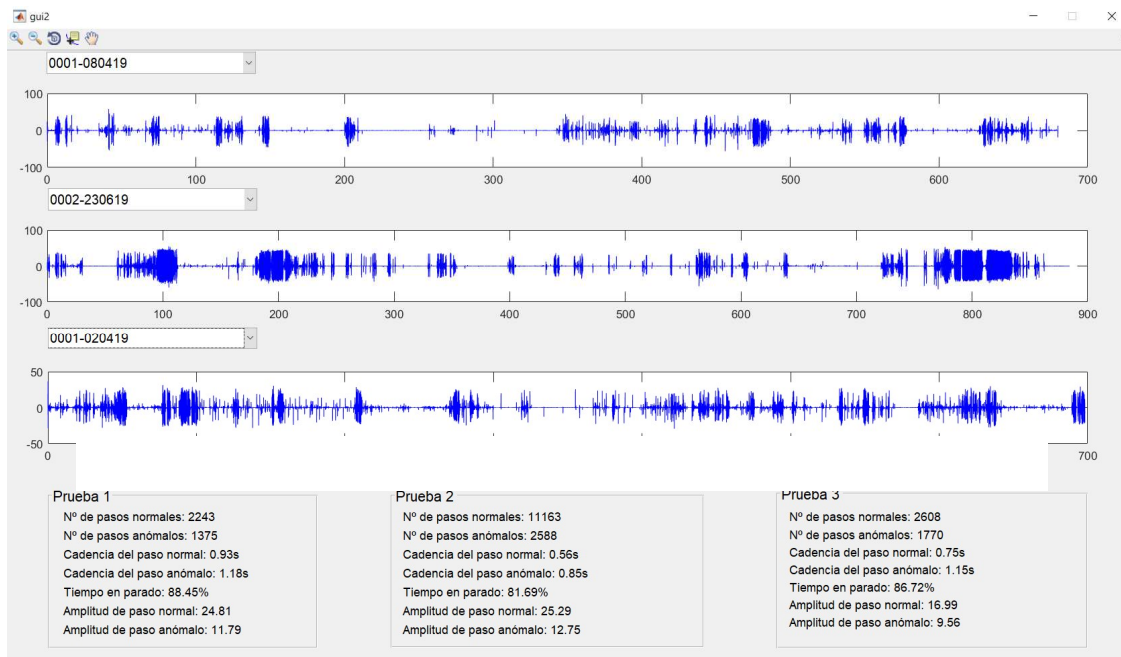


Figura 39. Página de comparación de pruebas.

VII – Planos

En este apartado se muestra el código desarrollado en Matlab para la realización del proyecto.

- **Filtro de Kalman**

Mediante estas dos funciones se obtiene la estimación de los ángulos de Euler a partir de los datos de los giróscopos y acelerómetros del sensor.

```
function [Rolle, Pitche, Yawe]=EKF (Time, AccX, AccY, AccZ, GyrX, GyrY, GyrZ, X, P, Q, R, H)
for i=2:1:length(AccX)
    dt=(Time(i)-Time(i-1));
    % Etapa de predicción: información relativa con el giroscopio.
    sinRoll=sin(X(1)); cosRoll=cos(X(1)); tanRoll=tan(X(1)); secRoll=sec(X(1));
    sinPitch=sin(X(2)); cosPitch=cos(X(2)); tanPitch=tan(X(2)); secPitch=sec(X(2));
    sinYaw=sin(X(3)); cosYaw=cos(X(3)); tanYaw=tan(X(3)); secYaw=sec(X(3));

    X_(1,1)=X(1) + dt*( GyrX(i) + GyrY(i)*sinRoll*tanPitch + GyrZ(i)*cosRoll*tanPitch);
    X_(2,1)=X(2) + dt*( GyrY(i)*cosRoll - GyrZ(i)*sinRoll );
    X_(3,1)=X(3) + dt*( GyrY(i)*sinRoll/cosPitch + GyrZ(i)*cosRoll/cosPitch );

    A=[];

    A(1,1) = GyrY(i)*cosRoll*tanPitch - GyrZ(i)*sinRoll*tanPitch;
    A(1,2) = GyrY(i)*sinRoll*secPitch^2 + GyrZ(i)*cosRoll*secPitch^2;
    A(1,3) = 0;
    A(2,1) = -GyrY(i)*sinRoll - GyrZ(i)*cosRoll;
    A(2,2) = 0;
    A(2,3) = 0;
    A(3,1) = GyrY(i)*cosRoll*secPitch - GyrZ(i)*sinRoll*secPitch;
    A(3,2) = GyrY(i)*sinRoll*secPitch*tanPitch + GyrZ(i)*cosRoll*secPitch*tanPitch;
    A(3,3) = 0;
    A = eye(3) + A*dt;

    for j=1:3
        if X_(j)>=pi
            X_(j)=X_(j)-2*pi;
        end

        if X_(j)<=-pi
            X_(j)=X_(j)+2*pi;
        end
    end

    % Etapa de actualización Roll y Pitch
    Acc=sqrt(AccX(i)^2+AccY(i)^2+AccZ(i)^2);
    if ((abs(Acc-1)<0.01) && (l==1))
        % Vector de observaciones: roll y pitch con los acelerómetros
        Z(1,1)=atan(AccY(i)/AccZ(i));
        Z(2,1)=atan(-AccX(i)/sqrt(AccY(i)^2+AccZ(i)^2));
        % Vector de estimaciones de las observaciones: Estimaciones a priori
        h_X_(1,1)=X_(1);
        h_X_(2,1)=X_(2);
        %Filtro EKF
        [X P]=Filtro_Kalman_Extendido(X_,P,A,H,Z,h_X_,Q,R);
        %Si el sujeto está en movimiento actualizo con mi estado a priori
    else
        X=X_;
    end
    end
    Rolle(i,1)=X(1)*180/pi;
    Pitche(i,1)=X(2)*180/pi;
    Yawe(i,1)=X(3)*180/pi;
end
end
```

```

function [X,P] = Filtro_Kalman_Extendido(X_,P,A,H,Z,h_X_,Q,R)
%FILTRO_KALMAN_EXTENDIDO Summary of this function goes here
% Detailed explanation goes here

% X_: estado a priori
% P: Matriz de covarianza del error
% A: Jacobiano ecuación transición
% H: Jacobiano ecuación de medida
% Z: Observaciones
% h_X_: Estimación de las observaciones a partir del estado a priori
% Q: Matriz covarianza error ecuación de transición
% R: Matriz covarianza error ecuación de medida

% Devuelve el estado actualizado X y la matriz de covarianza del error P

P_=A*P*A' + Q;
S=H*P_*H' + R;
K=P_*H'*(S^(-1));
V=Z-h_X_;
X=X_+K*V;
P=P_- K*S*K';

end

```

- **Detección de pasos**

Las siguientes dos funciones son las utilizadas para la detección del paso y para su clasificación en paso normal o paso anómalo.

```

function [ARRAY,Pitche_procesado]=Deteccion_pasos(Pitche,num)

up=0;
down=0;
semaforo_inicial=0;
zero=0;
maximo_procesado=0;
minimo_procesado=0;
contador=0;

for i=2:1:num
    if i>=101
        promediado(i,1)=mean(Pitche(i-100:i));
        Pitche_procesado(i,1)=Pitche(i,1)-promediado(i,1);

        if Pitche_procesado(i-1,1)<=0 && Pitche_procesado(i,1)>0 %cruce por cero,
direccion ascendente
            if semaforo_inicial>=1
                zero=zero+1;
            else
                semaforo_inicial=1;
            end
            up=1;
            down=0;
        end

        if Pitche_procesado(i-1,1)>=0 && Pitche_procesado(i,1)<0 %cruce por cero,
direccion descendente
            if semaforo_inicial>=1
                zero=zero+1;
            else
                semaforo_inicial=1;
            end
            down=1;
            up=0;
        end

        if up==1 && Pitche_procesado(i,1)>maximo_procesado
            maximo_procesado= Pitche_procesado(i,1);
            posmax_procesado=i;
        end
    end
end

```

```

if down==1 && Pitche_procesado(i,1)<minimo_procesado
    minimo_procesado= Pitche_procesado(i,1);
    posmin_procesado=i;
end
amplitud=maximo_procesado-minimo_procesado;

if zero==2 %Dos cruce por cero
    if up==1
        contador=contador+1;
        ARRAY(contador,1)=maximo_procesado;
        ARRAY(contador,2)=posmax_procesado;
        contador=contador+1;
        ARRAY(contador,1)=minimo_procesado;
        ARRAY(contador,2)=posmin_procesado;
    elseif down==1
        contador=contador+1;
        ARRAY(contador,1)=minimo_procesado;
        ARRAY(contador,2)=posmin_procesado;
        contador=contador+1;
        ARRAY(contador,1)=maximo_procesado;
        ARRAY(contador,2)=posmax_procesado;
    end
    zero=0;
    maximo_procesado=0;
    minimo_procesado=0;
end
end
end
end

```

```

function [ARRAY]=Clasificador_pasos(ARRAY)
for i=1:length(ARRAY)
    if ARRAY(i,1)>25 || ARRAY(i,1)<-25
        ARRAY(i,3)=1;
    elseif ARRAY(i,1)>5 || ARRAY(i,1)<-5
        ARRAY(i,3)=2;
    else
        ARRAY(i,3)=0;
    end
end
quitar=find(ARRAY(:,3)==0);
ARRAY(quitar,:)=[];

for j=1:5
    normal=find(ARRAY(:,3)==1);
    arrastra=find(ARRAY(:,3)==2);
    if isempty(normal)==0
        media_normal=mean(abs(ARRAY(normal,1)));
    else
        media_normal=30;
    end
    if isempty(arrastra)==0
        media_arrastra=mean(abs(ARRAY(arrastra,1)));
    else
        media_arrastra=15;
    end
    for k=1:length(ARRAY)
        [~,A]=min([abs(abs(ARRAY(k,1))-media_normal),abs(abs(ARRAY(k,1))-
media_arrastra)]);
        if A==1
            ARRAY(k,3)=1;
        elseif A==2
            ARRAY(k,3)=2;
        end
    end
end
end
end

```

- **Análisis espectral**

Las tres funciones mostradas a continuación son las utilizadas para realizar el análisis espectral y para decidir las diferentes zonas de paso.

```
function [Potencia,tam]=Analisis_frecuencia(Pitche_procesado)

[~,frec,~,Pot]=spectrogram(Pitche_procesado,rectwin(200),0,200,100,'yaxis');
tam=size(Pot);

[P,F]=pspectrum(Pitche_procesado,100);
[~,pos]=max(P);
frecuencia=F(pos);
[~,indice]=min([abs(frecuencia-frec(:,1))]);

Potencia=10*log10(Pot(indice,:));
FOG=find(Potencia<10);
Potencia(2,FOG)=1;
FOG2=find(Potencia>=10 & Potencia<25);
Potencia(2,FOG2)=2;
media_FOG2=mean(Potencia(FOG2));
FOG3=find(Potencia>=25);
media_FOG3=mean(Potencia(FOG3));
Potencia(2,FOG3)=3;
for i=1:5
    for j=1:tam(2)
        [~,k]=min([abs(Potencia(1,j)-media_FOG2),abs(Potencia(1,j)-media_FOG3)]);
        if k==2
            Potencia(2,j)=3;
        elseif Potencia(2,j)~=1
            Potencia(2,j)=2;
        end
    end
    media_FOG2=mean(Potencia(1,find(Potencia(1,:)>=10 & Potencia(1,*)<25)));
    media_FOG3=mean(Potencia(1,find(Potencia(1,*)>=25)));
end
end
```

```
function [FOG_p,FOG_p2,FOG_p3]=ordenar_FOG(Potencia,num,tam)
contador1=0;
contador2=0;
contador3=0;

for l=1:tam(2)
    if l==1
        if Potencia(2,l)==1
            contador1=contador1+1;
            FOG_p(contador1)=1;
        elseif Potencia(2,l)==2
            contador2=contador2+1;
            FOG_p2(contador2)=1;
        elseif Potencia(2,l)==3
            contador3=contador3+1;
            FOG_p3(contador3)=1;
        end
    else
        if Potencia(2,l)~=Potencia(2,l-1)
            if Potencia(2,l-1)==1
                contador1=contador1+1;
                FOG_p(contador1)=(l-1)*200;
            elseif Potencia(2,l-1)==2
                contador2=contador2+1;
                FOG_p2(contador2)=(l-1)*200;
            elseif Potencia(2,l-1)==3
                contador3=contador3+1;
                FOG_p3(contador3)=(l-1)*200;
            end
        end
        if Potencia(2,l)==1
```

```

        contador1=contador1+1;
        FOG_p(contador1)=( (1-1)*200)+1;
    elseif Potencia(2,l)==2
        contador2=contador2+1;
        FOG_p2(contador2)=( (1-1)*200)+1;
    elseif Potencia(2,l)==3
        contador3=contador3+1;
        FOG_p3(contador3)=( (1-1)*200)+1;
    end
end
end
if l==tam(2)
    if Potencia(2,l)==1
        contador1=contador1+1;
        FOG_p(contador1)=num;
    elseif Potencia(2,l)==2
        contador2=contador2+1;
        FOG_p2(contador2)=num;
    elseif Potencia(2,l)==3
        contador3=contador3+1;
        FOG_p3(contador3)=num;
    end
end
end
end
if contador3==0
    FOG_p3=[];
end
end
end

```

```

function
[ARRAY,FOG_p_copia,FOG_p2_copia,FOG_p3_copia]=Decisor(ARRAY,FOG_p,FOG_p2,FOG_p3)
FOG_p3_copia=FOG_p3;
if isempty(FOG_p3)==0
    for i=2:2:length(FOG_p3)
        v_normal=find(ARRAY(:,2)<FOG_p3(i) & ARRAY(:,2)>FOG_p3(i-1));
        p_n3=find(ARRAY(v_normal,3)==1);
        p_a3=find(ARRAY(v_normal,3)==2);
        if FOG_p3(i)-FOG_p3(i-1)<200
            if isempty(v_normal)==1
                u=find(FOG_p<=FOG_p3(i-1));
                if isempty(u)==1 && FOG_p3(i)~=FOG_p(1)
                    FOG_p=[FOG_p3(i-1:i),FOG_p];
                elseif isempty(u)==1 && FOG_p3(i)==FOG_p(1)
                    FOG_p=[FOG_p3(i-1),FOG_p(2:end)];
                elseif FOG_p(u(end))==FOG_p3(i-1)-1 && u(end)==length(FOG_p)
                    FOG_p=[FOG_p(1:u(end)-1),FOG_p3(i)];
                elseif FOG_p(u(end))~=FOG_p3(i-1)-1 && u(end)==length(FOG_p)
                    FOG_p=[FOG_p,FOG_p3(i-1:i)];
                elseif FOG_p(u(end))==FOG_p3(i-1)-1 && FOG_p(u(end)+1)==FOG_p3(i)+1
                    FOG_p(u(end):u(end)+1)=[];
                elseif FOG_p(u(end))==FOG_p3(i-1)-1 && FOG_p(u(end)+1)~=FOG_p3(i)+1
                    FOG_p=[FOG_p(1:u(end)-1),FOG_p3(i),FOG_p(u(end)+1:end)];
                elseif FOG_p(u(end))~=FOG_p3(i-1)-1 && FOG_p(u(end)+1)==FOG_p3(i)+1
                    FOG_p=[FOG_p(1:u(end)),FOG_p3(i-1),FOG_p(u(end)+2:end)];
                elseif FOG_p(u(end))~=FOG_p3(i-1)-1 && FOG_p(u(end)+1)~=FOG_p3(i)+1
                    FOG_p=[FOG_p(1:u(end)),FOG_p3(i-1),FOG_p3(i),FOG_p(u(end)+1:end)];
            end
            a=find(FOG_p3_copia==FOG_p3(i) | FOG_p3_copia==FOG_p3(i-1));
            FOG_p3_copia(a)=[];
            u=[];
        else
            if length(p_n3)<length(p_a3)
                v=find(FOG_p2<=FOG_p3(i-1));
                if isempty(v)==1 && FOG_p3(i)~=FOG_p2(1)
                    FOG_p2=[FOG_p3(i-1:i),FOG_p2];
                elseif isempty(v)==1 && FOG_p3(i)==FOG_p2(1)
                    FOG_p2=[FOG_p3(i-1),FOG_p2(2:end)];
                elseif FOG_p2(v(end))==FOG_p3(i-1)-1 && v(end)==length(FOG_p2)
                    FOG_p2=[FOG_p2(1:v(end)-1),FOG_p3(i)];
                elseif FOG_p2(v(end))~=FOG_p3(i-1)-1 && v(end)==length(FOG_p2)
                    FOG_p2=[FOG_p2,FOG_p3(i-1:i)];
                elseif FOG_p2(v(end))==FOG_p3(i-1)-1 &&
                    FOG_p2(v(end)+1)==FOG_p3(i)+1
                    FOG_p2(v(end):v(end)+1)=[];
            end
        end
    end
end

```

```

elseif FOG_p2(v(end))==FOG_p3(i-1)-1 && FOG_p2(v(end)+1)~=FOG_p3(i)+1
    FOG_p2=[FOG_p2(1:v(end)-1),FOG_p3(i),FOG_p2(v(end)+1:end)];
elseif FOG_p2(v(end))~=FOG_p3(i-1)-1 && FOG_p2(v(end)+1)==FOG_p3(i)+1
    FOG_p2=[FOG_p2(1:v(end)),FOG_p3(i-1),FOG_p2(v(end)+2:end)];
elseif FOG_p2(v(end))~=FOG_p3(i-1)-1 && FOG_p2(v(end)+1)~=FOG_p3(i)+1
    FOG_p2=[FOG_p2(1:v(end)),FOG_p3(i-1),FOG_p3(i),FOG_p2(v(end)+1:end)];
end
    d=find(FOG_p3_copia==FOG_p3(i) | FOG_p3_copia==FOG_p3(i-1));
    FOG_p3_copia(d)=[];
    v=[];
end
end
end
end
end
end
FOG_p2_copia=FOG_p2;
if isempty(FOG_p2)==0
    for j=2:2:length(FOG_p2)
        v_arrastra=find(ARRAY(:,2)<FOG_p2(j) & ARRAY(:,2)>FOG_p2(j-1));
        p_n2=find(ARRAY(v_arrastra,3)==1);
        p_a2=find(ARRAY(v_arrastra,3)==2);
        if FOG_p2(j)-FOG_p2(j-1)<200
            if isempty(v_arrastra)==1
                w=find(FOG_p<=FOG_p2(j-1));
                if isempty(w)==1 && FOG_p2(j)~=FOG_p(1)
                    FOG_p=[FOG_p2(j-1:j),FOG_p];
                elseif isempty(w)==1 && FOG_p2(j)==FOG_p(1)
                    FOG_p=[FOG_p2(j-1),FOG_p(2:end)];
                elseif FOG_p(w(end))==FOG_p2(j-1)-1 && w(end)==length(FOG_p)
                    FOG_p=[FOG_p(1:w(end)-1),FOG_p2(j)];
                elseif FOG_p(w(end))~=FOG_p2(j-1)-1 && w(end)==length(FOG_p)
                    FOG_p=[FOG_p,FOG_p2(j-1:j)];
                elseif FOG_p(w(end))==FOG_p2(j-1)-1 && FOG_p(w(end)+1)==FOG_p2(j)+1
                    FOG_p(w(end):w(end)+1)=[];
                elseif FOG_p(w(end))==FOG_p2(j-1)-1 && FOG_p(w(end)+1)~=FOG_p2(j)+1
                    FOG_p=[FOG_p(1:w(end)-1),FOG_p2(j),FOG_p(w(end)+1:end)];
                elseif FOG_p(w(end))~=FOG_p2(j-1)-1 && FOG_p(w(end)+1)==FOG_p2(j)+1
                    FOG_p=[FOG_p(1:w(end)),FOG_p2(j-1),FOG_p(w(end)+2:end)];
                elseif FOG_p(w(end))~=FOG_p2(j-1)-1 && FOG_p(w(end)+1)~=FOG_p2(j)+1
                    FOG_p=[FOG_p(1:w(end)),FOG_p2(j-1),FOG_p2(j),FOG_p(w(end)+1:end)];
                end
                c=find(FOG_p2_copia==FOG_p2(j) | FOG_p2_copia==FOG_p2(j-1));
                FOG_p2_copia(c)=[];
                w=[];
            else
                if length(p_n2)>length(p_a2)
                    x=find(FOG_p3_copia<=FOG_p2(j-1));
                    if isempty(x)==1 && FOG_p2(j)~=FOG_p3_copia(1)
                        FOG_p3_copia=[FOG_p2(j-1:j),FOG_p3_copia];
                    elseif isempty(x)==1 && FOG_p2(j)==FOG_p3_copia(1)
                        FOG_p3_copia=[FOG_p2(j-1),FOG_p3_copia(2:end)];
                    elseif FOG_p3_copia(x(end))==FOG_p2(j-1)-1 && x(end)==length(FOG_p3_copia)
                        FOG_p3_copia=[FOG_p3_copia(1:x(end)-1),FOG_p2(j)];
                    elseif FOG_p3_copia(x(end))~=FOG_p2(j-1)-1 && x(end)==length(FOG_p3_copia)
                        FOG_p3_copia=[FOG_p3_copia,FOG_p2(j-1:j)];
                    elseif FOG_p3_copia(x(end))==FOG_p2(j-1)-1 && FOG_p3_copia(x(end)+1)==FOG_p2(j)+1
                        FOG_p3_copia(x(end):x(end)+1)=[];
                    elseif FOG_p3_copia(x(end))==FOG_p2(j-1)-1 && FOG_p3_copia(x(end)+1)~=FOG_p2(j)+1
                        FOG_p3_copia=[FOG_p3_copia(1:x(end)-1),FOG_p2(j),FOG_p3_copia(x(end)+1:end)];
                    elseif FOG_p3_copia(x(end))~=FOG_p2(j-1)-1 && FOG_p3_copia(x(end)+1)==FOG_p2(j)+1
                        FOG_p3_copia=[FOG_p3_copia(1:x(end)),FOG_p2(j-1),FOG_p3_copia(x(end)+2:end)];
                    elseif FOG_p3_copia(x(end))~=FOG_p2(j-1)-1 && FOG_p3_copia(x(end)+1)~=FOG_p2(j)+1
                        FOG_p3_copia=[FOG_p3_copia(1:x(end)),FOG_p2(j-1),FOG_p2(j),FOG_p3_copia(x(end)+1:end)];
                    end
                    d=find(FOG_p2_copia==FOG_p2(j) | FOG_p2_copia==FOG_p2(j-1));
                    FOG_p2_copia(d)=[];
                    x=[];
                end
            end
        end
    end
end
end
end
FOG_p_copia=FOG_p;
if isempty(FOG_p)==0
    for k=2:2:length(FOG_p)

```



```

v_fog=find(ARRAY(:,2)<FOG_p(k) & ARRAY(:,2)>FOG_p(k-1));
p_n1=find(ARRAY(v_fog,3)==1);
p_a1=find(ARRAY(v_fog,3)==2);
if FOG_p(k)-FOG_p(k-1)<200
    if isempty(v_fog)==0
        if length(p_n1)>length(p_a1)
            z=find(FOG_p3_copia<=FOG_p(k-1));
            if isempty(z)==1 && FOG_p(k)~=FOG_p3_copia(1)
                FOG_p3_copia=[FOG_p(k-1:k),FOG_p3_copia];
            elseif isempty(z)==1 && FOG_p(k)==FOG_p3_copia(1)
                FOG_p3_copia=[FOG_p(k-1),FOG_p3_copia(2:end)];
            elseif FOG_p3_copia(z(end))==FOG_p(k-1)-1 && z(end)==length(FOG_p3_copia)
                FOG_p3_copia=[FOG_p3_copia(1:z(end)-1),FOG_p(k)];
            elseif FOG_p3_copia(z(end))~=FOG_p(k-1)-1 && z(end)==length(FOG_p3_copia)
                FOG_p3_copia=[FOG_p3_copia,FOG_p(k-1:k)];
            elseif FOG_p3_copia(z(end))==FOG_p(k-1)-1 && FOG_p3_copia(z(end)+1)==FOG_p(k)+1
                FOG_p3_copia(z(end):z(end)+1)=[];
            elseif FOG_p3_copia(z(end))==FOG_p(k-1)-1 && FOG_p3_copia(z(end)+1)~=FOG_p(k)+1
                FOG_p3_copia=[FOG_p3_copia(1:z(end)-1),FOG_p(k),FOG_p3_copia(z(end)+1:end)];
            elseif FOG_p3_copia(z(end))~=FOG_p(k-1)-1 && FOG_p3_copia(z(end)+1)==FOG_p(k)+1
                FOG_p3_copia=[FOG_p3_copia(1:z(end)),FOG_p(k-1),FOG_p3_copia(z(end)+2:end)];
            elseif FOG_p3_copia(z(end))~=FOG_p(k-1)-1 && FOG_p3_copia(z(end)+1)~=FOG_p(k)+1
                FOG_p3_copia=[FOG_p3_copia(1:z(end)),FOG_p(k-1),FOG_p(k),FOG_p3_copia(z(end)+1:end)];
            end
            f=find(FOG_p_copia==FOG_p(k) | FOG_p_copia==FOG_p(k-1));
            FOG_p_copia(f)=[];
            z=[];
        elseif length(p_a1)>=length(p_n1)
            zz=find(FOG_p2_copia<=FOG_p(k-1));
            if isempty(zz)==1 && FOG_p(k)~=FOG_p2_copia(1)
                FOG_p2_copia=[FOG_p(k-1:k),FOG_p2_copia];
            elseif isempty(zz)==1 && FOG_p(k)==FOG_p2_copia(1)
                FOG_p2_copia=[FOG_p(k-1),FOG_p2_copia(2:end)];
            elseif FOG_p2_copia(zz(end))==FOG_p(k-1)-1 && zz(end)==length(FOG_p2_copia)
                FOG_p2_copia=[FOG_p2_copia(1:zz(end)-1),FOG_p(k)];
            elseif FOG_p2_copia(zz(end))~=FOG_p(k-1)-1 && zz(end)==length(FOG_p2_copia)
                FOG_p2_copia=[FOG_p2_copia,FOG_p(k-1:k)];
            elseif FOG_p2_copia(zz(end))==FOG_p(k-1)-1 && FOG_p2_copia(zz(end)+1)==FOG_p(k)+1
                FOG_p2_copia(zz(end):zz(end)+1)=[];
            elseif FOG_p2_copia(zz(end))==FOG_p(k-1)-1 && FOG_p2_copia(zz(end)+1)~=FOG_p(k)+1
                FOG_p2_copia=[FOG_p2_copia(1:zz(end)-1),FOG_p(k),FOG_p2_copia(zz(end)+1:end)];
            elseif FOG_p2_copia(zz(end))~=FOG_p(k-1)-1 && FOG_p2_copia(zz(end)+1)==FOG_p(k)+1
                FOG_p2_copia=[FOG_p2_copia(1:zz(end)),FOG_p(k-1),FOG_p2_copia(zz(end)+2:end)];
            elseif FOG_p2_copia(zz(end))~=FOG_p(k-1)-1 && FOG_p2_copia(zz(end)+1)~=FOG_p(k)+1
                FOG_p2_copia=[FOG_p2_copia(1:zz(end)),FOG_p(k-1),FOG_p(k),FOG_p2_copia(zz(end)+1:end)];
            end
            g=find(FOG_p_copia==FOG_p(k) | FOG_p_copia==FOG_p(k-1));
            FOG_p_copia(g)=[];
            zz=[];
        end
    end
else
    quitar=find(ARRAY(:,2)<FOG_p(k) & ARRAY(:,2)>FOG_p(k-1));
    if quitar~=0
        ARRAY(quitar,:)=[];
    end
    quitar=0;
end
end
end
end

```

- **Estadísticas**

Esta última función es la encargada de obtener las características adicionales del paso.

```

function
[num_pasos_n,num_pasos_a,cadencia_n,cadencia_a,tiempo_parado,amplitud_n,amplitud_a]=esta
disticas (ARRAY, FOG_p, FOG_p2, FOG_p3, num_horas)
hora=1;
num_pasos_n=0;
num_pasos_a=0;
tiempo_parado(1:num_horas)=0;
cadencia_n(1:num_horas)=0;
num_cad_n(1:num_horas)=0;
cadencia_a(1:num_horas)=0;
num_cad_a(1:num_horas)=0;
amplitud_n=0;
amplitud_a=0;

for i=2:2:length(FOG_p3)
pasos_n=find(ARRAY(:,2)<FOG_p3(i) & ARRAY(:,2)>FOG_p3(i-1));
if length(pasos_n)>1
for l=2:length(pasos_n)
cadencia_n(hora)=cadencia_n(hora)+(ARRAY(pasos_n(l),2)-ARRAY(pasos_n(l-1),2));
num_cad_n(hora)=num_cad_n(hora)+1;
end
end
if FOG_p3(i)>360000*hora && FOG_p3(i)<=360000*hora
a=find(ARRAY(pasos_n,2)<=360000*hora);
num_pasos_n(hora)=num_pasos_n(hora)+length(a);
amplitud_n(hora)=amplitud_n(hora)+sum(abs(ARRAY(a,1)));
num_pasos_n(hora+1)=length(pasos_n)-length(a);
amplitud_n(hora+1)=sum(abs(ARRAY(pasos_n(a(end)+1):end),1));
hora=hora+1;
elseif FOG_p3(i)>360000*hora && FOG_p3(i)>360000*hora
hora=hora+1;
num_pasos_n(hora)=length(pasos_n);
amplitud_n(hora)=sum(abs(ARRAY(pasos_n,1)));
else
num_pasos_n(hora)=num_pasos_n(hora)+length(pasos_n);
amplitud_n(hora)=amplitud_n(hora)+sum(abs(ARRAY(pasos_n,1)));
end
end
hora=1;
for y=1:length(amplitud_n)
if amplitud_n==0
amplitud_n(y)=0;
else
amplitud_n(y)=amplitud_n(y)/num_pasos_n(y);
end
end
for m=1:length(cadencia_n)
if cadencia_n(m)==0
cadencia_n(m)=0;
else
cadencia_n(m)=cadencia_n(m)/(100*num_cad_n(m));
end
end
end

for j=2:2:length(FOG_p2)
pasos_a=find(ARRAY(:,2)<FOG_p2(j) & ARRAY(:,2)>FOG_p2(j-1));
if length(pasos_a)>1
for n=2:length(pasos_a)
cadencia_a(hora)=cadencia_a(hora)+(ARRAY(pasos_a(n),2)-ARRAY(pasos_a(n-1),2));
num_cad_a(hora)=num_cad_a(hora)+1;
end
end
if FOG_p2(j)>360000*hora && FOG_p2(j)<=360000*hora
b=find(ARRAY(pasos_a,2)<=360000*hora);
num_pasos_a(hora)=num_pasos_a(hora)+length(b);
amplitud_a(hora)=amplitud_a(hora)+sum(abs(ARRAY(b,1)));
num_pasos_a(hora+1)=length(pasos_a)-length(b);
amplitud_a(hora+1)=sum(abs(ARRAY(pasos_a((b(end)+1):end),1)));
hora=hora+1;
elseif FOG_p2(j)>360000*hora && FOG_p2(j)>360000*hora
hora=hora+1;
num_pasos_a(hora)=length(pasos_a);
amplitud_a(hora)=sum(abs(ARRAY(pasos_a,1)));
else
num_pasos_a(hora)=num_pasos_a(hora)+length(pasos_a);
amplitud_a(hora)=amplitud_a(hora)+sum(abs(ARRAY(pasos_a,1)));
end
end

```

```

end
hora=1;
for o=1:length(cadencia_a)
    if cadencia_a(o)==0
        cadencia_a(o)=0;
    else
        cadencia_a(o)=cadencia_a(o)/(100*num_cad_a(o));
    end
end
for x=1:length(amplitud_a)
    if amplitud_a==0
        amplitud_a(y)=0;
    else
        amplitud_a(x)=amplitud_a(x)/num_pasos_a(x);
    end
end

for k=2:2:length(FOG_p)
    if FOG_p(k)>360000*hora && FOG_p(k-1)>360000*hora
        hora=hora+1;
        tiempo_parado(hora)=tiempo_parado(hora)+(FOG_p(k)-FOG_p(k-1));
    elseif FOG_p(k)>360000*hora
        tiempo_parado(hora)=tiempo_parado(hora)+(360000*hora-FOG_p(k-1));
        tiempo_parado(hora+1)=FOG_p(k)-360000*hora;
        hora=hora+1;
    else
        tiempo_parado(hora)=tiempo_parado(hora)+(FOG_p(k)-FOG_p(k-1));
    end
end
tiempo_parado=tiempo_parado/100;
end

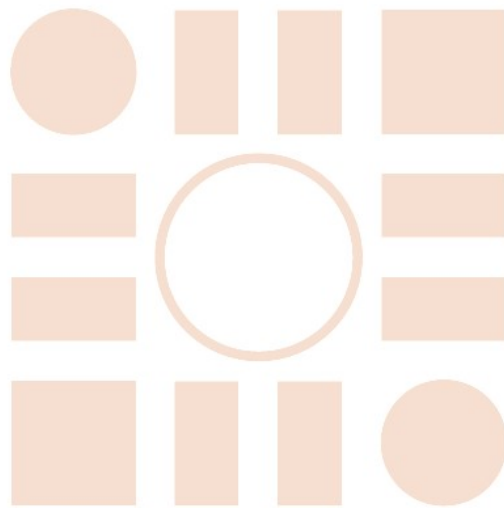
```


VIII – Bibliografía

- [1] C. Ramaker, J. Marinus, A. M. Stiggelbou, y B. J. van Hilten, “Systematic evaluation of rating scales for impairment and disability in Parkinson’s Disease”, *Movement disorders*, vol. 17, n° 5, pp. 867-876, 2002.
- [2] A. Rodríguez-Molinero, et al., “Analysis of Correlation between an Accelerometer-Based Algorithm for Detecting Parkinsonian Gait and UPDRS Subscales”, *Frontiers in neurology*, vol. 8, n° 431, 2017.
- [3] G. Cai et al., “Quantitative assessment of parkinsonian tremor based on a linear acceleration extraction algorithm”, *Biomedical Signal Processing and Control*, vol. 42, pp. 53-62, 2018.
- [4] J. Barth et al., “Biometric and Mobile Gait Analysis for Early Diagnosis and Therapy Monitoring in Parkinson's Disease”, en *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 868-871, 2011.
- [5] S. Okuda et al., “Gait analysis of patients with Parkinson's disease using a portable triaxial accelerometer”, *Neurology and Clinical Neuroscience*, vol. 4, n° 3, pp. 93-97, 2016.
- [6] M. H. Pham et al., “Validation of a Step Detection Algorithm during Straight Walking and Turning in Patients with Parkinson’s Disease and Older Adults Using an Inertial Measurement Unit at the Lower Back”, *Frontiers in Neurology*, vol. 8, n° 457, 2017.
- [7] A. Suppa, et al., “L-Dopa and Freezing of Gait in Parkinson’s Disease: Objective Assessment through a Wearable Wireless System”, *Frontiers in neurology*, vol. 8, n° 406, 2017.
- [8] Q. K. Dang, H. G. Seo, D. D. Pham, y Y. Chee, “Wearable Sensor Based Stooped Posture Estimation in Simulated Parkinson’s Disease Gaits”, *Sensors*, vol. 19, n° 2, pp. 223, 2019.
- [9] A. Pantall, S. Del Din, y L. Rochester, “Longitudinal changes over thirty-six months in postural control dynamics and cognitive function in people with Parkinson's disease”, *Gait & posture*, vol. 62, pp. 468-474, 2018.

- [10] A. Rodríguez-Molinero, et al., “A Kinematic Sensor and Algorithm to Detect Motor Fluctuations in Parkinson Disease: Validation Study Under Real Conditions of Use” JMIR rehabilitation and assistive technologies, vol. 5, nº 1, 2018.
- [11] “NGIMU”, 2019. [En línea]. Disponible en: “<https://x-io.co.uk/ngimu/>”. [Accedido: 12-dic-2019]
- [12] R. Cervigón Rey, “Implementación de un EKF sobre plataformas Android para posicionamiento de dispositivos portables en espacios interiores a partir de medidas de sensores inerciales”, trabajo de fin de grado, Universidad de Alcalá, 2017.
- [13] E. Muñoz Díaz, “Inertial Pocket Navigation System for Pedestrians”, tesis doctoral, Universidad de Alcalá, 2016.

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá

